**기초 컴퓨터 그래픽스**

**HW3 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]**

1. 물체의 배치 및 움직임 구현

(a)

구현 여부: O

호랑이가 sun temp 내부에서 이동한다. 호랑이는 한 주기에 360도 회전 > -y 방향 이동 > +x 방향 이동 > -y, -z 방향 이동 > -y 방향 이동 > -x 방향 이동 > +y 방향 이동 > +y, +z 방향 이동 > -x 방향 이동 > +y 방향으로 이동한다.

(b)

구현 여부: O

animateFlag를 선언하여 호랑이의 애니메이션을 제어한다. animateFlag가 1이면 호랑이가 이동하며, 0이면 정지한다. 정지는 호랑이의 좌표, 프레임을 고정시키는 것을 의미한다. animateFlag는 0으로 설정되며, ‘l(L)’ 키를 눌러 animateFlag를 on/off할 수 있다.

(c)

구현 여부: O

dynamic objects로 ben, wolf를 사용한다. ben은 x, y = 0, 0 근처에 위치한다. ben은 한 주기에 큰 원 360도 회전 > 벽 올라가기(+z 방향 이동) > +y 방향 이동 > 작은 원 180도 회전 > 벽 내려가기(-z 방향 이동) > +y 방향 이동 > 큰 원 180도 회전으로 이동한다. wolf는 x, y, z = 0, -4000, 1500 근처에 위치한다. wolf는 한 주기에 -y 방향 이동 > +y 방향 이동 > sun temp 벽 한 바퀴 회전 > 반지름이 150인 원 360도 회전으로 이동한다.

(d)

구현 여부: O

tank 2대, bus 2대, bike 1대를 사용하며, x, y, z = 0, -4500, 1500 근처에 위치한다. 프로그램을 실행하면 tank 2대, bus 2대를 볼 수 있다. bike 1대는 ‘a’ 키를 누르고 직접 찾거나, ‘u’ 키를 누르고 줌-아웃을 하여 쉽게 찾을 수 있다. ‘a’, ‘u’ 키에 대한 설명은 2번 문제에서 하겠다.

2. 가상 카메라의 배치 및 조절 기능 구현

(a)

구현 여부: ⅰ: O, ⅱ: O

‘u’, ‘i(I)’, ‘o’, ‘p’ 키를 눌러 카메라를 설정할 수 있다. 이 상태에서 CTRL 키를 누르고, 마우스 휠을 움직이면 줌-인/줌-아웃을 할 수 있다. 마우스 휠을 올리면 줌-인, 내리면 줌-아웃을 할 수 있다. zoomFlag를 선언하여 줌-인/줌-아웃을 제어할 수 있다. zoomFlag는 0으로 초기화되며, ‘u’, ‘i’, ‘o’, ‘p’, ‘a’, ‘e’ 키를 누르면 on이 된다. ‘t’, ‘g’키를 누르면 off가 된다. Camera\_u는 (d)에서 배치한 물체를 볼 수 있다. Camera\_i, Camera\_o는 sun temp을 아래에서 위로 관찰한다. Camera\_p는 석상의 뒤를 관찰한다.

(b)

구현 여부: ⅰ: O, ⅱ: O, ⅲ: O, ⅳ: O

‘a’키를 눌러 카메라를 설정할 수 있다. Camera\_a는 석상의 앞을 관찰한다. 방향키 위쪽을 누르면 -n 방향으로 이동하며, 아래쪽을 누르면 +n 방향으로 이동한다. 방향키 오른쪽을 누르면 +u 방향으로 이동하며, 왼쪽을 누르면 -u 방향으로 이동한다. ‘n’키를 누르면 +v 방향으로 이동하며, ‘m’키를 누르면 -v 방향으로 이동한다.

마우스를 x축으로 이동하면 v에 대해 회전한다. 마우스가 움직이는 방향으로 회전한다. 마우스를 y축으로 이동하면 u에 대해 회전한다. 마우스가 움직이는 방향으로 회전한다. ‘j’, ‘l’키를 누르면 n에 대해 회전한다. dynamicFlag를 선언하여 이동, 회전을 제어할 수 있다. dynamicFlag는 0으로 초기화되며, ‘a’키를 누르면 on이 된다. 나머지 카메라에 대해 off가 된다.

(a)와 마찬가지로, CTRL 키를 누르고, 마우스 휠을 움직이면 줌-인/줌-아웃을 할 수 있다. 마우스 휠을 올리면 줌-인, 내리면 줌-아웃을 할 수 있다.

(c)

구현 여부: ⅰ: O, ⅱ: O

‘t’키를 눌러 호랑이의 눈으로 세상을 관찰할 수 있다. viewFlag를 선언하여 이를 제어할 수 있다. viewFlag는 0으로 초기화되며, ‘t’키를 누르면 on이 된다. 나머지 카메라에 대해 off가 된다. 고개를 끄덕이는 효과가 있다.

(d)

‘g’키를 눌러 호랑이를 뒤에서 관찰할 수 있다. observeFlag를 선언하여 이를 제어할 수 있다. observeFlag는 0으로 초기화되며, ‘t’키를 누르면 on이 된다. 나머지 카메라에 대해 off가 된다.

3. 추가 기능 구현

구현 여부: O

speedFlag를 선언하여 n 방향으로 이동할 때 fovy를 제어한다. speedFlag는 0으로 초기화되며, ‘q’키를 눌러 on/off가 가능하다. on 상태에서 -n 방향으로 이동하면 fovy는 증가하며, +n 방향으로 이동하면 fovy는 감소한다. fovy는 최대 3까지 증가한다.

displayFlag를 선언하여 새로운 wolf를 표시한다. displayFlag는 0으로 초기화되며, ‘w’키를 눌러 on/off가 가능하다. on 상태에서 새로운 wolf가 표시되며, 변수가 갱신된다. off 상태에서 새로운 wolf의 변수는 갱신되지 않는다.

새로운 wolf는 x, y = 0, 0 근처에 위치한다. 새로운 wolf는 한 주기에 나선 운동(1080도 회전, +z 방향 이동) > -y 방향 이동 > +x, -y 방향(45도) 이동 > -x, -y 방향(45도) 이동 > +x, -y 방향(45도) 이동 > 나선 운동(1080도 회전, -z 방향 이동) > +y 방향으로 이동한다.

observeWolfFlag를 선언하여 새로운 wolf를 관찰하는 카메라를 제어한다. observeWolfFlag는 0으로 초기화되며, ‘e’키를 누르면 on이 된다. 나머지 카메라에 대해 off가 된다. on 상태에서 새로운 wolf를 관찰하는 카메라로 전환한다. 또한, zoomFlag가 활성화되어 동일한 명령으로 줌-인/줌-아웃을 할 수 있다.