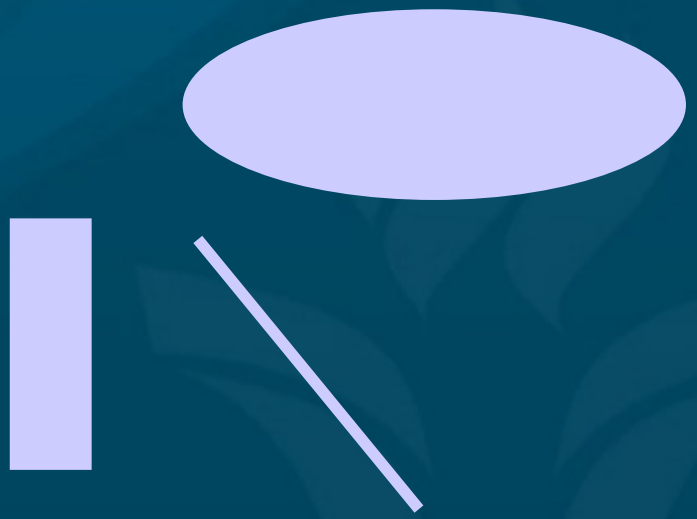
Java2D API: Graphics and Graphics2D

15 Feb 2008
CMPT166
Dr. Sean Ho
Trinity Western University

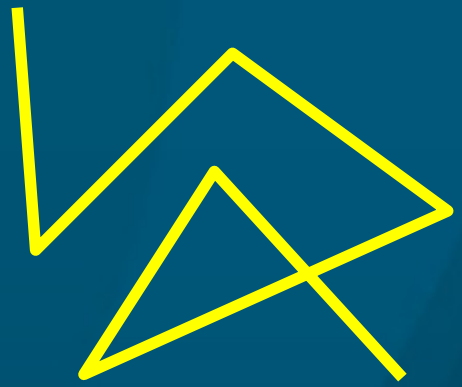
TRINITY WESTERN UNIVERSITY

Basic drawing: Graphics class

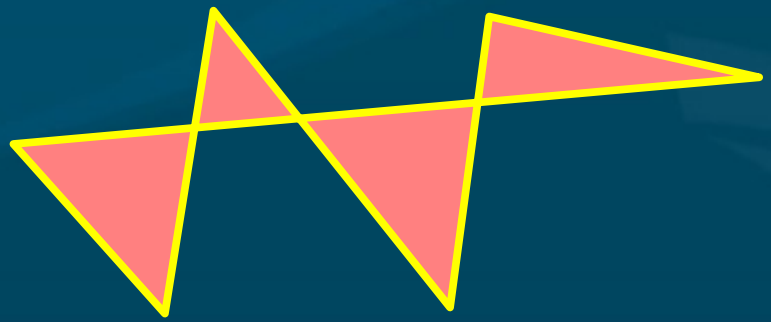
- Subclass JFrame and override paint()
  - Or subclass JPanel and override paintComponent()
- Current drawing context: Graphics object (g)
  - Pen colour: setColor()
  - Also: (x,y)-origin, clip, font, XOR-mode
- Basic drawing commands:
  - drawLine(), Rect, Oval, Arc



