

# DSD

Team Prepper

## 요약문

본 문서는 본 팀에서 개발하고자 하는 게임의 DSD 문서이다. 본 문서에서는 본 프로젝트 게임 개발을 위해 앞으로 개발할 게임 프로그램의 표준이 되어 게임 프로그램을 각 블록 레벨로 기술한다. 설계에 들어가기 전 각 모듈에 대한 내용을 기술하며 게임 외부 인터페이스에 대한 내용 또한 기술하며 상세한 코드는 기술하지 않는다.

전체 과제의 기능을 소개하며 각 블록으로 분해한 다이어그램을 이용하여 게임 시스템을 설명한다. 소프트웨어와 소프트웨어 각 블록 간 인터페이스를 기술하고 각 소프트웨어 블록에 따라 기능을 기술하고 모듈의 입출력 파라미터 정의 및 인터페이스, 모듈에서 호출하는 함수 또는 시스템 호출을 클래스 다이어그램을 이용하여 기술한다.

본 프로젝트의 게임은 플레이어 로컬과 서버로 이루어져 있다. 게임 프로그램의 경우 언리얼 엔진 5를 이용하여 개발하였고, 게임 서버의 경우 언리얼 엔진에서 제공하는 멀티 플레이 서버를 이용하였다. 본 문서는 게임의 기능적인 시스템 설계를 중심으로 클래스 다이어그램의 함수를 구성하였으며 입출력 및 알고리즘이 Description을 통해 서술되어 있다. 본 문서의 클래스 다이어그램을 통해 시스템 구현에 있어 시스템 구조를 쉽게 파악할 수 있도록 한다.

## 목차

1. 시스템 개요 .....	01
1.1 프로젝트의 개요 .....	01
1.2 블록 다이어그램 .....	02
1.3 UI .....	03
2. 세부 설계 .....	11
2.1 Character .....	11
2.1.1 Character의 기능 .....	11
2.1.2 Character의 클래스 다이어그램 .....	11
2.2 Item .....	17
2.2.1 Item의 기능 .....	17
2.2.2 Item의 클래스 다이어그램 .....	17
2.3 Interface .....	22
2.3.1 Interface의 기능 .....	22
2.3.2 Interface의 클래스 다이어그램 .....	22
2.4 Weapon .....	25
2.4.1 Weapon의 기능 .....	25
2.4.2 Weapon의 클래스 다이어그램 .....	25
2.5 Car .....	30
2.5.1 Car의 기능 .....	30
2.5.2 Car의 클래스 다이어그램 .....	30
2.6 Component .....	33
2.6.1 Component의 기능 .....	33
2.6.1 Component의 클래스 다이어그램 .....	33
2.7 GameMode .....	35
2.7.1 GameMode의 기능 .....	35
2.7.2 GameMode의 클래스 다이어그램 .....	35
3. 수행 방법 .....	37
3.1 설계 및 개발 환경 .....	37
3.2 설계 및 개발 도구 .....	37
3.3 테스트 방안 .....	37
4. 참조 자료 .....	38

## 그림 목차

그림 1 프로그램 개념도 .....	1
그림 2 전체 블록 다이어그램 .....	2
그림 3 로그인 화면 UI 설계 구조 .....	3
그림 4 게임 로비 화면 UI 설계 구조 .....	4
그림 5 멀티 플레이 룸 UI 설계 구조 .....	5
그림 6 게임 필드 UI 설계 구조 .....	6
그림 7 인벤토리 UI 설계 구조 .....	7
그림 8 아이템 세부 정보 팝업 UI 설계 구조 .....	8
그림 9 아이템 조합 UI 설계 구조 .....	9
그림 10 알람 메시지 UI 설계 구조 .....	10
그림 11 Character의 클래스 다이어그램 .....	11
그림 12 Item의 클래스 다이어그램 .....	17
그림 13 Interface의 클래스 다이어그램 .....	22
그림 14 Weapon의 클래스 다이어그램 .....	25
그림 15 Car의 클래스 다이어그램 .....	30
그림 16 Component의 클래스 다이어그램 .....	33
그림 17 GameMode의 클래스 다이어그램 .....	35

# 개발 요약

## ● 목적

본 문서는 게임 프로그램 시스템의 요구사항을 기록한 DSD 문서로 앞으로 개발할 게임 프로그램 시스템의 표준이 된다.

## ● 관련 기술

### ◇ 언리얼 엔진 5.3.2

#### 렌더링

- 나나이트 명시적 렌더링
- 루멘
- 패스 트레이서
- 버추얼 새도 맵
- 템포럴 슈퍼 해상도
- 랜더 리소스 뷰어

#### 월드 빌딩

- 레벨 인스턴스 액터 필터
- 나나이트 랜드스케이프
- 랜드스케이프 피지컬 머티리얼
- 랜드스케이프 LOD 그룹

#### 프로시저럴 콘텐츠 생성

- 계층형 생성
- 서브그래프 조사
- 그래프 인스턴스 파라미터
- 커스텀 PCG 엘리먼트

#### 개발자

- 버추얼 에셋
- C++20 디폴트 버전
- C++ 언리얼 헤더 툴

## ● 개발 환경

### ◇ Window 11

MacOS

### ◇ 언리얼 엔진 5.3.2

Visual Studio Code 1.88

Rider 2024.1

### ◇ Adobe

Figma

## ● 제품 요약

생존 탈출형 오픈월드 TPS 게임으로, 플레이어는 솔로 플레이, 멀티 플레이, 데스매치 3가지 모드로 게임을 플레이할 수 있다.

게임을 플레이하는 플레이어는 게임 필드를 돌아다니며 필드에 흩어진 물자들을 수집하고 조합하여 생존 및 탈출에 필요한 물자를 획득하고, 무기를 이용하여 적들과 싸워 생존 및 탈출을 통한 게임의 엔딩을 목표로 게임을 플레이한다.

## ● 개발 비용

### ◇ 환경구축

Software Part

Design Part

### ◇ 임금

10,000원/시간

## ● 목표 시장

### ◇ 주 시장

온라인 게임 유통 플랫폼 Steam

## 1. 시스템 개요

### 1.1 프로젝트의 개요

본 프로젝트는 언리얼 엔진 5를 이용하여 TPS 게임 소프트웨어를 개발하고자 한다. 게임을 이용하는 플레이어는 필드를 돌아다니며 물자들을 수집하고 조합하여 필요한 물자를 획득하고, 무기를 이용하여 적들과 싸워 필드를 탈출하는 게임 프로그램 서비스를 제공하는 것이 목표이다. 프로젝트의 개념도는 아래 그림 1과 같다.

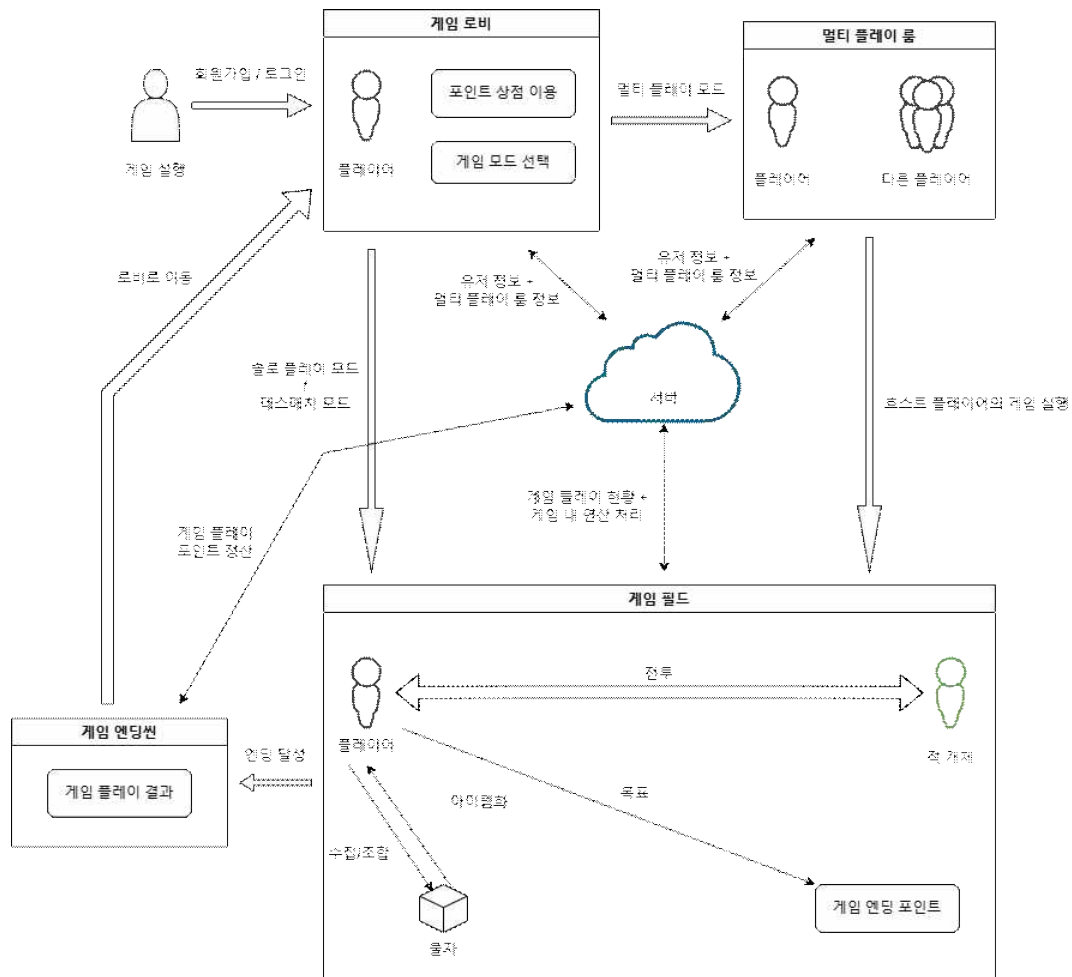


그림 1 프로그램 개념도

게임을 플레이하는 플레이어는 처음 회원가입 및 로그인을 통해 게임 로비로 들어와 게임에 접속할 수 있다. 서버는 로그인한 플레이어의 유저 정보와 개설된 멀티 플레이 룸에 대한 정보를 실시간으로 주고받아 플레이어가 다른 플레이어가 개설한 멀티 플레이 룸 정보를 받거나 본인이 개설한 멀티 플레이 룸 정보를 다른 플레이어가 확인할 수 있도록 한다.

