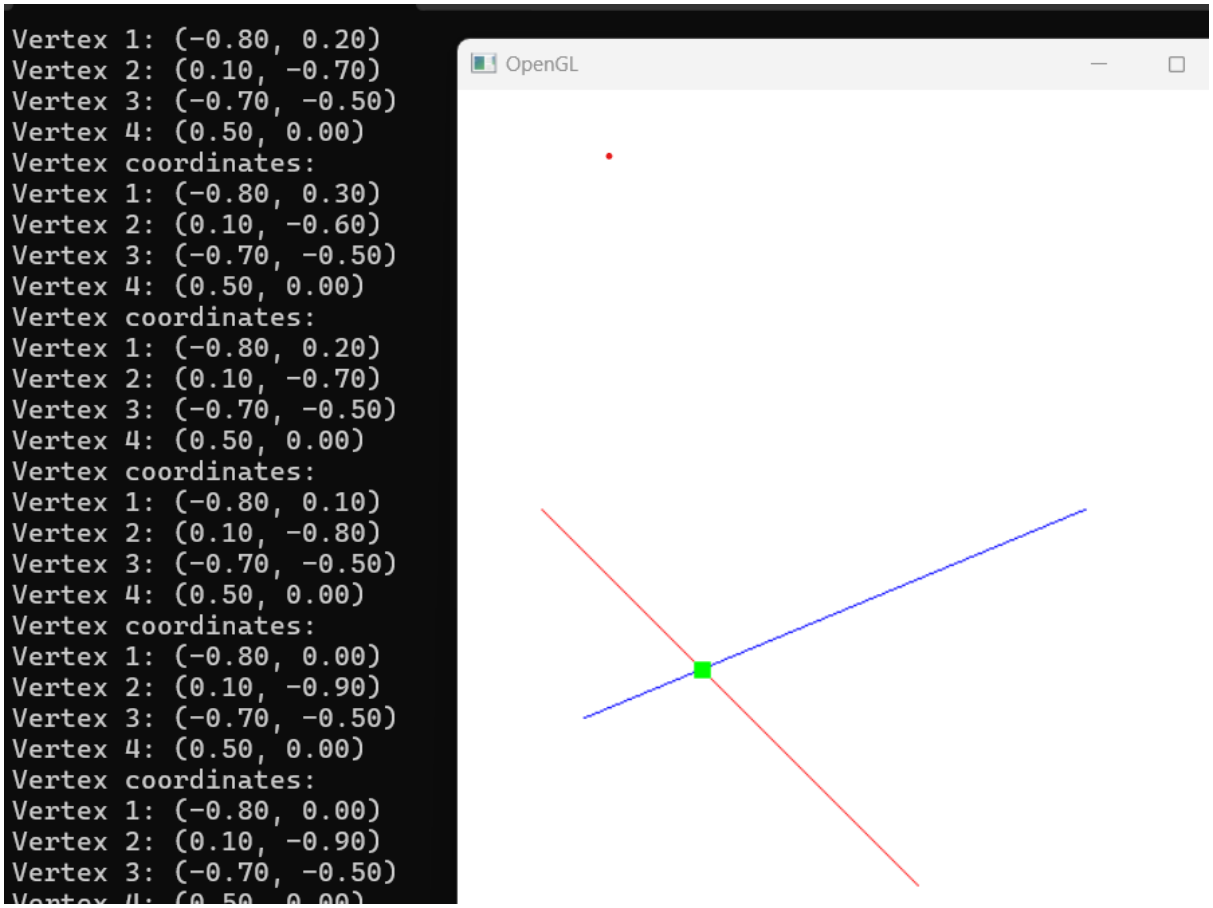
Homework 2 : 선분과 선분의 교차

컴퓨터공학과 1924073 최유현

추가구현 여부 : O

1. 렌더링



2. 설명

OpenGL의 gl_lines을 이용하여 선분을 그린후, glBegin와 glEnd라는 선분의 두개의 점 사이에서 각 선분의 시작과 끝을 glVertex2f()를 통해 지정한 후, 교차지점을 계산했습니다.

교차지점 계산법은 선분의 방정식을 이용하여 계산했습니다. 교차점이 생기면 추가한 greendot()함수로 점을 표시합니다.

(+add) 만약 두 선분이 평행하지 않고, 교차점이 선분 내부에 있을 경우에만 교차점을 표시하는 기능을 구현했습니다.

