

# *Ultimate Video Game Systems*

**XBox 360, Nintendo Wii, Sony Playstation 3, Sony PSP**

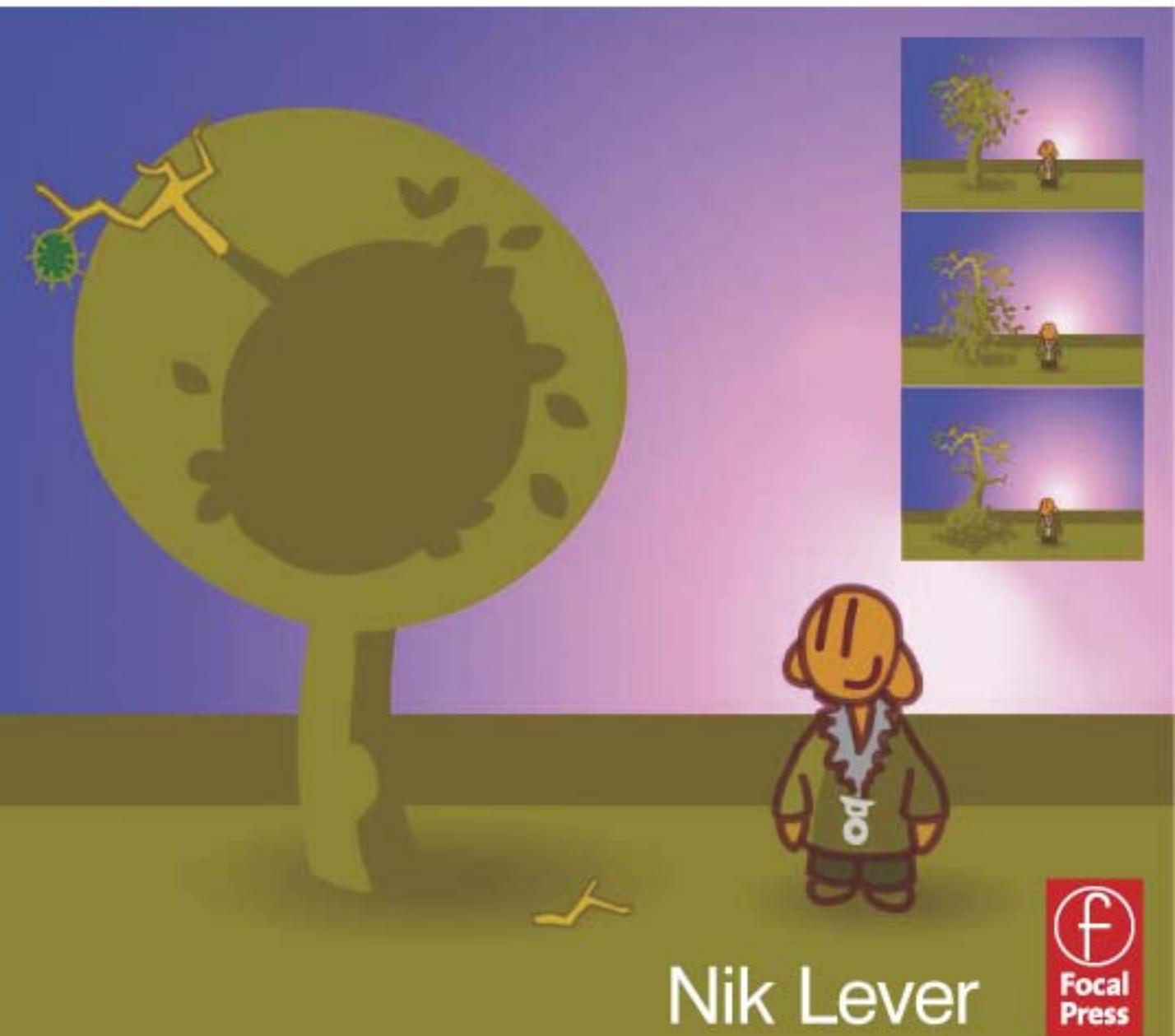




CD-ROM  
INCLUDED  
FOR  
PC & MAC

# FLASHMX 2004 Games

## Art to ActionScript



Nik Lever



# Flash MX 2004 Games

*Thanks to Pam*

# Flash MX 2004 Games

## Art to ActionScript

Nik Lever



AMSTERDAM • BOSTON • HEIDELBERG • LONDON • NEW YORK • OXFORD  
PARIS • SAN DIEGO • SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Focal Press is an imprint of Elsevier



