CS 683 Emerging Technologies
Fall Semester, 2004
Doc 21 MIDlet UI pt 3
Contents

References

J2ME in a Nutshell, Kim Topley, O'Reilly, 2002, Chapter 4

Examples in this lecture are based on examples in the above reference

Graphics

Canvas

http://www.eli.sdsu.edu/courses/fall04/cs683/j2me/docs/api/midp/javax/microedition/lcdui/Canvas.html

Graphics

http://www.eli.sdsu.edu/courses/fall04/cs683/j2me/docs/api/midp/javax/microedition/lcdui/Graphics.html

Example with Attributes

Based on an example from J2ME in a Nutshell

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.MIDlet;

class SampleCanvas extends Canvas
   {
   protected void paint(Graphics g)
      {
      }
   }

public class Attributes extends MIDlet
   {

   public void startApp() { }
   public void pauseApp() { }
   public void destroyApp( boolean unconditional ) { }
```

```
public Attributes()
  {
  Display display = Display.getDisplay(this);
  Canvas canvas = new SampleCanvas();

  Form form = new Form("Attributes");
  boolean isColor = display.isColor();
  String colorOrGray = isColor ? "Colors: " : "Grays: ";
  form.append(new StringItem(colorOrGray,
              String.valueOf(display.numColors())));

  form.append(new StringItem("Width: ",
              String.valueOf(canvas.getWidth())));

  form.append(new StringItem("Height: ",
              String.valueOf(canvas.getHeight())));

  form.append(new StringItem("Has Pointer? ",
              String.valueOf(canvas.hasPointerEvents())));

  form.append(new StringItem("MotionEvents? ",
              String.valueOf(canvas.hasPointerMotionEvents())));

  form.append(new StringItem("RepeatEvents? ",
              String.valueOf(canvas.hasRepeatEvents())));

  form.append(new StringItem("Buffered? ",
              String.valueOf(canvas.isDoubleBuffered())));
  display.setCurrent(form);
  }
}
```

Lines

Based on an example from J2ME in a Nutshell

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.MIDlet;

class LineCanvas extends Canvas
  {
  public void paint(Graphics g)
    {
    int width = getWidth();
    int height = getHeight();

    int black = 0;
    g.setColor(black);
    g.fillRect(0, 0, width, height);

    int white = 0xFFFFFF;
    g.setColor(white);
    g.drawLine(0, height/2, width - 1, height/2);

    int yellow = 0xFFFF00;
    g.setStrokeStyle(Graphics.DOTTED);
    g.setColor(yellow);
    g.drawLine(0, height/4, width - 1, height/4);

    g.setGrayScale(255);
    g.setStrokeStyle(Graphics.SOLID);
    g.drawLine(0, 0, width - 1, height - 1);
    }
  }
```

```
public class Lines extends MIDlet
   {
   public Lines()
      {
      Display display = Display.getDisplay(this);
      Canvas canvas = new LineCanvas();
      display.setCurrent(canvas);
      }

   public void startApp() { }
   public void pauseApp() { }
   public void destroyApp( boolean unconditional ) { }
   }
```

Rectangle

Based on an example from J2ME in a Nutshell

```
class RectanglesCanvas extends Canvas
  {
  final static int BLACK = 0;
  final static int WHITE = 0xFFFFFF;

  public void paint(Graphics g)
    {
    int width = getWidth();
    int height = getHeight();

    g.setColor(WHITE);
    g.fillRect(0, 0, width, height);

    g.setColor(BLACK);
    g.drawRect(width/4, 0, width/2, height/4);

    g.setStrokeStyle(Graphics.DOTTED);
    g.drawRect(width/4 + 4, 4, width/2 - 8, height/4 - 8);

    g.setStrokeStyle(Graphics.SOLID);
    g.drawRoundRect(width/4, height/2, width/2, height/4, 16, 8);
    }
  }
```

Translating Origin

```
class TranslateCanvas extends Canvas
   {
   final static int BLACK = 0;
   final static int WHITE = 0xFFFFFF;

   public void paint(Graphics g)
      {
      int width = getWidth();
      int height = getHeight();

      g.setColor(WHITE);
      g.fillRect(0, 0, width, height);

      g.setColor(BLACK);
      g.drawRect(0, 0, width/4, height/4);
      g.translate( 15, 15);
      g.drawRect(0, 0, width/4, height/4);
      g.translate( 20, 0);
      g.drawRect(0, 0, width/4, height/4);

      //Go back to 0,0 Not needed just to show how to do it
      g.translate(-g.getTranslateX(), -g.getTranslateY());
      }
   }
```

