OpenGL 그래픽스 라이브러리

윤진용

3D Modeling and Processing Lab

OpenGL

- 2차원 및 3차원 그래픽 이미지를 정의하기 위한 컴퓨터 산업계의 표준 응용프로그램 인터페이스
- http://www.opengl.org/
- 특징
  - Mouse, keyboard, menu등을 통한 interaction
  - 기본적인 drawing primitive들 제공
  - 간단한 animation 기능 제공

OpenGL APIs

- OpenGL core library
  - gl.h opengl32.lib opengl32.dll
- GLU(OpenGL Utility Library) - part of openGL
  - glu.h glu32.lib glu32.dll
- GLUT(OpenGL Utility Toolkit) - not part of openGL
  - glut.h glut32.lib glut32.dll
- OpenGL Extensions
  - glew, etc..

Installation

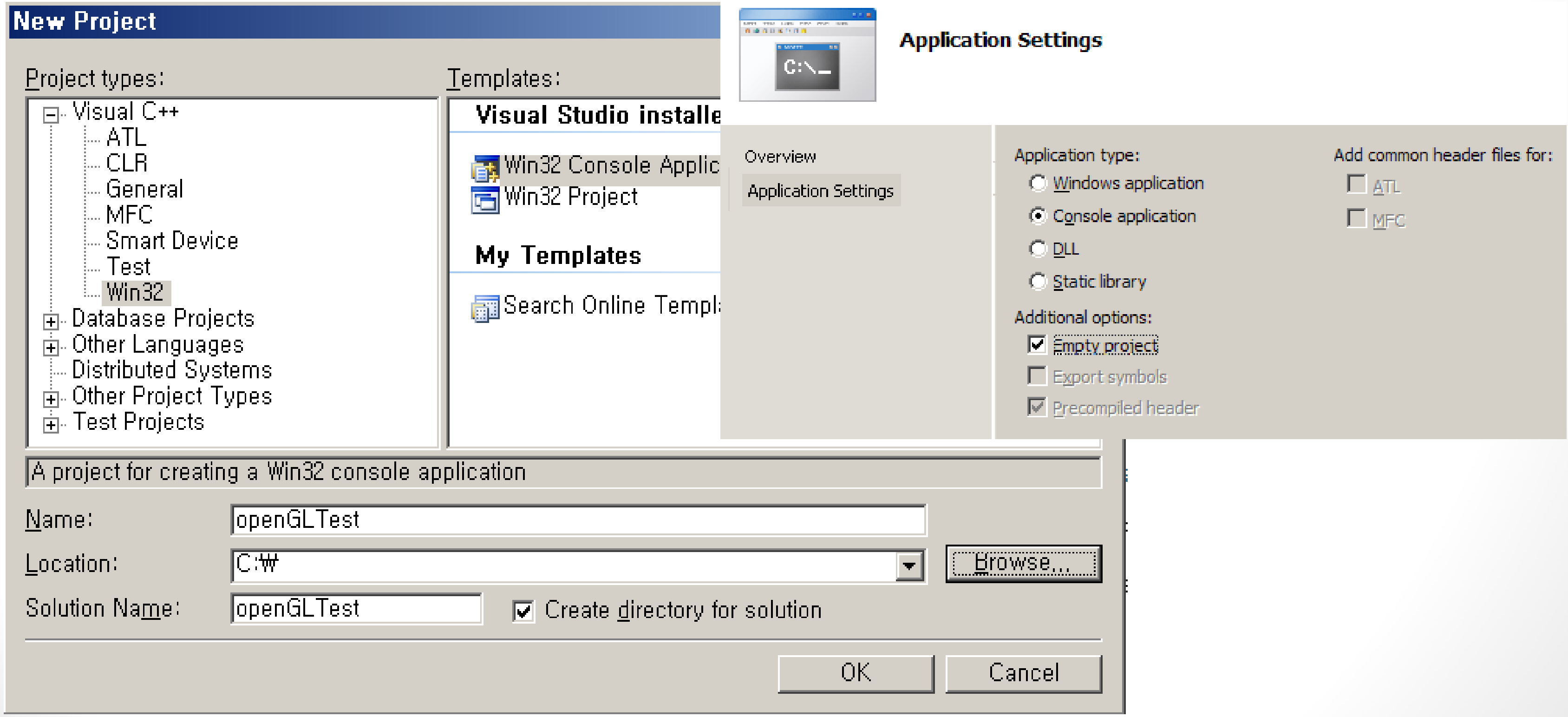
- OpenGL core, GLU는 이미 windows에 설치되어있음
- GLUT
  - Mark Kilgard가 개발한 GLUT는 open source가 아니고 오랫동안 업데이트 되지 않았기 때문에 freeglut를 대신 사용
  - http://freeglut.sourceforge.net/
  - http://www.transmissionzero.co.uk/software/freeglut-devel/ (prepackaged for MSVC and MinGW)

Installation

- **freeglut 2.6.0 MSVC Package** 다운, 압축해제
- header
    - $(VSInstallDir)\VC\include에 GL폴더 복사
- lib
    - $(VSInstallDir)\VC\lib에 freeglut.lib 복사
- dll
    - windows\system32 또는 실행폴더에 freeglut.dll 복사

Project Setting (1/3)

- File>New>Project>Win32 Console Application
- Select empty project



New Project

Project types:

- Visual C++
  - ATL
  - CLR
  - General
  - MFC
  - Smart Device
  - Test
  - Win32
- Database Projects
- Other Languages
- Distributed Systems
- Other Project Types
- Test Projects

Templates:

Visual Studio installe

Win32 Console Applic

Win32 Project

My Templates

Search Online Templ

A project for creating a Win32 console application

Name: openGLTest

Location: C:₩

Solution Name: openGLTest

☑ Create directory for solution

Application Settings

Overview

Application Settings

Application type:

○ Windows application

● Console application

○ DLL

○ Static library

Additional options:

☑ Empty project

☐ Export symbols

☑ Precompiled header

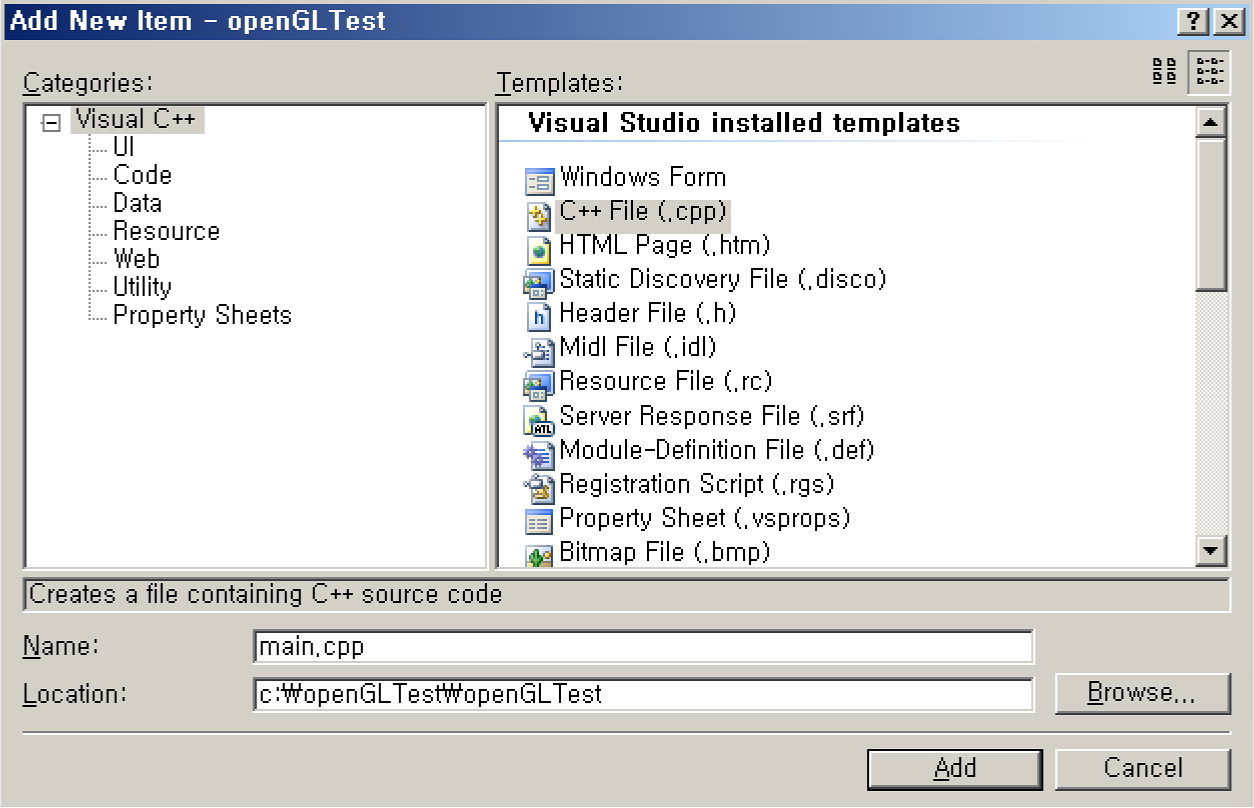
Add common header files for:

☐ ATL

☐ MFC

Project Setting (2/3)

- Project>Add New Item>C++ File



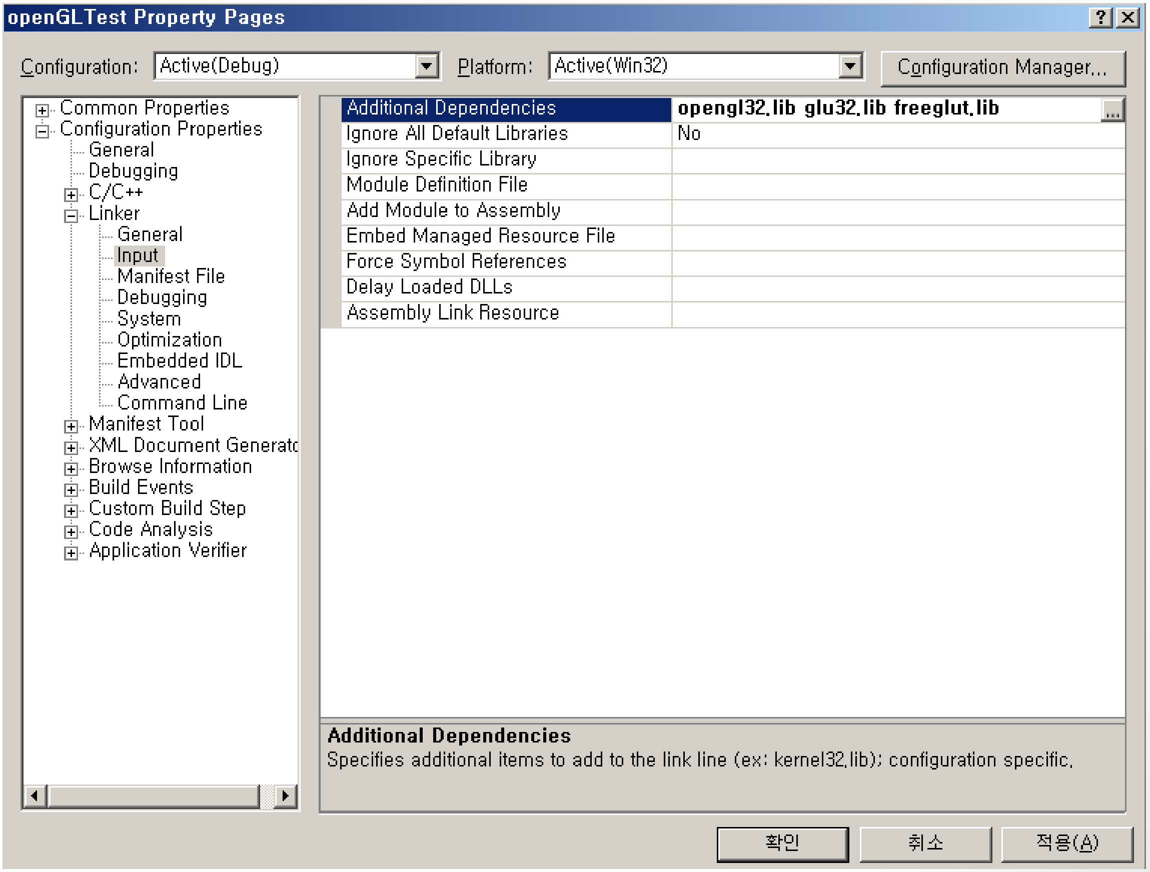
Creates a file containing C++ source code

