컴퓨터그래픽스

과제 3:

글로벌캠퍼스 네비게이션

201903868

컴퓨터전자시스템공학과

황가은

<소스코드>

```
#include <windows.h>
#include <stdio.h>
#include <GL/glut.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <math.h>
#include "bmp.h"     // replacing the obsolate file : <gl/glaux.h>
#include <GL/freeglut.h>


#define MAX_NO_TEXTURES 4

GLuint texture[MAX_NO_TEXTURES]; //총 4개의 사진을 넣기 위한 텍스쳐

void init(void);
```

```
//1인칭좌표
int eye_x = 20;
int eye_y = 20;
int eye_z = 10;

//내가 보는 시점좌표
int view_x = 20;
int view_y = 20;
int view_z = 0;

void OpenGlStart(void) //OpenGL초기세팅함수
{
	glEnable(GL_DEPTH_TEST);
	glEnable(GL_NORMALIZE);
	glEnable(GL_SMOOTH);
	glEnable(GL_LIGHTING); //조건4 : Lighting 넣기

	GLfloat ambientLight[] = { 0.5,0.5,0.5,1 };
	GLfloat diffuseLight[] = { 0.9,0.9,0.9,1 }; //어두워서 값을 좀 올려줌
	GLfloat specular[] = { 1,1,1,1 };
	GLfloat specref[] = { 1,1,1,1 };
	GLfloat position[] = { 400,300,700,1 }; //햇빛의 위치

	glLightfv(GL_LIGHT0, GL_AMBIENT, ambientLight);
	glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuseLight);
	glLightfv(GL_LIGHT0, GL_SPECULAR, specular);
```

```
	glLightfv(GL_LIGHT0, GL_POSITION, position);


	glEnable(GL_LIGHT0);
	glEnable(GL_COLOR_MATERIAL);


	glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);
	glMateriali(GL_FRONT, GL_SHININESS, 128);
	glClearColor(0.0, 0.6, 0.8, 0.5);


	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluPerspective(90, 4 / 3, 1, 700); //관점
}


void drawLand()//바닥
{
	//녹색 땅
	glPushMatrix();
	glBegin(GL_POLYGON);
	glNormal3f(0, 1, 0);
	glColor3f(0.0, 1.0, 0.0);
	glVertex3f(400, 0, 0);
	glVertex3f(400, 0, -1000);
	glVertex3f(-400, 0, -1000);
	glVertex3f(-400, 0, 0);
	glEnd();
```

```
glPopMatrix();

//중앙 도로
glPushMatrix();
glTranslatef(20, 2.5, 0);
glBegin(GL_POLYGON);
glColor3f(0, 0, 0);
glVertex3f(10, 0, 0);
glVertex3f(10, 0, -950);
glVertex3f(-10, 0, -950);
glVertex3f(-10, 0, 0);
glEnd();
glPopMatrix();

//기숙사 옆 도로
glPushMatrix();
glTranslatef(100, 2.5, -70);
glBegin(GL_POLYGON);
glColor3f(0, 0, 0);
glVertex3f(0, 0, 10);
glVertex3f(-80, 0, 10);
glVertex3f(-80, 0, 20);
glVertex3f(0, 0, 20);
glEnd();
glPopMatrix();
```

```
//공대앞도로
glPushMatrix();
glTranslatef(20, 2.5, -250);
glBegin(GL_POLYGON);
glColor3f(0, 0, 0);
glVertex3f(0, 0, 10);
glVertex3f(-60, 0, 10);
glVertex3f(-60, 0, 20);
glVertex3f(0, 0, 20);
glEnd();
glPopMatrix();

//명수당 앞 도로
glPushMatrix();
glTranslatef(10, 2.5, -600);
glBegin(GL_POLYGON);
glColor3f(0, 0, 0);
glVertex3f(0, 0, 10);
glVertex3f(-60, 0, 10);
glVertex3f(-60, 0, 20);
glVertex3f(0, 0, 20);
glEnd();
glPopMatrix();

//자대 앞도로
glPushMatrix();
```

```
glTranslatef(20, 2.5, -150);
glBegin(GL_POLYGON);
glColor3f(0, 0, 0);
glVertex3f(0, 0, 10);
glVertex3f(-100, 0, 10);
glVertex3f(-100, 0, 20);
glVertex3f(0, 0, 20);
glEnd();
glPopMatrix();

//도서관앞도로
glPushMatrix();
glTranslatef(80, 2.5, -350);
glBegin(GL_POLYGON);
glColor3f(0, 0, 0);
glVertex3f(0, 0, 10);
glVertex3f(-60, 0, 10);
glVertex3f(-60, 0, 20);
glVertex3f(0, 0, 20);
glEnd();
glPopMatrix();

//도서관앞주차장
glPushMatrix();
glTranslatef(90, 2.5, -255);
glBegin(GL_POLYGON);
```

```
		glColor3f(0, 0, 0);
		glVertex3f(35, 0, 0);
		glVertex3f(35, 0, -100);
		glVertex3f(-35, 0, -100);
		glVertex3f(-35, 0, 0);
		glEnd();
		glPopMatrix();

}

void drawDorm()//기숙사
{
		//앞의두건물
		glColor3f(1, 1, 1);
		glTranslatef(80, 20, -100);
		glScalef(1.5, 4, 4);
		glutSolidCube(10);
		glTranslatef(-15, 0, 0);
		glutSolidCube(10);

		//뒤의두건물
		glPushMatrix();
		glTranslatef(0, 0, -20);
		glScalef(1.5, 4, 4);
		glutSolidCube(4);
		glTranslatef(10, 0, 0);
```

```
		glutSolidCube(4);
		glPopMatrix();
}
void drawGongDae()//공대
{
		glColor3f(0.8, 0.8, 0.8);
		glTranslatef(-80, 0, -50);
		glScalef(4, 3, 2);
		glutSolidCube(7);

}
