

Written by Billy D. Spelchan for www.BlazingGames.com

Copyright © 2003-2005 Blazing Games Inc. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the file called fdl.txt

Chapter 3

Introduction to Flash

Contents

In this chapter we take a tour of Flash and it's animation capabilities.

- What is Flash - A look at what Macromedia Flash is.
- Java vs. Flash - A comparison between Flash and Java.
- Drawing in Flash - Basic drawing in Flash.
- Symbols - What Symbols are.
- Symbols Creation Tips - Tips on Creating Symbols.
- Keyframe Animation - Basics of Animating in Flash.
- Streaming - A look at one of Flash's nicest features.
- Spelchan part of logo - We start building a logo.
- Software part of logo - Second part of logo.
- Presents part of logo - Third part of logo.
- Fading out logo - Finishing logo.
- Transitions - Ideas for creating transitions for your own sequences.

What is Flash

At the heart of Flash is a very powerful vector animation system. If you are not familiar with graphics terminology, the term Vector Animation may not be familiar to you. Vector graphics are images that are made by combining mathematically defined shapes (made up of lines and curves and optionally filled with color). This has the advantage that you can zoom in or out of an image without any loss in quality. Another less noticeable but very important advantage is that simple vector images take up a lot less room than the equivalent bitmap image.

Bitmap images are made up of large numbers of colored dots. This is the type of image that is normally displayed on your monitor. The power from bitmaps comes from its ability to handle complex images, such as photographic images. Flash supports this type of image as well, though bitmap based images take up much more space so if you are designing a flash animation for use over the internet you will want to go easy on the number of bitmap images that you use in your animation.

Flash streaming capability is designed for the internet. What streaming does is it allows the animation to play while it is actually being downloaded from the internet. In some cases, such as with more complex games, you may need most or all of the game loaded before the game can be started. Flash allows you to get by this limitation by having a loading animation that can play while the rest of the animation is loaded.

What makes flash powerful enough to allow for the creation of games is the scripting language that Flash supports. This language is called Action Script and is based on the Java Script scripting language. As Action Script can be scary to non-programmers, I will try to slowly introduce the elements of it that are needed to develop games.

Flash is owned by Macromedia. They have freely available flash players for most Operating Systems and Browsers. Because they own Flash, the language runs fairly consistently on any platform that they support. Unfortunately, this means that they are a single source provider of the Flash standard which has disadvantages, which I will not get into.

Java vs. Flash

Having developed games in both Java and in Macromedia's Flash, some of you are probably wondering which of the two I prefer. In all honesty, I would have to say that I like both of them but for different reasons.

Java is a very powerful programming language and its object oriented structure and very well designed package system makes Java ideal for very large projects. With the ability to write Java applications, which can theoretically have full access to the computers resources, it is possible to create pretty much anything.

Flash is more of an animation system with a scripting language tacked on. As far as animation goes, it is far easier to do animated sequences and heavily animated games in Flash than Java, but the programming capabilities of Flash tend to limit what you can do.

Ideally, I would like to have a Java program that could run Flash animations. While that is theoretically possible, the solutions in place for doing so are not that ideal. Still, if being able to easily add Flash to Java becomes more acceptable, then that is certainly something I will be doing.

Drawing in Flash

Flash, as mentioned earlier, is vector based. All drawing in Flash is built around shapes. Shapes are made up of a series of connected lines and curves. You create the shapes by using basic building blocks such as lines, rectangles and ovals. You also have text, which can be treated as text or converted into a shape so that you can modify the shape.

For more complex shapes you can use a pencil tool. The pencil tool takes whatever shape you draw and turns it into a series of lines and curves. You can control how close to what you draw the pencil will be, with the more accurate the representation, the more anchor points in the object.

An anchor point is a point that defines the shape of the object. The more anchor points, the more memory is required to represent a shape. Anchor points connect to other anchor points and can either be connected as a line or a curve. Curves can be controlled by adjusting the tangent line for that line segment. You can adjust the anchor points and tangent lines by using the sub-selection tool. You can also use the pen tool which lets you place and adjust anchor points as you build a shape.

Shapes can have an outline and a fill, or can be just one of the two. The outline has a color, a thickness, and a pattern associated with it. One of the most unique thicknesses is the hairline. This is a special thickness as it will always be drawn as thinly as possible.

The fill of a shape can be a solid color, a gradient, or a bitmap fill. Gradients are a series of colors that gradually change from one color to the next color in the series. You may have as many colors in a series as you desire and can vary where in the gradient range the colors will change. Gradients can be linear or radial and the orientation, scale, and center spot can be adjusted. A bitmap fill is a bitmap image that is used to fill the shape. Bitmap fills are tiled, though you can adjust the orientation, scale, skew, and center spot.

