# 2. 그래픽스프로그래밍의 소개

## 2.1 OpenGL 이란 ?

### 2.1.1 OpenGL의 정의

■ 2차원 또는 3차원 드로잉을 위한 표준 그래픽스 라이브러리
- 그래픽스 하드웨어에 대한 소프트웨어 인터페이스
- C나 C++과 같은 프로그래밍 언어는 아님
- 그래픽스 하드웨어에 잘 구현될 수 있음
- C 언어 기반 라이브러리
- 상태 기반 아키텍쳐
- 즉시 모드 (Immediate mode) 기반

■ 그래픽스 라이브러리 = 수백 개의 함수들
- Primitive functions : 무엇을 그릴 것인가 ?
  = 점, 삼각형, 사각형, 도형 모양
- Attribute functions : 어떻게 그릴 것인가 ?
  = 색상, 패턴 모양, 선분의 두께, 점의 크기 등
- Viewing functions : 카메라를 어떻게 다룰 것인가 ?
  = 카메라의 위치, 방향
- Transformation functions : 변환을 어떻게 처리할 것인가 ?
  = 위치변환, 크기변환, 회전변환, 투영변환
- Input functions : 입력을 어떻게 받아들일 것인가 ?
  = 마우스의 좌(우)측 버튼, 드래깅 여부, 키보드, 조이스틱
- Control functions : 윈도우를 어떻게 관리할 것인가 ?
  = 윈도우의 크기, 위치, 이벤트 처리 방법

### 2.1.2 OpenGL 라이브러리 종류

■ GL (Graphics Library): 저수준 렌더링 루틴으로 구성된 기본 그래픽스 라이브러리
- 표기법 : gl---- 로 시작, 예) glColor(1.0, 0.0, 0.0);

■ GLU (OpenGL Utility Library) : 고수준 함수들을 제공하는 OpenGL의 보완물
- 2D 이미지 확대/축소
- 구, 원통, 원반 등을 포함한 3D 객체들의 렌더링
- 하나의 이미지로 자동적으로 여러 단계의 mipmap 이미지 생성
- NURBS를 통한 곡면 지원
- 오목 다각형의 테셀레이션 지원
- 특수 용도의 변환과 행렬
- 표기법 : glu---- 로 시작, 예) gluSolidSphere(1.0);

■ GLUT(OpenGL Utility Toolkit) : 주요 플랫폼에서 사용할 수 있는 보조 라이브러리
- 윈도우 설정
- 윈도우 처리

- 이벤트 처리
- 표기법 : glut---로 시작, 예) glutMainLoop();

## 2.2 OpenGL 기본 구조

### 2.2.1 자료형

| OpenGL 자료형 | 내부표현방식 | C언어 자료형 | C 언어 접미사 |
|---|---|---|---|
| GLbyte | 8-bit integer | signed char | b |
| GLshort | 16-bit integer | short | s |
| GLint, GLsizei | 32-bit integer | long | i |
| GLfloat, GLclampf | 32-bit floating point | float | f |
| GLdouble, GLclampd | 64-bit floating point | double | d |
| GLubyte, GLboolean | 8-bit unsigned integer | unsigned char | ub |
| GLushort | 16-bit unsigned integer | unsigned short | us |
| GLuint, GLenum, GLbitfield | 32-bit unsigned integer | unsigned long | ui |

### 2.2.2 함수 명령 규칙

OpenGL 함수들은 모두 일정한 규칙을 가지고 있다. 그래서 그 함수가 어느 라이브러리 속하는지, 몇 개의 인자를 받고 각 인자의 타입이 무엇인지, 어떤 역할을 하는지 에 관한 내용을 함수이름에서 알 수 있도록 한다. 각 함수 이름은 다음의 네 부분으로 구성된다.

- ■ 라이브러리 종류를 나타내는 접두사
- ■ command를 나타내는 어근
- ■ 인자의 개수 (선택)
- ■ 인자의 타입 (선택)

예) glColor3f(1.0, 0.0, 0.0)

- ■ gl : GL 라이브러리 함수임을 의미
- ■ Color : 색상에 관련된 함수임을 의미 (첫 글자는 대문자로 시작한다.)
- ■ 3 : 3 개의 인자를 받겠다는 의미 (red, green, blue)
- ■ f : 받아지는 인자들은 float 형식의 자료들임을 의미

