

CT5510: Computer Graphics

# Basic OpenGL

---

BOCHANG MOON



# Graphics Pipeline

---

- A series of computer operations to generate images from 3D objects
- Hardware pipeline
  - Real-time rendering (e.g., games)
  - APIs like OpenGL and DirectX
- Software pipeline
  - High-quality but offline rendering (e.g., animated films)
  - APIs like RenderMan
- Scope of this course?



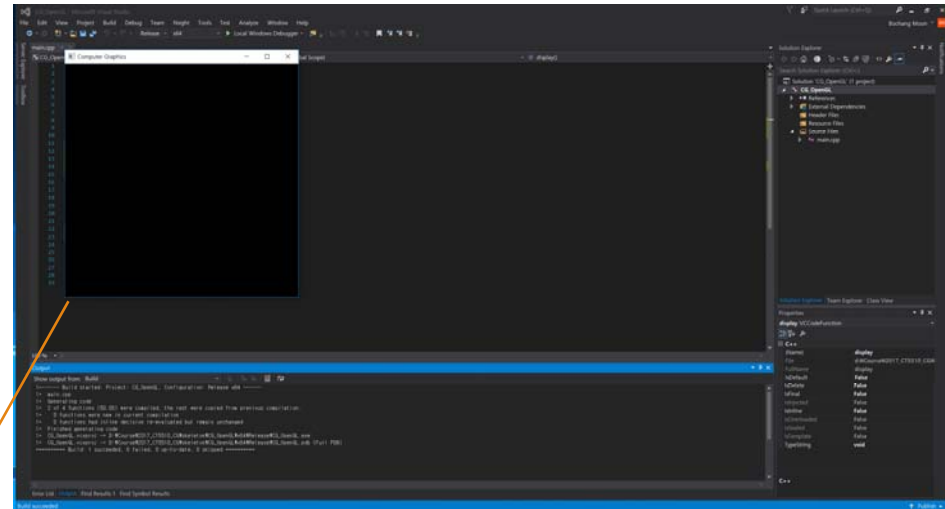
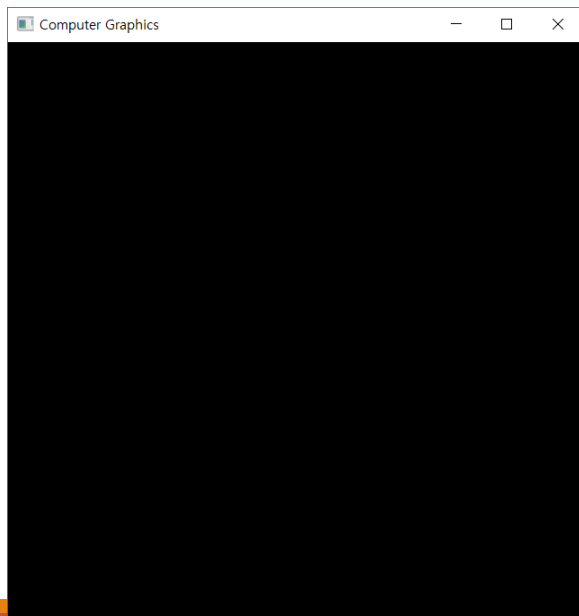
# OpenGL

---

- Open Graphics Library (OpenGL):
  - Cross-platform application programming interface (API)
  - Typically interact with GPUs
  - Widely used API for interactive rendering
  
- Additional libraries
  - GLU
  - FreeGLUT (OpenGL Utility Toolkit)
    - <https://sourceforge.net/projects/freeglut/>

# OpenGL Tools

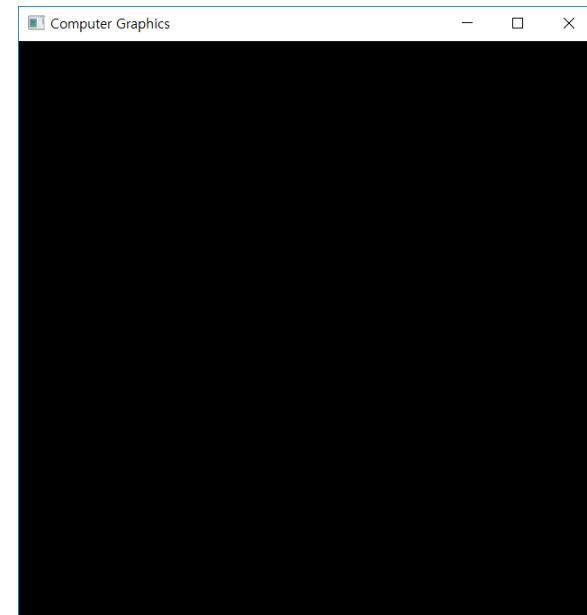
- `void main() {`
  - ...
  - `glutInitWindowSize(512, 512);`
  - `glutInitWindowPosition(100, 100);`
  - `glutCreateWindow("Computer Graphics");``}`



