[ 개발 문서 ]

FOCUS

프로젝트 진행 인원 : 1인

프로젝트 기간 : 2020.08.19. ~ 2020.09.15.

작성자 : 김예슬

목차

**1. 프로젝트 진행 1**

1) 기획 1

2) 개발 환경 1

3) 예상 일정 2

4) 실제 일정 3

5) 진행 결과 5

**2. 상세 설계**

1) 프로그램 구성 설계

2) 클래스 설계

**3. 프로젝트 흐름**

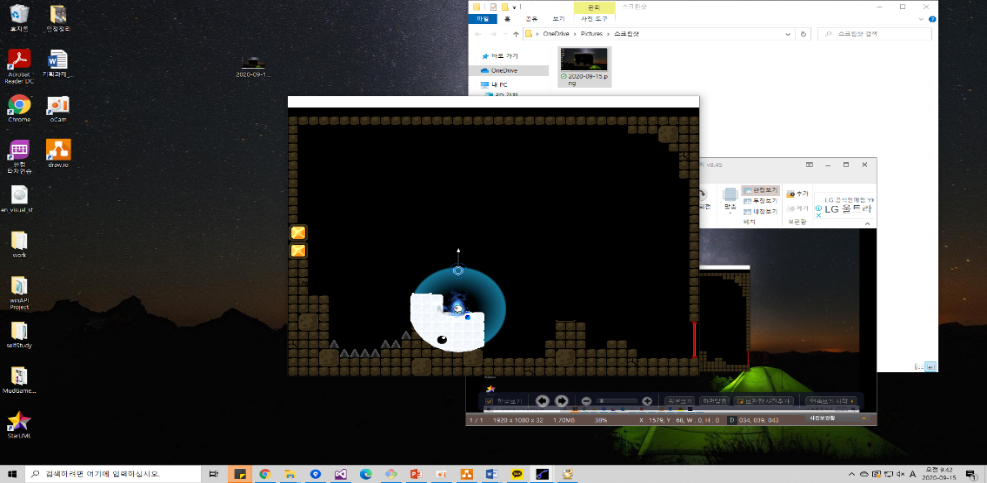
1) 순서도

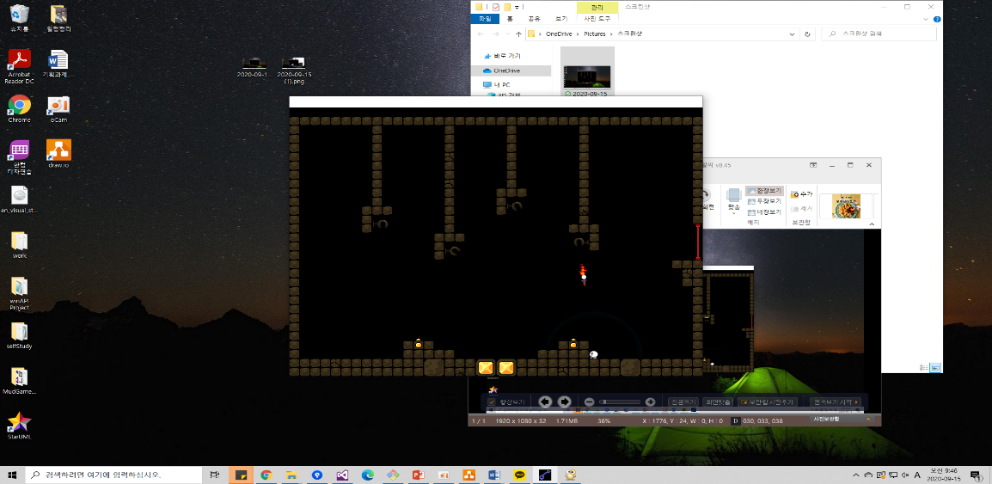
2) UML

**4. 함수 설명**

**5. 문제 해결 과정 및 소감 // 제출 시 제외**

**1. 프로젝트 진행**

1) 기획



WinAPI 프로젝트 주제로 Armor Games의 ‘Focus’란 게임을 모작 해보기로 했다. Focus는 플레이어를 조작해서 맵에 존재하는 버튼을 꺼서 게이트를 열어서 탈출하는 게임이다. 플레이어는 순간이동 능력을 사용할 수 있으며, 이 능력을 이용하여 갈 수 없는 지형을 가거나 범위 안에 들어온 투사체의 시간을 느리게 할 수 있다.

2) 개발 환경

|  |  |
| --- | --- |
| OS | Window10 |
| Development Tool | Visual Studio 2015 update 3 |
| Use | WinAPI |
| Language | C, C++ |
| Configuration Management | Git version 2.24.1.window.2  Github |
| Source(BGM, SEF, IMG) | Motion Element, Open Game Art |
| Else Tool | Clip Stuido, Asprite, GoldWave |

2) 예상 일정

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 일정표(1 Week) | | | | | | | | |
|  | 8/19 | 8/20 | 8/21 | 8/22 | 8/23 | 8/24 | 8/25 | 8/26 |
| 프로토 타입 제작 & 테스트 | \* |  |  |  |  |  |  |  |
| 프로젝트 기본 세팅 |  |  |  |  |  |  |  |  |
| 플레이어 조작 및 기능 |  |  |  |  |  |  |  |  |
| 적 구현\_가시 |  |  |  |  |  |  |  |  |
| 적 구현\_미사일 |  |  |  |  |  |  |  | \* |

\* : 발표일

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 일정표(2 Week) | | | | | | | |
|  | 8/27 | 8/28 | 8/29 | 8/30 | 8/31 | 9/1 | 9/2 |
| 적 구현\_미사일 |  |  |  |  | \* |  |  |
| 적 구현\_레이저 |  |  |  |  |  |  |  |
| 적 구현\_지뢰 |  |  |  |  |  |  |  |
| 스테이지 구현 |  |  |  |  |  |  | \* |

