




Multiple Coordinate Spaces (Revisited)

CS 355: Interactive Graphics and Image Processing

Forward (Drawing)

- Object Coordinates


Lab #2
Lab #3



\mathbf{O}_i

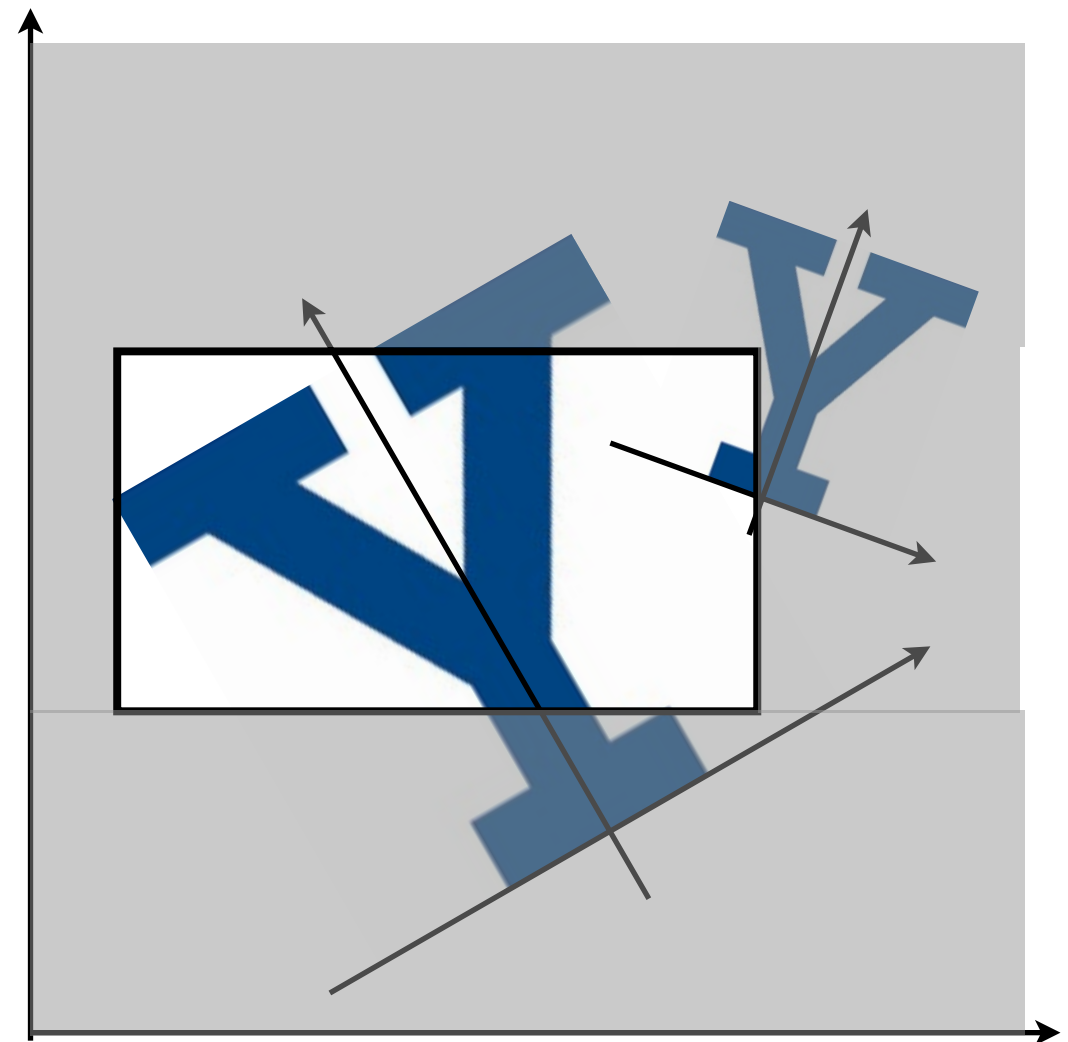
- World Coordinates

Lab #4



\mathbf{V}

- Viewing Coordinates



Backward (Selecting)

