

# 컴퓨터 그래픽스 실습(2) 프로젝트 보고서

B777009 김석환

## 목차

1. 조작법
2. 게임 특징
3. 구현 방법
4. 어려웠던 점

### 1. 조작법

먼저 타이틀 장면에서 스페이스바를 통하여 게임으로 진입한다.

메인 장면에서는 W, A, S, D키를 통하여 이동하고 마우스를 통하여 좌우 상하의 회전이 이루어진다.

십자가 오브젝트를 얻거나, 그 후 유령을 물리치는 등의 행동은 직접 접촉을 통하여 이루어진다.

50초의 기간 내로 모든 유령을 물리칠 경우 승리, 그러지 못할 경우 패배한다.

메인 장면, 승리 장면, 패배 장면에서 재시작을 원할 경우 R키를 통하여 가능하다.

승리 및 패배 장면에서는 Q키를 통하여 타이틀로 돌아갈 수 있다.

언제든 ESC키를 통하여 종료할 수 있다.

### 2. 게임 특징

기본적인 형태는 무덤가에 나타난 유령들을 십자가를 통하여 제령 하는 게임이다.

묘비는 총 16개로 4\*4의 직사각형 형태로 배치되어 있고 그 중 4개의 묘비에서 유령이 그 주위를 돌고 있지만 그냥은 보이지 않고, 가까이 가야만 확인할 수 있다.

묘비에 닿게 되면 이동이 정지하게 되고, 유령과 닿게 되면 유령은 제령 된다.

하지만 십자가를 들고 있지 않을 경우, 유령을 제령 시키는 상호작용을 할 수 없다.

십자가는 묘비가 있는 범위 내에서 무작위로 생성되는데, 그 위치를 알게 하기 위하여 빛의 기둥이 존재한다. 이 기둥은 묘비에는 가려지지만 배경의 건물에는 가려지지 않는다.

십자가에 닿게 되면 빛의 기둥은 사라지고 십자가를 손에 들게 된다. 이 때부터 유령을 제령 시키는 상호작용을 할 수 있게 된다.

유령을 제령 시킬 때까지 50초의 시간이 주어지는데 해당 시간 내로 제령 시킬 경우, 유령들이 모두 성불하여 자유로이 날아가는 모습과 함께 성공할 때까지 걸린 시간이 화면

에 나타난다.

하지만 시간이 다 지나가는 경우, 유령들이 캐릭터의 주위를 돌며 기만하는 모습이 화면에 나타난다.

### 3. 구현 방법

대략적인 부분은 이전학기에 있었던 컴퓨터 그래픽스 실습(1)의 팀프로젝트에서 사용했던 코드들을 활용하여 당시 구현했던 기능들을 재연하였다.

예시로는 카메라 이동, 카메라 회전, 사운드 재생, AABB 콜라이더, 스카이 박스, 텍스트 출력, 시간 계산, 행렬을 이용한 동일 모델의 다중 렌더 등이 있다.

추가적으로 구현한 기능에는 파티클 시스템, 멀티 씰 제작 및 변환, 필요에 따른 라이팅 변경, 스카이 박스의 스카이 맵(스피어 화), 비트맵의 픽셀 셰이더를 통한 투명도 조절, 빌보드, 오브젝트의 랜덤 위치 구현이 있다.

AABB 콜라이더의 경우, 이전에는 닿을 경우 닿는 해당 오브젝트를 다른 공간으로 이동시키는 형태였으나, 이번에는 닿을 경우 플레이어를 멈추는 오브젝트가 존재했다. 그리하여 이동에 사용되는 변수 중 하나인 속도를 0으로 하는 형태로 구현되었다.

파티클 시스템의 경우에는, 인터넷의 코드를 찾아 임포트 하였다. 추가로 변수를 입력 받을 수 있는 형태로 변형하여, 특정 파티클에서는 원하는 식으로 색이나 크기 등을 바꿀 수 있도록 구현하였다.

멀티 씰은 렌더 함수를 추가로 구현하여 상황에 맞는 렌더 함수를 실행하는 형태로 구현하였다. 그래픽클래스의 프레임 함수에서 시스템클래스로부터 받은 장면 변수에 따라 스위치 함수를 이용하여 첫 시작의 경우 타이틀, 이후 스페이스 키를 누를 경우 메인 장면으로 이동하는 등으로 구현되었다. 타이틀, 승리, 패배 장면에서는 이동이나 마우스를 통한 카메라 회전을 막아 두었는데, 이는 원하는 모습의 화면을 보이기 위해서이다.

라이팅의 경우, 추가 라이팅 변수를 만들고자 하였으나 변화시키고자 하는 부분의 변수만 바꿀 수 있는 형태로 개선하여 구현되었다. ambient 값, diffuse 값, specular 값, specular 크기 등의 변수를 헤더 파일에 추가로 만들어, 필요한 부분에서는 기존 라이트 값과 다른 ambient 값, diffuse 값, specular 값을 가진 라이트가 되도록 만들었다.

스카이 맵은 수업 중에 사용한 예시를 기본으로 하여 동기인 한태욱의 도움을 받아 구현하였다.

비트맵의 픽셀 셰이더를 통한 투명도 조절은 텍스트 픽셀 셰이더를 응용하여 상황에 맞도록 알파값을 조절하는 형태로 구현하였다. 그런 식으로 타이틀, 승리, 패배 장면에서 텍스트 대신에 나오도록 구현하였다.

빌보드는 인터넷의 코드를 찾아 임포트 하였는데, 이전 버전의 그것이라 XM의 형태로 재구성하여 구현하였다.

오브젝트의 랜덤 위치는 처음에는 생각치 않았으나, 파티클 시스템의 임포트 중에 랜덤

값을 사용하는 것을 보고 응용하여 구현하게 되었다.

#### 4. 어려웠던 점

스산한 분위기를 주기위해서 안개 이펙트를 찾아 추가하고자 하였으나, 찾아본 안개 이펙트는 스카이 맵과 함께 사용하기 어려운 구조였기 때문에 스카이 맵을 우선시하여 포기하게 되었기에 다른 방법을 찾다 노이즈 매핑을 찾았지만, 원하는 형태의 노이즈 텍스처를 찾지 못하여 또다시 포기하게 되었다.

추가 라이팅에서 스포트 라이트처럼 해당 오브젝트와 그 주변 또한 밝히고자 했지만, 지면을 터레인이 아닌, 플레인 큐브에 텍스처를 입힌 형태로 했기에 되지 못하였다. 또한 그저 위에서 밝히는 빛의 경우, 오브젝트가 잘 보이지 않는 문제점 또한 있었기에 포기하게 되었다.

승리 및 패배 장면에서 폰트를 바꿔보라는 피드백을 받았기에 폰트 텍스처와 폰트 데이터를 새로 하고자 하였으나 텍스처는 간단히 되었음에도 불구하고 폰트 데이터의 계산이 잘 되지 않아서 비트맵 이미지로 대체하게 되었다.