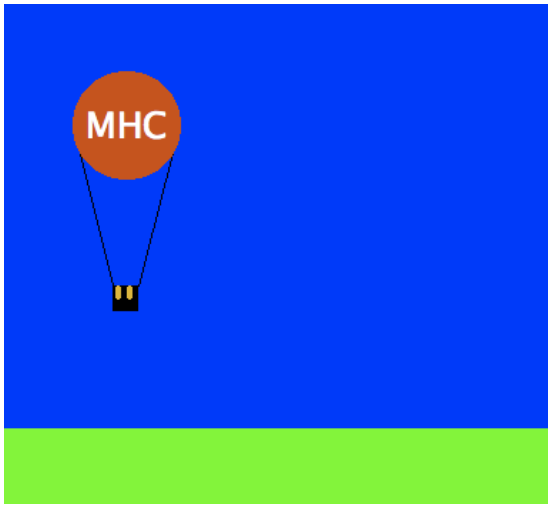
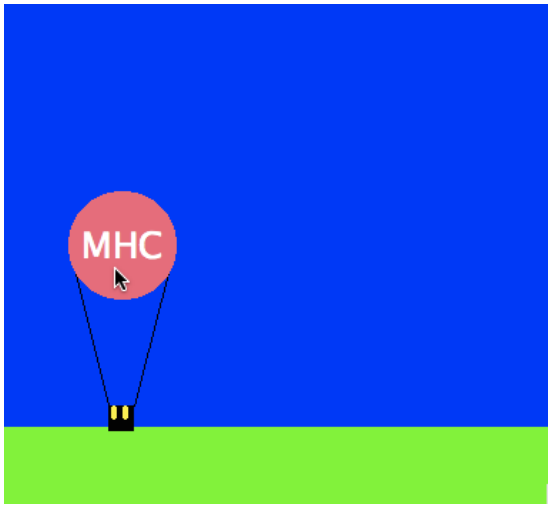
Computer Science Department

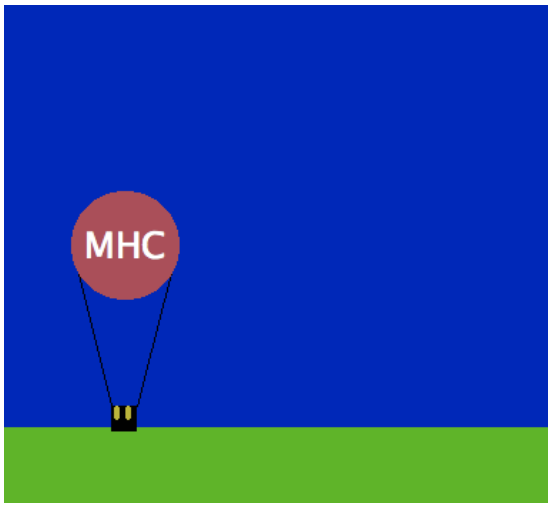
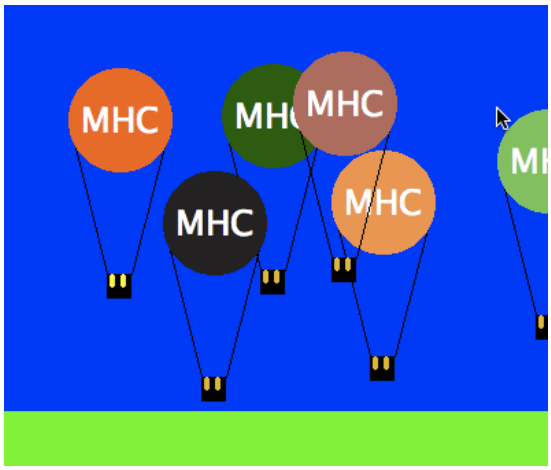
Mount Holyoke College

CS 101 - Problem Solving and Structured Programming
Lab 6 - Riding the MHC Balloon
Due: November 4/5

In this lab, you will create a drawing of a hot air balloon that the user can cause to rise and fall. The user will be able to drag the mouse vertically only. All horizontal mouse motion should be ignored. When the user clicks the mouse a new hot air balloon will be drawn, where the user clicks. The user will only be able to move the last balloon that is created. When the user moves the mouse out of the window, the scene will darken. When the user returns the mouse to the window, the scene will brighten.

Here are some snapshots from this program.

	The initial screen when the program starts
	Here, the user has dragged the balloon down.

	The user moved the mouse out of the window, causing the scene to get darker.
	The user has clicked the mouse several times in different places to create new balloons. Notice that the balloons have a random color.

The Program Design

This project will have 2 classes: BalloonController and HotAirBalloon.

The BalloonController class

The BalloonController is the class that extends WindowController. It is responsible for defining the begin method and the event handlers.

- begin method - This method should draw the background consisting of the grass and sky. It should also construct one HotAirBalloon.
- onMouseClick method - This method should construct a new HotAirBalloon where the user clicks.
- onMousePress / onMouseDrag / onMouseRelease - These methods should provide dragging code to allow the user to reposition the HotAirBalloon. Remember that the balloon should only go up and down. It should not move left and right at all.
- onMouseExit - This method should darken the scene.
- onMouseEnter - This method should restore the scene to its original colors.

The BalloonController class needs instance variables to hold the sky, grass, and HotAirBalloon so it can manipulate them. It also needs the usual instance variables to allow dragging.

The HotAirBalloon class

The HotAirBalloon is drawn using several shapes as you did with the penguin. You should define constants for the sizes of the shapes, relating those sizes with equations like you did with the parts of the penguin. Each time that you create a balloon, you should generate a random color for the balloon.

