CECS 277 – Lecture 18 – Animation using Threads

Threads can be used to animate drawings on a window. An object that you want to draw will have variables, such as location, size, color, etc. Draw that object to the screen in the paintComponent() method, then update the values of the variables by a small amount and then draw it again. This process of updating and redrawing can be performed many times per second to produce an animation. In the example below, a circle is drawn on an empty field, the location of the circle is modified and then the image is repainted. A thread is used to continuously repeat the process of calling the methods that modify and repaint the circle. In the thread's run method, there is a call to repaint(). This method redraws everything to the screen by calling your paintComponent() method. You should not call paintComponent() yourself, allow repaint() to do it instead. The thread's run method sleeps the application for 16 ms before calling repaint() again, this has our drawing recalculating and redrawing ~60 times a second.

Example: A Bouncing Ball Application. The Ball class extends Rectangle to take advantage of its built in x, y, width, and height, and its many functions.

```
import java.awt.*;
public class Ball extends Rectangle {
   private int dx, dy;
   private Color color;

   public Ball(int x, int y, int radius, Color c, int sp) {
      setBounds( x, y, 2*radius, 2*radius );
      dx = sp;
      dy = -sp;
      color = c;
   }

   public void draw( Graphics g ) {
      g.setColor(color);
      g.fillOval( x, y, width, height);
   }

   public void move( int winWidth, int winHeight ) {
      if( dx > 0 && x > winWidth - width ) {
            dx = -dx;
      } else if( dx < 0 && x < 0 ) {
            dx = -dx;
      }
      if( dy > 0 && y > winHeight - height ) {
            dy = -dy;
      } else if( dy < 0 && y < 0 ) {
            dy = -dy;
      }
      translate( dx, dy );
   }
}
```

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

public class Window extends JFrame {
   public static void main( String [] args ) {
      Window w = new Window();
   }
   public Window(){
      setBounds( 100, 100, 500, 500 );
      setTitle( "Bounce" );
      setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
      Panel p = new Panel();
      setContentPane( p );
      Thread t = new Thread( p );
      t.start();
      setVisible( true );
   }
   public class Panel extends JPanel implements Runnable {
      private Ball b1, b2;
      public Panel() {
         setBackground( Color.BLUE );
         b1 = new Ball( 50, 50, 25, Color.GREEN, 3 );
         b2 = new Ball( 100, 100, 15, Color.RED, 2 );
      }
      public void paintComponent( Graphics g ){
         super.paintComponent( g );
         b1.move( getWidth(), getHeight() );
         b2.move( getWidth(), getHeight() );
         b1.draw( g );
         b2.draw( g );
      }
      public void run() {
         while( true ) {
            repaint();
            try {
               Thread.sleep( 16 ); //~60 fps
            } catch ( InterruptedException e ) {}
         }
      }
   }
```