The Games Factory

Manual Update February 6th 2000

The following document is the original Europress© "The Games Factory" manual that shipped with the CD-ROM version of the program. The wonderful people at Europress have allowed us to distribute this document.

Thank you Europress!

A couple notes before you read the manual.

1. Please do not ask Europress with questions involving "The Games Factory"
2. Clickteam has taken responsibility for supporting all past, current and future users. You can reach our support forums at www.clickteam.com
3. The installation instructions in the manual refer to the CD-ROM version of TGF. The download version is very easy to install. Please email jvance@clickteam.com if you have any problems installing the program.

Changes/additions:

1. "Selling your game" page #297
   You will need to purchase a copy of "The Games Factory Pro" from Clickteam and agree to the terms of the "TGF PRO" runtime agreement.
   Note: "TGF PRO" will be released soon. Check Clickteam.com for information on its availability.

2. "Save as internet game"
   This is a feature introduced after the TGF manual was produced. Save as internet game produces a "Vitalized" file. This Vitalized file can be embedded into a web page and accessed by modern versions of the major web browsers. For more information visit the Clickteam web site.

3. The updated version of "The Games Factory" available from Clickteam allows the user to add extension to the runtime. More information about adding, using or creating extensions can be found at the Clickteam.

Happy Creating,

Francois, Yves, and the rest of Clickteam

Contents

Chapter

The Games Factory™ was programmed and designed by:

François Lionet & Yves Lamoureux.

Construction of The Games Factory™ was greatly assisted by the following people:

Extension Objects created by: Philippe Paquet and Yves Lamoureux.

Graphic artwork created by: Nolan Worthington and Etrange Libellules.

The Games Factory was managed by: Richard Vanner.

The Games Factory manual was written by: Ian Young, assisted by Philip Chapman, Richard Vanner and Lee Bamber.

The example games were created by: Lee Bamber, Nolan Worthington and Merlyn Lear.

The tutorial games were written by: Lee Bamber.

Help files written by: Stephen Hill.

Box design: Julian Marshall and Nolan Worthington.

Box artwork by: Bill Worthington.

Sound & Music by: Mel Croucher, Ian Waugh and Darren Ithell.

Manual Designed and Typeset by: Mark Nolan

Welcome to The Games Factory!

The Games Factory is the very latest easy to use games creator from Europress software. It represents state-of-the-art animated graphics and sound, multimedia functions and fabulous game-structuring routines that makes it very quick and easy to produce your own games of stunning quality with no programming experience whatsoever! Whether its platform games, maze games, shoot-'em-ups or graphic adventures, The Games Factory makes it easy! You don't even have to make the characters for your games, The Games Factory has huge graphics libraries containg literally thousands of different characters and backdrops ready made for your games.

For those who really want to be creative, though, you can make your own animated characters, backdrops and even record your own sound effects, making a game completely unique.

This manual will take you step-by-step through five tutorial games, from the most basic games to a fully fledged beat-'em-up with amazing graphics. By the time you finish creating the tutorial games you will be fully competent in all the major features of The Games Factory, fully skilled to create virtually any game you desire.

Should you be an experienced user of Klik & Play already, and you want to get stuck straight into the game creation, the rest of the manual will detail all the functions of all the objects, including a host of new options that will make games creation easier than ever before.

Don't forget that you also have the extensive Help files on hand at any time for handy hints and descriptions.

Using This Manual

To get the best out of this manual, it is heavily recommended that you read the following overviews of the functions of the different Editor screens, and read the introduction pages of each of the chapters.

Most of this manual describes the parameters for each different type of object. It is important to understand that everything that you can do within The Games Factory is related to an object. You have Player objects for Lives and Scores, CD Objects for playing music from a CD player, Sound objects for playing music and samples.

Each one of these objects can be changed in several ways, almost always done by selecting on an object with the right mouse button.

In the Level Editor, you get all the initial options for the objects. These are described in great detail in the ‘Objects’ chapter.

In the Event Editor, you choose all the Events and Actions that can happen to an object, such as ‘Play a sample’, ‘Destroy’, ‘Collides with...’, ‘Lives =0’, etc, etc.

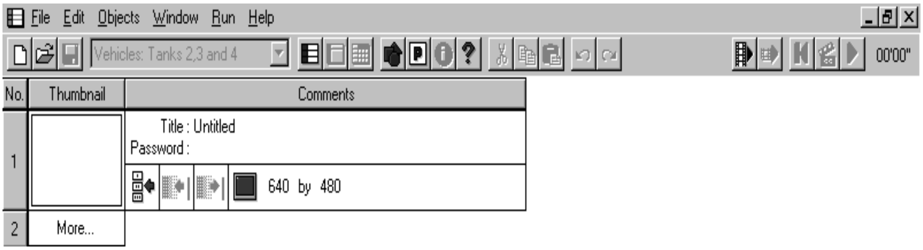
Each Object is listed in the ‘Objects’ chapter, with its icon below the heading, and a picture of the menu that you will get when you click on it with the right mouse button on the first page. This format is carried on into the Event Editor, so for a quick list of the menu, just go to the first page of that object’s topic.

The Editors

In order to simplify your games creation, The Games Factory divides the creation process into several different stages. This is done by the use of the ‘Editor’ screens. Each Editor screen allows you to manipulate the characters of your game in a different way. One will allow you to decide which characters to put on the screen and how to animate them, another will allow you to add music and movement. Some of the functions overlap, so look out for the ‘Handy Hints’ along the way.
You can move very quickly and easily from one editor screen to the next simply by clicking on the relevant icon from the toolbar at the top of each editor screen. Don’t worry if you cannot remember what each one looks like straight away. As you move the mouse pointer over the icons a text-balloon will appear underneath the mouse pointer, telling you what each one is.

Below is a brief description of the functions of each of the editor screens. These functions are described in more detail in the chapters relating to them.

Storyboard Editor



No.	Thumbnail	Comments
1		Title : Untitled Password : 640 by 480
2	More...	

Most games are composed of several different levels, and this screen allows you to add levels to your game, copy levels and to change the order of the levels by moving them around. This is also where you decide on the size of your playing area, adding and editing professional looking fading in and out of each level, plus using passwords to enter each level.

Note that until you have created a level, the only way to move out of this screen is by clicking in the 'Thumbnail' (so called because it shows a tiny screen shot of that screen) with the right mouse button, and select in the 'Edit this level' option. There is more detailed information in the chapter 'The Storyboard Editor' on page 112 but it is recommended that you at least read the summaries of each of the Editors before you go straight into your games creation!

Level Editor

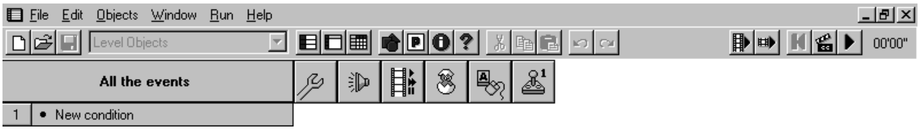
The Level editor is the initial 'Blank Page' of each of your levels. It displays your play area and is where you decide to put background objects and the main characters of your game.

It is from this screen that you have access to the libraries of all the different objects that you can use, and is also where you can create your own animated objects, text and other object types. Basically, all the objects that you want to play with have to be placed on this screen first before you can start manipulating them.

It is also here that you change the animation of objects, their movement and change the basic setup of all your objects. You will frequently find that before you can manipulate an object from the Event Editor you must make sure that it is set up correctly on this screen.

For more information on the Level Editor, got to page 117, but we do recommend that you finish this chapter at least before doing so!

Event Editor



This is where your game will really come to life. Once you become experienced with The Games Factory you will find that this editor is where you will spend most of your time. It is here that you decide all the events in your game, e.g. What happens when the alien collides with the spaceship? What happens when your hero eats the hot porridge?

All of these events are defined here and you decide which actions to apply to them; which sound to play, create an explosion or destroy an object, add to the score, subtract a life, or even complicated events like changing the direction of the evil cowboys chasing after your hero when he picks up a gun.

This is a very large chapter, so some knowledge of how to move around it is useful. The first portion of this chapter is essential, as it explains the differences between the Events and the Actions. All the objects are listed in the same order throughout this chapter, for their Events, Actions and Functions. Functions are called up using the Expression Editor.

Step Through Editor

The Step Through Editor can be very useful for the inexperienced Games Factory user. You can have the Step Through Editor detect all the possible Events that can happen in your game and put them into the Event Editor. More experienced users will find that it will 'overload' them with conditions. For a lot of games you will want nothing at all to be done for most of the Events. It will run your game step-by-step and stop it every time an Event occurs, and then ask you which actions you want to put to that Event. Bear in mind, though, the number of Events that are possible in your games is usually very large indeed, even with a very simple game. It is thus best used only initially to help you in the construction of your Event grid when using the Event Editor. Don't worry, these terms are explained fully later in the manual!

Installing The Games Factory

WINDOWS 95 USERS

Windows 95 users can install the 32 bit version of The Games Factory to their machine. To do this simply reset your machine and insert The Games Factory CD into your CD ROM drive.

If you have not installed The Games Factory previously then it will automatically install when you run Windows 95.

If you have installed The Games Factory before, or you have not reset your computer, (we recommend that you do) then use the following procedure;

Double-click on the ‘My Computer’ icon from your Windows desktop.
When you are presented with the ‘My Computer’ dialogue box, click on the pull-down button next to the text-box ‘My Computer’.
This will pull-down a list of all the available drives for your PC. Select the CD ROM icon from the list.
You will now be presented with a list of the contents of the CD. Look down the list for a file called ‘Setup.exe’. Double-click on this to start the installation.

Windows 3.x users

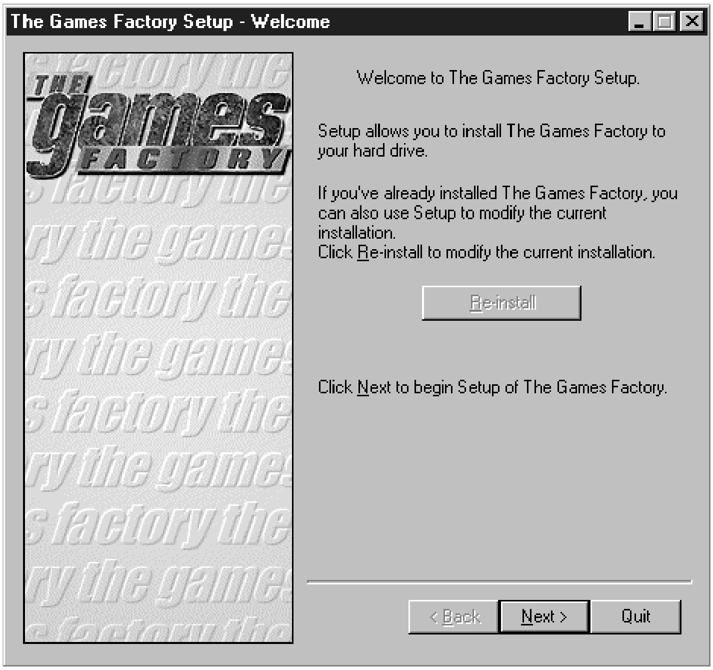
Users of Window 3.x can install the 16 bit version of The Games Factory.
First, reset your computer and insert The Games Factory CD into your CD ROM drive.
Now double-click on the Main icon from your Windows desktop. From here double-click on the File Manager icon.

From the File Manager select the CD ROM drive from the list of drives on the toolbar.
The window in the centre of the screen should now display the contents of the CD.
Look for a file in the right of the window called ‘Setup.exe’. Double-click on this to start the installation.

The Installation

The following procedure is very similar for both the 16 and 32 bit versions. The screen shots are all from the 32 bit version. Where applicable the differences are described.

You will now be presented with a screen that asks if you have already installed The Games Factory and if you want to modify the settings. Click ‘Next’ at the bottom of the screen.



The Games Factory Setup - Welcome

Welcome to The Games Factory Setup.

Setup allows you to install The Games Factory to your hard drive.

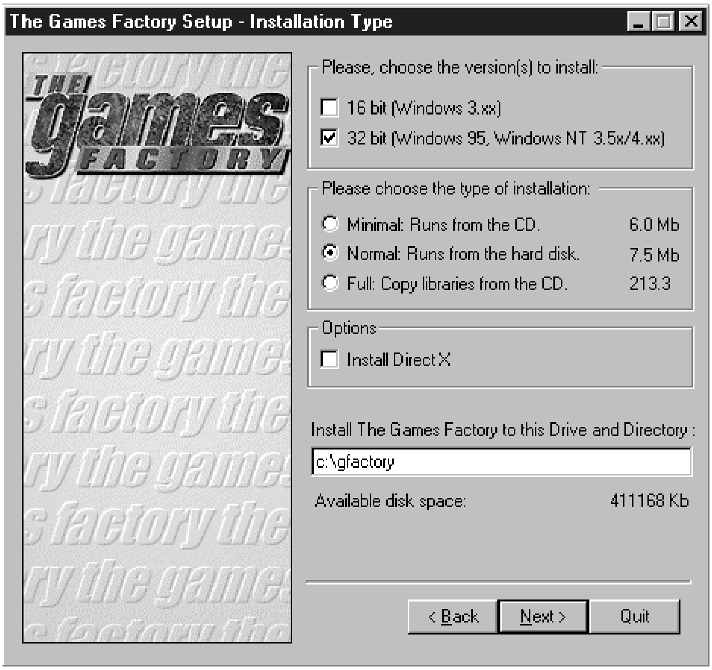
If you've already installed The Games Factory, you can also use Setup to modify the current installation.
Click Re-install to modify the current installation.

Click Next to begin Setup of The Games Factory.

The next screen asks if you want to install either Video for Windows or a QuickTime video driver for showing AVI and QuickTime compatible video clips. Unless you want to install one of the drivers, click ‘Next’.

The next screen asks you to type in your full name. When you have done so, click ‘Next’.

At the top of the next screen are two tick-boxes, one for the 16 bit version and one for the 32 bit version. If you are running Windows 95 on your machine, you can install the 32 bit version. The tick-box will be filled already. If you are running Windows 3.x then the 16 bit tick-box will be filled.



The Games Factory Setup - Installation Type

Please, choose the version(s) to install:

□ 16 bit (Windows 3.xx)
☑ 32 bit (Windows 95, Windows NT 3.5x/4.xx)

Please choose the type of installation:

○ Minimal: Runs from the CD. 6.0 Mb
● Normal: Runs from the hard disk. 7.5 Mb
○ Full: Copy libraries from the CD. 213.3

Options

□ Install Direct X

Install The Games Factory to this Drive and Directory :

c:\gfactory

Available disk space: 411168 Kb

Below this are three radio-buttons, asking how complete an installation you want to perform. The minimal installation places as little as possible of the contents of the CD into your PC, but because it runs from the CD, it will slow down the running of the program a little.

Normal installation installs everything except the object libraries and sound files onto your PC. This radio-button will be filled by default, and is the normal option, as it does not take up too much memory, but still runs the program quickly. Note you will need to have the CD in your drive when you run the program.

The Full installation installs the main program and all the graphics and sound libraries from the CD onto your PC. This allows for the fastest running of the program, but it does take up a very large amount of hard disk space and this is undesirable for many people.

For 32 bit (Windows 95) users, there is a tick-box for ‘Direct X’, which is Microsoft’s own graphics driver. Unless you specifically don’t want to install Direct X, place a check-mark in the button by clicking with the mouse pointer. Should you experience problems using Direct X you should change the graphics driver. See ‘General Setup’ on page 85 after installation.

For 16 bit (Windows 3.x) users there is a tick-box for ‘Win G’, which is Microsoft’s own graphics driver. Unless you specifically don’t want to install Win G, place a check-mark in the button by clicking with the mouse pointer. Should you experience problems using ‘Win G’ you should change the graphics driver. See ‘General Setup’ on page 85 after installation.

At the bottom of this screen is the file name that The Games Factory will use.
Unless you specifically want to enter a different name, click the ‘Next’ button.
You will then be told that the directory does not already exist, and asked if you want to create it.
Click on the ‘OK’ button to proceed.

Now wait while The Games Factory installs.

When it is finished installation, click on the ‘Return to Windows’ button to return to Windows. Now you can double-click on The Games Factory icon to run the program!

We recommend that you look through the examples that we have laid on for you after you have first loaded up The Games Factory.
To view these, click on the ‘File’ heading at the top left of the screen, then click on the ‘View examples’ option. You can then select whether you want to look at the tutorial games, the fantastic games ‘Zeb’, ‘Lobotomy’ and ‘Magicians Lair’, or the advanced techniques.

The Tutorial Games

This section of the manual is dedicated to taking you on a step-by-step mission to construct five tutorial games, from a very basic, one-level shoot-'em-up, to a platform game complete with high scores, music, scrolling scenery and animated characters that will really show you how easy it is to create games of stunning quality.

It is recommended that you occasionally refer to parts of the main manual though, as this will give you more information on the background of some of the objects and operations. This will not be necessary for the construction of the tutorial games, but it will help you learn your way around the manual, useful for when you embark upon your own gaming construction.

It is assumed that before starting on this section that you have successfully installed The Games Factory to your hard drive. If you have not done this, please turn to 'Installation' on page 7 to correctly install The Games Factory to your machine.

Loading Spacebattle

Select the 'Open' option from the 'File' menu at the top of the screen, then look for the 'tutorial' directory on your CD ROM drive.

Within the 'tutorial' directory look for the '01space' sub-directory. Now look for the directory 'tutorial' and within this look for the file '01 space.gam'. Open this file.

Tutorial Game 1

Spacebattle

Spacebattle is the first of the tutorial games and will show you the very basics of creating games with The Games Factory. Even so, a lot of the procedures that you will be using for this game are basic to all games and extremely useful.

This tutorial will help teach you some of the different functions of the different Editor screens and how they relate to each other. You will also get a basic introduction into the use of the pull-down menus at the top of each Editor screen.

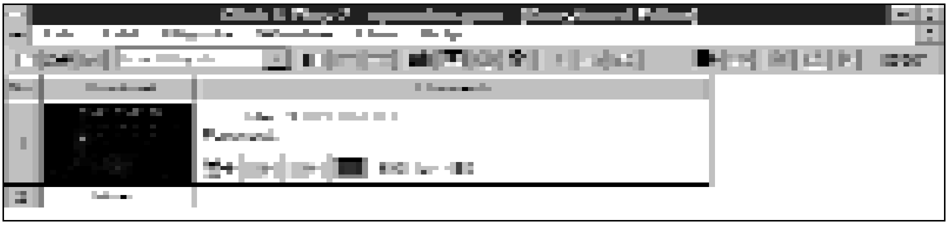
Perhaps most importantly, though, is the step-by-step creation of different events and actions in the Event Editor, and how this affects how the game is played.

Although a very basic game, you can always add further features to this game to make it far more complex - changing the movement paths, adding sound, power-ups, high score tables. It is recommended that you work your way through all the other tutorials or the relevant sections of the manual before doing this, though.

Important Note!
For those of you who have not used 'Windows' software before, it is important to note that when we talk of 'Selecting' an object, that means moving the mouse pointer over the object and clicking the left mouse button. When the right button is to be used, you will be told specifically.

Spacebattle – The Storyboard Editor

Once you have loaded in the Spacebattle game, you will be presented with a Storyboard Editor screen that looks like this;



Starting from the top of the screen, we shall describe the various features of the Storyboard Editor screen. Note that a lot of the features are the same for all of the Editor screens, but what they actually perform on each Editor screen can be quite different.
At the top right is the Window Manipulation icon. By clicking on this you can maximize or minimize the size of the window displaying The Games Factory, or exit The Games Factory. Note that the Storyboard Editor is a window within The Games Factory window, and it has its own window manipulation icon immediately beneath The Games Factory's.

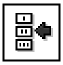
On the very top of the screen is the The Games Factory heading bar. This displays the current game name, in this case 'Space.gam'. Next to this is the name of the current Editor screen, 'Storyboard Editor'

Below this is the pull-down menu bar. From here you can save your games, change the size of the screen, change preferences, access the Help pages and customize your display. This is described in detail in the chapter 'Menu bar' on page 84.

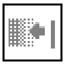
Below the menu bar is The Games Factory toolbar. From here you can very quickly and easily move around the Editor screens by clicking on their icons, save games, cut and paste objects or events, plus test running your games. All the different icons are described in the 'Toolbar' chapter on page 110.

On the main window area of the Storyboard Editor is where each level is displayed in summary. By default a ‘Thumbnail’, or very small screen shot of that level is displayed immediately next to the number of that level.
Next to the thumbnail are the comments for that level, the title of the level and the password. To change these, simply click on the text that you want to change with the left mouse button. You can then change / add a title or password.

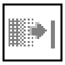
Underneath the comments are several ‘buttons’;



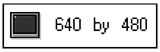
This denotes a multimedia level, which is what all your levels will be by default. If you were to make your level an animation frame, this icon would change.



You can add a fade-in transition to your level by using this icon.



You can add a fade-out transition to your level using this icon.



640 by 480

This is where you can select the size of your play area. The play area can be much larger than the screen size if you want, then you can scroll around it using a scroll function. Scrolling is dealt with on page 180 and in the later tutorials. By selecting on the numbers you can enter your own sizes via the keyboard, or by clicking on the monitor icon you can pick one of the pre-determined sizes that The Games Factory has to offer.

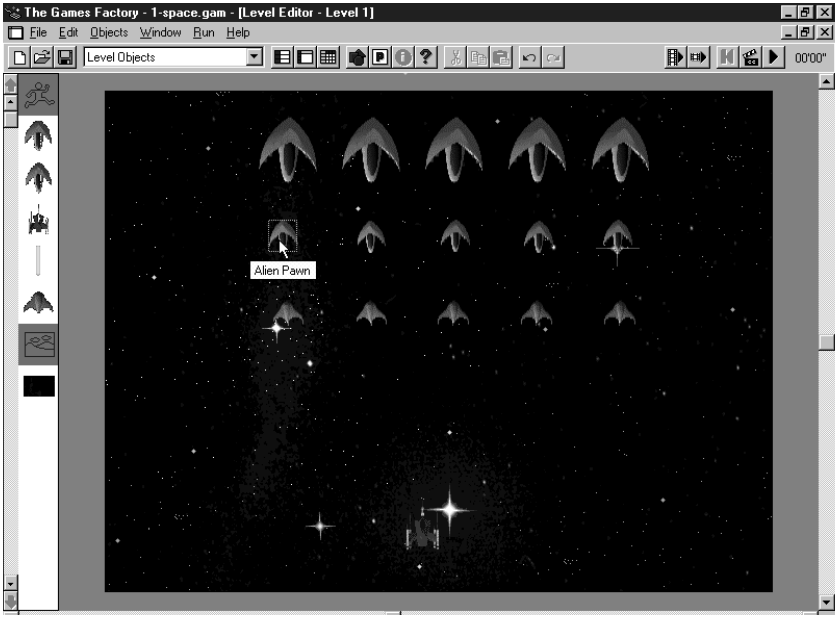
Right, now that you’ve got a general idea of what all the different features are on the Storyboard Editor screen are, let’s go and look at how we put all the objects on the screen. Don’t worry if you have questions about the Storyboard Editor, we are coming back to it soon, and you can always turn to the ‘Storyboard Editor’ chapter on page 112 for detailed information.

Spacebattle – The Level Editor

To get into the Level Editor when you first load a game, or start a new one, you need to click inside the ‘Thumbnail’ of the level display with the right mouse button, then from the menu that produces select the ‘Edit this level-Level editor’ option. This will then take you to the Level Editor.
Note that once you have been into the Level Editor or the Event Editor once using this method, all you have to do is click on the icons on the toolbar to move through the different Editor screens instead.
Now you are in the Level Editor, let’s have a look around.

You will notice that the toolbar is exactly the same as on the Storyboard Editor screen, only now you can access the object libraries via the 'Level Objects' window in the toolbar.
Notice also that you can see all of the objects that have been used to create 'Spacebattle' on their own Object Window on the left hand side of the screen.
Not all of the objects fit inside that window, so there is a scroll bar to look through them. Try looking through the contents. You will also find a 'Score' object at the bottom of the library. When you are making a very large level you will need to get used to scrolling through the window to view all of your objects.
(You can also change the window to a 'Moveable Window' from the Menu bar at the top of the screen, using the 'Objects-Display objects in a moveable window' option. For more information turn to the chapter 'Menu Bar', but be sure to carry on here if you do!)

If you move your mouse pointer over the objects in the object window a 'Handy hint' will show you the name of each object. You could now select one of the objects with the left mouse button and then place it anywhere on the screen. Try it!
When you have tried placing a few objects on the screen, select them with the left mouse button and use the <Delete> key to erase them. You must do this before you carry on with the tutorial otherwise it will change what you find in the other Editor screens.


The Games Factory - 1-space.gam - [Level Editor - Level 1]
File Edit Objects Window Run Help
Level Objects
00'00"
Alien Pawn

Try also moving the mouse pointer over the objects that have already been placed on the screen. You can move them around by selecting on them with the left mouse button and ‘dragging’ them around by holding the left mouse button down.

Now, go to the scroll bars at the side of the screen, and try moving around the play area using them. You will notice that there is a grey area which is the edge of the play area. Any objects placed here will not be shown on the screen when you play your game, but it can be a useful ‘holding area’ for placing objects in a level, which you can then move onto the play area later on, by using the Event Editor.

Note: You must have an object placed on the screen on the Level Editor screen before you can manipulate it in the Event Editor, the exceptions being when you create or shoot an object from the Event Editor.

Now let’s have a look at the menu options for each object. Different types of object have different menus, as you’ll come to learn.

Try pressing the right mouse button when you have the mouse pointer over the ‘Space Backdrop’ object. Have a look through all the different options, but don’t change anything. Now try it over any of the other objects on the screen. We shall be using these menus in a later tutorial.

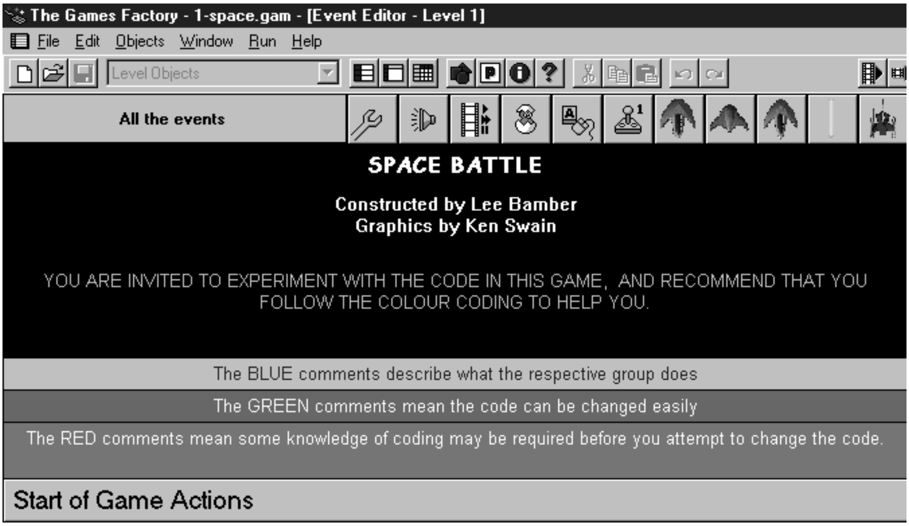
Right, now that we’ve had a quick tour around The Games Factory’s more basic features, let’s have a look at what the game plays like!
To be sure that you have made no changes to the game at all, re-load the game. To do this, click on the ‘File’ menu at the top of the screen, then select the ‘Open’ option. Now re-load Spacebattle, right click in the thumbnail, select the ‘Level Editor’ option.

Look at the top right of the Toolbar. There are several control buttons there which control the testing of your game. For now we are just going to use the Play / pause button ▶ and the restart button ⏮.

The controls for the game are Left / Right (cursor keys) and fire is the <Ctrl> key.
Now click on the Play button. To stop the action, click on it again. When you want to finish playing, be sure to use the restart button to reset everything.

Spacebattle – The Event Editor

To get into the Event Editor, simply click on the ▶ icon on the toolbar at the top of the screen. You should now be presented with a screen that looks like this;


The Games Factory - 1-space.gam - [Event Editor - Level 1]
File Edit Objects Window Run Help
Level Objects
All the events

SPACE BATTLE

Constructed by Lee Bamber
Graphics by Ken Swain

YOU ARE INVITED TO EXPERIMENT WITH THE CODE IN THIS GAME, AND RECOMMEND THAT YOU FOLLOW THE COLOUR CODING TO HELP YOU.

The BLUE comments describe what the respective group does

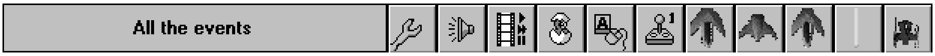
The GREEN comments mean the code can be changed easily

The RED comments mean some knowledge of coding may be required before you attempt to change the code.

Start of Game Actions

In the Event Editor is where you decide upon the action and strategy in your games creation, where you add sound, explosions, move onto the next level, display Hi Score tables, the list is almost endless! The Events and Actions are described in the Event Editor chapter, but we shall be going into some detail about how to create the Event lines used in ‘Spacebattle’ here.

In addition to the menu bar and toolbar, the Event Editor basically consists of one horizontal line, which contains the icons of all the level’s objects.
It also has a vertical column of grey ‘Event’ lines. Look below the heading ‘Start of Game Actions’ and you will notice that next to each Event line, there is a grid of boxes, each box lining up with an objects icon. For example, you can insert an Event that reads, ‘(Active object X) collides with (Active object Y)’. You now have an Event. What you would then want to do would be to decide upon an Action to assign to that Event. If Active object X was a bullet that you fired, and Active object Y was an alien, you might want to ‘Destroy (Active object Y). ‘We shall discuss how to insert Events and Actions in a moment.
You can see the horizontal line at the top of the Event Grid. It looks like this in ‘Spacebattle’;



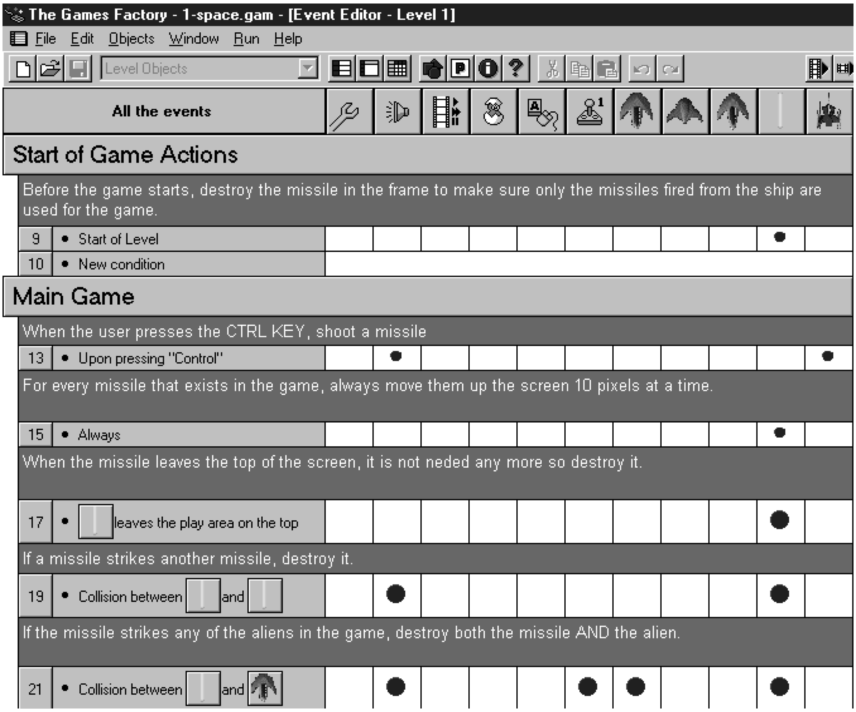
'All the events' is the heading for the vertical column, which is where you insert all the Events. We shall describe this after talking you through the icons held in the horizontal bar.

The icons at the right of the bar refer to all the level's objects. Shown here are all of the objects (except backdrop) that have been placed onto the play area. You can see the alien objects here, as well as the players ship and the missile. It will also show any object that you use as bullets or create from within the Event Editor, even if they are not already on the play area. Shooting objects is described later in the chapter, in step 36.

The first six icons denote Game Objects, and are always on the event grid by default. They are, in order;

- Special object. Performs special functions when an Event occurs.
- Sound object. Plays sound or music.
- Storyboard object. Moves you around the different levels of your game.
- Create object. Allows you to place a new object on the screen.
- Mouse and Keyboard object. Lets you interact with the mouse and keyboard.
- Player object. Allows you to change lives and scores.

Now, before we go any further, use the vertical scroll buttons on the right side of the screen to move down through the Events until you can see Event number 21 just appearing on the bottom of the screen. You should now have a screen that looks like the one overleaf;



The Games Factory - 1-space.gam - [Event Editor - Level 1]

All the events											
Start of Game Actions											
Before the game starts, destroy the missile in the frame to make sure only the missiles fired from the ship are used for the game.											
9 • Start of Level										●	
10 • New condition											
Main Game											
When the user presses the CTRL KEY, shoot a missile											
13 • Upon pressing "Control"		●									●
For every missile that exists in the game, always move them up the screen 10 pixels at a time.											
15 • Always										●	
When the missile leaves the top of the screen, it is not neded any more so destroy it.											
17 • leaves the play area on the top										●	
If a missile strikes another missile, destroy it.											
19 • Collision between and		●								●	
If the missile strikes any of the aliens in the game, destroy both the missile AND the alien.											
21 • Collision between and		●				●	●			●	

You can now see three Event lines within the group ‘Start of Game Actions’, the green comment line, and the two actual Events.

Below this you can see the group ‘Main Game’, which is so large that it doesn’t fit onto the screen.

Now you can see what the heading ‘All the events’ relates to. It refers to all the Events that are listed down the left hand side of the screen.

You can also see that there is a grid with vertical columns that correspond to the objects displayed on the bar at the top of the Event grid. In some of the boxes of the grid there are ‘check-marks’. These mean that an Action has been inserted in that box.

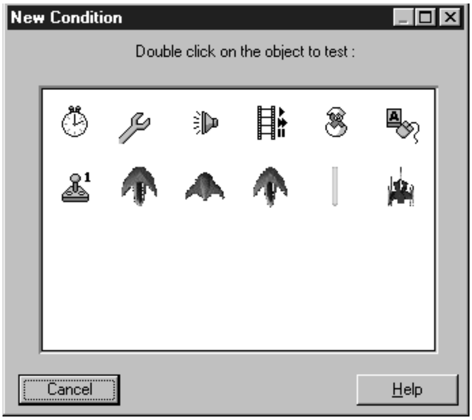
For example, look at Event line number 9. If you move your mouse pointer over the check-mark on this Event line you can see a ‘Handy Hint’ text box that says ‘Destroy’. Event number 9 reads ‘Start of level’, so, at the start of the level, destroy the missile object.

This ensures that they are all removed from the play area before you start your game, and that the only missiles on the screen will be the ones you fire from the spaceship.

Have a look through the other Events, move your mouse pointer over the check-marks so that you can see which Actions are associated with which objects. The comments above each set of Event describes what each Event line does, too.

Right, let's have a go at reproducing one of the Event lines.
We are going to produce a copy of Event line number 9. There is a very easy way of copying Events, but we are not going to use that here. We are going to go step-by-step to show you how to create your own Event lines.

1. To create a new Event line, move the mouse pointer directly over the text 'New condition' of Event line number 10. 'New condition' means that this line is completely empty. Notice that it has no grid next to it.
2. Now, click on the text with the right mouse button.
You will now have a dialogue box that looks like this;



3. If you move your mouse pointer over each object you will be shown what each one is via the 'Handy Hint' text balloon.
4. Now, click on the 'Storyboard controls' icon with the right mouse button. You will pull-down a menu.
5. Click on the 'Start of level' option with the left mouse button.
You will now be taken back to the Event grid, and event number 10 will read 'Start of level'. You've created your first Event!

Now all you need to do is put in an Action.
1. Go across the empty check-boxes on line number 10 until you get to the box underneath the icon of the missile.
2. Click in the empty box with the right mouse button. You will then be presented with a menu of all the things to do to the missile object.
3. Now move down the menu and using the left mouse button, click on the 'Destroy' option. You will then be taken back to the Event grid and Event line 10 should look exactly the same as Event line number 9. Well done!

Creating Spacebattle Yourself

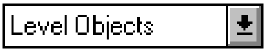
Note: It is assumed that for this part of the tutorial especially, that you are actually sat in front of your PC with the game Spacebattle already loaded.

Okay, now you've had a 'guided tour' around how Spacebattle was made, let's have a go at making it yourself.
First of all we'll need a 'blank page' to work with.
1. Go to the 'File' heading on the menu bar at the top of the screen. Select the 'New' option. Games Factory will ask you if you want to save the changes that you have made to 'Spacebattle'. We already have it safe on disk, so click the 'No' option.

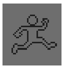
2. You will now be asked to select a playfield size. We want something that fills the screen, so click on the '640 by 480' size, then click 'OK'.

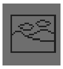
3. You will now be taken to the Storyboard Editor, with an empty level 1 display. To enter a name for the level, click on the 'Untitled' text within the level 1 display, and enter a name, then press <Return>.
Click inside the empty 'Thumbnail' with the right mouse button. From the menu produced click on the 'Edit this level' option, then the 'Level Editor' option. You will then be taken to an empty Level Editor screen, ready for you to place your objects on.

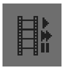
4. To place the objects on the screen, go to the 'Level objects' text box on the toolbar at the top of the screen, and click on the pull-down button at the side of the text.



5. Now, look through all the different object libraries until you get to one called '(Tutorial) SPACE BATTLE'. Click on this.
6. You will now have all the objects from this library displayed in the Object window down the left hand side of the screen. You will notice that there are large icons within this window. These denote the type of object, for example;


denotes Active objects.


denotes Backdrop objects.


denotes Storyboard objects. In this case a Score object.

Try moving your mouse pointer over the different objects in the window. A ‘Handy Hint’ text box will tell you what each one is called.
Before we place the Active objects of our game onto the play area, we are going to put a backdrop into place, align it perfectly with the edges of the screen, then ‘lock’ it into place so it cannot be selected or moved whilst we are putting all the other objects onto the play area.

Placing the backdrop and locking it in place

7. To place the backdrop onto the screen, go to the object window and click on the backdrop icon, called “Space Battle Backdrop”, using the left mouse button.

8. Now move the mouse pointer over the play area, anywhere will do, then press the left mouse button again.
This places the backdrop object onto the play area. Note that when the backdrop object and the size of the play area are identical, The Games Factory will automatically align the backdrop object to fit.

9. We are now going to stop the backdrop object being selected every time you click on it with the mouse. This will make it far easier when you start putting other objects on the play area.

Go to the toolbar at the top of the screen and click on the preferences icon P with the left mouse button. (You can learn more about preferences from the Menu Bar chapter on page 84. Look under the option ‘Edit-Editor preferences)

10. Now you will be presented with a dialogue box. Under the ‘Lock objects’ heading, look for ‘Backdrop’. There will be an empty check-box. Click in the check-box with the left mouse button. It will now be filled with a small ‘X’.
Now click on the ‘OK’ button.
Now when you try clicking on the backdrop object, you cannot! As you saw from the different options in the preferences dialogue box, you can lock lots of different types of object into place. To unlock the backdrop object, click on the preferences icon and then click in the ‘Backdrop object’ radio-button again.

Placing the main characters neatly onto the screen

Now that we have a nice backdrop for our game, we are going to place Active objects onto the play area, the Aliens, the bullet and the main characters spaceship.

So, before we put the object 5 onto the screen, we are going to put a grid on the screen that we can use to align the Aliens neatly, rather than somewhat haphazardly using the mouse.

11. Go to the preferences icon again, P at the top of the screen. Click on this using the left

mouse button. Now when you are presented with the Preferences dialogue box, click on the second ‘Layer’, called ‘Grid’, with the left mouse button.
You will now be shown the second layer of the preferences screen.

12. Look for the heading ‘Square size’. You will see that both the size and height are set to 32, which means that the grid is composed of lots of squares 32 pixels wide by 32 pixels high.
We are going to leave the height alone, but we want to change the width to a smaller number, allowing us to horizontally line up the aliens a little more finely.
So, click on the actual numbers in the ‘Width’ box using the left mouse button. Now, using the cursor and <Delete> key, erase the numbers in the box and type in ‘10’.

13. Now go towards the bottom of the dialogue box where there are two radio-buttons. Go to the ‘Snap-to’ radio-button and click in the empty box with the left mouse button. This means that any object you place on the screen will ‘snap’ to the grid. We are not too bothered about looking at the grid for this game, so leave the ‘Show grid’ box empty. It does not interact with the Editor screens in any way apart from to show the grid when you are on the Level Editor screen.
Now click the ‘OK’ box at the bottom of the dialogue box to take you back to the Level Editor screen.

Placing Objects on the screen using the left and right mouse buttons

Now we are ready to actually start placing the objects on the screen, which will be nice and neat because we have enabled the grid.

To place objects onto the screen, first go to the object window and select the object you want, using the left mouse button. You can then move the mouse pointer to wherever you want that object to be placed on the screen. You can now either click with the left mouse button, which places a single copy down, or use the right button, which allows you to place multiple copies down without having to return to the object shelf every time you want to place an object.

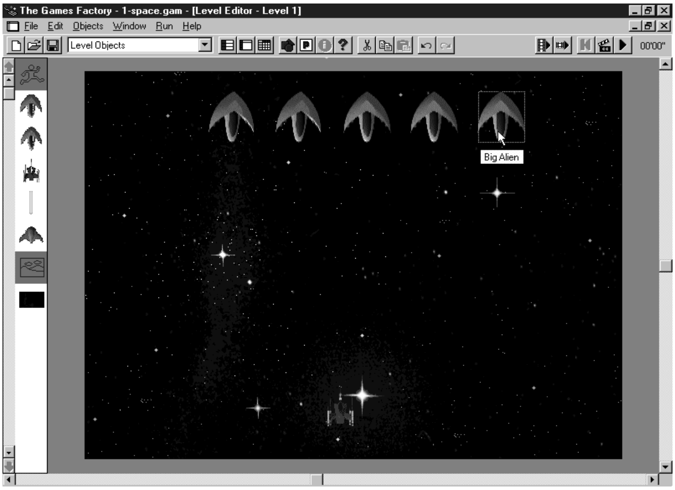
So, now you know the difference between placing the objects using the left and right keys, let’s lay out a wave of aliens!

14. Go to the object window and click on the ‘Big Alien’ with the left mouse button.

15. Now move the mouse pointer over the play area. Let’s start at the top left of the screen and start placing our Aliens there. Notice that the object jumps about as you move it. This is the effect of having the grid ‘snap’ enabled.
Select the place that you want to place the first alien, but don’t press any buttons yet. Don’t place it too close to the edge of the screen, otherwise it will disappear off the screen when you play the game. Note that you can still scroll around the display area using the scroll bars even when you are ‘carrying’ an object.

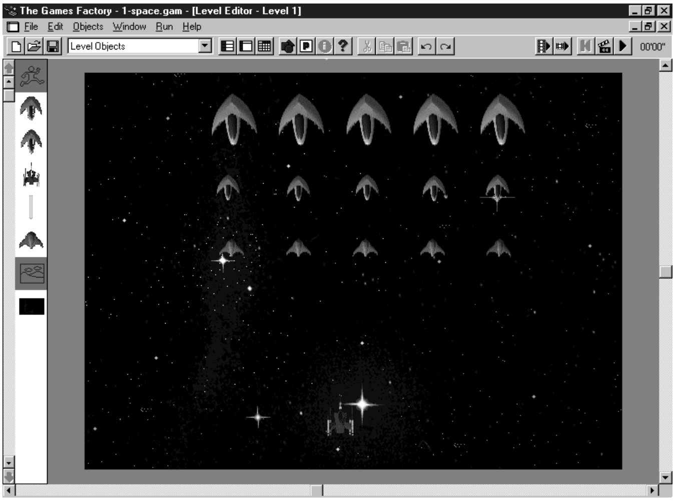
16. Use the right mouse button to place an alien there. Now move onto the next place that you want to put the next alien. We are trying to make a line of aliens evenly spaced, across the top of the screen. Stop when you have placed four aliens. You can now see how easy it is to line all of the aliens up exactly with the grid-snap on.

17. When you have placed four aliens, place the last one using the left mouse button. This will stop you placing any more clones.


The Games Factory - 1-space.gam - [Level Editor - Level 1]
File Edit Objects Window Run Help
Level Objects
00'00"
Big Alien

18. Now go to the object window and click on an ‘Alien Pawn’, and place a row of five of them immediately beneath the ‘Big Aliens’ using the same method as above. Try not to have them too close together.

19. Repeat the above process for the ‘Tiny Alien’. You should now have an entire attack wave consisting of three types of aliens, fifteen in all. A veritable fleet!


The Games Factory - 1-space.gam - [Level Editor - Level 1]
File Edit Objects Window Run Help
Level Objects
00'00"

20. The next stage is to place our hero, the ‘Players ship’ object, onto the bottom of the play area. Place him just slightly above the bottom of the screen.

21. Now place the bullet, ‘Missile’, anywhere on the screen. When we use the Event Editor, we are going to remove any missiles from the display when the level first starts, so we are only placing it on the screen so that it will appear in the Event Editor.

22. Now scroll through the object window until you find a ‘Score’ object. Place that on the bottom of the screen, out of the way of everything else.

To save you the complication of having to define the movements for all the different objects in this first tutorial, we have done it already. If you want to learn about movement right now, turn to the Objects chapter and look up ‘Active objects-Movement’. We will be covering movement in later tutorials, though. You can look at how the movement of each object was created, however, by right mouse clicking on the object, then select the ‘Movement-Edit movement’ option. The aliens movement was defined as a path, which you will be able to see.

You can have a look at what you’ve done so far, in action, by clicking on the ▶ button on the toolbar at the top of the screen. Remember to press it again to pause the game, then use the ⏮ button to reset the game.

You should have a field of aliens moving around the screen smoothly, and you should be able to move the spaceship from left to right by using the cursor keys.

Now that we have the basic objects on the screen, we need to add some actions, sound, etc.
To do all this you need to go to the Event Editor.
But, before we do this, let's save the work that we have done.
Go to the 'File' text on the menu bar at the top of the screen. Click on it using the left mouse button. Now click on the 'Save as...' option.
You will now be presented with a file selector. Select the disk drive that you want to save to. Now type in the name that you want to save it as. If you are using Windows 3.1, this can be up to 8 letters long, and must always have '.gam' after it, denoting this as a game file. Windows 95 allows you file names with up to 256 characters. To keep things nicely ordered, enter the name 'Space1', then click 'Save'.

You can also load up The Games Factory version of the game so far. Look in the directory 'gametuts', then for the sub-directory '01space', then 'tutorial', then 'spacetu1'.

Once you have safely saved your work, click on the ▦ button on the toolbar. This will take you to the Event Editor.

Making Spacebattle Events and Actions

The Event Editor is where your gaming really comes to life. You have seen how easy it is to place your objects on the screen, but now you want to fire missiles, destroy aliens, add sound and change the score.

Step by step, we are going to take you through creating a complete set of Events that brings Spacebattle to life. A lot of what you will learn in this chapter is basic to all games creation using Games Factory, and will not be repeated elsewhere, so reading this chapter is a must before you head on.

To start with, before we even create Events, we are going to show you how to create 'Comment' lines, which do nothing other than provide you with notes to refer to in the Event Editor. Although Spacebattle is a very simple game and presents no problem for the experienced user to find their way around in, writing comments in the Event Editor is a very good habit to get into for when you start writing longer, complicated games.
We are also going to show you how to define 'Groups' of Events, which again make your Event editing far easier to follow, and allow you to simplify your game writing.

So, enough of this merry banter, let us make a start on the Events!

Creating a Group of Events

The first thing we are going to do is to create a group of Events called ‘Start of Game Actions’.

23. To do this, right-mouse button click on the number 1 of the ‘New condition’ line.
Now, select the ‘Insert’ option from the menu, then the ‘A group of events’ option from that menu.

24. You will now have a dialogue box asking you to enter in the name of the group. Type in the text “Start of Game Actions”. When you have done this make sure that the radio-button ‘Active when frame starts’ is filled. It will be filled by default, but check it anyway. When you are happy with the name entered, either press <Return> or click on the ‘OK’ button.

25. You should now have a group called ‘Start of Game Actions’. Event line number 2 will be indented slightly, denoting it as being within that group. We are now going to make Event line number 2 a ‘Comment’.

Creating a Comment line

26. Now right-mouse click on the number 2 of Event line number 2. Select the ‘Insert’ option, then the ‘A comment’ option.

27. You now have a dialogue box where you can select the font, colour and background colour of your text. We are going to use the standard font, but write in white, so...
(i) Click on the ‘set font colour’, then click on the white square. Then click the ‘OK’ button.
(ii) Click on the ‘set back colour’, then select dark green, then ‘OK’.
Now click in the top left of the empty text box. Now you can type in the words that you want to appear in the Comment.
Enter the text, “Before the game starts, destroy the missile in the frame to make sure only the missiles fired from the ship are used in the game.”
Now click ‘OK’.
You will now have a comment with a green background and white letters. If you have made a mistake, you can edit your comment line by right mouse clicking on the text of the comment, then select the ‘Edit comment’ option.

Creating an Event Line

Right, time to get down to the nitty-gritty of your games creation!

28. We are going to insert an Event ‘Start of level’, which means that when it is the start of the level, all the Actions associated with this Event line will be performed.

This Event is going to be within the group ‘Start of Game Actions’.
So, click on the text ‘New condition’ of Event line number 3 with the right mouse button.

29. You will now be presented with a dialogue box titled ‘New Condition’ with the sub-heading ‘Double click on the object to test’. Move your mouse pointer over the ‘Storyboard controls’ icon, (The ‘Handy Hint’ text will tell you which one it is as you move your mouse pointer over the objects) and either double-left click, or single right click on it. This will produce a menu from the object. Select the ‘Start of level’ option with the left mouse button.

30. You will now have an Event line number 3 that reads ‘Start of level’. You can now insert an Action.

Save Your Game

Before you go any further, save your game so far.

So, click on the ‘File’ menu at the top of the screen, then select the ‘Save as...’ option.
Now under the ‘File name’ text box, enter the name you want to use. To keep things nicely ordered, enter the name ‘Space2’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘01space’, then ‘tutorial’, then ‘spacetu2’.

Adding Actions to Event Lines

We are now going to insert an Action into Event line number 3, to destroy the missile.

Destroying Objects

31. Right-mouse click in the empty box below the missile icon, on line number 3. This will produce a large menu of all the different possible Actions that you could do to the missile.

32. Select the ‘Destroy’ option with the left mouse button. You will now have a complete Event line that will destroy the missiles at the very start of the level, within the group ‘Start of Game Actions’.

Now it’s time to start adding the rest of the Events. These are not going to be described in as great detail, as it is assumed that you will have learnt how to perform the more basic functions. They will be referenced by number to previous steps, though, so if you get stuck you can see how previous steps were achieved.

33. Create a Group called ‘Main Game’. Use Event line number 5 to do this. See step # 23 for more information.

34. Now insert a Comment using Event line number 6. The text of the comment is “When the user presses the SHIFT KEY, shoot a missile”. See step # 26 for more information.

Testing the keyboard for a specific key depression

35. We are now going to create an Event ‘Upon pressing “Shift”’. To do this, right click on the ‘New condition’ text of Event line number 7. Now use the right mouse button to select the ‘Mouse pointer and keyboard’ icon. Select the ‘Keyboard’ option from this menu, then the ‘Upon pressing a key’ option.
You will now be asked to press the key that you want to associate your actions with. So, press the <Shift> key. You will now be taken back to the Event Editor screen, with an Event line that reads ‘Upon pressing “Shift”’.

Shooting an object

36. Now that you’ve made your Event line to test when the <Shift> key has been pressed, you need to actually shoot something when it is pressed.
(i) Go to the empty box that lies directly underneath ‘Players ship’ and on the same line as Event line number 7. Click in it with the right mouse button. Now select the ‘Shoot an Object...’ option from the menu produced.
(ii) You will now be presented with a dialogue box displaying all the Active objects from the current level. We want to use the ‘Missile’ object as our bullet, so select that, then click ‘OK’.
(iii)You will now be asked to select a speed and direction for the bullet.
By default the speed is 100, which is very fast. Use the mouse to move the pointer to 50.
(iv) Now you want to select the direction to fire in, which is always up. To do this select in the ‘Shoot in selected directions’ button.
(v) You can now select the direction that you want to fire the bullet in from the direction ‘Clock face’. You can select / un-select directions by clicking on the black buttons around the clock face. Make sure you have just one arrow pointing straight up. Note that if you have more than one direction showing, Games Factory will select out of those directions at random.
Now press all the ‘OK’ buttons to take you back to the Event Editor screen.
You should now have a complete Event line, so that when you press the <Shift> key, a missile is fired up the screen.

Save Your Game

Before you go any further, save your game so far.

So, click on the ‘File’ menu at the top of the screen, then select the ‘Save as...’ option.
Now under the ‘File name’ text box, enter the name you want to use. To keep things nicely ordered, enter the name ‘Space3’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘sound’, then ‘tutorial’, then ‘spacetu3’.

Adding sound effects

37. Now to add a little extra to the action, we are going to add a noise to the action of firing a missile.
(i) Right mouse click in the empty box beneath the ‘Sound’ icon on Event line number 7.
(ii) Now select the ‘Play sample’ option from the menu produced. You will then be taken to a file selector.
(iii) First, you have to select the correct drive to look in. If you have done a normal installation, all of Games Factory’s noises are kept on the CD. Make sure the drive is set to your CD drive. You can click on the button under the ‘Drives’ heading to pull down a list of the drives available.
(iv) Now look down the list of directories (use the scroll bars) until you find one called ‘samples’. Double-click on this with the left mouse button.
(v) Now look down this list until you find ‘weapons’. Double-click on it.
(vi) You will now have a list of all the noises contained within the directory ‘weapons’.
(vii) Scroll through the list of all the noises until you get to one called ‘phaser03.wav’. Select it once with the left mouse button.
(viii) Now press the ‘Play’ button on the right of the dialogue box. You can preview the noises using this button. You can always select another sample and preview that using the ‘Play’ button, or even move to another sound directory. Simply double click back on the ‘samples’ directory, then choose another sound directory.
(ix) When you are happy with the sound that you want to play every time a missile is fired, click the ‘OK’ button.

38. Create a Comment on line 8 containing the text “For every missile that exists in the game, always move them up 10 pixels at a time.” See step #26 for more information.

39. Create an Event ‘Always’ on line 9. To do this, right mouse select on the ‘New Condition’ text, then right mouse select on the ‘Special’ object, and from the ‘Always / Never’ option select the ‘Always’ option. See step #28 for more information on creating Events.

Moving an object by changing its coordinates

40. Now insert an Action on the ‘Always’ event line (number 9), beneath the ‘Missile’ icon. Select in the empty box with the right mouse button. From the menu produced select the ‘Position’ option.
(i) From this menu select the ‘Set Y coordinate’.
(ii) Now press the ‘Edit’ button. This will take you to another Expression Editor.
(iii) Now press the ‘Retrieve data from an object’ button.
(iv) Now right-mouse select the missile from the objects displayed.
(v) When you select the missile it will produce a menu. Select the ‘Position’ option.
(vi) From the menu that this produces, select the ‘Y coordinate’ option.
(vii) You will now have the text ‘Y(“MISSILE”)’ in the text box of the Expression Editor. Without changing this text, add on ‘-10’ to the end of the text, so that you have an expression that reads ‘Y(“MISSILE”)-10’. This means that every time Games Factory cycles through the Event Editor it will always subtract 10 from the Y coordinate of the Missile. Press the ‘OK’ buttons to take you back to the Event Editor.

This method of movement is actually somewhat redundant in this game, as we are already firing it at speed 50. It does show you that you can move objects around quite easily by changing their coordinates. Also, you may want an object to move faster than the maximum speed allowed by Games Factory normally, which is speed 100. To make it go faster than this, you would have it moving at maximum speed *and* manipulate its coordinates.

41. Now insert a Comment on Event line 10 that reads "When the missile leaves the top of the screen, it is not needed any more, so destroy it." See step # 26 for more information.

Testing the position of an Object

42. Now insert an Event on line 11 to test the position of the missile.
(i) Right click on the 'New condition' and select the missile from the dialogue box.
(ii) Select the 'Position-Test position of "Missile"' option.
(iii) You will now have a dialogue box that allows you to test the position of the missile in several different ways. Move the mouse pointer over the different buttons and have a look at all the options available.
(iv) Click on the button 'Is the object outside?'. Note that from here you could select more than one button. You can un-select a button by clicking on it again.
(v) When you have just got the button 'Is the object outside?' depressed, click the 'OK' button to take you back to the Event Editor.
You will now have an Event line that reads '(Missile) is out of the play area'.

43. Now insert an Action to destroy the missile, on the same Event line. See steps #31 and # 32.

44. Now insert the Comment "If a missile strikes another missile, destroy it." (Right mouse button on the number 12, select 'Insert-Comment'). See step # 26 for more information.

45. Now insert an Event 'Collision between (Missile) and (Missile)' on line 13.
(i) Right click on the 'New condition' text, right click on the Missile object, select the 'Collisions-Another object' option from the menu.
(ii) Select the Missile from the dialogue box with the left mouse button, then press 'OK'.
You will now have an Event line as described above.
So, if you are very fast on the keyboard and fire out missiles too fast, they will run into one another and destroy each other.

46. Now insert an Action to destroy the missile. Look at steps 43, 32 and 31 for more information, or copy the Action from one of the other Event lines by left mouse clicking and holding on one of the 'Destroy' tick-marks, then dragging it into the empty box beneath the Missile icon, on the Event line 'Collision between (Missile) and (Missile). You could add a sound here, see step # 37 above for more details.

Save Your Game

Before you go any further, save your game so far.

So, click on the ‘File’ menu at the top of the screen, then select the ‘Save as...’ option.
Now under the ‘File name’ text box, enter the name you want to use. To keep things nicely ordered, enter the name ‘Space4’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘01space’, then ‘tutorial’, then ‘spacetu4’.

47. Insert a comment “If the missile strikes any of the Aliens in the game, destroy both the missile AND the alien.” See step # 26 for more information. Make sure this is on line 14.

48. Now insert an Event on line 15, ‘Collision between (Missile) and (Big Alien)’. Right click on the ‘New condition’ text, then right click the Missile, select the ‘Collisions-Another object’ option. Now select the ‘Big Alien’ object. See step 45 for more information.

49. Insert an Event ‘Collision between (Missile) and (Alien Pawn) on line 16. See above for more information.

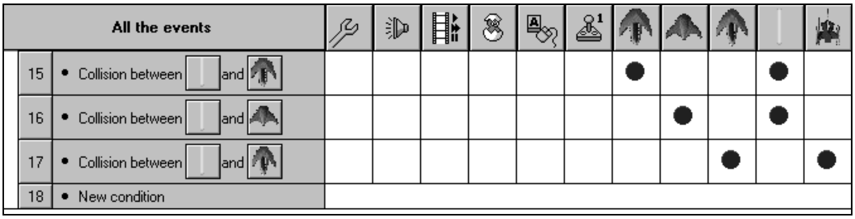
50. Insert an Event ‘Collision between (Missile) and (Tiny Alien) on line 17. See above for more information.

51. Now insert a sound action on line 15. Pick something appropriate for an explosion. (Right mouse in the empty box under the sound icon, select the ‘Play sample’ option, look on the CD ROM drive, look in ‘Samples-Impacts’. See step # 37 above.

52. Now copy the sound action into the lines 16 an 17. Left mouse click and hold on the tick-mark on line 15 and drag it into the empty sound boxes on line 16, then do the same for line 17. You will now have that sound being played every time a missile collides with any of the aliens.

53. Now insert an Action ‘Destroy’ into the empty box under the missile on line 15. (Right click in the box, select the ‘Destroy’ option.)
54. To save a lot of time, drag the ‘Destroy’ action, using the left mouse button, into all the empty boxes beneath the missile, for when the missile collides with an alien.

55. You can now drag the ‘Destroy’ action into the boxes under each alien. Make sure that it is the Big Alien that is destroyed when a missile collides with a Big Alien, and not one of the others! If you make a mistake you can select a check-mark and then press the <Delete> key.
So, on the line where the missile collides with the Alien Pawn, put the ‘Destroy’ action in the column directly under the Alien Pawn. Do the same for the Tiny Alien, on the line where the missile collides with Tiny Alien. The Events should look something like the screen shot below. Make sure that you are not accidentally destroying one of the other objects by mistake, that is, ensure the ‘Destroy’ actions are in the correct columns.



	All the events											
15	• Collision between and							●			●	
16	• Collision between and								●		●	
17	• Collision between and									●		●
18	• New condition											

Save Your Game

Before you go any further, save your game so far.

