컴퓨터 그래픽스 최종 프로젝트

개발 결과 보고서

**2016184001 강진주**

**2016184024 신봉섭**

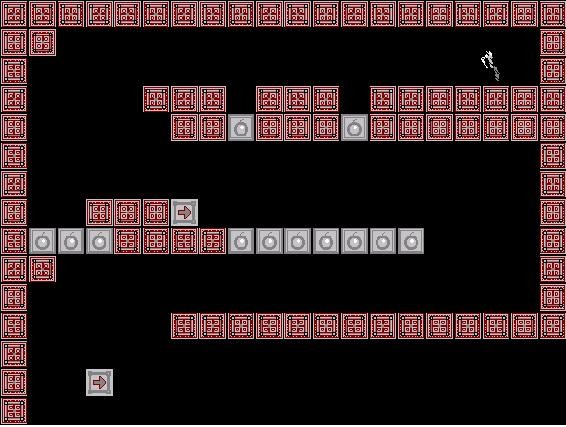
# 프로젝트 소개

개체, 시계이(가) 표시된 사진

자동 생성된 설명개체이(가) 표시된 사진

자동 생성된 설명

## **게임 이름: 3D 공튀기기(BouncingBall#3D)**

프로젝트를 시작할 때 사진과 같은 기존의 2D 공튀기기 게임과 유사하게 3D로 변환한 게임을 만드는 것이 목표였다.

최종 구현 결과, 3차원 일 때만 표현할 수 있는 다양한 요소들을 프로젝트 게임의 요소들로 구현하여 3D 공튀기기 게임을 완벽하게 구현하였다.

## **플레이 방법:** 다양한 특성을 가진 블록(발판)들을 떨어지지 않고 밟아가며 튀어 오르는 공을 골인지점까지 옮겨라!

## **게임 특징**

게임을 진행하는데 두드러지는 큰 특징들은 다음과 같다. 세부적인 구현 내용 및 이미지들은 구조 소개하기 문단과 결과물 분석 문단에서 다룰 것 이다.

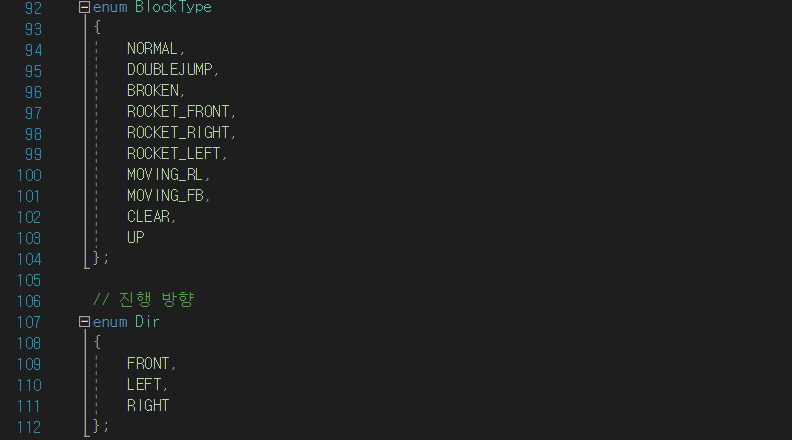
1. **카메라 시점 :** 카메라가 하늘에서 공을 따라가며 바라보는 관측자 시점 / 카메라가 튀기고 있는 공의 뒤에 붙어 따라가는 팔로잉 시점이 있다. 관측자 시점에서는 카메라를 회전(공전)시켜 주위 맵을 둘러볼 수 있다.
2. **블록(발판) :** 일반 블록 / 점프량 증가 블록 / 깨지는 블록(알파블렌딩 적용) / 직선 비행 블록 / xz평면에서 상하좌우 움직이는 블록 / y축으로 상승하는 블록
3. **맵 진행 :** 다양한 블록의 테마 별로 맵의 레벨(스테이지)을 구현. 한 레벨당 블록 5~6개 존재. 총 12개의 레벨로 이루어져 있다.맵의 레벨이 바뀔 때 마다 공의 진행 방향이 바뀌고 카메라 역시 바뀐 진행 방향으로 역동적으로 회전하여 고정된다.
4. **각 레벨의 블록마다 세이브 포인트 존재 :** 세이브 포인트인 블록을 밟으면 세이브 포인트가 저장되며 블록의 색깔이 주황색으로 바뀐다. 다음 세이브 포인트를 찍기 전에 공이 떨어져 죽으면 이전에 저장해 놓은 세이브 포인트에서 공이 살아난다.
5. **튀기는 공** : obj를 받아와 조명이 적용된 공은 블록을 밟으면 중력 가속도를 고려하여 튀어 오르고 밟지 못하면 중력이 적용된 상태 그대로 아래로 떨어진다. 아래로 떨어질 땐 떨어지는 느낌을 강조하기 위해 시점이 팔로잉 시점으로 고정된다. 일정량 떨어지면 공이 죽고 세이브 포인트에서 되살아난다.

