

[ 개발 문서 ]

## FOCUS

프로젝트 진행 인원 : 1인

프로젝트 기간 : 2020.08.19. ~ 2020.09.15.

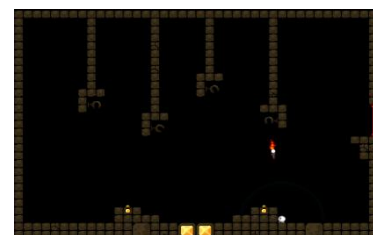
작성자 : 김예슬

# 목차

<b>1. 프로젝트 진행</b>	<b>1</b>
1) 기획	1
2) 개발 환경	1
3) 예상 일정	2
4) 실제 일정	3
5) 진행 결과	5
<b>2. 상세 설계</b>	<b>6</b>
1) 프로그램 구성 설계	6
2) 클래스 설계	8
<b>3. 프로젝트 흐름</b>	<b>11</b>
1) 순서도	11
2) UML	11
<b>4. 함수 설명</b>	<b>22</b>
1) FocusGame	22
2) MapTool	51
<b>5. 문제 해결 과정 및 소감 // 제출시 삭제</b>	<b>56</b>

## 1. 프로젝트 진행

### 1) 기획



WinAPI 프로젝트 주제로 Armor Games의 'Focus'란 게임을 모작 해보기로 했다. Focus는 플레이어의 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							
계 내 이미지 추가							
스테이지 구현_데이터 파싱							
UI							
BGM & SEF							
지형 추가_반 블록							
아이템 추가_포커스							

비고 : 코로나로 인한 휴무

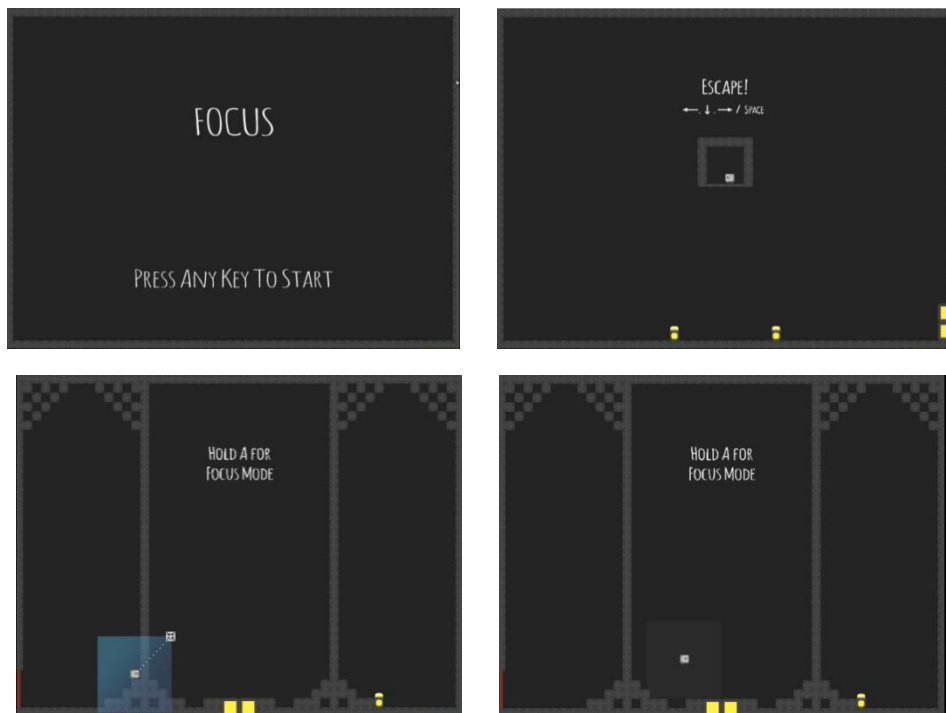
일정표(4 Week)						
	9/10	9/11	9/12	9/13	9/14	9/15
맵 툴 제작						
밸런스 조정						
키 입력 수정						
애니메이션 수정						
맵 디자인						
UI 수정						
발표 준비 및 서류 제작						

비고 : 추가 개발 진행

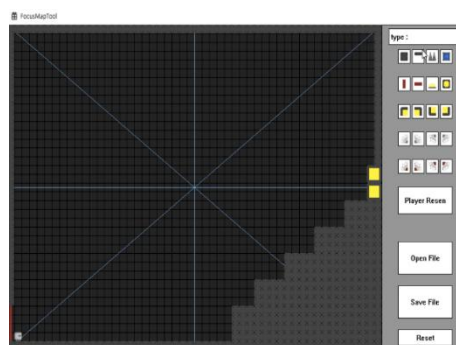
## 5) 진행 결과

개발 기간	2020.08.19. ~ 2020.09.09.	
추가 개발 기간	2020.09.09. ~ 2020.09.15.	
GitHub URL	<a href="https://github.com/tireDep/InhaUniv_winAPI_Project">https://github.com/tireDep/InhaUniv_winAPI_Project</a>	
영상 URL	<a href="https://blog.naver.com/wing997/222095747202">https://blog.naver.com/wing997/222095747202</a>	영상 및 자료

### 게임 스크린 샷



### 맵 툴 스크린 샷



## 2. 상세 설계

### 1) 프로그램 구성 설계

[ FocusGame ]

#### 〉 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(미사일 속도 증가 등)
  - 플레이어의 상태에 따른 UI 출력
- 

## [ MapTool ]

## 〉 맵 툴

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

## 2) 클래스 설계

### [ FocusGame ]

#### > 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 미사일을 미리 발사해서 맞는 위치에 따른 판정(플레이어 피격 -> 발사)

## > Bullet

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

## › Explode

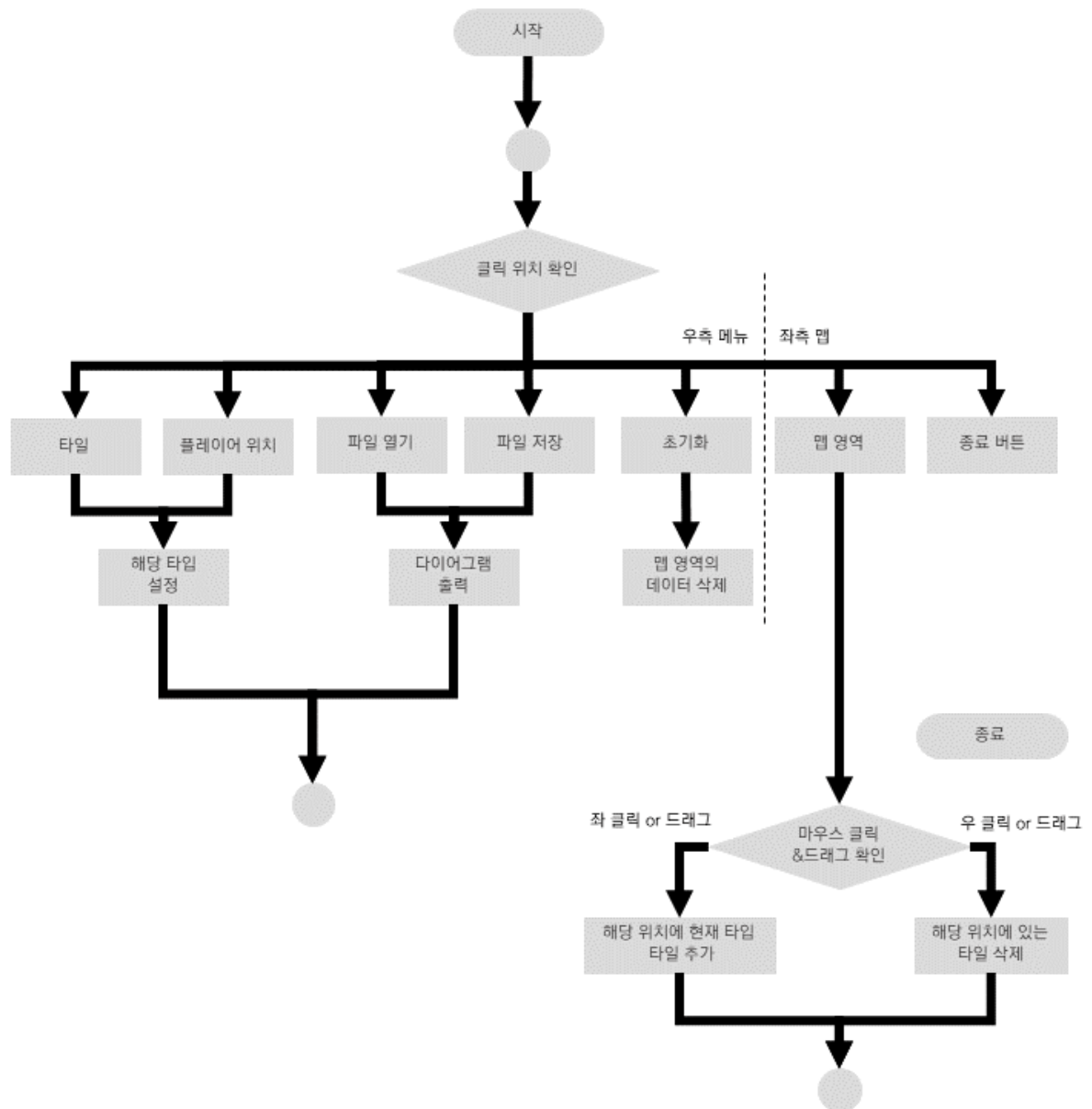
- 싱글턴 패턴
  - 미리 생성해 놓은 후 On/Off 하는 형식으로 구현
  - 미사일이 부딪힌 위치에 발생
  - 충돌 판정(플레이어, 스위치 없는 버튼)
- 