So, click on the ‘File’ menu at the top of the screen, then select the ‘Save as...’ option. Now under the ‘File name’ text box, enter the name you want to use. To keep things nicely ordered, enter the name ‘Space5’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘01space’, then ‘tutorial’, then ‘spacetu5’.

Changing the score

56. Now we want to add to the score every time one of the Missiles hits one of the Aliens. To change the score, right-click in the empty box underneath the Player 1 object, on line 15. Select the ‘Score’ option, then the ‘Add to score’ option.
You can now choose how much you are going to add to the score when a Big Alien is destroyed. To keep it the same as the Games Factory version, enter 100, using the keyboard or the slider controls to enter the numbers. Then press ‘OK’.

57. Do the same as the above for the Alien Pawn (add 10 to the score) and the Tiny Alien (add 50 to the score).
58. Now insert a Comment on Event line 18 that reads “Check the area above the player’s ship where the aliens move. When no more aliens are within this area, the player must have destroyed them and the game is complete.” See step # 26 for more information.

Comparing the number of objects in a zone to a number

59. Now we are going to insert an Event to end the game when all the aliens have been destroyed. To do this we are going to test if there are any objects in a zone on the play area.
(i) Right mouse click on the ‘New condition’ text of line 19.
(ii) Right mouse click on the ‘New Objects’ icon.
(iii) Select the ‘Compare to number of objects in a zone’ option. You will now be taken to a ‘Zone setup’ screen.
(iv) Select in the first box, ‘Horizontal’. Place a 0 in here. Now go to the next box on that line, ‘to’, click in there and type in 640.

(v) Now click in the first box on the next line down, ‘Vertical’. Enter a 0 in here.
(vi) Now enter the figure 320 in the last box, ‘to’. You should have the four boxes filled as follows- ”’0’ to ‘640’, then underneath, ‘0’ to ‘320’”. If you have done this successfully, click ‘OK’.
(vii) You will now be taken to a dialogue box where you can enter a number which you are comparing to the number of objects in the zone that you just defined.
In the top of the dialogue box are several radio-buttons, which is where you decide how you are going to compare the number. By default it is ‘Equal’. This is the comparison that we want to use, because when the number of objects is equal to zero, we want to end the game.
So, make sure the ‘Equal’ radio-button is filled.
(viii) Now you can enter the number you want to compare. In this case it is 0, so make sure the text box at the bottom of the dialogue box has a ‘0’ in it. You can change it by using the slider, or by selecting in the text box and entering the numbers directly.
When you are happy that you have the comparison set to ‘Equals 0’, press the ‘OK’ button.
You will now have an Event line that reads ‘Number of objects in zone (0,0) to (640,320) = 0’.

60. We are now going to insert an Action to finish the game. On line 19, go to the empty box underneath the ‘Storyboard controls’ icon.
(i) Right mouse click in here.
(ii) From the menu produced, select the ‘End the game’ option.

That’s it! You now have Spacebattle ready to play!

Save Your Game

Before you go any further, save your game so far.

So, click on the ‘File’ menu at the top of the screen, then select the ‘Save as...’ option.
Now under the ‘File name’ text box, enter the name you want to use. To keep things nicely ordered, enter the name ‘Space6’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘01space’, then ‘tutorial’, then ‘spacetu6’.

To give your version of Spacebattle a try, click on the ▶ icon. Click on it again to pause the action.

Tutorial Game 2

Pointblank!

It is assumed that before you embark upon this tutorial that you have successfully completed the first tutorial, and that you are comfortable with the conventions used in the first tutorial game. As before, we are going to take you on a 'guided tour' of Pointblank, so that you can see what we are going to be creating, then you will be taken on a step-by-step guide to creating your own version. Slightly less time will be spent explaining the exact mechanics of producing the game, as much of this was done in the first tutorial, but there will be more information on how certain steps can be used in your own games.

Loading Pointblank

Select the 'Open' option from the 'File' menu at the top of the screen, then look for the 'gametuts' directory on your CD ROM drive.
Within the 'gametuts' directory look for the '02pointb' sub-directory. Now look for the directory 'tutorial' and within this look for the file '1-pointb.gam'. Open this file.

Pointblank – The Storyboard Editor

When you have loaded Pointblank you will have a Storyboard Editor screen that looks very similar to the one for Spacebattle, and all the same options are available.
Before we move on to the Level Editor, though, we are going to have a look at some of the preferences available here.

Try clicking on the preferences icon P with the left mouse button.
You will be presented with a dialogue box which allows you to change the 'Density' of the display via the slider. Try moving it, then click 'OK' and see the difference it has made to the display. Click back on the preferences icon to change it again.
You can also change which information is included in the Level displays by clicking in the 'Display top bar' and 'Display comment(s)' tick points. Change these, then click the 'OK' button to see the changes.

When you are happy that you know how the preferences on the Storyboard Editor work, go to the Level Editor. Remember that the first time you load a game or start a new one, you must right select on the 'Thumbnail' of that level. Do this, select the 'Edit this level' option, then the 'Level Editor' option.
This will take you to the Level Editor of Pointblank.

Pointblank – The Level Editor

You can now look around the play area of Pointblank and see how it was created.
Move the mouse pointer over the different objects on the play area to see what they are via the Handy Hints.
When the game is running, all the gangsters are going to be hidden from view, then at random they are going to emerge from hiding, which is going to be the only time that you can shoot them, using the mouse to move the cross-hairs over them, then pressing the left mouse button to fire at them.
Normally, Active objects will appear in front of backdrop objects, so to hide the Gangsters properly, we are going to place an Active object, the ‘Mask’ object, in front of them. This prevents them from being shot whilst they are hiding. The ‘Mask’ fits exactly into the background, so it is not obvious when the game runs that there is another object on the screen.

Pointblank – The Event Editor

To enter the Event Editor, click on the ▦ icon on the toolbar at the top of the screen.

You will now be able to look through all of the Events used to make the game actions for Pointblank.
Scroll through the Events and move you mouse pointer over the check-marks in the boxes to see the Actions that are associated with the Events.

You can see that Groups are used extensively in this game. This actually makes it far easier to control the game, by deactivating and activating certain groups when the game starts and ends. For example, there is a group ‘Level Won’ that is inactive until the player wins the game, when it is activated, and all the others are deactivated.

You can ‘close’ a group by double-clicking on it. Try doing this to all the groups and you will be able to see how they can make it far easier to move around the Event grid.
You should be able to see all the groups on the screen at once. Double click on them to see all the Events that are contained within them.

Now that you have had a look around, have a play of the game before we go and re-create it. This time use the ‘Run Game’ button on the toolbar at the top of the screen.
If you want to exit the game before the game has finished, press <Alt> and <F4> together.

When you have done finished playing, click on the 🗋 icon on the toolbar. Select the ‘640 by 480’ play area size, then click ‘OK’.

Making Pointblank Yourself

Level Editor

To get into the Level Editor from the Storyboard Editor, click in the empty 'Thumbnail' with the right mouse button, select the 'Edit this level' then 'Level Editor' options.

1. Now, go to the Level Objects tool on the toolbar. Click on the button at the side of this to pull down a list of all the object libraries.
Select the library called '(Tutorial) POINTBLANK'
You can now see all the objects used to create Pointblank in the object window down the left hand side of the screen.

2. First of all we are going to place the backdrop object onto the screen. Select it then place it anywhere in the play area. Because it is exactly the same size as the play area it will align itself automatically.

3. Now select on the Gangster, then, using the right mouse button, place copies of him in the window part of the door on the bottom, with his waist at exactly the same height as the bottom of the window edge.

4. Place another copy of him, using the right button, in the window at the top left, again making sure that his waist is at window height.

5. Place another in the third window from the left, using the right mouse button.

6. Place the fourth and fifth in the window at the top right, using the right mouse button, and place the sixth behind the low wall on the very far left of the play area, using the right mouse button.

7. Put the last in the large window on the ground floor, using the left mouse button.
You should now have a gangster in every window except the one that is second from the left on the top floor.

8. Now you can place the 'Mask' object in front of the gangsters. Be careful to line it up with the backdrop perfectly. It should cover up the legs of the gangsters at their waist level. The mask is to stop you being able to shoot the gangsters when they are hiding, because they will be 'behind' an Active object, the 'Mask'.

Changing the view of objects

You could have placed the mask on the screen before the gangsters, then placed the gangsters over the top of the mask. If you had it this way, the gangsters legs would not be hidden by the mask.
To change this you could have changed the 'View' of either the gangsters, or of the mask. It would

be quicker to change the image of the mask than to change the view of every gangster.
(i) To do this, you would right select on the ‘Mask’ object. This will produce a menu.
(ii) Select the ‘View’ option.
(iii) Now select the ‘To front’ option. This will place the ‘Mask’ object ‘in front’ of all the other objects on the screen.

9. Now we are going to place the object that we are going to use as the cross-hairs of our gun onto the play area. Go to the object window and select the ‘Cross Hair’ object. Place it anywhere on the screen. It does not matter where you place it as we are going to hide them at the very beginning of the level, before giving it the same coordinates as the mouse pointer, then making it visible.

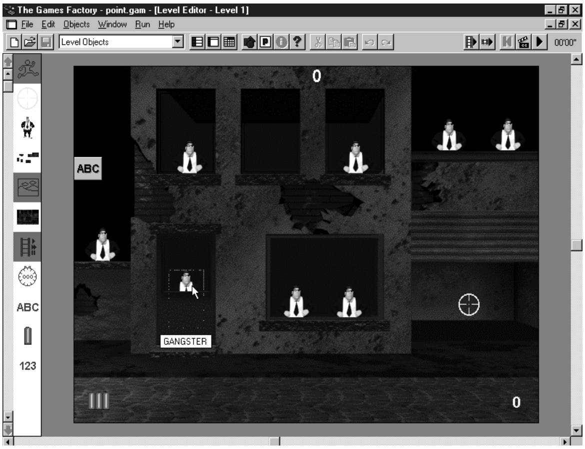
10. Now select the ‘Countdown’ object and place this on the play area in a place that you will be able to see readily, but not so as to interfere with the gangsters.

11. Now place the ‘Good Shooting’ object on the far left of the play area, at about the same height as the waist of the gangster in the far left window, just outside the wall of the building.

12. Place the ‘Live Ammo’ on the bottom left of the play area. This is the display for the number of bullets that you have left.

13. Now place the ‘Score’ object on the play area near the ‘Countdown’ object. This will display your score.

You should now have a screen that looks something like this;



14. Before we proceed, we are going to change the game setup. It is important to do this as some of the objects that we are using need to have their initial values changed. We also want to hide the mouse pointer and the Menu bar that is normally displayed when you do a full run of your game using the ‘Run Game’ button.
(i) Select the ‘File’ menu heading at the top of the screen.
(ii) Select the ‘Game setup’ option from this menu.
(iii) On the first ‘layer’ of the dialogue box you can enter a name for your game in the ‘Title’ text box, then your name underneath in the ‘Author’ text box.
(iv) Now go to the ‘Window’ layer by selecting on its tab at the top of the dialogue box.
(v) Now make sure that all of the tick-boxes are un-selected, i.e. that they are all empty. This will stop the menu bar and heading being displayed when you run your game.
(vi) When you have done that, select the Players tab. In the ‘Lives’ box at the bottom right, change the ‘Initial’ setting to 15. You can enter the numbers in directly by selecting in the text box, then using the <Delete> key to clear the box, then enter 15.
(vii) Now change the ‘Maximum’ value to 15 also.
(viii) Now click the ‘OK’ button.

If you are satisfied that you have placed all the game objects onto the play area the way you want, go to the Event Editor. Select its icon from the toolbar at the top of the screen to do this.

Pointblank–The Event Editor

Before you go any further, save your game so far. We are going to save the game every 10 steps, so if you encounter a problem you can always revert to a previous version of the game and try again.

So, click on the ‘File’ menu at the top of the screen, and then select the ‘Save as...’ option.
Make sure that the drive is set to your hard drive.
Double click on the ‘gfactory’ directory. This means your game will be placed in that directory.
Now under the ‘File name’ text box, enter the name you want to use. To keep things nicely ordered, enter the name ‘point-1’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘02pointb’, then ‘tutorial’, then ‘pointut1’.

We are now going to insert all of the Events and Actions to bring your game to life, add sound effects, a timer countdown, score, bullets counter and an end message.

15. The first thing we are going to do is insert an Event ‘Start of level’, then put in some actions.
Right click on the ‘New condition’ text of Event line number 1.
Right click on the ‘Storyboard’ icon, then select the ‘Start of level’ option from it’s menu.

Playing Music

16. Next we are going to insert an Action to play some music at the start of the level. We are also going to add four actions to deactivate groups of actions, but as we have not yet created these groups, The Games Factory will not let us deactivate them yet.

(i) Right mouse click in the empty box directly beneath the ‘Sound’ icon. Select the ‘Play and loop music’ option. You will then be presented with a file selector. If you have done a normal installation of The Games Factory, the sound files will be on your CD ROM drive. If you did a full installation, then all the files will be on the internal hard drive of your computer.

You are looking for a directory called ‘Midi’, and then within this you are looking for a file called ‘melripp1’. When you have selected this, click the ‘OK’ button. You will now be asked to enter the number of times you want the music to loop, which is 99 times. Either enter these numbers in directly by selecting within the text box, or click on the slider buttons. When you have entered the number 99, click on the ‘OK’ button.

You will now have a check-mark under the sound icon. If you move your mouse pointer over it will read ‘Play music melripp1 99 times’.

Inserting a Group

17. Now we are going to create a group called ‘Start of Game Actions’. Right click on the number 2, select ‘Insert’ then ‘A Group of events’. Type in ‘Start of Game Actions’, then click ‘OK’.

Inserting a Comment

18. We are now going to place Events within this group. Make sure that each Event line that you insert a new condition on is indented from the left. This denotes that specific Event line as being within the group directly above.

Insert a Comment on line number 3 that reads ‘When each game starts, make sure the player gets 15 bullets. As the player will lose if all the bullets are used, it is fitting we should use the players LIVES COUNT to store our bullet count.’

To insert a comment, right click on the number of the Event line, then select the ‘Insert’ option, then the ‘A Comment’ option. To make the comments stand out, you can change the text and background colour.

Inserting an Event

19. Insert an Event ‘Always’ on line number 4. Right click on the ‘New condition’ text, right select the Special object, then from the ‘Always / Never’ option, select the ‘Always’ option.

Setting Lives

20. Now insert the Action ‘Set number of lives to 15’. Right click in the box under the Player 1 object, on line 4. Select the ‘Number of lives’ option, then the ‘Set number of lives’ option. You can now enter the number in the text box, or use the slider. When it is set to 15, click ‘OK’.

Comment

21. Insert a comment ‘When the game starts, the mouse pointer and cross hairs should not be

seen. These will eventually be made visible when the level starts.' on line 5. See step 18 above for information on how to create a Comment.

Inserting an 'Always' Event

22. Insert an Event 'Always' on line 6. (Right click on 'New condition', right click the 'Special' object, select the 'Always' option.)

Hiding the Mouse pointer

23. On the same line insert the action 'Hide windows mouse pointer'. To do this right mouse click in the empty box beneath the 'Mouse pointer and Keyboard' icon and then select the 'Hide windows mouse pointer' option.

Making an object invisible

24. Insert another action on line 6 to make the cross hairs invisible. To do this right click in the box under the 'Cross Hairs' object and select the 'Visibility' option, then the 'Make object invisible' option.

Inserting an 'Always' Event

25. Insert an Event 'Always' on line 7. We are going to activate and deactivate some of the groups here, but as explained earlier, you cannot do this until you have created those groups.

Save Your Game

Before you go any further, save your game so far.

So, click on the 'File' menu at the top of the screen, and then select the 'Save as...' option. Now under the 'File name' text box, enter the name you want to use. To keep things nicely ordered, enter the name 'point-2', then click 'Save'.

You can also load up The Games Factory version of the game so far. Look in the directory 'gametuts', then for the sub-directory '02pointb', then 'tutorial', then 'pointut2'.

Inserting a Group of Events

26. Insert a group of events called 'Start of Level Actions' on line 9. Note that you must create this group on a line that is not indented. When you have done this line 8 should be empty, and still within the group 'Start of Game Actions'.

Inserting a Comment

27. Now insert a comment on line 10, 'The level waits for 1.5 seconds before hiding all the gangsters, making visible the cross-hairs and activating the main game loop. In order to create some urgency, the time limit is set to 30 seconds.' (Right click on the number, select the 'Insert-A Comment' options. Enter the text, select 'OK'.

Reading the Timer

28. Now insert an Event on line 11, ‘Timer is greater than 1”-50’. To do this right click on the ‘New condition’ text and right select on the ‘Timer’ object. Select the ‘Is the timer greater than a certain value?’ option. Now you can enter in the time. By default the value is 1 second, so you only need to change the ‘1/100’ box. Select in it and enter the number 50. Now select ‘OK’.

Changing an animation sequence

29. Insert an Action ‘Change animation to hiding’ under the Gangsters icon on line 11. Right click in the box, select the ‘Animation’ option, then the ‘Change’ option, then the ‘Animation sequence’ option.

From the dialogue box that this produces, select the ‘Hiding’ option. Now click on ‘OK’.

Making the Crosshairs Reappear

30. Insert another Action on line 11, ‘Reappear’ under the Cross Hairs icon. Right click in the box and then from the menu select the ‘Visibility’ option, then the ‘Make object reappear’ option.

Then insert an Action to set the counter to 30. Right click in the empty box under the ‘Countdown’ object on line 11, then select the ‘Set counter’ option. You will now be presented with a dialogue box where you can either enter the numbers directly in the text box, or use the buttons on the slider to change the number. Make sure that the value entered is 30, then click ‘OK’.

Inserting a Group

31. Insert a group called ‘Main game’. Note that you must do this on line 13, which is outside of the group ‘Start of Level Actions’. If you used line 12, which is indented, then you would put a group within a group, which Games Factory will not allow.

To insert a group, right click on the number of the line (13), select the ‘Insert-A group of events’ options, type in the text, click ‘OK’.

Inserting a Comment

32. Insert a comment on line 14 that reads ‘The main game group controls all the events that occur within the game itself.’ (Right click on the number, ‘Insert-A comment’, type in the text, ‘OK’.

Inserting a Comment

33. Insert another comment, preferably in a different colour, ‘The CROSS HAIRS must follow the mouse pointer throughout the game. Even though the mouse pointer has been hidden, it still moves about the screen as though it were visible. By taking the mouse pointers X and Y coordinates, we can make the cross hairs follow the movement of the mouse pointer.’

Inserting an ‘Always’ Event

34. Insert an Event ‘Always’ on the next Event line. (Number 16.)

Making an object follow the mouse pointer

35. Insert an Action on line 16 to set the coordinates of the cross hairs to be the same as the mouse.
(i) Right click on the empty box under the Cross Hairs object, on line 16.
(ii) Select the ‘Position’ option.
(iii) Select the ‘Set X coordinate...’ option.
(iv) Now press the ‘Edit’ button.
(v) Now press the ‘Retrieve data from an object’ button.
(vi) Now right mouse select the ‘Mouse pointer and Keyboard’ object.
(vii) Now select the ‘Current X position of the mouse’.
You should now have an action under the Cross Hairs object that reads ‘Set position to Xmouse’.
To set the Y coordinate to be the same as the mouse, do the following;
Right click on the filled box under the Cross Hairs object, on line 16.
(ii) Select the ‘Position’ option.
(iii) Select the ‘Set Y coordinate...’ option.
(iv) Now press the ‘Edit’ button.
(v) Now press the ‘Retrieve data from an object’ button.
(vi) Now right mouse select the ‘Mouse pointer and Keyboard’ object.
(vii) Now select the ‘Current Y position of the mouse’, then click ‘OK’.
This will set the Cross Hairs object to follow the movement of the mouse pointer exactly.

Save Your Game

Before you go any further, save your game so far.

So, click on the ‘File’ menu at the top of the screen, and then select the ‘Save as...’ option. Now under the ‘File name’ text box, enter the name you want to use. To keep things nicely ordered, enter the name ‘point-3’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘02pointb’, then ‘tutorial’, then ‘pointut3’.

Inserting a Comment

36. Insert a Comment on line 17, ‘Every second, deduct 1 from the counter that holds the time limit.’

Testing the Timer

37. Now insert an Event ‘Every 1”-00’. Right click on the ‘New Condition’ text of line number 18, then right select the Timer object, then select the ‘Every’ option. Make sure the time entered is 1 second, then click the ‘OK’ button.

Changing the value of the counter

38. Insert the Action ‘Subtract 1 from Counter’. Right click in the box under the ‘Countdown’ object, select the option ‘Subtract from Counter’. Now enter the number 1 in the text box of the Expression Editor produced, then click ‘OK’.

Inserting a Comment

39. Insert a Comment on line 19, ‘Every 2 seconds, select ONE of the many GANGSTER objects from the game and if that gangster is currently hiding, run its APPEAR FROM HIDING animation.’ (Right click on the number, select the ‘Insert-A Comment’ options, etc.,.)

Testing the Timer Every 2 seconds

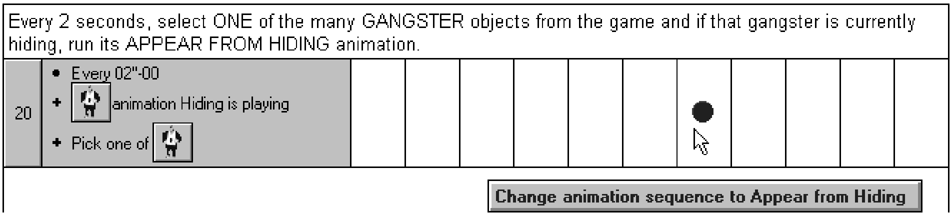
40. Insert an Event, ‘Every 2”-00’. Right click on the ‘New Condition’ text of line number 20, then right select the Timer object, then select the ‘Every’ option. Make sure the time entered is 2 seconds, then click the ‘OK’ button.

Adding a Condition to a Timer Event

41. Now add a Condition to the above Event. To do this, right select on the text ‘Every 2”-00’ and select the ‘Insert’ option.
Now right select the Gangster object, and select the ‘Pick or Count’ option, then the ‘Pick “GANGSTER” at random’ option.

Adding a Further Condition

42. Now add a further Condition to line number 20, ‘“GANGSTER” animation Hiding is playing’. To do this, right Click on any of the texts of line 20, select the ‘Insert’ option.
Now right select the Gangster, then select the ‘Animation’ option. Now select the ‘Which animation of “GANGSTER” is playing’ option.
Now select the ‘Hiding’ animation from the list in the dialogue box. Then click the ‘OK’ box.
You will now have three Conditions all on one Event line. You should move the Event ‘Every 2”-00’ to the top of this Event line, so that The Games Factory will only go through the rest of the Conditions once every 2 seconds. This will help to speed up your game play slightly, especially if you had many Event lines like this.



Every 2 seconds, select ONE of the many GANGSTER objects from the game and if that gangster is currently hiding, run its APPEAR FROM HIDING animation.

20
• Every 02"-00
+ animation Hiding is playing
+ Pick one of
●

If you, for example, you had placed the Condition ‘Pick one of ‘GANGSTER’ at the top, then The Games Factory would go through the actions of picking a gangster before looking at the other Conditions.

Changing an Animation sequence

43. Now insert an Action into line number 20, ‘Change animation sequence to Appear from Hiding.’ To do this, right click in the box under the Gangster object on line 20, then select the ‘Animation’ option, then the ‘Change’ option, then the ‘Animation sequence...’ option.
From the dialogue box produced select the ‘Appear from Hiding’ animation, then click ‘OK’.

Inserting a Comment

44. Now insert a Comment on line 21, ‘Four times a second, select ONE of the many GANGSTER objects from the game and if that gangster is currently hiding, run its APPEAR FROM HIDING animation.’ (Right click on the number 21, select the ‘Insert-A Comment’ options, etc.)

Testing the Timer

45. Now insert an Event on line 22, ‘Every 00”-25’. To do this right click on the ‘New condition’ text, then right select the Timer object, select the ‘Every’ option from its menu. Make sure that the time entered is 25 hundredths of a second, then press ‘OK’.

Save Your Game

Before you go any further, save your game so far.

So, click on the ‘File’ menu at the top of the screen, and then select the ‘Save as...’ option.
Now under the ‘File name’ text box, enter the name you want to use. To keep things nicely ordered, enter the name ‘point-4’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘02pointb’, then ‘tutorial’, then ‘pointut4’.

Picking a Gangster at random

46. Now insert a Condition to line 22, ‘Pick one of (Gangster). To do this right select on the text ‘Every 00”25’ of line 22, select the ‘Insert’ option, then right select the Gangster object, and select the ‘Pick or count’ option from its menu, then the ‘Pick “GANGSTER” at random’ option.

Testing which animation is playing

47. Now insert a further Condition to line 22, ‘“GANGSTER” animation Stopped is playing’. To do this, right select on either of the Condition texts on line 22, select the ‘Insert’ option, then right select the Gangster object, from its menu select the ‘Animation option’.
Now select the ‘Which animation of “GANGSTER” is playing?’ option. From the dialogue box produced select the ‘Stopped’ animation, then click ‘OK’.

Changing the Animation sequence

48. You can now insert an Action into line 22 to hide the Gangster that has been picked. Right click in the empty box on line 22, underneath the Gangster object. From its menu select the ‘Animation’ option, then the ‘Change option, then the ‘Animation sequence’ option. From the dialogue box select the ‘Hiding’ animation, then click ‘OK’.

Inserting a Comment

49. Insert a Comment on line 23, ‘If the player manages to click the mouse pointer (appearing as a cross-hair) on a gangster, destroy the gangster and add 10 points to the score.’

Testing the mouse to see if it is clicked on a Gangster

50. Now insert an Event on line 24, ‘User clicks with left mouse button on “GANGSTER”. To do this, right click on the ‘New condition’ text of line 24, then right select the ‘Mouse pointer and keyboard’ object. Select the ‘The mouse’ option, then the ‘User clicks on an object’. You will now be taken to a dialogue box. Make sure that there are tick marks in the ‘Left button’ and ‘Single click’ radio-buttons, then press ‘OK’.
Now select the Gangster object, then click ‘OK’.

Inserting Sound

51. Now insert 2 sound actions on line 24. Right select in the empty box under the ‘Sound’ object, select the ‘Play sample’ option. You will now be presented with a file selector. If you have done a normal installation, the sound files are on your CD ROM drive, on the Games Factory CD. Make sure the ‘Drives’ is set to your CD ROM drive.
Now look for the directory ‘samples’.
Within ‘samples’, look for the sub-directory ‘weapons’. It is from here that we are going to select the noise for the gun being fired.
Try the file ‘gun3.wav’. You can preview the sound by press the ‘Play’ button. Then click the ‘Open’ button.
Now insert a second noise, the scream of the Gangster.
To insert a further Action in the same check-box, left click in the box you want to add an Action to. So, left click under the ‘Sound’ object again, then right select the ‘New action’ option. Select the ‘Play sample’ option.
Now, look again in ‘samples’, but this time look in the sub-directory ‘human’. Look for the file ‘screamm1.wav’. Try playing it a few times. If you are happy with your selection, click the ‘OK’ boxes until you are returned to the Event Editor.

Changing the Score

52. You are now going to change the score at the same time as you destroy a gangster, so that every time the user clicks on Gangster, not only are you going to play a gun sample and a scream sample, but you add to the score, and destroy the gangster.
To do this, right click in the empty check-box underneath the ‘Player 1’ object. From the menu produced, select the ‘Score’ option, then the ‘Add to score’ option. Here you can enter the score in via the keyboard by clicking within the text box, or you can click on the slider. You can add any score you like, but we are going to add 10 to the score. When you are happy that you have the score entered correctly, click on the ‘OK’ button.

Destroying a Gangster

52.b. Now right select in the empty check-box underneath the Gangster object. From the menu produced select the ‘Destroy’ option.

Deducting a life (bullet)

53. Now, add an Action to deduct a life from Player 1. This means that Player one has used a bullet when he hits a Gangster. To do this, left click in the box underneath the Player 1 object, where you just inserted the action to add to their score.
Right select the ‘New action’ option, then select the ‘Number of lives’ option, then the ‘Subtract from Number of Lives’ option. Now enter the number 1 into the text box, or use the slider buttons to change the value. When you have correctly set this value to 1, click the ‘OK’ buttons to return you to the Event Editor.

Inserting a Comment

54. Insert a Comment on line 25, ‘When the player presses the left button and misses, deduct 1 life and play the sample ‘gun04.wav’. The sound condition makes sure that these actions are not repeated very rapidly.’

Testing the mouse to see if the left button has been clicked

55. Now insert an Event on line 26 to check whether the user has clicked the left mouse button. We are then going to add a condition to this, so the Actions associated with this line will only take place if a sound is not playing, i.e. if the player has not hit a Gangster.
So, right select the ‘New condition’ text of line 26 and select the ‘Mouse pointer and keyboard’ object, then select the ‘The mouse’ option, then the ‘User clicks’ option. Then make sure that the ‘Left button’ and ‘Single click’ radio-buttons are filled, then click ‘OK’.

Save Your Game

Before you go any further, save your game so far.

So, click on the ‘File’ menu at the top of the screen, and then select the ‘Save as...’ option.
Now under the ‘File name’ text box, enter the name you want to use. To keep things nicely ordered, enter the name ‘point-5’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘02pointb’, then ‘tutorial’, then ‘pointut5’.

Testing to see if no samples are already being played.
That is, to make sure that the user has not already just killed a Gangster

56. Insert a Condition to line 26 to check that no samples are being played. To do this right select on the text of line 26 and select the ‘Insert’ option. Then right select on the ‘Sound’ object. Choose the ‘Samples’ option, then select the ‘Is a sample not playing?’ option.

Playing a sample and deducting a life

57. Now you can insert Actions into line 26 to play a sample and to deduct 1 from Player 1’s lives, i.e. one bullet.

First, right click in the empty box under the ‘Sound’ object and select the ‘Play sample’ option. Look in the directory ‘samples’ then in the sub-directory ‘weapons’. From here select the sample ‘gun04.wav’.

Now, right click in the empty box under the ‘Player 1’ object and select the ‘Lives’ option, then the ‘Subtract from number of lives’ option. Set the Expression Editor value to 1, then click ‘OK’.

Inserting a Comment

58. Insert a Comment on line 27, ‘If ALL the gangsters in the building have been destroyed, the player has completed the level. Reveal the ‘Good Shooting’ message and activate the ‘Level Won’ group.

Testing to see if all the Gangsters have been shot

59. Insert the Event ‘Last “GANGSTER” has been destroyed’ on line 28. Right select the ‘new condition’ text, then right select the Gangster object, then select the ‘pick or count’ option, then select the ‘Have all “GANGSTER” been destroyed?” option.

Playing some ‘Victory’ music and displaying the text “Good Shooting”

60. As previously outlined, we cannot activate or deactivate a group until it has been created, so we will insert the Action to activate the ‘Level Won’ group after it has been created.

For now you can insert several Actions on line 28. First, insert an action to play some music.

Right select under the ‘Sound’ object, select the ‘Play Music’ option, then look in the ‘midi’ directory and select the ‘mrcool.mid’ file.

Now insert an Action to display the text ‘Good Shooting’.

Right select under the object ‘Good Shooting’, then select the ‘Display text...’ option.

You will now be presented with a position selector. Enter 0 into the X coordinate box and 64 into the Y coordinate box, then click ‘OK’.

Inserting a Comment

61. Now insert a comment onto line 29, ‘There are two ways in which the player can lose the game. The bullets could run out or the time limit could have expired. In each case, the Game Lost group must be activated.’

62. Insert an Event on line 30, right click on the text ‘New condition’, then select the ‘Player 1 object’, then select the ‘Compare to players number of lives’ option. Now, make sure that the value is 0, and that the ‘Equals’ radio-button is filled. Now click then ‘OK’ button.

Inserting sound

63. Now insert a ‘Sound’ action on line 29. Play a piece of music that you feel is appropriate to losing the game.

Comparing the value of the Countdown object to a number

64. Insert an Event on line 31. Click on the line’s text, then select the ‘Countdown’ object. Then select the ‘Compare the counter to a value’ option. Make sure the value it is being compared to is 0, then make sure that the ‘Equals’ radio-button is filled.

Copying an Action

65. Now left click and drag the music Action from line 30 down into line 31. This copies that Sound Action onto line 31.

Save Your Game

Before you go any further, save your game so far.

So, click on the ‘File’ menu at the top of the screen, and then select the ‘Save as...’ option. Now under the ‘File name’ text box, enter the name you want to use. To keep things nicely ordered, enter the name ‘point-6’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘02pointb’, then ‘tutorial’, then ‘pointut6’.

Inserting a Group

66. On line 33, insert a group ‘Level Won’. Note that this must go on this line so that it falls outside of the group “Main Game”. If you were to place it on line 32, The Games Factory would be unable to run your game.

Inserting a Comment

67. On line 34, insert a Comment, ‘When the congratulations music has finished, RESTART THE LEVEL which keeps the score and allows the player to play again with all the gangsters back as they were at the start of the level.’

Testing to see if no music is playing

68. Insert an Event on line 35. Right select the line, then select the ‘Sound’ object, then the ‘Musics’ option, then the ‘Is a music not playing?” option.

Restarting the current level

69. Now insert an Action on line 35. Right select under the ‘Storyboard’ object, then select the ‘Restart the current level’ option.

Inserting a Group

70. Now insert a group, ‘Game Lost’, on line 37.

Inserting a Comment

71. Now insert a Comment on line 38, 'When the game lost sound has ended, RESTART THE GAME to clear the score and begin the entire game from the very start.'

Copying an Event

72. Copy an Event on to line 39. Left mouse click and hold on the 'No sample is playing' text of line 35 and drag it into line 39. This will copy that Event.

Restarting the game

73. Now insert an Action on line 39 to restart the game. Right click in the box under the 'Storyboard' object, then select the 'Restart game' option.

Save Your Game

Before you go any further, save your game so far.

So, click on the 'File' menu at the top of the screen, and then select the 'Save as...' option. Now under the 'File name' text box, enter the name you want to use. To keep things nicely ordered, enter the name 'point-7', then click 'Save'.

You can also load up The Games Factory version of the game so far. Look in the directory 'gametuts', then for the sub-directory '02pointb', then 'tutorial', then 'pointut7'.

Inserting all the Actions to Activate / Deactivate the groups

74. That's almost it. All we need to do now is go back and insert all the Actions to activate / deactivate the groups, now that we have created them all.
So, go to line 1, right click in the box under the 'Special' object, select the 'Group of events' option, then the 'Deactivate' option. You will then be taken to a dialogue box where you can select the groups that you want to deactivate. First of all pick the 'Start of level actions' group, then click 'OK'. Now you need to insert the rest of the actions into the same check-mark. To do this, left click on the check-mark. This will produce a dialogue box which lists all the actions from within that check-mark. You also have a 'New action'. Right select on this. You will see the same menu produced as when you first right selected under the 'Special' object. Select the 'Group of events' option, then the 'Deactivate' option. Now select the group 'Main Game'.
You will now be able to deactivate the groups 'Level won' and 'Game Lost' using the method above.

Now go to line 7 and insert the Actions, 'Deactivate group', 'Start of game actions' and 'Activate group', 'Start of level actions'. So, right click under the 'Special' object, select 'Group of events', then 'Deactivate', then "Start of game actions", then 'OK'. Then left click the check-mark, then right click 'New actions', then 'Group of events', then 'Activate', then "Start of level actions", then 'OK'.

Now go to line 11 and insert the Actions ‘Deactivate group “Start of level actions”’ and ‘Activate group “Main Game”’.

Go to line 28 and insert the Actions ‘Deactivate group “Main Game”’ and ‘Activate group “Level Won”’.

Go to line 30 and insert the Actions ‘Deactivate group “Main Game”’ and ‘“Activate group “Game Lost”’. Now copy these actions into line 31 by left click and holding on the check-mark, then dragging it into line 31.

There, that really is it, and your own version of ‘Pointblank’ should be ready to play!

Save Your Game

Don’t forget to save your work before you carry on. Save it as ‘Pointfin’.

Hit the ‘Run Game’ button, and have a good play! Note that the menu bar is no longer displayed, so to quit the game you have to use the keyboard shortcut <Alt> + <F4>.

Have fun!

Tutorial Game 3

Pulverise

‘Pulverise’ is a version of the arcade classic ‘Asteroids’. This tutorial is divided into two parts, one to deal with the movement of the ship, and another to deal with the creation of the asteroids.

So, let’s go and load up ‘Pulverise’ version one.

Select the ‘Open’ option from the ‘File’ menu at the top of the screen, then look for the ‘gametuts’ directory on your CD ROM drive.
Within the ‘gametuts’ directory look for the ‘03pulver’ sub-directory. Now look for the directory ‘tutorial’ and within this look for the file ‘1-pulver.gam’. Open this file.

When you are presented with the Storyboard Editor for this screen, before proceeding into the other Editor screens, have a look in the ‘Game setup’ option. To do this go into the ‘File’ menu at the top of the screen, select the ‘Game setup’ option, then select the ‘Windows’ layer of this dialogue box. Notice which options are enabled and which are disabled.

Now go to the Level Editor. To do this, right select within the ‘Thumbnail’ of the level 1 display, then choose the ‘Edit this level’ option, then the ‘Level editor’ option.

Pulverise part 1 – The Level Editor

Right then, let’s have a look around the Level Editor.

We are mainly going to have a look at the ‘Pulvercraft’, which is the players spaceship. It has been manipulated in several ways from the Level Editor. We shall have a tour around these different changes, with an explanation as to why this was done.

Movement of the Pulvercraft

First, let’s have a look at the movement of the pulvercaft.
Right click on the Pulvercraft. This will produce a menu. Select ‘Movement’.
From the ‘Movement’ menu select the ‘Edit movement’. Do not use the ‘Change movement’ option.

Now you can see all the settings that were used to move the craft around. As you can see if you look at the title of the dialogue box, it has a ‘Race car’ type movement. this means that the left and right keys will rotate the object, and the forward and reverse will accelerate / brake or reverse the

movement of the object.
You can see that the speed has been set quite low; if it were too high then the craft would be difficult to control. The acceleration is set high, making it fast to get out of the way of careering asteroids, but the deceleration is quite low, making it drift fairly realistically. Having this set to zero would give you an endless drifting motion. You can change this if you want to make the craft more difficult to control.

Now have a look on the 'Rotation' layer of the dialogue box. You can see that the number of different directions is set to 32, making for very smooth-looking rotation. You must remember to add directions to the animations if the number of different directions that the object can move in is higher than the number of different animation directions. We will discuss how this was done in a moment.
The 'Rotation speed' is set quite high, meaning that the turning rate of the craft is quick.

Now look at the layer 'Initial direction'. This simply decides which direction the craft will point in initially.

Animation of the Pulvercraft

When you have finished looking around the movement of the Pulvercraft, click the 'Cancel' button to leave the settings as they were, or 'OK' if you want any changes you have made to be imposed on the Pulvercraft.

Now, select the 'Edit Animations...' option from the Pulvercraft's menu.

This will take you to the 'Animation Editor' of the Pulvercraft. Initially, the Pulvercraft only had one direction, and therefore only one animation associated with it.
To make it move in 32 directions, the 'Movement' menu option was changed, as outlined above. This, however, would result in the 'craft moving in 32 directions, but always looking like it was pointing up! Not good!
So, to have 32 different pictures, one for every direction, the Animation Editor was used.
The first step was to create 32 directions in the Animation Editor. To do this, the radio-button '32' was filled under the heading 'Number of different directions'. This would then create 31 empty frames, ready to be filled.

To quickly and easily fill these frames with 32 different directions, the ↻ button was used, which automatically rotates your initial frame by just the right amount to fill all the different direction frames. Easy!

Now let's go and have a look at how we put the Pulvercraft into an Object group. Not to be confused with Event groups, Object groups allow you to greatly simplify your Event Editing. For example, in Spacebattle, instead of having a separate Event line for each alien to be destroyed when a bullet hit it, it would be far easier to put all the aliens in a group called 'Baddies', and then have an Event line that checks to see if the group 'Baddies' has collided with a bullet instead.

So, ‘OK out of the Animation Editor, and select the ‘Object preferences...’ from the Pulvercraft’s menu. The first layer allows you to add the Pulvercraft to a group by using the ‘Add’ button. You can see all the groups that an object belongs to in the window. If you move your mouse pointer over the icon in the window, you can see that it is in the group ‘Neutral’. You could delete this by selecting it and then clicking on the ‘Delete’ button. When you have finished looking around the Preferences dialogue box, click the ‘Cancel’ button to exit without making any changes.

Now let’s go and have a look in the Event Editor.

Pulverise part 1 – The Event Editor

We are only going to have a brief tour around the Event Editor before trying to reproduce it for ourselves. Notice that all the main groups for the game have already been produced, meaning that they can be activated or deactivated already. When it is time for you to make this game for yourself, we are going to make the groups first, then place the Events in afterwards. We shall not be using line numbers anymore, but using group names and asking you to place Events and Actions in certain groups. Remember, to place an Event into a group, just right click on the indented ‘New condition’ line at the bottom of each group.

Have a look around and notice how the movement of the craft is started, and that already the group neutral is going to be ‘Wrapped’ around the play area when it leaves it. All the boulders are going to be in the group neutral as well, so you can see already that it only takes one Event line to wrap all the objects in a group around the screen, instead of testing each object in turn, and having separate ‘Wrap’ actions for each one.

Notice also that the group ‘Get ready’ turns itself off as soon as it sets the ship in motion, which will prevent this from happening again in the game.

There are also actions to destroy the missile when it goes off the screen, not to ‘Wrap’ that as well.

Now that you have had a brief tour around, run the game (don’t forget to use <Alt> + <F4> to escape) and see how it all fits together. Use the cursor keys to control the rotation and acceleration / deceleration of the ship, and the <Ctrl> key to fire.
Notice that there are no asteroids yet, just the ship. The second part of the tutorial controls the movement of the asteroids, explosion, etc.
When you have finished playing with part one, load up part two of this tutorial.

Select the ‘Open’ option from the ‘File’ menu at the top of the screen, then look for the ‘gametuts’ directory on your CD ROM drive.
Within the ‘gametuts’ directory look for the ‘03pulver’ sub-directory. Now look for the directory ‘tutorial’ and within this look for the file ‘2-pulver.gam’. Open this file.

Pulverise part 2 – The Level Editor

Now you can see all the different objects in position as they will appear when you first start playing the game. Notice, however, that you cannot see the 'BULLET' object, or the 'BOULDER(TINY)' on the play area, and yet they are on the 'Level objects' library. This is because they have been 'created' via the Event Editor. The bullet was created by having the Pulvercraft 'Shoot an object..' and then using the option from this, to look in other libraries. The bullet was picked from another library, and then Games Factory automatically placed it on the 'Level Objects' library.
The tiny boulder was created in a similar way, but using the 'Create objects' action, again in the Event Editor.
Now go and have a look in the Event Editor to see how this looks.

Pulverise part 2 – The Event Editor

Now have a look around here at all the different Actions. Let's start at line 16. Here you can see how the levels are going to get increasingly harder, the higher the score, the faster the asteroids are going to be moving.

Now look at line 23. Here you can see that all the asteroids have been put into the group 'Bad' from their 'Object preferences..' option in the Level Editor. (Right click on them in the Level Editor to show this option.) Instead of testing each asteroid individually, the group is tested to see if it collides with the Pulvercraft, meaning only one Event line is used, instead of one for every type of asteroid.

Read the Comment below this, which describes the 'splitting' of the large boulders into smaller ones. Move the mouse pointer over all of the check-marks to see how this was done.

Have a look through all the Event lines, and note how the use of Object groups has reduced the number of Event lines. If there were several more types of boulders, you can see how much easier it is to manipulate a group from the Event Editor, instead of writing an Event line for every single one of the object types.

When you are happy that you have a rough understanding of how the game works, play the game using the 'Run Game' button (don't forget to use + to exit) and then select the 'File' menu heading from the top of the screen, then 'New'. Set the play area size to 640 by 480.
You are now going to make Pulverise yourself.

Creating Pulverise
The Level Editor

All the objects that you need to create Pulverise are in the object library '(Tutorial) PULVERISE'. Click on the button next to the 'Level objects' text on the toolbar at the top of the screen to have a look through all the different object libraries. When you have selected '(Tutorial) PULVERISE' all the objects will be displayed in the object window on the left of the screen.

1. Select the object 'Backdrop object 1' and place it onto the play area.

2. Now select the 'Pulvercraft' object and place it in the middle of the play area.

3. Now Select a 'Boulder(Big)' from the object window, then place three copies using the right mouse button, and place the fourth using the left mouse button. Space them out towards the corners of the play area.

4(a). Now place the 'Score 1' object onto the play area, preferably somewhere near the bottom of the play area. It must be easily visible, but not interfere too much with the action.

4(b). Place the 'Warping to next zone' object anywhere onto the play area.

In order to save you a lot of time, all of the objects have already had their movements and animations completed. They have also been assigned to the groups that are going to be used in the Event Editor. You can have a look at what has been changed by right clicking on them. Always use the 'Cancel' button to escape out of the dialogue boxes that you open, in order to keep the settings as they are.

Now click on the icon to take you to the Event Editor.

The Event Editor

To speed up your games creation and to make you to start thinking for yourselves, instead of in previous tutorials where every single step was outlined, the steps are going to be abbreviated into single words.

For example, instead of saying 'Now insert an Event on line 22, 'Every 00"-25'. To do this right click on the 'New condition' text, then right select the Timer object, select the 'Every' option from its menu. Make sure that the time entered is 25 hundredths of a second, then press 'OK', this will now become;
'Insert Event: Timer, Every 0.25 seconds'.

A similar format will be used for Actions. For example, instead of saying 'Now, add an Action to deduct a life from Player 1. This means that player one has used a bullet when he hits a Gangster. To do this, left click in the box underneath the Player 1 object, where you just inserted the action to add to their score.
Right select the 'New action' option, then select the 'Number of lives' option, then the 'Subtract from Number of Lives' option. Now enter the number 1 into the text box, or use the slider buttons to change the value. When you have correctly set this value to 1, click the 'OK' button.'

This will become;

'Insert Action: Player 1, Lives, -1'.

Where appropriate, the specific line will be mentioned.

So let's make a start!

The following will deactivate all the groups, except "Start of Level Actions" to stop their Events and Actions occurring.
5. Insert Event: Storyboard, 'Start of level'.

6. Insert a Group "Start of level Actions".

7. Insert a Group "Get Ready".

8. Insert a Group "Main Game".

9. Insert a Group "Game Lost".

10. Insert a Group "Level Won".

11. Insert Actions to Event line 1: Special, deactivate groups "Get Ready", (left click to insert more groups...) "Main Game", "Game Lost", "Level Won".

Save Your Game

Before you go any further, save your game so far.

So, click on the 'File' menu at the top of the screen, and then select the 'Save as...' option.
Now under the 'File name' text box, enter the name you want to use. To keep things nicely ordered, enter the name 'pulv-1', then click 'Save'.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘03pulv’, then ‘tutorial’, then ‘pulvtut1’.

The following stops the Pulvercraft from moving and then deactivates itself at the same time as activating the group “Get Ready”.
12. Within the Group “Start of Level Actions”, insert the Event: Special, ‘Always’.

13. To this Event line insert the Actions;
(i) Special, deactivate group “Start of Level Actions” and activate group “Get Ready”.
(ii) Pulvercraft, Movement, ‘Stop’.

The group “Get Ready” simply starts the Pulvercraft moving, and sets the speed of the big boulders relative to the score. The higher the score, the faster they will move!
14. Within the Group “Get Ready” insert the Event: Timer, ‘is the timer greater than 2 seconds’.

15. On the same Event line insert the Actions;
(i) Special, Deactivate group “Get Ready” and activate group “Main Game”.
(ii) Pulvercraft, Movement, ‘Start’.
(iii) Pulvercraft, (left button to insert another action) Movement, ‘Set speed to 10’.

16. Now insert an Event: Special, ‘Always’ on the next line.

17. Insert an Action: Big Boulder, Movement, Set Speed to, ‘5+’, Edit, Retrieve data from an object, Player 1, Score, Current value, ‘/25’. So the text in the Expression Editor should read ‘5+score(“PLAYER 1”)/25’.

18. All Events from here on until specified otherwise are within the group “Main Game”.
This Event tests the keyboard for the <Ctrl> key being pressed.
Insert the Event: Mouse and Keyboard, Keyboard, Upon pressing a key, <Ctrl>.

These Actions fire a bullet from the Pulvercraft and play a sound sample.
19. On this line insert the Actions;
(i) Sound, Play sample, samples, weapons, phaser04.wav’.
(ii) Pulvercraft, Shoot an Object..., a library, (Tutorial) PULVERISER, “Bullet”, ‘Use the direction of PULVERCRAFT, at speed 75.

Save Your Game

Before you go any further, save your game so far.

So, click on the ‘File’ menu at the top of the screen, and then select the ‘Save as...’ option. Now under the ‘File name’ text box, enter the name you want to use. To keep things nicely ordered, enter the name ‘pulv-2’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘03pulv’, then ‘tutorial’, then ‘pulvtut2’.

This Event tests for if the Pulvercraft collides with any of the boulders and the Action below destroys it.
20. Insert the Event: Pulvercraft, Collisions, Another object, ‘Group.Bad’.

21. On the same Line insert the Action: Pulvercraft, ‘Destroy’.

This tests for when a bullet strikes a Boulder(Big).
22. Insert an Event: Bullet, Collisions, Another object, ‘Boulder(Big)’.

These Actions destroy the Boulder(Big) that a bullet has hit, and also creates 2 Boulders(Tiny), one in the same place, one at X-Y coordinates 20,20 relative to the Boulder(Big). It also plays a sample.
23. Onto this line insert the Actions;
(i) Sound, Play a sample, samples, weapons, ‘Explode3.wav’.
(ii) Create, Create object, a library, (Tutorial) PULVERISER, “BOULDER(TINY)”, Relative to:, BOULDER(BIG), X: 0, Y: 0.
(iii) Left click to insert another Action into Create, Create object, “BOULDER(TINY)”, Relative to:, BOULDER(BIG), X: 20, Y:20.
(iv) Player 1, Score, Add to Score, ‘10’.
(v) Bullet, ‘Destroy’.
(vi) Boulder(Big), ‘Destroy’.
(vii) Boulder (Tiny), Direction, Select Direction, (Select 5 different directions on the clock face. One of these will be then be picked at random to decide the direction of the tiny boulders. Try to include an upwards, downwards, leftwards and rightwards direction to make the results as random as possible.)

This Event tests for when the bullet strikes a Boulder(Tiny) and the Action underneath destroys it, and plays a sample.
24. Insert an Event: Bullet, Collisions, Another object, ‘Boulder (Tiny)’.

25. Onto this line insert the Actions:
(i) Sound, Play a sample, samples weapons, ‘explode2.wav’.
(ii) Player 1, Score, Add to Score, ‘5’.
(iii) Bullet, ‘Destroy’.
(iv) Boulder(Tiny), ‘Destroy’.

This Event test if the player or any of the boulders leave the play area. If they do, then they are 'Wrapped' around the play area.
26. Now insert an Event: Group.neutral, Position, Test position of "Group Neutral", ('Leaves the play area'. To do this select all of the 'Leaves in...' buttons.).

27. On this line insert the Action: Group.neutral, Movement, 'Wrap around Play Area'.

Save Your Game

Before you go any further, save your game so far.

So, click on the 'File' menu at the top of the screen, and then select the 'Save as...' option.
Now under the 'File name' text box, enter the name you want to use. To keep things nicely ordered, enter the name 'pulv-3', then click 'Save'.

You can also load up The Games Factory version of the game so far. Look in the directory 'gametuts', then for the sub-directory '03pulv', then 'tutorial', then 'pulvtut3'.

To stop you getting shot by your own bullets, they are destroyed if they leave the play area instead of wrapped.
28. Now insert an Event: Bullet, Position, Test position of "Bullet", ('Leaves the play area').

29. On the same line insert the Action: Bullet, 'Destroy'.

This tests to see if you have destroyed all the boulders.
30. Insert an Event: Group. Bad, Pick or count, 'Compare to the number of "Group.Bad" objects', 'Equal to', '0'.

This group of Actions deactivates the "Main Game" group and activates the "Game Won" group. It displays the text object 'Warping to next zone". It also plays a sample and destroys all remnants of the group "Bad".
31. On the same Line insert the Actions:
(i) Special, deactivate group "Main Game" and activate group "Level Won". Use the left button to insert the second Action in the same place.
(ii) Sound, play a sample, samples, weapons, 'explode3.wav'
(iii) Group.Bad, 'Destroy'.
(iv) Warping to next zone..., Display text..., at X=161 and Y=180.

Save Your Game

Before you go any further, save your game so far.

So, click on the 'File' menu at the top of the screen, and then select the 'Save as...' option.
Now under the 'File name' text box, enter the name you want to use. To keep things nicely

ordered, enter the name ‘pulv-4’, then click ‘Save’.

You can also load up The Games Factory version of the game so far. Look in the directory ‘gametuts’, then for the sub-directory ‘03pulv’, then ‘tutorial’, then ‘pulvtut4’.

This Event tests to see if the Pulvercraft has been destroyed. if it has, it deactivates the group “Main Game” and sends you to the group “Game Lost”.
32. Now insert an Event: Pulvercraft, Pick or count, ‘Have all “Pulvercraft” been destroyed?’.

33. On this Event line insert the Actions:
(i) Special, deactivate group “Main Game” and activate group “Game Lost”.
(ii) Sound, Play a sample, samples, weapons, ‘explode3.wav’.
(iii) Group.Bad, ‘Destroy’.

The following Event and Condition waits three seconds after the last sample finished playing before sending you back to the start of the game, which automatically resets your score.
34. The following Event is to go within the group “Game Lost”.
Insert the Event: Timer, Every, ‘3 seconds’.
Now insert a Condition into this line. Right click on the text, select ‘Insert’.
Now insert the Condition: Sound, Samples, ‘Is a sample not playing?’.

35. On this line insert the Action: Storyboard, ‘Restart the game’.

The following Event and Condition waits three seconds after the last sample finished playing before sending you back to the start of the level, which will keep your score the same. It also plays a sample of music to signify this.
37. The following Event is to go within the group “Level Won”.
Insert the Event: Timer, Every, ‘3 seconds’.
Now insert a Condition into this line. Right click on the text, select ‘Insert’.
Now insert the Condition: Sound, Samples, ‘Is a sample not playing?’.

38. On this line insert the Actions:
(i) Sound, Play a sample, samples, fantasy, ‘scifi09.wav’.
(ii) Storyboard, ‘Restart current level’.

Okay, that’s it! Before you give it a try, don’t forget to save your wonderful game!

When you’ve done that, all you need to do is press the ‘Run Game’ button, and away you go!

If you have encountered any troubles during your game creation, don’t forget that you can very quickly and easily drag Actions around, or delete them.

Try playing different samples instead of the ones specified, just to get a feel of creating for yourself, or try going into the Level Editor and editing the movement of the Pulvercraft, just to see the differences that can be made.

Tutorial 4

Submission

By now you should really be getting a feel for the different Editor screens and their purpose. You can already see that most of your game action is created in the Event Editor, and that the other Editor screens set up the look of your games, and some of the movements, but it is via the Event Editor that the players interact with your game.

Now it's time to take you on a tour of some fairly complex game creation techniques. 'Submission' is a submarine race game which, although fairly simple in terms of game play, uses some fairly fancy techniques to achieve the end result.

Before we go any further, let's load it up and start to have a look around.

Go to the 'File' menu heading at the top of the screen, select the 'Open' option. Now look in the directory 'gametuts' and then look in the sub-directory '04submis' then look for the directory 'tutorial' and then there are three parts to this tutorial, '1-submis.gam', then '2-submis.gam', and also '3-submis.gam'.

Load up part one of this tutorial, '1-submis.gam'.

Storyboard Editor – Submission part 1

Let's have a look around the Storyboard Editor, and see the changes that have been made here, as well as looking into the various preferences that have been changed.

First, go up to the 'File' menu, and select the 'Game setup' option. You can see on the first layer of this dialogue box the author and title of the game. You can see that the panic key has been enabled. This means that if you hit during the game, The Games Factory will take you back to the Windows screen, leaving the game to run in the background. Very useful if you are not supposed to be playing games and your boss walks into the room!

Now look on the next layer, 'Window'. There have been some significant changes from the default settings made here. These have a dramatic effect on the final look of the game when it is run.

The first line is 'Maximised on boot up' which maximises the window used to display the game when it is first loaded. That is, as in any Windows application, The Games Factory uses a window to display your game. Having this option turned on will stretch the window out to fill the screen to the very edge. Note that you will not be able to change the dimensions of the window using the mouse pointer when this option is enabled.

‘Resize display to fill current window size’ will fill whatever window size you are using with the game display. If you disable the option ‘Maximised on boot up’ option when you use this option then you will be able to shrink or stretch the window used to display your game using the mouse pointer to ‘grab’ the edges of the window. Note that when the window is large this option can dramatically slow down the game action due to the complicated nature of the calculations necessary to resize the display.

‘Full screen at start’ is for use on the 16 bit version only. It was designed specifically for use with the 320 by 200 graphics mode that is supported by the 16 bit version.
Note that changing this option will have no effect when used with any other window sizes, only on the 320 by 200. It allows very fast game play of a low resolution that still fills the screen.
Notice at the top of this layer the size of the window has been changed to 320 by 200, which means that only an area 320 by 200 will be shown on the screen. By enabling the ‘Full screen at start’ you ‘expand’ this small chunk of the play area to fill the screen.
When you go back to the Storyboard Editor, have a look at the size of the play area and note that it is in fact larger than the size of the screen. The ‘Window’ preference being set to 320 by 200 means that only a window 320 by 200 will be shown from this play area, and ‘Full screen at start’ means this is expanded out to fill the screen.
This is a very important feature of The Games Factory and can be used to change the way your games are displayed.

For 32 bit users to achieve the same effect as the ‘Full screen at start’ has on the 16 bit version, they should enable the ‘Maximised on boot up’ and ‘Resize display to fill current window size’. This will make the window being used to display the game fill the screen to the edge. Then the display is resized to fit this larger window.
Note that if you maximised the window without resizing the display, then the screen would show a chunk of the play area 320 by 200, surrounded by black.

Whilst having a look around this layer, notice that the ‘Heading’ and ‘Game to include menu bar’ options have been disabled. this means that when your game is run there will be no menu heading at the top of the screen, and no heading telling what the game is. Only the actual play area window will be displayed. Notice that the only way to escape from a game when these options are turned off is to use the + keys. The mouse pointer is hidden via the Event Editor.

No other changes have been made to the default settings, so you can ‘OK’ your way out of this dialogue box.

Now have a good look around the level 1 display of ‘Submission’.

The main thing to note is the play area size. It is 2560 by 480, a long horizontal strip. This has been done because one of the things we are going to do in Submission is to scroll the screen around the main character, so we need a long screen for it to scroll along.
Now let’s have a look in the Level Editor.

Level Editor – Submission part 1

As you have done in previous tutorial games, run your mouse pointer over the different objects on the screen to see what they are. Not all of the objects that we are going to use are on this screen yet - they are being saved for later tutorials.

Move your mouse pointer over the backdrop object ‘Slime floor’ and right click on it. Select the menu option ‘Obstacle’ and note that the ‘No’ has a tick next to it. This means that Active objects cannot collide with it. This is the ‘Water’ that our submarine is going to move through.

By contrast, have a look at the ‘Obstacle’ menu option of any of the other backdrop objects and notice that the ‘Yes’ has a tick next to it. This means that Active objects can collide with them.

Now go and find the submarine object, ‘Craft’. It is on the far left of the play area, which is where the race starts from.
Right click on the ‘Craft’ to show the menu options.
Have a look in the option ‘Movement’ then ‘Edit movement’. Note that the ‘Eight direction’ movement is being used. The maximum speed is set quite low and the deceleration is set quite high, making the object slow, and slows down quickly, just like a real submarine. To make the craft reasonably responsive, the acceleration is set quite high, though.

Continue having a look around the menu options of all the different objects on this screen, but do not change anything. Then, have a look in the object window at all the different Active objects. Note that here are 2 different walls, but they look exactly the same. One is going to be indestructible, the other you will be able to shoot out of the way in the game.
Having 2 objects that look exactly the same was done by placing the first on the play area, and then from its menu, selecting the ‘New object’ option, making the second object an Active object, then selecting its ‘Edit name and icon’ option to change its name. Apart from that they are identical, the other differences in the way they behave are all done in the Event Editor.

Now have a look through all the other objects in the object window. You can see that the background has been made out of many different objects, placed individually onto the screen. Using The Games factory’s many different libraries, you will be able to the same thing to create similar backgrounds or platforms.

At the very bottom of the object library is the ‘Counter’ object, which is used to display the time left, and also a ‘Lives’ object, which in this case is used to keep count of the number of torpedoes the player has left.

When you are finished having a look around the Level Editor, move onto the Event Editor by clicking on its icon from the toolbar.

