

Comp 360 Tutorial

Lab 0 Framework

August 24, 2011

Outline

- Go over the given framework
- Learn some basic OpenGL
- Answer questions

Environment

- C/C++ and OpenGL
- Code will be given with VS 2010 files (or CMake)
 - Uses glut, should be cross-platform
- Submissions should be compilable in VS 2010 (or have a CMakeLists.txt)
- Using the framework is not mandatory.
 - But it makes grading easier
 - Probably helps you as well

Prerequisites

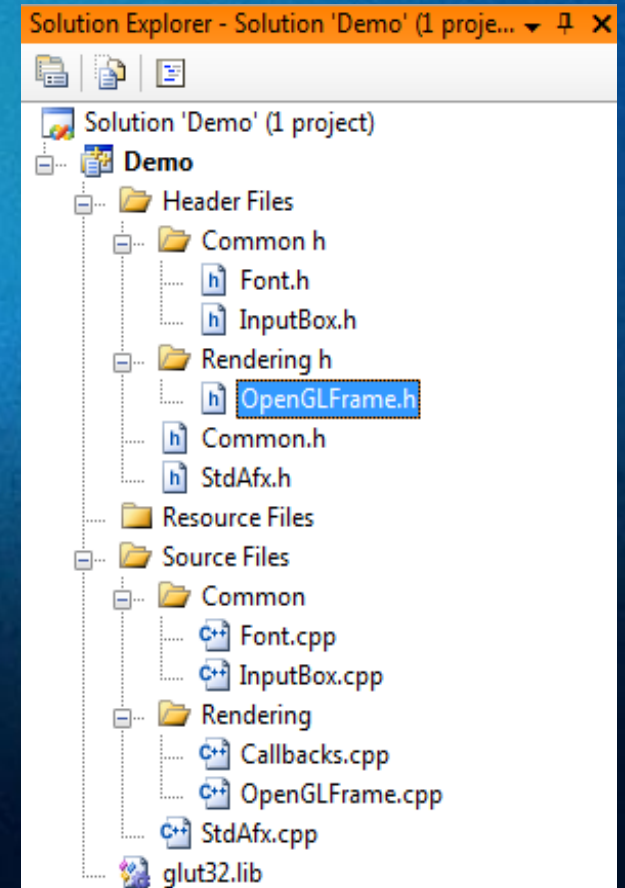
- At least you need to
 - Have programmed before
 - Know how to compile/run/debug/test code
- Better if you
 - Know C/C++
 - Have extensive programming experiences.
- Awesome if you
 - Already know OpenGL

Framework

- Go to <http://www.clear.rice.edu/comp360/>
- Find the Labs/Homeworks/Exams heading
 - Download Base Code for Lab 0
- Unzip framework.zip
- Open Framework.sln
- Convert project to VS 2010

Files

- OpenGLFrame/.cpp/.h
 - Wrapper around GLUT
 - In most cases, don't need to modify this
- InputBox/.cpp/.h
 - Handles the "InputBox" functionalities
 - Don't need to modify
- Callbacks.cpp
 - Needs to be modified



OpenGLFrame

- `printString()`
 - Print a string on screen at specified coordinates
- `setOrthographicProjection()`
 - Sets the screen to draw 2D (not exactly)
- `resetPerspectiveProjection()`
 - Sets the screen to draw 3D (not exactly)
- `getWidth()` and `getHeight()`
 - Returns the dimension of the window