What Happens if you Don't Clear the Screen?

```
class NoClearScreenCanvas extends Canvas
   {
   final static int BLACK = 0;
   final static int WHITE = 0xFFFFFF;

   public void paint(Graphics g)
      {
      int width = getWidth();
      int height = getHeight();

      //Note no code to clear screen

      g.setColor(BLACK);
      g.drawRect(0, 0, width/4, height/4);
      g.translate( 15, 15);
      g.drawRect(0, 0, width/4, height/4);
      }
   }
```

Filling Shapes

Based on an example from J2ME in a Nutshell

```
class RectangleFillsCanvas extends Canvas
  {
  final static int BLACK = 0;
  final static int GREEN = 0x00FF00;
  final static int YELLOW = 0xFFFF00;

  public void paint(Graphics g)
    {
    int width = getWidth();
    int height = getHeight();

    // Create a black background
    g.setColor(BLACK);
    g.fillRect(0, 0, width, height);

    g.setStrokeStyle(Graphics.DOTTED);
    g.setColor(GREEN);
    g.fillRect(width/4, height/4, width/2, height/2);
    g.setColor(YELLOW);
    g.drawRect(width/8, height/8, width/2, height/2);
    }
  }
```

Arcs

Based on an example from J2ME in a Nutshell

```
class ArcsCanvas extends Canvas
  {
  final static int BLACK = 0;
  final static int WHITE = 0xFFFFFF;

  public void paint(Graphics g)
    {
    int width = getWidth();
    int height = getHeight();

    g.setColor(BLACK);
    g.fillRect(0, 0, width, height);

    // A quarter circle, clockwise 90 degrees
    // from the 3 o'clock position. Show the
    // bounding rectangle as well.
    g.setColor(WHITE);
    g.drawArc(0, 0, width/2, height/2, 0, 90);

    // A quarter circle, anticlockwise 90 degrees
    // from the 3 o'clock position.
    g.setStrokeStyle(Graphics.SOLID);
    g.setColor(WHITE);
    g.drawArc(width/2, 0, width/2, height/2, 0, -90);
    }
  }
```

Text

Based on an example from J2ME in a Nutshell

```
class TextCanvas extends Canvas
  {
  final static int BLACK = 0;

  public void paint(Graphics g)
    {
    int width = getWidth();
    int height = getHeight();

    g.setColor(BLACK);
    g.drawString("Top left", 0, 0, Graphics.TOP | Graphics.LEFT);

    Font font = g.getFont();
    g.drawString("Below top left", 0, font.getHeight(),
        Graphics.TOP | Graphics.LEFT);

    g.drawString("Bottom right", width, height,
      Graphics.BOTTOM | Graphics.RIGHT);

    String str = "Multi-font ";
    font = Font.getFont(Font.FACE_PROPORTIONAL,
        Font.STYLE_UNDERLINED, Font.SIZE_LARGE);
    g.setFont(font);
    g.drawString(str, 0, height/2,
      Graphics.LEFT | Graphics.BASELINE);
    }
  }
```

Some Motion

```
class SimpleMotionCanvas extends Canvas {
   final static int BLACK = 0;
   final static int WHITE = 0xFFFFFF;

   int boxX = 0;
   int boxY = 0;
   int boxLength = 25;
   int width = getWidth();
   int height = getHeight();
   Timer timer;

   public void paint(Graphics g)  {
      g.setColor(WHITE);
      g.fillRect(0, 0, width, height);

      g.setColor(BLACK);
      g.fillRect(boxX, boxY, boxLength, boxLength);
   }

   protected void showNotify() {
      startTimer();
   }

   protected void hideNotify() {
      stopTimer();
   }
```

Some Motion Continued

```
    protected void moveBox() {
      boxX += 5;
      if (boxX > height)
        boxX = 0;
      boxY += 2;
      if (boxY > width)
        boxY = 0;
      repaint();
    }


    protected void startTimer() {
      timer = new Timer();

      TimerTask updateScreen = new TimerTask() {
            public void run() {
              moveBox();
            }
          };
      int frameRate = 10;
      int interval = 1000/frameRate;
      timer.schedule(updateScreen, interval, interval);
    }

    protected void stopTimer() {
      timer.cancel();
    }
}
```

Clip Regions

```
protected void moveBox()
   {
   repaint(boxX, boxY, boxLength, boxLength);
   boxX += 5;
   if (boxX > height)
      boxX = 0;
   boxY += 2;
   if (boxY > width)
      boxY = 0;
   repaint(boxX, boxY, boxLength, boxLength);
   }
```

Input Events

```
class EventsCanvas extends Canvas
  {

  protected void keyPressed(int keyCode) {
  }

  protected void keyRepeated(int keyCode) {
  }

  protected void keyReleased(int keyCode) {
  }

  protected void pointerPressed(int x, int y) {
  }

  protected void pointerDragged(int x, int y) {
  }

  protected void pointerReleased(int x, int y) {
  }
```