Event Editor – Submission part 1

One of the first things that you will notice is that there are lots of objects displayed across the top of the Event grid, so many, in fact, that you may have to scroll through them using the scroll bar at the base of the screen. You can make them all fit onto the screen, however, by clicking on the 'Preferences' icon on the toolbar, and then selecting the 'Display' layer of the dialogue box, then moving the 'Number of events on the screen' slider to 'High'. Try experimenting with this slider until you are happy with the result.

Now read through all the comments in the Event grid. These will fairly clearly explain what has been done at each stage, but there is more detail in this chapter.
The first actual Event line that we come to looks very simple, and is in fact very easy to do, but what it achieves used to take hours of extensive programming. Now it can be done in just one line! This Event line makes the rest of the screen scroll about the Craft object when it is moved, so the Craft will always be in the centre of the screen and the play area moves instead.
To do this, the 'Centre horizontal position in the playfield' and 'Centre vertical position in the playfield' options were selected from the Storyboard Actions, and the 'Retrieve data from an object' button was used to retrieve the X and Y coordinates of the 'Craft' object, assigning those values to the screen scrolling actions.

Lines 11 and 14 simply set all the values that are going to be used in the game and plays some music.

Line 18 makes the counter decrease in value by 1 for every second that passes. This was done by creating an Event line 'Every 1"00', then selecting the 'Subtract from Counter' option, and setting the value to 1.

Line 20 moves the 'Mine' objects by making their Y values the same as the Y value of the craft. This will mean that they are always 'in the way' and the player has to shoot them in order to get past them.
This was done by selecting the 'Position' option of the Mine, then the 'Set Y coordinate', option, and then the 'Retrieve data from an object' button was used to retrieve the Y coordinate from the 'Position' option of the Craft.

Use the 'Play' button on the top right of the toolbar to see how all of these events and actions fit together.

Okay, if you've managed to digest that comfortably, let's break off a slightly larger chunk and dive into part 2!

Level Editor – Submission part 2

There has been only one change in the Level Editor compared to part one, but it is a very significant change.

Look in the object window for an object called the ‘Real Collider’. This is the object that is actually going to be tested for collisions instead of the Craft object. The reason for this is that the Craft object has some fairly complex animations that include leaving a stream of bubbles behind it . The problem is that a collision would be detected if any part of the Craft collided with anything else, including the bubbles. Obviously this would not be very realistic, so an Active object was created that has an outline shape that matches that of the Craft object. The ‘Real Collider’ is then made invisible, and given the same X and Y coordinates as the Craft, via the Event Editor.
Pretty nifty stuff, huh? Let’s see how that looks in the Event Editor.

Event Editor – Submission part 2

The first thing to note is the change in line 11. There have been 2 additional groups created, “Lost Race” and “Won Race”. These groups perform groups of actions when the race is either won or lost. They are deactivated here to stop their actions being performed during the game.
Then, when the game is won or lost, one of the two groups is activated and the “Main Game” group will be deactivated.
You can see that this very much simplifies your game creation. Rather than having a condition on all the lines for “Lost Race” that test to see if the game has been lost, the group is simply deactivated, then activated when needed.
Also on this line is the Action to make the Real Collider invisible.

The next significant new line is line 23. This is where the Real Collider is given exactly the same position as the Craft. To do this, after setting up an Event ‘Always’ (Special object) the ‘Position’ option from the Real Collider was selected, then the ‘Select position’ option. Then the position was set to be at 0,0, relative to the Craft.

Line 25 is where the game can be lost. The Event line detects if a collision has occurred between the Real Collider and the background. To set this line up, after right selecting on the ‘New condition’ text and right selecting the Real Collider, it’s menu option ‘Collisions’ was selected, then the option ‘Backdrop’ was selected.
Then the “Main Game” group is deactivated, to stop the game carrying on, and the “Lost Race” group is activated.
An explosion sample is also played on this line.
The players control is taken away from them here, and the Craft object is destroyed.
Because the group “Main Game” is deactivated here, all the lines hereafter will be ignored when the game is lost, and only the lines in the group “Lost Race” will be active.

The next line, (27) is the one that controls when the game is won. To give the game a true race flavour, only the X coordinate of the 'Finishing Post' has to be passed in order to win, making it a true 'Finishing line'.
To create this line, the 'Position' option of the Craft object was selected, then the 'Compare X position to a value' option. Then the X position of the Finishing Post was retrieved via the 'Retrieve data from an object' button. This was compared using the comparison 'Greater than'.
Then, on this line the "Main Game" group is deactivated and the group "Game Won" is activated. To stop the player moving any further the players control is ignored.

The "Lost Race" group, at this stage, merely waits 4 seconds and then restarts the game, which resets the score.

The "Game Won" group waits 1 second then restarts the level, which does not reset the score. It also adds 2 to the value of the score. The score is not going to be used to display the score, but will be used to subtract from the time at the beginning of the game, so each time you win you have less and less time to reach the end.

Now run this version of Submission and see how the craft collides with obstacles and how the won / lost game sections work.

Right, now let's have a look at the last part of this tutorial which gives you a finished, running race game!

Level Editor – Submission part 3

Again, no huge differences here, but the change is still very important. If you look in the object window and on the play area you will see a new object, 'Ammo'. This is going to be used to replenish your stock of torpedoes, without which you would be unable to get past the various mines and walls.

You also need, of course, a torpedo to shoot things with.

Something that is not immediately apparent is that several of the objects have been placed into groups. The groups are 'Group.Bad' and 'Group.Neutral'. Find out which belong to which groups by right clicking on them on the play area, then selecting their 'Object preferences' menu. This will show you which groups they belong to.

Event Editor – Submission part 3

Again, it's line number 11 that has the first changes. This time it's just to destroy all Torpedo objects that might be left on the play area before the game begins.

Line 14 now adds a neat little start to the game. Whilst the music is being played, it moves the Craft down and along, as though it has just been launched into the water. As soon as the music that was played from line 11 has finished, this line plays another sample, then deactivates the group "Start of Race Actions", which stops the craft being moved any further. It also launches you into the main game by activating the group "Main Game".

Line 23 is the next one to look at. If the number of lives (torpedoes) is greater than 0 and the key is pressed, then all the actions on this line will be performed.
This line was created by first of all creating the Event 'Upon pressing the key'. Then the Condition 'Number of lives >0' was inserted by right clicking on the 'Upon pressing the key' text. Then the 'Insert' option was selected, and the Player 1 object was tested by selecting its 'Compare to the player's number of lives' option.
The actions associated with this were to play a sample, and then to create a torpedo just in front of the Craft. Note that the torpedo is not fired out, and is in fact stationary when it is first created. It needs an Event line to accelerate it placed elsewhere in the "Main Game" group. The reason that the torpedoes movement was controlled in this way was to create a realistic effect of the torpedo slowly accelerating as though it really were under water.
In addition to the above actions, 1 is subtracted from the players number of lives (torpedoes) to signify that a torpedo has been used up.

Line 25 is where the Torpedoes are accelerated up to a maximum speed of 50. This was done by selecting the Torpedo objects 'Movement' menu, then its 'Compare speed of "TORPEDO" to a value' option. This was compared with the value 50, using the comparison 'Less than'.
The action to set it's direction was placed in the check box first. The action to increase it's speed was inserted afterwards by left clicking in the box, right selecting the 'New action' text, then the 'Movement' option, then the 'Set speed' option. Its own speed was then retrieved using the 'Retrieve data from an object' button, then '+5' was added onto the end of the expression. So, the speed of the object is set to the speed of itself, +5.

Line 28 is where the torpedoes are destroyed if they collide with the background. This is associated with a sound sample.

Line 30 is where anything belonging to the 'Group.Neutral' is destroyed when hit by the torpedo.

Line 32 is where the torpedo strikes the indestructible wall. The wall changes appearance, but does not get destroyed. The torpedo is destroyed and a sample is played.

Line 33 is where a hole is blasted through the weak wall. The object is not completely destroyed, but the animation is changed to ‘CREATE HOLE’, the torpedo is destroyed and a sound sample is played.

Line 38 controls the actions that ‘collect’ the crate of torpedoes. This is done simply by adding 2 to the number of lives and destroying the crate when the ‘Real Collider’ collides with the ‘Ammo’ object.

Line 40 tests the ‘Countdown’ object. If the value of the counter is zero, then the game has been lost. The associated actions deactivate the “Main Game” group and players controls, as well as playing a sample and activating the group “Lost Race”.

Line 42 is where the game is lost if the Real Collider hits any of the backdrop objects. The same actions are performed as on line 40, with the addition of destroying the Craft object.

Line 44 tests to see if the Real Collider hits any of the other Active objects that belong to the ‘Group.Neutral’, which includes the walls, both Weak and Indestructible, any of the Mine objects or Boulder objects.

That’s pretty much it. Hopefully by now you can understand how this game was constructed as well as why it was constructed in this way.

If there are any points you are finding it difficult to understand, it can be very helpful to try and re-create an Event line on a spare ‘New condition’ line. You can always delete it afterwards. You can also have a ‘look around’ most of the Actions and Events by right clicking on them and selecting the ‘Edit’ option. Note that you cannot edit some of the very simple actions, such as ‘Destroy’.

Tutorial 5

Kung Fu

Welcome to the last of the tutorial games. This game could be based on any of the arcade fighting games and will really show you how to construct a fairly complex game, including many hidden values such as players energy levels, number of hits, etc.

You are very much encouraged to experiment with changes to these games to see what effect they have and to explore all the menu options of the objects as they appear on the Level Editor screen. You can try changing counter values and lives values, even change the appearance of the characters.

The Games Factory already has libraries full of fighter characters, so once you have learnt how to create a realistic fighting game by working your way through this tutorial, you can very quickly and easily create your own using these ready-made Active objects.

Without further ado, let’s load up part one of the Kung Fu tutorial.

Select the ‘Open’ option from the ‘File’ menu at the top of the screen, then look for the ‘gametuts’ directory on your CD ROM drive.
Within the ‘gametuts’ directory look for the ‘05kungfu’ sub-directory. Now look for the directory ‘tutorial’ and within this look for the file ‘1-kungfu.gam’. Open this file.

Have a look at the Storyboard Editor screen, and notice that as in Submission we are going to use a play area larger than the screen size. Again, this is because we are going to scroll around the play area, centring the screen about our hero.

Three are no other special changes here, but have a look at the ‘Game preferences’ option from the ‘File’ heading at the top of the screen, and notice that on the ‘Window’ layer of this dialogue box, we are again not going to display the menu bar or title bar when the game is run.

Let’s go and have a look at the Level Editor of part one of this tutorial.

Level Editor – Kung Fu part 1

Unlike in Submission, here we are using just 4 very large backdrop objects that the main characters are not going to collide or interact with. The backdrop is placed on the screen merely to ‘set the scene’ for the game, a dark city street, full of menace and brooding.

Have a look around the large play area by using the scroll bars at the sides of the screen. Move over to the far right of the play area to see where the main character, 'Hero', has been placed. We have not yet given him any movement, which you can see if you right click on him and select the 'Movement' option. Instead of there being an option to 'Edit movement', meaning that a movement style has been selected for him, you will see there is only a 'Select movement' option. We shall be adding his movement later.

Now have a look around the rest of the play area.
Along the bottom of the screen there is a 'Health' object. This is going to be used to increase the health level of our hero, which is going to be decreased every time he gets hit by any of the baddies.

Note that we have used this object as the icon used to denote his health level in the top left of the screen, in the 'Lives 1' object. You should have a look in the 'Object preferences' option of this object to see how it is prevented from being scrolled around with the rest of the play area. Right select on the object, select the 'Object preferences; option, then have a look on the 'Extra' layer, and then look under the 'Scrollings' heading. Note that the check-box 'Follow the playfield' is un-checked. This is normally checked by default. So, to keep things like score and lives objects in view at all times, you must un-check this box on those objects that you want to remain stationary.

If you have a further look around the left of the play area, using the mouse pointer, you will find a Text object which is going to be used to display several different paragraphs. You can have one text object display several different paragraphs by using its 'Edit Data Strings' option. Right click on the Text object, then select this option to have a look at how this was done. You can see that only this object is going to be used to display the texts for the beginning of the game, as well as for the end of the game and the Victory text.

Right, time to move on to the Event Editor.

Event Editor – Kung Fu part 1

The first line that we see is line number 7, which gets the game ready to start.
This line deactivates the "Main Game" group, to stop the game starting until several other Actions have been performed. It also plays a sample of music and displays the paragraph number 1 of the text object.
In addition it hides the windows mouse pointer to prevent this from being displayed whilst the game is running.

The next thing to note is the group "Start of Level Actions". This further sets up the game, plus it performs a neat trick with the display on line 10.
The way this works is to set the X coordinate of the centre of the display to the same of the centre of the screen, +18.

It does this by reading the X of the left of the playfield and adding 320. As the screen is 640 pixels wide, this centres the display in the middle of the screen. If we left the line like this, the screen would not scroll at all, so we add 18 to this. This means that every time the group "Start of Level Actions" is cycled through, the screen is centred in the left of the playfield, 320 pixels from the left edge, +18, so the screen will be scrolled 18 pixels at a time toward the right. Try changing the number 18 to a larger number, then running the program. The screen will then be scrolled far faster. If you made it less than 18, it would be slower.
To really understand this process, left click on the check-mark and then select the 'New action' with the right mouse button. Now select the 'Scrollings' option, then the 'Centre horizontal position in playfield' option. now click on the 'Edit' button, then the 'Retrieve data from an object' button. From the dialogue box produced, right select the 'Storyboard' object, and select the 'Playfield' option, then select the 'X coordinate of left visible edge' option.
You should now have an Expression that reads 'X Left Playfield'. Now add the text '+320+18' to this, then click 'OK'.

That's one of the new, harder things to learn out of the way, let's carry on!

The next line simply waits 2 seconds and then erases the text of the text box. Note that this is quite different from destroying the object. It is still on the playfield, but invisible.

Line 14 stops the further rightward scrolling of the screen by deactivating the "Start of Game Actions" group when the X value of the left of the screen is 360 less then the X value of the hero character. You must be very careful when using this event as The Games Factory will always stop scrolling when the edge of the play area is reached anyway.
Thus, if you had the hero too close to the right edge of the screen, the scroll would be stopped before the left edge of the screen was within 320 pixels of the hero, and all the associated actions, such as deactivating the group "Start of Game Actions", would never happen.

Now let's have a look at the group "Main Game", which at the moment only contains 1 Event line.

This is a very simple line, but very significant. It horizontally centres the display about the X coordinate of the hero character. This was done by constructing the Event line 'Always' (Special Event) and then inserting the Action by right selecting in the Storyboard column and selecting the 'Scrollings' option, then 'Centre horizontal position of window in playfield', then pressing the 'Edit', then 'Retrieve data from an object' buttons. Then the X position of the Hero was retrieved by right selecting him from the dialogue box produced, then selecting the 'Position' option, then the 'X coordinate' option. It is worth noting that if you already know the exact syntax of an Expression, you can enter that directly into the Expression Editor rather than using the 'Retrieve data from an object' button all the time.

Now use the 'Play' button to run the game to see how all of this combines. If you use the 'Run Game' button, don't forget that you will have to use + to exit the game.

In part 2 of this tutorial there have been no changes to The Storyboard Editor or the Level Editor. There have been, however, some very significant changes made to the Event Editor.

Select the ‘Open’ option from the ‘File’ menu at the top of the screen, then look for the ‘gametuts’ directory on your CD ROM drive.
Within the ‘gametuts’ directory look for the ‘05kungfu’ sub-directory. Now look for the directory ‘tutorial’ and within this look for the file ‘2-kungfu.gam’. Open this file.

Event Editor – Kung Fu part 2

The second part of the Kung Fu Event editor is where the game really starts to come to life, adding movement to the main character, as well as making him kick and punch.

Before we get that far, however, take another look at line 8. There is now an Action to set the number of lives to 5. Remember, we are actually going to use the lives object as a counter for the Hero’s health level.

Let’s have a look now at how we control the main characters movement and animation’s. There are some fairly complex conditions that control all of this. They have been put in place to stop the character jumping from one animation sequence to another mid-way through.
Have a look at line number 22. All the animation changes and movement changes that are associated with this event line can only be performed when the character is either standing or walking. Rather than have one event line to test if the character is performing the walking animation and another event line to test if the standing animation is playing, we simply test to see if no other animations are playing, such as the kicking or punching animation, etc., etc.

To do this we insert the ‘Negate’ option once we have inserted an Event or Condition. This changes the logic of a line, and reverses it.
For example, look on line 22. The first Condition is ‘Repeat while left key is pressed’. By placing the Negate in front of this Condition, the logic of this line now becomes ‘Repeat while left key is not pressed’.
By the same virtue, the line that reads ‘“HERO” animation instep is playing’ becomes ‘“HERO” animation instep is not playing’
The Negate function is denoted by the red cross placed in front of the Condition.

Making the Hero stand moodily and caste about when he is not moving

So, let’s have a thorough look at line 22. This is used to play the characters ‘Standing’ animation when he is not walking, kicking, punching, or falling. Rather than test separately to see if the walking animations are playing, which would happen every time the player moved, we test to see if the keyboard controls are not being pressed.

(i) First of all, the keyboard is tested to see if the left arrow key is not pressed. To do this, first of all we insert the Event 'Repeat while "Left Arrow" is pressed'. (Right click on 'New Condition', right click on 'Mouse and Keyboard', 'Repeat while key is pressed', press the "Left Arrow" key.)
Now, right click on the text 'Repeat while "Left Arrow" is pressed and select the 'Negate' option from the menu produced. You will now have a negated Event line.

(ii) To add the second keyboard Condition, use exactly the same procedure, only press the right cursor key when you are prompted to press a key. Obviously, insert this condition on the same Event line.

(iii) Now let's see how we tested to see if any significant animations are playing.
Right click on the text, 'X Repeat while "Right Arrow" is pressed', 'Insert', then right select the 'Hero' from the dialogue box produced.
Then the select the 'Animation' option, then the 'Which animation of "HERO" is playing?' option. Choose the 'instep' option, then 'OK'.

(iv) Now a 'Negate' was placed in front of the Condition '"HERO" animation instep is playing'.

(v) The above two steps were repeated for the animations 'mid punch' and 'Falling', placing a 'Negate' in front of them each time.

So, by the end of all this, you have an Event line that tests to see if nothing significant is happening to the main character. If all these Conditions are true, then the Action 'Change Animation sequence to Stopped' will be performed. So if the player does not move the main character, and he is not mid-way through kicking and punching or falling, then his mean and moody 'Stopped' animation will be played.

Making the Hero move left and right

The 'Negate' option is used in a very similar way to decide when the Hero can move or not. He will be moved when either the left or right cursor keys is depressed, and there are no significant animations playing. This is to stop him being moved mid-punch or kick, or when he is falling down.

The main thing to note here is that rather than change the Hero's speed, we are going to move him by changing his X coordinate. The reason that we are changing his movement by this method is because if we used his internal speed settings, then every time he performed an Action such as kicking, punching, falling or standing still, we would have to insert an Action to 'Stop' him.
By only changing his X coordinate when a key is pressed, he will automatically stop moving when the direction keys are not being pressed.
To make sure that he faces in the correct direction, the Action to change his direction to left when the left cursor key is pressed is inserted, and to change his direction to right when the right key is

pressed.
We also have to make sure that his ‘Walking’ animation is playing here. If this Action were not inserted then his ‘Stopped’ animation would keep on playing whilst he were being moved. Somewhat unrealistic!

Making the Hero Kick

Line 27 is where we make our Hero kick. He can perform a kick either when he is walking or standing still, so we only test to see if he is not kicking, punching or falling already. Note that the direction of the kick has already been decided by setting his direction in the previous Event lines. To make him perform a kick we simply change his animation sequence to ‘instep’. To add a little life to the action, we also play the sample ‘kungfu-1’.

Making the Hero Punch

Line 29 is where we make our Hero punch. As with kicking, he can only kick when he is not in the middle of doing anything else. We simply insert the Action ‘Change animation sequence to ‘mid punch’, as well as the sound Action ‘Play sample ‘kungfu-2’.

Letting the Hero ‘Pick-up’ health points

To allow the Hero character to ‘pick-up’ health points we simply test to see if he collides with any of the Health objects scattered about the play area. When he does, we add 3 to his number of lives, and than destroy the Health object.

Now use the ‘Play’ button to run the game to see how all of this combines. If you use the ‘Run Game’ button, don’t forget that you will have to use <ALT> + <F4> to exit the game.

Now we are ready to look at the third part of our tutorial, where we are going to have some ‘baddies’ for our hero to fight against.

Select the ‘Open’ option from the ‘File’ menu at the top of the screen, then look for the ‘gametuts’ directory on your CD ROM drive.
Within the ‘gametuts’ directory look for the ‘05kungfu’ sub-directory. Now look for the directory ‘tutorial’ and within this look for the file ‘3-kungfu.gam’. Open this file.

Again, there are no significant changes to the Storyboard Editor, so let’s go straight to the Level Editor.

Level Editor – Kung Fu part 3

The only real change made in the Level Editor has been the addition of the ‘Baddies’ characters. Have a look around the play area to see where they have been placed, and then have a look at their preferences by right clicking on one of them.
Notice that they belong to the ‘Group.Bad’, which will make handling certain aspects of their behaviour easier in the Event Editor.

Well, that’s it for the Level Editor, let’s get back to the Event Editor and see what we are going to do with this new character and how he is going to interact with our Hero!

Event Editor – Kung Fu part 3

The first thing of significance to note is on line 8, where we set all the values for the start of the game.
In addition to all the previous values, move the mouse pointer over the check-mark under the ‘Enemy Fighter’ to view the actions that have been inserted there.
‘Set alterable value A to 8’ is used to set the energy level for each fighter.
‘Set alterable value C to 400’ is used to count how long the fighter has been knocked unconscious for.

Testing to see if the Hero has landed a kick on an Enemy Fighter

The next Event line that has been changed is line 31. This is where the Hero strikes an Enemy Fighter with a kick. To make the reaction as realistic as possible, only animation frames 2,3 or 4 will have any effect upon the Enemy Fighter, i.e. when the kick is fully extended. To do this, the animation ‘level’ of the Hero is tested to see if it is greater than 1 and less than 5.
The next pair of Conditions test to see if the Enemy Fighter is performing any animations of consequence. The Enemy Fighter can only be struck when he is not Falling or Disappearing.
The last pair of Conditions test to see that the Enemy is on the left of the fighter by checking if his X coordinate is less than that of the Hero character, and that the Hero is facing the Enemy Fighter by testing the Hero’s direction.
When all of these Conditions are satisfied at the same time, then the sample ‘impact12’ is played, the Enemy Fighters health has 3 subtracted from it (alterable value A) and his animation sequence is changed to Falling.

All of the above Conditions were copied into a new Event line (32), changing only the Conditions that read the Enemies X value in relation to the Hero, and changing the Condition that reads the Hero’s direction to give a new Event line that test to see if an Enemy Fighter has been struck by a kick when they are on the other side of the Hero and the Hero is facing them.

Testing to see if the Hero landed a punch on an Enemy Fighter

All the above Conditions were copied into 2 separate Event lines, one for each direction, changing the Condition that read the players Animation sequence from ‘instep’ to ‘mid punch’. Then the Action ‘Subtract 3 from alterable value A’ was edited so that only 1 was subtracted from alterable value A, meaning that the kicks that land on the Enemy Fighters are 3 times more damaging than the punches that are landed.

Testing to see if the Enemy Fighters’ health is so low that they become unconscious

Line 36 tests to see if the health value (alterable value A) of any of the Enemy Fighters is less than 0, and that they are not already lying down. If both these conditions are satisfied then the sample ‘whuah’ is played, and their lying down (‘Disappearing’) animation is played.

If the Enemy Fighter is further away than 30 pixels and within 320 pixels, have them ‘lock-on’ to the Hero character and start walking toward him

This is tested by seeing if the Enemy Fighter’s X value is less than the Hero’s X value, minus 30. So, if the Hero’s X value was 400 and the Enemy Fighters was 369, then we subtract 30 from the Hero’s X value to make it 370, we can see that 369 is less than 370, so this Condition would be satisfied.
Had the fighter been any closer, e.g. had an X value of 390, then his X value would have been greater then that of the Hero’s, after 30 had been subtracted.

The next Condition tests to see if the Enemy Fighter is within visual range to the left of the Hero character.
For example, if the Enemy Fighter has an X value of 200 and the Hero has an X value of 500, and then we are subtracting 320 from the Hero’s X value so it becomes 180.
200 is greater then 180 therefore this Condition would be satisfied.

The next 5 Conditions simply test to make sure that the Enemy Fighter is not already involved in doing something else by testing to see that no other animations are being played.

When all the above Conditions are satisfied, then the following Actions are performed to the Enemy Fighter;
(i) His direction is changed to face the right.
(ii) His X value is increased by 2, moving him 2 pixels to the right.
(iii) His animation sequence is changed to ‘Walking’.
So, for all the time that an Enemy fighter lies between 320 and 30 pixels to the left of the Hero and is not performing an animation of note, he will be moved 2 pixels to the right and his animation will

be changed to 'Walking', and he will be turned to face the right.

Making the Enemy Fighter attack the Hero if they are within 50 pixels to the left of the Hero and a random number generated between 0 and 10, is 0.

The first Condition of this line simply test to see if the Enemy Fighter has an X coordinate that is less than that of the Hero, i.e. they are to the left of the Hero.
The second condition tests to see if the Enemy fighter is no further than 50 pixels from the left of the Hero.
These two Conditions combine so that only when the Enemy fighter is to the left and no further than 50 pixels away will the possibility of an attack be launched.

The next Condition tests to see if the animation of the Enemy Fighter is the 'Walking' animation, i.e. that they are not doing anything else already.

The last Condition generates a random number between 0 and 10, and then immediately tests to see if that number is 0. This generates a 1 in 10 chance that the Enemy Fighter will attack every time they are within 50 pixels of the Hero.

If all of these Conditions are satisfied, then the direction is set to be in the direction of the Hero, which may be necessary if the Hero has just walked past the Enemy Fighter, and then the animation of the Enemy Fighter is changed to 'mid punch'.
A sound sample, 'Arrow3', is also played here.

The next two Event lines, numbers 43 and 45, perform exactly the same Actions as lines 39 and 41, except they are related to when the Enemy Fighter is on the RIGHT of the Hero. Have a look through and notice the differences in the logic of the Conditions that deal with the X coordinates of the Enemy Fighter and the Hero, and that the direction that the Enemy Fighter is sent in is to the left, not to the right.

Testing when the 'Group.Bad' has managed to hit the Hero

Here we test to see if any of the 'Group.Bad' have managed to score a hit on the Hero character whilst he is not recoiling from a hit already.

The first Condition tests that the 'Group.Bad' is playing it's 'mid punch' animation, i.e. that they are punching the Hero.

The next line tests if it is the dangerous part of the punching animation that is being played, i.e. when the arm of the Enemy Fighter is extended.

The third Condition tests to see if the ‘Group.Bad’ is overlapping the Hero, and the last Condition checks to see if the Hero characters ‘Falling’ animation is not already playing.

If all of the Conditions are satisfied at the same time, then the following Actions are performed;
(i) The sound sample ‘whip02’ is played.
(ii) 1 is subtracted from Player 1’s lives.
(iii) The animation sequence of the Hero is changed to the ‘Falling’ animation.

Detecting if all the Player’s lives are used, and ending the game if they are

This line simply test the Player 1 object to see if the number of lives is 0.
If it is, then it performs the following Actions;
(i) Deactivates the “Main Game” group to prevent the game carrying on.
(ii) Activates the group “Game Lost” to end the game.
(iii) Changes the animation of the Hero character to one where he lies fallen on the ground.

Using a variable as a counter

Line 51 tests to see if the ‘Disappearing’ animation of the Enemy Fighter is playing. If it is, then add 1 to the value of that fighters alterable value B, which is used as a ‘count-up’ timer. Note there is no specific time reference, 1 is added to the value every time The Games Factory cycles through the Events.

Line 53 tests to see if value B is greater then value C. C is 400 initially, so it will take 401 cycles through the Event grid before value B is greater than value C. The time that this takes is used as the ‘stunned’ period that the Enemy Fighters lie on the ground for after they have been knocked out.
When B is greater than C, the following Actions take place;
(i) Alterable value A is set to 1. This means that a fighter that has been knocked out can only take 1 more hit before being knocked unconscious again.
(ii) Alterable value B is set to 0. This resets the internal ‘timer’ of that fighter.
(iii) Alterable value C is increased by 200. This means that they will stay on the ground for 200 more cycles the next time that they are knocked out.
(iv) The animation sequence is changed to ‘Bouncing’, which flips the fighter back onto their feet.

Now use the ‘Play’ button to run the game to see how all of this combines. If you use the ‘Run Game’ button, don’t forget that you will have to use + to exit the game.

Okay, we’re now ready to look at the last part of this tutorial game.

Select the ‘Open’ option from the ‘File’ menu at the top of the screen, then look for the ‘gametuts’ directory on your CD ROM drive.

Within the ‘gametuts’ directory look for the ‘05kungfu’ sub-directory. Now look for the directory ‘tutorial’ and within this look for the file ‘4-kungfu.gam’. Open this file.

Level Editor – Kung Fu part 4

Let’s have one last look around the Level Editor and see the final changes that have been made to the game.
If you look in the object window you can see that there are three new Active objects, the ‘Drop-Enemy’, which actually looks like a lobster! And there is also the ‘End-O-Level Enemy’ plus the ‘Trigger-Jumping-Enemy’ objects.

The ‘Drop-Enemy’ is actually going to be invisible for most of the game, so it does not actually matter what it looks like. It is simply a trigger for making an Enemy Fighter appear to jump out of a window.

The object that will trigger the Enemy to jump out of the window is the ‘Trigger-Jumping-Enemy’ object. When the game is played, this is also going to be invisible, and is placed some distance above the Hero’s head, and is given the same X coordinate as the Hero in the Event Editor, so it will follow his movement perfectly. If you scroll across the play area you will be able to see where it has been placed. You will also be able to see where all of the ‘Drop-Enemy’ objects have been placed.
So, as the Hero walks across the screen, the invisible ‘Trigger-Jumping-Enemy’ will collide with the ‘Drop-Enemy’ objects. Then, via the Event Editor, we are going to make an ‘Enemy Fighter’ appear to jump out of the window and land right next to the Hero, ready to fight.

Now have a look at the far left of the screen where the ‘End-O-Level Enemy’ has been placed onto the play area. Have a look at his object preferences and see which group he belongs to. (Right click on him, then select the ‘Object preferences...’ option, and select the ‘Object handling’ layer of this dialogue box.)

When you have done this and are happy that you understand the changes made, click on the Event Editor’s icon at the top of the screen to take you to the last changes made in the Event Editor.

Event Editor – Kung Fu part 4

Setting initial values and making the collision triggers invisible

Again, it is line 8 that has the first changes made. Move the mouse pointer across the check-marks to see what they are.
There is now a group “Game Won” which is deactivated here. The only way that this group can be

activated is by reducing the energy level (alterable value A) of the ‘End-O-Level Enemy’ to 0, which is tested later in the Event grid.

If you have a look under the columns for the ‘Drop-Enemy’ and the ‘Trigger-Jumping-Enemy’, you can see that they are both made invisible here.

Under the ‘End-O-Level Enemy’ is the Action to set his alterable value A to 10. This means that his energy level is higher than that of the other fighters to start with, but not only that, when he is hit by the Hero, we are not going to deduct as many energy points, making him very hard to knock out.

Testing to see if the Hero has kicked the ‘End-O-Level Enemy’

Line 38 is where we test to see whether the Hero has managed to hit the ‘End-O-Level Enemy’ with a kick. To make it even harder to knock him out, only kicks are going work against him.
So, the first Condition tests to see if the Hero is overlapping the ‘End-O-Level Enemy’.
The next tests to see if the Hero’s kicking animation (‘instep’) is playing.
The third Condition tests to see which frame of animation it is that is playing when the Hero overlaps the ‘End-O-Level Enemy’. Only those frames where the kick is fully extended will actually harm the ‘Enemy.
The last three lines of Conditions check that the ‘End-O-Level Enemy’ is not already performing an animation of note.

Now have a look at the Actions that have been inserted on this Event line. The first is to play the sample ‘impact12’
The next two are to play the animation ‘falling’ and to subtract only 1 point from the ‘End-O-Level Enemies’ alterable value A. Note that for all the other enemies 3 points are deducted when a kick lands.

Testing the ‘End-O-Level Enemy’ to see if they have lost all their energy points

The next Event line is where the ‘End-O-Level Enemy’ is tested to see if his energy level has reached 0. Notice that this Event line comes immediately after he may have had his last energy point deducted. This was done to prevent any more game actions occurring after the Hero has finished off the ‘Enemy.
The first Condition simply tests to see if the alterable value A of the ‘End-O-Level Enemy’ is less than or equal to 0.
The second tests to see if the ‘disappearing’ animation sequence is already playing.
If both these Conditions are satisfied, then the following actions will be performed;
(i) The group “Main Game” is deactivated to stop any further game actions from being performed.
(ii) The group “Game Won” is activated.
(iii) The sample ‘zing05’ is played.
(iv) The ‘End-O-Level Enemy’ has his animation sequence changed to ‘disappearing’.
(v) The text ‘Victory’ is displayed.

Making the 'End-O-Level Enemy' attack

Line 42 is where the 'End-O-Level Enemy' attacks. This is done very simply with two Conditions. The first tests that he is not already punching, the second tests the timer every 1.5 seconds.
If both these Conditions are satisfied, then the following Actions are performed;
(i) The sample 'arrow02' is played.
(ii) The 'End-O-Level Enemy' has his animation sequence changed to 'mid-punch'.

Note that there is no need to have separate Event lines to test if the 'End-O-Level Enemy' strikes the Hero. This has already been done on line 52, because all of the enemies belong to the 'Group.Bad'.

Controlling the invisible triggers to make an 'Enemy Fighter' jump out of a window

Now have a look at Event line 60. This is where the invisible 'Trigger-Jumping-Enemy' object is made to follow the X coordinate of the Hero. This is done by using the Event 'Always' and then inserting the Action 'Set X position to X("Hero")'. Wherever the Hero goes, the invisible object will follow. We have not manipulated the Y coordinate at all, so that will stay exactly the same as when it was placed onto the play area on the Level Editor screen.

Line 62 tests to see if the invisible trigger object has collided with the invisible 'Drop-Enemy' object. If it has, then the sample 'drumskin' is played.
The 'Drop-Enemy' is also made visible at this stage.
Now, here's the clever part. Although the 'Stopped' animation of the 'Drop-Enemy' looks like a lobster, any other animation can be inserted into its other animation sequences. We have inserted an animation that looks like an Enemy Fighter falling, into the 'Drop-Enemy' objects 'Falling' animation sequence.
So, an object that looks like an Enemy Fighter falling will suddenly appear at the window when the Hero walks underneath a window that has an invisible 'Drop-Enemy' placed there.
Now all we need to do is have them actually fall, i.e. move them down the screen, and then turn the 'Drop-Enemy' into an actual Enemy Fighter.

This is done on line 64. We simply test to see if the 'Falling' animation of the 'Drop-Enemy' is playing. If it is, then the following Actions will be performed;
(i) The Y position is incremented by 16 pixels every time The Games Factory cycles through the Event grid.
(ii) The 'Drop-Enemy' is made to reappear. Note that this is done after the action to change the animation sequence, so the player will only ever see the 'Falling' animation of the 'Drop-Enemy'.

The next Event line tests to see if the 'Drop-Enemy' has reached the ground. The first condition

tests to see if its animation 'Falling' is playing, and the next tests to see if the Y coordinate of the 'Drop-Enemy' is greater than that of the Hero.
If both these Conditions are satisfied, then the following Actions are performed;
(i) The sample 'drun19' is played.
(ii) A real enemy, an 'Enemy Fighter' is shot at speed 0, in no direction. This has the effect of creating an Enemy Fighter at exactly the same coordinates as the 'Drop-Enemy'.
(iii) To further add to the illusion that the 'Drop-Enemy' was actually an Enemy Fighter all along, the 'Drop-Enemy' is destroyed at exactly the same instant as the Enemy Fighter is created.
(iv) The next Action sets the Y position of the new Enemy Fighter to be 15 pixels to the left of the Hero, with anywhere between 0 and 3 pixels added to this value.

The final line that has been changed lies within the group "Game Won". All this does is restart the game.
Now, if you have worked your way through all the tutorial games you should feel competent to start making your own games.

Before you do, though, try changing the tutorial games. Here are some example ideas;
(i) Try changing the initial value of the alterable value A of the enemies to make them easier or harder to knock out.
(ii) Try adding a score object to the game. You could change the score every time the Hero lands a punch or kick, or scores a knockout.
(iii) In the "Game Won" group, substitute the action 'Restart Level' for the action 'Restart Game'. If you are going to use a score object, this will prevent the score from being reset when the game is won.
(iv) Try changing the values of A and C when the game is won each time to make it harder and harder to knock out the fighters.
(v) Try creating a whole new level using the many backdrop objects The Games Factory has to offer. You could make an immense playfield with many fighters.

These are just a few ideas, we're sure you've got loads more. Don't be afraid to experiment with the tutorial games, try making changes throughout them, in the Level Editor as well as in the Event Editor. Read the Picture Editor chapter and Animation Editor chapter to learn how to change individual animation frames.

There are also loads more examples on the Games Factory CD. Look in the Event Editor chapter. At the end of every Event and Action for a given object there is a file pathway for you to look up an example from the CD.

There is also the huge reference section of this manual to extract detailed information about each Editor screen, as well as all the menu options available for all of the objects in all of the Editor screens.

Above all, experiment and play, get an idea in your head, and then go and create it!

The Menu Bar

You will notice that along with almost all other Windows software, The Games Factory has a menu bar across the top of the screen. By clicking on these menu headings you will pull down a list of menu options.

Some of these options are exactly the same for all the Editor screens, but most are slightly different in what they achieve on each Editor screen.

This chapter will discuss all the different menu options and how they relate to each Editor screen. Note that this chapter contains information on 'General setup' and 'Game setup' which you will find invaluable in changing the look of your finished games, not to mention setting up keyboard control and joystick control. Both of these options can be found in the File menu option, which begins on page 85.

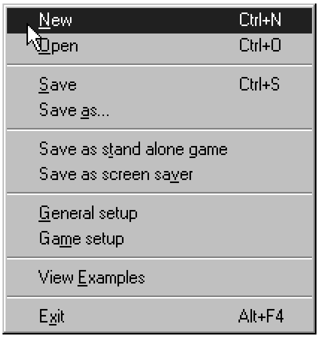
The different menu options will be discussed in the order as they appear on the screen;

- 'File' – Starting overleaf
- 'Edit' – page 94
- 'Objects' – 103
- 'Window' – page104
- 'Run' – page 105
- 'Help' – page 105

Within each heading the effects of each option on each Editor screen will discussed in the following order;

- Storyboard Editor
- Level Editor
- Event Editor

File



The 'File' menu heading contains all the options relating to saving and loading your games from disk, as well as two 'Setup' options. Note that all the options have exactly the same functions on all the Editor screens.

'New' will give you a completely 'blank page' to work with. Note that if you have already been working on a game, The Games Factory will erase this from memory unless you save it to disk. You will be prompted to save your work when you select this option.

'Open' will load in a game previously saved onto disk. You might have to specify the drive that you wish to load the game in from. By default The Games Factory will look on the hard drive (C drive) first.

'Save' will save your game with whatever it's current name is. When you first save a game, it is best to use the 'Save as...' option, then whenever you have made any major changes use the 'Save' option.
If you do not specify a name for your game before you save it then The Games Factory will give it a default name. Also, this option will save to the last disk drive used, by default the hard drive.

'Save as...' allows you to give your game a specific name for it to be saved as, as well as specifying the disk drive that it is saved to.

'General Setup' will take you to a dialogue box with 2 layers.

The first layer of the screen is 'General', there are several tick boxes. Clicking in these turns that option on or off.

'Show handy hints' attaches a text balloon to the mouse pointer so that when you move the mouse pointer over an icon you will be shown what that icon is. For those learning their way around it is best to leave this function on.

‘Show status line’ displays a text line at the bottom of the screen which will display things like mouse coordinates and handy hints.

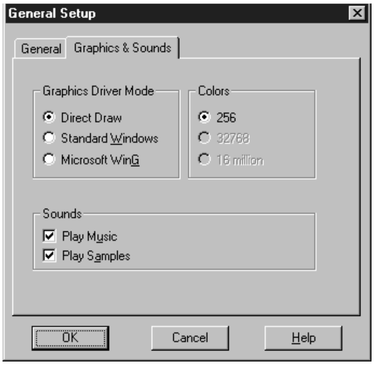
‘Use a small font in dialogs’ means that The Games Factory will use small text in all the dialogue boxes that are used. Normally on.

‘Ask user size of playfield when starting new game’ means that The Games Factory will ask you to choose the size of the playfield for your new game when you use the ‘New’ option from the ‘File’ menu. If you have this option turned off, then by default The Games Factory will choose 640 x 480.

‘Allow picture and animation frames to be stored externally’ means that instead of loading animation and picture frames into the main game file structure, they can be kept on disk until they are needed. Note that if you have this option on and write a game that requires a certain picture from floppy disk, then the game will not run without that floppy disk.

‘Undo levels’ is the number of times that you can ‘Undo’ something that you have changed within The Games Factory.

On the second layer, ‘Graphics and Sounds’, you will be presented with a screen that looks like this;



Graphics Driver Mode
● Direct Draw
○ Standard Windows
○ Microsoft WinG

Colors
● 256
○ 32768
○ 16 million

Sounds
☑ Play Music
☑ Play Samples

By clicking in the tick boxes you can turn the relevant option on or off.

Under the heading ‘Graphics driver mode’ you have several options.

‘Direct Draw’ which only appears in the 32-bit version uses Microsoft’s latest 32-bit graphics driver, Some PC’s might have problems with this driver, if you experience such difficulties then switch to Standard windows.

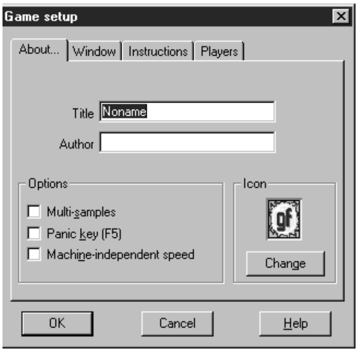
For 16-bit users of The Games Factory, a very fast, custom written graphics driver will be used. It is recommended that you use this driver, if possible, to improve the speed of your game. If you experience any problems you should use the Standard Windows driver.

‘Standard Windows’ will use Windows own graphics drivers.
‘Microsoft WinG’ uses the latest Microsoft 16-bit graphics driver.

The ‘Colour’ option allows you to change the number of colours The Games Factory can use. Note that to get the higher colour options you must first have Windows set to those higher colour resolutions. See your Windows manual for how to change your Windows setup.

‘Sounds’ allow you to play music or not, and play samples, or not.

‘Game Setup’ will take you to a four-layered dialogue box that looks like this;



Title Noname
Author

Options
□ Multi-samples
□ Panic key (F5)
□ Machine-independent speed

The first layer is ‘About..’
You have several options here;

‘Title’ lets you enter the name of your game. This will be displayed when you select the ‘About’ option from the menu bar when you run your game as a whole.

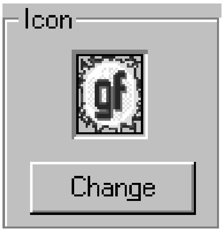
‘Author’ lets you put your name to the game. Displayed as ‘Title of game’.

'Multi-samples' will mix the music and sound samples together when they are played. Note that it takes a moment to do this, so you may well want this option turned off for fast games. The 16 bit version can mix upto 4 channels of sound but the 32 bit version can mix 8. There is a slight delay in mixing in the 16 bit version.

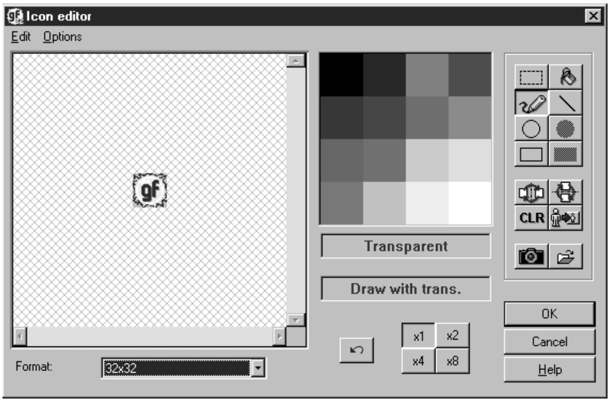
'Panic Key (F5)' -enabling this gives you a panic key (F5) that takes you out of The Games Factory directly to the Windows desktop. It still leaves The Games Factory running in the background, but turns off the sound and minimizes the display. Great for if your boss walks in unexpectedly!

'Machine-independent speed' will make your game run at the same speed no matter what PC it is played on. Note that to make your game run fast on a low specification PC The Games Factory has to cut corners such as having less smooth animation and movement.

'Icon change'



When you save a game in The Games Factory a standard icon is used to represent your game, however you can change this icon. If you wish to change the icon, simply select the 'Change' button and you will be presented with the following dialogue box.



This editor contains all the features of the Picture Editor and works in exactly the same way, see page 287 for more details.

The next level of the dialogue box is ‘Window’. Again, you have several tick-boxes that will alter the look and behaviour of your game.

‘Maximized on boot-up’ maximises the window used to display the game when it is first loaded. That is, as in any Windows application, The Games Factory uses a window to display your game. Having this option turned on will stretch the window out to fill the screen to the very edge. Note that you will not be able to change the dimensions of the window using the mouse pointer when this option is enabled.

‘Resize display to fill current window size’ will fill whatever window size you are using with the game display. If you disable the option ‘Maximised on boot up’ option when you use this option then you will be able to shrink or stretch the window used to display your game using the mouse pointer to ‘grab’ the edges of the window. Note that when the window is large this option can dramatically slow down the game action due to the complicated nature of the calculations necessary to resize the display.

‘Full screen at start’ is for use on the 16 bit version only. It was designed specifically for use with the 320 by 200 graphics mode that is supported by the 16 bit version.
Note that changing this option will have no effect when used with any other window sizes, only on the 320 by 200. It allows very fast game play of a low resolution that still fills the screen.
Notice at the top of this layer the size of the window has been changed to 320 by 200, which means that only an area 320 by 200 will be shown on the screen. By enabling the ‘Full screen at start’ you ‘expand’ this small chunk of the play area to fill the screen.

These three options are a very important feature of The Games Factory and can be used to change the way your games are displayed.

‘Heading’ displays a heading at the top of the screen when the game is run, showing the name of the game.

‘Heading when maximized’ will show the heading when the window for the game is maximized. You must have the ‘Heading’ option enabled to enable this function.

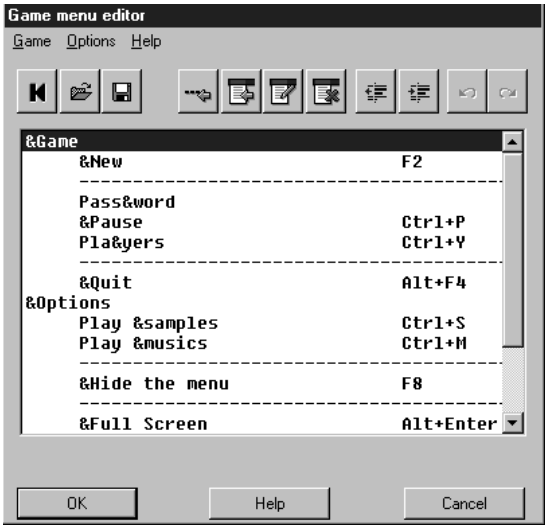
‘Game to include menu bar’ displays the menu bar when the game is run. Having this switched off hides the menu bar. Associated with the ‘Edit menu’ button, described below.

‘Menu displayed on boot up’ is only available when the ‘Game to include menu bar’ option is turned on. If this is turned off then the menu will only be displayed when you make it visible via the Event Editor.

The ‘Edit Menu’ option

In some games you may want to use a menu bar and the mouse pointer to make decisions or access game functions. The Games Factory already has a menu bar for use in your games, which can be shown or hidden from the above options, as well as during your game from the Event Editor. See pages 176 for information on how to display menu options, and how to perform actions when they are selected.

The ‘Edit menu’ button takes you to another dialogue box where you can Edit existing menu options, or create new ones.



```
&Game
     &New                          F2
     ----------------------------------------
     Pass&word
     &Pause                        Ctrl+P
     Pla&yers                      Ctrl+Y
     ----------------------------------------
     &Quit                         Alt+F4
&Options
     Play &samples                 Ctrl+S
     Play &musics                  Ctrl+M
     ----------------------------------------
     &Hide the menu                F8
     ----------------------------------------
     &Full Screen                  Alt+Enter
```

This button resets all the menu options to their default settings.

This button allows you to load a menu that you have already saved from disk.

This button allows you to save your menu onto disk.

This will display a separator in th pull-down menu.

This button allows you to enter a new menu heading or option. You simple insert the text that you want into the text box. You change its status from Menu heading to an option by using the ‘Move option right’ button, descibed below.

Allows you to edit the text of the selected option or heading.

Deletes the selected item.

Moves all the items beneath and including the item selected, to the left.

Moves all the items beneath and including the item selected, to the right.

Undoes the last thing you did.

Redoes the last thing you did.

There is a specific structure to the menu and how each item becomes defined as a heading, or an option.
All the text lines that are on the far left are headings. You can turn an option into a heading by selecting it, and then using the 'Shift option to the left button'. Note that this will also move all the items beneath to the left. To move them back, select the item at the top of the items that you want to move right, and then use the 'Shift option to the right' button.

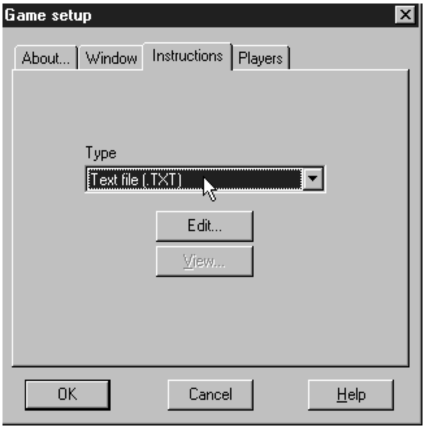
Notice also that when you insert an item, it is inserted before the item currently selected.

It is highly recommended that you work your way through the following tutorial to assist in your understanding of how to create your own menu's. Before embarking on this save any games and use the 'New' option to make sure there are no settings changed.
Make sure the 'Game to include menu bar' is enabled on the 'Window' layer of the 'Game setup' dialogue box.

- Use the 'Reset to default' button to make sure that no settings have been changed.
- Now, select the '&Game' item. Now click the 'New seperator' button. Note that the seperator goes above the '&Game' item. The seperator should now be highlighted.
- Now click on the 'New item' button. Enter the text 'My Heading', then click 'OK'. Now select the seperator again.
- Now click on the 'New item' button and enter the text 'Option 1', then 'OK'. Select the seperator again.
- Click 'New item', enter the text 'Option 2', then 'OK'. Select the seperator again.
- Click the 'New item' and enter the text 'Option 2 A', then 'OK'. Select the seperator again.

- Click the ‘New item’ and enter the text ‘Option 2 B’, then ‘OK’.
- Now select ‘Option 1’ and then press the ‘Shift item to the right’ button. Note that everything beneath this item is shifted to the right.
- Now select the item ‘Option 2 A’ and shift that to the right also. If we left this as it was, then all the other headings, ‘&Game’ etc, would become options within the heading ‘My Heading’. We can restore them by shifting them to the left again
- So, select the ‘&Game’ item, and then press the ‘Shift item to the left’ button twice. Note how all the items underneath are also moved 2 columns to the left.
- Now click all the ‘OK’ buttons to take you back to the Storyboard Editor.
- Now try running the game using the ‘Run Game’ button. You will now have the heading ‘My Heading’ at the top left. When you click on it you wil pull down ‘Option 1’ and ‘Option 2’. ‘Option 2’ will have a small arrow attached and clicking on it will pull down options ‘Option 2 A’ and ‘Option 2 B’.

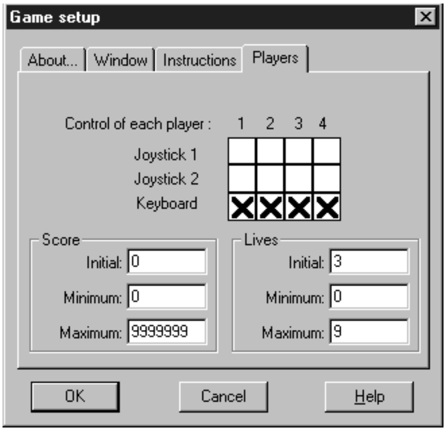
The next level of the ‘Game options’ dialogue box is the ‘Instructions’ layer. From this layer you can decide upon the contents of the ‘Help-How to use’ dialogue box when you run your game.
That is, when you run your game with the menu bar displayed and you choose the ‘Help’ heading, and then choose the ‘How to use’ option you will be presented with a dialogue box. It is from the ‘Instructions’ layer that you can decide what text to put into that dialogue box. An example is shown;



Type: Text file (.TXT)

When you click on the pull-down button at the side of the ‘Instructions’ text box you will be asked to choose the format of text file you wish to use. You can select either a standard .txt file, a write document or a Windows help file. You can then select a file from disk to be displayed in the dialogue box, or you can write your own by using the 'Edit' button. This will then present you with a blank dialogue box which you can then write into. When you are happy with the text you simply save it onto disk.

The next layer of the 'Game Setup' menu is 'Players'. This allows you to change the keys used for direction, as well as the setup of joysticks.
This is what the screen looks like on the 'Players' level;



Control of each player :	1	2	3	4
Joystick 1	□	□	□	□
Joystick 2	□	□	□	□
Keyboard	☒	☒	☒	☒

Score
Initial: 0
Minimum: 0
Maximum: 9999999

Lives
Initial: 3
Minimum: 0
Maximum: 9

Under the 'Control of each player' heading is a grid that represents the device by which each players character will be controlled. Each player has a vertical column assigned to them. As default, all the players are assigned keyboard control, and all use the cursor keys and the 'Shift' and 'Control' keys. Obviously if you have more than one player in a game at a time, you are going to want them to either be using different joysticks or different keyboard controls.

To change the control of Player 1, simply go to the vertical column for player one and tick in the horizontal column to correspond with the keyboard, or joystick 1 or 2.
If you tick in either of the joystick columns, The Games Factory will automatically assign that player movement to correspond with the joystick movements.

Note that when you click on one of the Keyboard options, you will be presented with a keyboard setup dialogue box like this;

In order to change the key that moves your player, press the relevant button. You will then be asked to press a key, which is the key that will then be used for that direction.

For example, to change the key that moves player 1 ‘up’ to the ‘A’ key, click on the “Up” button from the above dialogue box. Then, when your are asked to press a key, press the ‘A’ key. Now you will see that next to the button instead of ‘Up Arrow’ you will see ‘A’, like this;

The ‘Score’number boxes allow you to set the initial values for the score objects. You can also set the maximum and minimum values.

The ‘Lives’ number boxes allow you to set the initial values for the lives objects. You can also set the maximum and minimum values.

View Examples

This simply lets you play examples of what can be done using The Games Factory. The examples are on the CD ROM and give you a chance to view what the more experienced user will be able to achieve using The Games Factory.
You will have the choice to play Zeb, Lobotomy, Magicians Lair or any of the tutorial games.
You can also load up the advanced techniques here.

Exit

Simply exits you from The Games Factory back to the previous Windows application. You will be prompted to save your game if you have made any changes to it since you last saved it.

Edit

Most of the options from the ‘Edit’ heading have a very similar function on all the Editor screens, but what they actually do on each Editor screen can be quite different.
For example, the ‘Copy’ option will copy a selection into the clipboard. If you use it on the Level Editor it will copy an object, but if you used it on the Event Editor it would copy Events, and if you use it on the Storyboard Editor it will copy an entire level, including all the objects from that level, all the Events and Actions. Where applicable, the differences will be explained.

Undo

Simply undoes your last change. Note that you must ‘do’ something before you can ‘Undo’ it. You can keep on ‘Undoing’ steps up to 50 times.

Redo

‘Redoes’ anything that you have just ‘Undone’

Cut

‘Cuts’ a selection into the clipboard. Note that whatever you have ‘Cut’ will be removed from its original location completely into the clipboard, unlike the ‘Copy’ option. By pressing the ‘Shift’ key as well you can select more than one thing at a time.

- Using Cut from the Storyboard Editor allows you to cut levels out and move them to a new location.
- Using Cut from the Level Editor allows you to cut an object and then go to either another level, or even another game, and place it there.
- Using Cut from the Event Editor allows you to cut Events and transfer them to another level, or another game. Note that if you copy Events to another Level or game, all the objects icons will be replaced by a little 'Bomb' icon unless those objects already exist in the level or game that you are transferring to. You can change the 'Bombs' to the new objects by right mouse button selecting on the 'Bombs'.
- Use the 'Paste' option to put your selection into place

Copy

Does exactly the same as 'Cut', but it leaves the original selection behind, moving a copy of the original selection. Use the 'Paste' option to put your copy into place.

Paste

Will take whatever selection is on the clipboard and 'Paste' it wherever you have selected with the mouse pointer. Note that you cannot do things like 'Copy' a selection of events and 'Paste' them into the Level Editor screen.

Delete

Will delete the current selection.

Unselect

Removes the current items from the selection set. Often easier to click on 'empty' space with the mouse pointer. When this is not possible you can use the 'Unselect' option.

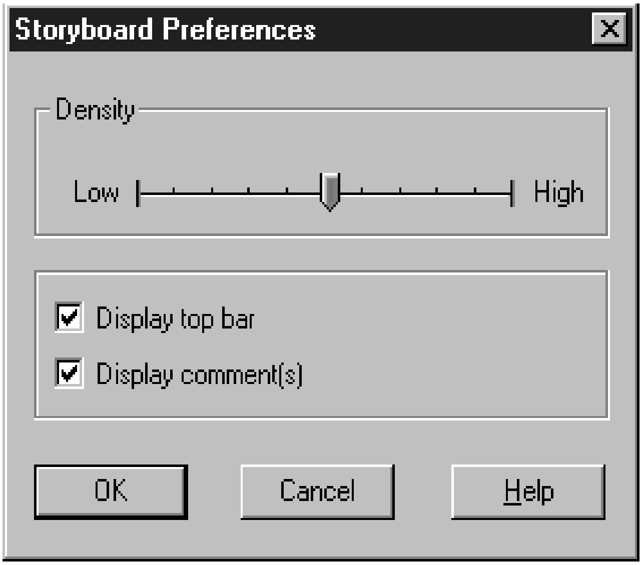
Select all

Will select all objects / events etc., from the current Editor screen, ready for 'Cutting', 'Copying' or 'Deleting', etc.
If you use this option on the Storyboard Editor screen, all the Levels will be selected.
If you use this option on the Level Editor screen, all the objects will be selected.
Using this option from the Event Editor screen selects all the Events and Actions.

Editor Preferences

Allows you to set preferences for each Editor screen, such as how many Events are displayed on the Event Editor, what information is included on the Storyboard Editor.
Each Editor screen has entirely different preferences, so this option is split into headings according to the Editor screens. So, first we have;

Storyboard Editor preferences


Storyboard Preferences

Density

☑ Display top bar

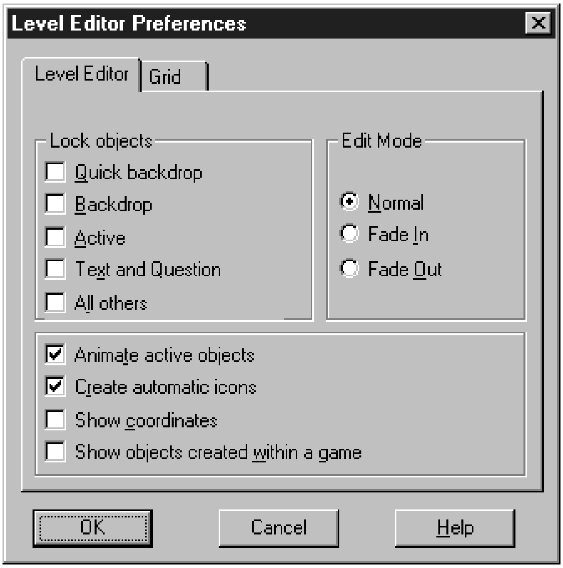
☑ Display comment(s)

‘Density’ allows you to change the size of the ‘Thumbnail’ and associated text for each level. Having a low density will show a large ‘Thumbnail’, but not allow many levels to be shown on the screen at once, whereas having the density high will show minimal information for each level and a tiny ‘Thumbnail’, but will allow a very high number of levels to be viewed on the screen at once.

‘Display top bar’ simply displays a bar above the Levels, describing ‘Number of level’, ‘Thumbnail’ and ‘Comments’.