플레이어는 게임 로비에서 솔로 플레이 및 대사매치 모드를 선택하여 바로 게임 필드로 이동할 수 있고, 멀티 플레이 모드를 선택하여 멀티 플레이 룸에 진입해 같은 멀티 플레이 룸에 있는 다른 플레이어들과 게임을 진행할 수 있다. 멀티 플레이 룸에서도 실시간으로 서버와 유저 정보 및 멀티 플레이 룸 정보를 주고받아 다른 플레이어의 정

보와 호스트 플레이어에 의한 게임 시작 정보를 주고받을 수 있다.

게임 필드에서는 필드 내 물자를 수집하고 조합하는 것으로 아이템을 획득하여 인벤토리에 추가하여 사용할 수 있으며, 적 개체와 전투를 벌일 수 있다. 모든 플레이어는 게임 엔딩 포인트를 목적으로 게임의 엔딩을 향해 플레이 하며, 해당 엔딩을 달성하는 것으로 게임 엔딩씬으로 넘어가 게임을 마무리할 수 있다. 게임 필드 역시 서버와 실시간으로 통신하며 게임 내 플레이 현황과 게임 내 발생하는 연산 처리 결과들을 주고받아 서버 중심으로 플레이 현황과 연산들이 처리되어 플레이어 로컬에서는 실시간으로 동기화된 게임 화면을 제공할 수 있도록 한다.

게임 엔딩씬에서는 플레이어마다 게임 플레이 결과를 통해 플레이 결과를 포인트로 전환하여 플레이어에게 제공하는 것으로 플레이어가 획득한 포인트를 이용하여 다음 게임을 유리하게 풀어나갈 수 있는 자원들을 얻을 수 있는 요소 등으로 활용할 수 있도록 한다.

## 1.2 블록 다이어그램

개발에 있어 전체 시스템의 구조를 도식화한 블록 다이어그램은 그림 2와 같다. 본 시스템은 크게 플레이어 로컬과 서버로 구분된다.

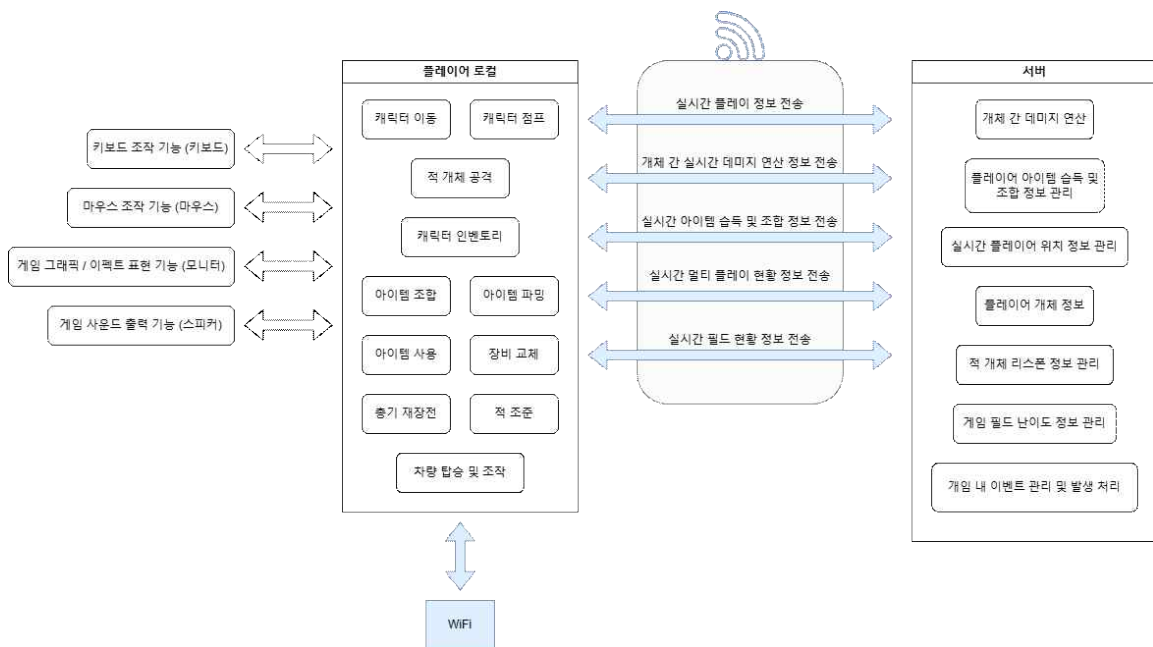


그림 2 전체 블록 다이어그램

플레이어 로컬에서는 실제 플레이어 조작이 구동되는 부분이며 키보드, 마우스 입력을 받아 플레이어 캐릭터 조작이 이루어진다. 게임 상황에 대한 출력은 모니터, 스피커를 통해 그래픽과 사운드가 출력되고 플레이어는 이를 통해 자신의 조작에 대한 캐릭터 움직임과 게임 상황에 대한 정보를 전달받을 수 있다. 플레이어 로컬에서 일어나는 모든 일들은 WiFi를 통해 서버와 연결되어 실시간으로 서버에서 각종 데미지 연산이나 플레이어 정보, 게임 상황에 대한 정보를 주고받으며 플레이어 로컬에서 인식한 플레이어 조작을 서버와 통신하여 서버에서 모든 정보를 처리한 뒤 다시 플레이어 로컬로 처리 결과를 전달해주는 것으로 게임의 멀티 플레이 시스템에 대한 실시간 동기화 및 게임 내의 보안성을 유지할 수 있다.

### 1.3 UI

이 부분에서는 예상 UI와 UI 안의 각 구성요소를 설명한다. 실제 개발에 따라 UI 디자인은 달라질 수 있으며, 본 문서에서 제공되는 UI는 원활한 설명을 위해 UI 프레임을 기준으로 구조를 설명하며, 실제 구현되는 디자인과는 차이가 있을 수 있으나 해당 UI 설계 구조를 따른다.

아래 그림 3은 게임 처음 실행 시 표기되는 로그인 화면 UI 설계 구조이다. 아이디 입력 칸과 비밀번호 입력 칸, 회원가입과 비밀번호 찾기 등 로그인과 회원가입에 필요한 간단한 요소들로 구성되어 있으며 해당 화면에서 로그인을 통해 게임 시스템에 접속하여 게임 로비 화면으로 이동하게 된다.



그림 3 로그인 화면 UI 설계 구조



아래 그림 4는 게임 로그인 후 표시되는 게임 로비 화면 UI 설계 구조이다. 게임 메뉴 및 게임 중 확인할 수 있는 플레이어 캐릭터 상세 정보 등 세부 정보를 확인하기 위한 UI를 통합적으로 설계한 그림이다. 상단 바를 통해 메뉴 카테고리를 변경할 수 있으며, 현재 게임 모드 및 플레이어 수 확인, 통합 인벤토리 및 현재 무기 등 게임을 플레이하면서 세부적인 정보를 표시하기 위한 프레임이다.

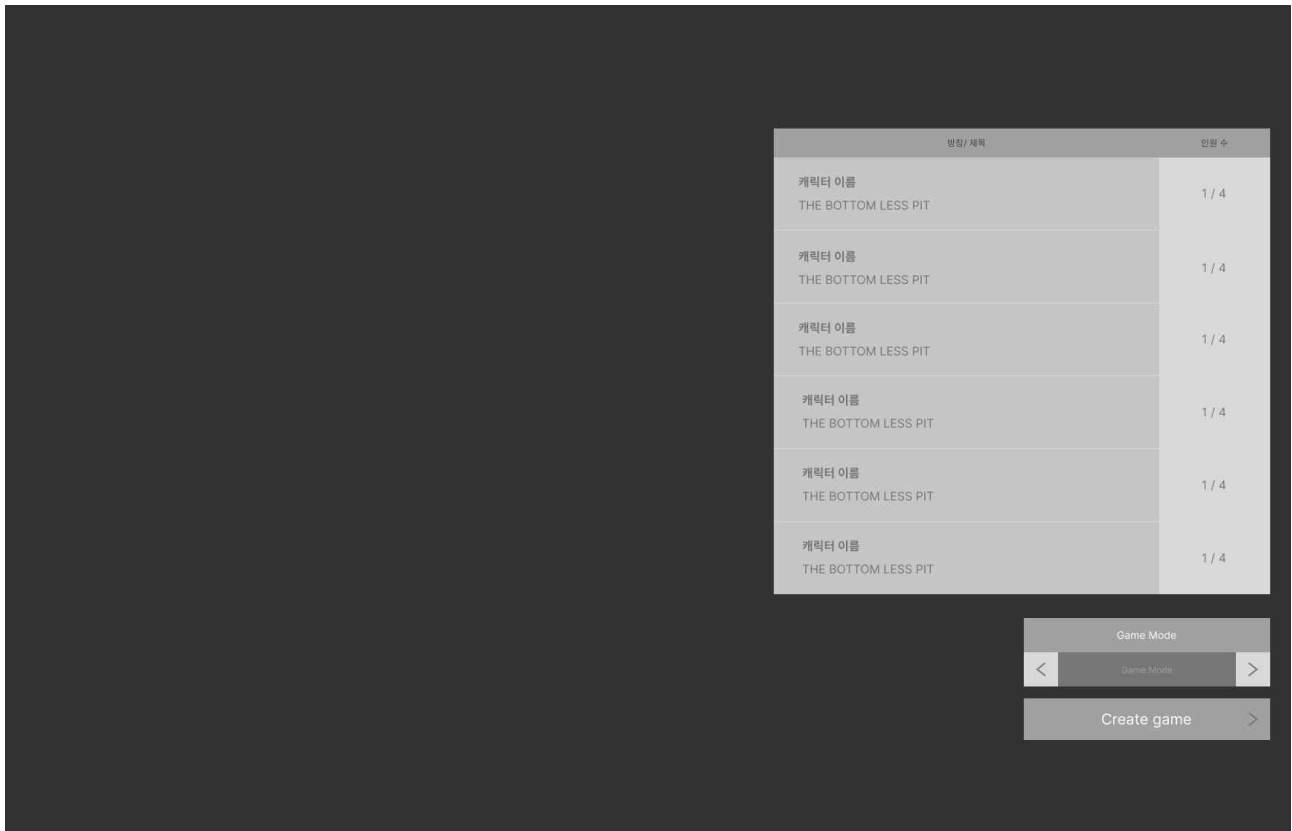


그림 4 게임 로비 화면 UI 설계 구조

아래 그림 5는 멀티 플레이를 위한 멀티 플레이 룸의 UI 설계 구조로, 호스트 플레이어가 상단에 표시되고 아래 호스트 플레이어 포함 최대 4인까지 참여할 수 있는 구조를 나타낸 것이다. 위 그림 6의 통합 정보 구조에서 우측 하단 게임 로비에 필요한 모드 선택과 멀티 룸 탐색 비가 같이 표시되어 있는데, 이 UI를 통해 멀티 플레이 룸을 구성 또는 참여하게 되면 그림 7과 같은 멀티 플레이 룸 비가 표시되어 멀티 플레이를 진행할 수 있다.

