OpenGL

장준영

3D Modeling and Processing Lab

OpenGL

- 2차원 및 3차원 그래픽스 표준 API
- Cross-language, multi-platform
  - 개발: 실리콘 그래픽스->크로노스 그룹
  - 비영리
  - 등등…

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 등등

Freeglut

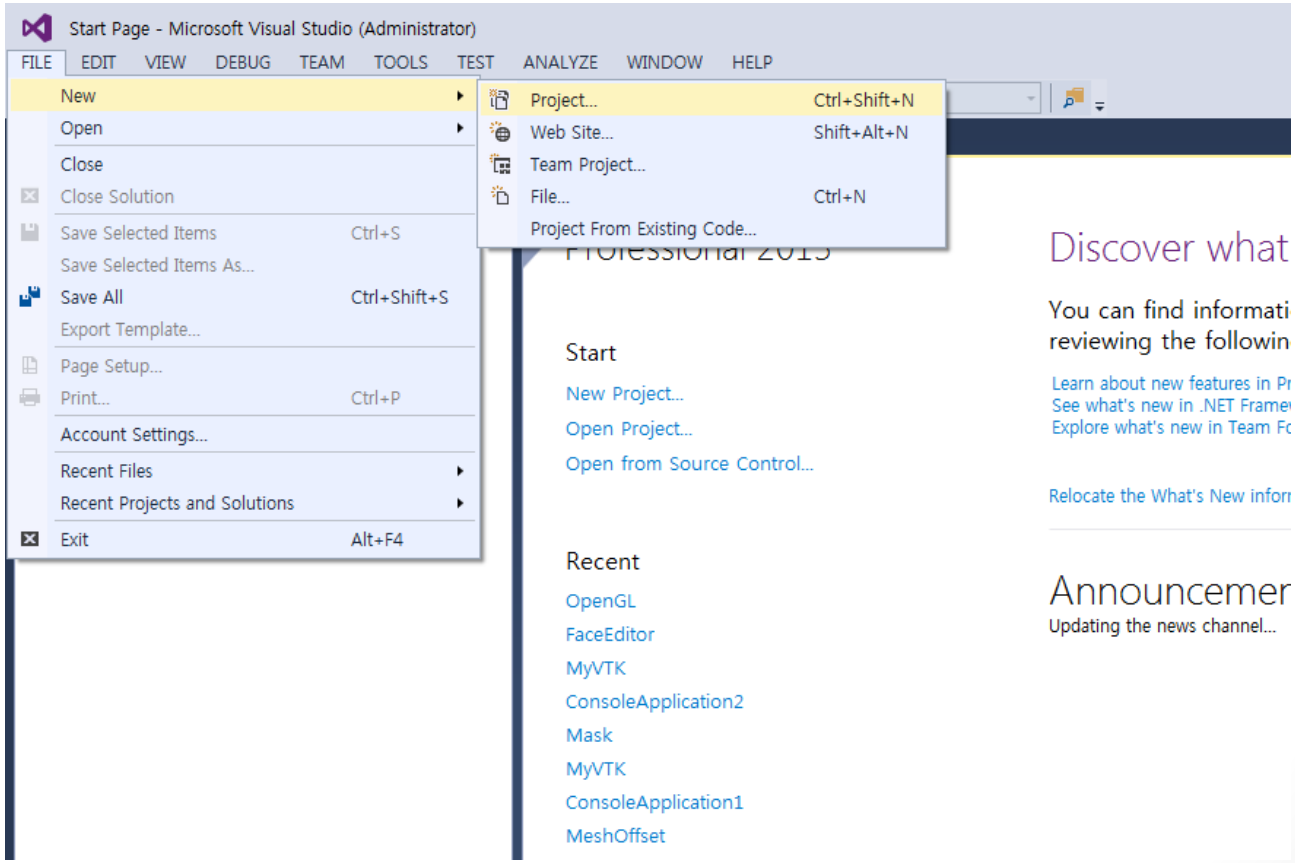
- GLUT
  - Keyboard, Mouse input, Window size제어 등의 편리한 기능을 제공하지만...
  - open source가 아니고, update도 없다.
- 이 수업에서는 Freeglut을 사용한다.

- http://www.transmissionzero.co.uk/software/freeglut-devel/
  - **freeglut 2.8.1 MSVC Package**를 다운로드

  - 압축을 풀고, 9페이지에서 지시하는 대로 파일을 복사한다.

