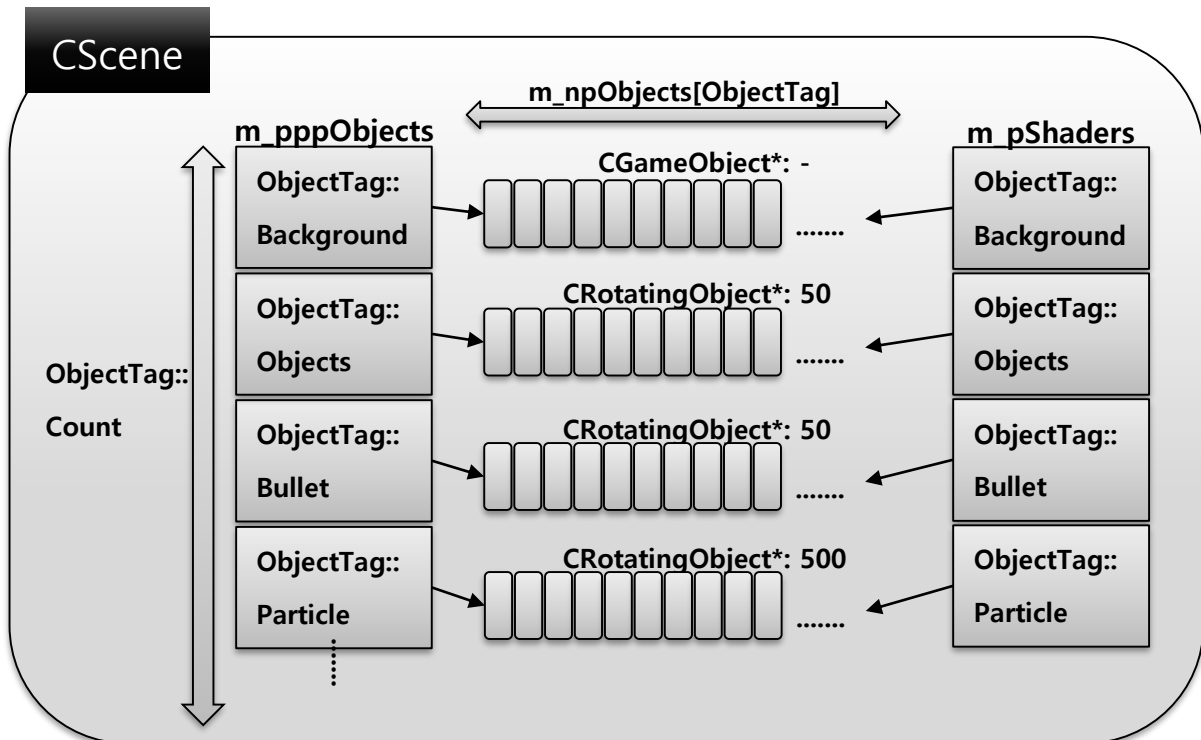


➤ [조작법]

키	설명
W	앞으로 이동
A	왼쪽으로 이동
S	뒤로 이동
D	오른쪽으로 이동
F2 / F3	자유 시점(플레이어를 그리지 않는다) / 플레이어 시점
M	해가 떠오르게 한다.
N	달이 떠오르게 한다.
P	플레이어의 위치를 미로의 입구로 이동한다.
F	손전등을 켜다
G	손전등을 끈다.
ESC	프로그램 종료

➤ [사용한 자료구조]



- 사용한 자료구조는 3D Game Programming 1 의 3 번 과제와 같다.
오브젝트의 3 차원 포인터 배열에 세로축엔 오브젝트의 종류, 가로축엔 같은 오브젝트의 개수를 담아서 같은 종류의 오브젝트들마다 하나의 인스턴싱 셰이더를 통해 animate 및 render 연산을 수행했다.

