Flash CS4 - Lab 3
Introduction to Classes:

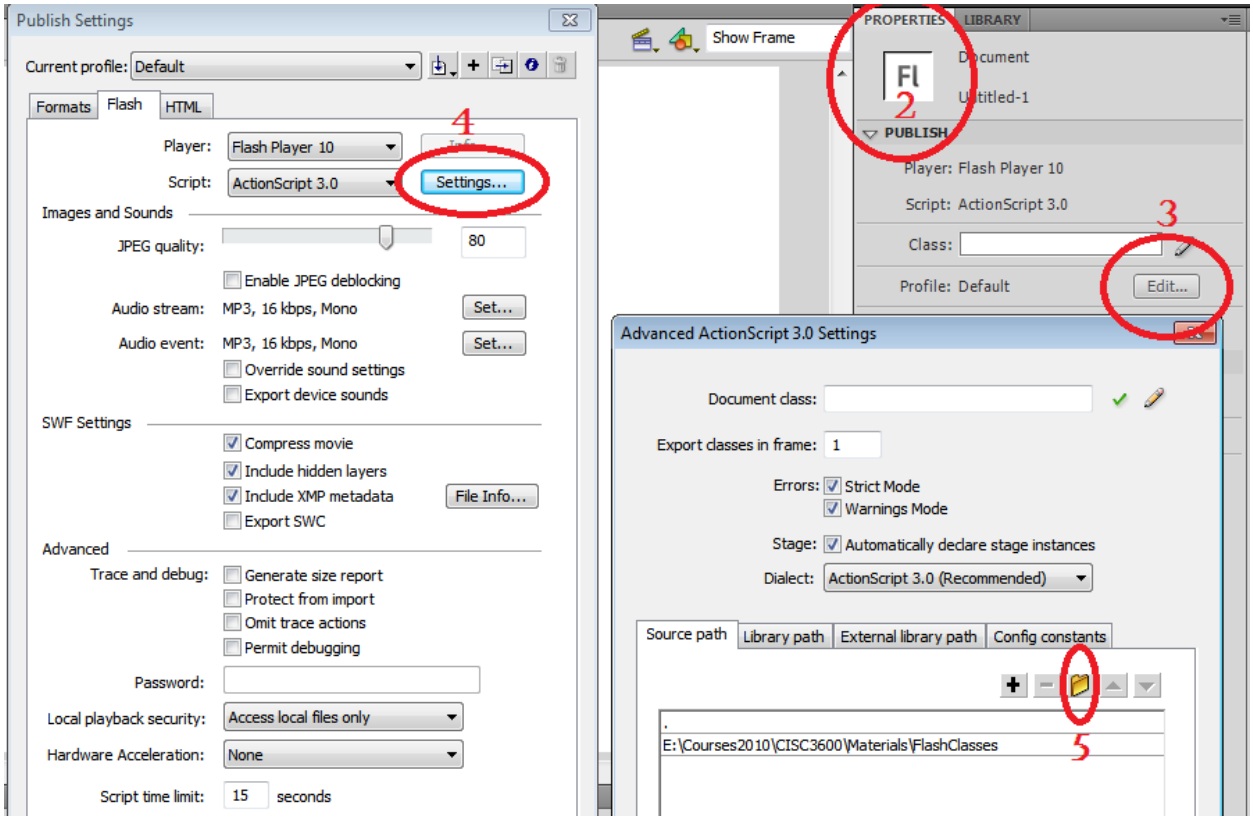
I. Setting (and resetting) the class path:

You will want to be able to use and reuse the classes that you create. You will also want to be able to carry those classes around on your USB. Therefore, for this class and this project, please take the following steps.

1. On your USB drive, create a FlashClasses folder to store your custom classes.
2. Create a new Flash document (or open the Flash document you are already working with) and go to the publish settings for that document.
3. Click on the Edit Button.
4. In the Publish Settings box that appears, select the Flash tab and then click on the settings button next to the ActionScript3.0 drop down box.
5. In the "Advanced ActionScript 3.0 Settings" window that appears, click on the "Source Path" tab, and on the button that looks like a file, and then select the folder (FlashClasses) that you created on your USB.

NOTE: You may need to redo steps 2-5 if you switch machines, or remove/replace your USB drive and it gets assigned a new drive letter. But following this method you should be able to share Class files.

Note: You can set a custom class path for ALL new Flash documents (inside Flash, go to Edit, Preferences.)