Name: main.cpp

Location: c:₩openGLTest₩openGLTest

Project Setting (3/3)

- View>Property Pages>Add additional dependencies



openGLTest Property Pages

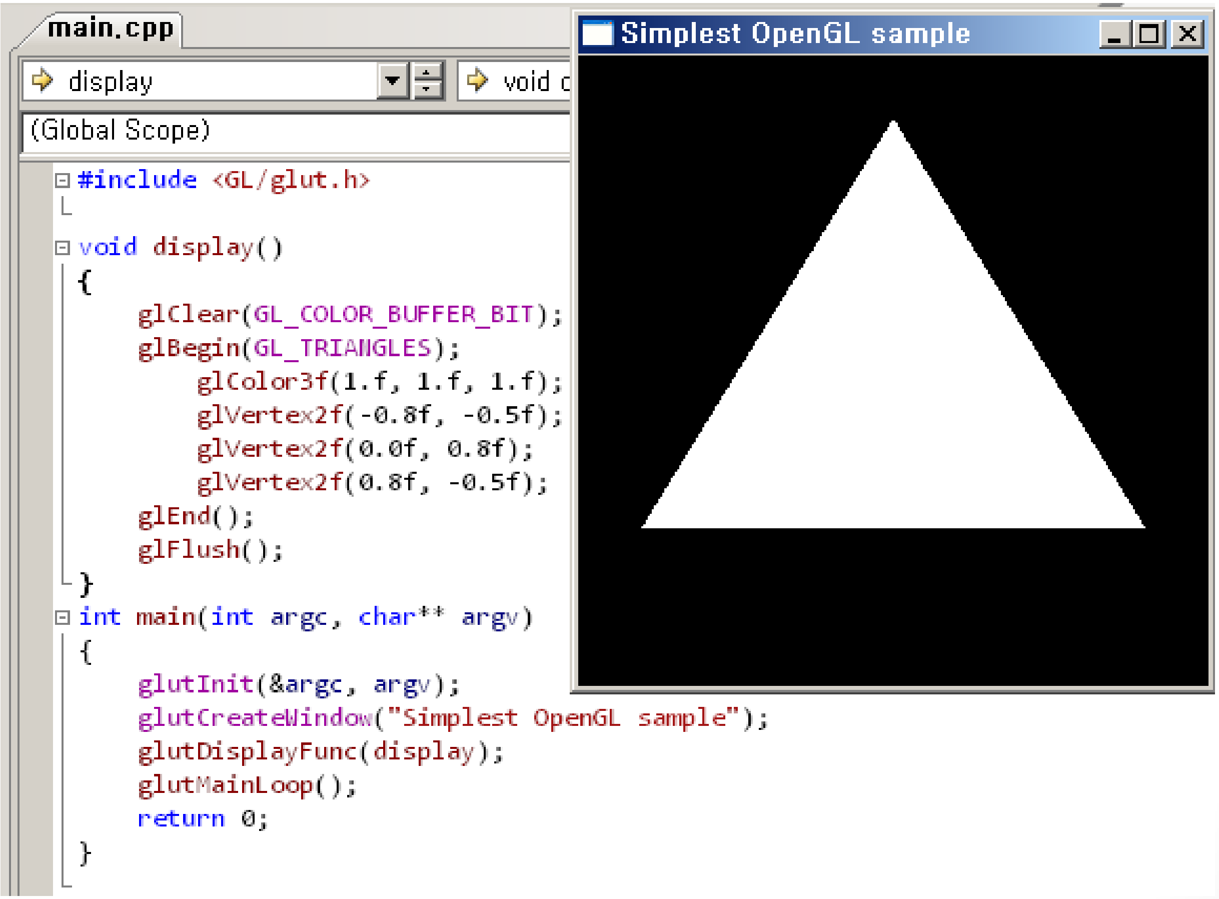
Configuration: Active(Debug)

Platform: Active(Win32)

Additional Dependencies	opengl32.lib glu32.lib freeglut.lib
Ignore All Default Libraries	No
Ignore Specific Library	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded DLLs	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the link line (ex: kernel32.lib); configuration specific.

Simple Project



```
#include <GL/glut.h>

void display()
{
	glClear(GL_COLOR_BUFFER_BIT);
	glBegin(GL_TRIANGLES);
		glColor3f(1.f, 1.f, 1.f);
		glVertex2f(-0.8f, -0.5f);
		glVertex2f(0.0f, 0.8f);
		glVertex2f(0.8f, -0.5f);
	glEnd();
	glFlush();
}
int main(int argc, char** argv)
{
	glutInit(&argc, argv);
	glutCreateWindow("Simplest OpenGL sample");
	glutDisplayFunc(display);
	glutMainLoop();
	return 0;
}
```

Basic Template

- Main function

```
int main(int argc, char **argv)
{
    glutInit(&argc, argv);

    // Initialize windows..
    // Register callback functions
    // Initialize openGL properties

    glutMainLoop();
    return 0;
}
```

Basic Template

- Initialize windows

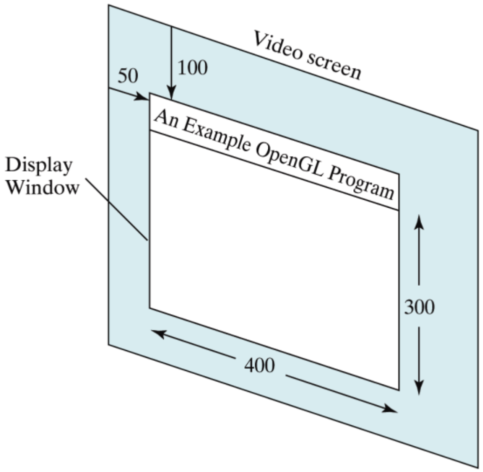


Figure 2-61

A 400 by 300 display window at position (50, 100) relative to the top-left corner of the video display.

```
void initWindow()
{
    glutInitWindowPosition(50, 100);
    glutInitWindowSize(400, 300);
    glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);

    glutCreateWindow(”An Example OpenGL Program”);
}
```

Basic Template

- Register callback functions

```
void initCallbackFunc()
{
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);

    glutKeyboardFunc(keyboard);
    glutMouseFunc(mouse);
    glutMotionFunc(motion);
    glutIdleFunc(idle);
}
```

Basic Template

- Frequently used callback functions

```
void display();
void reshape(int width, int height);
void keyboard(unsigned char key, int x, int y);
void reshape(int button, int state, int x, int y);
void reshape(int x, int y);
void idle();
```