## 2. MapTool

### › 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

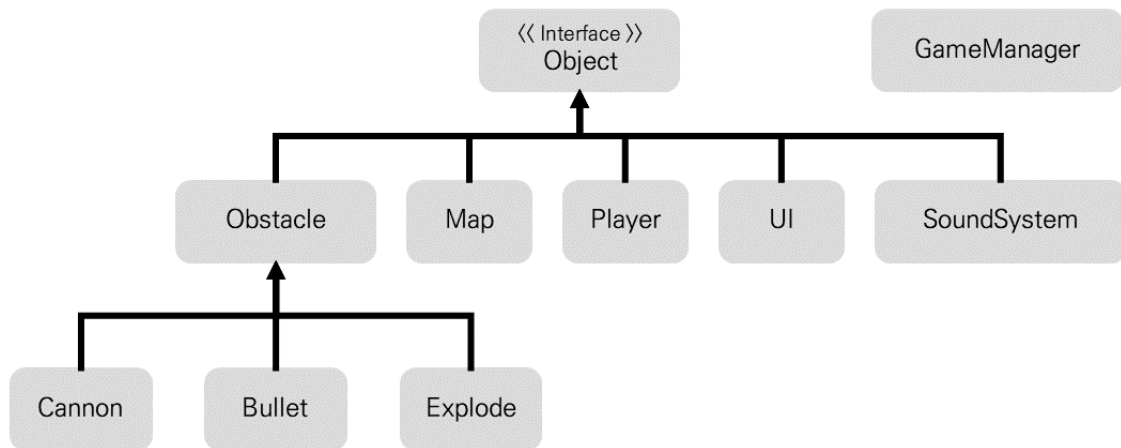




## 2) UML

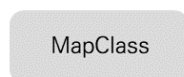
- 간략한 UML

[ FocusGame ]



포커스 게임의 클래스 중 Object는 순수 가상 클래스로 선언되어 있다.

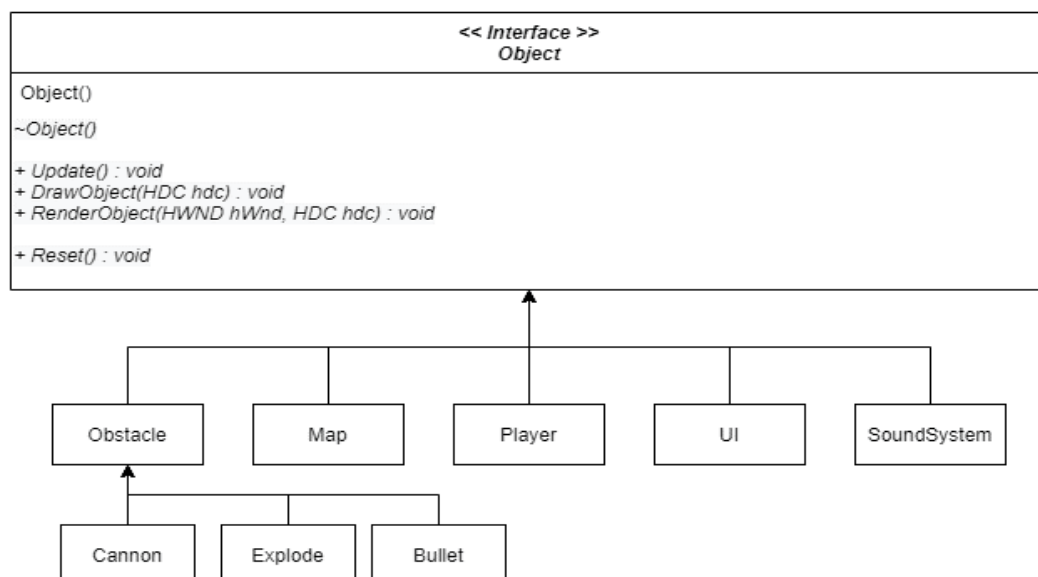
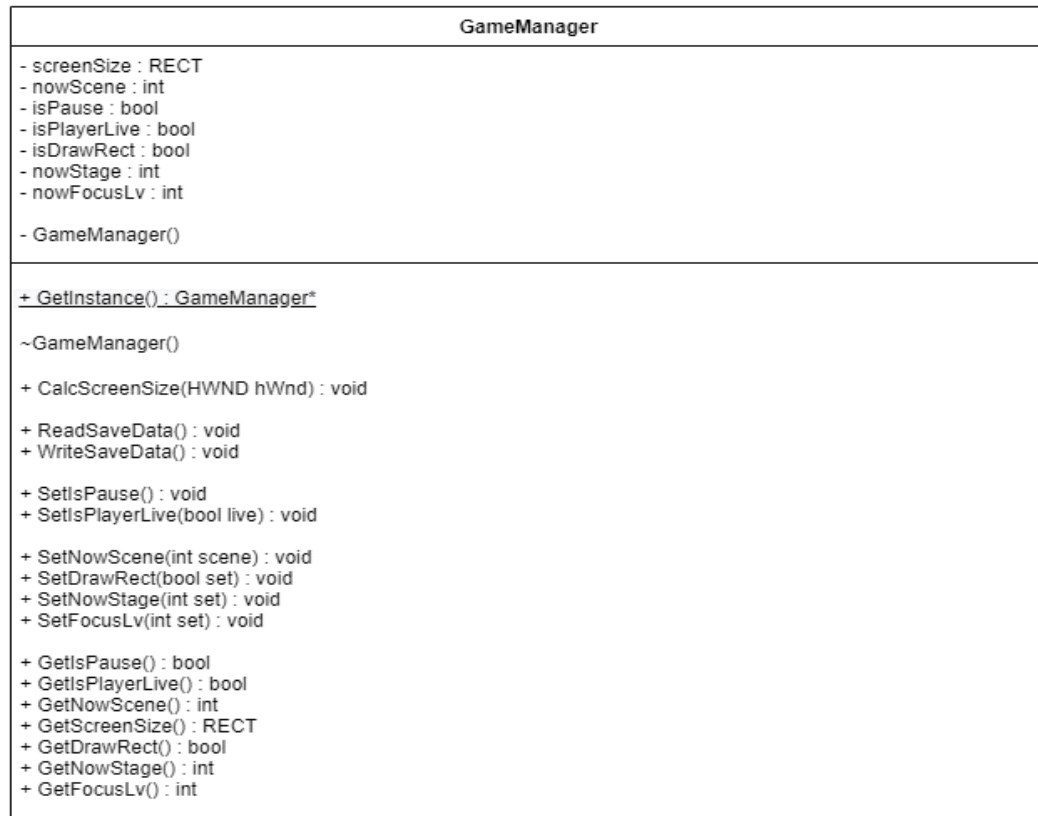
[ FocusMapTool ]



맵 툴의 클래스는 MapClass 단일 클래스로 이루어져 있다.