Focal Press  
An imprint of Elsevier  
Linacre House, Jordan Hill, Oxford OX2 8DP  
200 Wheeler Road, Burlington, MA 01803

First published 2004

Copyright © 2004, Nik Lever. All rights reserved

The right of Nik Lever to be identified as the author of this work  
has been asserted in accordance with the Copyright, Designs and  
Patents Act 1988

No part of this publication may be reproduced in any material form (including  
photocopying or storing in any medium by electronic means and whether  
or not transiently or incidentally to some other use of this publication) without  
the written permission of the copyright holder except in accordance with the  
provisions of the Copyright, Designs and Patents Act 1988 or under the terms of  
a licence issued by the Copyright Licensing Agency Ltd, 90 Tottenham Court Road,  
London, England W1T 4LP. Applications for the copyright holder's written  
permission to reproduce any part of this publication should be addressed  
to the publisher

Permissions may be sought directly from Elsevier's Science & Technology Rights  
Department in Oxford, UK: phone: (+44) (0) 1865 843830; fax: (+44) (0) 1865 853333;  
e-mail: [permissions@elsevier.co.uk](mailto:permissions@elsevier.co.uk). You may also complete your request on-line via  
the Elsevier homepage ([www.elsevier.com](http://www.elsevier.com)), by selecting 'Customer Support'  
and then 'Obtaining Permissions'

**British Library Cataloguing in Publication Data**

A catalogue record for this book is available from the British Library

**Library of Congress Cataloguing in Publication Data**

A catalogue record for this book is available from the Library of Congress

ISBN 0 240 51963 9

For information on all Focal Press publications visit our website at: <a href="http://www.focalpress.com">www.focalpress.com</a>
---

Typeset by Newgen Imaging Systems (P) Ltd., Chennai, India

Printed and bound in Great Britain

# Contents at a glance

<i>Introduction</i>		xi
Chapter 1	Your first game	1
<b>Section 1</b>	<b>Animation</b>	<b>15</b>
Chapter 2	Drawing with Flash	17
Chapter 3	Simple cut-out animation	34
Chapter 4	Using CGI programs to create animation	46
Chapter 5	Background art	60
Chapter 6	Creating artwork for mobile devices	73
<b>Section 2</b>	<b>Action</b>	<b>81</b>
Chapter 7	So what is a variable?	83
Chapter 8	In tip-top condition	101
Chapter 9	Using loops	118
Chapter 10	Keep it modular	136
Chapter 11	Debugging	152
Chapter 12	Using external files	168
<b>Section 3</b>	<b>Putting it into practice</b>	<b>183</b>
Chapter 13	Small games to keep them loading	185
Chapter 14	Quizzes	205

## Contents

Chapter 15	Mazes	226
Chapter 16	Board games	244
Chapter 17	Platformers	260
Chapter 18	Sports simulations	282
<b>Section 4</b>	<b>Flash for mobiles</b>	<b>307</b>
Chapter 19	Flash Lite – how does it affect game developers?	309
Chapter 20	Using Flash on a PocketPC	318
<b>Section 5</b>	<b>Flash for boffins</b>	<b>327</b>
Chapter 21	High score tables	329
Chapter 22	Multi-player games using sockets	347
Chapter 23	Using Flash Communication Server	360
Chapter 24	Embedding Flash	375
Appendix A	Integrating Flash with Director	389
Appendix B	Tweening in code	395
Bibliography		411
<i>Index</i>		413

