
CS770/870 Fall 2008

OpenGL Matrices

Related material in Hill and Kelley: 5

12:12

CS770/870 Fall 2008 Bergeron/Plumlee

1

Preview

- Problem: How to specify transformations efficiently
- Solution: OpenGL Matrix Stack

12:12

CS770/870 Fall 2008 Bergeron/Plumlee

2

The Problem

- How should one specify transformations?
 - Lots of parts in a matrix
 - May want to re-use some combinations of matrices, change others
 - Graphics cards have accelerated matrix capabilities—How do we leverage those?

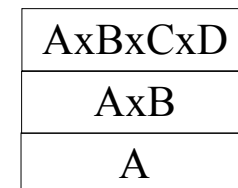
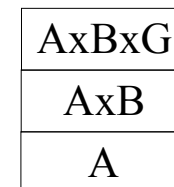
12:12

CS770/870 Fall 2008 Bergeron/Plumlee

3

OpenGL's Solution

- Matrix Stack
 - $A \times B \times C \times D \times v$
 - $A \times B \times G \times v$



12:12

CS770/870 Fall 2008 Bergeron/Plumlee

4

OpenGL Commands

- `glMatrixMode(mode)`
 - `GL_MODELVIEW`—for drawing objects
 - `GL_PROJECTION`—for controlling camera
 - `GL_TEXTURE`—for texture mapping
- `glLoadIdentity()`
 - Replace top matrix on stack with I
 - Does not alter rest of stack

OpenGL Commands

- `glPushMatrix()`
 - Push a copy of the top matrix on the stack
 - Allows you to transform top matrix further, and later...
- `glPopMatrix()`
 - Remove top matrix from the stack
 - Current matrix now the one on top prior to last `glPushMatrix()` call

OpenGL Commands

- `glTranslated(tx, ty, tz)` (or `glTranslatef(...)`)
 - Post-multiply top matrix by translation matrix
- `glRotated(angle, vx, vy, vz)` (or `glRotatef(...)`)
 - Post-multiply top matrix by rotation matrix, rotated *angle* degrees around the vector $[vx, vy, vx]$
- `glScaled(sx, sy, sz)` (or `glScalef(...)`)
 - Post-multiply top matrix by scale matrix

OpenGL Commands

- What is the effect of post-multiplication?
 - Transformations closest to the vertex calls are applied first
 - Effect is bottom-up order of operations
 - Multiplication still actually happens top-down

OpenGL Matrix Example

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity(); //Always good to start with I
glTranslated(5.0, 5.0, 0.0); //move to (5,5)
glPushMatrix();
glTranslated(-1.0, -2.0, 0.0); //move "back"
glRotated(45.0, 0.0, 0.0, 1.0); //rotate around z
glTranslated(1.0, 2.0, 0.0); //move to center
glBegin(GL_LINES);
    glVertex2d(0.0, 0.0);
    glVertex2d(2.0, 0.0);
    glVertex2d(2.0, 4.0);
    glVertex2d(0.0, 4.0);
glEnd(GL_LINES);
glPopMatrix();
```

Review

- OpenGL matrix stack as solution to matrix specification problem
- OpenGL functions:
 - glMatrixMode()
 - glLoadIdentity()
 - glPushMatrix(), glPopMatrix()
 - glTranslated(), glRotated(), glScaled()