* 게임 소개

저희가 만든 게임은 3차원 비행 게임입니다. 플레이어는 드론과 장애물들을 피해가며 링 안의 아이템을 먹어야 하며, 맵 안의 모든 아이템을 먹으면 게임이 끝나게 됩니다.

드론이나 장애물, 바닥에 부딪히면 플레이어의 비행기가 터지며 맵의 경계를 표시하는 구의 바깥으로 나가도 플레이어는 폭발합니다.

플레이어는 스텔스 기능과 순간 가속 기술을 가지고 있습니다. 스텔스 기능을 사용하면 초당 10의 마나를 소모하여 마나를 모두 소모할 때까지 투명해지며 드론의 추적에서 잠시 벗어날 수 있습니다. 마나를 소모하지 않는 동안엔 초당 5의 마나가 자동으로 회복됩니다.

순간 가속 기술은 짧은 시간 동안 속력을 급속도로 올릴 수 있는 기술입니다. 가속이 되는 동안에는 최대 속력의 한계가 늘어나며 시간이 지나면 다시 원래의 최고 속력으로 돌아오게 됩니다.

* 프로그램 구조

-Object Class : 게임 객체의 기본이 되는 클래스로 위치, 각도, 월드변환 행렬을 가지고 있고, 바운딩 구를 가지고 있습니다..

-CubeObject Class : Object 클래스의 자식 클래스로 큐브의 크기 값을 가지고 있습니다. 이 큐브와 행렬을 이용하여 OOBB충돌 체크를 합니다.

-게임 오브젝트들(Player, Drone, Ring, Building, etc ) : Object 클래스의 자식 클래스들이며 내부는 CubeObject들 여러 개로 구성됩니다. 이 내부의 CubeObject들은 Object에 대하여 상대 좌표를 가지며, 충돌체크시 이 CubeObject들이 계층적 충돌체크가 되도록 행렬을 만들어 놓았습니다.

-A.I State Class : Drone 인공지능은 상태패턴을 이용하여 상황에 따라 필요한 상태를 선택하도록 구현하였습니다.

-Arm Bullet Class : Unit이 소유하고 있는 Arm에 대하여 Shot함수를 호출하면 Arm클래스가 Bullet을 생성하여 BulletManager한테 넘겨 줍니다.

-Manager들 : InputManager는 사용자 입력을 처리하고, EffectManager는 객체가 Effect를 요청하면 해당 효과를 생성 및 관리합니다.

StageManager는 플레이어를 제외한 게임 객체들을 관리하는데, 사실상 게임 프레임 워크의 역할을 해주고 있습니다.

* 프로젝트 진행 상황

Python으로 맵툴을 제작하여 12km\*12km크기의 맵을 제작하였고 맵툴에서 드론이 이동할 수 있는 노드들 또한 생성해 주었습니다.

맵툴에서 건물을 겹치지 않는 한 자유롭게 배치할 수 있으며, 아이템을 포함하는 링 또한 건물과 겹치지 않는 한 어디든 자유롭게 배치할 수 있습니다.

건물 배치와 링 배치가 모두 끝내면 맵툴이 자동적으로 건물을 피해서 각 노드들을 연결해 줍니다. 이 노드들과 연결 선은 드론 AI인 A\* 알고리즘에서 사용됩니다.

비행 게임에서 가장 중요한 플레이어의 시점 변환도 구현 완료 하였고, A\*를 이용하여 플레이어를 추적하는 드론 AI 또한 구현 완료 하였습니다.

플레이어는 비행기를 조종하기 때문에 비행기의 조종이 직관적이고 자유로워야 하는데 OpenGL에서 제공하는 glRotatef 함수는 세 개의 회전 축에 대하여 적용하는 순서에 따라 결과물이 달라지고, 화면을 다시 그릴 때 마다 월드 기저를 축으로 해서 회전하기 때문에 직관적인 조작을 하기 힘들었습니다.

그래서 플레이어 객체가 자신의 현재 회전 행렬을 가지고 있게 하면서 새로운 회전을 적용하기 전에 회전 행렬을 곱해서 이미 회전된 좌표축을 기준으로 회전할 수 있도록 만들었습니다.

