THE UNIVERSITY OF AUCKLAND

COMPSCI 230

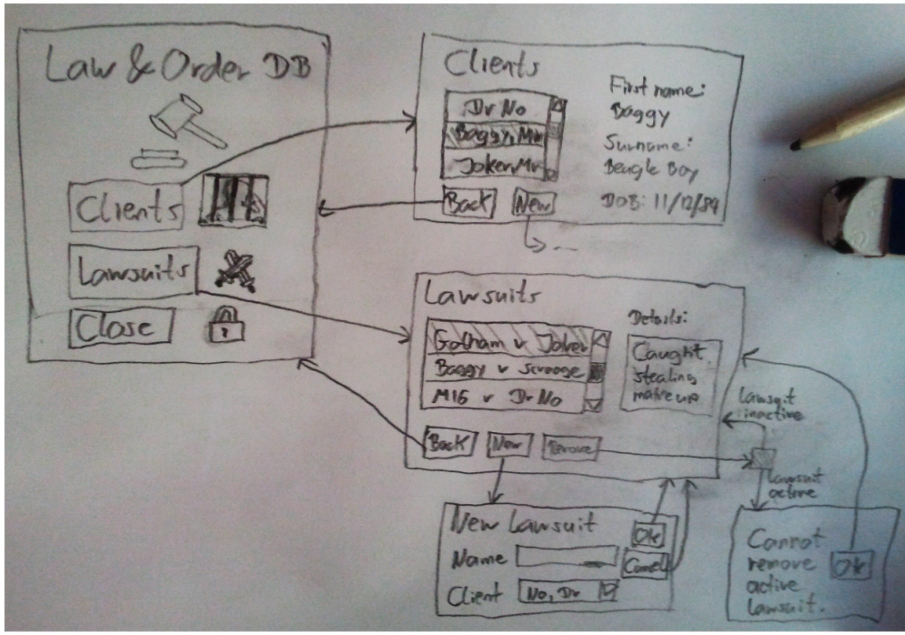
Software Design and Construction

GUI Examples

2013-05-08

How to get an A in the assignment

THE UNIVERSITY OF AUCKLAND

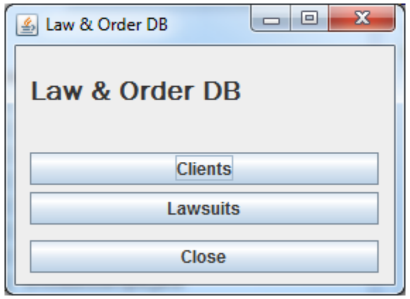


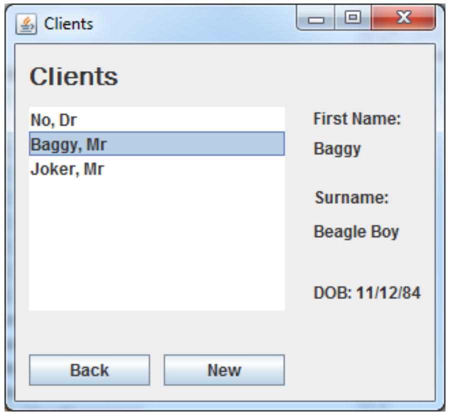
Q1 Screen Diagram

- At least 4 screens
- At least 20 widgets
- "Real" data
- Arrows between screens
- Can use drawing app but must submit JPG or PDF

Q2 Click Dummy

- Implement at least 4 screen as **JFrame**, **JPanel** filling a **JFrame** or **JDialog**
- At least 20 widgets
- Executable **MainFrame**
- Possible to open all screens when running the app
- No functionality required





