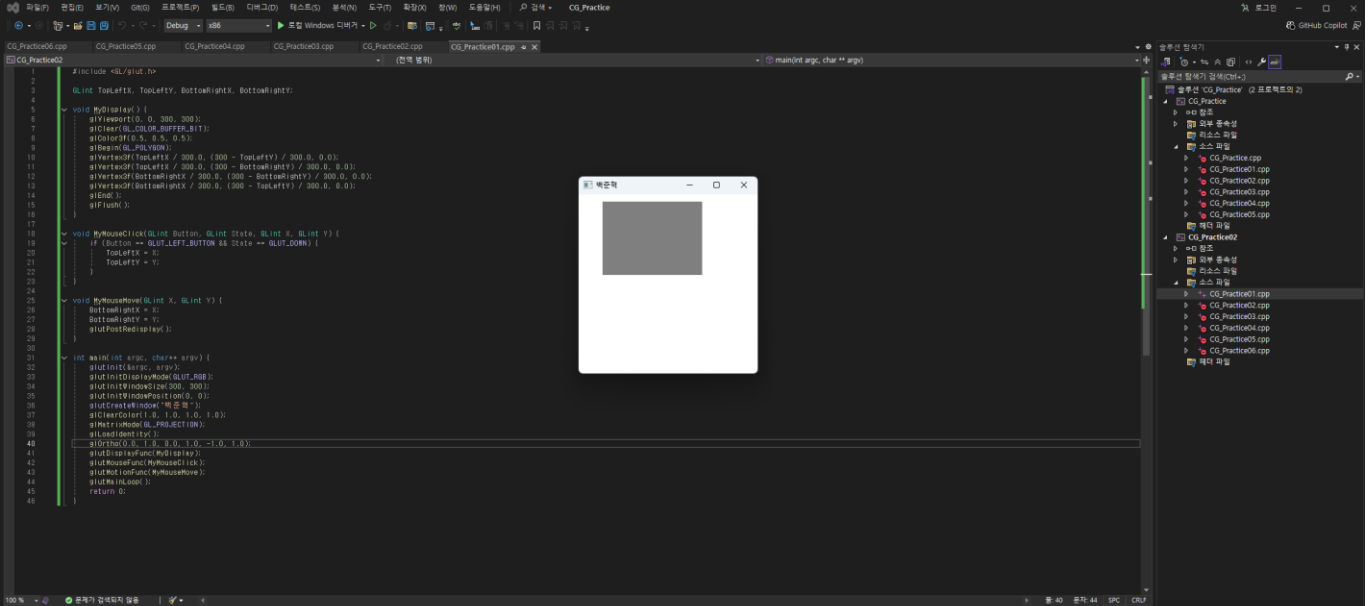
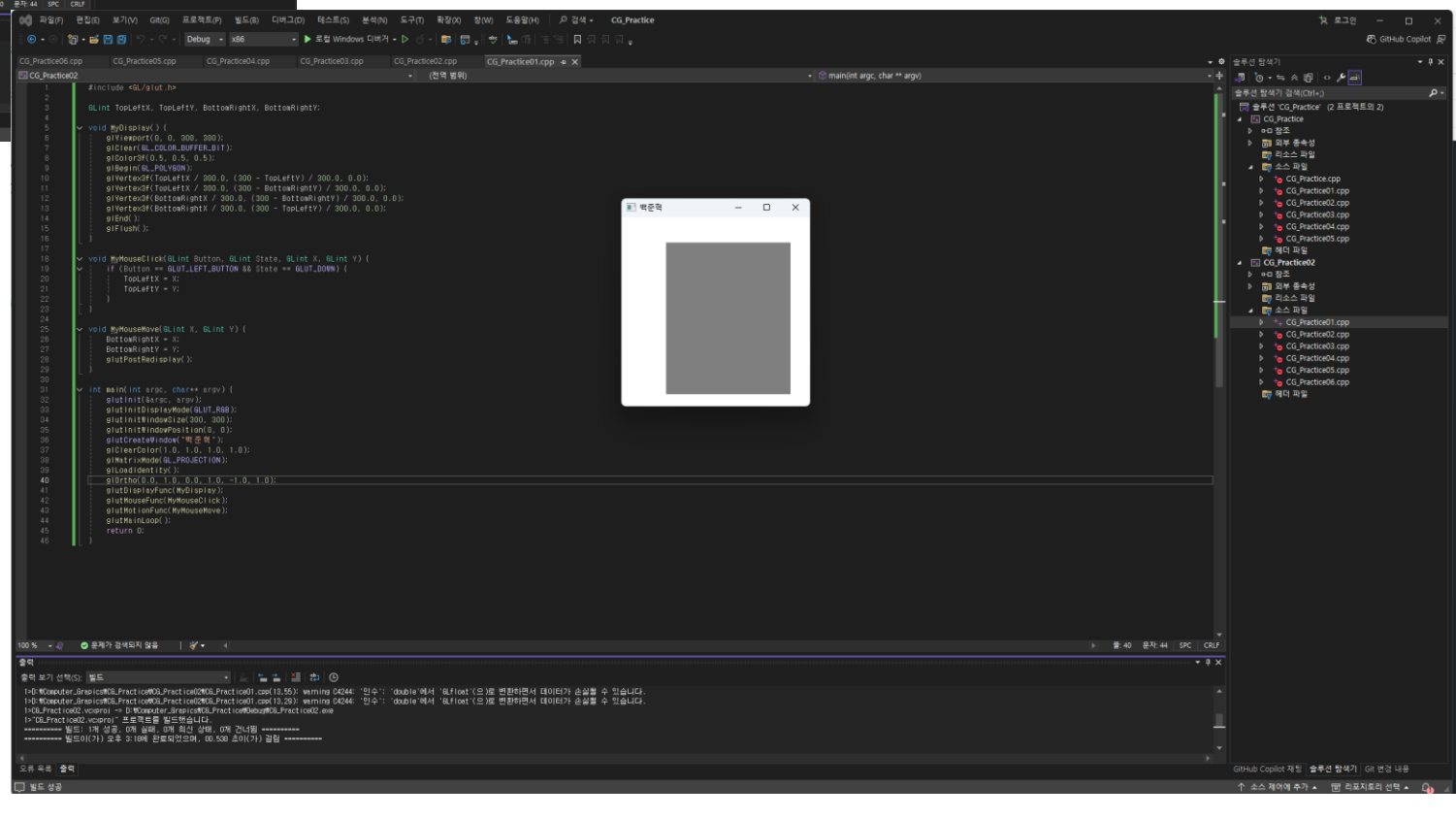
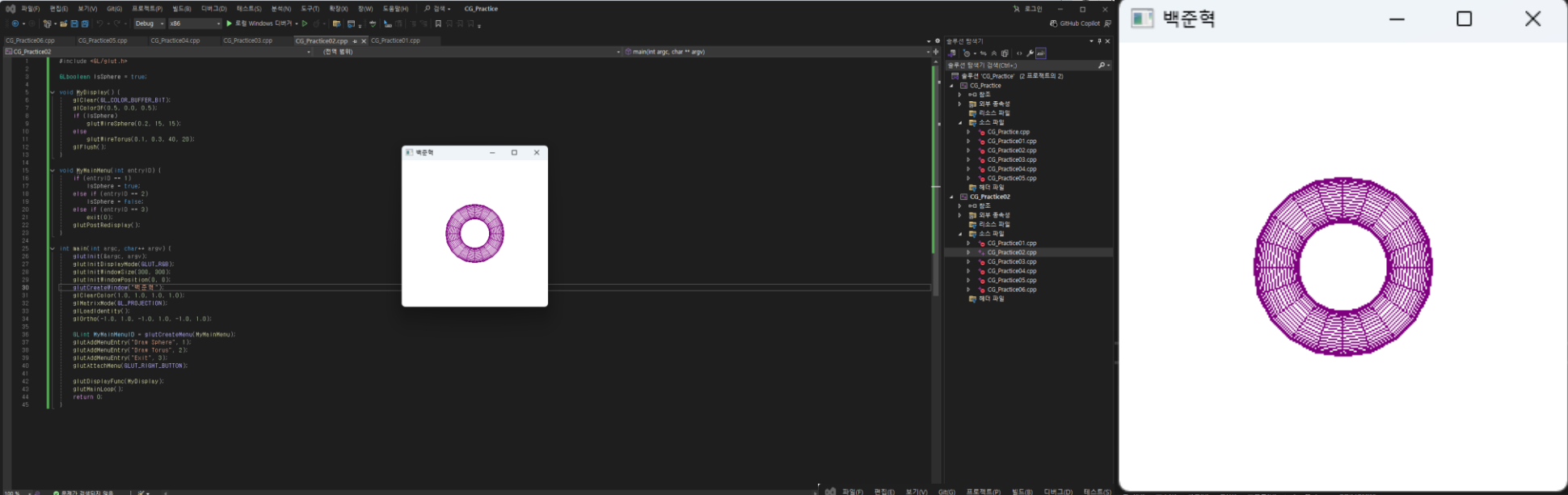


```
#include <GL/glut.h>

GLint TopLeftX, TopLeftY, BottomRightX, BottomRightY;

void MyDisplay() {
    glViewport(0, 0, 300, 300);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.5, 0.5, 0.5);
    glBegin(GL_POLYGON);
    glVertex3f(TopLeftX / 300.0, (300 - TopLeftY) / 300.0, 0.0);
    glVertex3f(TopLeftX / 300.0, (300 - BottomRightY) / 300.0, 0.0);
    glVertex3f(BottomRightX / 300.0, (300 - BottomRightY) / 300.0, 0.0);
    glVertex3f(BottomRightX / 300.0, (300 - TopLeftY) / 300.0, 0.0);
    glEnd();
    glFlush();
}

void MyMouseClick(GLint Button, GLint State, GLint X, GLint Y) {
    if (Button == GLUT_LEFT_BUTTON && State == GLUT_DOWN) {
        TopLeftX = X;
        TopLeftY = Y;
    }
}

void MyMouseMove(GLint X, GLint Y) {
    BottomRightX = X;
    BottomRightY = Y;
    glutPostRedisplay();
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGB);
    glutInitWindowSize(300, 300);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("백준책");
    glClearColor(1.0, 1.0, 1.0, 1.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
    glutDisplayFunc(MyDisplay);
    glutMouseFunc(MyMouseClick);
    glutMotionFunc(MyMouseMove);
    glutMainLoop();
    return 0;
}
```