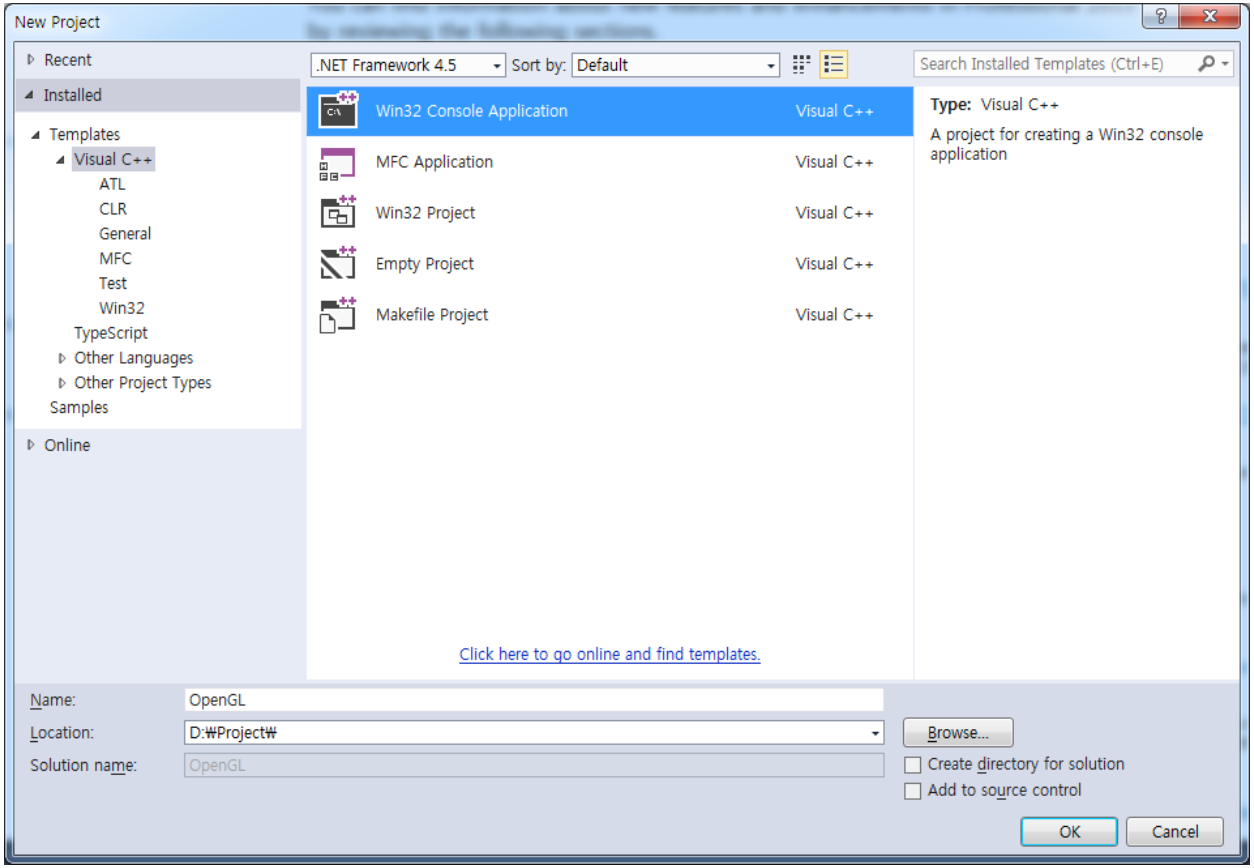
Project Setting

- New Visual Studio project



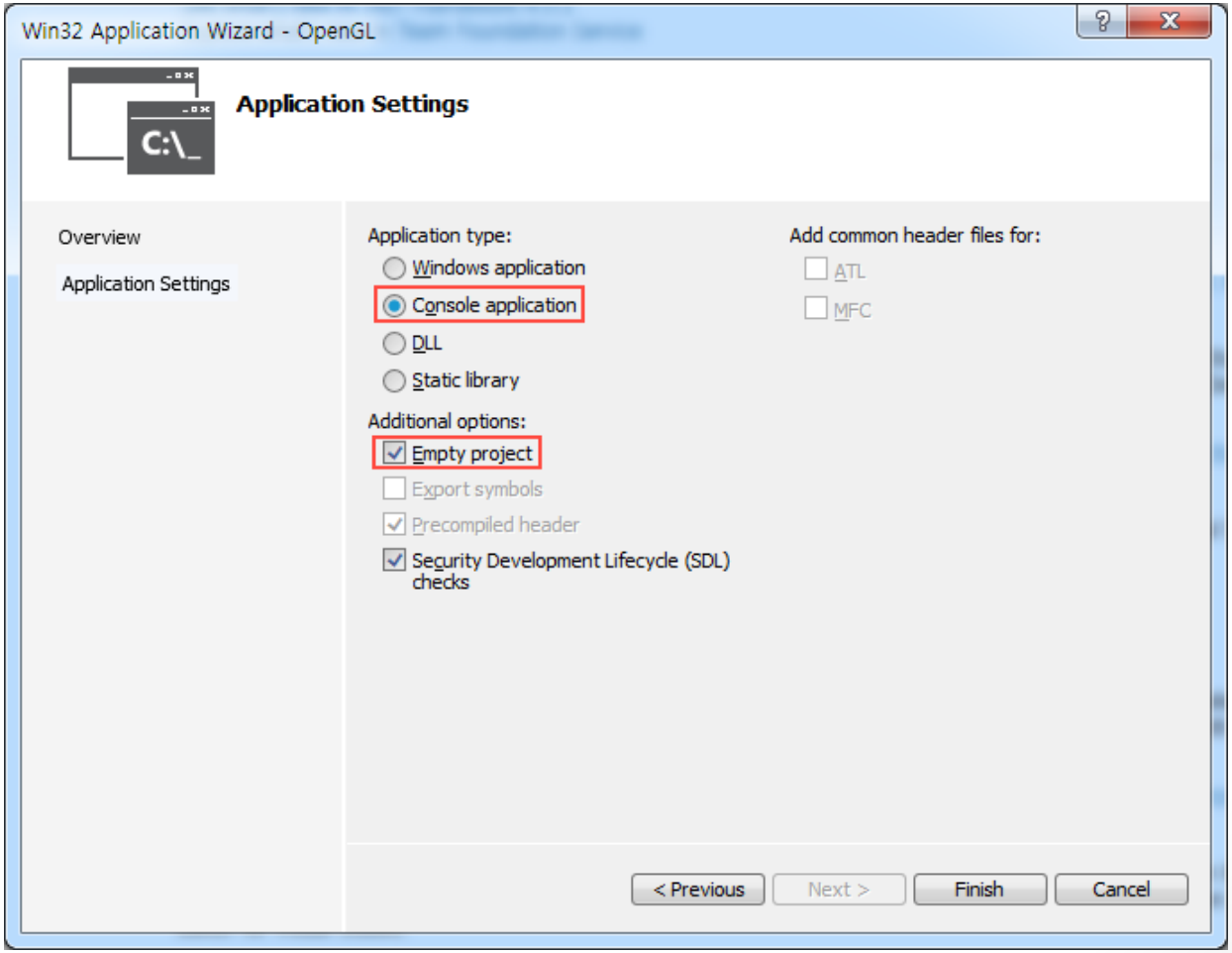
Project Setting

- Win32 Console Application



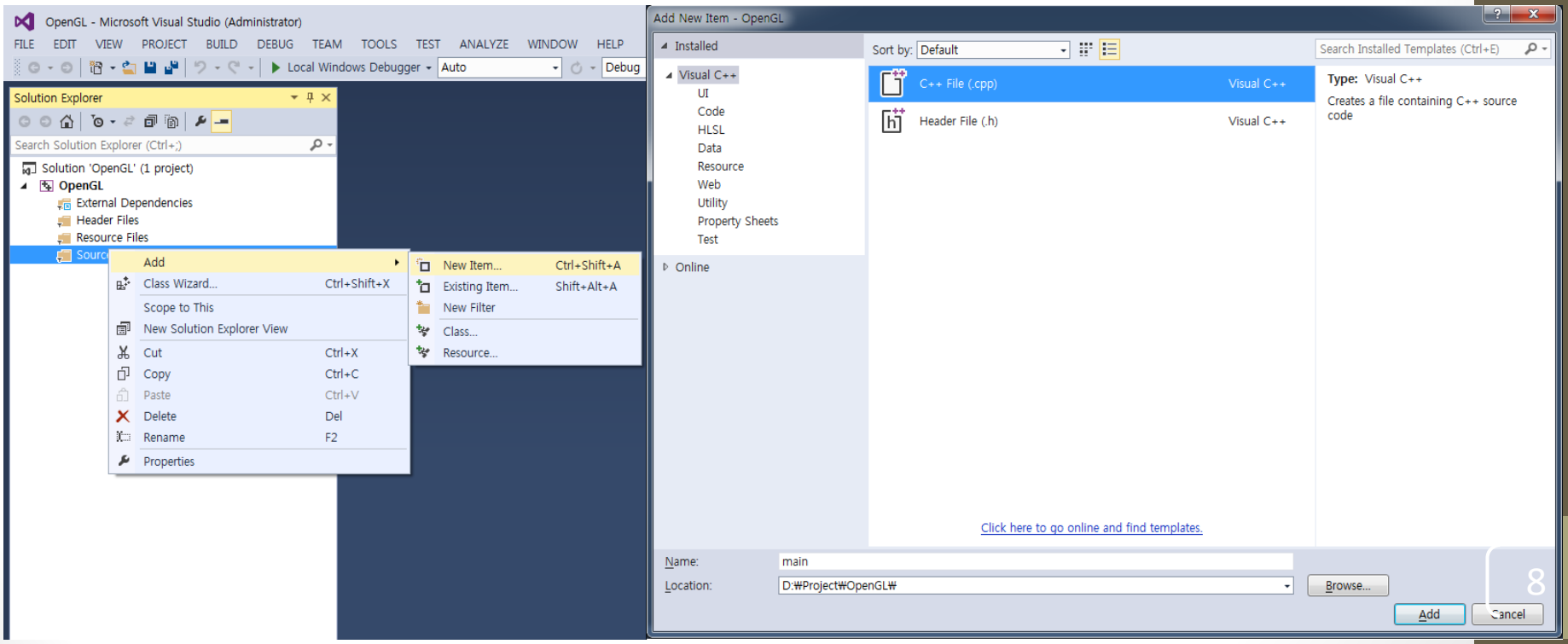
Project Setting

- Empty project



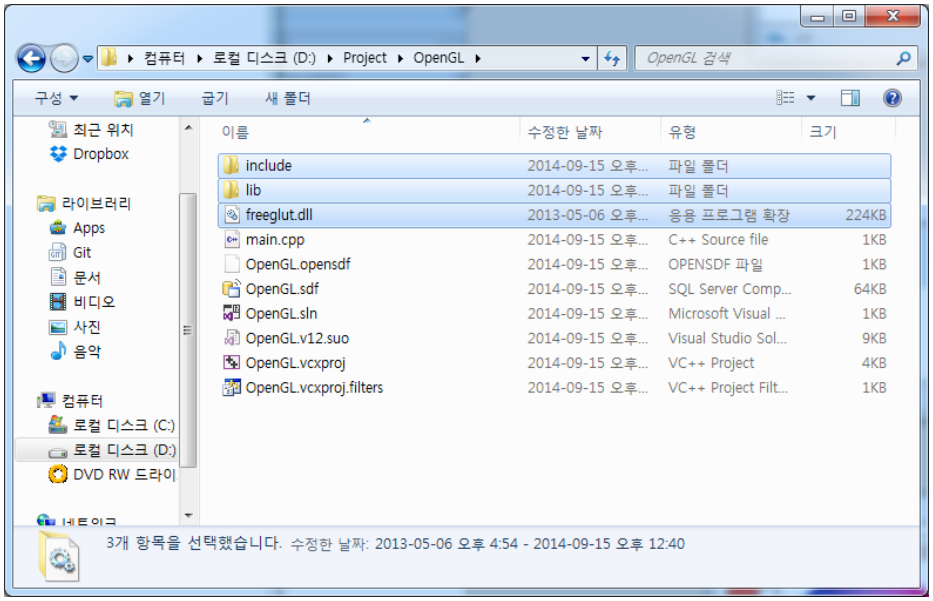
Project Setting

- Add new item
- C++ file



Project Setting

- Copy files to project root folder(32-bit)
  - Copy include, lib folders to root folder