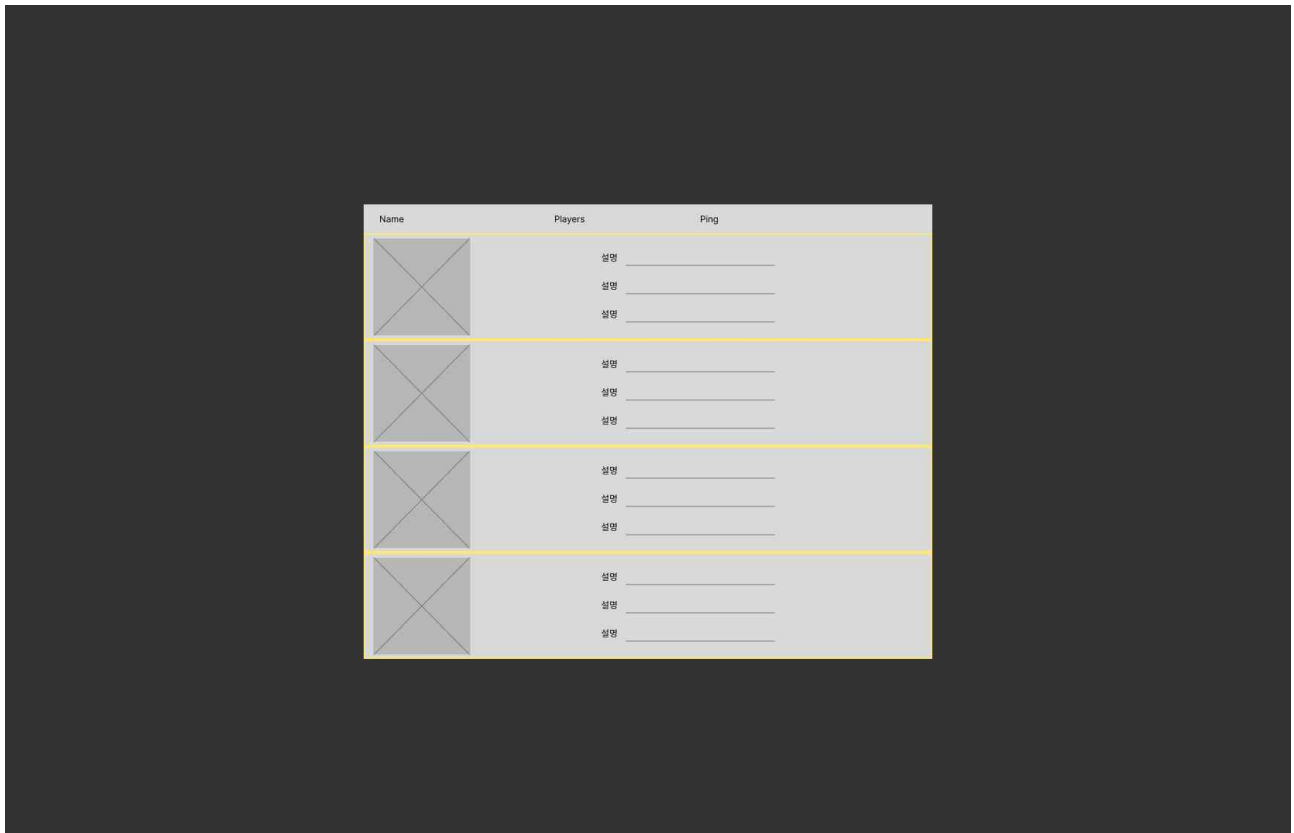


그림 5 멀티 플레이 룸 UI 설계 구조

아래 그림 6은 게임 필드 화면 UI 설계 구조이다. 체력과 포만감, 수분과 같은 플레이어 관련 수치들을 UI 바를 통해 시각적으로 표기하고 하단 중앙에 보유한 장비류들, 우측 하단에 콕슬롯 아이템, 잔탄량 등 게임 플레이에 필요한 정보들을 시각화하였다.

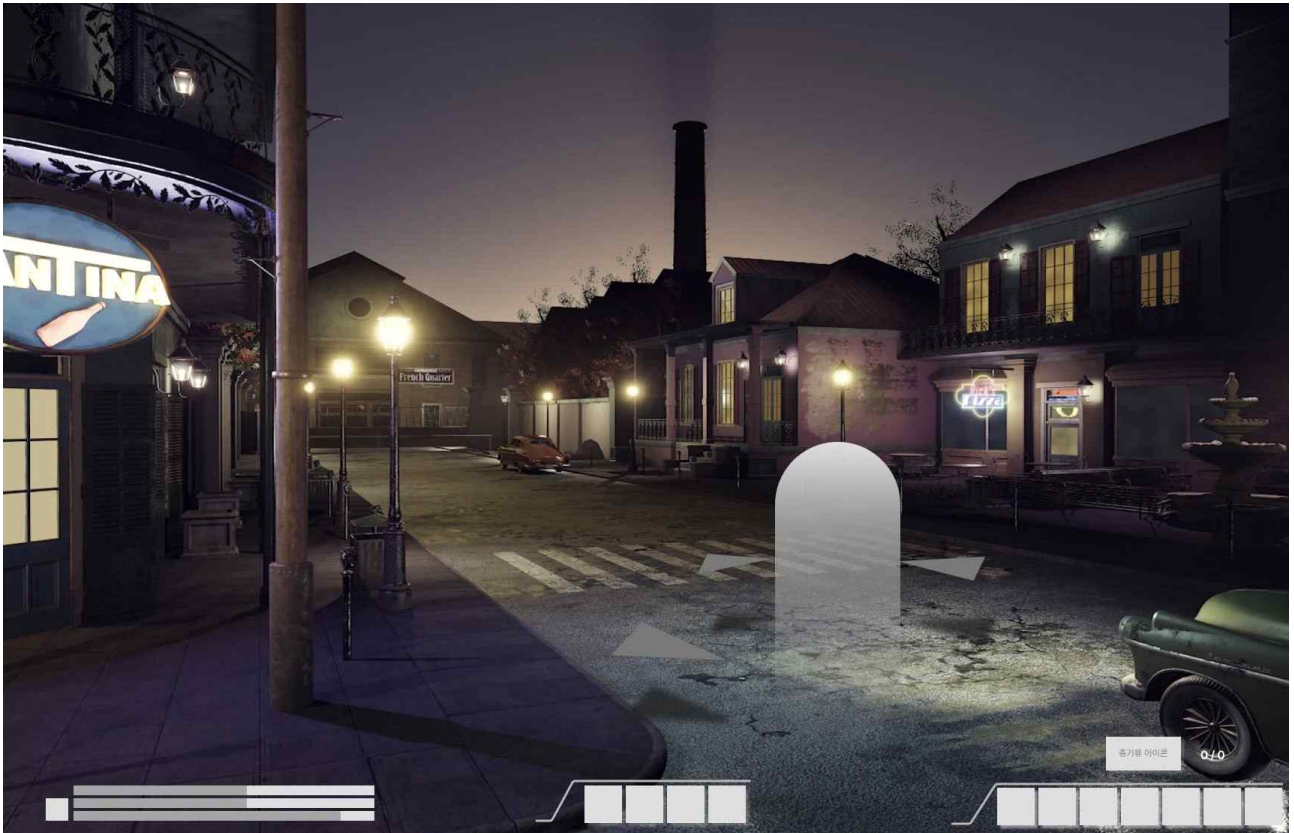


그림 6 게임 필드 UI 설계 구조

아래 그림 7은 게임의 인벤토리 UI 설계 구조이다. 화면에서 각 카테고리 분류된 아이템들의 카테고리를 선택하면 해당 카테고리의 보유한 아이템들이 하단 슬롯에 표시된다. 아이템 조합 기능을 위한 버튼과 아이템 폐기를 위한 버튼이 인벤토리 UI 하단에 배치되어 있다.