- 구체적인 UML

[ FocusGame ]





<i>Obstacle</i>
Obstacle() ~Obstacle() + Update() : void + DrawObject(HDC hdc) : void + RenderObject(HWND hWnd, HDC hdc) : void + Reset() : void + AddWeapon(vector<Obstacle *> &obstacle, vector<ParceCannoneStruct> cannonSet) : void + DeleteAllData(vector<Obstacle *> &vec) : void

<i>Cannon</i>
- type : int - hitRect : RECT - centerPos : POINT - testShot : RECT - tempCenter : POINT - nextSpot : POINT - shotCd : RECT - nowTime : time_t - *tmTime : struct tm - isCanShoot : bool - timer : float - countDownSec : int - shootedTimer : float - shootedDownSec : int - isShooted : bool - checkTimer : float - hitRangeSec : int - isPlayer : bool - hitRange : int - countDownTime : int - shootedDownTime : int - hitRangeDownTime : int - tShotSpeed : int
Cannon() Cannon(ParceCannonStruct set) ~Cannon() + Update() : void + DrawObject(HDC hdc) : void + RenderObject(HWND hWnd, HDC hdc) : void + CheckInPlayer() : void + MoveTestShot() : void + CheckHit() : void + CheckTileMap(TileMap mapTile) : bool + GetHitRect() : RECT + Reset() : void + CalcSetValue(int set) : int

Bullet
- nBulletList : vector<BulletSctruct> - hBulletList : vector<BulletSctruct>  - Bullet()
+ <u>GetInstance() : Bullet*</u>  ~Bullet()  + Update() : void + DrawObject(HDC hdc) : void + RenderObject(HWND hWnd, HDC hdc) : void  + Shoot(const RECT &hitRect, const POINT &cannonCenter, const POINT &playerCenter, int bulletType) : void + CalcBullet(BulletSctruct &bullet, const POINT &cannonCenter, const POINT &playerCenter, int bulletType) : void + CheckShot() : void + MoveShot(BulletSctruct &bullet) : void + CheckHit(BulletSctruct &bullet) : void + CheckTileMap(TileMap mapTile) : bool  + Reset() : void + ResetBullet(BulletSctruct &bullet) : void  + ConversionRect(POINT pos) : RECT + CalcSpeed() : int

Explode
explodeList : vector<explodStruct>  Explode()
+ <u>GetInstance() : Explode*</u>  ~Explode()  + Update() : void + CheckHitPlayer(explodStruct &effect) : void  + DrawObject(HDC hdc) : void + RenderObject(HWND hWnd, HDC hdc) : void + SetNextFrame() : void  + StartExplode(POINT bulletPos) : void  + Reset() : void + ResetExplode(explodStruct &effect) : void

Player
<ul style="list-style-type: none"> <li>- playerPos[4] : POINT</li> <li>- focusPos[4] : POINT</li> <li>- fMovePos[4] : POINT</li> <li>- centerPos : POINT</li> <li>- fCenterPos : POINT</li> <li>- lastPlayerPos[4] : POINT</li> <li>- lastMoveCenter : POINT</li> <li>- resenPos : POINT</li> <li>- moveDirection : int</li> <li>- moveSpeed : int</li> <li>- gravity : int</li> <li>- isCharging : bool</li> <li>- playerState : int</li> <li>- maxFocusGauge : int</li> <li>- focusGauge : int</li> <li>- focusLv : int</li> <li>- isJump : bool</li> <li>- jumpPower : int</li> <li>- isBtmGround : bool</li> <li>- hPlayerBitmap : HBITMAP</li> <li>- hFocusBitmap : HBITMAP</li> <li>- hSpotBitmap : HBITMAP</li> <li>- playerBitmap : BITMAP</li> <li>- focusBitmap : BITMAP</li> <li>- spotBitmap : BITMAP</li> <li>- isRightSight : bool</li> <li>- nowFrame : int</li> <li>- isEndAni : bool</li> <li>- nowTime : time_t</li> <li>- tmTime : struct tm *</li> <li>- timer : int</li> <li>- countDownSec : int</li> <li>- isGetItem : bool</li> <li>- aniPos : POINT</li> </ul>
Player()  <u>+GetInstance(): Player*</u>  ~Player()  + Update(): void + Gravity(): void  + CheckBtmGround(int &lengthDiff): bool + CollisionMap(int direction, int &lengthDiff): bool  + CheckOut(POINT pos[], int direction): void + CheckOutMap(POINT pos[], int direction, int &lengthDiff): bool + CheckBlockMap(): bool  + DrawObject(HDC hdc): void + RenderObject(HWND hWnd, HDC hdc): void  + CalcFocusMove(): void + FocusMomentum(): bool + CanMovePlayer(): void + MovePlayer(POINT pos[], int direction, int num, float mulNum, float addNum): void  + CheckFocusRange(int direction, int mulNum): int + CheckTileMap(TileMap tile): bool  + SetPos(POINT pos[], int xPos, int yPos, int addNum): void  + CalcCenterPos(): void + CalcFCenterPos(): void  + ConversionRect(POINT pos[]): RECT  + ReturnLastPos(): void + Reset(): void  + GetPlayerPos(): RECT + GetFocusPos(): RECT + GetIsFocusMode(): bool + GetIsEndAni(): bool  + SetIsPlayerDead(bool set): void + GetIsPlayerDead(): bool  + SetFocusLv(): void + GetFocusLv(): int  + SetFocusGauge(): void

Map
<ul style="list-style-type: none"> <li>- mapPos : vector&lt;TileMap&gt;</li> <li>- resetPos : vector&lt;TileMap&gt;</li> <li>- resenSpot : POINT</li> <li>- hMapBitmap : HBITMAP</li> <li>- mapBitmap : BITMAP</li> <li>- isNextStage : bool</li> <li>- Map()</li> </ul>
<ul style="list-style-type: none"> <li>+ <u>GetInstance() : Map*</u></li> <li>~Map()</li> <li>+ Update() : void</li> <li>+ CheckOffBtn() : bool</li> <li>+ CheckShotOffBtn(const RECT &amp;hitPos) : void</li> <li>+ DrawObject(HDC hdc) : void</li> <li>+ RenderObject(HWND hWnd, HDC hdc) : void</li> <li>+ SetNextStage() : void</li> <li>+ GetMapPos() : vector&lt;TileMap&gt;</li> <li>+ CheckInCannon() : vector&lt;ParceCannonStruct&gt;</li> <li>+ Reset() : void</li> <li>+ ReadMapData() : void</li> <li>+ GetResenSpot() : POINT</li> <li>+ SetIsNextStage(bool set) : void</li> <li>+ GetIsNextStage() : bool</li> <li>+ PlayerGetItem() : bool</li> </ul>

