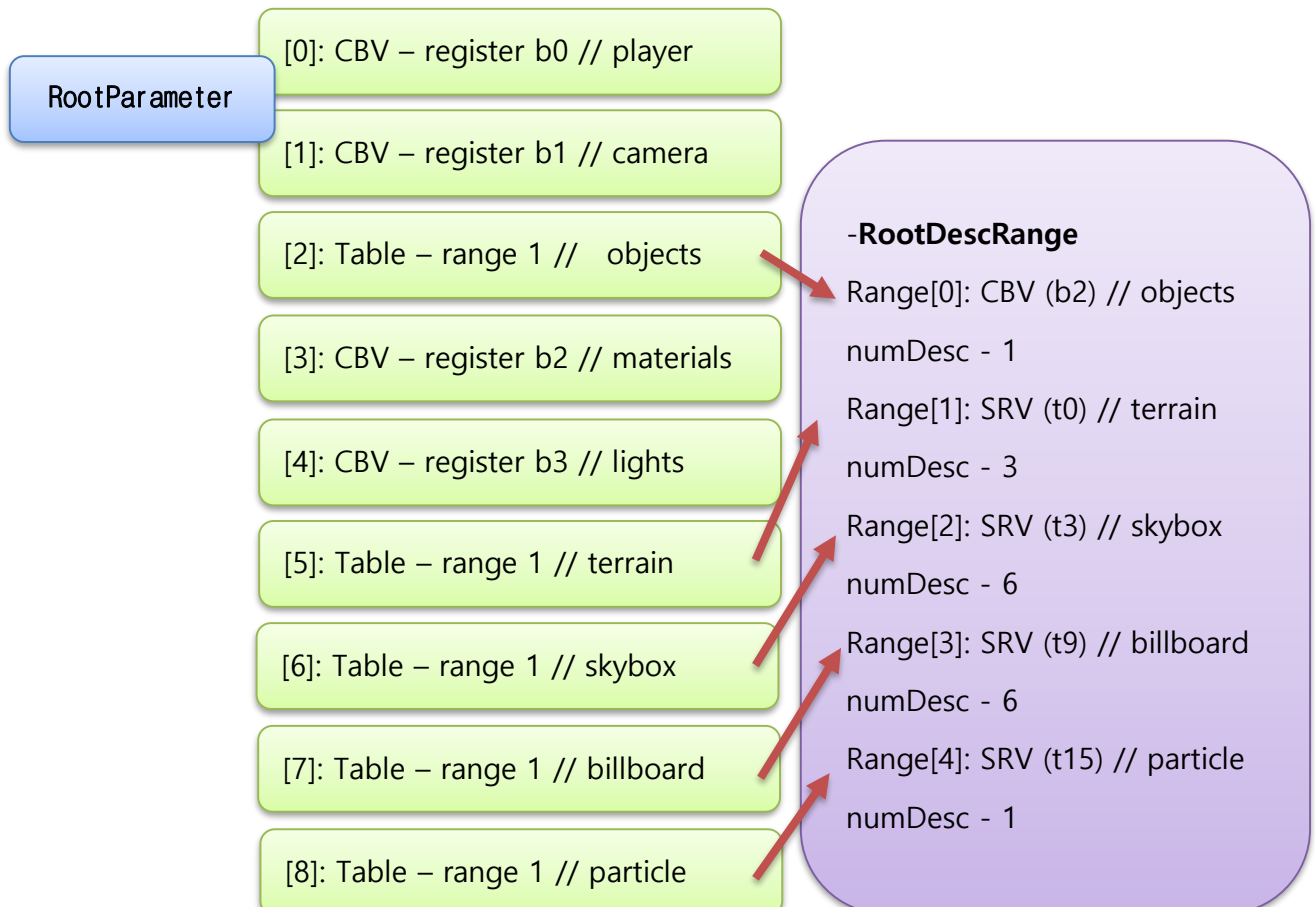


➤ [조작법]

키	설명
W, A, S, D	이동
Space	탄 발사
F2 / F3	자유 시점(플레이어를 그리지 않는다) / 플레이어 시점
R / F	상승 / 하강.
ESC	프로그램 종료

➤ [사용한 자료구조]



- 사용한 자료구조는 샘플 프로젝트 8-8 에서 루트 디스크립터 테이블에 사용할 텍스처들을 차례대로 range 에 등록한 후, 테이블을 루트 시그니처에서 사용할 오브젝트별로 나누었다.

➤ [구현하지 못한 내용]

- 구현 못한 내용은 과제 04 번을 제출할 때 보충하겠습니다
- 미니맵과 적 헬기가 플레이어를 따라오는 등의 움직임을 구현하지 못했습니다.

