

1. Object manipulation

HW5_skeleton code에서 시작했고, 큐브의 회전은 (q,w,e), (r,t,y)키를 눌렀을 때 되도록 구현했습니다.

2. Lighting

HW4의 Lighting을 그대로 가져왔습니다. directional light, point light, spotlight를 구현했습니다. default로 directional light가 켜져 있고, 각각 1, 2, 3키를 눌렀을 때 on/off됩니다. 여러 개의 light를 동시에 on하는 것도 가능합니다.

3. Texture

A. Map the textures to differentiate 9 cube

9개의 큐브에 각각 task1.bmp부터 task9.bmp까지의 texture를 입혔습니다. 이를 위해 texture[]와 textureID[][]의 크기를 조정해주었고, 이 texture들을 불러오는 과정은 sky_box까지 읽은 후에 진행하였습니다 (init_texture()).

B. Texture mapping with modified texture coordinate (except A)

init_texture_cube를 진행할 때 quad3 대신 quad4를 사용하도록 하였고, quad4함수에서 num에 따라 texture를 입히는 부분(좌표)을 달리하여 cube의 각 면마다 texture가 다르게 입혀지도록 하였습니다.

C. Apply two or more texture

bump mapping을 할 때 (program 2), 기존에 읽어온 task와 bump mapping이 동시에 이루어지도록 하였습니다.

D. Bump mapping

Lab #4에서 사용한 normal mapping을 사용하였습니다. 코드도 거의 동일합니다.

E. Cube mapping

sky-box를 불러오는 과정은 Lab #4와 동일하게 진행하였습니다. (main, Env~shader 모두 동일)

다만, floppy cube에 reflection대신 refraction을 적용하였습니다.

```
vec4 Kd = texture(myTextureSampler, UV);
```

```
//color = mix(Kd, texColor, 0.85).xyz;
```

```
float ratio = 1.00 / 1.52;  
  
vec3 I = normalize(Position - campos);  
  
vec3 R = refract(I, Normal, ratio);  
  
color = vec3(texture(cubemap, -R));  
  
color = mix(Kd, vec4(color,1.0), 0.85).xyz;
```

(EnvFragmentShader.glsl)

4. Creativity

A. Refraction model

refraction model을 사용했습니다.

B. object rotation

HW2에서 했던 것처럼 floppy cube 전체의 회전을 구현하였습니다. 키보드 O key를 이용하여 object change가 가능합니다.