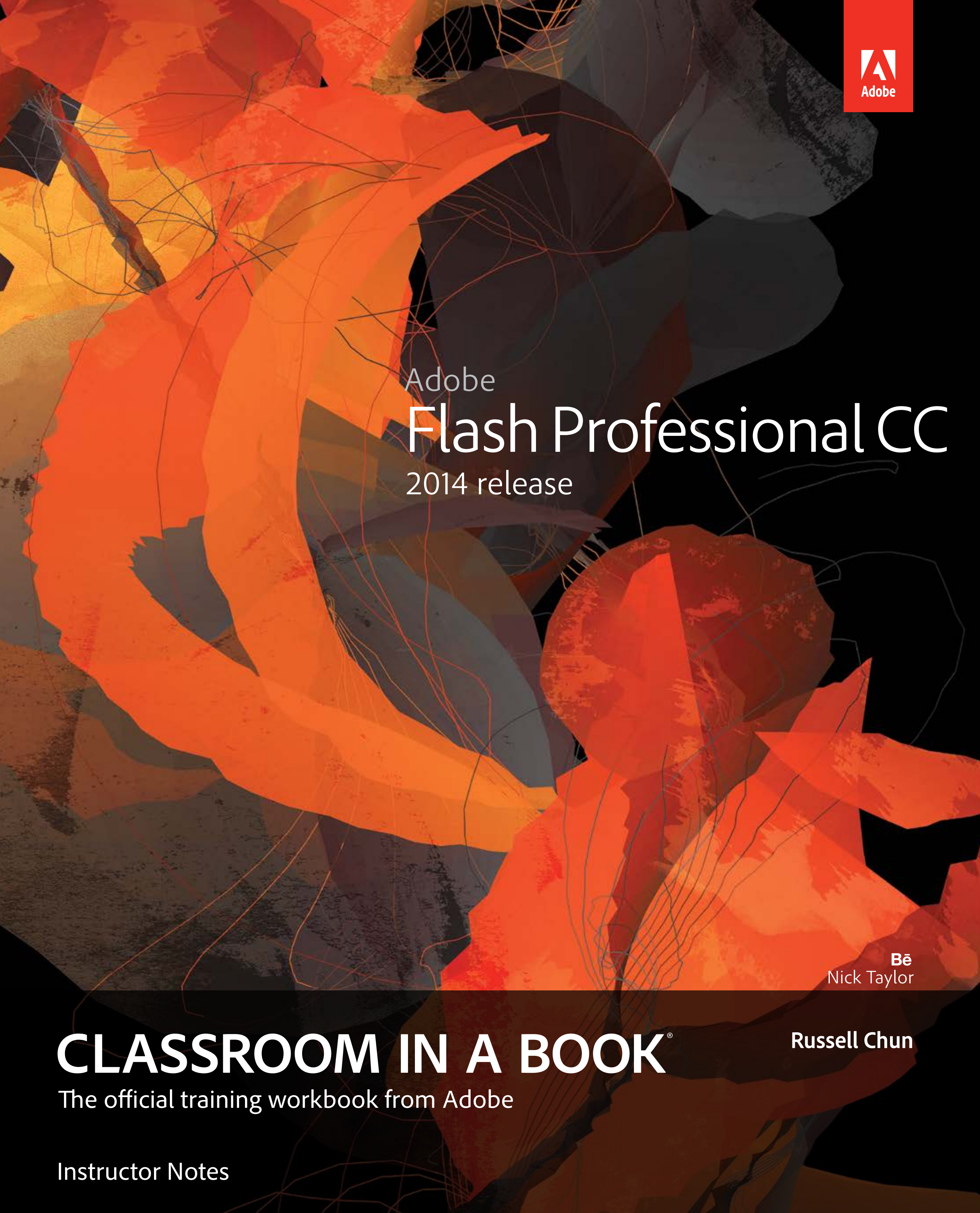

Adobe Flash Professional CC

2014 release

Bē
Nick Taylor

CLASSROOM IN A BOOK®

The official training workbook from Adobe

Russell Chun

Instructor Notes

Adobe® Flash® Professional CC Classroom in a Book® (2014 release)

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110-2704, USA

Adobe Press books are published by Peachpit, a division of Pearson Education located in San Francisco, California. For the latest on Adobe Press books, go to www.adobepress.com. To report errors, please send a note to errata@peachpit.com. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

Book:
ISBN-13: 978-0-133-92710-8
ISBN-10: 0-133-92710-5

Instructor Notes:
ISBN-13: 978-0-134-03520-8
ISBN-10: 0-134-03520-8

INSTRUCTOR NOTES

Getting Started

The *Adobe Flash Professional CC Classroom in a Book (2014 release)* presents students with a series of real-world projects to learn techniques, solutions, and tips using the 2014 release of Adobe Flash Professional CC software. The instructor notes are intended to complement the information in the *Adobe Flash Professional CC Classroom in a Book (2014 release)*. The lessons in this book require assets, including Flash files that are available for download from the Lesson & Update Files tab on your Account page at www.peachpit.com. Before beginning to work with the lessons, copy the files to each student's computer's hard drive as described in the Getting Started section of the book.

You may want to use all the lessons in this book, or just particular lessons. You can use the lessons in any order, but some tools and techniques are described in detail only in the first lesson in which they occur.

Introduce students to additional resources, such as Adobe Help, Adobe TV, Adobe Design Center, and Adobe Developer Connection, where they can access information such as tutorials, updates, and technical help provided by other Flash users and by Adobe.

Course Strategy

This book contains ten lessons, so if you're teaching a 15-session course, you can spend extra time on some of the more difficult topics and time-consuming lessons. The lessons that deserve at least two class sessions are Lesson 2 on creating graphics, Lessons 4 and 5 on animation, and Lesson 7 on interactive navigation. Depending on the focus of your course, you can spend the remaining extra session on Lesson 6, which covers shape tweening and masking; or Lesson 9, which covers video and sound. If you want to cover new publishing options, such as exporting content to HTML5 and JavaScript, or publishing AIR apps for mobile devices, you can spend more time on Lessons 10.

For beginners, the lessons that should be covered, at an absolute minimum, are Lesson 1, which acquaints the students with the interface and the concept of the Timeline and Lesson 4, which covers motion tweening.

Enhancements

Flash Professional CC (2014 release) adds many powerful new features for more expressive design, better control over complex motion tweening, and wider support for a variety of playback platforms and devices. The *Flash Professional CC Classroom in a Book (2014 release)* integrates many of the enhancements within the lessons. Some of the new features of Flash Professional include:

- A new drawing tool, the Variable Width tool, which allows you to create beautiful and expressive thick and thin strokes.
- A new and improved Motion Editor panel, which provides precision control over how the properties of a motion tween change over time. The Motion Editor panel also includes a variety of sophisticated eases to create more natural motion.
- A restored Object-level undo option
- Support for SVG (Scaleable Vector Graphics) export
- Separate Flash documents that target different playback platforms: HTML5 Canvas, WebGL, ActionScript 3.0 for Flash Player, AIR for mobile, and AIR for desktop.
- Integration with Kuler panel, a cloud-based application to create color themes using an iPhone or desktop browser.
- New publishing option for projectors for either Macintosh or Windows desktops
- HTML5 Canvas Javascript code snippets, making it easier than ever to create interactivity for HTML5 projects.

Lesson 1: Getting Acquainted

Lesson Overview

Lesson 1 uses a sample project to provide an overview of the Flash workspace. Students are introduced to the Tools panel, Library panel, Properties inspector, and Timeline. They also learn about layers and keyframes.

Starting Flash and opening a file

Demonstrate two ways of opening a document in Flash: by double-clicking the Flash document or by starting Flash, choosing File > Open. Let your students know that a Flash file can either have the extension, FLA or XFL. The XFL document is a newer Flash format (requiring CS5 or later), which is contained within a folder of other related assets. Explain the difference between a published project and the source file. The FLA file is the editable source document while the published project are the files necessary to play the movie in the target environment.

In an ActionScript 3.0 document, the SWF file is the published file. In an HTML5 Canvas or a WebGL document, HTML and JavaScript documents are the published files. In an AIR for desktop, an AIR (Adobe Integrated Runtime) installer or application is the final published file. In an AIR for mobile document, a mobile app is the final published file.

Emphasize the importance of choosing the correct Flash document type based on the intended target platform.

Getting to know the work area

Identify and describe the following areas of the workspace: Timeline, Tools panel, Stage, Library panel, Properties inspector, and the Edit bar above the Stage. Explain to students how they can undock and move the panels around to suit their working style, and how to return to the default layout by choosing Window > Workspace > Essentials.

If you have other Adobe applications installed, you can show students the similarities between panels in Flash Professional CC (2014 release) and panels in other Adobe Creative Cloud applications. The Timeline is similar to the Timeline in Adobe Premiere Pro and Adobe After Effects. The Properties inspector is similar to the one in Dreamweaver. Many commands work across applications, which will ease the anxiety of learning a new program. For example, copy and paste, or holding down the Shift key to select multiple items behave the same in Flash.

Understanding the Timeline

Layers overlap each other in the order in which they appear in the Timeline, with the top layer overlapping the bottom layers. As designers plan projects, they should consider the stacking order of the layers and need to be aware of how the objects in different layers appear in front of or behind objects in other layers.

Invite students to move the playhead through (or scrub) the Timeline to see the animation. As you move the playhead, Flash displays the contents of the frame on the Stage. Hide and show different layers on frames to demonstrate how layers are stacked and that each frame may contain content from multiple layers.

Keyframes are essential in animation. A keyframe indicates a change, whether it be in the position or other qualities of something on the Stage or the beginning or end of an audio file. Students will use keyframes in every Flash project, so it's important that they understand the keyframe's purpose clearly.