# Contents in summary

<b>Introduction</b>	<b>xi</b>
<b>Chapter 1: Your first game</b>	<b>1</b>
<p>Jumping headfirst into a tutorial-based ‘getting started’. In this chapter the reader will be guided through the process of creating a simple pairs game. In a tutorial format the user is taken through creating artwork in Flash and then adding some limited interactivity. The basics of using Flash buttons are presented along with keeping track of the board state in a way that the computer can understand.</p>	
<b>Section 1: Animation</b>	<b>15</b>
<p><i>The first section looks at creating the artwork and animation that will feature in the games we create.</i></p>	
<b>Chapter 2: Drawing with Flash</b>	<b>17</b>
<p>Flash is the perfect application to create line animation and in this chapter we look first at the drawing tools available and then move on to the principles of animation.</p>	
<b>Chapter 3: Simple cut-out animation</b>	<b>34</b>
<p>Flash is often used to display animation within a web browser. In such a context the size of the animation file is of vital importance to the viewers. In this chapter we will look at how you can create animation that looks fluid with only a small number of different pieces of artwork.</p>	
<b>Chapter 4: Using CGI programs to create animation</b>	<b>46</b>
<p>Computer animation packages can offer an excellent way to create the content for your games. In this chapter we look in general at the benefits and techniques involved.</p>	
<b>Chapter 5: Background art</b>	<b>60</b>
<p>Your animated characters will need to appear in some context. In this chapter we look at creating backgrounds to act as a setting for the action that takes place in your games.</p>	
<b>Chapter 6: Creating artwork for mobile devices</b>	<b>73</b>
<p>With the new Flash Lite player for mobile devices, you can create games that will play on a mobile phone. This chapter examines issues of size, frame rate and user input that influence the designer producing content for these memory-limited machines.</p>	

### **Section 2: Action** **81**

*Having developed the animation we now look at how to add interactivity to this art using *ActionScript*.*

### **Chapter 7: So what is a variable?** **83**

After the first blistering introduction to *ActionScript* in Chapter 1, we now take a leisurely tour through the basics of programming starting with variable types.

### **Chapter 8: In tip-top condition** **101**

In this chapter we look at using conditional code using the marvellous ‘if’ statement. We also look at how we can extend the flexibility of the condition using Boolean logic.

### **Chapter 9: Using loops** **118**

No programming language would be complete without the ability to repeat sections of code. Games regularly need to run a section of code several times. In this chapter we will look at creating sections of code that repeat.

### **Chapter 10: Keep it modular** **136**

Programming can be made very hard or very easy. Good structure makes it so much easier. In this chapter we examine the options.

### **Chapter 11: Debugging** **152**

Nobody is perfect and in this chapter we look at what to do when your code doesn’t work.

### **Chapter 12: Using external files** **168**

The game you create can be made to operate differently on different servers. Using ASP pages we can load variable data from a database. We look at linking to static files or generating dynamic data on the fly using server-based ASP pages.

### **Section 3: Putting it into practice** **183**

*You know your stuff, so how about using this knowledge to make some games?*

### **Chapter 13: Small games to keep them loading** **185**

If your game is over a meg then you will have lost your audience before they even see it. In this chapter we look at creating 20 K or less games to keep their interest while the remainder of the game loads.

### **Chapter 14: Quizzes** **205**

An interactive quiz where the questions are pulled in from a database forms the basis of this chapter. This develops the presentation of using external data first covered in Chapter 12.

- Chapter 15: Mazes** **226**  
Mazes can be generated dynamically or statically. The graphic view can be overhead, isometric or first person. In this chapter we look at the options and how to show the user when they are doing well or badly.
- Chapter 16: Board games** **244**  
A board game involves a legal move generator and a mini-max search routine for a good computer move. In this chapter we look at the strategies involved in presenting some classic board games in Flash form.
- Chapter 17: Platformers** **260**  
Creating a platform-based game is quite a challenge, but by this stage in the book the reader has all the skills necessary. In this chapter we look at the problems of game world and screen space calculations.
- Chapter 18: Sports simulations** **282**  
One effective way of getting input is to use the keyboard. In this chapter we look at creating games where the user has to press keys at a fast rate in order to score highly in sports simulations.
- Section 4: Flash for mobiles** **307**  
*The mobile games market is growing exponentially. With Flash MX 2004 developers have a new target platform, Flash Lite. This new section shows the limitations and possibilities of the new platform.*
- Chapter 19: Flash Lite – how does it affect game developers?** **309**  
Using the many new templates available in Flash MX 2004 Professional we can target lots of mobile devices. Flash Lite is a compromise between Flash 4 and 5; this chapter introduces you to how developing Flash Lite games differs from Flash games for desktop PCs.
- Chapter 20: Using Flash on a PocketPC** **318**  
The PocketPC is as near to a desktop as mobile devices get; as such, the content you can successfully handle includes most games you can develop for a desktop. We look at porting existing content to the PocketPC.
- Section 5: Flash for boffins** **327**  
*Sometimes you will need to extend what you can do in Flash using external methods. This last section introduces you to things you should be thinking about.*
- Chapter 21: High score tables** **329**  
High score tables encourage users to stay online in order to see their name appear in the high score list. The tables are usually stored in a database and then loaded into Flash when a call to a high score table is required. This chapter explores the options.