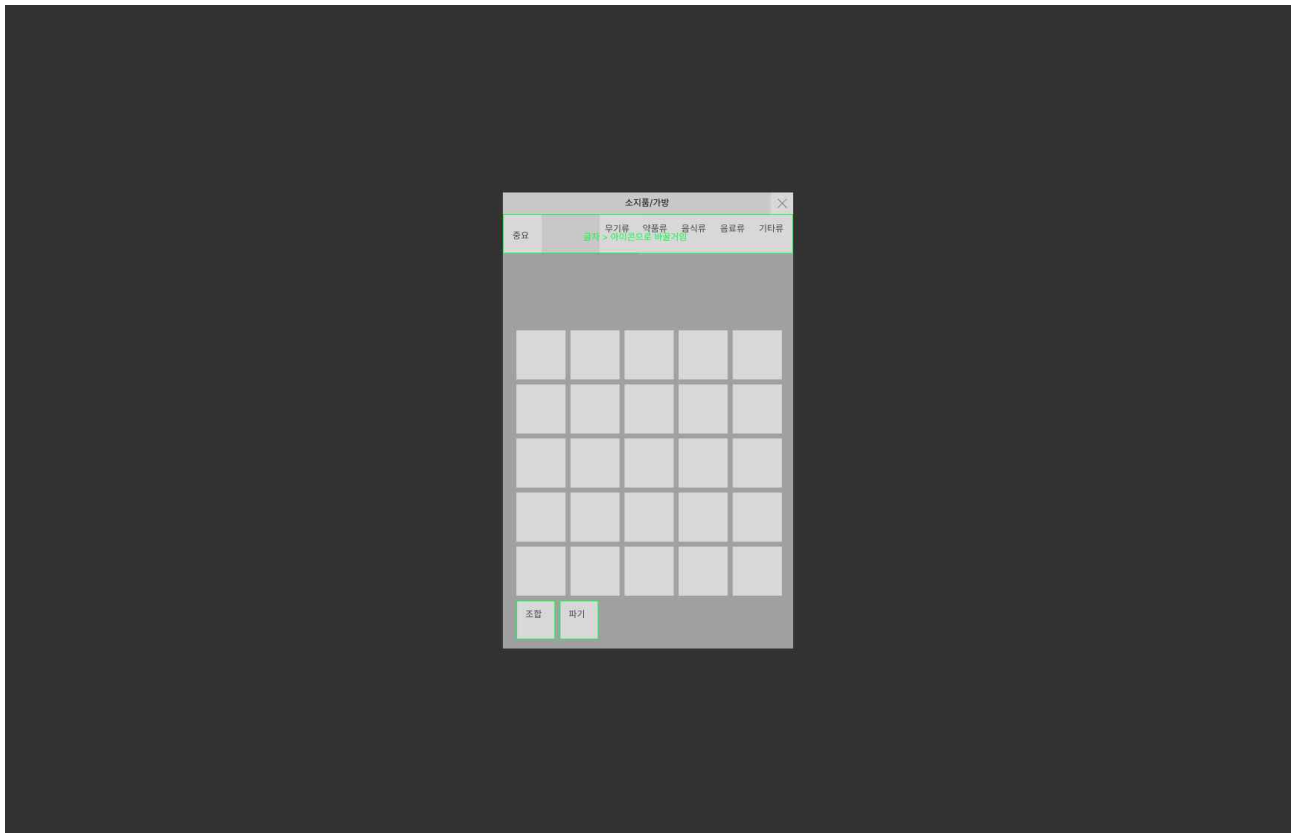


그림 7 인벤토리 UI 설계 구조

인벤토리 화면의 카테고리 하단 슬롯에 아이템에 마우스를 가져다 대면 아래 그림 8과 같이 아이템의 세부 정보를 확인할 수 있는 팝업이 표시된다.

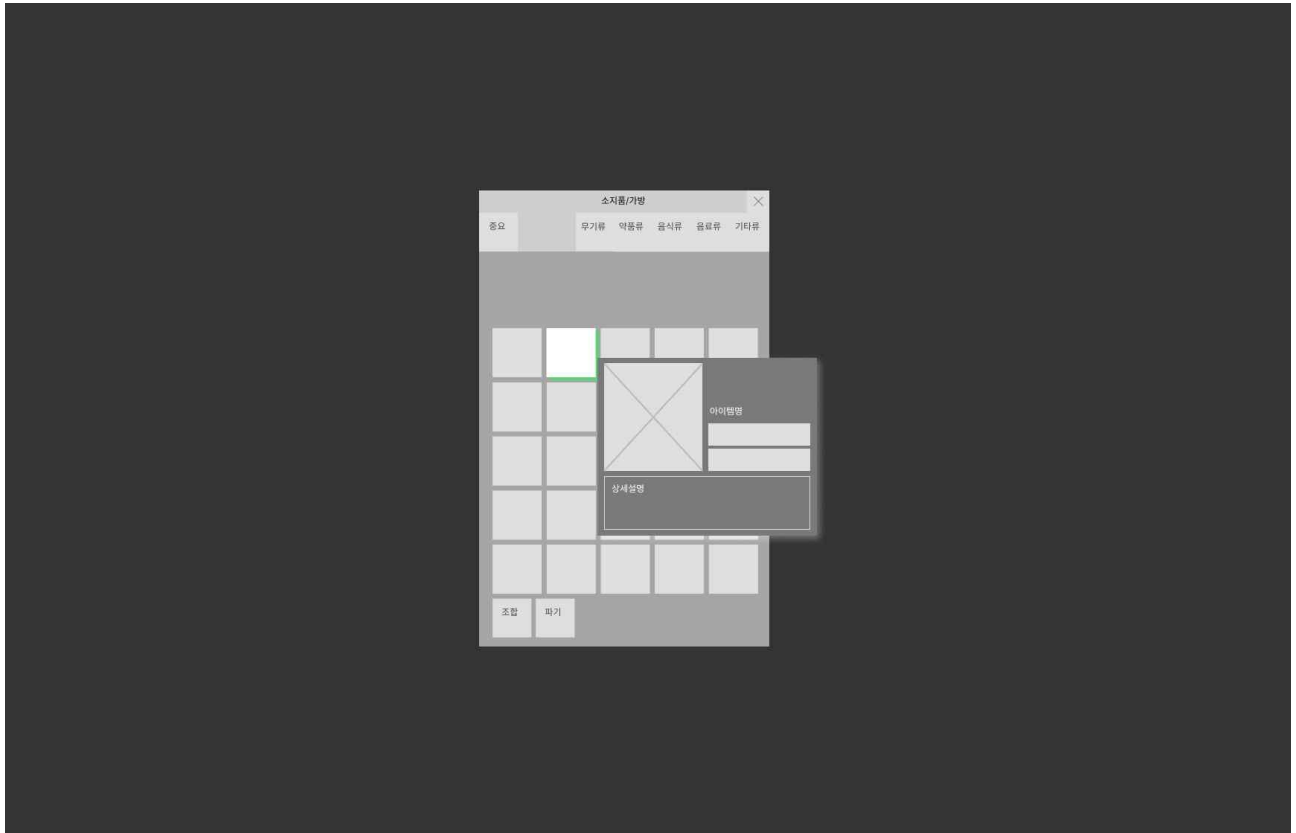


그림 8 아이템 세부 정보 팝업 UI 설계 구조

아래 그림 9는 인벤토리의 아이템 조합 기능을 실행하였을 때 조합 기능을 사용하기 위한 UI이다. 아이템 조합 시스템 UI의 1, 2 칸에 아이템을 올려두고 실행 버튼을 누르면 조합에 따라 조합된 결과 아이템이 하단 물음표 박스 위치에 표시되어 아이템을 획득할 수 있다. 만약 재료 아이템을 잘못 올린 경우, 하단 새로고침 버튼을 통해 재료 아이템 칸을 비울 수 있다.

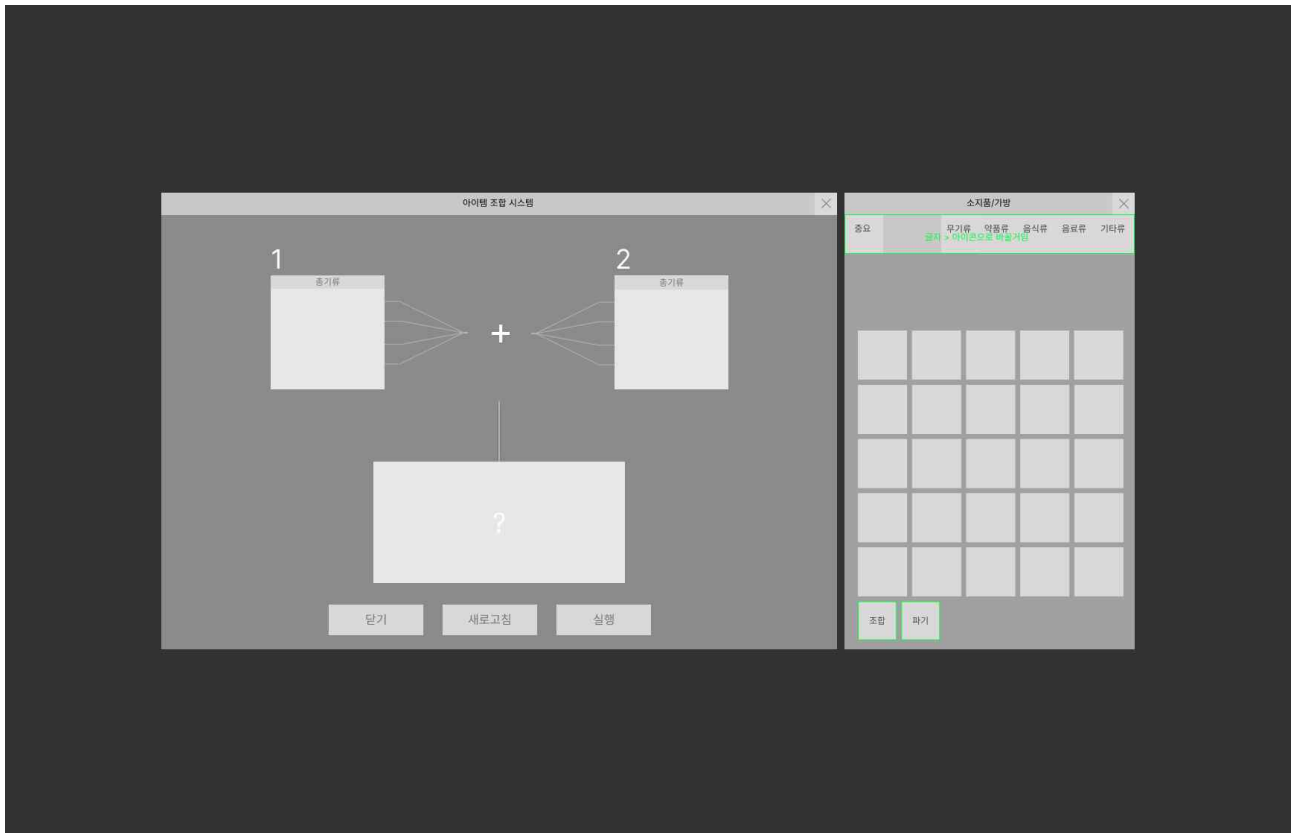


그림 9 아이템 조합 UI 설계 구조

아래 그림 10은 게임 진행 중 표시되는 클리어 메시지나 사망 메시지 등 알람 메시지를 표현하기 위한 알람 메시지 UI 설계 구조이다.