\* : 발표일

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 일정표(3 Week) | | | | | | | |
|  | 9/3 | 9/4 | 9/5 | 9/6 | 9/7 | 9/8 | 9/9 |
| 스테이지 구현 |  |  |  |  |  |  |  |
| UI 구현 |  |  |  |  |  |  |  |
| 최종 마무리 |  |  |  |  |  |  |  |
| 버그 픽스 |  |  |  |  |  |  |  |
| 개발 문서 정리 |  |  |  |  |  |  | \* |

\* : 발표일

4) 실제 일정

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 일정표(1 Week) | | | | | | | | |
|  | 8/19 | 8/20 | 8/21 | 8/22 | 8/23 | 8/24 | 8/25 | 8/26 |
| 프로토 타입 제작 & 테스트 | \* |  |  |  |  |  |  |  |
| 프로젝트 기본 세팅 |  |  |  |  |  |  |  |  |
| 리소스 탐색 및 제작 |  |  |  |  |  |  |  |  |
| 플레이어 조작 및 기능 |  |  |  |  |  |  |  |  |
| 기본 맵 설정 |  |  |  |  |  |  |  |  |
| 장애물\_가시 |  |  |  |  |  |  |  |  |
| 리팩토링 & 버그 수정 |  |  |  |  |  |  |  | \* |

\* : 발표일

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 일정표(2 Week) | | | | | | | |
|  | 8/27 | 8/28 | 8/29 | 8/30 | 8/31 | 9/1 | 9/2 |
| 리팩토링 & 버그 수정 |  |  |  |  | \* |  |  |
| 장애물\_대포, 미사일 발사 및 판정 |  |  |  |  |  |  |  |
| 장애물\_미사일, 폭발  On/Off 및 판정 |  |  |  |  |  |  |  |
| 리셋 구현 |  |  |  |  |  |  |  |
| 스위치 및 게이트 생성 |  |  |  |  |  |  |  |
| 이미지 추가 |  |  |  |  |  |  |  |

\* : 코로나로 인한 휴무 시작

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 일정표(3 Week) | | | | | | | |
|  | 9/3 | 9/4 | 9/5 | 9/6 | 9/7 | 9/8 | 9/9 |
| 리팩토링 & 버그 수정 |  |  |  |  |  |  |  |
| 게 내 이미지 추가 |  |  |  |  |  |  |  |
| 스테이지 구현\_데이터 파싱 |  |  |  |  |  |  |  |
| UI |  |  |  |  |  |  |  |
| BGM & SEF |  |  |  |  |  |  |  |
| 지형 추가\_반 블록 |  |  |  |  |  |  |  |
| 아이템 추가\_포커스 |  |  |  |  |  |  |  |

비고 : 코로나로 인한 휴무

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 일정표(4 Week) | | | | | | |
|  | 9/10 | 9/11 | 9/12 | 9/13 | 9/14 | 9/15 |
| 맵 툴 제작 |  |  |  |  |  |  |
| 밸런스 조정 |  |  |  |  |  |  |
| 키 입력 수정 |  |  |  |  |  |  |
| 애니메이션 수정 |  |  |  |  |  |  |
| 맵 디자인 |  |  |  |  |  |  |
| UI 수정 |  |  |  |  |  |  |
| 발표 준비 및 서류 제작 |  |  |  |  |  |  |

비고 : 추가 개발 진행

5) 진행 결과

|  |  |  |
| --- | --- | --- |
| 개발 기간 | 2020.08.19 ~ 2020.09.09. | Qr code  (영상 추가) |
| 추가 개발 기간 | 2020.09.09. ~ 2020.09.15. |
| GitHub URL | https://github.com/tireDep/InhaUniv\_winAPI\_Project |
| 게임 스크린 샷 | | | |
|  | | | |
| 맵 툴 스크린 샷 | | | |
|  | | | |

**2. 상세 설계**

1) 프로그램 구성 설계

[ 시작 화면 ]

* 세이브 데이터 확인, 존재 x 시 초기 값 세팅
* 아무 키 입력 시 화면 전환(게임)

[ 게임 화면 ]

* 맵 파싱
* 플레이어 이동 및 이동 제한
* 장애물 충돌 판정
* 미사일 발사 범위 내 판정
* 미사일(일반, 유도), 폭발 이펙트
* 플레이어 사망 시 스테이지 재 시작
* 다음 스테이지 불러오기, 존재하지 않으면 화면 전환(엔딩)
* 일시 정지(메인, 이어하기)

[ 엔딩 화면 ]

* 초기 값 세팅
* 끝나고 화면 전환(메인)

[ 기타 ]

* 게임 종료 시 세이브 데이터 저장
* 스테이지가 증가할수록 난이도 상승(총알 속도 증가 등)
* 플레이어의 상태에 따른 UI 출력



[ 맵 툴 ]

* 파일 열기, 다른 이름으로 저장
* 타일 배치 및 삭제(클릭, 드래그)
* 초기화

2) 클래스 설계

[ GameManger ]

* 싱글턴 패턴
* 현재 씬, 플레이어 생존 여부, 세이브 데이터 확인 등 게임 전체에 필요한 변수 필요

[ Object ]

* 순수 가상 클래스, 부모 클래스

[ Player ]

* 싱글턴 패턴
* 상하좌우, 포커스 모드의 조작
* 사망 시 스테이지 재시작

[ Map ]

* 싱글턴 패턴
* 데이터 파싱으로 맵 구현(타일, 대포, 아이템 등)
* 플레이어 리젠 위치
* 스테이지가 넘어가는 순간 다음 스테이지 정보 로드
* 플레이어와 장애물 충돌 판정(가시)

[ UI ]

* 싱글턴 패턴
* 현재 씬에 맞는 UI 출력

[ SoundSystem ]

* 싱글턴 패턴
* 현재 씬에 맞는 BGM 및 SEF 출력

[ Obstacle ]

* 장애물 관련 부모 클래스

[ Cannon ]

