

심화전공실습

HW #03



Self-scoring table

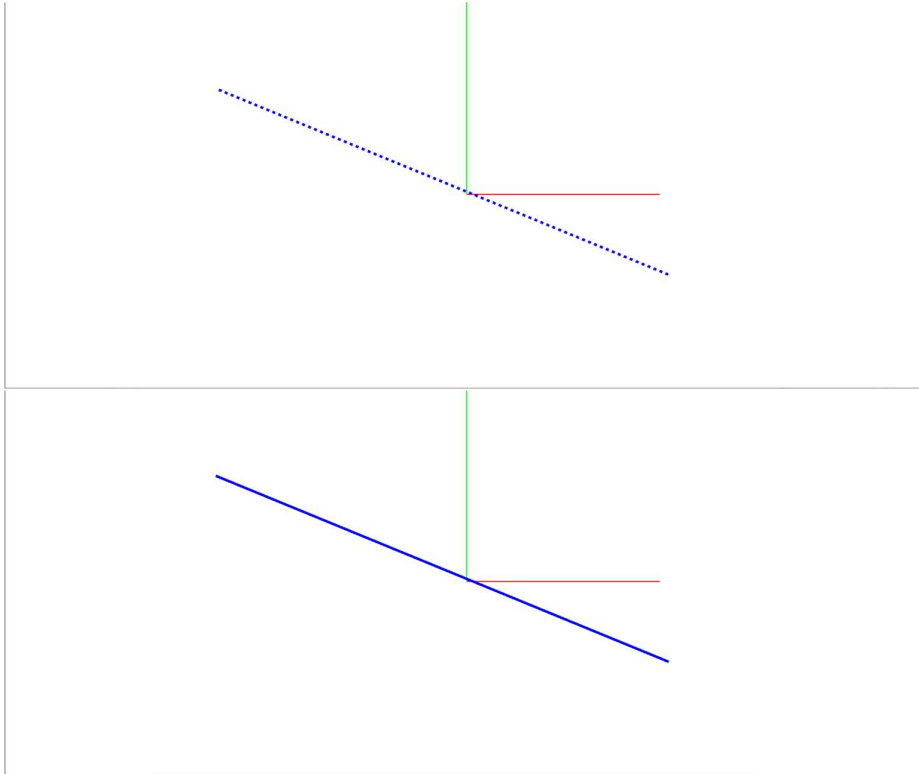
	P01	P02	E01	E02	E03	Total
Score	1	1	1	1	1	5

과목명	심화전공실습
학부	소프트웨어학부
학번	2019203010
이름	김민철
제출일자	2020년 09월 24일

I. Practice

Practice 01. Draw line segment

01. 선분을 그리기 위한 마우스 입력



02. 선분 애니메이션을 위한 타이머

```
float now = (float)glfwGetTime();
float delta_t = now - previous;
previous = now;

elapsed += delta_t;

if (elapsed > timeStep)
{
    if (!pause && inputMode == InputMode::COMPLETE)
    {
        float delta_theta = float(2.0 * M_PI) / period * elapsed;

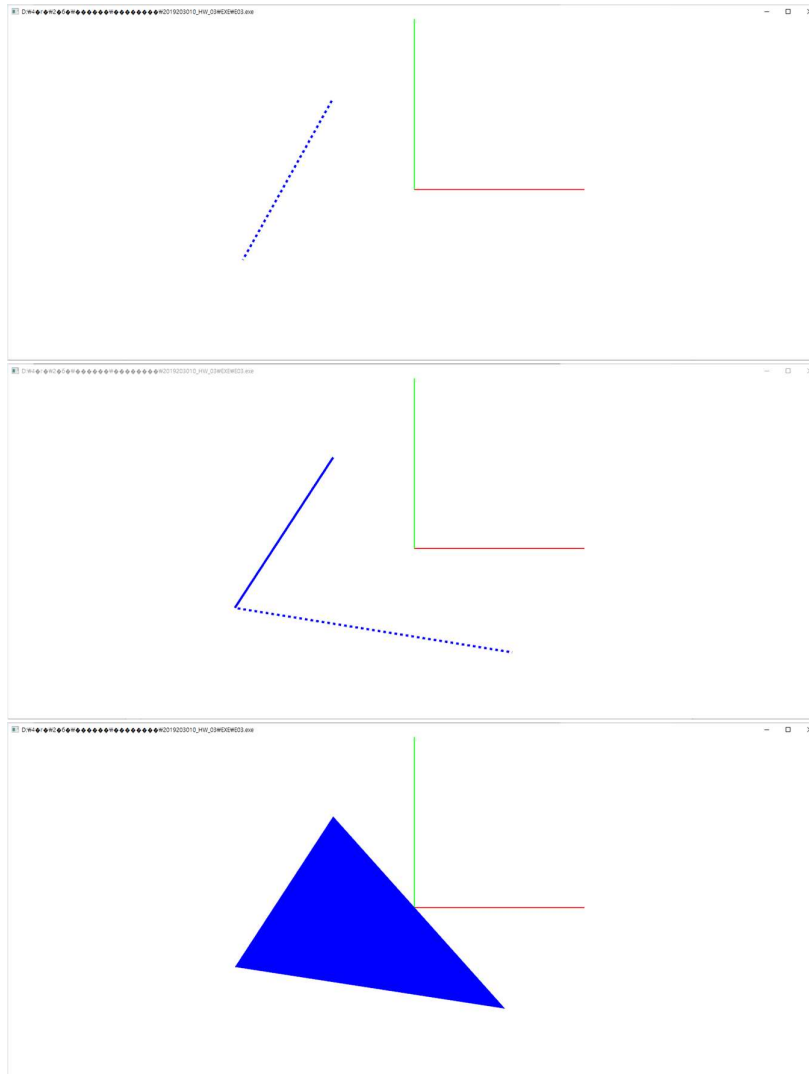
        rotate(point[0], delta_theta);
        rotate(point[1], delta_theta);
    }

    elapsed = 0;
}
```

