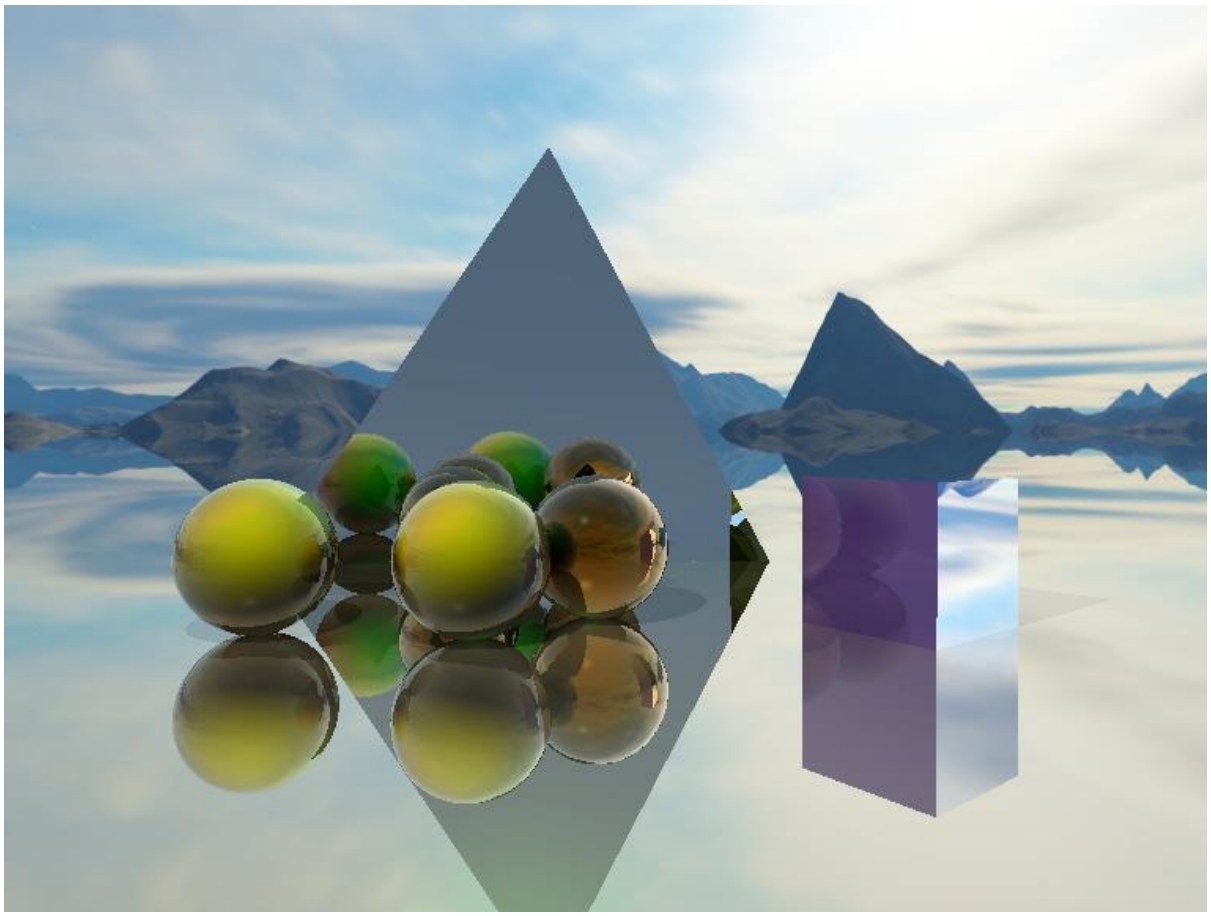
그래픽스 프로그래밍 과제 5 보고서

2022-25448

박기범



castRay 에서 getray로 우리 시점에서 보일 빛을 가져 오게 됩니다.

Castray 안의 for (int bounce = 0; bounce < 4; bounce++) 루프에서 여러 번의 트레이싱을 시도 합니다.

if(trace(ray,hit))에서 만약 빛이 물체에 닿을경우 레이트레이싱을 합니다.(아닐경우 col += (texture(environmentMap, tmp.xyz)).xyz * total_attenuation;를 통하여 하늘을 매핑합니다.)

for(int li = 0 ; li < lights.length(); li ++){ 이 안에서 모든 빛에 대하여 트레이싱을 하는데

if(trace(tmp_ray,tmp_hit) 까지 참일 경우 즉 관찰자 입장에서도 빛이 닿고, 광원에서도 빛이 닿을때에 한하여 빛을 더해주게 됩니다.

if(trace(tmp_ray2,tmp_hit2) && li == 0){ sh = 0.0;}
를 통하여 만약 만난 부분에서 레이트레이싱한 빛이 광원에 닿는다면 sh값을 1로 아니면 0으로 바꾸어

vec3 col_t = ambient + (diffuse + specular) * sh ;
col += col_t * total_attenuation * (vec3(1.0,1.0,1.0) - schlick(cosine, hit.mat.R0,hit.mat.material_type))
를 통하여 phong 라이팅을 구현해 줍니다.(col이 최종 아웃풋에 누적 됩니다.)

반사되지 않고 남은 부분의 빛만 현재 루프에서의 빛에 더해지게 됩니다.

이제 반사를 구현하기 위하여 반사된 빛을 구현 하여야 합니다.
total_attenuation *= schlick(cosine,hit.mat.R0,hit.mat.material_type) ;
를 통하여 다음 빛을 계산 하기 위하여 total_attenuation를 계산 합니다. total_attenuation이 잔여 빛으로 계산 되는데,

```
ray.origin = hit.p;
vec3 lightDir = ray.direction ;
vec3 reflectDir = reflect(lightDir, hit.normal);
ray.direction = normalize(reflectDir);
```

를 통하여 다음 빛의 방향과 오리진을 계산해주고 다음 루프로 들어 가게 됩니다.

또한 인바이로번트 매핑과 프레스넬까지 적용하여 리플렉션 계수를 변경 하였습니다