➤ [프로그램 요구사항]

1. 스카이 박스와 계곡 지형이 존재하고 지형에는 노멀맵핑이 되어 빛 반사를 한다.

- 가정
 - ① 8-8 샘플 프로젝트에 계곡지형과 스카이 박스를 적용한다.
 - ② 지형 높이 맵은 김병진 학우의 높이 맵을 참조하였다.
 - ③ 과제 02 때 구현했던 hsl 코드를 기반으로 한다.
- 구현
 - ① 루트 시그니처를 [사용한 자료구조]와 같은 형태로 만든다.
 - ② 과제 02 에서 터레인과 스카이박스의 오브젝트클래스, 셰이더클래스, 셰이더코드, mesh 클래스를 옮겨와서 붙인다.
 - ③ 스카이박스는 그대로 두고, 터레인의 오브젝트를 생성할 때 디테일텍스처의 노멀맵 로드하는 부분을 추가한다.
 - ④ Ppt 의 노멀맵핑을 설명하는 부분에서 탄젠트 계산식을 가져와서 height map image 클래스에 get height map normal 함수를 약간 수정하여 height map tangent 함수를 만들고, hsl 코드에서 터레인의 정점 셰이더에서 ppt 에 나온 그대로 tbn 행렬을 만들어 연산하여 픽셀

셰이더에서 노멀을 색으로 출력하여 노멀값이 제대로 들어왔는지 확인한다.

- ⑤ 노멀값이 제대로 들어왔다면 (0, 1, 0)벡터와 내적하고 실수 결과 값이 음수일 경우 -1 을 곱하여 양수로 만든 후에 이 값을 베이스텍스처 색에 곱하고, 1 에서 뺀 값을 디테일 텍스처 색에 곱하여 기울기가 평평할 수록 초록색, 많이 기울수록 돌 색을 나게 하여 절벽을 표현하였다.
- ⑥ 빛계산은 light.hlsl 함수를 사용하여 텍스처 결과색과 5:5 로 섞었다.

2. 빌보드로 풀과, 나무를 그리고, 파티클 애니메이션으로 폭발 효과를 구현하라.

● 가정

- ① 풀과 나무 빌보드는 과제 02 를 참조한다.
- ② 기하셰이더를 사용한다.
- ③ 파티클 애니메이션은 빌보드 정점에 텍스처 정보를 추가하여 구현하였다.
- ④ 플레이어가 쏘는 투사체도 빌보드 애니메이션이다.
- ⑤ 적과 플레이어가 쏘는 투사체의 충돌은 위치의 차이로 처리한다.

● 구현

- ① 과제 02 에 구현했던 빌보드 관련 소스코드를 현재 프로젝트에 적용한다.
- ② 과제 02 에서 가져온 빌보드 셰이더는 그래픽 파이프라인을 설정하는 부분과 그리기 전 디스크립터 힙을 셋하는 등의 전처리를 하는 부분만 남겨두고 나머지는 제거한다.
- ③ mesh 클래스에서 상속을 받아 staticBillboardMesh 클래스를 만들고 생성자에서 사용할 빌보드 정점 개수, 월드공간 위치(지형 어느부분에 풀과 나무를 그릴건지), 빌보드 크기, 월드공간 위치로부터의 반경(반지름), 지형 클래스의 포인터(풀과 나무가 놓일 높이값을 참조하기 위해)를 인자로 받아 정점버퍼를 default 로 만든다.

- ④ 정점의 위치는 입력 받은 월드공간 위치에서 반경안의 공간에 랜덤한 위치에 정점이 생성되고, 인자로 받은 빌보드 크기가 설정된 후, 터레인의 노말을 이용하여 경사가 심한 면에는 풀과 나무가 위치하지 않도록 하고, 터레인의 높이 값을 받아서 정점의 높이를 조정하였다.
- ⑤ 텍스처는 과제 02 에서 셰이더클래스에서 로드 했었는데, 이것을 텍스처 종류별로 오브젝트를 따로 만들어서, 풀은 grassBillboard, 나무는 treeBillboard 오브젝트를 만들고, 생성 설정은 텍스처를 6 개씩 로드하는 것 빼고는 터레인 오브젝트와 똑같이 하였고, 과제 02 와 같은 방식으로 hsl 에서 출력하였다.