II. Exercise

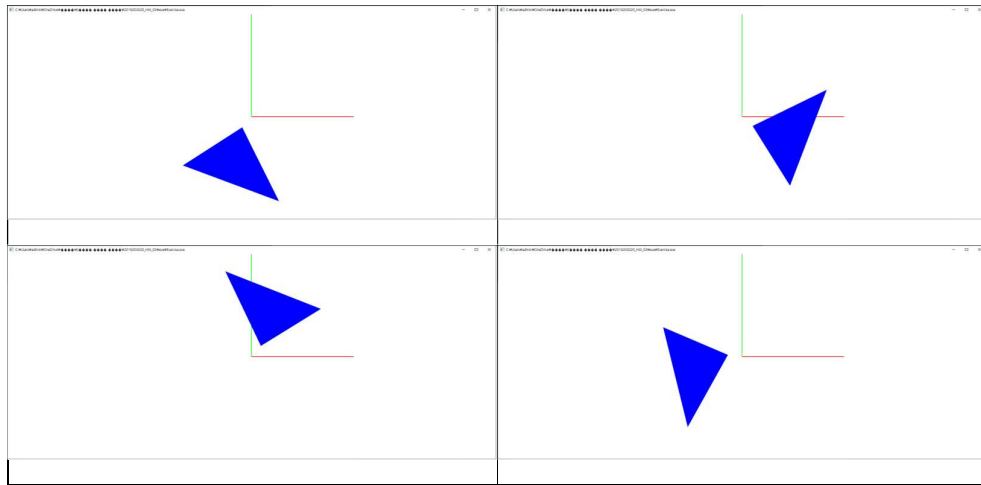
Exercise 01. Draw a triangle by clicking

1. Draw a triangle & dashed lines



2. Rotate the triangle

스페이스 바를 키보드로부터 입력 받아 rotation 진행



3. 코드

1. 포인트 설정, 포인트 개수

```
//endpoints of the three line segments to convert triangle
float point[3][2] = { {0,0},{0,0},{0,0} };//
float line_point[2][2] = { {0,0},{0,0} };//

int point_num = 0;

//mouse input mode
enum class InputMode {
    NONE = 0,
    CLICKED = 1, //
    COMPLETE = 2,
};
```

마우스를 사용해 삼각형과 선분을 그리기 위해 정점과 관련된 2차원 배열을 각각 설정해둔다. point[3][2]는 삼각형의 3개의 꼭짓점의 x, y 좌표를 저장하기 위한 2차원 배열이고, line_point[2][2]는 선분의 x, y 좌표를 저장하기 위한 2차원 배열이다. 그리고 마우스로부터 몇 개의 점을 찍었는지 알아보기 위한 지표로 point_num을 선언하고 초기값으로 0을 설정한다.

2. Main 함수에서의 로테이션 설정

```
if (elapsed > timeStep) {
    if (!pause && inputMode == InputMode::COMPLETE) {
        float delta_theta = float(2.0 * M_PI) / period * elapsed;

        rotate(point[0], delta_theta);
        rotate(point[1], delta_theta);
        rotate(point[2], delta_theta);
    }

    elapsed = 0;
}
```

elapsed 값이 timeStep 값보다 커지고 inputMode가 COMPLETE이 되었을 경우 rotation 애니메이션을 실행하도록 한다. 이때 키보드로부터 spacebar가 입력되어

pause 의 값이 바뀌면서 rotation 이 재생 및 정지를 반복하게된다. 마우스를 통해 그려진 도형은 삼각형이므로 3 개의 정점이 존재하게 된다. 이 3 개의 정점 모두 rotation 이 적용되어야 하므로 point[0], point[1], point[2] 모두에게 rotate 함수를 delta_theta 값만큼 회전하도록 적용한다.