‘Display comment(s)’ displays the title of the level, the password, play area size and the icons for adding fade ins / outs. Having this option turned off will only show the ‘Thumbnail’ for that level.

Level Editor Preference



Level Editor Preferences

Lock objects

- □ Quick backdrop
- □ Backdrop
- □ Active
- □ Text and Question
- □ All others

Edit Mode

- ● Normal
- ○ Fade In
- ○ Fade Out

- ☑ Animate active objects
- ☑ Create automatic icons
- □ Show coordinates
- □ Show objects created within a game

‘Lock Objects’ locks the relevant object in place on the play area, and prevents them from being selected by the mouse or manipulated in any way on the Level Editor screen. This can be a very useful feature when you start to get a lot of objects on the screen, particularly if you are trying to select a small, Active object, say, and you end up selecting the Backdrop object behind it instead. To stop this from happening you would ‘Lock’ all the backdrops into place.
When a tick box is filled, it means that set of object is locked into place. You can ‘unlock’ them by clicking in the box again.

‘Edit mode’ selects whether you want to edit the game screen, or if you want to add to the fade-in or fade-out transitions. You can place an Active object on the fade-in and then move it around to gradually display the screen underneath. A fairly advanced feature.

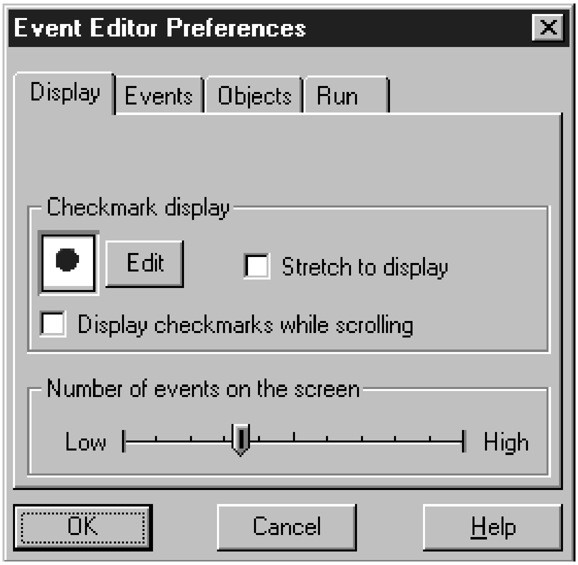
‘Animate active objects’ animates active objects when you place them onto the Level Editor screen. Can be useful to preview an animation or to show when you have changed the animation for 2 similar looking objects.

‘Create automatic icons’ means that The Games Factory will create an icon for any new objects that you create by shrinking an image of the object. Having this option turned off results in The Games Factory not changing the appearance of the icon even if you change the appearance of that object.
‘Show coordinates’ shows the coordinates of the mouse pointer on the status line at the

bottom of the screen. Note that you have to have the status bar showing in order to show the coordinates. To show the status bar, go to the ‘File’ heading, then select the ‘General Setup’ option, and click in the ‘Show status bar’ tick box.

‘Show objects created within a game’ displays the objects that you ‘create’ from the Event Editor on the Level Editor screen. For example, if you ‘Shoot an Object’ and the object that you use as your bullet does not already exist on that frame, i.e., you chose it from another frame or library, then by default you will never see those objects on the Level Editor. By clicking in this box, though, all those objects will be shown on the Level Editor screen on top of the ‘Parent’ object, i.e. the object that ‘Shoots an Object’.

Event Editor Preferences


Event Editor Preferences

Display | Events | Objects | Run

Checkmark display

● Edit

□ Stretch to display

□ Display checkmarks while scrolling

Number of events on the screen

Low — High

OK | Cancel | Help

There are several layers to the Event Editor preferences dialogue box. The first is the ‘Display’ layer.

Checkmark Display

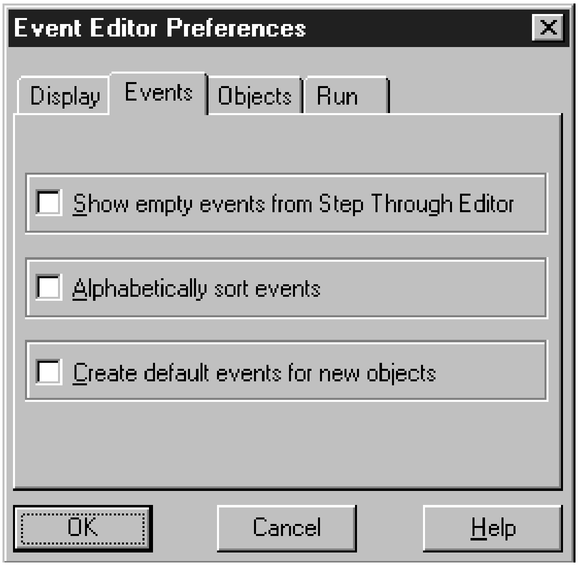
This allows you to change the appearance of the icon that is used as a checkmark in the Event Editor to denote an Action. By clicking on the ‘Edit’ button you can change the icon using a Picture Editor. To learn more about the Picture Editor turn to page 287.
‘Stretch to display’ will stretch the check mark to fill its box when the display density is set low

and the boxes are very large.

‘Display checkmarks while scrolling’ shows the checkmarks event when you are scrolling through a large number of Events. Having this turned off will speed up the rate at which you can scroll through your events.

The ‘Number of events on the screen’ sets the density of the display. To make it easier to read individual Events you would want this setting low, but to show more Event lines at once you would set this high.

The next layer of the Event Editor Preferences dialogue box is ‘Events’.



☐ Show empty events from Step Through Editor

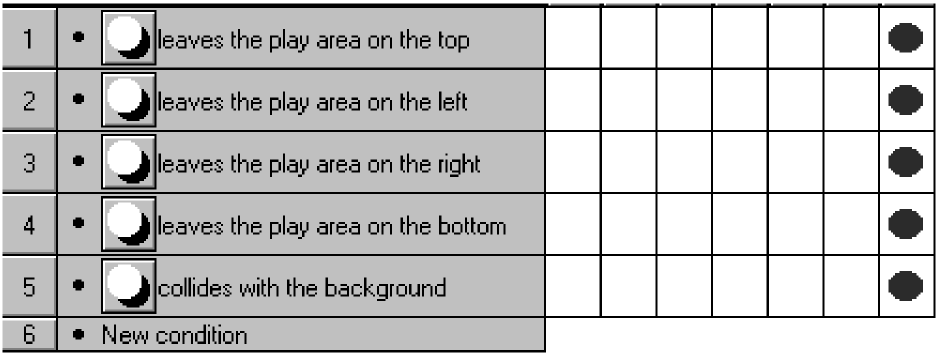
☐ Alphabetically sort events

☐ Create default events for new objects

‘Show empty events from Step Through Editor’ will show all the Events created by the Step Through Editor including those for which no Action has been assigned. Can be a quick and easy way to fill an Event grid with Events which you can then add Actions to. It does create masses of Events, however, and can make it difficult to sort through them all. Many Events created by the Step Through Editor will never have Actions assigned to them, so by default this option is turned off.

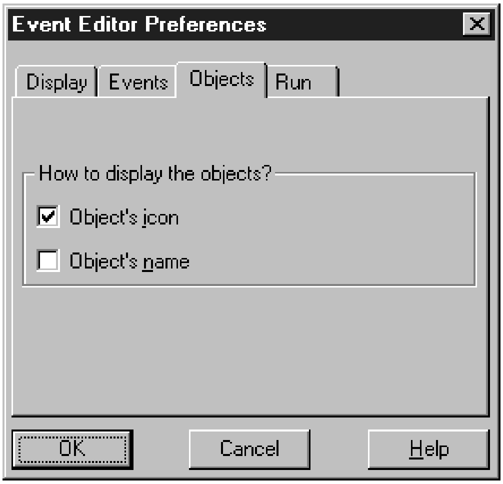
‘Alphabetically sort events’ can be useful for sorting out a large number of Events. However, this can destroy carefully ordered Events. If you are manipulating values in a specific order step-by-step, this option can ruin the logic of your Events. It is generally useful for ordering masses of Events that do not matter which order they run in.

'Create default events for new objects' will automatically create empty Event lines for new objects. For example, if you created a 'Bouncing ball' type object and had this option on, The Games Factory will automatically create the following events;



1	• leaves the play area on the top							●
2	• leaves the play area on the left							●
3	• leaves the play area on the right							●
4	• leaves the play area on the bottom							●
5	• collides with the background							●
6	• New condition							

This can be handy at first, but as you get more experienced you will find that it can 'clog up' the Event Editor with a lot of often irrelevant events.
The next layer of the Event Editor Preferences dialogue box is 'Objects';



Event Editor Preferences

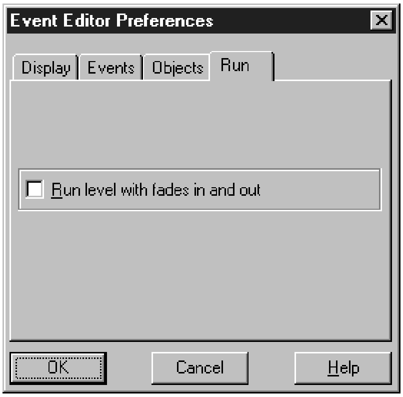
How to display the objects?

☑ Object's icon

□ Object's name

From here you can decide by what method The Games Factory displays your objects on the Event Editor screen. ‘Object’s icon’ means that the icon for that object will be used.
‘Objects name’ will use the name of the object instead, or as well as, the objects icon. Note that you must have one of these tick boxes filled. The Games Factory will not let you have none filled.

The next layer of the Event Editor Preferences dialogue box is ‘Run’;

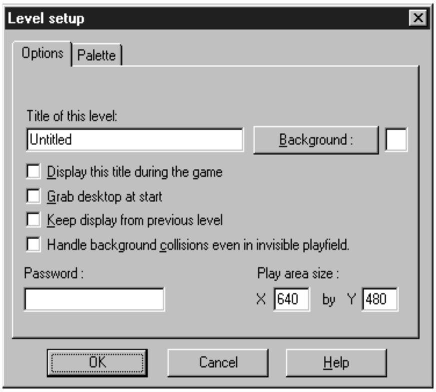


□ Run level with fades in and out

‘Run frame with fades in and out’ will run the fades for your frame when you use the ‘Run game’ button. Note that the fades will not be used when you just use the ‘Play’ button.

Level setup

The options under this heading allow you to change the look and behaviour of each level of your game. There are 2 layers to the ‘Level setup’ dialogue box. The first is the ‘Options’ layer.



Title of this level: Untitled

□ Display this title during the game

□ Grab desktop at start

□ Keep display from previous level

□ Handle background collisions even in invisible playfield.

Password :

Play area size : X 640 by Y 480

‘Title of this level’ allows you to enter a name for that level of your game. Simply click within the text box with the mouse pointer to enable you to write in text.

Below this are several tick boxes. The relevant option will be enabled when they are filled.

‘Display this title during game’ will display the title that you have given to that level when your game is run.

‘Grab desktop at start’ will use the Windows desktop as the backdrop for your game. Everywhere that is transparent on your level will have the desktop showing through.

‘Keep display from previous level’ uses your previous level as a backdrop for your current level. The previous level will be shown where there is a transparent area on your current screen. Note that the animation’s from the previous level will be frozen, and that it is treated as transparency. You could not collide with any objects from the previous level as it is being displayed as a transparency.

‘Handle background collisions even in invisible playfield’ will force The Games Factory to work out collisions that occur with backdrop objects even when those collisions occur well off the visible area of the screen. For example, if you have a platform game and the play area is much larger than the screen, The Games Factory would normally only calculate the collisions that are visible, even when collisions are occurring off screen. This makes game play fast, but not 100% accurate. By having this option turned on, The Games Factory will calculate all the collisions that occur on the entire playfield at once, even those that are a long way from the visible part of the play area.

You can change the password and play area size here, as well as from the Storyboard Editor.

The second layer of the ‘Frame setup’ dialogue box is the ‘Palette’ layer. The current palette is displayed on this layer of the dialogue box. The only option you have here is to ‘Load’ another palette. To do this, click on the ‘Load’ button. You will then be taken to a File Selector dialogue box. This allows you to load in a palette from disk.
This function can be useful if you are creating specific scenes for your games and you do not want to be limited to the standard Games Factory colours. Say, for instance, one of your scenes was a desert scene, you would not really want purples, greens and greys, but you would probably want about 100 shades of light brown! If you created a palette using an art package, then saved it onto disk as a PCX file, you could load in that palette to create the colours for that level. You could do the same for a forest scene, say, which you would want lots of different shades of green.

Objects

The ‘Objects’ heading allows you to change the way objects are displayed on each Editor screen. Note that this heading is not available to use on the Storyboard Editor or the Event Editor, only on the Level Editor.
The ‘Objects’ heading allows you to change how the object library is displayed, save your objects as a library, useful if you have spent hours creating fantastic animation’s, or loading in objects from another game.

‘Show Objects’ is further split down to;
‘In a moveable window’ displays the objects from the library in a window that you can move around the screen, and allows you to change the size of the window with the left mouse button, very useful when you have a very large object library and you want to be able to view them all at once. Simply click and hold down the left mouse button on the edges of the box to ‘drag’ the sizes of the box to the size you want.

‘In a fixed window’ is on by default, and displays the objects from a library in a thin window down the left hand side of the screen with scroll buttons to view through the contents. Is less intrusive on the display than having the objects in a moveable window.

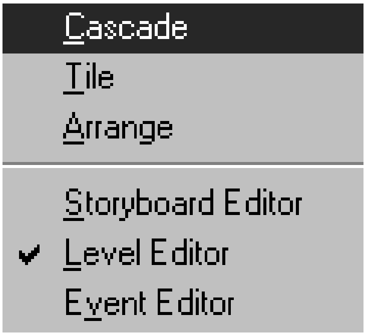
‘Hide Objects’ simply hides the current object library.

‘Pick objects from a game’ allows you to load in all the objects from another game that you have saved on disk, and to use them as a library.

‘Save game as a library’ allows you to save just the objects from your game onto disk as an Object library.

Window

The 'Window' heading allows you to dramatically change the way The Games Factory displays the Editor screens. You can have all the different Editor screens displayed at once in their own windows, which you can then move around the screen and resize. Can be useful for when your games construction gets complicated. The options contained within this heading have the same effect on all the Editor screens.



Note: to turn any of the following options off again, simply double click on the bar at the top of an Editor Screens window.

'Cascade' arranges the editor screens as a cascade of windows. You will be able to see the title bars of the non-selected screens 'behind' the window you have selected. Clicking in the title bar of another window will bring that window to the front of the 'cascade'.

'Tile' arranges the Editor screens in windows side-by-side across the screen. You will only be shown a vertical 'slice' of that screen, but you can scroll the 'slice' from left to right using the scroll bars.

'Arrange' allows you to move the windows around the screen in any arrangement; side-by-side, overlapping, etc., etc.

Clicking on the relevant title below these options will select that Editor screen. If you have any of the above options turned on, you will notice that the title bar of that screen will become highlighted, making that window 'Active'. It is often easier to simply click within the window that you want to make it 'Active', though.

If you have the above options turned off (as default) clicking on one of the names will take you to that Editor screen. It is often easier to click on the icons on the toolbar at the top of the screen.

Run

The 'Run' heading allows you to perform the same actions as the 'Run Game', 'Run Level', 'Play', 'Pause' and 'Restart' icons from the toolbar at the top of the screen.

Help

The 'Help' title takes you to the The Games Factory help pages via a number of different methods. If ever you get stuck and need a very quick reference, the Help pages are always there. You can either click on the 'Help' title, or press F1 to get you into the Help files. If you click on the Help icon, the pointer changes to the Help icon and you can then click on an object that you want information on. The following pages describe how to get about once you are in the Help system.

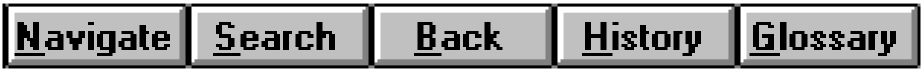
How to move around the Help pages

Before you move onto reading the following options, you will need to know how to move around the Help pages in order to make the most of them. Reading this first could save you a lot of time going round in circles!

The help pages layout

When you have a heading that contains many chapters and sub-chapters, the Help pages will be divided into two windows, one to display the actual text of the Help files, the other to help you navigate your way around the substantial pages of information, known here as the 'Find Topics' window.

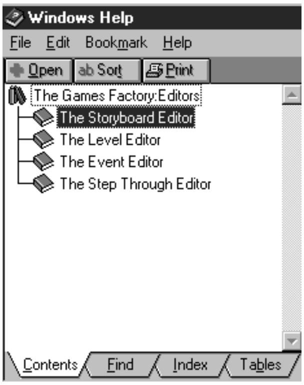
We shall concentrate mainly on how to navigate your way quickly and easily through the different topics via the 'Find Topics' window.
Note that you can click on the layer tabs at the bottom of the 'Find Topics' screen, or use the buttons on the top of the text display box to change which layer you are viewing of the 'Find topics' window.



'Navigate' corresponds to the 'Contents' layer, 'Search' to the 'Index' layer and 'Glossary' to the 'Tables' layer.
The other buttons are defined on page 108

Contents

The 'Contents' layer looks like this;

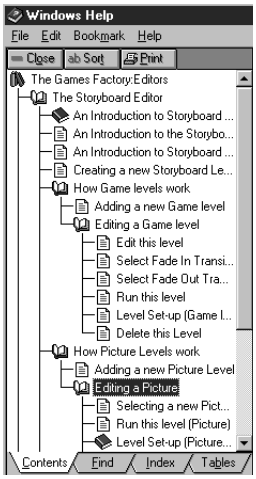


- The Games Factory:Editors
  - The Storyboard Editor
  - The Level Editor
  - The Event Editor
  - The Step Through Editor

In the text box are listed all the topics within a particular heading, by default. This is known as a 'Tree', so called because it resembles a family-tree diagram. This screen shot was taken from the 'Editors' help pages. The icons next to each topic tell you something about what is going to be contained within that topic.

The 🗎 icon denotes a single page of text. By clicking on this icon, that page of text will be displayed in the window on the right of the screen.

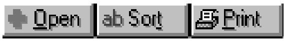
The 📚 icon denotes a very large topic and that this is the 'Root' of that topic, and that many sub-topics are contained within that heading. By clicking on it all the sub-topics will be listed. Note there can be further headings containing many pages of text within a Root Topic;



- The Games Factory:Editors
  - The Storyboard Editor
    - An Introduction to Storyboard ...
    - An Introduction to the Storybo...
    - An Introduction to Storyboard ...
    - Creating a new Storyboard Le...
    - How Game levels work
      - Adding a new Game level
      - Editing a Game level
        - Edit this level
        - Select Fade In Transi...
        - Select Fade Out Tra...
        - Run this level
        - Level Set-up (Game l...
        - Delete this Level
    - How Picture Levels work
      - Adding a new Picture Level
      - Editing a Picture
        - Selecting a new Pict...
        - Run this level (Picture)
        - Level Set-up (Picture...

Note that the 📖 has changed and pulled down a list of text pages and sub-topics. You can view the pages within a sub-topic by clicking on its 📘 icon. You will then be able to read the individual pages from that sub-topic by clicking on the relevant 📄 icon.

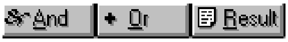
On the top of the 'Contents' layer there are three buttons;



'Open' displays a selected topic, sub-topic or page.
'abSort' sorts the topics alphabetically. You will be presented with a list of all the topics in alphabetical order, starting from any topic you had selected.
Note that when you are in the alphabetical sort mode, that the button changes to 'Tree'. This means that if you select a topic and then click on this button, you will be shown the 'Tree' for that topic.
The 'print' button will print the current Help pages.

Find

The next layer of the 'Find Topics' window is the 'Find' layer. From here you can select subjects to look up. All the options are listed in alphabetical order. Select the words you want to look up by clicking on them with the left mouse button, then pressing the 'And, or ' Or' buttons.



For example, if you selected 'Actions' and 'Events' you would be shown all the topics that involved Actions and Events.
Alternatively, by clicking in the text box at the top of the list, you can directly enter your own words to use for the search, including 'and' / 'or'.

When you are happy with what you want to search for, you can either use the key, or press the 'Result' button. You will then be presented with one or more topic headings, which you can then select with the left mouse button to be displayed in the text display window on the right of the screen.
Note that when you have pressed the 'Result' button, it changes to 'Find', which will take you back to the find pages.

Index

The next layer of the 'Find Topics' window is the 'Index' layer.
This simply lists the index to the Help pages in alphabetical order. You can select a topic to look up either by entering it in as text, or selecting it from the list using the left mouse button, then either press the return key or use the 'Result' button to view those pages. Use the 'Print' button to print them out onto paper.

Tables

The next layer of the ‘Find Topics’ window is the ‘Tables’ layer. This shows a glossary of terms used by The Games Factory. By selecting one of the glossary terms you will pull down a text box defining that term.

Using the buttons on the text box

Navigate changes the layer of the ‘Find topics’ window to the ‘Contents’ layer. Shows you a ‘Family-tree’ type diagram of the help topics. See the ‘Contents’ heading above for more information.

Search changes the layer of the ‘Find topics’ window to the ‘Index’ layer. Shows you an index of the Help pages. See the ‘Index’ heading above for more information.

Back changes the page being displayed to the previous page that was displayed.

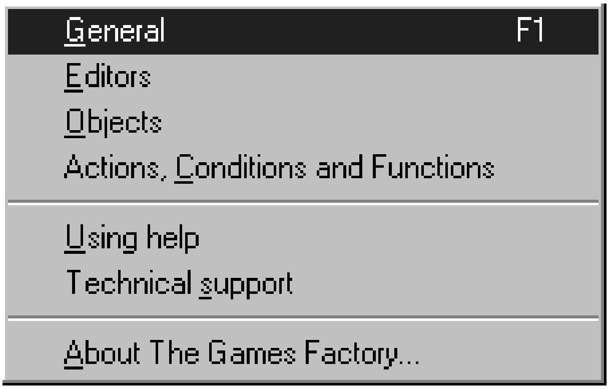
History produces another window which lists all the pages that you have displayed in the text window, i.e. the ‘History’ of your Help usage. You can select a page from here and it will be re-displayed.

Glossary changes the layer of the ‘Find topics’ window to the ‘Tables’ layer. See the ‘Tables’ heading above for more information.

The << and >> buttons take you forwards and backwards through the Help pages, starting from the page you have selected using the other Help options outlined above. Note that these buttons do not go to the previous topic that you selected, but go in numerical order from the page you are on. To go back to a previous topic use either the ‘Back’ or ‘History’ buttons, described immediately above.

The Help Options menu

When you click on the 'Help' heading you will pull down a menu that looks like this;



'General' takes you to the general overview of The Games Factory, including some handy hints and a brief description of what The Games Factory can do.

'Editors' takes you to the Help pages for the Editor screens.

'Objects' takes you to the help pages for the Objects.

'Actions, conditions and functions' takes you to the Help pages that will list all the available Actions, conditions and functions for different object types.

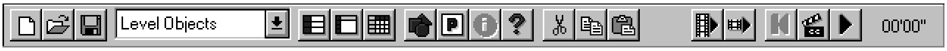
'Using Help' takes you to Help pages to teach you how to use the Help pages!

'Technical support' Describes the support facilities that are available should you be encountering any technical difficulties.

'About The Games Factory' Describes the version of The Games Factory that you are using, as well as the credits for the development team of The Games Factory.

The Toolbar

The Toolbar exists on all of the Editor screens and has broadly similar functions on each of them. It lies at the top of the screen just below the window heading.



Note that not all of the icons are active; some, in fact, are mutually exclusive. For example, you cannot restart a frame until you have played it, nor can you play a frame unless it is paused or restarted. You can only Cut or Copy something once you have selected it, and you can only Paste something after you have Cut or Copied it.

A lot of the functions of the toolbar are replicated by the pull-down menus. The toolbar is supposed to be a short cut to using the menus.

We shall discuss each of the icons in turn, starting at the left of the toolbar.

New. This re-sets The Games Factory as completely blank. You will be prompted to save any un-saved work.

Open. This allows you to load in a game from disk.

Save. Saves the current game to disk.

Storyboard Editor icon. Takes you to the Storyboard editor.

Level Editor icon. Takes you to the Level Editor. Note that you must click in an empty frame on the Storyboard Editor to take you to the Level Editor when you are starting with a completely blank game.

Event Editor Icon. Takes you to the Event Editor. Note that you must click in an empty frame on the Storyboard Editor to take you to the Event Editor when you are starting with a completely blank game.

The Create icon. Has different functions depending on which Editor screen you are on. When on the Storyboard Editor it allows you to decide on the type of level you are creating, e.g. pictures or animation’s. When used on the Level Editor it allows you to create your own objects. When used on the Event editor, allows you to create a new line, group of events or comment. See the relevant Editor chapters for more information.

The Preferences icon. Has the same function as the Editor Preferences option from the pull-down menu. See Editor Preferences on page 96

The Information icon. Tells you how much memory is being used by the various different parts of your the Games Factory game.

The Help icon calls up a pointer Help. Changes the mouse pointer to a question mark, and then when you place it over an object and click with the left mouse button, The Games Factory will call up the Help pages related to that object.

The Cut icon. Cuts a selection into the clipboard. See the ‘Edit-Cut’ option from the Menu bar chapter on page 94 for more information.

The Copy icon. Copies a selection into the clipboard. See above for more details.

The paste icon. Pastes a selection from the clipboard. See above for more details.

The Run Game icon. Runs your game as a complete entity, re-sizing the screen where applicable, displaying the game menu bar, going from level to level as a fully working game.

The Run Level icon. Runs the level you are working on as though it were a finished game, complete with menu bar etc.

The Restart Icon. Stops the game running and resets the timer to zero.

The Step Through Icon takes you to the Step Through Editor.

The Play icon. Allows you to play your game to test movement, collisions, sound etc. This is intended as a ‘minimal run’ and will not re-size the screen, produce a menu bar or mouse pointer. Because of this, though, it is very fast to be entered into and exited from. Especially useful for testing Shoot’-em-up-games. Note also that it only plays your current level and that the icon changes to a ‘Pause’ symbol whilst your game is being run. You cannot make any changes to your game whilst it is paused, you have to restart your game to do that.

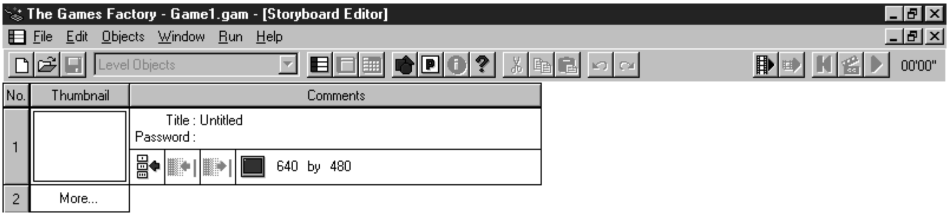
The Storyboard Editor

The Storyboard Editor is the initial editor that you will be presented with upon loading The Games Factory. It is from here that you would normally load up a game that you wanted to work upon, although you can actually do this from any of the Editor screens.

It is from the Storyboard Editor that you decide upon the initial look of each of your levels, inserting new levels, changing the numerical order of the levels, adding passwords and titles.

From here you can copy a whole level, ready to changed in subtle ways, or go into one level, 'Cut' out a list of Events', move to another level and 'Paste' them into the Event Editor for that level.

When you first load up The Games Factory, this is what the top of the Storyboard Editor screen will look like;



The Games Factory - Game1.gam - [Storyboard Editor]
File Edit Objects Window Run Help
Level Objects
00'00"

No.	Thumbnail	Comments
1		Title : Untitled Password : 640 by 480
2	More...	

At the top of the screen is the name of the game you are currently working on. 'Game1.gam' is the default name that The Games Factory gives to a game that has not been given a name already.

Below this are the pull-down menus. These are discussed in detail on page 84.

Below this is the Toolbar, which uses icons for many of the functions from the menu bar, plus several more functions that are not in the menu bar. Note that many of these are de-highlighted and therefore not available. That is because many of them relate to doing something with a current selection. Until you have created a level and / or selected something, you will not be able to use them. See the 'Toolbar' chapter on page110 for more information on the function of each of the icons on the toolbar.

On the main window of the Storyboard Editor is where all the different levels of your game are displayed. The number of each level is displayed down the left hand side of each level. By default you are shown a tiny screen-shot of that level, called a 'Thumbnail', next to the number of that level.

Next to the Thumbnail are the comments for that level.

In this area are displayed the title for that level, and the password. They will remain empty by default.

Title

You can add or edit a level's title by simply clicking on the 'Title' text. You then enter the name of the level using the keyboard.

Password

To add or change the password to a level is equally as easy as changing the Title. Simply click on the 'Password' text and then enter the password you want to be used to enter that level.

Underneath the comments are several 'buttons';

Type of Level

This denotes the type of level. By default all your levels will display this icon, which denotes them as a game level. If you were to make your level an animation frame, this icon would change.

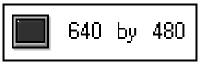
Fade In Transition

You can add a fade-in transition to your level by using this icon. The fades are in the 'Fades' directory.

Fade Out Transition

You can add a fade-out transition to your level using this icon in a similar way to the fade-in transitions. See above.

Play Area Size Selection



640 by 480

This is where you can select the size of your play area. The play area is the size of the area that your characters are going to be moving around on. It can be much larger than the actual screen area, and you can in fact change the amount of the play area being shown, using the 'File-Game Setup-Window' option. This can be useful for making scrolling games; you would have a very large play area, much larger than would fit on the screen, then you could move the view of that area around, creating a 'Scrolling' effect.

To change the size of the play area you can either select on the 'Monitor' icon, from which you can select any of the pre-determined sizes The Games Factory has to offer, or you enter your own sizes directly by clicking on the numbers of the screen size and then changing them via the keyboard.

Adding a Level to your Game

There are many different ways to add a new level to your game from the Storyboard Editor, but they all have the same end result, that is, producing a new, numbered level at the end of all your current levels. If you then want to move that level around, you can simply click on it and 'drag' it to the position you want.

Method 1

Simply click on the number text of the empty level with the left mouse button. Or, click on the 'More...' text with the right mouse button.
From the list of icons pulled down, select the icon. You will then have an entirely new, blank level.

Method 2

You can also add another level by clicking on the icon from the toolbar at the top of the screen.
Then select the icon.

Method 3

The third method does not create a blank level, but copies a level you have already created into another level, including all the objects, actions and events.
Simply click on the level that you want to copy with the left mouse button, then, holding the button depressed, 'drag' that level over the 'More..' text and release the mouse button.
You can, of course, use this method when you first load up The Games Factory and make multiple copies of the blank first level. You will then have many levels ready for you to fill.

Copying Events or Objects

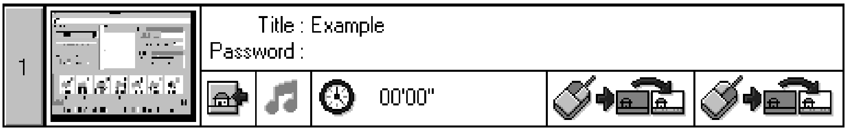
You can Cut, Copy or Paste events or objects from the relevant Editor screens to the same Editor screens on another level via the Storyboard Editor. Once you have Cut or Copied a selection into the clipboard, you can then Paste them into another level by going to the Storyboard Editor screen, selecting the level that you want to Paste your selection into, go into the relevant Editor screen and then Paste your selection there.
Note: If you are copying Events, you must paste them into the Event Editor of the next screen. The same with objects, they can only be copied to and from the Level Editors.
If you copy Events from one level to another that relate to objects that do not exist on the new level, all the icons for those objects will be replaced by 'Bomb' icons.
You can then change these to the objects from that level by selecting on them in the Event Editor screen.

Making a Slide Show Using the Storyboard Editor

You do not have to use The Games Factory for just games creation. You can also manipulate pictures to make a slide show, controlled by clicking one of the mouse buttons.

To have each level displaying just a picture, click on the create icon , then select the Picture Icon.

You can then place the mouse pointer over an empty frame that you want to have as a picture frame. Once you have done this you will be asked to select your picture from disk. You are prompted as to the different file types you are allowed to load in as pictures.
When you have decided upon a picture to load in, you will have a level display that looks like this;



1

Title : Example
Password :

00'00"

Notice that instead of the game level icon in the bottom left of the comments box, there is now a picture icon.

Music

Next to this is a ‘Note’ icon. By clicking on this you can add music to the level from The Games Factory’s ‘Midi’ files.

Time Delay

Next to this is a clock symbol. This shows the time that level will be shown before being moved onto the next frame. To change the time delay simply click on the numbers of the clock with the left mouse button and enter in the time delay via the keyboard.

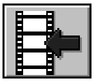
Slide Show controller

Just like a real slide show, you can flick between the frames using the buttons on the mouse. The first icon where the left mouse button is highlighted, controls what happens to the levels when you use the left mouse button.
To decide which level you want to jump to when you press the left mouse button, click on the picture of the two frames, next to the highlighted mouse. This will pull down a menu, from which

you can choose an option. The same procedure is used to decide what to do when the right mouse button is used.

Adding an Animation Frame

In a very similar way to making a slide show using pictures inserted into the levels, you can insert an animation frame into a level and use that to provide you with an animated slide show. Almost exactly the same procedure is used to create a picture level, except that you click on the animation icon instead of the picture icon when you have clicked on the create icon. The animation icon looks like this;



Note that you can only load in animation files, i.e. FLI or FLC files. Apart from this, the animation levels are treated in exactly the same way as the picture levels.

The Level Editor

Translation Error.
G INT (1):GAME FACTORY:GAME FACTORY:GAME FACTORY MANUAL:Level editor section :Levele

You can get to the Level Editor screen by clicking on the AME icon when you are using the other editor screens, or from the Storyboard Editor by selecting a particular level, then the ‘Edit Frame’ option.
The Level Editor is the initial place for your games creation. It is here that you decide on your background, platforms, ladders etc, and it is also here that you place all the baddies and goodies that you want in your game. It is also through this screen that you access basic movement controls, as well as the animation of your main characters.
You must place an object on this screen first before you can assign an event to it in the Event Editor screen, or perform any other manipulations on it.
It is from this screen that you can create text for headings, question objects for adventure games, or counter objects, which can be displayed or hidden. You will find counter objects extremely useful for deciding all sorts of different conditions, defining movement or energy levels.

Placing objects on the screen

The play area

This is the area you will actually be using for your game, the overall “background” for your game. If it is too large to fit on the Level Editor screen, you can move around it using the scroll bars at the sides of the screen. You can change the size of your play area from the Storyboard Editor.
To move around using the scroll bars you can either click on the arrows at the end of each scroll bar, or grab the central slider and move that around using the mouse.

The grey area of the screen surrounding the play area can be used for storing objects so that they will appear in the Event Editor but not necessarily on the play area when the game is first run.

Object Libraries

The Games Factory already has thousands of different objects ready made for you to play with, backgrounds, goodies and baddies, spaceships and tiles. These are kept in their own Object Libraries. Generally, you will find objects grouped together by type.

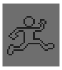
In order to open an object library, select on the Level Objects ⬇ at the top of the screen. This pulls down a list of all the different libraries. If you select one, its contents will be displayed on the Object Shelf. If it is very large you may well need to scroll through it in order to view all its contents.

Handy Hint!

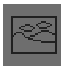
To get a better view of the objects in each library, you can go to the ‘Objects’ pull-down menu at the top of the screen, then select the ‘Show objects’ - ’In a moveable window’ option. This will pull-down a window (The Object shelf) which you can stretch out to fit in all the objects of a library at once.

The Object Shelf

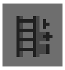
basically, this is the holding place for all the objects in a given library. It has some useful features to help you decide on the objects that you are going to be using in your game. Each type of object has a heading describing its type, e.g. Active objects. This is denoted by a specific icon, outlined below.



denotes all the Active Objects on the object shelf.



denotes all the Backdrop Objects on the object shelf.



denotes all the Game objects, such as text, counters, high score tables and other special objects.

It is from the object shelf that you select the objects to use in your game. There are two ways of doing this;

Using the left mouse button

By using the left mouse button you can select any object from the Object Shelf and that object will be placed onto the playfield just once.

Using the right mouse button

If you use the right mouse button to place objects on the screen you will end up with clones of the original object.

Deleting Objects

To delete an object, simply select it and then press the “Delete” key. To delete a group of objects, simply drag a box around them and press “Delete”.

Note: In order to stop the background being selected when you throw a box around objects, you can go to the “Edit” menu from the toolbar and by using the “Lock Objects” option, stop any different group of objects from being selected. (See overleaf, ‘Problems selecting animated objects’.)

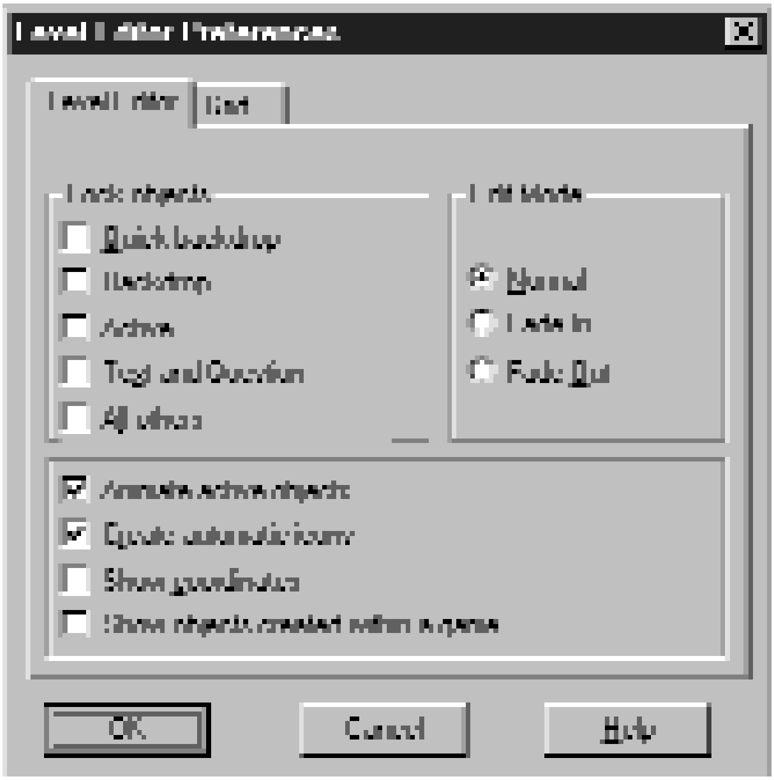
Level Objects

Once you have placed an object on the screen, it will also be on your own “Level Objects” library. This is useful if you are using several objects from several different libraries; every time you select one it will be stored on the Level Objects library.

Problems selecting animated objects

Sometimes with small or ‘whispy’ animated objects it can be hard to select them without selecting the background, particularly if the animation moves around a lot.

To get around this, use the ‘Edit’ pull down menu (bar at the top of the screen), then select the ‘Editor preferences’ option. You can then ‘lock’ a selection of objects in place while you select other objects. You will find that this comes in very useful when the screen starts to get a little crowded!



Level Editor Preferences

Level Editor | Grid

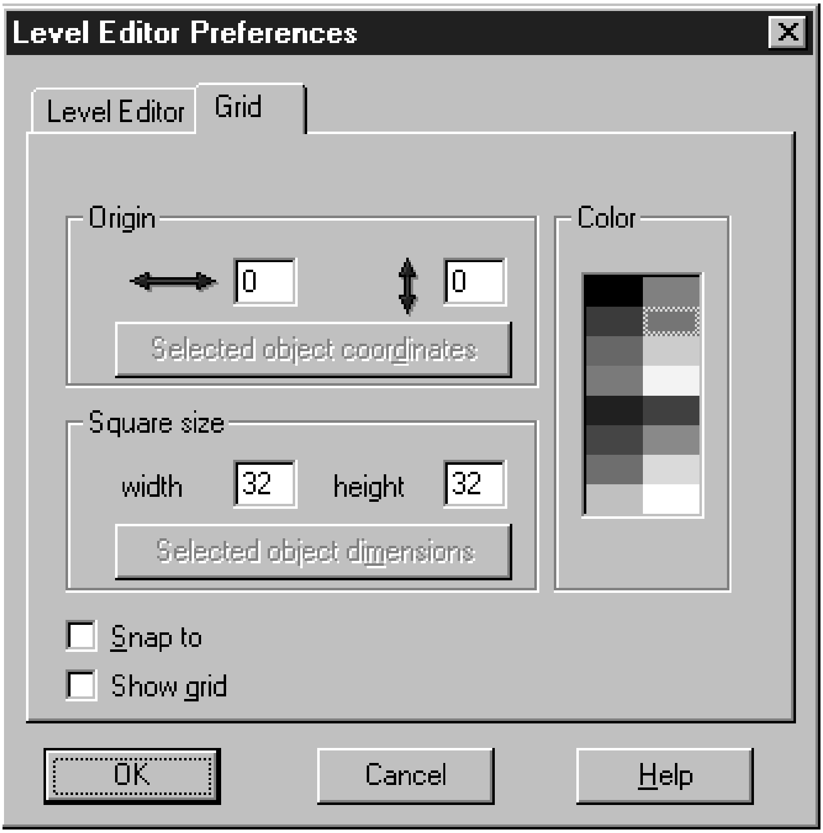
Lock objects
- □ Quick backdrop
- □ Backdrop
- □ Active
- □ Text and Question
- □ All others

Edit Mode
- ● Manual
- ○ Fade In
- ○ Fade Out

- ☑ Animate active objects
- ☑ Create automatic icons
- □ Show coordinates
- □ Show objects created within a game

Placing objects on a grid

You can place your objects very precisely using a background grid for the objects to “snap” to, very useful for Pac-man type games or platforms. To display a grid, select the “Edit” menu from the bar at the top of the screen, then select the “Preferences” option.
Go to the ‘Grid’ page.



Origin: ↔ 0 ↕ 0

Square size: width 32 height 32

□ Snap to

□ Show grid

Snap to

Will ‘snap’ any object to the grid in use, only allowing them to be placed on the play area along the lines of the grid.

Show grid

Displays the grid on the screen. You can change the colour of the grid using the palette.

Origin

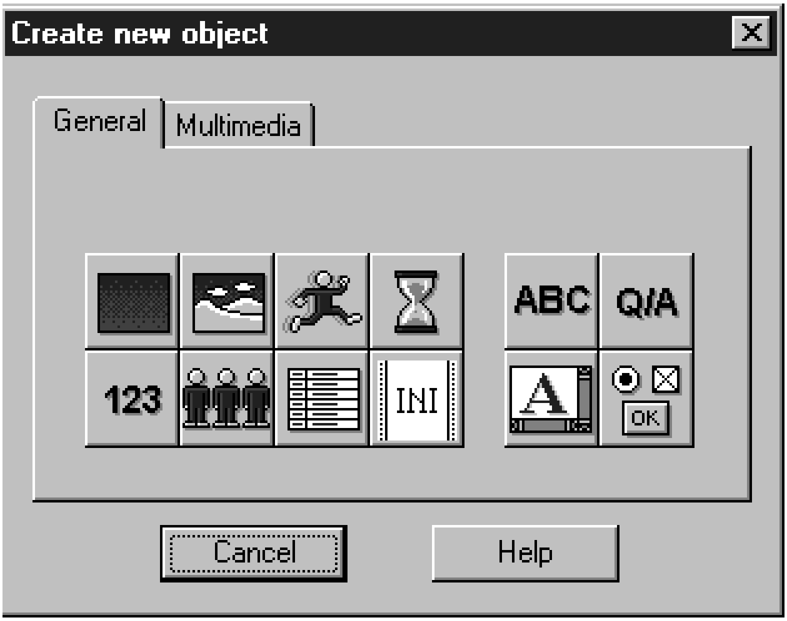
Chooses a point for the origin of the grid, allowing you to move the whole grid. If you have selected an object before entering the Editor preferences screen, you can use that object as the point of origin of the whole grid by clicking on the 'Selected object coordinates'.

Square size

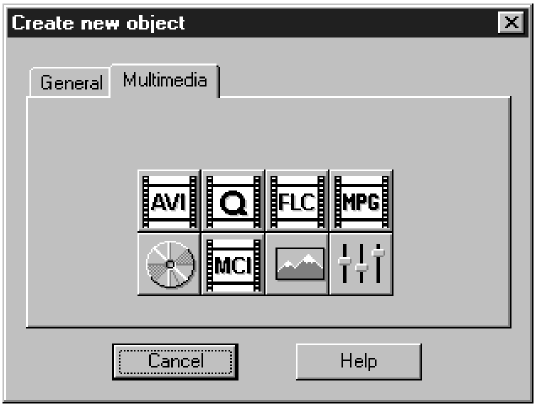
Changes the square size of the grid. Again, if you have selected an object before entering the Editor preferences, you can use the size of that object to select the size of the grid by clicking on the 'Selected object dimensions' box.

Creating new objects

It is from the Level editor that you can create new objects, such as counters, text, new backdrops etc, by using the icon. This will then take you to a menu where you can decide on the type of object to create. For more detail on the individual objects, go to the Objects chapter on page 223.



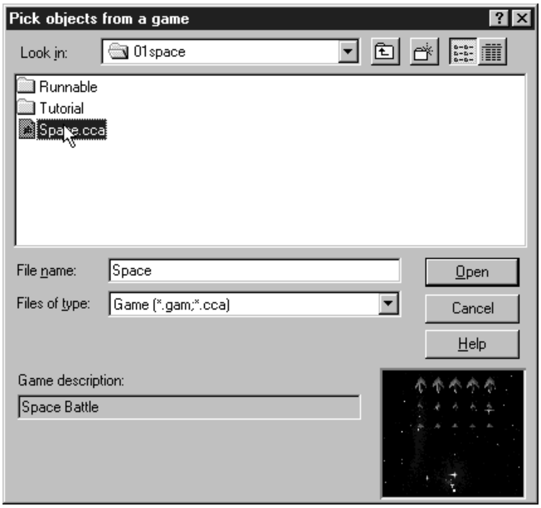
See over for 'Multimedia' layer.



Grabbing objects from another game

If you like the look of objects from another game, or if you spent a long time creating a particular object from another game and you want it in your current game, you can grab it from that game, as long as the game is on disk. Simply go to the ‘Objects’ pull down menu (bar at the top of the screen), then choose the ‘Pick objects from a game’ option.

This then takes you to a screen where you can select the game you want. You must specify the correct drive for this. As an example, try grabbing the objects from one of the tutorial games that are supplied on the CD. You must specify the CD drive to do this, then chose the game from the window on the left.



Look in: 01space

File name: Space

Files of type: Game (*.gam;*.cca)

Game description: Space Battle

Testing your Level

Note: You can use all of these buttons from the Storyboard editor, Level editor, the Step through editor and the Event editor.

You can test the level at any time using the ‘Video’ player buttons on the top right of the screen. These allow you to test a level, or run a level as a complete game.

Run Game

This will run the whole game from level one onwards. Normally only used when you are near completion of your entire game.

Run Frame

This button runs just the frame that you are currently working on, but as though it were being played as a game. Normally used when you are close to finishing a Level.

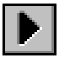
Restart

When you have used the Play and Pause button, re-sets everything to its original position and value.

Step Through editor

Takes you to the Step through editor. It will run your game as normal, but when an event occurs (collision, going off screen, being destroyed etc, etc,) it will pause the game and ask you what you want to happen when that event occurs. For example, if there is a collision, do you want to play a sample, destroy an object, change the score etc.
Can be very useful for the inexperienced as it will fill the Event editor with events which you can add actions to later. However, can be very time consuming as frequently a lot of events will occur that you want absolutely nothing done about, but the Step through editor will pause the game every time an event, no matter how small, occurs. For more information go to the chapter Step Through Editor on page 292.

Play / pause



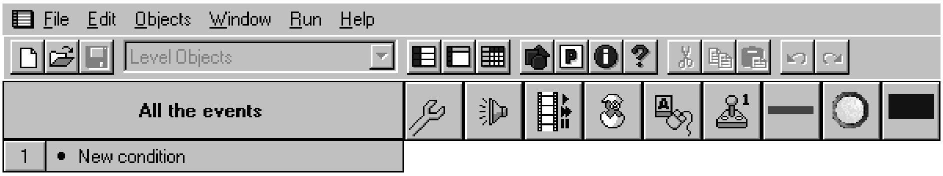
Will normally be displayed as the play icon unless you are ‘playing’ a level, when it is shown as the ‘pause’ icon. Differs from the ‘Run Game’ and ‘Run level’ buttons in that it will allow you to very quickly test new conditions that you add to a game, pausing it at a very particular point and examine what is happening. If you have paused a game and wish to continue modifiying it, it is best to use the ‘Restart’ button after pausing the game.

The Event Editor

The Event Editor is what allows you to really decide what goes on in your games.
It is here that you can define alien movement, assign noises and effects, change scores and lives plus several other special functions.

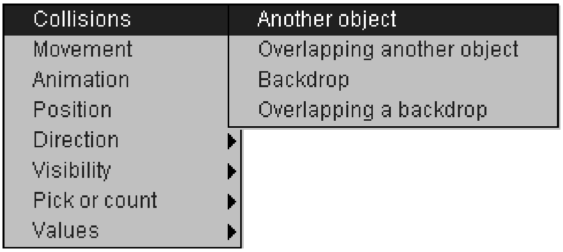
Everything that can happen in your game can be represented by an EVENT.
For example, ‘Bullet collides with Alien’.
You then decide what happens when a particular event line occurs.
This is called an ACTION.
For example, you could insert the Action ‘Destroy’ to the example above.
You insert this in the box that lies on the line ‘Bullet collides with Alien’, directly underneath the icon of the object you want to destroy.
You can also add CONDITIONS to your Events, which limit the Events.
For example, if you had an Event ‘Bullet collides with Alien’, you could add the Condition, ‘and Aliens health value is 0’ then do...
If this were used in the example above, then the alien would be destroyed if a bullet collided with it and the aliens health value was 0.

Before you go to the Event Editor screen, try placing a bat, ball and brick onto the screen from the Tutorial (Bat & Ball) object library. When you have done this, go to the Event Editor screen, which should now look like this, with the top of the Event grid at the top of the screen

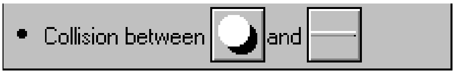


Note that all the Active Objects you placed on the Level Editor are now displayed on the top of the Event Grid, and down the left hand side of the screen is space for all the events. Number 1 should be ìNew Conditionî. The New Condition box is where you decide on an event, e.g. ‘Collision between bat and ball’. You then decide on an Action to be assigned to this event. You do that by clicking on the tick box underneath the character you want to change, on the same horizontal line as your Event.
As an example, click on the ‘New Condition’ box. You will then be presented with an ‘Object to test’ dialogue box. Now you can select the object that has an Event happen to it.

Now, when you right click (or double left click) on the ball, you will get a whole list of options, like this;

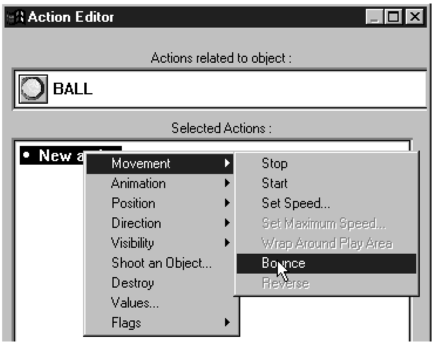


Select the 'Collisions' option. Then select the 'Another Object' option. You will then be asked to select the object the ball collides with from another dialogue box. Select the Bat, then 'OK'.
Now you should have a Event line that reads like this;



You can actually read this as follows;
'(When there is a) Collision between the ball and the bat (do the following Actions...)'

You can then insert an Action by clicking under the column of the object that you want something to happen to. If you now go to the vertical column under the ball and click it, you will be presented with a screen that list all the Actions that happen with that Event. Now, click the 'New Action' text and you will be presented with the menu headed 'Movement'.(See below.) Click on this and another menu headed 'Stop' will appear, as below.



You should then select the 'Bounce' option, then select 'OK' to go out to the Event Editor screen. Now you should see the tick box filled, and if you move the mouse pointer over the tick box, it should be flagged 'Bounce'.

So, when the ball collides with the bat, the ball bounces. You could have selected several options from the Balls' menu, like changing its visibility, direction or speed, or even all of them at once.

Now you have felt your way around the basic structure of the Event Editor, we shall explore the various different features in more detail. If you feel you need more worked examples, simply go to the Tutorial games on pages 11.

Events

Important!

Editing Events

All the Events of the game are listed down the left hand side of the Event grid, as described earlier in this chapter, and are contained in their own numbered box. You can have more than one Condition in an Event. For example, a plain event would be 'When ball collides with bat' do...

You can add a Condition to this by using the right mouse button and clicking the text of the Event you want to add to. Then select the 'Insert' option. This will let you add your new Condition.

Negating Events / Conditions

Something very important that you can do to your Events or Conditions is insert a 'Negate' function in front of them, which reverses the functioning of the line.

For example, if you had an Event line that was 'If Counter Object = 0, then...' and you inserted a 'Negate' in front of it, then the Event logic would be 'If the Counter Object is not = 0 then...'

To insert a negate, you first need to insert your Event or Condition onto an Event line, then right select on it and choose the 'Negate' option. The Negate function is denoted by a red cross in front of the Event or Condition.

Getting around the Event Editor screen

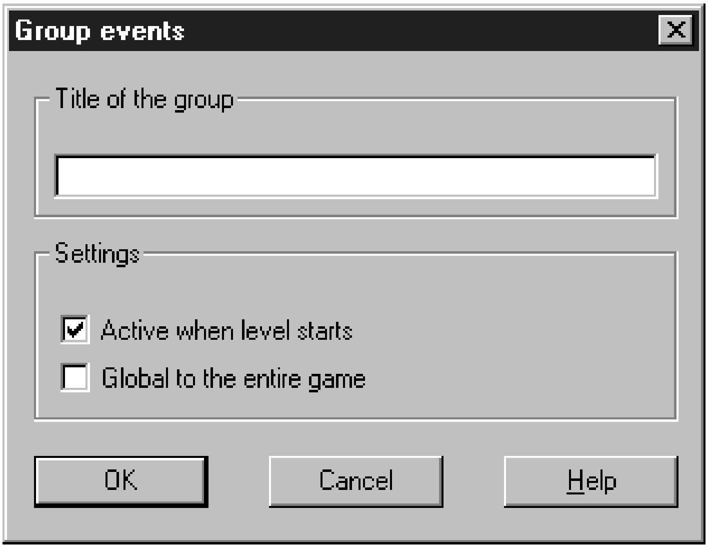
When you start to add large numbers of events and actions, it can be difficult to see what is going on, not to mention that not all of your games events and actions may fit on the screen at the same time. You can move the display around by using the scroll bars at the side of the screen. The vertical scroll bar will simply scroll upwards and downwards through the list of events.

The left and right scroll bars will actually scroll left and right through all the different objects that you have in your game. In many games you may well have many different Active objects, which will be displayed across the top of the screen. If they will not fit on the screen, then The Games Factory will let you scroll the whole Event grid to change your view. The Event lines will remain stationary, so you will always know which Event line you are on.

You can display only the events and actions related to an object by clicking on its icon at the top of the Event Editor screen. By doing this you can get a far clearer view of the action and events related to one object at a time. You can reverse this simply by clicking on that object again.

Grouping Events

You can group events together with a heading to make your games construction far easier to follow, as well as cutting down on the screen space needed to view your events.
For example, you can select a group of events and put them under the heading, 'Alien movement', or 'Screen scrolling control'.
You can do this by selecting the numbered part of a 'New condition' line with the right mouse button. Then select the 'Insert' option, which will pull down a further menu.
Select the 'A group of events' option. You will then be presented with a dialogue box like this;



Group events

Title of the group

Settings

☑ Active when level starts

□ Global to the entire game

Here you can enter the name of the group, this will be displayed at the top of the group.

'Active when frame starts' means that the group of Events will be active upon starting the frame. By de-selecting this option all the events and actions within that group will not happen until you make that group active. You can do this from the 'Special Actions' object. See Special Actions, on page 176.
'Global to entire application' means that all the events within that group will be copied into every frame of your game. If you are going to use the same movement for your hero on every level, this

can save you time re-writing events.
You can also copy all the events by using the Storyboard Editor screen. See page 110 for more information.

Once you have typed in a title, select the 'OK' button. You will now have an Event line that contains only text. If you move your mouse pointer over the group you will see that it is flagged by the text 'Double click to open / close group'
Before you can do that, though, you must put in the events that you want in that group. To generate an event you simply click on the 'New Condition' box below the group title.
You can repeat this process as many times as you like. See the Tutorial games for many examples.

Page Index

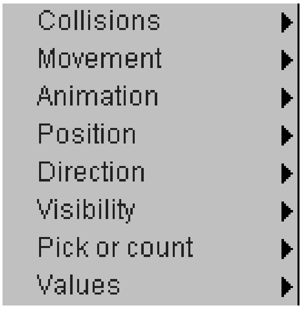
As the number of different objects is so large, we have included this page reference to help you find your way around all the different Events and Actions.

Object type	Event	Action
Active	Page 130	Page 165
Special	Page 137	Page 175
Sound	Page 141	Page 177
Timer	Page 142	N/A
Storyboard	Page 143	Page 179
Create	Page 144	Page 184
Mouse & Keyboard	Page 146	Page 185
Player 1-4	Page 150	Page 186
Text	N/A	Page 187
Question & Answer	Page 152	Page 188
Counter	Page 153	Page 189
High Score	Page 154	Page 189
Edit	Page 154	Page 191
AVI & QuickTime	Page 156	Page 193
MPEG	Page 158	Page 194
FLI / FLC	Page 159	Page 196
CD Audio	Page 160	Page 197
MCI	Page 161	Page 197
INI	N/A	Page 199
Picture	Page 162	Page 201
Button	Page 163	Page 202
Mixer	N/A	Page 204

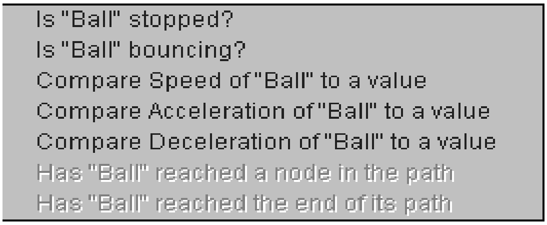
Active object events

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘active’, then ‘events’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable. You should look at the Event Editor of each of these to see how it was done.

The following menu is the first menu that you will see when you select a new condition from an active object;

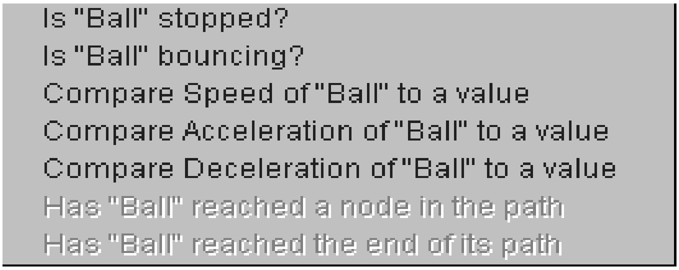


Collisions


Is "Ball" stopped?
Is "Ball" bouncing?
Compare Speed of "Ball" to a value
Compare Acceleration of "Ball" to a value
Compare Deceleration of "Ball" to a value
Has "Ball" reached a node in the path
Has "Ball" reached the end of its path

As briefly outlined previously, lets you decide upon an action to happen when there is a collision. As you have already decided upon the first object, all you have to do is decide upon the second object, the one the first collides with. You can assign this to be a group of objects if you like, by selecting the groups’ icon. This is where it can be very useful to assign all aliens to be ‘Baddies’ using their ‘Object preferences’. (Level Editor screen.) For example, you could have a group called ‘Bullets’ and a group called ‘Baddies’ and whenever they collide, decide on one or more actions, e.g. destroy Bullet, destroy Baddy, play an explosion noise. If you have many different bullets and many different this will save you having to write out a condition for every single collision.
Look up on the CD: Minituts\Active\Events\Active01, Active02

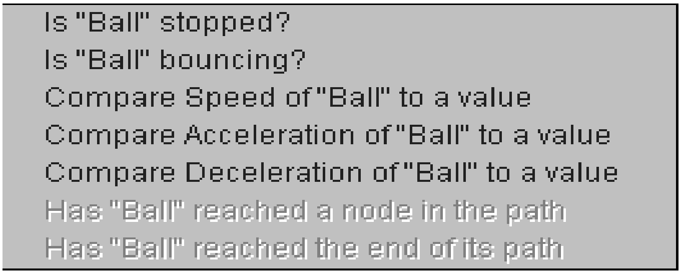
Movement



Is "Ball" stopped?
Is "Ball" bouncing?
Compare Speed of "Ball" to a value
Compare Acceleration of "Ball" to a value
Compare Deceleration of "Ball" to a value
Has "Ball" reached a node in the path
Has "Ball" reached the end of its path

This allows you to test what is going on in the object's movement.
In the above example, you could, for instance, test to see if the ball has stopped moving. If it has, you could set it moving again, destroy it, play a sample of music, etc.
You could compare the speed of an object to a value, and if it is too fast, either slow it down, blow it up, or even change the animation sequence, from walking to running, for example.
Look up on the CD: Minituts\Active\Events\Active03.

