

BuildYourOwnScene_OpenGL

Scene 을 openGL 을 이용하여 구현.

[실행 환경]

Ubuntu 18.04

OpenGL version string: 2.1 Mesa 20.0.8

```
sudo apt-get install g++
```

```
sudo apt-get install freeglut3-dev
```

```
sudo apt-get install mesa-utils
```

```
sudo apt install libglm-dev
```

[실행 방법]

```
g++ -o ./scene ./scene.cpp -lGL -lglut -lm -lGLU  
./scene
```

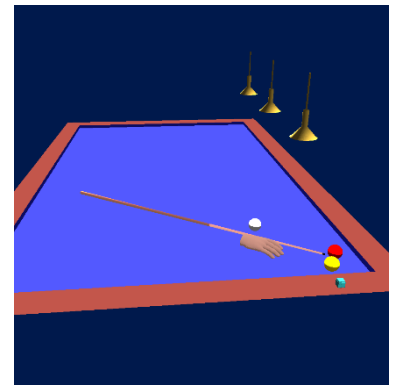
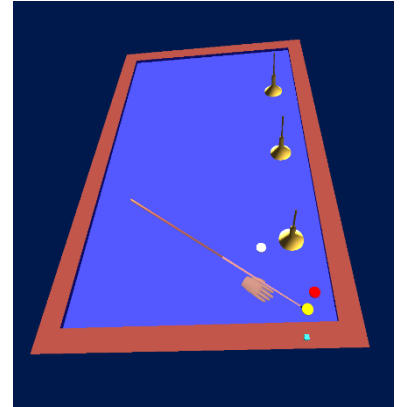
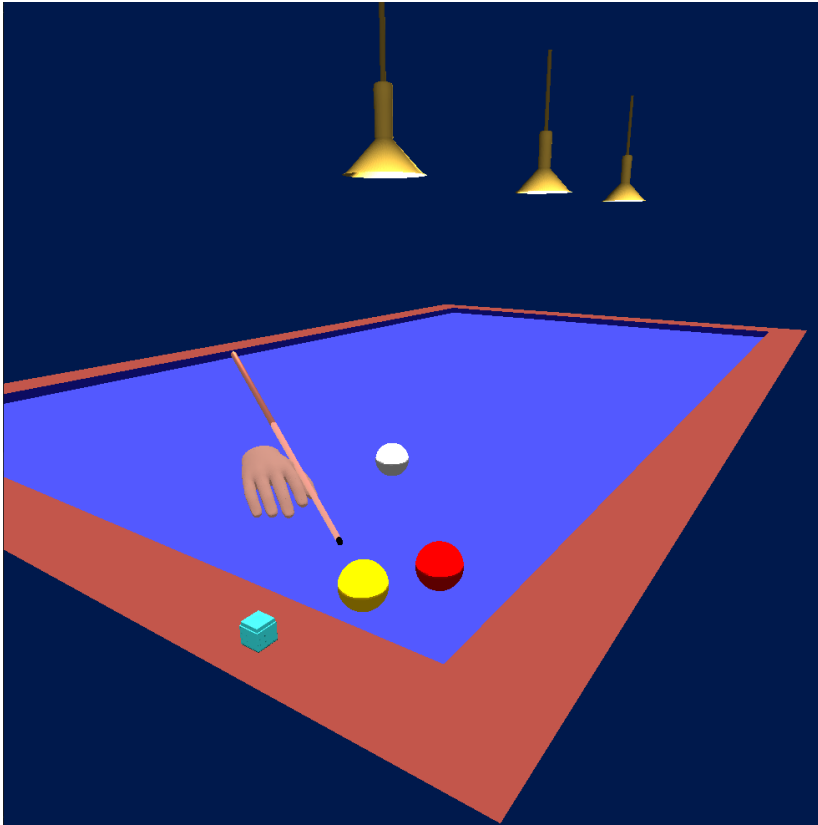
Drag 하여 rotate

Shift 누른 채로 drag 하여 translation

Ctrl 누른 채로 위아래로 drag 하여 zoom in / out

Alt 누른 채로 위아래로 drag 하여 dolly in / out

1. Picture of my own scene.



Reflection과 Sphere 하면 당구공이 떠올라 당구대를 만들어 보게 되었다.

Scene에는 polygon, implicit surface, swept surface, 외부 geometry file 모두가 포함되어 있다.

먼저 polygon으로는 푸른색 당구대 다이, 당구대 테두리 나무가 있다. GL_QUADS 를 이용하여 사각형으로 표현하였다.