# OpenGL Tools

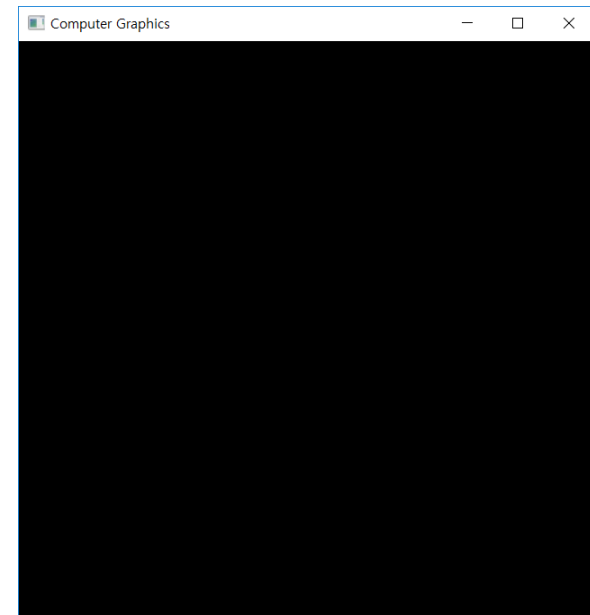
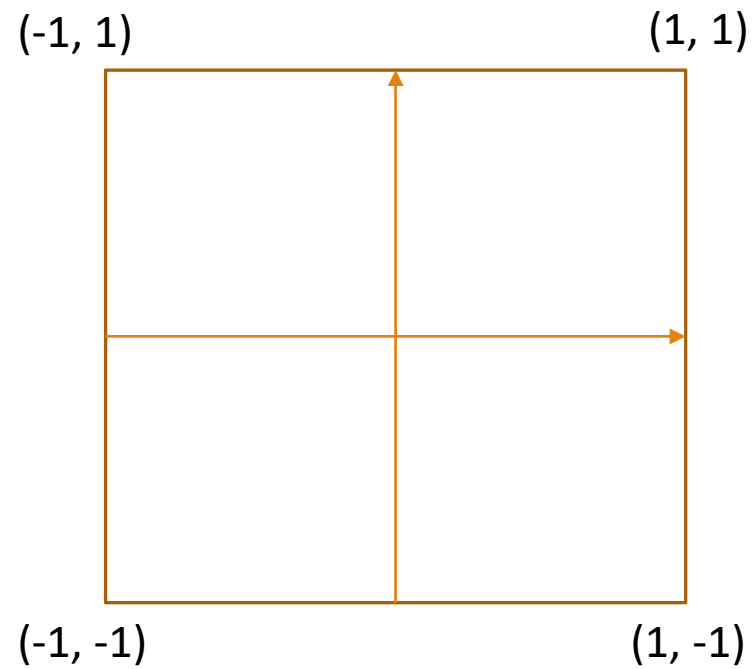
---

- `void main() {`
  - `...`
  - `glutDisplayFunc(display);`
  - `...``}`
  
- `void display() {`
  - `// draw some objects``}`



# OpenGL Coordinates

---



# Draw Triangles

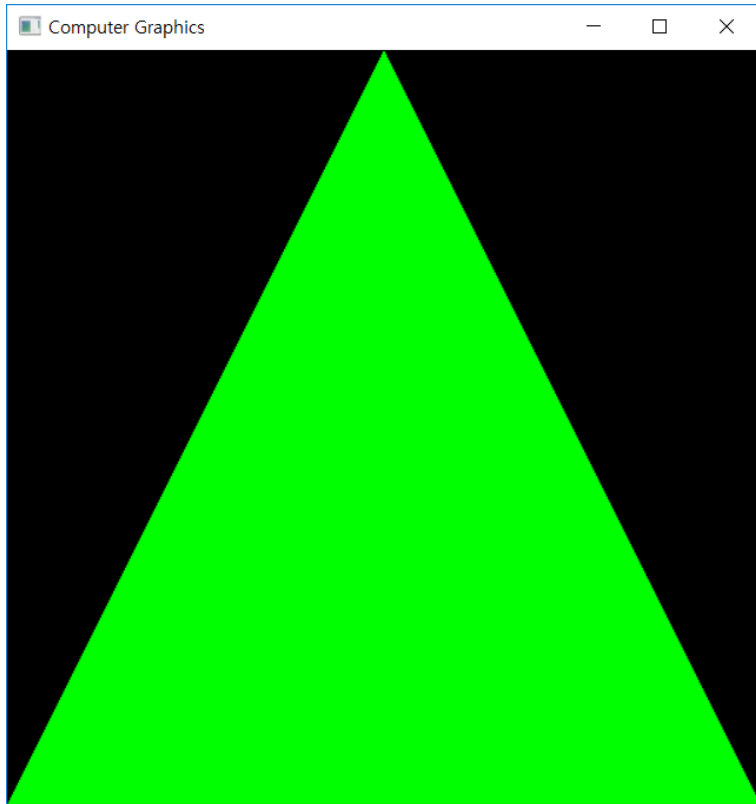
---



- `glColor3d(1.0, 0.0, 0.0);`
- `glBegin(GL_TRIANGLES);`
- `glVertex2d(-1.0, 1.0);`
- `glVertex2d(-1.0, -1.0);`
- `glVertex2d(1.0, -1.0);`
- `glEnd();`