Sound
<ul style="list-style-type: none"> <li>- mciOpen : MCI_OPEN_PARMS</li> <li>- mciGame : MCI_OPEN_PARMS</li> <li>- mciEnd : MCI_OPEN_PARMS</li> <li>- mciPlay : MCI_PLAY_PARMS</li> <li>- dDeviceID : DWORD</li> <li>- dwID : vector&lt;int&gt;</li> <li>- nowID : int</li> <li>- SoundSystem()</li> </ul>
<ul style="list-style-type: none"> <li>+ <u>GetInstance() : SoundSystem*</u></li> <li>~SoundSystem()</li> <li>+ Update() : void</li> <li>+ PlayChangeStageSound() : void</li> <li>+ PlayDeadSound() : void</li> <li>+ PlayShotSound() : void</li> <li>+ PlayExplodeSound() : void</li> <li>+ PlayFocusSound() : void</li> <li>+ PlayGateBreakSound() : void</li> <li>+ PlayBtnOffSound() : void</li> <li>+ PlayGetItemSound() : void</li> <li>+ PlayResultBgm() : void</li> <li>+ SetIsPause(bool set) : void</li> <li>+ SetIsStop(bool set) : void</li> <li>+ SetFirstPos() : void</li> <li>+ DrawObject(HDC hdc) : void</li> <li>+ RenderObject(HWND hWnd, HDC hdc)</li> <li>+ Reset() : void</li> </ul>

UI
<ul style="list-style-type: none"> <li>- hUIBitmap : HBITMAP</li> <li>- uiBitmap : BITMAP</li> <li>- mainBtn : RECT</li> <li>- playBtn : RECT</li> <li>- nowTime : time_t</li> <li>- tmTime : struct tm *</li> <li>- timer : int</li> <li>- countDownSec : int</li> <li>- nowFrame : POINT</li> <li>- isGoMain : bool</li> <li>- isEndScene : bool</li> <li>- UI()</li> </ul>
<ul style="list-style-type: none"> <li><u>+GetInstance() : UI*</u></li> <li>~UI()</li> <li>+ Update() : void</li> <li>+ DrawObject(HDC hdc) : void</li> <li>+ RenderObject(HWND hWnd, HDC hdc) : void</li> <li>+ CheckPushBtn(POINT pos) : void</li> <li>+ Reset() : void</li> <li>+ SetIsGoMain(bool set) : void</li> <li>+ GetIsGoMain() : bool</li> </ul>

## [ FocusMapTool ]

MapClass
<ul style="list-style-type: none"> <li>- basicBlock : RECT</li> <li>- halfBlock : RECT</li> <li>- spike : RECT</li> <li>- item : RECT</li> <li>- blockVerticalRazer : RECT</li> <li>- blockHorizenRazer : RECT</li> <li>- switchOn : RECT</li> <li>- btnOn : RECT</li> <li>- gate1 : RECT</li> <li>- gate2 : RECT</li> <li>- gate3 : RECT</li> <li>- gate4 : RECT</li> <li>- basicCannon1 : RECT</li> <li>- basicCannon2 : RECT</li> <li>- basicCannon3 : RECT</li> <li>- basicCannon4 : RECT</li> <li>- homingCannon1 : RECT</li> <li>- homingCannon2 : RECT</li> <li>- homingCannon3 : RECT</li> <li>- homingCannon4 : RECT</li> <li>- playerResenBtn : RECT</li> <li>- openFileBtn : RECT</li> <li>- saveFileBtn : RECT</li> <li>- nowTypeRect : RECT</li> <li>- resetBtn : RECT</li> <li>- hMapBitmap : HBITMAP</li> <li>- mapBitmap : BITMAP</li> <li>- tileMap : vector&lt;TileMap&gt;</li> <li>- nowType : int</li> <li>- typeName[128] : TCHAR</li> <li>- typePos : POINT</li> <li>- Map()</li> </ul>
<ul style="list-style-type: none"> <li>+ <u>GetInstance() : Map*</u></li> <li>~Map()</li> <li>+ DrawMap(HDC hdc) : void</li> <li>+ RenderMap(HWND hWnd, HDC hdc) : void</li> <li>+ DrawBtn(HDC hdc, RECT rect) : void</li> <li>+ AddTile(POINT pos) : void</li> <li>+ AddTile(POINT startPos, POINT lastPos) : void</li> <li>+ PushBack(RECT addRect) : void</li> <li>+ DeleteTile(POINT pos) : void</li> <li>+ DeleteTile(POINT startPos, POINT lastPos) : void</li> <li>+ CalcBtn(RECT rect) : RECT</li> <li>+ CalcTileRange(POINT pos) : RECT</li> <li>+ CalcMulNum(int moveX, int moveY, int &amp;mulX, int &amp;mulY) : void</li> <li>+ CheckIsSameRect(RECT rect) : int</li> <li>+ CheckIsDoublePos(RECT rect) : void</li> <li>+ SetNowType(HWND hWnd, POINT pos) : void</li> <li>+ SetShowType(int type) : POINT</li> <li>+ FileOpen(HWND hWnd) : void</li> <li>+ ReadData(HWND hWnd, TCHAR *OpenFileName) : void</li> <li>+ FileSave(HWND hWnd) : void</li> <li>+ SaveData(HWND hWnd, TCHAR *SaveFileName) : void</li> <li>+ Reset() : void</li> </ul>

## 4. 함수 설명

### 1) FocusGame

[ GameManager ]

함수	GameManager()
해설	기본 생성자

함수	~GameManager()
해설	기본 소멸자

함수	static GameManager* GetInstance()
해설	static 변수 반환 함수

함수	void CalcScreenSize(HWND hWnd)
해설	윈도우 창 크기 측정 함수. 내부 변수에 값 저장

함수	void ReadSaveData()
해설	세이브 데이터 불러오는 함수. (현재 스테이지와 포커스 레벨) 세이브 데이터가 있을 경우 해당 값을, 없을 경우 초기값을 내부 변수에 저장

함수	void WriteSaveData()
해설	게임이 종료될 때, 세이브 데이터를 저장하는 함수. (현재 스테이지와 포커스 레벨)

함수	void SetIsPause()
해설	현재 일시정지 상태가 아니면 일시정지 설정, 아니면 일시정지를 해제하는 함수, 일시 정지 상태에 따라 BGM이 재생 유무가 결정됨



함수	void SetNowScene(int scene)
매개변수	int scene : 현재 씬 넘버
해설	내부변수에 현재 씬 넘버를 저장하는 함수.

함수	void SetIsPlayerLive(bool live)
매개변수	bool live : 플레이어 생존 여부
해설	내부 변수에 플레이어 생존 여부 값 저장 함수

함수	void SetIsPlayerLive(bool live)
매개변수	bool live : 플레이어 생존 여부
해설	내부 변수에 플레이어 생존 여부 값 저장 함수

함수	bool GetIsPause()
해설	일시정지 상태 값 반환 함수.

함수	bool GetNowScene()
해설	현재 씬 넘버 값 반환 함수

함수	RECT GetScreenSize()
해설	화면 크기 값 반환 함수

함수	void SetDrawRect(bool set)
매개변수	bool set : 모드 설정 on/off
해설	디버그 모드 설정 함수. true일 경우 물체의 히트박스를 출력함.