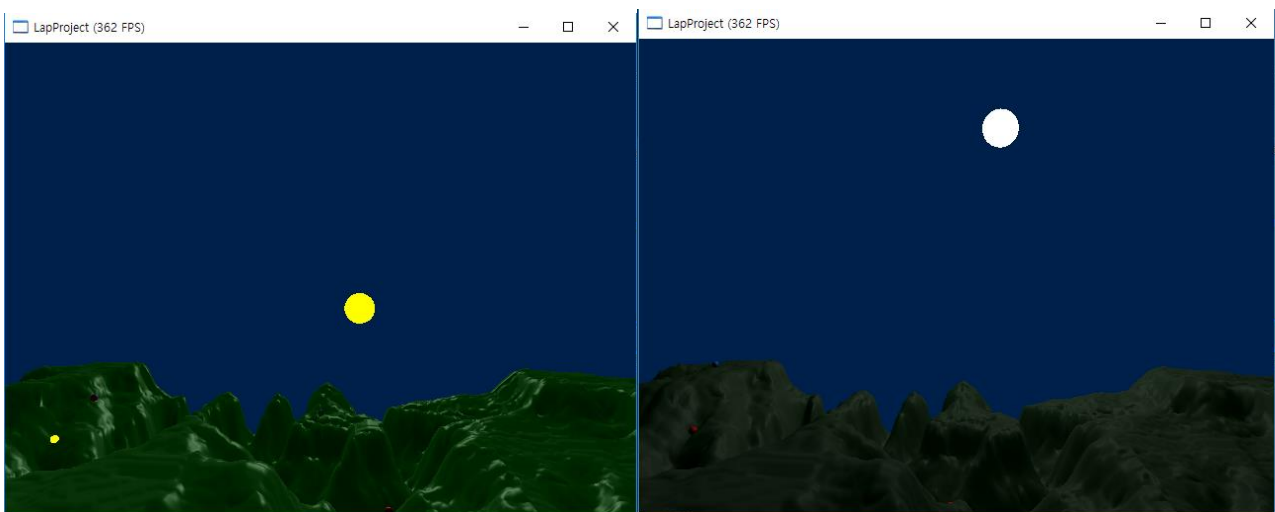
➤ [프로그램 요구사항]

1. 태양과 달이 있고, 지형 중심을 기준으로 공전하면서 가상공간에 낮과 밤을 만든다.

- 가정
 - ① 태양과 달은 구로 표현한다.
 - ② 태양과 directional light, 달은 point light 로 표현한다.
 - ③ 태양과 달은 180 도 반대편에 위치한다.

● 구현

- ① 태양과 달은 샘플 코드에 선언 되어있던 CRevolvingObject 를 사용했으며 mesh 로는 CSphereMeshIlluminated 를 사용했다.
- ② Material 의 Emissive 를 설정하여 구가 빛이 없어도 밝아 보이게 하였다.
- ③ 태양과 달의 공전은 CRevolvingObject 의 Animate 에서 처리하였다. 월드 matrix 의 41, 42, 43 부분에 terrain 의 중심 위치가 (800, 0, 800)이므로 $(1200 \cdot \cos + 800, 1200 \cdot \sin, 800)$ 으로 설정하고 cos, sin 의 세타값을 매 프레임마다 fTimeElapsed 를 누적시켜 CRevolvingObject 가 공전하게 만들었다.
- ④ 태양과 달의 cos, sin 세타의 초기값을 각각 0, 3.14 로 설정하여 태양과 달이 180 도 반대편에 위치하게 하였다.
- ⑤ m_pLights 의 1 번 2 번을 각각 directional light, point light 로 설정하고 scene 의 animate 에서 매 프레임마다 position 을 각각 CRevolvingObject 의 position 으로 갱신하면서 directional light 의 direction 을 $\text{normalize}((800, 0, 800) - \text{position})$ 로 구한 벡터로 설정하여 빛의 방향이 지형의 중심으로 가도록 하였다.
- ⑥ 태양과 달의 y 값이 0 보다 작으면 비활성화하고 0 보다 크면 활성화하여 낮과 밤을 나타냈다.



태양

달

2. 지형과 오브젝트들은 태양과 달빛을 받아서 빛 반사를 한다.