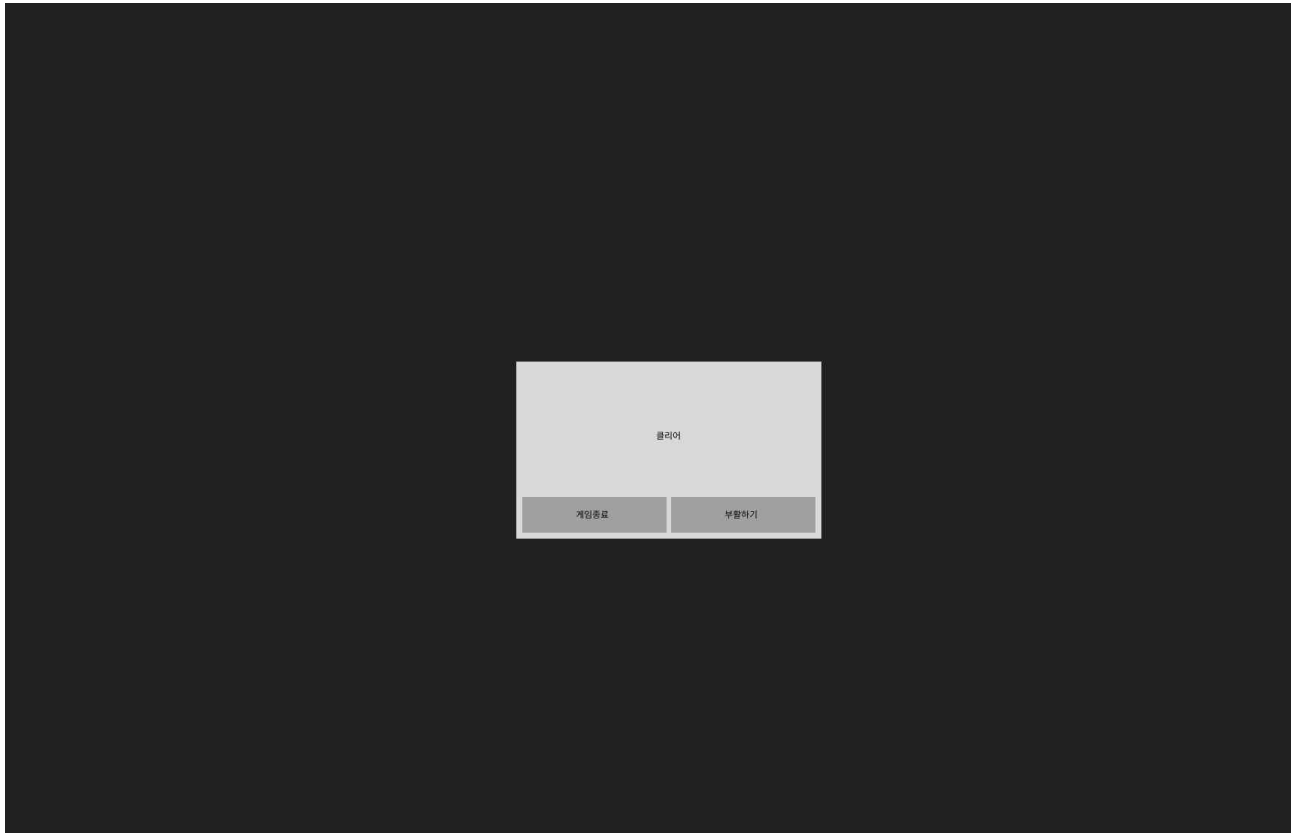


그림 10 알람 메시지 UI 설계 구조

## 2. 세부 설계

### 2.1 Character

#### 2.1.1 Character의 기능

Character(이하 캐릭터)는 플레이어 캐릭터와 적 캐릭터를 모두 통칭하는 것으로, 플레이어 캐릭터와 적 캐릭터 모두 이동 속도, 달리기 속도, 최대 체력과 현재 체력을 지니며 데미지를 받았을 때 이를 처리하는 기능과 캐릭터가 사망하였을 때의 처리 및 사망 상태 확인 기능을 수행한다.

Player(이하 플레이어)는 플레이어 캐릭터에 한정하여 인벤토리 기능과 장비 장착 기능을 수행하고 달리기, 장전 등과 같이 플레이어의 키보드 조작과 관련된 기능들을 수행한다.

Enemy(이하 적)는 적 캐릭터에 한정하여 플레이어를 인식하고 추적하는 적 움직임에 대한 기능을 수행한다.

#### 2.1.2 Character의 클래스 다이어그램

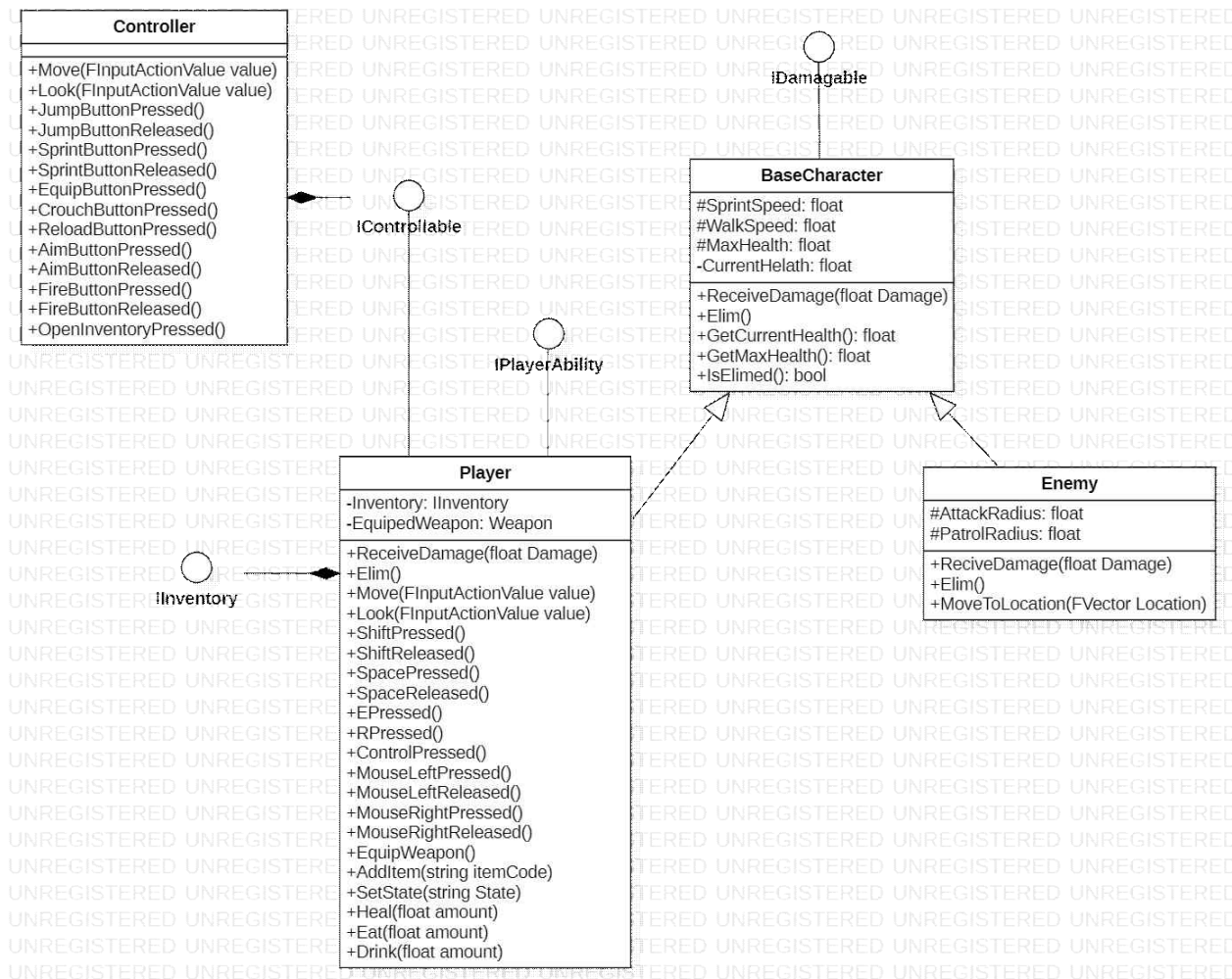


그림 11 Character의 클래스 다이어그램



BaseCharacter			
Class	캐릭터의 기본 기능을 담당하는 클래스		
Description			
구분	Name	Type	Visibility
Attribute	Description		
	SprintSpeed	float	protected
	캐릭터 기본 이동 속도		
	WalkSpeed	float	protected
	캐릭터 달리기 속도		
	MaxHealth	float	protected
	캐릭터 최대 체력		
	CurrentHealth	float	private
	캐릭터 현재 체력		
구분	Name	Type	Visibility
Operations	Description		
	ReceiveDamage	void	public
	실제 데미지 계산이 일어날 때 호출되는 함수이다. 데미지를 받는 캐릭터와 데미지량, 데미지 타입, 데미지를 준 캐릭터와 그 컨트롤러를 입력으로 받아 데미지 계산을 처리한다. 데미지를 받은 컨트롤러는 현재 체력에서 받은 데미지만큼 체력을 감소시킨다.		
	Elim	void	public
	캐릭터의 체력이 0 이하로 떨어졌을 때 호출되는 함수이다. 캐릭터의 체력이 0 이하로 된 경우 캐릭터 사망을 처리한다.		
	GetCurrentHealth	float	public
	캐릭터의 현재 체력값을 반환한다.		
	GetMaxHealth	float	public
	캐릭터의 최대 체력값을 반환한다.		
	IsElimed	bool	public
	캐릭터의 사망 상태값을 반환한다.		

Player			
Class	플레이어 캐릭터의 기능을 담당하는 클래스		
Description			
구분	Name	Type	Visibility
Attribute	Description		
	Inventory	Inventory	private
	플레이어의 인벤토리		
	EquipedWeapon	Weapon	private
	플레이어의 장비		
구분	Name	Type	Visibility
Operations	Description		
	ReceiveDamage	void	public
	실제 데미지 계산이 일어날 때 호출되는 함수이다. 데미지를 받는 캐릭터와 데미지량, 데미지 타입, 데미지를 준 캐릭터와 그 컨트롤러를 입력으로 받아 데미지 계산을 처리한다. 데미지를 받은 컨트롤러는 현재 체력에서 받은 데미지만큼 체력을 감소시킨다.		
	Elim	void	public
	캐릭터의 체력이 0 이하로 떨어졌을 때 호출되는 함수이다. 캐릭터의 체력이 0 이하로 된 경우 캐릭터 사망을 처리한다.		
	Move	void	public
	캐릭터를 움직일 때 호출되는 함수이다. 캐릭터 움직임 입력값을 받아 해당 움직임 입력값에 해당하는 방향, 각도, 위치로 캐릭터를 이동시킨다.		
	Look	void	public
	캐릭터 시야를 움직일 때 호출되는 함수이다. 캐릭터 시야 입력값을 받아 해당 시야 입력값에 해당하는 방향, 각도로 시야를 이동시킨다.		
	ShiftPressed	void	public
	키보드 Shift키를 누르면 호출되는 함수이다. Shift키가 눌리고 있는 동안 캐릭터 이동 상태를 걷기 상태에서 달리기 상태로 움직이도록 바꾼다.		
	ShiftReleased	void	public
	키보드 Shift키를 누르고 있다가 떼면 호출되는 함수이다. 캐릭터 이동 상태를 달리기 상태에서 걷기 상태로 움직이도록 바꾼다.		
	SpacePressed	void	public
	키보드 Space키를 누르면 호출되는 함수이다. Space키가 눌리고 있는 동안 캐릭터가 점프 동작을 수행하도록 한다.		
	SpaceReleased	void	public
	키보드 Space키를 누르고 있다가 떼면 호출되는 함수이다. 캐릭터가 점프 동작을 멈추도록 한다.		
	EPressed	void	public
	키보드 E키를 누르면 호출되는 함수이다. 캐릭터가 소지한 2개의 장비 중 현재 사용하고 있는 장비와 뒤에 장비한 장비를 교체한다.		
	RPressed	void	public
	키보드 R키를 누르면 호출되는 함수이다. 현재 사용하고 있는 장비가 총기류일 경우 총기를 재장전한다.		

