CECS 174 – Lecture 30 – Applets

Applets – An applet is a java program that can be embedded in an HTML page.

To make an applet you will need to import two packages. The javax.swing package to create the applet, and the java.awt package to draw graphics on your applet.

```
import javax.swing.JApplet;
import java.awt.*;
```

Your class also needs to extend the JApplet class in order to use the applet methods:

```
public class DrawApplet extends JApplet {
```

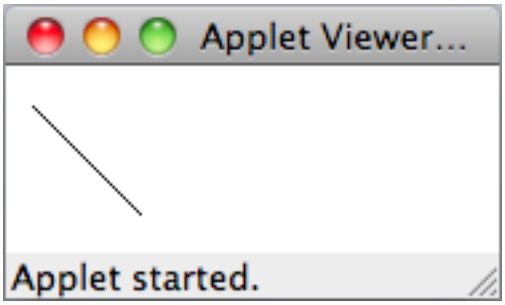
Applets do not use a main() method. Instead we will use the paint method as a place to start. The paint method uses the Graphics class to draw objects.

```
    public void paint(Graphics g) {
```

Drawing Shapes – The Graphics class allows you to draw different shapes and colors on the applet window. The upper left hand corner of the window is the location (0,0). The horizontal axis is the x direction (ranges from 0 to getWidth()) and the vertical axis is the y direction (ranges from 0 to getHeight()).

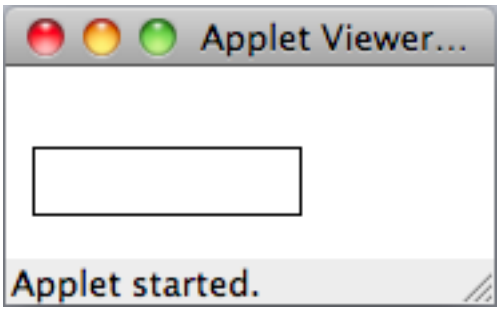
Line – drawLine draws a line from the first position x1,y1 to the second position x2,y2.

```
g.drawLine( 10, 15, 50, 55);
```



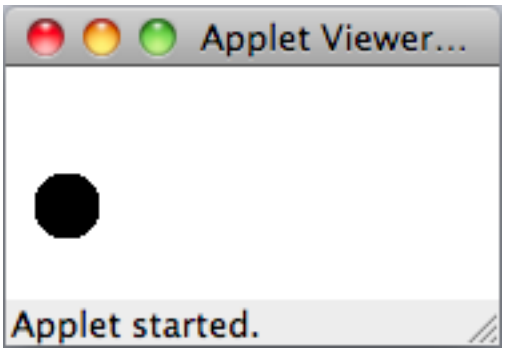
Rectangle – drawRect draws the outline of a rectangle and fillRect draws a filled in rectangle. They both take in parameters of the x and y location of the upper left hand corner of the rectangle, and the width and height of the rectangle. If the width and height are the same, then it will draw a square.

```
g.drawRect( 10, 30, 100, 25);
```



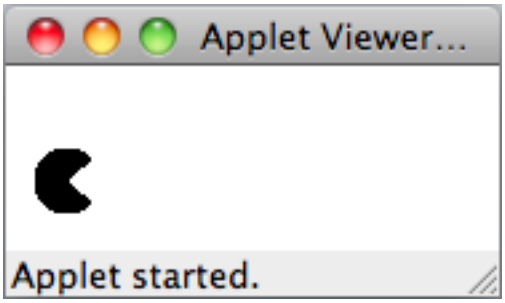
Oval – drawOval draws the outline of an oval and fillOval draws a filled in oval. They both take in parameters of the x and y location of the upper left hand corner of a bounding box containing the oval, and the width and height of the oval. If the width and height are the same, then it will draw a circle.

```
g.fillOval( 10, 30, 25, 25);
```



Arc – drawArc draws the outline of a portion of an oval and fillArc draws a filled in portion of an oval. They both take in several parameters, the x and y location of the upper left hand corner of a bounding box containing the oval, the width and height of the oval, the angle measurement to the beginning of the arc, and the angle of the arc.

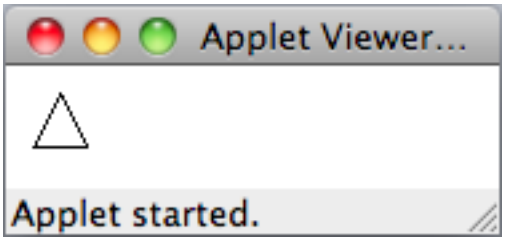
```
g.fillArc( 10, 30, 25, 25, 40, 280);
```



Polygon – drawPolygon draws the outline of a polygon, fillPolygon draws a filled in polygon, and drawPolyline draws a segmented line. They all take in parameters of an array of x locations, an array of y locations and the number of points in the polygon.

```
int [] xPts = {10, 20, 30};
int [] yPts = {25, 15, 25};

g.drawPolygon( xPts, yPts, 3);
```



String – drawString draws a string at the given x and y location starting from the lower left hand corner of the string.

```
g.drawString( "Hello", 15, 25);
```

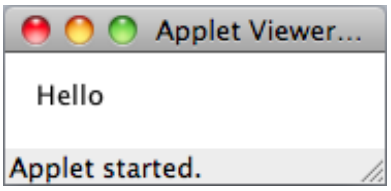


Image – Draws an image to the window. You can only use JPG, GIF, or PNG image files, and they need to be stored in the src folder of your project.

```
Image pic = getImage(getCodeBase(), "img.gif");
g.drawImage(pic, 10, 10, this); //x, y coords
```

Changing Colors – You can change colors of your shapes by either using default colors (Black, Blue, Green, Gray, Yellow, Red, etc) or by creating your own using RGB values (which each range from 0 to 255).

```
g.setColor(Color.GREEN);

Color purple = new Color(125, 35, 200);
g.setColor(purple);
```

Making a Pattern Method – You can create a new shape by drawing a series of objects.

```
                drawFlower(g, 75, 75, 50);
        }
    public void drawFlower(Graphics g,int x,int y,int s){
        g.setColor(Color.PINK);
        g.fillOval(x, y, s, s);
        g.fillOval(x-s, y-s, s, s);
        g.fillOval(x, y-s, s, s);
        g.fillOval(x-s, y, s, s);
        g.setColor(Color.YELLOW);
        g.fillOval(x-s/2, y-s/2, s, s);
    }
}
```