- Object Coordinates



$$\mathbf{O}_i^{-1}$$

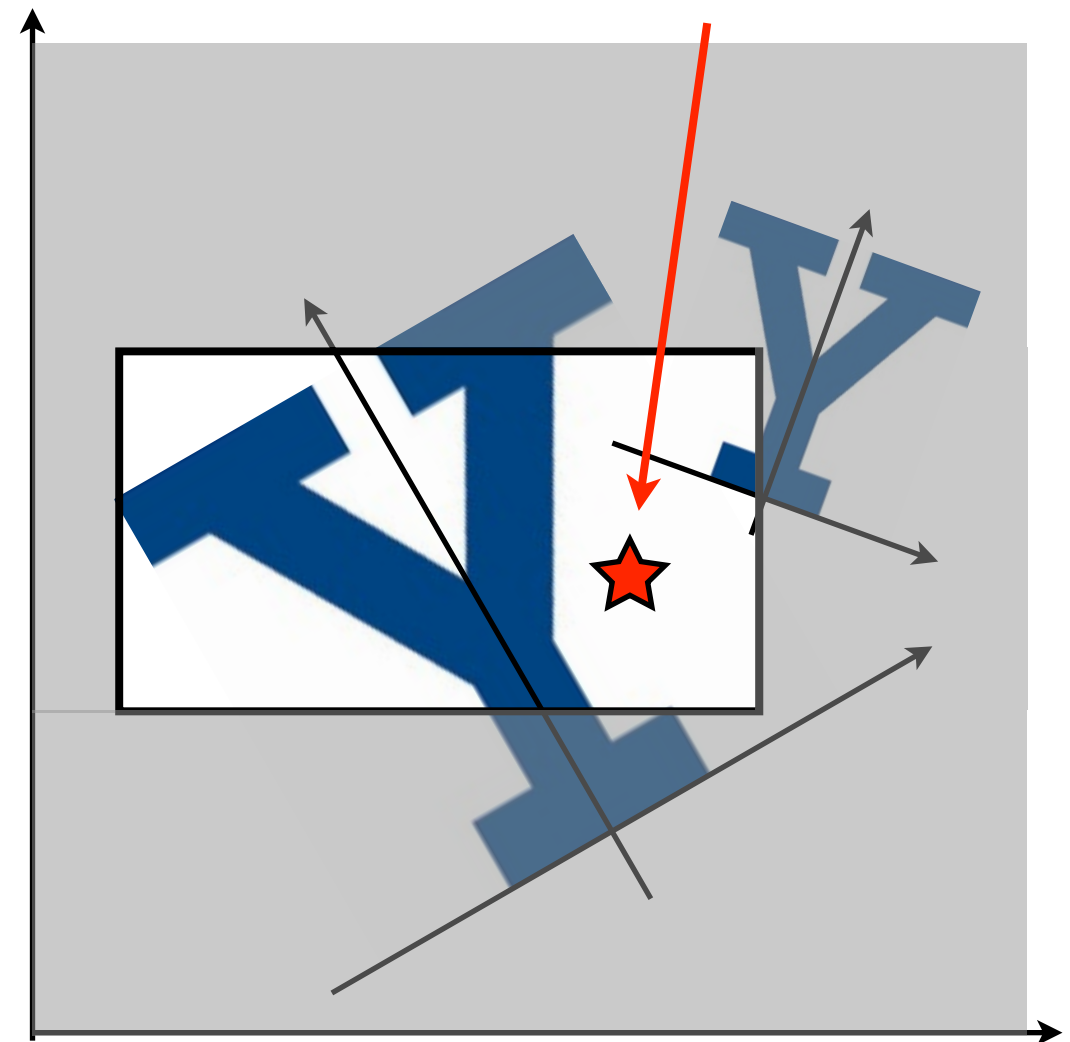
- World Coordinates



$$\mathbf{V}^{-1}$$

- Viewing Coordinates

Mouse click
in screen coordinates



Object to World

- To transform a point from object space to the world space:
- Rotate to orientation
- Translate to position

$$\mathbf{O}_i = \overset{\text{order} \leftarrow}{\mathbf{T}(\mathbf{c}_i) \mathbf{R}(\theta_i)}$$

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}(\mathbf{c}) = \begin{bmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