void drawMath()//자연대
{
		glColor3f(1, 1, 1);
		glTranslatef(-10, 0, 15);
		glScalef(3, 3, 4);
		glutSolidCube(2);

}

void drawLibrary()//도서관
{
		glPushMatrix();
		glColor3f(1, 1, 1);
		glTranslatef(12, 0, -7);
		glScalef(1, 1, 1);
```

```
	glutSolidCube(2);
	glPopMatrix();
}

void drawMyungsu()//명수당(원으로 구현함)
{
	glTranslatef(-6, 0, 0);
	glPushMatrix();
	glTranslatef(2, -0.5, -15);
	glColor3f(0, 0, 1);
	glScalef(8, 0.1, 5);
	glutSolidSphere(0.9, 100, 100);
	glPopMatrix();
}


void Navigation_Program() //네비게이션 프로그램
{
	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
	gluLookAt(eye_x, eye_y, eye_z, view_x, view_y, view_z, 0, 1, 0);
	glEnable(GL_TEXTURE_2D);
	glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);

	drawLand(); //땅을 먼저 만들고
```

```
glBindTexture(GL_TEXTURE_2D, texture[0]); //처음 기숙사를 텍스쳐 맵핑을 통해 한면만 입혀줌
drawDorm();
glBegin(GL_QUADS);
glTexCoord2f(0.0f, 0.0f); glVertex3f(-6.0, -5.0, 5.5);
glTexCoord2f(1.0f, 0.0f); glVertex3f(5.0, -5.0, 5.5);
glTexCoord2f(1.0f, 1.0f); glVertex3f(5.0, 5.0, 5.5);
glTexCoord2f(0.0f, 1.0f); glVertex3f(-6.0, 5.0, 5.5);
glEnd();

glBindTexture(GL_TEXTURE_2D, texture[1]); //두 번째로 공대를 텍스쳐 맵핑을 통해 한면만 입혀줌
drawGongDae();
glBegin(GL_QUADS);
glTexCoord2f(0.0f, 0.0f); glVertex3f(-4.0, -2.0, 3.6);
glTexCoord2f(1.0f, 0.0f); glVertex3f(4.0, -2.0, 3.6);
glTexCoord2f(1.0f, 1.0f); glVertex3f(4.0, 3.5, 3.6);
glTexCoord2f(0.0f, 1.0f); glVertex3f(-4.0, 3.5, 3.6);
glEnd();

glBindTexture(GL_TEXTURE_2D, texture[2]);  //세 번째로 자연대를 텍스쳐 맵핑을 통해 한면만 입혀줌
drawMath();
glBegin(GL_QUADS);
glTexCoord2f(0.0f, 0.0f); glVertex3f(-1.0, -0.6, 1.05);
```

```
		glTexCoord2f(1.0f, 0.0f); glVertex3f(1.0, -0.6, 1.05);
		glTexCoord2f(1.0f, 1.0f); glVertex3f(1.0, 1.0, 1.05);
		glTexCoord2f(0.0f, 1.0f); glVertex3f(-1.0, 1.0, 1.05);
		glEnd();

		glBindTexture(GL_TEXTURE_2D, texture[3]);  //마지막 으로 도서관을 텍스쳐 맵핑을 통해 한면만 입혀줌
		drawLibrary();
		glBegin(GL_QUADS);
		glTexCoord2f(0.0f, 0.0f); glVertex3f(11.0, -1.0, -5.95);
		glTexCoord2f(1.0f, 0.0f); glVertex3f(13.0, -1.0, -5.95);
		glTexCoord2f(1.0f, 1.0f); glVertex3f(13.0, 1.0, -5.95);
		glTexCoord2f(0.0f, 1.0f); glVertex3f(11.0, 1.0, -5.95);
		glEnd();

		drawMyungsu(); //명수당을 그려줌

		glDisable(GL_TEXTURE_2D);
		glutSwapBuffers();

}

void InsertKey(unsigned char key, int x, int y)//입력된키처리
{
		switch (key) {
		case 's': //앞으로 전진
```

```
            eye_z -= 10;
            view_z -= 10;
            glutPostRedisplay();
            break;
case 'd': //뒤로 후진
            eye_z += 10;
            view_z += 10;
            glutPostRedisplay();
            break;

            //시점변경
case 'u': //고개를 약간 들기
            view_y += 2;
            glutPostRedisplay();
            break;
case 'b': //고개를 약간 내리기
            view_y -= 2;
            glutPostRedisplay();
            break;
case 'l': //시계방향으로 5도회전
            view_x -= 1;
            glutPostRedisplay();
            break;
case 'r': //시계반대방향으로 5도 회전
            view_x += 1;
            glutPostRedisplay();
```

```
                break;
        }
}


AUX_RGBImageRec* auxDIBImageLoad(const char* FileName);


AUX_RGBImageRec* LoadBMP(char* Filename)       // Loads A Bitmap Image
{
        FILE* File = NULL;              // File Handle

        if (!Filename)                  // Make Sure A Filename Was Given
        {
                return NULL;            // If Not Return NULL
        }

        File = fopen(Filename, "r");            // Check To See If The File Exists

        if (File)                       // Does The File Exist?
        {
                fclose(File);           // Close The Handle
                return auxDIBImageLoad((const char*)Filename);       // Load The Bitmap And
Return A Pointer
        }
```

