Aaron Turner

Mimi Opkins

4/30/15

Due: 5/13/15

CECS 277

GUI Project

**MainClass.java**

**import** javax.swing.JFrame;

/\*\*

\* Aaron Turner

\*

\* April 30th, 2015

\*

\* Date Due: May 13th, 2015

\*

\* Purpose: This is the class that creates our frame, and places our Jpanel game

\* within it, which is continuously run

\*

\* Input: N/A

\*

\* Output: Jframe with out game

\*

\*

\*/

**public** **class** MainClass

{

**public** **static** **void** main(String[] args) **throws** InterruptedException

{

//Create our frame

JFrame frame = **new** JFrame("GUI Project");

//Create our game Jpanel

Ball game = **new** Ball(600);

//Add our game to our frame

frame.add(game);

//Set our frame size, and other properties

frame.setSize(game.getFrameSize(), game.getFrameSize());

frame.setResizable(**false**);

frame.setVisible(**true**);

frame.setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);

//Continuously run the game

**while** (**true**)

{

game.moveBall();

game.repaint();

Thread.*sleep*(game.getSpeed());

}

}

}

**Ball.java**

**import** java.awt.Color;

**import** java.awt.Font;

**import** java.awt.Graphics;

**import** java.awt.Graphics2D;

**import** java.awt.Point;

**import** java.awt.RenderingHints;

**import** java.awt.event.MouseEvent;

**import** java.awt.event.MouseListener;

**import** java.util.Random;

**import** javax.swing.JPanel;

/\*\*

\* Aaron Turner

\*

\* April 30th, 2015

\*

\* Date Due: May 13th, 2015

\*

\* Purpose: This is the class that creates our game, it handles all of the animations,

\* calculations, and mouse clicks that the user interacts with

\*

\* Input: N/A

\*

\* Output: Our animations, score, and ball that the game contains(Jpanel)

\*

\*

\*/

**public** **class** Ball **extends** JPanel **implements** MouseListener

{

// Our default serializable ID

**private** **static** **final** **long** *serialVersionUID* = 1L;

// Our coordinates for our ball

**private** **int** x = 0;

**private** **int** y = 0;

// The count for our number of clicks

**private** **int** clicks;

// our game speed

**private** **int** speed;

// our frame size

**private** **int** frameSize;

// Our random

Random rand;

// Counstructor

**public** Ball(**int** frameSize)

{

// inititalize our random

rand = **new** Random();

// Set our speed to 1000

speed = 1000;

//Set our framesize

**this**.frameSize = frameSize;

//Add a mouselistener

addMouseListener(**this**);

}

// Function to change the location of the ball

**public** **void** moveBall()

{

// Give it a coordinate less than the frame size

// Going to use pixel padding

x = rand.nextInt(frameSize - 50) + 1;

y = rand.nextInt(frameSize - 150) + 30;

}

// Function to return our game speed

**public** **int** getSpeed()

{

**return** speed;

}

// Function to return our game speed

**public** **int** getFrameSize()

{

**return** frameSize;

}

//Overriding repaint to draw our game

@Override

**public** **void** paint(Graphics g)

{

**super**.paint(g);

//Get our graphics

Graphics2D ball = (Graphics2D) g;

//Create our ball

ball.setRenderingHint(RenderingHints.*KEY\_ANTIALIASING*, RenderingHints.*VALUE\_ANTIALIAS\_ON*);

ball.setColor(Color.*MAGENTA*);

ball.fillOval(x, y, 60, 60);

// Draw our score

g.setFont(**new** Font("TimesRoman", Font.*PLAIN*, 20));

g.drawString("Score: " + Integer.*toString*(clicks), 245, 25);

}

// Get the mouse events

@Override

**public** **void** mouseClicked(MouseEvent e)

{

}

@Override

**public** **void** mouseEntered(MouseEvent e)

{

}

@Override

**public** **void** mouseExited(MouseEvent e)

{

}

@Override

**public** **void** mousePressed(MouseEvent e)

{

//Get our mouseclick

Point mouseClick = e.getLocationOnScreen();

//get coordinates of our mouse click

**int** mouseX = (**int**) mouseClick.getX();

**int** mouseY = (**int**) mouseClick.getY();

//check if they clicked the ball (-60 since ball size)

//ball is drawn from left to right top to bottom, so location is top center!