3. 코드

```
#include "gl/glut.h"
#include <cstdio>

struct Vec2 {
    float x, y;
};

Vec2 linePt[4] = {
    {-0.3f, 0.2f},
    {0.6f, -0.7f},
    {-0.7f, -0.5f},
    {0.5f, 0.0f}
};

void greendot() {
    // 접점 계산 함수
    float x1 = linePt[0].x, y1 = linePt[0].y;
    float x2 = linePt[1].x, y2 = linePt[1].y;
    float x3 = linePt[2].x, y3 = linePt[2].y;
    float x4 = linePt[3].x, y4 = linePt[3].y;

    float xMin = -1.0f, xMax = 1.0f, yMin = -1.0f, yMax = 1.0f; // 화면의 범위
    bool intersectionFound = false;

    float denominator = ((y4 - y3) * (x2 - x1)) - ((x4 - x3) * (y2 - y1));

    if (denominator == 0) {
        return;
    }

    float ua = (((x4 - x3) * (y1 - y3)) - ((y4 - y3) * (x1 - x3))) /
denominator;
    float ub = (((x2 - x1) * (y1 - y3)) - ((y2 - y1) * (x1 - x3))) /
denominator;

    if (ua >= 0 && ua <= 1 && ub >= 0 && ub <= 1) {
        float intersectionX = x1 + (ua * (x2 - x1));
        float intersectionY = y1 + (ua * (y2 - y1));

        // Check if intersection point is within screen boundaries
        if (intersectionX >= xMin && intersectionX <= xMax && intersectionY
>= yMin && intersectionY <= yMax) {
            glColor3f(0.0, 1.0, 0.0);
            glPointSize(10.0);
            glBegin(GL_POINTS);
            glVertex2f(intersectionX, intersectionY);
            glEnd();
            intersectionFound = true;
```

```
49.         }
50.     }
51.
52.     if (!intersectionFound) {
53.         if ((x1 == x2 && (x1 >= xMin && x1 <= xMax) && ((y3 <= y1 && y4 >=
    y1) || (y3 >= y1 && y4 <= y1))) ||
54.             (y1 == y2 && (y1 >= yMin && y1 <= yMax) && ((x3 <= x1 && x4 >=
    x1) || (x3 >= x1 && x4 <= x1)))) {
55.             glColor3f(0.0, 1.0, 0.0);
56.             glPointSize(10.0);
57.             glBegin(GL_POINTS);
58.             glVertex2f(x1, y1);
59.             glEnd();
60.         }
61.     }
62. }
63.
64. void display() {
65.     glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
66.     glClear(GL_COLOR_BUFFER_BIT);
67.
68.     glColor3f(1.0, 0.0, 0.0);
69.     glBegin(GL_LINES);
70.     glVertex2f(linePt[0].x, linePt[0].y);
71.     glVertex2f(linePt[1].x, linePt[1].y);
72.     glEnd();
73.
74.     glColor3f(0.0, 0.0, 1.0);
75.     glBegin(GL_LINES);
76.     glVertex2f(linePt[2].x, linePt[2].y);
77.     glVertex2f(linePt[3].x, linePt[3].y);
78.     glEnd();
79.
80.     greendot();
81.
82.     glutSwapBuffers();
83. }
84.
85. void keyboard(unsigned char key, int x, int y) {
86.
87.     switch (key) {
88.     case 'w':
89.         linePt[0].y += 0.1f;
90.         linePt[1].y += 0.1f;
91.         break;
92.     case 's':
93.         linePt[0].y -= 0.1f;
94.         linePt[1].y -= 0.1f;
95.         break;
96.     case 'a':
```

```
97.         linePt[0].x -= 0.1f;
98.         linePt[1].x -= 0.1f;
99.         break;
100.            case 'd':
101.                linePt[0].x += 0.1f;
102.                linePt[1].x += 0.1f;
103.                break;
104.            case 27: // 종료
105.                exit(0);
106.                break;
107.            }
108.            glutPostRedisplay();
109.        }
110.
111.        int main(int argc, char **argv)
112.        {
113.            glutInit(&argc, argv);
114.            glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);
115.            glutInitWindowSize(500, 500);
116.            glutInitWindowPosition(1480, 100);
117.
118.            glutCreateWindow("OpenGL");
119.            glutDisplayFunc(display);
120.            glutKeyboardFunc(keyboard);
121.            glutMainLoop();
122.
123.            return 0;
124.        }
```