  - Copy bin\freeglut.dll to root folder



  - If you want to make a 64-bit program, it's different(see next page)

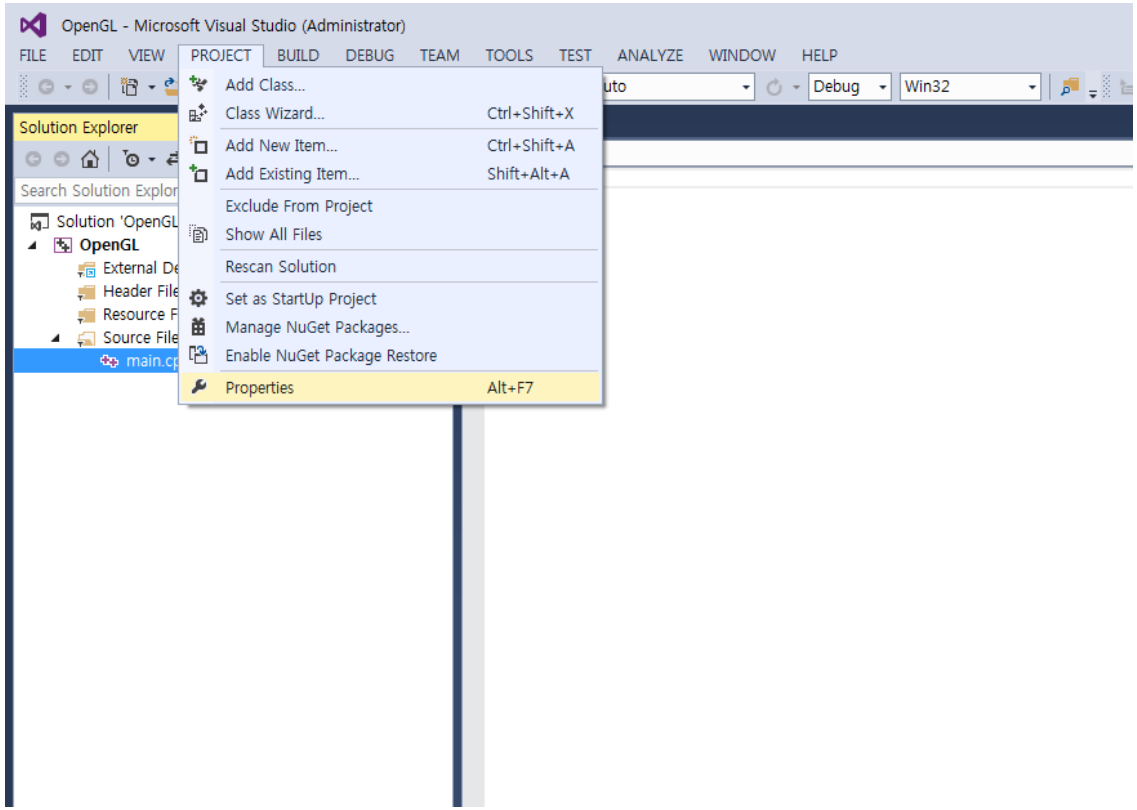
Project Setting

- Copy files to project root folder(64-bit)
  - Copy include folder (same!)
  - Copy **lib\x64\freeglut.lib** to lib folder
  - Copy **bin\x64\freeglut.dll** to root folder

- Note: You can make a 32-bit program on a 64-bit system.