II. Creating & testing our first class.

1. Inside of Flash, click the File menu, choose "New" and then "Actionscript File." In the new Actionscript document's script pane, type the following:

```
package  edu.cisc3600.testing {
        public class TestClass {
            public function TestClass() {
                trace("TestClass working");
            }
        }
}
```

2. Note that the folders "edu," "cisc3600," and "testing" don't exist yet, but that's okay. We will create them in the process of saving the file. Go to File, Save As. In the Save As dialog box, first navigate to the folder you created earlier as a "portal" directory. Next, create in the directory a new folder called "edu". Inside the edu folder create a folder called "cisc3600", and inside cisc3600 create a folder called "testing." Double click testing to enter it. Finally, save the file as TestClass.as.
3. Let's put the TestClass to the test and make sure it works. In the flash file that you opened, click on the first frame of the timeline, and press F9 to go to the actions panel. Type the following two lines:

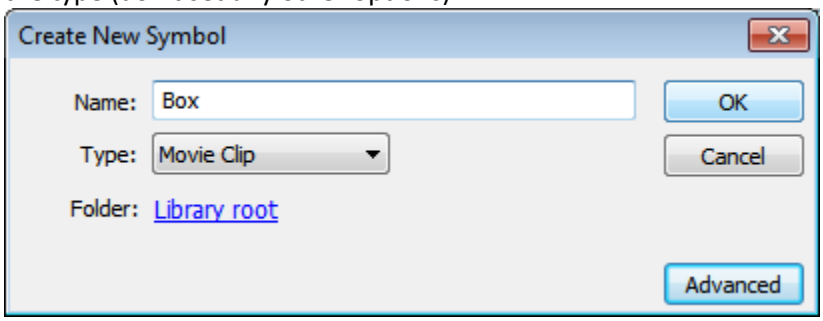
```
import edu.cisc3600.testing.TestClass;
var testClass:TestClass = new TestClass();
```

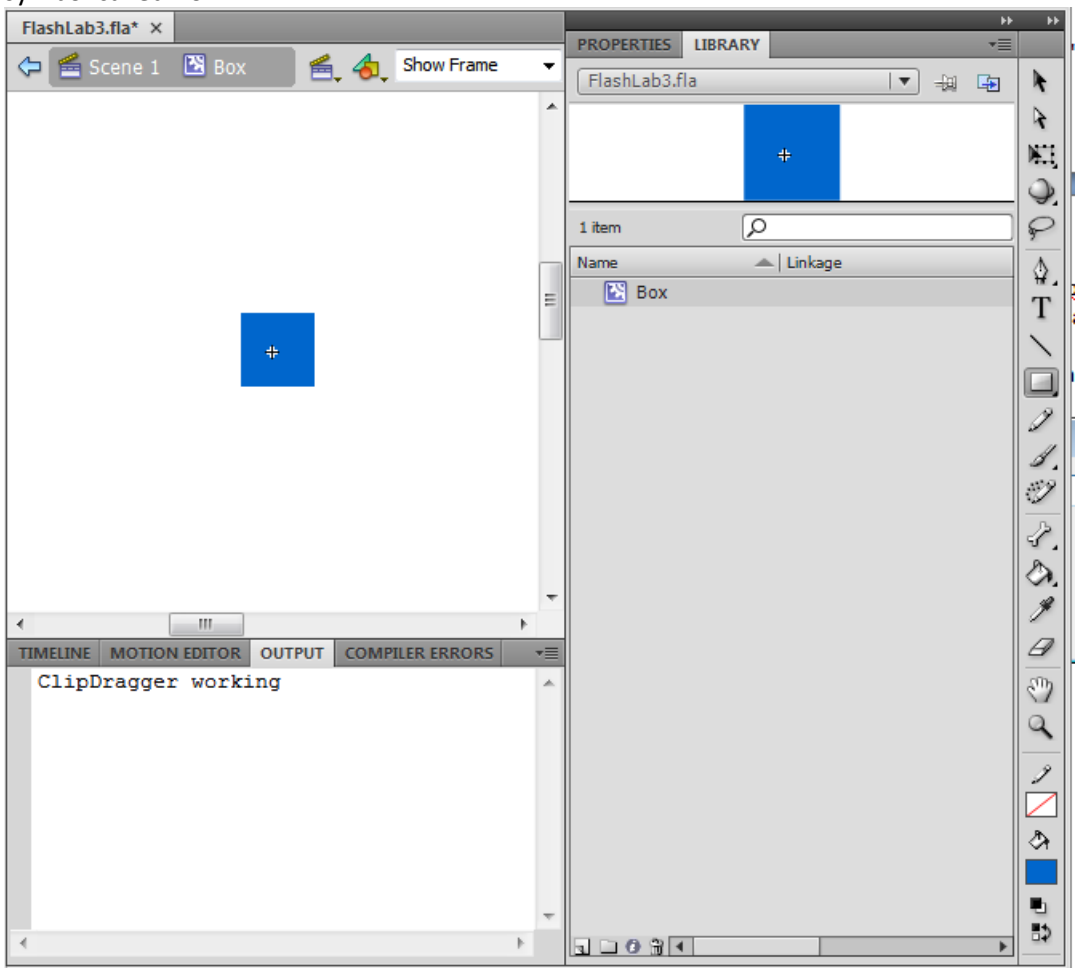
4. Press CTRL-Enter to test the movie. You should immediately get a message traced to the output window saying “TestClass working.”

