CS 349
Affine Transforms

Byron Weber Becker
Spring 2009

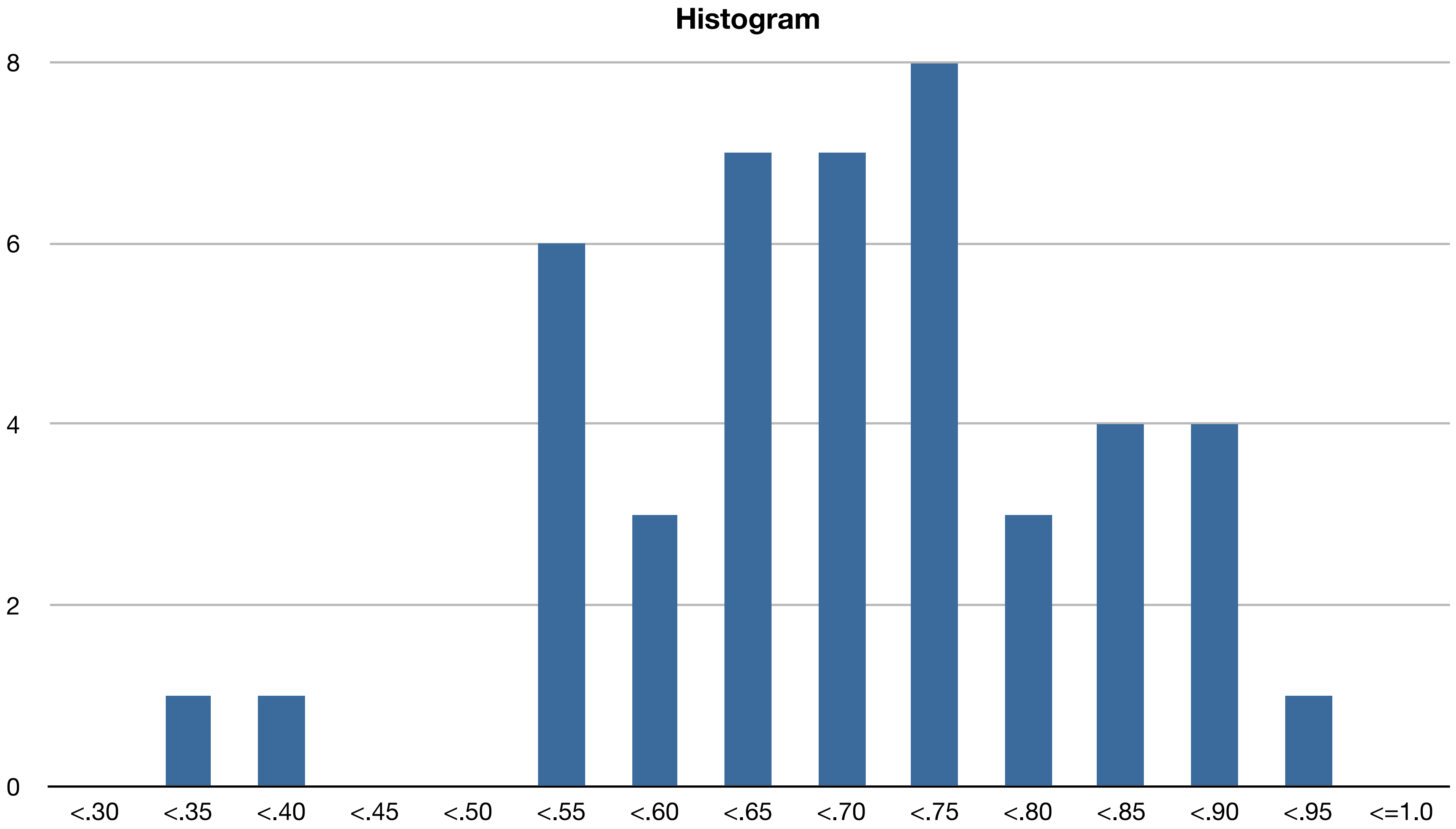
University of Waterloo

6-July-09 Announcements

- Jaime won’t be holding regular office hours this week. Please email him for an appointment instead.
- Agenda:
  - Hand back midterm
  - Review midterm
  - UI Video
  - Affine Transforms