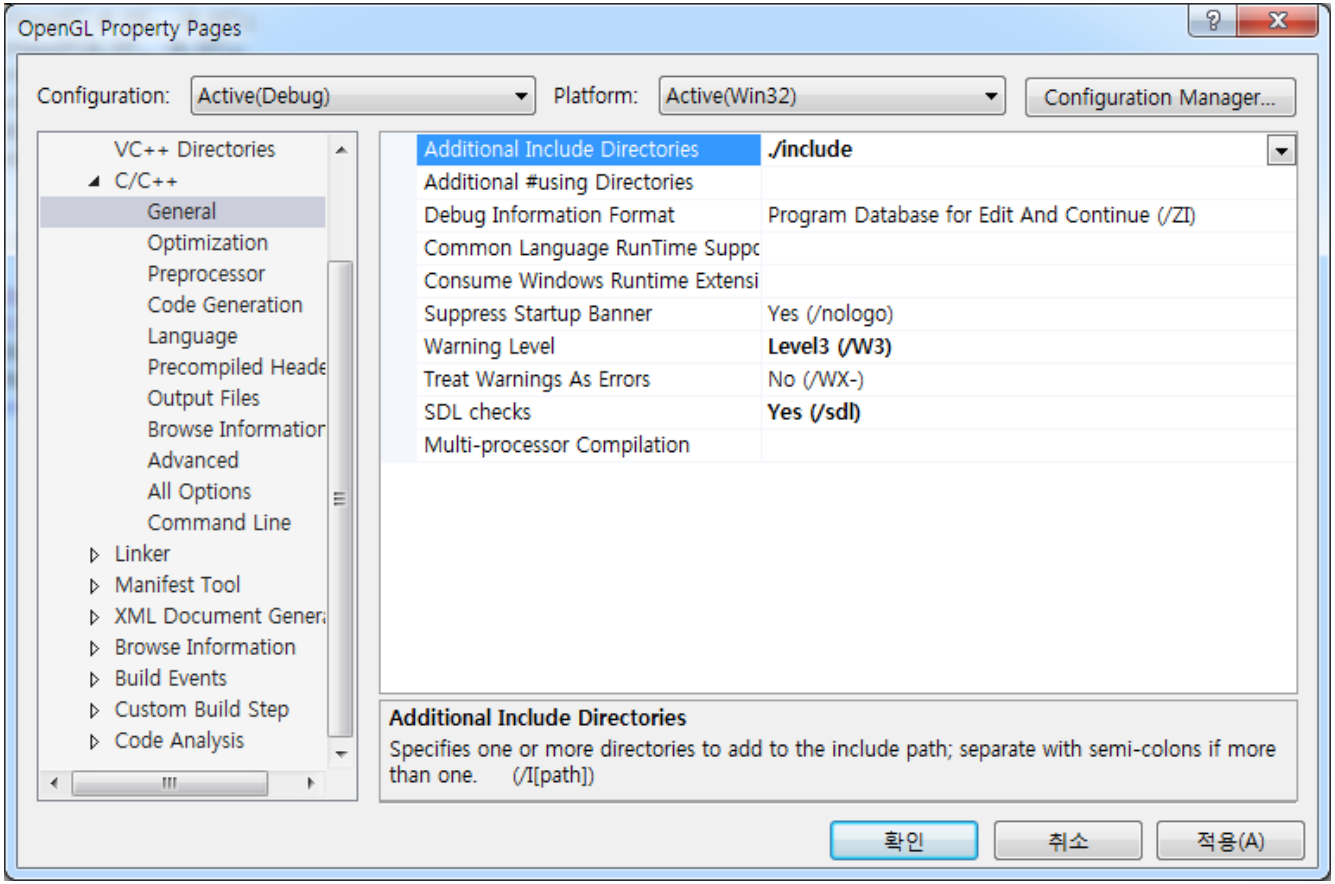
Project Setting

- Project properties



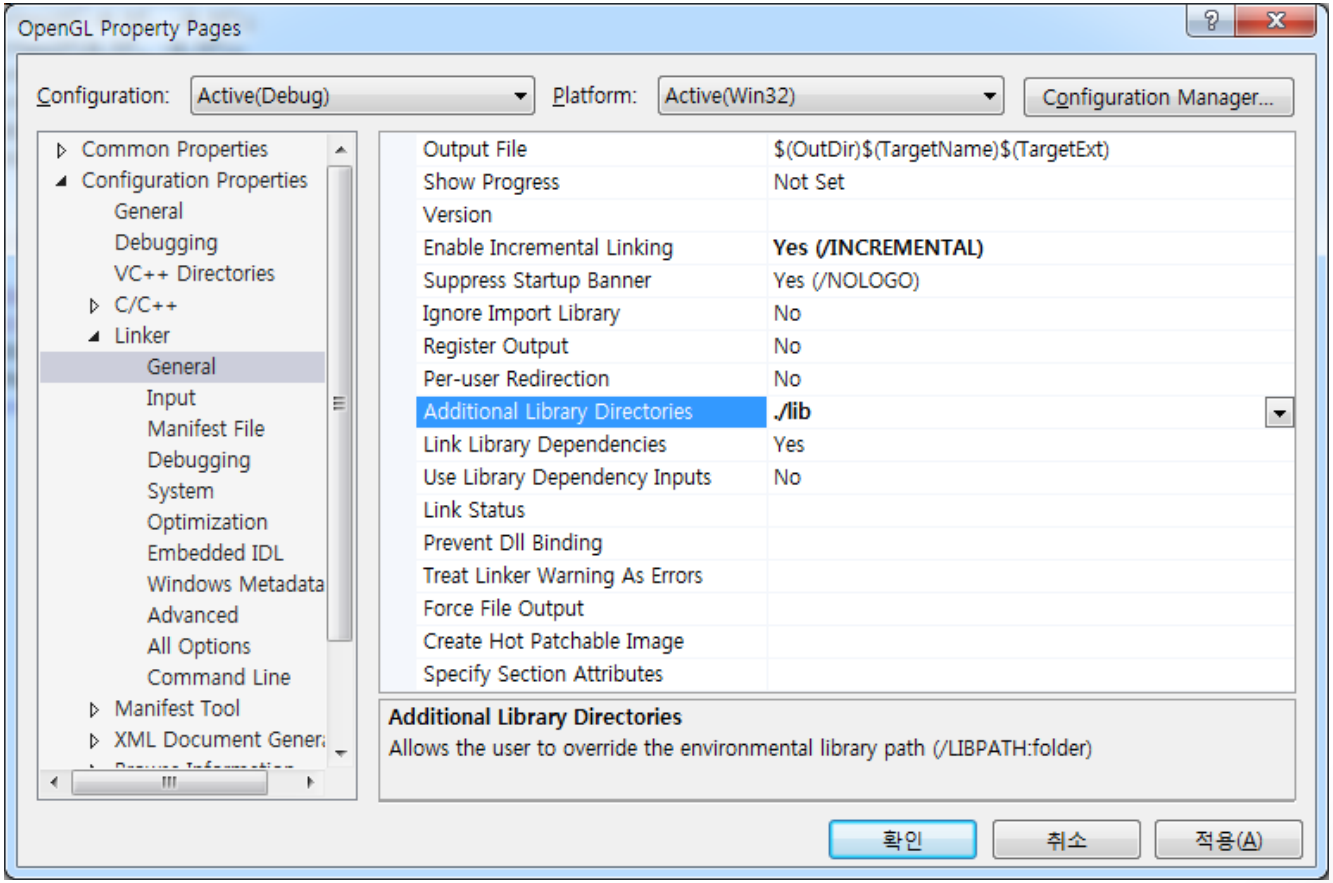
Project Setting

- Addition Include Directories



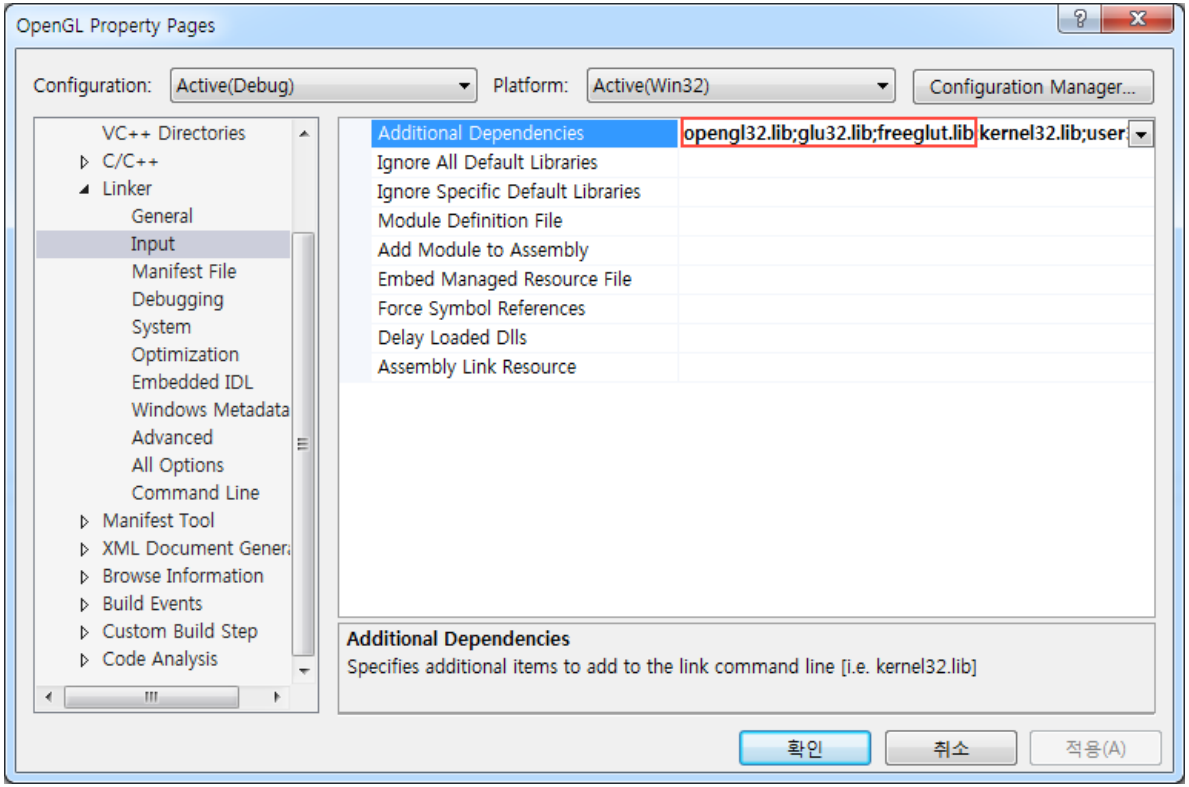
Project Setting