다음으로 implicit surface로는 sphere로, 당구공 3개가 있다. theta 와 pi 를 동일한 각도로 나누어 표면을 cubic하게 mapping하여 아주 작은 사각형들로 나타냈다. 이는 과제 5에서 reflection을 구현할 때 필요할 것으로 예상된다.

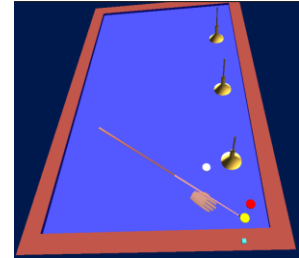
당구대 위쪽의 전등 3개, 당구 큐(당구 채)는 swept surface를 이용하여 구현하였다. 실제 광원도 전등과 동일한 위치에 두어 그림 에서와 같이 실제 전등처럼 표현될 수 있었다. 또한 당구 큐의 경우 swept surface를 3개로 나누어 구현하였다.

외부 geometry file로는 손 모양과 초크를 불러왔다. 두 obj file 모두 인터넷에서 (turbosquid.com, free3d.com) 무료로 다운받았다. Obj file에서 vertex 정보와 normal 법선 벡터, face 면 정보를 읽어온 후 직접 texture를 지정해 주었다.

2. Material

먼저 파란색 당구대 다이 material 은 ambient diffuse specular 모두 파란 빛이 잘 나오도록 설정해 주었으며, shininess도 실제 다이를 생각하며 적절히 주었다.

```
float blue[13] = { 0.04725, 0.0995, 0.5745, 1.0, /* ambient */  
                  0.05164, 0.00648, 0.82648, 1.0, /* diffuse */  
                  0.228281, 0.255802, 0.766065, 1.0, /* specular */  
                  30.0 /* shininess */  
};
```



다음으로는 당구 다이 테두리 나무이다. 앞선 다이와 비슷하게 나무 색상 RGB 값을 참고하여 세가지 빛에서 모두 나무 색이 잘 나오도록 설정하였다. 반짝거리는 나무 당구대 테두리를 생각하며 shininess는 높게 주었지만 opengl 낮은 버전에서 잘 표현되지는 않는 것 같았다.

```
float wood[13] = { 0.2588, 0.1137, 0.1, 1.0, /* ambient */  
                  0.2588, 0.1137, 0.1, 1.0, /* diffuse */  
                  0.2588, 0.1137, 0.1, 1.0, /* specular */  
                  80.0 /* shininess */  
};
```

전등 같은 경우에는 금색으로 구현하였으며 shininess도 높게 주어 금속 느낌을 내도록 하였다.

```
float gold[13] = { 0.24725, 0.1995, 0.0745, 1.0, /* ambient */  
                  0.75164, 0.60648, 0.22648, 1.0, /* diffuse */  
                  0.628281, 0.555802, 0.366065, 1.0, /* specular */  
                  50.0 /* shininess */  
};
```

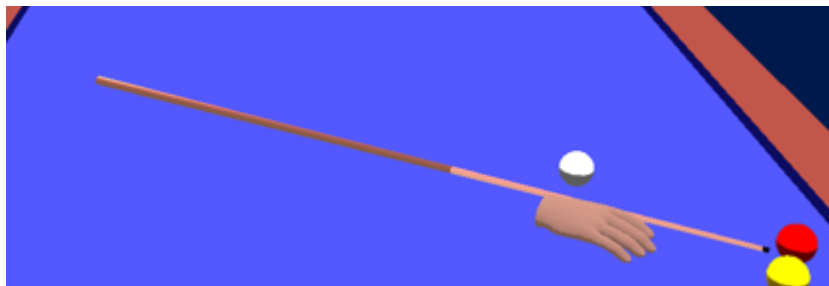


다음으로, 큐의 경우 swept surface 3개로 이루어져 있으며, 각각 밝은 갈색, 밝은 살색, 검은색을 지정해 주었다.

```
float gol[13] = { 0, 0, 0, 1.0, /* ambient */  
                 0, 0, 0, 1.0, /* diffuse */  
                 0, 0, 0, 1.0, /* specular */  
                 10.0
```