Symbols

Flash revolves around symbols. Symbols are a very efficient way of building movies, as a symbol only needs to be loaded once. After it has been loaded, you can create as many copies - known as instances - as you want to. More importantly, every instance of a symbol can have properties independently applied to it. You can adjust its size, orientation, and skew. In addition you can apply tinting, control the brightness, adjust the alpha (transparency) level. There are three basic types of symbols that Flash uses. The Graphic symbol, the Movie symbol and the Button symbol.

Graphic symbols are just a drawing converted into a symbol. The drawing can be as simple or complex as you desire, and can even contain other symbols as part of it. A movie symbol is essentially a Flash movie within a Flash movie. You have as much control over a Flash movie symbol as you do over a Flash movie. And, of course, Flash movies can contain other Flash movies as part of them.

A Button is simply a special symbol that has special actions associated with it whenever the mouse is over it or the mouse has been clicked while over it. To be more precise, a button has 4 frames associated with it. These frames have special significance.

The up frame is simply the normal appearance of the button. The over frame is how the button looks when an object is over it. The down frame is what the button looks like when the mouse is over it and the mouse button has been pressed. Finally, the hit frame defines the over/hit areas.

The hit frame may seem confusing to people new to Flash. This frame is not actually seen by the viewer, but is instead used by Flash. How it works is any area in the hit frame that is solid will react to the mouse being over it. Likewise an area that is not solid will not react when the mouse is over it.

Some of you may be wondering why you would want or need such a thing? Couldn't you just use the existing button image? The answer to that is you could, but then the button would only work if the user had his or her mouse positioned in a solid part of the image. In some cases, such as with our Play the Game button, we want the button to react if the mouse is within a block or an area that covers more than the frames cover.

Well then, you may ask, why not just have Flash use the rectangular bounds of the button as its click area. This would work great for our Play the Game button, but what if we wanted to have a round or oval button? What if we had an interactive image where the person had to click on parts to bring up information? In both these cases having rectangular buttons would be a hindrance.

Flash is able to provide the best of both situations by having the hit frame. In fact, the hit frame also allows for decorated buttons where only a small portion of the button is clickable. Because of the hit frame, Flash's button handling is extremely flexible, which is always nice.

Symbol Creation Tips

When creating an object for use in a Flash animation, almost anything is possible. In fact, you can even use bitmap images created with a paint program if you so desired. It should, however, be pointed out that if your animation is going to be viewed over the internet, then keeping the image small is probably a very high priority.

The storage size of a vector image is based on how many points make up the image, not on how big the image is. The sub selection tool can be used to see how many points an image requires by clicking it. You can also use this tool to help eliminate redundant points.

If you need a lot of detail in an image, a small bitmap can be used as a texture. Complex gradients can also be created. Bitmap images can be used in multiple objects, so you may be able to re-use a texture. In fact, by modifying the scale, orientation and positioning of a bitmap you can even make it look like different objects are using different textures even though they are all using the same bitmap as a texture.

Base the amount of detail an image has on the largest size of an image. Also take into consideration how fast an object is moving. If the object is going to be small or moving fast then you don't need very much detail as most people won't see the details anyway.

Finally, Remember that an object can be made up of other object, and that object can be a flipped, rotated, scaled, and skewed. Symmetrical objects can be created in halves or quarters, for instance.

Keyframe Animation

Flash uses a layer based keyframe animation system. For people who are not familiar with the creation of animation, this sounds complicated. In reality it is a very quick and effective way of creating animation.

A keyframe is essentially a frame of the movie. Within the frame you place objects where you want them to be. Every time you want to change the contents of the frame you create a new keyframe. This, in itself, is not that powerful. Where the power comes in is with tweening. Tweening is a method where you let the computer animate the object for you. Flash has two types of tweening. Motion Tweening and shape tweening.

Motion tweening moves an object from the location it is on the starting keyframe to the location it is when it is on the ending frame. You can also apply rotation to the object that is moving. While technically not motion, you can also adjust any color adjustments (such as the alpha level) and the adjustment will smoothly transition between the frames. As objects in nature don't move at steady rates, you can use leading to have the object start the motion at a faster rate and slow down as it approaches the end keyframe or have it start slow and speed up as it reaches the keyframe.

For even more advanced animation effects, you can even have guides. Guides let you specify the path that a moving object will follow. You can have the object follow the path while maintaining the same orientation or you can have the object orient itself to the path.

Shape tweening is a bit complicated. Shape tweening is when you have one shape and it transforms itself into the second shape. This is easy to do, but hard to do well. You simply need two shapes, a starting shape on the start keyframe, and an ending shape on it's ending keyframe. To make sure the shape morphs the way you want, you add key points to the shapes.

A Flash movie consists of one or more layers. Layers go from back to front with closer layers overlapping further layers. Every layer is independent of the other layers. In other words, you can have keyframes in one layer but not any of the other layers. Flash lets you define as many layers as you need and also lets you group layers into folders. It is advisable to keep each animated object on it's own layer.