World to Object

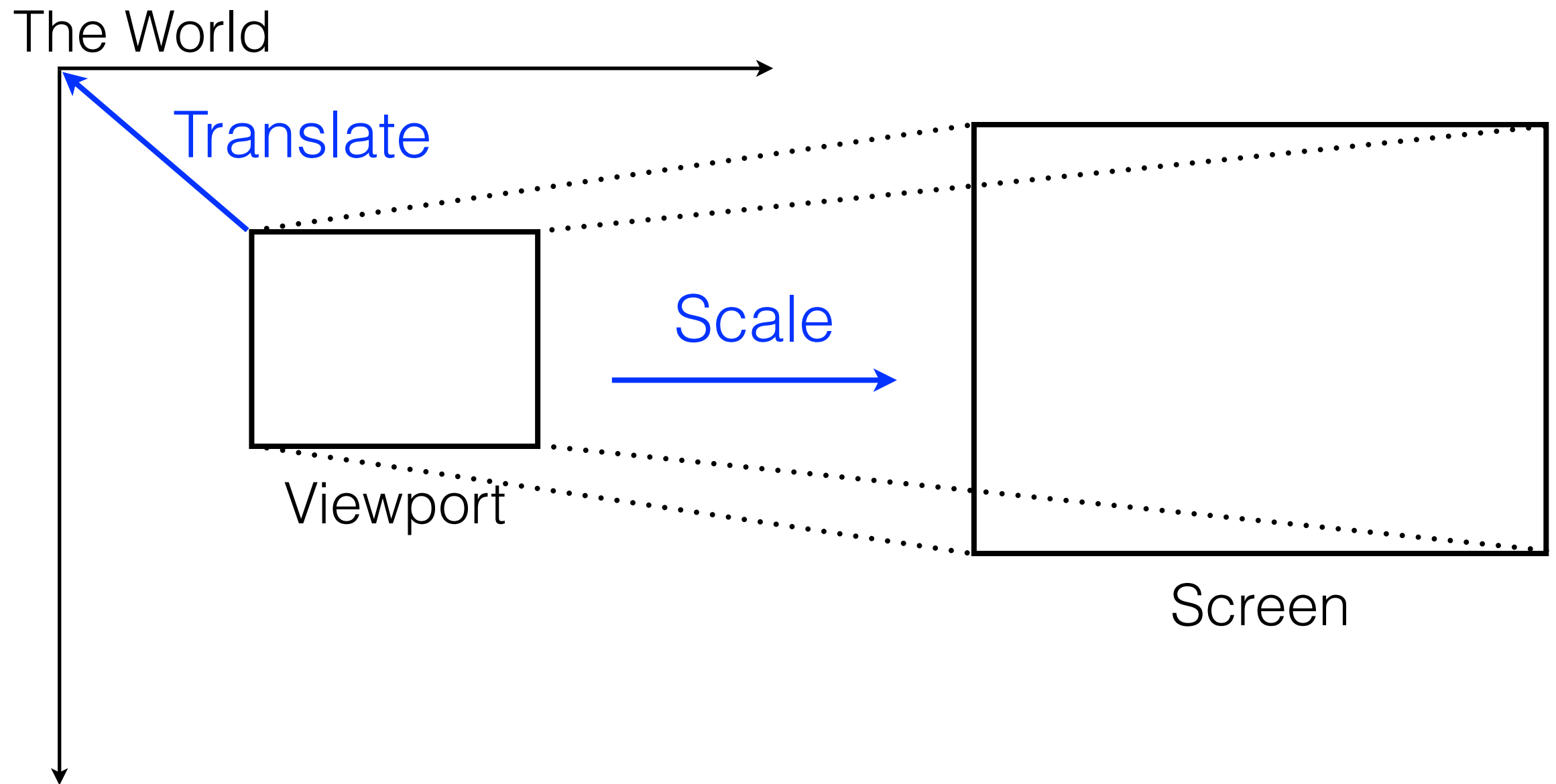
- To transform a point from world space to object space:
- Translate from position to origin
- Rotate back from orientation

$$\mathbf{O}_i^{-1} = \mathbf{R}(-\theta_i) \mathbf{T}(-\mathbf{c}_i)$$

$$\mathbf{R}(-\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}(-\mathbf{c}) = \begin{bmatrix} 1 & 0 & -c_x \\ 0 & 1 & -c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Viewports



World to View

- To transform a point from world space to viewing space:

$$\mathbf{V} = \mathbf{S}(f) \overset{\text{order}}{\longleftarrow} \mathbf{T}(-\mathbf{p})$$

- Translate from origin of viewport to origin of the world
- Scale by the zoom factor

$$\mathbf{T}(-\mathbf{p}) = \begin{bmatrix} 1 & 0 & -p_x \\ 0 & 1 & -p_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{S}(f) = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

View to World

- To transform a point from viewing space to world space:

$$\mathbf{V}^{-1} = \mathbf{T}(\mathbf{p}) \mathbf{S}(1/f)$$

- Scale by the inverse zoom factor

$$\mathbf{T}(\mathbf{p}) = \begin{bmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

- Translate from origin of the world to origin of the viewport

$$\mathbf{S}(1/f) = \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Drawing