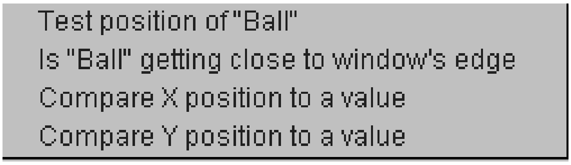
Animation



Is "Ball" stopped?
Is "Ball" bouncing?
Compare Speed of "Ball" to a value
Compare Acceleration of "Ball" to a value
Compare Deceleration of "Ball" to a value
Has "Ball" reached a node in the path
Has "Ball" reached the end of its path

This lets you decide on something to do depending on what is going on with a particular animation. For example, you could have a character be invulnerable to attack if they are 'Crouching', or play a sample of up-beat music when a character is running, etc.
You can use this function to get rid of explosion animation's when they have finished running. You would have an Event 'Has animation (of the explosion) finished? if so, do...(Destroy explosion object).
Look up on the CD: Minituts\Active\Events\Active04, Active05, Active06.

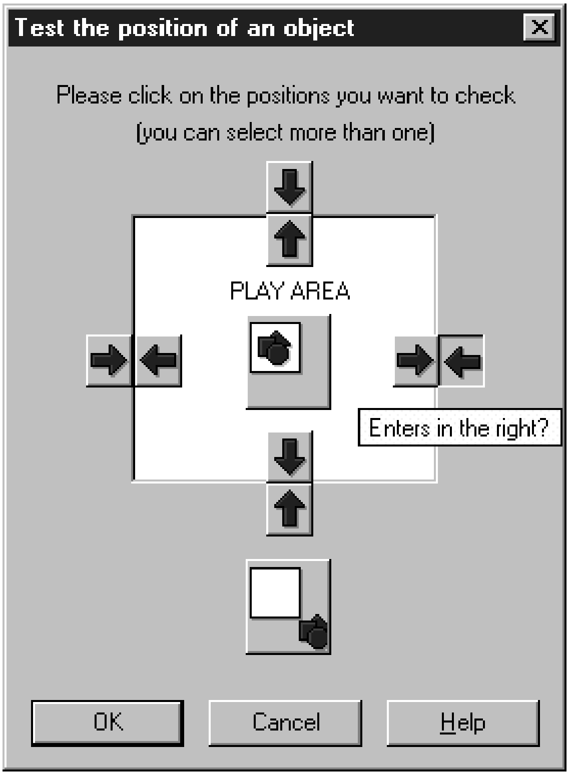
Position



Test position of "Ball"
Is "Ball" getting close to window's edge
Compare X position to a value
Compare Y position to a value

Look up on the CD: Minituts\Active\Events\Active07, Active08, Active09

Test position’



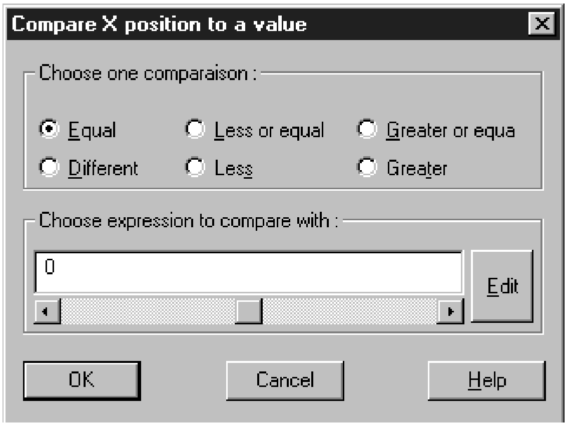
Test the position of an object

Please click on the positions you want to check (you can select more than one)

This is one of The Games Factory’s more useful features from the Event Editor screen. It lets you keep track of the objects as they fly around the screen, and decide what to do when they go off the edge of the screen. In a ‘Breakout’ type of game for example, you could test to see if the ball exits on the top and sides of the screen, then make it bounce back again, but if it leaves at the bottom of the screen you would want to destroy the ball and subtract 1 from the players lives, then

create a new ball in the middle of the playing area. (Done in the appropriate Action columns.)
As you can see above, a text balloon has been called up 'Enters in the right?' There are similar text balloons displayed for each of the other icons, so don't worry if you can't remember what each one means, you'll be told as you go along. If you were to select the 'Leaves in the right?' option, then you would have a condition that read, '(If) Active object [X] leaves in the right (Then do...)'

'Test coordinates'
Comparing the X and Y values can be very useful indeed for 'Invisible zones' on the screen that if the player goes over, will set off an Action or movement.



Compare X position to a value

Choose one comparaison :

● Equal ○ Less or equal ○ Greater or equa
○ Different ○ Less ○ Greater

Choose expression to compare with :

0

You can have the X or Y compared to a specific number by typing the number directly into the 'Choose expression to compare with', or you can compare it to that of another object. For example, 'Is Y coordinate of ball greater than or equal to that of the bat?'
You can select the coordinates of a specific object by selecting the 'Edit' box from the above screen, then when you are presented with the screen to edit the expression, select the 'Retrieve data from an object' box. You will then be shown a window with all the Level objects in, and then be asked to chose one.
See also 'Expression Editor' on page 205
Look up on the CD: Minituts\Active\Events\Active10

Direction

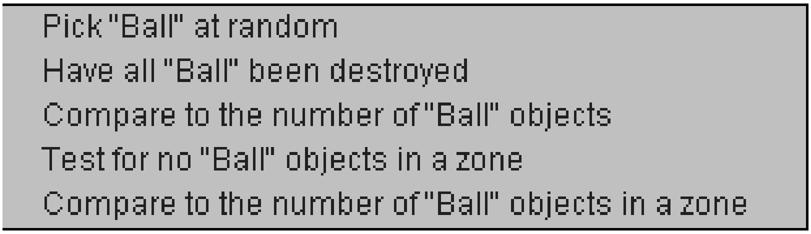
You can ask to compare the direction of an object to a specific direction, or the direction of another object. So, for example, 'is Active object [X] in direction (up), etc., or, 'is Active object [X] in direction of Active object [Z]. You will be presented with a 'Clock-face' direction selector if you are selecting a specific direction. If you want to compare the direction to that of another object,

select on the ‘1+1’ button, then the ‘Edit’ buttons until you get to the ‘Retrieve data from another object’ button. You should read the chapter Expression Editor for more information. See page 205
Look up on the CD: Minituts\Active\Events\Active11

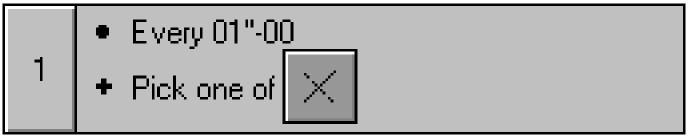
Visibility

Checks on the current state of visibility of an object. e.g. ‘is Active object [X] visible?’ then, for example, it can be hit by bullets, etc.
Look up on the CD: Minituts\Active\Events\Active12

Pick or Count


Pick "Ball" at random
Have all "Ball" been destroyed
Compare to the number of "Ball" objects
Test for no "Ball" objects in a zone
Compare to the number of "Ball" objects in a zone

‘Pick [X] at random’ will choose one of those particular active objects, at random. You can then assign an action to happen to it when it has been picked. Note: This function is best used with the timer function to slow down the rate at which objects are picked. For example;

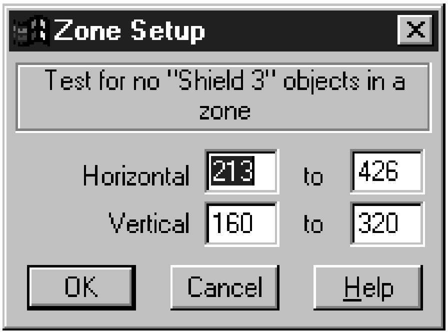

1
● Every 01"-00
✚ Pick one of

This can be done by inserting a timer event. Once you have set up the event to pick an object at random, select on that event with the right mouse button, then select the ‘Insert’ option. You can then select the timer, then select the ‘Every’ option. This will take you to a screen where you decide on the time elapsed. When you have finished you will end up with a condition that looks like that shown above.
Look up on the CD: Minituts\Active\Events\Active13

Handy Hint!
'Have all [X] been destroyed' will let you decide upon an action. Particularly useful for deciding the end of a level.
For example, you could have a test like 'Have all (Group Baddies) been destroyed' attached to the action 'Go to next level' etc.
Look up on the CD: Minituts\Active\Events\Active14

'Compare to the number of [X] objects' lets you compare the number of a type of object to a specific value.
For example, you could test to see if there were more than ten baddies on the screen. If so, don't allow any more on the screen. Or, you could compare the number of goodies and baddies, and when one is greater or less than the other, have an Action occur.
See Expression Editor' on page 205
Look up on the CD: Minituts\Active\Events\Active15

'Test for no [X] objects in a zone' will test to see if there are any [X] objects in a portion of the screen. You can choose the size and placement of the zone using the 'Zone Setup' editor, shown below. You can resize the pick box using the left mouse button on the pick points on the sides of the box.


Zone Setup

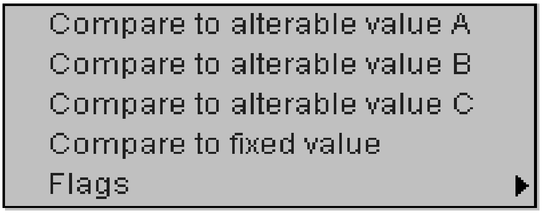
Test for no "Shield 3" objects in a zone

Horizontal 213 to 426
Vertical 160 to 320

Look up on the CD: Minituts\Active\Events\Active16

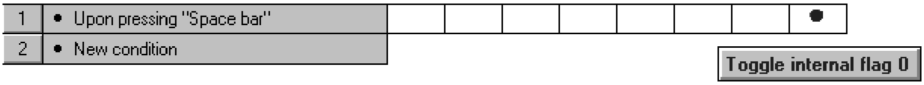
'Compare to the number of [X] objects in a zone' lets you do the same as above, but you can decide on the number of objects. You use a 'Zone setup' editor identical to that shown above.

Values


Compare to alterable value A
Compare to alterable value B
Compare to alterable value C
Compare to fixed value
Flags ▶

Each Active object has its own set of internal variables, a fixed value, plus many internal 'flags'. You can compare the variables to a value, or to the internal variables of another object.
See 'Expression Editor' on page 205
Look up on the CD: Minituts\Active\Events\Active17

Handy Hint!
The flags option can be very useful indeed. A flag is a 'switch' within an object that can either be on or off. Each flag is assigned its own number, so you can read or change a flag by its number. For example, to have a direction change button in a game, all you would need to do would be to have the direction decided by reading whether a flag was on or off.
You could then 'Toggle' that particular internal flag, i.e. if the flag is on, turn it off and vice versa. You could do this when a key is pressed.
The whole Event would look like this;

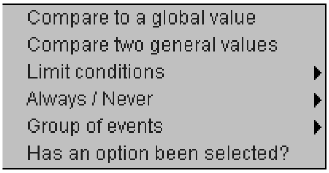


Then, elsewhere in the game, you would have an Event to read the internal flag 0, and change the movement accordingly.

Special object events

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘special’, then ‘events’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable. You should look at the Event Editor of each of these to see how it was done.

You will notice that the icon for ‘Special’ occurs in the ‘Object to test’ window. If you select it you will be presented with a menu that looks like this;


Compare to a global value
Compare two general values
Limit conditions ▶
Always / Never ▶
Group of events ▶
Has an option been selected?

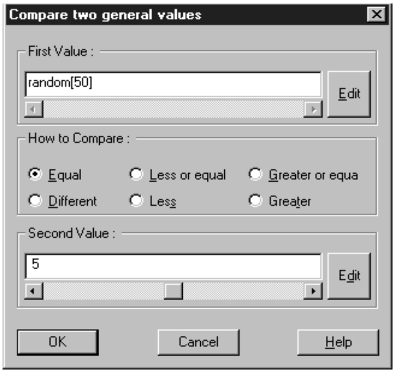
This menu will allow you to perform a variety of special functions that do not occur elsewhere.

Refer to the ‘Expression Editor’ (page 205) chapter for more information on how to define values that you will use for the following functions.

‘Compare to a global value’ allows you to read the value of a global variable. A global variable is one which occurs throughout all the levels of the game. By using this function you can compare the value of a global variable to either a number or an expression (like the X-coordinate of an Active object, for example).
Look up on the CD: Minituts\Special\Events\Specie01

‘Compare two general values’ allows you to compare two variables or values. You could for example use it to compare the X-coordinate values of two different Active objects.
Look up on the CD: Minituts\Special\Events\Specie02

Random Numbers



Compare two general values

First Value : random[50]

How to Compare :

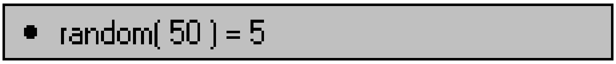
● Equal ○ Less or equal ○ Greater or equa

○ Different ○ Less ○ Greater

Second Value : 5

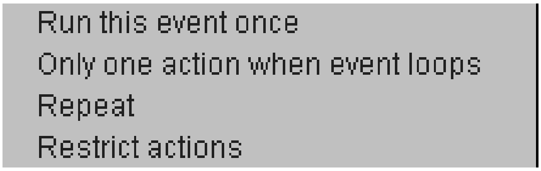
You can create a random number using the ‘Compare two general values’ by using the Random [X] function. The example shown below will create a random number from 0 to 49. The second line of the comparison will then see if the number is 5. This means that when the computer generates a random number, and that number is five, then do...(the Actions that you decided) You create the random function by selecting in the empty space of the ‘First line’ box, (it could also be in the second line) and then simply type in ‘Random (the number of choices you want)

You will then end up with an Event that looks like this;



• random(50) = 5

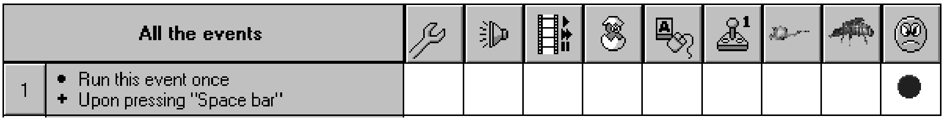
‘Limit conditions’



Run this event once
Only one action when event loops
Repeat
Restrict actions

‘Run this event once’ limits the number of times the Actions for that Event will be performed, to once only.
For example, you might be allowed once per level to depress a key that will destroy all the aliens on the screen. (See ‘Keyboard events’ later in this chapter.) You would create an Event to detect

when that key is pressed, then insert the ‘Run this event once’ condition, then in the action column, destroy the group ‘Aliens’. See the chapter ‘Objects-Object preferences’ on page 226 to learn how to group objects.



	All the events									
1	• Run this event once + Upon pressing "Space bar"									●

Only one action when event loops’ means that the Actions will only be performed once even if the Event loops continuously.

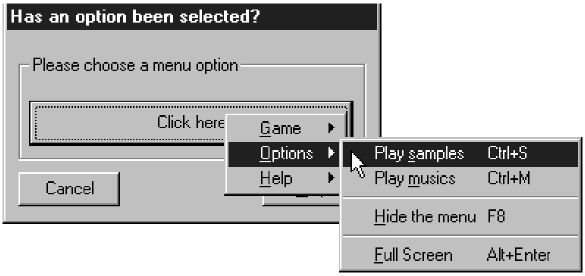
‘Repeat’ will repeat the Actions either a set number of times, or you can enter an expression, e.g. Random(5) will repeat the actions anywhere between one and five times. See the Expression Editor chapter on page 205 for more information.
‘Restrict actions’ will perform a given set of actions after a given period of time, very similar to the ‘Every’ condition. The main difference is that this condition performs the Actions immediately and sequences subsequent executions from when the actions were initially performed.
Look up on the CD: Minituts\Special\Events\Specie06, Specie07, Specie08, Specie09.
‘Always / Never’ allows you to choose between Always and Never. So if you chose Always followed by an Action, The Games Factory will always do that Action.
For example, if you want to centre the play area around your hero, so that when he moves the play area scrolls around him but he stays central on the screen, you would use ‘Always’- (Centre play area around Active Object [X]). To see how to centre the play area go to ‘Storyboard Actions-Scrolling’ on page 180
By selecting the Never option it forbids The Games Factory to do the actions associated with a particular Event line. This is useful when you want to temporarily disable an Event line without deleting it for good. You would insert the ‘Never’ condition to an existing Event line, test the game, and then remove it afterwards.
Look up on the CD: Minituts\Special\Events\Specie05

‘Group of events’ Checks to see if a group of events is active or not. Normally, all groups are active by default, but you can make them inactive via the ‘Special Actions’ function. See page 175 for more information on Special Actions, and see page 128 for information on how to create groups of events.
Look up on the CD: Minituts\Special\Events\Specie03

‘Has an option been selected’ reads whether or not a menu option has been selected from the menu bar at the top of the game’s screen. Note that you are only presented with a menu when you are running the game as a whole using the ‘Run Game as finished’ button. You cannot test it from the ‘Play’ button.

There are several menu options displayed by default, on the menu bar at the top of the screen when you run your game. This function allows you to decide upon an action to do if one of these menu items is selected with the mouse pointer. You can change the look and contents of the menu items via the 'File-Game preferences' option, which can be accessed on any of the Editor screens via the menu bar at the top of the screen. The set-up and editing of the menu's is discussed in more detail in the chapter 'Menu bar options-Game setup' on page 90.

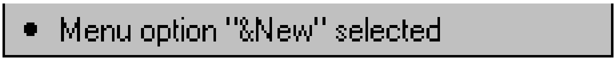
When you select this function you will be asked to select a menu option. What this means is for you to select the menu option that you want to attach an Action to. You will be asked to click in the large button labelled 'Click here'.. When you do this all the menu options will be shown, allowing you to select the one you want to attach an Action to. The following picture shows that one of the options from the 'Application' menu is about to be selected.



Has an option been selected?

Please choose a menu option

If you had selected the 'New' option, you would end up with an Event line like this;



- Menu option "&New" selected

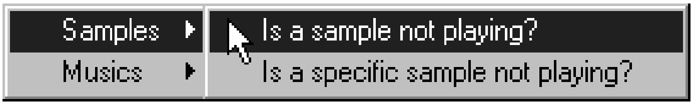
You could then go across the event grid and assign an Action to this event.
Look up on the CD: Minituts\Special\Events\Specie04

Sound object events

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory 'minituts', then 'sound', then 'events'. There are two examples here, which are identified by their file name. You will be prompted to look up each example where applicable.

You should look at the Event Editor of each of these to see how it was done.

This allows you to decide upon an action depending on whether or not there are any samples or music being played. When you click on the sound icon from the 'Object to test' window, you will pull down a menu 'Samples / Music'. This lets you test whether it is a sound sample that is playing, or a music sample.
By selecting 'Music' for example, you will be asked to choose between the two option shown below;



The 'Is a music not playing?' option will test to see if any music is not playing at that time. If it isn't, then do...
Look up on the CD: Minituts\Sound\Events\Sounde01

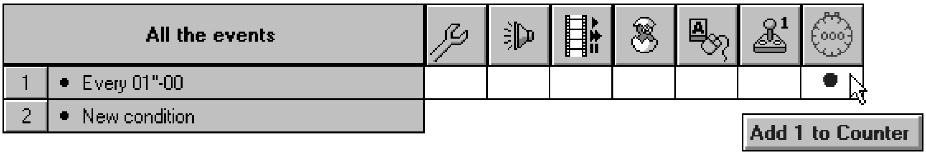
The 'Is a specific music not playing?' option will test to see if a specific, named piece of music is not playing.
The same applies to the sample options, only it tests whether a sample is not playing. This function could be useful for playing two tunes, one after the other. You could have some music played once (see Special functions-limit conditions, previous topic) and then test to see when it ended and then play another music sample.
Look up on the CD: Minituts\Sound\Events\Sounde02

Timer object events

In addition to the information here you can look for the examples on the CD ROM. Look for the directory ‘minituts’, then ‘timer’, then ‘events’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.

You should look at the Event Editor of each of these to see how it was done.

This allows you to choose an Action dependent upon the timer. The timer runs from zero from each level. You could choose to do an action every ten seconds, for example, or to do an action after a certain time period was up.
When you select the Timer object from the ‘Object to test’ screen, you will be given a pull-down menu like this;

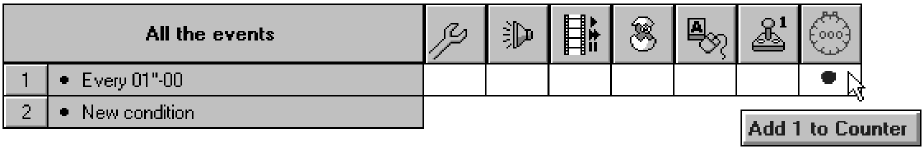

All the events

1 • Every 01"-00

2 • New condition

As briefly outlined in the ‘Handy hint’ below, the ‘Every’ option allows you to do an action repeated every [X] amount of time.

Handy Hint!
To make an on-screen clock, all you would have to do would be to place a counter on-screen (Level editor) and then use the timer event ‘Every (1 second)’, to add 1 to the value of the counter. This will give you an event line like this;


All the events

1 • Every 01"-00

2 • New condition

‘Is the timer equal to a certain value’ tests to see when the timer reaches a time [X]. If it does, then do... Useful for ending a level after a certain time. Note that this will only do an action once when applied to a fixed value.
Look up on the CD: Minituts\Timer\Events\Timere01

‘Is the timer less than a certain value’ tests to see if a time has been reached. If it hasn’t,

do... Note that this will rapidly repeat doing all the associated Actions until that time has been reached.
Look up on the CD: Minituts\Timer\Events\Timere02

'Is the timer greater than a certain value' also tests to see if a certain time has been reached, but note that this option will keep on doing all its' associated actions once the time has been reached. For example, if you set this function up as follows; 'Is the timer greater than ten seconds' then after ten seconds all the associated actions would keep on occurring after ten seconds had elapsed.
Look up on the CD: Minituts\Timer\Events\Timere03

'User has left computer since' checks to how along it has been since the user (you) moved the mouse or depressed a key. If it has been longer than the time you specify, then this can be used to start a demo rolling when there's no user activity for example.
You may well need to add the Special event 'Run this action only once when event loops' to stop the actions associated with this timer function from being repeated endlessly. See 'Special events-limit conditions' earlier in this chapter.
Look up on the CD: Minituts\Timer\Events\Timere04

Storyboard object events

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory 'minituts', then 'story', then 'events'. There are two examples here.
You should look at the Event Editor of each of these to see how it was done.

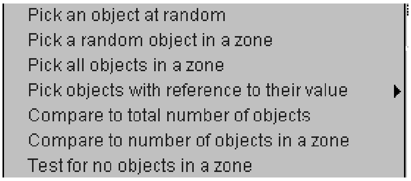
Allows you to decide on an action when it is the beginning of a level, the end of a level, or the end of the game.
Look up on the CD: Minituts\Story\Events\Storye01, Storye02

Create object events

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘newobj’, then ‘events’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.

You should look at the Event Editor of each of these to see how it was done.

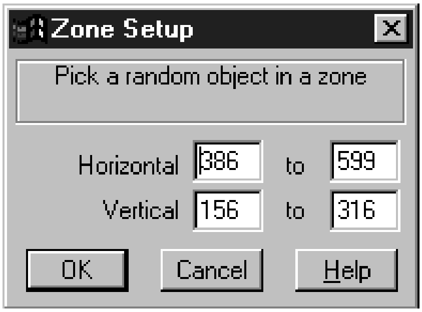
Allows you to select an action depending on several different object parameters, listed below;



Pick an object at random
Pick a random object in a zone
Pick all objects in a zone
Pick objects with reference to their value ▸
Compare to total number of objects
Compare to number of objects in a zone
Test for no objects in a zone

‘Pick an object at random’ will pick an Active object at random and then do the actions that you decide to it, e.g. destroy it, move it, set its position, etc. It’s important to note that the object picked by The Games Factory must have actions assigned to it, otherwise nothing will happen.
Look up on the CD: Minituts\Newobj\Events\Newobje01

‘Pick a random object in a zone’ will pick an Active object at random within a zone that you decide the size and position of. If you select this or any of the other zone options you will be presented with a Zone editor like this;



Zone Setup

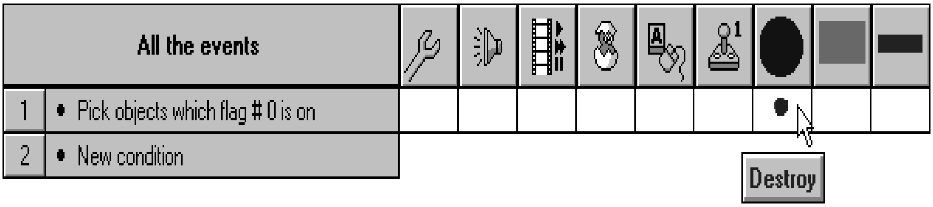
Pick a random object in a zone

Horizontal 386 to 599
Vertical 156 to 316

This allow you to decide on the size and position of the zone on the play area that you want to select your objects from.
Look up on the CD: Minituts\Newobj\Events\Newobje02

'Pick all objects in a zone' will trigger off actions to all the objects in a zone that you setup. See above for Zone setup.
Look up on the CD: Minituts\Newobj\Events\Newobje03

'Pick objects with reference to their value' will read specified internal values of objects. If that specified value is equal to the one you want, then that object will be selected. For example, if the internal flag (0) of the objects is on, then all those objects will be selected and have [X] actions (for this example 'Destroy') done to them. See below for an example of what this will look like on an event line.



1 • Pick objects which flag # 0 is on

2 • New condition

Look up on the CD: Minituts\Newobj\Events\Newobje04

'Compare to total number of objects' checks to see what the total number of objects is and compares that to a value.
For example, 'If the total number of objects is greater then four, then do...'
Look up on the CD: Minituts\Newobj\Events\Newobje05

'Compare to number of objects in a zone' does the same as the condition outlined above, but restricts the area to a zone which you decide upon. You could use this to test whether or not a 'room' is full, for example, by making the zone fit the 'room'. See the 'Zone setup', conditions above.
Look up on the CD: Minituts\Newobj\Events\Newobje06

'Test for no objects in a zone' tests to see if there are no objects in a zone that you decide upon. If there are no objects in that zone, then do...
Look up on the CD: Minituts\Newobj\Events\Newobje07

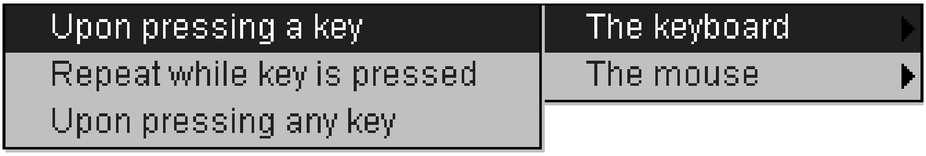
The mouse pointer and keyboard events

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘input’, then ‘events’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.

You should look at the Event Editor of each of these to see how it was done.

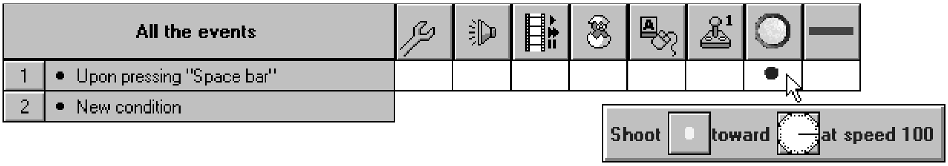
This is one of The Games Factory’s more important features as it controls how the user interacts with the game. By default you can already change the direction of animated objects using the joystick and cursor keys, but sometimes manipulating their X & Y coordinates is more useful. You can do this directly by using different keys. What if you want your main character to pick something up, fire a gun, eat something? All of these can be done from here.
You also have access to what is happening to the mouse pointer, checking to see if it is in a certain position on the screen, whether or not it is over a particular object, etc., etc.
Below is a picture of what you get when you select the ‘Mouse pointer and keyboard’ option from the ‘Object to test’ screen.

The Keyboard



‘Upon pressing a key’ means that every time you press a particular key your [X] actions will be done. Note that the events will only occur once per key depression, even if you keep the key depressed. To repeat the actions whilst the key is depressed, see the next item.
When you select this option you will be asked to press a key. This is the key that you want to use. If, for example, you wanted the player to fire a gun when the spacebar is pressed, you would go to the ‘New condition’ box, click on it with the right mouse button, then select the ‘Mouse pointer and keyboard’ object (right or double-left click), then select the ‘Upon pressing a key’ option. Then press the spacebar to choose that as the action key. Once you have done this you would go across the Event grid on the same horizontal line as the ‘When spacebar is pressed’ event to the tick-box underneath your main character. In here you could select the Action, ‘Shoot an object’.
For more information on firing objects go to the ‘Event editor-Actions-Shoot an object’ section on

page 171. When you have done this you will end up with an Event line that looks something like this;



All the events									
1	• Upon pressing "Space bar"							●	
2	• New condition								

Shoot toward at speed 100

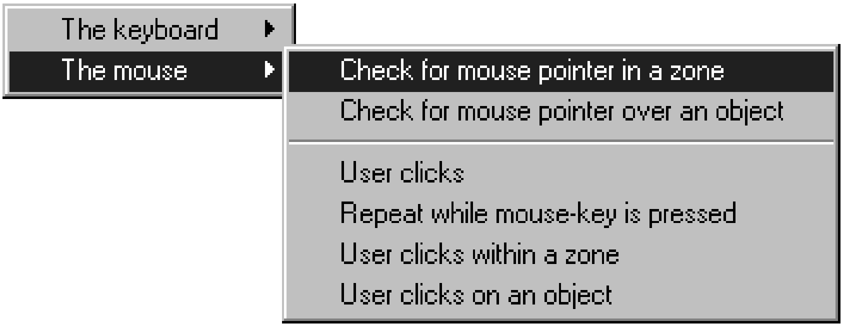
Look up on the CD: Minituts\Input\Events\Inpute01

'Repeat while key is pressed' will keep on repeating those actions as long as that key is held down, useful for shooting things rapidly, but not for picking objects up!
Look up on the CD: Minituts\Input\Events\Inpute02

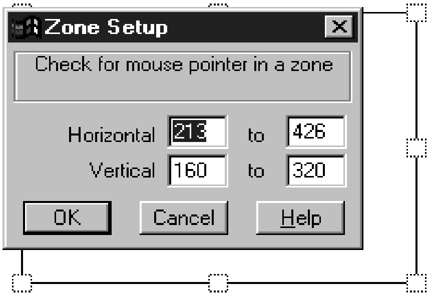
'Upon pressing any key' will do all [X] actions when any key is depressed. Useful on game introduction screens.
For example, when any key is pressed, move onto the next level.
Look up on the CD: Minituts\Input\Events\Inpute03

The mouse pointer

When you select the ‘mouse’ option, you will be presented with another menu, like that below.


The keyboard
The mouse
Check for mouse pointer in a zone
Check for mouse pointer over an object
User clicks
Repeat while mouse-key is pressed
User clicks within a zone
User clicks on an object

‘Check for mouse pointer in a zone’ tests to see if you have moved the mouse pointer over a particular part of the screen. You decide on the size and position of that zone.
You will be presented with a zone editor like that overleaf;


Zone Setup

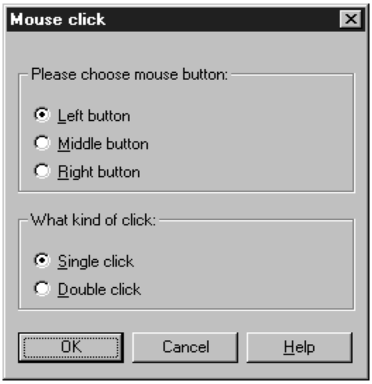
Check for mouse pointer in a zone

Horizontal 213 to 426
Vertical 160 to 320

When the mouse pointer is in that zone, the associated actions that you assign to that event will occur. You do not have to click either of the mouse’s buttons to activate this event. Note that you may well have to limit this event from continuously performing the actions by inserting a Special condition to run only once when the event loops. See ‘Editing events’ at the beginning of this chapter and ‘Special object events-limit conditions’ on page 138.
Look up on the CD: Minituts\Input\Events\Inpute04

‘Check for mouse pointer over an object’ checks to see if the mouse pointer is over an object. As with the condition above this function will repeat the associated actions endlessly unless you insert a Special object condition. See above item for references.
Look up on the CD: Minituts\Input\Events\Inpute05

‘User clicks’ tests to see if the user has clicked one of the mouse buttons. When you select this option you will be presented with a screen that looks like this;



Mouse click

Please choose mouse button:

● Left button
○ Middle button
○ Right button

What kind of click:

● Single click
○ Double click

From here you can select to detect which button is pressed and the type of click has been used.
Look up on the CD: Minituts\Input\Events\Inpute06
‘Repeat while mouse key is pressed’ will repeat the Actions associated with that event for as long as that key is held down. Again you will be asked to chose which mouse button is being pressed.
Look up on the CD: Minituts\Input\Events\Inpute07

‘User clicks within a zone’ tests to see if the user has clicked the mouse button within a zone that you define. This is a combination of the ‘Check for mouse pointer within a zone’ and ‘User clicks’ options. You will first be asked to decide upon which mouse button to use and the type of click, and then asked to define the zone within which it must be clicked.
Look up on the CD: Minituts\Input\Events\Inpute08

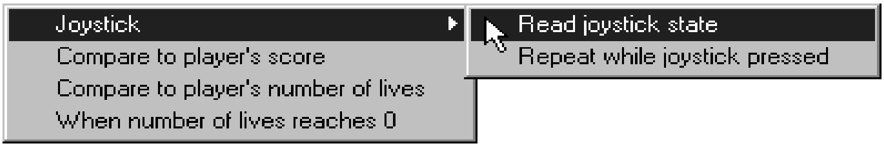
‘User clicks on an object’ tests to see if you have clicked one of the mouse buttons whilst over a specific object. Again, you will be asked to select which button and the type of click, then you will be presented with a dialogue box containing the Active objects from that level. Simply select the object you want.
Look up on the CD: Minituts\Input\Events\Inpute09

Player object events

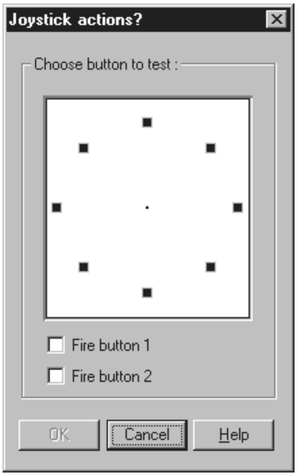
In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘player’, then ‘events’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.

You should look at the Event Editor of each of these to see how it was done.

This group of events lets you test what is happening to the players. Each player is denoted by the joystick icon and the relevant number in the top right hand corner.
Within this group you can test joystick controls, compare the players score to a number or variable, compare a players number of lives to a number or variable, or doing something when the player’s lives reaches zero, like end the game. When you select one of the player objects from the ‘Object to test window’ you will be presented with a menu like this;


Joystick
Compare to player's score
Compare to player's number of lives
When number of lives reaches 0
Read joystick state
Repeat while joystick pressed

‘Joystick’ allows you to test what is happening with a players joystick controls.
There are two options within this. ‘Read joystick state’ will read the current joystick state and do the actions once every time the joystick is moved to that state, whereas ‘Repeat while joystick is pressed’ will repeat the actions for as long as the joystick is in that state. Both options will bring up the following options screen;



You select the direction by clicking on the black squares and the relevant fire button in the tick-boxes. On the picture above no choices have been made.
Look up on the CD: Minituts\Player\Events\Playre01, Playre02

'Compare to a players score' will bring up an Expression Editor that will let you decide on an action when the players score reaches a certain level.
For example, when the players score passes 10,000, you could move up their number of lives by one.
Look up on the CD: Minituts\Player\Events\Playre03

'Compare to players number of lives' will bring up an Expression Editor that lets you compare the number of lives to a specific number, or variable.
For example, you could have a warning message appear when the player is down to his last life.
See the Expression Editor chapter on page 205 for more information.
Look up on the CD: Minituts\Player\Events\Playre04

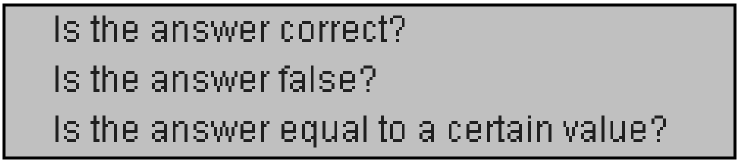
'When players lives reaches 0' means that an action will be done when all the players lives are used up. It is a simpler version of the option above. Normally used to end the game and go to a different screen, such as a high score table. See page 225 for how to set up high score tables.
Look up on the CD: Minituts\Player\Events\Playre05

Question and Answer Object Events

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘question’, then ‘events’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.

You should look at the Event Editor of each of these to see how it was done.

These are used to ask the player a question and retrieve an answer. The answers are retrieved when the player selects an answer ‘button’ on the Question and Answer object. So you must have a question and answer object already placed in the play field - done from the Level Editor.
To bring up the Event options available, right mouse click on a ‘New Condition’, then right mouse click on the Q / A icon that you want to test, from the ‘Object to test’ dialogue box. You will then be presented with a menu that looks like this;



‘Is the answer correct’ tests whether the answer was correct. If it was, do...
Look up on the CD: Minituts\Question\Events\Queste01

‘Is the answer false’ tests to see if the wrong answer was given. If it was, do...
Look up on the CD: Minituts\Question\Events\Queste02

‘Is the answer equal to a certain value’ asks to see if the number of the answer (i.e. was answer number [X] given?) is equal to a numerical value. This can be a set number, or a value retrieved from an expression. See the Expression Editor chapter on page 205 for more information.
Look up on the CD: Minituts\Question\Events\Queste03

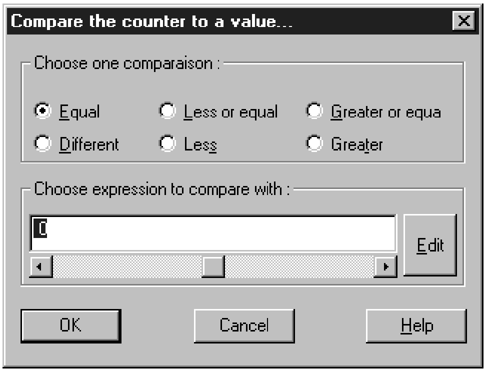
Counter Object Events

In addition to the information here you can look for the example on the CD ROM.
Look for the directory ‘minituts’, then ‘counter’, then ‘events’ then ‘counte01’
You should look at the Event Editor of this to see how it was done.

Reading counter objects is one of The Games Factory’s most useful game creation features. You can read the values of counters to decide on alien movement, speed, energy levels, or for your player’s health, object store, food supply, speed, temperature, in fact anything that you want to count the value of.

They are so useful that you will probably find yourself using many different counters in your games. In order to make it easier to distinguish between them, you can change the name and icon of each one from the Level Editor by right mouse clicking on them. For more detail on this turn to page 249 of the Objects chapter.

To access the Counter’s events, right mouse click on a ‘New Condition’ and then right mouse click on the Counter from the ‘Object to test’ dialogue box. You will then be presented with one option only - ‘Compare counter to a value’. Selecting this will bring up the following dialogue box;



Compare the counter to a value...

Choose one comparaison :

● Equal ○ Less or equal ○ Greater or equa
○ Different ○ Less ○ Greater

Choose expression to compare with :

0

You can then choose the way you wish to compare the value of the counter, (greater than, less then, etc.) and the value you want to compare it to. You can enter a fixed value either by entering the numbers directly, or by using the slider.

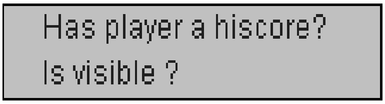
This value can also be an expression, that is, a mathematical function, or a value retrieved from another object, like its speed, direction, internal flag, etc., etc. See the Expression Editor chapter on page 205 for more information.

High Score Object Events

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘hiscore’, then ‘events’. There are two examples here, which are identified by their file name. You will be prompted to look up each example where applicable.

You should look at the Event Editor of each of these to see how it was done.

High Score objects are used to display not only your previous best score, but those of your friends, so you can add an exciting competitive edge to your games. No more lying about the score you got in your games, it’s up there on the High Score table, or you didn’t do it!
You can choose the options available as events for your High Score object by right mouse clicking on a ‘New Condition’ then right mouse clicking on the High Score object from the ‘Object to test’ dialogue box. You will then be presented with the following menu;



‘Has a player a High Score’ tests to see if a player has managed to get onto the High Score table. You may want to play a fancy tune if they have, for example.
Look up on the CD: Minituts\Hiscore\Events\Hiscoe01

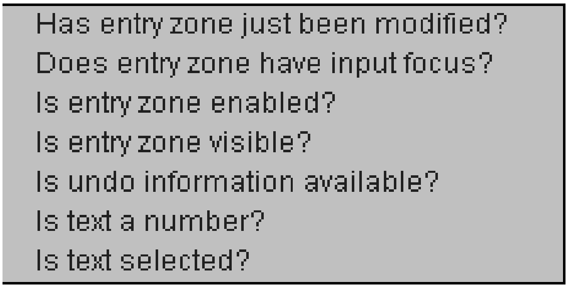
‘Is visible’ tests to see if the High Score table is visible. If it is, then do...
Look up on the CD: Minituts\Hiscore\Events\Hiscoe02

Edit Object Events

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘edit’, then ‘events’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.

You should look at the Event Editor of each of these to see how it was done.

Edit Objects are used for entering and retrieving text from the player. You can choose the options available as events for your Edit object by right mouse clicking on a ‘New Condition’ then right mouse clicking on the Edit object from the ‘Object to test’ dialogue box. You will then be presented with the following menu;



Has entry zone just been modified?
Does entry zone have input focus?
Is entry zone enabled?
Is entry zone visible?
Is undo information available?
Is text a number?
Is text selected?

‘Has entry zone just been modified?’ tells you if the player changes something in the Edit box. If they have, do...
Look up on the CD: Minituts\Edit\Events\Edite01

‘Does entry zone have input focus?’ asks if the edit zone has been selected by the user. If it has, do...
Look up on the CD: Minituts\Edit\Events\Edite02

‘Is entry zone enabled?’ tests if the zone can be selected by the user. You can disable an Edit box, making it inaccessible to the user. This tests if the Edit box is enabled.
Look up on the CD: Minituts\Edit\Events\Edite03

‘Is entry zone visible?’ tests to see if the object is visible on the screen. If it is, do...
Look up on the CD: Minituts\Edit\Events\Edite04

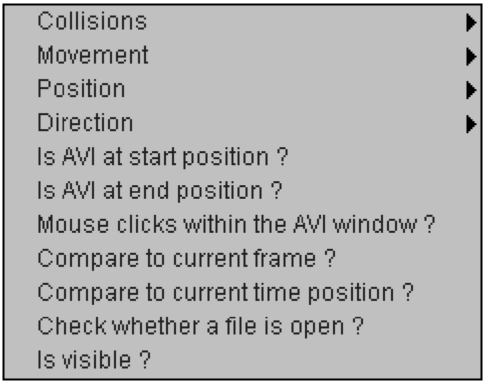
‘Is undo information available?’ tells you if the undo option is available, i.e. you can perform the function ‘Undo’ on whatever you have just done.
Look up on the CD: Minituts\Edit\Events\Edite05

‘Is text a number’ asks if the text is a whole number. If it is, do...
Look up on the CD: Minituts\Edit\Events\Edite06

‘Is text selected?’ asks if the player has highlighted some text with the mouse. If they have, do...

AVI Object Events

AVI objects are used to play video clips within a game. You can choose the options available as events for your AVI object by right mouse clicking on a ‘New Condition’ then right mouse clicking on the AVI object from the ‘Object to test’ dialogue box. You will then be presented with the following menu;


Collisions
Movement
Position
Direction
Is AVI at start position ?
Is AVI at end position ?
Mouse clicks within the AVI window ?
Compare to current frame ?
Compare to current time position ?
Check whether a file is open ?
Is visible ?

The ‘Collisions’, ‘Movement’, ‘Position’ and ‘Direction’ are all handled in a similar way to that of Active objects. See the ‘Active Objects Events’ on page 130 for more information on these options. Note also that QuickTime objects share almost all of these options.

‘Is AVI at start position’ tests if your Video for Windows object is at the beginning. You could use this condition to play a piece of music, run a music clip from the CD, or start a complex animation sequence in the background when the Video for Windows object begins.

‘Is AVI at end position’ tests if your Video for Windows object is at the end. You could use this condition for loading another video sequence, wait for a command from the user, move to the next frame, or stop the background music when the Video for Windows object ends, etc., etc.

‘Mouse clicks within the AVI window’ tests if the user has clicked in the video viewing window. If so, do...

‘Compare to current frame position’ tests if the video sequence has reached a specific frame. If so, do...

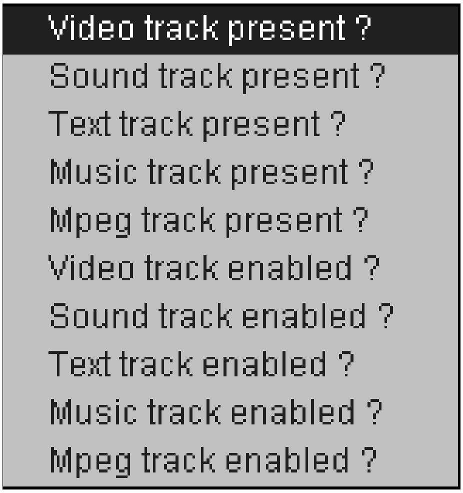
‘Compare to current time position’ waits until a selected period of time has elapsed from the start of the video. If it has, then do...

‘Check whether a file is opened’ enables you to test that there is a video sequence in memory, before you attempt to access it.

‘Is Visible?’ will be True if the AVI object is currently being displayed on the screen. You can conceal AVI objects by enabling the ‘Hide on Start’ check box in the Setup dialogue box or by using the ‘Make object invisible’ action from your application. You can display the object again with the Make object Reappear action.

QuickTime Object Events

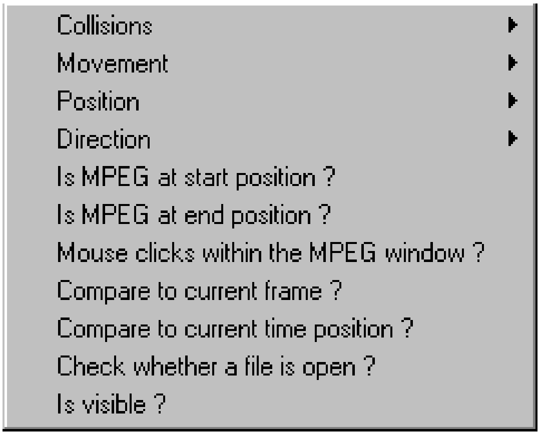
QuickTime objects are almost identical to AVI objects and share almost all the same parameters. However, in addition to the Events above, you have an additional option, ‘Tracks’. Selecting this will pull-down this menu;



These test to see if that parameter is true. If it is, then do [X] actions, normally related to the QuickTime object. If you want, QuickTime videos can have subtitles linked to them.

MPEG Object Events

Motion Picture Expert Group objects are used to play video clips using special MPEG drivers. You will need either an MPEG card fitted to your machine, or a software MPEG card emulator installed. You will then be able to play MPEG files.
This is the menu that you will be presented with when you right click on the ‘New condition’ text and select an MPEG object from the ‘Object to test’ window.


Collisions ▸
Movement ▸
Position ▸
Direction ▸
Is MPEG at start position ?
Is MPEG at end position ?
Mouse clicks within the MPEG window ?
Compare to current frame ?
Compare to current time position ?
Check whether a file is open ?
Is visible ?

‘Collisions’ detects if there is a collision between an MPEG object and another object on the play area in the same way as for Active Objects. For more detail look up Active Object Events, Collisions, on page 1.

‘Movement’ allows you to read the current state of movement of an MPEG object, again in much the same way as for an Active object. See Active Object Events, Movement, on page 131.

‘Position’ will test the position of the MPEG object in much the same way as for an Active object. See Active Object Events, Position, on page 132.

‘Direction’ tests to see if the MPEG is facing a particular direction. You can select the direction it is being tested with using a ‘Clock-face’ direction selector.

‘Is MPEG at start position’ tests to see if the MPEG sequence is a at frame number 1.

‘Is MPEG at end position’ tests to see I the MPEG sequence is on the last frame.

‘Mouse clicks within the MPEG window’ tests to see if the user clicks a mouse button whilst the mouse pointer lies within boundaries of the window being used to display the MPEG.

‘Compare to current frame’ tests the current frame number of the MPEG sequence to a number or expression. You could compare it to the value of a counter, or a players score, or just a particular number.

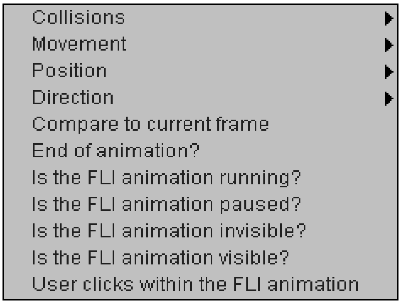
‘Compare to current time value’ tests the current time value of the MPEG sequence to a particular time. You can choose the time in minutes, seconds and hundredths of a second, and whether or not to compare it is as being less than or greater than the value you enter.

‘Check whether a file is open’ tests to see if an MPEG file has been opened.

‘Is visible’ test to see if the MPEG is currently visible. If it is, do...

FLI / FLC Objects Events

FLC / FLI objects are used for displaying animations within your game. They can be treated in many ways like Active objects. You can choose the options available as events for your FLI / FLC object by right mouse clicking on a ‘New Condition’ then right mouse clicking on the FLI / FLC object from the ‘Object to test’ dialogue box. You will then be presented with the following menu;


Collisions
Movement
Position
Direction
Compare to current frame
End of animation?
Is the FLI animation running?
Is the FLI animation paused?
Is the FLI animation invisible?
Is the FLI animation visible?
User clicks within the FLI animation

For ‘Collisions’, ‘Movement’, ‘Position’ and ‘Direction’ see Active Objects Events on page 130.

‘Compare to current frame’ allows you to compare the current frame number to a value. This can be a specific number, or it can be an expression. See the Expression Editor chapter on page 205 for more information.

‘End of animation’ checks to see if the animation has finished. If it has, do...

‘Is the FLI animation running?’ checks to see if the animation is running at that moment. If it is, do...

‘Is the FLI animation paused?’ checks to see if the animation is paused. If it is, do...

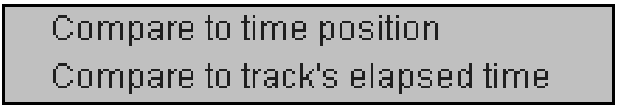
‘Is the FLI animation invisible’ checks to see if the animation is invisible. If it is, do [x], like make it visible, for example.

‘Is the FLI animation visible’ checks that the animation is visible. If it is, do...

‘User clicks within FLI animation’ checks to see if the user has placed the mouse pointer within an FLI object while the animation is running. You could use this to detect if the user has correctly selected a frame, or by extracting the coordinates of the mouse pointer as well, check the position of the mouse pointer also. See the Expression Editor chapter on how to retrieve values from objects.

CD Audio Object Events

CD Audio objects are used to play tracks from an audio CD whilst your game is running. You could therefore use it for moody background music for your game.
You can choose the options available as events for your CD Audio object by right mouse clicking on a ‘New Condition’ then right mouse clicking on the CD Audio object from the ‘Object to test’ dialogue box. You will then be presented with the following menu;

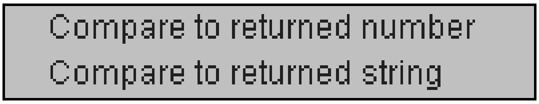


‘Compare to time position’ compares the current elapsed playing time to a value. This can be a specific value, e.g. 2mins 10secs.

‘Compare to track’s elapsed time’ – as above but only the time value of the current track.

MCI Object Events

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘mci’, then ‘events’, then ‘Mcie01’.
You should look at the Event Editor of this to see how it was done.


Compare to returned number
Compare to returned string

These are used to control multimedia devices. You can choose the options available as events for your MCI object by right mouse clicking on a ‘New Condition’ then right mouse clicking on the MCI object from the ‘Object to test’ dialogue box. You will then be presented with the following menu;

‘Compare to returned number / string’ compares a returned number / string to a value, such as a CD track number / name, or an expression.

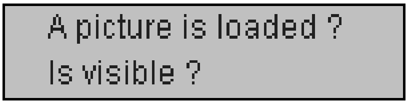
For example, you could enter the text ‘Play C:\samples\comic\ping.wav’ into the Expression Editor of an MCI, it would play the file ‘ping.wav’ from the directory ‘comic’, from the sub-directory ‘samples’. Note that these are fictitious file names, just used for example purposes only.

Picture Object Events

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory 'minituts', then 'picture', then 'events'. There are two examples here, which are identified by their file name. You will be prompted to look up each example where applicable.

You should look at the Event Editor of each of these to see how it was done.

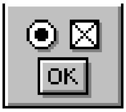
Picture objects are used for displaying pictures loaded in from disk. They can be used for slide shows, or as backdrops, loaded in as and when required. You can choose the options available as events for your Picture object by right mouse clicking on a 'New Condition' then right mouse clicking on the Picture object from the 'Object to test' dialogue box. You will then be presented with the following menu;



'A picture is loaded' tests to see if your picture object has loaded in a picture from disk. If it has, do...
Look up on the CD: Minituts\Picture\Events\Pictue01

'Is visible?' tests to see if your picture object is visible. It can be made invisible initially from its 'Setup' menu option (Level Editor screen) or via its Actions.
Look up on the CD: Minituts\Picture\Events\Pictue02

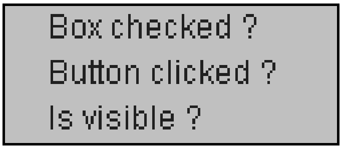
Button Object Events



In addition to the information here you can look for the examples on the CD ROM.
Look for the directory 'minituts', then 'button', then 'events'. There are two examples here, which are identified by their file name. You will be prompted to look up each example where applicable.

You should look at the Event Editor of each of these to see how it was done.

Button objects are used to provide a 'button' that the user can 'press' using the mouse pointer. You can use them for the user to enter choices in an adventure game, changing a CD track, or choosing a level to go to. When you right select a Button object from the 'Object to test?' dialogue box, you will be presented with a menu that looks like this;



'Box checked' tests to see if there is a check-mark in that particular button. Note that only certain types of button can have check-marks placed in them.

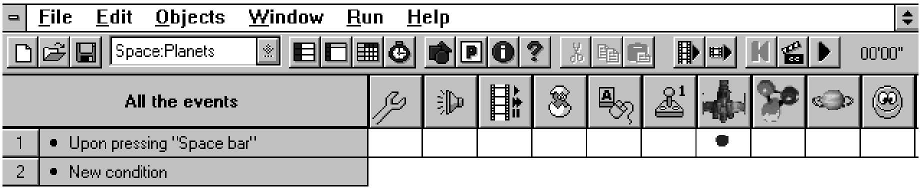
'Box clicked' tests to see if the user has clicked on the button with the mouse pointer. If they have, do...
Look up on the CD: Minituts\Button\Events\Button01

'Is visible?' tests to see if the button object is visible. If it is, do...
Look up on the CD: Minituts\Button\Events\Button02

Actions

Once you have decided upon an Event, you want to decide what to do when that Event occurs. When you press the fire button on the joystick, do you fire an object, or pick something up? When your bullets hit the aliens, do they destroy them, or reduce their energy? Do you play a sound sample when your hero falls off a cliff, and change his animation so that he flails his arms and legs around? All these are decided upon by deciding what Actions to assign to your Events. The previous chapter dealt with all the possible Events, this will deal with all the Actions. We shall start with the Active objects actions first, dealing with special actions later.

To place an Action to a particular event and a particular object, you look down the left hand side of the screen to find the event that you want, follow this horizontally across the screen until you get to the object that you want to have an Action happen to.



All the events

1 • Upon pressing "Space bar"

2 • New condition

In the example shown above an Action has been inserted into the tick box associated with the ‘Jet’ active object. This means that whenever the space bar is pressed, that Action will occur to that object.

There are two ways to place an Action in the relevant tick-box, the fastest is by using the right mouse button, which will instantly pull down a menu of choices. You can do it with the left mouse button, which will present you with an Object actions dialogue box. You add a new action by clicking the ‘New action’ text.

Both ways will produce the same menu options, but using the right mouse button avoids using the dialogue box. It can be handy to use the dialogue box instead, when you already have many actions already assigned to the same tick-box, as it will list them at the same time and allow you to change the order in which they are triggered.

You can view all of the actions assigned to a particular tick-box simply by moving the mouse pointer over the tick box. The actions will be shown in a box underneath.

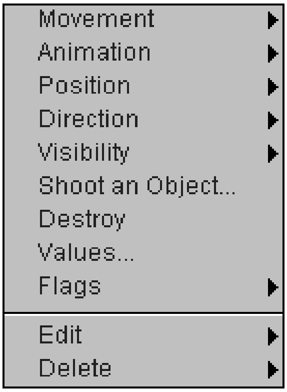
Edit / Deleting Actions

To edit or delete an action, select the tick-box with the right mouse button, then the relevant choice (edit / delete). If there is more than one Action associated with a given Event, then you will be shown them as a list, and asked to choose one to edit / delete.

Active objects Actions

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory 'minituts', then 'active', then 'actions'. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable. You should look at the Event Editor of each of these to see how it was done.

Primary Menu



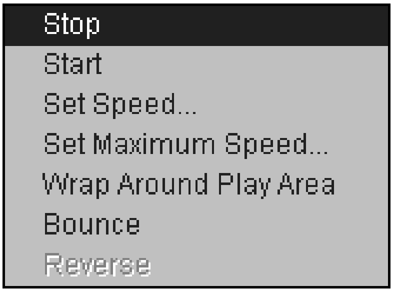
The primary menu is where you decide on the type of action you are going to assign to an event.

We will deal with these in order throughout this chapter.
Movement-see below.
Animation-see page 167
Position-see page 169
Direction-see page 170
Visibility-see page 171
Shoot an object..-see page 171
Destroy-see page 173
Values-see page 173
Flags-see page 174

Movement

Note: You will only be allowed to change the movement of an object in the Event Editor if you have already set the movement up in the Level Editor. If you want to have an objects movement decided entirely from the Event editor, simply give the object a Bouncing ball movement, then set all the parameters to zero. This will give you a ‘blank page’ to work with.

By entering the movement option, you will be presented with another menu, like this;



Note that not all the options are always available. For example, you may have an event line ‘Test to see if object leaves play area’ and therefore the option ‘wrap around play area’ will not be available. You can only ‘Reverse’ objects with a path-type movement.

You can only wrap an object around the play are if it leaves the play area. It will soon become common sense which actions you can apply to a given object or event. The event associated with the above menu was ‘Test to see if object leaves play area’, hence you can select ‘Wrap around play area’.
‘Stop’ simply stops the object from moving.

‘Start’ simply starts the object moving in its previous manner.

‘Set Speed...’ sets the speed of the object. This can be to a specific value or to an expression. See the Expression Editor chapter on page 205.
Look up on the CD: Minituts\Active\Actions\Activa01

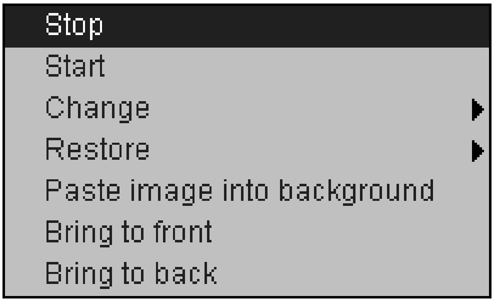
‘Set maximum speed’ puts a speed limit on an object, perhaps to slow it down. Note this value can also be set to either a fixed value or to an expression.

‘Wrap around play area’ will bring an object around to the other side of the screen if it exits on one side. If it leaves at the top it will reappear at the bottom, at the left edge it will reappear on the right.

‘Bounce’ imposes a bounce function to the object. Does not necessarily reverse its direction. If you have a high randomisation setting for a Bouncing ball movement, it could spring off anywhere.

‘Reverse’ as mentioned above can only be used when applied to an object with a path-type movement. Simply reverses the movement mid-path.

Animation

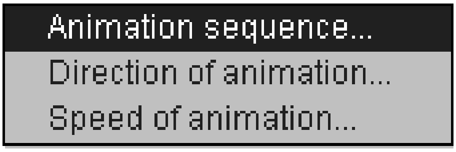


This action menu lets you change what is happening to an object’s animation.

‘Stop’ simply stops an object’s animation from playing, which ever frame it is on.

‘Start’ re-starts an object’s animation from what ever frame it was stopped on.