### 2.2.3 OpenGL 프로그램 예제를 통한 이해

```
#include <windows.h>
#include <gl/glut.h> // (or others, depending on the system in use)

void init (void) {
    glClearColor (1.0, 1.0, 1.0, 0.0) ; // Set display-window color to white.
    glMatrixMode (GL_PROJECTION); // Set projection parameters.
    gluOrtho2D (0.0, 200.0, 0.0, 150.0);
}

void RenderScene (void) {
```

```
    glClear (GL_COLOR_BUFFER_BIT); // Clear display window.
    glColor3f(1.0, 0.0, 0.0); // Set line segment color to red.
    glBegin (GL_LINES);
    glVertex2i (180, 15); // Specify line-segment geometry.
    glVertex2i (10, 145);
    glEnd();
    glFlush (); // Process all OpenGL routines as quickly as possible.
}

void main (int argc, char** argv) {
    glutInit (&argc, argv); // Initialize GLUT.
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB); // Set display mode.
    glutInitWindowPosition (50, 100); // Set top-left display-window position.
    glutInitWindowSize (400, 300); // Set display-window width and height.
    glutCreateWindow ("An Example OpenGL Program"); // Create display window.
    init(); // Execute initialization procedure.
    glutDisplayFunc (RenderScene); // Send graphics to display window.
    glutMainLoop(); // Display everything and wait.
}
```

■ 헤드 파일 부분

```
#include <windows.h>
#include <gl/glut.h>
```

"windows.h"라는 헤드 파일은 모든 응용 windows 애플리케이션에 필요하며 WIN32 함수 프로토타입의 대부분을 포함하고 있다. 그러나 이 헤더 파일을 꼭 추가할 필요는 없다. 그 이유는 WIN32 버전의 GLUT 라이브러리는 window.h를 glut.h에 포함하고 있기 때문이다. glut.h는 OpenGL과 GLU 라이브러리 함수를 정의하는 헤드 파일도 포함한다. 만약 GLUT를 사용하지 않는다면 두 개의 파일 gl.h 와 glu.h 도 추가하여야 한다.

■ main 부분

```
void main (int argc, char** argv) {
    glutInit (&argc, argv); // Initialize GLUT.
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB); // Set display mode.
    glutInitWindowPosition (50, 100); // Set top-left display-window position.
    glutInitWindowSize (400, 300); // Set display-window width and height.
    glutCreateWindow ("An Example OpenGL Program"); // Create display window.
    init(); // Execute initialization procedure.
    glutDisplayFunc (lineSegment); // Send graphics to display window.
    glutMainLoop(); // Display everything and wait.
}
```

콘솔 모드의 C나 C++ 프로그램은 항상 main 함수에서 시작한다. Windows 프로그램인 경우에는 WInMain이 그 역할을 담당한다고 볼 수 있다. 일반적으로 main 함수는 다음과 같은 네 개의 과정을

포함한다.

- ▶ 시스템 초기화
  - 시스템에 맞는 윈도우를 스크린에 띄우는 작업
  - OpenGL를 윈도우에 결합하는 작업
- ▶ OpenGL 초기화
  - OpenGL 좌표계 설정
  - 초기치 상태 값 설정
- ▶ 답신 함수 등록
  - 이벤트가 발생하면 실행될 내용을 하나의 함수로 설정
  - 이벤트와 답신함수 연결
- ▶ 사건 대기 및 실행
  - 사건이 발생하기를 기다렸다가 발생하면
  - 사건에 등록된 함수를 연결하여 실행

glutInit 함수는 GLUT를 초기화하는 함수이고 사용하려는 윈도우의 위치와 크기를 제어하기 위해서는 glutInitWindowPosition 함수와 glutInitWindowSize 함수를 사용한다. 여기에서 사용되는 좌표계는 윈도우 좌표계로서 모니터의 상단좌측이 (0,0)이 되며 오른쪽으로 갈수록 x값이 증가하며, 아래로 내려올수록 y값이 증가한다. 그림을 그릴 때 사용되는 OpenGL 좌표계는 좌측하단이 (0,0) 으로 여겨지는데, 오른쪽으로 갈수록 x값이 증가하며, 위로 갈수록 y값이 증가한다. 즉, 윈도우좌표계와 OpenGL 좌표계는 y축이 반대임을 명심하도록 하자.

GLUT 라이브러리가 윈도우를 생성할 때 어떤 종류의 디스플레이 모드를 사용할 것인지를 지정하는 함수가 glutInitDisplayMode 이다. 함수안의 인자는 플래그로서 싱글버퍼를 사용할 지 아니면 더블버퍼를 사용할 지를 정하는 플래그 (GLUT_SINGLE or GLUT_DOUBLE), 사용되는 색상 모드가 RGB인지 RGBA 인지를 결정하는 플래그 (GLUT_RGB or GLUT_RGBA), 3차원 장면을 표현할 때 보이지 않는 면을 그리지 않게 할 때 사용되는 깊이 정보를 저장하는 z-버퍼를 사용할 지 여부를 알리는 플래그 (GLUT_DEPTH) 들로 구성된다.