- 가정

- ① 플레이어를 제외한 오브젝트들의 메시를 모두 illuminated 로 만든다.
- ② 빛 계산은 cube mesh 의 정점이 러프하므로 hlsl 에서 픽셀라이팅으로 계산한다.
- ③ 샘플 코드에 나머지는 다 구현되어 있으므로 그대로 사용한다.

- 구현

- ① 루트 시그니처에 D3D12_ROOT_PARAMETER_TYPE_SRV 의 ShaderVisibility 를 D3D12_SHADER_VISIBILITY_ALL 로 변경하고 shader.hlsl 의 #define _WITH_VERTEX_LIGHTING 를 주석처리하여 정점라이팅에서 픽셀라이팅으로 변경한다.
- ② Cube 와 Sphere 의 mesh 는 샘플 코드에 이미 illuminated 가 구현되어있으므로 기존의 diffused mesh 를 이용하는 cube 와 sphere 오브젝트들을 illuminated cube, illuminated, sphere 로 변경한다.
- ③ terrain mesh 를 생성할 때 terrain image 의 GetHeightMapNormal 함수를 이용하여 기존의 diffused vertex 를 illuminated vertex 로 변경한다.

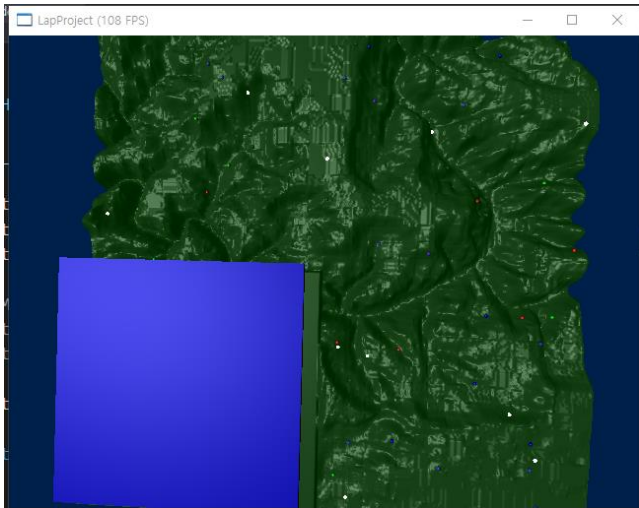
3. 키 조작으로 낮과 밤 상태를 조절할 수 있다.

- 가정

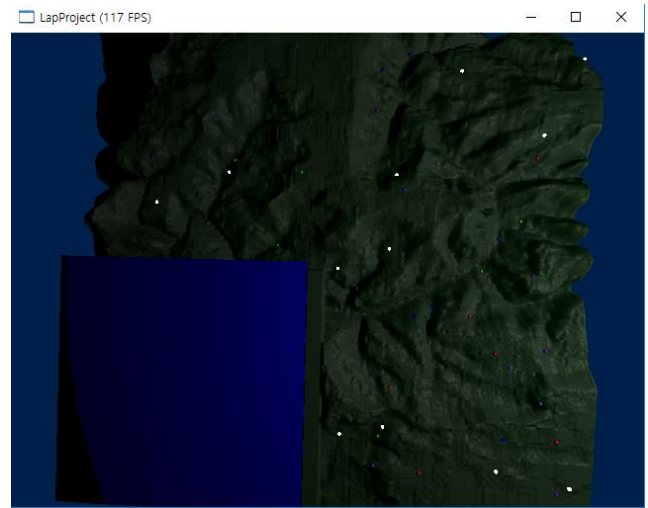
- ① M 키를 누르면 낮이 된다.
- ② N 키를 누르면 밤이 된다.

- 구현

- ① CRevolvingObject 의 누적되고있는 세타를 M 을 누르면 태양은 0.0, 달은 3.14 로 설정한다.
- ② CRevolvingObject 의 누적되고있는 세타를 N 을 누르면 태양은 3.14, 달은 0.0 로 설정한다.



낮



밤

4. 지형 위에 50 개 이상의 구들이 존재하고 각각의 구는 서로 다른 방향으로 이동하며 구끼리 충돌하면 튕겨나간다.