‘Change’ pulls down the following menu;



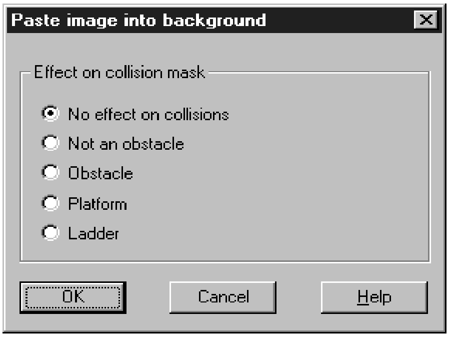
‘Animation sequence’ allows you to change an animation sequence to another one by its name. You can only choose one that has animation frames assigned to it.
For example, you could change an animation from ‘Walking’ to ‘Running’ when an object’s speed exceeds a certain amount. You would construct an Event line ‘If speed of Active Object [X] =>50, then (Action: Change animation to ‘Running’).

‘Direction of animation’ will change the direction of the animation sequence showing. Again, you can only change it to a direction, that already has an animation for that direction.

‘Speed of animation’ changes the speed of the animation to a value that you decide. This can be set as a figure or an expression, or retrieved from another object, such as a counter.

‘Restore’ will pull down a menu similar to that shown above, but selecting each option will restore that parameter to its previous value.

‘Paste image into background’ brings up another dialogue box.
Pasting the image into the background will paste an image of the Active object into the background as you move it around the screen, leaving a kind of ‘snail trail’ around the screen. You can then decide if you want other objects to interact with this trail or not.



Paste image into background

Effect on collision mask

● No effect on collisions
○ Not an obstacle
○ Obstacle
○ Platform
○ Ladder

‘No effect on collisions’ means that whatever image is laid down will not effect the collisions of other objects on the screen. This is the collisions being made between active objects and backdrop objects.

‘Not an obstacle’ will paste the image as a non-obstacle.

‘Obstacle’ will paste the image as an obstacle to all other active objects.

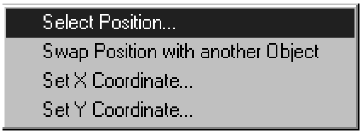
‘Platform’ will paste the image as a platform.

‘Ladder’ will paste the image as a ladder.
Look up on the CD: Minituts\Active\Actions\Activa02

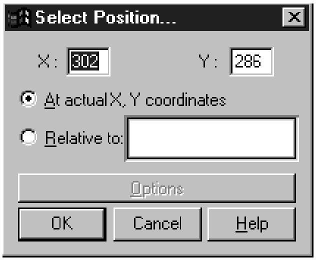
‘Bring to front’ changes the animated objects view, i.e. this function will put the animated object ‘in front’ of all other objects on the screen.

‘Bring to back’ pushes the animated objects view ‘behind’ all the other images on the screen.

Position



‘Select position’ will place an object at specific coordinates on the screen. You can do this either by typing in the X and Y coordinates directly, or by moving the mouse pointer. You can also select its position relative to another object on the screen by clicking in the ‘Relative to’ box, then click on the object you want to the position to be relative to. If X and Y are set to 0, your object will be placed where the crosshair that you can see appears.


Select Position...

X: 302 Y: 286

● At actual X, Y coordinates

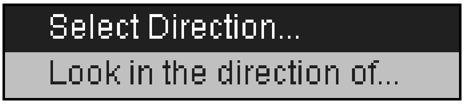
○ Relative to:

‘Swap position with another object’ does exactly that. note that the object you are going to swap position with must also have the ‘Swap position’ action inserted on the same event line. Also note that you can have 2 or more objects swapping position.

‘Set X coordinate’ allows you to set the X coordinate of the object to a specific value or expression, or to that of another object. See ‘Expression Editor’ on page 205

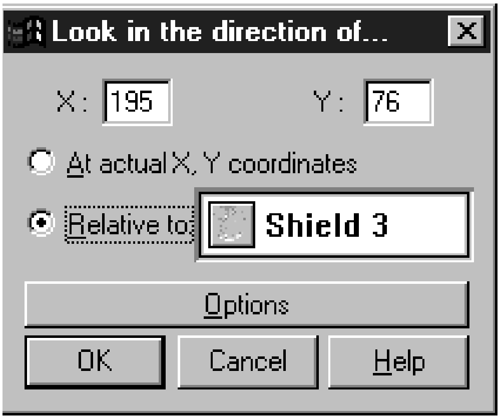
‘Set Y coordinate’ – as above but for the Y coordinate.
Look up on the CD: Minituts\Active\Actions\Activa03

Direction



'Select direction' allows you to select a specific direction using a direction 'clock-face'.
Look up on the CD: Minituts\Active\Actions\Activa04

'Look in the direction of' lets you select a direction from either the coordinates that you type in or specify with the mouse, or that of another object.



Look in the direction of...

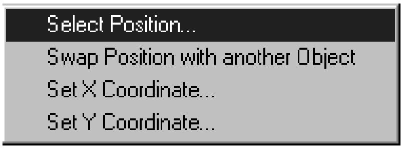
X: 195 Y: 76

○ At actual X, Y coordinates

● Relative to: Shield 3

Handy Hint!
This function can be very useful as the basis for defining the movement of objects that follow each other round the screen. You would enter the object that you wanted to chase after in the 'Relative to' box, then make sure the X and Y coordinates were set to 0. Note that this will mean that if there are no obstacles in the way that you would be followed unerringly by that object. You could link this to a Timer Event (See page 142) to make it follow less 'aggressively'.
For example, set a timer event to 'Every 2 seconds' (Look in the direction of [X]).

Visibility



‘Make object invisible’ does exactly that. It still exists on the screen, but is no longer visible.

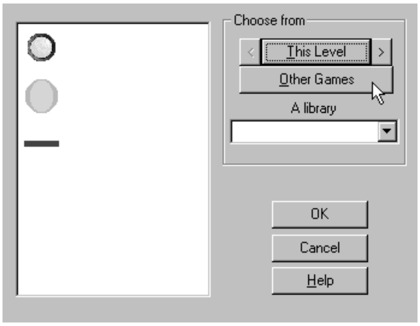
‘Make object reappear’ re-shows an invisible object.

‘Flash object’ will flash an object on and off at a rate that you can choose.

Shoot an object

Fairly fundamental to Shoot-’em-up games, allows you to select the objects that you want to use as bullets, the direction for them to be shot in and their speed.
You can use any Active object as a bullet, including objects from other levels and games.

Once you have fired your bullet you could add a further event line to ‘Look in the direction of’ to create your own guided missiles!
When you choose the ‘Shoot an object’ option you will be presented with the following dialogue box;

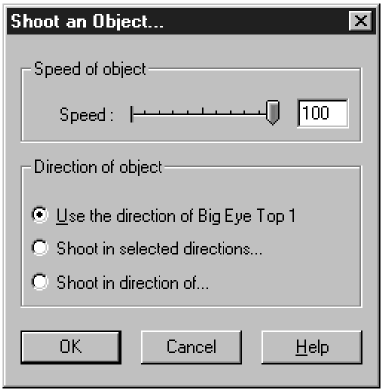


This is where you choose the object that you want to use as your bullet.
You can either choose from the objects in the dialogue box, or from another frame of your game.
You do this by pressing the direction buttons either side of the ‘This frame’ text with the mouse pointer.

‘Other games’ means that you can grab an object from another game, as long as you have it saved on disk.

‘A library’ lets you look through all the object libraries The Games Factory has to offer, and choose your bullet from one of them.

Once you have chosen the object that you want to use as your bullet, you will be asked to select the direction in which you want it to be fired. You will be presented with a dialogue box like this;



Shoot an Object...

Speed of object

Speed : 100

Direction of object

● Use the direction of Big Eye Top 1
○ Shoot in selected directions...
○ Shoot in direction of...

From here you can select the speed of the object using the slider at the top.
Underneath you have several choices for the direction that you want to fire the object in.

‘Use the current direction of [X]’ will read the current direction value of the object you are firing from, and impose that direction to the bullet.

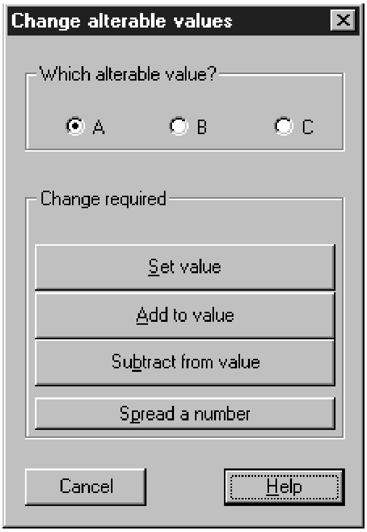
‘Shoot in selected directions’ allows you to select a specific direction using the direction ‘clock-face’. Note that if you select more than one direction here, The Games Factory will choose between the directions you have selected at random.

Handy Hint!
‘Shoot in direction of’ allows you to fire the bullet to either a specific coordinate on the screen, or in relation to another Active object. See ‘Direction’, previous topic.
This is very useful for defining the aiming of the bullets of your baddies. You would simply select the direction to be in the direction of the hero’s character and when the baddies fired their bullets they would always be fired in the hero’s direction.

Destroy

Removes the object from the game screen. Does not just make it invisible, but removes it entirely. Note that many animated objects have a 'Destroy' animation associated with them, such as an explosion. If there is such an animation associated with an active object, when it is 'destroyed' this animation will automatically be played before the object is removed.
Look up on the CD: Minituts\Active\Actions\Activa05

Values



Allows you to manipulate the internal values A, B and C of that object. This can be done to a specific number or to an expression. For more information on expressions, see the 'Expression Editor' chapter on page 205.

'Set value' sets that variable to a specific value or expression.

'Add to value' adds to the value of that variable by a specific number or expression.

'Subtract from value' subtracts from the value of that variable by a specific number or expression.

'Spread a number' assigns a number to each of one of the duplicates of that active object, starting with the number that you specify. For example, if you were to place ten of one type of object on the screen and then spread Alterable value A from 1, if you read the value of A from each one, it would be a number from 1 to 10, starting with the object that you first placed on the screen.

Flags

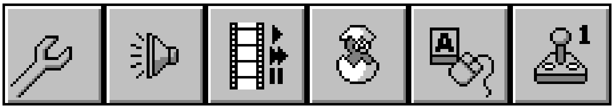
Each active object has a set of flags, called by a number. A flag is a ‘switch’ that can either be on or off. This option allows you to either turn a flag on or off, or to ‘toggle’ it, meaning that if it is on it will be turned off, and vice versa.

You will be asked to select which flag you are going to change. You do this by entering the number of that flag. An active object can have up to 32 flags.

Flags can be used for all sorts of things, such as picking up objects, (If you have picked up Active object [X] then set flag#1 to on, etc.) status, (when Active object [X] has been hit by a bullet, set flag#2 to off, etc.).

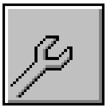
Look up on the CD: Minituts\Active\Actions\Activa06

Game Action Objects



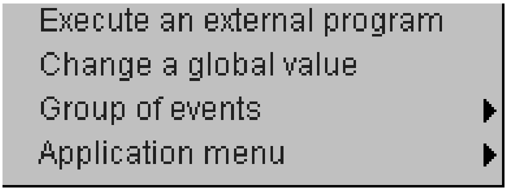
These Objects allow for special Actions in your game, such as changing a level, playing a sound, or creating a new object, like an explosion. Note that these always exist on the top of the Event Editor screen, you do not have to create them via the Level Editor in order to manipulate them.

Special Actions



In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘special’, then ‘actions’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.
You should look at the Event Editor of each of these to see how it was done.

The Special Actions perform actions that are not covered anywhere else

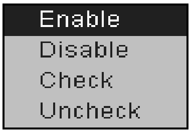


‘Execute an external program’ allows you to run an external program at the same time as you are running your game. Useful if you want to link in .exe files that may have been created via languages like C, Basic or Pascal.
Look up on the CD: Minituts\Special\Actions\Specia01

‘Change a global value’ allows you to change the value of global variables 1 - 16 Changing its value here will change its value throughout all levels of the game. Useful for preserving a value from one level to the next.
Look up on the CD: Minituts\Special\Actions\Specia02

‘Group of events’ will allow you to enable / disable a group of events. You can only use this if you have placed a number of events into a named group. You can then enable / disable that group by name. See Grouping Events on page 128 for information on creating a group of events.
Look up on the CD: Minituts\Special\Actions\Specia03

‘Application menu’ lets you select an action to impose upon the menu bar that is at the top of the screen when you run the game. You have four different options;



All of these options use the following menu selector;

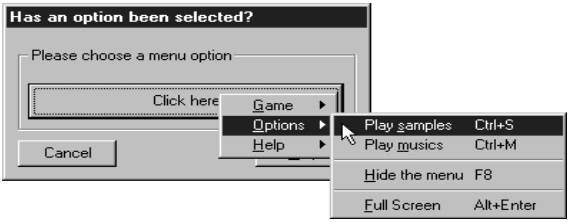
‘Enable’ will re-activate a previously disabled menu option.

‘Disable’ will disable a menu option.
Look up on the CD: Minituts\Special\Actions\Specia04

‘Check’ places a tick mark in a particular menu option.

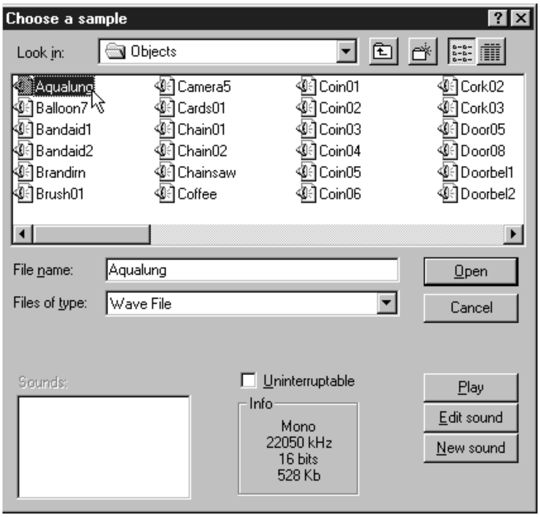
‘Uncheck’ removes the tick mark from a particular menu option.

When you select any of the above options you will be asked to select which menu option you want to apply that condition to. You will be presented with a screen like this;



Sound Actions

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory 'minituts', then 'sound', then 'actions'. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.
You should look at the Event Editor of each of these to see how it was done.



Choose a sample

Look in: Objects

Aqualung	Camera5	Coin01	Cork02
Balloon7	Cards01	Coin02	Cork03
Bandaid1	Chain01	Coin03	Door05
Bandaid2	Chain02	Coin04	Door08
Brandim	Chainsaw	Coin05	Doorbel1
Brush01	Coffee	Coin06	Doorbel2

File name: Aqualung

Files of type: Wave File

☐ Uninterruptable

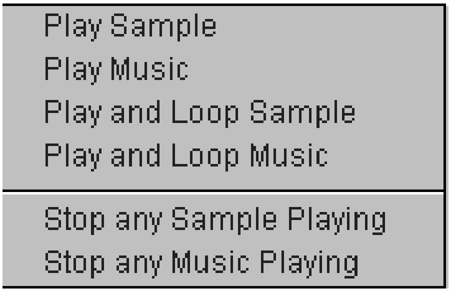
Info
Mono
22050 kHz
16 bits
528 Kb

This allows you to play sound or music when a particular Event occurs. You need to select a sound from the many files The Games Factory has to offer, but don't worry too much, as you can preview the sound before you decide on it.
Note that if you have done a normal install, The Games Factory will need to access the sound and music files from the CD drive, so you must specify your CD ROM drive, and then look for 'Samples'. You will then be presented with several different libraries of sounds. When you select one, its contents will be displayed down the left hand side of the screen, which can be scrolled through if they are very large. If you select a sound file you can 'Play Sound' before clicking the 'OK' button, which will play that sound every time the particular event occurs (you can also just double click on a sound to preview it)..
If you are playing a music sample, you will need to look in the 'Midi' directory.

By selecting the ‘Uninterruptable’ button, you can make that sample or music non-interruptable by any other sample.

‘Edit sound’ allows you to manipulate the sound sample using the ‘Sound recorder; editor. You can change the speed of the sound, add echo, as well as cutting parts from the sample or mixing with other sound files.

‘New sound’ allows you to add your own sound effects to your games using the microphone inlet on your PC. You can manipulate these sounds using the ‘Sound recorder’ editor also.
The initial menu that you will be presented with for the Sound Actions (right select in the tick-box under the Sound icon) looks like this;



‘Play sample’ will play a sample of music once.
Look up on the CD: Minituts\Sound\Actions\Sounda01

‘Play music’ will play a piece of music once.
Look up on the CD: Minituts\Sound\Actions\Sounda02

‘Play and loop sample’ will play a sample repeatedly.
Look up on the CD: Minituts\Sound\Actions\Sounda03

‘Play and loop music’ will play a piece of music repeatedly.
Look up on the CD: Minituts\Sound\Actions\Sounda04

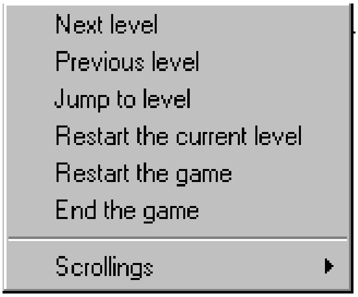
‘Stop any sample playing’ stops all samples from playing at a particular moment in your game.

‘Stop any music playing’ any piece of music being played at a particular moment in your game.

Storyboard Actions

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘story’, then ‘actions’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable. You should look at the Event Editor of each of these to see how it was done.

The Storyboard actions allow you to manipulate the levels in your game, jumping forward a level when all the Aliens have been killed, for example, or going to the ‘Game Over’ screen when all the hero’s lives have been used up.



‘Next level’ will jump to the frame that is next in numerical order.
Look up on the CD: Minituts\Story\Actions\Storya01

‘Previous level’ will jump to the numerically previous level number, not necessarily the level you were just in.

‘Jump to level’ allows you to move on to any level. Simply enter the level number that you want to go to, or you can use an expression to calculate the number of the level that you go to. See Expression Editor on page 205 for more information.
Look up on the CD: Minituts\Story\Actions\Storya02

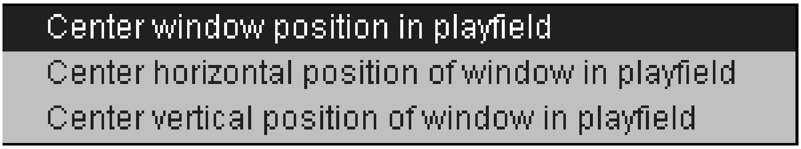
‘Restart the current level’ will re-start the current level, resetting all variables except global score and lives values.
Look up on the CD: Minituts\Story\Actions\Storya03

‘Restart the game’ will go back to the very beginning of the game, resetting all values.

Look up on the CD: Minituts\Story\Actions\Storya04

'End the game' halts the game entirely, and will take you back to an Editor screen from The Games Factory.
Look up on the CD: Minituts\Story\Actions\Storya05

'Scrolling'


Center window position in playfield
Center horizontal position of window in playfield
Center vertical position of window in playfield

For many games that you will be creating you will probably find it necessary to scroll around the play area. One of the easiest ways to do this is by selecting a very large play area from the Storyboard Editor screen, then using the 'Scrollings' option from the Storyboard Actions menu in the Event Editor screen.
If you are going to have a scrolling style the same all the way through your application, it would be a good idea to define an Event as 'Always', (see Special Events, page 137) for more information) and to put it in a group that is Global to the whole application. (See Editing Events on page 127) for more information.)

As an example, we shall take you step by step through scrolling, from a completely new game, to a scrolling system that would be applied to all the different levels of your game.

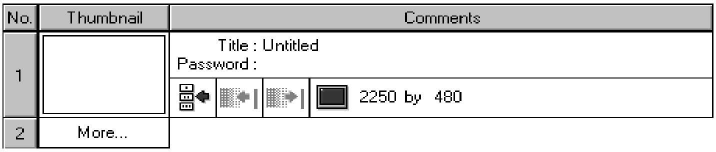
Step One

- Firstly start from scratch by selecting the 'New' option form the File menu.

- From the Storyboard Editor screen, select the X dimension of the play area, (The first of the two numbers that define the play area size) using the left mouse button.

- Enter 2550 as the size of the X dimension. This will give you a play area width of 2550 pixels. Note that this does not fit on the screen, but that you will view a small area in the middle of this. By moving the view point around you can scroll around the play area.

- Make sure the Y dimension is set to 480, which is a fairly standard height for most of the Backdrop objects The Games Factory has already in the object libraries.



No.	Thumbnail	Comments
1		Title : Untitled Password : 2250 by 480
2	More...	

Now go to the Level Editor of that frame.

Step Two

- Go to the ‘Object libraries’ and scroll through until you find the library ‘Space: Backdrops’.
- Select any four of the backdrops and place them on the play area so that they cover it entirely. This is going to be the background that you will scroll around.
- Next, go to the object library ‘Games: Space Walkers’ and select the man in the space suit with a jet pack on. Place him anywhere on the starfield.
- Once he is placed on the starfield, select on him with the right mouse button and choose ‘Movement-Change Movement’ from the menu.
- Choose the ‘Eight Directions’ icon, then ‘OK’ all the way out of the Movement Editor. For more information on movement, turn to page 232
- If you left the spaceman as he was now, you could move him around the starfield using the cursor keys, but the background would stay stationary.

Step Three

- Now go to the Event Editor screen. Select on the number part of the ‘New Condition’ line with the right mouse button. Select the ‘Insert-A group of events’ option.
- Now type in the name of the group into the text box. For this example use the text ‘Scroll control’. Do not use the <RETURN> key when you have finished typing in the name.
- Next, click in the ‘Global to the entire application’ box. This will make this group of events occur in every frame of your game, even the levels that you have not yet constructed. Now select the ‘OK’ button.
- When you have done this you should have a large grey bar with ‘Scroll Control’ on it, and underneath, indented slightly, event No. 2, with No. 3 normally laid out beneath.

Step Four

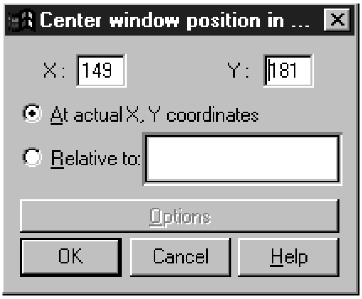
- Now select on the ‘New condition’ text of event No. 2 with the left mouse button.

- Select the icon. From its menu, select ‘Always’. This means that the Actions associated with this Event will always be performed. Event line No. 2 should now have the text ‘Always’

- Now go to the empty tick box along this line that lies directly under the ‘Storyboard’ icon.

- Select in the tick box with the right mouse button, then select the ‘Scrollings’ option.

- Select the Centre window position in playfield’ option from the menu.



Center window position in ...

X: 149 Y: 181

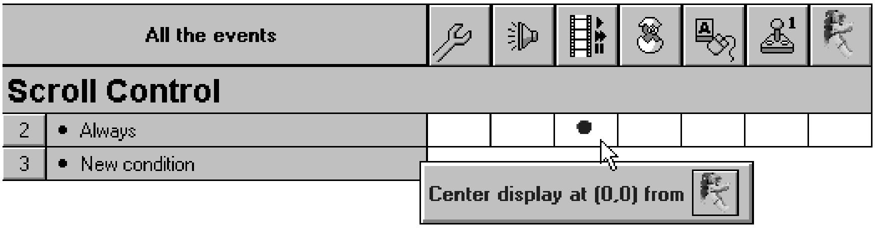
● At actual X, Y coordinates

○ Relative to:

- You will then be presented with a dialogue box like that shown above. You want the screen to be centred about your spaceman, so select in the ‘Relative to:’ box. Then select the spaceman’s icon.

- Now change the X and Y coordinates to 0. This means that the screen will be centred at 0 distance from the spaceman.

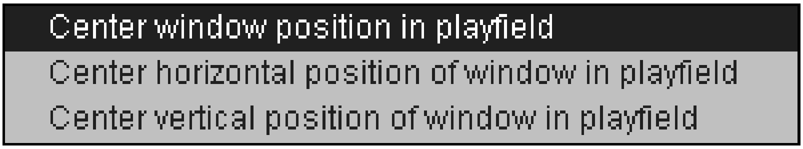
- ‘OK’ all the way out and you should now have an Event grid that looks like this;



- This means that throughout all of your levels the group ‘Scroll Control’ will exist, and that it will always centre the display around the spaceman.

- Now press the ‘Play’ icon on the top right of the screen, and you will be able to scroll about the starfield using the cursor keys, with the spaceman staying central in the screen.

Scrolling options


Center window position in playfield
Center horizontal position of window in playfield
Center vertical position of window in playfield

Centre window position in playfield’ will centre the playfield about a specific X, Y coordinate, or that of an Active object. See above tutorial for more information.

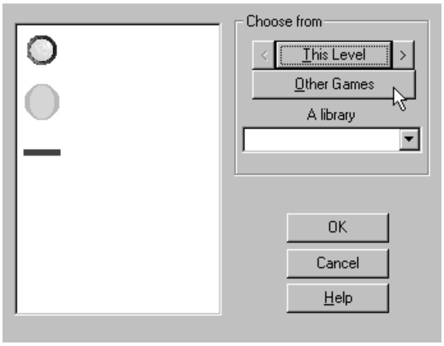
Centre horizontal position of window in playfield’ will centre the horizontal position of the playfield about a specific value. This can be a set number, or, more usefully, by using the Expression Editor, you could retrieve the X coordinate of an object, and centre it around that. When you get to the dialogue box to enter the number, select the ‘Edit’ option, then the ‘Retrieve data from an object’ option, then select the object, then its X coordinate. You would then have a playfield that scrolled horizontal but not vertically, useful for ‘Defender’ type games.

‘Centre vertical position of window in playfield’ does exactly the same as the function described above, except in the vertical instead of the horizontal.

Create New Object Actions

In addition to the information here you can look for the example on the CD ROM.
Look for the directory ‘minituts’, then ‘newobj’, then ‘actions’, then ‘newobja01’.
You should look at the Event Editor of this to see how it was done.

This action allows you to create a new object at a specific point on the screen. This action is very useful for displaying an explosion when two objects collide, for example. There are no options to this function, only the selection of the point where you are going to place the new object, and the selection of the object that you are going to place.
You can choose the object you wish to create from a dialogue box containing all the objects for that particular level, or from a library, or from another game.

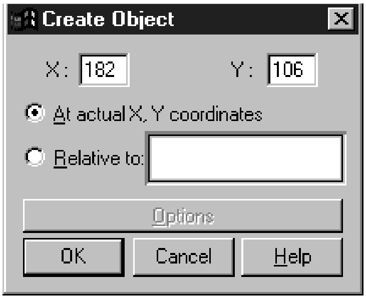


You can either choose from the objects in the dialogue box, or from another frame of your game. You do this by pressing the direction buttons either side of the ‘This frame’ text with the mouse pointer.

‘Other Games’ means that you can grab an object from another game, as long as you have it saved on disk.

‘A library’ lets you look through all the object libraries The Games Factory has to offer, and choose your new object from one of them.

Once you have selected the object you want to create, you select its position using the ‘Select position’ editor.



Create Object

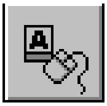
X: 182 Y: 106

● At actual X, Y coordinates

○ Relative to:

From here you can either select a specific X, Y coordinate, or select a position in relation to another object. Be aware of correctly setting the X and Y coordinates when you choose this option. Setting them to zero will create your new object at the same place as the reference object, useful for placing explosion animation's over the top of Active objects.

Mouse Pointer Actions



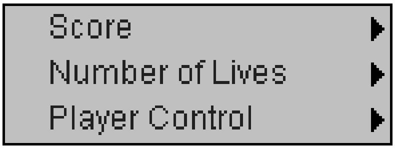
In addition to the information here you can look for the example on the CD ROM.
Look for the directory 'minituts', then 'input', then 'actions', then 'inputa01'.
You should look at the Event Editor of each of these to see how it was done.

Allows you to hide or show the windows mouse pointer.

Player Actions

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘player’, then ‘actions’. There are two examples here, which are identified by their file name. You will be prompted to look up each example where applicable. You should look at the Event Editor of each of these to see how it was done.

This object allows you to change values that are applied to each player, such as lives and score. You have three options;



‘Score’ allows you to alter a players score. You can add, subtract or set the score.
You can do this to a specific value or an expression. See Expression Editor on page 205
Look up on the CD: Minituts\Player\Actions\Playa01

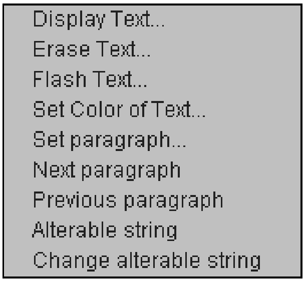
‘Number of lives’ allows you to change the number of lives a player has. Note that the game will not automatically end when the number of lives reaches 0’ you must insert an Event to end the game when the number of lives reaches 0, if that is what you want to happen. The number of lives can be manipulated in the same way that the score can.

‘Player control’ enables or disables a players controls. Disabling them will mean that The Games Factory will ignore any keyboard or joystick commands made by that player until control is restored. This is useful when you want to create a game ‘pause’ mode and want to stop the player moving his hero.

Note that you can define which keys are used for each control in the ‘File-Game setup’ menu from any of the Editor screens. (Situated at the top of each screen) For more information on Game setup see page 93
Look up on the CD: Minituts\Player\Actions\Playa02

Text Object Actions

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘text’, then ‘actions’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.
You should look at the Event Editor of each of these to see how it was done.
This action allows you to change parameters that are applied to your text objects. These are;



‘Display text...’ will display your text object at a specific location. You can either enter in the coordinates directly, use the mouse pointer, or have it in relation to another object, say, just underneath an object, for example.
Look up on the CD: Minituts\Text\Actions\Texta01

‘Erase text...’ simply erases the text box from the screen.
Look up on the CD: Minituts\Text\Actions\Texta02

‘Flash text...’ will flash on and off your text at a specific rate, in a particular place. You will be asked to choose its position, as in ‘Display text...’ above, then asked to set the time delay, which can be a set time or an expression. See the Expression Editor chapter on page 205 for more information.
Look up on the CD: Minituts\Text\Actions\Texta03

‘Set colour of text...’ does exactly that, from a range of 16 colours.
Look up on the CD: Minituts\Text\Actions\Texta04

‘Set paragraph’ allows you to choose to display a paragraph by its number. In this way you can use one text object to display lots of different texts without having to use lots of text objects. To set the many texts use the “Edit Data Strings” option from the level editor (see page 243).
Look up on the CD: Minituts\Text\Actions\Texta05

‘Next paragraph’ displays the next paragraph by numerical order.
Look up on the CD: Minituts\Text\Actions\Texta06

‘Previous paragraph’ displays the numerically previous paragraph. Not necessarily the last one to be shown.
Look up on the CD: Minituts\Text\Actions\Texta07

‘Alterable string’ allows you to display an alterable string in the text box. You can enter an alterable string via an Edit Object, then set text to the value of the alterable string using the Text Objects expression Editor from the ‘Change Alterable string’ to the value of the Text Object contents.
Look up on the CD: Minituts\Text\Actions\Texta08

‘Change alterable string’ sets the alterable string to a value, like the text from an Edit box, or you can enter a set text directly into the Expression Editor. Make sure that the text is entirely within quote marks (“For example”).
Look up on the CD: Minituts\Text\Actions\Texta09

Question and Answer Object Actions

Q/A

In addition to the information here you can look for the example on the CD ROM.
Look for the directory ‘minituts’, then ‘question’, then ‘actions’, then ‘questa01’.
You should look at the Event Editor of this to see how it was done.

These actions allow you to perform actions to your Question and Answer Objects. When you select on the tick-box under the Q / A icon, you can choose to ‘Ask the Question’. This means that if the Q / A object was invisible by default, The Games Factory will display the object and await a response. You then use other events to check the user’s response to the question, see page 152.

Counter Object Actions

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘counter’, then ‘actions’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.
You should look at the Event Editor of each of these to see how it was done.

By using this option you can change your Counter objects. For all three possible actions you can either enter a set number, or an expression. See the chapter Expression Editor on page 205 for more details.

‘Set Counter’ will set your counter to a specific value, or expression.
Look up on the CD: Minituts\Count\Actions\Counta01

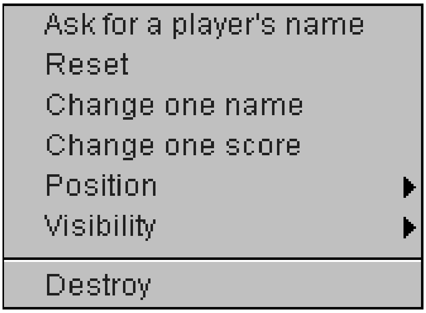
‘Add to counter’ will add to the counter by a set value or expression.
Look up on the CD: Minituts\Count\Actions\Counta02

‘Subtract from counter’ will subtract from the counter by a specific value or expression.
Look up on the CD: Minituts\Count\Actions\Counta03

High Score Object Actions

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘hiscore’, then ‘actions’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.
You should look at the Event Editor of each of these to see how it was done.

Allows you to change your High Score table in a number of ways.



‘Ask for a players name’ will allow you to assign a name to players 1-4.
Look up on the CD: Minituts\Hiscore\Actions\Hiscoa01

‘Reset’ will reset the values in the High Score table.
‘Change one name’ allows you to change a name in the High Score table, identified by it’s numerical rank. You would enter the rank by number, then the name that you want to change it too, remembering to enter the name within quote marks. You will be reminded by a double quote mark.(“”)
Look up on the CD: Minituts\Hiscore\Actions\Hiscoa02
‘Change one score’ allows you to change one of the scores in the High Score table. Again, it is identified by its rank in numerical order.
Look up on the CD: Minituts\Hiscore\Actions\Hiscoa03

‘Position’ allows you select the position of the High Score table by its X and Y coordinates, or separately horizontally and vertically. You can also set its position in relation to another object. See the Expression Editor chapter on page 205 for more information.
Look up on the CD: Minituts\Hiscore\Actions\Hiscoa04

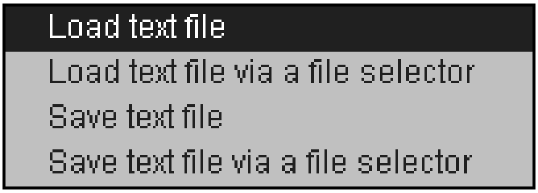
‘Visibility’ allows you to make the High Score table visible or invisible. You may want to make your High Score object invisible at the start of the level, making it visible when your hero has been killed and achieved a High Score, rather than keeping your High Score table on a different level.
Look up on the CD: Minituts\Hiscore\Actions\Hiscoa05

‘Destroy’ does not just make your High Score table invisible, but completely removes it from the play area.
Look up on the CD: Minituts\Hiscore\Actions\Hiscoa06

Edit Objects Actions

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory 'minituts', then 'edit', then 'actions'. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable. You should look at the Event Editor of each of these to see how it was done.

'Files' - From within the Files option, there is a further menu.


Load text file
Load text file via a file selector
Save text file
Save text file via a file selector

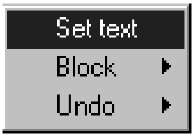
'Load text file' allows you to load a text file from disk.
Look up on the CD: Minituts\Edit\Actions\Edita01

'Load text file via a file selector' will ask the player which text file they want to load in by displaying a file selector on screen.
Look up on the CD: Minituts\Edit\Actions\Edita02

'Save text file' allows you to save text to disk.

'Save text file via a selector' asks the user where they want to save the text file.

'Edition'
By selecting the Edition option you will pull down a further menu.

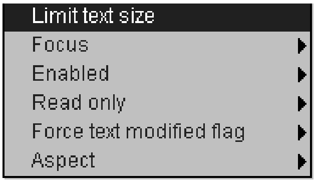


'Set text' allows you to set the text to a specific string variable.
Look up on the CD: Minituts\Edit\Actions\Edita03

‘Block’ allows you to Cut, Copy and Paste text.

‘Undo’ allows you to undo the previous user action within the Edit object.

‘Control’
By selecting the Control option you will pull down a further menu.



‘Limit text size’ allows you to put a limit on the number of characters in the text.

‘Focus’ is an on / off switch for that Edit object, which directs input from the keyboard to that Edit object or not

‘Enabled’ is an on / off switch for enabling / disabling an Edit object.
Look up on the CD: Minituts\Edit\Actions\Edita04

‘Read only’ is an on / off switch that stops you writing into an Edit object.
‘Force text modified flag’ is a switch that forces the ‘Modified’ flag to be on or off irrelevant to user input. This means that for the modified flag to be on the user must have edited the contents of the Edit object at some point. Forcing the modified flag off, will change the way future conditions detect for this flag, and will be fooled into thinking no change has been made.

‘Aspect’ allows you to set a font or background colour via a selector.
Look up on the CD: Minituts\Edit\Actions\Edita05

‘Position’ allows you to select the position of the Edit object.
Look up on the CD: Minituts\Edit\Actions\Edita06

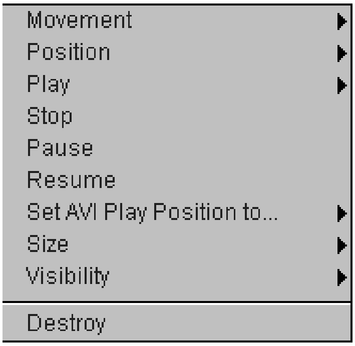
‘Size’ allows you to set the horizontal and / or the vertical size of the Edit object.
Look up on the CD: Minituts\Edit\Actions\Edita07

‘Visibility’ changes the visibility of the Edit object.
Look up on the CD: Minituts\Edit\Actions\Edita08

‘Destroy’ removes the Edit object from the play area.
Look up on the CD: Minituts\Edit\Actions\Edita09

AVI Object Actions

You can change your AVI object using the following options;



‘Movement’, ‘Position’, ‘Visibility’ and ‘Destroy’ are all defined in the Active Object Actions on page 165.

‘Play’ allows you to play your AVI Forwards, backwards, step by step, play a new file, or play a new file via a file selector that will allow the user to select the file.

‘Stop’ stops the AVI from running.

‘Pause’ pauses the AVI.

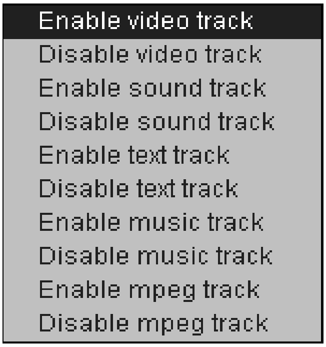
‘Resume’ re-plays the AVI from a paused position.

‘Set AVI Play Position to’ sets the AVI to the beginning, end, to a frame number or time position.

‘Size’ allows you to change the horizontal and / or the vertical size of your AVI object.

QuickTime Objects

All the options for QuickTime objects are the same as for AVI Objects except that you also have the options ‘Volume’, which allows you to set the volume, and ‘Tracks’ which allows you to enable / disable the relevant tracks. The options are listed below;



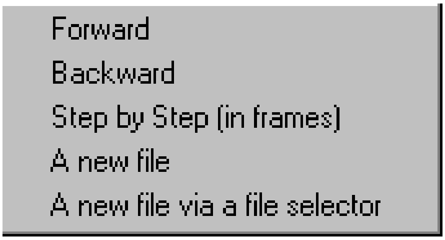
MPEG Object Actions

From these options you can change the movement and position of your MPEG objects, as well as the MPEG frame sequence.

‘Movement’ gives you the same options for movement as Active object. See Active Object Actions, Movement, on page 165.

‘Position’ allows you to set the position of your MPEG on the play area in the same way as for Active Objects. See Active Object Actions, Position, on page 165.

‘Play’ allows you to change the way in which your MPEG is currently playing. Selecting this option will pull down a further menu;



'Forward' plays the sequence of frames in a normal, forwardly sequenced manner.

'Backward' plays the frames in reverse order.

'Step by step' will play one frame at a time, dependant upon that event line.
For example, to have a step-by-step run through the frames, you could insert an Event to test when a key is pressed, and then insert the Action to play the MPEG 'Set by Step (in frames) and enter a value of one into the Expression Editor produced. This will allow the user to advance the sequence by one frame at a time every time that particular key is pressed.

'A new file' will play a new file. You will be asked to select your new file from a new file selector in the Event Editor.

'A new file via a selector' produces a file selector on screen during game time for the user to select the new file from.

'Stop' simply stops the MPEG playing. Note that when 'Frame mode' is off it does not stop the time reference from running. If it were stopped, then resumed, it would start from a new frame in the sequence relevant to how long the sequence had been stopped for.

'Pause' pauses the MPEG sequence as well as the time reference.

'Resume' starts a paused or stopped sequence.

'Set MPEG play position to...' allows you to set the position of the MPEG to either the beginning or end of the sequence. It also allows you to set it to a specific frame number or time reference. The sequence will then be played from that position using whichever play mode you have specified.

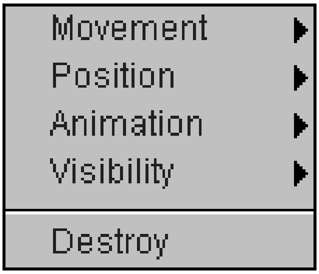
'Size' allows you to change the horizontal and / or vertical size of the display of the MPEG to a specific number of pixels high or wide.

'Visibility' allows you to make the MPEG invisible or visible. It is visible by default.

'Destroy' removes the MPEG completely from the play area.

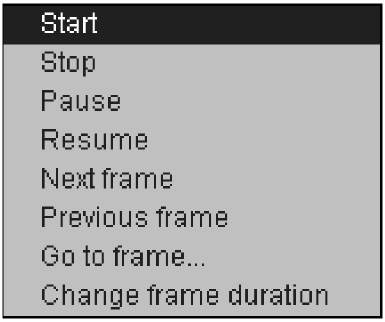
FLI / FLC Object Actions

From these options you can change your FLI / FLC objects;



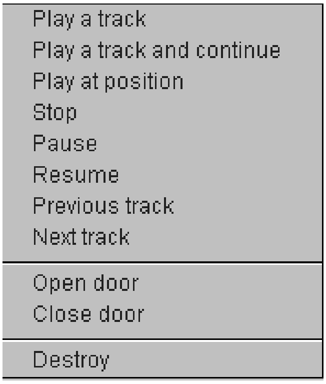
Note that all the options except the 'Animation' are virtually identical to that of Active objects.

'Animation' will pull down a further menu;



CD Audio Object Actions

Using the CD Actions you can change the tracks being played from your CD, etc., etc.
You will be presented with the following menu;



As you can see, the menu options are all fairly self-explanatory. Each option will do exactly what it says to the CD player, except the ‘Destroy’ option, of course, which merely removes the CD object from the play area, not blowing up part of your PC!
You could use these Actions to create your own track selector in a game, or to play some moody music as a background to an Adventure game.

MCI Object Actions

MCI objects give you a general way of controlling a device or animation system from within The Games Factory. You’ll be able to use these devices directly from your games providing you’ve installed an appropriate driver from the Windows Control panel.
For example, one common use of an MCI object is to play samples externally of the Games Factory,

which means large or numerous '.wav' files are not stored in memory all at once. Another could be to play a Midi file from Level 1 and still here it in Frame 7, as all MCI Actions perform external controls of Windows devices.
The available actions are:

'Send a command' this will send a single string to the Media Control Interface using established MCI syntax. All commands must be in lower case and should be in quotes as you enter them into the MCI's Expression Editor.

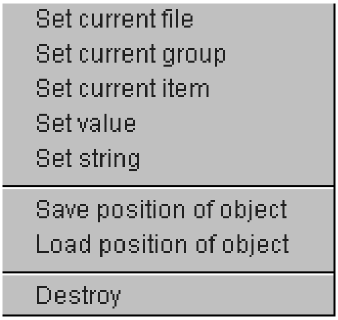
'Destroy' simply removes the object from the play area.

You can enter your commands using the standard the "MCI" instruction set. In order to communicate with the MCI device, you'll need to obey a few simple rules. Each line should be entered into a separate 'Send a command' action from the MCI object's action menu.
You'll also need to define the appropriate MCI objects in the Frame Editor,
and enter the device names into the box provided.
For example, to play a CD, you would specify the device as 'cdaudio' from its 'Setup' menu (Level Editor), and then as an Action you would select the 'Send a command' option and enter the text '"play cdaudio"', making sure that it is within quotes and in lower case only. To change the track to a number, enter the text '"play cdaudio from (X)'", where X is the track number.
As MCI commands are so diverse we cannot feasibly cover them here, and recommend that you consult a Windows manual for more information.

INI Object Actions

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘ini’, then ‘actions’. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable. You should look at the Event Editor of each of these to see how it was done.

It is from here that you can save your game positions and High Scores, load in saved information and decide where you are going to save that information to.
When you select under the INI icon with the right mouse button, you will be presented with a menu like this;



‘Set Current File’ chooses a new INI File to be used for all future operations. If the file doesn’t exist, it will be created automatically on the disk. The default directory will be your current Windows folder.
Look up on the CD: Minituts\Ini\Actions\Inia01

‘Set Current Group’ chooses the Group you wish to manipulate. You should enter the name between the quotation marks. You don’t need to include the square brackets. If the Group doesn’t exist, it will be added to the end of the currently open INI File.
Look up on the CD: Minituts\Ini\Actions\Inia02

‘Set Current Item’ selects the item you want to read or adjust. Enter the name between the brackets. If the item doesn’t exist, it will be added to the end of the currently selected Group.

'Set Value' saves a number in the currently chosen Group and Item.
For example:
'Set Current Group to [X]' will move to the Group [X]

'Set Current Item to "first_value"' will go to the Item called 'first_value'

'Set Item value to 3' will produce "first_value=3"

The text lines would look like this:
(Group)[X]
first_value=3
Look up on the CD: Minituts\Ini\Actions\Inia03

'Set String' saves a line of text into the previously chosen Group and Item.
Before using this feature, you should pick the Group with 'Set Current Group'.
You can now choose an Item you'd like to change with 'Set Current Item'.
Finally, you can enter the text with the 'Set String' action. For example;

Set string value to "High There"

Supposing you wanted to set Item "name" in Group "[User]" to "Linda"
You'd use the actions:

Set Current Group to "user"
Set Current Item to "name"
Set string value to "Linda"

The resulting File would look like this:

[user]
name=Linda
Look up on the CD: Minituts\Ini\Actions\Inia04

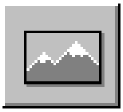
'Save Position of object' stores the current coordinates of an object in the current Group. You'll be asked to choose the object as you are entering the action. You don't need to bother with the "Item name", as it's set up automatically. The Games Factory will generate it by adding "pos." to the object name.
Look up on the CD: Minituts\Ini\Actions\Inia05

'Load Position of object' copies a saved object position from the INI File
and moves the object to the stored coordinates.
You'll be asked to choose an object to be loaded when you enter this action. This object should

have the same name as an object you've previously saved in the "INI" File.

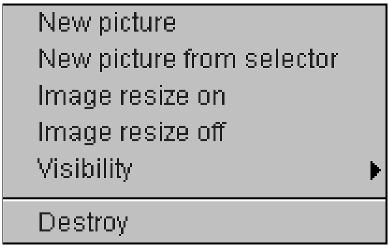
'Destroy' turns off the INI object, and deactivates it from your game.
Look up on the CD: Minituts\Ini\Actions\Inia06

Picture Object Actions



In addition to the information here you can look for the examples on the CD ROM.
Look for the directory 'minituts', then 'picture', then 'actions'. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.
You should look at the Event Editor of each of these to see how it was done.

When you select under the Picture icon with the right mouse button, you will be presented with a menu like this;



'New picture' selects a new picture to be displayed. You select this via a file selector.
Look up on the CD: Minituts\Picture\Actions\Pictua01
'New picture via a selector' allows the game player to select a new picture via a file selector.
Look up on the CD: Minituts\Picture\Actions\Pictua02

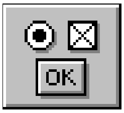
'Image resize on' enables the image to be sized to the size you select in the Level Editor.
Look up on the CD: Minituts\Picture\Actions\Pictua03

'Image resize off' displays the Picture object in its default size.

'Visibility' allows you to make the Picture object either Invisible or visible.
Look up on the CD: Minituts\Picture\Actions\Pictua04

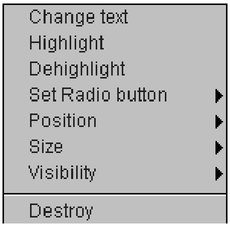
'Destroy' removes the Picture object from the play area.
Look up on the CD: Minituts\Picture\Actions\Pictua05

Button Object Actions



In addition to the information here you can look for the examples on the CD ROM.
Look for the directory 'minituts', then 'button', then 'actions'. There are several examples here, which are identified by their file name. You will be prompted to look up each example where applicable.
You should look at the Event Editor of each of these to see how it was done.

From here you can perform Actions to your Button objects, changing their look, the contents of the text they contain, even their size.
When you right select under the Button object, you will be presented with a menu that looks like this;



'Change text' allows you to change the text that is within a button object. Make sure that all the text you enter is within double-quote marks ("TEXT").
Look up on the CD: Minituts\Button\Actions\Buttoa01

'Highlight' makes the button object highlighted. It is highlighted by default.
Look up on the CD: Minituts\Button\Actions\Buttoa02

'Dehighlight' makes the button object de-highlighted, or 'greyed-out'.

‘Set radio button’ is only available for Radio-Buttons. Within this you can Change Text, Select a Radio Button and Enable / Disable a Radio button.
Look up on the CD: Minituts\Button\Actions\Buttoa03

‘Select position’ allows you to select the position of a button using a ‘Zone setup’ Editor, or to enter in the X and Y coordinates directly.
Look up on the CD: Minituts\Button\Actions\Buttoa04

‘Size’ allows you to set the size of your button object.
Look up on the CD: Minituts\Button\Actions\Buttoa05

‘Visibility’ lets you make the button visible / invisible.

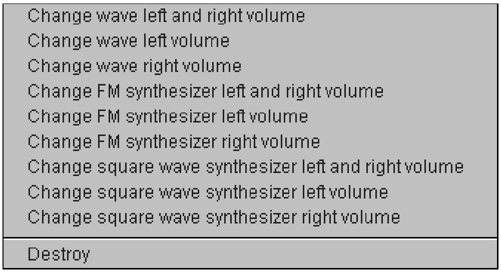
‘Destroy’ removes the button object from the play area.
Look up on the CD: Minituts\Button\Actions\Buttoa06

Mixer Object Actions

In addition to the information here you can look for the examples on the CD ROM.
Look for the directory ‘minituts’, then ‘mixer’, then ‘actions’. There are two examples here, which are identified by their file name. You will be prompted to look up each example where applicable.
You should look at the Event Editor of each of these to see how it was done.

Using the Mixer Actions, you can change the volume output to your speakers, either the volume used when you play samples, when you use the internal FM synthesizer to play ‘Midi’ files, or the Square wave synthesizer.

When you right select under the Mixer Object, you will be presented with a menu that looks like this;



Change wave left and right volume
Change wave left volume
Change wave right volume
Change FM synthesizer left and right volume
Change FM synthesizer left volume
Change FM synthesizer right volume
Change square wave synthesizer left and right volume
Change square wave synthesizer left volume
Change square wave synthesizer right volume
Destroy

The ‘Change wave’ options control the volume that samples are played through your speakers. ‘Left and right’ controls the overall volume, the others their respective volumes.
Look up on the CD: Minituts\Mixer\Actions\Mixera01

The ‘Change FM synthesizer’ option changes the volumes at which Midi files are played.
Look up on the CD: Minituts\Mixer\Actions\Mixera02

The ‘Change square wave synthesizer’ option changes the volumes at which Square wave synthesizer files are played.

‘Destroy’ removes the object from the play area.

The Expression Editor

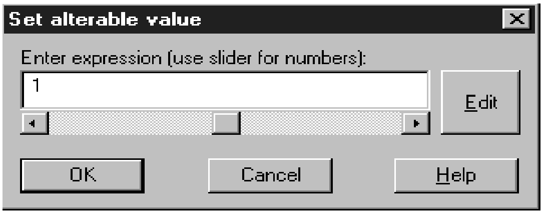
If you have already read through all the Events and Actions in this chapter, you will have noticed a referral to this chapter whenever a numerical value has to be selected.

The Expression Editor is what really gives you the ability to do very complex gaming using very simple Event lines. By using the Expression Editor you can read values such as a players score, internal values, flags, animation sequences, in fact any value that an object possesses.

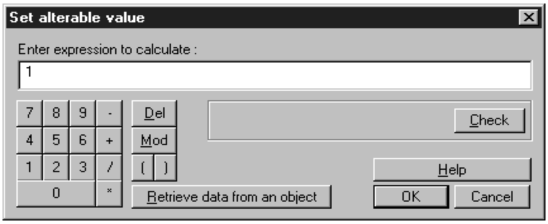
You can, of course, just enter specific values, but it is the ability to retrieve data from objects that really makes the Expression Editor so useful.

Each object has various different types of data that can be retrieved, known here as Functions. The available functions are listed toward the end of this chapter. The descriptions are kept fairly brief as many are self explanatory, e.g., ‘X-coordinate’ or ‘Y-coordinate’.

There are several levels to the Expression Editor, depending on the values that you want to use. If you only want to enter a number, or you know the exact text to enter, you can enter it into the text box of the initial dialogue box, which looks like this;



In this case placing a number in the text box would set one of the alterable values to that number. By pressing the ‘Edit’ button you can go to a further dialogue box which looks like this;



From this dialogue box you can check the syntax of your expression using the ‘Check’ box, or, if you select the ‘Retrieve data from an object’ button, you can retrieve a value from an object.
If you select this you will be able to retrieve virtually any information that you want from an object, its direction, position, coordinates, animation, flags, values, speed, etc.
As mentioned before, if you know the exact text to type in, you can type that into the text box instead of having to select the object, then select the value that you want to retrieve.

Example

For this example we are going to set the X coordinate of an object to be the same as that of another object upon pressing a key.

First of all you must have at least two Active objects places on the play area. You do this from the Level Editor screen.
Once you have done this, go to the Event Editor screen and set up an Event to test when a key is pressed, use the spacebar for this example.

Now you have an Event, you need to assign an Action to that Event.

- Click in the empty tick box with the right mouse button, under the first of your Active objects.
- Select the ‘Position’ option.
- From that menu select ‘Set X coordinate...’
- When you do this you will be taken to the first dialogue box of the Expression Editor.
- Now, you want to retrieve the X value of another Active object, so, select the ‘Edit’ option, which will take you to the second level of the Expression Editor.
- Click on the ‘Retrieve data from an object’ button.
- You will now be asked to select the object that you want to retrieve data from.
- Select the other object with the right mouse button.
- From the menu displayed select the ‘Position’ option.
- From this menu select the ‘X coordinate’ option.
- You should now have an Expression that reads something like ‘X(“Object Z”)’
- This means that the X value of the first object will be set to the X value of Object Z.
- Press all the ‘OK’ buttons to place this Action in the tick box.
- Now when you move the mouse pointer over that Action it should read something like;
- ‘Set X position to X(“Active object Z”)’

You could have retrieved any of the values from object Z and made that the new X coordinate of the first object.

You could, for example, when using the ‘Special Events-Compare two general values’ event, compare the coordinates of two objects, or maybe compare the “Energy” values of two fighters in an adventure game to decide upon a winner. The possibilities are literally endless!

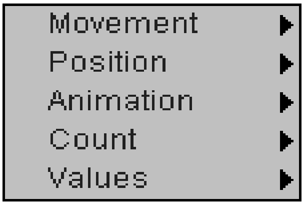
Object Functions

As mentioned above, each object has its own Functions. That is, a set of variables that can be read using the Expression Editor.

Only certain Actions require entering into an Expression Editor, those requiring a value to be entered. The following pages list all the available variables, or Functions, that you can retrieve from an object using the 'Retrieve data from an object' button when using the Expression Editor. See above for how to enter into the Expression Editor.

Active Object Functions

All the variables that can be retrieved from an Active object are listed below. You will be presented with the menu below after you have selected your Active object from the 'Object to test' via the 'Retrieve data form an object' button.



From this menu you can select the general type of information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

Movement

Speed("object")	Speed of the object.
Acc("object")	Acceleration
Dec("object")	Deceleration

Position

X("object")	X coordinate
Y("object")	Y coordinate
X Left("object")	X-Coordinate of left edge
X Right("object")	X-Coordinate of right edge
Y Top("object")	Y-Coordinate of top edge
Y Bottom("object")	Y-Coordinate of bottom edge

Animation

Dir("object")	Current direction value
Image("object")	Current frame

Count

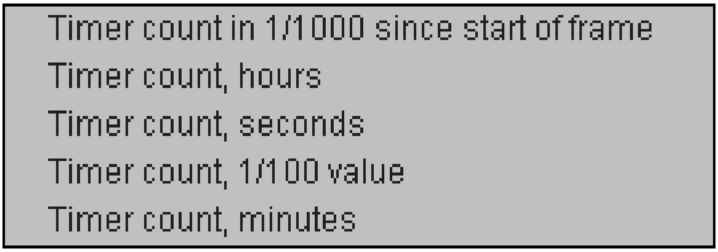
Number("object")	Number of objects on the playfield

Values

Value A("object")	Retrieve Alterable Value A
Value B("object")	Retrieve Alterable Value B
Value C("object")	Retrieve Alterable Value C
Fixed("object")	Retrieve Fixed Value
Flag("object", value)	Retrieve an Internal Flag

Timer Object Functions

This is the primary menu that you will be presented with when you select the Timer Object from the 'Object to test' dialogue box produced from the 'Retrieve data from an object' button of the Expression Editor;

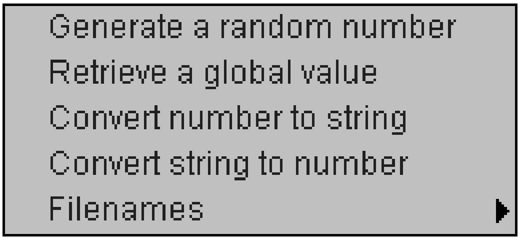


From this menu you can select the information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

timer	Timer Count in 1/1000 since start of frame
hours	Timer count, hours
seconds	Timer count, seconds
hundreds	Timer count, 1/100 value
minutes	Timer count, minutes

Special Object Functions

This is the primary menu that you will be presented with when you select the Special Object from the 'Object to test' dialogue box produced from the 'Retrieve data from an object' button of the Expression Editor;



From this menu you can select the general type of information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

Generate a random number

Random	Generate random number

Retrieve a global value

Global Value(identifier)	Retrieve a Global Value

Convert number to string

Str$(value)	Convert a string to a number

Convert string to number

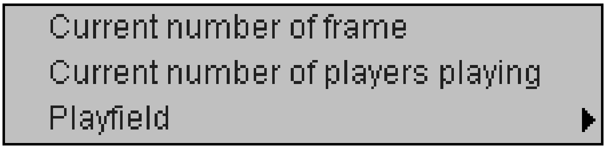
Val("value")	Convert a number to a string

Filenames

appdrive$	Application drive name
appdir$	Application directory
apppath$	Application path name
appname$	Application filename

Storyboard Object Functions

This is the primary menu that you will be presented with when you select the Special Object from the ‘Object to test’ dialogue box produced from the ‘Retrieve data from an object’ button of the Expression Editor;



From this menu you can select the general type of information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

Current number of frame

frame	Current number of frame

Current number of players playing

players	Current number of players playing

Playfield

Width	Width of the play area
Height	Height of the play area
X Left Playfield	X coordinate of left visible edge
X Right Playfield	X coordinate of right visible edge
Y Top Playfield	Y coordinate of bottom visible edge
Y Bottom Playfield	Y coordinate of top visible edge

Create Object Functions

There is only one Function available from this object, and that is the ‘Number of objects on the playfield’. The text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

nobjects	Total number of objects

Mouse and Keyboard Functions

There are only two Functions available from the Mouse and Keyboard object. These are listed below. The text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

Xmouse	Current X position of the mouse
Ymouse	Current Y position of the mouse

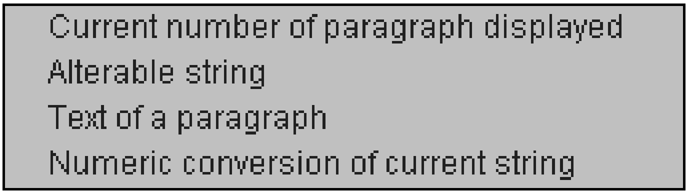
Player Functions

There are only two Functions available from the Player object. These are listed below. The text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

frame	Current number of frame
players	Current number of players playing

Text Object Functions

This is the primary menu that you will be presented with when you select the Text Object from the ‘Object to test’ dialogue box produced from the ‘Retrieve data from an object’ button of the Expression Editor;


Current number of paragraph displayed
Alterable string
Text of a paragraph
Numeric conversion of current string

From this menu you can select the general type of information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

Paragraph(“object”)	Current number of paragraph

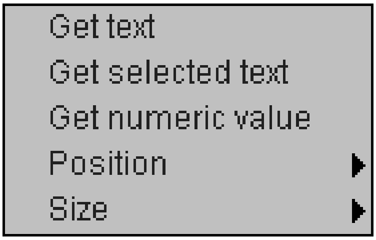
String$("object")	Alterable String
Paragraph$("object",p)	Text of a Paragraph
Value("text string")	Converts string to a numerical value

Counter Object Functions

There is only one Function available from counter objects and that is to retrieve its current value.