함수	void SetNowStage(int set)
매개변수	int set : 현재 스테이지 넘버
해설	내부 변수에 현재 스테이지 넘버를 저장 함수

함수	void SetFocusLv(int set)
매개변수	int set : 현재 포커스 레벨
해설	내부 변수에 현재 포커스 레벨 저장 함수

함수	bool GetDrawRect()
해설	디버그 모드 설정 값 반환 함수

함수	void GetNowStage()
해설	현재 스테이지 넘버 반환 함수

함수	void GetFocusLv()
해설	현재 포커스 레벨 값 반환 함수

[ ObjectClass ]

함수	Object()
해설	기본 생성자

함수	virtual ~Object() = 0
해설	기본 소멸자. 순수 가상 함수

함수	virtual void Update() = 0
해설	업데이트 함수. 순수 가상 함수

함수	virtual void DrawObject(HDC hdc) = 0
매개변수	HDC hdc : DC 핸들
해설	히트박스 출력 함수. 순수 가상 함수

함수	virtual void RenderObject(HWND hWnd, HDC hdc) = 0
매개변수	HWND hWnd : 윈도우 핸들 HDC hdc : DC 핸들
해설	이미지 출력 함수. 순수 가상 함수

함수	virtual void Reset() = 0
해설	초기화 함수. 순수 가상 함수

[ PlayerClass ]

함수	Player()
해설	기본 생성자

함수	~Player()
해설	기본 소멸자

함수	static Player* GetInstance()
해설	static 변수 반환 함수

함수	void Update()
해설	업데이트 함수. 타이머 시간 마다 호출되며 중력, 플레이어 이동, 플레이어 생존 여부를 판단함. 플레이어가 사망한 상태일 경우 음악 일시 정지, 사망 애니메이션이 출력된 후 GameManager의 플레이어 사망 판정 변수를 변경함

함수	void Gravity()
해설	플레이어가 점프 중이거나 포커스 모드가 아닐 경우에 중력이 적용되는 함수

함수	bool CheckBtmGround(int & lengthDiff)
매개변수	int & lengthDiff : 위치 보정 값을 받기 위한 변수
해설	플레이어가 바닥에 닿았는지 유무 판별 함수. 바닥에 닿아있을 경우 false 반환, 아닐 경우 true 리턴. 바닥에 닿았거나 그 이상일 경우 겹치는 부분 만큼의 값을 lengthDiff에 저장하고 바닥 판정 내부 변수 true로 변경.

함수	bool CollisionMap(int direction, int & lengthDiff)
매개변수	int direction : 이동 방향 int & lengthDiff : 위치 보정 값
해설	플레이어와 맵의 충돌 판정 함수. 지나 갈 수 있는 경우 true, 아닐 경우 false 리턴. 이동방향에 따라 검사하며 갈 수 없는 타일과 닿아 있을 경우 보정 값 저장 후 false 반환. 가시(장애물)과 부딪혔을 경우 플레이어 사망 판정 내부 변수 변경 후 true 반환. 아이템과 닿았을 경우 아이템 획득 내부 변수 변경 후 false 반환. 충돌이 없을 경우에는 true 반환

함수	void CheckOut(POINT pos[], int direction)
매개변수	POINT pos[] : 이동 체크할 좌표 int direction : 이동 방향
해설	이동 방향에 따른 이동 실행 함수. 이동 방향에 따른 이동 가능을 체크 한 후, 해당 방향에 따른 보정 값을 계산하여 해당 위치로 이동 시킴

함수	void CheckOutMap(POINT pos[], int direction, int & lengthDiff)
매개변수	POINT pos[] : 이동 체크할 좌표 int direction : 이동 방향 int & lengthDiff : 위치 보정 값
해설	이동 방향에 따른 이동 체크 함수. 이동이 가능할 경우 true, 아니면 false 리턴. 해당 위치가 맵 내부에 있는지 판별 후 맵 밖에 있을 경우 그에 따른 보정 값을 저장함. 단, 맵 밖에 위치하더라도 열린 게이트 타일과 부딪혔을 경우엔 이동 가능

함수	bool CheckBlockMap()
해설	포커스 좌표로 이동 가능 판별 함수. 이동 가능 시 true, 아니면 false 리턴 포커스 좌표나 플레이어 좌표가 이동 불가능 타일에 있을 경우, 혹은 맵을 벗어날 경우에 직전 위치로 설정 후 false 반환. 해당하지 않을 경우 직전위치를 저장한 후 true 반환

함수	void DrawObject(HDC hdc)
매개변수	HDC hdc : DC 핸들
해설	히트박스 출력 함수. 디버그 모드가 설정되어 있을 경우에만 출력함

함수	void RenderObject(HWND hWnd, HDC hdc)
매개변수	HWND hWnd : 윈도우 핸들 HDC hdc : DC 핸들
해설	플레이어 이미지 출력 함수. 플레이어의 상태에 따라 이미지 출력 및 애니메이션 실행

함수	void CalcFocusMove()
해설	포커스 운동량 계산 함수. 이동 가능한 좌표일 경우, 포물선 운동 계산 및 포커스 운동량 감소. 이동이 불가능할 경우 직전 위치로 설정

함수	bool FocusMomentum()
해설	포커스 값에 따른 포물선 운동 계산 함수. 이동 가능 시 true, 아니면 false 반환

함수	void CanMovePlayer()
해설	플레이어의 키 입력에 따른 이동, 포커스 모드 실행 함수. 키 입력에 따라 플레이어 상태가 변경되며, 해당 키에 따른 작업이 실행됨 포커스 모드 실행 유무에 따라 수행할 수 있는 작업이 변경 되며 포커스 모드가 꺼져 있을 경우 이동, 점프, 포커스 모드 On / 켜져 있을 경우 영역 내부에서만 이 동이 가능함.

함수	void MovePlayer (POINT pos[], int direction, int num, float mulNum, float addNum)
매개변수	POINT pos[] : 이동시킬 좌표 int direction : 이동 방향 int num : 이동 시킬 값 float mulNum : 이동 시킬 방향 값 float addNum : 보정 값
해설	이동 방향에 따른 좌표 이동 함수. 이동 방향에 따라 좌표를 이동 값 * 이동 시킬 방향 값 + 보정 값으로 설정함

함수	int CheckFocusRange(int direction, int mulNum)
매개변수	int direction : 이동 방향 int mulNum : 보정 값
해설	포커스 영역 내부에 이동방향이 존재하는지 판단 함수. 영역을 벗어날 경우 보정 값을 계산해서 반환함

함수	bool CheckTileMap(TileMap tile)
매개변수	TileMap tile : 맵 타일
해설	맵 타일이 갈 수 있는 타일인지 판단하는 함수. 갈 수 있는 경우 false, 아니면 true 반환

함수	void SetPos(POINT pos[], int xPos, int yPos, int addNum)
매개변수	POINT pos[] : 이동시킬 좌표 int xPos : x값 int yPos : y값 int addNum : 보정 값
해설	좌표를 해당 x, y 값으로 이동 시키는 함수

함수	void CalcCenterPos()
해설	플레이어의 중심 값 계산

함수	void CalcFCenterPos()
해설	포커스 좌표의 중심 값 계산

함수	RECT ConversionRect(POINT pos[])
매개변수	POINT pos[] : 변환시킬 좌표
해설	POINT 좌표를 RECT 좌표로 변경해 반환하는 함수

함수	void ReturnLastPos()
해설	플레이어를 직전 위치로 설정하는 함수

함수	void Reset()
해설	플레이어가 사망했을 경우 초기화하는 함수

함수	RECT GetPlayerPos()
해설	플레이어 좌표를 반환하는 함수

함수	RECT GetFocusPos()
해설	포커스 좌표를 반환하는 함수



함수	bool GetIsFocusMode()
해설	포커스 모드 실행 유무 값 반환하는 함수

함수	bool GetIsEndAni()
해설	플레이어 사망 애니메이션이 끝났는지 값 반환 함수

함수	void SetIsPlayerDead(bool set)
매개변수	bool set : 플레이어 생존 여부 값
해설	플레이어의 생존 여부 설정 함수. 사망 시 포커스 영역이 삭제되며 중력을 받아서 땅에 떨어지고, 관련 사운드가 출력됨. 플레이어 상태 사망으로 설정됨