터레인 노말매핑,
빌보드 풀, 나무

파티클 빌보드
오브젝트

```
class CParticleBillboards : public CGameObject
{
public:
    struct ParticleController {
        CParticleBillboardVertex& Particle;
        const XMFLOAT2 szSprite;
        XMFLOAT3 xmf3Direction;
        float fSpeed;
        float fAnimateSpeed;
        float fAnimateTime;
        float fTimer;
        bool bActive;

        ParticleController(
            CParticleBillboardVertex& particle,
            const XMFLOAT2& szSpriteSize) { ... }

        void StartAnimate(
            const XMFLOAT3& pos,
            XMFLOAT3& dir,
            float speed,
            float animateTime,
            float animateSpeed = 0.0f) { ... }

        void Reset() { ... }
        void Animate(float fTimeElapsed) { ... }
    };

public:
    ParticleController** m_ppControllers;
    UINT m_nControllers;

    UINT m_iCurrController;
```

- ⑥ 파티클 애니메이션은 빌보드 정점을 그대로 복사한 후, 스프라이트 이미지에서 현재 그려야 할 위치를 나타내는 변수와, 잘린 이미지의 크기를 나타내는 uv 변수를 추가로 가지고 있으며 파티클빌보드 메시는 정점버퍼가 디폴트가 아니라 업로드로 만들고 파티클 빌보드 메시의 멤버변수로 생성한 정점들을 유지하고, 정점버퍼를 거기에 바로 맵핑해서 비디오메모리의 정점버퍼 값을 변경할 수 있도록 하였다.
- ⑦ 파티클빌보드 오브젝트클래스를 만들어 위의 사진과 같이 클래스 내부에 메시의 맵핑된 정점을 참조자로 가지고 지속적으로 컨트롤할 수 있도록 구조체를 하나 만들어서 멤버함수를 통해 정점을 컨트롤하였다.
- ⑧ 추가적으로 오브젝트의 멤버변수로 다음에 바로 사용할 정점의 인덱스를 저장하는 멤버변수를 가지게해서 오브젝트의 멤버함수 popParticle()이 불리면 바로 현재 사용할 정점을 호출된 위치로 보내고 활성화시켜서 파티클 이펙트를 보여주도록 하고, 인덱스를 ++하여 미리 생성된 정점들을 돌아가면서 사용하였다.
- ⑨ Hls1 의 빌보드 셰이더코드를 복사한 후 정점에 추가된 정보들을 이용해 프레임마다 uv 값을 변화시켜 스프라이트 애니메이션이 가능하게 하였다.

3. 플레이어는 탄환을 발사하고 적 헬기에 맞으면 폭발하는 파티클 애니메이션이 실행된다.

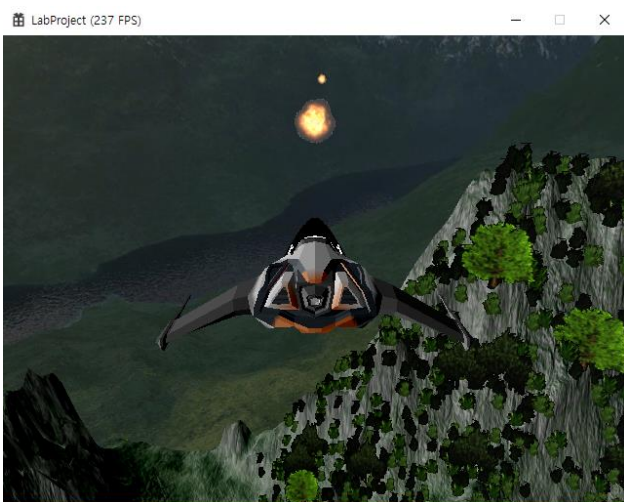
● 가정

- ① 플레이어가 발사하는 탄환도 빌보드를 이용한 파티클 애니메이션으로 구현한다.
- ② 적과 충돌처리는 파티클 정점 위치와 헬기의 위치값의 거리를 이용하였다.
- ③ 지형과 오브젝트의 충돌을 지형 높이맵을 이용하였다.

● 구현

- ① 스페이스바를 누르면 플레이어 위치에 파이어볼이 생성되고 앞으로 진행하는 애니메이션을 하며 10 초 후에 사라진다.

- ② 파티클빌보드 오브젝트를 그대로 사용하였다.
- ③ 씬의 animate()함수에서 모든 오브젝트의 위치값과 파티클 빌보드의 위치값의 차이를 구해서 거리가 일정 이하면 충돌하게 하였다.
- ④ 파이어볼 빌보드가 날아가다가 충돌하면, 파이어볼은 비활성화되고 폭발 파티클이 그 위치에서 활성화된다.
- ⑤ 헬기, 플레이어가 지형과 충돌하면 위치가 지형 에서 y 값이 50 만큼 큰 위치로 위치값이 보정된다.
- ⑥ 파이어볼이 지형과 충돌하면 헬기와 충돌했던 것과 마찬가지로 폭발 파티클이 활성화된다.
- ⑦ 헬기가 파이어볼에 맞으면 지형 중앙의 공중으로 위치가 이동된다.



플레이어의
공격인 파이어볼

폭발 파티클
빌보드 애니메이션