* 플레이어가 탐지 범위 내에 들어오면 피격되는지 판정 후 발사
* 안보이는 총알을 미리 발사해서 맞는 위치에 따른 판정

[ Bullet ]

* 싱글턴 패턴
* 미리 생성해 놓은 후 On/Off 하는 형식으로 발사
* 일반 총알, 유도 총알 2가지 필요
* 충돌 판정(플레이어, 스위치 없는 버튼)

[ Explode ]

* 싱글턴 패턴
* 미리 생성해 놓은 후 On/Off 하는 형식으로 구현
* 총알이 부딪힌 위치에 발생
* 충돌 판정(플레이어, 스위치 없는 버튼)



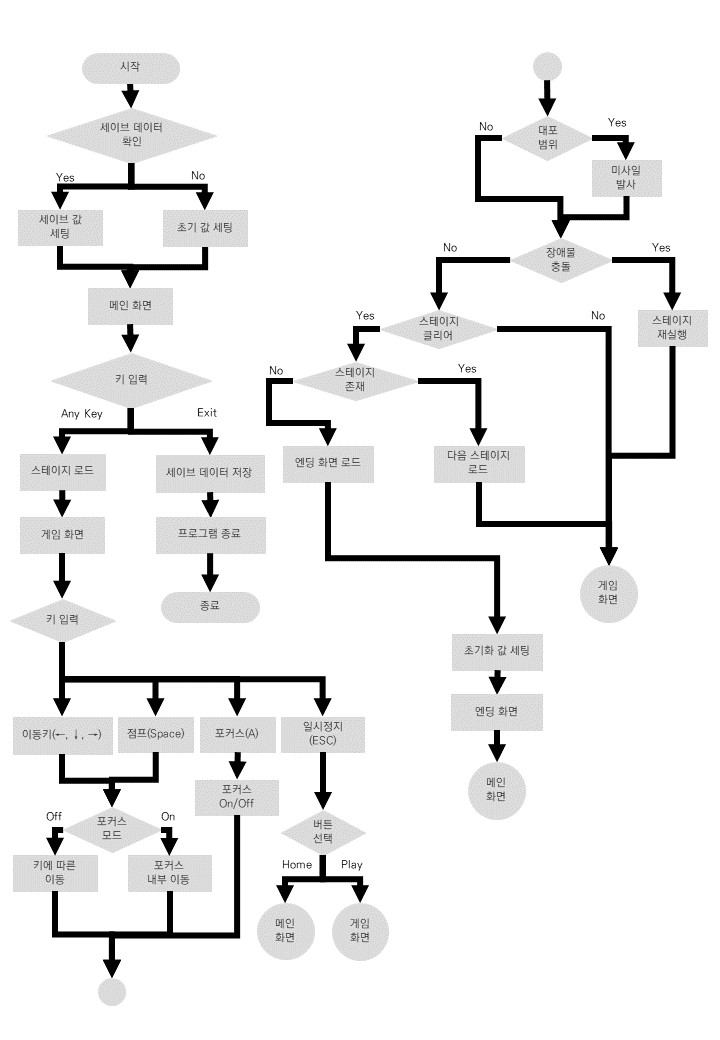
[ MapClass ]

* 좌 클릭 & 드래그 : 타입 선택, 해당 타입 맵에 배치
* 우 클릭 & 드래그 : 맵에 존재하는 타일 삭제
* 초기화 : 맵에 있는 타일 전체 삭제
* 파일 열기 : 만들어 놓은 맵 데이터 파일 열기
* 파일 저장 : binary 등을 적용한 맵 데이터 파일 저장

