CMPU 102 - Fall 2015
Homework 4: Making an Animated Graphics Application

To begin, create an assn4 directory. It is a good idea to save each file you incrementally change and start with a new file for the next task. By doing this, you can always get back to working code if you get into a nest of errors.

Download the starter file PaddleStarter.java

For this lab, you will write a Java animation of a bouncing ball with a moving paddle. You will be instructed to do this in a sequence of steps that leads up to the final product: the ball bouncing off the paddle until the ball drops off the bottom of the scene.

Take a moment to read through and run the starter file. When you run this file, it will show a small yellow circle on a blue background. The ball bounces off all walls, as in the implementation you wrote for lab 6. The file emphasizes the use of constants and you should understand what each of the constants does in the code (HINT: Print out the first page of the program and you will have a list of all constants and instance variables).

The PaddleStarter class extends the JPanel class and implements the ActionListener interface. It uses a Timer object to drive the animation. The only events generated in the starter file are ActionEvents, fired whenever the timer changes value. The actionPerformed method has a single line of code that calls repaint(). The effect is to redraw the scene every time the timer generates an ActionEvent. The paintComponent method is called when the program starts running and on every call to repaint() from the actionPerformed method.

Look through the starter file to find **PART 1**:

1. Save the file as PaddlePart1.

2. Draw a paddle (a rectangle really), at PADDLE_Y pixels from the bottom and the x coordinate of the middle of the scene. Use constants PADDLE_WIDTH and PADDLE_HEIGHT for the dimensions of the rectangle.

3. Use the variable paddleX, defined at the top of this file for the x coordinate of the paddle. The x coordinate of the paddle will change with mouse movement, but its y coordinate should not change.

4. Use the constant PADDLE_Y for the y coordinate of the paddle.

Make sure the file compiles and runs after you are done implementing PART 1. You should now see the ball bouncing and passing behind the paddle. When you modify the code, you should always start with a working file.

Look through the starter file to find **PART 2**:

Change the checkForWallHit method to stop the animation if the ball goes off the bottom of the scene. HINT: Use the variable inPlay which is also used in drawFrame.

Make sure the file compiles and runs after you are done implementing PART 2.

Look through the starter file to find **PART 3** (above PART 2 in code):

Add code to draw a string that says
        "Game over. Close window to quit."
in the middle of the window.

Look through the starter file to find **PART 4**:

The method checkForWallHit method is already implemented in the starter file. For part 4, write method checkForPaddleHit.

1. method checkForPaddleHit

    If ball is touching the top edge of the panel, is between the left and right x coordinates of the paddle, and is moving down, negate the y direction of ball movement. The figure below shows the information available to allow you to check when the ball is touching the paddle top.

2. Uncomment the call to checkForPaddleHit in the drawFrame method after you have written the checkForPaddleHit method.
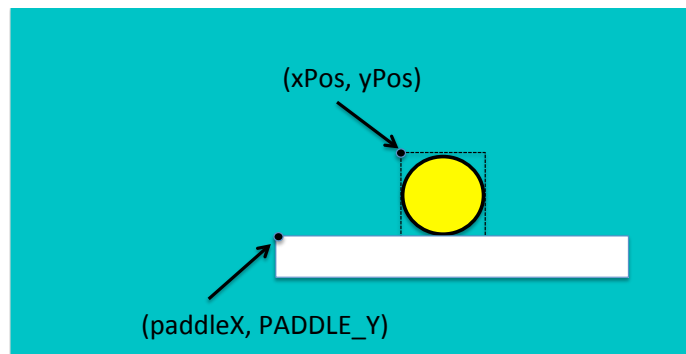
Make sure the file compiles and runs after you are done implementing PART 4. When this file runs, you should see a stationary "paddle" (choose color yourself) near the bottom of the scene. The bouncing ball should bounce off the paddle if it comes in contact with the paddle.

Look through the starter file to find **PART 5**:

In this part, you should add a MouseMotionListener to the setup method and set the drawingArea to listen for mouse moves using the addMouseMotionListener method. You can either write this as an anonymous inner class inside the setup method, or you can have the PaddlePart5 class implement both ActionListener and MouseMotionListener. You should implement a mouseMoved method to change the x-coordinate of the paddle to follow the mouse cursor moves across the scene. Make sure the paddle does not move off the scene. You do not need to implement the mouseDragged method of the MouseMotionListener interface unless you find a need for it.

At the end of working on this assignment, you should have an animation of a bouncing ball that bounces off top, left, right walls, a moving paddle, and a message that appears after the ball eventually drops off the bottom of the scene.

If you want to add some bells and whistles, you can count the number of times the ball hit the paddle and display that information with a small number in one of the frame corners.

Submit your compressed assn4 directory when you are finished and have tested your program.