- Transform from object space to world space
- Transform from world space to view space

$$\mathbf{M} = \overset{\text{order}}{\overleftarrow{\mathbf{V}}} \mathbf{O}_i$$

$$\mathbf{p}_{\text{view}} = \mathbf{M} \mathbf{p}_{\text{object}}$$

Selecting

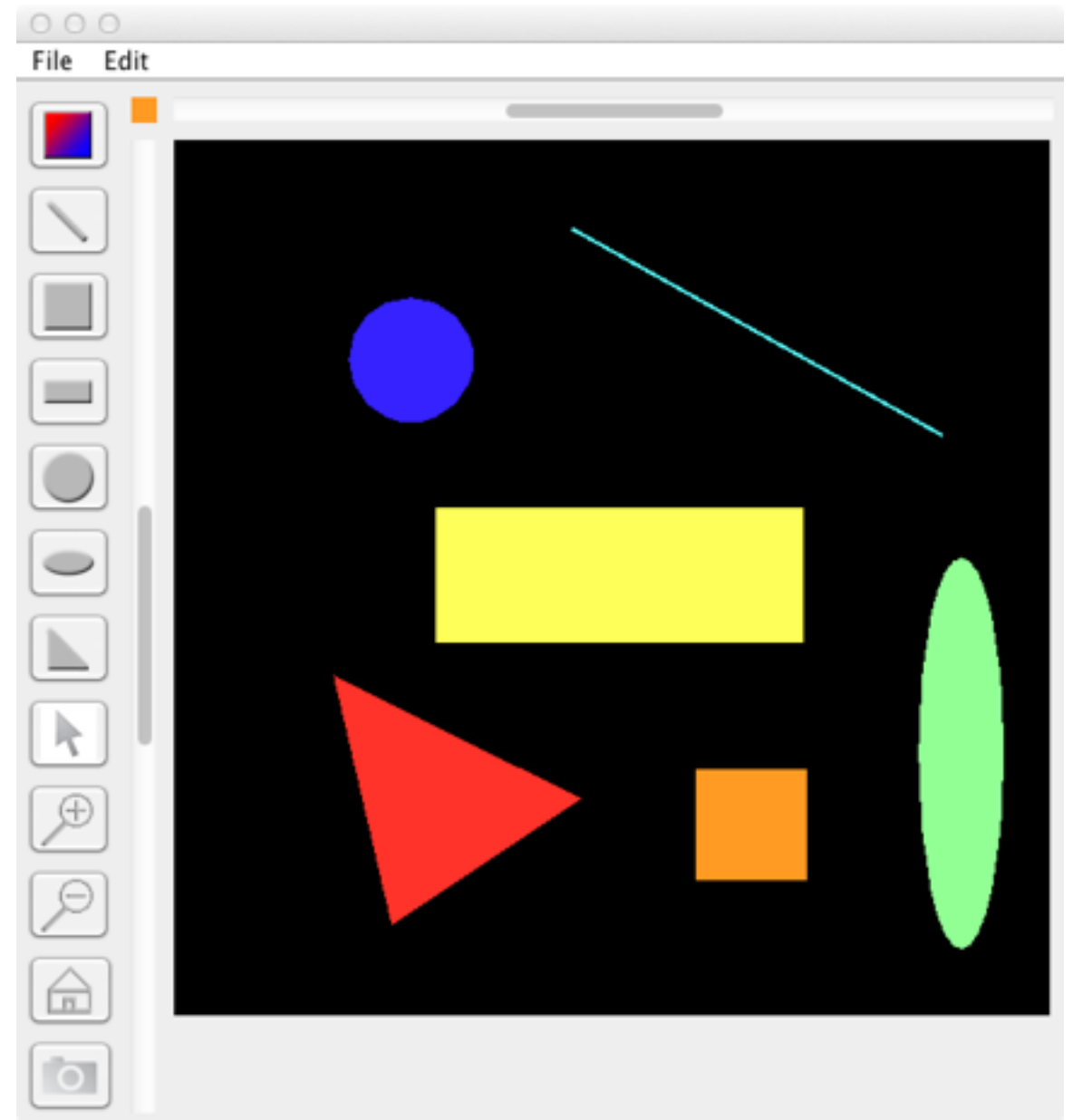
- Transform from view space to world space
- Transform from world space to object space

$$\mathbf{M}^{-1} = \overset{\text{order}}{\overleftarrow{\mathbf{O}_i^{-1} \mathbf{V}^{-1}}}$$