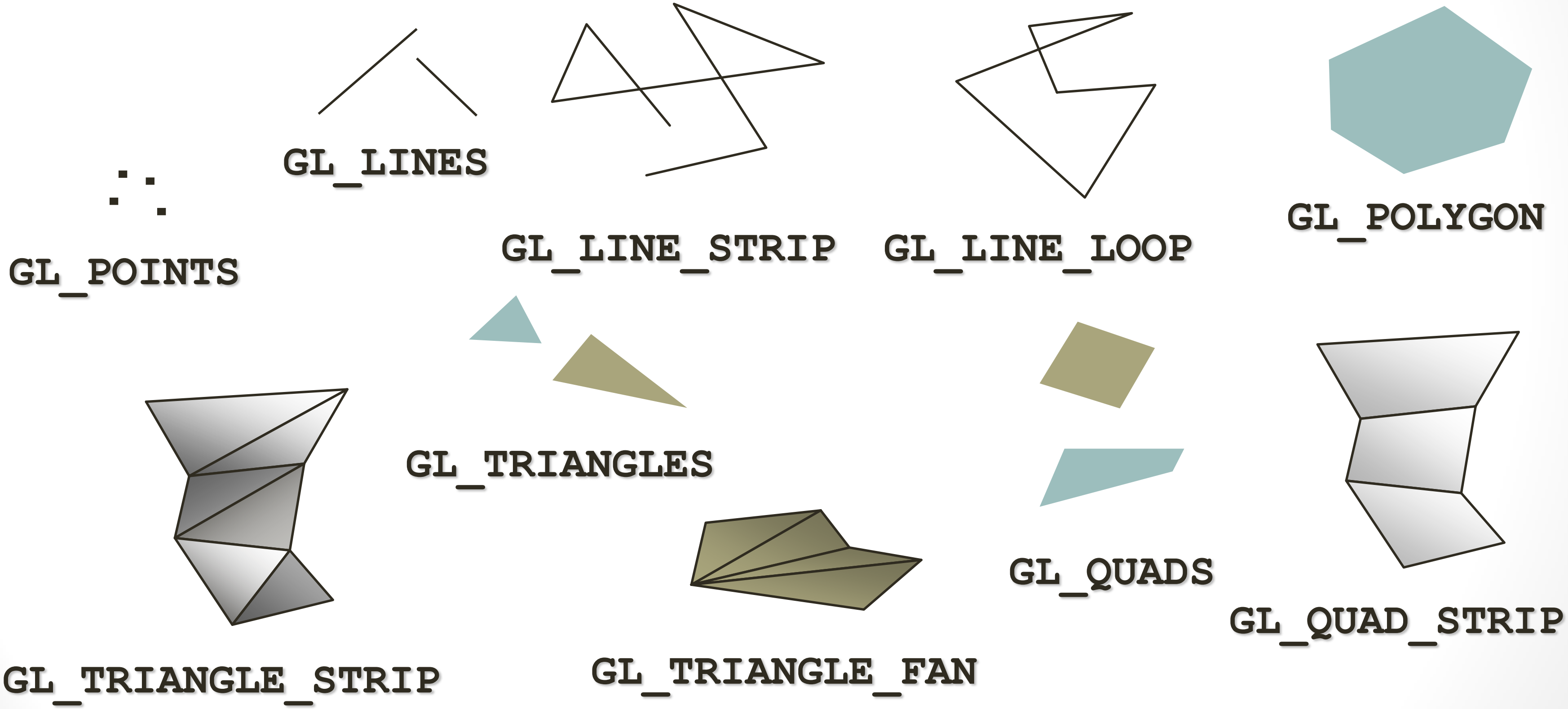
- 기타 callback function은 GLUT API spec을 참조.

Basic Template

- Initialize OpenGL properties

```
void initOpenGL()
{
    // light
    glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, specular);
    glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
    glLightfv(GL_LIGHT0, GL_POSITION, position);
    glEnable(GL_LIGHT0);
    glEnable(GL_LIGHTING);
    // materials, etc…
}
```