Clients

Clients

No, Dr
Baggy, Mr
Joker, Mr

First Name:
Baggy

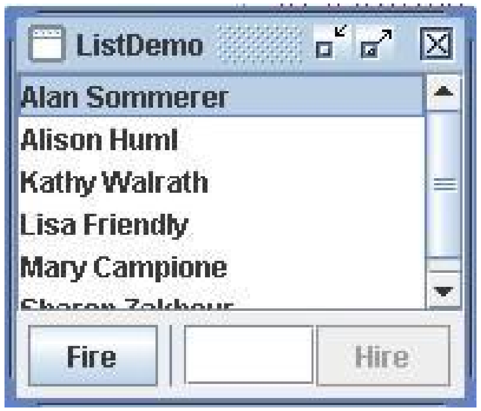
Surname:
Beagle Boy

DOB: 11/12/84

Back New

Q3 Model-View Separation

- All Swing widgets support separate models, e.g.
  - **`DefaultListModel`** for **`JList`**
  - **`DefaultTreeModel`** for **`JTree`**
- For at least one widget, show that you know how to use the separate model object for the data, by manually writing some code that adds the data
- Important to show real-looking data in your click dummy
- State where the separation is done (which file at which lines) in **`readme.txt`**



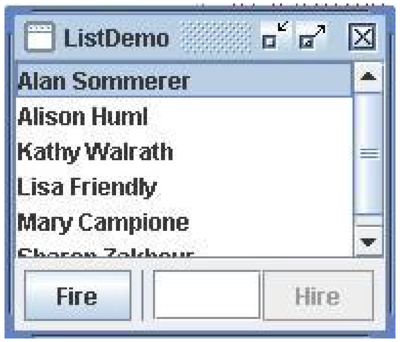
```
listModel = new DefaultListModel();
listModel.addElement("Alan Sommerer");
list = new JList(listModel);
```

Full source code at:
http://docs.oracle.com/javase/tutorial/uiswing/components/list.html

Q4 Event Handler with Simple Functionality

- Write at least one event listener that implements a simple functionality of your app, changing the GUI
- State where the event listener is located (which file at which lines) in **`readme.txt`**

```
…
hireButton.addActionListener(new ActionListener(){
  void actionPerformed(ActionEvent e) {
    listModel.addElement(nameField.getText());
  }});

fireButton.addActionListener(new ActionListener(){
  void actionPerformed(ActionEvent e) {
    int index = list.getSelectedIndex();
    listModel.remove(index);
  }});
…
```



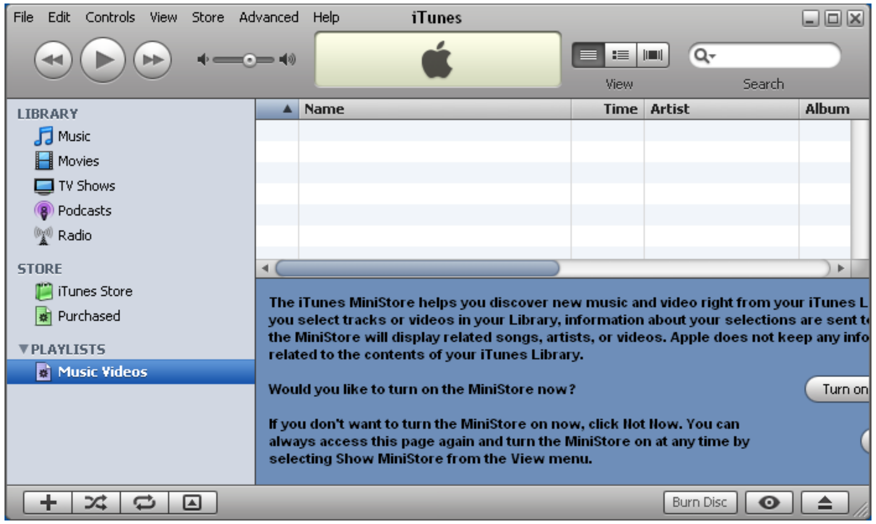
Full source code at:
http://docs.oracle.com/javase/tutorial/uiswing/components/list.html

Important: Don't copy

All assignments are automatically checked for copying.

So please don't copy. Thank you!

