

Programming Assignment 3 (100 Points)

Due: 11:59pm Thursday, October 23
START EARLY!

This programming assignment has two programs:

- 1) a Java application (Triangles) that displays alternating triangle patterns side-by-side on the console outputting individual '*' and ' ' (space) characters using nested while loops, and
- 2) a Java applet extending the Mickey program from PA2 to be more Object-Oriented and add some functionality.

START EARLY!

README (10 points)

You are required to provide a text file named **README**, NOT Readme.txt, README.pdf, or README.docx, etc. with your assignment in your pa3 directory. There should be no file extension after the file name "**README**". Your README should include the following sections:

Program Description (3 points) :

Explain how the user can run and interact with each program. What sort of inputs or events does it take and what are the expected outputs or results? How did you test your program? How well do you think your program was tested?

Write your README as if it was intended for a 5 year old or your grandmother. Do not assume your reader is a computer science major. **The more detailed the explanation, the more points you will receive.**

Short Response (7 points) : Answer the following questions:

Vim related Questions:

1. How do you jump to a certain line number in your code with a single command in vim?
For example, how do you jump directly to line 20? (Not arrow keys)
2. What command sets auto-indentation in vim? What command turns off (unsets) auto-indentation in vim?
3. What is the command to undo changes in vim?
4. In vim, in command mode, how do you move forward one word in a line of code?
Move back one word? (Not arrow keys)

Unix/Linux related Questions:

5. How can you remove all .class files in a Unix directory with a single command?
6. How do you remove a Unix directory? Define the commands you would run on the terminal when the directory is empty and when it is not empty. How do these command(s) differ?
7. What is the command to clear a Unix terminal screen?

STYLE (20 points)

Please see previous programming assignment for details on Coding Style.

Example 2:

```
java Triangles
```

```
Enter the size of the triangles to display: -1
Triangle size must be > 1; Try again.
```

```
Enter the size of the triangles to display: 0
Triangle size must be > 1; Try again.
```

```
Enter the size of the triangles to display: 5
```

```
*      *      *      *
**     *      *      **
***    *      *      ***
****   *      *      ****
***** *      *      *****
```

Program 2 – Mickey:

This program will build off of your last Mickey program, so you should make sure that program is working correctly first. Being more Object-Oriented, we will separate the GUI controller and the Mickey object.

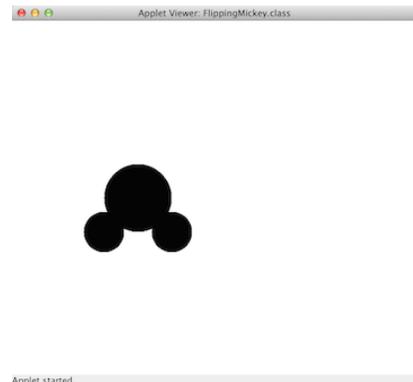
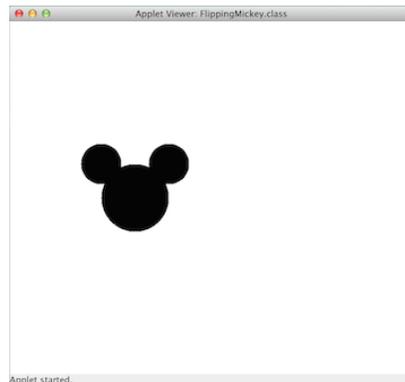
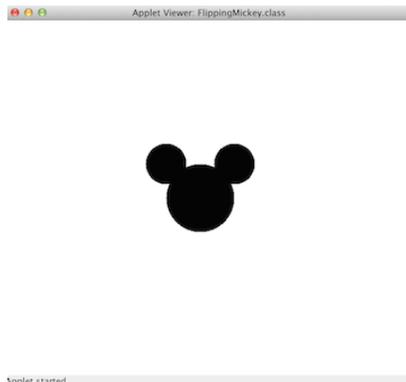
In this program you will be adding a feature that allows you to flip the Mickey left if the mouse is pressed on his left ear, flip right if the mouse is pressed on his right ear, and flip upside-down if the mouse is pressed on his face—all mouse presses must last for longer than 500 milliseconds (1/2 second) during a mouse click. Once the Mickey is flipped upside-down, if the mouse presses the face in the Mickey for longer than 500 milliseconds during a mouse click, the Mickey should flip right-side up again. Even when Mickey is upside-down, it could still be flipped either left or right by clicking on its ears. For flipping the Mickey either left or right, the flip should be centered on the ear that was clicked on. A long mouse press during a mouse click **not** in Mickey will not flip Mickey. Recall: a mouse click event is fired if there is a mouse press event followed by a mouse release event with no mouse movement between the press and release. A mouse press followed by moving the mouse one pixel and then a mouse release does not fire a mouse click event.