**Chapter 22: Multi-player games using sockets** **347**

Although multi-player games are possible using ASP pages, the best technique is to use a server-side listening program. In this chapter we look at creating a socket listening program using C++.

**Chapter 23: Using Flash Communication Server** **360**

The Flash Communication Server provides all the tools you will need to create multi-player games. A trial developer version is freely available; in this chapter we walk through an introduction to this important technology.

**Chapter 24: Embedding Flash** **375**

You can add Flash to a VB, C#, C++ or PocketPC program very easily. This chapter shows the options.

**Appendix A: Integrating Flash with Director** **389**

Director is starting to make a more significant impact on the Internet with the addition of Shock-wave 3D. Here you will find an introduction to controlling a 3D animation using a Flash interface in Director and how to package this for Internet distribution.

**Appendix B: Tweening in code** **395**

The easiest way to tween a graphic or a movie clip is to right-click on a frame layer in the timeline and chose 'Create Motion Tween'. Unfortunately the results are not always fantastic. In this appendix we look at how you can use ActionScript to provide dynamic tweening that gives a much smoother and more flexible result.

**Bibliography** **411**

**Index** **413**

# Introduction: Learn to write ActionScript and have fun doing it!

Flash MX 2004 provides the perfect platform to create fun games for Internet distribution. This book takes the reader through the entire process from creating the art and animation for these games, through to programming them ready for users across the world to enjoy the results. The book is split into five sections. The first section introduces the new user to the drawing tools in Flash and explains how to create the animation frames that you will need for your games while keeping the files bandwidth friendly. Section 2 takes the reader on a guided tour of using ActionScript. The emphasis is on explaining computer programming to readers from a design background; no previous computer programming is assumed. In Section 3 lots of practical examples are offered of specific games; all the source code is provided in Flash MX 2004 format on the included CD. The mobile games market is growing exponentially. With Flash MX 2004 developers have a new target platform: Flash Lite. Section 4 shows the limitations and possibilities on the new platform. Finally in Section 5 and the Appendices advanced topics are included about using sockets, databases and Flash Communication Server; there is even a section on embedding Flash into VB, C# and PocketPC programs.

## Using the CD

You can browse the folders on the CD. Each chapter and appendix has a unique folder in the 'Examples' folder. All the projects and source can be found here. To use a project file, open the file directly from the CD in Flash and then save it to your local hard drive if you intend to make changes. The 'Utilities' folder contains a small exe file for Windows machines that can create a MIME encoded query string from a text file.

## Who is the author?

It all started with a ZX81, a hobbyist computer released in Britain in 1981, which was soon replaced by the Sinclair Spectrum, an amazing computer produced by the same company. The Sinclair Spectrum boasted an amazing 48 K of memory, and an eight-colour display. From those early days the author was smitten, writing code became something of an obsession. Having graduated as a graphic designer in 1980 he had already started work as a professional animator when the interest in computers surfaced. All too soon, his computer hobby merged with professional life, the first example being a computer-controlled rostrum stand, a camera for filming 2D animation. Much more recently he has been producing CD-ROMs and web-based multimedia productions

for clients including Kellogg's, Coca-Cola, BBC, Cartoon Network and HIT Entertainment. The first interactive web-based work used Java, but the latest work all tends to be Flash or Shockwave 3D based. With compression and streaming abilities that are so web-friendly, Flash is set to be a standard for some time to come.

