



R E P O R T

컴퓨터그래픽스및실습 실습과제 03

과목명	컴퓨터그래픽스및실습
분반	01
교수	최 영 규
학번	2020136129
이름	최 수 연
제출일	2022년 9월 26일 월요일

실습 3.1: 컴퓨터에 GLUT 설치하기

freelut-MSVC-3.0.0-2.mp	2022-09-21 오후 9:19	파일 폴더
<p>https://www.transmissionzero.co.uk/software/freelut-devel/ 위 사이트에서, Download freelut 3.0.0 for MSVC (with PGP signature and PGP key) 를 클릭하여 위 사진의 폴더를 다운로드 한다.</p>		
<p>C:\Program Files (x86)\Windows Kits\10\Include\10.0.17763.0\um\gl 위 경로에, C:\Users\choi6\Desktop\3-2\컴퓨터그래픽스및실습\freelut-MSVC-3.0.0-2.mp\freelut\include\GL 위 경로에 있는 모든 파일을 복사 붙여넣기 한다.</p>		
<p>C:\Program Files (x86)\Windows Kits\10\Lib\10.0.17763.0\um\x86 위 경로에, C:\Users\choi6\Desktop\3-2\컴퓨터그래픽스및실습\freelut-MSVC-3.0.0-2.mp\freelut\lib 위 경로에 있는 freelut.lib 파일을 복사 붙여넣기 한다.</p>		
<p>C:\Windows\SysWOW64 위 경로에, C:\Users\choi6\Desktop\3-2\컴퓨터그래픽스및실습\freelut-MSVC-3.0.0-2.mp\freelut\bin 위 경로에 있는 freelut.dll 파일을 복사 붙여넣기 한다. 설치 완료.</p>		

실습 3.4: 자신의 이름 그리기

문제 분석 및 해결 방법

- 본 문제는 도형으로 이름을 그리는 문제로, 각 정점의 좌표를 계산하여 자신의 이름을 출력하는 함수를 구현하는 문제이다.
- 다각형(GL_POLYGON)을 사용하여 본인의 이름에 맞게 좌표를 직접 계산해 이름을 그린 코드를 구현한다.
- 다각형(GL_POLYGON)을 사용할 때는 좌표 순서를 잘 맞춰야 하는데, 좌표를 적은 순서대로 선이 연결되어 다각형이 만들어진다. 따라서 원하는 모양의 도형을 만들어내기 위해서는 좌표를 그릴 때 서로 이웃한 좌표를 찾아 순차적으로 그려야 한다.

주요 설명 코드

```
#include <GL/glut.h> // OpenGL을 사용하기 위해 해당 라이브러리 함수를 추가한다.

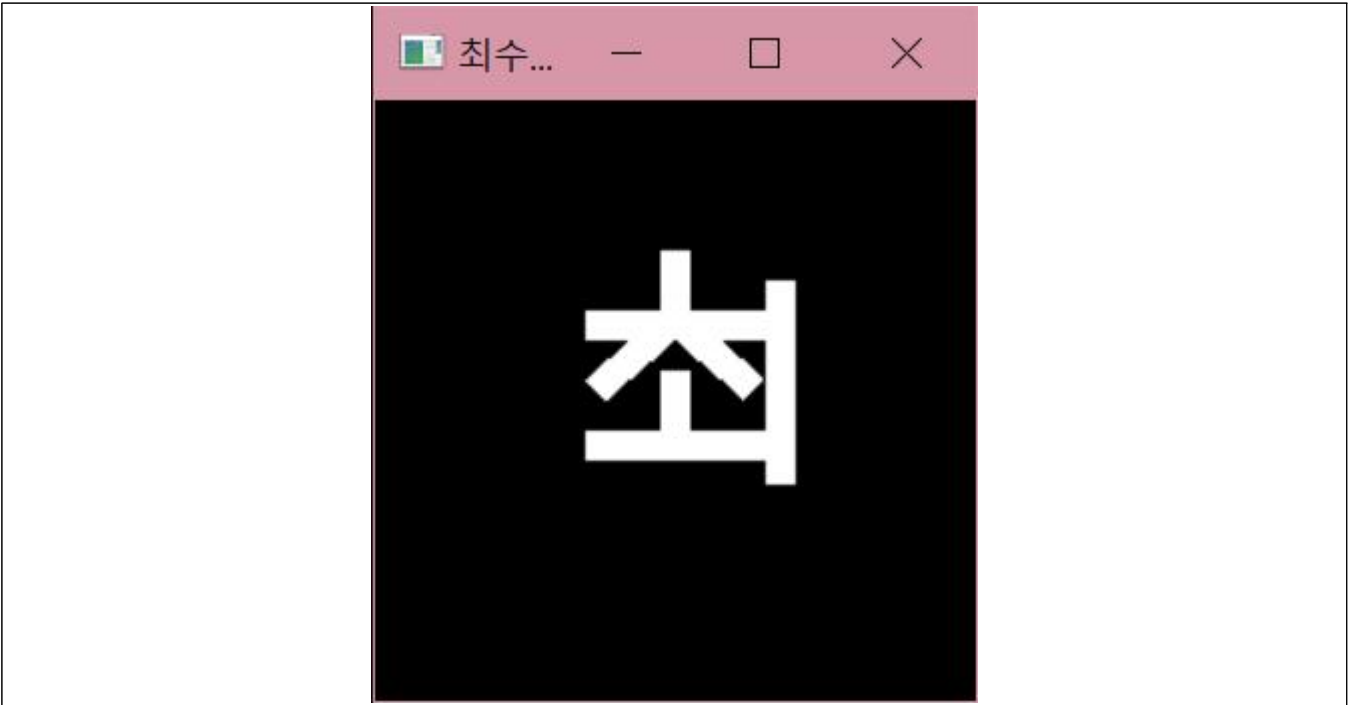
glClear(GL_COLOR_BUFFER_BIT); // 색상 버퍼를 지워준다.

glBegin(GL_POLYGON); // 정점이 연결된 다각형을 만들어낸다.
glVertex2f(-0.05, 0.5); // 정점 하나의 좌표를 의미한다.
glVertex2f(-0.05, 0.25);
glVertex2f(0.05, 0.25);
glVertex2f(0.05, 0.5);
glEnd(); // 다각형 그리기를 마무리한다.

glFinish(); // 모든 명령어가 종료된 후 화면에 출력하도록 한 번에 코드를 실행 완료 후 반환

glutInitWindowPosition(100, 100); // 직접 x, y 좌표를 지정하여 화면에서 Window 창의 위치를 정한다.
glutInitWindowSize(200, 200); // 직접 x, y 좌표를 지정하여 Window 창의 크기를 정한다.
glutCreateWindow("최수연_과제3.4"); // 새 Window를 생성하는 명령어이며 상단 title도 지정할 수 있다.
glutDisplayFunc(MyDisplay); // 콜백 함수인 MyDisplay() 함수를 호출한다.
glutMainLoop(); // OpenGL의 메인 함수가 끝났을 때 선언하는 함수이다.
```

