Programming Assignment 5 (100 Points)

Due: 11:59pm Thursday, November 6th

README (10 points)

You are required to provide a text file named **README,** NOT Readme.txt, README.pdf, or README.doc, with your assignment in your pa5 directory. There should be no file extension after the file name "**README**". Your README should include the following sections:

Program Description (5 points) : Provide a **high level** description of what your program does and how you can interact with it. Make this explanation such that your grandmother or uncle or someone you know who has no programming experience can understand what this program does and how to use it.

Write your READMEs as if it was intended for a 5 year old. **Do not assume your reader is a computer science major.**

Short Response (5 points) : Answer the following questions:

Vim Questions

1. Using vim/gvim, how can you open a file from the Linux command line so that you are taken directly to a specified line in the file?
2. How do you turn off special coloring of keywords and syntax in vim (this is most evident in gvim)? For example, by default your gvim colors comments differently from other parts of code (comments are colored blue by default). If you turn off syntax coloring then all the code in your file will be of same color, black by default. How do you turn the coloring back on?
3. What is the command that you type in to view more information on how to use vim/gvim? The specific command that we're looking for is the one you type in command mode in an open vim/gvim file, not on the terminal. Typing the command will bring up a help manual for vim.

Java Questions

4. What run time structure holds the local variables and formal parameters with each method invocation?
5. What are the only things that can be declared in a Java interface? Be specific with full access modifiers and keywords.

STYLE (20 points)

Please see previous programming assignment write-ups.

You will be specifically graded on commenting, file headers, class and method headers, meaningful variable names, judicious use of blank lines, not having more than 80 characters on a line, perfect indentation, no magic numbers/hard-coded numbers other than zero, using setters and getters, etc.

CORRECTNESS (70 points)

For this assignment, you will extend your PA4 ResizablePacMan applet to include standard Java GUI components (JButtons, JLabel, JScroller, JPanels) and standard Java Event Handling. You will want to reference Ch 11 in the textbook and the notes for Ch 11!

You will first want to make your new pa5 directory and copy the appropriate pa4 files over into this new pa5 directory:

```
> cd
> mkdir pa5
> cp pa4/Resizable*.java pa5
> cp pa4/objectdraw.jar pa5
> cp pa4/*.html pa5
```

If it is not already, you should probably name the html file ResizablePacManController.html

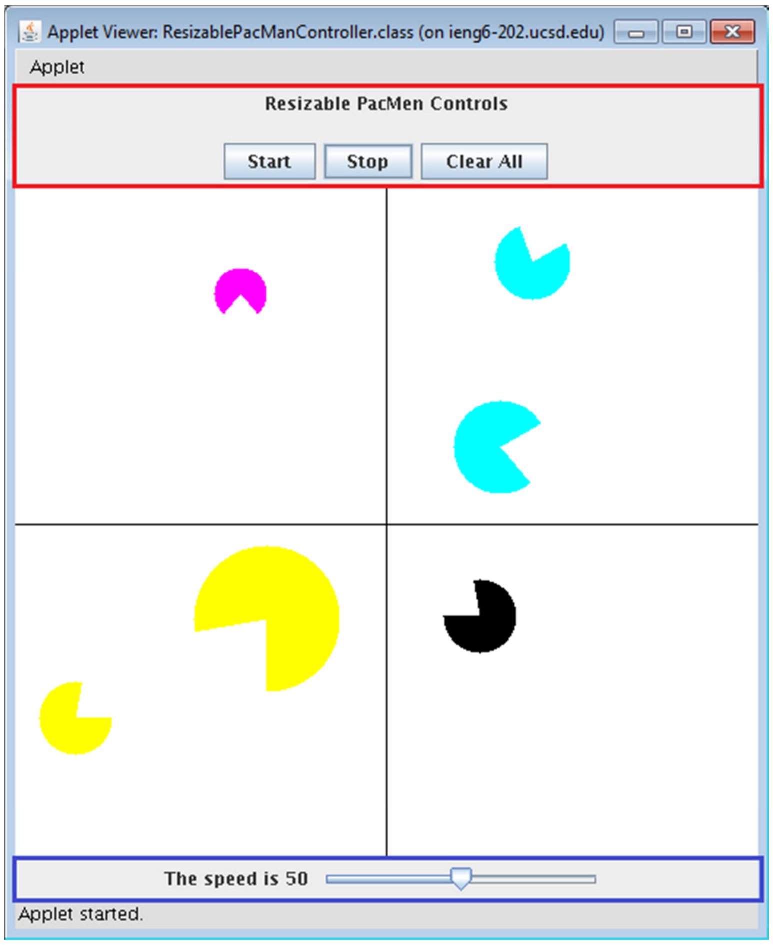
Two important quick changes from PA4:

