**기초 컴퓨터 그래픽스**

**HW5 README**

학번 20171687 이름 전민수

**1. [환경 명세]**

1) 본인 프로그램의 실제 구동 환경을 명시 할 것 (OS, CPU, GPU, Compiler 등등)

- ex 1) window10 64bit, AMD Ryzen 5600X, rtx 3070, visual studio 2019 – win 32, debug

**2. [요구사항]**

   1. Modeling Transformation

1) ) 자신이 임의로 선정한 3D 물체 (텍스처 매핑 미적용)에 대하여 Gouraud shading을 적용하라. (어떤 물체인지 명확히 기술하고 시각적으로 Gouraud shading 효과가 분명히 보이도록 하라.)

- 확인 방법:

용에 Gouraud shading을 적용하였다.

5번을 눌러 빛을 조종할 수 있게 바꾼 후, 용에 빛을 가져다 두면, gouraud shading효과가 발생했음을 알 수 있다.

(커맨드 : 5 1번 누르고 d 두번 누르고 e 꾹누르기.)

2) 자신이 임의로 선정한 또 다른 3D 물체 (텍스처 매핑 미적용)에 대하여 Phong shading을 적 용하라. (어떤 물체인지 명확히 기술하고 시각적으로 Phong shading 효과가 분명히 보이도록 하라.)

적용 물체 : Godzilla

확인 방법 :

5번을 눌러 빛을 조종할 수 있게 바꾼 후, 고질라를 빛에 가져다 두면, 빛을 smoothe하게 반사하는 고질라를 볼 수 있다.

(커맨드 : 5 1번 누르고, w로 고질라 근처로 빛을 이동 후, a누르기, e를 꾹 누르면 가까이에서 고질라를 볼 수 있다.)

3) 자신이 임의로 선정한 3D 물체 (정적인 물체와 동적인 물체 각각 한 개씩)에 대하여 웹에서 공개적으로 구할 수 있는 영상 (또는 자신이 촬영한 영상)을 사용하여 텍스춰를 입혀라. (호랑이와 같이 예제 코드에서 사용한 물체는 제외함.)

정적 물체에 입힌 텍스쳐 :



이 텍스쳐를 옵티머스 객체에 입혔다.

입힌 모습이다.



확인 방법 :

5번을 눌러 s를 꾹누른다면 확인이 가능하다.

동적 물체에 입힌 텍스쳐 :

포유류, 실외, 고슴도치, 너구리이(가) 표시된 사진

자동 생성된 설명

늑대가 좀 더 늑대처럼 보였으면 하는 바람에 넣었다.

입힌 모습이다.



확인 방법 :

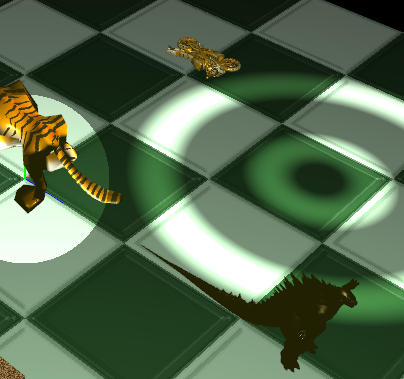
옵티머스에서 카메라를 조금 더 옮겨서 확인이 가능하다.

5를 처음 눌렀을 때, s를 누르다가 늑대 근처에서 a를 누르면 된다.

4. 세상 좌표계에 최소한 한 개 이상의 광원을 배치하라. (이 광원은 ‘w’ 키를 사용하여 동시에 on/off 할 수 있어야 한다.)

세상좌표계에 고정되어있는 광원을 하나 배치하였다.

5번을 누르면 바로 보이는 반짝이고 있는 광원이 그것이다.

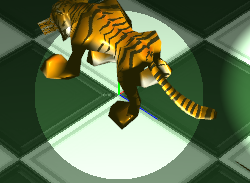


w에는 이미 카메라를 조작하는 것을 쓰고 있기 때문에, 0을 누르면 껐다 킬 수 있다.

5. 움직이는 카메라 (카메라는 마우스 또는 키보드를 사용하여 조작)에 대하여 눈 좌표계를 기 준으로 (즉 카메라에 상대적으로 고정되어 움직이는) 광원을 한 개 구현하라. (이 광원은 ‘e’ 키를 사용하여 on/off 하며, 눈 좌표계에서 설정한 광원임을 분명히 알 수 있어야 한다.)

5번 카메라를 따라다니는 눈 좌표계 광원을 하나 구현하였다.

5번을 누르면 카메라 중앙에 스폿조명이 비춰진다.