함수	bool GetIsPlayerDead()
해설	플레이어의 생존 여부 반환 함수

함수	void SetFocusLv()
해설	GameManager에 저장된 포커스 레벨 값을 내부변수에 저장하는 함수

함수	int GetFocusLv()
해설	포커스 레벨 값을 반환하는 함수

함수	void SetFocusGauge()
해설	포커스 레벨에 따른 최대 포커스 게이지 설정 함수

[ MapClass ]

함수	Map()
해설	기본 생성자

함수	~Map()
해설	기본 소멸자

함수	static Map* GetInstance()
해설	static 변수 반환 함수

함수	void Update()
해설	업데이트 함수. 타이머 시간마다 호출되며 플레이어가 생존해 있을 경우 아이템 / 가시 / 버튼 Off 등의 충돌 판정과 다음 스테이지 로드 등의 작업을 수행함

함수	bool CheckOffBtn()
해설	맵에 켜진 버튼이 남아있는지 판단 하는 함수. 켜진 버튼이 있으면 false, 아니면 true 반환

함수	void CheckShotOffBtn(const RECT & hitPos)
매개변수	RECT & hitPos : 충돌 위치
해설	미사일, 폭발 등이 버튼을 꺾는지 판단하는 함수. 버튼 위에 무언가가(스위치, 플레이어 등) 존재하지 않을 경우 버튼이 꺼짐

함수	void DrawObject(HDC hdc)
매개변수	HDC hdc : DC 핸들
해설	히트박스 출력 함수. 디버그 모드가 설정되어 있을 경우에만 출력함

함수	void RenderObject(HWND hWnd, HDC hdc)
매개변수	HWND hWnd : 윈도우 핸들 HDC hdc : DC 핸들
해설	맵 이미지 출력 함수. 타일의 속성에 따라 출력되는 이미지가 변경됨

함수	void SetNextStage()
해설	다음 스테이지를 로드하는 함수. 엔딩을 보고 오면 맨 처음 스테이지 로드 및 메인 UI 설정 게임 중 메인으로 돌아왔을 경우에 해당 스테이지 재 로드, UI 재 설정

함수	vector<TileMap> GetMapPos()
해설	맵 리턴 함수

함수	vector<ParceCannonStruct> CheckInCannon()
해설	맵에 존재하는 대포확인 및 설정 함수. 타입에 따라 일반 / 유도인 대포가 설정됨

함수	void Reset()
해설	플레이어 사망 시, 현재 스테이지 재 로드 하는 초기화 함수

함수	void ReadMapData()
해설	다음 스테이지를 읽어오는 함수. 맵 파일이 더 이상 존재하지 않을 경우 초기 값으로 설정하고 GameManager 의 현재 씬을 엔딩으로 변경함

함수	POINT GetResenSpot()
해설	플레이어 리젠 위치 반환 함수

함수	void SetIsNextStage(bool set)
매개변수	bool set : 다음 스테이지 로드 설정
해설	다음 스테이지 로드 유무 설정 함수

함수	bool GetIsNextStage()
해설	다음 스테이지 로드 유무 반환 함수

함수	bool PlayerGetItem()
해설	플레이어가 아이템을 획득했는지 판단 함수. 아이템에 닿았을 경우, 해당 사운드를 출력한 후 플레이어의 포커스 레벨을 상승 시킴. 획득한 아이템은 맵에서 삭제됨

[ ObstacleClass ]

함수	Obstacle()
해설	기본 생성자

함수	virtual ~Obstacle()
해설	기본 소멸자

함수	virtual void Update()
해설	업데이트 함수

함수	virtual void DrawObject(HDC hdc)
매개변수	HDC hdc : DC 핸들
해설	내용 존재 x

함수	virtual void RenderObject(HWND hWnd, HDC hdc)
매개변수	HWND hWnd : 윈도우 핸들 HDC hdc : DC 핸들
해설	내용 존재 x

함수	static void AddWeapon (vector<Obstacle *> & obstacle, vector<ParceCannonStruct> cannonSet)
매개변수	vector<Obstacle *> & obstacle : 대포를 저장할 벡터 vector<ParceCannonStruct> cannonSet : 대포 위치 및 속성 변수
해설	대포 추가하는 함수. 맵에 대포가 존재할 경우, 해당하는 대포를 생성함

함수	virtual void Reset()
해설	내용 존재 x

함수	static void DeleteAllData(vector<Obstacle *> & vec)
매개변수	vector<Obstacle *> & vec : 저장 값 삭제할 변수
해설	벡터에 저장된 값을 모두 동적할당 해제 및 삭제하는 함수

[ CannonClass ]

함수	Cannon()
해설	기본 생성자

함수	Cannon(ParceCannonStruct set)
매개변수	ParceCannonStruct set : 대포 위치 및 속성 변수
해설	대포 속성에 따른 생성자

함수	~Cannon()
해설	기본 소멸자

함수	void Update()
해설	업데이트 함수. 플레이어가 생존해 있으면 탐지범위 내에 있는지 판단 함수 실행, 아닐 경우 초기화

함수	void DrawObject(HDC hdc)
매개변수	HDC hdc : DC 핸들
해설	히트박스 출력 함수. 디버그 모드가 설정되어 있을 경우에만 출력함

함수	void RenderObject(HWND hWnd, HDC hdc)
매개변수	HWND hWnd : 윈도우 핸들 HDC hdc : DC 핸들
해설	내용 존재 x

함수	void CheckInPlayer()
해설	플레이어가 대포의 탐지 범위 내에 있는지 판단하는 함수. 탐지 범위 내에 탐지 시간 만큼 존재 할 경우 테스트 유도탄을 발사함. 테스트 유도탄이 플레이어에 부딪힐 경우 실제 미사일 발사, 아닐 경우 초기화. 탐지 시간 및 발사 관련 타이머가 있어서 바로 발사 되는 것 방지함

함수	void MoveTestShot()
해설	테스트 유도탄 이동 함수

함수	void CheckHit()
해설	테스트 유도탄이 충돌한 위치 판별 함수 맵에 부딪힐 경우 탐지 타이머 off, 플레이어에 부딪힐 경우 탐지 타이머 off, 발사 타이머 on

함수	bool CheckTileMap(TileMap mapTile)
매개변수	TileMap(mapTile)
해설	맵 충돌 판별 함수. 충돌 시 true 반환, 아니면 false 반환

함수	RECT GetHitRect()
해설	히트박스 반환

함수	void Reset()
해설	초기화 함수. 테스트 탄 초기 위치 설정, 타이머 초기 설정 적용



함수	int CalcSetValue(int set)
매개변수	int set : 초기화 할 값
해설	테스트 탄 속도, 히트 박스 범위, 탐지 타이머 등을 스테이지 레벨에 따라 설정하는 함수. 스테이지가 높아질수록 난이도가 올라감

[ BulletClass ]

함수	Bullet()
해설	기본 생성자

함수	~Bullet()
해설	기본 소멸자

함수	static Bullet* GetInstance()
해설	static 변수 반환 함수

함수	void Update()
해설	업데이트 함수. 플레이어가 죽지 않았을 때, 발사된 미사일이 있으면 이동 및 피격 판정

함수	void DrawObject(HDC hdc)
매개변수	HDC hdc : DC 핸들
해설	히트박스 출력 함수. 디버그 모드가 설정되어 있을 경우에만 출력함

함수	void RenderObject(HWND hWnd, HDC hdc)
매개변수	HWND hWnd : 윈도우 핸들 HDC hdc : DC 핸들
해설	이미지 출력 함수. 미사일의 속성에 따라 이미지가 변경됨