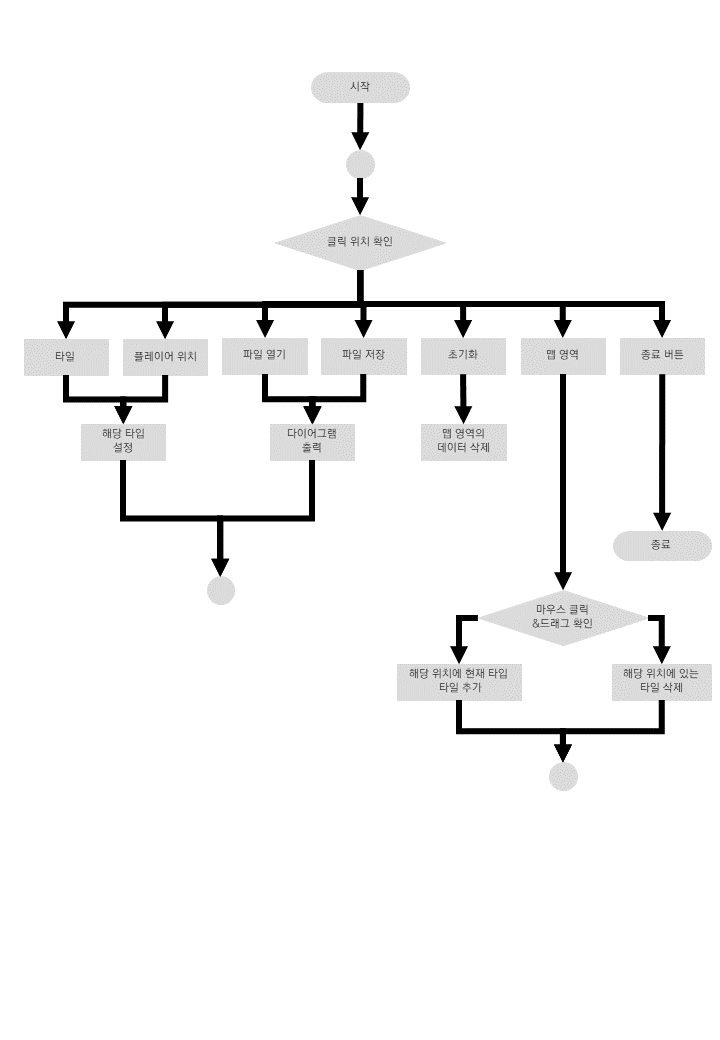
**3. 프로젝트 흐름**

1) 순서도

- FocusGame



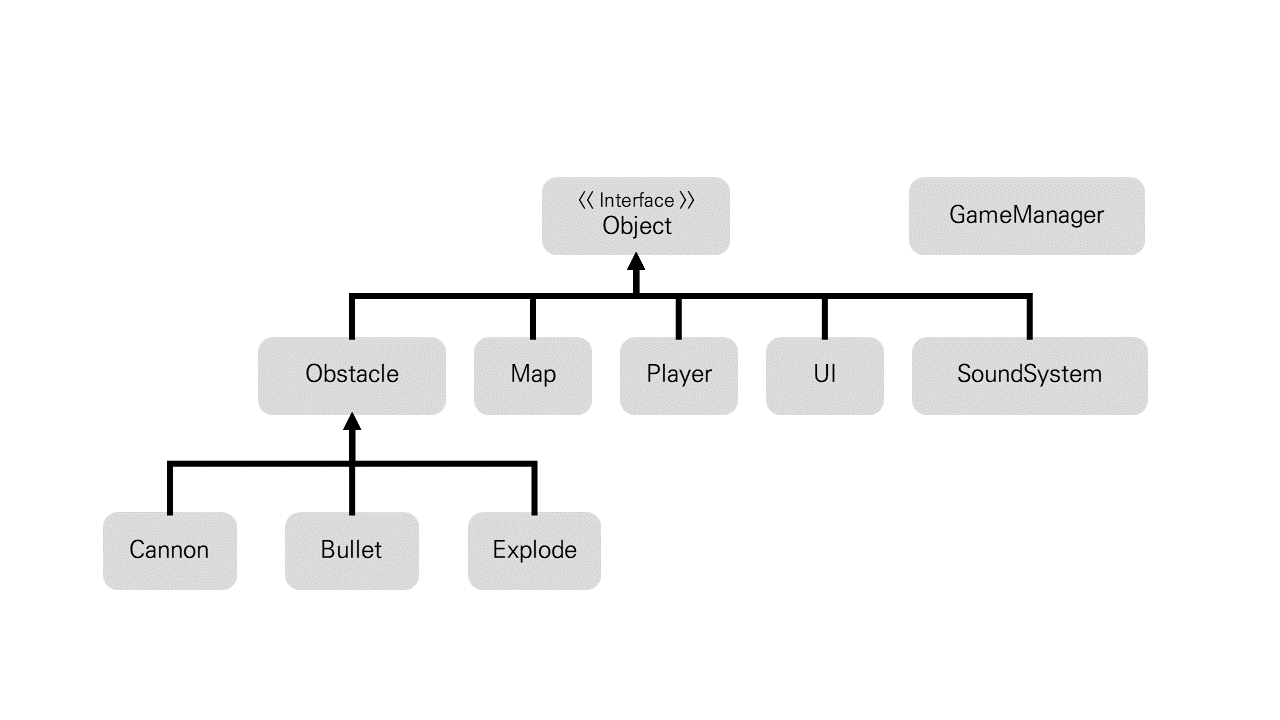
* FocusMapTool



2) UML

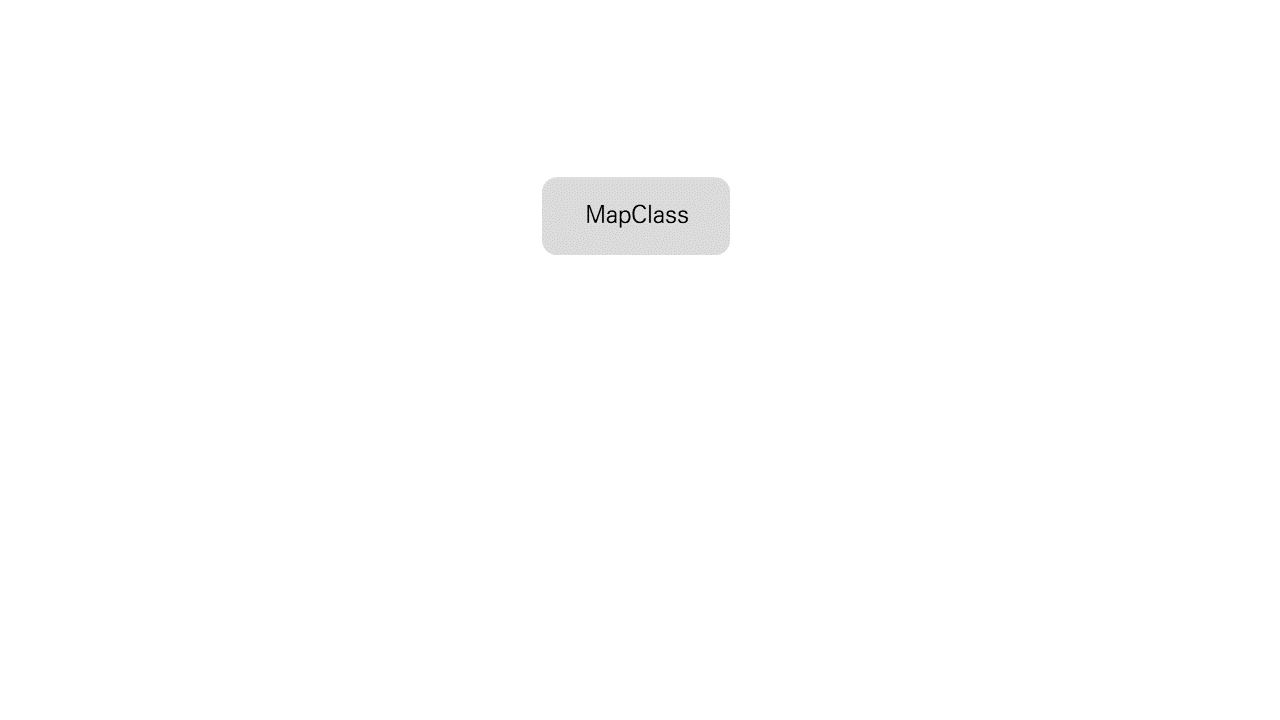
- 간략한 UML

[ FocusGame ]



포커스 게임의 클래스 중 Object는 순수 가상 클래스로 선언되어 있다.