1. Make sure your PacMen spin **clockwise.**
2. The PacMen should spin in increments of **10** degrees per call to setStartAngle(double).

Regardless of what you did in PA4, you need to make these changes for this assignment!!!

Begin the Coding

Let's take a look at a screen shot of what your new applet should look like:


Applet Viewer: ResizablePacManController.class (on ieng6-202.ucsd.edu)
Applet
Resizable PacMen Controls
Start
Stop
Clear All
The speed is 50
Applet started.

As you can see, the basic layout of the canvas with the horizontal and vertical lines and the resizable PacMen are the same along with their same functionality from PA4. This time, notice the control panel at the top (outlined in red) with a centered label, "Resizable PacMen Controls", and three buttons: "Start", "Stop", and "Clear All". Next, notice the bottom panel (outlined in blue) with a label indicating the current speed of the resizable PacMen, and a slider to control the speed (the smaller the speed value, the slower the PacMen resize -- MIN_SPEED is 1 and MAX_SPEED is 100). When the slider is at 1, the PacMen should spin and resize slowest. Conversely, when the slider is at 100 the PacMen should spin and resize the fastest.

You will have to do some calculation with the speed value extracted from the speed slider in order to have the speed of the PacMen work correctly. The default slider speed needs to be 50 when your program starts. <u>This corresponds to</u>

having a pause time of 50 milliseconds at the end of the forever loop in the run() method.

You should probably get the new layout with the panels, labels, buttons and scroller done first without worrying about the event handling on these GUI components. Ch 11 in the textbook explains what the default layout manager is for a WindowController. We will still be extending WindowController and using the drawing canvas for the PacMen. Resizing the applet should still keep the lines proportional but the PacMen don't move (same functionality as PA4).

The book is your best friend for this assignment!

Once you get the layout done, then add some standard Java event handling. You will need to change from using the objectdraw helper mouse event handlers to using the standard Java mouse event handlers. For example, change from using

```
public void onMouseClick( Location point )
```

to using (and defining)

```
public void mouseClicked( MouseEvent evt )
```

You will need to modify any code that used a Location object to using a MouseEvent object and add the appropriate interfaces. Again **Ch 11 and the notes should be excellent guides**. Remember you will need to provide an implementation for each abstract method declaration in the interfaces you implement. You will need to do this in both your ResizablePacManController.java and ResizablePacMan.java source files. **There should be no onMouse*() methods**. However, we are still using objectdraw library objects like Line and FilledArc and their methods like contains() which takes a Location object. So when necessary, you will need to create Location objects from the MouseEvent's x and y - for example, evt.getX() and evt.getY() - see the Java API docs for MouseEvent and the objectdraw docs for Location.