OpenGLFrame

- Let's take a deeper look
- Open `OpenGLFrame.cpp`

Callback.cpp

- We delegate all the action to this file.
 - Handles input/output
 - Rendering

```
/**
 * function called by the opengl frame to draw all screen contents
 * (called once per frame)
 *
 * @param timeElapsed  time elapsed since last frame in milliseconds
 */
void renderFunc (float timeElapsed)
{
    OpenGLFrame::setOrthographicProjection();
    glDisable(GL_LIGHTING);

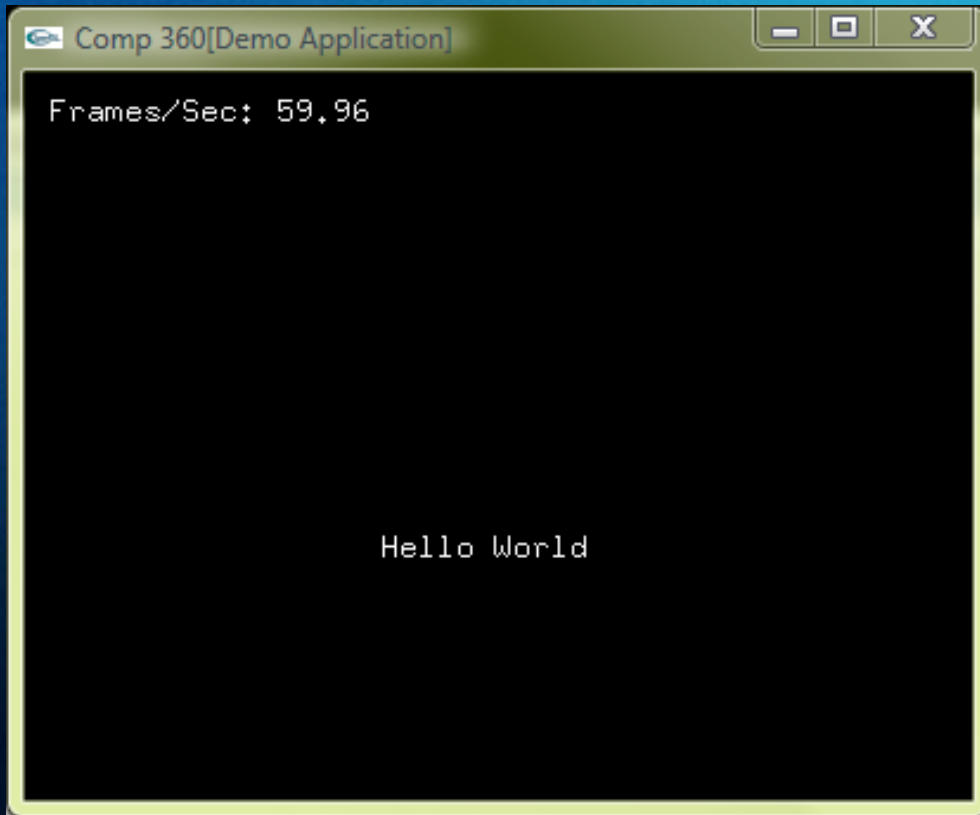
    /***** All drawing code goes here *****/

    OpenGLFrame::resetPerspectiveProjection();
}
```

Easy Exercise

- Quit the program when the key 'q' is pressed.
- Quit the program when 'quit' is entered in the InputBox.
 - C++ hint: you can use the `std::string` class for string equality comparison
 - C hint: `strcmp` returns 0 if two strings are equal

Easy Exercise



- Print "Hello World" on the screen when 'hello' is entered in the InputBox. Remember all rendering should go in the renderFunc, even text.
- Hint: Use function from OpenGLFrame

OpenGL

- A powerful graphics library
 - Widely-used in the industry/academia
 - Capable of rendering 3D geometries with varieties of effects.
- Keeps around a global state.
 - Color, line width, point size, etc.
 - glColor*, glLineWidth, glPointSize
 - Transformation matrices, lighting, etc.