● 가정

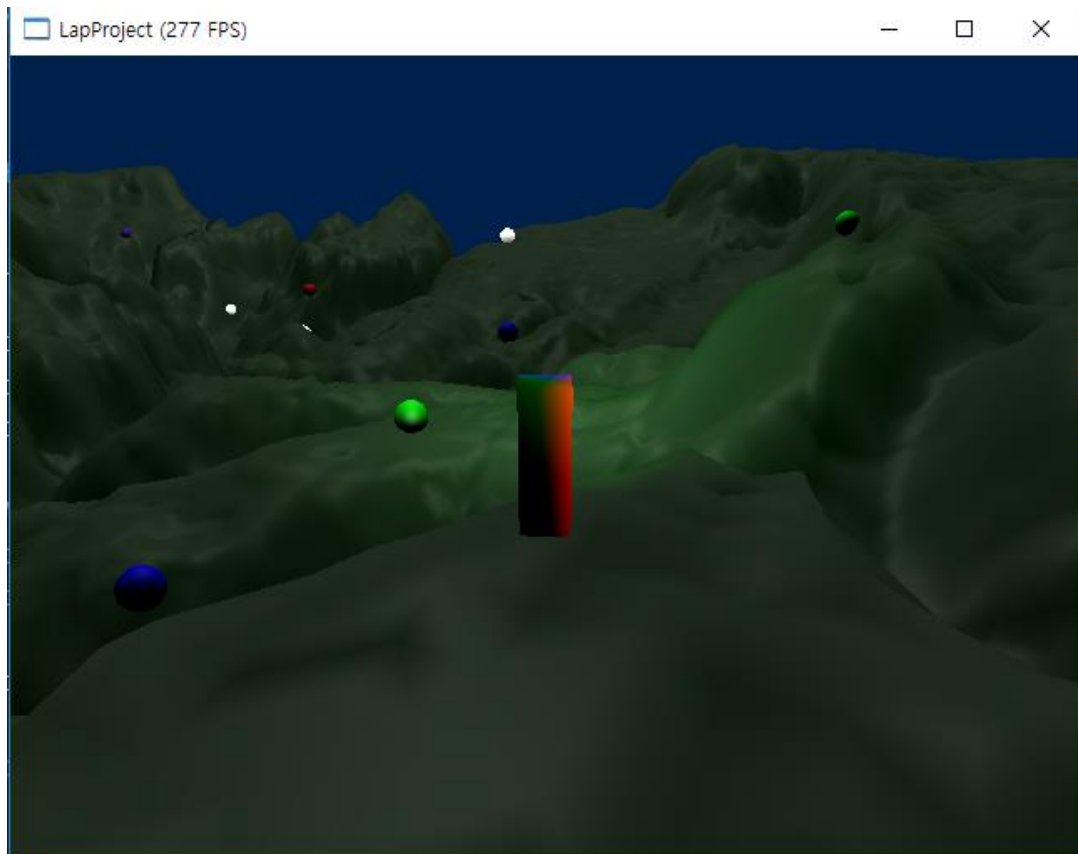
- ① 3D Game Programming 1 의 3 번 과제의 구들을 기반으로 한다.
- ② 미로 안에서 플레이어를 쫓아다니던 구들을 지형의 랜덤한 위치에서 생성되게 하고 플레이어를 쫓지 않게 한다.
- ③ 구들의 초기 이동 벡터는 랜덤이다.
- ④ 구들은 지형 밖으로 나가지 않고 경계에 도달하면 이동 벡터가 반사 방향으로 변경된다.
- ⑤ 구들은 실내 지형이 있는 곳에 들어갈 수 없고 경계에 도착하면 이동 벡터가 반사된다.
- ⑥ 구 끼리 충돌하면 방향 벡터가 충돌한 구와 반대 방향으로 변경된다.

● 구현

- ① 씬의 AnimateObjects 함수에서 구가 플레이어를 쫓게 만드는 부분과 실내 지형의 벽과 충돌하는 부분을 삭제한다.
- ② PhysicsProcessing 함수에서 충돌 후 후처리부분에 `setdirection (normalize(object->getposition() - collider->getposition()))` 또는 그 반대인

colliderPos - objectPos 를 추가하여 구끼리 충돌했을 때 충돌한 구의 반대 방향으로 이동 방향이 바뀌도록 한다.