Describe the difference between inserting a keyframe and inserting a blank keyframe. When you insert a keyframe, the keyframe contains the frame's original content until you change it. When you insert a blank keyframe, all content is removed from the frame so that you can add new content. Also, describe the difference between a frame and a keyframe. A frame is a measure of time on the Timeline. A keyframe is represented on the Timeline with a circle and indicates a change in content on the Stage.

Using the Properties inspector

Demonstrate that the options in the Properties inspector change to reflect what's currently selected. The Properties inspector can display properties for the entire document, the selected frame, the selection on the Stage, or the selected tool in the Tools panel.

Using the Tools panel

The Tools panel contains several tools for selecting, drawing, editing, and navigating. Point out that the main tool for selecting and moving objects is the Selection tool. Demonstrate that some tools are grouped together, and a tool may be hidden beneath another. To select a hidden tool, click the triangle in the icon of the displayed tool, and then select the hidden tool from the pop-up menu.

Beneath the tools is the tools options area. The options available depend on the selected tool. Demonstrate how the options area changes as you select different tools. When you select the Rectangle tool, the Object Drawing mode icon appears; when you select the Zoom tool, the Enlarge and Reduce icons are available; when you select the Pencil tool, the Pencil mode icons are available. Show students that additional options for a selected tool are also available in the Properties inspector.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

1 What is the difference between a *.fla file, the *xfl file, and a *.swf file, and how do you create them?

2 What do you do when you can't find a panel that you know exists?

3 How does Flash measure its X and Y coordinates on the Stage?

4 How do you change the dimensions of the Stage and keep the contents proportional?

5 If your panels have been moved out of their normal positions, how do you put them back to their default positions?

6 How do you change the appearance of the Timeline?

Answers

1 The *.fla and the *.xfl are both editable, source files for a Flash project. When you save a Flash project, you can save it as a *.fla document or as an uncompressed *.xfl document, which exposes the contents of the Flash file for other developers. The *.swf is the exported file that is generated when you publish an ActionScript 3.0 Flash document. The *.swf is the file that you upload to your Web server to play in the Flash Player in a browser.

2 You can access all the panels by choosing Window from the top menu.

3 The X and Y values are measured on the Stage from the top-left corner. X begins at 0 and increases to the right, and Y begins at 0 and increases downward.

4 In the Properties inspector, click the Edit button next to Size. In the Document Settings dialog box that appears, enter new pixel values in the Stage size fields and choose Scale content. The Anchor option lets you choose the origin from which your content is resized, if the proportions of the new Stage are different.

5 Choose Window > Workspaces > Reset Essentials to reset the Essentials workspace, which is the default arrangement of panels.

6 When you want to see more layers, select Short from the Frame View pop-up menu in the upper-right corner of the Timeline. The Short option decreases the height of frame cell rows. The Preview and Preview in Context options display thumbnail versions of the contents of your keyframes in the Timeline. You can also change the width of the frame cells by selecting Tiny, Small, Normal, Medium, or Large.

Lesson 2: Working with Graphics

Lesson Overview

Lesson 2 uses a sample project to introduce students to the drawing tools and to creating simple graphics. The project is a static banner ad showing a mug of steaming coffee over a wavy background with pieces of text that are aligned to each other. As they create the banner, students learn how to use the various drawing, editing, and selection tools in the Tools panel. They learn to create basic shapes, import a bitmap fill, make selections, group objects, align objects, create gradients and transparencies, create curves, and manipulate objects with the Free Transform tool. They also learn the difference between drawing modes in Flash. Be sure to introduce the Convert to Bitmap feature, which can be helpful to simplify artwork and reduce the performance load on mobile devices.

Flash drawing modes

By default, Flash uses the merge drawing mode when you draw rectangles, ovals, or lines. In merge drawing mode, the drawn elements are called shapes. Overlapping shapes merge together to form a single element on the Stage. When you select a shape, the selected area appears dotted. Though the elements merge together, you can select each separately. Each side of a rectangle's stroke is a separate element, for example. To select a shape's fill and stroke, drag the Selection tool around the entire shape. Demonstrate to students how to select the fill or the stroke of a shape by clicking on it, or to select the entire shape (the fill and the stroke) by double-clicking on it.

In object drawing mode, active when the Object Drawing icon is selected in the options area of the Tools panel, rectangles, ovals, and lines are discrete elements, even when they overlap. The discrete elements are called drawing objects. When you select a drawing object, its fill and stroke are both selected.

In primitive drawing mode, active when you use the Rectangle Primitive or Oval Primitive drawing tools, drawn elements are discrete objects, but you can modify their corner angles or inner radius.

Merge drawing mode offers advantages in creating unusual, complex shapes. Object drawing mode lets you move objects without concern that they will overlap and clip other objects. Primitive drawing mode makes it possible to draw more complex shapes without having to merge multiple shapes together.

Invite students to draw and manipulate elements using each of these drawing modes so they become familiar with the differences and can recognize them as they work with them. You can convert a shape to an object by selecting it and choosing Modify > Combine Object > Union. You can convert an object to a shape by selecting it and choosing Modify > Break Apart.

Breaking apart and grouping objects

Emphasize for students the difference between shapes and groups in Flash, and the advantage of groups when you want to align certain elements of a drawing with the Align panel. Demonstrate how you can have groups within groups, and mention that you can always break apart groups to their component parts.

Creating transparencies

Transparency, or inversely, opacity, is determined by the alpha value of a color. Decreasing the alpha value decreases the opacity and increases the transparency. Alpha values range from 0 (totally transparent) to 100 (totally opaque). Students can change the alpha value for a color in the Color panel or in the Tools panel, under Fill Color options, in the Alpha field.

Creating gradients

In a gradient, one color gradually changes into another. Flash can create linear gradients, which change color horizontally, vertically, or diagonally; or radial gradients, which change color moving outward from a central focal point. Encourage students to apply a radial gradient and a linear gradient in succession to the same shape to see how the gradients differ. Students can add color pointers beneath the gradient definition bar in the Color panel to add up to 15 color transitions. In addition to choosing colors and positioning the color pointers for a gradient, demonstrate how you can adjust the size, direction, or center of a gradient fill with the Gradient Transform tool.

Variable width strokes

The Variable Width tool is a new addition to Flash Professional CC (2014 release). The Variable Width tool allows students to easily create and edit thick and thin strokes. Show students the Width option in the Property inspector when creating strokes, and the different width profiles that Flash provides. Select the Variable Width tool and hover over any stroke. Click to insert an anchor point and drag out the handles to change the thickness of the stroke at that point.

Creating and editing text

Creating text in Flash is similar to creating text in Adobe InDesign, Adobe Photoshop, or numerous other applications. Use the Text tool to create an insertion point on the Stage and begin typing, or drag the Text tool to create a text box for your text. In this lesson, students use Static text, which is used for text used for display purposes only. Input and Dynamic text, the two other forms of text, are used when you want to control the text with ActionScript. To format text, select it with the Text tool, and then select formatting options in the Properties inspector. To manipulate a text box as an object, select it with the Selection tool and make changes in the Properties inspector or Transform panel, or simply drag the text box handles to resize it. Resizing the text box does not resize the text, but it does reflow the text in the text box.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

1 How do you create a corner point when editing the contours of a shape with the Selection tool?

2 What are two ways you can change the fill of a shape?

3 What's the difference between a linear gradient and a radial gradient?

4 What is the function of the Lock Fill option for the Paint Bucket tool?

Answers

1 Hold down the Alt (Windows) or Option (Mac) key while dragging the sides of a shape to add a new corner.

2 One way you can change the fill of a shape is to choose the Paint Bucket tool, then choose a new Fill color in the Properties inspector, and then click on the shape. The other way is to select the fill on the Stage, and then choose a new color in the Properties inspector.

3 A linear gradient changes color horizontally, vertically, or diagonally, whereas a radial gradient changes color moving outward from a central focal point.

4 The Lock Fill option locks the current gradient to the first shape where it was applied so that subsequent shapes extend the gradient.

Lesson 3: Creating and Editing Symbols

Lesson Overview

Lesson 3 uses a sample project to introduce symbols, instances, and imported Illustrator and Photoshop files as students create a static frame of a cartoon. Students learn to create new symbols and to convert selections on the Stage to symbols. Students learn how to edit the symbols from the Library panel and to edit the symbols in place on the Stage. Students also learn to adjust the color effect and blending, apply filters, and move the symbol instances in three-dimensional space.

Importing Illustrator files

Flash lets you import Adobe Illustrator files (*.AI) files and to a large extent preserves the artwork's editability and visual fidelity. Flash Professional CC (2014 release) provides an AI Importer that includes a great degree of control in determining how Illustrator artwork is imported into Flash.

Flash Professional automatically recognizes layers, frames, and symbols. If students are more familiar with Illustrator, you can suggest that students design layouts in Illustrator, and then import them into Flash to add animation and interactivity.

Choose File > Import > Import To Stage or File > Import > Import To Library to import the artwork into Flash. Alternatively, students can even copy artwork from Illustrator and paste it into a Flash document.

Importing Layers

When an imported Illustrator file contains layers, you can import them in any of the following ways:

- Convert Illustrator layers to Flash layers
- Convert Illustrator layers to Flash keyframes
- Convert all Illustrator graphics to a bitmap
- Convert all Illustrator layers to a single Flash layer

Importing Symbols

Working with symbols in Illustrator is similar to working with them in Flash. In fact, you can use many of the same symbol keyboard shortcuts in both Illustrator and Flash: Press F8 in either application to create a symbol. When you create a symbol in Illustrator, the Symbol Options dialog box lets you name the symbol and set options specific to Flash, including the symbol type (such as movie clip) and registration point location.

