**Project 3 – Draw Objects**

**(20 points)**

**Name:** Ozaner Hansha

**Due Date: Tuesday, November 17, 2015**

**Description:**

In this project, you will complete some code that I have started (at the end of the project) to learn more about drawing objects using the ACM Library.

Carry out the following steps:

1. Download the Eclipse project for this project – APCS2015Proj03DrawObjects.
2. Make sure that the ACMLibrary project is on your build path.
3. In the class DrawObjects, carry out the directions in the commented code.
4. Submission:
5. Fill out your name at the top of this page.
6. Copy and paste your Java source code listing to the end of this document. Be careful when copying and pasting to preserve the font information from Eclipse. It makes reading your code much easier.
7. Append a screen shot of your graphics window after your code at the end of this document.
8. Save this document as “Project 3 – Draw Objects (*Lastname Firstname*).doc”  
   Upload your Word document to the Hand-In folder that you created and shared with me. Do not submit a Google document. (You can drag-and-drop the document into the folder from a file browser.)

**Final Code for DrawObjects:**

**package** unit4;

**import** java.awt.Color;

**import** java.awt.Dimension;

**import** java.awt.Font;

**import** java.awt.Point;

**import** java.awt.event.KeyListener;

**import** java.util.Random;

**import** java.util.Timer;

**import** java.util.TimerTask;

**import** acm.graphics.G3DRect;

**import** acm.graphics.GCompound;

**import** acm.graphics.GImage;

**import** acm.graphics.GLabel;

**import** acm.graphics.GLine;

**import** acm.graphics.GOval;

**import** acm.graphics.GPolygon;

**import** acm.graphics.GRect;

**import** acm.graphics.GRoundRect;

**import** acm.graphics.GTurtle;

**import** acm.program.GraphicsProgram;

/\*\*

\* This class displays a number of simple GObjects.<br><br>

\*

\* AP Computer Science<br>

\* Dr. Jones<br>

\* November 13th, 2015<br>

\* **@author** Ozaner Hansha

\*\*/

@SuppressWarnings("serial")

**public** **class** DrawObjects2 **extends** GraphicsProgram **implements** KeyListener

{

/\*\*

\* Displays a number of simple GObjects.

\* **@param** args none expected

\*/

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

{

**new** DrawObjects2().start(args);

}

**public** **static** Dimension *space* = **new** Dimension(0,0);

**public** **static** Dimension addSpace(**int** w, **int** h)

{

*space*.setSize(*space*.width + w, *space*.height + h);

**return** *space*;

}

**public** **static** GPolygon newPolygon(**int** sides, **int** size)

{

GPolygon polygon = **new** GPolygon(0,0);

**for**(**int** x = 0; x <= sides; x++)

{

polygon.addPolarEdge(size,(360/sides)\*x);

}

**return** polygon;

}

/\*\*

\* The init method is a good place to do initializations, to create

\* graphical interfaces, and to set up event handlers.

\*/

@Override

**public** **void** init()

{

setSize(1000, 600); // window width, window height

setBackground(Color.***LIGHT\_GRAY***);

//------------------------------------------------

// GRect, GRoundRect, and G3DRect

//------------------------------------------------

// The following rectangle examples show 3 different placement methods

// for GRect, GRoundRect, and G3DRect objects.

GRect rect = **new** GRect(100, 100, 150, 100); // x, y, wid, len (x, y specified when created)

rect.setColor(Color.***RED***); // the color of the boundary

rect.setFillColor(Color.***CYAN***); // the color of the interior

rect.setFilled(**true**); // whether to fill it

add(rect); // add the object to the canvas

GRoundRect rect2 = **new** GRoundRect(150, 100); // wid, len

rect2.setFilled(**true**);

rect2.setColor(Color.***BLUE***);

add(rect2, 100, 250); // x, y specified when adding to the canvas

G3DRect rect3 = **new** G3DRect(100, 100); // wid, len

rect3.setFilled(**true**);

rect3.setColor(Color.***PINK***);

rect3.setRaised(**true**); // whether to make it look raised (3D)

rect3.setLocation(100, 400); // x, y specified explicitly

add(rect3);

//------------------------------------------------

// GOval

//------------------------------------------------

GOval oval = **new** GOval(300, 100, 150, 100);

oval.setFilled(**true**);

oval.setColor(Color.***GREEN***);

add(oval);

// Make two more ovals below the green one, to the right of the rectangles,

// in two different fill colors, using the other two placement methods.

// Make the boundary color a different shade than the fill color.

// Make one of the ovals a circle.

