Java on Mobile Devices Feb 19

16:00 – 16:50: J2ME Low Level User Interface

17:00 – 17:50: Garbage Collection (Rasmus)

18:00 – 18:30: Introduction to Mini Project

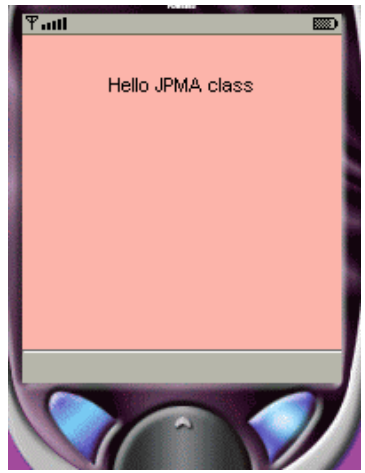
Canvas and Graphics Class

```
class MyCanvas extends Canvas {
        ...

    void paint (Graphics g) { ...

        g.setColor(255, 175, 150);
        g.fillRect(0, 0, getWidth(), getHeight());
        g.setColor(0, 0, 0);
        g.drawString("Hello JPMA class", getWidth()/2,
                     0, g.TOP | g.HCENTER);

    }
}
```

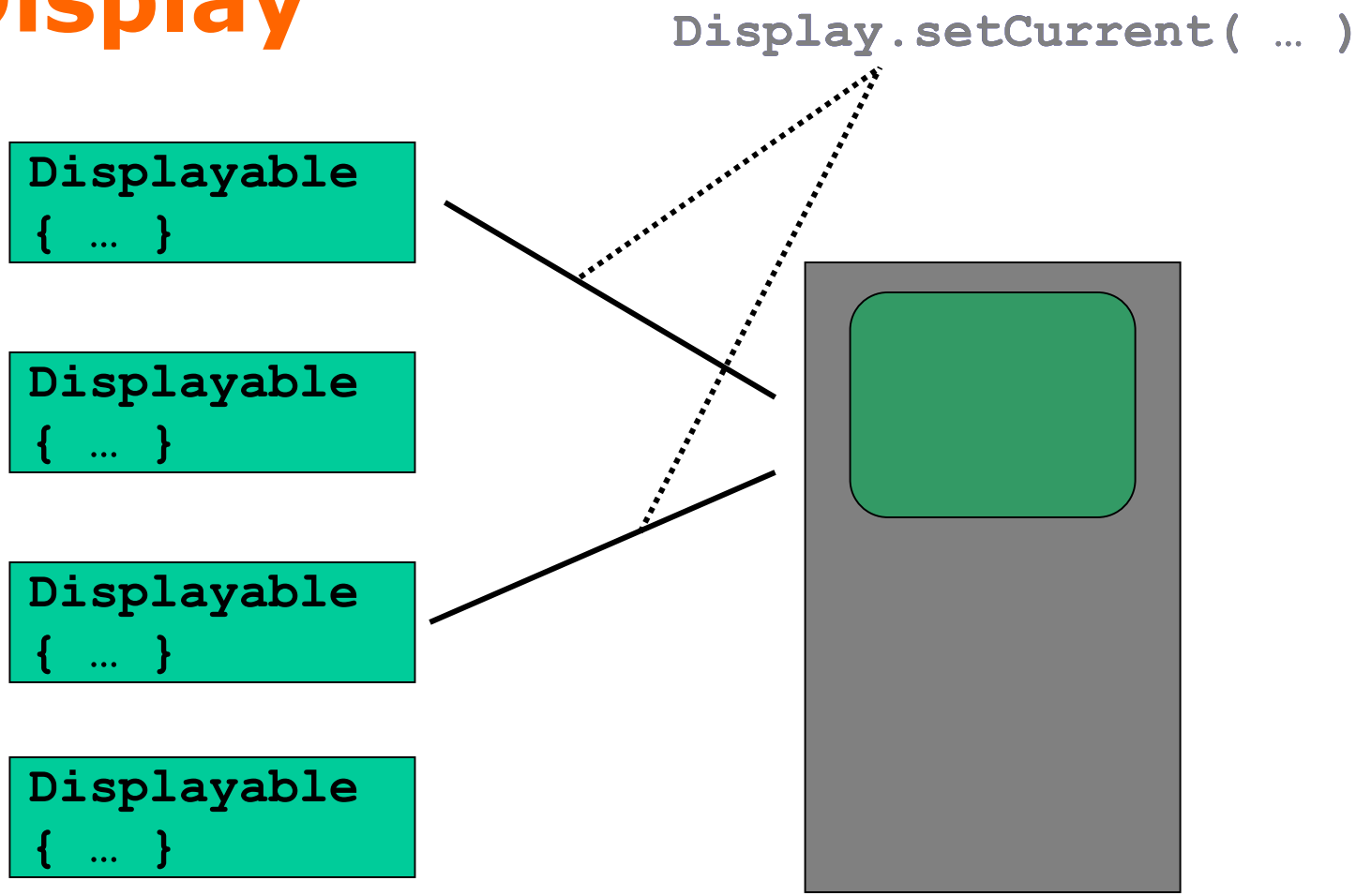


Canvas Characteristics

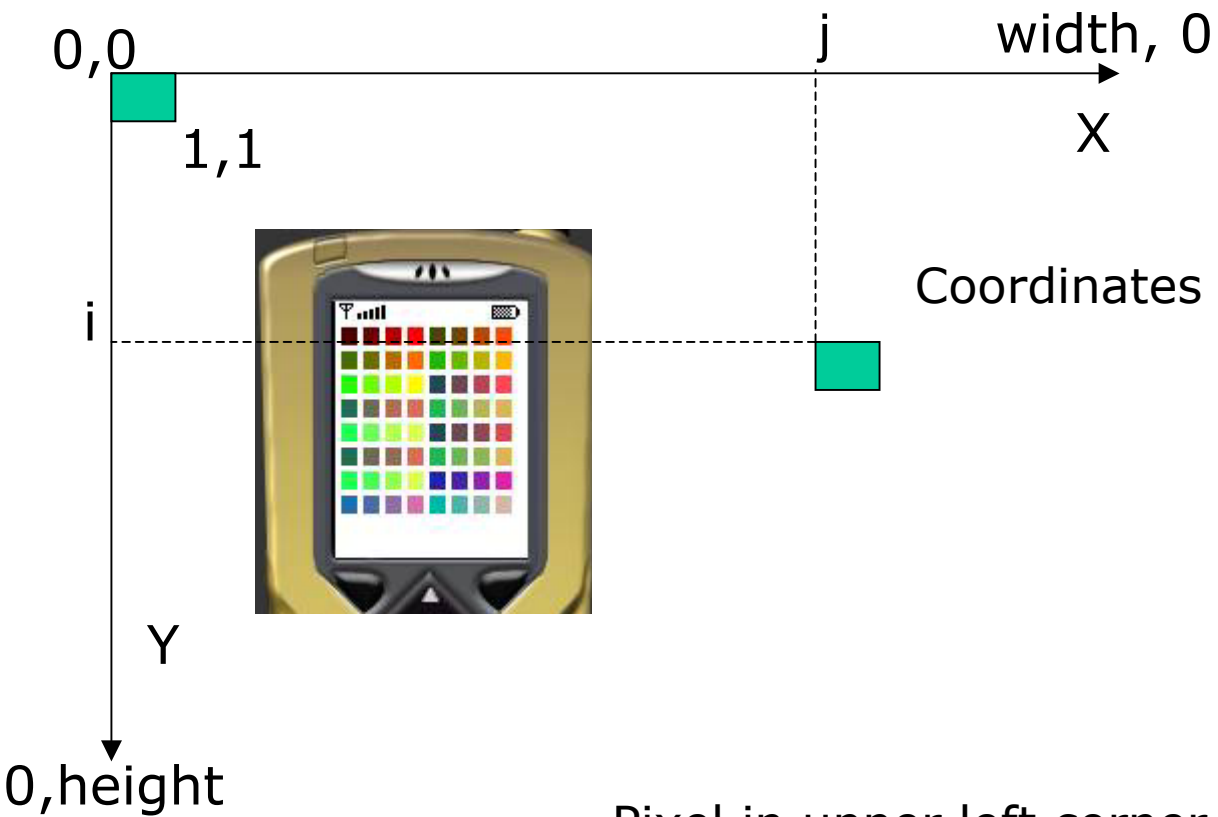
Canvas is a subclass of Displayable
-> may have soft buttons

```
public MyMIDlet(){
  display = Display.getDisplay(this);
  canvas  = new MyCanvas(this);
  canvas.addCommand(exitCommand);
  canvas.setCommandListener(this);
}
```

