

3D 게임프로그래밍

과제03 설명 문서

〈과제03 지형 전투 만들기〉

2021. 6. 24

2018182021 윤성주

목차

1. 게임 소개
2. 게임 목표
3. 구현 내용 설명 및 가정
4. 코드 설명
5. 사용한 자료구조와 알고리즘
6. 게임의 매커니즘
7. 조작법
8. 게임 플레이 화면

1. 게임 소개

지형 위에서 적군, 아군을 소환시켜 적군과 아군이 전투하는 게임이다. 적기지를 무너뜨리면 승리한다.

2. 게임 목표

- ① 지형을 추가하여 지형을 타야 하는 오브젝트(플레이어, 아군, 적군 일부)들은 지형을 타도록 한다.
- ② 적군과 아군에 모델을 적절히 사용한다.
- ③ 플레이어가 아군 생성 지점에 가서 아군을 소환하여 아군의 수를 늘린다.
- ④ 플레이어가 적군 생성 지점에 도달 시 적군 비행선으로부터 적군이 소환된다.
- ⑤ 적군이 소환되면 아군이 공격을 시작하고 적군은 플레이어를 향해 공격하여 전투하도록 한다.
- ⑥ 여러 종류의 적군을 생성한다.
- ⑦ 적군 생성 지점에서의 전투를 모두 수행하면 적군 기지가 가라앉으며 게임이 종료된다.

3. 구현 내용 설명 및 가정

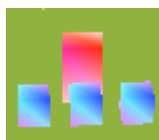
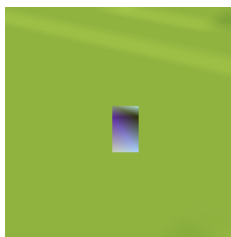
■ 가정

- 적군 기지는 4개이고 4개의 기지에서의 전투가 모두 끝나면 적기지가 가라앉는다.
- 지형을 타는 오브젝트들은 지형을 벗어나지 않는다.
- 계곡물에 빠져도 죽거나 타격을 입지 않는다.

■ 구현 내용 및 기능 설명

〈아군〉

① 플레이어

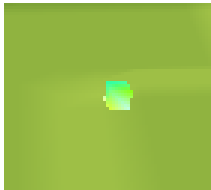


->피격 색깔(빨간색)

- 색깔을 랜덤으로 지정하므로 실행시킬 때마다 색상이 다름.
- 마우스로 시점 변경 가능하고 이동키로 이동 가능

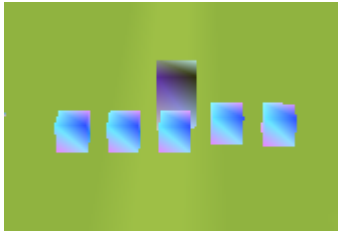
- 지형을 타도록 함
- 플레이어 중심으로 아군이 생성된다.
- 마우스 오른쪽 버튼 누르면 플레이어 총알 생성됨.
- 피격시 빨간색으로 색 변함. 이 부분은 셰이더에 변수를 연결해서 셰이더에서 색깔을 변경하도록 함.

② 플레이어 및 일반 아군 총알



- 플레이어 총알은 플레이어의 Look 방향으로 이동함.
- 회전하면서 이동
- 생성 후 일정시간 후에 객체 삭제
- 몬스터와 충돌 시 객체 삭제

③ 일반 아군 – 파란색 메쉬



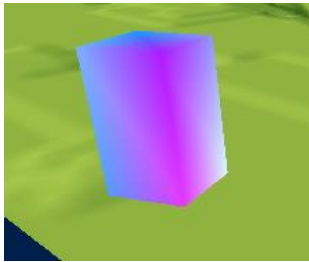
- 플레이어 뒤에 아군 생성 지점 방문시마다 5마리씩 생성
- 플레이어와 일정 offset 유지하고 지형 타면서 이동
- 몬스터가 있으면 랜덤한 시간마다 플레이어 Look방향으로 총알을 생성
- 몬스터 총알과 충돌 시 체력이 깎이고, 체력이 0보다 작아지면 소멸된다.
- 피격시 빨간색으로 색 변함.