Custom Widgets and Drawing



The iTunes MiniStore helps you discover new music and video right from your iTunes L
you select tracks or videos in your Library, information about your selections are sent t
the MiniStore will display related songs, artists, or videos. Apple does not keep any info
related to the contents of your iTunes Library.

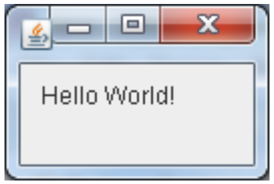
Would you like to turn on the MiniStore now?

If you don't want to turn the MiniStore on now, click Not Now. You can
always access this page again and turn the MiniStore on at any time by
selecting Show MiniStore from the View menu.

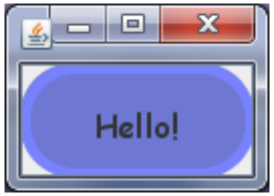
Recap: Creating a Custom Widget

1. Create new class that extends `JPanel`
2. Override **`paintComponent(Graphics g)`** with custom drawing code
   - Make sure to honor the **`width`** and **`height`** of the widget
   - Possibly call **`super.paint(g)`** to draw the superclass widget (e.g. unicolored background)
3. Override size methods such as **`getPreferredSize()`** to return the right sizes for your widget

```
class MyPanel extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawString("Hello World!",10,20);
    }

    public Dimension getPreferredSize() {
        return new Dimension(100,50);
}   }
```

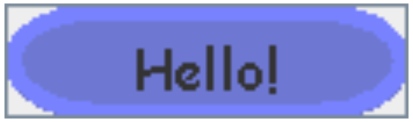


RoundedButton Part 1

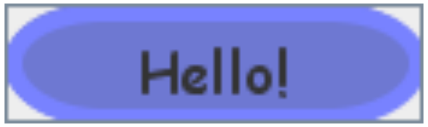


```
class RoundedButton extends JButton {
  public void paintComponent(Graphics g) {
    // Argument of paint() is actually a Graphics2D object,
    // which has more functionality than Graphics
    Graphics2D g2 = (Graphics2D) g;

    // Switch on anti-aliasing, which looks better
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
             RenderingHints.VALUE_ANTIALIAS_ON);
    g2.setRenderingHint(RenderingHints.KEY_TEXT_ANTIALIASING,
             RenderingHints.VALUE_TEXT_ANTIALIAS_ON);
```

Without Anti-Aliasing:

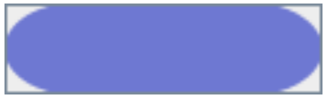


With Anti-Aliasing:

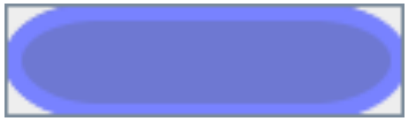


```
    g2.setColor(getBackground());
    g2.fill(new Rectangle2D.Float(
          0, 0, getWidth(), getHeight()));
```

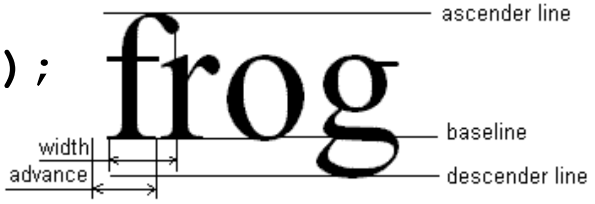
```
    g2.setColor(new Color(110, 120, 210));
    g2.fill(new RoundRectangle2D.Float(
          0, 0, getWidth(), getHeight(), 50, 50));

    ...
```