OpenGL Primitives



GL_POINTS

GL_LINES

GL_LINE_STRIP

GL_LINE_LOOP

GL_POLYGON

GL_TRIANGLES

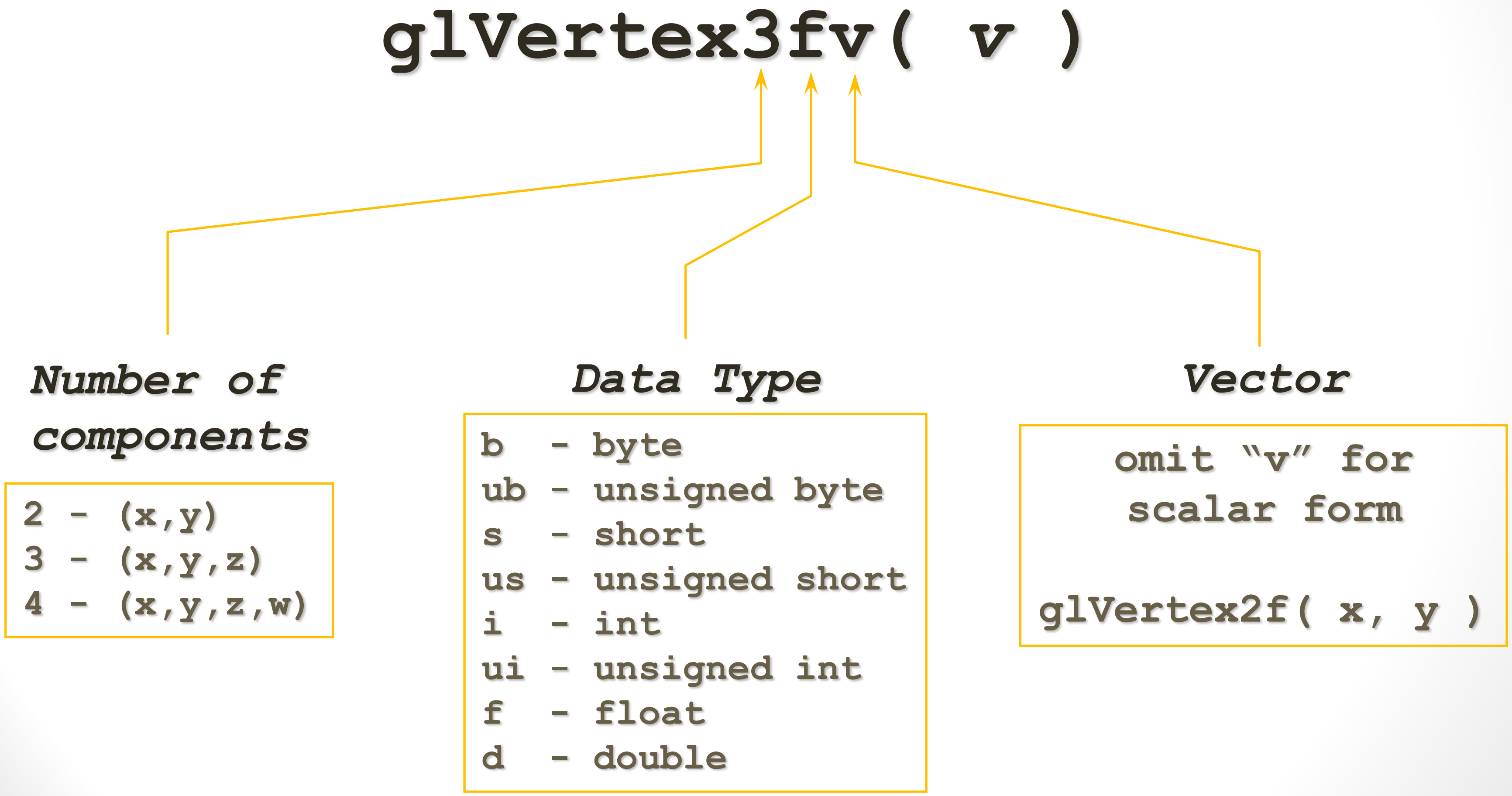
GL_QUADS

GL_QUAD_STRIP

GL_TRIANGLE_STRIP

GL_TRIANGLE_FAN

Drawing Primitives



glVertex3fv(*v*)

Number of components

2	-	(x,y)
3	-	(x,y,z)
4	-	(x,y,z,w)

Data Type

b	-	byte
ub	-	unsigned byte
s	-	short
us	-	unsigned short
i	-	int
ui	-	unsigned int
f	-	float
d	-	double

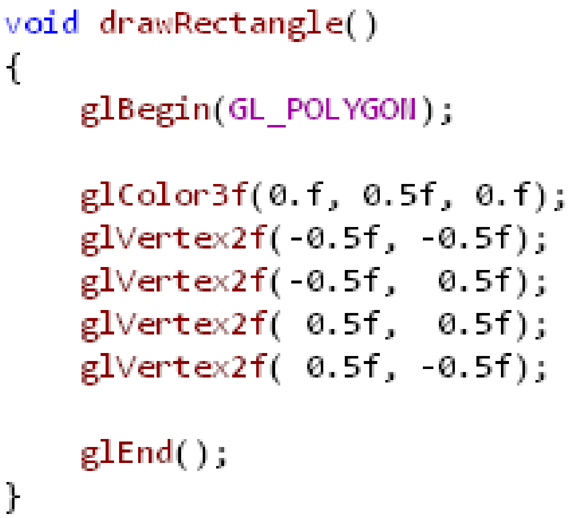
Vector

omit "v" for scalar form

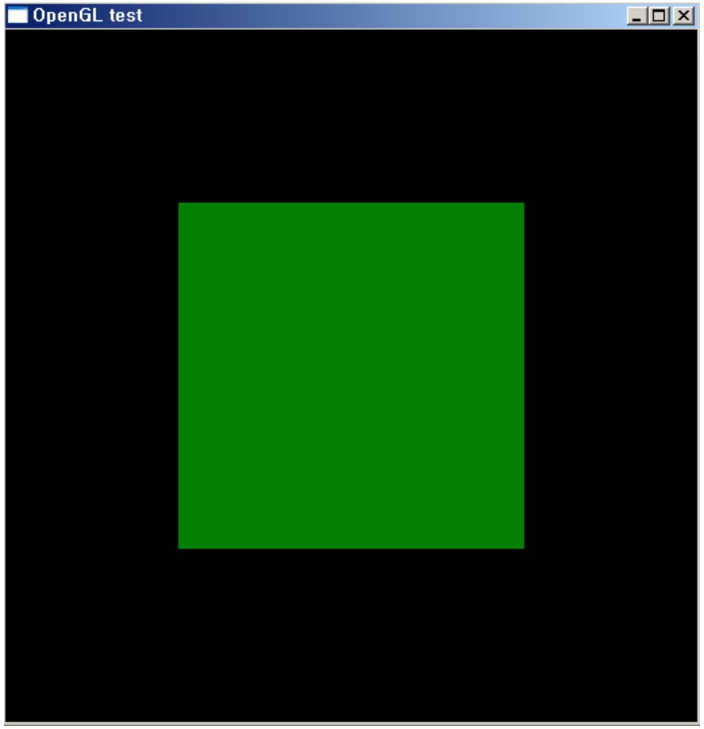
`glVertex2f( x, y )`

Drawing Primitives

- example



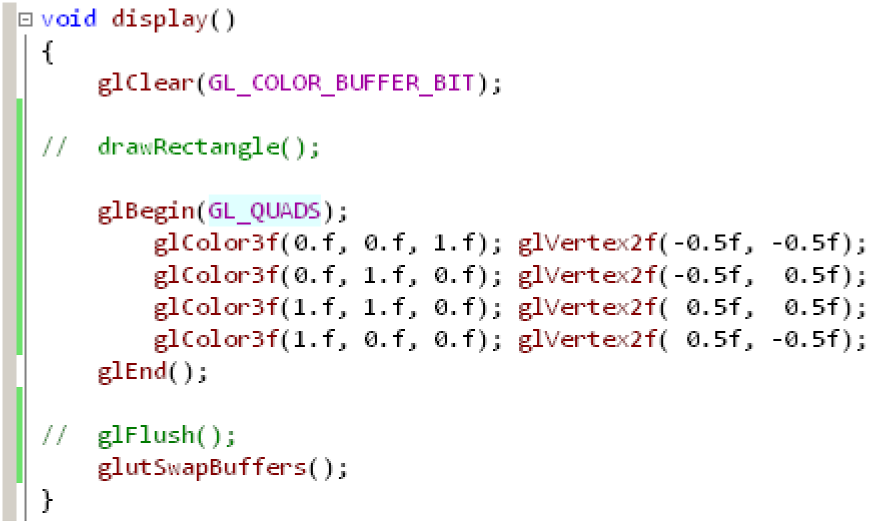
```
void drawRectangle()
{
    glBegin(GL_POLYGON);

    glColor3f(0.f, 0.5f, 0.f);
    glVertex2f(-0.5f, -0.5f);
    glVertex2f(-0.5f,  0.5f);
    glVertex2f( 0.5f,  0.5f);
    glVertex2f( 0.5f, -0.5f);

    glEnd();
}
```