III. A useful class: "ClipDragger"

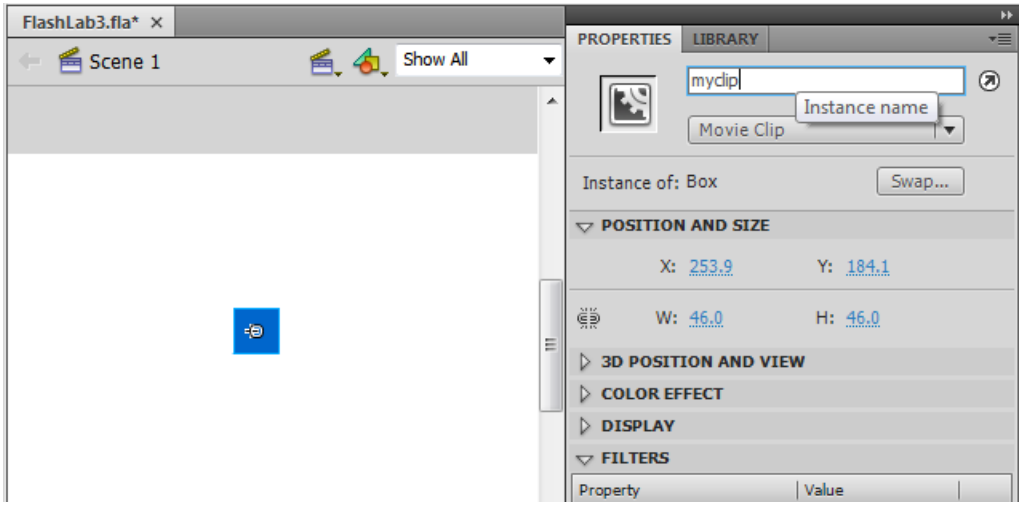
1. Inside of Flash, click the File menu, choose "New" and then "Actionscript File." In the new Actionscript document's script pane, type the following:

```
package edu.cisc3600.mouse {
    import flash.display.MovieClip;
    import flash.events.MouseEvent;
    public class ClipDragger {
        private var _clip:MovieClip;
        public function ClipDragger(clip:MovieClip) {
            _clip = clip;
            _clip.buttonMode = true;   // Will display a hand when moused over.
            _clip.addEventListener(MouseEvent.MOUSE_DOWN, drag);
            _clip.stage.addEventListener(MouseEvent.MOUSE_UP, drop);
        }
        private function drag(event:MouseEvent) {
            _clip.parent.setChildIndex(_clip, _clip.parent.numChildren - 1);
            // Parent is the stage, 0 is lowest child, so put _clip on top.
            _clip.startDrag();
        }
        private function drop(event:MouseEvent) {
            _clip.stopDrag();
        }
    }
}
```

2. In the cisc3600 folder, create a new folder called "mouse". Double click mouse to enter it. Finally, save the file as ClipDragger.as.
3. Our new class is expecting to be passed a MovieClip as a paramenter. So we will need to create a new MovieClip symbol, draw something in it, place it on the stage, and then give it an instance name of "myClip" in the properties panel.
    a. Go to Insert/Symbol. In the box that appears name your symbol "Box", select "Movie Clip" as the type (don't set any other options).
    b. 

    c. You should now see a Box symbol in the Library tool-bar. And the edit bar should show that you are editing the box. Use the rectangle tool to quickly draw a box in the center of the movie clips symbol called Box.
    d. 

    e. Go to the Edit Bar and switch to Scene 1. Then drag a copy of the Box from the library to the center of the stage. Switch to the Properties tab in the Panels window and give the box you just dragged onto the stage the instance name of myclip.

f. 