- Addition Library Directories



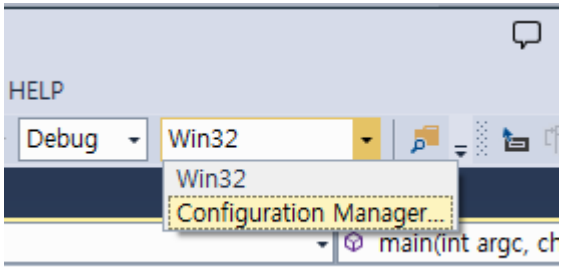
Project Setting

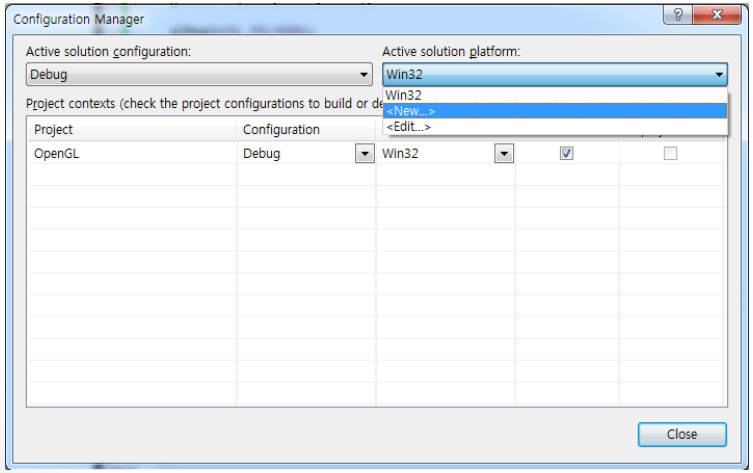
- Add three files to Addition Dependencies:
  - opengl32.lib, glu32.lib, freeglut.lib