Check out [www.catalystpics.co.uk](http://www.catalystpics.co.uk) to see the author's handiwork or [www.niklever.net/flash](http://www.niklever.net/flash) for the web page associated with this book.

## Who is this book for?

### Designers

If you are a designer who has worked with Flash and wants to go further, you will find lots of interest you here. You will learn about applying your creative skills to the many stages of game production. If you have never played with Flash then fear not, you will be guided through the art, animation and programming involved in creating sophisticated games.

### Animators

Perhaps you have created the sprites for other people's games and always meant to look into how to create your own; in this book you will learn how. You will see how to use motion tweening in Flash to create smooth animation and then how to add interactive functionality to your game using ActionScript. Even if you have never written a line of code before then you will find the tutorial style easy to follow.

### Web developers

If you haven't started writing Flash code then shame on you, start today. Flash offers a sophisticated development environment with good tools for the production of both artwork and code. In this book you will learn how to use Flash to the best by creating animation and then adding interactivity, producing exciting and dynamic Internet content.

### Students

The Internet is rapidly turning into a rich source of employment for visually literate students. If your skills also include adding the code then you will be in demand. In this book you will learn all the skills necessary to create highly dynamic web content.

## So what are you waiting for?

Boot up your computer, go straight to Flash MX 2004 and turn the page, where you are about to create your first game in a Flash!

# 1 Your first game

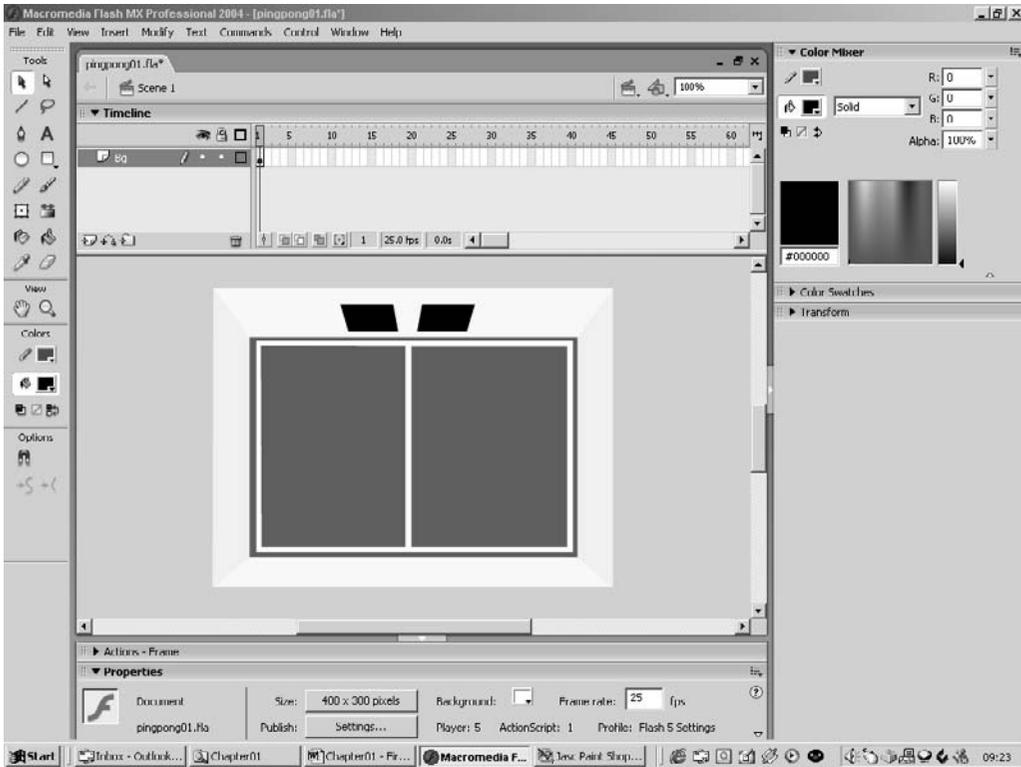
Rather alarmingly it is now over 20 years since I wrote my first bit of code using a Sinclair Spectrum computer and Sinclair Basic.