If you want to edit a symbol in Illustrator without disturbing anything else, double-click the symbol to edit it in isolation mode. Illustrator dims all other objects on the artboard. When you exit isolation mode, the symbol in the Symbols panel—and all instances of the symbol—are updated accordingly.

Use the Symbols panel or the Control panel in Illustrator to assign names to symbol instances, break links between symbols and instances, swap a symbol instance with another symbol, or create a copy of the symbol.

Copying and Pasting Artwork

When you copy and paste (or drag and drop) artwork between Illustrator and Flash, the Paste dialog box appears. The Paste dialog box provides import settings for the Illustrator file you're copying. You can paste the file as a single bitmap object, or you can paste it using the current preferences for AI files. Just as when you import the file to the Stage or the Library panel, when you paste Illustrator artwork, you can convert Illustrator layers to Flash layers.

In this lesson, students import an Illustrator file with the characters of the cartoon frame. Many of the same options are available when copying and pasting artwork. If Illustrator is installed on the computer, open the Illustrator file and copy and paste, or drag and drop, artwork from it to the Stage in Flash to demonstrate how the options are similar.

About symbols

A *symbol* is a reusable asset that you can use for special effects, animation, or interactivity. There are three kinds of symbols: the graphic, button, and movie clip. Symbols can reduce the file size and download time for many animations because they can be reused. You can use a symbol countless times in a project, but Flash includes its data only once.

Symbols are stored in the Library panel. When you drag a symbol to the Stage, Flash creates an *instance* of the symbol, leaving the original in the library. An instance is a copy of a symbol located on the Stage. You can offer your students the analogy of a symbol as a photographic negative. The instances on the Stage can be thought of as prints of the negative. With just a single negative, you can create multiple prints. It's also useful to talk about symbols as containers. Like containers, symbols hold different kinds of content—whether it's an imported Illustrator graphic, a bitmap, or a piece of text. You can always edit or replace the contents of the symbol, just like you can edit or replace the contents of a container.

Movie clip symbols, button symbols, and graphic symbols each have unique functions, benefits, and limitations. Create one of each, and then show students the differences in the Timeline and in the Properties inspector for each.

Creating symbols

In Flash, there are two ways to create a symbol. The first is to have nothing on the Stage selected, and then choose Insert > New Symbol. Flash brings you to symbol-editing mode, where you can begin drawing or importing graphics for your symbol.

The second way is to select existing graphics on the Stage, and then choose Modify > Convert to Symbol (F8). Whatever is selected will automatically be placed inside your new symbol. The "Convert to Symbol" language may confuse students, because there is really no "conversion." Explain that the selected items are simply placed inside the newly created symbol.

The registration point of a symbol is indicated by the crosshair on the Stage in symbol-editing mode. The registration point determines the point at which Flash reports the coordinates of the symbol and from which it resizes or transforms the symbol.

Editing and managing symbols

Demonstrate to students the two ways in which you can edit your symbols. First, you can double-click the symbol in the Library panel and edit its contents in symbol-editing mode. Return to the main Timeline by clicking the Scene 1 button on the Edit bar above the Stage. Second, you can double-click the instance on the Stage and edit the symbol in place. Editing in place lets you see all the other elements surrounding the instance on the Stage. When you edit a symbol, all of its instances on the Stage change to reflect the new edits.

Changing the appearance of instances

There are a number of ways you can change the appearance of instances. You can change the size of any instance with the Free Transform tool. You can also change the brightness, tint, or transparency of an instance by setting different values for the instance's Color Effect in the Properties inspector. You can also apply a Blend effect or a Filter on an instance. Blending refers to how the colors of an instance interact with the colors below it. Invite students to explore the different Blend options. Filters are special effects, such as blurs and glows, which you can apply to instances. Additionally, you can position or rotate any instance in three-dimensional space using the 3D Translation and 3D Rotation tool.

The Visible and Render options in the Properties inspector can help you manage movie clip instances. The Visible option is an easy way to position an instance on the Stage and control its visibility at a later point in time, either with code or with the Visible option. The Render options can help performance on mobile devices by simplifying the display of complicated vector art as bitmap art.

Using rulers, guides, and grids

Explain to students that many tools can help them position instances and other objects on the Stage. In Lesson 1, students used the Properties inspector to position objects by changing the X and Y coordinates, and in Lesson 2, they learned to use the Align panel to align several objects to each other. Another way to position objects on the Stage is to use rulers, guides, and grids. Rulers (View > Rulers) appear on the top and left edge of the Pasteboard to provide measurement along the horizontal and vertical axes. Guides (View > Guides > Show Guides) are vertical or horizontal lines that you can pull out from the rulers and position on the Stage but do not appear in the final published movie. The grid (View > Grid > Show Grid) superimposes vertical and horizontal lines on the Stage and also does not appear in the final published movie. Another useful feature is the Move Guide option, which appears when you double-click on a guide. In the Move Guide dialog box, you can enter pixel values to position a guide exactly where you want it.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

1 What are the three kinds of symbols, and how do they differ?

2 When you import a Photoshop document into Flash, how do you preserve the layer effects?

3 How do you convert a bitmap image into a vector graphic?

4 What are blending effects?

5 What is the difference between the Global and Local Transform option for the 3D Translation and 3D Rotation tool?

6 How do you reset the transformations of an instance on the Stage?

Answers

1 The three symbols are Graphic, Button, and Movie Clip. A Graphic symbol is the most basic of symbols, which you can use for animation, but does not support interactivity, blending effects, or filters. A Button symbol is used for interactivity and contains four special keyframes that describe how it appears when the mouse is interacting with it. A Movie Clip symbol contains its own independent Timeline. You can apply filters, color settings, and blending modes to a Movie Clip instance, and you can use ActionScript or JavaScript to control a Movie Clip instance.

2 In the Import to Stage dialog box, For Layer conversion, choose the Maintain editable paths and effects option. Any layer effects such as transparencies or blending will be preserved.

3 To convert a bitmap to a vector, import the bitmap image into Flash. Select the bitmap and choose Modify > Bitmap > Trace Bitmap. The options in the dialog box that appears determine how faithful of a trace the vector image will be to the original bitmap.

4 Blending refers to how the colors of an instance interact with the colors below it.

5 When the Global Transform option is selected, rotation and positioning is relative to the global, or Stage, coordinate system. The 3D display over the object that you're moving shows the three axes in constant position, no matter how the object is rotated or moved. When the Global Transform option is not selected, rotation and positioning is relative to the object. The 3D display shows the three axes oriented relative to the object, not the Stage.

6 If you want to reset the transformations of your instance, you can use the Transform panel (Window > Transform). Click the Remove Transform button in the lower-right corner of the Transform panel to reset the instance to its original settings.

Lesson 4: Animating Symbols

Lesson Overview

Lesson 4 uses an animated motion picture promotional Web site to introduce students to motion tweens, easing, editing the paths of motion, frame-by-frame animations, and nested animations.

About animation

Animation is the movement, or change, of objects through time. Animation can be as simple as moving a box across the Stage from one frame to the next. It can also be much more complex involving other properties other than an object's location. You can animate many different aspects of a single object: the position, the color or transparency, the size or rotation, and filters. You can also animate an object's position and rotation in three-dimensional space. You can control its path of motion, and its easing, which is the way an object accelerates or decelerates.

The basic workflow for creating motion goes like this: To animate objects in Flash, you select the object on the Stage, right-click/Ctrl-click, and choose Create Motion Tween from the context menu. Move the playhead to a different point in time and change one of its properties such as its position, transparency, or size.

Motion tweens require you to use a symbol instance. If the object you've selected is not a symbol instance, Flash will automatically ask to convert the selection to a symbol. Flash also automatically separates motion tweens on their own layers, which are called Tween layers. There can only be one motion tween per layer without any other element in the layer. Tween layers allow you to change various attributes of your instance at different keypoints over time. It's helpful to think of Motion tweening as "instance tweening," since you are changing the properties of an instance over time. This will help students differentiate between Motion tweening and Shape tweening, which they will encounter in a later lesson.

Changing the pacing and timing

Simply click and drag the end of a tween span to add more frames to it and expand the amount of time it takes to do the animation. All the keyframes within the tween span will be distributed evenly. You can also Shift-drag the end of the tween span to add frames without affecting the timing of the keyframes. Remind students that they can also right-click/Ctrl-click on the Timeline and choose Insert Frames to add frames, or press F5 on the keyboard to add frames.

To move individual keyframes, first Ctrl-click (Windows) or Command-click (Mac) to select the keyframe, and then click and drag it to a new position.

Demonstrate to students how the positions of the keyframes affect the timing of the animation and how the total length of the tween span affects the overall speed of the animation.

Span Based vs. Frame Based Selections

If students are having trouble selecting and editing their keyframes, double-check their preferences on span based or frame based selections. By default, Flash does not use span based selection, which means you can select individual keyframes within a motion tween. However, if you prefer to click on a motion tween and have the entire span (the beginning and end keyframes, and all the frames in between) be selected, you can enable span-based selection from the options menu on the top right corner of the Timeline.

With span based selection enabled, you can click anywhere within the motion tween to select it, and move the whole animation backward or forward along the Timeline as a single unit.

If you want to select individual keyframes while span-based selection is enabled, hold down the Ctrl (Windows)/Command (Mac) key and click on a keyframe.

Changing the path of the motion

The path that an object takes as it moves on the Stage during a motion tween is the motion path, and it is represented by a colored line. You can edit the path of the motion with the Free Transform tool, or by using the Selection tool, or, if you want Bezier precision, you can use the Convert Anchor Point tool and Subselection tool.