Display


Display.setCurrent(…)
Displayable
{ … }
Displayable
{ … }
Displayable
{ … }
Displayable
{ … }

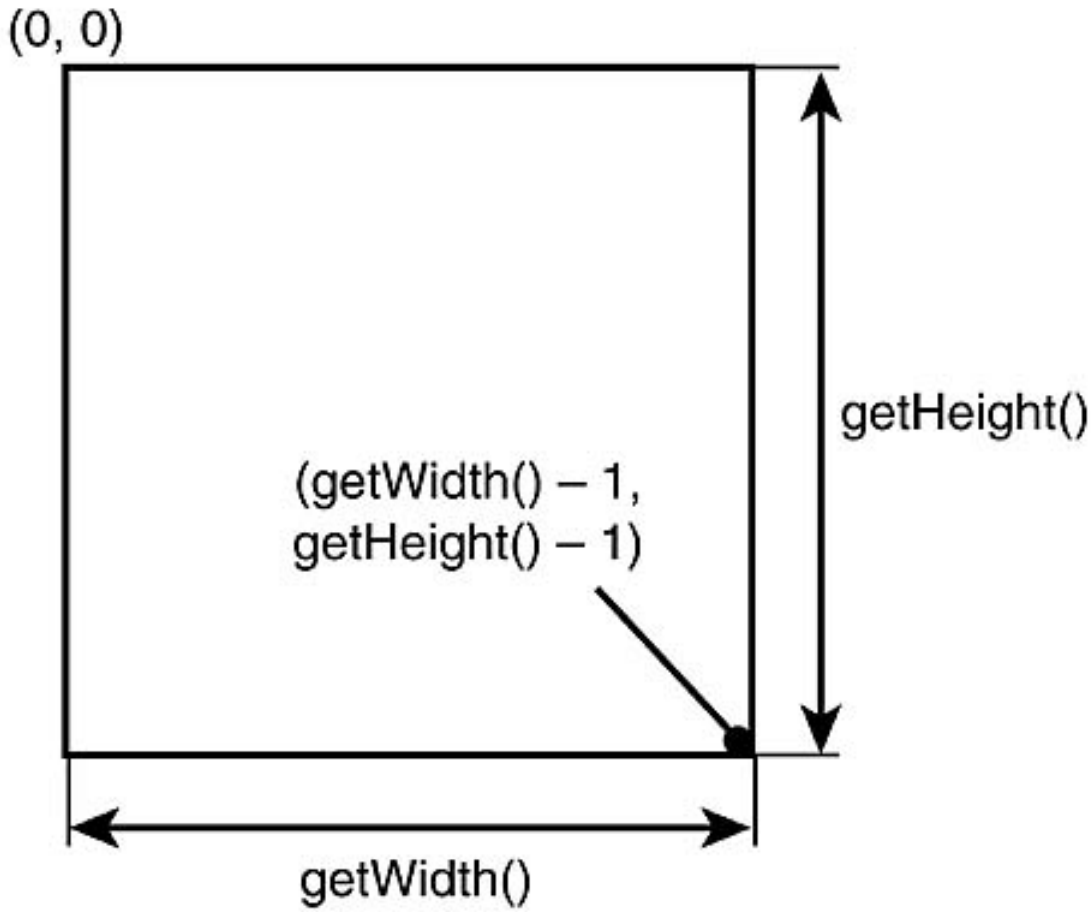
Graphics Coordinate System



Coordinates are between pixels

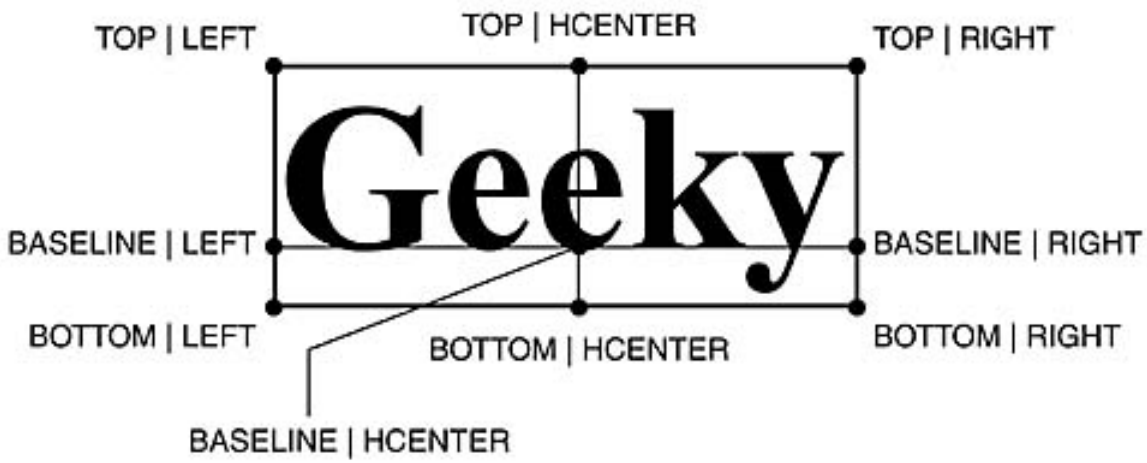
Pixel in upper left corner is between:
(0,0),(1,0),(0,1),(1,1)

Canvas Size



Anchor Points

```
g.drawString("Hello JPMA class", getWidth()/2, 0,
                                 g.TOP | g.HCENTER);
```



0 = TOP | LEFT

Low Level Events in Canvas

```
Class MyCanvas extends Canvas {
        ....
        void paint (Graphics g) {
        ...
}
        public void keyPressed(int key) {
                if (key == Canvas.KEY_NUM6 ||
