

Menus, Colors

CS116B

Chris Pollett

May 9, 2004.

Outline

- OpenGL for the stuff in Chapter 10
- OpenGL for Menus
- Start of Color

OpenGL Textures

- OpenGL supports 1D, 2D, 3D textures.
- The basic commands to do 1D texturing are:

```
// say what texture to use
glTexImage1D(GL_TEXTURE_1D, 0, GL_RGBA, nTexColors, 0,
             dataFormat, dataType, lineTexArray);
/*the first 0 is an offset into our array, nTexColors - is the size of our
  array (must be power of 2). The second 0 says no border.
  dataformat is used to say the color format for the texture. For
  example, GL_RGBA. datatype could be GL_UNSIGNED_BYTE,
  GL_INT, GL_FLOAT, finally lineTexArray is our array*/
//say how texture should be reduced/enlarged to fill an area
glTexParameteri(GL_TEXTURE_1D, GL_TEXTURE_MAG_FILTER,
                GL_NEAREST);
glTexParameteri(GL_TEXTURE_1D, GL_TEXTURE_MIN_FILTER,
                GL_NEAREST);
glEnable(GL_TEXTURE_1D); //enable
glBegin(GL_LINES);
    glTexCoord1f(0.0); glVertex3fv(pt1);
    glTexCoord1f(1.0); glVertex3fv(pt1);
glEnd();
```

2D and 3D Textures in OpenGL

- The idea is similar to the 1D case except now to set up use:

```
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, nWidth, nHeight, 0,  
            dataFormat, dataType, surfTexArray);
```

```
glEnable(GL_TEXTURE_2D);
```

//Width and height must both be a power of 2.

- Filter step now take 2D INSTEAD OF 1D:

```
glTexParameter(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,  
              GL_NEAREST);
```

```
glTexParameter(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,  
              GL_NEAREST);
```

- We now use `glTexCoord2f(x,y);` // where x and y between 0 and 1.0
- 3D is similar now have `glTexImage3D`, etc ..
- More texture stuff in book.... Also, see HW3 solutions.

Classification of Input Devices

- We are now going to talk a little about input devices. Mainly, filling in blanks that we have missed so far.
- A **logical input device** is an input device for a particular data type.
- The standard classification of logical input-data consists in:
 - LOCATORS - used to specify coordinates
 - STROKE - used to specify a set of coordinate positions
 - STRING - used for text input
 - VALUATOR - used for specifying a scalar value
 - CHOICE - a device for selecting a menu option
 - PICK -- a device for selecting a component of a picture.

OpenGL Menus

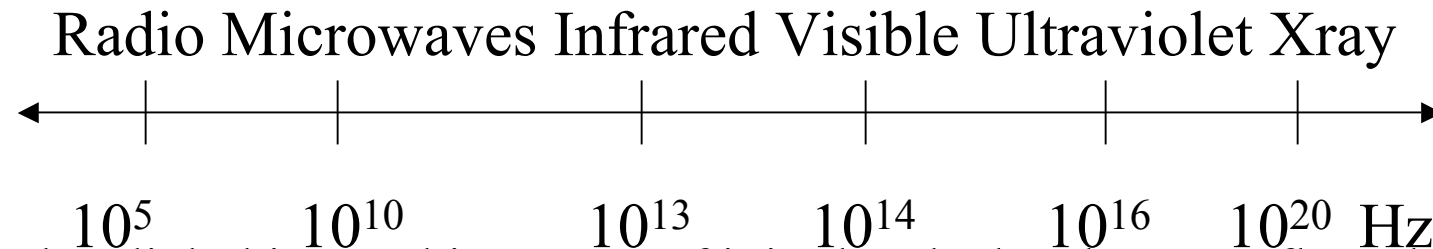
- Many of these inputs can be handled via a mouse, mouse wheel, or other pointing device or a keyboard and we've already seen how to do these in OpenGL.
- So how do we handle menu input?
- A pop-up menu can be created with the command:
`glutCreateMenu(menuFcn);`
*/*menuFcn is a callback with one argument GLint for what selected*/*
- Menu items can be added using lines like:
`glutAddMenuEntry("entry 1", 1);`

More Menu's

- Finally, we need to attach the menu to a mouse button click:
`glutAttachMenu(GLUT_RIGHT_BUTTON);`
- Only one menu is active at a given time.
- To set this we use:
`glutSetMenu(menuID);`
`//id = glutGetMenu();// get the active menu`
- Besides entries we can add submenu's to a menu with:
`glutAddSubMenu("Submenu Option",submenuID);`
- Finally, to get rid of a menu, get it off a button or remove an item we can do respectively:
`glutDestroyMenu();`
`glutDetachMenu(button);`
`glutRemoveMenuItem(itemNum);`

Color

- As we said before light is made of electromagnetic radiation. The frequency of this radiation determines the color we see.



- When light hits an object some of it is absorbed and some reflected. The color of a material is determined by the dominant frequency reflected (**hue**).
- There are a number of psychological factors that also determine how we see light: **Brightness** - related to the total energy of the light (recall luminance). **Purity/Saturation** - how close a color is to a pure spectral color such as red. For example, pastels have low purity. **Chromaticity** -- is a collective term for hue and saturation.