Midterm 2 Results

Average: 67.78%
StdDev: 12.65%



Midterm 2 Results

- Q1: History 59.0% (out of 8 rather than 9)
- Q2: MVC 62.4%
  - Misinterpreted intent of distributing the code
  - Didn't provide the interface
  - Didn't update the view when it was added
- Q3: Undo Manager 72.3% (out of 9 rather than 10)
  - Many didn't provide the interface (adjusted marking)
  - Many didn't clear the redo stack after pushing a new command
  - Many didn't do any error checking
- Q4: Design Patterns 85.0%

Midterm Results

- Q5: Videos 79.4%
- Q6: Cut ‘n Paste Transferables 91.5%
- Q7: Long Tasks 57.7%
  - a) marked very leniently
  - c) done poorly
    - Most put *everything* into a thread or *everything* into invokeLater. Neither is correct.

```
public void actionPerformed(ActionEvent e)
{
   Thread t = new Thread() {
      public void run() {
           Image img = getImage(…);

           SwingUtilities.invokeLater(
              new Runnable() {
                   public void run() {
                      picture.setImage(img);
                   }  // run
            }  // Runnable

      }  // run
   }  // Thread

   t.start();
}
```

UI Video

- GestureTek's gesture-based TV Remote
- Intro by David Rust-Smith

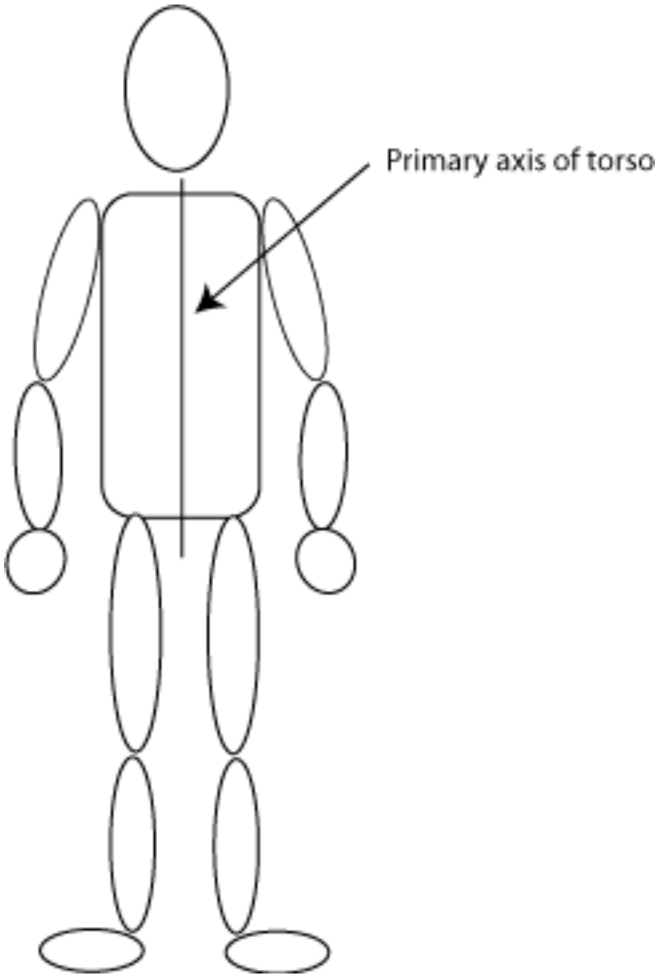
Direct Manipulation

- Graphical user interfaces often allow direct manipulation of on-screen objects with the mouse.
- Need to perform many *inside tests* to implement DM:
  - Easy for rectangles
  - Not so easy for other shapes, or if images have been scaled, or rotated, …

```
for(Item item : displayList)
{   if (item.contains(mouse.x, mouse.y))
    {   …
    }
}
```

  - Need a more general strategy.