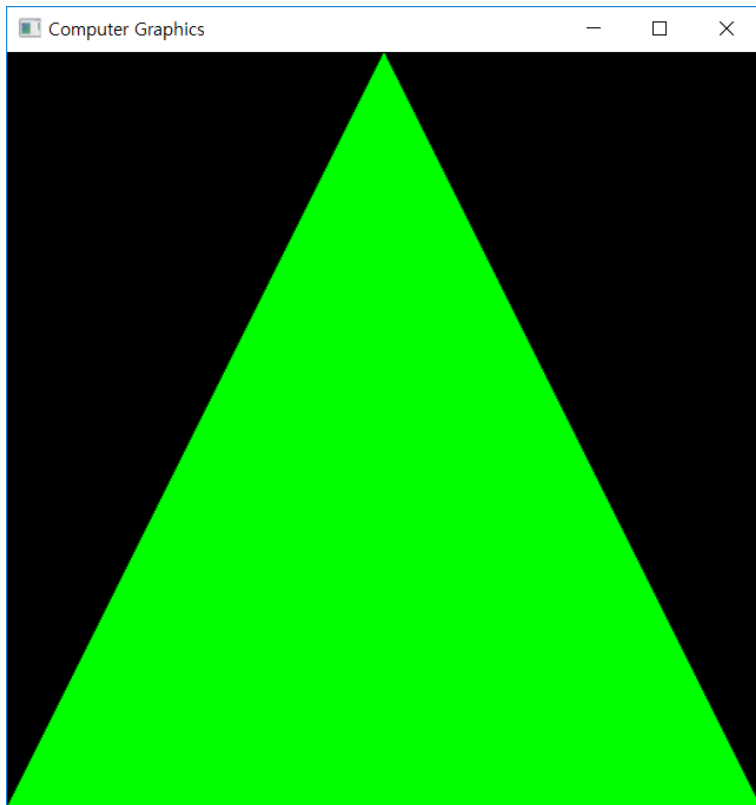
# Draw Triangles

---



# Draw Triangles

---



- `glColor3d(0.0, 1.0, 0.0);`
- `glBegin(GL_TRIANGLES);`
- `glVertex2d(0.0, 1.0);`
- `glVertex2d(-1.0, -1.0);`
- `glVertex2d(1.0, -1.0);`
- `glEnd();`



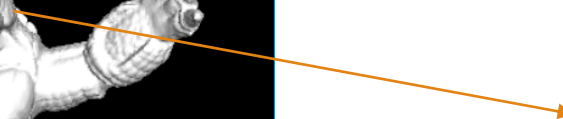
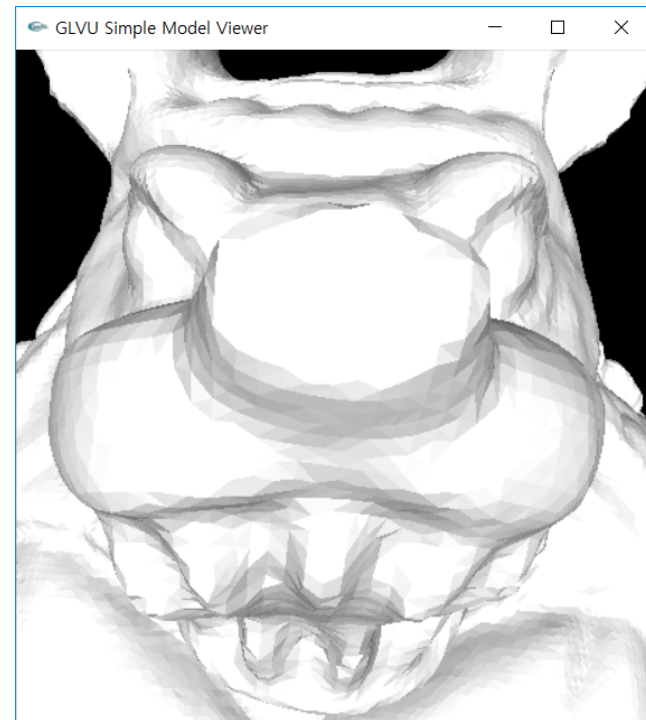
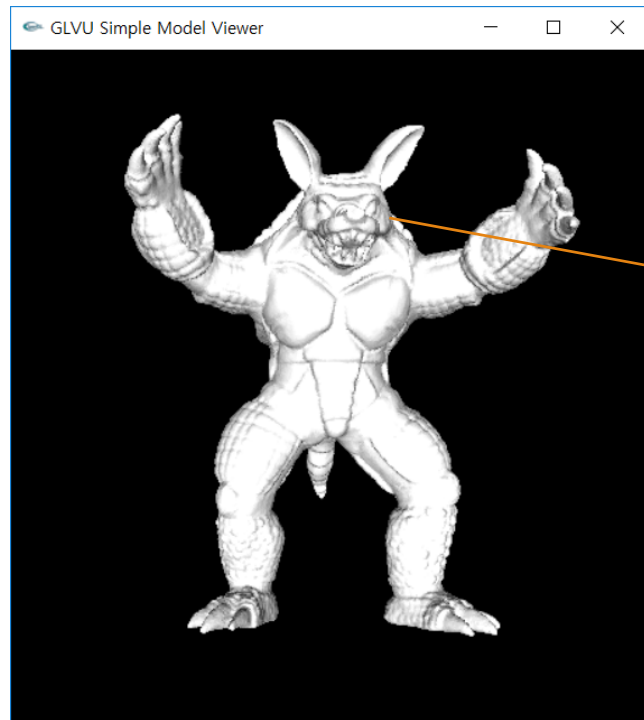
# Full Source Code