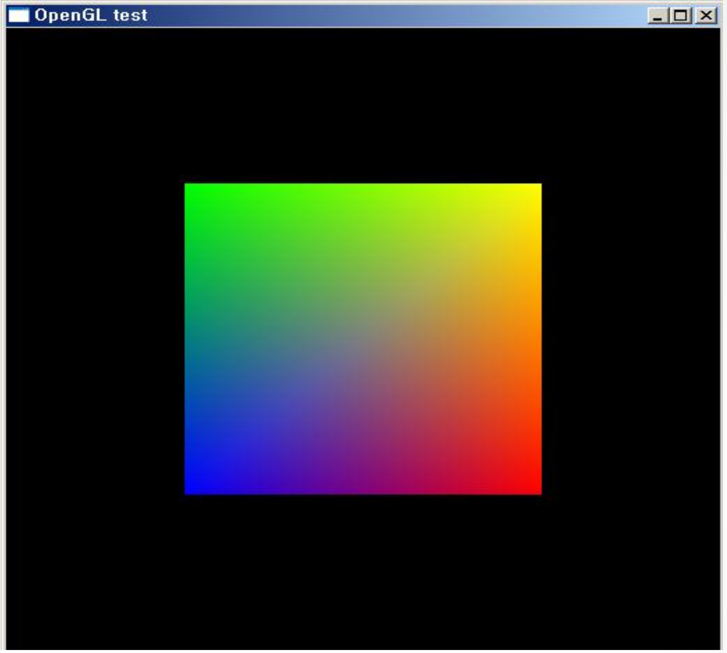
  - color
  - normal
  - texture coordinate

Display Function

- example



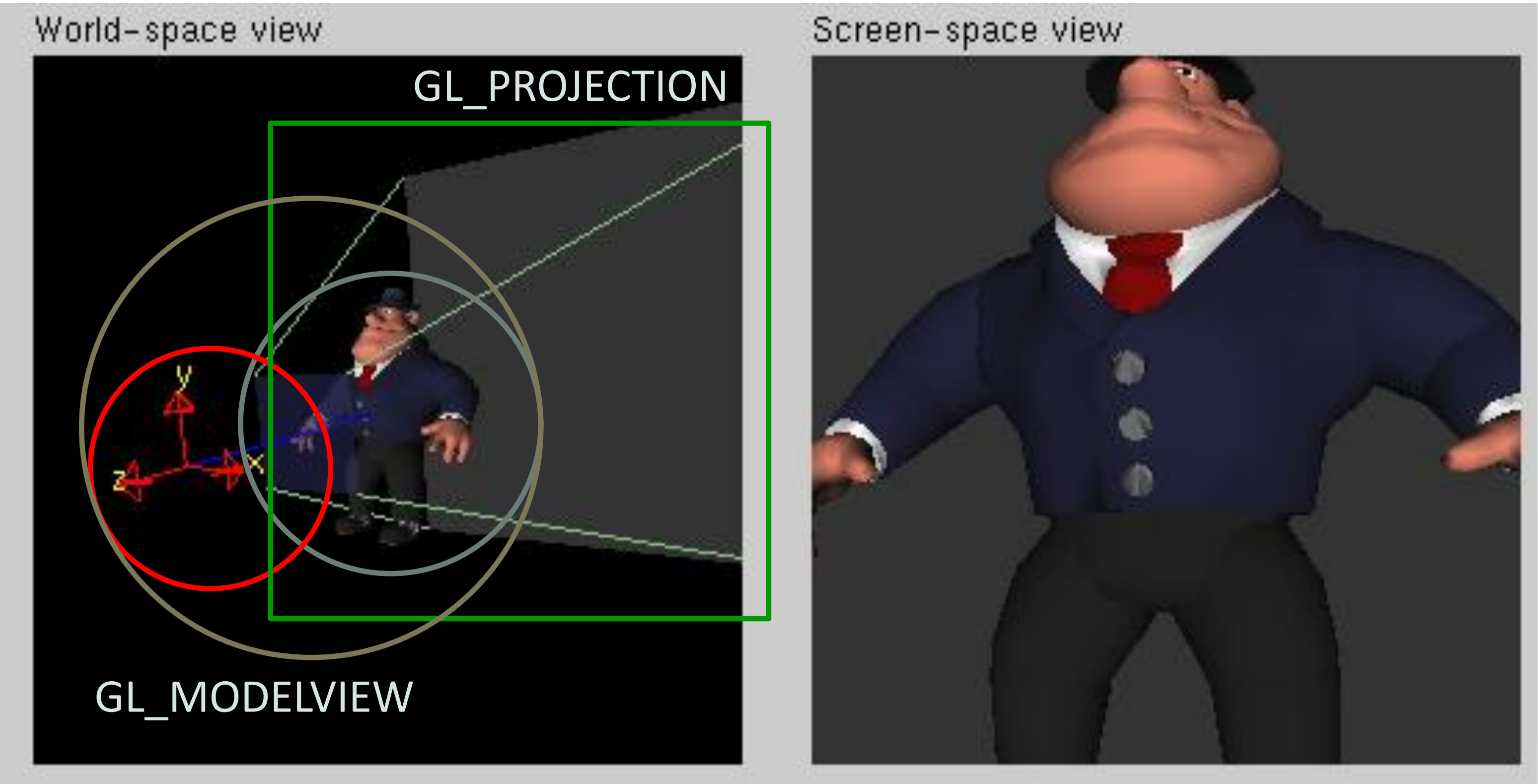
```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT);

//  drawRectangle();

    glBegin(GL_QUADS);
        glColor3f(0.f, 0.f, 1.f); glVertex2f(-0.5f, -0.5f);
        glColor3f(0.f, 1.f, 0.f); glVertex2f(-0.5f,  0.5f);
        glColor3f(1.f, 1.f, 0.f); glVertex2f( 0.5f,  0.5f);
        glColor3f(1.f, 0.f, 0.f); glVertex2f( 0.5f, -0.5f);
    glEnd();

//  glFlush();
    glutSwapBuffers();
}
```