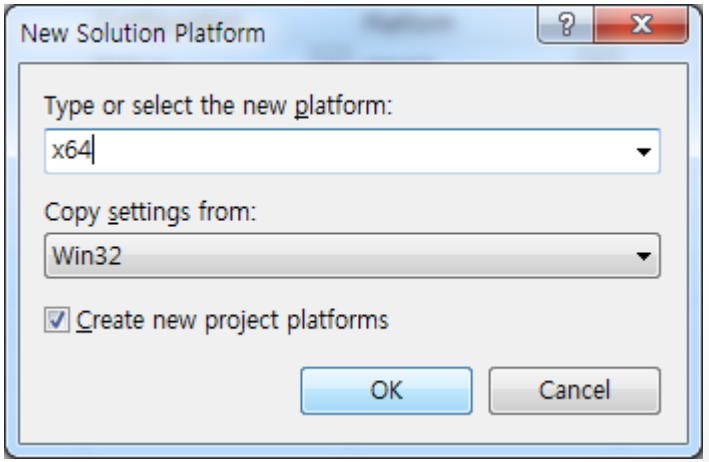


Project Setting

- (64-bit only)
  - Change configuration

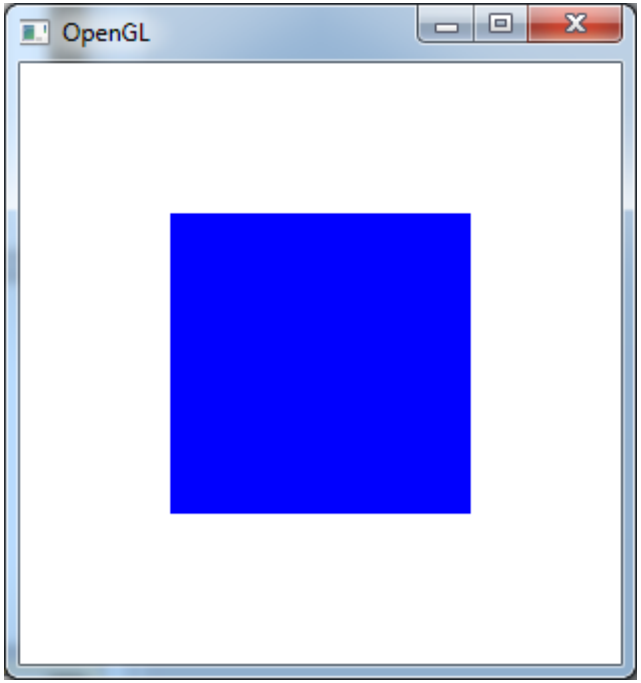






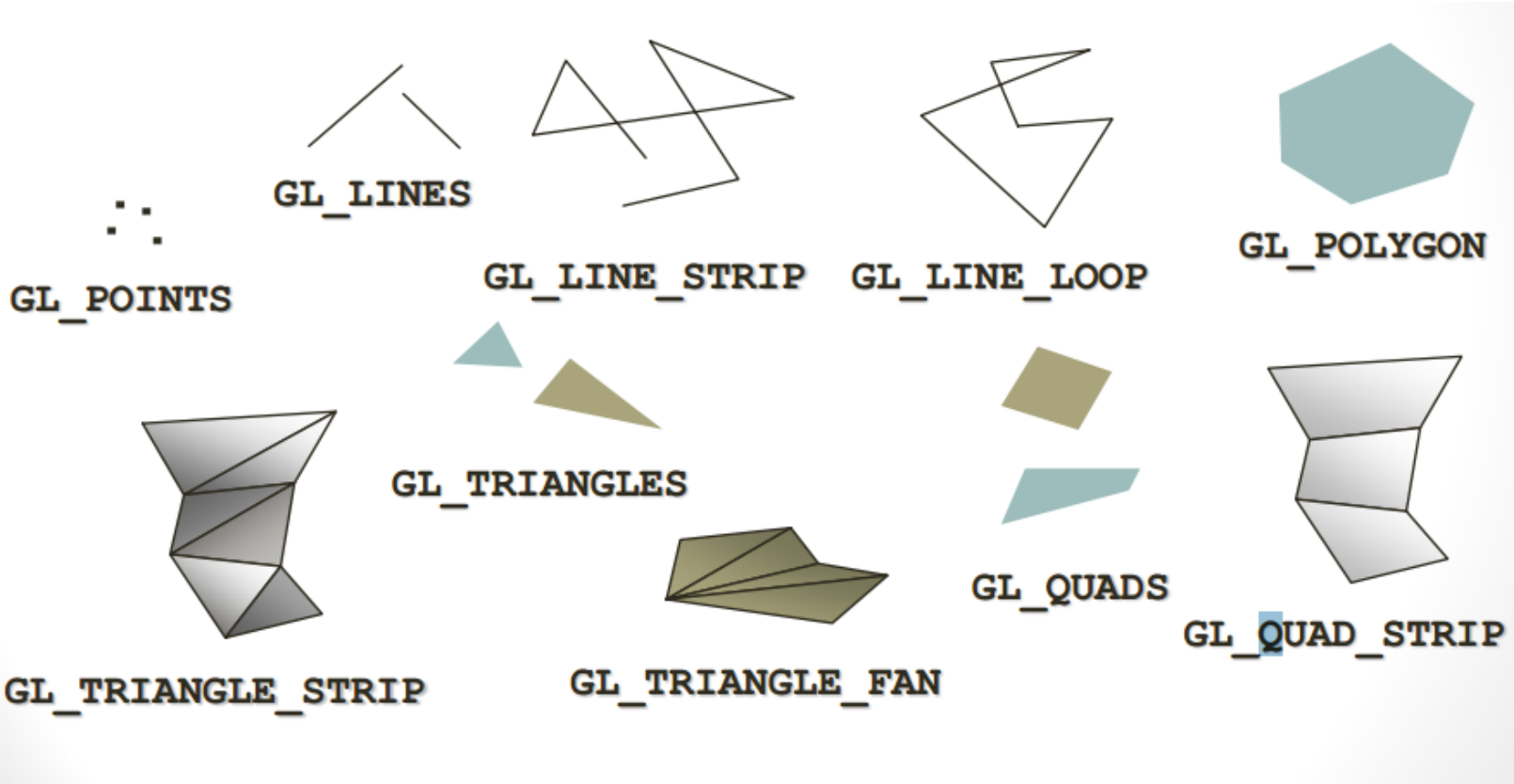
Code example

```
#include <gl/freeglut.h>

void display()
{
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0f, 0.0f, 1.0f);

    glBegin(GL_POLYGON);
    glVertex2f(-0.5f, -0.5f);
    glVertex2f(0.5f, -0.5f);
    glVertex2f(0.5f, 0.5f);
    glVertex2f(-0.5f, 0.5f);
    glEnd();
    glFinish();
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutCreateWindow("OpenGL");
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```

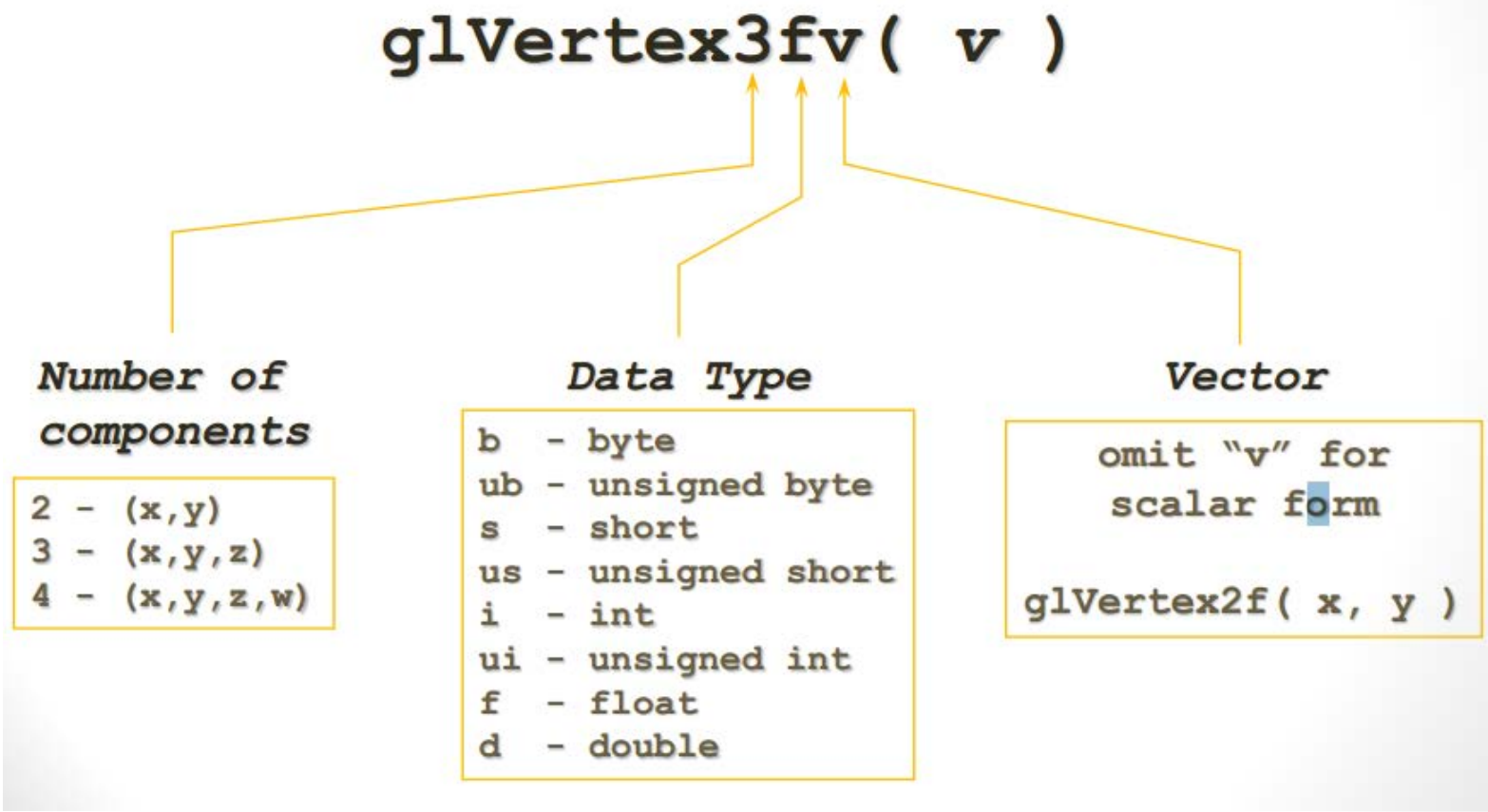


OpenGL Primitives

