The Modelview Matrix

Lecture 7
Section 4.10

Robb T. Koether

Hampden-Sydney College

Fri, Sep 11, 2009
Outline

1. Transformations and Matrices
2. The Modelview Matrix
3. Combining Transformations
4. Assignment
Each geometric transformation can be represented by a matrix.

If a point $P$ is represented by a column vector $p$ and a transformation is represented by a matrix $M$, then the product $Mp$ represents the transformed point.

That is, the transformed point $P'$ is represented by the column vector $p'$, where

$$p' = Mp.$$
Sequences of Transformations

Suppose that we perform two transformations in sequence.
If $M_1$ represents the first transformation and $M_2$ represents the second transformation, then the product $M_2M_1$ represents the sequence of transformations.
That is because the image $P''$ is represented by

$$p'' = M_2p' = M_2M_1p.$$ 

The same principle applies to any number of transformations in sequence.
The Modelview Matrix

- OpenGL maintains a matrix called the **modelview matrix**.
- It represents the **current transformation**, as determined by the various calls to `glTranslatef()`, `glRotatef()`, `glScalef()`, etc...
- It should be initialized to the identity matrix, using `glLoadIdentity()`.
The Modelview Matrix

Initializing the Modelview Matrix

```c
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
```

- OpenGL code to initialize the modelview matrix.
The Modelview Matrix

- If $M$ represents the current transformation and we wish to apply transformation $M_1$, then OpenGL computes $MM_1$ and makes that the new current matrix.
- The old $M$ is gone. It was replaced by $MM_1$.
- Note that $M$ was multiplied on the right by $M_1$. 
Suppose the current transformation is $M$ and we wish to apply a sequence of transformations $M_1$, $M_2$, and $M_3$ to an object.

We need to create $MM_3M_2M_1$ on the stack.

In other words, we call the various transformation functions in reverse order.
Example

- Suppose we begin with the unit sphere centered at the origin and we wish to
  - Scale it by a factor of 2 in the $y$-direction, and then
  - Rotate it $45^\circ$ about the positive $x$-axis, and then
  - Translate it $+2$ units in the $z$-direction.
A Sequence of Transformations

```c
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslatef(0.0, 0.0, 2.0);
glRotatef(45.0, 1.0, 0.0, 0.0);
glScalef(1.0, 2.0, 1.0);
glutSolidSphere(1.0, 40, 40);
```

- OpenGL code for this sequence of transformations.
Example

Transforming a Sphere

- The code.
- The executable.
Suppose we want to rotate an object about the point \((x, y, z)\) (not the origin).

We perform:
- A translation by \((-x, -y, -z)\).
- The rotation (about the origin).
- The inverse translation by \((x, y, z)\).

The first translation may not be necessary if we draw the object in model coordinates.
Homework

- Read Section 4.10 – transformations in OpenGL.