For example:

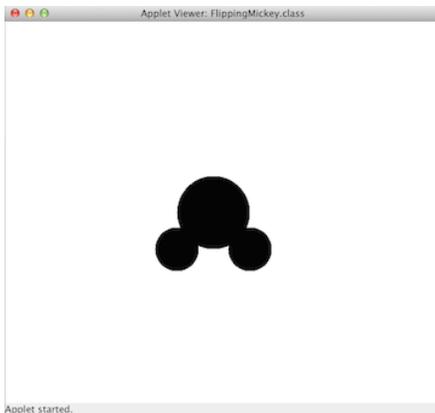
How the Mickey looks when you place it with initial mouse click:

How the Mickey looks after holding down the mouse more than 1/2 second on mouse click while mouse is on the left ear:

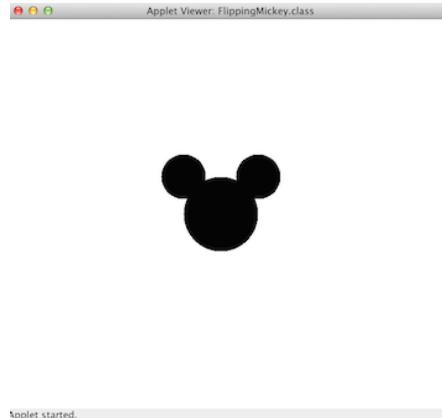
Then click on Mickey’s face holding the mouse button down more than 1/2 second to turn it upside-down:



How the Mickey looks after holding down the mouse button more than 1/2 second on mouse click while mouse is on the right ear:



How the Mickey looks after holding the mouse button down more than 1/2 second on mouse click while mouse is on Mickey's face:



Notes: You should be able to drag mickey around despite however the figure was flipped. Also, even if part of the Mickey is off the canvas, you can still flip Mickey based on its parts that are still left on the canvas.

You will need four files for this program:

Mickey.html	Timer.java
Mickey.java	FlippingMickey.java

Mickey.html - change the applet tag's code attribute to be FlippingMickey.class

FlippingMickey.java - this is the main GUI controller class that handles all mouse events and controls what we see on the canvas. This class needs to extend WindowController. Methods defined in this class include: begin(), onMouseClick(), onMouseDrag(), onMousePress(), onMouseRelease(), onMouseExit(), onMouseEnter(). Where appropriate, code in this class will create an instance of a Mickey and send messages to this Mickey object (tell the Mickey figure to flip, ask the Mickey figure to check if the current mouse Location is contained in the Mickey figure, tell the Mickey figure to move, tell the Mickey figure to remove itself from the canvas).

Mickey.java - this class defines what a Mickey figure is (the 3 filled ovals) and what a Mickey figure can do (what messages it can respond to). This class will define a constructor to initialize a new Mickey object placing it on the canvas as part of the FlippingMickey along with the methods to determine if the mouse pointer is contained in the Mickey figure, move the Mickey figure some delta, remove the Mickey figure from the canvas, and flip the Mickey figure.

Timer.java - this class calculates timing between events. Just copy the code on page 168 of the textbook for the class Timer.

It is important to note the difference between the FlippingMickey class and the Mickey class. The Mickey class should have all the relevant variables and methods specific to each Mickey created on the canvas (in this case we only create one at a time). The GUI controller class, FlippingMickey, handles all the activities that occur on the canvas – this includes the mouse interactions/events with the canvas and Mickey object on the canvas.

Since both classes interact with each other, the FlippingMickey (GUI) needs to have a reference to a Mickey object and the Mickey object needs a reference to the GUI canvas in the FlippingMickey. This is done by passing the canvas as an actual argument to the Mickey constructor when the code in onMouseClick() in FlippingMickey creates a Mickey object. Examples in Chapter 6 of the textbook show how to do this.

Specifications:

The program should still have same functionality as was specified in PA2 (able to create a Mickey on the canvas and drag it around).

