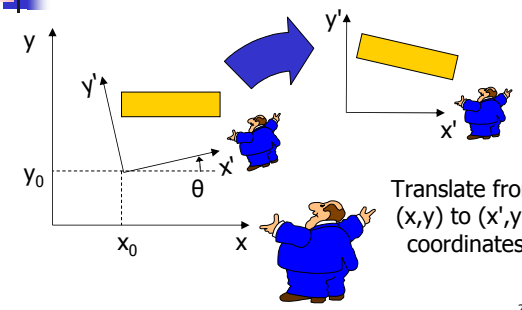
Coordinate System Transforms

- Object descriptions
  - Defined in source coordinates
  - Map to destination coordinates
  - For now, all Cartesian coordinates
- Applications
  - Describe view of part of whole
  - Map primitive into drawing

1

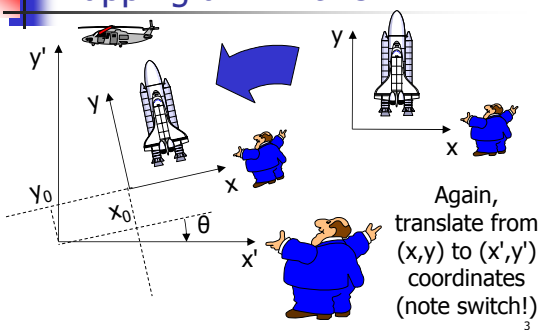
Mapping a View



Translate from (x,y) to (x',y') coordinates

2

Mapping a Primitive



Again, translate from (x,y) to (x',y') coordinates (note switch!)

3

Mapping Procedure

1) Translate origin (x_0,y_0) of x'y' system to origin of xy system

$$\mathbf{T}(-x_0,-y_0)=\begin{bmatrix}1 & 0 & -x_0\\ 0 & 1 & -y_0\\ 0 & 0 & 1\end{bmatrix}$$

2) Rotate x' axis onto the x axis

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

4

Composite Mapping Transform

$$\mathbf{M}=\mathbf{R}(-\theta)\cdot\mathbf{T}(-x_0,-y_0)$$

$$=\begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix}1 & 0 & -x_0\\ 0 & 1 & -y_0\\ 0 & 0 & 1\end{bmatrix}$$

$$=\begin{bmatrix}\cos\theta & \sin\theta & -x_0\cos\theta-y_0\sin\theta\\ -\sin\theta & \cos\theta & x_0\sin\theta-y_0\cos\theta\\ 0 & 0 & 1\end{bmatrix}$$

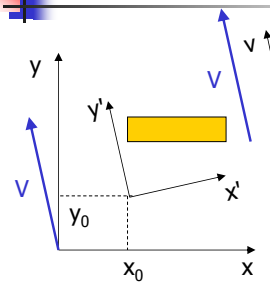
5

Alternate Angle Specification

- So far, assume we know θ
- But there are alternate ways
  - Have Cartesian vector V
  - Direction same as y' axis
  - Relative to xy system

6

Direction Vector Calculation



$$\mathbf{v} = \frac{\mathbf{V}}{|\mathbf{V}|} = (v_x, v_y)$$

$$\mathbf{u} = (v_y, -v_x) = (u_x, u_y)$$

Orthogonal unit vectors **u** and **v** describe x'y' system orientation

7

Direction Vector Transforms

1) Translate origin (x_0,y_0) of x'y' system to origin of xy system

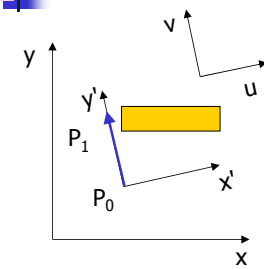
$$\mathbf{T}(-x_0, -y_0) = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

2) Rotate x' axis onto the x axis

$$\mathbf{R} = \begin{bmatrix} u_x & u_y & 0 \\ v_x & v_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

8

Direction from Points



$$\mathbf{v} = \frac{\mathbf{P}_1 - \mathbf{P}_0}{|\mathbf{P}_1 - \mathbf{P}_0|} = (v_x, v_y)$$

$$\mathbf{u} = (v_y, -v_x) = (u_x, u_y)$$

Direction vector from two (user-specified?) points

9

Raster Transformations

- What if data in raster form?
  - So far, have been transforming points, lines, etc.
- Translation
  - Just move pixels (bitBlt)
- Rotation (other than 90º steps)
  - More difficult (supersampling?, image/signal processing?)

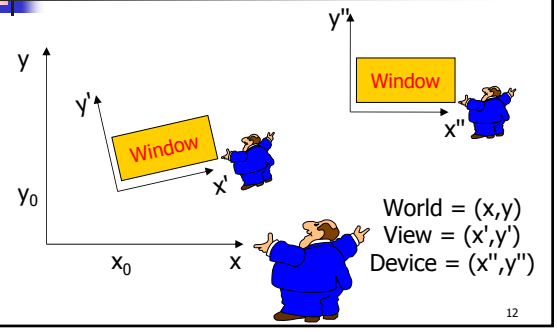
10

Viewing Process

- Construct elements in model coordinates
- Assemble scene in world coordinates
- Map to viewing coordinates
  - And to normalized viewing coordinates?
- Map to device coordinates

11

View Coordinate Mapping



12

Viewport (Device) Scaling

$$sx = \frac{xv_{max} - xv_{min}}{xw_{max} - xw_{min}}$$

$$sy = \frac{yv_{max} - yv_{min}}{yw_{max} - yw_{min}}$$

To maintain aspect ratio, force sx=sy

Force to min(sx, sy) to "letterbox"

$$xv = xv_{min} + sx(xw - xw_{min})$$

$$yv = yv_{min} + sy(yw - yw_{min})$$

13

Viewing Support (X, Windows)

- X Window System
  - Little direct support
- MS Windows - DC map mode
  - `MM_TEXT` - pixels, + down
  - Physical metrics, + up
    - `MM_LOENGLISH, MM_HIENGLISH`
    - `MM_LOMETRIC, MM_HIMETRIC`
    - Other specialized modes

14

Viewing Support (Qt)

- QPainter in Qt
  - World Transformation Matrix
  - Window (model) <-> Viewport (device)
- Pages 247-250 in Qt Book
- http://doc.trolltech.com/3.0/coordsys.html

15