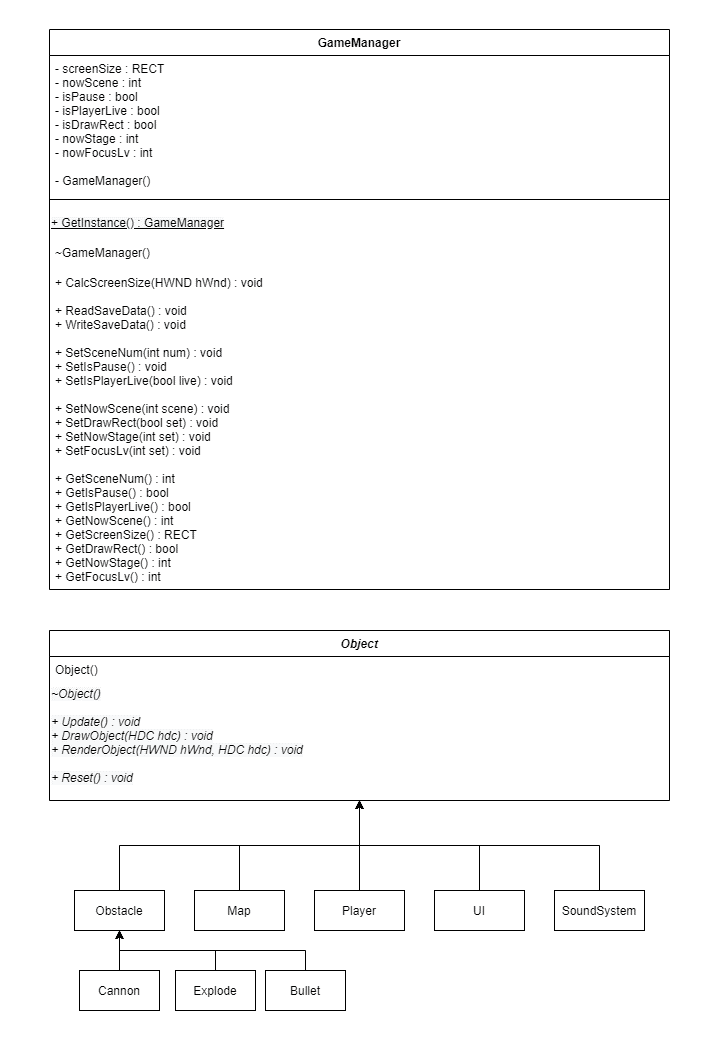
[ FocusMapTool ]