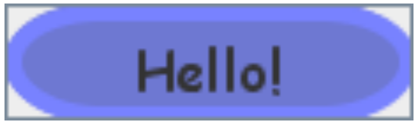


RoundedButton Part 2

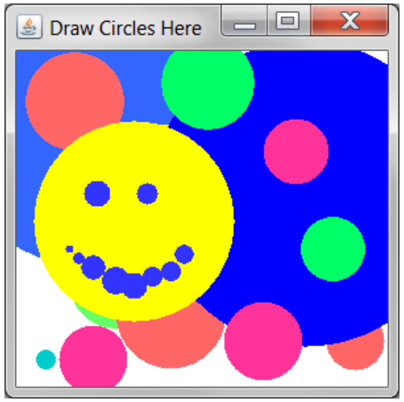


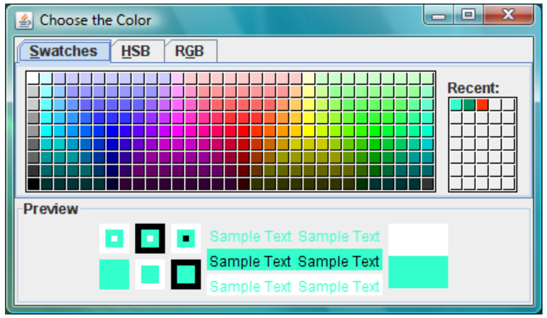
```
    g2.setColor(new Color(120, 130, 255));
    g2.setStroke(new BasicStroke(5));
    g2.draw(new RoundRectangle2D.Float(
        2, 2, getWidth() - 4, getHeight() - 4, 50, 50));
    g2.setStroke(new BasicStroke(1));

    FontMetrics metrics = g2.getFontMetrics(getFont());
    int h = metrics.getAscent();
    int w = metrics.stringWidth(getText());

    g2.setColor(getForeground());
    g2.drawString(getText(),
        (getWidth() - w) / 2, (getHeight() + h) / 2);
  }

  public static void main(String[] args) {
    JFrame frame = new JFrame();
    RoundedButton r = new RoundedButton();
    r.setText("Hello!");
    r.setFont(new Font("Comic Sans MS", Font.PLAIN, 16));
    frame.getContentPane().add(r);
    frame.pack(); frame.setVisible(true);
}  }
```





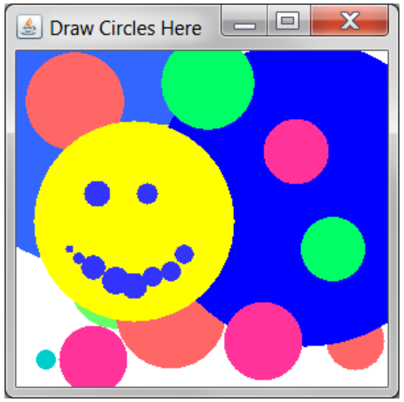
Application Example
"`CirclePaint`"



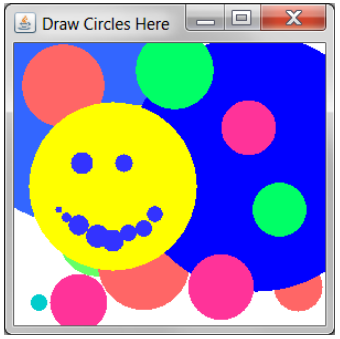


CirclePaint Part 1 Custom Component

```
public class Canvas extends JComponent {
   // our model, i.e. the data that is represented
   class Circle { float x, y, r; Color col; }
   Vector<Circle> circles
          = new Vector<Circle>();
   Circle current;

   public Canvas() {
      setOpaque(true);
      setBackground(Color.white);
      // add mouse listeners, see next slides…
    }


    public void paintComponent(Graphics g)  {
       // paint the background & circles, see next slides…
   }
}
```