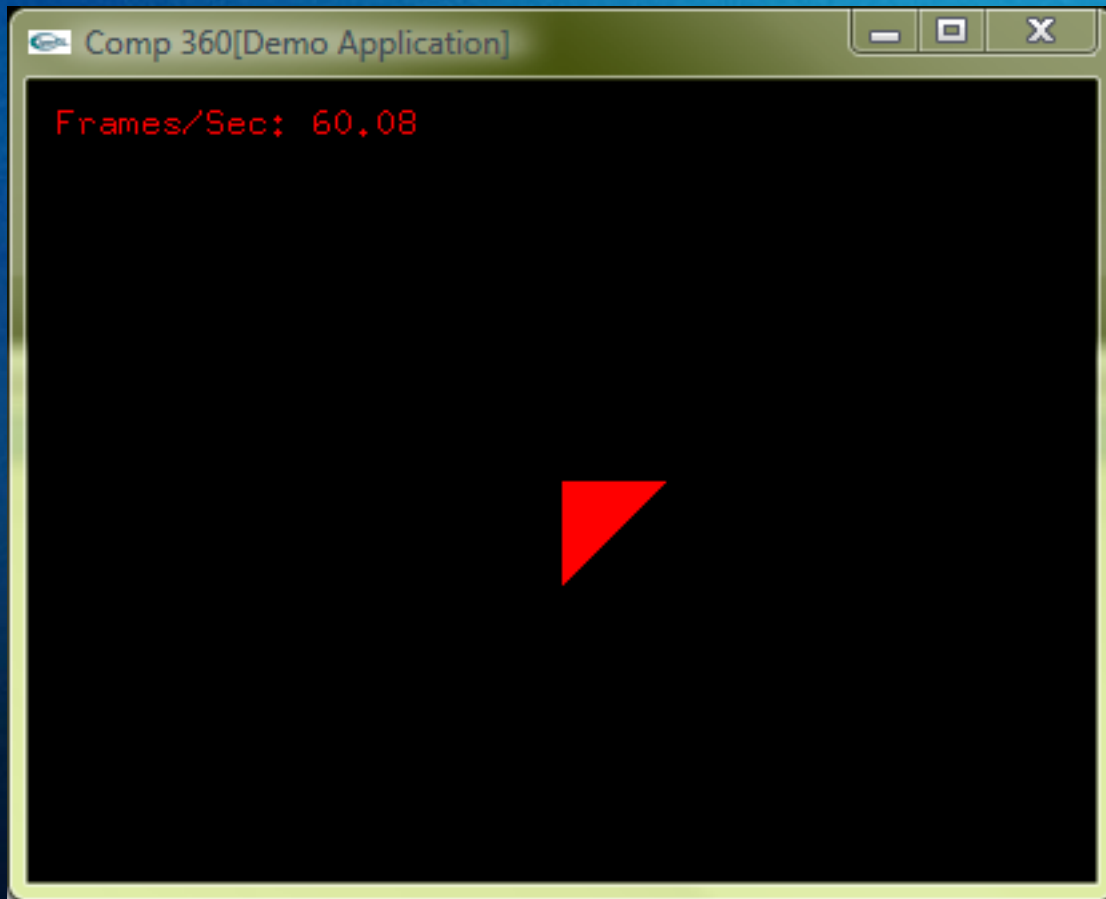
OpenGL

- Drawing code goes between glBegin and glEnd

```
glBegin(GL_POINTS);  
glVertex2f(100.f,100.f);  
glEnd();
```

- Can draw different geometric primitives
 - points, lines, triangles, quads, etc.
 - GL_POINTS, GL_LINES, GL_TRIANGLES, GL_QUADS

Exercise



- Draw a red triangle somewhere in the middle of the screen.

- see <http://www.opengl.org/sdk/docs/man/> for reference

Let's try some 3D stuff

- Comment out the lines
 - `OpenGLFrame::setOrthographicProjection();`
 - `glDisable(GL_LIGHTING);`
 - `OpenGLFrame::resetPerspectiveProjection();`