  - glutInitDisplayMode(GLUT_RGB|GLUT_DEPTH|GLUT_DOUBLE);
  - glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
  - glutPostRedisplay()

3D Rendering

- Camera analogy



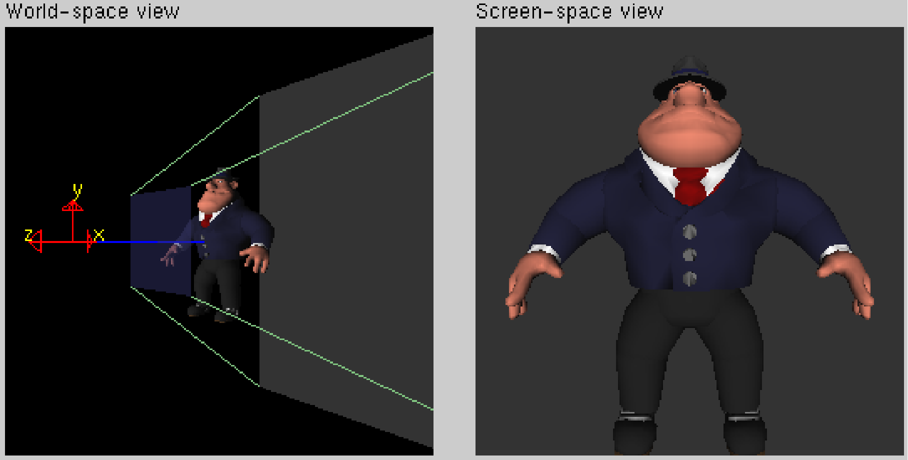
3D Rendering

- Transformation
  - Model-View matrix
    - glTranslate
    - glRotate
    - glScale
    - gluLookAt
  - Projection matrix
    - glOrtho
    - gluOrtho2D
    - glFrustum
    - gluPerspective