$$\mathbf{p}_{\text{object}} = \mathbf{M}^{-1} \mathbf{p}_{\text{view}}$$

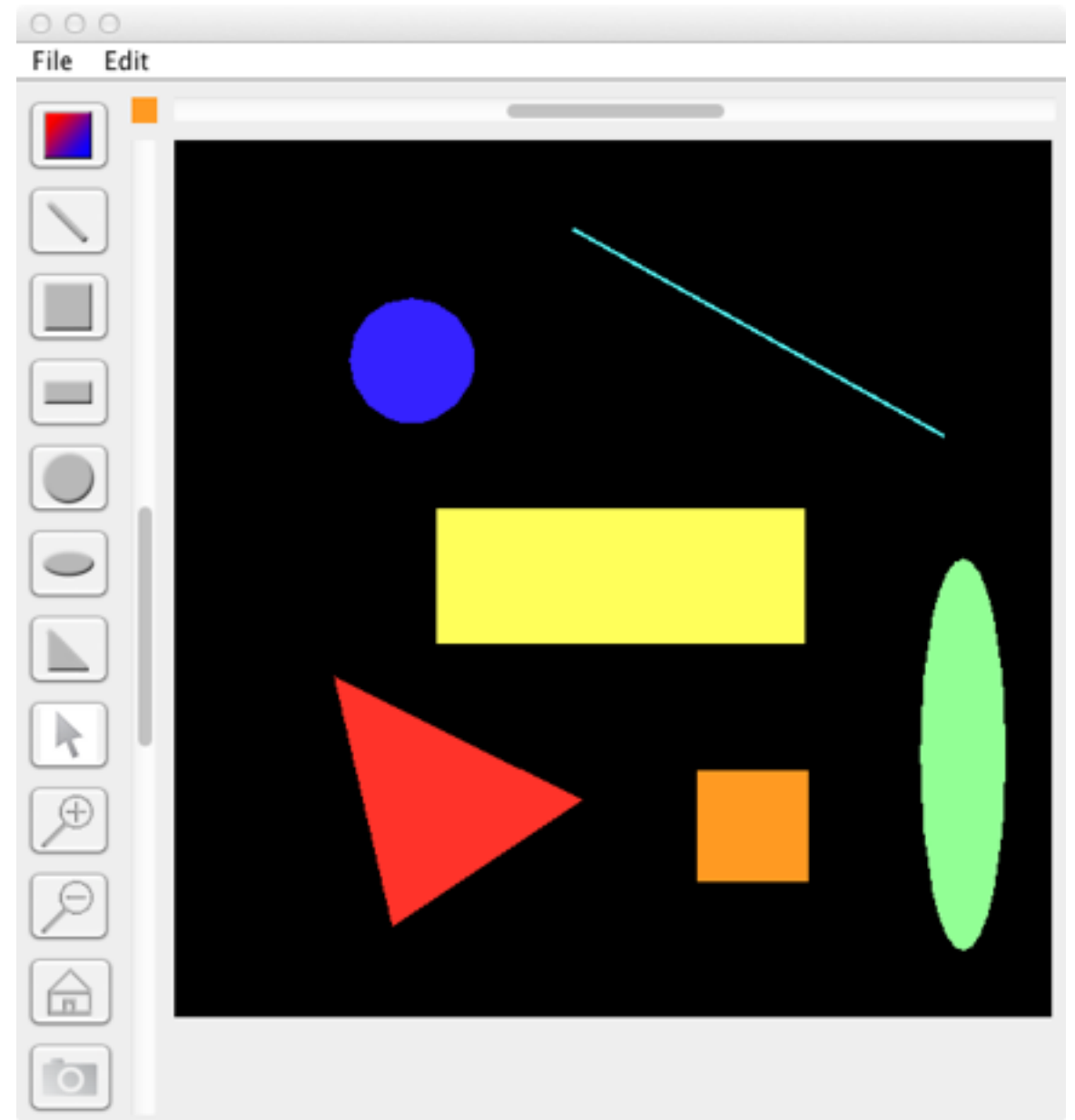
Lab #4 Preview

- New buttons for zoom in / out
- Scroll bars now used
- Maximum drawing area (“world”) is 2048 x 2048
- Viewing area is still 512 x 512
- Zoom levels: 25%, 50%, 100% (default), 200%, 400%
- Zooming in /out should keep the center of the current view still in the center of the new view
- Exception: entire viewing area must stay within the maximum drawing area



Lab #4 Preview

- Build AffineTransform objects for viewing transformation and inverse (these live in view/controller)
- Using AffineTransforms
 - Constructor that takes the 6 elements of the matrix
 - Concatenation (matrix multiply)
 - Transform point(s)
 - Pass to Graphics2D for drawing
 - No “rotate”, “translate”, “scale”



Coming up...

- Transformation hierarchies
- 3D!