함수	void Shoot (const RECT & hitRect, const POINT & cannonCenter, const POINT & playerCenter, int bulletType)
매개변수	const RECT & hitRect : 대포 히트박스 const POINT & cannonCenter : 대포 중심점 const POINT & playerCenter : 플레이어 중심점 int bulletType : 미사일 속성
해설	미사일 발사 함수. 플레이어가 히트박스 안에 있을 때, 속성에 맞는 미사일 중 발사되지 않은 상태 의 미사일이 발사됨. 발사 시 해당하는 사운드 출력됨

함수	void Shoot (const RECT & hitRect, const POINT & cannonCenter, const POINT & playerCenter, int bulletType)
매개변수	const RECT & hitRect : 대포 히트박스 const POINT & cannonCenter : 대포 중심점 const POINT & playerCenter : 플레이어 중심점 int bulletType : 미사일 속성
해설	미사일 발사 함수. 플레이어가 히트박스 안에 있을 때, 속성에 맞는 미사일 중 발사되지 않은 상태 의 미사일이 발사됨. 발사 시 해당하는 사운드 출력됨

함수	void CalcBullet (BulletStruct & bullet, const POINT & cannonCenter, const POINT & playerCenter, int bulletType)
매개변수	BulletStruct & bullet : 미사일 const POINT & cannonCenter : 대포 중심점 const POINT & playerCenter : 플레이어 중심점 int bulletType : 미사일 속성
해설	발사된 미사일의 이동 값 계산 함수. 일반 / 유도 속성에 따라 계산이 변경됨

함수	void CheckShot()
해설	발사된 미사일 탐지 함수. 발사된 미사일이 있으면 미사일의 이동 값을 계산함

함수	void MoveShot(BulletStruct & bullet)
매개변수	BulletStruct & bullet : 미사일
해설	미사일 이동 함수 미사일이 포커스 내부일 경우 이동 값이 0.25로 감소, 아닐 경우 이동 값 만큼 이동시킴

함수	void CheckHit(BulletStruct & bullet)
매개변수	BulletStruct & bullet : 미사일
해설	미사일 충돌 판정 함수 미사일이 맵에 충돌 시 폭발 생성, 맵 버튼 Off 함수 실행 미사일이 플레이어에 충돌 시 폭발 생성, 플레이어 사망 판정 충돌 된 경우나 맵 밖을 벗어난 경우에 미사일 초기화

함수	bool CheckTileMap(BulletStruct & bullet)
매개변수	BulletStruct & bullet : 미사일
해설	미사일 맵 충돌 판정 함수. 미사일이 맵에 충돌 했을 경우 true, 아니면 false 반환

함수	void Reset
해설	미사일 초기화 함수

함수	void ResetBullet(BulletStruct & bullet)
매개변수	BulletStruct & bullet : 미사일
해설	미사일 초기 세팅 및 Off 하는 함수

함수	RECT ConversionRect(POINT pos)
매개변수	POINT pos : 변환할 좌표 값
해설	POINT 좌표를 RECT로 변환해 주는 함수

함수	int CalcSpeed()
해설	미사일 속도 계산 함수. 스테이지에 따라 미사일 속도가 증가함

[ ExplodeClass ]

함수	Explode()
해설	기본 생성자

함수	~Explode()
해설	기본 소멸자

함수	static Explode * GetInstance()
해설	static 변수 반환 함수

함수	void Update()
해설	업데이트 함수 플레이어가 생존해 있을 경우, 실행되고 있는 폭발에 대해 플레이어 피격 판정

함수	void CheckHitPlayer(ExplodeStruct & effect)
매개변수	ExplodeStruct & effect : 폭발 이펙트
해설	폭발 피격 판정 함수. 폭발이 플레이어에 닿았을 경우 사망 판정, 맵에 닿았을 경우 맵 버튼 Off 함수 실행

함수	void DrawObject(HDC hdc)
매개변수	HDC hdc : DC 핸들
해설	히트박스 출력 함수. 디버그 모드가 설정되어 있을 경우에만 출력함

함수	void RenderObject(HWND hWnd, HDC hdc)
매개변수	HWND hWnd : 윈도우 핸들 HDC hdc : DC 핸들
해설	이미지 출력 함수. 폭발 이펙트 및 애니메이션 출력

함수	void SetNextFrame()
해설	폭발 애니메이션 프레임 설정 함수. 포커스 모드 안에 있으면 다음 프레임으로 넘어가지 않음, 아닐 경우 다음 프레임으로 증가. 마지막 프레임에 도달하면 폭발 Off 및 초기화

함수	void StartExplode(POINT bulletPos)
매개변수	POINT bulletPos : 미사일 위치
해설	폭발 시작 함수. 총알의 위치에 따라 Off 상태의 폭발이 On으로 상태변화 하면서 폭발이 실행됨 폭발이 시작되면 관련된 사운드가 출력됨.

함수	void Reset()
해설	초기화 함수

함수	void ResetExplode(ExplodeStruct & effect)
매개변수	ExplodeStruct & effect : 폭발 이펙트
해설	폭발 이펙트 초기 값 설정 및 Off 적용 함수

## [ UIClass ]

함수	UI()
해설	기본 생성자

함수	~UI()
해설	기본 소멸자

함수	static UI * GetInstance()
해설	static 변수 반환 함수

함수	void Update()
해설	업데이트 함수

함수	void DrawObject(HDC hdc)
매개변수	HDC hdc : DC 핸들
해설	내용 존재 x

함수	void RenderObject(HWND hWnd, HDC hdc)
매개변수	HWND hWnd : 윈도우 핸들 HDC hdc : DC 핸들
해설	이미지 출력 함수. 현재 씬과 플레이어 상태에 따른 UI 및 애니메이션 출력 메인 화면 : 키 입력 애니메이션 게임 화면 : 스테이지 재 로드 애니메이션, 첫 스테이지 알림, 첫 아이템 획득 알림 엔딩 화면 : 엔딩 애니메이션



함수	void CheckPushBtn(POINT pos)
매개변수	POINT pos : 마우스 위치
해설	버튼 입력 확인 함수. 게임화면 중 일시정지일 때 나타나는 HOME, PLAY 버튼 입력 확인. HOME 버튼 입력 시 메인으로 돌아가고 PLAY를 누르면 게임화면으로 돌아감.

함수	void Reset()
해설	내용 존재 x

함수	void SetIsGoMain(bool set)
매개변수	bool set : 메인으로 돌아갔는지 확인 값
해설	내부 변수에 게임화면에서 메인으로 돌아갔는지 여부 값 저장

함수	bool GetIsGoMain()
해설	메인으로 돌아갔는지 여부 값 반환

## [ SoundClass ]

함수	SoundSystem()
해설	기본 생성자

함수	~SoundSystem()
해설	기본 소멸자

함수	static SoundSystem * GetInstance()
해설	static 변수 반환 함수

함수	void Update()
해설	현재 씬이 게임화면일 경우, 해당하는 BGM 실행

함수	void PlayChangeStageSound()
해설	화면전환 소리 출력 함수

함수	void PlayDeadSound()
해설	플레이어 사망 소리 출력 함수

함수	void PlayShotSound()
해설	미사일 발사 소리 출력 함수

함수	void PlayExplodeSound()
해설	폭발 소리 출력 함수

함수	void PlayFocusSound()
해설	포커스 모드 On 소리 출력 함수

함수	void PlayGateBreakSound()
해설	게이트 파괴 소리 출력 함수

함수	void PlayBtnOffSound()
해설	버튼이 꺼지는 소리 출력 함수

함수	void PlayGetItemSound()
해설	아이템 획득 소리 출력 함수

함수	void PlayResultBgm()
해설	엔딩 화면 BGM 출력 함수

함수	void SetIsPause(bool set)
매개변수	bool set : 일시정지 유무 변수
해설	BGM play & pause 함수. 플레이어가 사망 시 BGM의 맨 처음부터 실행됨