glutDisplayFunc(RenderScene) 함수는 이벤트가 발생하였을 때 인자 안에 있는 답신함수를 실행하게 하는 역할을 하는 함수의 일종이다. 이 함수는 윈도우가 처음 화면에 나타날 때, 윈도우의 크기를 보벙할 때, 윈도우가 다름 윈도우에 의해서 가려졌다가 다시 나타날 때 Display 이벤트를 발생하여 이 함수의 인자인 RenderScene 이라는 함수를 실행하게 된다. 이 예제에서는 빨간색의 선분을 그리게 되는 셈이다. 이런 종류의 함수들은 이벤트를 발생할 수 있는 입력장치와 연관되는 경우도 있다. glutDisplayFunc, glutKeyboardFunc, glutMouseFunc 그리고 glutReshapeFunc, glutPostRedisplay 등이 추가로 알아두어야 할 중요한 함수들이다.

glutMainLoop() 함수는 그래픽스 프로그램과 일반 프로그램과의 차이를 명확하게 보여 주는 함수이다. 일반프로그램은 내부 프로그램을 실행하고 결과를 얻은 다음에는 자동적으로 종료하지만 그래픽스 프로그램은 계속해서 결과를 보여주고 사용자가 직접 종료 명령을 내리기 전까지는 종료되어서는 안 된다. 왜냐하면 사용자의 입력을 계속 받을 준비가 되어야 하기 때문이다. 이를 가능하게 하는 것이 이 함수이다. 이 함수는 GLUT 이벤트 핸들링 루프를 시작한다. 이벤트 루프에서 모든 키보드, 마우스, 다시 그리기 등과 여러 가지 윈도우 메시지를 처리하게 된다. 이 함수는 프로그램을 종료할 때 까지는 리턴하지 않는다.

■ RenderScene 부분

```
void RenderScene (void) {
    glClear (GL_COLOR_BUFFER_BIT); // Clear display window.
    glColor3f(1.0, 0.0, 0.0); // Set line segment color to red.
    glBegin (GL_LINES);
    glVertex2i (180, 15); // Specify line-segment geometry.
    glVertex2i (10, 145);
    glEnd();
    glFlush (); // Process all OpenGL routines as quickly as possible.
}
```

이 부분에서는 프로그래머가 그리고자 하는 물체들을 만들어서 그리는 영역이다. main 부분에서는 그리고자 하는 목적을 위해 전체적인 환경을 구성하는 역할을 한다면 이 부분은 그래픽스 프로그램의 꽃인 scene을 형성하는 곳으로 나타내고자 하는 모든 물체를 자유롭게 다루는 곳이다. 그러므로 OpenGL의 대부분의 함수들이 이곳에서 적용되는 셈이다. 위 예제를 간단히 설명하면, 현재 그려져 있는 그림이 저장되어 있는 Color buffer를 지우는 작업을 하는 glClear 함수를 먼저 실행하고 빨간색을 가지고 (180.15)에서 출발하여 (10.145)에 끝나는 선분을 그리는 함수를 수행하게 되어 있다. 그래픽스 프로그램은 내부적으로 여러 개의 명령들을 순차적으로 처리하는 렌더링 파이프라인을 사용한다. OpenGl 명령들은 큐에 저장되어 있다가 OpenGL 드라이버가 요청된 여러 개의 작업을 한꺼번에 처리하게 된다. 이러한 구조를 사용하면 성능이 향상되며 복잡한 물체를 그릴 때에는 두드러진 효과를 얻을 수 있다. 그리는 작업이 가속되는 원리는 일정한 드로잉 명령들을 한꺼번에 하드웨어에 액세스하기 때문이다. glFlush 함수는 OpenGL에게 더 이상 드로잉 명령들을 받아들이지 말고 지금까지 요청된 명령들을 처리하게 하는 임무를 수행한다.

■ 초기화 부분

```
void init (void) {
    glClearColor (1.0, 1.0, 1.0, 0.0) ; // Set display-window color to white.
    glMatrixMode (GL_PROJECTION); // Set projection parameters.
    gluOrtho2D (0.0, 200.0, 0.0, 150.0);
}
```

이 부분에서는 OpenGL의 상태들을 초기화 한다. window의 바탕색을 어떤 색으로 할 지, 카메라의 위치, 카메라 방향, 3차원 물체를 2차원 화면으로 투영하는 방법, 관심 영역을 의미하는 관측공간의 정의, 마지막으로 카메라로 찍은 사진을 현상하고자 할 때 현상되는 영역을 의미하는 Viewport의 위치 및 크기를 정하는 과정을 이곳에서 처리한다.

