Week 7 :
a. Animation

Anastasia.Bezerianos@lri.fr

(part of this class is based on previous classes from J.Garcia)

Animation

Used to draw images/objects that vary over time

Use the class Timer from package
javax.swing.Timer
(not to be confused with java.util.Timer)

Although all Timers perform their waiting using a single, shared thread (created by the first Timer object that executes), the action event handlers for Timers execute on another thread ***-- the event-dispatching thread****. This means that the action handlers for Timers can* ***safely perform operations on Swing components****. However, it also means that the handlers* ***must execute quickly*** *to keep the* ***GUI responsive****.*

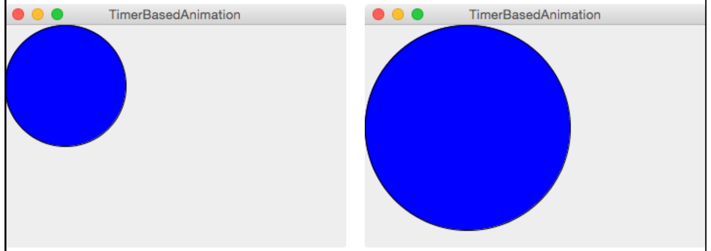
Timer example 1

Timer object, with animation step 100 milliseconds.
Each step creates an ActionListener event (calling MyTimerActionListener)

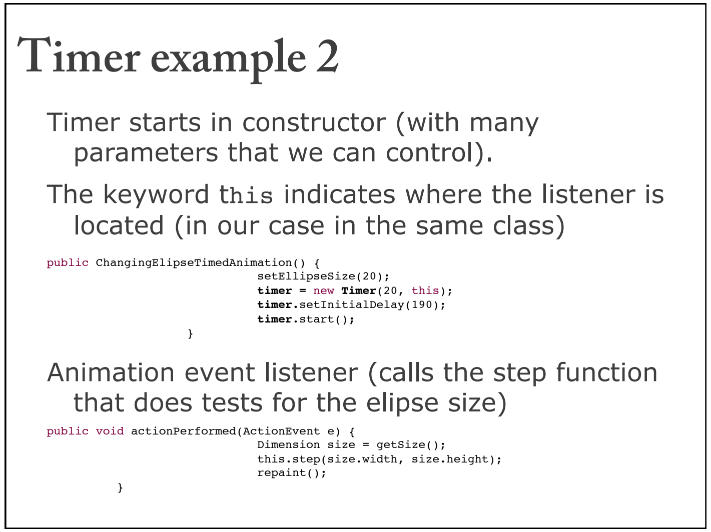
```
public class MyTimerExample extends JFrame {

    Timer mytimer = new Timer (100, new MyTimerActionListener() );

    public void init(){
          JToggleButton b = new JToggleButton ("o");
          b.addActionListener(new ActionListener(){
               @Override
               public void actionPerformed(ActionEvent e) {
                    if ( ((JToggleButton) e.getSource()).isSelected())
                        mytimer.start();
                    else
                        mytimer.stop();
                    }
               });
          getContentPane().add(b);
          }

          public static void main (String args[]){
              JFrame j = new MyTimerExample();
              ((MyTimerExample)j).init();
              j.setSize(50, 50);
              j.setVisible(true);
          }
}
```

```
class MyTimerActionListener implements ActionListener{

    public void actionPerformed(ActionEvent e){
          System.out.print("o");
    }
}
```

Timer example 2



Code available online



Timer example 2

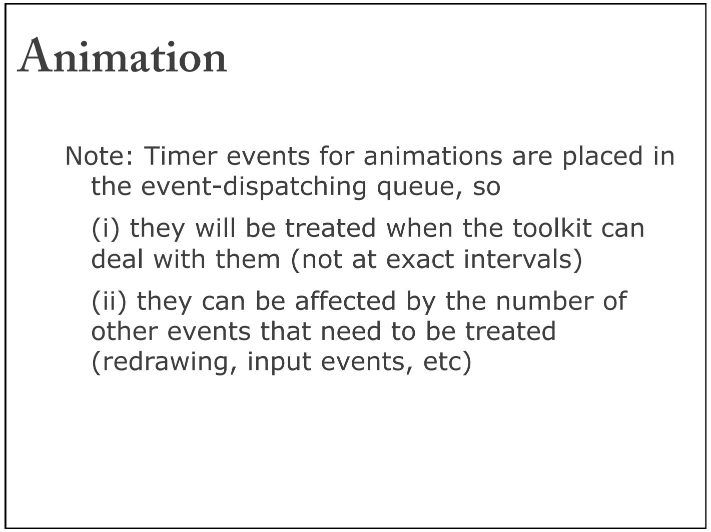
Timer starts in constructor (with many parameters that we can control).

The keyword `this` indicates where the listener is located (in our case in the same class)

```
public ChangingElipseTimedAnimation() {
                              setEllipseSize(20);
                              timer = new Timer(20, this);
                              timer.setInitialDelay(190);
                              timer.start();
                    }
```

Animation event listener (calls the step function that does tests for the elipse size)

```
public void actionPerformed(ActionEvent e) {
                              Dimension size = getSize();
                              this.step(size.width, size.height);
                              repaint();
          }
```



Animation

Note: Timer events for animations are placed in the event-dispatching queue, so

(i) they will be treated when the toolkit can deal with them (not at exact intervals)

(ii) they can be affected by the number of other events that need to be treated (redrawing, input events, etc)