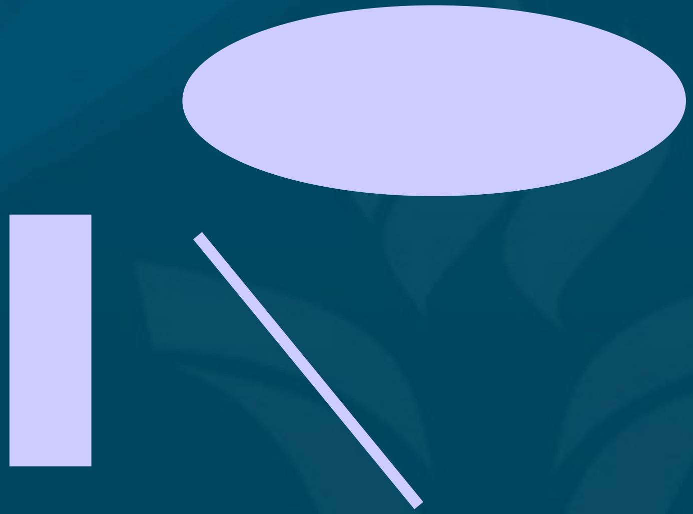
Java2D API: Graphics and Graphics2D

19 Feb 2010
CMPT166
Dr. Sean Ho
Trinity Western University

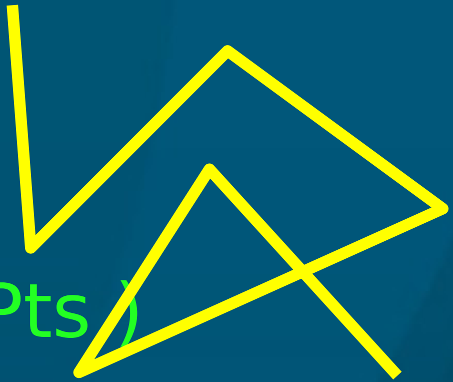
TRINITY WESTERN UNIVERSITY

Basic drawing: Graphics class

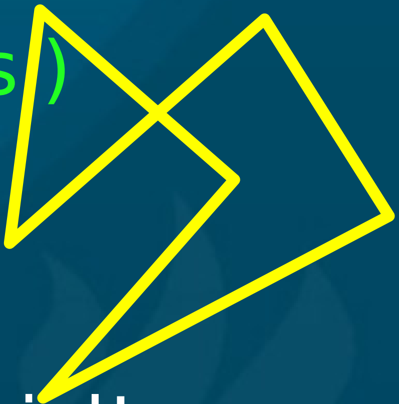
- Subclass JFrame and override paint()
  - Or JPanel and override paintComponent()
- Current drawing context: Graphics object (g)
  - Pen colour: setColor()
  - Also: (x,y)-origin, clip, font, XOR-mode
- Basic drawing commands:
  - draw or fill:
  - Line, Rect, Oval, Arc