## 2.3 OpenGL 설치

### 2.3.1 OpenGL Setup 과정

① Microsoft visual studio 6.0 이상 버전을 설치한다. 이미 설치되어 있다면 C:\..\Microsoft Visual studio\VC98\INCLUDE\GL 폴더에 **gl.h, glaux.h, glu.h** 파일들이 있는지 확인한다.

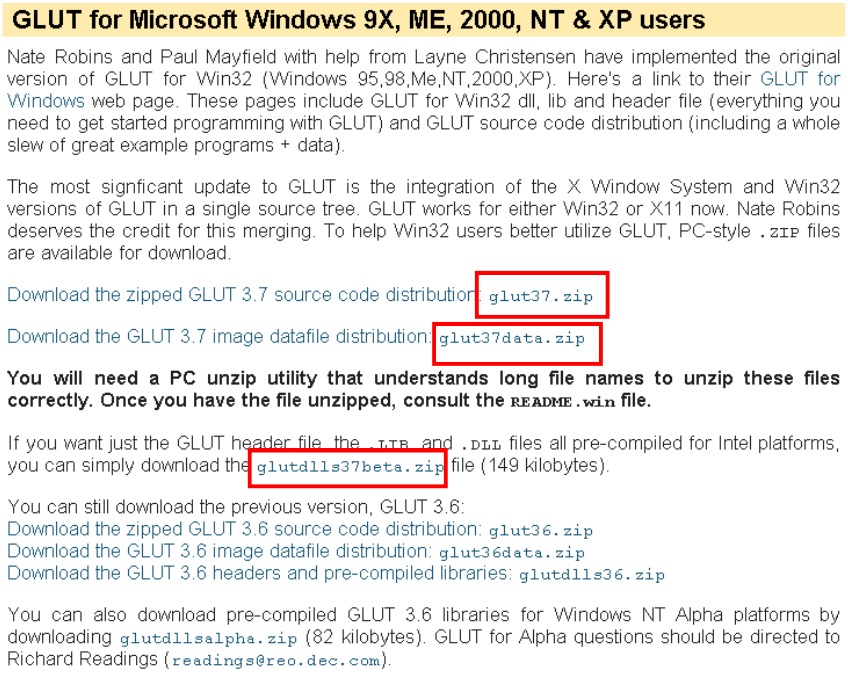
② glut 라이브러리를 http://www.opengl.org/resources/libraries/glut/glut_downloads.html 에서 glut 라이브러리를 다운로드 한다.

③ Microsoft visual studio를 실행하고 다운로드한 GLUT-3.7폴더의 경로를 지정해준다.

### 2.3.2 GLUT 라이브로리 다운로드 과정

① http://www.opengl.org/resources/libraries/glut/glut_downloads.html 를 방문하여 **Contents**목록에 나와있는 **GLUT for Microsoft Windows 95 & NT users**를 클릭하고 다음 그림과 같이 3개의 링크된 zip파일들을 다운로드 한다.

glut37.zip, glut37data.zip, glutdlls37beta.zip



**GLUT for Microsoft Windows 9X, ME, 2000, NT & XP users**

Nate Robins and Paul Mayfield with help from Layne Christensen have implemented the original version of GLUT for Win32 (Windows 95,98,Me,NT,2000,XP). Here's a link to their GLUT for Windows web page. These pages include GLUT for Win32 dll, lib and header file (everything you need to get started programming with GLUT) and GLUT source code distribution (including a whole slew of great example programs + data).

The most signficant update to GLUT is the integration of the X Window System and Win32 versions of GLUT in a single source tree. GLUT works for either Win32 or X11 now. Nate Robins deserves the credit for this merging. To help Win32 users better utilize GLUT, PC-style `.ZIP` files are available for download.

Download the zipped GLUT 3.7 source code distribution: `glut37.zip`

Download the GLUT 3.7 image datafile distribution: `glut37data.zip`

**You will need a PC unzip utility that understands long file names to unzip these files correctly. Once you have the file unzipped, consult the `README.win` file.**

If you want just the GLUT header file, the `.LIB` and `.DLL` files all pre-compiled for Intel platforms, you can simply download the `glutdlls37beta.zip` file (149 kilobytes).

You can still download the previous version, GLUT 3.6:
Download the zipped GLUT 3.6 source code distribution: `glut36.zip`
Download the GLUT 3.6 image datafile distribution: `glut36data.zip`
Download the GLUT 3.6 headers and pre-compiled libraries: `glutdlls36.zip`

You can also download pre-compiled GLUT 3.6 libraries for Windows NT Alpha platforms by downloading `glutdllsalpha.zip` (82 kilobytes). GLUT for Alpha questions should be directed to Richard Readings (`readings@reo.dec.com`).