            key == Canvas.FIRE)  ...

        }
        ...
}
```

Overview of Methods

javax.microedition.lcdui

Class Canvas

java.lang.Object
  |
  +--javax.microedition.lcdui.Displayable
        |
        +--javax.microedition.lcdui.Canvas

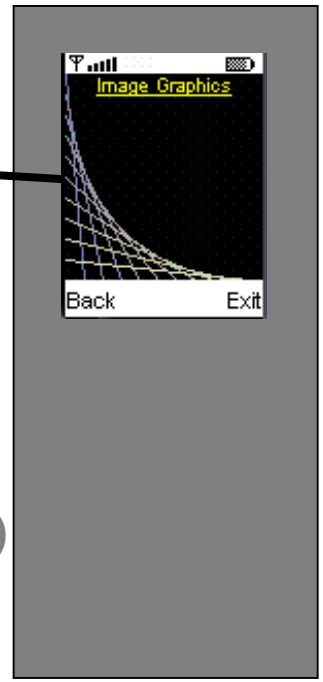
javax.microedition.lcdui

Class Graphics

java.lang.Object
  |
  +--javax.microedition.lcdui.Graphics

Paint and Repaint

```
void paint (Graphics g) {
        ...
}
```



1. Display setCurrent
2. When Canvas reappear (e.g. after alert)
3. Call of repaint

hideNotify and showNotify

Nightmare

Version 1

V

isColor

numColors

getWidth

getHeight

hasPointerEvents

...

