CSE328 Fundamentals of Computer Graphics

OpenGL Programming: A Quick Walkthrough (2)

Yicheng Lin

DEPT. OF COMPUTER SCIENCE, STONY BROOK UNIVERSITY

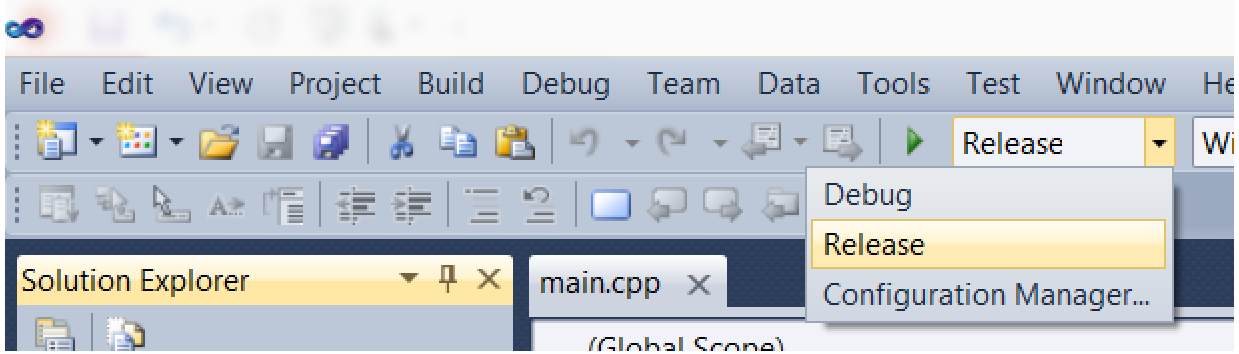
yiclin@cs.stonybrook.edu

Outline

- HW2 Info
- OpenGL 3D: View and Transformations
- Examples & Explanation
- Advanced Topics
- Course Project

HW2 Info

- Please allow some time for Windows config. and preparing your submission
- Remember to generate a RELEASE build because the debug version may lag



OpenGL 3D: View and transformation

- OpenGL 3D drawing is similar with 2D
  - glBegin(GL_…)
  - glNormal3f(…) (You need to normalize to unit length)
  - glVertex3f(…)
  - glEnd()
- The key is how to describe where you look
- OpenGL maintains 2 **matrix stacks** to describe the **ModelView/Projection**

OpenGL 3D: View and transformation

- **Model**: Local coordinate (object) -> World
- **View**: World -> Camera
- **Projection**: Camera -> Screen
- OpenGL combines Model and view into one
- In OpenGL the camera actually never moves; the world moves accordingly

OpenGL 3D: View and transformation

Useful commands:

- glLoadIdentity()
- glMatrixMode(GL_MODELVIEW/GL_PROJECTION)
- gluLookAt() **(works with ModelView)**
- glFrustum() **(works with Projection)**

OpenGL 3D: View and transformation

To set up the view:

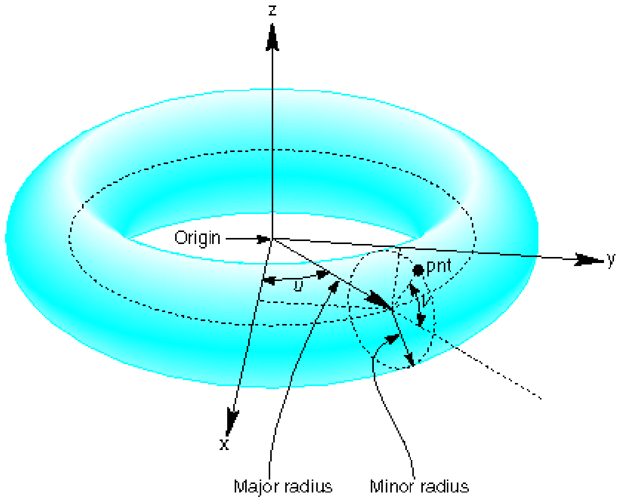
- Clear both stacks with identity matrix upon view change
- Set up the camera position
  gluLookAt() in ModelView mode
- Set up the view range
  glFrustum() in Projection mode

