

CS1007 lecture #16 notes

thu 8 nov 2001

<http://www.cs.columbia.edu/~sklar/cs1007>

today:

- news
- more GUI's: menus
- simple animation
- reading: *ch 9*

1

what constitutes help on your homework?

- NOT copying your friend's homework and changing the names
- if I see this again, then you will receive zero for the assignment AND your homework and your friend's will be posted on the class web page, with names, for all to see
- beyond that, there are more serious consequences

3

code from class: ex16.java.

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

public class ex16 extends JApplet {

    public JPanel    panel;
    public JTextField textfield;
    public JMenuBar  menubar;
    public JMenu     shape_menu, animate_menu;
    public JMenuItem string_item, circle_item, rect_item,
                    arc_item, go_item, stop_item;
    public Paper thePaper;

    public void init() {
        panel = new JPanel();
        panel.setLayout( new BorderLayout() );
        Ex16al_class al = new Ex16al_class();
        textfield = new JTextField( "welcome to our toy story emulator" );
        panel.add( textfield, BorderLayout.NORTH );
        thePaper = new Paper();
        panel.add( thePaper, BorderLayout.EAST );
        menubar = new JMenuBar();
        setJMenuBar( menubar );
        shape_menu = new JMenu( "shape" );
        menubar.add( shape_menu );
        string_item = new JMenuItem( "string" );
        shape_menu.add( string_item );
```

5

news.

- midterm #2 will be back next class
- yes, midterm grades will be scaled
- no, the final won't be harder or longer (though you'll have more time to complete it)
- homework #5 hardcopies are due NOW
- what is help (on your homework)?

2

menus.

- Swing components:
 - JMenuBar
 - JMenu
 - JMenuItem
- think of their relationship like a "tree" structure
 - JMenuBar is the root
 - JMenu(s) are children of JMenuBar
 - JMenuItem(s) are children of JMenu(s)
- for example:

```
      menubar
      |
+-----+-----+
|               |
| shape_menu    | animate_menu
|               |
| string_item    | go_item
| circle_item    | stop_item
| rect_item
| arc_item
```

4

```
string_item.addActionListener( al );
circle_item = new JMenuItem( "circle" );
shape_menu.add( circle_item );
circle_item.addActionListener( al );
rect_item = new JMenuItem( "rectangle" );
shape_menu.add( rect_item );
rect_item.addActionListener( al );
arc_item = new JMenuItem( "arc" );
shape_menu.add( arc_item );
arc_item.addActionListener( al );
animate_menu = new JMenu( "animate" );
menubar.add( animate_menu );
go_item = new JMenuItem( "go" );
animate_menu.add( go_item );
go_item.addActionListener( al );
stop_item = new JMenuItem( "stop" );
animate_menu.add( stop_item );
stop_item.addActionListener( al );
setContentPane( panel );
} // end of init()
```

```

private class Ex16a1_class implements ActionListener {

    public void actionPerformed((ActionEvent event) ) {
        Object source = event.getSource();
        if ( source == string_item ) {
            textfield.setText( "drawing string" );
            thePaper.setChoice( 0 );
        }
        else if ( source == circle_item ) {
            textfield.setText( "drawing circle" );
            thePaper.setChoice( 1 );
        }
        else if ( source == rect_item ) {
            textfield.setText( "drawing rectangle" );
            thePaper.setChoice( 2 );
        }
        else if ( source == arc_item ) {
            textfield.setText( "drawing arc" );
            thePaper.setChoice( 3 );
        }
        else if ( source == go_item ) {
            textfield.setText( "starting animation" );
            thePaper.animate();
        }
        else if ( source == stop_item ) {
            textfield.setText( "stopping animation" );
        }
    } // end of actionPerformed() method

} // end of Ex16a1_class

} // end of ex16 class

```

```

        break;
    case 2:
        g.drawRect( x,y,25,50 );
        break;
    case 3:
        g.drawArc( x,y,25,25,180,180 );
        break;
    } // end of switch
} // end of paint() method

public void setChoice( int c ) {
    choice = c;
    update( getGraphics() );
} // end of setChoice() method

public void animate() {
    for ( int i=0; i<10; i++ ) {
        y+=10;
        try {
            Thread.sleep( 1 );
        }
        catch( InterruptedException x ) {
        }
        update( getGraphics() );
    } // end for i
} // end of animate() method

} // end of Paper class

```

code from class: Paper.java.

```

import java.awt.*;
public class Paper extends Canvas {

    private int choice = -1;
    /* string -> 0
       circle -> 1
       rect  -> 2
       arc   -> 3 */

    private int x, y;

    // constructor
    public Paper() {
        setSize( 300,300 );
        x = 0;
        y = 0;
        setBackground( Color.red );
    } // end of Paper() constructor

    public void paint( Graphics g ) {
        g.setColor( Color.red );
        g.fillRect( 0,0,300,300 );
        g.setColor( Color.black );
        switch ( choice ) {
            case 0:
                g.drawString( "to infinity and beyond",x,y );
                break;
            case 1:
                g.drawOval( x,y,25,25 );

```