**if**(mouseX >= (x - 60) && mouseX <= (x + 60)

&& mouseY >= y && mouseY <= (y + 120))

{

//Increase the score, and if it is divisible by 5, increase the speed

++clicks;

**if**(clicks % 5 == 0 && speed > 150) speed = speed - 150;

//Now repaint the panel

repaint();

}

}

@Override

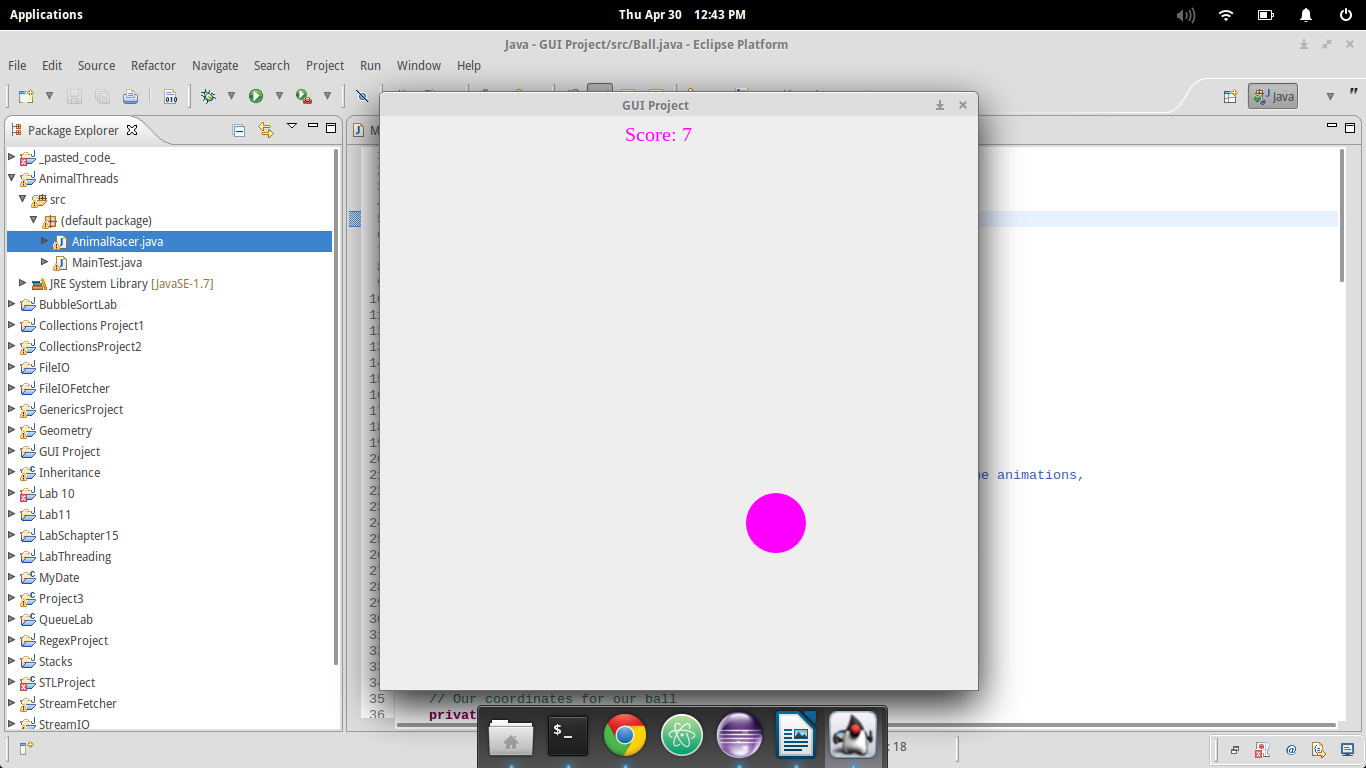
**public** **void** mouseReleased(MouseEvent e)

{

}

}

**Sample Output**

****