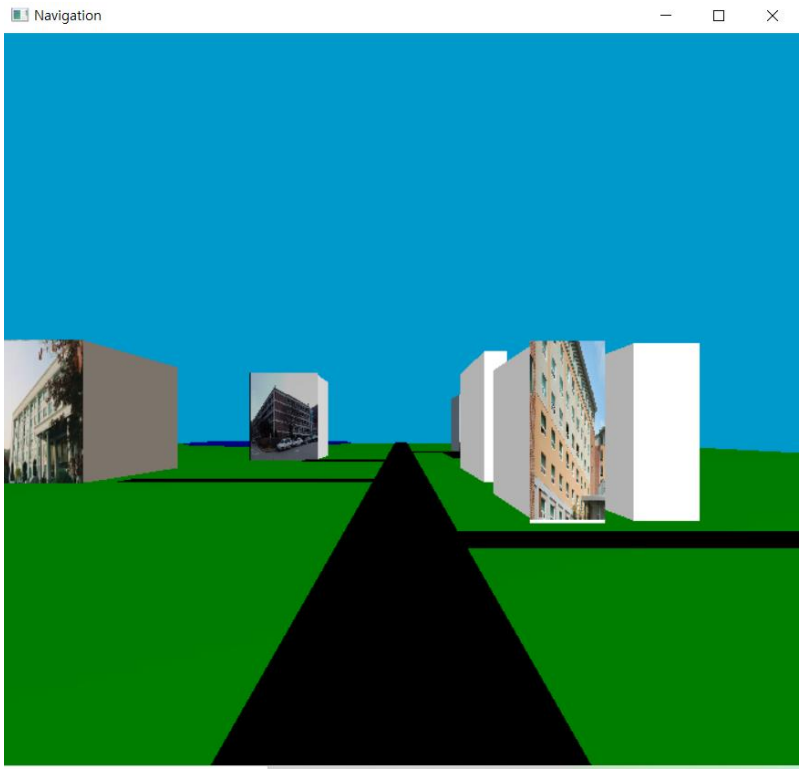
```
        return NULL;                // If Load Failed Return NULL
}

void init(void)
{
        glShadeModel(GL_SMOOTH);            // Enable Smooth Shading
        glClearColor(0.0f, 0.0f, 0.0f, 0.0f);       // Black Background
        glEnable(GL_COLOR_MATERIAL);
        glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);

        glEnable(GL_TEXTURE_2D);
        glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
        glGenTextures(MAX_NO_TEXTURES, texture);

        AUX_RGBImageRec* TextureImage[MAX_NO_TEXTURES];        // Create Storage Space For The Texture

        memset(TextureImage, 0, sizeof(void*) * MAX_NO_TEXTURES);                    // Set The Pointer To NULL

    // if ((TextureImage[0] = LoadBMP("삼성 4-1 SLIDE(2)_00001.bmp")) &&
    //   (TextureImage[1] = LoadBMP("삼성 4-1 SLIDE(2)_00002.bmp")))     //이미지 로딩
        TextureImage[0] = LoadBMP((char*)"./dorm.bmp");
        TextureImage[1] = LoadBMP((char*)"./Gongdae.bmp");
        TextureImage[2] = LoadBMP((char*)"./jaedae2.jpg");
        TextureImage[3] = LoadBMP((char*)"./Library3.bmp");
```

