Lines and Coordinate Systems – Part 1
Chapter 1 – Ammeraal & Zhang
In Java, drawing a line can be done by the code:

\[
g.\text{drawLine} \left( x_A, y_A, x_B, y_B \right)
\]

- “drawLine” is a method in class “Graphics”
- “g” (graphics context) is normally a parameter of the “paint” method
- Note: \( g.\text{drawLine} \left( x_A, y_A, x_B, y_B \right) \) is the same as \( g.\text{drawLine} \left( x_B, y_B, x_A, y_A \right) \)
For example, we can draw the largest possible rectangle in a canvas:

```java
class CvRedRect extends Canvas {
    public void paint (Graphics g) {
        Dimension d = getSize (); // to set canvas dimensions
        int maxX = d.width - 1;
        int maxY = d.height - 1;

        g.drawString ("d.width = " + d.width, 10, 30);
        g.drawString ("d.height = " + d.height, 10, 80);
        g.setColor (color.red);
        g.drawLine (0, 0, maxX, 0);              // top edge
        g.drawLine (maxX, 0, maxX, maxY);  // right edge
        g.drawLine (maxX, maxY, 0, maxY);   // bottom edge
        g.drawLine (0, maxY, 0, 0);       // left edge
    }
}
```

Note: same as `drawRect (0, 0, maxX, maxY);`
Result:

RedRect

Frame (200, 100)
d.width = 192
d.height = 73

Canvas (192, 73)

4 pixels

Lines and Coordinate Systems
1.1 Device Coordinate System

- So the rectangle size drawn by `g.drawRect(x, y, maxX, maxY)` is \((maxX + 1)\) by \((maxY + 1)\).

\[
d.\text{width} = maxX + 1 \\
d.\text{height} = maxY + 1
\]
• The smallest rectangle we can possibly draw is a square 2 pixels by 2 pixels: 
  :
  :
using `g.drawRect (x, y, 1, 1)`

• To draw one dot: `g.drawLine (x, y, x, y)`
To fill a rectangle of size \( w \) by \( h \), use `g.fillRrect(x, y, w, h)`

Note: `g.drawRect(x, y, w, h)` draws a slightly larger triangle

Lines and Coordinate Systems
The “filling problem”: suppose that we want to fill a triangular part of the square.

What color to fill the edge pixels?
1.2. Logical Coordinate System

- Suppose that we want the origin at the left-bottom corner, as in the usual math coordinate system

- In this case we simply do: \( y' = \max Y - y \)

- Note: the x-axis is unchanged since it has the same direction as in the math coordinate system
Contrasting the two coordinate systems:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Data type</th>
<th>Feature</th>
<th>Positive y-axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>Lower-case letters</td>
<td>Integer</td>
<td>Discrete</td>
</tr>
<tr>
<td>Logical</td>
<td>Upper-case letters</td>
<td>Float</td>
<td>Continuous</td>
</tr>
</tbody>
</table>
1.3. Converting between logical and device coordinates

1. Truncating:

```c
int ix (float x)
{
    return (int) x; //logical → device
}

float fx (int x)
{
    return (float) x + 0.5F; //device → logical
}
```

Example: \( ix(2.8) = 2 \) \( fx(2) = 2.5 \)
1.3. Converting between logical and device coordinates

2. Duplicating:

```java
int ix (float x)
{
    return Math.round (x);       //logical \rightarrow device
}

float fx (int x)
{
    return float (x);                 //device \rightarrow logical
}
```

Example: $ix (2.8) = 3$  $fx (3) = 3.0$
1.3. Converting between logical and device coordinates
   ◦ For both (1) and (2) above,

\[ |x - fx(i(x))| \leq 0.5 \]

meaning that the maximum loss of precision is 0.5

◦ However, in most (but not all) cases rounding produces more accurate results than truncating, so we will use that method (1)
Example program: “Triangles.java” (from the course textbook, Section 1.3)

// Triangles.java: This program draws 50 triangles inside each other.
import java.awt.*;
import java.awt.event.*;

Lines and Coordinate Systems
Example program: “Triangles.java” (from the course textbook, Section 1.3)

```java
public class Triangles extends Frame
{
    public static void main(String[] args){new Triangles();}
    Triangles()
    {
        super("Triangles: 50 triangles inside each other");
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e){System.exit(0);}});
        setSize (600, 400);
        add("Center", new CvTriangles());
        show();
    }
}

class CvTriangles extends Canvas
{
    int maxX, maxY, minMaxXY, xCenter, yCenter;

    void initgr()
    {
        Dimension d = getSize();
        maxX = d.width - 1; maxY = d.height - 1;
        minMaxXY = Math.min(maxX, maxY);
        xCenter = maxX/2; yCenter = maxY/2;
    }
    int iX(float x){return Math.round(x);}
    int iY(float y){return maxY - Math.round(y);}

Lines and Coordinate Systems
• Example program: “Triangles.java” (from the course textbook, Section 1.3)

```java
public void paint(Graphics g)
{
    initgr();
    float side = 0.95F * minMaxXY, sideHalf = 0.5F * side,
        h = sideHalf * (float)Math.sqrt (3),
        xA, yA, xB, yB, xC, yC,
        xA1, yA1, xB1, yB1, xC1, yC1, p, q;
    q = 0.05F;
    p = 1 - q;
    xA = xCenter - sideHalf;
    yA = yCenter - 0.5F * h;
    xB = xCenter + sideHalf;
    yB = yA;
    xC = xCenter;
    yC = yCenter + 0.5F * h;
    for (int i=0; i<50; i++)
    {
        g.drawLine(iX(xA), iY(yA), iX(xB), iY(yB));
        g.drawLine(iX(xB), iY(yB), iX(xC), iY(yC));
    }
}
```
Example program: “Triangles.java” (from the course textbook, Section 1.3)

```java
    g.drawLine(iX(xC), iY(yC), iX(xA), iY(yA));
    xA1 = p * xA + q * xB;
    yA1 = p * yA + q * yB;
    xB1 = p * xB + q * xC;
    yB1 = p * yB + q * yC;
    xC1 = p * xC + q * xA;
    yC1 = p * yC + q * yA;
    xA = xA1; xB = xB1; xC = xC1;
    yA = yA1; yB = yB1; yC = yC1;
```
• Example program: “Triangles.java” (from the course textbook)
  ◦ It shows the advantage of using floating point logical coordinates
  ◦ An important principle in this example is:
    • Step i + 1: Computation is performed on the floating point result of Step i, i.e.:
Example program: “Triangles.java” (from the course textbook)

- Rather than

```
float → float → float → ... → float
```

  ↓ rounding ↓ rounding ↓ rounding ↓ rounding ↓

  ↓ int ↓ int ↓ int ↓ int ↓

- This approach will accumulate rounding errors, so the previous approach is better
Questions/comments/How about the Kentucky Derby? Nothing?!? Seriously, I have student loans to pay off people!!!