함수	void SetIsStop(bool set)
매개변수	bool set : 엔딩 화면 확인 변수
해설	엔딩 화면 BGM 출력 및 메인 화면 BGM 설정 함수

함수	void SetFirstPos()
해설	BGM 시작 위치를 맨 처음 부분으로 설정함

함수	void DrawObject(HDC hdc)
매개변수	HDC hdc : DC 핸들
해설	내용 존재 x

함수	void RenderObject(HWND hWnd, HDC hdc)
매개변수	HWND hWnd : 윈도우 핸들 HDC hdc : DC 핸들
해설	내용 존재 x

함수	void Reset()
해설	내용 존재 x

## 2) MapTool

[ MapClass ]

함수	Map()
해설	기본 생성자

함수	!Map()
해설	기본 소멸자

함수	static Map * GetInstance()
해설	static 변수 반환 함수

함수	void DrawMap(HDC hdc)
매개변수	HDC hdc : DC 핸들
해설	버튼 및 중심 표시선 출력 함수

함수	void RenderMap(HWND hWnd, HDC hdc)
매개변수	HWND hWnd : 윈도우 핸들 HDC hdc : DC 핸들
해설	속성에 따른 맵 타일 이미지 출력

함수	void DrawBtn(HDC hdc, RECT rect)
매개변수	HDC hdc : DC 핸들 RECT rect : 버튼 그릴 변수
해설	실제 버튼 변수 보다 10 만큼 큰 버튼 화면 출력

함수	void AddTile(POINT pos)
매개변수	POINT pos : 마우스 좌표
해설	타일 선택 및 맵에 타일 배치하는 함수. 클릭 입력 플레이어가 기존에 배치되어 있으면, 배치된 플레이어 삭제됨. 타일 속성이 정해져 있고, 올바른 범위 내에 있을 경우 내부변수 맵에 타일이 추가됨

함수	void AddTile(POINT startPos, POINT lastPos)
매개변수	POINT startPos : 마우스 시작 좌표(눌린 위치) POINT lastPos : 마우스 끝 좌표(떼어진 위치)
해설	타일 선택 및 맵에 타일 배치하는 함수. 드래그 입력 플레이어가 기존에 배치되어 있으면, 배치된 플레이어 삭제됨. 타일 속성이 정해져 있고, 올바른 범위 내에 있을 경우 좌표 값 만큼 내부변수 맵에 타일이 추가됨

함수	void PushBack(RECT addRect)
매개변수	RECT addRect : 추가할 타일
해설	맵에 타일 추가하는 함수. 내부 변수 맵에 타일(속성, 좌표 값) 저장

함수	void DeleteTile(POINT pos)
매개변수	POINT pos : 마우스 좌표
해설	타일 삭제 함수. 클릭 입력. 마우스로 클릭한 위치의 타일을 내부변수 맵에서 삭제

함수	void DeleteTile(POINT startPos, POINT lastPos)
매개변수	POINT startPos : 마우스 시작 좌표(눌린 위치) POINT lastPos : 마우스 끝 좌표(떼어진 위치)
해설	타일 삭제 함수. 드래그 입력. 마우스로 클릭한 위치의 좌표 값 만큼 타일을 내부변수 맵에서 삭제

함수	RECT CalcBtn(RECT rect)
매개변수	RECT rect : 버튼 좌표
해설	버튼 좌표 계산 함수. 실제 크기보다 +10 만큼 큰 좌표 반환

함수	RECT CalcTileRange(POINT pos)
매개변수	POINT pos : 마우스 좌표
해설	마우스 좌표를 맵 좌표로 변환하는 함수, 맵 사이즈(800 x 592)를 타일 크기인 16으로 나누어서 마우스 좌표를 맵 좌표로 변환 후 반환

함수	void CalcMulNum(int moveX, int moveY, int &mulX, int &mulY)
매개변수	int moveX : 움직인 좌표의 가로 값 int moveY : 움직인 좌표의 세로 값 int &mulX : x 보정 값 int &mulY : y 보정 값
해설	좌표 값에 따른 보정 값 반환 하는 함수. 좌 - 우 이동 시 : mulX = -1 우 - 좌 이동 시 : mulX = 1 기타 : mulX = 0 상 - 하 이동 시 : mulY = -1 하 - 상 이동 시 : mulY = 1 기타 : mulY = 0 값이 저장됨

함수	int CheckIsSameRect(RECT rect)
매개변수	RECT rect : 삭제할 타일
해설	삭제할 타일의 인덱스를 반환하는 함수. 해당 타일이 위치한 맵 좌표의 인덱스를 찾아 값이 존재하면 인덱스 값, 존재하지 않을 경우 -1 반환

함수	void CheckIsDoublePos(RECT rect)
매개변수	RECT rect : 검사할 타일
해설	타일이 동일 좌표에 설정 되어 있는지 판별 후 삭제 하는 함수

함수	void SetNowType(HWND hWnd, POINT pos)
매개변수	HWND hWnd : 윈도우 핸들 POINT pos : 마우스 좌표
해설	좌표 값에 따른 맵 타입 설정 및 버튼 동작 함수. 좌표 값이 맵 타일 좌표일 경우 해당 타입을 내부 변수에 설정, 파일 열기 및 파일 저장 버튼, 초기화일 경우 해당 함수 실행

함수	POINT SetShowType(int type)
매개변수	int type : 맵 타입
해설	타입에 따라 현재 설정된 타입을 화면에 출력 및 해당 타입 이미지의 좌표 값을 반환하는 함수

함수	void FileOpen(HWND hWnd)
매개변수	HWND hWnd : 윈도우 핸들
해설	파일 열기 대화상자 출력 함수. 파일 열기를 선택하면 파일 데이터를 읽어오는 함수 실행



함수	void ReadData(HWND hWnd, TCHAR *OpenFileName)
매개변수	HWND hWnd : 윈도우 핸들 TCHAR * OpenFileName : 파일 이름
해설	<p>맵 데이터 읽어오는 함수.</p> <p>일치하는 확장자가 아닐 경우 파일 읽기 실패 메시지 박스 생성.</p> <p>올바른 확장자일 경우 플레이어 리젠 위치를 읽은 후 / keycode - keycode 공식을 통해 값을 해독해서 내부변수 맵에 저장. 타일 좌표를 읽은 후 / keycode - keycode 공식을 통해 값을 해독해서 임시 변수에 저장한 후 타입과 타입에 따른 이미지 좌표 값을 임시 변수에 저장 한 후 내부변수 맵에 해당 변수를 저장함</p>

함수	void FileSave(HWND hWnd)
매개변수	HWND hWnd : 윈도우 핸들
해설	<p>다른 이름으로 저장 대화상자 출력 함수.</p> <p>파일 저장을 선택하면 파일 데이터를 저장하는 함수 실행</p>

함수	void ReadData(HWND hWnd, TCHAR *OpenFileName)
매개변수	HWND hWnd : 윈도우 핸들 TCHAR * OpenFileName : 파일 이름
해설	<p>맵 데이터 읽어오는 함수.</p> <p>일치하는 확장자가 아닐 경우 파일 저장 실패 메시지 박스 생성.</p> <p>올바른 확장자일 경우 플레이어 리젠 위치의 중심값을 구한 후 (+ keycode) * keycode 공식을 통해 값을 변형해서 파일에 저장. 타일 좌표를 읽은 후 (+ keycode) * keycode 공식을 통해 값을 변형해서 파일에 저장함</p>

함수	void Reset()
해설	내부 변수 맵의 모든 데이터 삭제 함수

## 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.