플레이어와 다른 오브젝트 간의 충돌 검사는 OOBB를 이용했습니다. 그런데 월드에 존재하는 오브젝트(드론, 총알, 건물)이 꽤 많아서 프레임이 뚝 떨어지는 현상이 나타났습니다. 이 문제를 해결하기 위해서 저희 팀은 쿼드트리와 바운딩 구를 이용해서 계층적으로 충돌 체크를 했습니다.

모든 충돌 가능한 객체는 충돌 검사가 쉬운 바운딩 구를 1차적으로 가지고 있어서 바운딩 구 사이에 충돌 검사를 했습니다. 그 뒤에 바운딩 구가 충돌한 두 객체들 사이에 OOBB를 이용해서 정확한 충돌 검사를 해서 충돌 처리를 했습니다.

이렇게 해도 초당 프레임이 만족스럽지 않아서 정적인 객체인 건물들과의 충돌은 쿼드트리를 이용해서 공간을 나누어 충돌 검사 횟수를 대폭 줄였습니다.

게임프레임워크 제작은 시간 문제상 하지 못하였습니다.

* 팀원 간 작성한 내용
* 2014182021 소윤

바운딩 구 충돌체크가 적용된 오브젝트 객체 제작

OOBB 충돌체크가 적용된 큐브 오브젝트 제작

A\*알고리즘 및 노드를 벗어난 상황에선 장애물들을 체크하기 위해 OOBB Ray 충돌체크를 사용하는 드론 AI 제작

또한 드론 AI는 상태패턴을 이용하여 나름 FSM으로 제작

게임내의 건물 오브젝트제작 및 건물(정적인 객체)에 대한 충돌체크 쿼드트리(depth 4)로 구현

무기 로직 설계 및 구현

게임 효과 매니저 제작

Ending 화면 제작

* 2012182045 재용

플레이어 객체 모델링 및 내부(이동, 회전, 스킬, 체력 등) 구현

플레이어를 따라 이동하고 회전하는 카메라 구현

후방 카메라 기능 구현

맵 툴 제작 및 맵 디자인

링과 링 안의 아이템 모델링 및 구현

도로 모델링 및 배치

사운드 라이브러리 fmod를 통해 사운드 효과 구현

멀티 뷰를 이용한 미니맵과 UI 구현

사용자 입력 제어

Intro 화면 제작

팀원 사이의 개발 결과물 통합 및 관리(Git 및 Github 이용)

* 조작법

W: 가속

S: 감속

A, D: 플레이어 Z축으로 좌우 회전

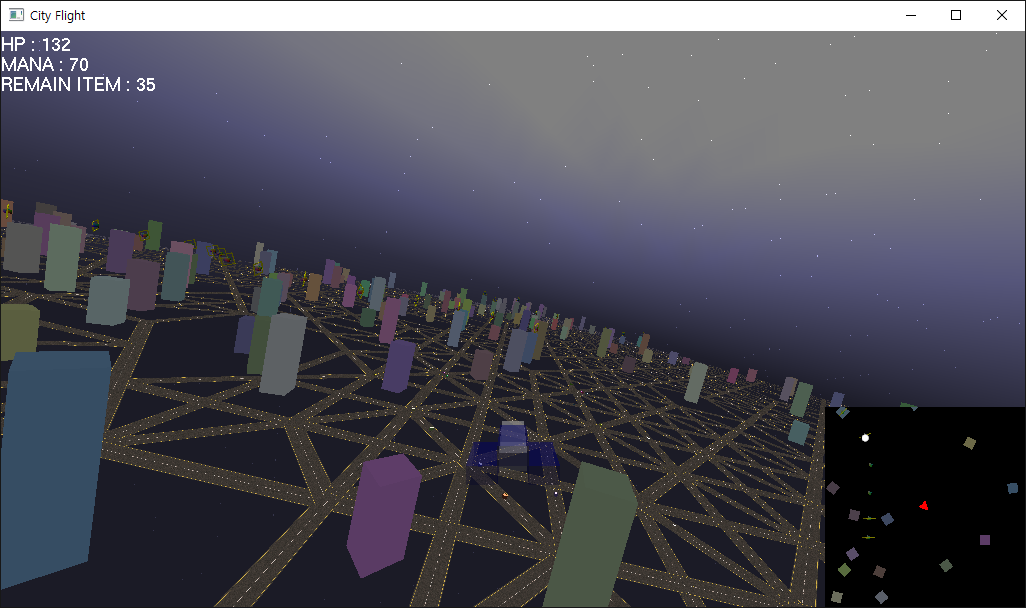
↑, ↓: 플레이어 X 축으로 상하 회전

→, ←: 플레이어 Y 축으로 좌우 회전

Q: 순간 가속

E: 스텔스 (반투명화)