ControlPressed	void	public
키보드의 Control키를 누르면 호출되는 함수이다. 캐릭터를 웅크리게 한다. 만약 캐릭터가 웅크린 상태라면 웅크린 상태를 해제한다.		
MouseLeftPressed	void	public
마우스 왼쪽 버튼을 누르면 호출되는 함수이다. 마우스 왼쪽 버튼을 누르고 있는 동안 현재 사용하고 있는 장비의 공격 함수를 실행한다.		
MouseLeftReleased	void	public
마우스 왼쪽 버튼을 누르고 있다가 떴을 때 호출되는 함수이다. 마우스 왼쪽 버튼을 누르고 있는 동안 실행되던 장비의 공격 함수의 실행을 멈춘다.		
MouseRightPressed	void	public
마우스 오른쪽 버튼을 누르면 호출되는 함수이다. 마우스 오른쪽 버튼을 누르고 있는 동안 조준 상태로 변경한다.		
MouseRightReleased	void	public
마우스 오른쪽 버튼을 누르고 있다가 떴을 때 호출되는 함수이다. 마우스 오른쪽 버튼을 누르고 있는 동안 유지되던 캐릭터 조준 상태를 해제한다.		
EquipWeapon	void	public
장비를 획득하면 호출되는 함수이다. 장비를 입력으로 받아 해당 장비를 장착한다.		
AddItem	void	public
아이템을 획득하면 호출되는 함수이다. 획득한 아이템의 아이템 코드를 입력으로 받아 해당 아이템을 인벤토리에 추가한다.		
SetState	void	public
캐릭터 상태가 변화할 때 호출되는 함수이다. 캐릭터 상태를 입력으로 받아 해당 상태에 맞게 캐릭터 상태를 변화시킨다.		
Heal	void	public
캐릭터 체력을 회복할 때 호출되는 함수이다. 체력 회복량을 입력으로 받아 체력 회복량만큼 캐릭터의 체력을 회복시킨다.		
Eat	void	public
캐릭터 포만감을 회복할 때 호출되는 함수이다. 포만감 회복량을 입력으로 받아 포만감 회복량만큼 캐릭터의 체력을 회복시킨다.		
Drink	void	public
캐릭터 수분을 회복할 때 호출되는 함수이다. 수분 회복량을 입력으로 받아 수분 회복량만큼 캐릭터의 체력을 회복시킨다.		

Enemy			
Class	적 캐릭터의 기능을 담당하는 클래스		
Description			
구분	Name	Type	Visibility
	Description		
Attribute	CombatRadius	float	protected
	적 캐릭터의 플레이어 감지 범위		
	PatrolRadius	float	protected
	적 캐릭터의 행동 반경		
구분	Name	Type	Visibility
	Description		
Operations	ReciveDamage	void	public
	실제 데미지 계산이 일어날 때 호출되는 함수이다. 데미지를 받는 캐릭터와 데미지량, 데미지 타입, 데미지를 준 캐릭터와 그 컨트롤러를 입력으로 받아 데미지 계산을 처리한다. 데미지를 받은 컨트롤러는 현재 체력에서 받은 데미지만큼 체력을 감소시킨다.		
	Elim	void	public
	캐릭터의 체력이 0 이하로 떨어졌을 때 호출되는 함수이다. 캐릭터의 체력이 0 이하로 된 경우 캐릭터 사망을 처리한다.		
	MoveToLocation	void	public
	적 캐릭터를 움직일 때 호출되는 함수이다. 적 캐릭터가 이동할 위치 벡터 구조체를 입력으로 받아 해당 위치로 적 캐릭터를 이동시킨다.		

Controller			
Class	캐릭터 조작 기능을 관리하기 위한 클래스		
Description			
구분	Name	Type	Visibility
	Description		
Operations	Move	void	public
	차량을 움직일 때 호출되는 함수이다. 차량 움직임 입력값을 받아 해당 움직임 입력값에 해당하는 방향, 각도, 위치로 차량을 이동시킨다.		
	Look	void	public
	차량 시야를 움직일 때 호출되는 함수이다. 차량 시야 입력값을 받아 해당 시야 입력값에 해당하는 방향, 각도로 시야를 이동시킨다.		
	JumpButtonPressed	void	public
	점프 버튼을 누를 때 호출되는 함수이다. 점프 버튼이 눌리고 있는 동안 캐릭터가 점프 동작을 수행하도록 한다.		
	JumpButtonReleased	void	public
	점프 버튼을 누르고 있다가 떼면 호출되는 함수이다. 캐릭터가 점프 동작을 멈추도록 한다.		
	SprintButtonPressed	void	public
	달리기 버튼을 누를 때 호출되는 함수이다. 달리기 버튼이 눌리고 있는 동안 캐릭터 이동 상태를 걷기 상태에서 달리기 상태로 움직이도록 바꾼다.		

SprintButtonReleased	void	public
달리기 버튼을 누르고 있다가 떼면 호출되는 함수이다. 캐릭터 이동 상태를 달리기 상태에서 걷기 상태로 움직이도록 바꾼다.		
EquipButtonPressed	void	public
장비 장착 버튼을 누르면 호출되는 함수이다. 장착 버튼을 누른 장비를 캐릭터에 장착한다.		
CrouchButtonPressed	void	public
웅크리기 버튼을 누르면 호출되는 함수이다. 캐릭터를 웅크리게 한다. 만약 캐릭터가 웅크린 상태라면 웅크린 상태를 해제한다.		
ReloadButtonPressed	void	public
재장전 버튼을 누르면 호출되는 함수이다. 현재 사용하고 있는 장비가 총기류일 경우 총기를 재장전한다.		
AimButtonPressed	void	public
조준 버튼을 누르면 호출되는 함수이다. 조준 버튼을 누르고 있는 동안 조준 상태로 변경한다.		
AimButtonReleased	void	public
조준 버튼을 누르고 있다가 떼면 호출되는 함수이다. 조준 버튼을 누르고 있는 동안 유지되던 캐릭터 조준 상태를 해제한다.		
FireButtonPressed	void	public
공격 버튼을 누르면 호출되는 함수이다. 공격 버튼을 누르고 있는 동안 플레이어가 공격을 수행한다.		
FireButtonReleased	void	public
공격 버튼을 누르고 있다가 떼면 호출되는 함수이다. 공격 버튼을 누르고 있는 동안 실행되던 플레이어의 공격을 멈춘다.		
OpenInventoryPressed	void	protected
인벤토리 버튼을 누르면 호출되는 함수이다. 게임 화면에 인벤토리 UI를 표시한다.		

## 2.2 Item

### 2.2.1 Item의 기능

Item(이하 아이템)은 아이템 코드를 통해 아이템을 획득하고 사용하는 등 아이템과 상호작용하는 기능과 아이템을 조합하여 새로운 아이템을 획득하는 기능을 수행한다.

### 2.2.2 Item의 클래스 다이어그램

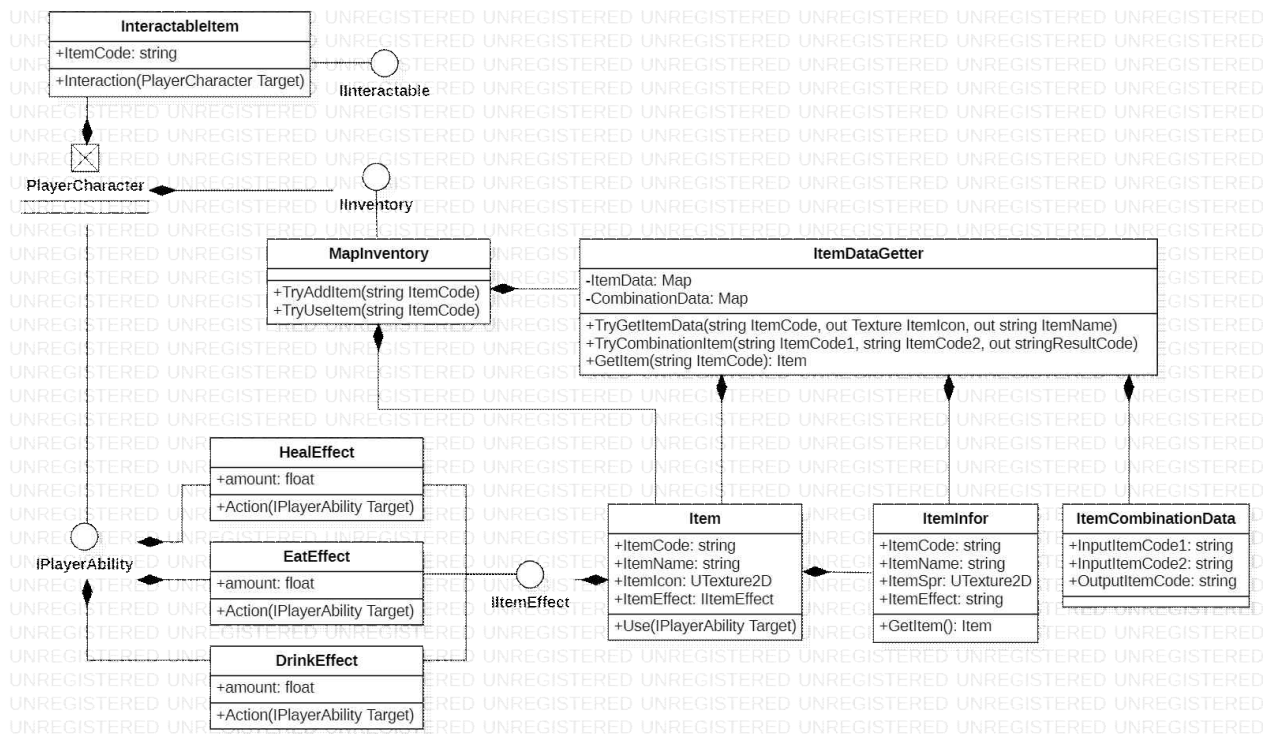


그림 12 Item의 클래스 다이어그램

InteractableItem			
Class			
Description	플레이어와 아이템 간 상호작용 기능을 담당하는 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	ItemCode	string	public
	아이템을 구분하기 위한 코드		
구분	Name	Type	Visibility
	Description		
Operations	Interaction	void	public
	<p>아이템을 습득할 때 호출되는 함수이다.</p> <p>아이템과 상호작용한 액터를 입력으로 받아 아이템 코드를 액터에게 넘겨주어 AddItem 함수를 통해 아이템을 습득할 수 있도록 하고 습득된 아이템을 필드에서 제거한다.</p>		