④ 아군 비행선



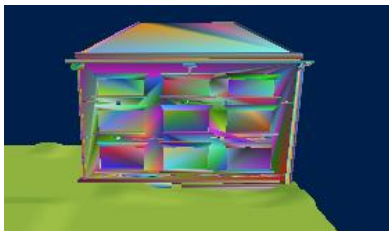
- 공중에 떠 있는 몬스터를 공격하기 위한 비행선.
- 첫번째 아군 생성 지점 방문 시 생성됨.
- 공중에 떠 있는 몬스터 방향으로 총알 생성
- 총알은 빠른 속도로 공격함.
- 체력이 줄지 않음.

⑤ 아군 생성 지점 - 파란색 메쉬



- 충돌 시 회전하면서 가라앉는다.
- 총 5개가 있으며 다 가라앉으면 각 지점에서 아군 5마리 생성
- 첫번째 지점에서는 비행선 2개가 추가로 생성된다.

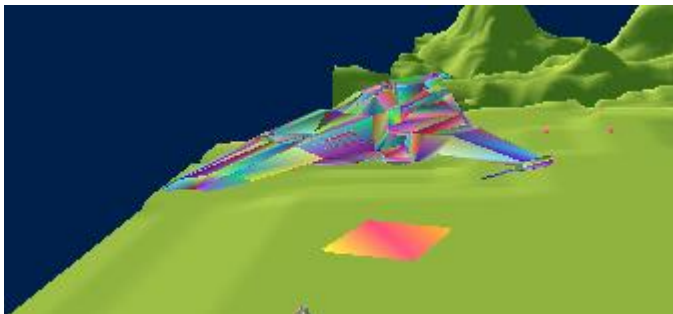
⑥ 아군 기지



- 플레이어 생성 위치 뒤에 아군 기지 배치

〈적군〉

① 적군 생성 비행선



- 적군 생성 지점과 플레이어 충돌 시 적군 생성지점 내려가고 다 내려가면 날아오

는 비행선. 적군 생성 지점에 도착 시 멈추고 몬스터 생성 후 날아감.

② 일반 적군 - 빨간색 메쉬



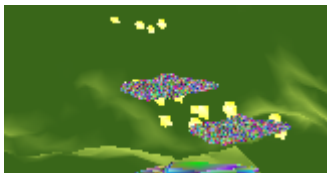
- 큐브 메쉬 사용.
- 비행선에서 떨어질 때 회전하면서 크기가 커지면서 지형에 착지하면 멈춤. 착지 후 공격 시작.
- 공격은 플레이어 방향으로 날아가는 총알 발사하는 것
- 피격 시 노란색으로 변함.

③ 플레이어 쫓아오는 적군 - 빨간색 메쉬



- 큐브 메쉬 사용
- 일반 적군보다 크기가 더 큼
- 랜덤한 속도로 지형을 타면서 플레이어 방향으로 이동.
- 피격 시 노란색으로 변함.

④ UFO 적군



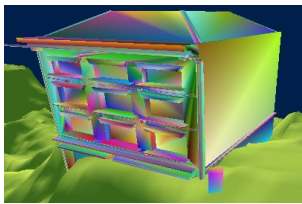
- 공중에서 원을 그리며 이동. 플레이어 방향으로 총알 발사.
- 피격 시 초록색으로 변함.