A5 Preview



Goal: A coherent framework

- Define graphics primitives
  - inside tests
  - drawing routines
- Allow a defined set of operations
  - translation
  - scaling
  - rotation
  - (reflection, skewing)

  - These are all linear transformations. Provide them to the GPU via a matrix.
- Basis for all modern printer and windowing systems as well as 3D graphics.

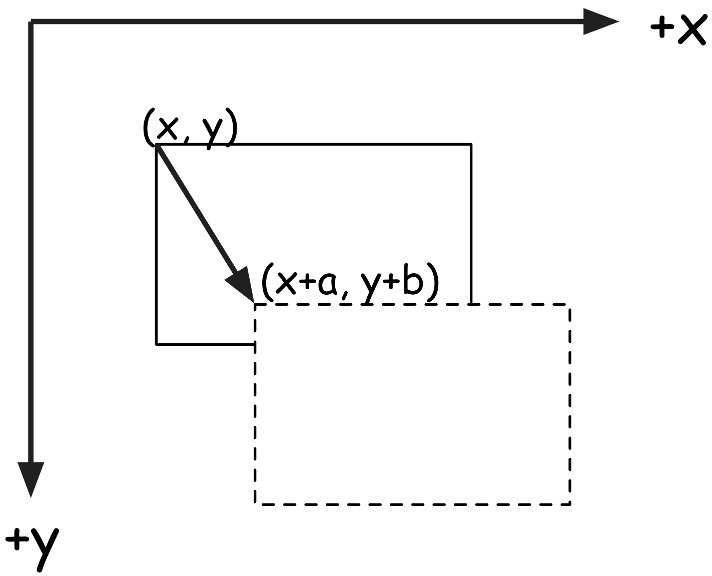
Fundamentals

- Point vs. Vector
- Vector operations: add, scalar multiply, point + vector
- Point operations: subtraction

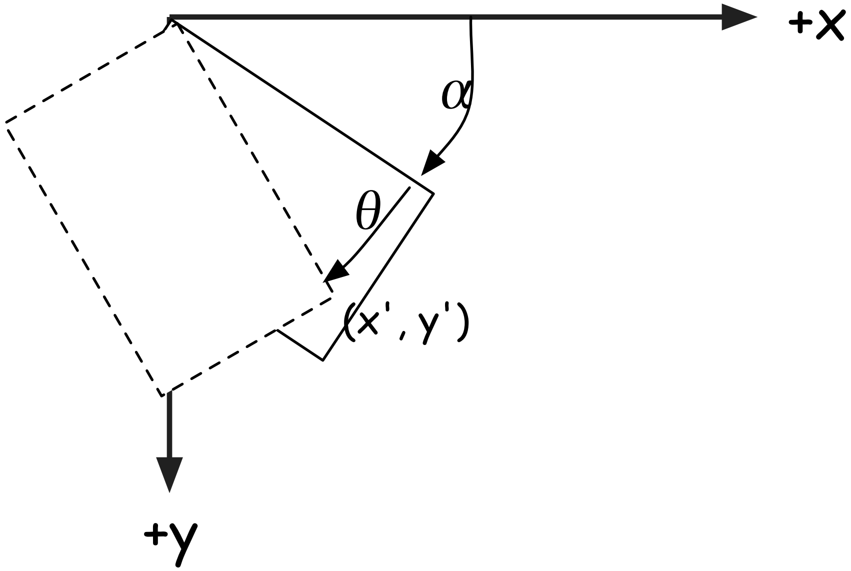
Homogenous Coordinates

- point:
- vector:
- Do the above operations still work?
- Can we define shapes in these terms?