MapInventory			
Class			
Description	인벤토리 기능을 담당하는 클래스		
구분	Name	Type	Visibility
	Description		
Operations	TryAddItem	bool	public
	<p>아이템을 습득할 때 호출되는 함수이다.</p> <p>아이템 코드를 입력으로 받아 인벤토리에 해당 아이템 정보를 추가하고 true 값을 반환한다. 또한 아이템이 존재한다면 해당 아이템의 소지 수를 1 늘리고 true 값을 반환한다. 만약 인벤토리에 아이템이 들어갈 공간이 없다면 false 값을 반환한다.</p>		
	TryUseItem	bool	public
	<p>아이템을 사용할 때 호출되는 함수이다.</p> <p>아이템 코드를 입력으로 받아 인벤토리에 아이템이 존재한다면 아이템 소지 수를 1 차감하고 아이템의 소지 수가 0이 된다면 인벤토리에서 해당 아이템 정보를 삭제하고 true 값을 반환한다. 만약 아이템이 인벤토리에 존재하지 않는다면 false 값을 반환한다.</p>		

ItemDataGetter			
Class	아이템 데이터를 가져오기 위한 클래스		
Description			
구분	Name	Type	Visibility
	Description		
Attribute	ItemData	Map	protected
	아이템 정보		
	CombinationData	Map	protected
	아이템 조합 정보		
구분	Name	Type	Visibility
	Description		
Operations	TryGetItemData	void	public
	아이템 정보를 가져올 때 호출되는 함수이다. 아이템 코드, 아이템 아이콘, 아이템 이름을 입력으로 받아 해당 아이템 정보를 UI에 표기한다.		
	TryCombinationItem	void	public
	아이템을 조합할 때 호출되는 함수이다. 조합에 필요한 아이템 코드 2개와 결과 아이템 코드를 입력으로 받아 재료 아이템을 소모하고 조합된 아이템을 생성한다.		
	GetItem	Item	public
	아이템 정보를 가져올 때 호출되는 함수이다. 아이템 코드를 입력으로 받아 해당하는 아이템 정보를 반환한다.		

Item			
Class	아이템을 관리하는 클래스		
Description			
구분	Name	Type	Visibility
	Description		
Attribute	ItemCode	string	public
	아이템 코드		
	ItemName	string	public
	아이템 이름		
	ItemIcon	Texture2D	public
	아이템 아이콘 이미지		
	ItemEffect	ItemEffect	public
	아이템 효과		
구분	Name	Type	Visibility
	Description		
Operations	Use	void	public
	아이템을 사용할 때 호출되는 함수이다. 대상 캐릭터를 입력으로 받아 해당 아이템을 소모하고 그 효과를 적용한다.		



ItemInfor			
Class			
Description	아이템 정보를 담당하는 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	ItemCode	string	public
	아이템 코드		
	ItemName	string	public
	아이템 이름		
	ItemSpr	Texture2D	public
	아이템 아이콘 이미지		
	ItemEffect	string	public
	아이템 효과		
구분	Name	Type	Visibility
	Description		
Operations	GetItem()	Item	public
	아이템 정보를 반환하는 함수이다. 선택한 함수의 아이템 정보를 반환한다.		

ItemCombinationData			
Class			
Description	아이템 조합 기능을 담당하는 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	InputItemCode1	string	public
	아이템 조합에 사용되는 아이템 코드 1		
	InputItemCode2	string	public
	아이템 조합에 사용되는 아이템 코드 2		
	OutputItemCode	string	public
	아이템 코드 1, 2를 재료로 획득하게 되는 아이템의 아이템 코드		

HealEffect			
Class			
Description	체력 회복 효과를 담당하는 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	amount	float	public
	체력 회복량		
구분	Name	Type	Visibility
	Description		
Operations	Action	void	public
	체력 회복 효과를 수행할 때 호출되는 함수이다. 대상 캐릭터를 입력으로 받아 체력 회복 효과를 적용한다.		

EatEffect			
Class			
Description	포만감 회복 효과를 담당하는 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	amount	float	public
	포만감 회복량		
구분	Name	Type	Visibility
	Description		
Operations	Action	void	public
	포만감 회복 효과를 수행할 때 호출되는 함수이다. 대상 캐릭터를 입력으로 받아 포만감 회복 효과를 적용한다.		

DrinkEffect			
Class			
Description	수분 회복 효과를 담당하는 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	amount	float	public
	수분 회복량		
구분	Name	Type	Visibility
	Description		
Operations	Action	void	public
	수분 회복 효과를 수행할 때 호출되는 함수이다. 대상 캐릭터를 입력으로 받아 수분 회복 효과를 적용한다.		

## 2.3 Interface

### 2.3.1 Interface의 기능

Interface(이하 인터페이스)는 상호작용, 데미지 연산, 캐릭터 컨트롤, 인벤토리, 아이템 관련 기능들을 서로 다른 범위에서 유용하게 사용되기 때문에 인터페이스를 통해 공통된 기능을 제공하기 위해 정의하였다.

### 2.3.2 Interface의 클래스 다이어그램

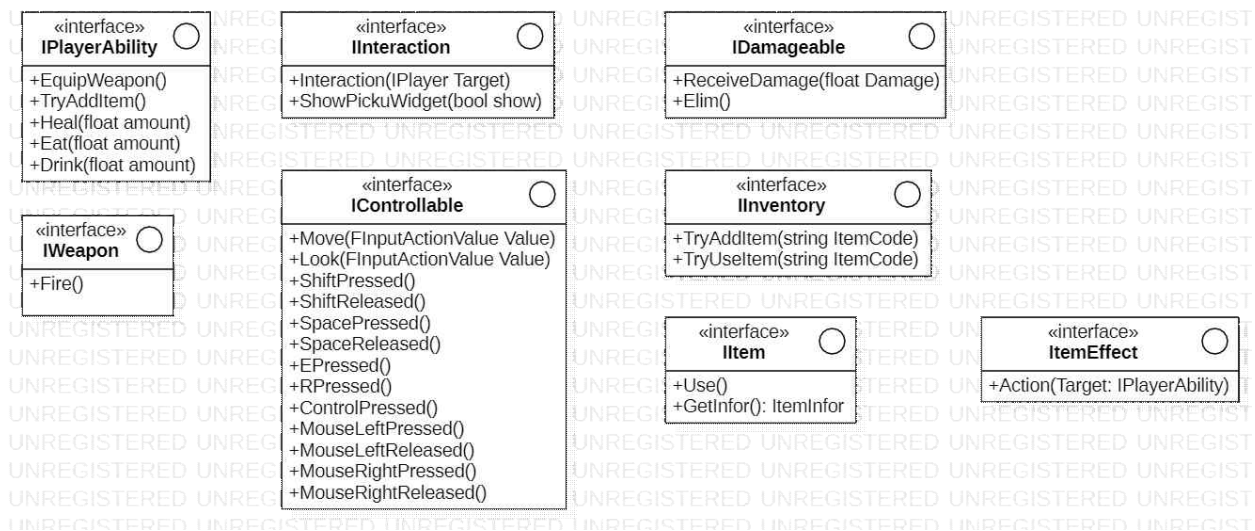


그림 13 Interface의 클래스 다이어그램

Interaction			
Class	상호작용 기능을 정의하기 위한 인터페이스		
Description			
구분	Name	Type	Visibility
Operations	Description		
	Interaction	void	public
	아이템 습득 기능을 정의한다.		
	ShowPickUpWidget	void	public
아이템 습득 시 표시되는 위젯 기능을 정의한다.			

IDamageable			
Class	데미지 연산 기능을 정의하기 위한 인터페이스		
Description			
구분	Name	Type	Visibility
Operations	Description		
	ReceiveDamage	void	public
	데미지 연산 기능을 정의한다.		
	Elim	void	public
캐릭터 사망 기능을 정의한다.			

Inventory			
Class			
Description	인벤토리 기능을 정의하기 위한 인터페이스		
구분	Name	Type	Visibility
	Description		
Operations	TryAddItem	bool	public
	인벤토리 아이템 추가 기능을 정의한다.		
	TryUseItem	bool	public
	인벤토리 아이템 사용 기능을 정의한다.		

Item			
Class			
Description	아이템 기능을 정의하기 위한 인터페이스		
구분	Name	Type	Visibility
	Description		
Operations	Use	void	public
	아이템 사용 기능을 정의한다.		
	GetInfor	void	public
	아이템 정보 표시 기능을 정의한다.		

IControllable			
Class			
Description	캐릭터 컨트롤 기능을 정의하기 위한 인터페이스		
구분	Name	Type	Visibility
	Description		
Operations	Move	void	public
	캐릭터 움직임 기능을 정의한다.		
	Look	void	public
	캐릭터 시야 조작 기능을 정의한다.		
	ShiftPressed	void	public
	캐릭터의 걷기에서 달리기로 전환되는 기능을 정의한다.		
	ShiftReleased	void	public
	캐릭터의 달리기에서 걷기로 전환되는 기능을 정의한다.		
	SpacePressed	void	public
	캐릭터의 점프 기능을 정의한다.		
	SpaceReleased	void	public
	캐릭터의 점프 상태를 멈추는 기능을 정의한다.		
	EPressed	void	public
	캐릭터의 장비 교체 기능을 정의한다.		
	RPressed	void	public
	캐릭터 무기의 재장전 기능을 정의한다.		
	ControlPressed	void	public
	캐릭터 웅크리기 기능을 정의한다.		
	MouseLeftPressed	void	public
	캐릭터 공격 기능을 정의한다.		
	MouseLeftReleased	void	public
	캐릭터 공격 멈춤 기능을 정의한다.		
	MouseRightPressed	void	public
	캐릭터 조준 기능을 정의한다.		
	MouseRightReleased	void	public
	캐릭터 조준 해제 기능을 정의한다.		