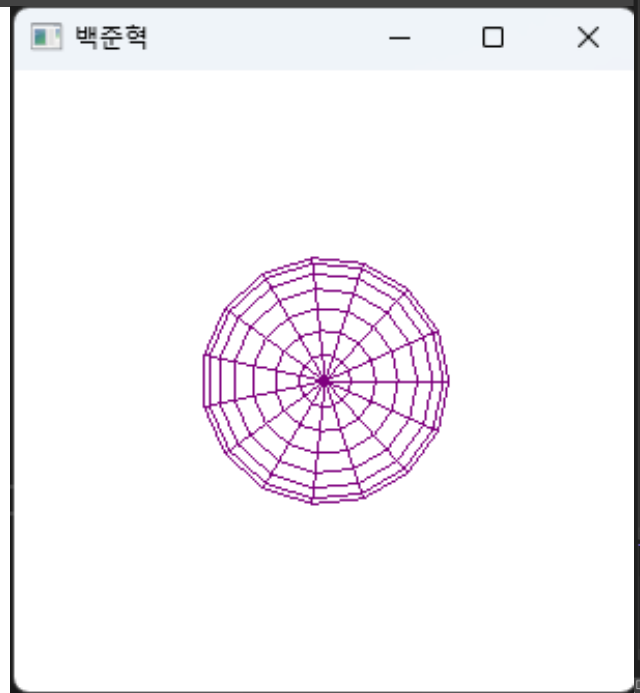


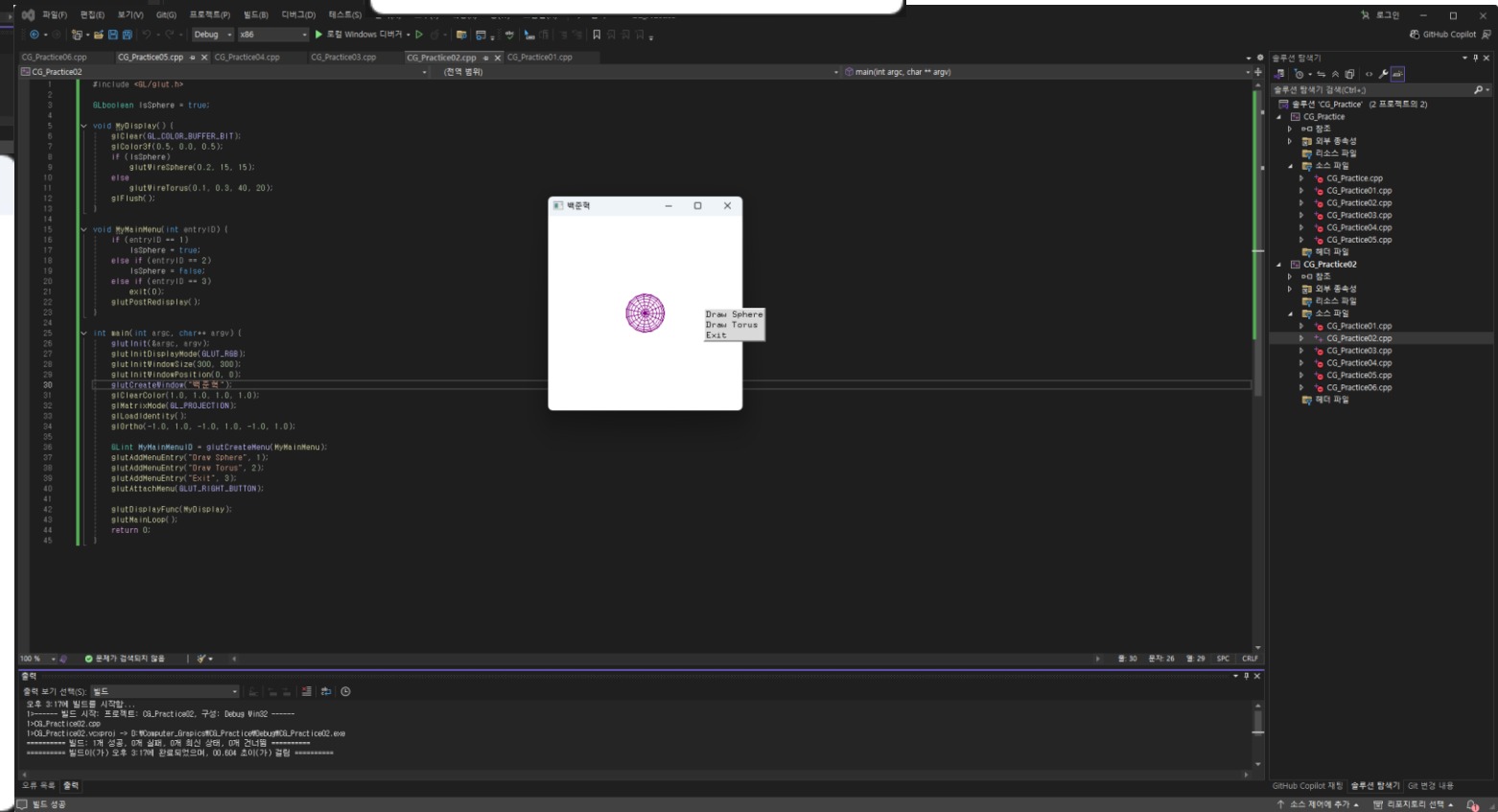
```
1>D:\Computer_Grapics\OGL_Practice02\OGL_Practice01.cpp(13,55): warning C4244: '인수': 'double'에서 'GLfloat'(으)로 변환하면서 데이터가 손실될 수 있습니다.
1>D:\Computer_Grapics\OGL_Practice02\OGL_Practice01.cpp(13,29): warning C4244: '인수': 'double'에서 'GLfloat'(으)로 변환하면서 데이터가 손실될 수 있습니다.
1>OGL_Practice02.vcxproj -> D:\Computer_Grapics\OGL_Practice02\Debug\OGL_Practice02.exe
1>"OGL_Practice02.vcxproj" 프로젝트를 빌드했습니다.
========== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뜀 ==========
========== 빌드이(가) 오후 3:18에 완료되었으며, 00.538 초이(가) 걸림 ==========
```