Each ResizablePacMan object will need to handle the events from each of the buttons and the scroller. The ResizablePacManController will also need to respond to the scroller and buttons in order to:

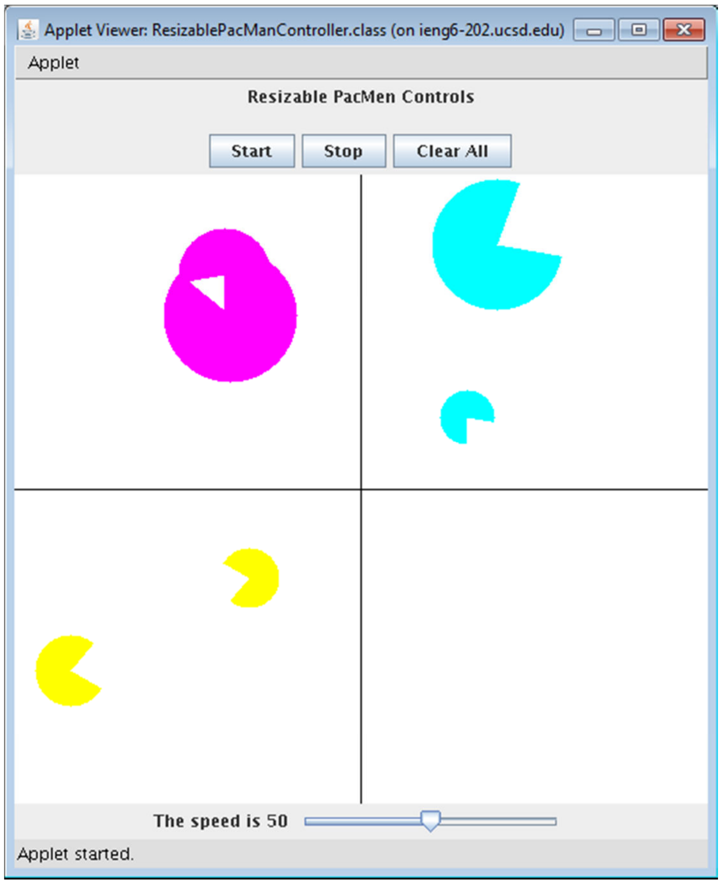
- Change the value in the label indicating the current speed.
- Respond to the start and stop buttons to pass an indication of whether a new ResizablePacMan when created and initialized should start off resizing or not. Clicking the mouse to create a new ResizablePacMan after the stop button has been pressed should create a new PacMan but the PacMan should not be resizing until the start button is pressed.

You will probably need to make heavy use of the on-line docs for the objectdraw library and the standard Java API libraries.

Tip: Bring the book and notes to the lab.
They have pretty much everything you need to implement this assignment.

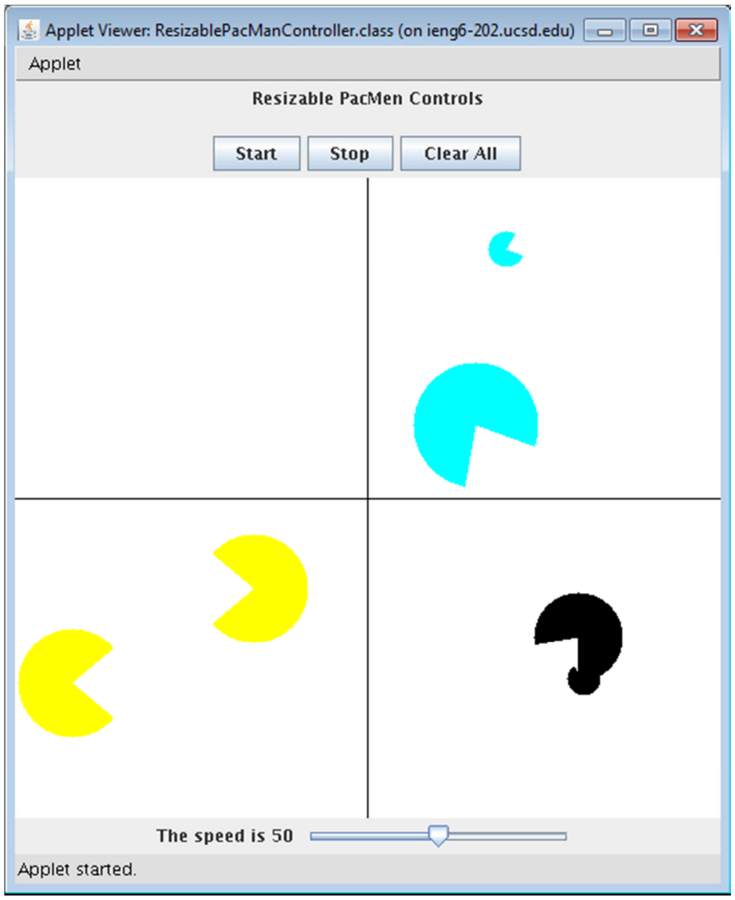
Once you get all that going where you can start and stop the PacMen resizing and add more PacMen and start and clear all and add and stop and clear some more and add ... etc., then you want to add the ability to grab a PacMan and drag/move it around the canvas. The PacMan should change color as it is getting dragged from one quadrant to another (based on the center of the PacMan). This should work whether the PacMan is resizing or not.