n 3

Version 4

Layers

javax.microedition.lcdui.game

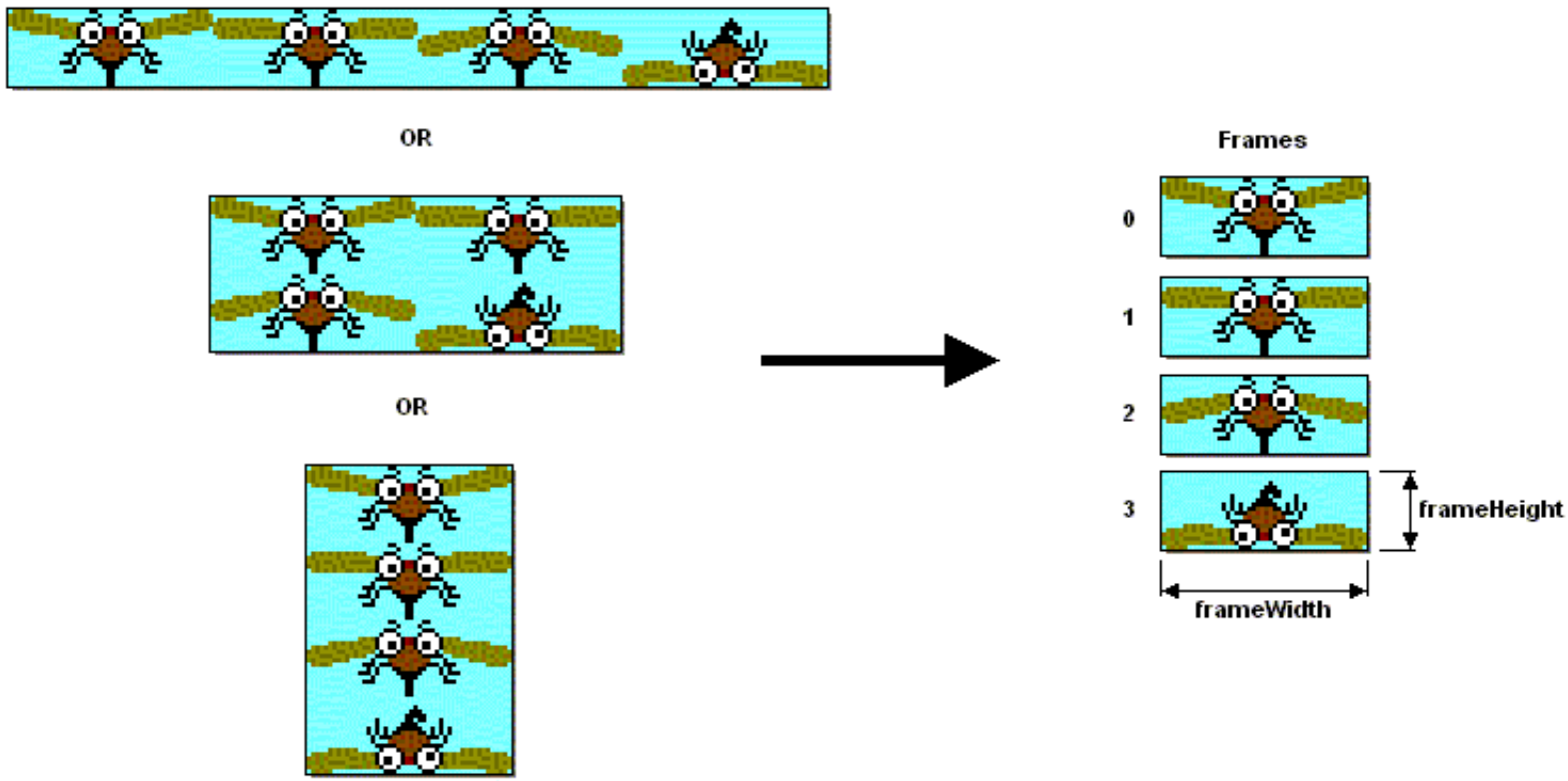
Class Layer

java.lang.Object
  |
  +--javax.microedition.lcdui.game.Layer

Direct Known Subclasses:
Sprite, TiledLayer

boolean	**isVisible**() Gets the visibility of this Layer.
void	**move**(int dx, int dy) Moves this Layer by the specified horizontal and vertical distances.
abstract void	**paint**(Graphics g) Paints this Layer if it is visible.
void	**setPosition**(int x, int y) Sets this Layer's position such that its upper-left corner is located at (x,y) in the painter's coordinate system.
void	**setVisible**(boolean visible) Sets the visibility of this Layer.

Sprites



public final int getFrame() Gets the current index in the frame sequence

public void nextFrame() Selects the next frame in the frame sequence

Sprite Example (1)

```
private Sprite s;

try {
   Image image = Image.createImage("/stars.png");
   s= new Sprite(image, 15, 15);
} catch (IOException ie) {}
```

Sprite Example (2)

```
private LayerManager mLayerManager;
mLayerManager = new LayerManager();
mLayerManager.append(s);
```

Sprite Example (3)

```
private LayerManager mLayerManager;
private Sprite s;
public StarCanvas() {
  super(true);
  mLayerManager = new LayerManager();
  try {
    Image image = Image.createImage("/stars.png");
    s= new Sprite(image, 15, 15);
  } catch (IOException ie) {}
  mLayerManager.append(s);
}
```

Break

Mini Project

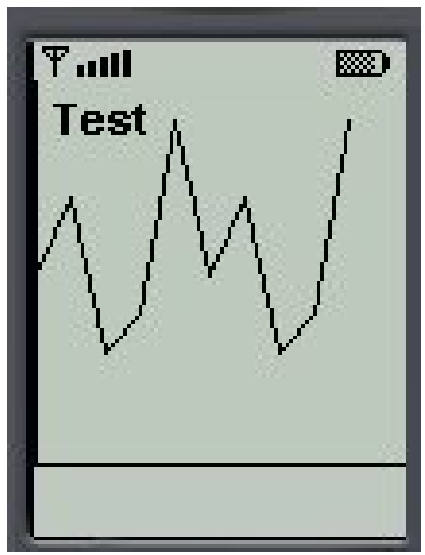
- February 19 – March 5 (12:00).