Polylines and polygons

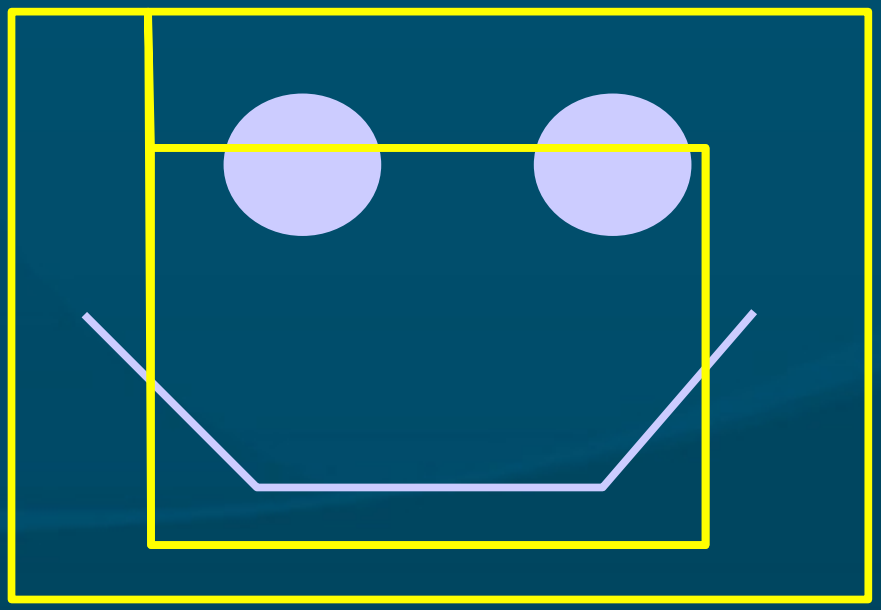


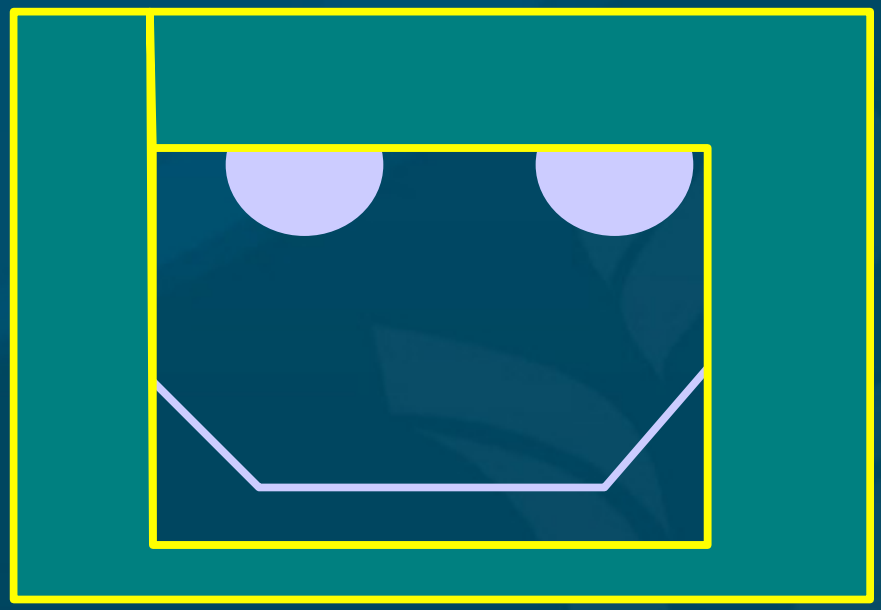
- drawPolyline(int[] x, int[] y, int numPts)
  - Arrays of x and y coordinates
  - Draws connected line segments
- drawPolygon(int[] x, int[] y, int numPts)
  - Connects last point to first point
- Also fillPolygon()
  - Filling an arbitrary polygon is not trivial! (tessellation)



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:
  - 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

Hello, World!

Working with images: read from file

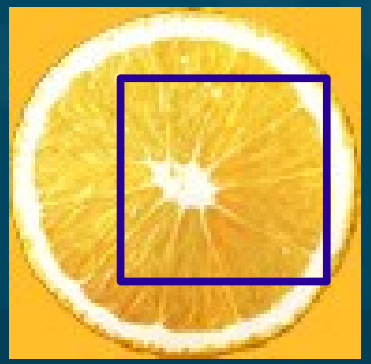
- ImageIO library understands jpg, gif, png, bmp
  - import javax.ImageIO;
- BufferedImage is the object that stores image data

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

  - IOException may happen if file doesn't exist, etc.
    - More on exceptions and file I/O soon

Drawing an image on the canvas

- g.drawImage(
  Image img, int x, int y, ImageObserver obs)
  - The ImageObserver is usually null
- Can also select a sub-rectangle of the image to draw
  - And scale it to fit in a target rectangle of the canvas
- g.drawImage(Image img, int
  dst_x1, dst_y1, dst_x2, dst_y2,
  src_x1, src_y1, src_x2, src_y2, 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