3D게임 프로그래밍

과제

게임공학과 3학년 2020182028 이규원

목차

1. **조작법**
2. **실행 결과**
3. **과제에 대한 목표 및 가정**
4. **구현 내용 & 추가적으로 구현한 내용**
5. **조작법**

이동 - 화살표 키, page up & down

무기 발사 – ctrl

쉴드 활성화 – Z키

캐릭터 시점 이동 – 마우스 왼쪽 드래그

카메라 회전 – 마우스 오른쪽 드래그

게임 종료 – esc

1. **실행결과**

스케치, 도표, 아동 미술, 디자인이(가) 표시된 사진

자동 생성된 설명스케치, 그림, 아동 미술, 디자인이(가) 표시된 사진

자동 생성된 설명

도표, 텍스트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

1. **과제에 대한 목표 및 가정**
2. **3D형태로 보이는 start 구현.**

START메뉴를 구현하기 위해 기존의 CExplosiveObject를 이용해 pCubeMesh, 육면체들로 START 글자를 만들고, 플레이어가 미사일을 발사해 Start 메시와 충돌 시 게임 시작이 되게끔 한다.

게임 시작 후, START 오브젝트와 플레이어 기체만 화면에 나올 수 있도록 만들고, 충돌 후에 터지는 이펙트 후, START 오브젝트를 비활성화 하고, 메인 게임을 구현할 수 있도록 한다.

플레이어가 적 미사일에 맞아 격추 시, 다시 START 화면으로 돌아오도록 한다.

1. **적기의 미사일 발사 구현.**

적의 미사일 발사를 구현하기 위해, 플레이어가 사용하고 있는, 미사일 발사를 적기에도 구현한다.

적기가 가만히 서서 쏘면 게임이 너무 쉽고 단조로워 지고, 너무 무작위성으로 이리 저리 날라다니면 플레이어 입장에서도 미사일 회피에 있어 너무 불합리하게 느껴질 수 있다. 그래서 좌우, 상하로 규칙성 있게 움직이면서, 정면으로 미사일을 발사하게 해, 플레이어로서 하여금 너무 단조롭지 않으면서 동시에 어느정도 컨트롤에 신경 쓸 수 있도록 한다.

위에 서술한 내용과 같이 플레이어와 적 미사일이 충돌 시 첫 화면으로 돌아갈 수 있게 한다.

1. **‘z’키를 누를 시 플레이어에게 쉴드 발동 구현**

z키를 누르면 플레이어 중심을 기준으로 육면체를 그려 쉴드가 생성됐음을 보여준다.

쉴드의 지속시간은 5초, 쿨타임은 쉴드가 꺼진 후 5초로 설정한다. 쉴드를 이미 발동중이거나, 쿨타임 중이면 쉴드 키를 눌러도 발동되지 않는다.

쉴드 발동 시 쉴드 바운딩 박스와 적 미사일이 충돌하게 되면, 적 미사일은 그 위치에서 소멸하게 한다.

1. **구현 내용 & 추가적으로 구현한 내용**
2. **3D형태로 보이는 start 구현.**

START 오브젝트 구조체는 위치 외의 모든 정보가 같기에 FOR문을 통해 만들었고, 위치는 case문을 통해 START오브젝트를 구현했다.

플레이어 미사일 – START 오브젝트 등의 충돌처리는 Scene.cpp의 Animate에 구현했다. 미사일과 START오브젝트 충돌시 즉시 넘어가는게 아니라 폭죽 애니메이션을 보여주기 위해 타이머를 돌려 1초 뒤에 화면 전환이 될 수 있게 구현했다.

CScene Render에 bool 변수를 통해 START맵에선 START 오브젝트만 보이게끔 렌더링하고, START 오브젝트와 충돌 후엔, bool 변수를 통해 메인게임으로 들어갈 수 있게 구현했다.

1. **적기의 미사일 발사 구현.**

CExplosiveObject class에 CAirplanePlayer의 FireBullet을 이용해 CExplosiveObject에도 FireBullet 함수를 생성했다. 클래스의 m\_ppBullets 배열을 통해 오브젝트들을 관리했다.

CExlposiveeObject Animate에 미사일 위치 정보를 업데이트 하고, Render에서 미사일을 출력했다.

적이 미사일을 쭉 발사하다가, 중간에 또 끊기고 발사하는 모습을 보여 이 문제를 해결하기 위해 통해 적이 미사일을 한번에 발사 할 수 있는 개수 ENEMYBULLETS를 정의해 3발로 한정 하여, 자연스럽게 출력되도록 만들었다.

적의 미사일과 쉴드 충돌 시 적기의 수만큼 적기에 담겨있는 미사일들을 쉴드와 OOBB 충돌 검사를 통해 쉴드와 충돌했는지 체크하고, 쉴드와 충돌 시 미사일이 소멸하게 만들었다.

적기의 수만큼 적기에 담겨있는 미사일들을 플레이어와 충돌 검사를 해 만일 충돌 시 플레이어를 위에 만든 START 맵으로 이동하게 만들었다.

1. **‘z’키를 누를 시 플레이어에게 쉴드 발동 구현**

CAirplanePlayer class에 쉴드 오브젝트, 쉴드가 켜져있는지에 대한 bool 변수값, 쉴드 지속시간, 쿨타임을 계산하는 float값 변수를 만들었다.

CAirplanePlayer 생성자에 쉴드 메쉬 생성 및 위치, 색상을 설정했다.

Animate에서 쉴드 쿨타임일 때 쉴드 쿨타임을 진행하고, 쉴드가 켜졌을 때, 쉴드의 위치를 플레이어를 따라가도록 업데이트하고, 쉴드의 지속시간을 계산해 지속시간이 다 지날 시, 쉴드가 꺼지고 쉴드 쿨타임이 돌아가도록 구현했다.

Render에서 쉴드를 켰으면 쉴드 오브젝트를 화면에 그려주도록 구현했다.

GameFramework에서 Z키를 누를 시, 쉴드 쿨타임이 아니며 쉴드 지속중이 아닐 때 쉴드 지속시간 타이머를 0초로 초기화하고, 쉴드를 킬 수 있도록 bool 변수를 true로 변경하게 구현했다.