Edit Object Functions

This is the primary menu that you will be presented with when you select the Text Object from the 'Object to test' dialogue box produced from the 'Retrieve data from an object' button of the Expression Editor;

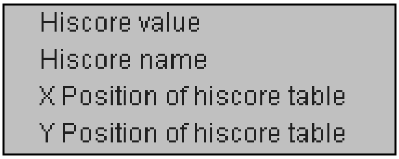


From this menu you can select the general type of information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

Edit Text$("Object")	Get text
Edit Select$("Object")	Get selection
Edit X Pos("Object")	Get X position
Edit Y Pos("Object")	Get Y position
Edit X Size("Object")	Get X size
Edit Y Size("Object")	Get Y size

High Score Object Functions

This is the primary menu that you will be presented with when you select the Hi Score Object from the ‘Object to test’ dialogue box produced from the ‘Retrieve data from an object’ button of the Expression Editor;

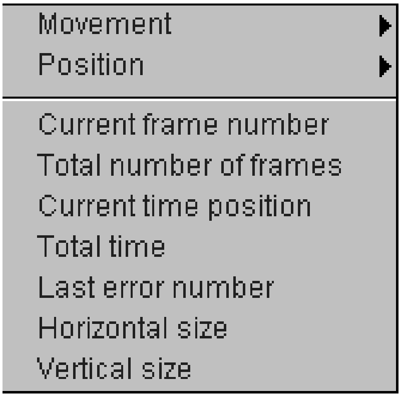


From this menu you can select the general type of information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

High Score(“object”,slot)	High Score value
High Score name$(“object”,slot)	High Score name
High Score X pos(“object”)	X Position of High Score table
High Score Y pos(“object”)	Y Position of High Score table

AVI Object Functions

This is the primary menu that you will be presented with when you select an AVI Object from the 'Object to test' dialogue box produced from the 'Retrieve data from an object' button of the Expression Editor;



From this menu you can select the general type of information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

Movement

Speed("object")	Speed
Acc("object")	Acceleration
Dec("object")	Deceleration

Position

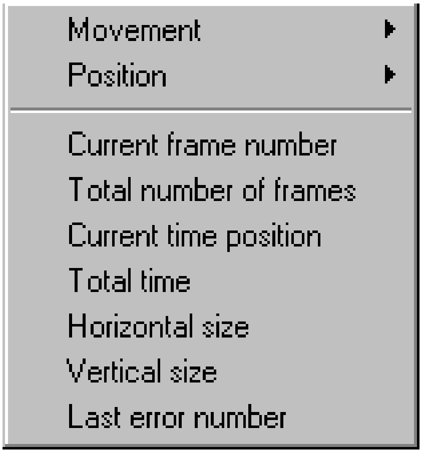
X("object")	X coordinate
Y("object")	Y coordinate
X Left("object")	X-Coordinate of left edge
X Right("object")	X-Coordinate of right edge
Y Top("object")	Y-Coordinate of top edge
Y Bottom("object")	Y-Coordinate of bottom edge

Special Video for Windows functions

Video Frame(“object”)	Current frame number
Video Time(“object”)	Total number of frames
Video Time(“object”)	Current time position
Video Total Time(“object”)	Total time
Last Error(“object”)	Last error number
Video X Size(“object”)	Horizontal size
Video Y Size(“object”)	Vertical size

MPEG Object Functions

This is the primary menu that you will be presented with when you select an AVI Object from the ‘Object to test’ dialogue box produced from the ‘Retrieve data from an object’ button of the Expression Editor;



You can test MPEG clips for a range of information. This information can be used as part of an event to trigger a series of actions in your application. Therefore, you can start a video playing when the video reaches a selected frame, or stop the clip at particular points in the sequence.

From the above menu you can select the general type of information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

Movement

Speed(“object”	Speed
Acc(“object”)	Acceleration
Dec(“object”)	Deceleration

Position

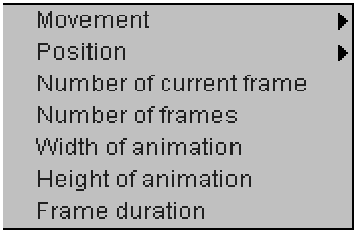
X(“object”)	X coordinate
Y(“object”)	Y coordinate
X Left(“object”)	X-Coordinate of left edge
X Right(“object”)	X-Coordinate of right edge
Y Top(“object”)	Y-Coordinate of top edge
Y Bottom(“object”)	Y-Coordinate of bottom edge

Special MPEG functions

Video Frame(“object”)	Current frame number
Video Total Frames(“object”)	Total number of frames
Video Time(“object”)	Current time position
Video Total Time(“object”)	Total time
Last Error(“object”)	Last error number
Video X Size(“object”)	Horizontal size
Video Y Size(“object”)	Vertical size

FLI / FLC Object Functions

This is the primary menu that you will be presented with when you select an FLI / FLC Object from the ‘Object to test’ dialogue box produced from the ‘Retrieve data from an object’ button of the Expression Editor;



From this menu you can select the general type of information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

Movement

Speed(“object”)	Speed
Acc(“object”)	Acceleration
Dec(“object”)	Deceleration

Position

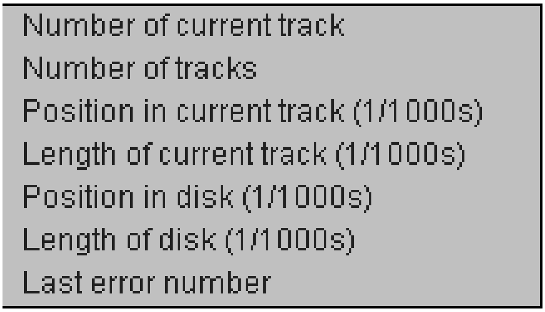
X(“object”)	X coordinate
Y(“object”)	Y coordinate
X Left(“object”)	X coordinate of left edge
X Right(“object”)	X coordinate of right edge
Y Top(“object”)	Y coordinate of top edge
Y Bottom(“object”)	Y coordinate of bottom edge

Special FLI Functions

Current Frame(“object”)	Number of Current Frame
total Frames(“object”)	Number of Frames
width(“object”)	Width of Animation
height(“object”)	Height of Animation
Frame duration(“object”)	Frame duration

CD Audio Object Functions

This is the primary menu that you will be presented with when you select a CD Object from the ‘Object to test’ dialogue box produced from the ‘Retrieve data from an object’ button of the Expression Editor;



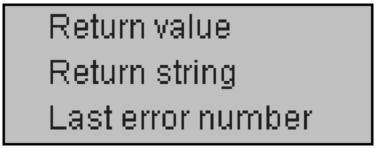
Number of current track
Number of tracks
Position in current track (1/1000s)
Length of current track (1/1000s)
Position in disk (1/1000s)
Length of disk (1/1000s)
Last error number

From this menu you can select the general type of information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

Track(“object”)	Number of current track
Maxtrack(“object”)	Number of tracks
PosInTrack(“object”)	Position in current track
Track Length(“object”)	Length of current track
Position(“object”)	Position in disc (1/1000s)
Length(“object”)	Length of disc(1/1000s)
Last Error(“object”)	Last error number

MCI Object Functions

This is the primary menu that you will be presented with when you select an MCI Object from the ‘Object to test’ dialogue box produced from the ‘Retrieve data from an object’ button of the Expression Editor;



From this menu you can select the general type of information that you want to retrieve. These are listed and described below underneath each menu heading.

Return Value

The Return value function returns the result of the last MCI operation called from your application. The value depends on the type of command you’ve called. In a CD audio object, you could read the total playing time or the number of tracks. In a AVI, you could read the current frame.
The command can be one of the following:

ready
left/right volume output
current track
disk space drive
forward
length
length track number
media present
number of tracks

pause mode
position
position track number
speed
start position
time format
window visible

Return String

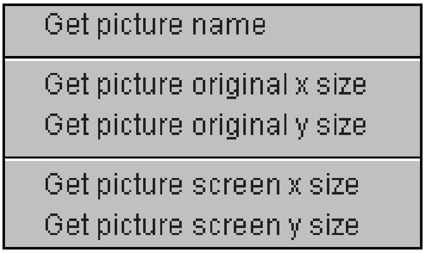
Return String reads a string of characters from the selected MCI device. This string might represent anything from the current filename, to the name of an instrument.

Last error number

Last error number reads the last error number generated by the MCI device. A value of zero means that there are no errors. Any other number indicates a problem.

INI Object Functions

This is the primary menu that you will be presented with when you select an INI Object from the 'Object to test' dialogue box produced from the 'Retrieve data from an object' button of the Expression Editor;


Get picture name
Get picture original x size
Get picture original y size
Get picture screen x size
Get picture screen y size

From this menu you can select the type of information that you want to retrieve. These are listed below underneath their menu headings.

Get Value

Usage: Value of("object")

Reads a value from the current position. You'll need to choose the [Group] and Item first.

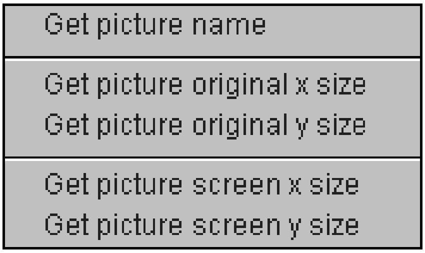
Get string

Usage: String of$("object")

Reads a line of text from the current File position. You can move to the appropriate line with the Set Current Group and Set Current Item actions.

Picture Object Functions

This is the primary menu that you will be presented with when you select Picture Object from the 'Object to test' dialogue box produced from the 'Retrieve data from an object' button of the Expression Editor;


Get picture name
Get picture original x size
Get picture original y size
Get picture screen x size
Get picture screen y size

From this menu you can select the type of information that you want to retrieve. These are listed below, the text on the left is what will actually be written in the Expression Editor (The Games Factory does this for you, although you can enter it in as text if you like) and on the right is a very brief description of the variable.

Get picture name

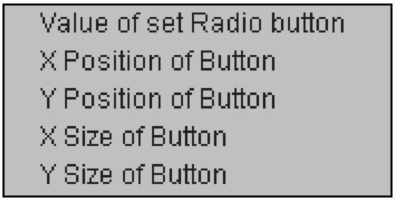
picture name$(object,"string")Get picture name
This function reads the full path name of the currently loaded file. It can now be copied into an alterable string, or saved in an "INI file" using an "INI" object.

Get X, Y variables

x original of(“object”)	Get picture original x size
y original of(“object”)	Get picture original y size
x screen of(“object”)	Get picture screen x size
x screen of(“object”)	Get picture screen y size

Button Object Functions

This is the primary menu that you will be presented with when you select a Button Object from the ‘Object to test’ dialogue box produced from the ‘Retrieve data from an object’ button of the Expression Editor;



‘Value of set Radio button’ retrieves the numerical order of the Radio-button that you have clicked on.
For example, if you had a Radio-button object with 5 buttons, the first button would be 0, the second would be 1 and so on. This function retrieves that number.

‘X position of Button’ retrieves the X coordinate of the Button object.

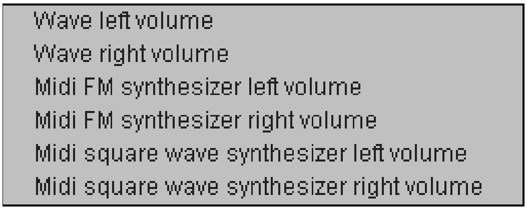
‘Y position of Button’ retrieves the Y coordinate of the Button object

‘X Size of Button’ retrieves the X size of the Button object.

‘Y Size of Button’ retrieves the Y size of the Button object.

Mixer Object Functions

This is the primary menu that you will be presented with when you select a Mixer Object from the 'Object to test' dialogue box produced from the 'Retrieve data from an object' button of the Expression Editor;


Wave left volume
Wave right volume
Midi FM synthesizer left volume
Midi FM synthesizer right volume
Midi square wave synthesizer left volume
Midi square wave synthesizer right volume

'Wave left volume' retrieves the current volume setting that samples are played at through the left speaker channel.

'Wave right volume' retrieves the current volume setting that samples are played at through the right speaker channel.

'Midi FM synthesizer left volume' retrieves the current volume setting that Midi files are played at through the left speaker channel.

'Midi FM synthesizer right volume' retrieves the current volume setting that Midi files are played at through the right speaker channel.

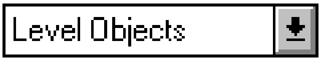
'Midi square wave synthesizer left volume' retrieves the current volume setting that Midi square wave files are played at through the left speaker channel.

'Midi square wave synthesizer right volume' retrieves the current volume setting that Midi square wave files are played at through the right speaker channel.

The Objects

All the Objects that are going to be used in a game must first be placed onto the play area in the Level Editor screen before you can do anything with them. There are several exceptions, however, the ‘Restricted Objects’. These are the Special, Sound, Storyboard, Creation, Mouse / Keyboard and Player 1-4 objects, which are always in existence, and are always on the Event Editor screen and cannot be deleted.

The object libraries contain mainly Backdrop, Quick Backdrop and Active Objects. They are situated on the toolbar at the top of the screen;



By clicking on the pull-down button at the side of the icon you can display / hide a list of all the object libraries. In The Games Factory the list is so extensive that the scroll buttons may have to be used for you to reach the library you require.
Bear in mind that once you have selected and placed an object on your play area it will always be in the ‘Frame Objects’ library, stored at the top of the list. This includes all the objects that you create, such as Counters, Text and Question objects, not just the games characters. Most of these can be created using the ‘Create Object’ icon at the top of the screen.

Summary of types of Objects

Active objects. The main moving characters of the game. Can be manipulated in a variety of ways, moved around and animated. (See page 232)

Backdrop objects. Used to set the scene for games. Can be defined to be an obstacle or not. (See page 240)

Quick Backdrop objects.Can be used to provide more scenery, or as building blocks, platforms etc. (See page 240)

Text objects. Used to put text up on the screen. (See page 243)

Question and Answer objects. For asking the player a question and selecting an answer. (See page 247)

Counter objects. Used for a very wide range of different tasks. Expressed as a number, can be displayed on screen or hidden. Useful for energy levels, status, direction controls etc. (See page 249)

Score Objects. Used for displaying a players score. (See Page 251)

Lives Objects. Used for displaying the number of 'Lives' a player has. (See Page 253)

High Score Object. Displays a High Score table. (See Page 255)

Edit Objects. Can be used for entering player's name, passwords, etc. (See Page 257)

AVI Objects. Used to contain Video clips within your game. (See Page 261)

QuickTime Objects. Very similar to AVI objects, but plays video clips stored in Apple's QuickTime format. (See Page 263)

MPEG Objects. Used for playing MPEG supported video. Note: You must have an MPEG card and drivers to use this object. (See Page 263)

FLI / FLC Objects. Used for playing animated sequences from FLC files. (See Page 266)

CD Audio Objects. Allows you to play audio CD's. (See Page 268)

MCI Objects. One of the main functions is to play various different files in your game. A fairly advanced feature. (See Page 269)

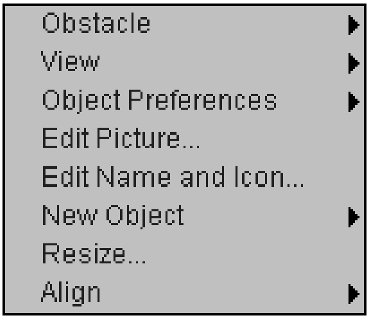
INI Objects. Normally used to provide initialisation information for a game, but can be used to store information like player's position, score etc. (See Page 271)

Picture objects. Can be used to display a picture on the screen. A wide variety of picture formats can be displayed. (See Page 272)

Button objects. Are used to place a 'Button' on the screen which the player can then click on with the mouse pointer, either to 'press' the button, or place a Check-mark. (See Page 274)

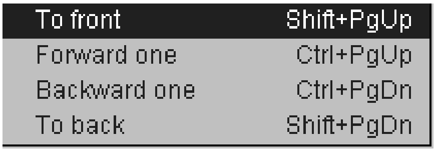
Mixer objects. Used to change the volumes of the audio channels in The Games Factory. (See Page 276)

Common parameters for most objects



Almost all of the objects can be changed in the ways described in the following paragraphs. Some can be changed in all the ways, some only one. The above menu was taken from an Active object, but the following options are open to many of the objects. There is a menu for each object underneath each heading. All of these parameters can be changed from the Level Editor screen when you select your object with the right mouse button. Some can be changed as the game is played, from the Event Editor.

View



Basically, this will allow you to ‘move’ the objects on the Level Editor screen to be displayed ‘in-front’ of, or ‘behind’ one-another. This can be very useful if you are building a complex background, such as a forest scene, where you may want put an object on the screen, behind something else. Normally The Games Factory will put the last object to be placed on the screen ‘in front’ of all the others.

Handy Hint!

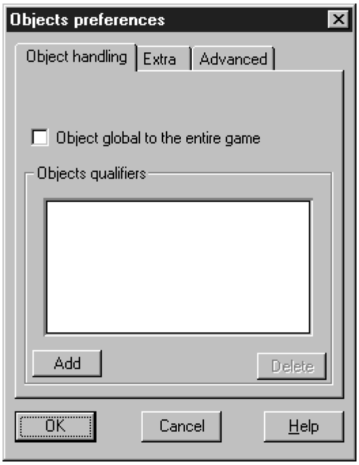
With Active Objects you can access this function from the Event Editor, so that you can put an Active Object ‘in front’ or ‘behind other objects in the middle of a game.
See ‘Event Editor-Active Object Actions-Animation’ on page 168 for more information on how to do this.

Object Preferences

All the different types of object can have various different preferences assigned to them, changing the way that they behave and appear.

All Except Backdrop Objects

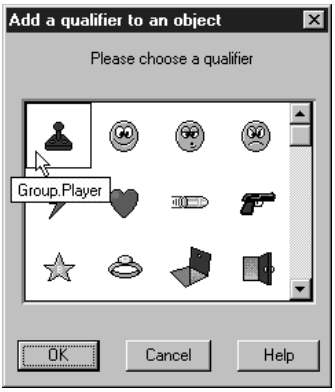
By clicking on your object with the right mouse button and then selecting the ‘Object preferences’ menu item, you will be presented with a screen that looks like this;



□ Object global to the entire game

As you can see, there are three layers to this screen. We will deal with ‘Object handling’ first. Starting from the top of the dialogue box, the first thing that you can see is the tick box for making the object global to the whole application. This means that this object will exist on all the levels of your game and will be stored on the ‘Frame objects’ library of each Level Editor screen. It also means that all the parameters that you assign to that object will be carried through to each level, such as its movement.

Handy Hint!
Below this is the heading for the ‘Object qualifiers’.
This is a very useful function indeed and can save you hours of time in the Event Editor.
What this does is it assigns an object to one or more groups. For example, you can have several different objects and call them all ‘Bullets’ and another group of objects called ‘Goodies’. Then, in the Event Editor, instead of having a separate event for each individual bullet and goody, you could just say “If the Group ‘Bullets’ collides with the group ‘Goodies’, do.....”
When you go into the object qualifiers you will be presented with a screen like this;

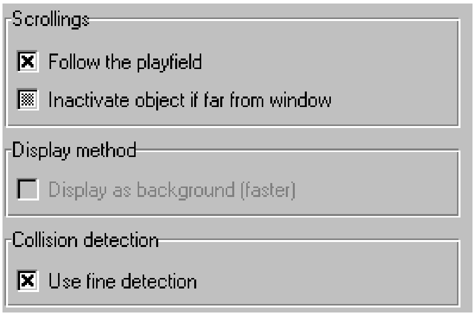


Add a qualifier to an object

Please choose a qualifier

This dialogue box displays the many different groups that there are available for you to put that particular object into. An object can belong to more than one group.
For example, you can have a group ‘Bullets’, and have Goodies Bullets and Baddies Bullets. One particular bullet would belong both to the group ‘Bullets’ and to either the ‘Goodies’ group or the ‘Baddies’ group also.
There is a scroll bar at the side of the window as there are so many different groups for you to choose from.
Once you have chosen your group it will be displayed on the first ‘Object Qualifiers’ screen.

Now let’s move on to the ‘Extra’ functions.



Scrollings

☒ Follow the playfield

▨ Inactivate object if far from window

Display method

□ Display as background (faster)

Collision detection

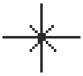
☒ Use fine detection

Under the ‘Scrollings’ heading we have the ‘Follow the playfield’ tick box. This means that unless otherwise changed, the object will be scrolled when the playfield is scrolled. Having this turned off means the object will stay still even when the playfield is moved around, handy for things like score and life displays, which you want to stay in the same position on the screen all the time.

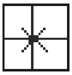
The next box down will inactivate an object if it strays too far from the game playing area. This can be useful if you are going to have a lot of objects flying around on and off the screen. It can speed up your games by deactivating all objects that are not going to come back onto the play area.

The display method, when left as default (off), will display your object as normal, but changing to background will merge it with the current background, making it faster to be displayed. You cannot do this to Active objects.

Collision detection will adjust between fine and box (only shown as on or off respectively). Using box detection uses the box around the Active object to determine whether or not there has been a collision. By using fine, The Games Factory will look at the edge of the 'Solid' area of an object and use that to determine whether or not a collision has been made.
The example below demonstrates the difference.

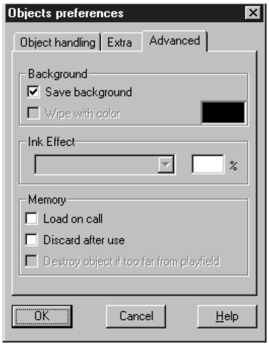


In this picture, a collision would only be detected if another object collided with the solid parts of the object, the actual structure of the star.



In this picture, a collision would be detected if there was a collision with the imaginary box that surrounds the image, which in some cases would look unrealistic.

Advanced Object Preferences



Background
☑ Save background
□ Wipe with color

Ink Effect
%

Memory
□ Load on call
□ Discard after use
□ Destroy object if too far from playfield

The 'Background' options allow you to save the background as an Active object is moved over it, or to wipe it with colour. The reason that you may want to do this is that in very fast 'Shoot `em ups' with many different Active objects whizzing around the screen, not having to replace the background as an object is moved over it can dramatically speed up a game.

When the tick box in the ‘Save Background’ switch is filled, The Games Factory will re-display the background as an object is moved over it. With the tick box off, it will either leave a trail of parts from the Active object, or...
‘Wipe with colour’ wipes the trail of an Active object with a specific colour. In space shoot `em ups this could be black. This means that a black trail would be left by an Active object as it moved around, but if the background was black anyway it would be invisible. Or you could use this function to leave a ‘Snail trail’ around the screen of the colour that you are wiping with.

‘Ink Effect’ gives you a wide range of effects to play with. Using this function you can make ghostly figures barely visible, ones that only appear when they pass over a background object, plus some that will result in some fairly psychedelic colour changes.
By pressing the pull-down button at the side of the text box you select your ink effect. Note that the % box will only be activated when you select the ‘Semitransparent’ option.

‘Transparent colour’ draws the object as normal, and draws the background of that object with a normal transparent colour. The default setting.

‘Opaque’ means that the areas normally transparent on the object will be displayed as an opaque colour, normally black. This is slightly faster for The Games Factory to do.

‘Semitransparent’ draws your figure with ‘translucent ink’. You can change the density of the ink by changing the percentage transparency. The higher the number the more transparent will be your image. You can create an excellent ‘Ghost’ effect using this function. But beware, this effect takes the processor a long time to produce and can slow down your game.

‘Inverted’ inverts the colours that your object is drawn with. Note that it does this by colour as well as shade. You can end up with some unexpected results, so exercise some caution with this and the following ink effects.

‘Xor’ will invert your objects colours, but re-invert them over a black background. Can result in very interesting colours when passed over the top of other colours.

‘And’ results in your object being invisible over a black background and unchanged over a white one. Again, results in some fairly fantastic colour changes when it interacts with other colours.

‘Or’ results in your object only being normally visible over a black background, and completely invisible over a white one. Again, interacts psychedelically with other colours.

‘Memory’
You can change the way The Games Factory stores the information about objects in various ways

by changing these tick-boxes. You may find that if you do a very large game indeed, an awful lot of memory is used, which may not be desirable.
So, The Games Factory will let you load in information only as and when it is needed. Normally, at the start of a game all the information about all the characters will be loaded in; all the animation frames, preferences, etc.
You can change this using the following functions.

'Load on call' will only load in animation frames to memory when it is required in the game. Before an animated character is displayed on screen, the animation frames associated with that character will not be stored.
Note: This can slow down action games noticeably as The Games Factory will have to pause the game for a fraction of a second as it loads in the animation frames. This function is most useful in adventure games where memory is at a premium and such momentary pauses would go unnoticed. Note also that those frames will stay in memory once loaded unless 'Discarded after use'.

'Discarded after use' discards all information not needed about a character once that character is removed from the screen. Normally used in conjunction with 'Load on call' to save memory. Also minutely slows down game play, so best used in slow paced games which have lots of animated characters.

'Destroy object if too far from playfield' removes an object from the display list. When used in conjunction with 'Discard after use' and 'Load on call' will remove the animation frames of that character from memory. Note that this function is normally on by default, so if a character wanders off screen too far, The Games Factory does not have to keep on displaying that character. This is useful if you have many animated objects flying around, when it would become difficult to display all the objects both on and off screen.
By turning this option off your objects will be permanently somewhere off the screen until you decide to destroy them via the Event Editor.

Edit name and icon

Normally, The Games Factory has a name and icon already for each object in the object libraries. Also, if you were to create your own object, The Games Factory would make an icon for that object using a small version of the graphic for that object, and would assign names in numerical order, e.g. 'Active Object 1', or 'Counter Object 15'
When you start getting complex game play you may well find it easier to keep track of all your different characters if you name them yourself as well as making up your own icons for them.
For example, the icons for counter objects are all the same, yet you may be using many counters in your game. Changing the appearance of the icon will help you greatly to keep track of them both on the Level Editor screen and particularly on the Event Editor screen. Assigning your own

names to them will help you even more.
You may well have two different Active objects that use exactly the same animation frames, and whose icons are identical, yet they perform two entirely different functions. It would be far easier to keep track of them with different icons and names.

For more detail on the actual mechanics of changing the icon, go to the ‘Picture Editor’ chapter on page 287.

Resize

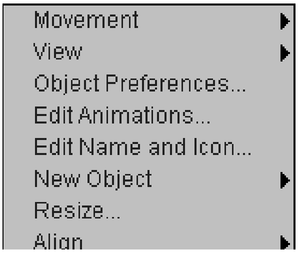
Allows you to resize your object by using pick-points on a box drawn around the object. By selecting and holding a pick-point, you can drag the box to the size that you want the object. With Active objects you also have the option to enter the size change as a percentage ratio directly.

Align

With this function you can precisely align your Object. You can push it up against the left hand or right hand edge of the playfield, or centre it horizontally from the ‘Horizontally’ option.
If multiple objects are selected you can align objects in rows and columns.

Active Objects

Note: Only the options related specifically to Active objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225.



Active Objects are used as the main characters of your games. These are the characters that you can control, either using the mouse or keyboard, or having the computer control them. You can also make your active objects animated, making them run, jump or whatever you decide.

Active objects are denoted by the icon on the object shelf (Level Editor Screen). Try grabbing one and dragging it onto the screen.
You can also create your own by clicking on the new object icon on the tool bar at the top of the screen (Level Editor).
Once you have done this you can change the options by selecting on the object with the right mouse button.

Movement

To change an objects movement from the Level Editor, select the object using the right mouse button, then select the ‘Movement’ option from its pull-down menu and then ‘Select Movement’.

Player Controlled Movement

Mouse Controlled

This assigns your object to exactly follow the movement of the mouse. Notice that when you call this option your object will be surrounded by a box which represents the limits of movement of your object. You can make this stretch or shrink by grabbing the pick points with your mouse and dragging them around. Note this box takes its position from the object, not the screen, so if you move your object to a new position on the Level Editor screen, you may well need to edit this box again.

The following movement parameters are used for several of the movements, so they have been summarised here.

‘Speed’ Sets the maximum speed your object will be able to move at.

‘Acceleration’ Sets the rate at which your object will reach its maximum speed. If your object was a car, for example, you may want the acceleration quite high. If it were a “heavier” object, like an elephant, you would want it set quite low.

‘Deceleration’ Affects the rate at which your object slows down. Having a high value will stop your object quickly when you release the key to move it. Having it set low will result in a very gradual slowing down, as though the object were really heavy, like an oil tanker.

‘Initial Direction’ Allows you to decide what the first direction will be of the object. If you select more than one direction, the computer will select one direction at random.

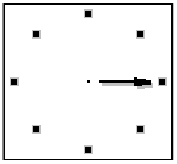
‘Moving at start’ Toggles whether or not you want your object to be moving at the start of a game.

‘Try Movement’ Tests your movement on the screen. Use the <Esc> key, or select the “Stop” icon to take you back to the direction editor.

Eight Direction Movements

This movement provides you with the classic eight different directions that are used by a joystick. You can also use the cursor keys to control this movement. There are several basic controls. Speed, acceleration and deceleration have been described above.

‘Possible directions’ allows you to select or un-select the number of directions your object can move in.



To select or un-select a direction simply click on the relevant box. Possible directions are shown by having an arrow pointing to that box. In the example above the only possible direction that object could move in would be right.

Race Car

This simulates a birds-eye view of a cars' movement, thus there are controls for steering, braking and accelerating. These are;

Accelerator	Up arrow	Joystick Up
Brake	Down arrow	Joystick Down
Turn Left	Left Arrow	Joystick Left
Turn Right	Right Arrow	Joystick Right

In addition to speed, acceleration and deceleration settings, there are;

'Enable Reverse movement' Gives your object reverse gears. With it turned off, you can only have forward movement.

'Directions' Lets you decide how many different directions it is possible for the vehicle to move in. Selecting 4 will only give you left, right, up and down, 32 the smoothest possible direction changes.

Note: You can easily create all the different animations needed for each direction by using the Animation Editor. (See Page 241)

'Rotating Speed' Sets the rate of turn. Having a high value will allow tight corners to be turned, having it low reduces the cornering ability.

Platform Movement

This movement is used mainly to define platform game-type movement, i.e. characters who walk along a platform on the screen viewed from the side.
They are controlled by the cursor keys or the joystick.
In addition to the usual Acceleration, Deceleration and Speed, there are a large number of controls peculiar to Platform movement.
See 'Backdrop objects-Obstacles' on page 241 to see how to go about making platforms and ladders out of backdrop objects.

Note: You must still test for a collision with a backdrop platform object otherwise your active object will fall through the platform.

'Gravity' Like it suggests, selects the effect of gravity. A high setting will have your object fall rapidly, allowing only short jumps.

'Jump Strength' Selects the jumping power of your character. Changing the gravity will also affect this parameter.

‘Jump Controls’ Used to change the control system for jumps.

‘No Jump’ Turns the jumping off.

‘Up Left/Right arrow’ Makes the object jump when both the ‘Up’ arrow and either the left or right cursor keys are pressed.

‘Button 1’ Uses fire button one or the “Shift” key to control the jump.

‘Button 2’ Uses the second fire button or the “Control” key to activate a jump.

Computer Controlled Movements

These movements are used to move around the other objects in your game, from a simple bouncing ball to an attack wave of aliens.

‘Bouncing Ball’ Is normally used to produce an object that will bounce around the screen, mimicking the movement of a bouncing ball. However, by changing a lot of the parameters and using the Event Editor, this movement can be used to control the movement of a host of Aliens, or platform Baddies that will chase you around.

‘Speed’ Speed is set as for all the other types of movement.

‘Ball Deceleration’ When this is set to zero, the ball will keep on bouncing around forever. Increasing this value gradually slows your object down until it eventually grinds to a halt.

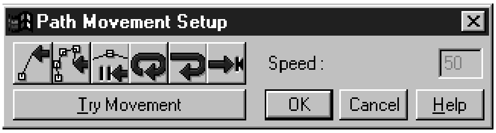
‘Bounce Randomiser’ As the name suggests, makes the bounces more random in their direction by having this control set high.

‘Bounce Security’ This jiggles your objects around to stop them getting stuck in corners, but as a result the rebound effects are made slightly more random.

Path Movement

Sets your object moving on a pre-determined path that you define. For example, a patrolling guard who walks a set distance then turns around, or has a pre-set walk around a corridor.

Path Editor



Path Movement Setup

Speed : 50

You can see that there are six different buttons for you to define your movement, plus the speed bar which changes the speed which that object moves along it’s path.
A Path-type movement is entered using your mouse to define the path.

‘New Line’ This function will add a single line to your movement. Note: If you already have a movement defined, New Line will be added at the end of this by default, unless you insert it by choosing the insertion point with the mouse.

‘Tape Mouse’ Allows you to set a very complex path movement. By holding down the left mouse button and dragging it around the screen you set the movement you want. Note that this function will change the speed of the object depending on how fast you move the mouse around.

‘Pause’ Stops your object at its current position for a pause that you define.

‘Loop the Movement’ This will run whatever movement you have defined over and over. Note that it will re-position the object to its original starting position to continue the loop, so try and ensure that your path finishes at the objects start point, or it will ìjumpî around the screen.

‘Reverse at End’ Simply reverses an objects movement and sends it back along the original path, backwards.

‘Reposition Object at End’ Replaces your object to its original starting position when it has completed the movement.

‘Try Movement’ Allows you to try the movement before finally deciding upon it.

Editing a path

Once you have added a movement to your object, you can edit it very easily afterwards from the Level Editor by selecting the object, then select the ìEdit Movementî option.

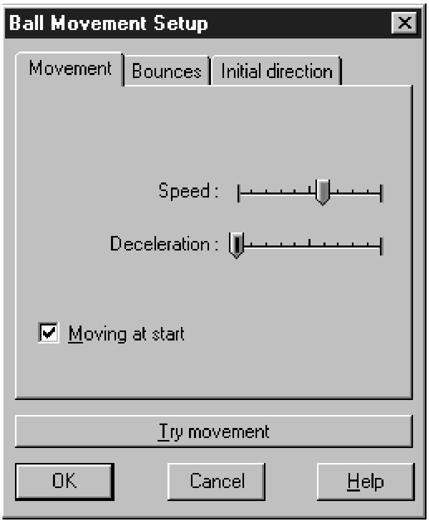
This will present you with the Path Editor again. You can select individual points of the movement, or whole sections by dragging a box around them. You can then manipulate these selected pieces by either deleting them, or using the Cut, Copy & Paste from the Edit menu. (Drop-down menu at the top of the screen).

You can simply drag one of the points that you select using the left mouse button.

Inserting a condition

By using the right mouse button on one of the points, you can insert one of the previously defined conditions at any point along a path e.g. set a pause, tape mouse, new line etc.

Bouncing Ball Movement



The Bouncing Ball movement is normally used to provide a ball-like movement that will bounce off other obstacles on the screen.

However, by setting all the parameters to zero, you will have a movement 'blank page' which you can then manipulate entirely from the Event Editor, useful for Baddies movements.

When you first enter the 'Ball movement setup', you will be presented with a three layered dialogue box. You can change the layer being shown by clicking on its text.

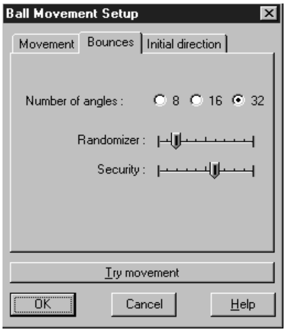
Movement

'Speed' sets the maximum speed which your object can move at.

'Deceleration' sets the rate at which your bouncing object will slow down. Having this option set to zero will bounce your object around endlessly.

'Moving at start' will set off the movement in one of the directions you choose (see below) right from the start of the game. Having it set off means that the object will remain stationary until it is bumped into by another object.

Bounces



Number of angles : ○ 8 ○ 16 ● 32

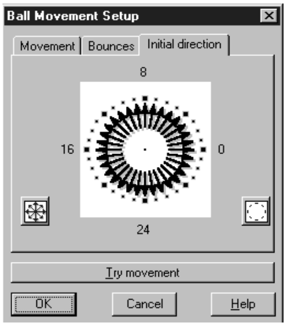
‘Number of angles’ sets the number of angles that it is possible for you object to bounce in. Having it set to 32 will result in the smoothest, most realistic effect.
Having it set to 8 will result in an object that can only move left, right, up, down and diagonally in-between and would not suit an object that is supposed to bounce like a ball.

‘Randomiser’ gives your object a chance off bouncing of in a different direction to that you would normally expect. The higher the setting the more unpredictable your bounces will be.

‘Security’ jiggles your objects around to stop them getting stuck in corners. As a result, the rebound effects do become slightly more random.

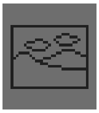
Initial Direction

Allows you to choose one or more directions for your object to move in at the start of the game. Having more than one direction selected will result in a random choice being made between the directions that you have selected.



You can select and un-select the directions individually by clicking on the box at the end of each direction arrow. A selected direction will show an arrow to that box, un-selecting a direction will remove the arrow. You can select all the possible directions using the icon on the bottom left, or un-select all directions by using the icon on the bottom right.

Backdrop Objects and Quick Backdrop Objects



Note: Only the options related specifically to Backdrop objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225.
All the following parameters apply to both Backdrop and Quick Backdrop objects, except that Quick Backdrop objects can be constructed without using the Picture Editor, if their appearance is going to be relatively simple. How to do this is discussed at the end of the this topic on page 242.

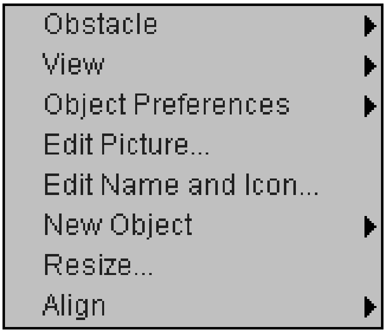
Backdrop objects are normally used to ‘set the scene’ in games. Provides a backdrop for your players to move over, or even to interact with.

You cannot move backdrop objects or change their appearance when the game is playing, as you can with Active objects, but you can change their position on the screen, their size, shape and colour from the Level Editor.

You can either select one of the many Backdrop objects from an object library or create you own

Backdrop and Quick Backdrop objects using the icon from the toolbar at the top of the Level Editor screen.

Most of the changes that you make to the actual appearance of the backdrop object are done using the Picture Editor (see page 287). You only use this when you have selected ‘Mosaic’ when using a Quick Backdrop object.
All the others are done from the main menu that you pull down when you select the backdrop object from the Level Editor screen.
You will be presented with a menu like this;



‘Obstacle’ You can change the way that other objects on the screen will interact with a backdrop by turning on or off the menu options here. There are four options;

‘No’ means that the backdrop object will not be an obstacle to Active objects. You will not be able to detect a collision with an Active object with this option turned on.

‘Yes’ means that it will be possible to detect a collision with an Active object. You must test for a collision with a backdrop in the Event Editor and insert a ‘Stop’ action

‘Platform’ means that the backdrop object will act as a platform for Active objets controlled by platform-type movement. Not the same as ‘Obstacle’ because you will not be able to detect a collision from an Active object that has been assigned a platform-type movement.

‘Ladder’ will treat the backdrop object as a ladder when you are using a platform-type movement. For those animated objects that have a relevant animation sequence, the animation will automatically be changed when they are climbing a ladder. If not, you could change the animation via the Event Editor. See ‘Active-Actions-Animation’ on page 167

‘Edit Picture’ allows you to change the appearance of the backdrop object using the Picture Editor. To learn more about the Picture Editor turn to page 287

‘New Object’ is available for both Backdrop Objects as well as Active objects, and means that you can produce another object that looks exactly the same as the original object, only it exist as either a Backdrop, Quick Backdrop or an Active object. You can then place this object on the Level Editor screen and although it looks exactly the same as the original Backdrop, it takes on all

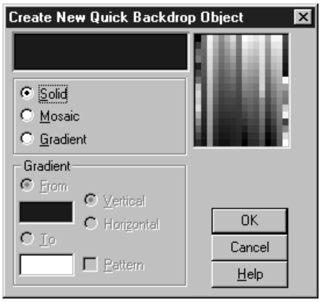
the qualities of whichever object type that you selected it to be.
'New Quick Backdrop Object' is only available from the menu of a Quick Backdrop object. It produces a clone of the original, but with a different name, one up in numerical order. For example, if you created a Quick Backdrop object and it was the first one that you had produced in that game, The Games Factory would call it 'Quick Backdrop 1'. If you cloned that object using this method, even though it would take on all the same parameters as the original object, it would be called 'Quick Backdrop 2'.
Note that this differs from using the right mouse button to lay down multiple copies of the same object. Using that method results in producing genuine clones with the same name. If you made a change to one, that change would occur to all of them.

'Resize' allows you to resize your object by using the pick-points on the box shown around the object.

'Align' will correctly align your Backdrop object. You can push it up against the left hand or right hand edge of the playfield, or centre it horizontally from the 'Horizontally' option.
You can align it against the top, bottom or centre it vertically using the 'Vertically' option from the menu.

Special Options for Quick Backdrop Objects

The reason that there are two different types of backdrop object is that for many purposes it is only necessary to produce a block of colour, or gradient of colour.
For example, to produce a plain black backdrop, it is very easy to simply select a block colour for a quick backdrop and then stretch it to fit the screen rather than go into the picture Editor and fill a picture. For simple structures like platforms and ladders they are very quick to be produced.



● Solid
○ Mosaic
○ Gradient
● From
○ To
● Vertical
○ Horizontal
□ Pattern

You have several options within the Quick Backdrop Editor.

‘Solid’ will fill your object with a solid colour, selected from the palette.
‘Mosaic’ will take you to the Picture Editor. See page 287 for more information.

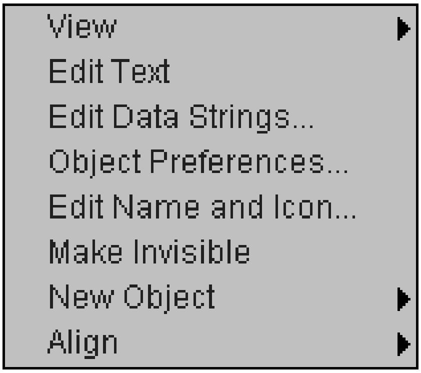
‘Gradient’ will produce a smooth gradient from one colour to another. To use this function, tick in the ‘Gradient’ tick-box with the left mouse button, then the ‘From’ button. Now choose your first colour from the palette. Then select the ‘To’ button, then select the second colour from the palette. You will be presented with a smooth graduation of colour from one to another. You can change the orientation of the gradation using the ‘Vertical’ and ‘Horizontal’ buttons.

‘Pattern’ uses a cross-hatch to grade the colours from one to another rather than smoothly fading the colour.

Text Objects

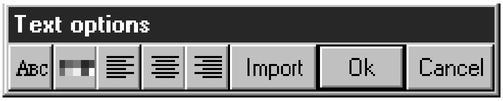
ABC

Note: Only the options related specifically to Text objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225.



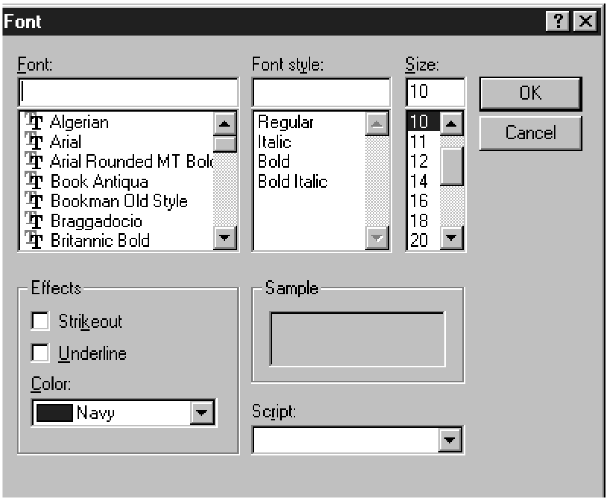
Text objects are used to put text up on the screen. You can use them for instructions, comments, end of game displays or just about anything where you need to place text on the screen. There are some texts already in the object libraries, but you really want to be making your own. To do this select the icon from the toolbar at the top of the Level Editor screen, then click on the text icon. (Shown above.)
The mouse pointer will be replaced with a cursor. You place this cursor at the place that you want the text to start.

You will also have a dialogue box which is where you select your fonts, size and colour of text, as well as the justification style you want to use.



Selecting a font, size and colour

Clicking on the font icon will take you to a dialogue box where you can choose the size, style and font that you want to use for the text in your text object.



Size: 10

□ Strikeout

□ Underline

Color: Navy

To change the parameters simply click on each text box. All the text boxes have scroll bars so that you can view all the options available. The ‘Sample’ box will give you a preview of the style you have chosen.

‘The colour selector’ This is merely a shortcut to changing the colour of the text via the ‘Font

Editor’, and brings up a palette for you to choose the text colour from.

‘Left Alignment’ When you have more than one line of text, this option will align it all so that all the lines start evenly from the left-most margin.

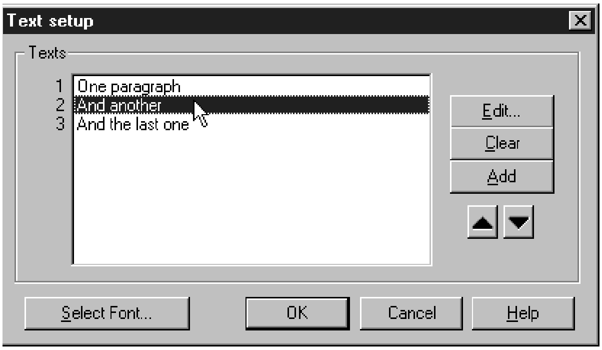
‘Centre Align’ This will align all the text centrally, so the length of the lines will be mirrored about a central line.

‘Right Align’ Aligns all your text to butt-up against the right margin.

Import

‘Import’ This allows you to load up .txt files and place them into your text object from disk.

Once you have places your text object onto the play area, you will be able to right click on it to produce it’s menu. All the other options have already been discussed, the ‘Setup’ option was described above. The following page describes the function of the ‘Edit Data Strings...’ option. This option allows you to have one Text object used to display several different paragraphs of text. When you select this option you will be presented with a dialogue box like this;



Text setup

Texts

1 One paragraph
2 And another
3 And the last one

This screen shot was taken with three paragraphs entered, ‘One paragraph’, ‘And another’, ‘And the last one’.

You can ‘Edit’ or ‘Clear’ the selected paragraph, or change its order using the up and down arrows on the right of the dialogue box.

You add more paragraphs using the ‘Add’ button, which then takes you to a text box where you enter in the text for your new paragraph.

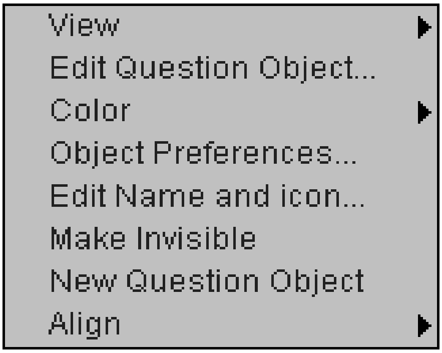
Note that on the left of each paragraph is number. This number is used to identify the paragraph when you display it in the Event Editor.

For example, an Action you could insert under the Text object on the Event grid would be ‘Display paragraph 1’, which in this case would display ‘One paragraph’ on the play area during a game.

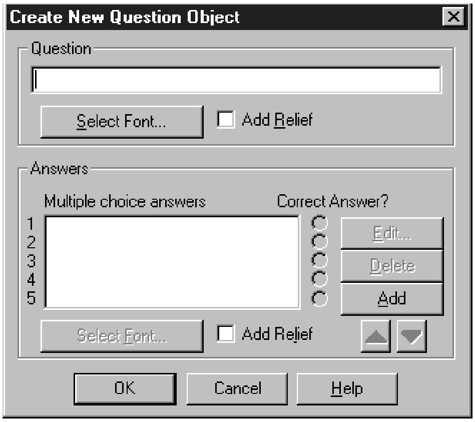
Question and Answer Objects

Q/A

Note: Only the options related specifically to Question and Answer objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225.



These are used to ask the player a question and retrieve an answer. The finished object will be in the form of a dialogue box and the answers are buttons that you can click upon with the mouse pointer.



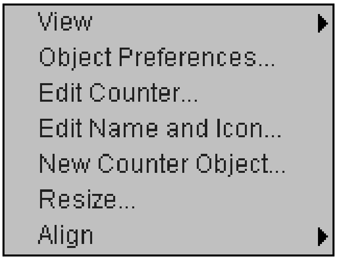
At the top of the Question Editor is a text box for you to enter the text of the question. You can

select the font of the text used here by clicking the ‘Select Font...’ button.
You can also have the question text ‘raised’ by clicking in the ‘Add Relief’ box.

Underneath this is a text box where your answers are displayed. To put in the text of an answer, click the ‘Add’ button. This takes you to another text box where you can type in the text of your answer. You can keep on adding answers until you are satisfied that you have enough answers to choose from.
To choose which answer is the correct one, select in one of the ‘Correct Answer?’ tick boxes. This can then be read from the Event Editor.

Counter Objects

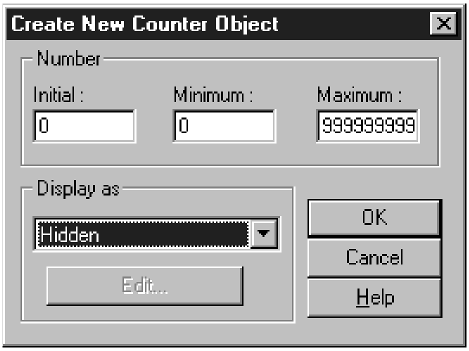
Note: Only the options related specifically to Counter objects are described below. All the other options available are described in the heading 'Common parameters for most objects' on page 225.



Counter objects can be placed on the screen using the icon on the toolbar at the top of the Level Editor screen, then selecting the Counter Object icon.
They can be hidden, to count special values to help decide direction, timers, etc., or shown, to represent power levels, speed, temperature, etc.

To stop a displayed counter being moved around the screen with the play area, go to the 'Object preferences-Scrollings-Follow the playfield' option, and turn it off.

Select the counter with the right mouse button after you have placed it on the play area to pull down a menu, now select "Edit Counter...". You will be presented with a dialogue box like this;


Create New Counter Object

Number

Initial : 0
Minimum : 0
Maximum : 999999999

Display as: Hidden

Here you can select the initial value of the counter, the minimum that it can ever reach, and the maximum value it can reach. You can change these by selecting in the relevant text boxes.
By default the counters are ‘Hidden’ which means that they are not visible when you run your game.
To show the other options use the button at the side of the ‘Hidden’ text. This will pull down all the available options.

‘Numbers’ will display your counter as numbers. You can change the style and colour of the numbers using the ‘Import new font’ button, or change the appearance of each individual number by using the right mouse button to select the number that you want to change. You can then use the Picture Editor to change the appearance of the numbers. For more information on using the Picture Editor turn to page 287

‘Vertical Bar’ will display your counter as a vertical bar that changes in length with the value of the counter. Note that the scale of the bar is worked out on the difference between the minimum and maximum possible values, so if you have a minimum of 0 and a maximum of 100 and the value of the counter is 10, the bar will be one-tenth its maximum length. You can change the appearance of the bar in a similar way to editing a Quick Backdrop object.

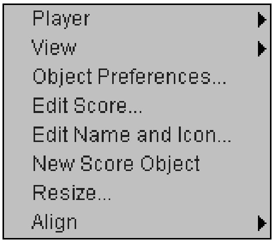
‘Horizontal Bar’- as ‘Vertical Bar’ but horizontal.

‘Animation’ will display a frame of animation for a different counter value.
For example, if you have ten frames of animation, and the minimum value of the counter is 0 and the maximum 100, then when its value is between 0 and 9, the first frame of animation will be shown. If the value was between 50 and 59, then the fifth frame of animation will be shown.
You will then be presented with a string of empty frames ready for you to insert your animation pictures. By selecting a frame with the right mouse button you can alter each frame using the Picture Editor. See page 287 for more information on the Picture Editor.

Score Objects

123

Note: Only the options related specifically to Score objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225.

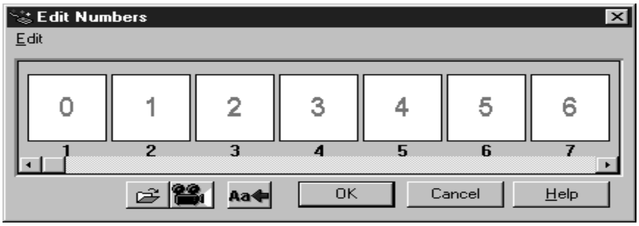


To stop a Score object being moved around the screen with the play area, go to the ‘Object preferences-Scrollings-Follow the playfield’ option, and turn it off. Select the Score object with the right mouse button after you have placed it on the play area to pull down a menu to do this.

Score objects are normally used to display a players score. You can create a score object by using the icon from the toolbar on the Level Editor screen.
You can then place your object anywhere on the play area.

To change which player the Score object relates to, select on the Score object with the right mouse button, then the ‘Player’ option. You can then choose which of the four players you want that particular score object to relate to.

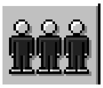
You can change the appearance of the numbers being displayed by selecting the ‘Edit Score...’ option from the score objects pull-down menu. When you do this you will be presented with a number editor like this;



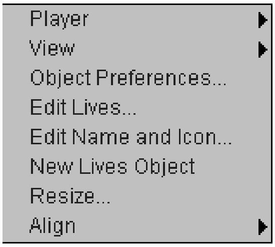
You can edit the individual frames by selecting them with the right mouse button. You can also use Cut, Copy and Paste from the ‘Edit’ menu at the top of the Number Editor.
To change the font size and colour, select the ‘Font’ button. This will take you to a Font Editor dialogue box where you can select the size and style, as well as the colour of the font that you want to use.

You also have the option to import a new font from disk using the ‘Import’ button.

Lives Objects



Note: Only the options related specifically to Lives objects are described below. All the other options available are described in the heading 'Common parameters for most objects' on page 225.

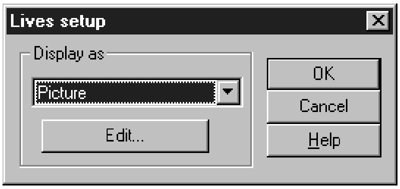


To stop a Lives object being moved around the screen with the play area, go to the 'Object preferences-Scrollings-Follow the playfield' option, and turn it off. Select the Lives object with the right mouse button after you have placed it on the play area to pull down a menu to do this.

Lives objects are normally used to display the number of 'Lives' that a player has left.

You can create a new Lives object by selecting on the icon at the top of the Level Editor screen.
'Player' allows you to assign the Lives object to players 1, 2, 3 or 4.

When you select the 'Edit lives...' option, you will be presented with a dialogue box like this;


Lives setup

Display as: Picture

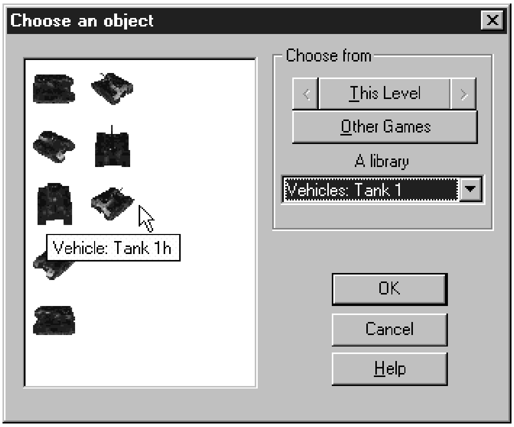
From here you can decide upon the look of the objects that are used to show the number of lives a player has left.

By selecting the button next to the text box, you will pull-down the following options;

‘Picture’ will display a picture for the object that you use. See the Picture Editor chapter on page 287 for more information on how to draw your own pictures.

‘Numbers’ will display the number of lives as a number. You can change the look of the numbers using the Number Editor. See ‘Score Objects-Number Editor’ from the previous topic for more information.

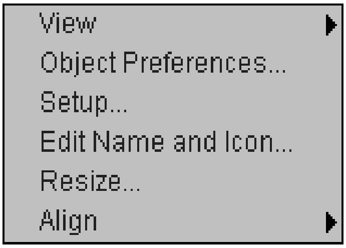
‘Pick image from object’ allows you to use an object already made up, either from one of The Games Factory’s extensive object libraries, or from another game.



This dialogue box shows ‘Vehicles: Tank 1’. You could then choose one of these objects and that would be used as the Lives character.

High Score Objects

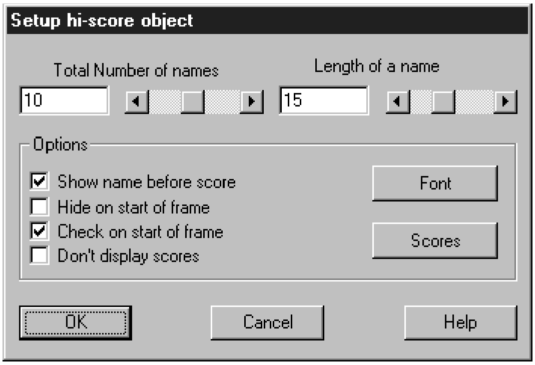
Note: Only the options related specifically to High score objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225.



You can create a new High Score object by selecting on the icon at the top of the Level Editor screen.

High Score objects are used to display a table of high scores for your game. You can change the appearance of the table, as well as the names that are shown in it to start with, as well as the initial scores. You may want to make it fairly difficult to get onto the High Score table at all by having the initial scores set quite high.

When you create a new High Score object, you will be asked to decide how it is set-up from this dialogue box;



Setup hi-score object

Total Number of names: 10

Length of a name: 15

Options

☑ Show name before score
□ Hide on start of frame
☑ Check on start of frame
□ Don't display scores

You have several options open to you.

‘Total number of names’ changes the length of the table. The higher the number, the more names that you will be able to fit onto your High Score table.

‘Length of a name’ sets the maximum length for a name to be displayed on the table.
The ‘Options’ tick-boxes toggle each option on or off.

‘Show name before score’ will show the name in front of the score. Having this option off means that the scores will be displayed down the left hand side of the table instead.

‘Hide on start of level’ will make the High Score table invisible when the level is first started. You can then make it visible after an Event has occurred from the Event Editor.

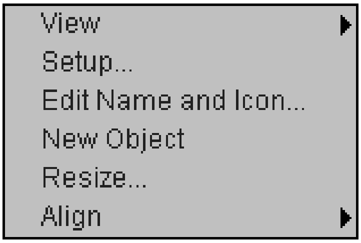
‘Check on start of level’ tests the table at the start of the level. Whenever the player moves to a new level, the table will be updated automatically.
‘Don’t display scores’ means that only when a player has reached the lowest level of the High Score table will a score be displayed.

‘Font’ allows you to select the font that the scores and names are displayed in. Beware that you do not choose so large a font that not all the scores can be displayed at once.

‘Scores’ allows you to set the initial scores. This ensures that the table will not be empty the first time that you use it, and means that a player will have to reach a minimum score before they can get onto the table.

Edit Objects

Note: Only the options related specifically to Edit objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225.

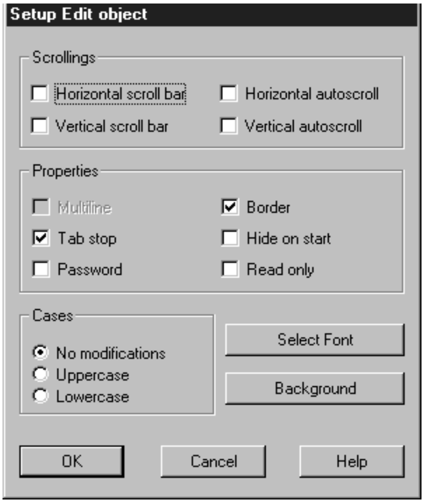


You can create a new Edit object by selecting on the icon at the top of the Level Editor screen.

Edit objects make it possible to retrieve information from the users playing your games. By default, the Edit object is displayed as a single line. You can use these objects to enter codes, passwords, etc.
You can also display them in a rectangular box to allow for multiple lines of text using the ‘Multiline’ option from the ‘Setup’ option.
After the user has entered some text, you can cut, copy or paste the text into another Edit object, or the Windows clipboard.

Once you have placed the object on the Level Editor screen, you can select the ‘Setup’ option from its pull down menu by using the right mouse button.
The Setup option enables you to define the settings for your Edit objects in the Level Editor.
Display the text in any colour, and any of the available True Type fonts. You can also hide your Edit object in the background for future use.