Let's try some 3D stuff

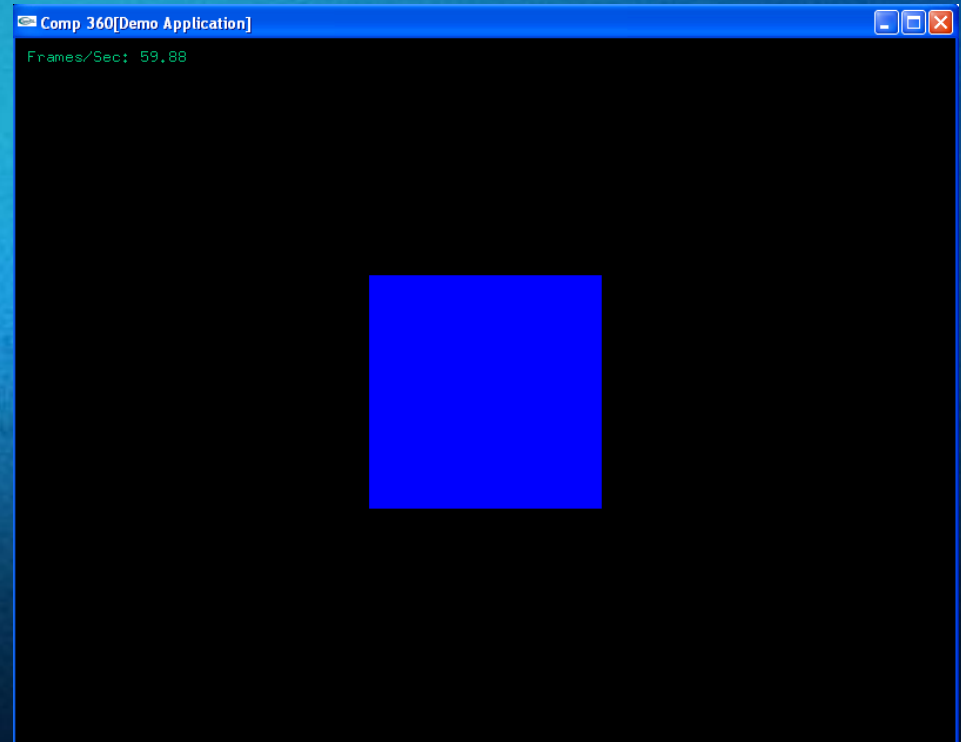
- Add these lines
 - `float ratio =`
`OpenGLFrame::getWidth()/(float)OpenGLFrame::getHeight();`
 - `glMatrixMode(GL_PROJECTION);`
 - `glPushMatrix();`
 - `glLoadIdentity();`
 - `glFrustum(-.8*ratio,.8*ratio,-.8,.8,1,1000);`
 - `glMatrixMode(GL_MODELVIEW);`
 - `glPushMatrix();`
 - `glLoadIdentity();`
 - `glTranslatef(-.5f,-.5f,-.5f);`
 - `glTranslatef(.0f,.0f,-2.5f);`
 - `// draw here`
 - `glMatrixMode(GL_PROJECTION);`
 - `glPopMatrix();`
 - `glMatrixMode(GL_MODELVIEW);`
 - `glPopMatrix();`

Let's try some 3D stuff

- We want to draw a 3D cube of size 1. How do we do that?

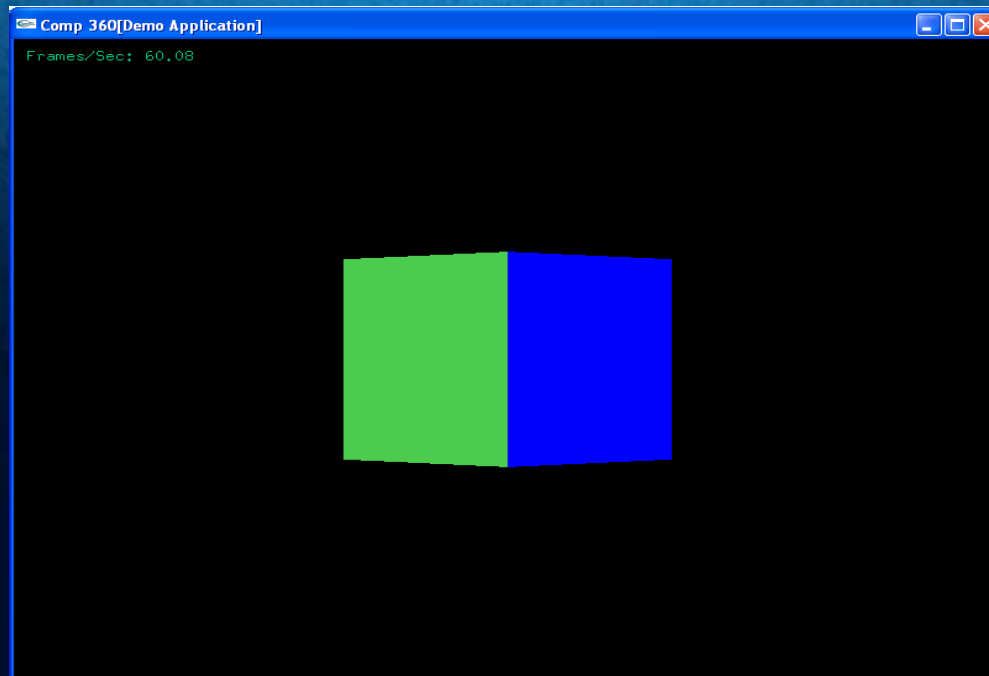
Let's try some 3D stuff

- Let's draw 6 quads. Think about the coordinates and then code them.
- We can use different colors for each quad to distinguish them from each other.