4. Click on the first frame of the timeline, and press F9 to go to the actions panel. Type the following two lines:

```
import edu.cisc3600.mouse.ClipDragger;
var clipDragger:ClipDragger = new ClipDragger(myclip);
```

5. Press CTRL-Enter to test the movie. And you should be able to click on the box and move it.

III. A useful class: "KeyMover"

1. Inside of Flash, click the File menu, choose "New" and then "Actionscript File. Save the actionscript file as KeyMover.as in the following directory structure.
   ... (portal location on USB) ... \FlashClasses\edu\cisc3600\keyboard\KeyMover.as
2. Add the following text to the KeyMover.as file and save it again.

```
// The package name should match the folder layout from the portal link you created.
package edu.cisc3600.keyboard{
        // Familiarity with the class libraries will come with time.
        // But what do you think each of these libraries do?
        import flash.display.MovieClip;
        import flash.events.KeyboardEvent;
        import flash.ui.Keyboard;
        import flash.events.Event;

        // This is the class we are creating.
        public class KeyMover {
                private var _clip:MovieClip; // We will pass in a movie clip.
                private var rightArrow:Boolean;
                private var leftArrow:Boolean;
                private var upArrow:Boolean;
                private var downArrow:Boolean;
                private var speed:uint = 10; // How far should the clip move.

                // Constructor
                public function KeyMover(clip:MovieClip) {
                        _clip = clip;

                        // Every display object (of which our MovieClip _clip is one)
                        // has a property called "stage" that's an instance of the Stage class.
                        // In other words, every display object "knows" what stage it's on.
                        // Our _clip is for events on the stage that it belongs too.
```

```
                _clip.stage.addEventListener(KeyboardEvent.KEY_DOWN, keyPressed);
                _clip.stage.addEventListener(KeyboardEvent.KEY_UP, keyReleased);
                // Note, we could add the following line here, but we won't.
                // _clip.stage.addEventListener(Event.ENTER_FRAME, update);
        }
        private function keyPressed(event:KeyboardEvent) {
                if (event.keyCode == Keyboard.RIGHT) rightArrow = true;
                if (event.keyCode == Keyboard.LEFT) leftArrow = true;
                if (event.keyCode == Keyboard.UP) upArrow = true;
                if (event.keyCode == Keyboard.DOWN) downArrow = true;
        }
        private function keyReleased(event:KeyboardEvent) {
                if (event.keyCode == Keyboard.RIGHT) rightArrow = false;
                if (event.keyCode == Keyboard.LEFT) leftArrow = false;
                if (event.keyCode == Keyboard.UP) upArrow = false;
                if (event.keyCode == Keyboard.DOWN) downArrow = false;
        }
        // Notice how this function is public.
        public function update(event:Event) {
                if (rightArrow) {
                        _clip.x += speed;
                        if (_clip.x > _clip.stage.stageWidth) {
                                _clip.x = -(_clip.width);
                        }
                }
                if (leftArrow) {
                        _clip.x -= speed;
                        if (_clip.x < -(_clip.width)) {
                                _clip.x = _clip.stage.stageWidth;
                        }
                }
                if (upArrow) {
                        _clip.y -= speed;
                        if (_clip.y < -(_clip.height)) {
                                _clip.y = _clip.stage.stageHeight;
                        }
                }
                if (downArrow) {
                        _clip.y += speed;
                        if (_clip.y > _clip.stage.stageHeight) {
                                _clip.y = -(_clip.height);
                        }
                }
        }
    }
}
```

3. In the flash file that you have opened, click on the first frame of the timeline, and press F9 to go to the actions panel. Type the following lines:

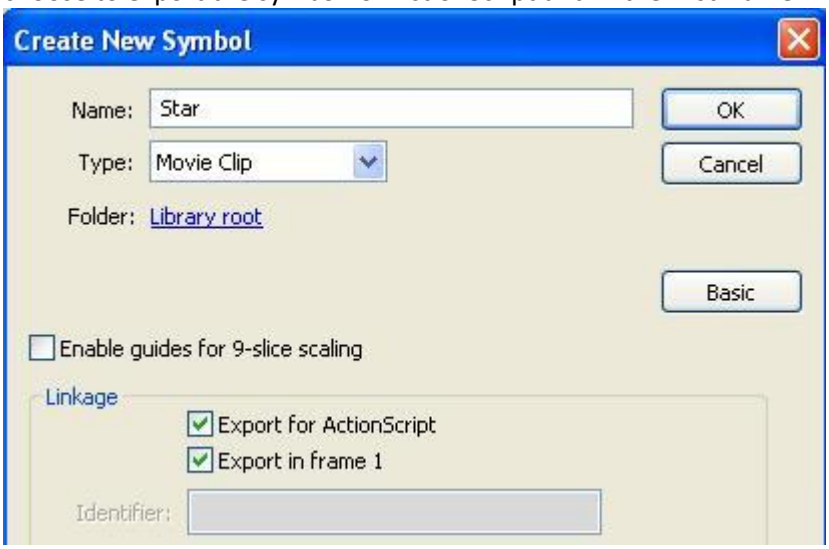
```
import edu.cisc3600.mouse.ClipDragger;
var clipDragger:ClipDragger = new ClipDragger(myclip);
import edu.cisc3600.keyboard.KeyMover;
var keyMover:KeyMover = new KeyMover(myclip);

stage.addEventListener(Event.ENTER_FRAME, keyMover.update);
```