② **glut37.zip, glut37data.zip** 압축 해제하고 glut-3.7폴더를 생성한다.(예 D:\glut-3.7 경로로 압축이 풀렸다고 가정)

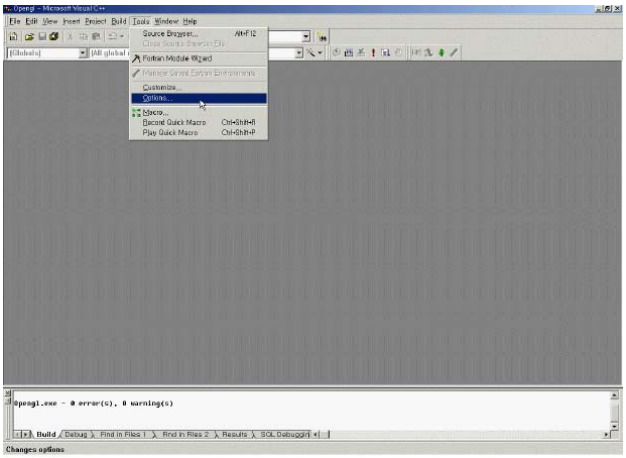
③ **glutdlls37beta.zip** 압축 해제하여 다음을 실행한다.

- ***.lib → D:\glut-3.7\lib\glut**
- **glut.h → VC98\Include\GL**
- **glut.dll → C:\WINNT\system32**
- **glut32.dll → C:\WINNT\system32**

④ Microsoft visual studio 실행하고 GLUT-3.7폴더의 경로를 지정해야 한다.

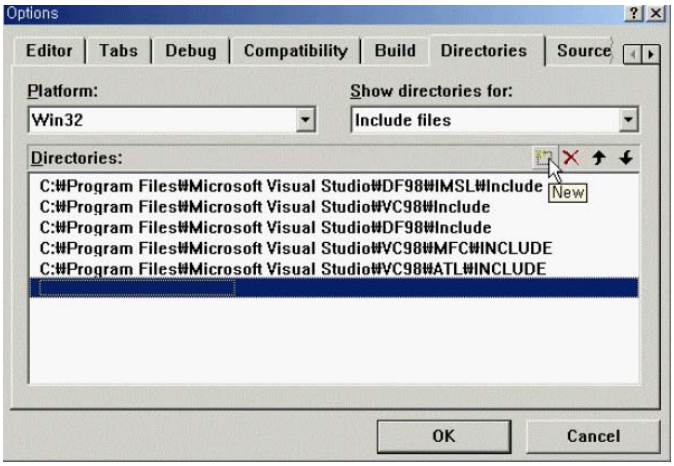
⑤ Microsoft visual studio의 Option 메뉴 설정

- 메뉴바에서 Tools --> Options..를 클릭

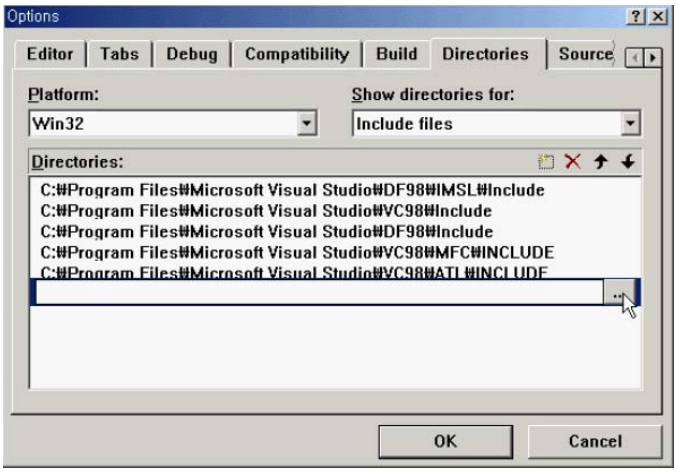


⑥ 헤드파일 경로 설정

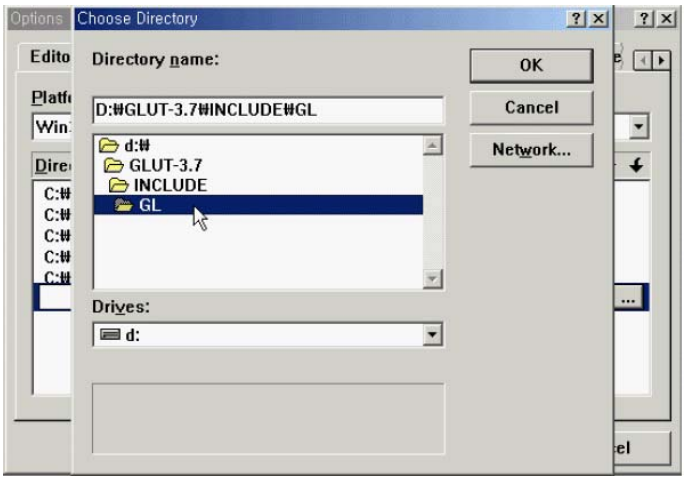
- Option 탭 메뉴에서 directories 선택
- show directories for 에서 include files 선택
- new버튼 클릭