A skeleton example of FlippingMickey.java:

```
import objectdraw.*;

public class FlippingMickey extends WindowController
{
    private Text instr1, instr2, instr3;
    private static final int INSTR1_X = 50;
    private static final int INSTR1_Y = 50;
    private static final int INSTR2_X = INSTR1_X;
    private static final int INSTR2_Y = INSTR1_Y + 20;
    private static final int INSTR3_X = INSTR1_X;
    private static final int INSTR3_Y = INSTR2_Y + 20;

    private static final int FLIP_PRESS_THRESHOLD = 500; // Half a second
    private Timer timer;

    ...

    // additional variables you might need and begin() and the event handling methods
}
```

A skeleton example of Mickey.java:

```
import objectdraw.*;

public class Mickey
{
    private FilledOval leftEar, rightEar, face;
    private static final int FACE_RADIUS = 50;
    private static final int EAR_RADIUS = 30;
    private static final int EAR_OFFSET = 50; // Center of each ear is this
                                              // offset up and over (x and y)
                                              // from center of face.

    private DrawingCanvas canvas;

    ...

    // additional variables you might need and ...

    public Mickey( Location point, DrawingCanvas canvas ) // ctor

    public boolean contains( Location point ) {}

    public boolean inFace( Location point ) {}

    public boolean inLeftEar( Location point ) {}

    public boolean inRightEar( Location point ) {}

    public void move( double dx, double dy ) {}

    public void removeFromCanvas() {}

    public void flipLeft() {}

    public void flipRight() {}
}
```

```

    public void flipUpDown() {}
}

```

Use calls to the **Timer** class methods in FlippingMickey to determine how long the mouse button was pressed:

- Create a Timer object in begin()
- When the mouse is **pressed**, the timer should **reset** (start the timer).
- When the mouse is **released**, get the elapsed time in milliseconds and if the mouse was on the Mickey (grabbed) and the elapsed time was longer than ½ second (500 milliseconds) then flag that the Mickey figure should be flipped if this was indeed a mouse click (vs. a drag).
- In mouse click event handler send the flip message to the Mickey object if it should be flipped. Note the mouse click event will be fired after a mouse press event and mouse release event with no mouse motion between the press and release.
- The Mickey should **not flip** if the mouse is **dragged** or if the mouse press/click is **not on the Mickey**.
- When the mouse is **clicked**, if the mouse was pressed for enough time (1/2 second), the Mickey should **flip** upside-down or right-side up. Otherwise the Mickey does not change.
- You should use the following constant to compare with the elapsed time Mickey was pressed:

```

private static final int FLIP_PRESS_THRESHOLD = 500; // Half a second

```

The starting text/instructions of your program should be adjusted to look like the following (and should disappear when a Mickey is placed, just like in your last assignment):

```

Click to display a Mickey silhouette centered at the mouse click.
Mouse press in any part of the image and drag to move image around.
Mouse click in any part of the image with a mouse press for more than 0.5 seconds to flip image.

```

EXTRA CREDIT (5 points)

Create another main GUI controller class that instead of flipping Mickey, **rotates** Mickey and **repositions** Mickey. Before you start the extra credit, make sure that FlippingMickey.java is running perfectly. From there, type the following commands:

- cd ~/pa3
- cp FlippingMickey.java RotatingMickey.java
- cp FlippingMickey.html RotatingMickey.html

By using the previous Unix commands, you are creating a copy of FlippingMickey.java and naming that copy RotatingMickey.java. BOTH FILES ARE NEEDED IF YOU ATTEMPT THE EXTRA CREDIT.

After this, make sure to change the class definition in RotatingMickey.java to the new name. ie:

```

public class RotatingMickey extends WindowController {
    ...
}

```

You must also change FlippingMickey.class to RotatingMickey.class in the html file.

The following features must be implemented:

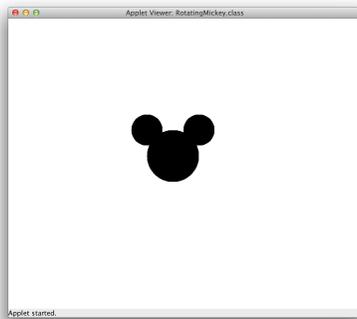
- 1) If the Mickey is pressed on his left ear, Mickey will rotate counter-clockwise by 90 degrees around the center of the left ear.

- 2) If the Mickey is pressed on his right ear, Mickey will rotate clockwise by 90 degrees around the center of the right ear.
- 3) If the Mickey is pressed on his face, Mickey will be repositioned to the middle of the window pane.