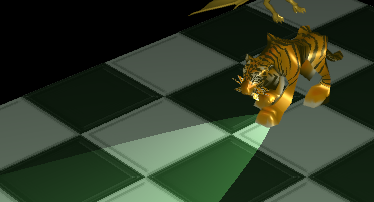


=을 누르면 껐다 킬 수 있다.

6. 움직이는 물체에 고정된 광원, 즉 그 물체의 모델링 좌표계에 배치한 광원을 한 개 구현하라. (이 광원은 ‘m’ 키를 사용하여 on/off 하며, 모델링 좌표계에서 설정한 광원임을 분명히 알 수 있어야 한다.)

움직이는 호랑이에 고정된 광원을 하나 구현하였다.

호랑이가 움직임에 따라 스폿조명이 같이 움직이는 것을 관찰할 수 있다.



이 광원은 -를 눌러서 껐다 켤 수 있다. 호랑이는 p를 눌러서 움직일 수 있다.

7. ‘스크린 효과’ 또는 ’블라인드 광원’과 같은 재미있는 쉐이더 효과를 최대 3개까지 구현하라. (당연히 이 두 효과와는 전혀 다른 자신만의 창의력을 발휘한 3D effect이어야 하며, 무엇을 대략적으로 어떻게 구현했는지 REAMDE 파일에 기술하라.)

1. 움직이는 빛

블라인드 효과를 보고, 이 블라인드 효과가 움직이면 멋있겠다 라는 생각을 하였다.

이는 shader에 타이머 변수를 하나 더 넣어줌으로써 구현이 가능했다.

tmp\_float = cos(30.0f\*acos(tmp\_float)+30.0f\*((timer/5)%4));

코드로 설명을 하자면, 블라인드 효과를 내기 위해 곱하는 주기를 좁히는 인자 말고도, 그 뒤에 타이머를 통해 주기를 조정해준다면, 시간에 따라 바뀌는 빛의 모양을 볼 수 있다.

이는 세상 좌표계에 고정된 빛을 가지고 구현하였으며, 5번을 누르면 바로 볼 수 있다

1. 거리에 따라 색이 변하는 늑대

늑대의 색은 가장 가까울 때에는 빨간색, 거리가 조금 멀어졌을 때에는 파란색, 좀 더 멀어졌을 때에는 초록색을 띈다. 너무 멀면 원래 색을 띈다.

거리는 카메라와의 거리이다. 따라서 카메라를 바꿔가며 본다면, 늑대의 색이 카메라의 영역으로 들어왔을 때 바뀌는 것을 관찰 할 수 있는데,

1번 카메라로 두고, 늑대를 움직인다면(1을 누르고 o를 누름)

늑대가 원형궤도를 돌며 카메라에 가까워짐에 따라 파란색에서 빨간색이 되었다가 파란색으로 바뀐다.

2번 카메라로 두고, 늑대를 움직인다면(2을 누르고 o를 누름)

늑대가 원형궤도를 돌며 카메라에 가까워짐에 따라 초록색에서 파란색이 되었다가 초록색으로 바뀐다.

3번 카메라로 두고, 늑대를 움직인다면(3을 누르고 o를 누름)

늑대가 원형궤도를 돌며 카메라에 가까워짐에 따라 파란색에서 빨간색이 되었다가 파란색으로 바뀐다.

4번 카메라로 두고, 늑대를 움직인다면(4을 누르고 o를 누름)

늑대가 원형궤도를 돌며 카메라에 가까워짐에 따라 원래 색인 하얀색이었다가 초록색이 되었다가 하얀색으로 바뀐다.

이는 카메라와 눈 좌표계 사이의 거리를 length(v\_position\_EC.xyz) 를 통해 구한 후, if문으로 색을 정해주고, 원래 색과 mix를 해주었다.

1. 흙 같은 큐브

큐브의 원래 색에 돌이 박힌듯한 효과를 줌으로써, 흙큐브 같은 느낌을 주고 싶었다.

구현은 우선 큐브를 열심히 만든다.

GLfloat cube\_vertices[72][3]

큐브를 그려준 후, 각 점에 대해 하나의 랜덤 값을 뽑아야하는데,

float random (vec2 st) {

return fract(sin(dot(st.xy,

vec2(12.9898,78.233)))\*

43758.5453123);

}

다음과 같은 랜덤 비스무리한 함수를 사용하였다.

이후, vec3(random(st)),1.0의 색은 흰색~검은색의 색 중 하나일텐데, 이 색과 큐브의 색을 mix하여 돌이 박힌듯한 효과를 만들어보았다.

실행하고 1번을 누르면 바로 확인이 가능하다.