- ~27 lines
- Very simple & easy
- You need to install freeglut
  - <https://sourceforge.net/projects/freeglut/>
  - Need to compile the code (with cmake) and generate lib files

```
main.cpp # X
CG_OpenGL (Global Scope)
1 #pragma comment(lib,"opengl32.lib")
2 #pragma comment(lib,"glu32.lib")
3 #pragma comment(lib,"freeglut_static.lib")
4 #include "freeglut.h"
5 #include <GL/glu.h>
6 #include <math.h>
7
8 void display() {
9     glClearColor(0, 0, 0, 0); // Clear the screen
10    glColor3d(0.0, 1.0, 0.0);
11    glBegin(GL_TRIANGLES);
12    glVertex2d(0.0, 1.0);
13    glVertex2d(-1.0, -1.0);
14    glVertex2d(1.0, -1.0);
15    glEnd();
16    glFlush();
17 }
18
19 void main(int argc, char* argv[]) {
20     glutInit(&argc, argv);
21     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
22     glutInitWindowSize(512, 512);
23     glutInitWindowPosition(100, 100);
24     glutCreateWindow("Computer Graphics");
25     glutDisplayFunc(display);
26     glutMainLoop();
27 }
```

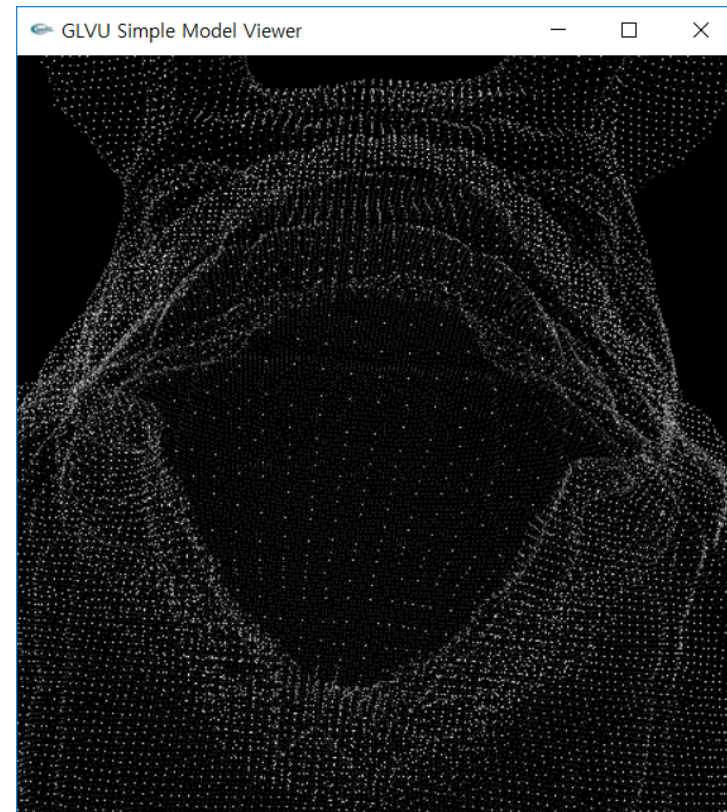
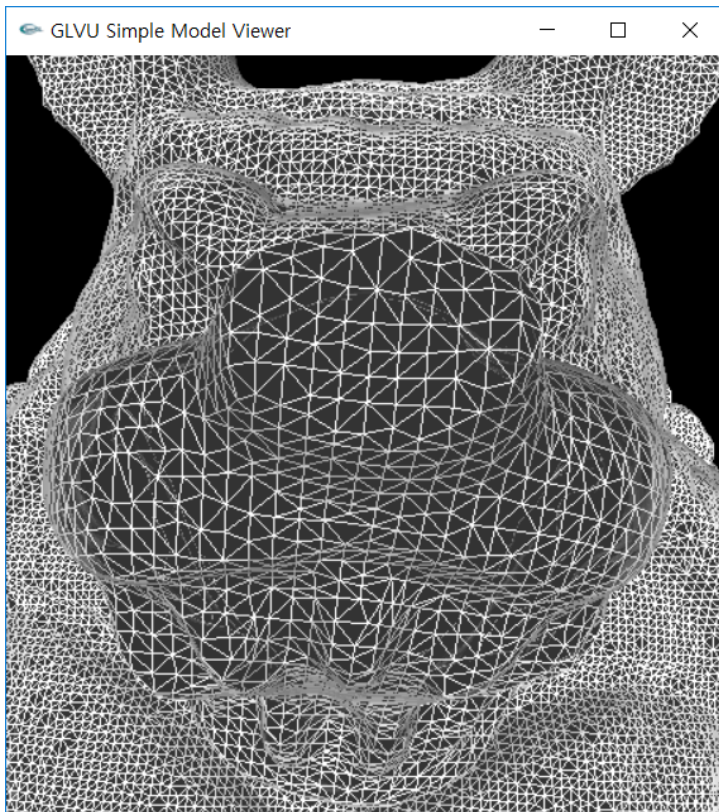
# Triangles