- 헤더파일의 경로를 찾기 위해 ... 클릭

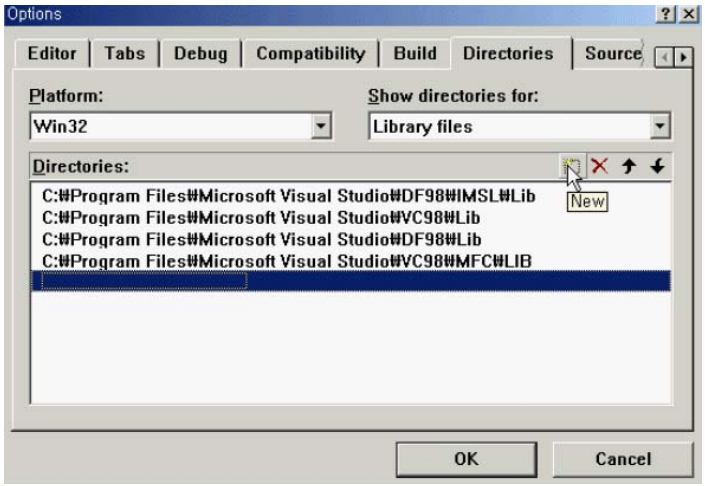


- 그림과 같이 헤더 파일들이 들어있는 폴더를 찾아 클릭

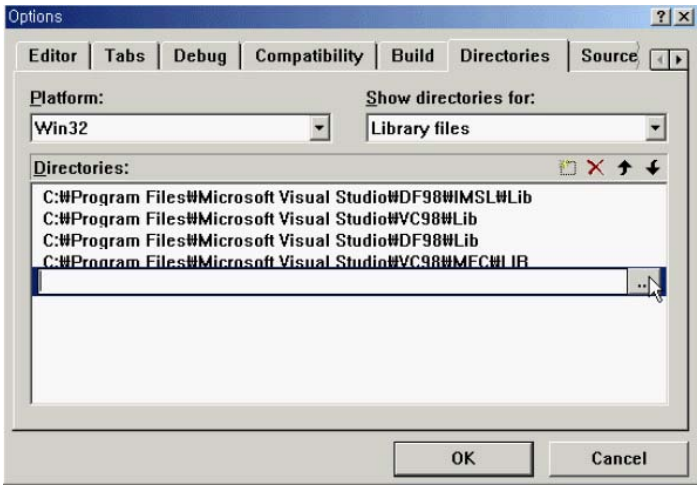



⑦ 라이브러리 경로 설정

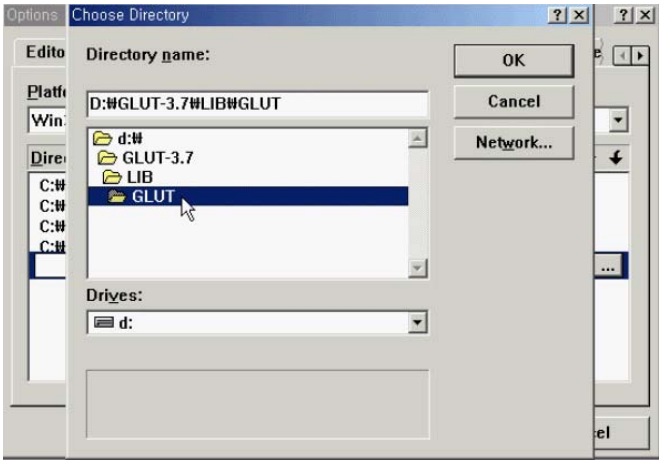
- Option 탭 메뉴에서 directories 선택
- Show directories for에서 Library files를 선택, new 클릭