4. Now hit Ctrl + Enter and see what happens.

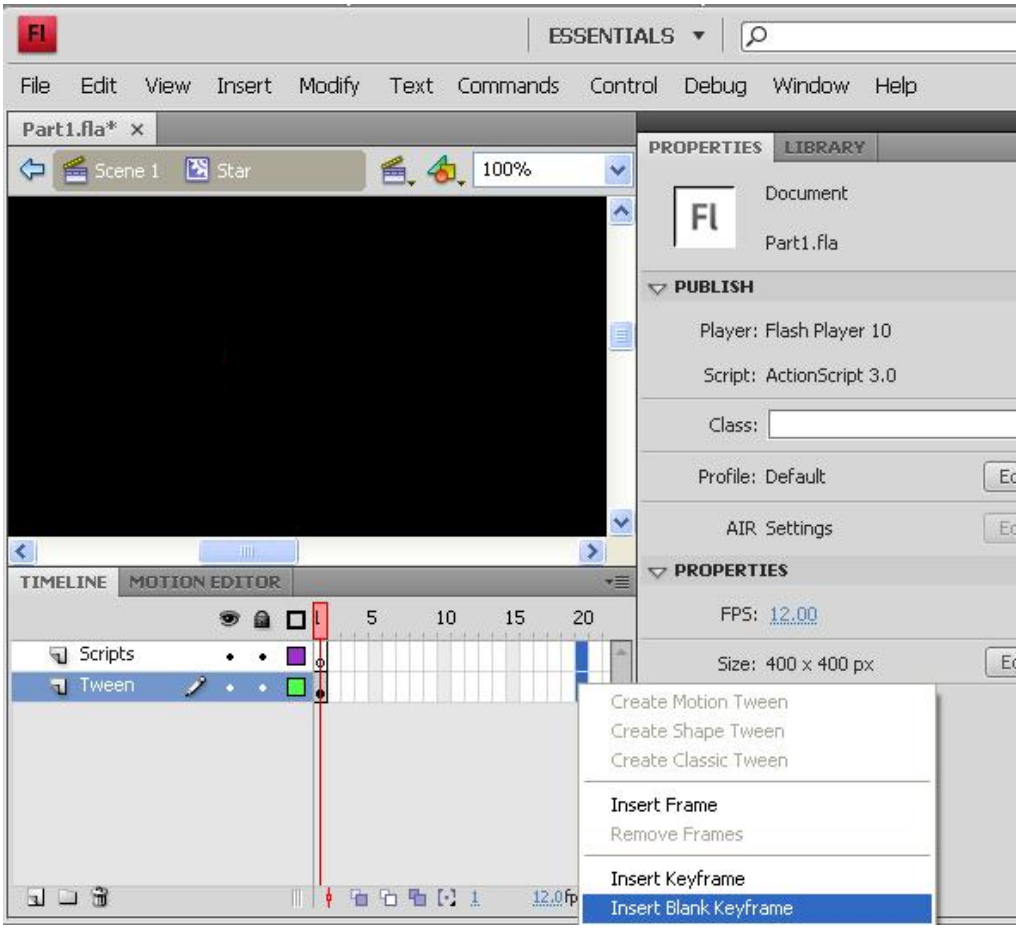
III. A useful class: "AnimatedObject" class (think object factory).