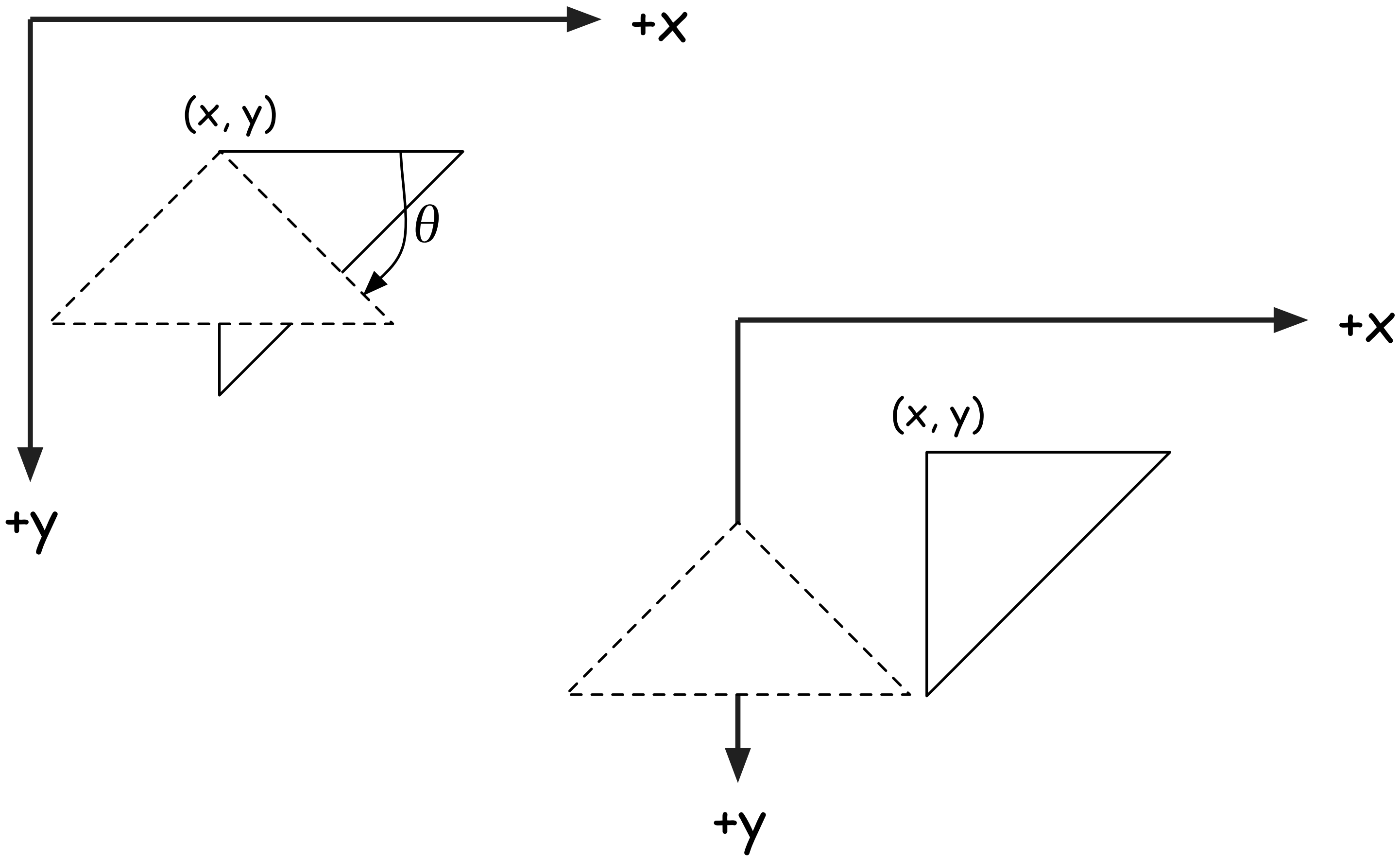
Translation



Rotation



Combining Transformations



08-Jul-09 Announcements

- Midterm Remark requests to Byron by July 20.
- Minor simplification/clarification to A4.
- Read *The Anti-Mac Interface* for Monday's class
- Class cancelled on Wednesday, July 15.
- Agenda:
  - Advising Email: only 7 responses from CS349 students
  - UI Video "siftables"
  - Summary of affine transformations do far
  - Composing transformations
  - Scaling, Reflections
  - Inside Tests, Mouse stuff
  - Scene graphs