Creating nested animations

Movie clip symbols have their own Timelines, similar to the main Timeline. A movie clip symbol may have other symbols nested within it. In this lesson, students create a nested animation of the alien waving his arms and another nested animation of the car jerking up and down. Ask your students to brainstorm what other kind of nested animations would require an animation inside a movie clip symbol.

For nested animations, demonstrate the export functions that generate a PNG sequence or a sprite sheet. PNG sequences and sprite sheets are used as animation assets for alternative (non-Flash) ways to create animation.

Frame by frame animation

Frame-by-frame animation refers to the illusion of movement created by seeing the incremental changes between every keyframe. It's the closest to traditional hand-drawn cel animation, and it's as tedious. In Flash, you can change a drawing in every keyframe, and create a frame-by-frame animation. In this lesson, students insert a frame-by-frame animation inside a movie clip symbol of a car to make it move up and down in a jittery fashion. When the movie clip loops, the car will rumble slightly to simulate the idle of the motor.

Ask students what situations would require frame-by-frame animations. Remind students that frame-by-frame animations increase your file size rapidly because Flash has to store the contents for each keyframe, so warn them to use the technique sparingly.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

1 Why is animation referred to as "tweening"?

2 If you want to animate an airplane moving across a static background, how many layers are required?

3 Identify two ways that you can lengthen the duration of a motion tween.

4 What is the Motion Presets panel and how do you use it?

5 What does the Orient to path option do?

6 What's the difference between an ease-in and an ease-out?

Answers

1 The term "tweening" comes from the world of classic animation. Senior animators would be responsible for drawing the beginning and ending poses for their characters. The beginning and ending poses were the keyframes of the animation. Junior animators would then come in and draw the "in-between" frames, or do the "in-betweening." Hence, "tweening" refers to the smooth transitions between keyframes.

2 You need two layers: one layer to create a motion tween for the airplane and a second layer for the static background.

3 To lengthen the duration of a motion tween, you can drag the end of a tween span to lengthen its total time. You can also choose Insert > Timeline > Frame (F5) to add frames to a tween span.

4 The Motion Presets panel (Window > Motion Presets) stores particular motion tweens so you can apply them to different instances on the Stage. To save a motion tween, select the motion tween on the Timeline or the instance on the Stage. In the Motion Presets panel, click the Save selection as preset button and name your motion tween to be stored in the Motion Presets panel.

5 The Orient to path option in the Properties inspector adjusts the object traveling along its path so it is oriented in the direction in which it is heading.

6 Easing refers to the way in which a motion tween proceeds. An ease-in refers to motion that gradually begins from its starting keyframe. An ease-out refers to motion that gradually slows down as it approaches its ending keyframe.

Lesson 5: Advanced Motion Tweening

Lesson Overview

In Lesson 5, students create a short animated infographic illustrating the idea of emerging market currencies as a threat to the highest-traded currencies in the world. The animation is something you might see accompanying a story in the business and economy section of a news site. In the animation, blocks with the American, European Union, and Japanese flags, representing the dollar, the Euro, and the Yen, respectively, stack themselves up. A moment later, a large block representing the emerging market currencies bumps the stack, sending them tumbling down. Students learn advanced motion tweening using the Motion Editor to make refinements in the motion tweening of the blocks, including the bouncing movements as they fall.

About the Motion Editor

It's important to remind students that they don't have to use the Motion Editor at all to create animations. However, the Motion Editor can make complex animation tasks much simpler.

The Motion Editor is an advanced panel that is only accessible when you have a motion tween on your Timeline. The panel shows you how the properties of your animation change over the course of the tween span as lines on a graph. You can modify the curves on the graph, called *property curves*, by adding or deleting anchor points, and change their curvature with Bézier precision. You can copy property curves and apply them to other properties or other motion tweens.

The Motion Editor also allows you to apply sophisticated eases to your animation. Although you can apply eases through the Property Inspector, the Motion Editor provides a wider range of ease types, with options to customize any ease. The Motion Editor also visually shows you how your ease affects a property curve. You can even apply different eases to different property curves.

Give your students some examples of how the Motion Editor would simplify the job of creating a complex animation. For example, in the previous lesson, students made the cars move up and down in a jittery fashion with frame-by-frame animations. With the Motion Editor, you could create the same effect by simply applying a Random ease to the motion.

Editing property curves

Double-click a motion tween, or right-click/Ctrl+click and choose Refine Tween, and Flash expands the motion tween, revealing the Motion Editor panel. The Motion Editor contains lines that represent the changes in the instance's properties. The units on the vertical axis of the Motion Editor measure the values of a selected property.

To change the shape of a property curve, you add anchor points. From each anchor point, you can change the curvature or direction of the curve. Select the Add Anchor point option at the bottom of the Motion Editor, and click directly on a curve to add an anchor point. Alternately, you can double-click on a curve to add a new anchor point. Drag the handle bars of any anchor point to change the curvature of the property curve.

You can move any anchor point, including the ones at the first and last keyframes, to new property values along the vertical axis. You can also move any of the anchor points (except the first and last ones) to new times along the motion tween. In effect, when you move anchor points to new positions on the curve, you're moving the keyframes on the tween span.

Reinforce the connection between the shape of the property curve and the behavior of the animation to your students. It's often difficult for beginners to understand how animation is visualized as curves, especially when separated into X and Y components.

If a student has added too many anchor points, reassure them that they can always remove the anchor points (except for the first and last anchor points). Removing an anchor point has the same effect as removing a keyframe of the motion tween. Hold down the Ctrl/Cmd key and click any anchor point (except the first or last) to remove the anchor point.

Viewing options in the Motion Editor

Demonstrate the Zoom and the Fit to View options in the Motion Editor to change the size and viewing area of the panel.

Applying and understanding eases

The Motion Editor provides a host of different ease types that can give a simple motion a more interesting appearance.

Click the Add Ease option in the Motion Editor. Flash opens the Ease panel, which lists the different ease options, from No Ease at the top, to Custom ease at the bottom. The different eases change the way the motion tween proceeds from the first keyframe to the last. The Simple eases, for example, affect one side of the motion tween (either the beginning or the end). The eases in the Stop and Start category affect both sides of the motion tween. The eases in the other categories are more complex ease types. For Simple eases, the Easing value determines the strength of the ease and the direction of the ease. A positive number represents an ease-out, and a negative number represents an ease-in.

When you apply an ease to a property curve, a dotted line is superimposed on the original property curve. The dotted line shows the effect of the ease on the tween.

Ease curves vs. property curves

Students may be confused about the difference between ease curves and property curves, and it's valuable to take the time to explain the difference and provide examples. When you apply an ease to a motion tween with the Motion Editor, the original property curve is not altered permanently. Applying eases is akin to a filter, which influences the outcome of an original property, but the ease, like a filter, can be edited or removed altogether.

Complex eases

In addition to ease-ins and ease-outs, the Motion Editor is powerful because of the more complex eases that students can apply to a motion tween. In this lesson, students use the Bounce-In ease to simulate a block falling to the ground. The ease changes the simple linear trajectory of the fall into one with multiple bounces.

Students also apply a different ease to different property curves. Applying an ease-out ease to the horizontal motion of the falling block helps provide a sense of weight, as the object slowly settles to a halt.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

1. How do you change the view in the Motion Editor?
2. What are the units of the vertical axis in the Motion Editor?
3. What's the relationship between an anchor point on a property curve and a keyframe of the motion tween?
4. You can use the Motion Editor with what kinds of animation?

Answers

1. Change the zoom level with the magnification menu at the bottom of the Motion Editor, or click the Fit to View option at the bottom to expand the panel to fit the existing space on the Timeline. Click the Fit to View option again to restore the panel to its default size.
2. The Motion Editor displays the vertical axis in the units of the property curve. For example, the x and y property curves are displayed in pixels, while the alpha property curve is displayed in percentage units.
3. An anchor point on the property curve represents a keyframe in the motion tween.
4. The Motion Editor can be used for motion tweens only. Classic tweens, shape tweens, and frame-by-frame animations are not supported.

Lesson 6: Animating Shapes and Using Masks

Lesson Overview

Lesson 6 introduces shape tweening and masking. Shape tweening allows students to morph one shape to another, and is ideal for creating organic animations. Masking is a way to selectively show only parts of a layer. Students explore both techniques in combination to create more sophisticated effects.

Animating shapes

Shape tweening is a technique for interpolating amorphous changes between shapes in different keyframes. Shape tweens make it possible to smoothly morph one thing into another. Any kind of animation that requires that the contours of a shape change—for example, animation of clouds, water, or fire—is a perfect candidate for shape tweening. Explore with your students different animation challenges and have them identify which techniques would be most appropriate to complete the animation. In this lesson, students create a simple animated logo that contains an animation of a flickering flame.

Emphasize to students that both the fill and the stroke of a shape can be smoothly animated. You can even animate changes in variable width strokes. Also, point out that because shape tweening only applies to shapes, you can't use groups, symbol instances, or bitmap images.

Students should be familiar with the basic workflow: Create a shape in the initial keyframe, insert a new keyframe further down the Timeline, change the shape in the newly-created keyframe, then apply a shape tween between the two keyframes. Create a broken shape tween by clearing the last keyframe, and help them recognize the broken shape tween. Identify ways to fix the tween.

Changing the pace

If students are comfortable editing the pacing and timing of motion tweens, then they should have no trouble doing the same with shape tweens. Reinforce that the process is identical. If you want the animation to proceed quicker, then the time between keyframes must be shorter. You can select individual keyframes and move them closer or farther apart to change the pacing of the shape tween.