Streaming

One feature of Flash that I particularly like is Flash's streaming capability. Streaming is when a movie starts playing while it is still being loaded. This is nice for the viewer as they do not have to wait for the movie to finish downloading before starting to watch it so they are not wasting their time. The problem is that some movies, and almost all games, are too complex to properly stream and require pre-loading some things, or even require the entire movie to be in memory before the game can be played.

Just because streaming can not be fully utilized, that does not mean it is not useful. In fact, many of the Flash games that I have seen have not really utilized the streaming capabilities of flash as much as they could have. Far too often I have seen a please wait splash screen as the flash movie loads.

At a minimum, you should have a loading bar, which is starting to be used more frequently. A better approach, which happens to be what I do, is to have a logo and animated title sequence. By making these sequences as small as possible. This allows the logo and titles to be loaded fairly quickly and while they are playing, the rest of the movie is loaded. This makes the waiting a bit more tolerable.

Another approach that I have seen, and I think would be useful if the movie was particularly large, is what I call the entertainment loading bar. This is like the boring load bars, but have some type of entertaining sequence that is played when the bar reaches key points. The mini-movie sequences keep the viewer entertained while the movie is being loaded. For a game, one thing that could be done would be to have game play sequences as the preview movie. This would have the added advantage of not requiring new resources for the entertainment sequences.

Spelchan part of logo

As my company is named Spelchan Software, it only makes sense that the logo be a Spelchan Software logo. The following will give people new to Flash an idea of what can be done and may also give readers some ideas on how to create their own small logo.

Spelchan part of Logo

We will start the logo with the word "Spelchan". If you have played the released version of this game you know that the letters forming the word "Spelchan" come one at a time from off screen zooming from close to the user to their final position on the screen. The following screen shot shows the final position of the letters, with the n just starting to move into it's final position.



Figure 1

This effect is very easy to do. To start you use the font tool to type in the text that you are going to use in the font that you wish to use. You then break apart the text twice. The first time you break the text apart it breaks the text into separate letters. The second time it turns the individual letters into objects. Now use the distribute to layers command which places the individual letters into their own layers.

Now, convert all the letters into symbols. Figure out how long you want the sequence to take. For each letter create a key frame on the time position when the letter should reach it's ending point. Now move the starting frame to the point where that letter should first appear. Next move the position of the letter in the starting frame to the area outside the screen you want it to originate from. Scale the starting frame, click on any frame between the starting and ending frame and use the create motion tween.

Software part of logo

The "Software" part of my logo has the letters sliding in from the left and stopping in the proper positions.



Figure 2

Like the first part of the logo this effect is a very easy. To start you use the font tool to type in the text that you are going to use in the font that you wish to use. You then break apart the text twice. The first time you break the text apart it breaks the text into separate letters. The second time it turns the individual letters into objects. Now use the distribute to layers command which places the individual letters into their own layers.

Now, convert all the letters into symbols. Figure out how long you want the sequence to take. For each letter create a key frame on the time position when the letter should reach it's ending point. Now move the starting frame to the point where that letter should first appear. Next move the position of the letter to the area outside the screen you want it to originate from. Click on any frame between the starting and ending frame and use the create motion tween. You may also want to adjust the easing of the frame. Easing is a way of having the object that is moving speed up or slow down as it is moving.

Presents part of logo

The presents part of the logo has the last four letters of presents moving in from the left side of the screen moving into it's proper position and the first four letters of presents moving in from the right side of the screen into it's proper position.



Figure 3

This was created by taking the word "pres" and the word "ents", moving them to their final position on the screen, each in their own layer. Creating a key frame. Moving the "pres" off to the right and the "ents" off to the left in their respective starting key frames. Motion tweening is then added.

Fading out logo

Now we finish the logo by fading it to a solid color. This is a very fancy effect that is very simple to do. All that is needed is a solid filled rectangle that is the same color as the color you want to fade to. You then create a top layer that has the rectangle appear in the frame that you want the fade effect to start in. Create a key frame on the last frame of the logo. You then create a motion tween between the two frames. Using the effect property to select an alpha level. The alpha level is essentially a level of transparency. Have the starting frame have an alpha value of 0 and the ending frame have an alpha value of 100.

Transitions

Fading out is not the only transition that can be used. There are many effects that can be used for transitioning from one scene to another. I chose a fading transition as it is a fairly decent looking effect and can be put into the game very quickly. If you watch television or movies, I am sure that you will see a huge number of transition effects. Lets quickly take a look at a few of the effects that can be done.

Sweeps are a very simple effect to build. Horizontal and vertical sweeps can easily be accomplished by having a solid block move across the field. Other shapes could be used as well. By taking advantage of Flash's mask layer and the ability of the mask layer to be a movie, pretty much any type of sweep is possible.

If you really want to get fancy with transitions, you could create an object that contains everything in the last frame. You could then have the end sequence consist of a single used layer which would be this full image object. This object could then be moved off screen, shrunk, or even spun away.