<The beginning of this section is taken from Lab II: Movie Clips and Instances>

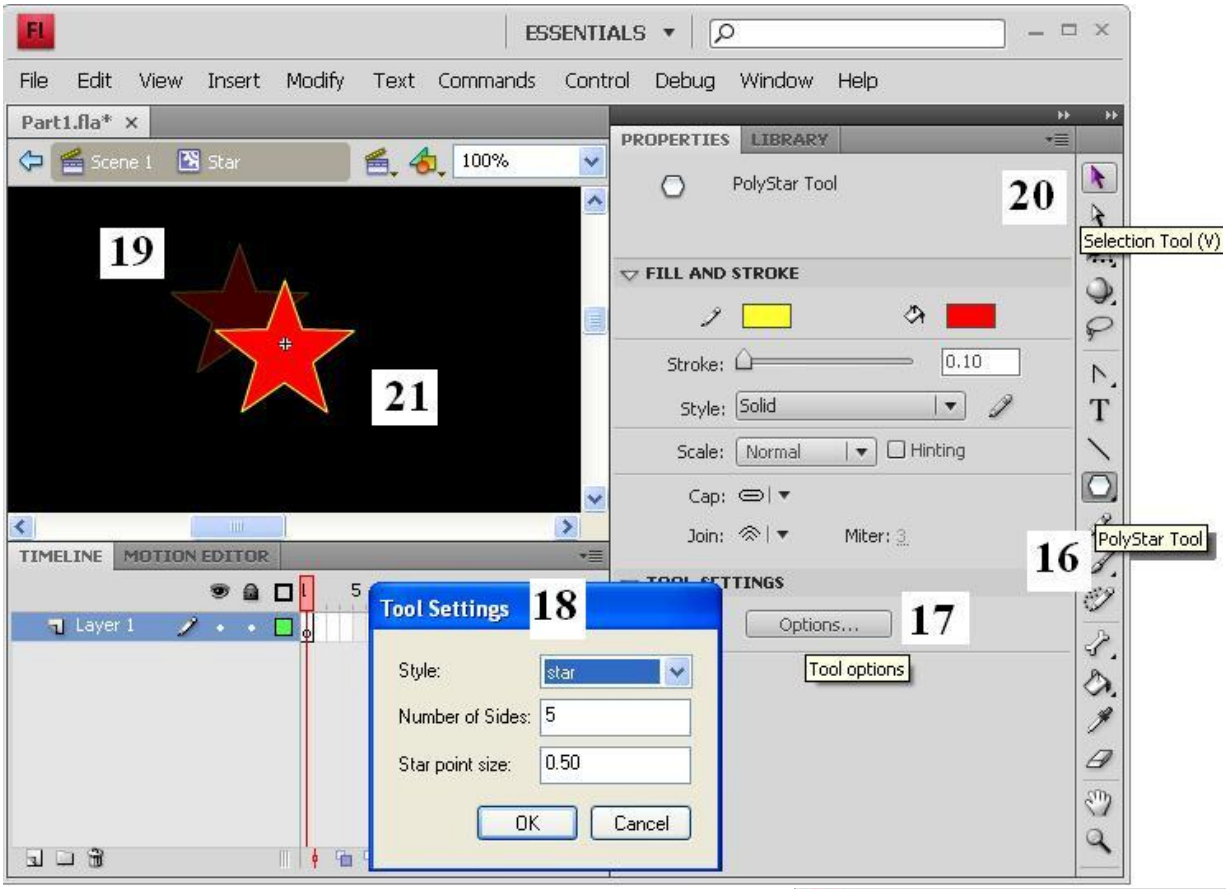
1. Go to Insert/Symbol. In the box that appears name your symbol "Star", select "Movie Clip" as the type and choose to export the symbol for ActionScript and in the first frame.



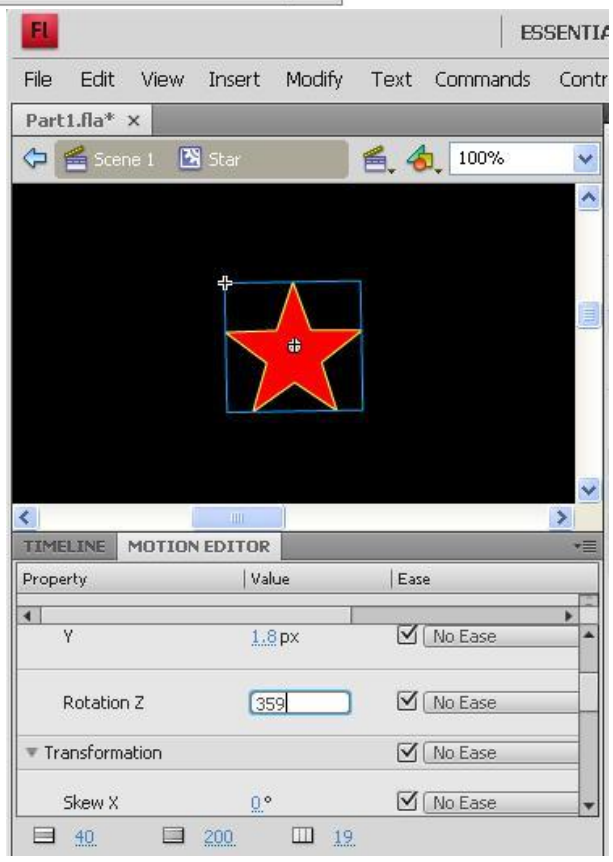
2. Add a new layer to the timeline for your "Star" Movie Clip. Name the two layers "Scripts" and "Tween". Then Select the 20th frames of both of your layers, right-click and choose "Insert Blank Keyframe".