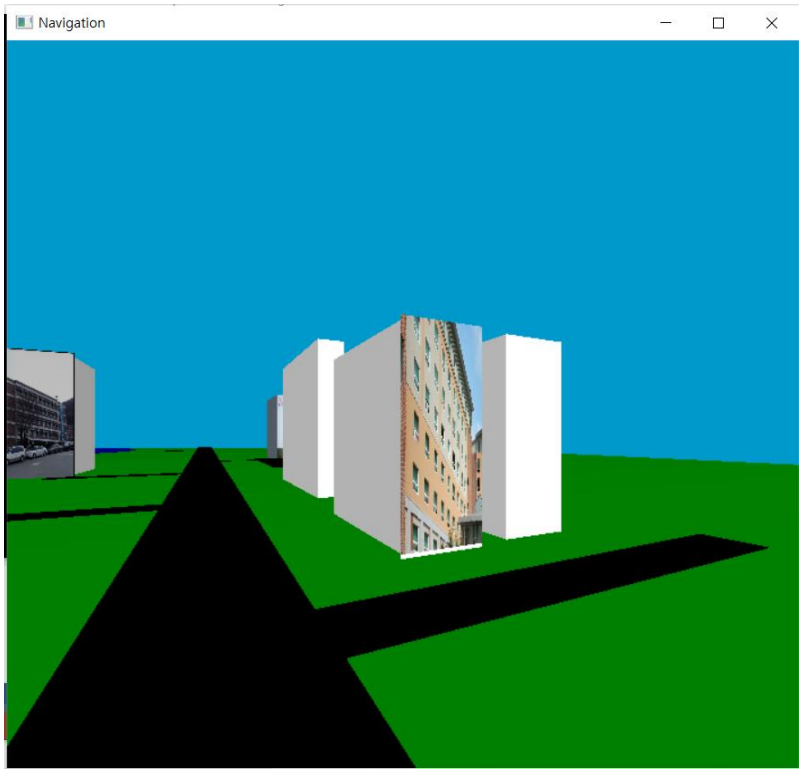
```
{
        glGenTextures(MAX_NO_TEXTURES, texture);       //텍스쳐 생성

                                        //텍스쳐에 이미지 1 넣기
        glBindTexture(GL_TEXTURE_2D, texture[0]);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
        glTexImage2D(GL_TEXTURE_2D, 0, 3, TextureImage[0]->sizeX, TextureImage[0]->sizeY, 0, GL_RGB, GL_UNSIGNED_BYTE, TextureImage[0]->data);
        //텍스쳐에 이미지 2 넣기

        glBindTexture(GL_TEXTURE_2D, texture[1]);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR_MIPMAP_NEAREST);
        glTexImage2D(GL_TEXTURE_2D, 0, 3, TextureImage[1]->sizeX, TextureImage[1]->sizeY, 0, GL_RGB, GL_UNSIGNED_BYTE, TextureImage[1]->data);
        //gluBuild2DMipmaps(GL_TEXTURE_2D, 3, TextureImage[1]->sizeX, TextureImage[1]->sizeY, GL_RGB, GL_UNSIGNED_BYTE, TextureImage[1]->data);

        glBindTexture(GL_TEXTURE_2D, texture[2]);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR_MIPMAP_NEAREST);
        glTexImage2D(GL_TEXTURE_2D, 0, 3, TextureImage[2]->sizeX, TextureImage[2]->sizeY, 0, GL_RGB, GL_UNSIGNED_BYTE, TextureImage[2]->data);
```