glBegin(GL_xxxx); ~~ glEnd();



Drawing Primitives

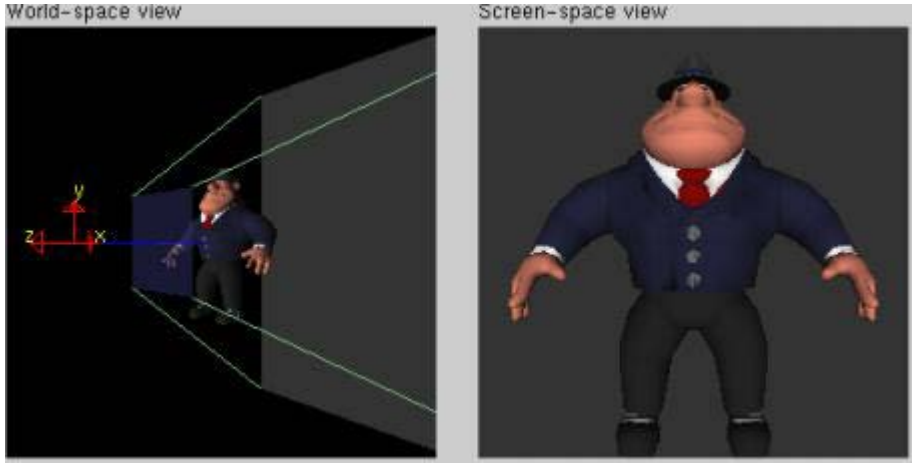


Modelview & Projection matrix

- Modelview Matrix
  - The relative transformation between object and camera
  - glTranslate()
  - glRotate()
  - glScale()
  - gluLookAt()
- Projection Matrix
  - Clipping volume (viewing frustum)
  - Projection to screen
  - glOrtho()
  - gluOrtho2D()
  - glFrustum()
  - gluPerspective()
- Common
  - glMatrixMode()
  - glLoadIdentity()
  - glPushMatrix()
  - glPopMatrix()

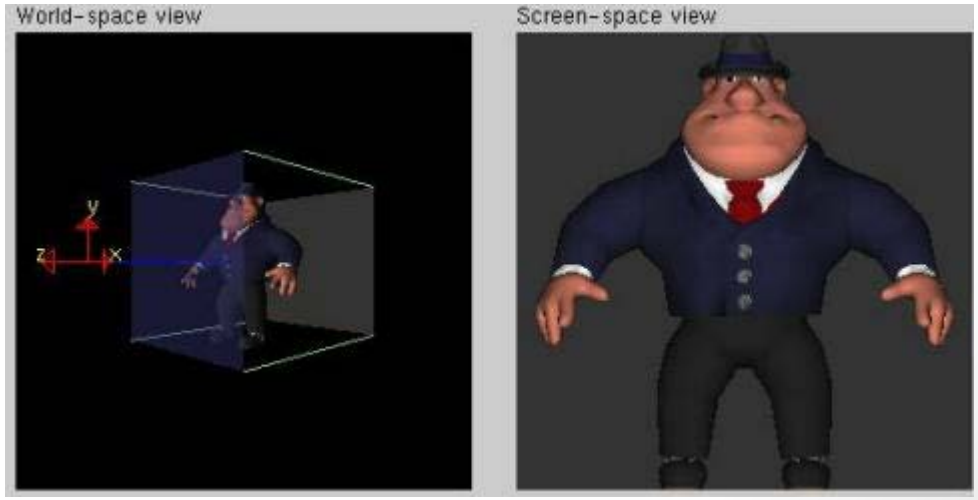
Projection transformation

- Projection transformation



gluPerspective

glOrtho

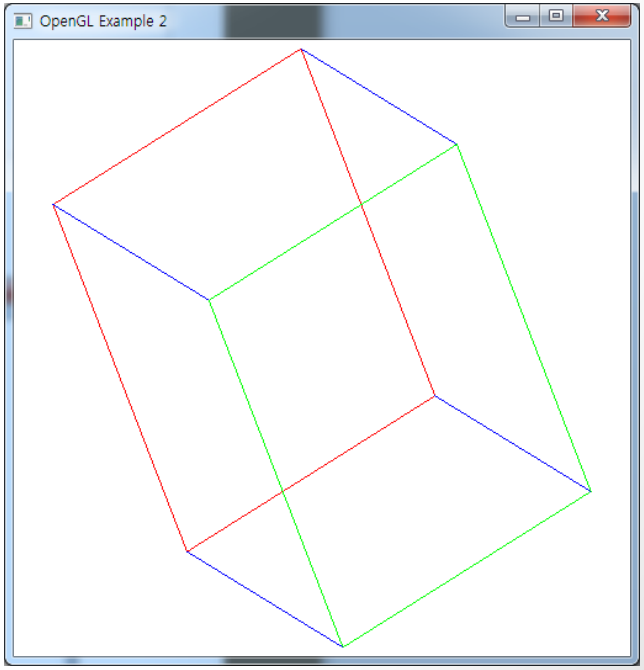


Code example(2)