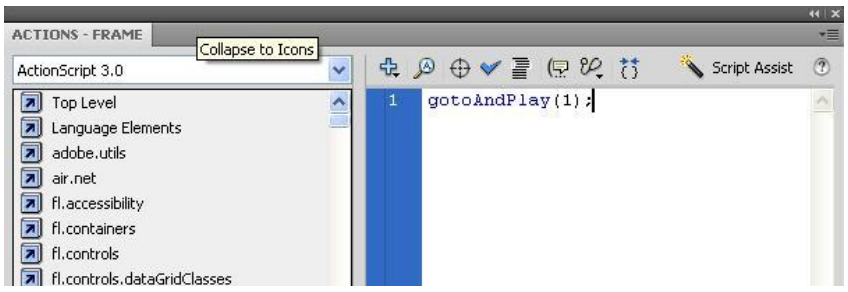


3. Now select the first frame of your "Tween" layer.
4. Select the PolyStar Tool.
5. Select the "Options..." button.
6. Set the Style to star, the Number of Sides to 5 and the Star point size to 0.50.



7. Create a star in the stage area.
8. Switch to the Selection Tool.
9. Center the star in the center of the stage area.
10. Select the 19th frame of your "Tween" layer. Right click and choose "Create Motion Tween".
11. The Tab next to the "Timeline" tab is called "Motion Editor". Click on the motion editor tab.
12. Find the row labeled "Rotation Z" and change the 0 to 359.
13. Go back to the Timeline and test your Movie Clip by going to Control/rewind and then Control/play.
14. Select the 20th frame of the Scripts layer, right-click and choose "Actions".
15. In the window that appears type: gotoAndPlay(1);
16. Close the actions window.
17. Click on the Scene 1 icon in the edit bar to return to Scene 1.





<Now you have a Star with animation, exported as a class>

18. Inside of Flash, click the File menu, choose "New" and then "Actionscript File. Save the actionscript file as KeyMover.as in the following directory structure.
    ... (portal location on USB) ... \FlashClasses\edu\cisc3600\gameobjects\AnimatedObject.as
19. Add the following text to the AnimatedObject.as file and save it again.

```
// The package name should match the folder layout from the portal link you created.
package edu.cisc3600.gameobjects{
        // Familiarity with the class libraries will come with time.
        // But what do you think each of these libraries do?
        import flash.display.*;
        import flash.display.MovieClip;
        import flash.events.KeyboardEvent;
        import flash.ui.Keyboard;
        import flash.events.Event;
        import flash.utils.getDefinitionByName;

        // This is the class we are creating.
         public class AnimatedObject extends MovieClip {
                private var _stage:MovieClip;  // A reference to the Stage. We will need to pass this in.
                private var _clip:MovieClip;    // Our object.
                private var speed:uint = 10;    // How far should the clip move.
                private var className:String; // The name of the class we are using.
                private var classRef:Class;     // We will use this to capture the Class type

                // Constructor
                public function AnimatedObject( s:MovieClip, MovieClipClassName:String, xPos:uint, yPos:uint ) {
                        _stage = s;
                        className = MovieClipClassName;
                        classRef = getDefinitionByName(className) as Class;

                        _clip = new classRef(); // Turn the class into a MovieClip
                        _clip.x = xPos;
                        _clip.y = yPos
                        _stage.addChild(_clip);
                        _clip.stop();       // We don't want the clip to play right away.

                        trace("AnimatedObject class working");
                }

                public function update(event:Event) {
                        _clip.x += speed;
                        if (_clip.x > _clip.stage.stageWidth) {
                                _clip.x = -(_clip.width);
                        }
                }
                public function animate(event:Event) {
                        _clip.nextFrame();
                        // Could also use _clip.play(); but would need to add .stop() somewhere else.
                }
         }
}
```