The screen shot to the right shows how the PacMan from quadrant IV moved to upper left quadrant while the PacMan are still resizing and spinning →


Applet Viewer: ResizablePacManController.class (on ieng6-202.ucsd.edu)
Applet
Resizable PacMen Controls
Start
Stop
Clear All
The speed is 50
Applet started.

If PacMen overlap each other, you can grab multiple PacMen in the overlapping area and drag them together around the canvas. If a PacMan overlaps one or both lines you can grab the PacMen and the line(s) at the same time and move them all together.

Below is a screen shot where the two PacMen that overlapped each other in upper left quadrant are both simultaneously moved together to lower right quadrant:

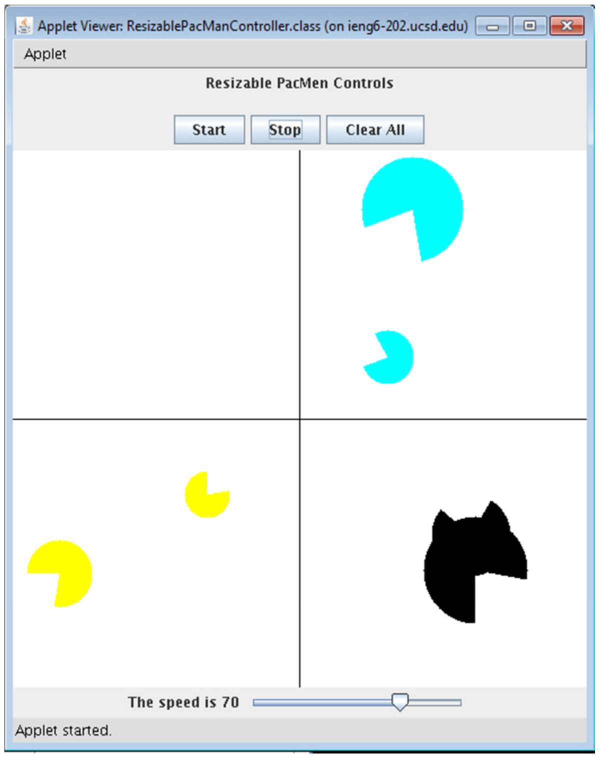

Applet Viewer: ResizablePacManController.class (on ieng6-202.ucsd.edu)
Applet
Resizable PacMen Controls
Start
Stop
Clear All
The speed is 50
Applet started.

Note: Do not allow a PacMan to be dragged off the canvas area. Use the same 5 pixels margins you used in PA4 to prevent either line from being dragged off the canvas and do not allow the center of a PacMan to be dragged past that margin.

