**기초 컴퓨터 그래픽스**

**HW1 README**

20181536 엄석훈

**1. [환경 명세]**

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

* Windows 11 Education, i5-1240P, 내장그래픽, Visual Studio 2022

**2. [요구사항]**

(a) 윈도우 화면

- 구현 여부 : 예

- 작동 확인 방법 : 프로그램을 실행하면 윈도우가 화면에 뜬다.

- 구현 방법 : main 함수에서 glutInitWindowSize()를 이용해서 화면의 크기를 가로 750, 세로 750이 되도록 설정한다.

(b) 선분 그리기

- 구현 여부 : 예

- 작동 확인 방법 : 프로그램을 실행하면 윈도우에 선분이 화면 우상단에 그려진다.

- 구현 방법 : initialize\_render()함수에서 px, py, qx, qy에 선분의 끝 점의 좌표를 설정하고 display()함수에서 draw\_line()함수를 호출해서 선분을 그려준다.

(c) 비대칭 다각형 그리기

- 구현 여부 : 예

- 작동 확인 방법 : 프로그램을 실행하면 윈도우에 비대칭 다각형이 화면 좌하단에 그려진다.

- 구현 방법 : initialize\_render()함수에서 다각형을 구성하는 점의 좌표를 설정하고 display()함수에서 draw\_object()함수를 호출해서 다각형을 그려준다.

(d) 선분 회전

- 구현 여부 : 예

- 작동 확인 방법 : 마우스 휠을 위쪽으로 스크롤 하면 선분의 파란색 꼭지점을 기준으로 선분이 반시계 방향으로 회전하며, 마우스 휠을 아래쪽으로 스크롤 하면 선분의 파란색 꼭지점을 기준으로 선분이 시계 방향으로 회전한다.

- 구현 방법 : register\_callbacks()함수에서 glutMouseWheelFunc()을 mousewheel()함수를 파라미터로 하여 호출한다. mousewheel()함수에서는 아핀변환의 translation을 통해 파란색 꼭지점이 원점에 오도록 선분을 이동시킨 뒤, 아핀변환의 rotation 공식을 이용해 direction의 방향대로 선분의 각 꼭지점을 회전시킨 뒤, 다시 아핀변환의 translation 공식을 통해서 처음 위치로 다시 이동시킨다. 그리고 glutPostRedisplay()함수를 통해 화면을 다시 그려준다.

(e) 파란색 꼭지점 이동

- 구현 여부 : 예

- 작동 확인 방법 : SHIFT키를 누른 상태에서 파란색 꼭지점을 마우스 왼쪽 버튼으로 클릭하고 마우스를 이동하면 선분의 파란색 꼭지점이 마우스를 따라서 이동한다.

- 구현 방법 : mousepress()함수에서 왼쪽 마우스 버튼이 눌리면 leftbuttonpressed변수를 1로 set하고 떼지면 0으로 reset해준다. 그리고 마우스가 클릭된 위치를 윈도우 좌표계에서 glut좌표계로 바꿔주고 파란색 꼭지점의 위치와 오차가 작다면 center\_selected변수를 1로 set해준다. 그리고 마우스가 이동해 mousemove()함수가 호출되었을 때 leftbuttonpressed와 center\_selected변수가 1로 set되어 있고 SHIFT키가 눌려 있는 상태라면 파란색 점의 좌표를 이동한 마우스 위치만큼 del\_x와 del\_y를 계산해서 파란색 꼭지점을 이동시켜주고 glutPostRedisplay()함수를 호출해 이동한 좌표를 토대로 화면을 다시 그려준다.

(f) 다각형 이동

- 구현 여부 : 예

- 작동 확인 방법 : ALT키를 누른 상태에서 마우스 오른쪽 버튼을 클릭하고 마우스를 움직이면 마우스의 움직임과 같이 다각형이 같은 속도와 방향으로 이동한다.

- 구현 방법 : mousepress()함수에서 오른쪽 마우스 버튼이 눌리면 rightbuttonpressed변수를 1로 set하고 떼지면 0으로 reset한다. 그리고 마우스가 이동해 mousemove()함수가 호출되었을 때 ALT키가 눌려있는 상태라면 마우스가 이동한 위치만큼 del\_x와 del\_y를 계산해서 다각형의 각 점을 이동시켜주고 glutPostRedisplay()함수를 호출해 이동한 좌표를 토대로 화면을 다시 그려준다.

(g) 다각형 크기 조절

- 구현 여부 : 예

- 작동 확인 방법 : CTRL키를 누른 상태에서 마우스 오른쪽 버튼을 클릭하고 마우스를 왼쪽 방향으로 이동하면 다각형이 작아지고, 마우스를 오른쪽으로 이동하면 다각형이 커진다.

