To Do

- Submit HW 0 by tomorrow (any issues?)
- Start looking at HW 1 (simple, but need to think)
  - Axis-angle rotation and gluLookAt most useful
  - Probably only need final results, but try understanding derivations.
- Usually, we have review sessions per unit, but this one before midterm. (If you got Shirley-Marschner text, look at problems)

Course Outline

- 3D Graphics Pipeline
  - Modeling ➔ Animation ➔ Rendering
  - Motivation
    - Many different coordinate systems in graphics
      - World, model, body, arms, ...
    - To relate them, we must transform between them
    - Also, for modeling objects, I have a teapot, but
      - Want to place it at correct location in the world
      - Want to view it from different angles (HW 1)
      - Want to scale it to make it bigger or smaller
    - Demo of HW 1

Goals

- This unit is about the math for these transformations
  - Represent transformations using matrices and matrix-vector multiplications.
- Demos throughout lecture: HW 1 and Applet
- Transforms Game Applet
  - Brown University Exploratories of Software
  - http://graphics.cs.brown.edu/research/exploratory/freeSoftware/catalogs/repository/Applets.html
  - Credit: Andries Van Dam and Jean Laleuf
**General Idea**

- Object in model coordinates
- Transform into world coordinates
- Represent points on object as vectors
- Multiply by matrices
- Demos with applet

**Outline**

- 2D transformations: rotation, scale, shear
- Composing transforms
- 3D rotations
- Translation: Homogeneous Coordinates (next time)
- Transforming Normals (next time)

---

**(Nonuniform) Scale**

\[
\text{Scale}(s_x, s_y) = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix}, \quad S^{-1} = \begin{pmatrix} s_x^{-1} & 0 & 0 \\ 0 & s_y^{-1} & 0 \\ 0 & 0 & s_z^{-1} \end{pmatrix}
\]

\[
\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}
\]

**Shear**

\[
\text{Shear} = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix}, \quad S^{-1} = \begin{pmatrix} 1 & -a \\ 0 & 1 \end{pmatrix}
\]

---

**Rotations**

2D simple, 3D complicated. [Derivation? Examples?]

\[
\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}
\]

- Linear \( R(X+Y)=R(X)+R(Y) \)
- Commutative

**Outline**

- 2D transformations: rotation, scale, shear
- Composing transforms
- 3D rotations
- Translation: Homogeneous Coordinates
- Transforming Normals

---

*transformation_game.jar*
### Composing Transforms

- Often want to combine transforms
- E.g. first scale by 2, then rotate by 45 degrees
- Advantage of matrix formulation: All still a matrix
- Not commutative!! Order matters

### E.g. Composing rotations, scales

\[
x_3 = Rx_2 \quad x_2 = Sx_1
\]

\[
x_3 = R(Sx_1) = (RS)x_1
\]

\[
x_3 \neq SRx_1
\]

---

### Inverting Composite Transforms

- Say I want to invert a combination of 3 transforms
- Option 1: Find composite matrix, invert
- Option 2: Invert each transform and swap order
- Obvious from properties of matrices

\[
M = M_1 M_2 M_3
\]

\[
M^{-1} = M_3^{-1} M_2^{-1} M_1^{-1}
\]

\[
M^{-1} M = M_3^{-1} (M_2^{-1} (M_1^{-1} M_1) M_2) M_3
\]

---

### Outline

- 2D transformations: rotation, scale, shear
- Composing transforms
- 3D rotations
- Translation: Homogeneous Coordinates
- Transforming Normals

---

### Rotations

**Review of 2D case**

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix}
= \begin{bmatrix}
  \cos \theta & -\sin \theta \\
  \sin \theta & \cos \theta
\end{bmatrix}
\begin{bmatrix}
  x \\
  y
\end{bmatrix}
\]

- Orthogonal?, \( R^T R = I \)