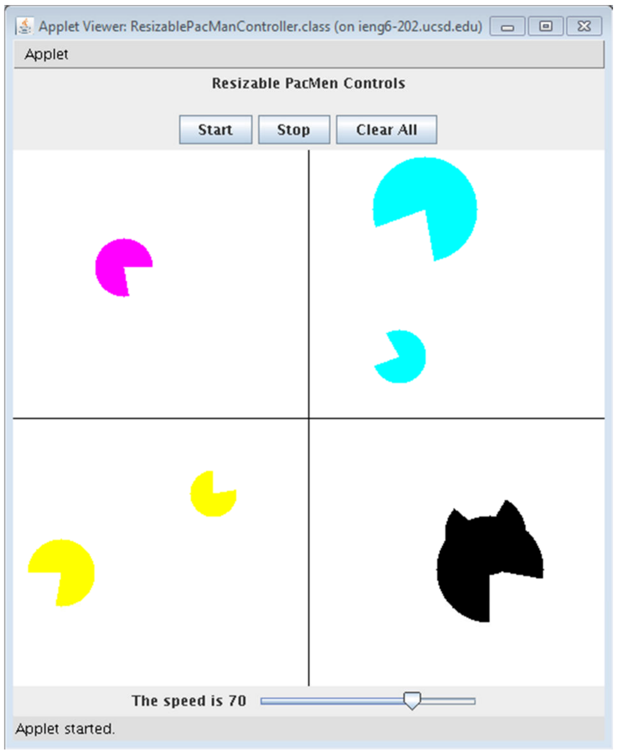
Some of you may want to implement dragging the PacMen around before the buttons and scroller event handling. It is up to you.

You will probably need to modify the ResizablePacMan constructor to deal with new information that needs to be sent to each PacMan so each PacMan can operate independently. There should be no mechanism in the controller object that knows anything about how many PacMen there are or where they are or if they have been cleared from the canvas, etc. Each PacMan handles the various events (mouse, button, scroller) on its own and makes its own decisions based on things like where its center is located, and how big it is in its resizing cycle.

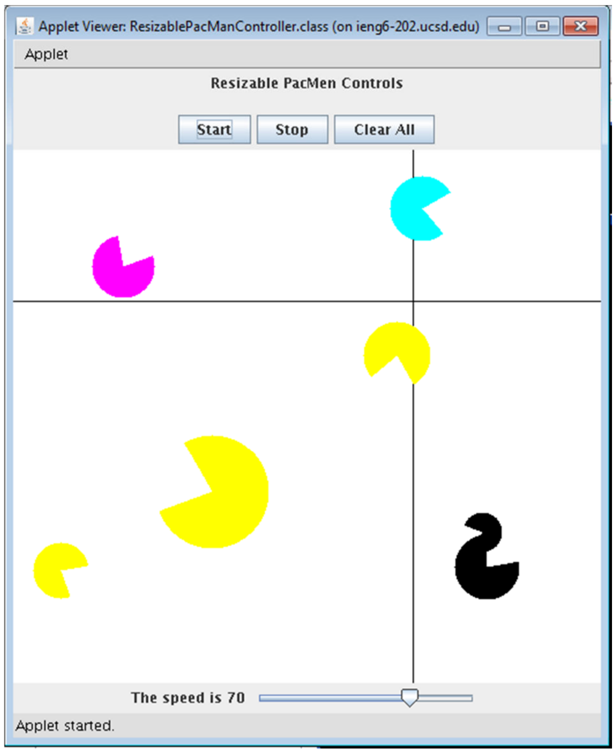
The screen shot below demonstrates changing resizing speed, and hitting the stop button:


Applet Viewer: ResizablePacManController.class (on ieng6-202.ucsd.edu)
Applet
Resizable PacMen Controls
Start
Stop
Clear All
The speed is 70
Applet started.