- Fundamental modeling primitives



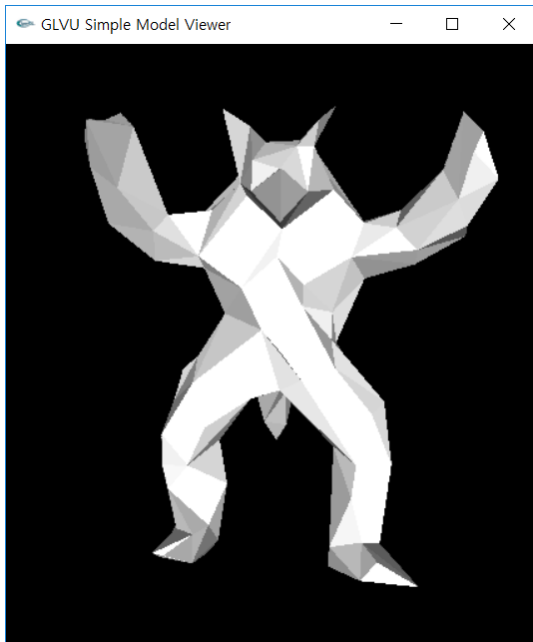
# Triangles

- Fundamental modeling primitives

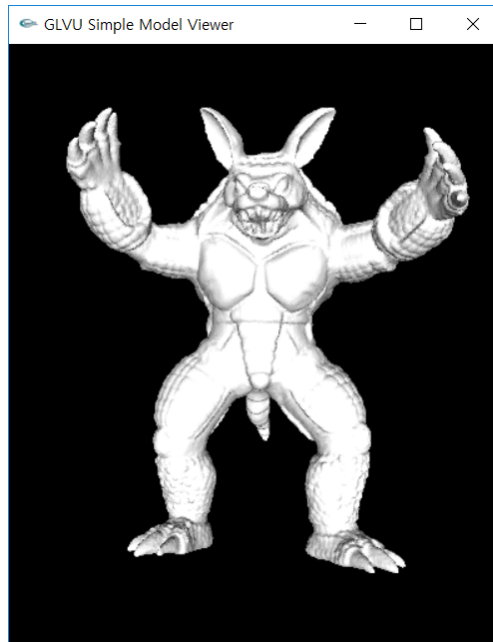


# Triangles

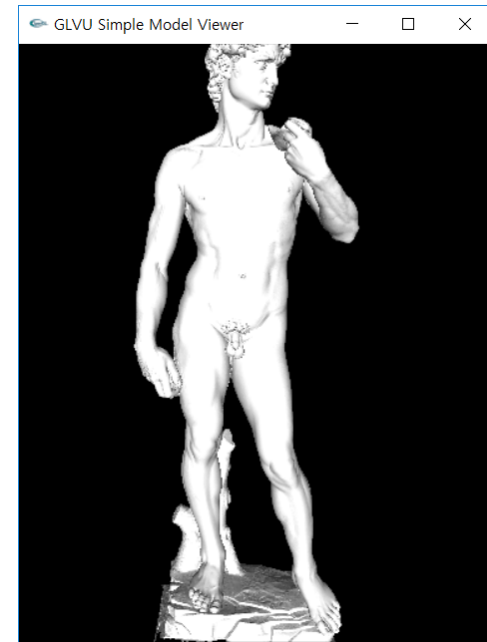
---



300 triangles



345,944 triangles



8,254,150 triangles



# Triangles

---

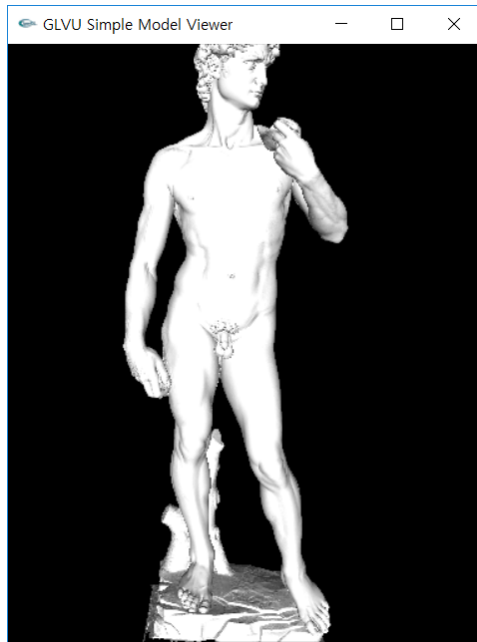
