**CEIT 225 INSTRUCTIONAL DESIGN**

**ADVANCED ANIMATION**

**Convert to Frame by Frame Animation**

Right Click to timeline of the animation that you want to convert, and Click to “Convert to Frame by Frame Animation”. This will convert your motion tween to frame by frame animation which will give you the flexibility of changing a specific spot of your animation.



**Reverse Frames**

Select the frames you want to revert and right Click to timeline and Click to “Reverse Frames”. This will revert the animation that you were created. You can use this function to create looping animations.

**Using and Saving Motion Presets**

You can apply default motion presets to your symbols by selecting the symbol (or shape) and select the preset and click to “Apply”



You can also save and use a motion that you have created before. To save the motion preset, you can either right click to symbol and select “Save as motion preset” or after selecting the symbol click the “Save selection as preset” from Motion Presets window. You can also export and import motion presets.



**3D**

“Use the 3D Rotation tool to rotate an object along any of its three axes and the 3D Translation tool to move an object along any of its three axes. For example, create a Star Wars-style opening scrolling text screen by rotating the text along its x-axis to tilt it, and then translating it along the y- and z-axes to have it disappear in the horizon. Create confetti that realistically tumble in 3D, or develop games with cards that flip as they are dealt. Your only limit is your imagination.” –From the Book *Flash Professional CS5 Advanced*

**Shape Hint**

Select the first keyframe of the shape tween, and choose Modify > Shape > Add Shape Hint. Move the first shape hint to a point on your shape. Select the last keyframe of the shape tween, and move the matching circled letter to a corresponding point on the end shape. This shape hint turns green and the first shape hint turns yellow, signifying that both have been moved into place correctly.

**Advanced Masking Examples**

You can use shape or motion tween in your mask layer. Other than that you can use alpha property to create a spotlight effect. You can do different examples with combination of those (Car windshield, spotlight etc.)

**Bone Tool:**

Bone tool that enables you to link symbols together quickly and easily in a parent/child relationship commonly referred to as inverse kinematics. The entire bone structure is also referred to as an armature. You can apply an armature to a series of movie clip symbols or to a raw vector shape that can then be manipulated across time by dragging the armature to a new pose.

***Example 1***

*Create two square on the stage and convert them into Movie Clips. Then select bone tool from toolbar and draw a bone from first square to second. The squares will be shown like the picture below. In the layers section, a new layer named armature is created. After that, it can be used as motion tween property. For example, click 20th frame and press F5. Then click fifth frame and change objects’ positions according to your animation. Prees Ctrl+Enter to test the animation.*

******

***Note: Bone tool can be used for raw vector shapes in the same way.***

***To see more information about bone tool and examples you can visit:***

<http://www.adobe.com/devnet/flash/articles/character_animation_ik.html>

**Introduction to ActionScript 3.0**

ActionScript is the programming language for the Adobe® Flash® Player and Adobe® AIR™ run-time environments. It enables interactivity, data handling, and much more in Flash, Flex, and AIR content and applications. Unlike the previous versions of ActionScript, this new version enables users to create complex applications with large data sets and object-oriented, reusable code bases.

**Variables & Constants**

A variable is a name that represents a value in the computer’s memory. To create a new variable, the parameter var has to be used. A variable can get different values through the application.

***Example***

*Var Adi: String*

A constant is similar to a variable. It is a name that represents a value in the computer’s memory with a specified data type. The difference is that a constant can only be assigned a value one time in the course of an ActionScript application. Once a constant’s value is assigned, it is the same throughout the application. To define a constant const parameter has to be used

***Example***

*Const pi: Number*

**Data Types**

 **Simple Data Types**

* String: a textual value, like a name or the text of a book chapter
* Numeric: ActionScript 3.0 includes three specific data types for numeric data:
	+ Number: any numeric value, including values with or without a fraction
	+ int: an integer (a whole number without a fraction)
	+ uint: an “unsigned” integer, meaning a whole number that can’t be negative
* Boolean: a true-or-false value, such as whether a switch is on or whether two values are equal

**Some Complex Data Types**

* MovieClip: a movie clip symbol
* TextField: a dynamic or input text field
* SimpleButton: a button symbol
* Date: information about a single moment in time (a date and time)

In ActionScript object-oriented programming, there are three types of characteristics that any class can include:

* Properties
* Methods
* Events

Properties

A property represents one of the pieces of data that are bundled together in an object.

***Examples***

*Star.x=100;*

*Square.rotation= Triangle.rotation*

*Circle.alpha=.1*

**Methods**

A method is an action that an object can perform.

 ***Examples***

 *MymovieClip.play*

*MymoviClip.stop*

**Events**

Events are things that happen that ActionScript is aware of and can respond to.

 ***Examples***

*function tasi(e:MouseEvent):void*

 *{*

*e.target.x+=5;*

*}*

*Mycircle.addEvetnListener(MouseEvent.CLICK, drag);*

**Commonly Used Operators**

|  |  |  |  |
| --- | --- | --- | --- |
| **+** | **Addition** | **==** | **Equality** |
| **-** | **Subtraction** | **!=** | **Inequality** |
| **\*** | **Multiplication** | **===** | **Strict equality** |
| **/** | **Division** | **!==** | **Strict inequality** |
| **%** | **Modulo** | **as** | **Checks data type** |
| **<** | **Less than** | **is** | **Checks data types** |
| **>** | **Greater than** | **In** | **Checks for object properties** |
| **<=** | **Less than or equal** | **instanceof** | **Checks prototype chain** |
| **>=** | **Greater than or equal** | **Void** | **Returns undefined value** |

**Using Code Snippets to add ActionScripts**

Code Snippets are the code pieces that Flash can add to a frame automatically.

***Example 1***

* Draw a rectangle on the stage and convert it to a Movie Clip (Press F8)
* Check whether Action Script window is active. If not, activate it from Window Menu
* Select the object that you want to add scripts
* In Action Script Tab, Code Snippets button is placed on right of the Action Script



* Select drag and drop property (Right Click and select “add to frame”)
* Press Ctrl+Enter to run application

***Example 2***

Use steps explained above and this time, select “Move with Keyboard Arrows” under Animation folder

***Note: Analyze these two examples carefully to understand main structure of writing Action Script 3.0***