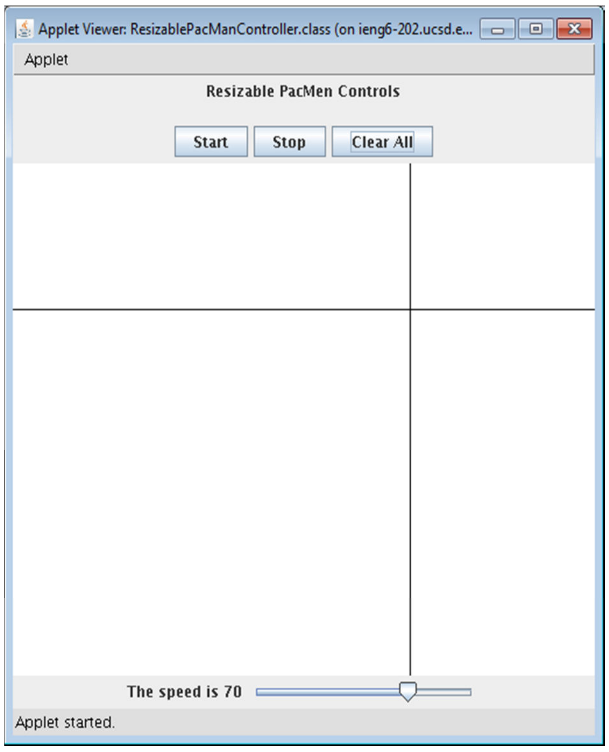
Even though the PacMan has stopped moving, another PacMan can be created on the canvas:


Applet Viewer: ResizablePacManController.class (on ieng6-202.ucsd.edu)
Applet
Resizable PacMen Controls
Start
Stop
Clear All
The speed is 70
Applet started.

Screen shot of after moving the lines around a bit and hitting the start button back again:


Applet Viewer: ResizablePacManController.class (on ieng6-202.ucsd.edu)
Applet
Resizable PacMen Controls
Start
Stop
Clear All
The speed is 70
Applet started.

Finally, this is how the canvas should look after clicking on the “Clear All” button:


Applet Viewer: ResizablePacManController.class (on ieng6-202.ucsd.e...
Applet
Resizable PacMen Controls
Start
Stop
Clear All
The speed is 70
Applet started.

Further note on “Clear All” functionality: When a PacMan is cleared from the canvas, it's run() method running in its own thread should end (change the forever loop in run() to have a condition that will fail when the PacMan is cleared from the canvas). The lines should stay in the same place. **There should be absolutely no exceptions thrown back in the terminal window where you started appetviewer.**

Running

To run (and view) your applet, first create an html file named **ResizablePacManController.html**, which contains the following code:

```
<html>
   <body>
      <applet code="ResizablePacManController.class"
       archive="objectdraw.jar" width="500" height="500">
      </applet>
   </body>
</html>
```

Then use the appletviewer command specifying this html file:

> **appletviewer ResizablePacManController.html**

You must have all the files in the pa5 directory and run appletviewer in the pa5 directory for it to display correctly.

Turnin

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

> **turnin pa5**

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 pa5**

If you are unsure your program has been turned in successfully, 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:

- README
- objectdraw.jar
- ResizablePacMan.java
- ResizablePacManController.html
- ResizablePacManController.java

Extra Credit (5 points)

1. Switch the Spin Direction of the PacMen [2pts]

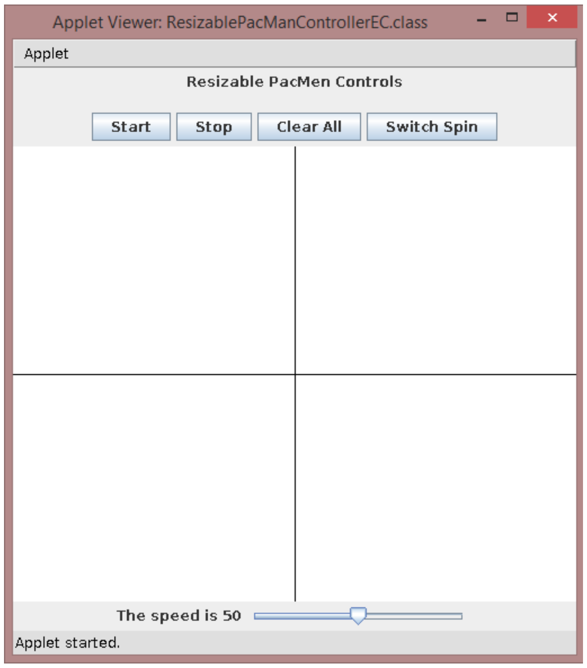
Create a new button on the GUI called “Switch Spin”. Pressing this button should switch the spin direction of the PacMen. That is, if the PacMen are currently spinning clockwise, pressing this button will make them spin counterclockwise. Conversely, pressing this button when the PacMen are spinning counterclockwise will make them spin clockwise.