In 1980 in the UK, Clive Sinclair, an entrepreneurial inventor, advertised for sale a very simple computer. The ZX80 connected to the TV set and enjoyed 1 K of RAM. Each machine came with a simple programming language, BASIC (Beginners All-purpose Symbolic Instruction Code). This early machine started a revolution of young bedroom and garage programmers. Some of these early starters went on to become very successful in the games industry. Sinclair's inexpensive computers, including the Spectrum, are one of the main reasons why the UK has an internationally respected games industry.

Animation had for several years been my main motivation. I still get a buzz from making things move. Although initially unconnected, programming shares many similarities with animation. It is a creative pursuit. If you come to programming from an art background then you may find this doubtful. Surely *creative* implies the visual, written or musical arts. But programming is not a case of just one solution to a problem; there are many solutions and it is the path you take to the solution where the 'art' comes in. I got such a buzz from seeing my first programs work that I now spend rather more of my time writing code than creating animation artwork. In this chapter we are going to rush headlong into creating a game. Not a massive multi-player 3D first-person extravaganza, we may have to wait until day two for that! No, the game we will produce will be familiar to anyone who has ever played a computer game. It is a very simple version of 'Pong'. Along the way we will discover the Flash interface. You will learn how to create simple artwork with Flash, where to find all those panels and what they do. You will add a little code, test your work and do some simple debugging. So without further hesitation, let's get going. I strongly advise reading this chapter while using your computer. You will get much more from it if you enter the code and follow the tutorial throughout rather than just dissecting the final program.

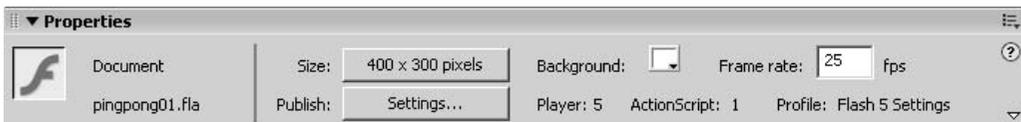
## Creating simple artwork using Flash

Figure 1.1 shows the basic Flash interface. The biggest area is used for the stage. This is where we put the imagery that the user will see. Above this in the figure is a grid; this is used as a timeline. In Flash a scene is broken down using time segments and layers. A single column in the upper section represents one time segment and a row represents a layer of graphics. So how long is a time segment? By default a time segment is one twelfth of a second. The user can alter the duration



**Figure 1.1** *The Flash MX 2004 interface*

of a time segment by clicking in an empty area of the stage and selecting the properties window. The window should look like that shown in Figure 1.2.



**Figure 1.2** *Altering movie properties*

The first thing to do when following this tutorial is to open the file ‘Examples/Chapter01/Pingpong01 fla’ from the CD. You must save the project to your local machine if you intend to edit and keep a project. You should be looking at something similar to the view shown in Figure 1.1. Here in addition to the timeline and main stage area you will see a panel on the left containing the important tools you will use when creating graphics with Flash. The graphics produced using Flash are vector based. That is, you define lines, curves and fills and Flash paints the screen using this information. So let’s look at creating our first graphic element. To help you, the example already includes the background for the game. The background shows an overhead

view of a sports hall. We are going to create a ball and two bats, then add some code to allow the user to move the bats, and finally add some code so that the ball bounces off the bats and walls.

## Creating new layers

The first stage of this tutorial chapter is creating the ball. When creating a game in Flash we can sometimes create everything in the same layer, but usually we will use extra layers for different elements. Look at Figure 1.3; here you can see that three layers are involved. To add a layer, click on the icon shown in the lower left corner of the figure. To delete a layer click on the trash can. Try adding a layer and deleting it now. When you have the feel for this add a layer and click on the layer name; it may be 'Layer 2' or another number. When you click the name it goes into renaming mode; now you can give the layer a useful name, for this tutorial call it 'Bats and Ball'. When there are only a few layers and you are just creating a game you may be tempted to take the default name. Fight this! Get into the habit of naming everything; if the project gets bigger then you will be glad you did. Also you may come back to a project months after it was completed and you will find it much easier to find your way around if everything has a useful name. If you are working in a team then you will enjoy working on someone else's project if the project is well organized with easily identified names; if everything is poorly structured then the project becomes a nightmare to maintain.