- Common
  - glMatrixMode
  - glLoadIdentity
  - glPushMatrix
  - glPopMatrix

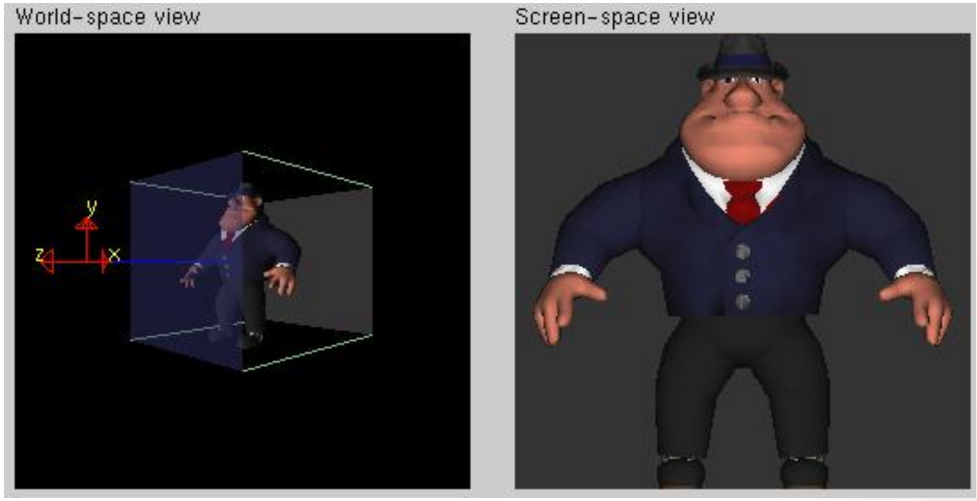
3D Rendering

- Projection transformation



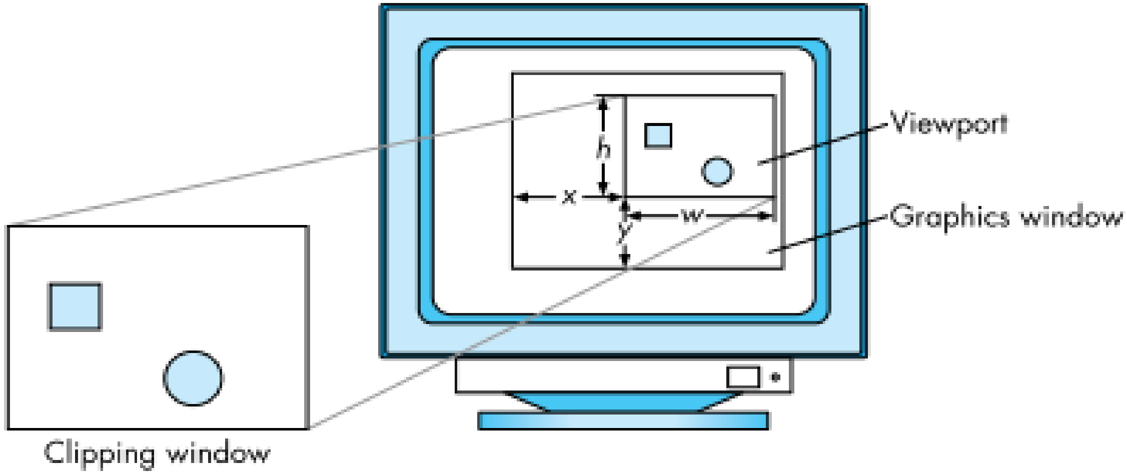
gluPerspective

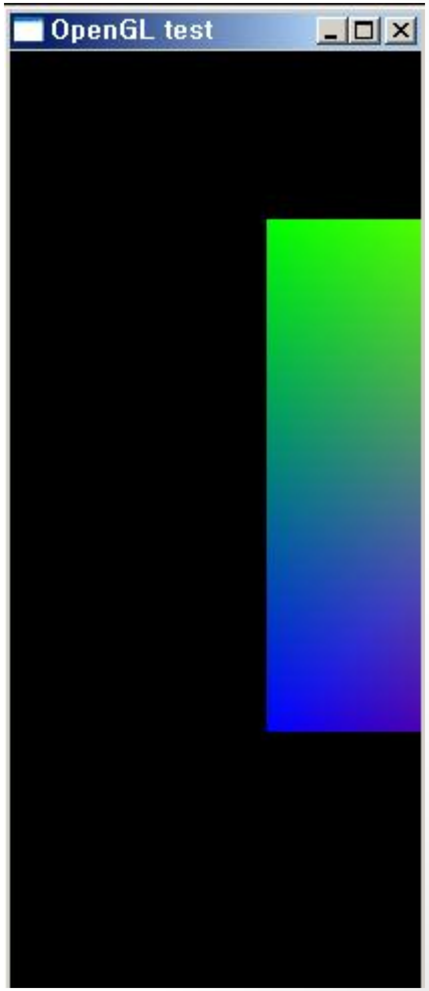
glOrtho



Reshape Function

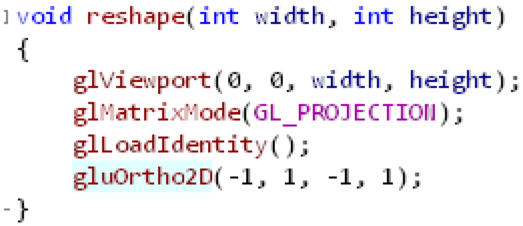
- void reshape(int width, int height)
  - glViewport(x, y, width, height)



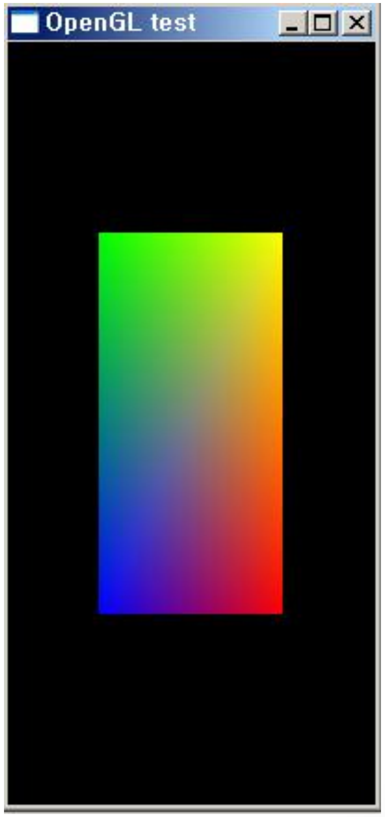


Reshape Function

- generate new viewing volume
  - glOrtho(-width/2, width/2, -height/2, height/2, -1, 1)
  - gluPerspective(fovy, width/height, zNear, zFar)

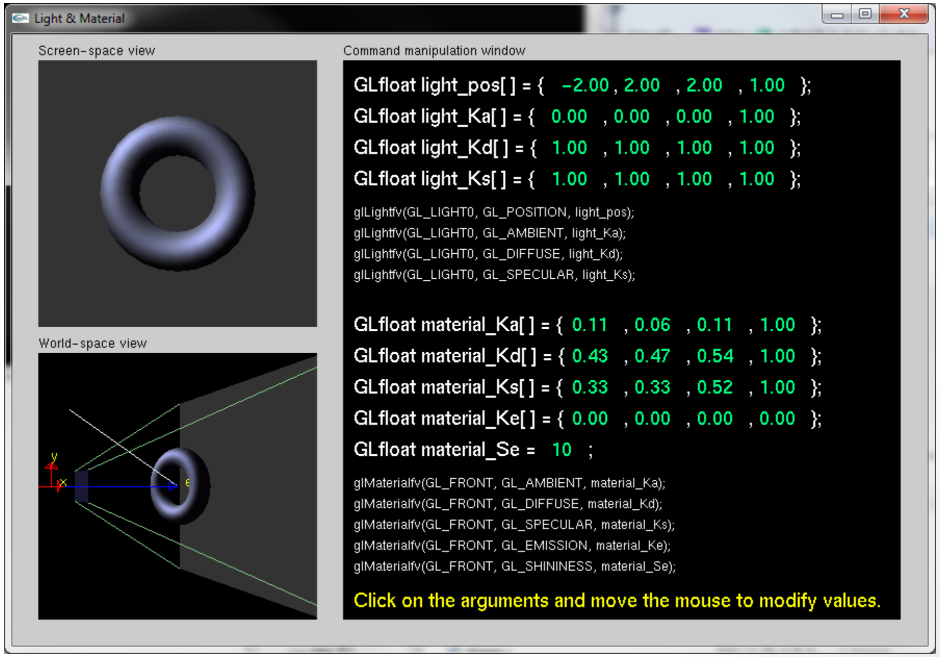


```
void reshape(int width, int height)
{
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-1, 1, -1, 1);
}
```



References

- http://www.opengl.org/sdk/docs/man/
- http://www.opengl.org/documentation/specs/glut/spec3/spec3.html
- http://freeglut.sourceforge.net/docs/api.php
- http://www.xmission.com/~nate/tutors.html



```
GLfloat light_pos[ ] = {  -2.00 , 2.00  , 2.00  , 1.00  };
GLfloat light_Ka[ ] = {  0.00  , 0.00  , 0.00  , 1.00  };
GLfloat light_Kd[ ] = {  1.00  , 1.00  , 1.00  , 1.00  };
GLfloat light_Ks[ ] = {  1.00  , 1.00  , 1.00  , 1.00  };
glLightfv(GL_LIGHT0, GL_POSITION, light_pos);
glLightfv(GL_LIGHT0, GL_AMBIENT, light_Ka);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_Kd);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_Ks);

GLfloat material_Ka[ ] = { 0.11  , 0.06  , 0.11  , 1.00  };
GLfloat material_Kd[ ] = { 0.43  , 0.47  , 0.54  , 1.00  };
GLfloat material_Ks[ ] = { 0.33  , 0.33  , 0.52  , 1.00  };
GLfloat material_Ke[ ] = { 0.00  , 0.00  , 0.00  , 0.00  };
GLfloat material_Se =  10  ;
glMaterialfv(GL_FRONT, GL_AMBIENT, material_Ka);
glMaterialfv(GL_FRONT, GL_DIFFUSE, material_Kd);
glMaterialfv(GL_FRONT, GL_SPECULAR, material_Ks);
glMaterialfv(GL_FRONT, GL_EMISSION, material_Ke);
glMaterialfv(GL_FRONT, GL_SHININESS, material_Se);
```

Click on the arguments and move the mouse to modify values.