**기초 컴퓨터 그래픽스**

**HW1 README**

학번 20181360 이름 함형원

**1. [환경 명세]**

본인 프로그램의 실제 구동 환경을 명시할 것 (OS, CPU, GPU, Complier 등)

* Windows 11 Education 64bit, AMD RYGEN 5 3600 6-Core, NVIDIA GeForce RTX 2060 SUPER, Visual Studio 2022 Debug x64

**2. [요구사항]**

(a)

- 구현 여부 : 예

- 작동 확인 방법 : 프로그램을 실행하면 윈도우가 화면에 뜬다.

- 구현 방법 : 예제 프로그램과 동일하다.

(b)

- 구현 여부 : 예

- 작동 확인 방법 : 프로그램을 실행하면 생성되는 윈도우에 나타난다.

- 구현 방법 : 예제 프로그램의 initialize\_renderer 함수에서 px, py, qx, qy의 값을 수정했다.

(c)

- 구현 여부 : 예

- 작동 확인 방법 : 프로그램을 실행하면 생성되는 윈도우에 나타난다.

- 구현 방법 : 전역 변수로 n\_object\_points(다각형의 점의 개수), object[6][2](다각형의 좌표), object\_center\_x, object\_center\_y(다각형의 무게중심의 좌표)를 선언했다. initialize\_renderer 함수에서 sq\_cx, sq\_cy, sq\_side를 선언하여 다각형의 좌표를 설정하고, 다각형의 무게중심의 좌표를 계산했다. 예제 프로그램의 draw\_object 함수에서 for 반복문의 반복 횟수를 6으로 수정했다.

(d)

- 구현 여부 : 예

- 작동 확인 방법 : 마우스 휠을 위 또는 아래로 움직인다.

- 구현 방법 : mousewheel 함수를 선언했다. 먼저, T(-px, -py)에 해당하는 계산을 통해 qx, qy의 좌표를 이동한다. 다음으로, R(dir \* TO\_RADIAN)에 해당하는 계산을 통해 qx, qy의 좌표를 이동한다. 마지막으로, T(px, py)에 해당하는 계산을 통해 qx, qy의 좌표를 이동한다. register\_callbacks 함수에 glutMouseWheelFunc 함수를 호출했다. glutPostRedisplay를 호출하여 화면을 업데이트한다.

(e)

- 구현 여부 : 예

- 작동 확인 방법 : shift 키를 누른 상태에서 선분의 파란 점을 마우스를 왼쪽 클릭하여 이동한다.

- 구현 방법 : 전역 변수로 picking을 선언하여 0으로 초기화하고, picking이 1이면 기능이 활성화되며, 0이면 비활성화된다. mousepress 함수에서 if 조건문을 구성했다. shift 키를 누른 상태에서 마우스를 왼쪽 클릭하면 if 조건문을 만족한다. glutGetModifier 함수를 통해 modifier 키를 누른 상태인지 확인할 수 있다. shift, alt ctrl 키가 modifier 키에 해당하며, 각각 GLUT\_ACTIVE\_SHIFT, GLUT\_ACTIVE\_ALT, GLUT\_ACTIVE\_CTRL와 비교하여 상태를 확인할 수 있다. 마우스로 정확하게 선분의 파란 점을 클릭하기 어려우므로, px, py와 mousepress의 x, y가 opengl 좌표 기준으로 0.01보다 작으면 picking을 1로 할당한다. 이외에는 picking을 0으로 할당한다. mousemove에서 picking을 사용한 if 조건문을 구성하여 mousemove의 x, y를 opengl 좌표로 변환하여 px, py를 이동하고, glutPostRedisplay를 호출하여 화면을 업데이트한다.

(f)

- 구현 여부 : 예

- 작동 확인 방법 : alt 키를 누른 상태에서 마우스를 오른쪽 클릭하여 이동한다.

