

CISC 110 Lab 9

In this lab, you will practice defining functions for buttons and will experiment with the index levels of objects.

Complete the following during your lab time.

1. Download and open the Flash file called Lab9Start fla, and then test the movie to see what you can do so far (you can only see the button states change and type in a character in the text field). This file contains:
 - Three button symbols with instance names **orangeBtn**, **pepperBtn** and **removeBtn**
 - Two static text fields
 - One input text field with instance name **numberBox**
 - Two graphic symbols in the library called **orange** and **pepper**, each of which contains a bitmap that was imported and is also in the library. These graphic symbols were converted to buttons to create the corresponding button symbols.
2. Convert the two graphics symbols, **orange** and **pepper**, to movie clips and create classes for them, so you can create and display Orange and Pepper objects dynamically. To do this:
 - Open the Library Panel and select the graphics symbol **orange**
 - Select the Symbol Properties button (the “i” symbol at the bottom of the Library Panel)
 - In the resulting dialog, set the type to MovieClip, check the Export for ActionScript option, and add **Orange** as the class name in the class field. Select ok and ignore the warning if it appears, which is just telling you Flash is creating a class for you, which is what you want.
 - Follow the same process to convert **pepper** to a movie clip and create a class for it called **Pepper**.
3. Try out your new classes by writing ActionScript to create two new objects, one of type **Orange** and one of type **Pepper** and use the **addChild** method to add them to the display list; then test your movie to see them appear. Recall that to create a new object, you use the syntax:

```
var <variable name> <Class name> = new <Class name> ( );
```

4. Write a function that will display any object at a specified x and y position. It sets the x and y properties of the object to the parameters given and then calls the addChild method to add it to the display. Here is the header line for your function:

```
function placeObject( obj: DisplayObject, x: int, y: int)
```

5. Test your function on the two orange and pepper variables you have created.

Next you will add functionality to the buttons. You will make **orangeBtn** create a new **Orange** object each time the button is clicked and place it using the **placeObject** function. Similarly you will make **drawPepper** create and place a new **Pepper** object each time it's clicked. You will make **removeBtn** remove an object each time it's clicked. Test your movie after you add each button handler function.

6. Add three MouseEvent.CLICK listeners: to orangeBtn that will call function drawOrange, to pepperBtn that will call function drawPepper, and to removeBtn that will call function removeObject.
7. Define two variables, called **orangeInc** and **pepperInc** and set them both to 0. You will use these variables to place each new orange and pepper offset from the previous one. Each time you place a new object, you'll increment **orangeInc** or **pepperInc**.
8. Write the **drawOrange** function:
 - Create a new Orange object
 - Call the placeObject function for the new object, with both the x and y position set to $20 * \text{orangeInc}$
 - Increment orangeInc
9. Similarly write the **drawPepper** function, except:
 - Create a new Pepper object
 - Set the x position to $60 + 40 * \text{pepperInc}$ (so it won't be on top of any oranges)
 - Set the y position to $40 * \text{pepperInc}$
 - Increment pepperInc
10. Write **removeObject** function with the single statement: `removeChildAt(0);`

Now your movie should draw oranges on top of each other as you press the **orangeBtn** and peppers on top of each other as you press the **pepperBtn** and should remove them in the order you created them. Note that the remove button will also remove the objects that were created at author time if you keep pressing it (the text fields and the buttons). Next you will modify your script to allow you to add objects at different levels.

11. Modify your **placeObject** function. Place the following line in your function, which reads a digit typed by the user in the **numberBox** text field, converts it to a **Number**, and stores it in a variable called **n**. Then use the **addChildAt** method instead of the **addChild** method to place the object at index level **n** in the display list.

```
var n: int = Number(numberBox.text);
```

12. Modify your **removeObject** function by adding that same line to it and then removing the child at index level **n** instead of the child at level 0.
Experiment with your movie to see how levels are assigned in Flash.