Take a look at the top left corner of Figure 1.3; here you will see that it says 'Scene 1'. In addition to breaking a project down into time segments and layers, Flash also allows you to divide it up using Scenes. When producing a game or any other computer program, structure can make a huge difference to how difficult the project is to produce and maintain. Flash allows the developer to use a very good structure, but equally you can also use a very poor structure. Another aim of this book is to help you understand how to create well-structured projects and to learn how to attack a complex problem so that the task becomes manageable.

Now that we have created a layer to store the ball artwork, make sure that this layer is highlighted before we move on to the next section.

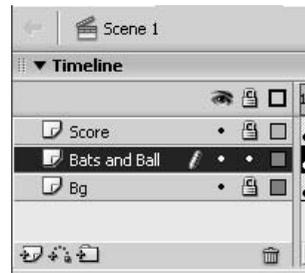


Figure 1.3 Adding and naming layers

## Creating the ball

Take a look at Figure 1.4; here the ring highlights the tool that you should select. Before we do any drawing in Flash let's briefly look at each of the tools in the Tools palette. Starting in the top left is the selection arrow, which allows you to select and edit a part of the artwork. The white arrow lets you edit a curve using the control handles that tell Flash how to draw the curve. The next row down has the Line tool for drawing lines and the Lasso tool for more complex selection. Moving down you will find the Pen and Text tools, the Oval and Rectangle tools, the Pencil and Brush tools,



**Figure 1.4** *Using the Oval tool*



**Figure 1.5** *Selecting the colour*

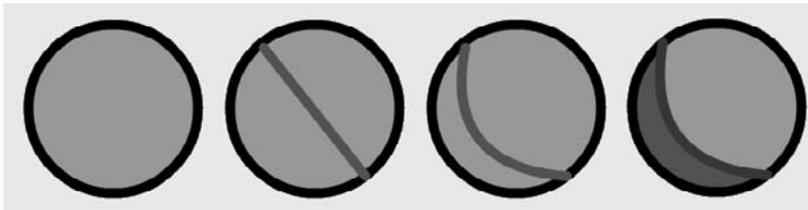
the Resize and Edit Fill tools, the Ink and Fill tools and finally the Eyedropper for picking up colours and the Eraser for deleting part of the artwork. If you have a tablet for drawing then you will find that some of the tools use the pressure sensitivity of the tablet to adjust line width. Recall that Flash is a vector-based drawing package; it distinguishes between lines and fills. A line has a consistent width that is set using the Properties panel for the Oval tool. When you have selected the Oval tool, take a look at the Colors palette (Figure 1.5). Notice that there are two colours, a colour for the lines you draw, which is to the immediate right of the pencil icon and a colour for the filled areas, to the right of the paint bucket icon. When you create a circle or oval it can include an outline, a fill or both. Click on the colour square and this brings up a further dialog box. In this box you can select from a pre-defined palette or you can use the mixer panel, 'Window/Design Panels/Color Mixer', to create a new colour. For the ball we are now going to create we will select both an outline colour and a fill colour; in the example we use black for the outline and a mid blue for the fill colour.

Before we draw our first element we need to select one further setting, the stroke width. With the Properties panel for the Oval tool visible set the stroke width to '1'. The type of stroke will be 'Solid'. Now we are ready to draw.



**Figure 1.6** *Setting the width of a line*

Figure 1.7 illustrates the four stages in drawing the ball. Stage 1 – on the stage area click to define the top left extent of the ball, then drag to the bottom right. This sets the size of the ball and creates the ball we see on the left of Figure 1.7. We are going to enhance this simple drawing by adding a shadow to the bottom left. To do this we will use the Line tool.



**Figure 1.7** *Stages in drawing the ball*

## Using the Line tool

Select the Line tool by clicking on the icon highlighted with a ring in Figure 1.8.

A line is initialized using a click. Click on the ball to define the start of the line and drag to the end. If the line is not quite in the right place, then select the arrow ‘Selection tool’ icon and click on the line. The line is highlighted; now press the delete key to remove this line.