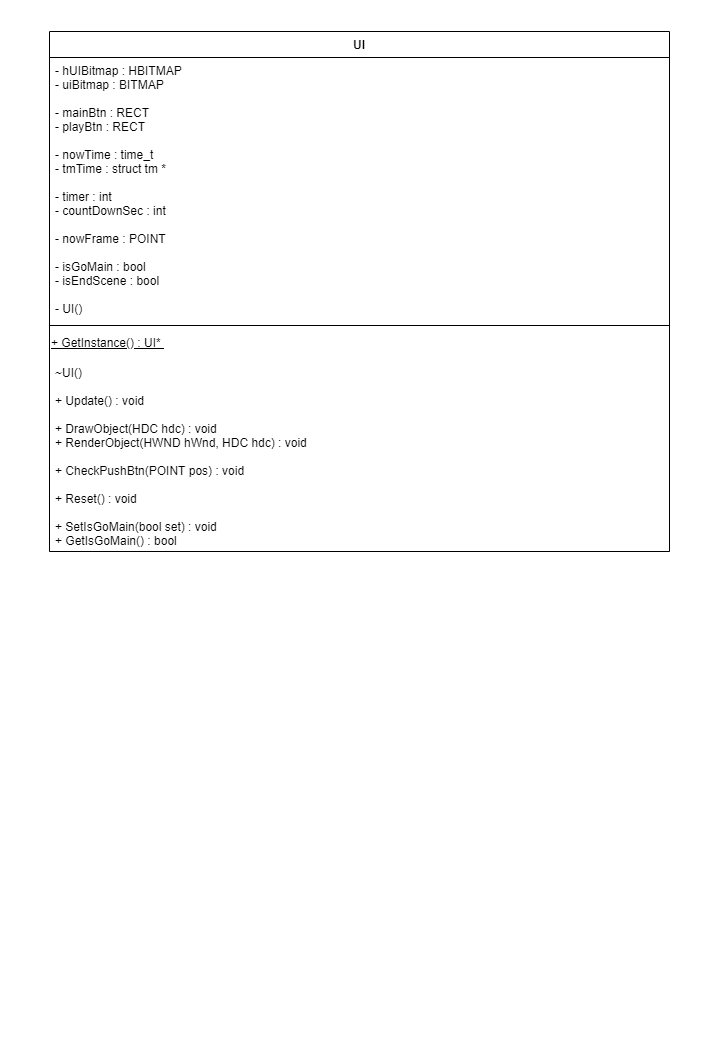
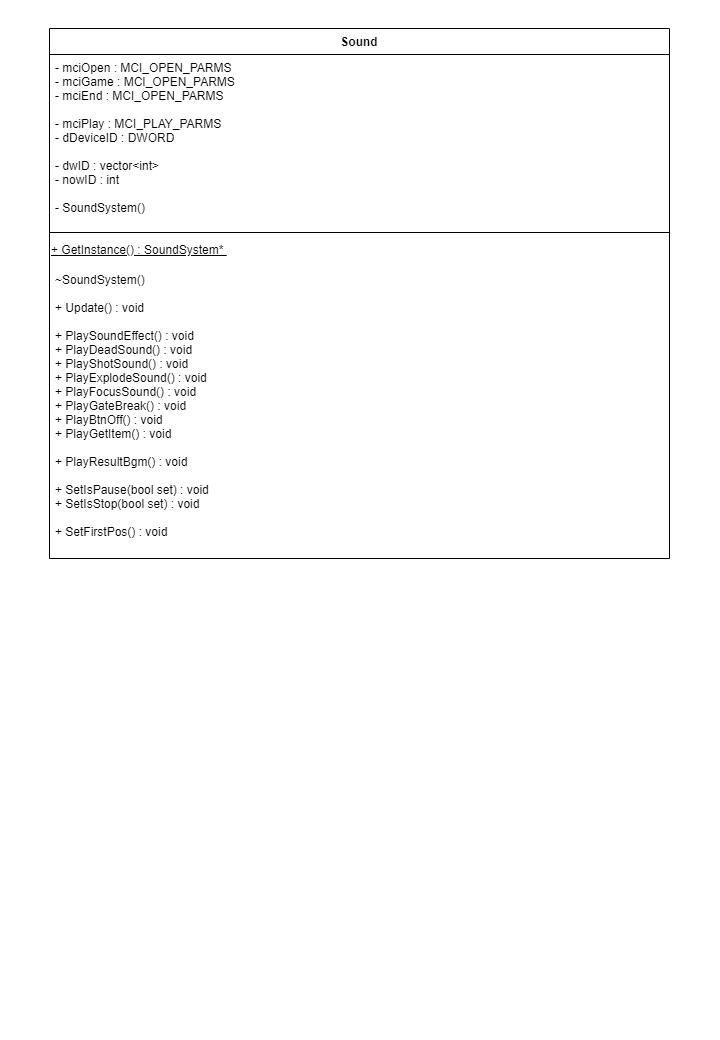
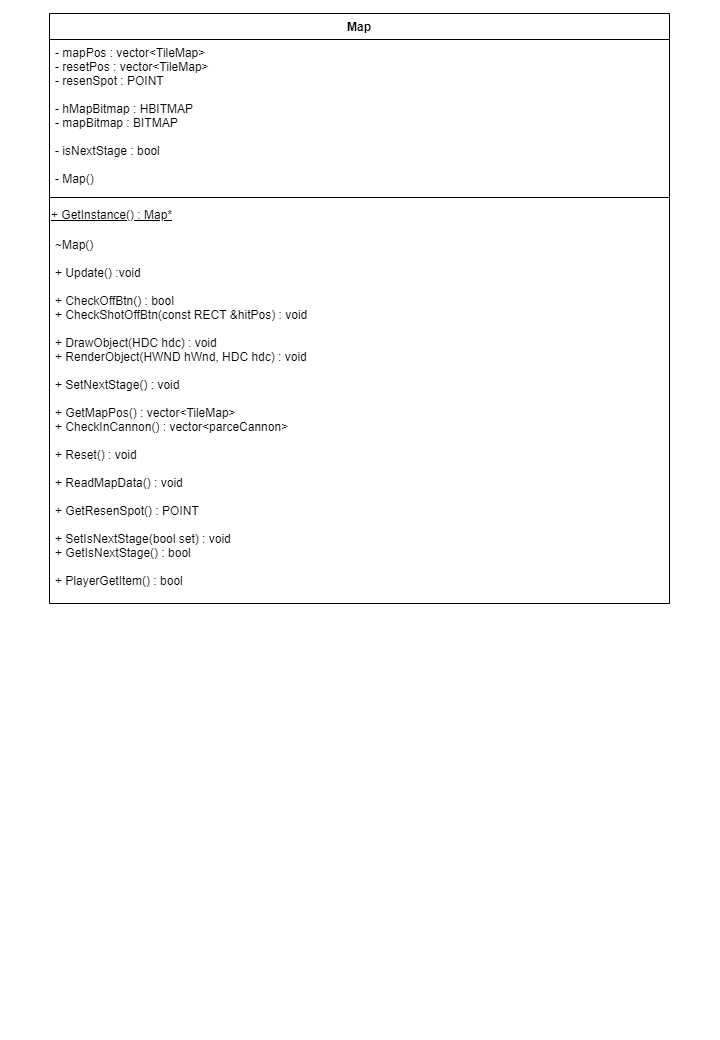