- ③ 구가 지형의 좌우 경계에 도달하면 이동 벡터의 x 를 $-x$ 로 반전시키고 상하 경계에 도달하면 이동 벡터의 z 를 $-z$ 로 반전시킨다.
- ④ 구가 실내 지형이 존재하는 지형의 1/4 크기의 구역 경계의 위쪽에 도달하면 이동 벡터의 z 를 $-z$ 로 반전시키고 오른쪽에 도달하면 이동 벡터의 x 를 $-x$ 로 반전시켰다.
- ⑤ 구가 생성되는 위치는 $x: 800 \sim 1500$, $z: 800 \sim 1500$ 으로 설정하였고 이동 벡터도 xyz 를 모두 $-1000 \sim 1000$ 의 랜덤한 정수를 넣고 normalize 하여 설정하였다.
- ⑥ 구의 이동 벡터가 설정될 때 마다 구의 회전축을 이동 벡터와 $(0,1,0)$ 의 외적 벡터로 설정하여 구가 이동할 때 이동하는 방향으로 구르게 하였다.



굴러다니는 구들

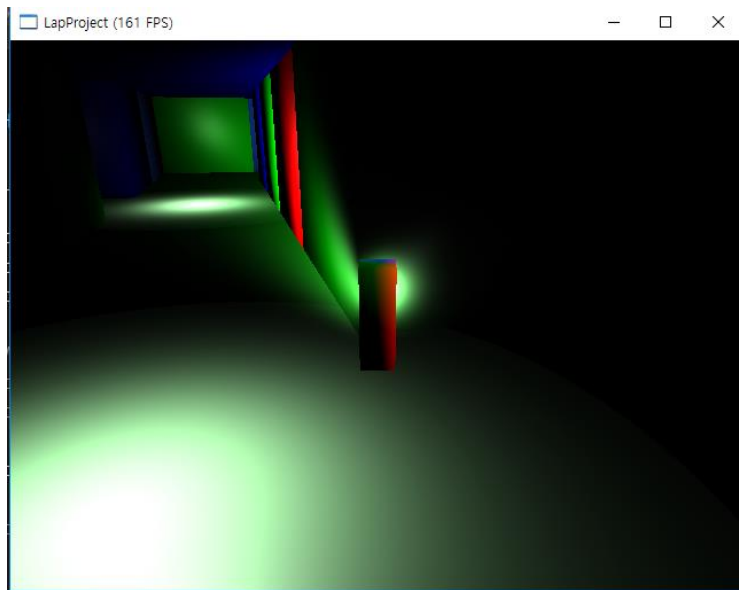
5. 지형 어느 위치에 실내 지형이 있고 실내 지형에 20 개 이상의 조명이 있으며 플레이어가 실내에 들어가면 조명이 활성화되고 실내에는 태양과 달이 영향을 줄 수 없다.

● 가정

- ① 실내 지형은 3D Game Programming 1 의 3 번 과제의 미로를 기반으로 한다.
- ② 태양, 달, 플레이어 손전등을 제외한 실내 지형의 조명은 27 개이며 미로의 pathes 위치에 일정 간격으로 존재한다.
- ③ 플레이어가 실내 지형으로 진입하면 27 개의 실내 조명이 활성화되고 태양과 달의 light 가 비활성화 된다.
- ④ 실내 조명은 spot light 만 사용한다.