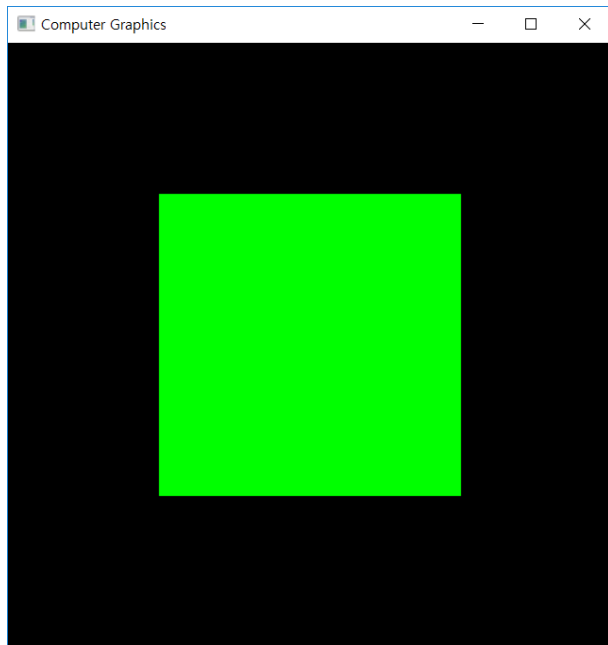


Image from [graphics.stanford.edu](http://graphics.stanford.edu)



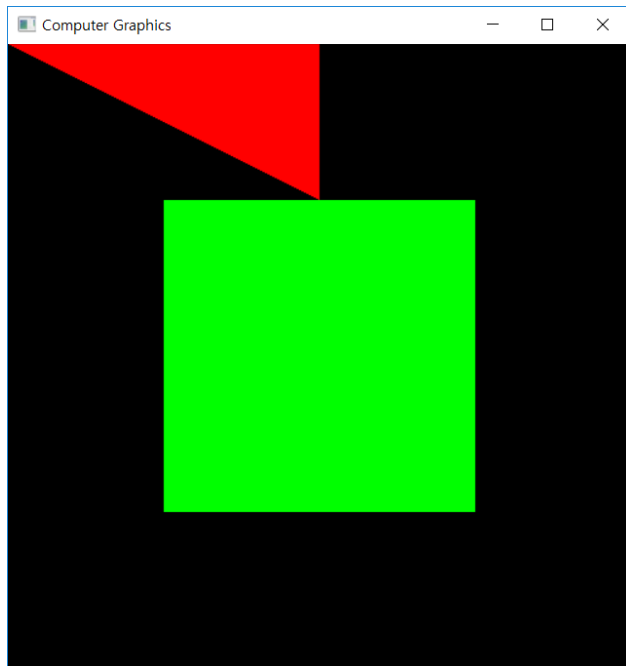
# Draw Other Shapes

---



- `glColor3d(0.0, 1.0, 0.0);`
- `glBegin(GL_QUADS);`
- `glVertex2d(-0.5, -0.5);`
- `glVertex2d(-0.5, 0.5);`
- `glVertex2d(0.5, 0.5);`
- `glVertex2d(0.5, -0.5);`
- `glEnd();`

