**기초 컴퓨터 그래픽스**

**HW3 README**

20181625 남현준

1. **프로그램 작성 및 실행환경**

본 HW 3의 작성 및 구동은 Windows 10 64bit, AMD Ryzen 5 5600X, NVIDIA GeForce RTX 3070, Visual Studio Release x64에서 진행되었습니다.

1. **HW3 구현사항**
   1. **물체의 배치 및 움직임 구현**
      1. 움직이는 호랑이
         * + 구현여부 : 예
           + 확인방법 : 프로그램이 실행된 다음, 실행되는 윈도우 상에 배치된 물체들과 기본적으로 주어진 bistro\_exterior, skybox가 화면 상에 출력된다.
           + 구현 방법 :

텍스트, 전자제품, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Bistro\_exterior와 함께 주어진 3D\_Objects\_Anim 파일에서 필요한 함수들과 호랑이에 관련된 함수들을 가져와서 구현했다.

Read\_geometry\_20181625 함수로 모델링에 필요한 정보를 읽어들이고, draw\_tiger, prepare\_tiger 함수를 이용해서 화면 상에 호랑이 모델링을 출력하게 된다. 이 때, 모델링은 wire-frame 형식이어야 하기 때문에 해당 구현을 위해 glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_LINE)을 이용해 wire-frame 형식으로 출력하고 오브젝트의 출력이 끝났다면 다시 glPolygonMode(GL\_FRONT\_AND\_BACK, GL\_FILL)로 set하였다.

* + 1. 호랑이 물체의 움직임 제어
       - * 구현여부 : 예
         * 확인방법 : 프로그램이 실행된 다음, 키보드의 특정 값이 입력으로 주어지면 호랑이의 움직임의 멈춤을 조절할 수 있다.
         * 구현방법 : ‘s’가 입력으로 들어오게 될 경우, 호랑이가 움직이거나 멈추게 된다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

* + 1. 추가 동적 오브젝트들의 구현
       - * 구현여부 : 예
         * 확인방법 : 프로그램의 뷰에서 호랑이를 빼고 2개의 추가적인 오브젝트들이 동적으로 움직이는 것을 확인할 수 있을 것이다.
         * 구현방법 :

스크린샷, 디스플레이, 패턴, 다채로움이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Wolf object의 경우, 하트와 비슷한 모양으로 움직일 수 있도록 설정했다.

스크린샷, 디스플레이, 텍스트, 다채로움이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

spider object의 경우, 사용한 bistro\_exterior에서의 길의 일정 부분을 따라서 움직이도록 경로를 설정한 뒤, timer function을 통해 시간이 지남에 따라 움직임을 화면 상에서 보일 수 있도록 하였다.

* + 1. 추가 정적 오브젝트들의 구현
       - * 구현여부 : 예
         * 확인방법 : 가상의 세상에 정적 오브젝트들이 구현되어 있다. 세상 이동 카메라를 통해 가상의 세상을 돌아다니며 확인할 수 있다.
         * 구현방법 : 정적 object들의 파일 정보를 가져온 뒤, 원하는 위치에 배치될 수 있도록 ModelViewMatrix를 설정하여 해당 위치에 object들이 출력될 수 있도록 하였다.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

해당 코드는, 프로그램에 배치된 옵티머스 오브젝트의 코드 예시이다. 다른 오브젝트들도 이와 유사하게, ModelViewMatrix의 설정을 통해 특정 위치에 배치되어 있다.

* 1. **가상 카메라의 배치 및 조절 기능 구현**
     1. 세상 관찰 카메라
        + - 구현여부 : 예
          - 확인방법 : u, I, o, p 입력이 키보드를 통해 주어질 때, 프로그램 상에서 가상의 세상을 관찰하는 카메라에 변화가 생길 것이다.
          - 구현방법 :

텍스트, 스크린샷, 멀티미디어 소프트웨어, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

해당 코드는 u의 입력이 들어올 때의 카메라 설정값에 대한 코드이다. 위의 코드와 비슷하게 카메라를 설정할 때 pos (카메라의 위치), uaxis, vaxis, naxis의 값을 수정하여 카메라가 원하는 위치에서, 원하는 방향을 향하도록 설정해주었다.

* + 1. 세상 이동 카메라
       - * 구현여부 : 예
         * 확인방법 : ‘a’가 입력으로 들어올 때, 세상 관찰 카메라에서 세상 이동 카메라로 변경된다.
         * 구현방법 :

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명

세상 관찰 카메라와 마찬가지로 카메라의 pos, uaxis, vaxis, naxis의 값을 수정하여 원하는 위치, 원하는 방향을 향하도록 설정한다. 카메라의 move factor를 1로 설정하여, 카메라가 이동할 수 있도록 설정한다.