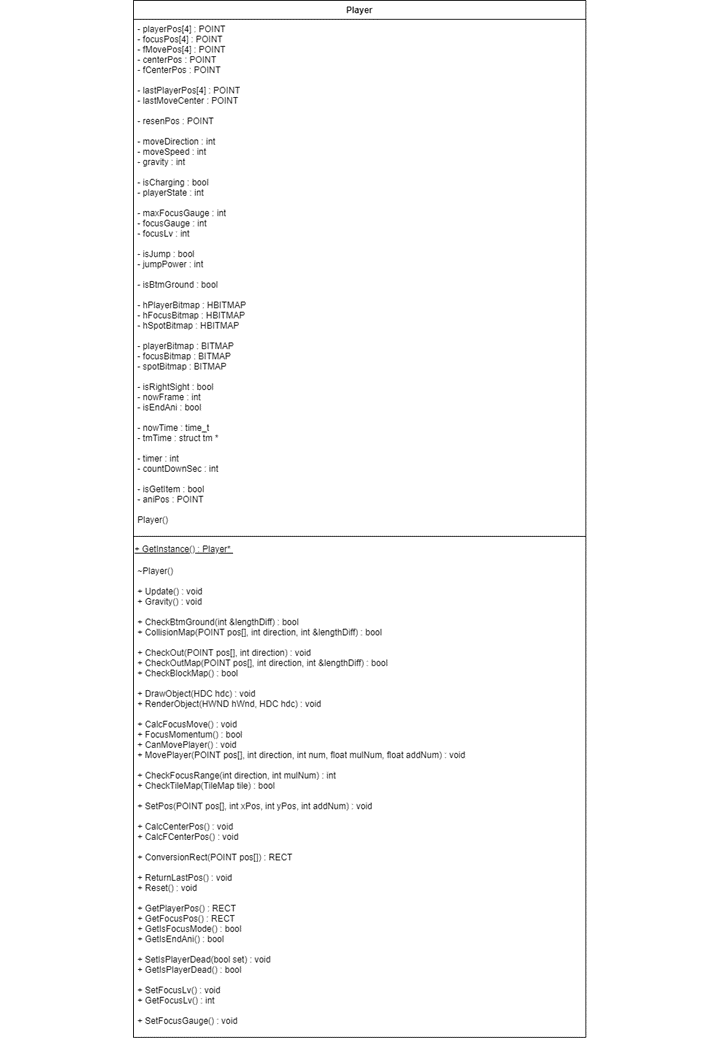
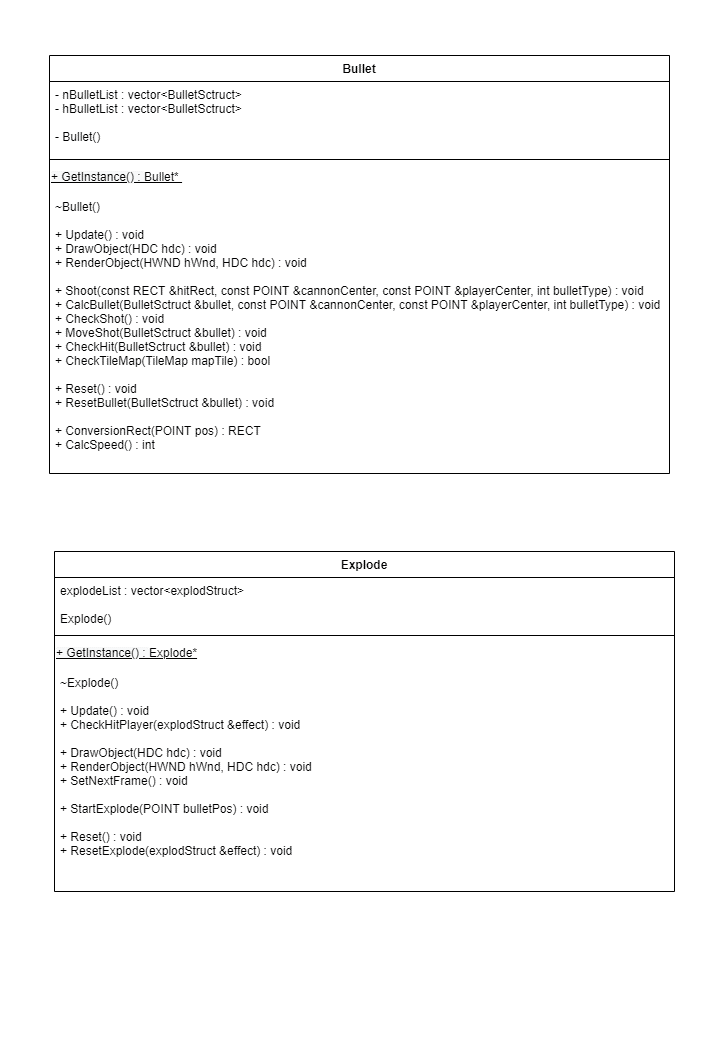
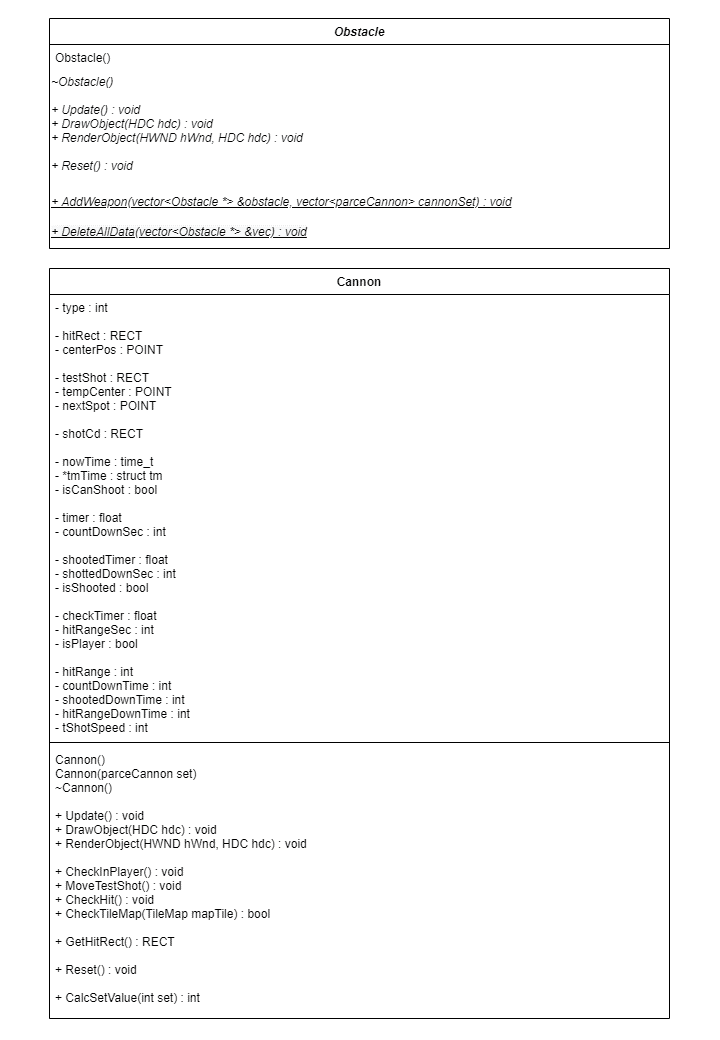