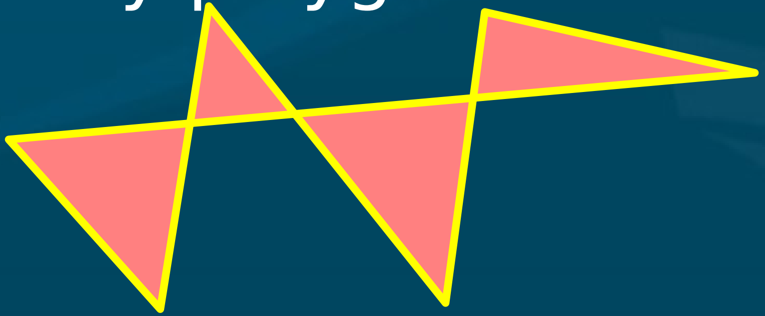


Polylines and polygons



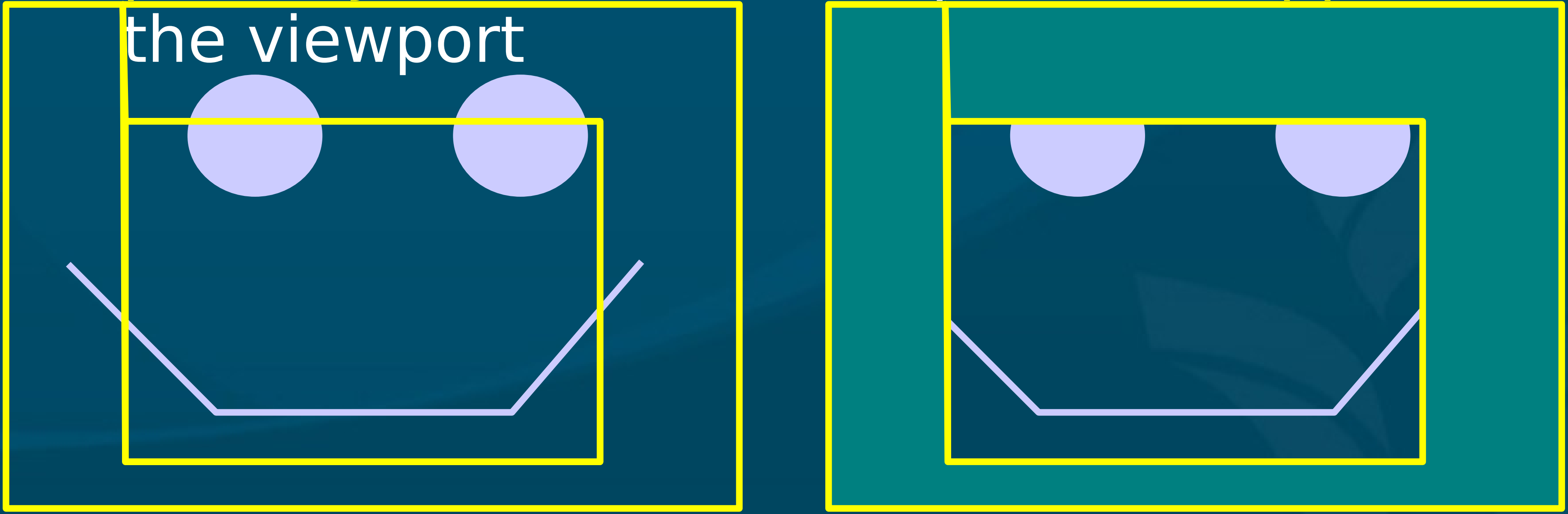
- drawPolyline(int[] x, int[] y, int numPts)
  - Arrays of x and y coordinates
  - Draws connected line segments
- drawPolygon(int[] x, int[] y, int numPts)
  - Connects last point to first point
- Also fillPolygon(...)
  - Filling an arbitrary polygon is not trivial! (tessellation)





Clipping

- The current clip is the viewport of the canvas which is being drawn on
  - Anything drawn outside the clip is not visible
  - Primitives (ovals, polygons, etc.) that lie partially outside the viewport are clipped to the viewport



Setting the clip region

- setClip(int x, y, w, h)
  - Sets the clip region to the given rectangle
  - Useful if you want to "protect" parts of the window/panel from being drawn over
- setClip() is also overloaded to take a Shape
  - For more complex clip regions
    - Polygon, Line2D, Arc2D, CubicCurve2D, etc.
  - See documentation for Shape interface

Drawing text

- drawString(String text, int x, int y)
  - Uses current colour and font
- setFont(Font f)
  - Sets the current font in the graphics context
- Font class:

  Hello, World!

    - import java.awt.Font;
    - new Font(Font.SANS_SERIF, Font.PLAIN, 18)
  - Family (can also specify name as string)
  - Style: plain, italic, bold
  - Size: in points

Reading images from file

- ImageIO library understands jpg, gif, png, bmp
  - **import javax.ImageIO;**
- BufferedImage stores the image data:

```
BufferedImage img;
try {
    img = ImageIO.read(
      new File( “apples.jpg” ) );
} catch (IOException e) { }
```

  - May raise IOException if file doesn't exist, etc

Drawing images on the canvas

- g.drawImage(
  Image img, int x, int y, ImageObserver obs)
  - The ImageObserver is usually null
- Or select a sub-rectangle of the image to draw and scale it to fit within a rectangle on canvas:
- g.drawImage(Image img, int
  dest x_1, dest y_1, dest x_2, dest y_2,
  src x_1, src y_1, src x_2, src y_2, ImageObserver)
  - Source rectangle in the image
  - Destination rectangle in the canvas

Java2D: more in Graphics2D

- The Graphics2D class extends Graphics and adds more functionality for
  - Fancier primitives: cubic curves, etc.
  - Coordinate transforms: skewing, shearing
  - Colour management
  - Text layout
  - Filtering images: sharpening/blurring, etc.
  - More: see Java2D API tutorial