3. 마우스 버튼 설정, 마우스 움직임 설정

```
//
void mouseButton(GLFWwindow* window, int button, int action, int mods) {
    double xs, ys;
    glfwGetCursorPos(window, &xs, &ys);

    float xw, yw;
    screen2world((float)xs, (float)ys, xw, yw);

    if (action == GLFW_RELEASE && button == GLFW_MOUSE_BUTTON_LEFT) {
        if (point_num < 3)
        {
            inputMode = InputMode::CLICKED;
            line_point[0][0] = xw; line_point[0][1] = yw;
            line_point[1][0] = xw; line_point[1][1] = yw;

            point[point_num][0] = line_point[0][0];
            point[point_num][1] = line_point[0][1];
            point_num += 1;

            if (point_num == 3) inputMode = InputMode::COMPLETE;
        }

    }
    else if (action == GLFW_PRESS && button == GLFW_MOUSE_BUTTON_LEFT) {
        if (point_num == 3) point_num = 0;
    }
}

//
void mouseMove(GLFWwindow* window, double x, double y) {
    if (inputMode == InputMode::CLICKED) {
        screen2world((float)x, (float)y, line_point[1][0], line_point[1][1]);
    }
}
```

마우스가 클릭함에 따라 정점이 찍힌다. 이때 정점 즉, point_num 에 따라 마우스의 움직임 및 버튼이 달라진다. point_num 의 개수가 3 보다 작다면 정점이 1 혹은 2 개라는 의미이다. 이때 line_point 에 정점 2 개를 저장해 stipple 패턴이 나타나는 선분을 그리도록 한다. 그리고 point 에 클릭된 점을 저장한다. 클릭될 때마다 point_num 을 1 씩 증가시켜 point 가 가리키는 점을 바꿔주도록 한다. point_num 이 3 이되면 이는 즉 클릭된 점의 개수가 3 이라는 의미이므로 삼 각형이 완성됨을 알 수 있다. 이때 inputMode 를 COMPLETE 으로 바꿔준다.

point_num 이 3 개가 되었을 시점에 다시 클릭한다면 삼각형이 사라지고 처음과정이 재시작 되도록 한다. 이를 위해 클릭된 순간 point_num 을 0 으로해 점의 개수를 0 으로 초기화 시켜 준다.

4. 렌더 함수

```

if (inputMode > InputMode::NONE) {
    if (point_num == 1) {
        glEnable(GL_LINE_STIPPLE);
        glLineStipple(int(3 * dpiScaling), 0xcccc);
        glBegin(GL_LINES);
        glVertex2f(line_point[0][0], line_point[0][1]);
        glVertex2f(line_point[1][0], line_point[1][1]);
        glEnd();
    }
    else if (point_num == 2) {
        glBegin(GL_LINES);
        glVertex2f(point[0][0], point[0][1]);
        glVertex2f(point[1][0], point[1][1]);
        glEnd();

        glEnable(GL_LINE_STIPPLE);
        glLineStipple(int(3 * dpiScaling), 0xcccc);
        glBegin(GL_LINES);
        glVertex2f(line_point[0][0], line_point[0][1]);
        glVertex2f(line_point[1][0], line_point[1][1]);
        glEnd();
    }
    else if (point_num == 3) {
        glBegin(GL_TRIANGLES); // triangle??
        glVertex2f(point[0][0], point[0][1]);
        glVertex2f(point[1][0], point[1][1]);
        glVertex2f(point[2][0], point[2][1]);
        glEnd();
    }
}

```

렌더 함수를 통해서 windows 창에 점, 선, 삼각형이 그려진다.

이때 point_num 이 1 이라면 클릭된 점이 1 개이고 창에는 stipple 패턴이 나타나는 선분이 그려진다. 선분의 두 점은 line_point 에 저장된 점으로 나타나고 한 점은 클릭된 점, 다른 한 점은 마우스가 움직이고 있는 점을 나타낸다.

point_num 이 2 라면 클릭된 점이 2 개라는 의미이다. 이때 선분 1 개와 그 선분의 끝점에서부터 마우스와 이어지는 stipple 패턴의 선분이 나타나게 된다. 앞서 찍어두었던 2 개의 정점은 point 에 저장되어있어 연속된 선분으로 나타난다. 그리고 point_num 이 1 이었던 경우와 마찬가지로 line_point 에 저장된 두 점, 즉 선분의 한 끝점과 마우스가 움직이고 있는 점을 통해 stipple 패턴이 나타나는 선분이 그려진다.

point_num 이 3 이라면 정점이 모두 찍힌 상태이다. 이때 point 에 저장된 3 개의 점을 정점으로 해 삼각형이 그려지게 된다.