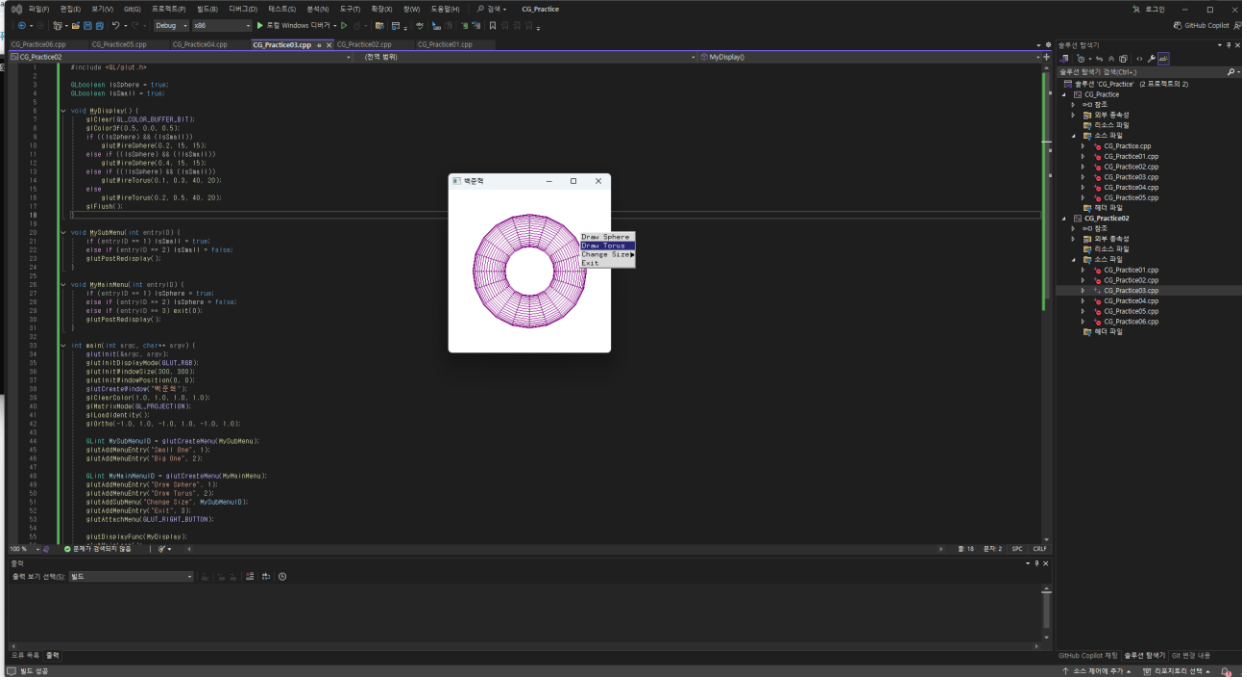


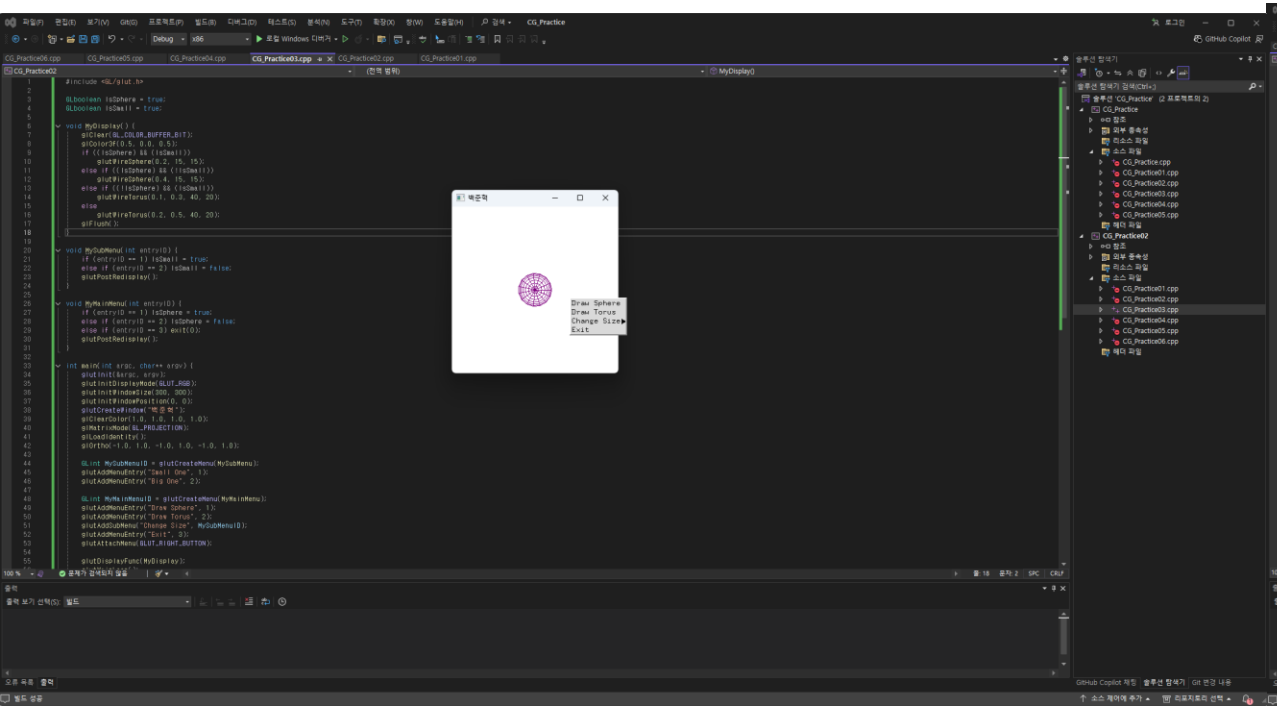

#include <GL/glut.h>
GLboolean isSphere = true;
void MyDisplay() {
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(0.5, 0.0, 0.5);
if (isSphere)
glutWireSphere(0.2, 15, 15);
else
glutWireTorus(0.1, 0.3, 40, 20);
glFlush();
}
void MyMainMenu(int entryID) {
if (entryID == 1)
isSphere = true;
else if (entryID == 2)
isSphere = false;
else if (entryID == 3)
exit(0);
glutPostRedisplay();
}
int main(int argc, char** argv) {
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_RGB);
glutInitWindowSize(300, 300);
glutInitWindowPosition(0, 0);
glutCreateWindow("백준혁");
glClearColor(1.0, 1.0, 1.0, 1.0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);
GLint MyMainMenuID = glutCreateMenu(MyMainMenu);
glutAddMenuEntry("Draw Sphere", 1);
glutAddMenuEntry("Draw Torus", 2);
glutAddMenuEntry("Exit", 3);
glutAttachMenu(GLUT_RIGHT_BUTTON);
glutDisplayFunc(MyDisplay);
glutMainLoop();
return 0;
}
백준혁