⑤ 적군 생성 지점 - 빨간색 메쉬



- 충돌 시 회전하면서 가라앉는다.
- 총 4개가 있으며 다 가라앉으면 각 지점에 맞는 적군 생성. 순서는 방문한 순서대로임.
- 지점1: 일반 적군 5마리 생성
- 지점2: 플레이어 쫓아오는 적군 5마리 생성
- 지점3: 일반 적군 15마리 생성
- 지점4: UFO 적군 2마리 생성

⑥ 적군 기지



- 적군 생성 지점 4개가 모두 가라앉고 몬스터가 다 죽었으면 가라앉음.
- 적군 기지가 모두 가라앉으면 플레이어 승리!

〈지형〉

① 터레인

- 하이트맵을 이용하여 지형 추가. 플레이어가 지형의 높이값을 받아와서 지형을 탄다.

② 계곡

- 큐브를 통해 구현. 따로 충돌은 구현하지 않았고 그냥 물에 빠질 수 있다.

4. 코드 설명

프로젝트는 따라하기 15까지의 코드를 기반으로 만들었으며 셰이더를 사용하였다. 또한 모델을 추가하기 위해 모델을 불러오는 함수를 추가하였다.

〈Player, Camera, Timer, GameFramework 소스코드 및 헤더파일〉

- 따라하기 15에서 추가된 부분 이외에 따로 수정하지 않음.

〈GameObject 소스코드 및 헤더파일〉

- CGameObject 클래스

- ✓ 충돌을 확인하기 위해 BoundingBox 변수를 추가하여 충돌체크 함.
- ✓ UpdateBoundingBox 함수를 통해 바운딩박스를 업데이트함.
- ✓ CObjectsShader 객체를 추가하여 CObjectsShader의 함수들에 접근할 수 있도록 함. 예를 들어 몬스터 객체에서 총알 객체를 생성하고 싶을 때 이 객체를 통해 총알 객체를 생성할 수 있음.
- ✓ MoveByDir 함수와 Dir 변수를 추가하여 객체가 Dir 방향으로 이동할 수 있도록 함.
- ✓ UpdateShaderVariables 함수에

```
pd3dCommandList->SetGraphicsRoot32BitConstants(0, 4, &m_xmf4HitColor, 16);
```

를 추가하여 변수를 연결함.

- CRotatingFlagObject 클래스
 - ✓ 적군, 아군 생성 지점 객체 클래스. 상태를 지정하여 일반 상태, 플레이어와 충돌 시 내려가는 상태, 다 내려가면 그에 맞는 행동을 취하는 상태로 나누어 관리됨.
- CEnemyFlyShip 클래스
 - ✓ 적 비행선 객체 클래스.
 - ✓ 날아오는 상태, 적 생성 상태, 날아가는 상태로 나누어 상태에 맞는 행동 취하도록 함.
- CTerrainObject 클래스
 - ✓ 지형을 태워하는 하는 오브젝트를 위한 클래스.
- CEnemy 클래스
 - ✓ CTerrainObject 클래스를 상속받는 클래스로 적이 비행선에서 내려오는 상태, 다 내려온 상태로 구분하여 관리됨.
 - ✓ Animate에서 일정시간마다 플레이어 방향으로 총알을 생성하도록 함.
- CFollowingEnemy 클래스
 - ✓ 플레이어를 쫓아오는 적을 구현하기 위한 클래스. 플레이어를 Set 해준다.
 - ✓ Animate에서 플레이어 좌표와 현재 좌표를 Vector3::Substract 함수를 통해 빼고 벡터를 정규화하여 Dir을 정한다. 일정시간마다 플레이어 방향으로 총알을 생성함.