맵 툴의 클래스는 MapClass 단일 클래스로 이루어져 있다.

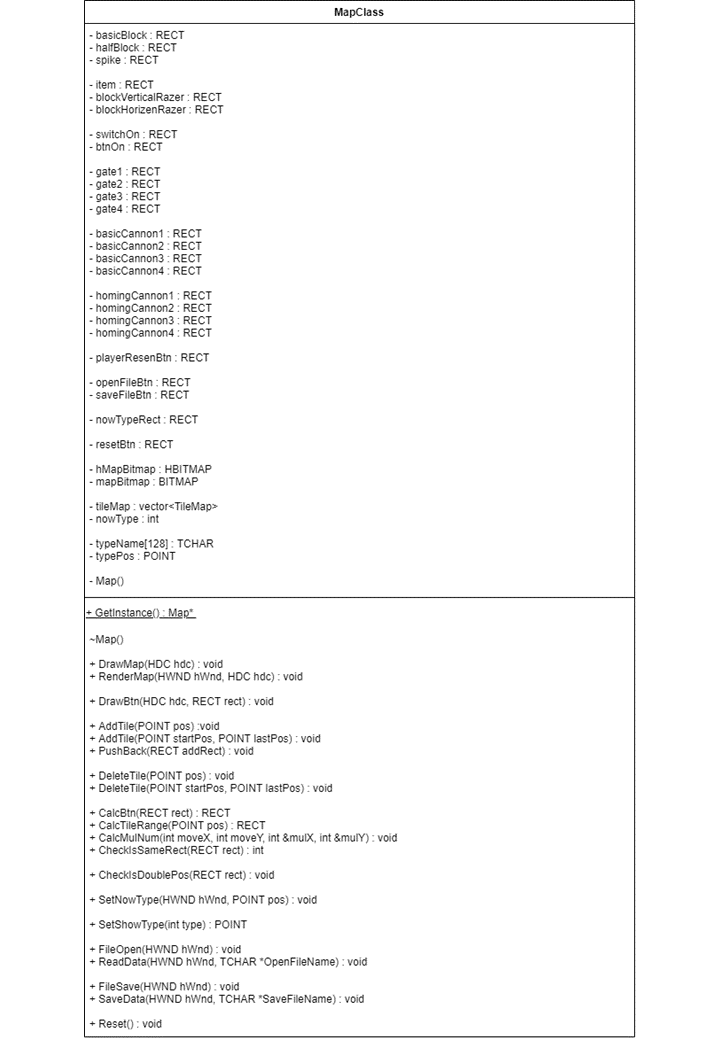
- 구체적인 UML

[ FocusGame ]

****

****

[ FocusMapTool ]

****

**4. 함수 설명**

[ GameManager ]

|  |  |
| --- | --- |
| 함수 | GameManager() |
| 해설 | 기본 생성자 |

|  |  |
| --- | --- |
| 함수 | ~GameManager() |
| 해설 | GameManager 기본 소멸자 |

|  |  |
| --- | --- |
| 함수 | static GameManager\* GetInstance() |
| 해설 | 스테틱 변수 반환 함수 |

|  |  |
| --- | --- |
| 함수 | void CalcScreenSize(HWND hWnd) |
| 해설 | 윈도우 창 크기 측정 함수  ( 내부 설명 추가 ) |

|  |  |
| --- | --- |
| 함수 | void ReadSaveData() |
| 해설 | 세이브 데이터 불러오는 함수  ( 내부 설명 추가 ) |

|  |  |
| --- | --- |
| 함수 |  |
| 매개변수 |  |
| 해설 |  |

**5. 문제 해결 과정 및 소감 // 제출 시 제외**

[ 문제 해결 과정 ]

* 설계 순서를 잘못 생각해서, 일을 효율적으로 하지 못했고 기능을 추가할 때 마다 예상치 못한 버그들이 많이 발생하였다. 효율적으로 개발하는 순서가 있다는 것을 느낄 수 있었다.
* 맵과 플레이어의 충돌 판정을 할 때, 플레이어에 닿아 있는 블록이 아닌 같은 라인에 있는 블록과 판정이 되어서 원하는 판정이 나오지 않는 문제가 있었다. 사각형의 충돌을 좌표 값을 이용한 계산이 아닌 IntersectRect() 함수를 이용해서 각각의 블록과 겹치는 부분이 있는지를 계산해서 충돌판정을 완료할 수 있었다.
* 플레이어 조작 중, 점프를 구현하는데 어려움이 있었다. 처음에는 좌표 값 변경으로 구현하였으나, 중력 값을 넣으면서 상당히 부자연스럽게 떨어진다는 것을 알 수 있었고, 수정하던 중 점프를 하는 동안에 미치는 힘을 계산하는 것이 생각보다 어려웠다. 여러 자료를 찾아보고 ‘점프하는 맨 처음 순간 힘 부과 -> 서서히 힘 감소 -> 최고점을 찍고 중력에 의해 내려옴’ 이라는 과정을 통해 점프를 더욱 자연스럽게 구현할 수 있었다.

[ 소감 ]

* 설계 미스, 많은 버그 발생, 코로나로 인한 휴무 등 예상치 못한 일이 많이 발생해서 일정표와 다르게 진행되었고, 예상 계획표를 작성할 때 이러한 부분도 생각해 두어야 한다는 것을 느꼈다. 또한, UI나 조작에 있어서 생각보다 많은 디테일이 들어가고, 이를 구현하는데도 시간이 생각보다 많이 소요됨을 느꼈다. 또한, 이러한 디테일을 구현할수록 게임의 완성도가 올라간다는 것을 느낄 수 있었다.
* 시간이 부족해서 사용해보고 싶었던 툴이나 디자인 패턴을 사용하지 못한 것, UI 이미지를 통으로 사용하는 등 아쉬운 부분이 많다.
* 메인-게임-엔딩으로 이어지는 완성된 게임을 구현하는 걸 목표로 삼았고, UI와 사운드까지 추가 해서 목표에 어느정도 부합한 거 같다.