Using shape hints

Shape hints force Flash to map points on the first shape to corresponding points on the second shape. By placing multiple shape hints, you can control more precisely how a shape tween appears. Insert a shape hint by selecting the first keyframe of a shape tween. Then choose Modify > Shape > Add Shape Hint. A tiny, circled letter appears in the first keyframe and a corresponding circled letter appears in the next keyframe. Move both circled letters to matching points on the contours of the shapes.

A maximum of 26 shape hints are allowable; for best results, place shape hints in a clockwise or counterclockwise fashion.

Previewing the Animation with Onion Skin Outlines

It's sometimes useful to see how your shapes are changing from one keyframe to another on the Stage all at once. Seeing how the shapes gradually change lets you make smarter adjustments to your animation. You can do so using the Onion Skin Outlines option, available at the bottom of the Timeline.

Onion Skin Outlines show the contents of the frames before and after the currently selected frame.

The term "onion skin" comes from the world of traditional hand-drawn animation, when animators would draw on thin, semi-transparent, tracing paper known as onionskin. When creating an action sequence, animators flip back and forth quickly between drawings held between their fingers. This allows them to see how the drawings smoothly connect to each other.

To use Onion Skin Outlines, click the Onion Skin Outlines button to enable it. Drag the beginning or end marker to select the range of frames around the playhead that you want to show. You can also select the preset marker options in the Modify Markers menu.

Only the content in the keyframes of a shape tween are displayed fully rendered. All other frames are shown as outlines. To see all frames fully rendered, click the Onion Skin option.

Creating masks

Masking is a way of selectively hiding and displaying content on a layer. Masking allows you to control the content that your audience sees. Provide several examples of the kinds of effects that masking can do: For example, you can make a circular mask to create a spotlight effect, and you can animate both the shape of the spotlight or the image under the spotlight.

In Flash, you put a mask on one layer and the content that is masked in a layer below it. After the students grasp the basic concept of masks, reveal the possibilities of an animation in the Mask layer or an animation in the Masked layer.

To create a mask, double-click the Layer icon in front of the layer name. In the Layer Properties dialog box that appears, select Mask. That layer becomes a Mask layer. Anything that is drawn in this layer will act as a mask for a masked layer below it.

To create the Masked layer, simply drag another normal layer under the Mask layer. Or, double-click the Layer icon in front of the layer name in the layer below the Mask layer. In the Layer Properties dialog box that appears, select Masked. Both the mask and masked layers must be locked in order for you to see the results of the mask in the authoring environment.

Warn students of the limitations of masking: Flash does not recognize different Alpha levels of a mask. For example, a mask at an Alpha value of 50% will still mask at 100%. However, with ActionScript you can dynamically create masks that will allow transparencies. Masks also do not recognize strokes.

For the animated logo students create in this lesson, they add a mask and an animation in the masked layer that makes the text a little more visually interesting.

Traditional Masks

It might seem counter-intuitive that the shapes in the Mask layer reveal, rather than hide, the content in the Masked layer. However, that's exactly how a traditional mask in photography or painting works. When a painter uses a mask, the mask protects the painting from paint splatters. When a photographer uses a mask in the darkroom, the mask protects the photo-sensitive paper from the light, to prevent those areas from getting any darker. So thinking of a mask as something that protects the lower, Masked, layer, is a good way to remember which areas are hidden and which areas are revealed.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

1 Of the following properties, which one CANNOT be smoothly animated in a shape tween? Gradient fill, stroke size, variable stroke width, transparent fill, blur filter.

2 How do you remove shape hints?

3 What is the difference between using an opaque shape and a shape with a fill at 50% opacity in the Mask layer?

4 A mask requires two layers: a Mask layer, and a Masked layer. Which one appears above the other on the Timeline?

Answers

1 The blur filter can not be smoothly animated with a shape tween. Shape tweens require shapes, and filters are only applied to symbol instances.

2 To remove a shape hint, drag it entirely off the Stage and Pasteboard. To remove all shape hints, choose Modify > Shape > Remove All Hints. Removing a shape hint in one keyframe will remove its corresponding shape hint in the other keyframe.

3 There is no difference; Flash does not recognize different Alpha levels in a mask, so a semi-transparent fill in the Mask layer has the same effect as an opaque fill, and edges will always be hard-edged. However, with ActionScript you can dynamically create masks that will allow transparencies.

4 The Mask layer appears above the Masked layer on the Timeline. The Masked layer is also indented, indicating that it is affected by the Mask layer.

Lesson 7: Creating Interactive Navigation

Lesson Overview

Lesson 7 introduces ActionScript and the button symbol. Students learn how to structure the Timeline to create a project with non-linear navigation—a project in which the user chooses what content to see and when to see it. Students learn about basic event handling and the commands for controlling the playhead. The end result is an interactive restaurant guide in which the user can click on any thumbnail image of a restaurant to see more information and reviews of that restaurant.

About interactive movies

Students should learn what it means to create non-linear navigation. In this lesson, students create a Flash movie that doesn't play straight from the beginning to the end. ActionScript provides the instructions to make the Flash playhead to jump around and to go to different frames of the Timeline based on which button the user clicks. Different frames on the Timeline contain different content. Users don't actually know that the playhead is jumping around the Timeline—all they see (or hear) is different content appearing as they click the buttons on the Stage.

Emphasize that there are many other ways to interact with Flash content: clicking a button is the simplest and is the one demonstrated in this lesson. You can also interact with content with mobile gestures such as pinching or swiping, or by tilting a mobile device.

Creating buttons

The buttons in this lesson include small thumbnail bitmap images with some text. Review the three kinds of symbols and the benefits of each. Button symbols contain four unique keyframes. Demonstrate and explain each keyframe: Up, Over, Down, and Hit:

- The Up state determines the button's default appearance when the mouse is not interacting with it.
- The Over state appears when the mouse rolls over the button.
- The Down state appears when the mouse is depressed over the button.
- The Hit state defines the button's active area, which is the area that responds to the mouse rolling over or clicking the button.

A particularly effective demonstration of the Hit state is to delete half of the Hit state graphic in the button symbol. Test the movie with the button instance on the Stage and show how only the remaining half of the button is responsive to the mouse.

For more advanced students, show how button symbols with only their Hit states defined can be used as invisible buttons. Place an invisible button instance on the Stage to make that particular area clickable and responsive to the mouse.

Naming the button instances

Giving names to instances is a critical step in creating interactive Flash projects. Emphasize the importance of naming instances so ActionScript (or JavaScript, in the case of an HTML5 Canvas document) can refer to them. The most common beginner mistake is not to name, or to incorrectly name, a button instance. The instance names are not the same as the symbol names in the Library panel. The names in the Library panel are simple organizational reminders.

Instance naming follows these simple rules:

- Do not use spaces or special punctuation. Underscores are OK to use.
- Do not begin the name with a number.
- Do not use any words that are also ActionScript key words, or commands.
- Be mindful of uppercase and lowercase letters.

One source of confusion is the optional, but helpful, _btn suffix. Some students think that the suffix is only required in the Instance Name field in the Properties inspector but not in the actual ActionScript coding, or vice versa. Remind them that the names must match exactly.

Name instances in a way that is meaningful, so you'll remember what you've called them when you're writing ActionScript. Shorter names are easier to type without error, but longer names may be easier to remember and to identify when you're editing the ActionScript later.

Understanding ActionScript 3.0

For ActionScript 3.0 documents, Adobe Flash Professional CC (2014 release) uses ActionScript 3.0, the scripting language that extends Flash functionality and enables interactivity in Flash projects. Previous versions, ActionScript 1.0 and 2.0, are not supported. Students who haven't worked with scripting languages before may initially be intimidated and confused by ActionScript, which is an object-oriented scripting language. This lesson introduces basic scripting concepts, terminology, and syntax, as well as specific ActionScript coding to make their buttons functional.

About ActionScript

The ActionScript scripting language lets you add complex interactivity, playback control, and data display to your ActionScript 3.0 document. You can add ActionScript in the authoring environment by using the Actions panel.

ActionScript follows its own rules of syntax, includes reserved keywords, and lets you use variables to store and retrieve information. ActionScript includes a large library of built-in classes that let you create objects to perform many useful tasks. For detailed information on ActionScript, tell students that they can always refer to *Programming ActionScript 3.0* or the *ActionScript Language References*; both are available in Flash Help. It's particularly important for students to be able to find and learn how to use an ActionScript element on their own by reading the Help entries.

You don't need to understand every ActionScript element to begin scripting; if you have a clear goal, you can start building scripts with simple actions.

Because there are multiple ways of incorporating ActionScript into your Flash document files, there are several different ways to learn ActionScript.

Dealing with errors

All students struggle with syntax errors when they learn ActionScript. Explain to your students that this is a normal process, and even veteran developers deal with debugging. Share with them the ways in which they can identify and fix the mistakes in their code:

- Use the errors that appear in the Compiler Errors panel to identify the kind and source of the problem.
- Use the color-coding of the code as it appears in the Script pane to check if the correct keywords are colored.
- More advanced students can begin to use the trace() command to send messages to the Output panel.

Creating event handlers for buttons

Events happen in the Flash environment that Flash can detect and respond to. Explain the concept of an event to students, and give examples of various types of events. For example, a mouse click, a mouse movement, and a key press on the keyboard are all events. Other mobile device interactions are also events, such as pinching, swiping, or even accessing the accelerometer by tilting the device. These events are produced by the user, but some events can happen independently of the user, like the successful loading of a piece of data or the completion of a sound. With ActionScript, you can write code that detects events and responds to them with an event handler.

The first step in event handling is to create a listener that will detect the event. A listener looks something like this:

```
wheretolisten.addEventListener(whatevent, responsetoevent);
```

The actual command is `addEventListener()`. Clarify to students which of the words are placeholders for objects and parameters for their situation. Wheretolisten is the object where the event occurs (usually a button), whatevent is the specific kind of event (such as a mouse click), and responsetoevent is a function that gets triggered when the event happens.