- CMyTeamObject 클래스
 - ✓ 일반 아군 객체를 위한 클래스. CTerrainObject를 상속받아 지형을 타도록 한다.
 - ✓ 플레이어와 offset을 설정하여 offset을 유지하고 이동한다.
 - ✓ Animate에서 일정시간마다 총알을 생성하도록 함.
- CMyTeamShip 클래스
 - ✓ 공중 적에게 공격하는 아군 비행선을 위한 클래스. CMyTeamObject 클래스를 상속받아 플레이어와 offset을 유지하며 이동하도록 함.
 - ✓ 공중 적 생성시 Target이 설정됨.
 - ✓ Animate에서 일정시간마다 공중 적 방향으로 총알을 생성하도록 함.
 - ✓ 공중 적이 없을 경우 공격하지 않음.
- CBullet 클래스
 - ✓ 플레이어, 아군, 적 객체의 총알을 구현하기 위한 클래스.
 - ✓ 설정된 Dir 방향으로 이동함.
- CUFOEnemy 클래스
 - ✓ 공중적을 구현하기 위한 클래스
 - ✓ Animate 함수에서 sin, cos을 통해 원을 그리며 이동하도록 하였고 일정시간마다 플레이어 방향으로 총알 쏘.

〈Mesh 소스코드 및 헤더파일〉

- CModelMeshDiffused 클래스
 - ✓ LoadMeshFromFile 함수를 통해 bin, txt 파일에서 모델을 불러오도록 함.
 - ✓ 샘플 프로젝트의 코드를 수정하여 사용함.
 - ✓ 색상은 랜덤하게 지정

〈Scene 소스코드 및 헤더파일〉

- CreateGraphicsRootSignature 함수 수정
 - ✓ `pd3dRootParameters[0].Constants.Num32BitValues = 20/*16*/;`
 - ✓ 이 부분의 크기를 4만큼 늘려 float4 변수를 추가 연결하도록 함.

〈Shader 소스코드 및 헤더파일〉

- CObjectsShader 클래스 수정 및 추가
 - ✓ 게임 오브젝트에 쓸 메쉬들을 미리 생성하여 한번 생성된 메쉬를 공유하도록 함.
 - ✓ 리스트를 통해 게임 객체들을 관리함.

```
enum OBJID{MYTEAM, ENEMY, FLAG, MY_BULLET, MONSTER_BULLET, ITEM, BASE, FLYSHIP, NONE, END_OBJID};
```

(객체의 종류)

- ✓ BuildEnemy 함수에서 적군 생성 지점에 따라 그에 맞는 적군 생성하도록 함.
- ✓ BuildMyTeam 함수에서 아군 생성하도록 함.
- ✓ Collision_Check 함수에서 바운딩박스를 통해 충돌체크 Animate 마지막에 수행하도록 함.

〈Shader.hlsl〉

```
//게임 객체의 정보를 위한 상수 버퍼를 선언한다.  
cbuffer cbGameObjectInfo : register(b0)  
{  
    matrix gmtxWorld : packoffset(c0);  
    float4 hitColor : packoffset(c4);  
};
```

- 게임 객체가 피격인 상태에 색상을 변경하기 위해 hitColor 변수를 추가하여

```
//정점 셰이더를 정의한다.  
VS_OUTPUT VSDiffused(VS_INPUT input)  
{  
    VS_OUTPUT output;  
    //정점을 변환(월드 변환, 카메라 변환, 투영)  
    output.position = mul(mul(mul(float4(input.position, 1.0f), gmtxWorld), gmtxProjection), gmtxView);  
    output.color = input.color;  
    output.color.x += hitColor.x;  
    output.color.y += hitColor.y;  
    output.color.z += hitColor.z;  
    return(output);  
}
```

- 정점 셰이더 정의 부분에 색상을 추가하도록 함.

5. 사용한 자료구조와 알고리즘, 디자인 패턴

- ObjectsShader에서 리스트를 사용하여 객체 관리. Animate 함수에서 객체의 return 값이 OBJ_DEAD이면 반복자를 사용하여 객체를 삭제하도록 함.