---

### Rotations in 3D

**Rotations about coordinate axes simple**

\[
R_x = \begin{bmatrix}
  \cos \theta & -\sin \theta & 0 \\
  \sin \theta & \cos \theta & 0 \\
  0 & 0 & 1
\end{bmatrix}
\]

\[
R_y = \begin{bmatrix}
  \cos \theta & 0 & \sin \theta \\
  0 & 1 & 0 \\
-\sin \theta & 0 & \cos \theta
\end{bmatrix}
\]

- Always linear, orthogonal
- Rows/cols orthonormal

\[
R(X+Y) = R(X) + R(Y)
\]
Geometric Interpretation 3D Rotations

- Rows of matrix are 3 unit vectors of new coord frame
- Can construct rotation matrix from 3 orthonormal vectors

\[
R_{uvw} = \begin{bmatrix}
  x_u & y_u & z_u \\
  x_v & y_v & z_v \\
  x_w & y_w & z_w
\end{bmatrix}
\]

\[
R_p = \begin{bmatrix}
  x_u & y_u & z_u \\
  x_v & y_v & z_v \\
  x_w & y_w & z_w
\end{bmatrix}
\begin{bmatrix}
  x_p \\
  y_p \\
  z_p
\end{bmatrix} = \begin{bmatrix}
  u \cdot p \\
  v \cdot p \\
  w \cdot p
\end{bmatrix}
\]

Non-Commutativity

- Not Commutative (unlike in 2D)!!
- Rotate by x, then y is not same as y then x
- Order of applying rotations does matter
- Follows from matrix multiplication not commutative
  - \(R1 \cdot R2\) is not the same as \(R2 \cdot R1\)

- Demo: HW1, order of right or up will matter

Arbitrary rotation formula

- Rotate by an angle \(\theta\) about arbitrary axis \(a\)
  - Homework 1: must rotate eye, up direction
  - Somewhat mathematical derivation but useful formula

- Problem setup: Rotate vector \(b\) by \(\theta\) about \(a\)
  - Helpful to relate \(b\) to \(X\), \(a\) to \(Z\), verify does right thing

- For HW1, you probably just need final formula

Axis-Angle formula

- Step 1: \(b\) has components parallel to \(a\), perpendicular
  - Parallel component unchanged (rotating about an axis leaves that axis unchanged after rotation, e.g. rot about z)

- Step 2: Define \(c\) orthogonal to both \(a\) and \(b\)
  - Analogous to defining \(Y\) axis
  - Use cross products and matrix formula for that

- Step 3: With respect to the perpendicular comp of \(b\)
  - \(\cos \theta\) of it remains unchanged
  - \(\sin \theta\) of it projects onto vector \(c\)
  - Verify this is correct for rotating \(X\) about \(Z\)
  - Verify this is correct for \(\theta\) as 0, 90 degrees

Axis-Angle: Putting it together

\[
(b \cdot a)_{ROT} = (l_{3,3} \cos \theta - aa^T \cos \theta)b + (A' \sin \theta)b
\]

\[
(b \rightarrow a)_{ROT} = (aa^T)b
\]

\[
R(a, \theta) = l_{3,3} \cos \theta + aa^T(1 - \cos \theta) + A' \sin \theta
\]

- Unchanged (cosine)
- Component along \(a\) (hence unchanged)
- Perpendicular (rotated comp)
(b \ a)_{\text{ROT}} = (I_{3 \times 3} \cos \theta - a a^\top \cos \theta)b + (A^\top \sin \theta)b

(b \rightarrow a)_{\text{ROT}} = (aa^\top)b

\begin{align*}
R(a, \theta) &= I_{3 \times 3} \cos \theta + aa^\top (1 - \cos \theta) + A^\top \sin \theta \\
R(a, \theta) &= \cos \theta \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + (1 - \cos \theta) \begin{pmatrix} x^2 & x y & x z \\ x y & y^2 & y z \\ x z & y z & z^2 \end{pmatrix} + \sin \theta \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}
\end{align*}

(x, y, z) are cartesian components of a