You will need to deal with the location in a different way, however, since you want to create balloons where the user clicks, rather than always in the same location. As a result, you will need to pass the location information as parameters to the HotAirBalloon constructor, instead of using constants for the locations like you did with the penguin. The location parameters should tell the constructor where the balloon part of the balloon should go and use equations to determine where to place the basket and lines to draw the rest of the balloon.

You will need instance variables to hold the parts of the balloon so that you can manipulate them later.

In addition to the constructor, the HotAirBalloon class must provide methods to:

- drag the balloon up and down. You will need two methods: contains and move. These will be called from onMousePress and onMouseDrag, respectively.
- darken the balloon. This is called from onMouseExit in BalloonController.
- restore the original color. This is called from onMouseEnter in BalloonController.

Documentation

Please remember that there is documentation on all the objectdraw methods available in Appendix B of the text book as well as online at
http://www.mtholyoke.edu/~blerner/cs101/objectdrawJavadocV1.1.2/index.html.

Writing the Program

Step 1: Creating the initial display

Start by writing the begin method in BalloonController to draw the sky and grass.

Next, write the HotAirBalloon class, containing just a constructor. Initially, you can draw the balloon at a fixed location. The balloon only requires the basket, lines and actual balloon. Adding sandbags (my son's idea!) or text on the balloon is extra credit. Save that for last if you want to do it.

This portion of the assignment should feel very similar to what you did for the penguin lab.

Step 2: Giving the balloon a random color

A balloon should have a random color. You will need a random number generator, ranging in value from 0 to 255 to generate each of the red, green, and blue values used in your color. Refer to the Draw With Random Colors example from October 16 to refresh your memory on how to do this.

If you like, you can restrict the range used for any of the 3 components if you want to limit the colors produced. Remember that large numbers give lighter colors, while small numbers give darker colors.

Step 3: Dragging the HotAirBalloon

Dragging the HotAirBalloon requires similar code to code that we looked at in class to drag the train. You will need to decide if the user has selected the HotAirBalloon. This requires you to define a contains method in the HotAirBalloon class, like we did with the train.

To do the actual movement, you will need to provide a move method in the HotAirBalloon class. To keep the HotAirBalloon moving only vertically, you should ignore the x component of the mouse's motion when calculating how far to move the HotAirBalloon. As a result, the HotAirBalloon's move method only needs a vertical offset. Again, this is similar to what we did to drag the train in class.

Remember that to move the balloon, the move method that you define simply **delegates** to its parts, telling each of them to move.

Step 4: Creating additional hot air balloons

Next, you should modify your program so that the user can create additional balloons.

To do this, you need to provide an onMouseClick event handler. The event handler should call the HotAirBalloon constructor passing in the location where the new balloon should be created. This will require you to modify your HotAirBalloon constructor to include parameters for the top and left coordinates that should be used to determine where to draw the balloon.

Note that you will also need to change the call to the constructor that you already have to pass in top and left arguments to the constructor for the position of the initial balloon drawn.

When your program has more than one balloon, the user will only be able to drag the last balloon created. (Also, only the last balloon created will darken after you complete Step 5.)

Step 5: Darkening the scene

To darken the scene, you need to change the color of all the objects in the scene that are not already black. You could do this by running a color chooser and finding darker colors to use. A simpler way, however, is to call a method provided by Java:

```
public Color darker()
```

This method creates a new Color that is a darker version of the Color it is called on.

You should use it as shown in this example. In BalloonController, you should have a constant GRASS_COLOR:

```
private static final Color GRASS_COLOR = Color.GREEN;
```

You can pick a different shade of green if you like. Whatever shade you choose, you should next have a second constant to represent a darker shade.

```
private static final Color DARK_GRASS = GRASS_COLOR.darker();
```

To change the color of an object, call setColor:

```
grass.setColor (DARK_GRASS);
```

The HotAirBalloon class should declare a method called darker that takes no parameters. You will then call this method to make the balloon darker.

Note that the balloon has a random color, so you will need to first get the color of the balloon before making it darker.

Darkening should happen in the onMouseExit event handler.

Step 6: Restoring the original colors

To restore the original colors, you will define a method in HotAirBalloon that calls setColor on all the objects, this time calling the predefined method

```
public void brighter ()
```

You need to carry out similar steps to what you did to darken the scene, this time brightening it instead.

Brightening should happen in the onMouseEnter event handler.

Extra credit: Decorating the balloon

If you would like, you can add decorations to your balloon, such as the sand bags or text in the demo version of the program.

Style Issues

Constant declarations

Like you did last week, be sure to use constant names (like BALLOON_SIZE) in your program rather than literal values (like 10) whenever possible. Also, define constants using equations based off of earlier constants when logical relationships exist between the constants.

Grading

5	Step 1: Drawing the initial display
10	Step 2: Giving the balloon a random color
10	Step 3: Dragging the HotAirBalloon
10	Step 4: Creating additional hot air balloons
5	Steps 5 & 6: Darkening the scene and restoring the original colors
10	Comments
10	Constant declarations
5	Indentation
10	Naming
5	Extra credit: decorating the balloon

Turning in Your Work

Go to ella. Click on COMSC 101 in the toolbar menu across the top. Then click on Assignments in the left column. Click the submit link for this assignment. Click the Add Attachment button. Use the Browse… button to upload a local file. You only need to submit the 2 Java files (BalloonController.java and HotAirBalloon.java), not the other files created by BlueJ. Click Continue. Then click the Submit button.