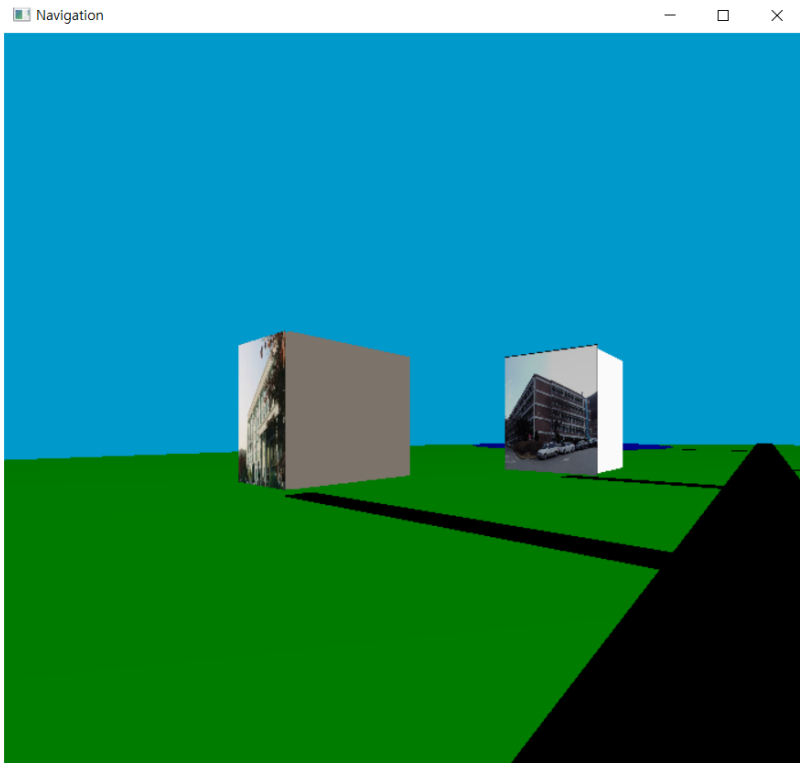
```
                glBindTexture(GL_TEXTURE_2D, texture[3]);
                glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
                glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,
GL_LINEAR_MIPMAP_NEAREST);
                glTexImage2D(GL_TEXTURE_2D, 0, 3, TextureImage[3]->sizeX, TextureImage[3]-
>sizeY, 0, GL_RGB, GL_UNSIGNED_BYTE, TextureImage[3]->data);
        }

        glEnable(GL_CULL_FACE);
        // glEnable(GL_DEPTH_TEST);
}

void main(int argc, char** argv)//main
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_DEPTH | GLUT_SINGLE | GLUT_RGBA);
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        glColor3f(0.2, 1, 1); //하늘은 하늘색
        glutInitWindowSize(700, 700);
        glutInitWindowPosition(150, 150);
        glutCreateWindow("Navigation");
        init();
        glutDisplayFunc(Navigation_Program);
        glutKeyboardFunc(InsertKey);
        OpenGlStart();
        glutMainLoop();
```