To change the parameters of your Edit object, select on it with the right mouse button, then select the ‘Setup’ option from the menu. This will take you to a screen that looks like this;



Setup Edit object

Scrollings

□ Horizontal scroll bar
□ Horizontal autoscroll
□ Vertical scroll bar
□ Vertical autoscroll

Properties

□ Multiline
☑ Border
☑ Tab stop
□ Hide on start
□ Password
□ Read only

Cases

● No modifications
○ Uppercase
○ Lowercase

The Scrolling options

The Scrolling options are used to add horizontal or vertical scroll bars to your edit zone.

Select in the 'Horizontal scroll bar' tick-box to add a horizontal scroll bar to the bottom of the Edit box. The player can drag a slider left or right to scroll through the contents.

Select in the 'Vertical scroll bar' check box to add a vertical scroll bar to the right side of the Edit box. The player can drag a slider up or down to scroll through the text.

'Horizontal Autoscroll' is used when you're displaying text which is too wide to fit into the current editing area. Once this check box is enabled, The Games Factory will scroll your text to the left whenever the user moves the cursor past the edge of the zone. The user will always be able to see the currently edited text on the screen. If this feature is disabled, the cursor will stop when it reaches the right-hand side of the Edit zone.

'Vertical Autoscroll' automatically scrols the text up when the user moves the cursor past the bottom of the editing area. If this check box is disabled, the cursor will stop when it reaches the last visible line.

Note: You can only use the Autoscroll feature if you're displaying your text over several lines, using the "Multiline" feature.

Properties

The Properties options change the way the Edit object will work on the screen.

'Multiline'

By default, your Edit zone will be displayed on a single line. This makes it easy to create editing areas for your dialogue boxes. You can define your dialogue box using a number of different Edit zones. Each object can read a separate piece of information into your game such as a name, password, or a search command.

If you want to display larger amounts of information, you can enable the 'Multiline' check-box. This feature arranges your text in a rectangular box on the screen. You could use it to display the pages of a story, the text in an encyclopaedia, or the product details in an advertisement. You could also create editing windows to enter free-form comments or notes from the user.

Note: Multiline mode doesn't change the size of the display area. You'll need to set the size separately using the Resize option from the objects pull-down menu in the Level Editor.

'Tab Stop'

The Tab Stop option lets you move around the Edit objects using the TAB and SHIFT + TAB keys, just like a standard Windows dialogue box. If the Tab Stop check box is enabled, the user can press the TAB key to jump directly to the next line. Press SHIFT + TAB will move the user backwards through the available Edit zones. The zones are arranged in the same order they were created in the Frame Editor. So, the first zone created will be the top of the list, and the last zone created will be at the bottom.

Note: You can only use the Tab Stop feature for SINGLE lines. This option won't be available if you're using Multiline mode.

'Password'

The Password feature provides a simple way of entering private information such as passwords or credit card details. The password feature hides the user's entries, replacing each letter with a star (*).

Note: The Password feature has no effect on the actual text. It just changes the way that the text is displayed on the screen. Read the password protected lines directly from your game using the 'Get text' action from the Event Editor screen. The Password option is not compatible with the Multiline feature. It's only available if you're displaying your zone as a single line.

'Border' adds a thin border line around the Edit zone.

'Hide on start' conceals the Edit zone at the start of the current level. You can display the Edit objects again using the 'Make object Reappear' action, or make it invisible again with 'Make object Invisible'.

'Read only' limits the Edit objects to a text display and stops the user from editing its contents.

The Control buttons

The Control buttons are used to set the style and appearance of the text and to choose a colour for the background areas.

Click the ‘Select Font’ button to display a standard font selector to choose the colour, size, style, and type of your text. The Games Factory lets you display your text in any True Type font and in any colour.

Click the ‘Background button’ to choose a background colour for your text. All Windows colours can either be a “Solid” colour or a mixture of two shades or a “composite” shade. If you chose a “composite” shade, only a single colour will be shown in the display area.

Cases

The Case options enable you to change the way the text will be displayed in the Edit zone.

‘No modifications’ shows the text exactly as it was originally entered by the user.

‘Uppercase’ automatically converts the Edit zone’s contents into CAPITAL letters.

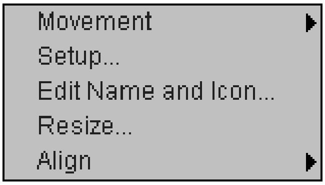
If you typed in a line such as: “Hello world”, it would be automatically changed to “HELLO WORLD”.

This feature is very useful because it forces the user’s commands into a standard format. Supposing you wanted to test for an instruction such as “Attack”. The user could type this command into an Edit object. You could now read it into your application with the ‘Get Text’ function, and test its contents using the ‘Compare two general’ values condition. Unfortunately, the user might type the instruction as “Attack”, “attack”, or “ATTACK”. This means you’d need THREE tests for each possible command. You could avoid this by enabling the Uppercase check box in the Setup dialogue box. This text would be automatically converted into the same format so you’d only have complete one test to check for “ATTACK”.

Lowercase forces the text to displayed in lower case letters. Taking the previous example, the line “Hello World”, would be changed to “hello world”.

AVI Objects

Note: Only the options related specifically to AVI objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225.



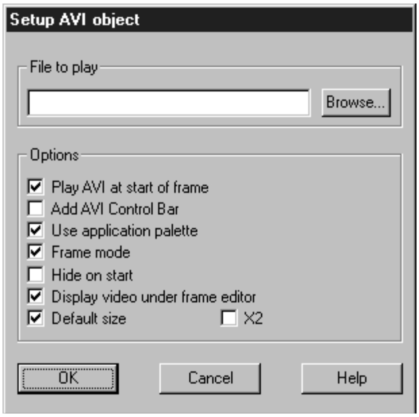
You can create a new AVI object by selecting on the icon at the top of the Level Editor screen.

AVI objects, or ‘Video for Windows’ objects can be used to add professional looking movie sequences to your Games Factory games. These sequences are displayed in their own “Viewing window” that can be moved around the screen.
Actions and conditions are available from the Event Editor so that you can fast forward, freeze frame, or jump to a new position in your clip.

Video for Windows or AVI files can be played on practically any Windows computer. You’ll be able to display your sequences in anything from 16 to 16 million colours (providing you’ve got an appropriate driver installed). The video will be smoother if you’ve got a little extra memory and a local bus video card. But you can get acceptable effects on even the most limited hardware.
Video for Windows objects require the Video for Windows driver before you can use them. If you don’t already have this driver, you can install it during The Games Factory setup.

Video clips can be taken from a range of sources. You can create your own using a Video Capture board or display existing files from a PD library or CD-ROM. You can also edit your clips into a professional video sequence using a commercial software package like ‘Adobe Premier’.

Selecting on the AVI object with the right mouse button will pull down a menu. Selecting the ‘Setup’ option will display the ‘Setup AVI’ dialogue box:


Setup AVI object

File to play

Options

☑ Play AVI at start of frame
☐ Add AVI Control Bar
☑ Use application palette
☑ Frame mode
☐ Hide on start
☑ Display video under frame editor
☑ Default size ☐ X2

In many cases, you'll want to choose a video clip for your object. You can type the filename into the File to play box, or click the Browse button to select a file from a disk. You can change the file at any time in your application using a Play new file action.

The 'Options' section enables you to customise the video.

'Play AVI at start of level' runs the movie automatically when the level is first entered.

'Add AVI Control bar' displays a control bar below the video area. You can select this control bar with the mouse to forward or reverse the video.

'Use The Games Factory Palette' converts the movie to the Games Factory's standard colours. This may make your video look rather odd.

'Hide on Start' makes the video completely invisible. You can display it again using a Visibility/Make object Reappear action.

'Display Video under the Level Editor' shows the first level of the current video file from the Level Editor.

'Default Size' displays the video in its original size. If this check box is enabled, you won't be able to resize the object from the Frame Editor.

'X2' is only available if you've enabled the Default Size option. Zoom X2 expands your video to twice its original size.

After you've entered you choices, click OK. The mouse pointer will change to an insertion point.

Click on the place on the play-area to drop the AVI object into place. If you haven't enabled the 'Default size' check box, you can adjust the playback window with the 'Resize' option from the object's menu.

Quick Time Objects

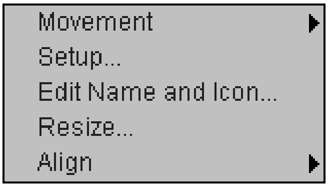
Note: Only the options related specifically to Quick Time objects are described below. All the other options available are described in the heading 'Common parameters for most objects' on page 225.

You can create a new Quick Time object by selecting on the icon at the top of the Level Editor screen.

Quick Time objects are virtually identical to AVI Objects and can be manipulated in the same way. See 'AVI Objects' (above) for more details.

MPEG Objects

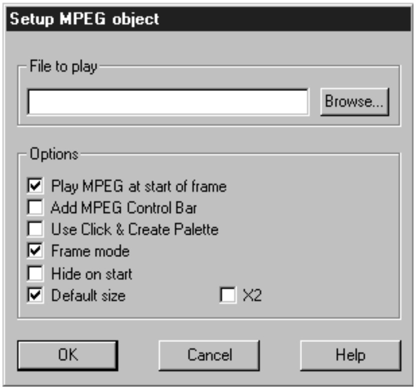
Note: Only the options related specifically to MPEG objects are described below. All the other options available are described in the heading 'Common parameters for most objects' on page 225.



You can create a new MPEG object by selecting on the icon at the top of the Level Editor screen.

The only option that is specific to MPEG objects is its 'Setup' option. Note that the 'Movement' option is identical to that of Active objects.

You call up the objects ‘Setup’ option by right selecting it when it is already on the play area. It is automatically produced when you first try to create it.
You will then be shown a dialogue box like this;



Setup MPEG object

File to play

Options

☑ Play MPEG at start of frame
□ Add MPEG Control Bar
□ Use Click & Create Palette
☑ Frame mode
□ Hide on start
☑ Default size □ X2

In the ‘File to play’ text box you enter the drive name and file pathway of the file that you want to play. Alternatively, you can click on the ‘Browse’ button and produce a file selector and then select your file to play from this.

Under the ‘Options’ heading there are several tick-boxes which will enable / disable these options;

‘Play MPEG at start of frame’ will automatically start playing the MPEG sequence when that frame (level) first starts. It would be useful to have this turned off if you wanted to start the sequence when something specific happens in your game, like the hero reaching the end of a level, or having them turning on a virtual TV!

‘Add MPEG control bar’ adds a ‘slider’ control bar, much like a scroll bar, at the bottom of the MPEG window which you can then move forwards and backwards to control the position of the MPEG sequence. A play / pause button is also displayed.

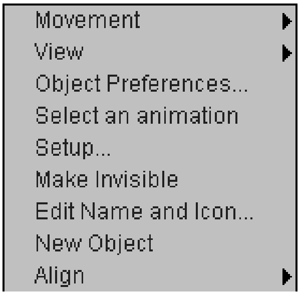
‘Use Games Factory palette’ enforces The Games Factory palette to be used by the MPEG object rather than it’s own colours.
‘Frame mode’ stops the time reference at the same time as the sequence is stopped. So, if you were to ‘Stop’ the MPEG, then ‘Resume’ it with this option on, it would simply pause the sequence, and carry on from where it was stopped.
If this option is disabled though, then the time reference will carry on running. Say, for instance, that the sequence was ‘Stopped’ for 2 seconds, and then ‘Resumed’, the frame it would resume from would be from 2 seconds further along the sequence.

‘Hide on start’ simply makes the MPEG object invisible at the start of the level. It can then be made visible via actions in the Event Editor.

‘Default size’ displays the MPEG at the size it was created in. Having this option on allows you to enable the ‘x2’ box which doubles the size of the MPEG display.

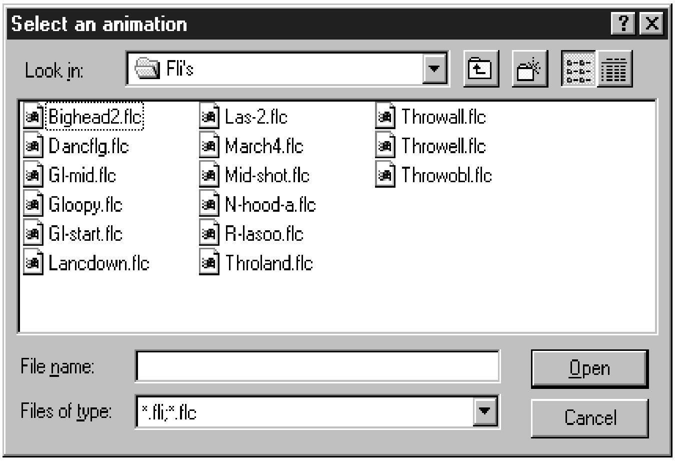
FLI / FLC Objects

Note: Only the options related specifically to FLI /FLC objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225.



You can create a new FLI / FLC object by selecting on the icon at the top of the Level Editor screen.

When you first create the object, or choose the ‘Select an animation’ from its menu (Pulled down by selecting on it with the right mouse button) you will be presented with the following dialogue box;



Select an animation

Look in: Fli's

Bighead2.flc
Dancflg.flc
Gl-mid.flc
Gloopy.flc
Gl-start.flc
Lancdown.flc
Las-2.flc
March4.flc
Mid-shot.flc
N-hood-a.flc
R-lasoo.flc
Throland.flc
Throwall.flc
Throwell.flc
Throwobl.flc

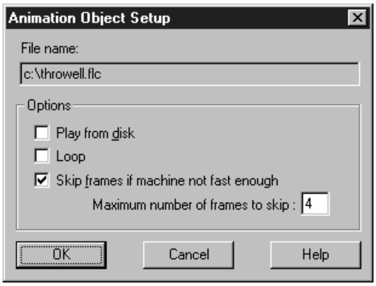
File name:

Files of type: *.fli;*.flc

It is from here that you select which animation to load in from disk. The Games Factory already has some animations on the CD under the ‘examples-media’ headings.

FLI objects run Autodesk compatible ‘FLI’or ‘FLC’ Animations directly from one of your levels in The Games Factory.
You can use FLI objects as an animated background for your games. They are perfect for the intro. before a game begins. You can also move them around the screen like an Active object, and test them using a wide variety of conditions and functions.

Check the Object help for detailed information about the available movement options:
FLI’s are similar to Active objects. They both generate their displays using a series of still pictures, or Animation Frames. These Frames are flashed onto the screen in sequence, creating a smooth illusion of movement.
The Games Factory CD contains a number of these animations for your use. Look at “Libs/Anims” directory for a list of the available animations.
You can treat them in a similar way to Active objects, and many of the parameters are very similar. However, they are normally much larger and take up considerably more memory space. You can change the way that they are handled from the objects menu. Select the ‘Setup...’ option. You will then be presented with a dialogue box like this one;



Animation Object Setup

File name: c:\throwell.flc

Options

☐ Play from disk
☐ Loop
☑ Skip frames if machine not fast enough
Maximum number of frames to skip : 4

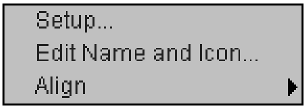
‘Play from disk’ will play the file from disk rather than loading it into memory. This can result in a fairly slow animation, but it does save a lot of memory space.

‘Loop’ will loop the playing of the animation continuously.

‘Skip frames if machine not fast enough’ does exactly that. On very slow machines, or if you are loading the image in from CD ROM, the animation may well be slow. To get around this you can skip frames to speed up the animation. Note that each object has its own time scale, so if your machine is fast enough already, changing this number will not make any difference to the speed of the animation, and The Games Factory will not skip any of the animation frames.

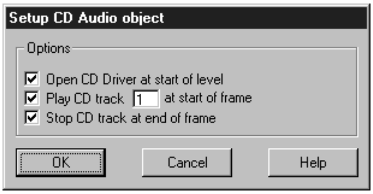
CD Audio Objects

Note: Only the options related specifically to CD Audio objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225.



You can create a new CD Audio object by selecting on the icon at the top of the Level Editor screen. CD audio objects add a professional touch to your Games Factory Games. You can include superb sound effects, real speech, or CD quality music. Don’t worry if you can’t create your own CDs. (Recordable CD-ROM’s are still quite rare!) You can still have a lot of fun producing your own jukeboxes, creating disco effects, or playing your favourite song for the background music in your game.

To change the CD Object parameters, select on the object with the right mouse button and choose ‘Setup...’ from the menu. You will be presented with a dialogue box like this;


Setup CD Audio object

Options

☑ Open CD Driver at start of level
☑ Play CD track 1 at start of frame
☑ Stop CD track at end of frame

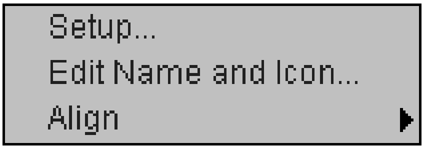
‘Open CD Drive at start of level’ activates your CD driver the moment your frame starts, and deactivates when it finishes. If you disable this check box, you’ll force The Games Factory to reset the CD. This lets you read files from the CD-ROM when the CD is not playing an audio track.

‘Play CD track [1] at start of level’ will begin your CD playing from the specified track number when your game starts. Enter a new track number to start the CD playing at a different track.

‘Stop CD track at end of level’ will stop the CD Audio system from playing at the end of the level.

MCI Objects

Note: Only the options related specifically to MCI objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on pag 225 .



You can create a new MCI object by selecting on the icon at the top of the Level Editor screen. Media Control Interface (MCI) is a standard way of controlling any multimedia device connected to a personal computer. MCI objects let you use this Media Control Interface directly from your Click & Create applications.

MCI objects work in the background, using the Windows multi-tasking system. They are completely independent from the rest of your The Games Factory game.

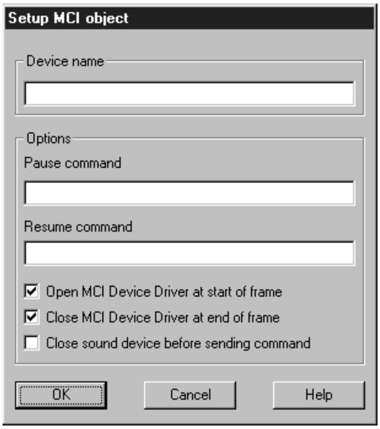
If you used an MCI object to play a large sound sample directly from the “.WAV” files on the disc, these samples would continue to play even when The Games Factory moves to a new frame, or exits from the game. You can play music from a CD, from beginning to end, without the slightest interruption, Run a “QuickTime” or “AVI movie file” in its own revisable Window and control external devices such as “MIDI sequencers” or “MPEG cards”.

The MCI object doesn’t appear on the screen. It works in the background so you can send commands to any media device currently running under Windows. The MCI object also lets you read information from any device used by The Games Factory.

MCI enters your instructions using command strings. These strings are in plain English, so they’re not hard to use, even for a non programmer. The only complication, is that you have to use complicated looking device names.

Before using these commands, it’s worth noting that The Games Factory also provides a range of objects for playing AVI files, QuickTime movies, or CD audio tracks directly. These offer a simple alternative to the MCI system.

If you select on the MCI object with the right mouse button and select the ‘Setup...’ option, you will be presented with a dialogue box that looks like this;



☑ Open MCI Device Driver at start of frame
☑ Close MCI Device Driver at end of frame
☐ Close sound device before sending command

‘Device name’

This box assigns your MCI object to the selected device. The name should be entered in lower case letters only. The Games Factory will test that you’ve entered a valid name before using the device. If you make a mistake, the commands will be ignored.

‘Options’

Some MCI devices required you to enter additional information when you pause the display:
‘Pause command’ enters the Pause options.
‘Resume command’ enters the Resume options.

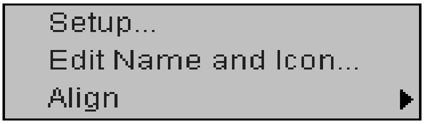
‘Open MCI Device Driver at start of level’ - by default, your MCI object will be activated the moment your level starts, and deactivated when it finishes. If you disable this option, the MCI device will need to be re-initialised before each command, using a separate ‘Open’ instruction in your game, from the Event Editor. This process will be
slower but you can use it when you want to control several devices at a time.

‘Close MCI Device Driver at end of level’

If this check box is enabled, the device driver will close when the game reaches the end of a level.

INI Objects

Note: Only the options related specifically to INI objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225 .



You can create a new INI object by selecting on the icon at the top of the Level Editor screen. When you have done this you can enter the ‘Setup...’ option by clicking on the object with the right mouse button and selecting the ‘Setup...’ option from the menu.

INI objects are vitally important to your Games Factory application even though they don’t actually appear when the application is run. INI objects save and load values into an “INI” file on the disc. They can be used to store configuration details and keep track of a player’s progress in a game. INI Files provide a simple way of saving your information for future use. You can use them to:

Save the player’s status in an adventure
Store the current level in an arcade game, with the player’s score and lives values
Hold the settings of a screen saver
Keep track of the current application directory
Enter the name and address of the current user

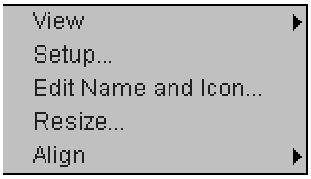
INI Files are divided into sections called ‘Groups’. Here are some examples:

[This is a Group]
[Here’s another one]
[Hero’s position]

As you can see, the names are surrounded by square brackets. These brackets are added automatically by The Games Factory.
Each ‘Group’ has its own set of ‘Items’. These ‘Items’ can be loaded with either a number, some text, or the current screen coordinates of an object.
The contents of these items can be read from the disk when they’re needed or changed at any time in your game.
All the functions of INI files are accessed from the Event Editor. See page 199 for more details.

Picture Objects

Note: Only the options related specifically to Picture objects are described below. All the other options available are described in the heading ‘Common parameters for most objects’ on page 225.



You can create a new Picture object by selecting on the icon at the top of the Level Editor screen.

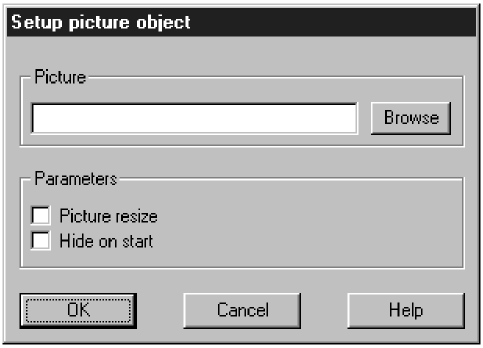
Picture objects provide an easy way of adding images to your application. Each picture object can be loaded with a file in either: “BMP”, “DIB”, “RLE”, “PCX”, “LBM”, or “IFF” formats.

The Picture will be displayed in its own window on the playfield. You can set the position of the window in the Level Editor. When you add a Picture Object to your application, the picture will be displayed as an “envelope” on the screen. This envelope sets the physical limits of the image. If the picture is larger than the envelope, the image will be resized to fit into the area.

You can change the size of the area using the “Resize” option from the object’s pop-up menu. You can also let the user adjust the size during the application. See the “Picture resize” option in the Setup dialogue box.

You can enter the ‘Setup...’ option by clicking on the object with the right mouse button and selecting the ‘Setup...’ option from the menu.

You will then be presented with a dialogue box that looks like this;



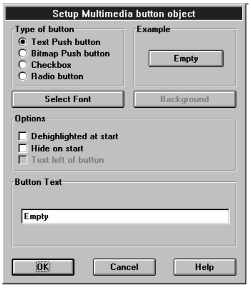
□ Picture resize
□ Hide on start

Click the file box to type in the name of an image to be loaded from the disk. If you'd prefer to choose the file from a directory, click the Browse button to display the selector.
'Picture Resize' allows you to resize the picture via the Level Editor. With this tick-box off, the object will be fixed in its current size.

'Hide on start' makes your picture invisible at the start of the level. You can make it visible at any time via the Event Editor using the 'Show' function.

Button Objects

As there are no Button objects already in The Games Factory's object libraries, you will always be creating your own, using the 'Create' icon from the toolbar at the top of the screen.
You will be presented with a setup dialogue box that looks like this;



Setup Multimedia button object

Type of button
- ● Text Push button
- ○ Bitmap Push button
- ○ Checkbox
- ○ Radio button

Options
- □ Dehighlighted at start
- □ Hide on start
- □ Text left of button

Button Text: Empty

At the top left of this dialogue box is the type of Button object you want to create.

'Text push button' will create a Button object that contains text which 'stands-out' from the screen, and can be 'pressed-in' by clicking on it with the mouse pointer.

'Bitmap push button' allows you to draw your own Button. If you selected this option instead of the 'Button text' heading you would be shown a 'Bitmaps' heading, with three choices, each of which will take you to a Picture Editor so that you can decide on the look of your button. See page 287 for more information on the Picture Editor.
The three choices are;
'Normal' allows you to decide what the button looks like in its normal state, i.e. not pressed or disabled.
'Pressed' allows you to decide how your button will look when it is pressed with the mouse pointer.
'Disabled' lets you change the look of your button when it is disabled.
Note that you cannot select a font with this type of button, nor type in text, as the whole look of the button is decided via the Picture Editor for that button.

'Checkbox' will create a Button object that will contain a check-mark that can be placed / removed by clicking in the box with the mouse pointer.

'Radio button'will create a Button object that normally has several options to it, associated with a line of text. When you click on one option, it turns any others off, and by the same virtue, you can only turn off an option by clicking on another.

‘Example’ displays an example of what your button will look like.

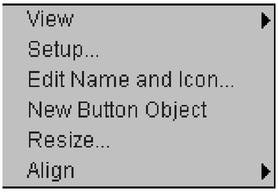
‘Background’ changes the colour of the background of your buttons. Only available with ‘Checkbox’ and ‘Radio’ buttons.

‘Select font’ allows you to select the font you are going to use for any of the text that you use for your buttons.

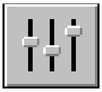
‘Options’ allows you to set the status of the button when it is first displayed.
‘Dehighlighted at start’ will dehighlight the button by default, and make it disabled. You would then have to enable it via an Action performed via the Event Editor.
‘Hide on start’ hides the button object initially. Note that you cannot select on it whilst it is invisible. You can make it visible by performing an Action upon the button via the Event Editor.
‘Text left of button’ displays the text associated with a button on the left of the button instead of the right. Only available to ‘Checkbox’ and ‘Radio’ buttons.

‘Button text’ is not available for ‘Bitmap’ buttons. You can directly enter the text that you want to be associated with a button in here. Simply select in the box with the mouse pointer to change the text. When you are creating a ‘Radio’ button object, however, this box will display the text associated with each different radio-button. You can use the ‘Add’ option to add another radio-button option to your object.
Use the ‘Edit’ option to change the text of a particular button option, ‘Delete’, to delete it. You must select the option you want to change / delete, using the left mouse button. It will then be highlighted in your Windows highlight colour.

When you have setup you button object, you can then place it on the play area by clicking the ‘OK’ button at the bottom of the dialogue box, then place it using the left mouse button.
All the other options available are described in the heading ‘Common parameters for most objects’ on page 225. You can access these by right-clicking on the button object once it has been placed on the play area. The menu looks like this;



Mixer Objects



There are only two options available for the Mixer objects on the Level Editor, ‘Edit name and icon’ and ‘Align’, which are both discussed in the ‘Common parameters for most objects’ on page 225.

Mixer objects are used to change the left / right volumes of samples and Midi tracks whilst they are being played through your speakers. Virtually all of the options available to you for Mixer objects are accessed via the Event Editor, on page 204.

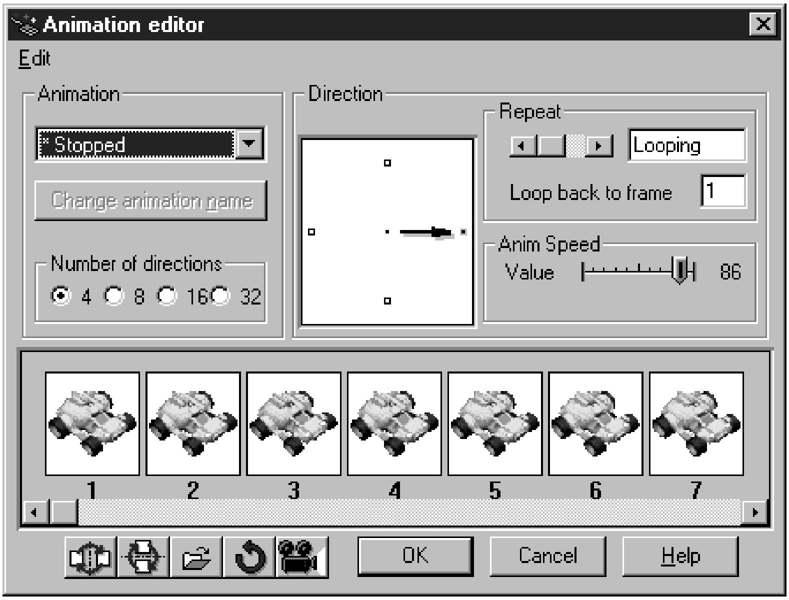
Animation Editor

In order to edit or create an animated object, you must be in the Level Editor screen, and you must have either picked an object from the object shelf and placed it in the play area, or be creating an Active object using the ‘New Object’ tool on the toolbar at the top of the screen.

Note: only Active objects can be animated from the Animation editor.

First we shall deal with editing an Active object that already exists. This will also show you most of the steps required to create your own animated object.

Select the object that you want to edit (on the Play Area of the Level Editor screen) by using the right mouse button. This will pull down a menu of options for that object. Move the pointer to the ‘Edit animations...’ option and click the left mouse button.
You should now have a screen that looks something like this;



Animation editor

Animation: * Stopped

Number of directions: ● 4 ○ 8 ○ 16 ○ 32

Repeat: Looping

Loop back to frame: 1

Anim Speed Value: 86

You can use this screen to change the name of your animation, preview what it looks like when you run it all together, create different animations depending on the direction your character is moving in, or different animations if you character is climbing, running, etc.

Important note; To ‘select’ all the frames of an animation, use the ALT+A keyboard shortcut. This will allow you to move or delete a whole sequence of animation in one go.
Let’s explore this screen a bit more.

Animation Speed

To the top right are the controls for the speed of the animation as well as the number of times it will repeat the total animation loop before it stops. You can select ‘Looping’ by moving the ‘Repeat’ slider below 1, which will repeat the animation sequence endlessly. You can also change which frame number it loops back to in the ‘Loops back to Frame’ box, which could be used for a man getting up from a crouched position, then running away, the first couple of frames being crouched down, then looping the animation back to the running sequence only.

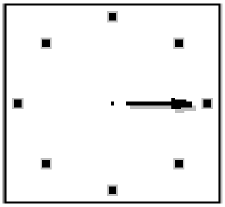
Note that for some animations there are two sliders for the control of speed. There are only two controls for when the object would normally be moving e.g. the ‘Walking’ animation of a character. If the character was ‘Stopped’ then there would only be one slider, which controls the speed of the animation regardless of movement.

Minimum speed controls the speed when the character is not moving. Setting this to zero will halt the animation when the character is not moving, having it set higher will have the animation running all of the time, which may well look unrealistic if your character is running frantically without moving!

Maximum speed controls the maximum rate of the animation when the character is at full speed. Note that the rate of animation will be proportional to the speed of the
character, in between the ‘Minimum’ and ‘Maximum’ settings. To create a realistic running action you may well need to change the ‘Maximum’ setting to a similar value as a characters speed across the screen.
For example, if you were to set a characters movement speed high and the animation speed low, it would appear as though they were being ‘dragged’ across the screen, if you had the animation speed high and the actual movement speed low, it would look as though they were trying to run fast on an icy floor.

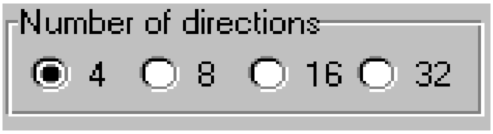
Animation Direction

The control for which animation plays when your character moves in a particular direction is situated in the ‘Direction’ box. You can select a direction by using the mouse to click on one of the buttons on the arrow ‘clock face’.



Note: An easy way to create several different directions from one animation is to click the ↻ icon. This may may well turn your objects upside down for some of the directions but you can correct this using the controls outlined below.

You can select the total number of different directions by clicking on one of the buttons. Obviously, the higher the number of different directions the smoother will be the turning effect, particularly useful for race-car movement, but 4 different directions will be fine for a platform character.

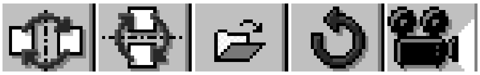


Number of directions

● 4 ○ 8 ○ 16 ○ 32

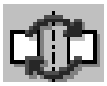
You can ‘manually’ fill the other directions using Cut, Copy and Paste from the Edit pull-down menu (on the Animation Editor). It is recommended that you use Copy and Paste, otherwise you will simply move one frame to another location rather than copying it.

Manipulation Icons



These icons are used to manipulate all the frames of the animation being displayed on the Animation Editor screen, so some care must be excercised when using them, except the ‘View Animation’ icon which merely previews the whole sequence for you.(Described below.)

Horizontal Flip



The horizontal flip reverses all the images from left to right. Great for turning your characters to face the other way. Note that both the Action Point and Hot Spot will also be moved.

Vertical Flip

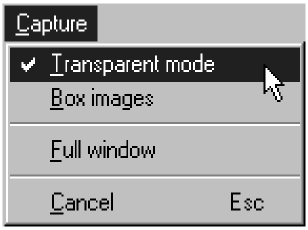
Turns all the images upside-down, Useful for when you have created several new directions from your original animation using the ↻ icon and some are the wrong way up, or for when you want a character to walk on the ceiling! It also moves the Action Point and Hot Spot.

Import

This will grab one or more images from disk and load them into the Animation editor, where you will be able to manipulate them as a normal animated, Active object. You can load up images stored in PCX, LBM, GIF, TIF or BMP formats. You can even load up FLC and FLI files, but beware! These files are normally very large indeed, and may contain well over 50 separate images, therefore will use up large amounts of memory space.

Once you have entered the Import function, you will be asked to select a drive and the types of files to look for, which can be any of those described above.

Once you have selected the file that you want, you will be asked to select a 'Capture' style. This decides the method by which you grab one or more images from the file you are looking in.



Transparent mode will have the background colour of the images that you pick as being transparent, rather than the actual colour on the original image.

For example, if the background colour of the images that you select is red, and you have Transparent mode on, instead of pasting them with a red background, The Games Factory will paste them into the Animation Editor with a transparent background, not a red one.

If Transparent mode is off, then whatever colour the background is in the frames that you grab will be transferred as well. So, if the background colour was red, then it will be pasted into the Animation Editor as red.

Box Images grabs all of the images that are within boxes, within the selection area that you define with the mouse, and will paste each boxed image into its own animation frame, in order from top left to bottom right.
If, for example, there are ten boxed images on the screen, and you drag a selection area that completely encloses nine of them, but only half crosses the tenth, only nine of the images will be taken. If box mode were off, one image comprising the entire contents of the selection area would be taken and pasted into only one animation frame.

Full Window will grab the whole area inside the ‘Capture’ window (not just inside your selection area) and paste that into your animation frame. Note this only produces one frame of animation comprising the entire contents of the window.
If, however, you have Box Images on at the same time, all the boxed images in the window will be grabbed and treated as separate frames of animation.

Create different directions

As briefly outlined above, allows you to quickly and easily create many different directions from only one animation direction. Be aware though, if you have several animation frames and 32 different directions, you will be using a lot of memory space. Also, it literally just turns your object around, so if you start with an object that faces to the right, when it is rotated to the left it will be upside-down.

Right, let’s get on with drawing a frame of animation and then copying it out to fill several frames!

Creating / Editing a single frame of animation (Level Editor)

Now try selecting an Active object with the right mouse button, then enter the ‘Edit Animations’ option. When you are presented with the ‘Animation Editor’ screen click on the middle of frame number one with the right mouse button. This will pull down a menu. Select the ‘Edit Frame’ option. This then takes you to the drawing screen, or ‘Picture Editor’ for the animation. Here you decide the colour, size and shape of your animation frame. To learn more about this you can go to the Picture Editor chapter, on page 287. Don’t forget to come back, though!

In the picture editor you can modify each frame individually to suit your own requirements. To save you laboriously copying each frame out, you can go back to the animation editor and use the following commands, which are select from the menu produced when you right click on the frame you want to modify.

Adding animation frames

The key to animation is to create several frames that are slightly different to one another, that when run quickly together give the impression of movement. Should you be starting from scratch, or using one initial frame, it is far easier to copy each frame and then modify it as you go along than it is to draw the whole thing out for each frame.

If you look at the Vehicle animation you can see how this is done.

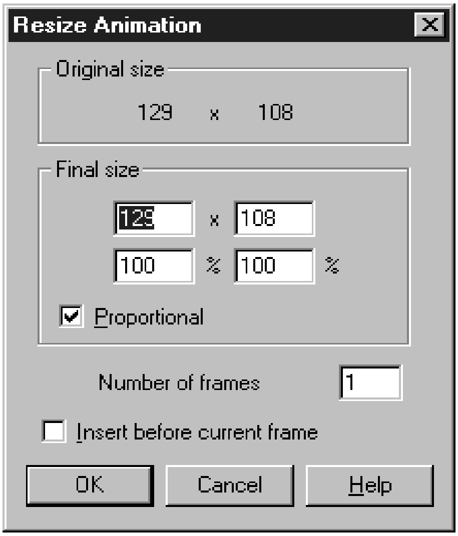
First of all, click on the first frame of your animation with your right mouse button, and select the 'Insert' option. This will insert a direct copy of your first frame into frame number two.

You can also do the same thing using the left mouse button. Simply select the image that you want to copy with the left mouse button, hold the button down, then drag the image to the frame you wish it to be copied to (in this case frame number two) and let go of the button.

You can now enter the Picture Editor for that frame and modify it slightly, and when you are happy, insert a copy of that frame into frame three, which you can then modify further, continuing the process until you are happy with it. You can check on your progress at any time by using the 🎥 icon.

Resize Zoom

The Resize Zoom option allows you to automatically create several frames of animation that will gradually shrink or grow in relation to the frame that you start off with. When you select the Resize Zoom option you will be presented with a dialogue box that looks like this;



Resize Animation

Original size

129 x 108

Final size

129 x 108

100 % 100 %

☑ Proportional

Number of frames: 1

□ Insert before current frame

Under the 'Original Size' is displayed the size of the frame that you are zooming from.

Under the 'Final Size' heading you can select the size of the frame that you want to end up with. You can either enter in the actual size of the X and Y dimensions of the final frame, i.e. the number of pixels wide and high that the final frame will be,or you can enter in a percentage ratio. For

example, 50% would shrink that particular dimension by half.
Note that when the 'Proportional' tick box is filled and you change any of the values, then the other dimension will be changed proportionally. For example, if you changed the width to 50%, then the height would also be changed by 50%.
With the 'Proprotional' tick box off, you can change the dimensions independently.

The 'Number of frames' allows you to select how many frames are going to be used to grow / shrink the object. The higher the number, the smoother the effect will be.

'Insert before current frame' will grow / shrink the object before the current frame.
For example, if you had selected the first frame of an animation, and you were shrinking it by half, and you were using ten frames to do so, and you inserted them before the current frame, then when you had finished the process, the first frame would be the smallest, getting gradually bigger over the next ten frames until you arrived back at the original frame.
If you did exactly the same process, but left this option off, then the ten frames following the initial frame would become gradually smaller and smaller, until they reached the one that was half the size of the original.

Rotate

The Rotate option can be used to insert several frames of animation that gradually rotate either clockwise or counterclockwise from the initial frame. You can choose the number of frames used to perform the rotation, ranging from 4 to 32. Obviously, the higher the number of frames used, the smoother the rotation effect will be.
The rotation frames will be inserted after the initial frame.

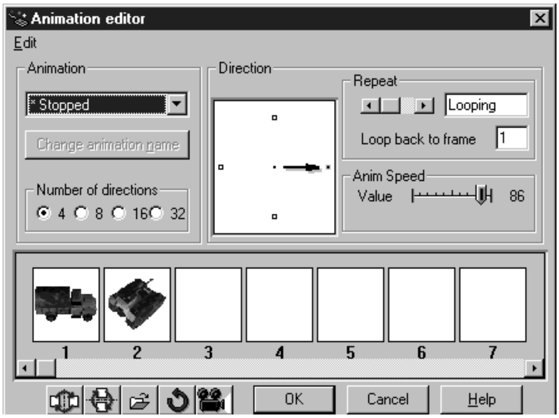
Morphing

The Morphing function allows you to change one frame into another, allowing for stunning transformations. You can use this function to smoothly upgrade your spaceships, rather than clumsily have them change appearance, or have creature transformations in adventure games.
You may well have seen advanced versions of this technique in Hollywood movies. Now you can do it on your PC!

To 'Morph' an object, you first of all need an object to start with, which must be an Active object, plus the object which you want to end up with, which must also be an Active object.

The easiest way to do this is to select your second (destination) object first, enter the animation editor, select the frame you want, then 'Copy' it using the Edit function from the animation editor (top left), then exit the animation editor.

Now, select the first (start) object, enter the animation editor for that object, and 'Paste' the finishing frame after the the frame you want first. You may well have to enter the Edit mode and unselect the ‘Remove all’ option, which will remove all previous animation frames if you try to paste a new frame into that animation sequence.



Animation: * Stopped

Number of directions: ● 4 ○ 8 ○ 16 ○ 32

Repeat: Looping

Loop back to frame 1

Anim Speed Value 86

Note: This function works best when there is only one frame for the original object. If there are more than one, delete the redundant frames so that you are only left with the start frame and the finish frame. You can leave part of the original animation if you want, but bear in mind that the morphing part of the animation will not start until the original animation has run. This can also use a lot of memory.

When you have done this, select the first (start) frame with the right mouse button, then select the ‘Morphing’ option. You will then be presented with a screen that looks like this;

Using this editor screen you can select the number of frames used to ‘Morph’ from the original frame to the final one. This is done by selecting in the “No. Images” box. The more frames that you have, the smoother and more realistic the end result will be. However, this will also be time consuming and uses a lot of memory. By default The Games Factory uses 10 frames, which is a compromise between smoothness of the effect, and speed of execution. You can also choose the definition of the change in the grid box at the top of the Editor.(By default set to 5X5). Having a higher number here will raise the number of fine changes that are made in between each frame, although again be aware that a very high definition combined with a large object will take a long time to do, and may use up lots of memory.

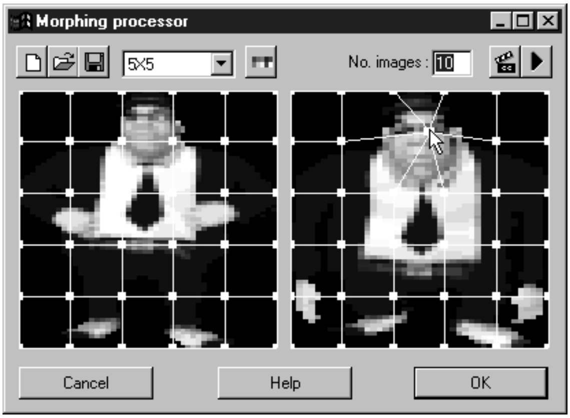
The 🎬 icon actually starts the whole process off. When it has finished you will be presented with a whole new animation strip, ‘morphing’ smoothly from the first image to the last.

Use of the Grid

You can change the colour of the grid using the 🎬 icon, which you may well find very useful depending on the colour of the images that you are changing.

The grid is used to define common points between the two images that you want to move. The grid is composed of a number of ‘elastic bands’ that you can stretch to fit various strategic points on the objects that you are morphing.

For example, if you were to morph from one face to another, you would stretch the grid so that each point corresponds to an eye, nose, edge of mouth etc. Make sure that the same grid point on the other object is used to correspond to the same feature.



Morphing processor

5X5

No. images : 10

For example, if the point one from the left and one from the top is placed on the eyebrow on the first object, the very same grid point (one from the left, one from the top) must be used on the eyebrow of the second object, even if it means stretching the point right across the image window. Try to ensure that you do not get grid points ‘crossing over’ each other, as this will spoil the effect also.

Setting the Hot Spot of an object

The Hot Spot is an invisible handle used to drag your images around on the screen. It is used as a reference for the X, Y coordinates of your object. Each image can have its own seperate Hot Spot. As a default, when you create a new Active object the hot spot is automatically positioned at the top left corner of each image. You can, however, move it anywhere you like. You can show the Hot Spot by going to the 'Options' heading and select the 'Show-Hot Spot'.

Try to position it centrally if your object is going to have several different directions, otherwise it will 'jump' when you change direction.

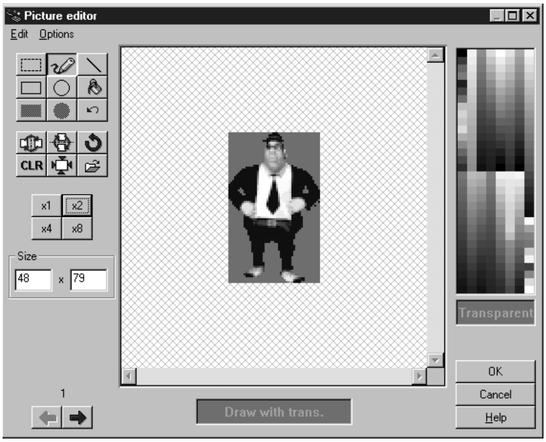
Setting the Action Point of an object

This is the point where things like bullets are fired from objects. If, for example, you had a large spaceship with a gun mounted aboard, you would want to set the Action Point to the end of the barrel of the gun. This means that this is where your bullet would first appear.

You can show the Action Point by going to the 'Options' heading and select the 'Show-Action Point'.

The Picture Editor

The Picture Editor is used for creating your own animation, background objects, icons and Quick backdrop objects. Because many of the features are identical for all these types of objects, they are summarised in one chapter here.



Size
48 x 79

In the centre of the screen is the area you will be drawing / editing on. If the area is to large to fit in this screen, you can either scroll around it, or zoom in or out using the zoom control buttons. (On the left, x1, x2, x4 & x8)

To the right of this is the palette, which allows you to pick and choose the colours you draw with.

The Colour bars

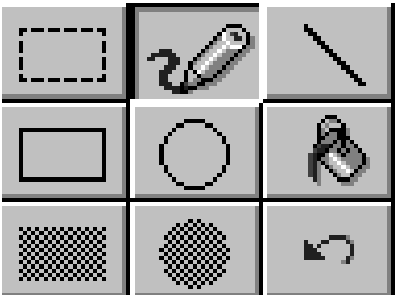
These allow you to select whether you draw with a transparency, or solid colour. You can draw with a normal colour by selecting in the palette, to draw with the transparency click on the ‘Transparency’ box. When you do this the colour box wording will change to ‘Draw with trans.’ This will allow you to remove parts of the object and make them transparent, rather like an eraser. The transparent areas are denoted by a specific colour, normally green-blue, but you can change this using the ‘Options’ menu at the top of the Picture editor.

Anything that you decide as being transparent will ‘show through’ that area, very useful for leafless trees, or large animated objects that have gaps in the structure. Anything in the background will be seen through the transparent areas.

The Palette

You can chose a colour to draw with simply by selecting a colour from the palette with the mouse (and the left mouse button). Note that if you were previously drawing with transparency, this turned off as soon as you select a colour. The current drawing colour will be displayed in the 'Draw with' box.

The drawing tools



These are the tools used to actually draw with, including an 'Undo' tool to undo the last thing that you did.

The Pen tool

This tool is used to either draw one pixel at a time or to draw a freehand line. Not that if you are drawing a freehand line and move the mouse too fast, you will end up with a broken line.

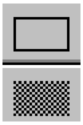
Handy Hint!

To create a 'grainy' effect, use this tool with several different colours and go over the drawing area very fast. This will lay down pixels of colour spaced well apart. Doing this over a gray background with brown, black and white will give the effect of a brown, stone tile.

The Line tool

This lets you draw perfectly straight lines using the mouse. Simply click on the point that you want the line to start from, hold down the mouse button, now notice that when you move the mouse it 'drags' a line behind it. Move the other end of the line to where you want the end of the line, then let go of the mouse button and you will have drawn a line between the two points.

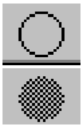
The Rectangle and Filled Rectangle tools



These tools allow you to draw rectangles and squares more easily than by trying to construct them of 4 separate lines. The principle is the same as that for the line command. After selecting the icon you want, place the pointer at the place you want the top left corner to be, press and hold down the mouse button then drag the rectangle to the shape you want.

Using the same procedure using the filled rectangle icon will do exactly the same, except that it will produce a solid rectangle of the colour you have selected.

The Ellipse and Filled Ellipse tools



These tools allow you to create ellipses and circles, both filled and empty. In order to use them, select the icon you want, then place the mouse pointer at the place you want as the centre of the circle. Holding down the mouse button, move the pointer away from the centre of the circle. You will notice that an ellipse is drawn out in proportion with how far you move the pointer from the centre point, moving sideways from the centre will stretch out the ellipse horizontally, moving it up (or down) from the centre will stretch it out vertically.

The Fill tool

The fill icon fills an area with a solid block of colour. The area to be filled should be completely enclosed. If there is a gap of even only one pixel, the colour will ‘leak’ out into other areas of your frame.

The Undo tool

This is used to undo the last step that you performed. It will only undo the last thing you did, though. Pressing it again will undo what you have undone, in effect re-doing it!

The Selection tool

This defines a rectangular block which can be cut or copied from your image. When it is selected, you can move the mouse where you want the top left corner of your block and then drag a box down around the area you want. If you make a mistake, click once on another part of your image and try again.

Once you have selected a block you can save it into memory using the ‘Cut’ option from the Edit menu. (Top left of the Picture Editor.) The original area will be replaced with a block of the transparent colour.

Using the ‘Copy’ image will do the same thing without affecting the original image.

After you have grabbed your area, you can now copy it onto the image using the ‘Paste’ command. This provides you with a rectangular box which can be used to position your block over the image. You can fix it in place with a single click of the mouse button.

Horizontal flip

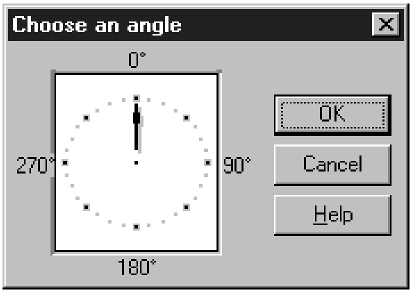
Reverses your image from left to right, just like a mirror.

Vertical flip

Turns the whole image upside down.

Rotate

Allows you to rotate the whole image very finely. When you select this function you will be asked to select an angle to rotate the image to. The arrow will point to the current direction. Simply click a new direction on the clock face then select ‘OK’ and the image will be turned by the angle you have specified.



Clear

CLR

This will blank the image window, ready for you to start from scratch.

Shrink

The shrink function removes any unnecessary transparent border areas from the image, reducing its size. This is very useful for saving memory and will speed up your game by removing overly large images, trimming them to the minimum required.

Import

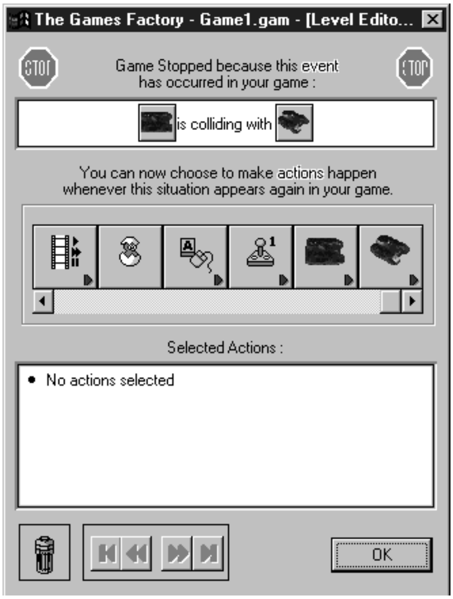
Loads an image from disk. For more detail see ‘Import’ under the Animation Editor on page 280.

The Step Through Editor

The Step Through Editor can be very useful for testing your games. The Games Factory will tell you every time an Event occurs that you have not already defined in the Event Editor. Until you 'find your feet' using the Event Editor, it can make constructing an Event grid very easy. Every time you run your game using the Step Through Editor, The Games Factory will stop the action when a new Event occurs, tell you what that Event is, then give you the option of adding an Action to that Event. You don't have to write out a new Event line, simply select which object that you want to change.

To enter the Step Through Editor, simply select on the 🎬 icon at the top of the screen. Note that in order to do anything at all, you must have at least one Active object on the screen.

In this example, objects were placed on to the playfield from the 'Vehicles: Tank 1' library. When the Step through button was selected Tank 1a moved around the screen until it collided with tank 1b, and the following dialogue box was then displayed.



The Games Factory - Game1.gam - [Level Edito...

Game Stopped because this event has occurred in your game :

is colliding with

You can now choose to make actions happen whenever this situation appears again in your game.

Selected Actions :

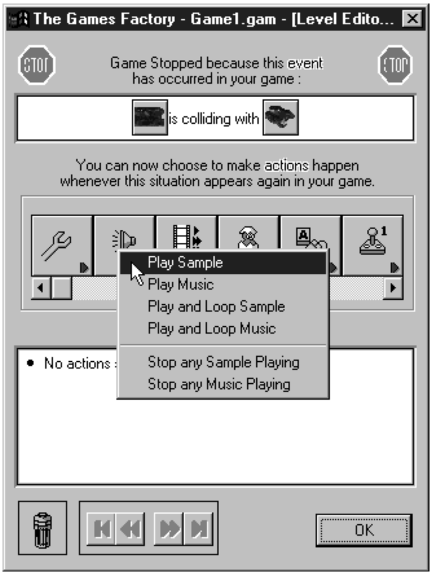
- No actions selected

The current Event that has been detected is displayed in the text box at the top of the Game Control Panel. In this case the event that has been detected is that the ball has collided with a brick, and from this dialogue box you can decide on the Actions to associate with the event.

Simply click on the relevant object that you want to 'change', then you will be presented with a menu of options to do to that object. When you select one, The Games Factory will make an Event line in the Event Editor describing the Event and the Action that you have just decided upon.

In this example an appropriate action to perform would be to play a music sample when the two objects collide.

For a complete description of all the Actions that you can perform on an Object, turn to that Object heading in the Event Editor chapter, within the Actions sub-chapter.



The Games Factory - Game1.gam - [Level Edito...

Game Stopped because this event has occurred in your game :

is colliding with

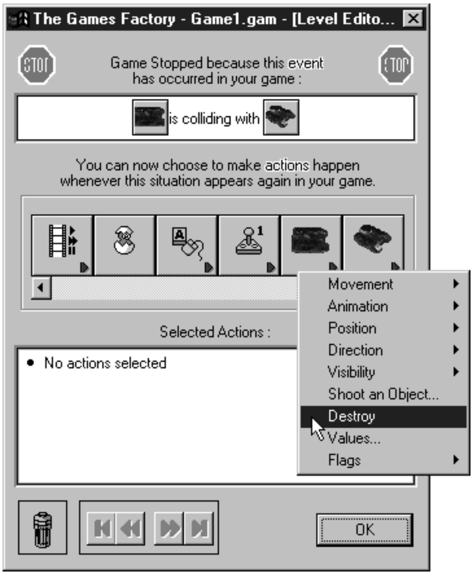
You can now choose to make actions happen whenever this situation appears again in your game.

No actions :

If you cannot immediately see the object that you want to 'change', use the mouse pointer on the scroll buttons at either side of the scroll bar to move through the objects.

You can add further Actions by clicking on the objects on the scroll bar. In this way you can very easily create new Event lines, however, it can be difficult to do things like manipulate Actions depending upon hidden values such as Flags or internal values. For the most part, you will need to construct Event lines in the Event Editor to do this.

In this example, an action has been created whereby the tank 1a is destroyed when tank 1b collides with it.



The Games Factory - Game1.gam - [Level Edito...

Game Stopped because this event has occurred in your game :

is colliding with

You can now choose to make actions happen whenever this situation appears again in your game.

Selected Actions :

- No actions selected

All the actions that you have chosen are summarized in the text box at the bottom of the Game Control Panel. By selecting on the text of an Event from this text box, you can delete it by dragging it into the trashcan at the bottom of the Game Control Panel, or by clicking on it with the right mouse button.

In addition to the controls described above, you also have the Event Control buttons at the very bottom of the Game Control Panel. These allow you to look through all the Events that have so far occurred, then add or delete actions to them.

Event Control Buttons

⏮ Clicking on this button will take you to the first Event to happen using the Step Through Editor. You can then add, edit or delete the Actions that you have associated with that Event.

⏪ This will move you to the previous Event, which you can then edit.

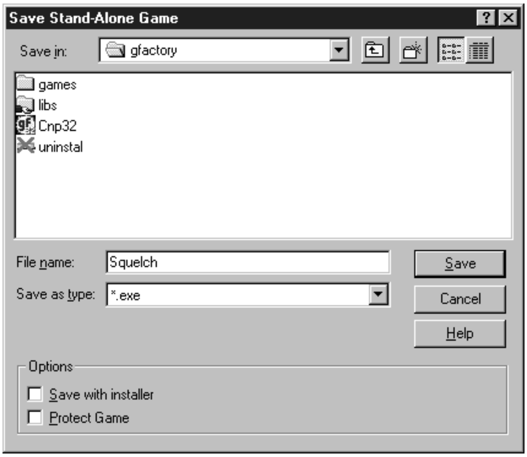
⏩ This will move you onto the next Event, which you can then edit. Note that you can only use this button after you have been looking at a previous Event.

⏭ This button moves you to the last Event to be produced by the Step Through Editor, which you can then edit. Note that you can only use this button if you have been looking at a previous Event.

Saving your Games as Standalone Software

When you perform a 'Save' or 'Save as...' using The Games Factory, you are actually just saving data which The Games Factory uses to create your game.
In other words, you need to be running The Games Factory before you can run these games, they will not run on their own.
When you save as a standalone, you can choose to save your game as just an executable file, or as a discrete application, i.e. you could save it on to floppy disk and run it as a game on a machine that does not have The Games Factory installed. Note that some Active object files are very large and may even use an entire disk just for the data for that object, so beware, if you are using a lot of complex sound and graphics, you may need a lot of floppy disks! The Games Factory does compress the data so considerably less space is required on disk than just a straightforward save.

When you choose the 'Save game as standalone' option you will be shown a warning about selling your games.
You are prohibited from selling any game or program created by you either as an independent product or as Shareware.
Once you have read this information you will be shown a special file selector;



Save Stand-Alone Game

Save in: gfactory

games
libs
Cnp32
uninstal

File name: Squelch

Save as type: *.exe

Options

☐ Save with installer

☐ Protect Game

In the top left is the text box for you to enter the file name for your game to be saved as, the text box on the right is the drive and directory you wish it to be saved in.

At the bottom are the ‘Options’ for the way your game will be saved. By default, The Games Factory will simply save your game as an executable file.

If you enable the ‘Save with installer’ box then your game will be saved as a complete entity that does not require The Games Factory to run, complete with installation files.
You will then have a file ‘setup.exe’ saved with your game. When you run this it will install your game to your machine, complete with name and icon.

‘Protect game’ prevents anybody using The Games Factory from editing your game. With this option enabled you cannot gain access to the different Editor screens for that game. Be careful not to overwrite unprotected versions of your game with a protected version as you will not be able to make changes to your game afterwards.

You can now save your games to give to your friends and family. You can make unlimited copies, but you are prohibited from making any financial advantage.

Selling your games

If you wish to sell your games, you are advised that you should purchase ‘Corel Click & Create’, available from the Corel Corporation.
Click & Create is a sister product to The Games Factory, and you can run games you have created with The Games Factory on Corel’s Click & Create.
The purchase of Click & Create includes a licence to sell your games as standalones.
You should contact either your local software dealer, or contact Corel directly at;

Internet: http:\\www.corel.com\click&create

Addresses:

Customer Service
Corel Corporation, c/o UCA&L, 49 B Bracken Road, Sandyford Industrial Estate, Sandyford, Dublin, Ireland

Corporate Headquarters:
Corel Corporation, 1600 Carling Avenue, Ottawa, Ontario, Canada, K1Z 8R7

In North America Phone: 1-800-77COREL (772-6735)
Outside of North America: (353) 1-706-3912

Saving as a Screen Saver

You can also save your games as a screen saver. This is subject to the same restrictions as saving as a standalone.

The Games Factory will save your game so that you can then select it from the ‘Desktop’ of your Windows program as your screen saver.

In Windows 3.x you can go to the ‘Main’ icon, then the ‘Control Panel’, then the ‘Desktop’ setup.

In Windows 95 you can go to the ‘Start’ icon, then the ‘Control Panel’ option, then ‘Display’, then ‘Screen saver’.

Europress Web Site

If you are on the Internet you can contact the development team directly. We would be very interested to hear from you, whether you want the latest handy hints, general help, or to show us your games.

The Games Factory is always undergoing further development, so if you want the latest news, we’ll keep you posted on new object and sound libraries, links with other developers, and game tutorials and ideas etc.

The address is;

http:\\www.europress.co.uk,

Or email us at : gamesfactory@europress.co.uk

Hope to hear from you soon!

INDEX