테스트 결과



실습 3.5: Sierpinski Gasket

문제 분석 및 해결 방법

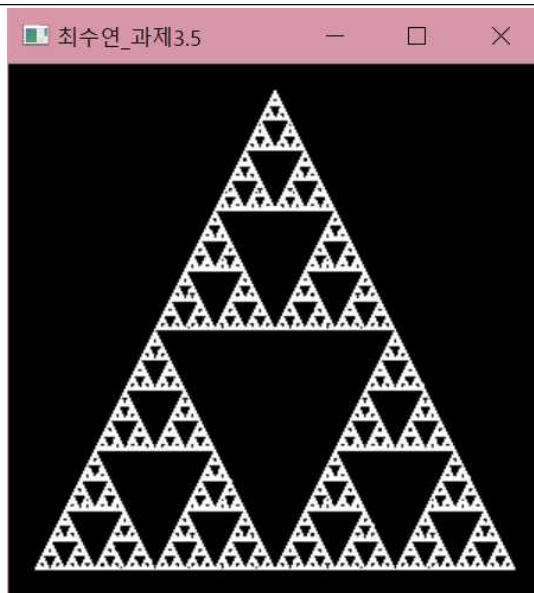
- ① 삼각형 내부에서 무작위로 한 개의 시작점을 잡는다.
 - ② 무작위로 세 개의 정점 중 한 개를 선택한다.
 - ③ 시작점과 무작위로 선택된 정점의 중간점을 잡는다.
 - ④ 그 위치에 작은 원 같은 표지를 함으로써 새로운 점을 표시한다.
 - ⑤ 이 새로운 점을 시작점으로 대치한다.
 - ⑥ 단계 2로 돌아간다. (이 과정을 주어진 횟수만큼 반복)
- 시어핀스키 삼각형은 정삼각형 하나에서 세 변의 중점을 이어 무한으로 작은 삼각형을 만드는 것으로, 중점을 찾아 각 변의 중점을 이어 정점을 찍어내는 것이 관건이다.

주요 설명 코드

```
GLfloat vertices[3][2] = { {-0.9, -0.9}, {0.9, -0.9}, {0, 0.9} }; // 큰 삼각형의 세 정점 배열을 선언한다.
GLfloat p[2] = { 0.3, 0.1 }; // 윗글의 ①번으로 float 타입으로 무작위로 하나의 정점 배열을 선언한다.

glBegin(GL_POINTS); // 독립적인 점을 찍어낸다.
for (int k = 0; k < 500000; k++) { 윗글의 ⑥번으로 50만 번 본 과정을 반복한다.
    int v = rand() % 3; // 윗글의 ②번으로 무작위로 세 개의 정점 중 하나를 선택한다.
    p[0] = (p[0] + vertices[v][0]) / 2.0; // 윗글의 ③번으로 무작위로 선택된 정점의 중간 점을 잡는다.
    p[1] = (p[1] + vertices[v][1]) / 2.0;
    glVertex2fv(p); // 꼭짓점을 지정하는 함수로 하나의 정점을 찍어낸다.
```

테스트 결과



고찰 및 느낀점

일반적인 코딩과는 달리 코드를 작성하는 대로 바로 한눈에 확인할 수 있어서 좋았다. 그러나 내가 원하는 대로 쉽게 그려지지 않아서 생각보다 시간이 오래 걸리고 번거로운 작업인 것 같다. 그래도 직관적으로 볼 수 있는 것이 흥미로웠고, 운영체제 팀 프로젝트에서 했던 GUI가 생각나서 더 재밌었다.