```
#include <gl/freeglut.h>

void drawBox()
{
	glColor3f(1.0f, 0.0f, 0.0f);
	glBegin(GL_LINE_LOOP);
	glVertex3f(-1.0f, -1.0f, -1.0f);
	glVertex3f(1.0f, -1.0f, -1.0f);
	glVertex3f(1.0f, 1.0f, -1.0f);
	glVertex3f(-1.0f, 1.0f, -1.0f);
	glEnd();
	glColor3f(0.0f, 1.0f, 0.0f);
	glBegin(GL_LINE_LOOP);
	glVertex3f(-1.0f, -1.0f, 1.0f);
	glVertex3f(1.0f, -1.0f, 1.0f);
	glVertex3f(1.0f, 1.0f, 1.0f);
	glVertex3f(-1.0f, 1.0f, 1.0f);
	glEnd();
	glColor3f(0.0f, 0.0f, 1.0f);
	glBegin(GL_LINES);
	glVertex3f(-1.0f, -1.0f, -1.0f);
	glVertex3f(-1.0f, -1.0f, 1.0f);
	glVertex3f(1.0f, -1.0f, -1.0f);
	glVertex3f(1.0f, -1.0f, 1.0f);
	glVertex3f(1.0f, 1.0f, -1.0f);
	glVertex3f(1.0f, 1.0f, 1.0f);
	glVertex3f(-1.0f, 1.0f, -1.0f);
	glVertex3f(-1.0f, 1.0f, 1.0f);
	glEnd();
}
```

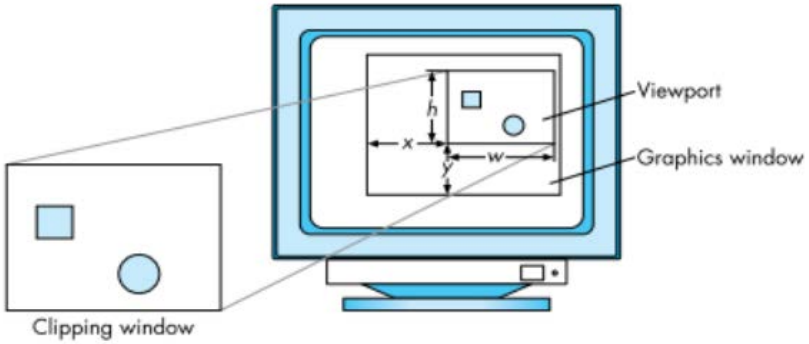
```
void display()
{
	glClearColor(1.0f, 1.0f, 1.0f, 1.0f);

	glClear(GL_COLOR_BUFFER_BIT);

	glColor3f(0.0f, 0.0f, 1.0f);

	glPushMatrix();
	glTranslatef(0.0f, 0.0f, -2.0f);
	glRotatef(45.0f, 1.0f, 1.0f, 1.0f);
	glScalef(0.5f, 0.7f, 0.5f);
	drawBox();
	glPopMatrix();

	glutSwapBuffers();
}

void reshape(int w, int h)
{
	glViewport(0, 0, w, h);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	glOrtho(-1.0f, 1.0f, -1.0f, 1.0f, 0.1f,
	50.0f);
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
}
```

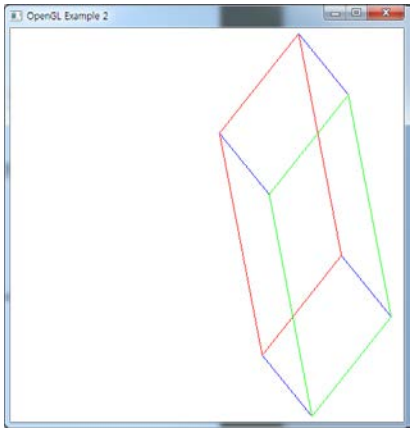
```
int main(int argc, char** argv)
{
	glutInit(&argc, argv);
	glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
	glutInitWindowSize(500, 500);
	glutInitWindowPosition(250, 250);
	glutCreateWindow("OpenGL Example 2");

	glutDisplayFunc(display);
	glutReshapeFunc(reshape);

	glutMainLoop();
	return 0;
}
```



Reshape funtion

- 처음 윈도우를 열 때, 위치를 옮길 때, 윈도우 크기를 조절할 때 호출되는 함수
- glViewport(GLint x, GLint y, GLsizei w,GLsizei h)





← glViewport(250, 0, w-250, h);

Callback functions

- glutDisplayFunc(void (*func)(void));
- glutReshapeFunc(void (*func)(int width, int height));
- glutKeyboardFunc(void (*func)(unsigned char key, int x, int y));
- glutMouseFunc(void (*func)(int button, int state, int x, int y));
- glutMotionFunc(void (*func)(int x, int y));
- glutIdleFunc(void (*func)(void));

Learning OpenGL

- The Red Book
- opengl.org
- nehe.gamedev.net
- lighthouse3d.com/opengl
- Google