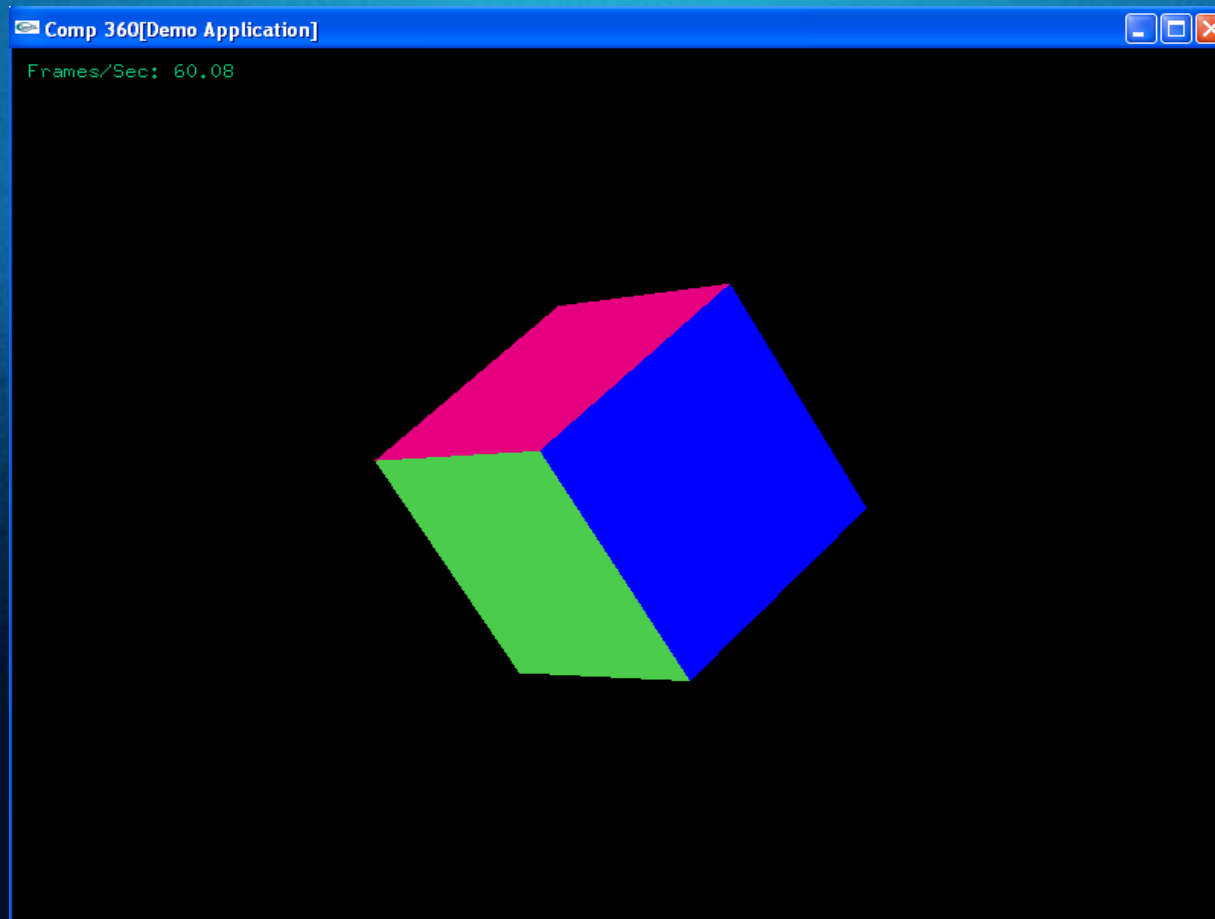
Let's try some 3D stuff

- We can rotate it a little bit.
- Add the line
 - **glRotatef(45.f,0.f,1.f,0.f);**
 - after the 2
 - **glTranslate**
 - calls.



Exercise

- Make it rotate continuously.



The End

We are done! Questions?