The next step is to create the function that will respond to the event. A function simply groups a bunch of actions together, which you can trigger by referencing its name. A function looks something like this:

```
function responsetoevent (myEvent:MouseEvent):void{ };
```

In this particular example, the name of the function is called responsetoevent. It receives one parameter (within the parentheses) called myEvent, which is an event that involves a mouse event. If this function is triggered, all the actions between the curly braces are executed.

Students should memorize the most common mouse events and ActionScript commands for navigating the Timeline. The mouse events they should remember are

- MouseEvent.CLICK
- MouseEvent.MOUSE_MOVE
- MouseEvent.MOUSE_DOWN
- MouseEvent.MOUSE_UP
- MouseEvent.MOUSE_OVER
- MouseEvent.MOUSE_OUT

The commands to control the playhead that they should remember are

- stop();
- play();
- gotoAndStop(framenumber or “framelabel”);
- gotoAndPlay(framenumber or “framelabel”);
- nextFrame();
- prevFrame();

Creating destination keyframes

When the user clicks each button, Flash will move the playhead to a new spot on the Timeline. Students are asked to create new content at those particular frames. In this lesson, students learn to put new content in different keyframes and to label each keyframe to make referencing each from ActionScript easier.

A few essential points for students to remember: Use straight double quotes around the name of the frame label in their ActionScript code. If they want to use the gotoAndPlay() command, they will also need to add a stop() command on the Timeline to stop the playhead before it displays all the other keyframes along the Timeline.

Using the Code Snippets panel

The Code Snippets panel provides commonly used ActionScript or JavaScript code for you to easily add simple interactivity to your Flash project. If students are unsure of coding their own buttons, they can use the Code Snippets panel to learn how to add interactivity. You can also save, import, and share code between a team of developers to make the development and production process more efficient. In this lesson, students use the Code Snippets panel to add interactivity to their home button.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

1 What happens if a button symbol has an empty keyframe in its Up state but filled keyframes in its Over, Down, and Hit states?

2 What is the significance of the color of a shape in a button symbol's Hit keyframe?

3 In what kind of situation would you want a button's Hit state to be larger than its Up state?

4 What is wrong with the instance name of this button, 02restaurantbtn, and how would you fix it?

5 What additional information is needed in between the parentheses of the command gotoAndStop()? What information is needed in between the parentheses of the command stop()?

6 How do you use the Code Snippets panel to program a button to go to a specific frame?

Answers

1 If a button symbol has an empty keyframe in its Up state, the button instance on the Stage is invisible. The mouse still can interact with the button instance, but the user won't be able to see it.

2 There is no significance to the color of a shape in a button symbol's Hit keyframe. Only the size and position of a shape in the Hit state determines the clickable area.

3 You may want the clickable area to cover a larger area than the visible button. For example, a typical pause button on consumer electronics is represented by two parallel vertical lines. You would want the Hit state to cover the gap between the vertical lines so users have an easier time clicking the button.

4 The instance name cannot begin with a number. You may want to add an underscore before btn, but it is not incorrect without it.

5 The gotoAndStop() command requires the destination frame number or name of the frame label, in quotation marks. The stop() command doesn't require any information between its parentheses.

6 Choose Window > Code Snippets, or if your Actions panel is already open, click the Code Snippets button at the top right of the Actions panel to open the Code Snippets panel. The code snippets are organized in folders that describe their function. First, select a button instance on the Stage. In the Code Snippets panel, expand the folder called ActionScript > Timeline Navigation and double-click the Click to Go To Frame and Stop option. You can also click the Add to current frame button. Flash automatically adds the ActionScript code to the Timeline. Customize the ActionScript code by providing the correct destination frame in the gotoAndStop() command.

Lesson 8: Using Text

Lesson Overview

In Lesson 8, students explore the layout, formatting, and interactive features available with the Text tool to create a mock Android mobile app for a real estate company showcasing a property for sale. As they work through the project, students learn the difference between Static, Dynamic, and Input text, they learn to add hyperlinked text, and use ActionScript to read user-inputted text and dynamically change text on the Stage. Students also learn how text is integrated in the different Flash document types.

How and where you add text depends on the Flash document type that you're working with.

- ActionScript 3.0 documents: Use static, dynamic, or input text to integrate text with your animations and graphics to display in the Flash Player.
- AIR for mobile or desktop documents: Use static, dynamic, or input text to integrate text with your animations and graphics to display in AIR for the desktop or AIR for mobile.
- HTML5 Canvas documents: Text support is limited, so the best approach is to include text in the HTML document, and not from Flash Pro.
- WebGL documents: Text is not supported, as WebGL documents are geared for animation and graphics only. Include text in the HTML document, as text within Flash Pro in a WebGL document is disabled.

Text types

When you add text to a Flash document, you have three options: Static text, Dynamic text, and Input text.

- Static text displays text that can't be selected or edited by the end user. Flash embeds the fonts that you use when you add Static text so your audience sees the fonts on the Stage as you intend it. Use Static text for display text, such as titles, labels, and other informational text.
- Dynamic text displays text that can be controlled with code. You can dynamically change the contents of the text to respond to different conditions while your movie plays. For example, use Dynamic text for a scoreboard that shows a player's current score during an interactive game.
- Input text displays text that the user enters from the keyboard or touch screen. With code, you can analyze the text, store the information, and process it for different responses during the playback of the project. For example, use Input text for questionaires or for order forms for online purchasing.

Adding text

Adding text is a rather straightforward process. Encourage students to explore many of the formatting options available in the Properties inspector. You should also remind students that text remains editable, which means that they can always double-click the text box to edit the text or change its formatting options. Students can break apart text (Modify > Break Apart) to convert each letter into separate drawing objects whose stroke and fill they can modify. However, once broken apart, the text will no longer be editable.

Hyperlinking text

Hyperlinks are easy to include in text, and they don't require any coding. Highlight text, and in the Options section of the Properties inspector, enter a URL (be sure to include the http://) for the Link and choose _blank in the Target pull-down menu. The selected words in your text box become underlined, indicating that it is hyperlinked.

Remind students that the hyperlinked text is underlined only in the Flash Pro authoring environment. When the published movie plays back, the hyperlinked text appears like the other text, so you should distinguish the hyperlinked text in some way. The standard visual cue for hyperlinks is a different color, or an underline.

Creating user-input text

Students create a Mortgage Calculator, which accepts input from the user through the keyboard, and displays estimated monthly payments based on those inputs. Students create user-input text with Dynamic and Input text. The contents of a Dynamic or Input text box are referenced by the keyword text. Although the Mortgage Calculator part of this project may appear complicated, emphasize for students two take-home messages: First, the contents of text boxes are referenced by the text property. Second, text boxes contain String data. To make numeric calculations, you must first convert the text to a number with Number(). To convert a number back to text, you use String().

Embedding fonts and device fonts

For any text that may be edited during runtime, you should remind students to embed their fonts. Since the user can enter any kind of text in editable text boxes, you need to include those potential characters in the final published document to ensure that text appears as you expect it, with the same font that you've chosen in the Properties inspector. The alternative to embedding fonts is to use device fonts. These options find and use the fonts on a user's computer that most closely resemble the specified device font. When you use device fonts, you don't have to worry about embedding fonts, and you can be assured that your viewer sees text that is similar to the text that you see in the authoring environment.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

1 Which Flash documents fully support text?

2 How do you add a hyperlink to text?

3 What is the Target option in hyperlinked text?

4 Why would you need to use the ActionScript commands String() or Number()?

5 What are device fonts?

Answers

1 ActionScript 3.0, AIR for Desktop, AIR for Android, and AIR for iOS documents fully support text within Flash Pro. While you can use the Text tool in an HTML5 Canvas document, it's recommended that you add text in the HTML file instead.

2 Select the words you want to be hyperlinked. In the Advanced Character section of the Properties inspector, enter the URL for the Link and choose _blank in the Target pull-down menu. The selected words in your text box become underlined, indicating that it is hyperlinked.

3 The Target field determines where the Web site loads. An _blank Target means that the Web site loads in a blank browser window. An _self Target loads the URL in the same browser window, taking over your Flash movie. An _top or _parent Target refers to the arrangement of a frameset and loads the URL in a particular frame relative to the current frame.

4 Text boxes contain text, or String data. To convert the contents of a text box into a number for the sake of calculations, use Number(). To convert a number back to text, you use String().

5 Device fonts are three generic options grouped at the top of the Character Family pull-down menu. Use device fonts as an alternative to embedding fonts. You can also choose the Use device fonts option from the Anti-alias pull-down menu. The three device fonts are _sans, _serif, and _typewriter. These options find and use the fonts on a user's computer that most closely resemble the specified device font. When you use device fonts, you don't have to worry about embedding fonts, and you can be assured that your viewer sees text that is similar to the text that you see in the authoring environment.

Lesson 9: Working with Sound and Video

Lesson Overview

In Lesson 9, students bring audio and video files together to create a dynamic, interactive wildlife preserve kiosk. As they work through the project, students learn how to edit audio files, use Adobe Media Encoder to encode their video files to the correct format, and integrate video into their project. Students also learn to use Media Encoder to export Flash content to video in a variety of formats.

Using sounds

To add an audio file to a frame in Flash, first import it to the Library panel. Then select the frame at which you want the audio file to begin playing. In the Properties inspector, choose the audio file from the Sound menu. You can also simply drag the sound file from the Library panel to the Stage.