- 구현 방법 : mousepress()함수에서 오른쪽 마우스 버튼이 눌리면 rightbuttonpressed변수를 1로 set하고 떼지면 0으로 reset한다. 그리고 마우스가 이동해 mousemove()함수가 호출되었을 때 CTRL키가 눌려있는 상태라면 마우스가 이동한 위치만큼 del\_x를 계산한다. 그리고 다각형을 중심이 원점에 오도록 아핀변환을 통해 translation해준 뒤 del\_x가 양수라면 아핀변환의 scaling을 해 크기를 키워주고 del\_x가 음수라면 scaling을 통해 크기를 줄여준다. 그리고 다시 아핀변환을 통해 다각형의 중심을 원래의 위치가 되도록 모든 점을 translation을 해준다.

(h) 추가 구현

- 구현 여부 : 예

- 작동 확인 방법 : 프로그램을 실행하면 별모양이 윈도우의 우상단에 그려진다. SHIFT키를 누른 상태에서 마우스 오른쪽 버튼을 클릭하고 마우스를 움직이면 마우스의 움직임과 같이 별이 같은 속도와 방향으로 이동한다. 별이 이동하면서 현재의 별 모양과 똑같이 잔상을 남기고 그 잔상은 시간이 지나면서 제자리에서 회전하면서 크기가 작아지다가 사라진다.

다음으로 키보드에서 ‘1’키를 누르면 상하좌우로 별이 반사되어 나타나며 시간이 지날수록 회전하며 크기가 작아지다 사라진다. ‘2’키를 누르면 대각선 방향으로 별이 반사되어 나타나며 시간이 지날수록 회전하며 크기가 작아지다 사라진다. ‘3’키를 누르면 상하좌우, 대각선 방향으로 별이 반사되어 나타나며 시간이 지날수록 회전하며 크기가 작아지다 사라진다.

마지막으로 키보드에서 x키를 누르면 별이 양의 우측 방향으로 찌그러지고, z키를 누르면 별이 음의 좌측 방향으로 찌그러지고, y키를 누르면 별이 위쪽 방향으로 찌그러지고, t키를 누르면 별이 아래쪽 방향으로 찌그러진다.

- 구현 방법 : 먼저 STAR라는 구조체를 만들어 별을 그릴 수 있도록 해준다. 그리고 make\_star()함수를 통해 화면에 나타날 별을 그려준다. 그리고 initialize\_render()함수에서 make\_star()함수를 호출하게 하고 display()함수에 draw\_star함수를 포함시켜 화면에 그려지도록 한다.

다음으로 mousepress()함수에서 오른쪽 마우스 버튼이 눌리면 rightbuttonpressed변수를 1로 set하고 떼지면 0으로 reset한다. 그리고 마우스가 이동해 mousemove()함수가 호출되었을 때 SHIFT키가 눌려있는 상태라면 마우스가 이동한 위치만큼 del\_x와 del\_y를 계산해서 별의 각 점을 아핀변환의 transtaion을 이용해 이동시켜준다. 그리고 마우스가 움직일 때 3번에 한번 꼴로 make\_star\_copy()함수를 호출해준다. make\_star\_copy()함수를 호출하면 현재의 별의 모양과 크기를 그대로 복사해서 새로운 별을 하나 만들어주고 display()함수에 draw\_star\_copy()함수를 호출하도록 해준 뒤 glutPostRedisplay()함수를 호출해 이동한 좌표를 토대로 화면을 다시 그려준다.

그리고 keyboard()함수에서 1, 2, 3키가 눌리면 star\_reflect()함수를 호출해준다. star\_reflect()함수는 키보드에서 어떤 키가 눌렸는지 flag로 받아들여 각 flag에 따라서 별을 아핀변환 translation을 이용해 x축 y축에 접하도록 이동시킨 뒤, 아핀변환의 reflection을 이용해서 x축 또는 y축 또는 y=x직선 또는 y=-x직선에 반사시킨 뒤 반사된 점을 star\_copy2에 저장해서 새로운 잔상을 저장해준다. 그리고 display()함수에 draw\_star\_copy2()함수를 호출하다록 해준 뒤 glutPostRedisplay()함수를 호출해 새로운 별의 잔상을 화면을 다시 그려준다.

그리고 잔상 별이 생기도록 한 뒤 register\_callbacks()함수 내부에 glutTimerFunc()을 추가해서 매 30ms마다 timer()함수를 호출하고 timer()함수는 매번 star\_copy\_rotate()함수를 호출하도록 해준다. star\_copy\_rotate()함수는 star\_copy와 star\_copy2의 잔상 별들이 매 30ms마다 아핀변환의 translation을 통해 원점으로 이동 후, rotation해주고 다시 translation해서 별의 중심을 기준으로 회전하면서 scaling을 통해 별의 크기까지 동시에 줄여준다. 그리고 별이 일정 횟수이상 회전하게 되면 해당 별의 state를 -1로 바꿔 별을 소멸시킨다.

마지막으로 keyboard()함수에서 z, x, t, y키가 눌리면 star\_shearing()함수를 호출해준다. 어떤 키가 눌렸는지 flag로 받은 뒤 star\_shearing함수에서 별을 원점으로 translation시킨 뒤 아핀변환의 shearing을 이용해서 원하는 방향으로 물체를 찌그러트리고 다시 translation시켜 별을 x, y축 방향으로 찌그려트린다.