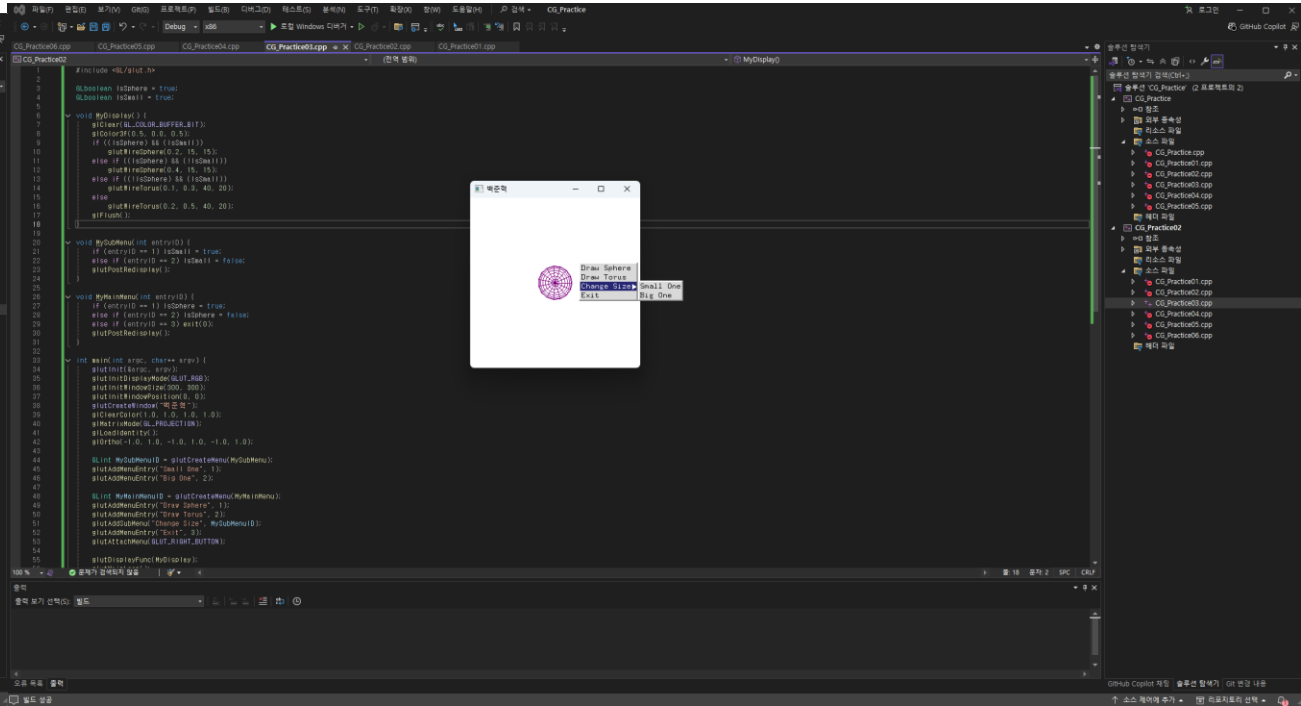
백준혁

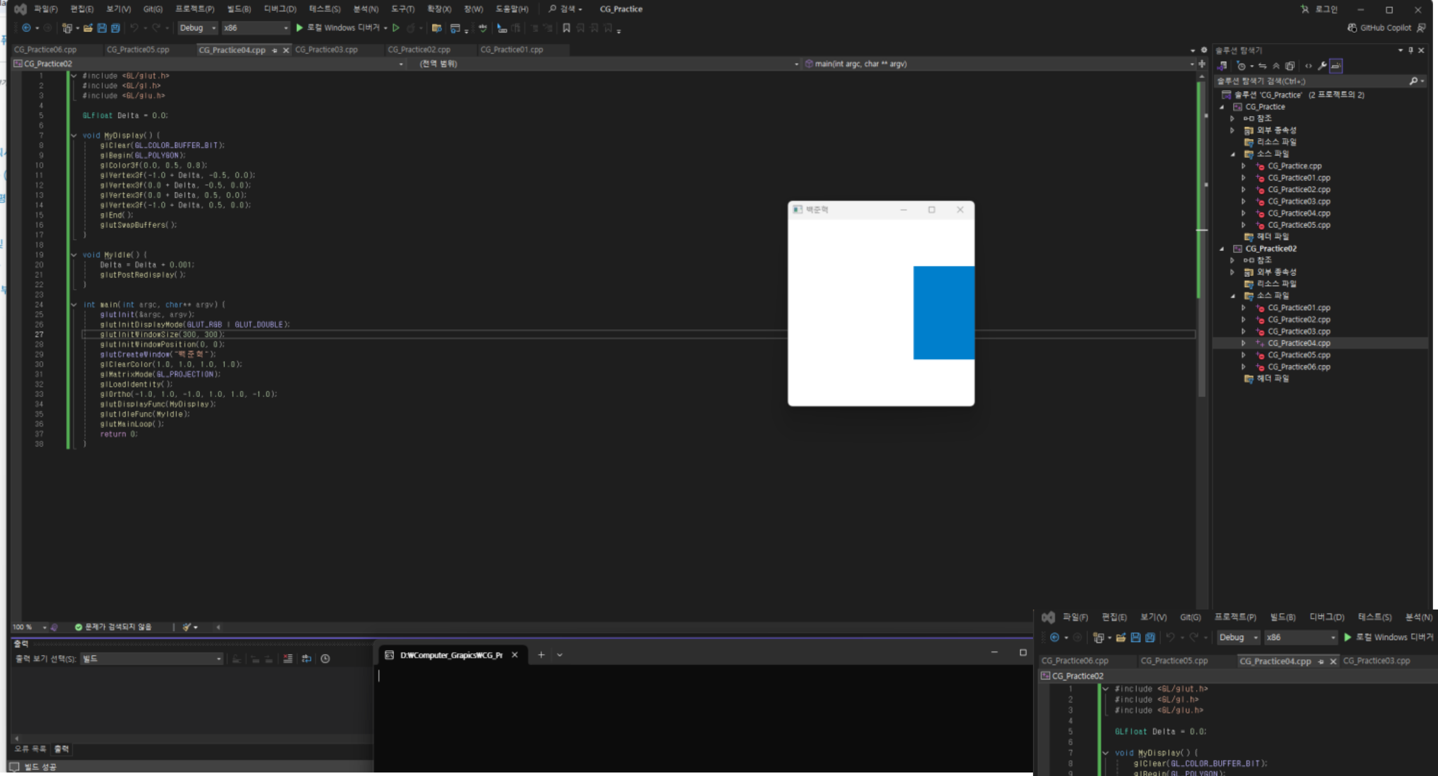

백준혁


백준혁
Draw Sphere
Draw Torus
Exit

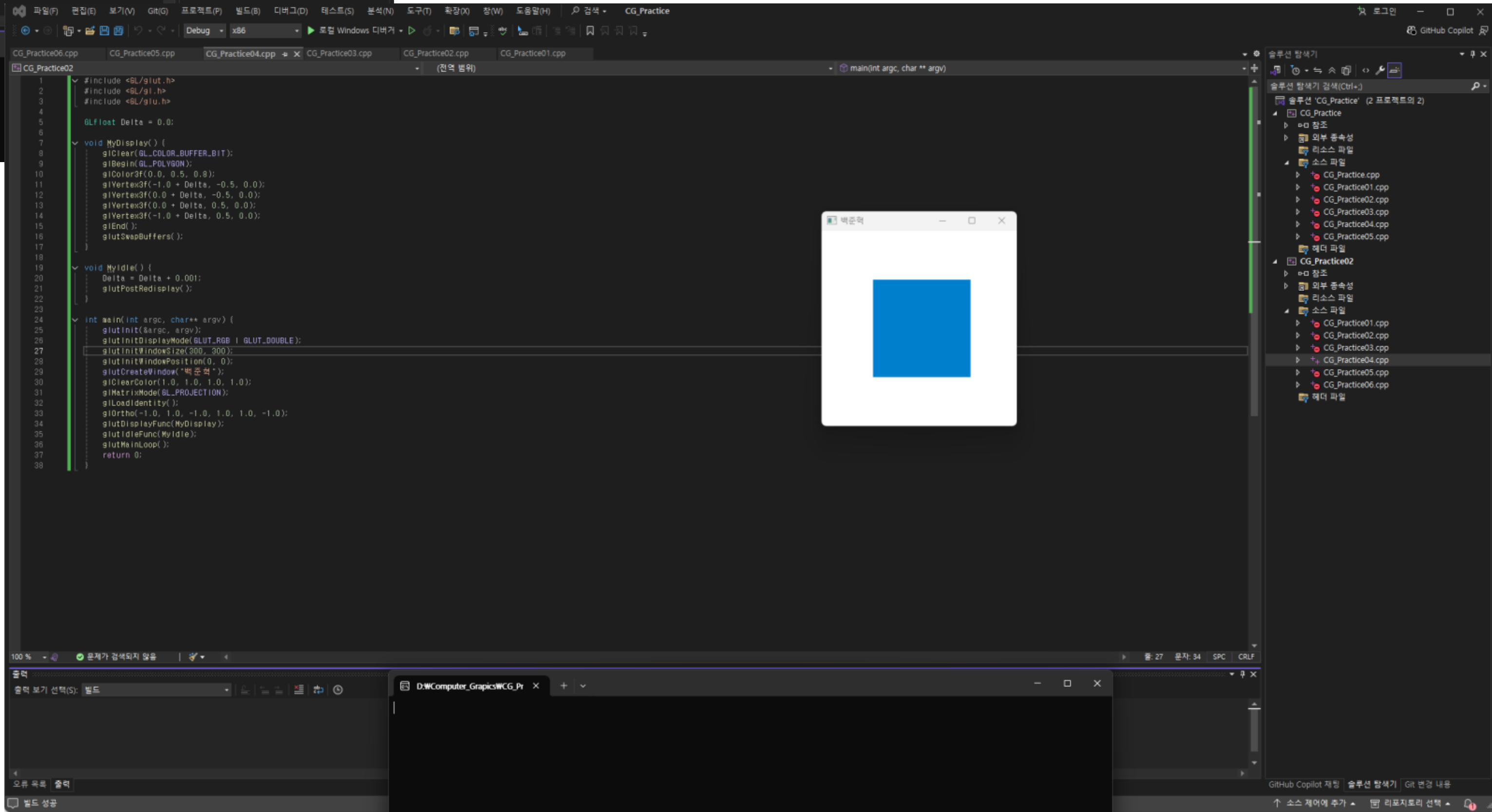

박준적
Draw Sphere
Draw Torus
Change Size
Exit


박준적
Draw Sphere
Draw Torus
Change Size
Exit

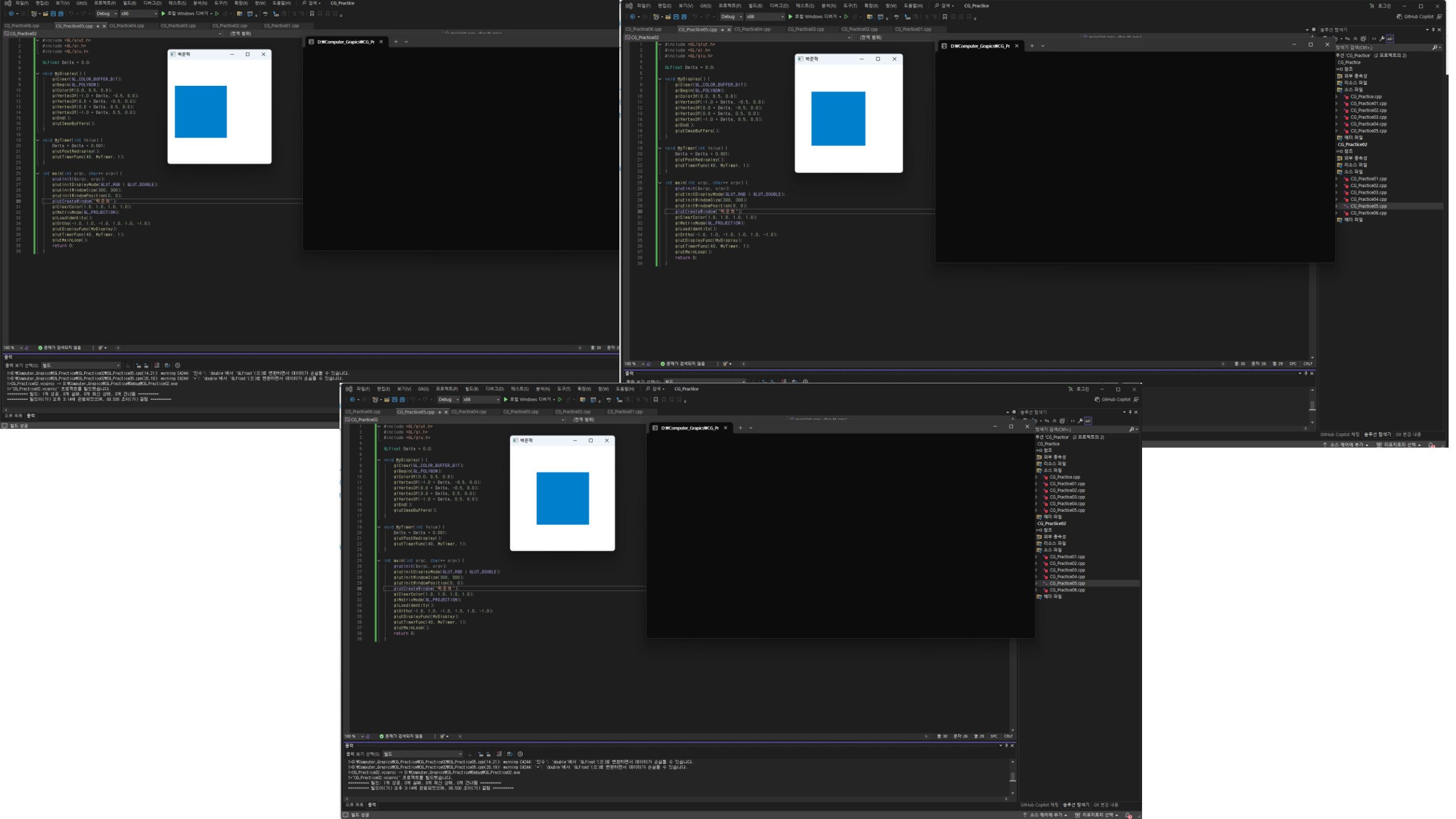

박준적
Draw Sphere
Draw Torus
Change Size
Exit
Small One
Big One

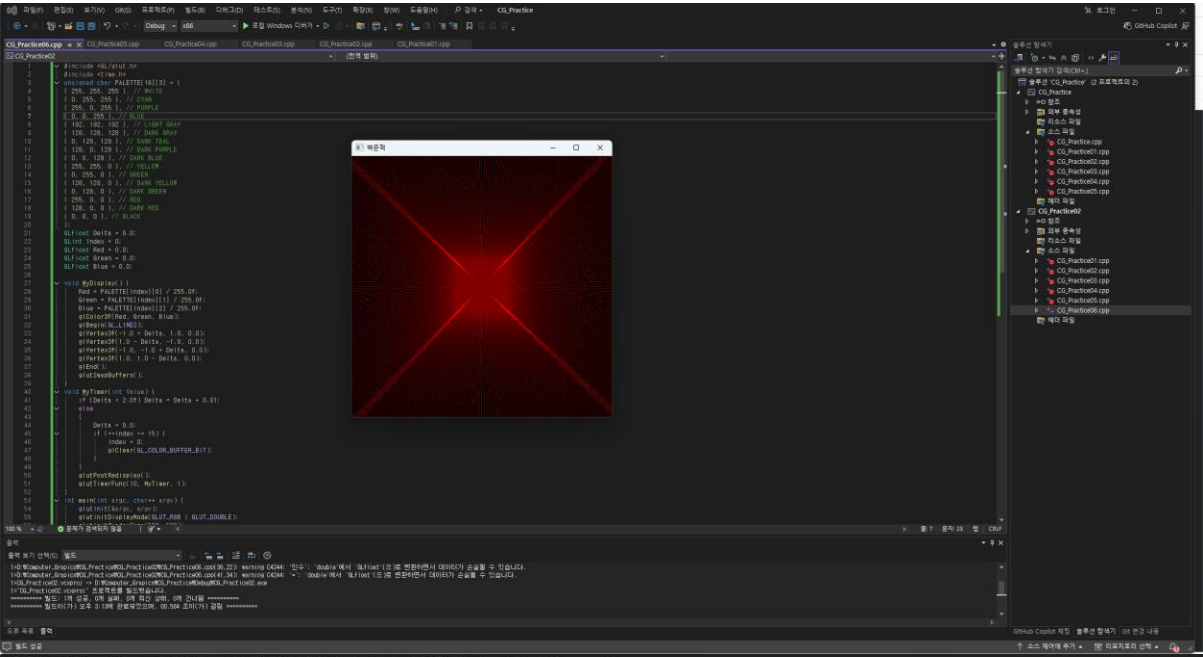


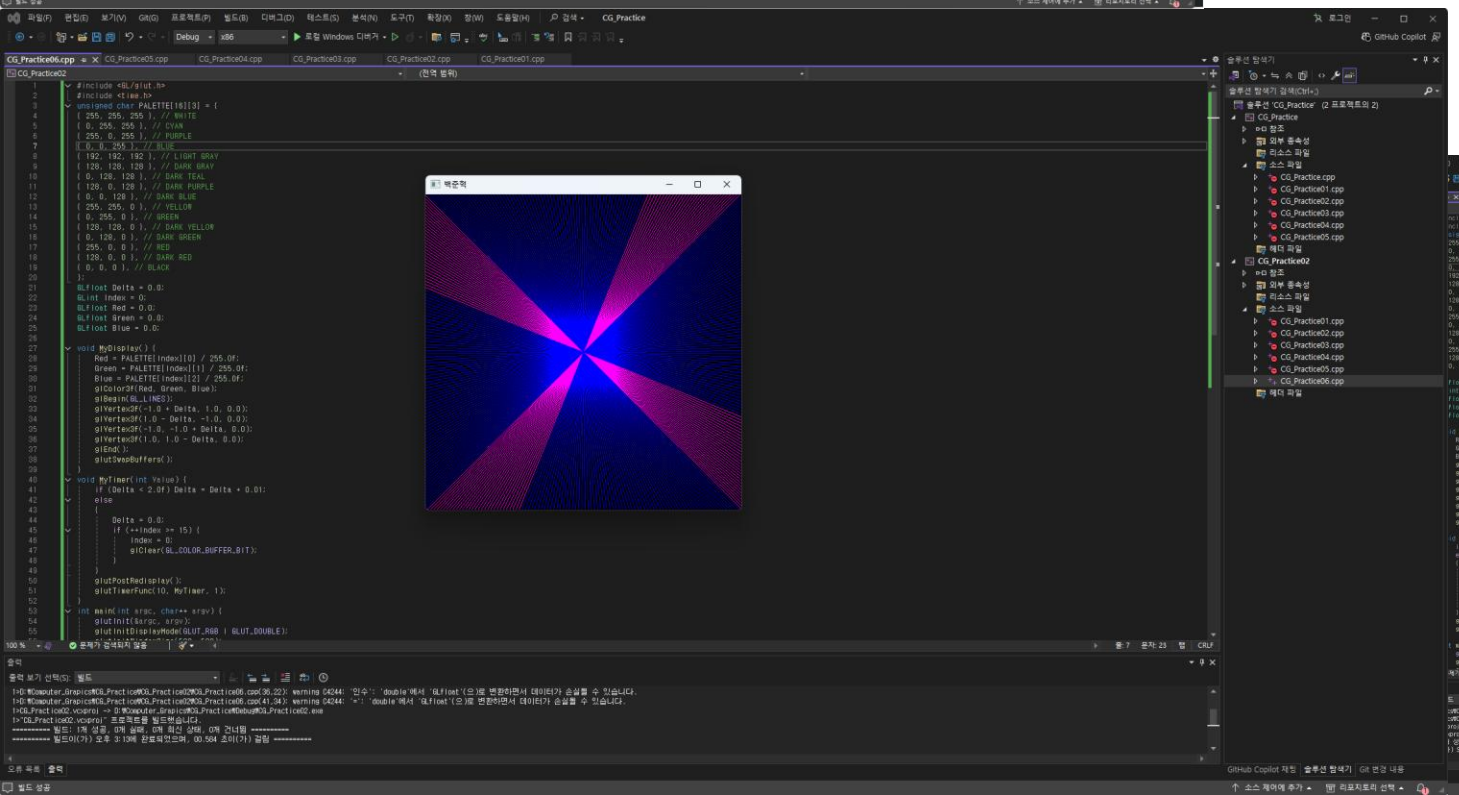
CG_Practice