Selecting the appropriate option from the Sync menu is important. If a sound file doesn't play when you think it should, check the Sync menu to ensure you've selected the right option:

- Event synchronizes the sound to the occurrence of an event, such as a button click. An event sound plays when its first keyframe appears and plays in its entirety, even if the project stops playing. If another event occurs, the first instance of the sound continues playing while the second instance of the sound plays.
- Start synchronizes the sound to the occurrence of an event, just as the Event option. However, if the sound is already playing, no new instance of the sound plays.
- Stop silences the specified sound.
- Stream synchronizes the sound for playing on a Web site. Flash forces animation to keep pace with stream sounds. If it can't draw animation frames quickly enough to keep up with the sound file, it skips frames. Stream sounds stop if the Flash Timeline ends. Next to the Sync menu is a menu that determines how many times the sound file plays.
- Repeat specifies the number of times the file plays; specify a number.
- Loop repeats the sound continuously.
- None applies no effect
- Left Channel/Right Channel plays sound in the left or right channel only.
- Fade Left To Right/ Fade Right To Left shifts the sound from one channel to the other.
- Fade In gradually increases the volume of a sound over its duration.
- Fade Out gradually decreases the volume of a sound over its duration.
- Custom lets you create custom in and out points of sound using Edit Envelope

● **Note:** Repeating stream sounds is not recommended, because frames are added to the file and the file size is increased by the number of times the sound is repeated. You can also apply effects to the sound file.

Editing a sound clip

In Flash, you can change the point at which a sound starts and stops playing, or control the volume of the sound as it plays. In this lesson, students clip the end of the waveform of a sound to shorten the sound clip and edit the volume levels:

- To edit a sound file, first add it to a frame, and click Edit in the Properties inspector.
- To change the start and end points of a sound, drag the Time In and Time Out controls in the Edit Envelope.
- To change the sound envelope, drag the envelope handles to change levels at different points in the sound. Envelope lines show the volume of the sound as it plays. To create additional envelope handles (up to eight total), click the envelope lines. To remove an envelope handle, drag it out of the window.
- To display more or less of the sound in the window, click the Zoom In or Out buttons.
- To switch the time units between seconds and frames, click the Seconds and Frames buttons.

Understanding Video for Flash

Before you start working with video, consider what video quality you need, what video format to use, what size can accommodate your layout, and how it will be integrated with the rest of your Flash project. The two ways in which you can use video in Flash is (1) to embed the video, or (2) to play the video from an external file. Embedding video requires that the video be in FLV format, an older format that is no longer supported in Adobe Media Encoder. Playing from an external file requires that the video be encoded in the H.264 codec.

When you embed a video, it increases the size of the final published file. The embedded video appears on the Timeline and can be synchronized with other Flash elements.

When you play external video, the final project remains small because the video is kept separate from the published document. Playback of external video depends on a video playback component, which you can configure, and an optional skin, which provides the look and feel and functionality of the playback controls.

Using Adobe Media Encoder

You can convert your video files into many different formats using Adobe Media Encoder, a stand-alone application that comes with Flash Professional CC (2014 release). Adobe Media Encoder can convert single files or multiple files (known as batch processing) to make your workflow easier.

You can customize many settings in the conversion of your original video into the required H.264 format. You can crop and resize your video to specific dimensions, just convert a snippet of the video, adjust the type of compression and the compression levels, or even apply filters to the video. To display the encoding options, click the Preset selection in the display window, or choose Edit > Export Settings. The Export Settings dialog box appears.

To simplify the conversion process, you can rely on the preset settings, which offer common encoding profiles for a one step operation.

Playback of external video

To play back an external video file encoded in the H.264 codec, in Flash, choose File > Import > Import Video. The Import Video wizard appears, which guides you through the process. During the process, you choose the external file that you want to play and an optional skin, which provides the look and feel and functionality of the controls.

Using external video files provides the following capabilities that are not available when using embedded (FLV) video:

- Long video clips. You can use longer video clips without slowing down playback. External video files are played using cached memory, which means that large files are stored in small pieces and accessed dynamically; they do not require as much memory as embedded video files.
- Different frame rate. An external video file can have a different frame rate from the Flash document in which it plays. For example, you can set the Flash document frame rate to 30 fps and the video frame rate to 21 fps, which gives you greater control in ensuring smooth video playback.
- No interruption. With external video files, Flash document playback does not have to be interrupted while the video file is loading. Imported video files can sometimes interrupt document playback to perform certain functions (for example, to access a CD-ROM drive). Video files can perform functions independently of the Flash document, so they do not interrupt playback.
- Captions and cue points. Captioning video content is easier with external video files because you can use callback functions to access metadata for the video. Cue points are special markers that can be tied to ActionScript code for additional functionality.

Embedding Flash Video

Embedded video lets you insert a video file directly within Flash. When you import video in this way, the video is placed in the Timeline where you can see the individual video frames represented in the Timeline frames. An embedded video file becomes part of the Flash document.

The frame rate of the video clip and the Flash file must be the same. If you use different frame rates for the Flash file and the embedded video clip, playback is inconsistent.

Embedded video works best for smaller video clips with a playback time of less than 120 seconds. If you are using video clips with longer playback times, consider keeping and playing the video as an external file.

The limitations of embedded (FLV) video include:

- Performance problems. You might encounter problems if the resulting Flash file become excessively large. Flash Player reserves a lot of memory when downloading and attempting to play large SWF files with embedded video, which can cause Flash Player to fail.
- Synchronization issues. Longer video files (over 120 seconds long) often have synchronization issues between the video and audio portions of a video clip. Over time, the audio track begins playing out of sequence with the video, causing a less than desirable viewing experience.
- Maximum length. There is a maximum length of 16,000 frames for embedded movies.

Despite these limitations, embedded video has a place for small, well-placed video segments. Use an older version of Adobe Media Encoder to export video to FLV to be used for embedding in Flash.

Exporting video

You can use Adobe Media Encoder to export your Flash content to video.

For example, use the powerful drawing and animation tools within Flash to create animation, and then export it as a HD broadcast quality video, or for playback on a variety of platforms such as an iPhone, Nook, Kindle, or Android device. Sound on the Timeline and even nested animations are exported.

In this lesson, students export the animation they created in Lesson 4, the cinematic splash screen for the fictional motion picture, *Double Identity*.

Questions

The following are additional questions that are not in the students' Classroom in a Book.

1. How do you set the quality of sounds?
2. How do you fade out the volume of a sound?
3. What are the Sync options for sound?
4. Which video formats are compatible with Flash?

5 Which of these options are *not* available in Adobe Media Encoder? (1) Cropping a video, (2) Trimming the length of a video, (3) Combining multiple video clips together, (4) Stripping the audio from a video, (5) Changing the frame rate of a video.

6 What is the SkinAutoHide option, and how do you enable it?

Answers

1 Set the sound quality and compression in the Publish Settings options. Select the Flash format from the left and click on the Audio Stream and Audio Event values. In the Sound Settings dialog box that appears, change the Bit rate, which determines the quality of the sound in your final, exported project. The higher the bit rate the better the quality.

2 You can change the volume levels in the Edit Envelope dialog box. Select the first keyframe that contains your sound and in the Properties inspector, click the Pencil button. The Edit Envelope dialog box appears. Click on the top horizontal line of the waveform to add keyframes. Drag the keyframes down to the bottom of the window to reduce the volume at that point in time. You can also select the Fade out option from the Effect menu in the Properties inspector to apply a global fade from start to finish.

3 Sound sync refers to the way the sound is triggered and played. There are several options: Event, Start, Stop, and Stream. Stream ties the sound to the Timeline so you can easily synchronize animated elements to the sound. Event and Start are used to trigger a sound when the playhead enters the keyframe. Event and Start are similar except that the Start sync does not trigger the sound if it is already playing (so there are no overlapping sounds possible with Start sync). The Stop option is used to stop a sound.

4 The video formats that are compatible with Flash are FLV (for embedded videos) and videos encoded with the H.264 codec (for external videos that play with the video playback component). The Adobe Media Encoder CC 2014 does not support FLV export, but earlier versions of Media Encoder do.

5 Option three. You cannot use Adobe Media Encoder to combine multiple video clips together.

6 To hide the video controller and only display it when users roll their mouse cursor over the video, use the SkinAutoHide option. Select your video on the Stage, and enable the SkinAutoHide option in the Component Parameters section of the Properties inspector.

Lesson 10: Publishing

Lesson Overview

In Lesson 10, students learn about the variety of publishing options available in the 2014 release of Flash Professional CC. Students publish sample projects for Flash Player, HTML5, an AIR for Desktop application, and projectors for Macintosh and Windows. Students also learn about the required steps to publish mobile apps from Flash Pro.

Understanding publishing

Publishing is the process that creates the required file or files to play your final Flash project for your viewers. Remind students that Flash Professional CC is the authoring application, which is a different environment from where viewers experience the final movie. In Flash Professional CC, you author content, which means you're creating art, animation, and adding text, videos, sound, buttons, and code. In the target environment, such as a desktop browser or a mobile device, your viewers watch the content when it plays back, or runs. So developers make a distinction between "author-time" and "runtime"environments.

Flash can publish content to various runtime environments, and your desired runtime environment determines the Flash document that you must choose when you first begin a project.

Playback environments

The Flash Player in a desktop Web browser is the runtime environment for an ActionScript 3.0 document. Flash Player 13 is the latest version and supports all the new features in Flash Professional CC. The Flash Player is available as a free plug-in from the Adobe Web site for all the major browsers and platforms. In Google Chrome, the Flash Player comes pre-installed and updates automatically.

If you want to target Web browsers without requiring the Flash Player, then begin your Flash project with an HTML5 Canvas document. To integrate interactivity in an HTML5 Canvas document, you use JavaScript rather than ActionScript. You can add JavaScript directly in the Actions panel yourself, or use HTML5 Canvas snippets.