6. 게임의 매커니즘

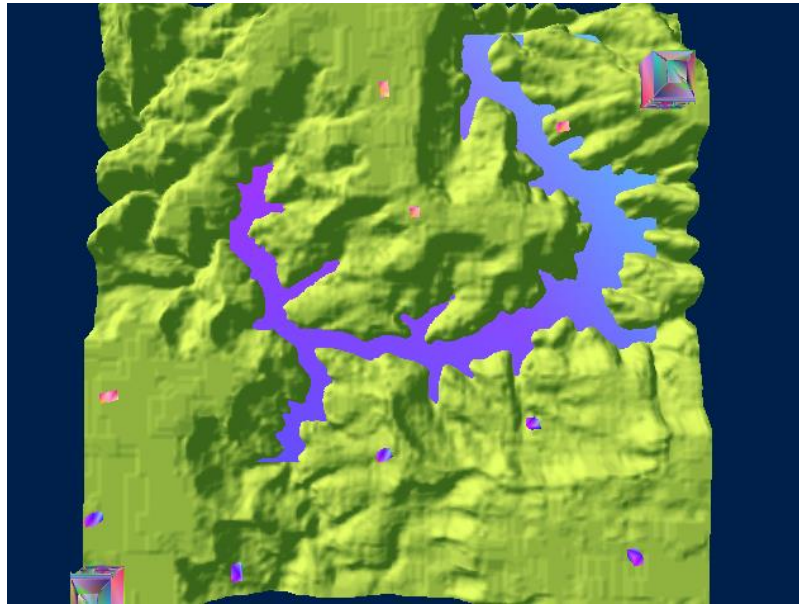
- ① 플레이어 아군 기지 앞에 생성
- ② 첫번째 아군 생성 지점과 충돌 시 아군 5마리와 아군 비행선 2마리 생성

- ③ 아군 생성 지점, 적군 생성 지점을 방문하면서 아군을 채우고 적군과 전투한다.
- ④ 적군 생성 지점이 다 내려가고 몬스터가 다 소멸되면 적군 기지가 가라앉으면서 플레이어 승리한다.

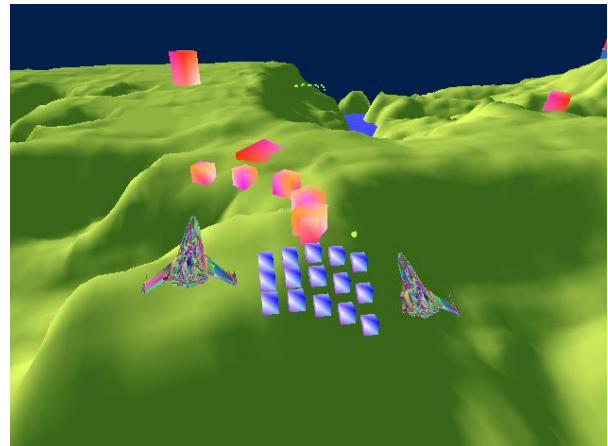
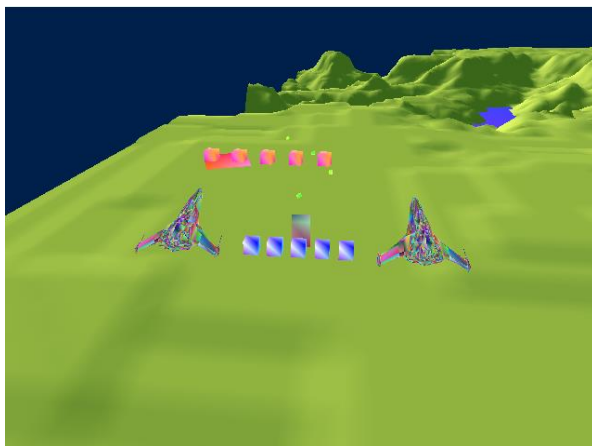
7. 조작법

- 키보드 이동키로 이동
- 마우스 좌클릭으로 시점 이동
- 마우스 우클릭으로 플레이어 공격
- F8키로 엔딩 보기 -> 치트키

8. 게임 플레이 화면



게임 전경



1->2->3->4