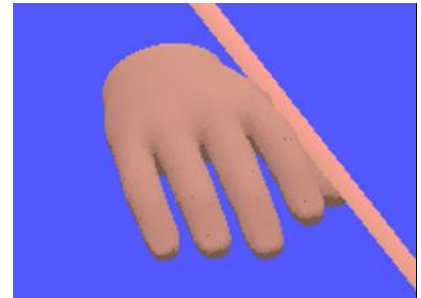
```
float gold[13] = { 0.9843, 0.6921, 0.5961, 1.0, /* am  
                 0.05164, 0.00648, 0.02648, 1.0, /* diffu  
                 0.000007843, 0.000003921, 0.000001961, 1.0,  
                 10.0 /* shini
```

```
float black[13] = { 0.5843, 0.3921, 0.2961, 1.0, /* a  
                  0.05164, 0.00648, 0.02648, 1.0, /* diffu  
                  0.000007843, 0.000003921, 0.000001961, 1.0,  
                  10.0 /* shini
```



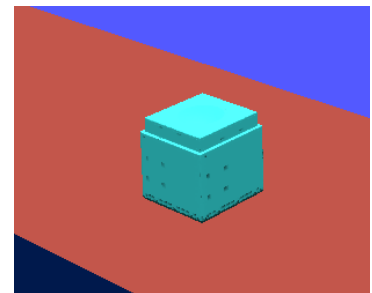
외부 obj file 중 hand 부분이다. 직접 texture를 지정하여 주었는데, 사람 피부의 rgb 색과 가장 비슷한 색을 골라 shininess를 0으로 주고, 인간 피부와 비슷하게 보이게 하기 위해 원 살색에 ambient 값을 0.6, diffuse와 specular 값에는 각각 0.2배를 해 주어 표현하였다.

```
float gold[13] = { 0.8915*0.6, 0.6547*0.6, 0.5664*0.6, 1.0, /* ambier
                  0.8915*0.2, 0.6547*0.2, 0.5664*0.2, 1.0, /* diffuse */
                  0.8515*0.2, 0.5547*0.2, 0.5664*0.2, 1.0, /* specular */
                  0.0 /* shininess */
};
```

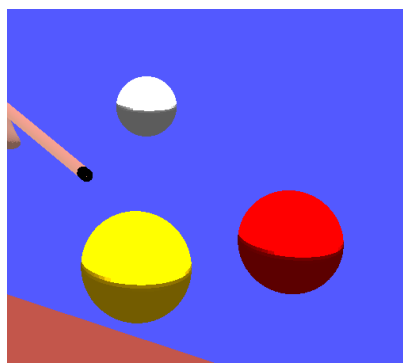


외부 obj file 중 chalk 부분이다. 그림자를 잘 드러내도록 하고 싶었지만 새 초크가 모델링이 되어있는 것인지 굴곡이 덜 패여 있어 그 부분은 아쉬웠다. 최대한 당구장의 초크 색과 비슷하게 하도록 했다.

```
= { 0.0543, 0.2221, 0.2261, 1.0, /* ambient */
    0.05164, 0.1648, 0.1648, 1.0, /* diffuse */
    0.0843, 0.36921, 0.36961, 1.0, /* specular */
    0.0 /* shininess */
```



당구공의 경우 세가지 색이 있으며, 각각 적절한 shininess 와 조명 바로 아래서도 색이 잘 나타날 수 있도록 각각 값을 잘 조절하였다.



3. Viewing and Lighting

먼저 조명 3개의 해당하는 위치에 실제 specular 조명을 각각 하나씩 총 3개를 위치시켰다. 그 결과 실제 조명과 같은 효과를 얻을 수 있었다.

```
GLfloat Specular1[] = { 0.3f, 0.3f, 0.3f, 0.0f };
GLfloat Position1[] = { 100.0f, 100.0f, 200.0f, 0.0f };
glLightfv( GL_LIGHT1, GL_DIFFUSE, Specular1 );
glLightfv( GL_LIGHT1, GL_POSITION, Position1 );
```



하지만 이 specular 3개만을 가지고는 물체들이 가진 색을 잘 보여주지 못해서 ambient light을 시선보다 z축으로 뒤 방향에 위치시켜 문제를 해결하였다.

```
GLfloat AmbientLightValue[] = {0.7f, 0.7f, 0.7f, 0.0f};
GLfloat PositionLightValue[] = { -10.0f, -10.0f, 200.0f, 0.0f };

glLightfv( GL_LIGHT0, GL_AMBIENT, AmbientLightValue );
glLightfv( GL_LIGHT0, GL_POSITION, PositionLightValue );
```

이전 과제에서 진행했던 viewpoint 이동의 경우에도 그대로 붙여 넣어 user가 viewpoint를 움직일 수 있도록 하였다.