CS 325 - Class 22

- Today
  - Java's Swing library
  - Animation
- Announcements
  - Project 4 is due tonight by midnight
  - Start on Project 5

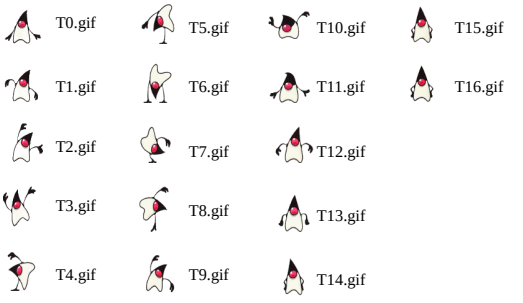
Animation

- Animation requires painting the window repeatedly with slightly changing pictures
  - If the time interval is short enough, it appears as though the picture is in motion
  - Same principle as video
- We will see how to develop animations using
  - Images
  - Graphics

Threads

- In order to draw pictures at specific time intervals, we will use a "thread"
  - A thread is similar to, and simpler than, a process
    - Recall C++ fork( ) function in Unix
- A thread can be told to "sleep" for a certain time interval
  - After it finishes sleeping, it will wake up and continue running

Animation example using images



Tumble example

```
public class Tumble {

   static final int num_images = 17;
   static ImageIcon[ ] icons;

   public static void main(String[ ] args) {
        JFrame frame = new JFrame("Tumble");
        frame.setDefaultCloseOperation
                              (JFrame.EXIT_ON_CLOSE);
        AnimationPanel p = new AnimationPanel( );
                    // class will be defined on upcoming slide
```


Tumble example (cont.)

```
        p.setPreferredSize(new Dimension(150,100));
        p.setBackground(Color.red);
        frame.getContentPane( ).add(p);
        frame.pack( );
        frame.setVisible(true);

        icons = new ImageIcon[num_images];
        for (int i=0; i<num_images; i++)
                icons[i] = new ImageIcon("T" + i + ".gif");

        Thread t = new Thread(p);  // p must be a Runnable object
        t.start( );        // calls p.run( )
   }
```

Tumble example (cont.)

```
static class AnimationPanel extends Panel
                        implements Runnable {
                        // Thread requires a Runnable object
    int counter = -1;

    public void run ( ) {
        while (true) {
            repaint( );
            try { Thread.sleep(200); }   // in milliseconds
            catch (Exception e) { }
        }
    }
```

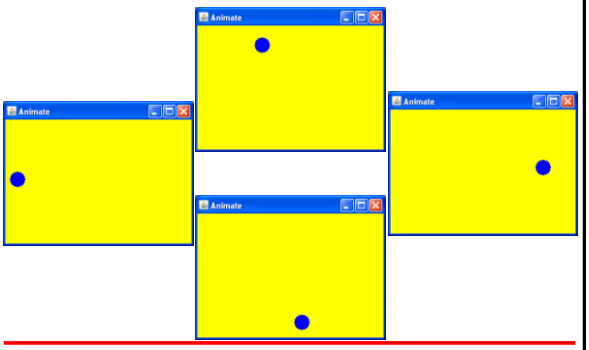

Tumble example (cont.)

```
    public void paint (Graphics g) {
        counter = (counter+1) % num_images;
        if (icons != null && icons[counter] != null)
            icons[counter].paintIcon(this, g, 10, 10);
    }
  }
}
```


Class Exercise

- Download from **http://cs.ua.edu/325/Summer2007/examples/Tumble.zip**
  - Also contains the 17 image files
- Extract, compile and run
- Adjust the sleep time
  - Decrease the sleep time to speed up the animation
  - Increase the sleep time to slow down the animation

Animation using graphics



Bouncing ball example

```
public class Animate {

  public static void main(String[ ] args) {
      JFrame frame = new JFrame("Animate");
      frame.setDefaultCloseOperation
                        (JFrame.EXIT_ON_CLOSE);

      AnimationPanel p = new AnimationPanel( );
      p.setPreferredSize(new Dimension(300,200));
      p.setBackground(Color.yellow);
      p.setForeground(Color.blue);
```


Bouncing ball example (cont.)

```
      frame.getContentPane( ).add(p);
      frame.pack( );
      frame.setVisible(true);

      Thread t = new Thread(p);
      t.start( );
  }
}

class AnimationPanel extends Panel implements Runnable {
      private int  x, y, oldx, oldy,
                   diameter=25, xdelta=4, ydelta=3;
```

Bouncing ball example (cont.)

```
public void paint (Graphics g) {
        g.setColor(getBackground( ));
        g.fillOval(oldx,oldy,diameter,diameter);
        // paints over old ball position with background color

        g.setColor(getForeground( ));
        g.fillOval(x,y,diameter,diameter);
        // paints ball in new position with foreground color
}
```


Bouncing ball example (cont.)

```
public void update (Graphics g) {
        paint(g);
}


// repaint( ) calls update( ), which calls paint( )

// Normally update( ) redraws the entire background
// before it calls paint( )

// But here we don't need the entire background painted
```


Bouncing ball example (cont.)

```
public void run () {
        int w = getSize( ).width;
        int h = getSize( ).height;
        while (true) {
                repaint( );
                try { Thread.sleep(10); }
                catch (Exception e) { }
```


Bouncing ball example (cont.)

```
                oldx = x;
                x += xdelta;
                if (x<0) { x=0; xdelta = -xdelta; }
                if (x>w-diameter) { x=w-diameter; xdelta = -xdelta;}

                oldy = y;
                y += ydelta;
                if (y<0) { y=0; ydelta = -ydelta; }
                if (y>h-diameter) { y=h-diameter; ydelta = -ydelta;}
        }
    }
}
```


Class Exercise

- Download from **http://cs.ua.edu/325/Summer2007/examples/Animate.java**
- Compile and run
- Try modifying the various parameters
  - Modify the sleep time
  - Modify the dimensions of the panel
  - Modify the diameter of the ball
  - Modify xdelta, ydelta
    - These values affect the speed/direction of the ball
- Try removing the first two lines of paint( )
  - What happens? Why?