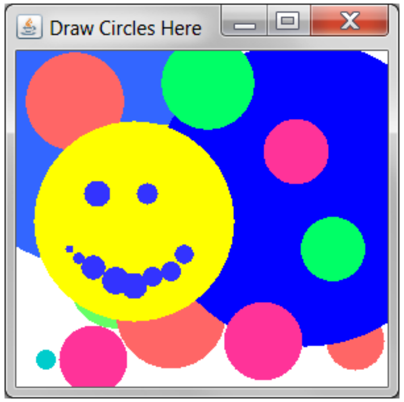
CirclePaint Part 2 Event Listeners



```
addMouseListener(new MouseAdapter() {
   public void mousePressed(MouseEvent e) {
      current = new Circle();
      current.x = e.getX(); current.y = e.getY();
      current.col = CirclePaint.colorChooser.getColor();
    }
   public void mouseReleased(MouseEvent e) {
      if(current!=null) circles.add(current); }
   public void mouseExited(MouseEvent e) {
      current = null; repaint(); }
});
addMouseMotionListener(new MouseMotionAdapter() {
   public void mouseDragged(MouseEvent e) {
      if(current==null) return;
      current.r =  (float)Math.sqrt(
         (e.getX() - current.x) * (e.getX() - current.x)
       + (e.getY() - current.y) * (e.getY() - current.y));
      repaint();
} });
```

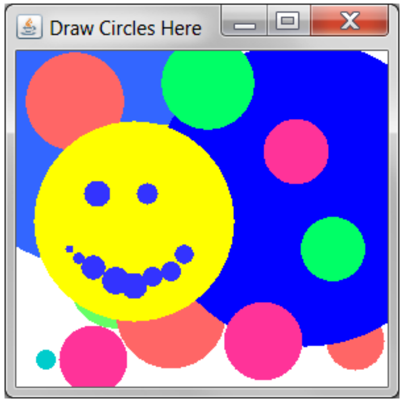
CirclePaint Part 3
Paint Method

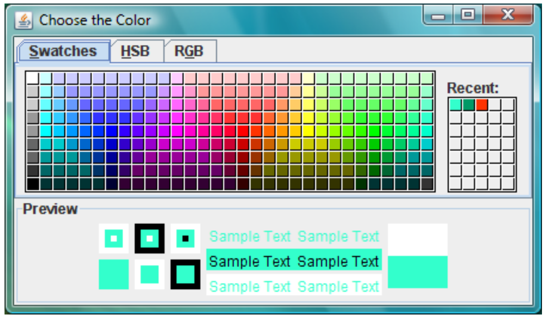
```
public void paintComponent(Graphics g) {
  Graphics2D g2 = (Graphics2D) g;
  g2.clearRect(0, 0, this.getWidth(), this.getHeight());
  for(Circle c : circles) {
    g2.setColor(c.col);
    g2.fill(new Ellipse2D.Float(
              c.x-c.r, c.y-c.r, 2*c.r, 2*c.r));
  }
  if(current!=null) {
    g2.setColor(current.col);
    g2.fill(new Ellipse2D.Float(
      current.x-current.r, current.y-current.r,
      2*current.r, 2*current.r));
  }
}
```



CirclePaint Part 4 Main Class

```
public class CirclePaint extends JFrame {
   public static JColorChooser colorChooser
         = new JColorChooser();
   public CirclePaint() {
      setTitle("Draw Circles Here");
      setDefaultCloseOperation(
         JFrame.EXIT_ON_CLOSE);
      setSize(300, 300);
      getContentPane().add(new Canvas());
   }

   public static void main(String[] args) {
      new CirclePaint().setVisible(true);
      JFrame paletteFrame= new JFrame("Choose the Color");
      paletteFrame.setSize(450, 260);
      paletteFrame.setDefaultCloseOperation(
         JFrame.EXIT_ON_CLOSE);
      paletteFrame.getContentPane().add(
         colorChooser);
      paletteFrame.setVisible(true);
   }
}
```





Summary

- **Custom components** can be created by overriding the method paintComponent(Graphics g) of a widget
- **Interactive graphics** apps can be created by using custom painting and event handlers

References:

- The Java Tutorials: 2D Graphics. http://docs.oracle.com/javase/tutorial/2d/
- The Java Tutorials: Performing custom painting. http://docs.oracle.com/javase/tutorial/uiswing/painting/

Quiz

1. Change the **`CirclePaint`** app so that it draws rectangles instead of circles.
2. What does the **`repaint()`** method of a widget do?