Summary of 6-July

- Represent points as (x, y, 1) and vectors as (x, y, 0).
- Translation matrix

$$\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}$$

- Rotation matrix
  - in radians
  - positive angle rotates +x to +y

$$\begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

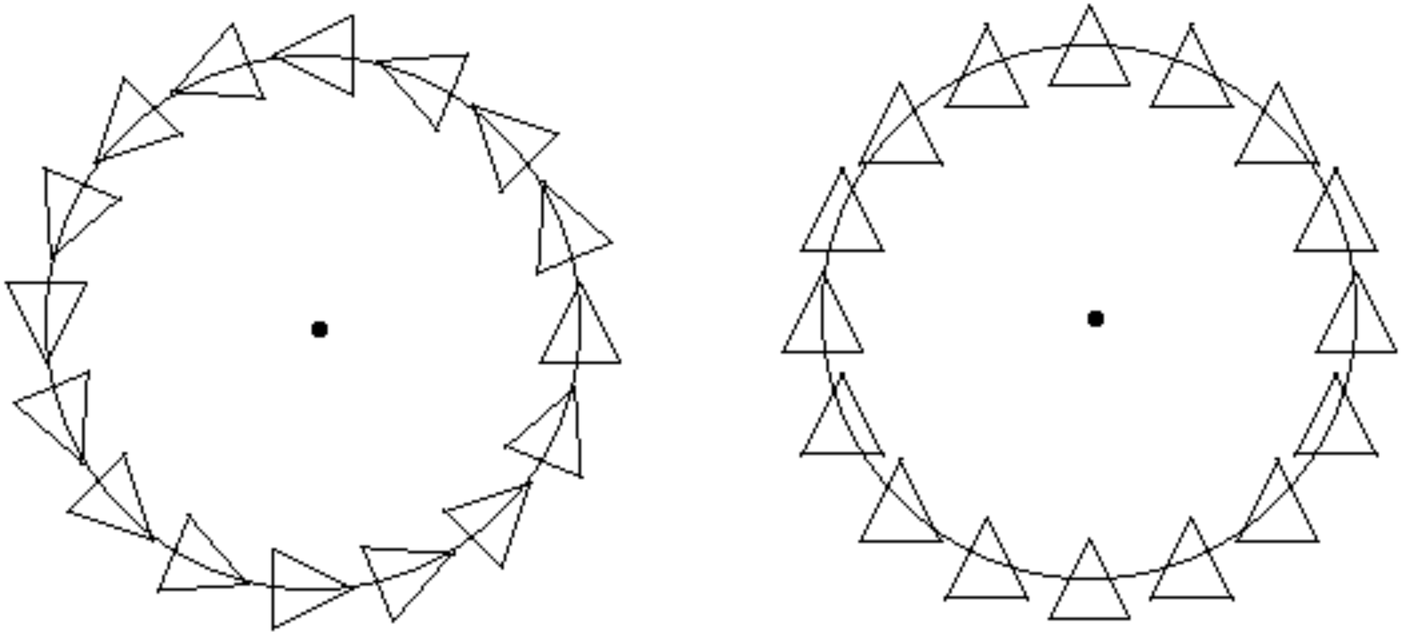
- Combining matrices
  - use associative property
  - order matters! (Matrix multiplication is associative but not commutative.)

$$T_{(x,y)}\left(R_{\Theta}\left(T_{(-x,-y)} \cdot p\right)\right) = \left(T_{(x,y)}R_{\Theta}T_{(-x,-y)}\right) \cdot p$$

```
AffineTransform t = new AffineTransform();
t.translate(x, y);
t.rotate(θ);
t.translate(-x, -y);
g2.setTransform(t);
```

Class Exercise

- Develop the transformations to animate a triangle so it rotates in a circle in two different ways:

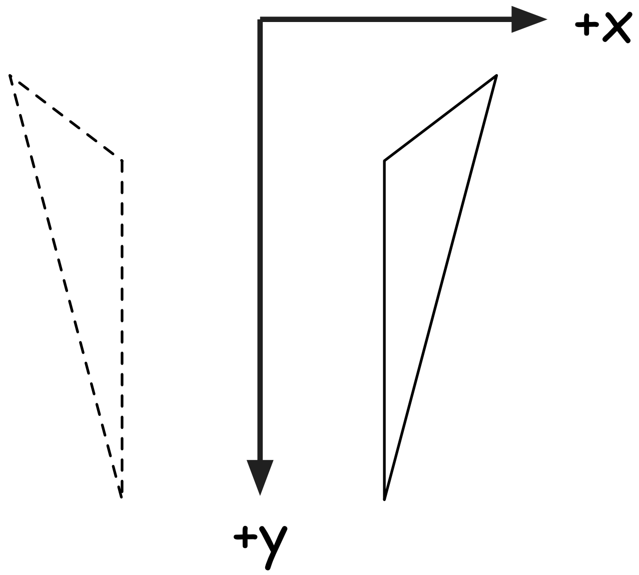


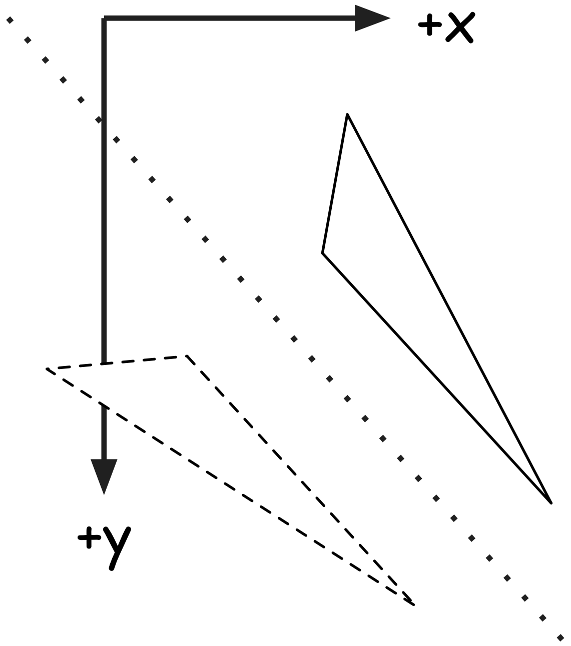
```
class Canvas extends JComponent {
    private double theta = 0.0;  // Updated by a Timer
    private int cx = 200;        // Timer omitted from code
    private int cy = 200;
    private int radius = 100;

    public Canvas() {
        new Timer(400, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                theta += .2;
                repaint();
            }
        }).start();
    }
```

// continued on next slide

```
    public void paintComponent(Graphics g) {
        Graphics2D g2 = (Graphics2D) g;
        g2.clearRect(0, 0, this.getWidth(), this.getHeight());

        AffineTransform t = AffineTransform.getTranslateInstance(cx, cy);
        g2.setTransform(t);
        g2.fillOval(-2, -2, 4, 4);
        g2.drawOval(-radius, -radius, radius * 2, radius * 2);

        // Define transformation here



        g2.setTransform(t);
        // Draw Triangle
        g2.drawLine(0, -15, 15, 15);
        g2.drawLine(-15, 15, 15, 15);
        g2.drawLine(-15, 15, 0, -15);
    }
}
```