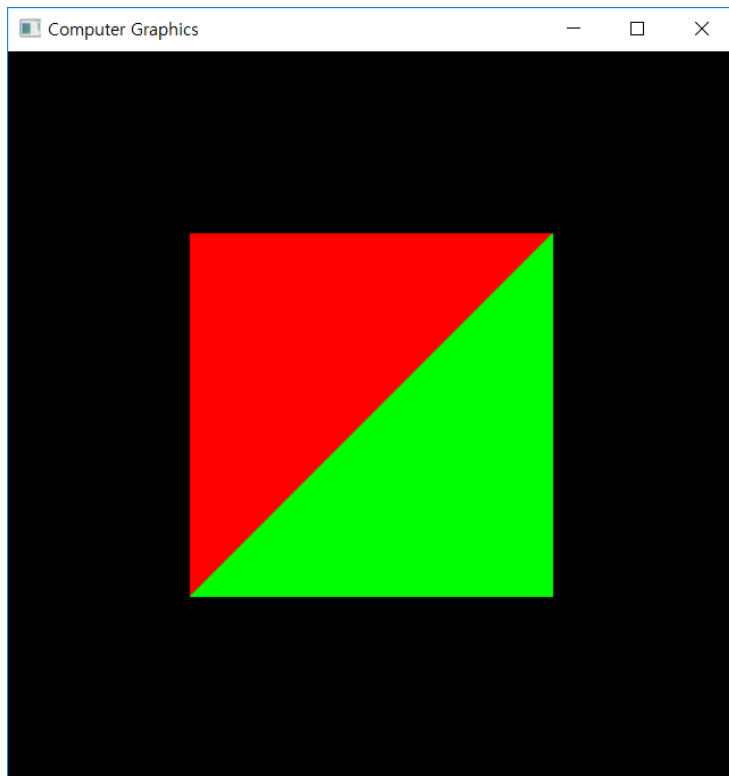
# Draw Multiple Primitives



- `glColor3d(1.0, 0.0, 0.0);`
- `glBegin(GL_TRIANGLES);`
  - `glVertex2d(-1.0, 1.0);`
  - `glVertex2d(0, 0.5);`
  - `glVertex2d(0, 1.0);`
- `glEnd();`
  
- `glColor3d(0.0, 1.0, 0.0);`
- `glBegin(GL_QUADS);`
  - `glVertex2d(-0.5, -0.5);`
  - `glVertex2d(-0.5, 0.5);`
  - `glVertex2d(0.5, 0.5);`
  - `glVertex2d(0.5, -0.5);`
- `glEnd();`

# Example: Draw Two Triangles

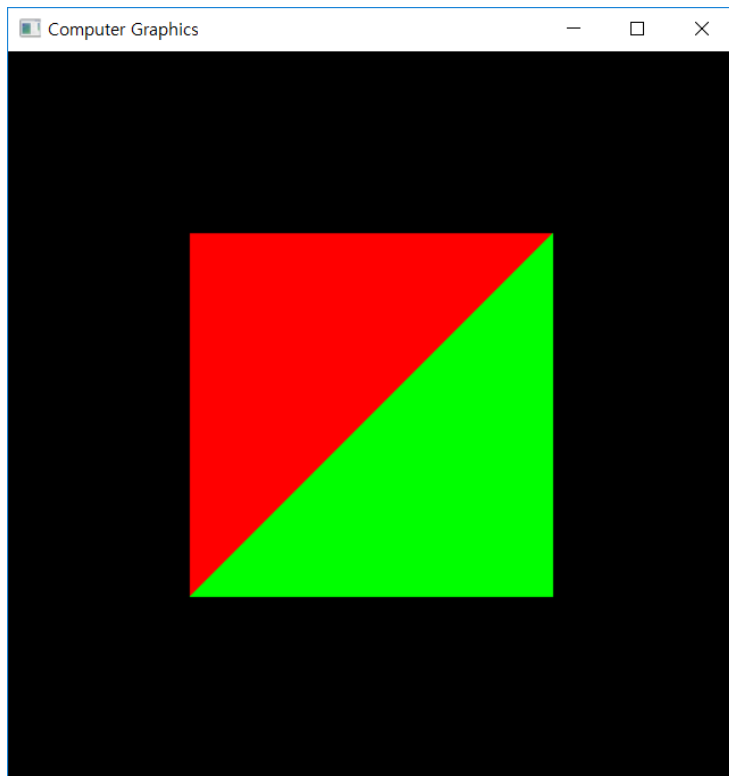
---



- `glColor3d(1.0, 0.0, 0.0);`
- `glBegin(GL_TRIANGLES);`
  - `glVertex2d(-0.5, -0.5);`
  - `glVertex2d(-0.5, 0.5);`
  - `glVertex2d(0.5, 0.5);`
- `glEnd();`
- `glColor3d(0.0, 1.0, 0.0);`
- `glBegin(GL_TRIANGLES);`
  - `glVertex2d(0.5, 0.5);`
  - `glVertex2d(-0.5, -0.5);`
  - `glVertex2d(0.5, -0.5);`
- `glEnd();`