- 구현 방법 : 전역 변수로 translation을 선언하여 0으로 초기화하고, translation이 1이면 기능이 활성화되며, 0이면 비활성화된다. 또한, 전역 변수로 prev\_x, prev\_y를 선언하여 각각 0으로 초기화하고, 마우스의 이전 좌표를 저장한다. mousepress 함수에서 if 조건문을 구성했다. alt 키를 누른 상태에서 마우스를 오른쪽 클릭하면 if 조건문을 만족한다. glutGetModifier 함수와 GLUT\_ACTIVE\_ALT를 비교하여 상태를 확인할 수 있다. 만족하면 translation을 1로 할당한다. 이외에는 translation을 0으로 할당한다. mousemove에서 translation을 사용한 else if 조건문을 구성한다. 먼저, del\_x, del\_y를 선언하여, 현재 좌표와 이전 좌표의 차를 저장한다. 다음으로, 다각형의 무게중심의 좌표를 0으로 초기화한다. for 반복문을 사용하여 다각형의 좌표를 이동하고, 다각형의 무게중심의 좌표를 계산한다. opengl 좌표로 통일하기 위해 del\_x, del\_y를 250.0f로 나누어 계산했다. del\_y의 부호가 음수인 것은 윈도우 좌표를 고려했기 때문이다. 마지막으로, prev\_x, prev\_y에 현재 좌표를 저장하고, glutPostRedisplay를 호출하여 화면을 업데이트한다.

(g)

- 구현 여부 : 예

- 작동 확인 방법 : ctrl 키를 누른 상태에서 마우스를 오른쪽 클릭하여 왼쪽 또는 오른쪽으로 이동한다.

- 구현 방법 : 전역 변수로 scaling을 선언하여 0으로 초기화하고, scaling이 1이면 기능이 활성화되며, 0이면 비활성화된다. 또한, 이전 좌표를 저장하기 위해 (f)에서 선언한 prev\_x, prev\_y를 사용한다.. mousepress 함수에서 if 조건문을 구성했다. ctrl 키를 누른 상태에서 마우스를 오른쪽 클릭하면 if 조건문을 만족한다. glutGetModifier와 GLUT\_ACTIVE\_CTRL를 비교하여 상태를 확인할 수 있다. 만족하면 scaling을 1로 할당한다. 이외에는 scaling을 0으로 할당한다. mousemove에서 scaling을 사용한 else if 조건문을 구성한다. 먼저, del\_x를 선언하여, 현재 좌표와 이전 좌표의 차를 저장한다. 다음으로, for 반복문을 사용하여 다각형의 좌표를 이동한다. T(-object\_center\_x, -object\_center\_y)에 해당하는 계산을 통해 다각형의 좌표를 이동한다. S(1 + del\_x / 100.0f)에 해당하는 계산을 통해 다각형의 좌표를 이동한다. T(object\_center\_x, object\_center\_y)에 해당하는 계산을 통해 다각형의 좌표를 이동한다. 적절하게 확대 또는 축소하기 위해 del\_x를 100.0f로 나누어 계산했다. 마지막으로, prev\_x에 현재 좌표를 저장하고, glutPostRedisplay를 호출하여 화면을 업데이트한다.

(h)

- 구현 여부 : 예

- 작동 확인 방법 : ‘t’ 키를 눌러 삼각형을 x 축 또는 y 축 반사한다.

- 구현 방법 : 전역 변수 reflection을 선언하여 0으로 초기화한다. 또한, 전역 변수로 n\_triangle\_points(삼각형의 점의 개수), trianglect[3][2](삼각형의 좌표), triangle\_center\_x, triangle\_center\_y(삼각형의 무게중심의 좌표)를 선언했다. initialize\_renderer 함수에서 tr\_cx, tr\_cy, tr\_side를 선언하여 삼각형의 좌표를 설정하고, 삼각형의 무게중심의 좌표를 계산했다. draw\_triangle 함수를 선언하여 윈도우에 삼각형이 그려지도록 설정했다. display 함수에 draw\_triangle 함수를 호출했다. keyboard 함수에서 switch 조건문을 구성했다. switch 조건문에서 if 조건문을 구성하여 ‘t’ 키를 눌렀을 때, scaling이 1이면 삼각형이 x 축 반사되며, 0이면 y 축 반사된다. glutPostRedisplay를 호출하여 화면을 업데이트한다. scaling이 1이면 x 축 반사에 해당하는 계산을 통해 삼각형의 좌표를 이동하고, 삼각형의 무게중심의 좌표를 계산한다. 다음으로, scaling을 0으로 할당한다. scaling이 0이면 y 축 반사에 해당하는 계산을 통해 삼각형의 좌표를 이동하고, 삼각형의 무게중심의 좌표를 계산한다. 다음으로, scaling을 1로 할당한다.