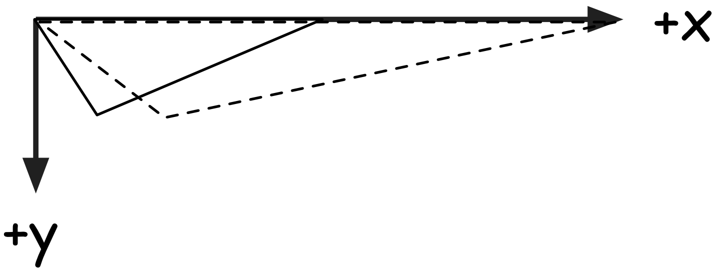
Reflection





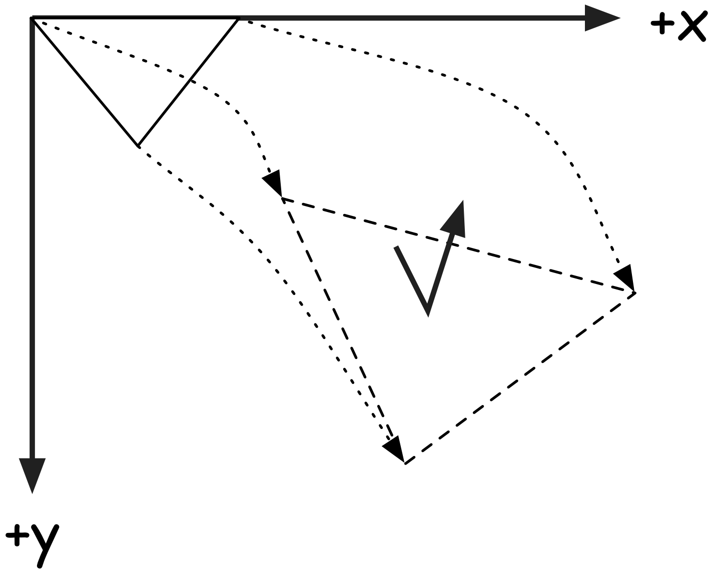
Scaling

- Issues:
  - Amount
  - Direction
  - Centre



Inside Tests

- Option 1
- Option 2
- Optimizations
- Obtaining/Maintaining M^{-1}



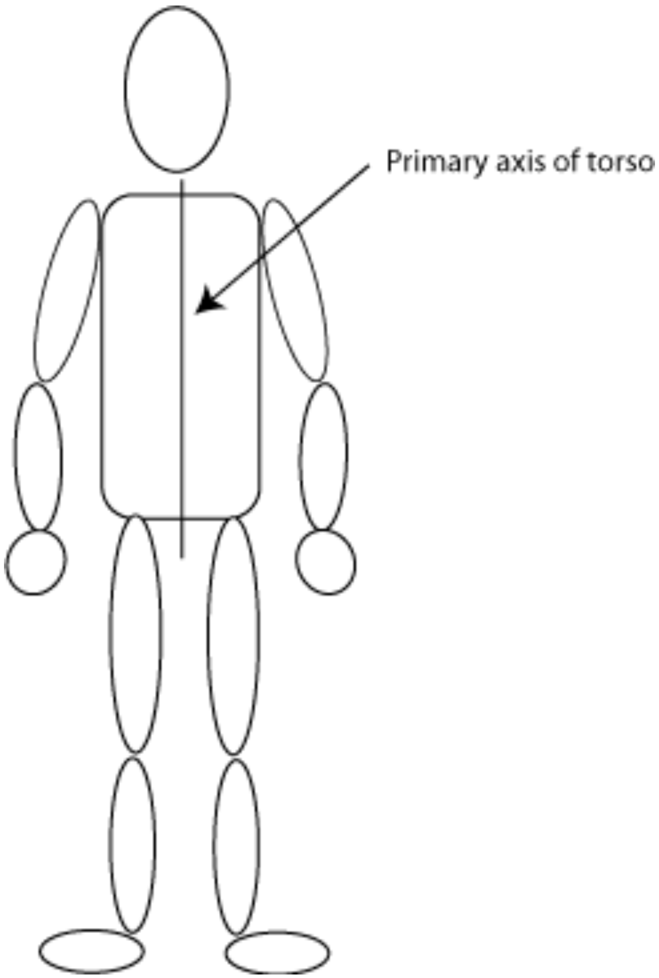
“Nearness” to a line

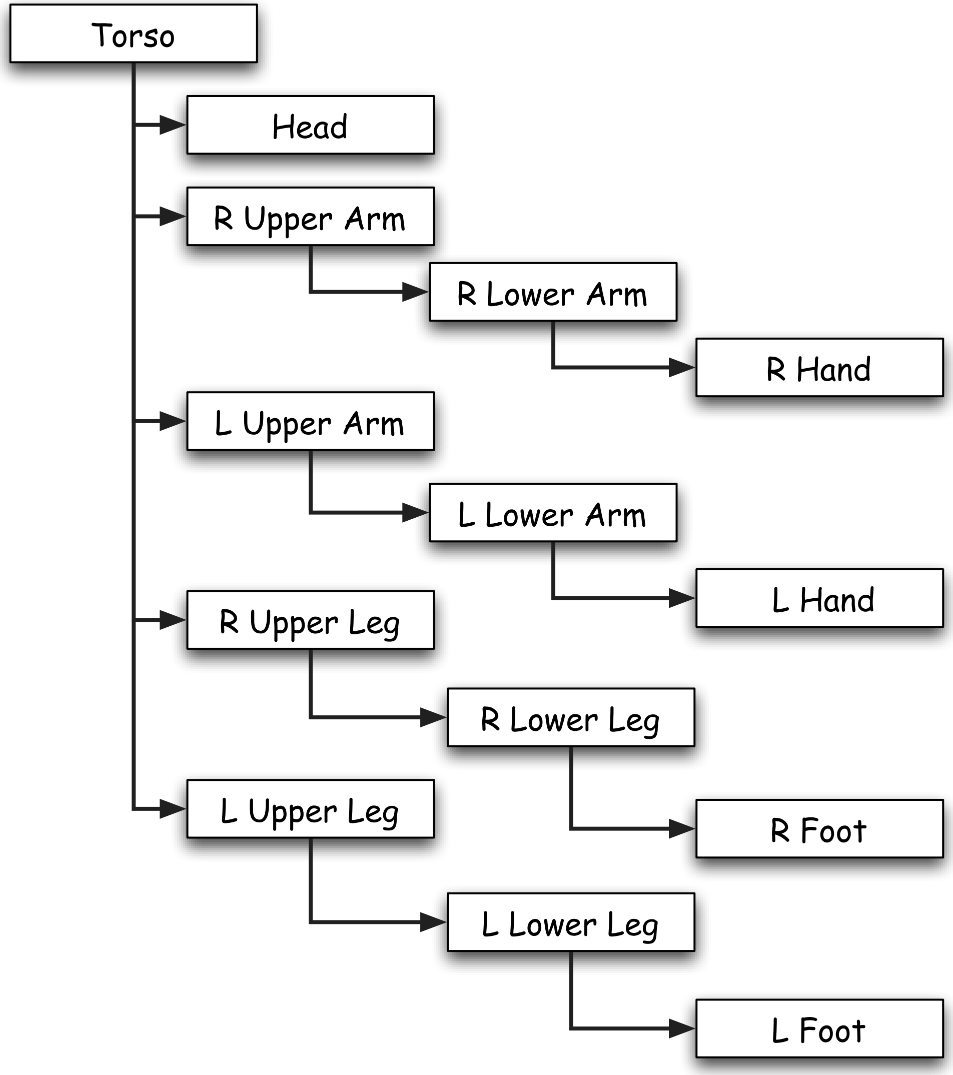
- World coordinates vs. Window coordinates
- If they are the same…
- If they are radically different...

Tracking the Mouse

- Keep track of the last mouse position (starting with mouseDown event)
- For each mouseDrag event:
  - Calculate Δx and Δy
  - form a translation matrix; concatenate it to the shape’s matrix
  - store the new mouse location

Scene Graphs





- Torso
  - Head
  - R Upper Arm
    - R Lower Arm
      - R Hand
  - L Upper Arm
    - L Lower Arm
      - L Hand
  - R Upper Leg
    - R Lower Leg
      - R Foot
  - L Upper Leg
    - L Lower Leg
      - L Foot