3D게임1 과제3 설명 문서

2018182025 이승준

1. 조작법

**상하좌우 방향키**: 플레이어 객체를 앞, 뒤, 좌, 우로 움직인다.

**PageUp, PageDown**: 플레이어 객체를 위, 아래로 움직인다.

**Ctrl**: 플레이어가 바라보는 방향으로 미사일을 발사한다.

**F4**: 적이 출현하는 위치를 플레이어의 근처로 변경하거나 다시 먼 거리로 변경한다.

1. 과제에 대한 목표 및 가정

본 과제는 LabProject14에서 다룬 것과 같은 *TerrainMap*을 활용하여 지형을 생성하고, 그 지형을 바탕으로 적을 생성하여 플레이어에게 다가오게 하는 과제이다. 플레이어는 미사일을 발사해서 적을 처치할 수 있어야 하고, 플레이어나 적이 지형 안으로 파고들지 않게 하는 등의 충돌 처리가 되어야 한다. 또한 조명과 색 처리가 되어야 한다.

기본적으로 조명과 색 처리가 되어 있는 LabProject7-9-1을 바탕으로 프로젝트를 제작하기로 하였다. 해당 프로젝트에서 헬리콥터의 마우스를 통한 화면 회전이나 상하좌우전후 이동 등이 잘 구현되어 있으므로 이 역시 그대로 사용하기로 하였다.

여기에 LabProject14의 *HeightMap*을 프로젝트에 적용시키고자 하였는데, LabProject7-9-1은 LabProject14와 다르게 조명 및 재질이 적용되어 있으므로 따라하기(18)의 LabProject17과 같이 조명 및 재질을 사용하는 방식으로 LabProject14를 수정해주는 과정이 필요할 것 같다.

여기에 적 생성 및 충돌처리, 미사일 발사와 적 추격 작업을 더해주는 방식으로 프로젝트를 개발하고자 하였다.

또한 저번 과제에서 오브젝트를 생성할 때마다 새롭게 파일을 읽어 생성하는 바람에 많은 메모리가 낭비되고 로딩에 매우 긴 시간이 필요한 문제가 있었다. 이번 과제에서는 같은 모델을 여러 오브젝트가 공유하게 하는 방법으로 구현하여 해당 문제를 해결하고자 하였다.

1. 실행 결과

침대, 실내, 어두운이(가) 표시된 사진

자동 생성된 설명침대, 자연, 어두운이(가) 표시된 사진

자동 생성된 설명

*(과제 실행 화면) (적 격추)*

어두운이(가) 표시된 사진

자동 생성된 설명침대, 실내, 야채, 베개이(가) 표시된 사진

자동 생성된 설명

*(적 접근)*

플레이어의 이동은 **상하좌우 방향키**와 **PageUp, PageDown** 버튼을 통해 이뤄진다. 맵의 특정 장소에서 적이 일정 간격을 두고 생성되며, 적은 플레이어를 향해 날아온다.

**Ctrl** 키를 이용하여 플레이어가 바라보는 방향으로 미사일을 발사할 수 있다. 적이 미사일에 피격되면 격추된다.

**F4** 버튼을 통해 플레이어의 근처에 적이 생성되도록 할 수 있다.

어두운이(가) 표시된 사진

자동 생성된 설명자연이(가) 표시된 사진

자동 생성된 설명

*(충돌 처리)*

*TerrainMap*과의 충돌 처리를 통해 플레이어가 지형 아래쪽으로 이동해도 이를 쫓는 적 헬리콥터가 지형 안으로 파고들지 않는다. 플레이어 역시 지형 안으로 들어갈 수 없다.

플레이어가 적 오브젝트와 충돌한다면 적 오브젝트를 파괴하고 폭발 파티클을 출력하도록 하였다.

1. 구현 내용

텍스트이(가) 표시된 사진

자동 생성된 설명

*(적 생성)*

*Scene*의 *BuildObjects()*에서 모델의 정보를 미리 객체에 저장하며, 이 모델을 이용하여 적의 생성이 필요할 때마다 *EnemySpawn()* 함수를 통해 생성한다. 과제 2에서처럼 객체를 생성할 때마다 파일을 읽지 않아 프로그램 실행 속도가 개선되었다.

생성한 적 객체는 전부 *std::list*에 저장하여 컨테이너 중간에서 적이 삭제되는 과정이 효율적으로 이뤄질 수 있게 하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

*(오브젝트끼리의 충돌 처리)*

텍스트이(가) 표시된 사진

자동 생성된 설명

*(지면과의 충돌 처리)*

오브젝트끼리의 충돌 처리는 바운딩 박스를 통해 확인하였다. 플레이어와 적이 충돌했는지 확인하여, 충돌했으면 파괴 이펙트를 생성하고 적 오브젝트를 파괴하였다.

지면과의 충돌 처리는 LabProject14에서 다뤘던 *GetHeight()* 함수를 사용하였다. 객체의 좌표를 인자로 하여 해당 좌표의 *HeightMap*의 높이를 받아와 객체의 위치보다 높다면 객체의 위치를 재조정해주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

*(HeightMap에 Material 적용)*

*HeightMap.raw* 파일은 헬기 모델과 달리 재질 정보가 입력되어 있지 않으므로, 직접 *Material* 및 *IlluminatedShader* 객체를 생성하여 색상 정보를 입력하고 *HeightMap* 객체에 적용해 주었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

*(적 추격)*

플레이어의 위치와 적 오브젝트 객체의 위치를 빼고 이를 정규화하여 방향벡터를 구하였다. 이는 내가 향하고자 하는 새로운 *Look벡터*일 것이다. 이를 원래 *Look벡터*와 외적하면 새로운 *Up벡터*를 구할 수 있고, 이는 음수일 수도 있으므로(비행기가 뒤집어져 보인다.) 이를 보정해준다. 이는 객체의 새로운 *Pitch*값이 되고, 새로운 *Look벡터*를 객체의 새로운 *Yaw*로 지정해 주면 객체가 향하고자 하는 방향으로 알맞게 회전된다. 이제 앞으로 이동하게 되면 플레이어를 향하는 방향으로 객체가 이동한다.