GOval projOval = **new** GOval(200, 100);

projOval.setFilled(**true**);

projOval.setFillColor(Color.***MAGENTA***);

projOval.setColor(Color.***BLACK***);

add(projOval, 300, 250); //add Parameter Placement Method

//Orange-Outlined, White-Filled, Circle

GOval projOval2 = **new** GOval(120, 120);

projOval2.setFilled(**true**);

projOval2.setFillColor(Color.***WHITE***);

projOval2.setColor(Color.***ORANGE***);

projOval2.setLocation(300, 400); //Explicit Placement Method

add(projOval2);

//------------------------------------------------

// GLine

//------------------------------------------------

// Create two GLine objects and place them as diagonals inside the

// upper-left, cyan rectangle

add(**new** GLine(100, 100, 250, 200));

add(**new** GLine(250, 100, 100, 200));

//------------------------------------------------

// GPolygon

//------------------------------------------------

GPolygon triangle = *newPolygon*(3, 150);

// triangle.addVertex(0, 0); // lower left vertex

// triangle.addEdge(150,0); // edge to lower right vertex

// triangle.addEdge(150\*GMath.cosDegrees(120), -150\*GMath.sinDegrees(120)); // edge to top vertex

triangle.setFilled(**true**);

triangle.setColor(Color.***ORANGE***);

add(triangle, 500, 200);

// Make a trapezoid below the triangle. Use a combination of

// the addVertex and addEdge methods.

GPolygon pinkTrapazoid = **new** GPolygon();

pinkTrapazoid.addVertex(0, 0);

pinkTrapazoid.addEdge(50,-100);

pinkTrapazoid.addEdge(100,0);

pinkTrapazoid.addEdge(50,100);

pinkTrapazoid.setFilled(**true**);

pinkTrapazoid.setColor(Color.***PINK***);

add(pinkTrapazoid, 500, 350);

// Make a stop sign (red octagon with white border) below the trapezoid.

// If you have know about polar coordinate systems, look up and use the

// addPolarEdge method.

GPolygon octagon = *newPolygon*(8,45);

octagon.setFilled(**true**);

octagon.setColor(Color.***WHITE***);

octagon.setFillColor(Color.***RED***);

add(octagon, 540, 500);

//------------------------------------------------

// GLabel

//------------------------------------------------

// Create a GLabel containing the string "STOP" and add it to

// the canvas so that it is centered inside the red hexagon.

GLabel stopString = **new** GLabel("STOP");

stopString.setColor(Color.***WHITE***);

stopString.setFont(**new** Font("Sans Sarif", 35, 25));

add(stopString,530,460);

//------------------------------------------------

// GCompound

//------------------------------------------------

// Create a GCompound containing the stop sign octagon,

// the STOP label, and a sign post. Place it to the right of

// the polygons that you created above.

GPolygon redOctagon = *newPolygon*(8,45);

redOctagon.setFilled(**true**);

redOctagon.setColor(Color.***WHITE***);

redOctagon.setFillColor(Color.***RED***);

add(redOctagon, 552, 523);

GLabel stopLabel = **new** GLabel("STOP");

stopLabel.setColor(Color.***WHITE***);

stopLabel.setFont(**new** Font("Sans Sarif", 35, 25));

add(stopLabel,540,475);

GRect post = **new** GRect(565,522,20,75);

post.setFilled(**true**);

post.setFillColor(Color.***GRAY***);

GCompound stopSign = **new** GCompound();

stopSign.add(redOctagon);

stopSign.add(stopLabel);

stopSign.add(post);

add(stopSign, 200, -350);

//------------------------------------------------

// GImage

//------------------------------------------------

// Find a jpg image of a stop sign. Drag it into the project.

// Create a GImage with it and place it below your GCompound.

GImage stopPic = **new** GImage("elmo.gif");

add(stopPic, 720, 375);

//------------------------------------------------

// GTurtle

//------------------------------------------------

addKeyListeners();

//Instantiate and add GTurtles

GTurtle[] gregArmy = **new** GTurtle[5];

**for**(**int** x = 0; x < gregArmy.length; x++)

{

gregArmy[x] = **new** GTurtle();

add(gregArmy[x], x\*50+200, x\*20+200);

}

Timer timer = **new** Timer();

timer.scheduleAtFixedRate(**new** TimerTask(){

**public** **void** run(){

//"A.I" controlled greg army

**for**(GTurtle g: gregArmy){

**switch**(**new** Random().nextInt(5))

{

**case** 0: g.left(5); **break**;

**case** 1: g.right(12); **break**;

**case** 2: g.forward(2); **break**;

**case** 3: g.setSpeed(1.5); **break**;

**case** 4: g.right(10); **break**;

}}}}, 25, 5);

//Game Logic Loop

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

{

Point p = getMousePosition();

**if**(p != **null**)

{

gregArmy[1].setLocation(p.getX(), p.getY());

System.***out***.println(p);

}

}

}

}

**Screen Shot**