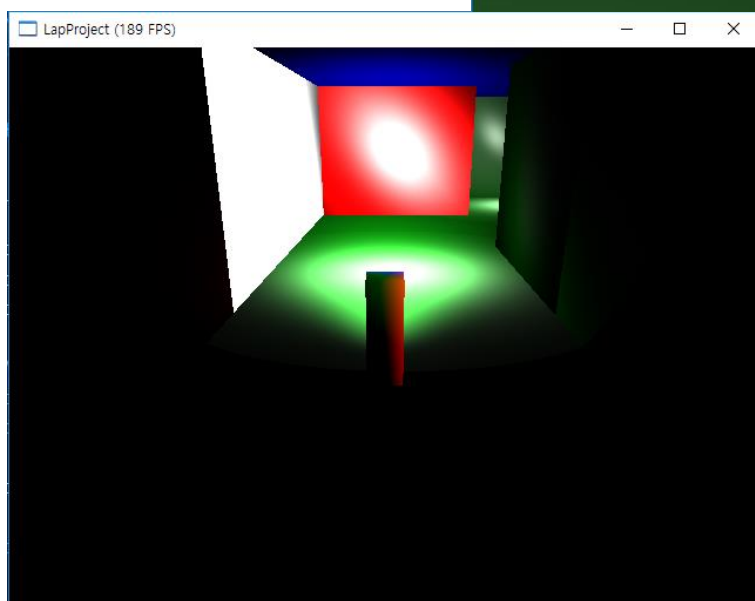
● 구현

- ① 미로 지붕의 크기가 지형의 1/4 이 되도록 terrain 의 scale 을 (6.5, 2, 6.5) 로 조정하고 terrain 의 1/4 의 높이 값을 100 으로 고정하여 미로가 위치한 지형을 약간 높게 평탄화 시켰다.
- ② 미로의 길 개수를 실내조명 개수로 나눈 값을 stride 로 두어 미로의 길 내부에서 일정간격마다 조명이 위치할 수 있도록 설정하였다.
- ③ 미로 지붕 크기를 기준으로 지붕의 x, z 범위 안에 플레이어가 들어오고 지붕의 y 보다 플레이어가 아래에 있다면 플레이어가 실내에 들어왔다고 판단을 하여 실내 조명을 모두 활성화하고 태양과 달을 비활성화 하였다.
- ④ 실내 조명은 미로의 길 천장 중심에 위치시켰고 0, -1, 0 방향으로 빛을 쏘도록 설정하였다.



실내 조명

실내 진입 전



실내 진입 후

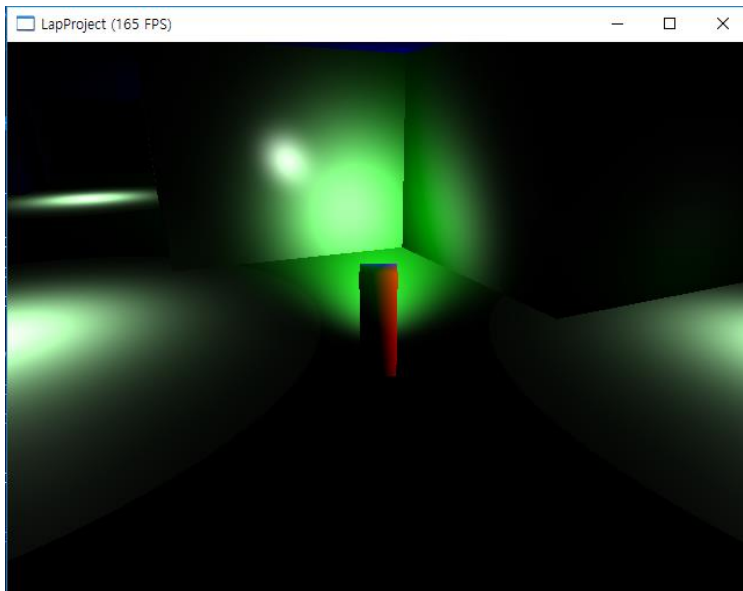
6. 플레이어는 손전등을 가지고 있으며 손전등은 키고 끄고 할 수 있다.

- 가정

- ① 샘플 코드에 있는 손전등을 그대로 사용하였다.
- ② F 키를 누르면 손전등이 활성화된다.
- ③ G 키를 누르면 손전등이 비활성화 된다.

- 구현

- ① F 키와 G 키로 손전등인 light[0]의 enable 플래그를 토글하도록 하였다.
- ② 손전등의 범위를 300 으로 늘리고 m_fPhi 를 70, m_fTheta 를 35 로 조정하여 손전등의 밝기와 퍼지는 정도를 더 크게 하였다.



손전등 켜

손전등 끄