# Example: Draw Two Triangles

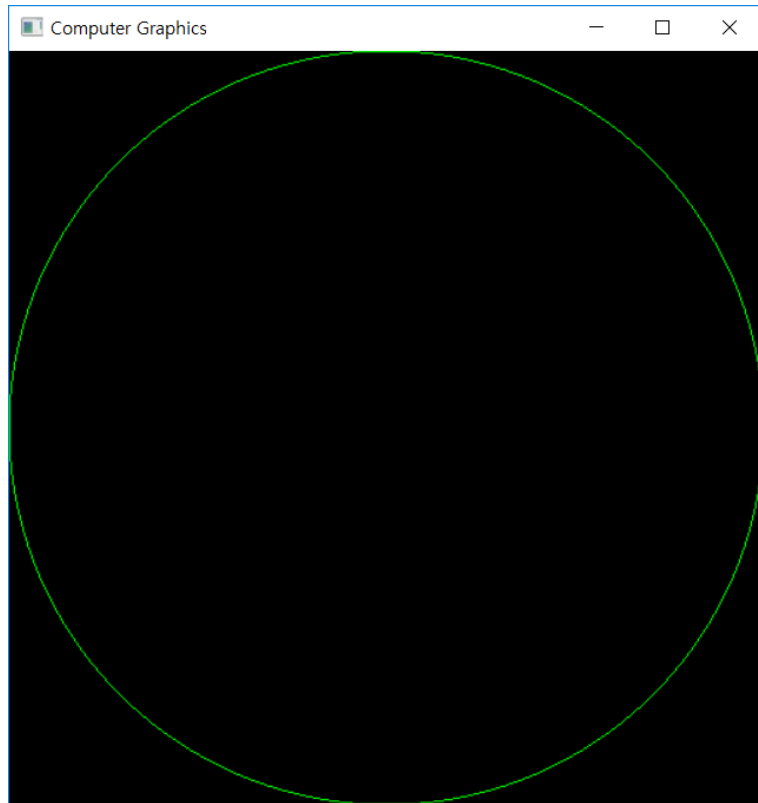
---



- `glColor3d(1.0, 0.0, 0.0);`
- `glBegin(GL_TRIANGLES);`
  - `glVertex2d(-0.5, -0.5);`
  - `glVertex2d(-0.5, 0.5);`
  - `glVertex2d(0.5, 0.5);`
- `glEnd();`
- `glColor3d(0.0, 1.0, 0.0);`
- `glBegin(GL_TRIANGLES);`
  - `glVertex2d(0.5, 0.5);`
  - `glVertex2d(-0.5, -0.5);`
  - `glVertex2d(0.5, -0.5);`
- `glEnd();`

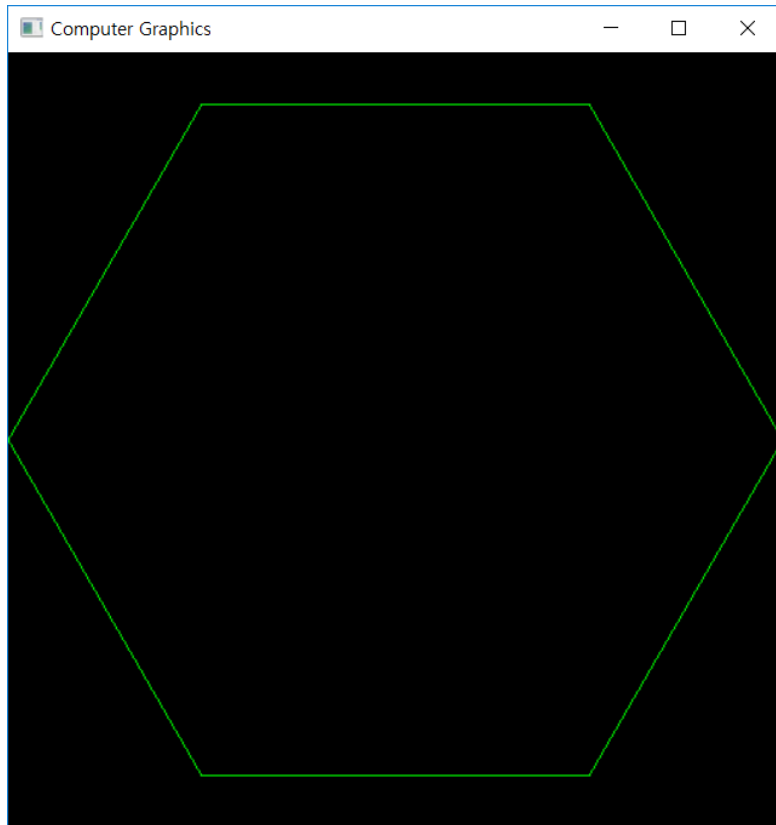
# Other Examples

---



- `glColor3d(0.0, 1.0, 0.0);`
- `glBegin(GL_LINE_LOOP);`
- `for (int i = 0; i < 360; i = i + 1) {`
  - `double x = cos(i * PI / 180);`
  - `double y = sin(i * PI / 180);`
  - `glVertex2d(x, y);`
- `}`
- `glEnd();`

# Other Examples



- `glColor3d(0.0, 1.0, 0.0);`
- `glBegin(GL_LINE_LOOP);`
- `for (int i = 0; i < 360; i = i + 60){`
  - `double x = cos(i * PI / 180);`
  - `double y = sin(i * PI / 180);`
  - `glVertex2d(x, y);`
- `}`
- `glEnd();`

# Other Examples

---



- `glBegin(GL_LINE_LOOP);`
- `for (int i = 0; i < 360; i = i + 1){`
  - `double x = cos(i * PI / 180);`
  - `double y = sin(i * PI / 180);`
  - `if (i < 180)`
    - `glColor3d(1.0, 0.0, 0.0);`
  - `else`
    - `glColor3d(0.0, 1.0, 0.0);`
  - `glVertex2d(x, y);`
- `}`
- `glEnd();`