# 구조 소개하기

프로젝트 진행에서 가장 중요한 플레이어인 공,블록,카메라를 각각 클래스로 구현하였다. 또한 특정 클래스에서 필요한 멤버변수 별로 열거체(enum)를 구현하여 편리함과 속도 향상을 고려 하였다.

1. **열거체 :** **BlockType, Dir**

* **BlockType: 블록 객체의 종류를 나눠 주기 위한 열거체**
* **Dir: 플레이어 공 객체의 진행 방향을 지정해주기 위한 열거체**



1. **플레이어 공 클래스: C\_PlayerBall**

* **멤버 변수**: 공의 위치, 충돌 처리를 위한 공의 크기, 움직임, 상태 변수(어느 블록과 충돌한 상태인지, 죽은 상태인지 등),진행 방향을 정해주는 Dir 변수 등을 멤버 변수로 가지고 있다. 텍스트이(가) 표시된 사진

  자동 생성된 설명
* **멤버 함수(Move):** 공으로 사용되고 있는 obj 크기를 glm::scale 함수로 줄여주면서 glm::translate 함수로 공을 이동 변환 시켜 움직이게 하는 함수이다.

실내이(가) 표시된 사진

자동 생성된 설명

1. **블록 클래스: C\_Block**

* **멤버 변수**: 블록의 위치, 충돌 처리를 위한 블록의 크기, 객체 별 블록 타입을 정해줄 수 있는 BlockType 변수, 상태 변수(공과 충돌 했는지 등), 움직이는 블록일 때 이동량, 진행방향이 바뀌는 블록일 때 충돌 시 공의 진행 방향을 바꿔주는 Dir 변수 등이 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* **멤버 함수 Set:** glm::translate 함수를 사용하여 블록 객체의 위치 지정, objColorLocation으로 블록 타입 별 색깔 지정, 움직이는 블록일 때 moving\_translater 행렬을 적용하게 하는 함수이다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

* **멤버 함수 Move:** 움직이는 블록일 때 moving\_translater에 적용되는 이동량 변수를 바꿔주는 함수이다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

1. **카메라 클래스: C\_Camera**

* **멤버 변수**: 카메라 변환에 필요한 변수, 카메라 위치, 회전 각도, 키 입력에 따른 카메라의 상태 변화를 적용하는 bool 변수 등이 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* **멤버 함수 Set:** 키 입력에 따라 카메라 시점을 바꿔주고 카메라가 회전되게 하는 함수이다.

스크린샷, 앉아있는이(가) 표시된 사진

자동 생성된 설명

# 팀원 간 작업한 내용(프로젝트 진행 사항)

|  |  |
| --- | --- |
| 강진주  (클래스&객체, 카메라, 특수블록 중점) | 공,블록,카메라 클래스 제작: 조작 및 움직임,위치 지정 |
| 카메라 시점 구현: 공의 진행 방향을 따라가는 카메라, 시점 변환 키, 클리어 시 카메라 워킹, 키 입력 시 카메라 회전 |
| 특수 블록 제작: 점프량 증가 블록, 직진으로 날아가는 블록, 밟으면 깨지는 블록, xz평면으로 움직이는 블록 |
| 신봉섭  (충돌처리,  알파블렌딩,  맵 중점) | 맵(지형) 구현: 블록 배치로 맵 레벨 구현  특수 블록 제작: y축으로 상승하는 블록/ 깨지는 블록 알파블렌딩 |
| 카메라 시점 구현: 맵의 진행방향 따라서 자동으로 시점 변환 |
| 공과 블록간 충돌처리 |
| 공동 작업 | Obj 적용, 프레임 워크 제작, 물리 적용, 텍스쳐, 사운드, 조명,치트 |

깃허브 commit을 활용해 공동 작업을 하였고 프로젝트 기간 내내 직접 만나서 작업을 진행하였다. 그래서 서로 중점적으로 맡은 부분이 있지만 바로바로 팀원 간 소통을 해서 막히는 부분이 있으면 서로 맡은 부분을 도와주며 함께 알고리즘을 짜고 작업을 하였다.