- library 파일들의 경로를 찾기 위해 클릭



- 그림과 같이 glut32.lib 가 있는 폴더를 찾아 클릭

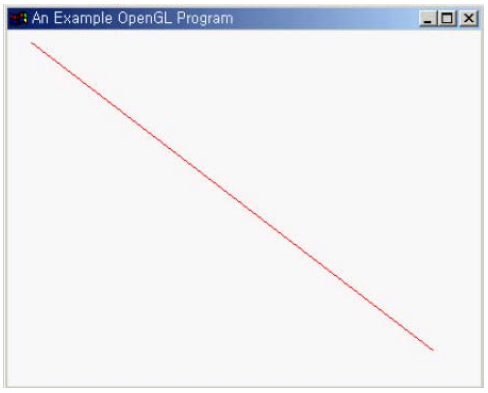



- ok버튼 클릭하면 셋업 과정 완료

### 2.3.4 Visual C++에서의 작업

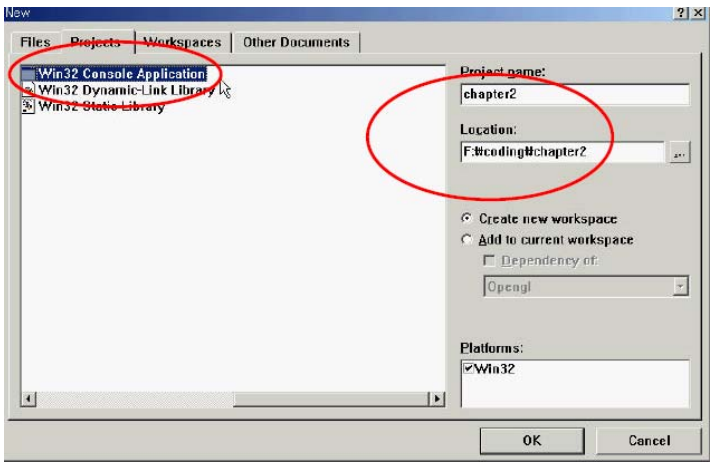
① 예제 프로그래밍 (The display window and line segment 만들기)

- 예제 프로그램을 실행한 결과

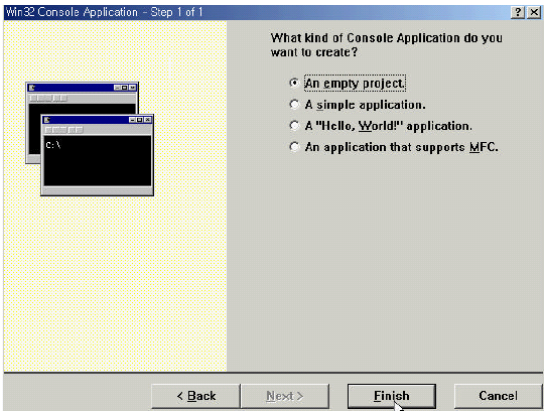



② Project 만들기

- [File] 메뉴의 new를 클릭하여 프로젝트 생성 마법사를 실행
- Project탭 win32 console Application을 선택한다.
- Project name과 프로젝트 파일들이 저장될 폴더를 지정한다.

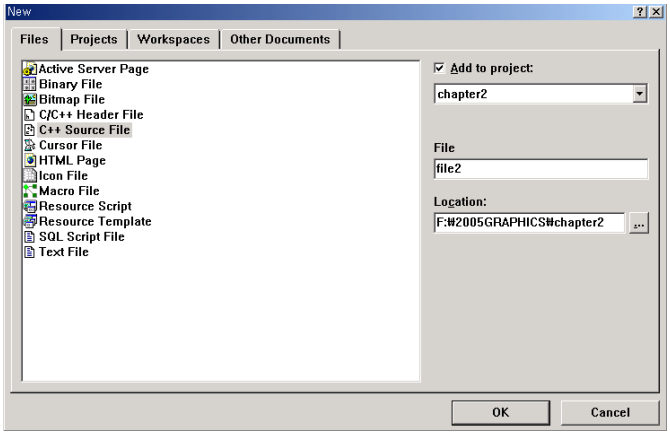


- An empty project 선택 Finish클릭



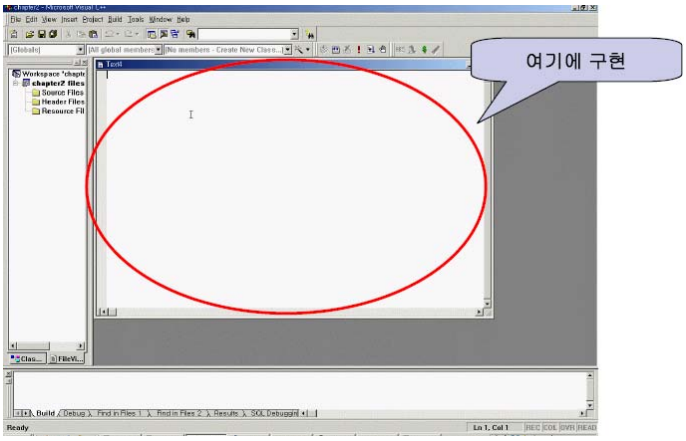

③ 새로운 파일 만들기

- [File] 메뉴의 new를 클릭
- Files탭 C++ source File을 선택한다.
- File name을 지정한다.




