Viewer Movement in OpenGL

Tom Kelliher, CS 320

Mar. 31, 2003

1 Administrivia

Announcements

Pong due now!

Assignment

Read Chapter 5.

From Last Time

Linear algebra basis for computer graphics transformations.

Outline

1. Understanding clipping volumes and their specifications.
2. Projections.
3. Movements in 3-D.
4. Toward a better movement model.

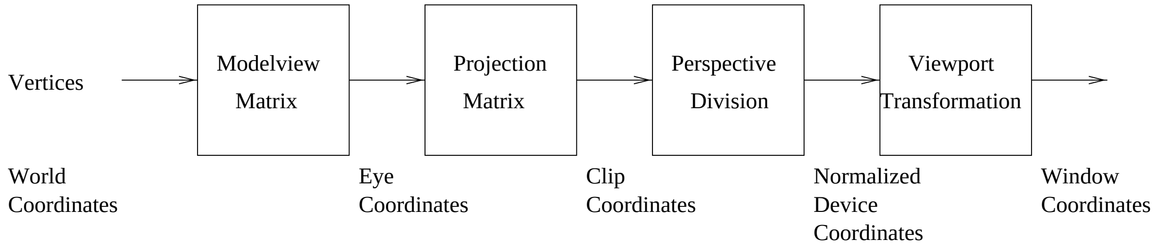
Coming Up

Movement through a room.

2 Preliminary: Viewing Volumes

Are our viewing volume coordinates relative or absolute?

Consider:



1. By default, the eye is at $(0, 0, 0)$ looking down the $-z$ axis.

2. What does

   ```
   glOrtho(-10.0, 10.0, -5.0, 5.0, -2.0, 2.0);
   ```

   mean?

3. Other viewing modes:

   (a) `glFrustum`: same parameters as `glOrtho`. What's a frustum? Truncated pyramid.

   (b) `gluPerspective`: fovy, aspect ratio, zNear, and zFar.

   znear and zfar need to be **positive**.

3 Moving and Positioning the Eye

View specification:

1. One way of specifying eye position and viewing angle:
   (a) Specify position of eye.
   (b) Specify center of field of view.
   (c) Specify "up."
2. Use of `gluLookAt()` in `cubeview.c`:

   ```
   void display(void)
   {

      glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

   /* Update viewer position in modelview matrix */

      glLoadIdentity();
      gluLookAt(viewer[0],viewer[1],viewer[2], 0.0, 0.0, 0.0,
                0.0, 1.0, 0.0);

   /* rotate cube */

      glRotatef(theta[0], 1.0, 0.0, 0.0);
      glRotatef(theta[1], 0.0, 1.0, 0.0);
      glRotatef(theta[2], 0.0, 0.0, 1.0);

       colorcube();

      glutSwapBuffers();
   }
   ```

   Note order of matrix multiplications: view, then model transformations.
3. Is it really necessary to have view and model transformations?

3.1 Example Runs

1. P1: Stock viewcube using frustum. Demonstrate clipping, invisibility when up vector is parallel to line of sight, walking through the cube.

2. P2: Perspective view with fovy 45, near 2, and far 20.

3. P3: Perspective view with fovy 135, near 0.1, far 100.

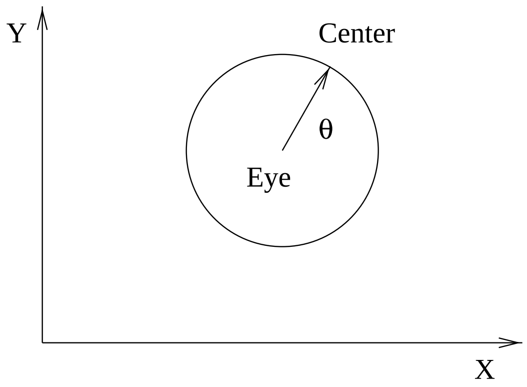
4 A Movement Model

Problems with viewer movement in cubeview:

1. Must specify movement in global coordinate values.

2. Can't speak of left, right, forward, backward, etc.

Consider this model:



Eye: (x, y)
Center: (x + Δx, y + Δ y)

1. What should the radius of the circle be?

2. Given x, y, and θ, what's Δx and Δy?

3. How do we handle left, right, forward and backward?

4. Suppose, to see the "big picture," I wanted to elevate on the Z-axis. What should I do with center? Is that easy to do?