F: 후방 카메라 보기

* 플레이 화면
* 프로젝트 개발 소감 및 후기
* 소윤

개인 적으로 이번 게임 제작이 그간 실험해 보고 싶었던 OOBB나, AI 상태패턴이나, A\*알고리즘 같은 것들을 구현해 볼 수 있는 기회가 돼주어 좋았습니다. 또한 3D환경에서 게임을 제작하다 보니 필요한 수학적 개념이 많아 졌는데 이 과정에 여러 가지 수학적 원리를 이곳 저곳 찾아가며 공부하는 것도 재미있었습니다.

이번 게임을 제작하면서 정말 제대로 게임을 만들어 보는 구나 하고 느낀 순간이 있었는데, bullet과 building들의 충돌체크를 할 때 갑자기 프레임 레이트가 확 떨어진 순간이 있었습니다. 어떻게 할까 고민하다 건물 오브젝트 충돌체크를 쿼드트리로 작성하였는데, 쿼드트리로 구현한 순간 프레임 레이트가 돌아왔습니다. 쿼드트리처럼 배운 것들이 절실히 필요한 순간이 많지 않은데 이번 게임을 만들면서는 그간 배운 것들을 사용해야만 하는 순간들이 여럿 있었습니다. 그리고 배운 것들이 실제로 돌아가는 것을 볼 때 굉장히 기뻤습니다.

처음 해보는 팀 프로젝트였는데 팀원이 제가 해보고 싶은 것들을 할 수 있도록 배려 해주고 제가 약한 부분에 대해 많이 보완해 주었습니다. 또한 각자 1씩 일을 해도 더한 결과가 2가 아닌 10이 되게 만들 수 있는 팀 프로젝트의 장점을 끌어내 주어 고마웠습니다.

일년 동안 다른 과목에 비해 힘들긴 했지만, 항상 그 학기를 뿌듯하게 끝낼 수 있게 해주셔서 감사합니다.

* 재용

항상 2차원에서만 놀다가 3차원으로 게임을 만들려다 보니 굉장히 생소하고 어려웠습니다. 개발하다 보니 결과물이 3D로 보이고 하니 점점 재미가 붙었습니다. 또한 3차원에서 객체들의 움직임을 표현하려다 보니 벡터와 행렬을 자주 다루게 되면서 3D 그래픽스에서 굉장히 중요한 개념이란 것을 깨달을 수 있었습니다.

팀 단위로 하는 프로젝트를 어떻게 하면 편하게 관리할까 하다가 Git과 Github를 써보기로 결정했습니다. 써보기 전에는 몰랐는데 실제로 사용해보니 개발에 굉장히 유용한 도구라는 것을 알게 되었습니다. 여러 방향으로 개발되어 있는 코드를 하나로 통합하기도 편리하고, 혹시 실수로 코드를 망가트렸더라도 항상 백업 파일이 시간대 별로 준비되어 있어서 안심하고 여러 종류의 테스트를 해볼 수 있었습니다.

태어나서 처음으로 팀 프로젝트를 한 것도 굉장히 뜻 깊은 경험이 되었습니다. 팀원과 서로 평소에 선호하는 코딩 스타일이 약간 달랐는데 많이 대화하면서 서로의 생각을 이해하고 의견을 맞춰 나가면서 처음 예상했던 것 보다 훨씬 뛰어난 결과물이 나오게 되었습니다.

제가 자신 없는 A\*나 OOBB 같은 알고리즘의 구현을 팀원이 기꺼이 맡아 주어서 너무 고마웠습니다. 덕분에 제가 만들고 싶은 비행기의 조작과 카메라를 플레이어에 고정하는 기능 등을 만들 수 있었습니다.

이번 텀 프로젝트로 여러가지로 깨닫고 성장할 수 있었습니다. 이런 귀중한 경험을 하게 기회를 주신 교수님께 감사드립니다!