④ 프로그램 작성하기

- Text창에 구현을 한다.

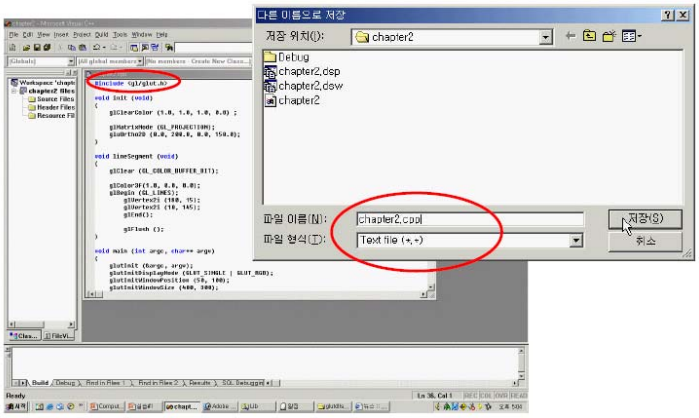


⑤ 예제 코드

```
#include <gl/glut.h>   // (or others, depending on the system in use)

void init (void)
{
      glClearColor (1.0, 1.0, 1.0, 0.0) ;    // Set display-window color to white.

      glMatrixMode (GL_PROJECTION);    // Set projection parameters.
      gluOrtho2D (0.0, 200.0, 0.0, 150.0);
}

void lineSegment (void)
{
      glClear (GL_COLOR_BUFFER_BIT);    // Clear display window.

      glColor3f(1.0, 0.0, 0.0);    // Set line segment color to red.
      glBegin (GL_LINES);
            glVertex2i (180, 15);    // Specify line-segment geometry.
            glVertex2i (10, 145);
            glEnd();

            glFlush ();    // Process all OpenGL routines as quickly as possible.
}
```

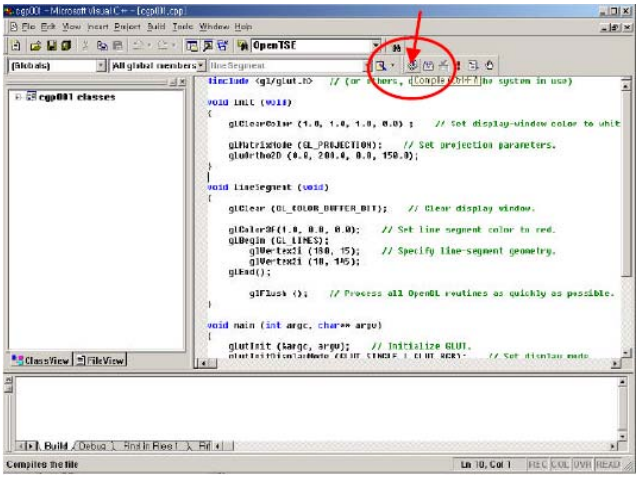
```
void main (int argc, char** argv)
{
       glutInit (&argc, argv);    // Initialize GLUT.
       glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);    // Set display mode.
      glutInitWindowPosition (50, 100);    // Set top-left display-window position.
      glutInitWindowSize (400, 300);    // Set display-window width and height.
      glutCreateWindow ("An Example OpenGL Program");    // Create display window.

      init();    // Execute initialization procedure.
      glutDisplayFunc (lineSegment);    // Send graphics to display window.
      glutMainLoop();    // Display everything and wait.
}
```

⑥ 파일 저장하기



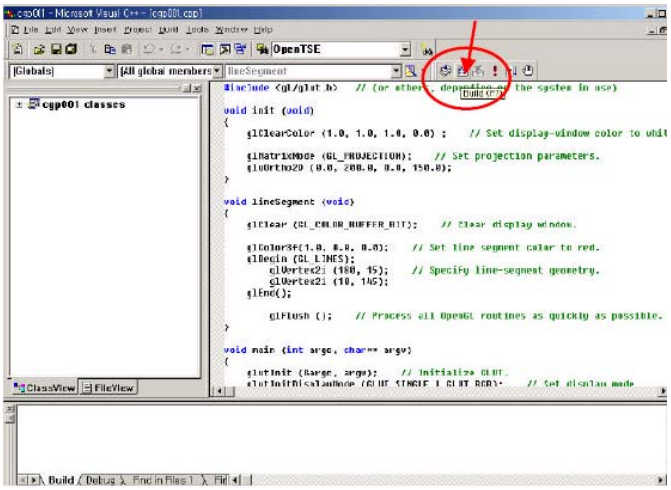

⑦ 컴파일하기

- Ctrl + F7키를 눌러 컴파일




⑧ 빌드하기

- F7를 눌러 빌드



⑨ 실행하기

- Ctrl + F5를 눌러 실행