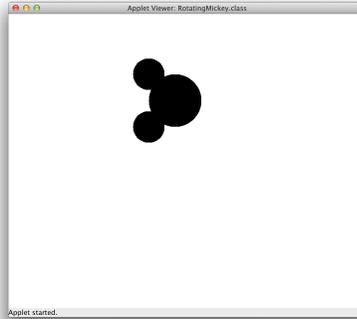
Just like for the FlippingMickey.java implementation, these presses must last longer than than ½ second (500 milliseconds) during the mouse click.

For example:

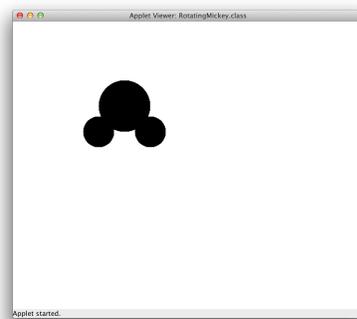
How the Mickey looks
When you place it with
initial mouse click:



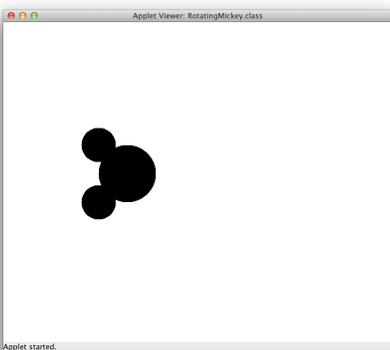
How the Mickey looks after
holding down the mouse more
than ½ second on mouse click
while mouse is on the left ear:



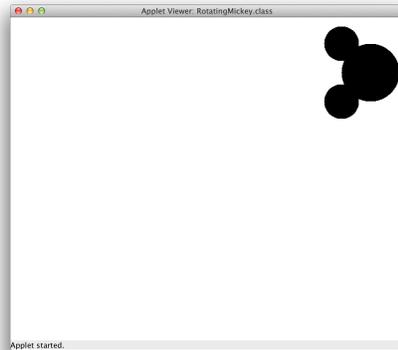
How the Mickey looks after
holding down the mouse more
than ½ second on mouse click
while mouse is on the left ear again:



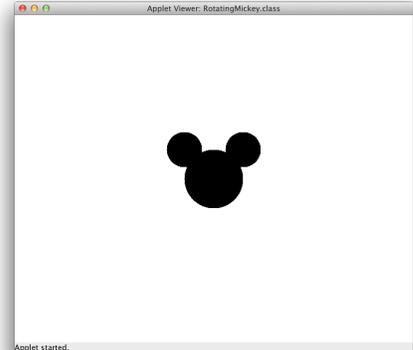
How the Mickey looks after
holding down the mouse button
than ½ second on mouse click
while mouse is on the right ear:



How the Mickey looks after
dragging Mickey over to the side
of the canvas



How the Mickey looks after holding
down the mouse button more than ½
second on mouse click while mouse
is in the face.



Just to be explicit, no matter what orientation the Mickey is currently in, clicking in the face for over a half second will center the figure upright in the center of the canvas.

Note: You will NOT receive extra credit for repositioning Mickey in the middle if you hardcode Mickey to be moved to the center of a window with a specific size; we will be checking this particular feature with arbitrary window sizes.

This time, the starting text/instructions of your program should be adjusted to look like the following (and should disappear when a Mickey is placed):

Click to display a Mickey silhouette centered at the mouse click

Mouse press in any part of the image and drag to move image around

Mouse click in ear parts of the image with a mouse press for more than 0.5 seconds to rotate image

Mouse click in face part of the image with a mouse press for more than 0.5 seconds to center image upright

Turnin

To turnin your code, navigate to your home directory and run the following command:

```
> turnin pa3
```

You may turn in your programming assignment as many times as you like. The last submission you turn in before the deadline is the one that we will collect.

Verify

To verify a previously turned in assignment,

```
> verify pa3
```

If you are unsure your program has been turned in, use the verify command. **We will not take any late files you forgot to turn in.** Verify will help you check which files you have successfully submitted. It is your responsibility to make sure you properly turned in your assignment.

Files to be collected:

- Triangles.java
- Mickey.java
- FlippingMickey.java
- FlippingMickey.html
- Timer.java
- objectdraw.jar
- README

Additional files to be collected for Extra Credit:

- RotatingMickey.java
- RotatingMickey.html

The files that you turn in must be EXACTLY the same name as those above.

NO LATE ASSIGNMENTS ACCEPTED.

DO NOT EMAIL US YOUR ASSIGNMENT!

Start Early and Often!