- Solutions must be turned in electronically by e-mail to Jørgen Staunstrup (**jst@itu.dk**) -> receipt.
- The project must be solved in groups of two persons.
- The solution must be approved (to get credit for the course).

Mini Project

```
public class Plot ... {

   public Plot(String title, int[] y) {

   }
}
```



Mini Project Description

http://www.itu.dk/courses/JPMA/F2004/miniproject.html

For steps 1-4 you may initialize y with a set of values given in the source program (through initialization or assignments).

1. Program Plot assuming y.length < width (number of pixels), and y values are in the range 0..Height (height of the display in pixels).

2. Extend the solution so the y values can be of arbitrary size and possibly negative.

3. Extend the solution so the length of the vector y can be much smaller and much larger than the width

4. Extend the solution with a zoom-function that makes it possible to show only a part of the graph.

5. Extend your solution with an input method that can read the values of y from an external file.

Reading From a File

```
InputStream is =
                    getClass().getResourceAsStream("large.txt");


try{
     int ch;
     while ((ch=is.read()) !=-1) {
          switch (ch) {
             case '\r': is.read();
             case '\n': v.addElement(new Integer(x));
                        x = 0;  break;
             default:   x = 10 * x + (ch - '0');
      }
   }
```

Next Week(s)

Exercise Sessions: Mini Project

Lecture: Low Level GUI and Threading
Garbage Collection