CG_Practice06.cpp
CG_Practice05.cpp
CG_Practice04.cpp
CG_Practice03.cpp
CG_Practice02.cpp
CG_Practice01.cpp
CG_Practice02
#include <GL/glut.h>
#include <GL/gl.h>
#include <GL/glu.h>
GLfloat Delta = 0.0;
void MyDisplay( ) {
glClear(GL_COLOR_BUFFER_BIT);
glBegin(GL_POLYGON);
glColor3f(0.0, 0.5, 0.8);
glVertex3f(-1.0 + Delta, -0.5, 0.0);
glVertex3f(0.0 + Delta, -0.5, 0.0);
glVertex3f(0.0 + Delta, 0.5, 0.0);
glVertex3f(-1.0 + Delta, 0.5, 0.0);
glEnd( );
glutSwapBuffers( );
}
void MyIdle( ) {
Delta = Delta + 0.001;
glutPostRedisplay( );
}
int main(int argc, char** argv) {
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);
glutInitWindowSize(300, 300);
glutInitWindowPosition(0, 0);
glutCreateWindow("백준혁");
glClearColor(1.0, 1.0, 1.0, 1.0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity( );
glOrtho(-1.0, 1.0, -1.0, 1.0, 1.0, -1.0);
glutDisplayFunc(MyDisplay);
glutIdleFunc(MyIdle);
glutMainLoop( );
return 0;
}
백준혁