* + 1. 카메라의 이동 및 줌-인/줌-아웃
       - * 구현여부 : 예
         * 확인방법 : 특정 동작을 통해 카메라가 이동하여 프로그램 상에 보이는 뷰의 갱신이 일어난다. 줌-인/줌-아웃을 담당하는 동작이 생길 때, 화면이 클로즈업되거나 멀어지게 된 다음 뷰의 갱신이 생겨 프로그램 상에 출력된다.
         * 구현방법 :

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

상하좌우의 방향키로 카메라가 보는 시점을 기준으로 x, y축의 방향으로 움직이도록 구현했다. 위, 아래 방향키는 y방향으로 +, -로 움직이도록 하였고 좌, 우 방향키는 x방향으로 +, -로 움직이도록 구현했다.

z축에 대한 이동의 경우, page up과 page down키를 활용하여 page up은 +z방향, page down은 -z방향으로 움직이도록 구현했다.

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

카메라의 줌-인과 줌-아웃은 ctrl + 마우스 휠의 움직임으로 구현했다. ctrl키와 휠이 입력으로 들어오면, 해당 방향에 맞는 동작을 하도록 했다. 휠이 위로 움직이면 줌-인, 아래로 움직이면 줌-아웃 하도록 구현했다.

* + 1. 카메라의 회전
       - * 구현여부 : 예
         * 확인방법 : 마우스의 작동에 따라 카메라가 특정 축을 기준으로 회전했을 때의 뷰가 프로그램 상에 출력된다.
         * 구현방법 :

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

마우스의 입력에 따라 다른 축을 기준으로 카메라가 회전할 수 있도록 했다. 좌클릭이 주어질 때, 회전 축을 카메라의 y축, 즉 vaxis에 대해 회전을 진행하도록 하여 좌우 방향으로의 회전이 마우스의 움직임에 따라 일어나도록 하였다. 카메라의 시점을 회전시킨 후, vaxis vector를 제외한 나머지 vector들 (uaxis, naxis)의 값을 갱신해주어 카메라의 시점에 변화가 생긴 후, 해당 변화를 유지할 수 있게 하였다.

우클릭이 입력으로 주어질 때, y방향으로의 시점 갱신이 일어날 수 있도록 설정했다. 회전 축을 카메라의 x축, 즉 uaxis를 기준으로 회전을 진행하도록 하여 상하 방향으로의 회전이 일어나도록 했다. 마찬가지로 카메라를 회전시킨 후, 나머지 vector인 vaxis, naxis vector들의 값 또한 갱신해주어서 해당 변화를 유지할 수 있도록 했다.

마우스 휠 부분을 눌렀을 때, 카메라의 z축, 즉 naxis를 기준으로 회전이 일어나도록 했다. 마찬가지로 카메라를 회전시킨 후, 나머지 vector인 vaxis, uaxis vector들의 값 또한 갱신해주어서 해당 변화를 유지할 수 있도록 했다.

* + 1. 호랑이 관점 카메라
       - * 구현여부 : 예
         * 확인방법 : ‘t’를 누르고 있으면 호랑이의 눈 시점에서 바라보는 풍경이 프로그램 상에 출력된다.
         * 구현방법 :

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

T의 입력이 주어질 때, ViewMatrix를 갱신하여 뷰가 계속 바뀌게 된다. 이 때, ViewMatrix에는 tiger object를 정의했을 때 사용한 ViewMatrix의 역행렬이 계산값으로 들어가게 된다. 호랑이의 고개 움직임을 구현하기 위해, rotation\_angle\_tiger의 값에 따라 translate하는 z의 값에 차이를 주어, 시점이 갱신되는 동안 위 아래의 움직임이 생기도록 설정했다.

* + 1. 호랑이 관찰 카메라
       - * 구현여부 : 예
         * 확인방법 : ‘g’를 누르고 있으면 호랑이를 뒤에서 따라가는 시점의 카메라가 바라보는 풍경이 프로그램 상에 출력된다.
         * 구현방법 :

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

G의 입력이 주어질 때, ViewMatrix를 지속적으로 갱신하게 된다. ViewMatrix에는 tiger object의 위치를 설정할 때 사용된 행렬들의 역행렬이 들어가게 된다.