By now you should have a ball with a line drawn through it. You can also use the arrow selector as an important editing tool. Move the mouse so that it is near one end of the line. Notice that the mouse cursor includes a corner at the bottom right. Now move the mouse so it is near the middle of the line, and you will see that the corner changes to a curve. Unlike most vector editing packages, Flash does not show editing handles on a line, unless you use the white ‘Subselection’ arrow. As a user you can determine what will happen when you edit a line based on the mouse cursor image. If you click and drag when the mouse cursor has a corner showing then you are moving one end of a line segment, whereas if the mouse cursor has a curve then you can stretch the curve without moving the endpoints by dragging the mouse. Try clicking and dragging near the middle of the line and you will see the curve bend. You may find that it is difficult to see any detail on the ball because it is too small. If so then try zooming in; remember that Flash is a vector package and as such you can zoom in without any loss of quality. Use the magnifying glass to select the zoom option, or select the level of zoom from the combo box in the top right corner of the application. When the Zoom tool is selected you can drag a box around your artwork and Flash will resize and centre your view of the artwork to this box.

At this stage your ball should look like the third ball from the left in Figure 1.7. To give the illusion of depth to the ball we will add a shadow of a darker colour to the bottom left of the ball. This area has a bounding shape defined by the circle and the line we have just drawn, so you can use the Fill tool shown highlighted in Figure 1.11. Use the mixer panel to create a darker blue for the bottom left and then click in this area to apply the colour.

At this stage you could choose to remove the outlines. Figure 1.12 shows the completed ball with all the lines and fills selected. The dots over the artwork indicate a selection. If you click on



Figure 1.8 Using the Line tool



Figure 1.9 Using the Selection tool



Figure 1.10 Zooming in

a frame layer in the timeline then everything in that frame layer will be highlighted. To select just a section of the image click away from the artwork to deselect, then click on the part you wish to select. To remove the outlines you would click on any outline in the ball, then double-click to select all the outlines of the same colour and stroke connected to the first selection. Try using firstly the timeline to select and then the arrow selection tool, and try selecting individual and connected parts. It is important when developing artwork in Flash that you are familiar with the selection tools. Finally highlight the completed ball; the easiest way to do this is by using the timeline selection method. Your selection should look like Figure 1.12.

## Using symbols

Flash lets you turn pieces of artwork into symbols. With the whole of the ball selected choose 'Modify/Convert to Symbol...' or press F8. This brings up the dialog box you can see in Figure 1.13.

When creating a symbol you have three choices. The symbol can be a 'Movie Clip' in which case you will have control over it using ActionScript. It can be a 'Button' so that it will respond to the position of the mouse or it can be a 'Graphic'. For now we will choose 'Movie Clip' and give it the name Ball. Having clicked 'OK', the highlighting will change from that shown in Figure 1.12 to a rectangular bounding box drawn around the symbol. Although you can still move and resize the symbol you cannot edit it using the drawing tools. To get at the actual artwork you will need to edit the symbol itself. You can get at the symbol by any one of three ways: either open the 'Library' window using 'Window/Library' or right-click on the symbol and then choose the option to 'Edit in Place'. This will change the timeline so that you are now editing 'Scene1/Ball' rather than 'Scene 1'. The final option for choosing to edit a symbol is to use the 'Edit Symbols' button at the top right of the stage window, as in Figure 1.14.



Figure 1.11 Using the Fill tool

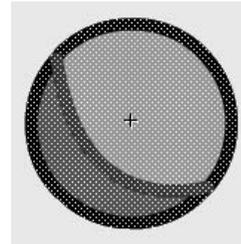


Figure 1.12 The ball selected

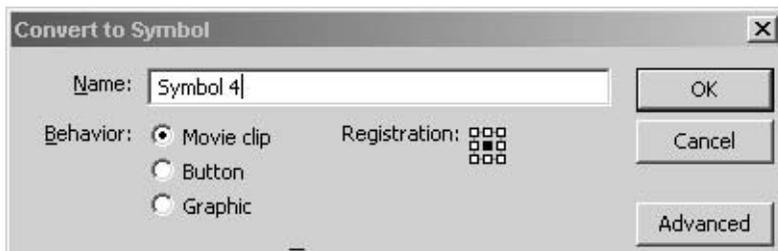


Figure 1.13 Creating a symbol