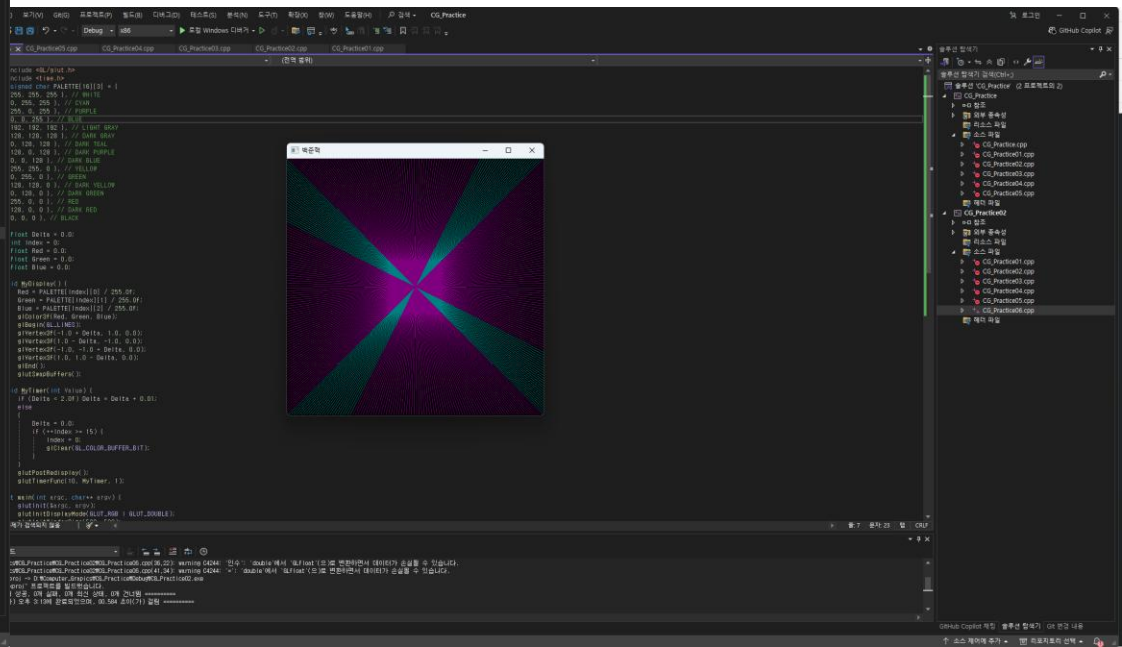


CG_Practice
CG_Practice06.cpp
CG_Practice05.cpp
CG_Practice04.cpp
CG_Practice03.cpp
CG_Practice02.cpp
CG_Practice01.cpp
CG_Practice02
#include <GL/glut.h>
#include <GL/gl.h>
#include <GL/glu.h>
GLfloat Delta = 0.0;
void MyDisplay( ) {
glClear(GL_COLOR_BUFFER_BIT);
glBegin(GL_POLYGON);
glColor3f(0.0, 0.5, 0.8);
glVertex3f(-1.0 + Delta, -0.5, 0.0);
glVertex3f(0.0 + Delta, -0.5, 0.0);
glVertex3f(0.0 + Delta, 0.5, 0.0);
glVertex3f(-1.0 + Delta, 0.5, 0.0);
glEnd( );
glutSwapBuffers( );
}
void MyIdle( ) {
Delta = Delta + 0.001;
glutPostRedisplay( );
}
int main(int argc, char** argv) {
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);
glutInitWindowSize(300, 300);
glutInitWindowPosition(0, 0);
glutCreateWindow("백준혁");
glClearColor(1.0, 1.0, 1.0, 1.0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity( );
glOrtho(-1.0, 1.0, -1.0, 1.0, 1.0, -1.0);
glutDisplayFunc(MyDisplay);
glutIdleFunc(MyIdle);
glutMainLoop( );
return 0;
}
백준혁




CG_Practice06.cpp
#include <GL/glut.h>
#include <time.h>
unsigned char PALETTE[16][3] = {
{ 255, 255, 255 }, // WHITE
{ 0, 255, 255 }, // CYAN
{ 255, 0, 255 }, // PURPLE
{ 0, 0, 255 }, // BLUE
{ 192, 192, 192 }, // LIGHT GRAY
{ 128, 128, 128 }, // DARK GRAY
{ 0, 128, 128 }, // DARK TEAL
{ 128, 0, 128 }, // DARK PURPLE
{ 0, 0, 128 }, // DARK BLUE
{ 255, 255, 0 }, // YELLOW
{ 0, 255, 0 }, // GREEN
{ 128, 128, 0 }, // DARK YELLOW
{ 0, 128, 0 }, // DARK GREEN
{ 255, 0, 0 }, // RED
{ 128, 0, 0 }, // DARK RED
{ 0, 0, 0 }, // BLACK
};
GLfloat Delta = 0.0;
GLint Index = 0;
GLfloat Red = 0.0;
GLfloat Green = 0.0;
GLfloat Blue = 0.0;
void MyDisplay() {
Red = PALETTE[Index][0] / 255.0f;
Green = PALETTE[Index][1] / 255.0f;
Blue = PALETTE[Index][2] / 255.0f;
glColor3f(Red, Green, Blue);
glBegin(GL_LINES);
glVertex2f(-1.0 + Delta, 1.0, 0.0);
glVertex2f(1.0 - Delta, -1.0, 0.0);
glVertex2f(-1.0, -1.0 + Delta, 0.0);
glVertex2f(1.0, 1.0 - Delta, 0.0);
glEnd();
glutSwapBuffers();
}
void MyTimer(int Value) {
if (Delta < 2.0f) Delta = Delta + 0.01;
else
{
Delta = 0.0;
if (++Index >= 15) {
Index = 0;
glClear(GL_COLOR_BUFFER_BIT);
}
}
glutPostRedisplay();
glutTimerFunc(10, MyTimer, 1);
}
int main(int argc, char** argv) {
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);
백준적
솔루션 탐색기
CG_Practice
CG_Practice02
출력
CG_Practice06.cpp(36,22): warning C4244: '인수': 'double'에서 'GLfloat'(으)로 변환하면서 데이터가 손실될 수 있습니다.
CG_Practice06.cpp(41,34): warning C4244: '=': 'double'에서 'GLfloat'(으)로 변환하면서 데이터가 손실될 수 있습니다.
CG_Practice02.vcxproj -> D:\Computer_Graphics\OGL_Practice\OGL_Practice02\OGL_Practice02\Debug\OGL_Practice02.exe
"OGL_Practice02.vcxproj" 프로젝트를 빌드했습니다.
========== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뜀 ==========
========== 빌드이(가) 오후 3:13에 완료되었으며, 00.584 초이(가) 걸림 ==========


CG_Practice06.cpp
#include <GL/glut.h>
#include <time.h>
unsigned char PALETTE[16][3] = {
{ 255, 255, 255 }, // WHITE
{ 0, 255, 255 }, // CYAN
{ 255, 0, 255 }, // PURPLE
{ 0, 0, 255 }, // BLUE
{ 192, 192, 192 }, // LIGHT GRAY
{ 128, 128, 128 }, // DARK GRAY
{ 0, 128, 128 }, // DARK TEAL
{ 128, 0, 128 }, // DARK PURPLE
{ 0, 0, 128 }, // DARK BLUE
{ 255, 255, 0 }, // YELLOW
{ 0, 255, 0 }, // GREEN
{ 128, 128, 0 }, // DARK YELLOW
{ 0, 128, 0 }, // DARK GREEN
{ 255, 0, 0 }, // RED
{ 128, 0, 0 }, // DARK RED
{ 0, 0, 0 }, // BLACK
};
GLfloat Delta = 0.0;
GLint Index = 0;
GLfloat Red = 0.0;
GLfloat Green = 0.0;
GLfloat Blue = 0.0;
void MyDisplay() {
Red = PALETTE[Index][0] / 255.0f;
Green = PALETTE[Index][1] / 255.0f;
Blue = PALETTE[Index][2] / 255.0f;
glColor3f(Red, Green, Blue);
glBegin(GL_LINES);
glVertex2f(-1.0 + Delta, 1.0, 0.0);
glVertex2f(1.0 - Delta, -1.0, 0.0);
glVertex2f(-1.0, -1.0 + Delta, 0.0);
glVertex2f(1.0, 1.0 - Delta, 0.0);
glEnd();
glutSwapBuffers();
}
void MyTimer(int Value) {
if (Delta < 2.0f) Delta = Delta + 0.01;
else
{
Delta = 0.0;
if (++Index >= 15) {
Index = 0;
glClear(GL_COLOR_BUFFER_BIT);
}
}
glutPostRedisplay();
glutTimerFunc(10, MyTimer, 1);
}
int main(int argc, char** argv) {
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);
백준적
솔루션 탐색기
CG_Practice
CG_Practice02
출력
CG_Practice06.cpp(36,22): warning C4244: '인수': 'double'에서 'GLfloat'(으)로 변환하면서 데이터가 손실될 수 있습니다.
CG_Practice06.cpp(41,34): warning C4244: '=': 'double'에서 'GLfloat'(으)로 변환하면서 데이터가 손실될 수 있습니다.
CG_Practice02.vcxproj -> D:\Computer_Graphics\OGL_Practice\OGL_Practice02\OGL_Practice02\Debug\OGL_Practice02.exe
"OGL_Practice02.vcxproj" 프로젝트를 빌드했습니다.
========== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뜀 ==========
========== 빌드이(가) 오후 3:13에 완료되었으며, 00.584 초이(가) 걸림 ==========


백준적
솔루션 탐색기
CG_Practice
CG_Practice02
출력