The two following functionalities must be implemented to receive full credit:

a) If you switch the spin direction of the current PacMen on the canvas.
b) If you switch the spin direction of future PacMen to be placed on the canvas. For example, if the current PacMen are spinning counterclockwise because you clicked "Switch Spin" then new PacMen should start off spinning counterclockwise also.

Note: When you start your program, all the PacMen should be spinning **clockwise**, as mentioned earlier.

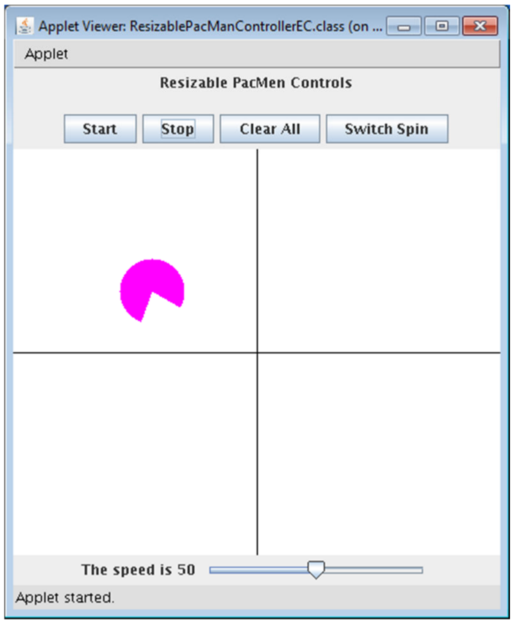
Here is a screenshot of what the GUI of your program should look like with the extra credit →


Applet Viewer: ResizablePacManControllerEC.class
Applet
Resizable PacMen Controls
Start
Stop
Clear All
Switch Spin
The speed is 50
Applet started.

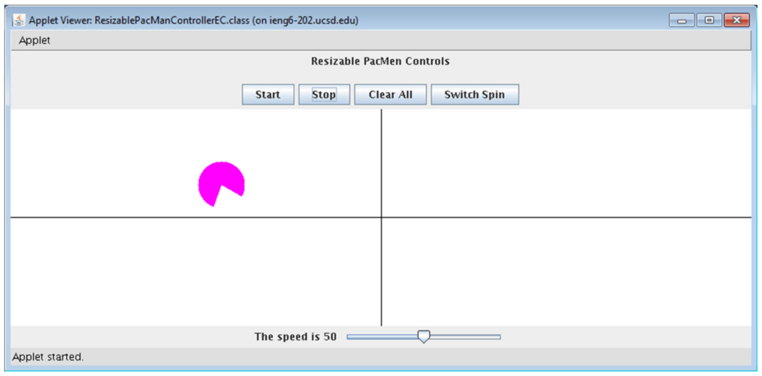
2. Keeping the PacMen’s position proportional to the canvas size [3pts]

Just how the Lines maintain their proportionality when resizing the window, implement the spinning PacMen to preserve their current proportions on the canvas when resizing the window. Use the center of the PacMan to determine its proportional position with respect to the canvas.

Before resizing the canvas:


Applet Viewer: ResizablePacManControllerEC.class (on ...
Applet
Resizable PacMen Controls
Start
Stop
Clear All
Switch Spin
The speed is 50
Applet started.

After resizing the canvas:


Applet Viewer: ResizablePacManControllerEC.class (on ieng6-202.ucsd.edu)
Applet
Resizable PacMen Controls
Start
Stop
Clear All
Switch Spin
The speed is 50
Applet started.

NO LATE ASSIGNMENTS ACCEPTED!

<u>START EARLY!!!</u>

and… HAVE FUN!