5. In the flash file that you have opened, click on the first frame of the timeline, and press F9 to go to the actions panel. Type the following lines:

```
import edu.cisc3600.mouse.ClipDragger;
var clipDragger:ClipDragger = new ClipDragger(myclip);
import edu.cisc3600.keyboard.KeyMover;
var keyMover:KeyMover = new KeyMover(myclip);
```

```
stage.addEventListener(Event.ENTER_FRAME, keyMover.update);

// Everything above is from the previous part of the lab.

import edu.cisc3600.gameobjects.AnimatedObject;
var animatedObject:AnimatedObject = new AnimatedObject(this, "Star", 100, 150);

stage.addEventListener(Event.ENTER_FRAME, animatedObject.update);
stage.addEventListener(Event.ENTER_FRAME, animatedObject.animate);
```

20. Hit Ctrl + Enter and see what happens. Then try commenting out the last line:
    stage.addEventListener(Event.ENTER_FRAME, animatedObject.animate);
    and running it again to see what happens.

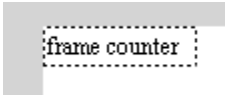
IV. Useful Class FrameRateTimer

1. Inside of Flash, click the File menu, choose "New" and then "Actionscript File. Save the actionscript file as FrameRateTimer.as in the following directory structure.
   ... (portal location on USB) ... \FlashClasses\edu\cisc3600\timers\ FrameRateTimer.as
2. Add the following text to the FrameRateTimer.as file and save it again.

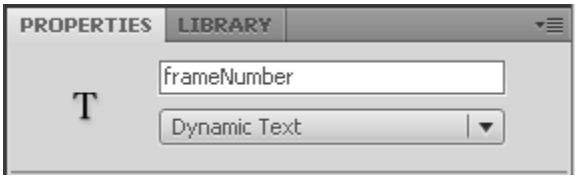
```
package edu.cisc3600.timers {
    import flash.events.Event;
    import flash.display.MovieClip;

    public class FrameRateTimer extends MovieClip {
        var _stage:MovieClip; //We need this to pass in a reference to the stage.

        var frame:int;
        var fps:int;
        var timeinit:Date;
        var time:Date;
        var lasttime:int;
        var timepassed:int;

        public function FrameRateTimer( s:MovieClip) {
            _stage = s;

            frame = 0; //initialize fps to 30
            fps = 30;
            timeinit = new Date;
            lasttime=timeinit.getMilliseconds(); //initialize lasttime to the current time

            addEventListener(Event.ENTER_FRAME,enterFrameHandler);
        }

        protected function enterFrameHandler(event:Event):void {

            time = new Date;
            //on each frame, figure out how much time has passed by
            // comparing the milliseconds of the last frame to
            //the milliseconds of the current frame
            timepassed=((time.getMilliseconds()-lasttime)>=0)?(time.getMilliseconds()-
lasttime):(1000+(time.getMilliseconds()-lasttime));
            //convert the time passed between frames to frames per second.
            //Round it to the nearest tenth.
            fps=Math.round(10000/timepassed)/10;
```

```
                    //set last time for the next frame.
                    lasttime=time.getMilliseconds();

                    frame++;
                    _stage.frameNumber.text = "Frame: " + frame + " & FPS: " + fps;
                }

        } // end of class FrameRateTimer

    } // end of package code
```

3. We need to add the text field that gets updated in the main loop. In the flash file, choose the text tool and drag a field onto the stage. Use the selection tool to select the text field. I typed "frame counter" into my control on the stage so that I know what that field is used for at a glance.



4. With the field selected, go to the properties tab. Set the name to "frameNumber" (the name we used in code above) and the type to "Dynamic Text."



6. In the flash file that you have opened, click on the first frame of the timeline, and press F9 to go to the actions panel. Add the following lines (to what is already there):
   import edu.cisc3600.timers.FrameRateTimer;
   var frameRateTimer:FrameRateTimer = new FrameRateTimer(this);
5. Hit Ctrl + Enter and see what happens.

IV. Useful Class MainGameLoop

Well I am afraid that this class doesn't exist yet. It will be something I will try and get around too next year. Unless of course, you want to create it ☺!

I have given you all of the pieces above. Think about what you would need to create. You would like to create a class (and then an associated object) that allowed you track whether your framerate was running too fast or too slow for your game.

If the frame rate is running too fast, you would like to like to pause the actions of your players and player objects for just a fraction of a second so that they game doesn't appear to fast and harder than it was designed to be.

If the frame rate is running to slow, then you would like to be able to turn off some functionality (like animating some objects) in order to help the game to run a little faster.

Can you see the solution? Good, go code it.