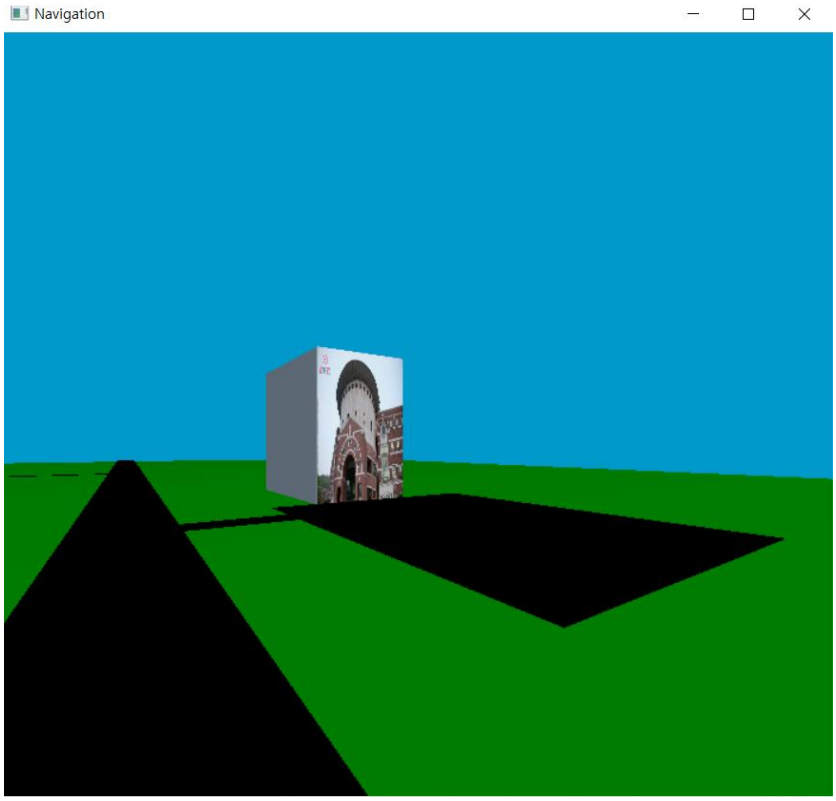
}

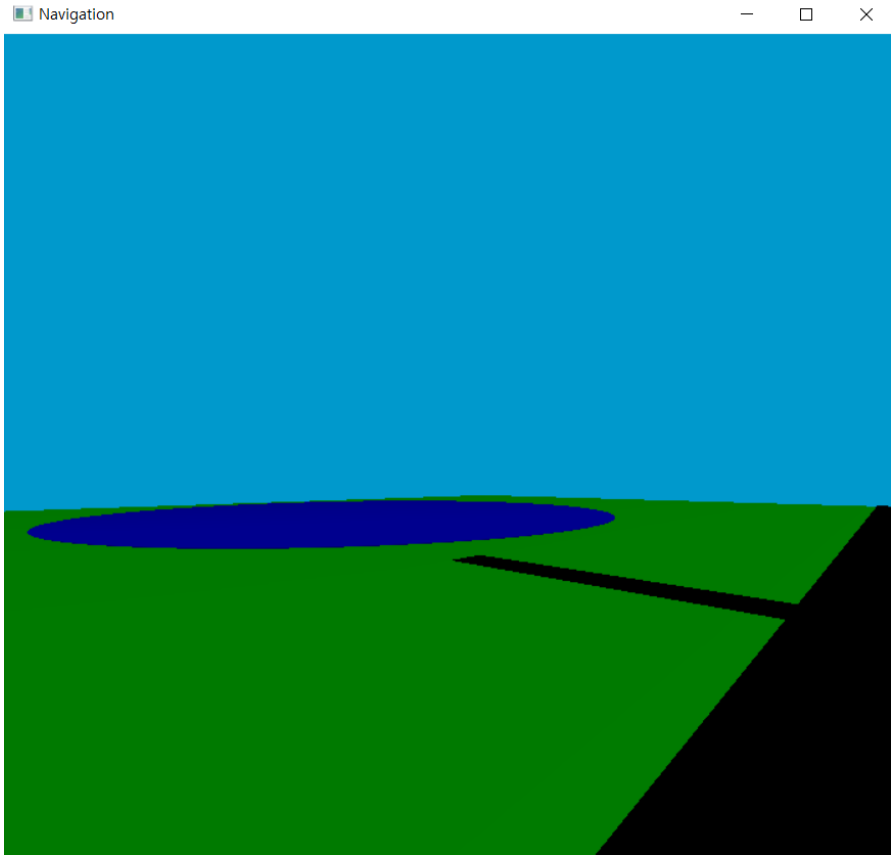
<화면 DUMP>




Navigation


Navigation


Navigation


Navigation

<소감>

종강후 마지막 과제인데 진짜 너무 어려웠습니다. 2D만 만들다가 3D를 만들려고 하니 하기가 굉장히 복잡했고 시간도 제일 오래 걸렸던 것 같습니다. 그리고 텍스쳐 맵핑은 한면 밖에 성공하지 못했습니다.