Adobe AIR is another runtime environment. AIR (Adobe Integrated Runtime) runs content directly from your desktop, without the need for a browser. When you publish your content for AIR, you make it available either as an installer that creates a standalone application, or you can make the application with the runtime already installed. You can also publish AIR applications that can be installed and run on Android devices and iOS mobile devices such as the iPhone or iPad, whose browsers do not support the Flash Player.

Publishing for Flash Player

When you publish for Flash Player with an ActionScript 3.0 document, you publish two files: a SWF file and an HTML document that tells the Web browser how to display the SWF file. You need to upload both files to your Web server along with any other files your SWF file references (such as video files and skins). By default, the Publish command saves all the required files to the same folder.

You can specify different options for publishing a movie, including whether to detect the version of Flash Player installed on the viewer's computer.

Publishing for HTML5

HTML5 is the latest version for marking up Web pages for the browser. HTML5, along with CSS3 and JavaScript, are the modern standard for creating content for the Web on the desktop, mobile phone, and tablet. Choosing the HTML5 Canvas document type in Flash targets HTML5 as the published runtime and outputs a collection of HTML5 and JavaScript files.

Canvas refers to the canvas element, a tag that's new in HTML5 that allows JavaScript to render and animate 2D graphics. Flash Professional CC relies on the CreateJS JavaScript libraries to generate the graphics and animations inside the canvas element of the HTML5 project.

CreateJS is a suite of JavaScript libraries that enable rich, interactive content with HTML5. The new HTML5 Canvas document in Flash replaces the Tookit for CreateJS, which was a separate panel in previous versions of the Flash application that generated JavaScript and published HTML5 content.

While the Flash Professional CC update still relies on CreateJS, it remains hidden under the hood of the publish options. CreateJS itself is a collection of several separate JavaScript libraries: EaselJS, TweenJS, SoundJS, and PreloadJS.

- EaselJS is a library that provides a display list that allows you to work with objects on the canvas in the browser.
- TweenJS is a library that provides the animation features.
- SoundJS is a library that provides functionality to play audio in the browser.
- PreloadJS is a library that manages and coordinates the loading of content.

With CreateJS, Flash Professional CC generates all the necessary JavaScript code to represent your images, graphics, symbols, animations, and sounds on the Stage. Flash also outputs dependent assets, such as images and sounds. You can easily configure how those image and sound assets are organized in the publish settings.

You can also include your own simple JavaScript commands directly in the Actions panel to control your animation. Those JavaScript commands get exported in the JavaScript files.

The HTML5 Canvas document does not support all Flash Professional CC features. Fortunately, the Output panel displays warnings about any features in your Flash file that can't be successfully exported.

Flash also disables any features that won't export successfully to HTML5 and JavaScript. For example, the 3D Rotation and Translation tool in the Tools panel is grayed out to indicate that 3D rotations and translations are not supported in an HTML5 Canvas document. Many of the blending modes and some Filter effects are not supported as well.

Classic tweens

While all animation types (motion tweens and shape tweens) are supported in HTML5 Canvas documents, the CreateJS libraries convert those animations to frame-by-frame animations. Classic tweens, an older way of creating animation with Flash Professional, are retained as a runtime tween, which saves file size and gives you the opportunity to control the animation dynamically via JavaScript.

Classic tweening is an older approach to animation, but it's still a valuable technique to show your students. Like motion tweens, classic tweens use symbol instances. Changes to the symbol instances between two keyframes on the Timeline are interpolated to create the animation. You can change an instance's position, its rotation, scale, transformation, Color Effect, of Filter effect.

The key differences between classic tweens and motion tweens are:

- Classic tweens require a separate Motion Guide layer to animate along a path.
- Classic tweens are not supported in the Motion Editor panel.
- Classic tweens don't support 3D rotations or translations.
- Classic tweens are not separated on their own tween layer. However, classic and motion tweens share the same restriction that no other object can exist in the same layer as the tween.
- Classic tweens are Timeline-based, and not object-based, meaning you add, remove, or swap the tween or the instance on the Timeline and not the Stage.

Learning to create animation with classic tweens will help you maintain leaner animation sizes when you publish to HTML5 Canvas.

Inserting JavaScript

If you want to add interactivity to your HTML5 Canvas project, you can add JavaScript directly in the Actions panel, which gets exported to the published JavaScript files.

In previous Flash versions that used the Toolkit for CreateJS, you were required to write `/* js` to begin a block of JavaScript code, and `*/` to end the block of JavaScript code. The comment block distinguished ActionScript code with JavaScript code since the same document could export to both SWF and HTML5/JavaScript. Since Flash Pro CC now uses a dedicated document type for HTML5, you can write JavaScript directly in the Actions panel and not worry about mixing code.

Note: Important! The CreateJS JavaScript libraries that drive the publishing process reference Timelines starting with frame numbers at 0 while Flash starts frame numbers at 1. Therefore, frame numbers in your exported JavaScript code are one less than what you would expect to see in Flash. Because of this difference, Adobe recommends that you always use frame labels instead of frame numbers for Timeline navigation. Keep this in mind when you use the *gotoAndStop()* or *gotoAndPlay()* commands to control the playhead of the Timeline.

Show students how the Code Snippets panel makes it easy to insert simple JavaScript code. Choose Window > Code Snippets, then click the arrow in front of the HTML5 Canvas folder to reveal the JavaScript code snippets. Double-click any snippet in a folder to add the code to the Actions panel. Some snippets will require that you select an instance as the target of the code.

Publishing an AIR application

Most desktop computers have the Flash Player installed with their browsers, but you may want to distribute your movie to someone who doesn't have the Flash Player or who has an older version. Perhaps you just want your movie to run without a browser.

You can output your movie as an AIR file, which installs an application on the user's desktop. Adobe AIR is a more robust runtime environment that supports a broader range of technologies.

To create an AIR desktop application, edit the AIR application settings and provide some additional information:

- You need to provide a certificate so that users can trust and identify the developer of the Flash content. In the Signature section of the Application Settings dialog box, you can guide your students to create their own self-signed certificate, which has the .p12 extension.
- The Icons section of the Applications Settings dialog box allow students to use PNG images as icons for the AIR application.
- The Advanced section of the Applications Setting dialog box contains settings for the initial window location and options for resizing.

Creating projectors

A low-tech alternative to an AIR desktop application is a projector. A projector is a self-contained file that essentially includes the Flash Player runtime so your audience can simply double-click the projector icon to play and view your multimedia content.

You can publish either a Macintosh or Windows Projector from an ActionScript 3.0 or AIR document. However, unlike an AIR application, you don't have the publishing options such as choosing thumbnails for the app icon, or specifying a transparent background or initial position when the app is launched.

To create a projector, simply choose Command > Export as Projector. In the Output to Projector window that opens, specify the location to save your projector and the choice of Windows or Macintosh projector.

Publishing WebGL animations

WebGL is technology for rendering graphics—especially 3D graphics—in a browser with JavaScript. WebGL uses hardware acceleration for graphics rendering, which potentially makes complex graphics faster and smoother to display. Use the Flash WebGL document type to create flat graphics and animation for export as WebGL assets. Your output files consists of various JavaScript files (including .json files, which are a specific kind of JavaScript file for storing data—json stands for "JavaScript Object Notation"), and image assets.

Warn your students that the Flash WebGL document type is limited (it doesn't support 3D tranformations, text, or masks), but tell them to keep an eye on WebGL as both the technology and the Flash document type develop.

Testing a mobile app

Flash provides the SimController, which allows you to simulate mobile app interactions. Since creating an app for mobile devices is a little more complicated because it requires specific developer certificates and the actual hardware, the SimController is very useful for early testing. Interactions such as swiping and pinching, or even tilting the device can be emulated directly within Flash.

In this lesson, students use the SimController to test out mobile interactions in an already completed Flash file that targets AIR for Android. To test a mobile app, choose Control > Test Movie > in AIR Debug Launcher (mobile), and the SimController automatically launches.

Mobile interactions that can be tested in the SimController include:

- Using the accelerometer in three directions (x-, y-, and z-axis)
- Click and drag, press and tap, two-finger tap, pan, rotate, zoom, and swipe
- Geolocation
- Menu, Back, and Search buttons

Questions

The following are additional questions that are not in the students' Classroom in a Book.

1 What files are exported when you publish an HTML5 Canvas document?

2 How do you know if a particular Flash feature is supported in an HTML5 Canvas document?

3 It's recommend that you use classic tweens in an HTML5 Canvas document. How do classic tweens and motion tweens differ in the way they each handle motion on a custom path?

4 Which one of these interactions are NOT available in the SimController: geolocation, swipe, pinch, accelerometer, microphone?

5 How are projectors and an AIR application for the desktop similar, and how are they different?

Answers

1 An HTML5 Canvas document publishes two files: a JavaScript file that contains all the necessary JavaScript code to represent your images, graphics, symbols, animations, and sounds on the Stage, and an HTML file that displays the animation in the browser. It also outputs any dependent assets, such as images and sounds.

2 When you publish an HTML5 Canvas document, Flash warns you of any incompatible Flash features that will not export or work as expected in the Output panel. Also, Flash grays out and disables any tools that are not supported in an HTML5 Canvas document.

3 The path of motion for a motion tween appears in the Tween layer, while the path of motion for a classic tween must be created in a separate Guide layer and paired with the layer containing the classic tween.

4 The microphone function is not available in the SimController.

5 Both projectors and AIR applications for the desktop are self-contained applications that run on a user's Windows or Mac desktop, without needing a browser or Flash Player. AIR applications support more robust features, however, and require a code-signing certificate, which provides a way for users to authenticate your identity as a trusted developer.