# 결과물 분석

**2019 컴퓨터 그래픽스 강의에서 배운 쉐이더,모델링 변환,obj파일 다루기,뷰잉 변환,투영 변환,텍스쳐,조명 모든 것을 활용하여 구현하였다.**

개체, 시계이(가) 표시된 사진

자동 생성된 설명개체이(가) 표시된 사진

자동 생성된 설명

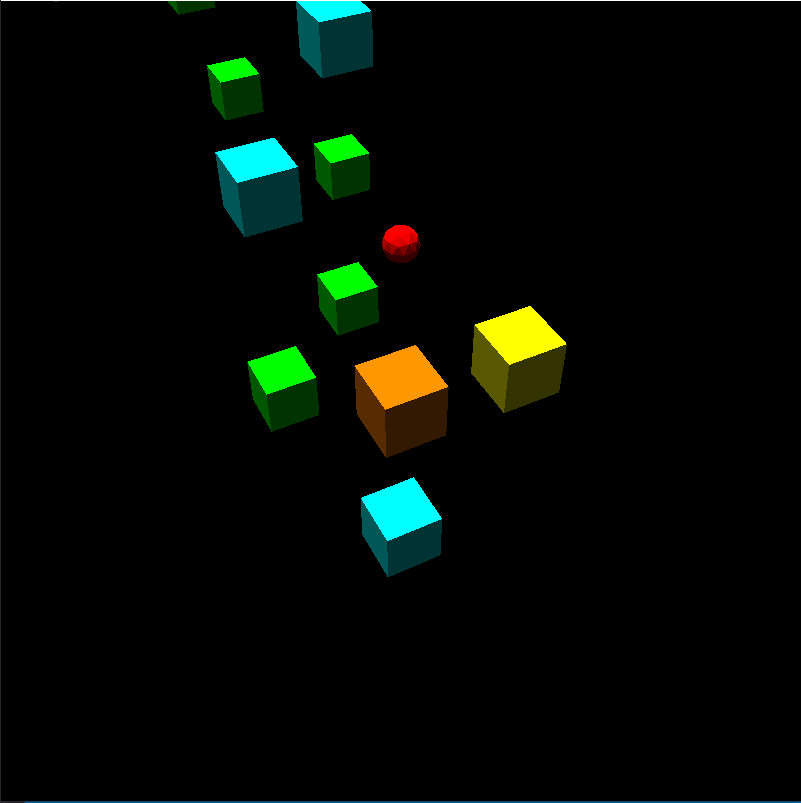
중력 가속도가 적용되어 튕기고 있는 공에 관측자 시점과 팔로잉 시점으로 카메라를 변환 하여 시점에 따른 원근감이 달라보인다.

또한 조명을 적용하여 공과 블록들의 입체감이 강조 되었다.

옅은, 빨간색이(가) 표시된 사진

자동 생성된 설명

텍스쳐를 적용하여 특정 키를 눌렀을 때 블록 별 설명이 적용되어 있는 텍스쳐가 화면에 살짝 기울여져 입체감 있게 뜨도록 구현하였다. 또한 **클리어시** 폭죽 텍스쳐가 나타나게 구현했고 카메라 워킹으로 전체적인 맵 볼륨을 볼 수 있도록 구현했다. 쉐이더,텍스쳐가 여러 번 로드되어 프로그램이 터지거나 버벅이는 현상을 한 번만 로드하도록 해서 해결했다.

신호등이(가) 표시된 사진

자동 생성된 설명

또한 관측자 시점일 때는 키 입력을 통하여 카메라를 회전(공전) 시켜 블록간의 위치와 거리감을 확인해서 이동할 수 있도록 구현하였다. 캡쳐에는 담을 수 없지만 레벨별로 시점이 변하는 블록이 있는데 그 블록과 공이 충돌하면 카메라시점이 자동으로 회전한다.개체이(가) 표시된 사진

자동 생성된 설명개체, 시계이(가) 표시된 사진

자동 생성된 설명

또한 여러 특수 블록들을 구현하여 특수 블록들과 충돌했을 시 각기 다른 상호작용을 하도록 구현했다. 사진은 깨지는 블록의 예시이다. 깨지는 블록에는 알파블렌딩을 적용했다. 깨지는 블록을 밟으면 사라지고 그 전에 다른 블록으로 이동하지 않으면 공은 그대로 떨어진다. 이 외에도 앞서 소개한 여러 특수 블록들이 있다.

특수블록들을 활용한 총 12개의 레벨로 이루어져 있으며 각 레벨당 세이브 포인트 블록이 존재한다. 세이브 포인트 블록을 밟아 지정되면 블록이 이미지와 같이 주황색으로 변한다.

추가적으로 사운드를 추가하여 플레이 하는 동안 배경 음악이 나오도록 하였다.

# 필요한 명령어 소개

|  |  |
| --- | --- |
|  | 방향키 |
| 이동 | W,A,S,D |
| 시점 변환 | P |
| 카메라 회전 | Q,E |
| 블록 설명보기 | M |
| 종료 | ESC |
| 치트키 | 0(공이 블록 5칸 앞으로 이동) |