Example 1

- Textbook p. 396
- Don’t forget to load identity
- Change the view range

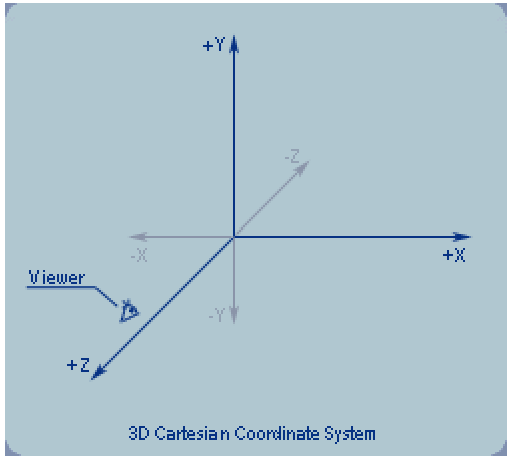
Example 2

- Draw a torus with parametric equation



Example 3

- Rotate the scene



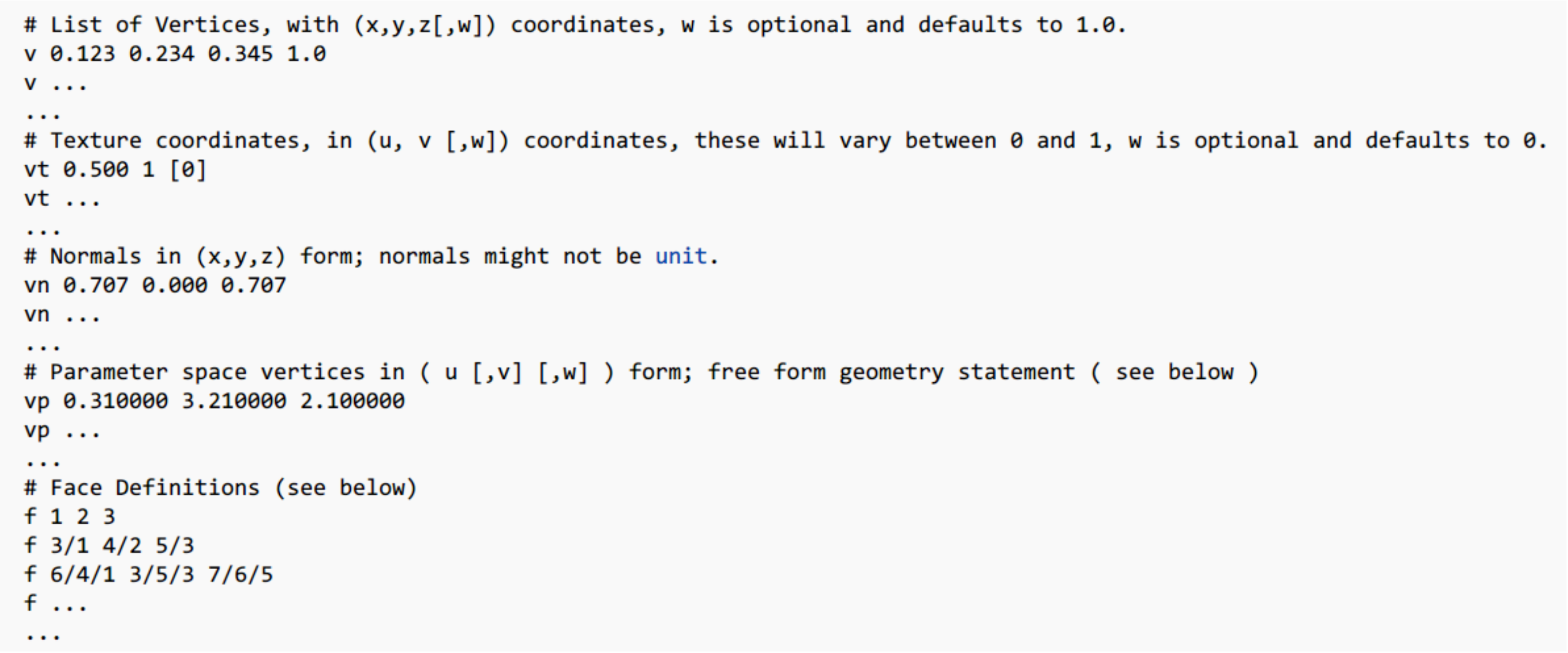
Advanced Topics

We have students working on

- Modeling
- Simulation
- Rendering and Shader

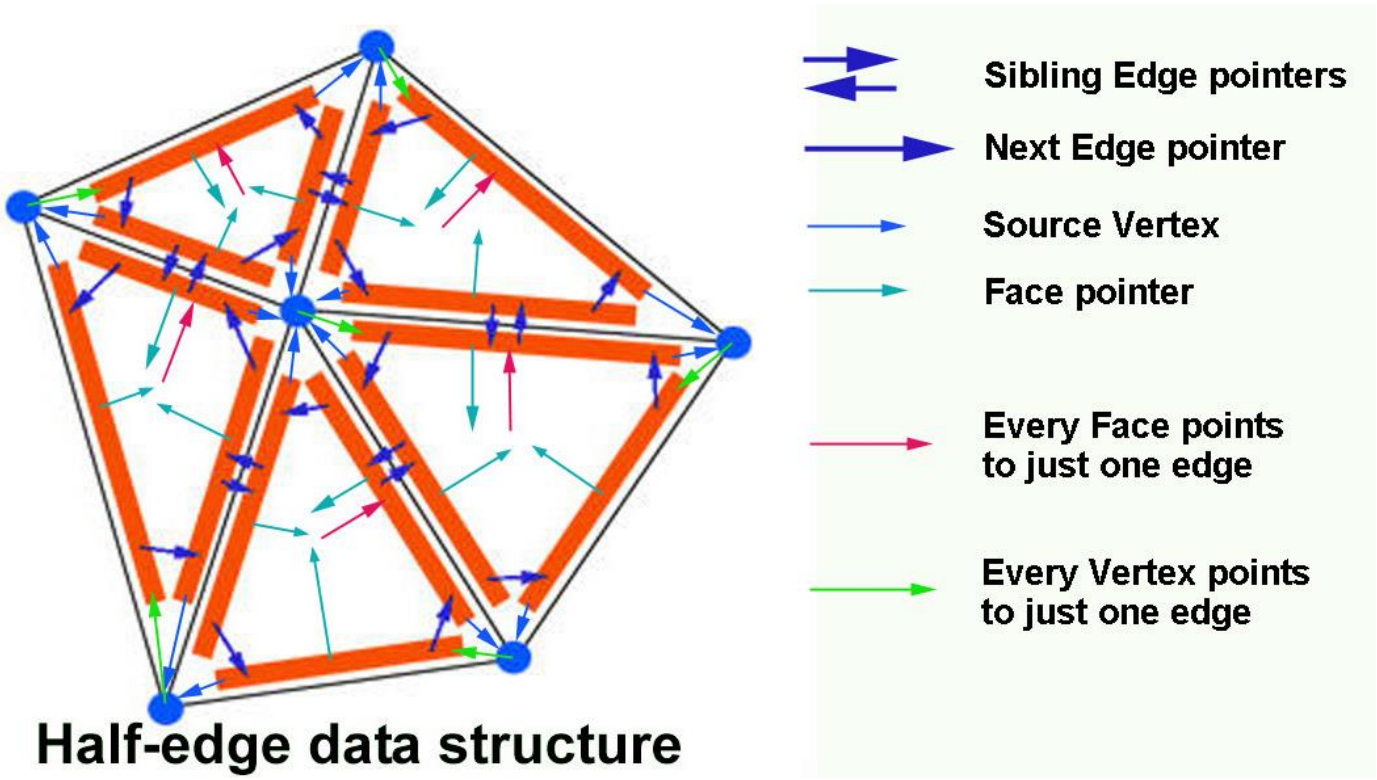
Triangular Mesh

- Mesh is made up of vertices, edges and faces (triangles)
- A good data structure is needed to handle mesh geometry



```
# List of Vertices, with (x,y,z[,w]) coordinates, w is optional and defaults to 1.0.
v 0.123 0.234 0.345 1.0
v ...
...
# Texture coordinates, in (u, v [,w]) coordinates, these will vary between 0 and 1, w is optional and defaults to 0.
vt 0.500 1 [0]
vt ...
...
# Normals in (x,y,z) form; normals might not be unit.
vn 0.707 0.000 0.707
vn ...
...
# Parameter space vertices in ( u [,v] [,w] ) form; free form geometry statement ( see below )
vp 0.310000 3.210000 2.100000
vp ...
...
# Face Definitions (see below)
f 1 2 3
f 3/1 4/2 5/3
f 6/4/1 3/5/3 7/6/5
f ...
...
```

Halfedge Data Structure



Sibling Edge pointers
Next Edge pointer
Source Vertex
Face pointer
Every Face points to just one edge
Every Vertex points to just one edge

Half-edge data structure

Modeling and Mesh Editing

- Find a simple mesh viewer to start with if you can
- Let the program framework deal with view control, save/load file and data structure so that you can focus on algorithm

Simulation

- Includes cloth, fluid, smoke, etc.
- The key is PDE and its numerical solution
- Understand the basic model your paper was based on and what were the improvements
- Discretize the timestep and spatial domain, design a proper approximation scheme
- Use glutIdleFunc/glutTimerFunc to update the animation
- Pay attention to your parameters and the convergence and stability of your approximation scheme
- Find a solver to start with if you can

Course Project

- Midterm demo on Apr 8
- You are expected to finish 1/3+ of your project and **have preliminary results to show**
- Using online resources and open source libraries is encouraged as long as there is adequate amount of your own coding

Q&A

yiclin@cs.stonybrook.edu

http://www3.cs.stonybrook.edu/~yiclin/cse328/index.html