IPlayerAbility			
Class			
Description	플레이어 캐릭터의 기능을 정의하기 위한 인터페이스		
구분	Name	Type	Visibility
	Description		
Operations	EquipWeapon	void	public
	플레이어 장비 장착 기능을 정의한다.		
	TryAddItem	void	public
	플레이어 아이템 획득 기능을 정의한다.		
	Heal	void	public
	플레이어 체력 회복 기능을 정의한다.		
	Eat	void	public
	플레이어 포만감 회복 기능을 정의한다.		
	Drink	void	public
	플레이어 수분 회복 기능을 정의한다.		

IWeapon			
Class			
Description	장비 기능을 정의하기 위한 인터페이스		
구분	Name	Type	Visibility
	Description		
Operations	Fire	void	public
	장비 공격 기능을 정의한다.		

## 2.4 Weapon

### 2.4.1 Weapon의 기능

Weapon(이하 웨폰)은 게임의 무기류들에 대한 기능들을 수행하며, 주로 무기를 이용한 공격 기능을 수행한다.

무기 종류에 따라 MeleeWeapon(이하 근접 무기)과 RangeWeapon(이하 총기류)로 구분되며, 총기류의 경우 HitScanWeapon과 ProjectileWeapon으로 구분되어 HitScanWeapon의 경우 레이저 판정 방식으로 공격을 가하고 ProjectileWeapon의 경우 발사체 충돌 판정 방식으로 공격을 가한다.

웨폰은 캐릭터 공격에 있어 실제 공격 범위 판정 기능 및 공격 기능을 수행하며, 캐릭터 데미지 연산을 위한 공격 정보 전달 기능 등 실제 공격 행위가 일어날 때 데미지 처리까지 필요한 기능들을 수행한다.

### 2.4.2 Weapon의 클래스 다이어그램

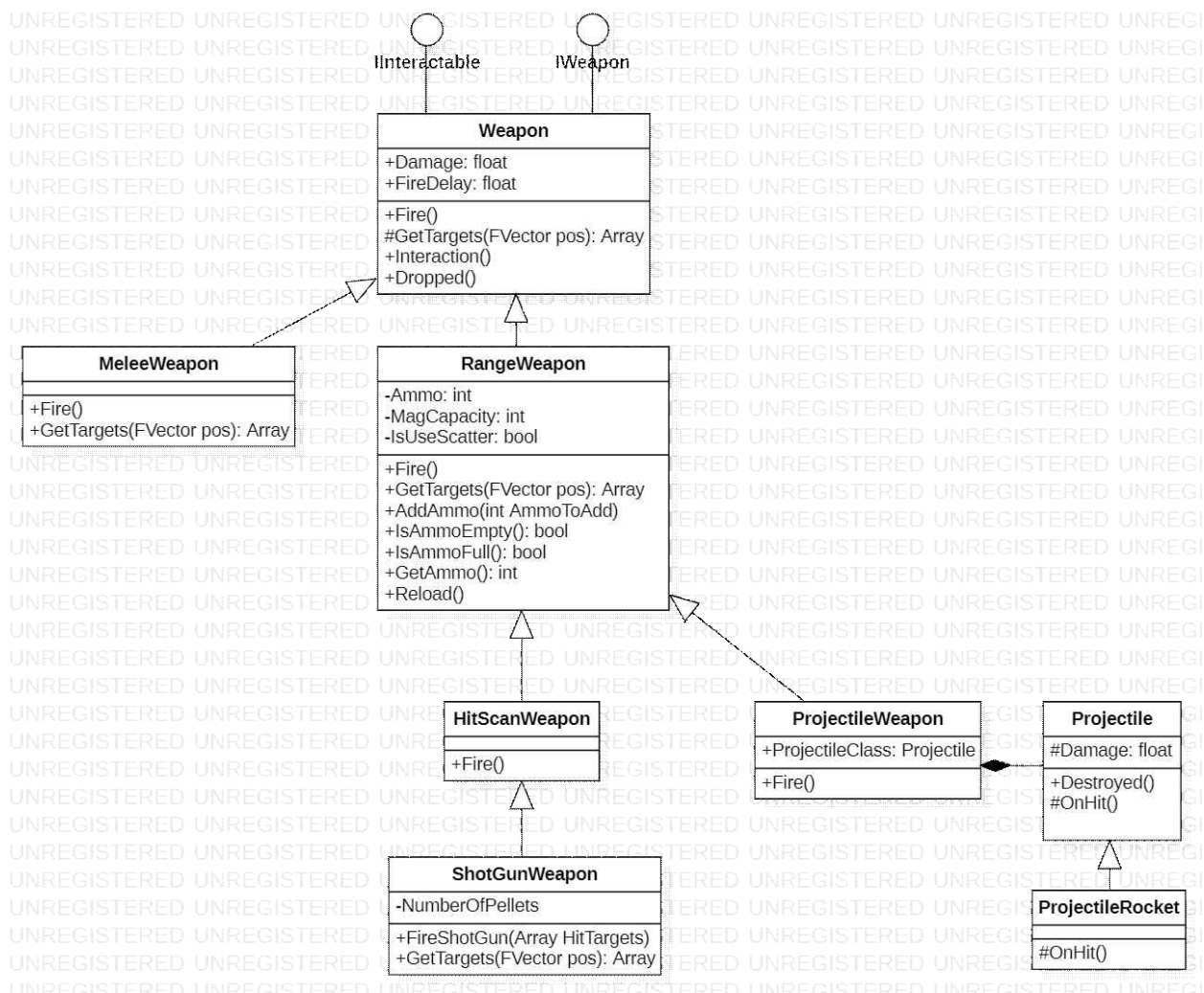


그림 14 Weapon의 클래스 다이어그램

Weapon			
Class	무기의 기본 기능을 담당하는 클래스		
Description			
구분	Name	Type	Visibility
	Description		
Attribute	Damage	float	public
	무기의 데미지량		
	FireDelay	float	public
	무기의 공격 간 딜레이(연사 속도)		
구분	Name	Type	Visibility
	Description		
Operations	Fire	void	public
	캐릭터가 공격할 때 호출되는 함수이다.		
	공격 범위 내 피격 대상들에게 데미지를 전달한다.		
	GetTargets	Array	protected
	캐릭터가 공격할 때 호출되는 함수이다.		
	무기 공격 범위를 입력으로 받아 무기 공격 범위에 해당하는 피격 대상들의 정보를 배열 형태로 반환한다.		
	Interaction	void	public
	캐릭터가 공격할 때 호출되는 함수이다.		
	피격 대상의 정보를 입력으로 받아 캐릭터가 장비한 무기에 해당 정보를 전달한다.		
	Dropped	void	public
	캐릭터가 사망할 때 호출되는 함수이다.		
	캐릭터가 지닌 장비를 드롭하여 필드에 해당 아이템을 생성한다.		

MeleeWeapon			
Class	근접 무기의 기능을 담당하는 클래스		
Description			
구분	Name	Type	Visibility
	Description		
Operations	Fire	void	public
	캐릭터가 공격할 때 호출되는 함수이다.		
	공격 범위 내 피격 대상들에게 데미지를 전달한다.		
	GetTargets	Array	public
	캐릭터가 공격할 때 호출되는 함수이다.		
	무기 공격 범위를 입력으로 받아 무기 공격 범위에 해당하는 피격 대상들의 정보를 배열 형태로 반환한다.		

RangeWeapon			
Class	총기류의 기능을 담당하는 클래스		
Description			
구분	Name	Type	Visibility
Attribute	Description		
	Ammo	int	private
	탄약의 수		
	MagCapacity	int	private
	탄창의 최대 탄약 수		
	IsUseScatter	bool	private
구분	총기가 산탄인지 아닌지를 표시		
	Name	Type	Visibility
	Description		
	Fire	void	public
	캐릭터가 공격할 때 호출되는 함수이다. 피격 대상들의 정보를 입력으로 받아 해당 대상들에게 데미지를 전달한다.		
	GetTargets	Array	public
Operations	캐릭터가 공격할 때 호출되는 함수이다. 무기 공격 범위를 입력으로 받아 무기 공격 범위에 해당하는 피격 대상들의 정보를 배열 형태로 반환한다.		
	AddAmmo	void	public
	총기를 재장전할 때 호출되는 함수이다. 더해지는 탄약의 수를 입력으로 받아 탄약 재장전 결과 총기 내 탄약의 수를 검사한다.		
	IsAmmoEmpty	bool	public
	총기 내 탄약이 비었는지 확인하는 함수이다. 총기 내 탄약의 수가 0보다 작거나 같으면 true를 반환한다.		
	IsAmmoFull	bool	public
	총기 내 탄약이 가득찼는지를 확인하는 함수이다. 총기 내 탄약이 탄창의 최대 탄약 수와 같으면 true를 반환한다.		
	GetAmmo	int	public
	현재 총기 내 탄약의 수를 확인하는 함수이다. 총기 내 탄약의 수를 반환한다.		
	Reload	void	public
	총기를 재장전할 때 호출되는 함수이다. 총기의 탄약의 수를 최대 탄약의 수까지 늘려 총기를 재장전한다.		

HitScanWeapon			
Class	레이저 판정 기능을 담당하는 클래스		
Description			
구분	Name	Type	Visibility
Operations	Description		
	Fire	void	public
	캐릭터가 공격할 때 호출되는 함수이다. 공격 범위 내 피격 대상들에게 데미지를 전달한다.		



ShotGunWeapon			
Class			
Description	샷건의 범위 판정 기능을 담당하는 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	NumberOfPellets	int	private
	샷건에 소비되는 탄약의 수		
구분	Name	Type	Visibility
	Description		
Operations	FireShotGun	void	public
	캐릭터가 공격할 때 호출되는 함수이다.		
	피격 대상들의 정보를 입력으로 받아 해당 대상들에게 데미지를 전달한다.		
	GetTargets	Array	public
	캐릭터가 공격할 때 호출되는 함수이다.		
	무기 공격 범위에 해당하는 피격 대상들의 정보를 배열 형태로 반환한다.		

ProjectileWeapon			
Class			
Description	발사체 무기 기능을 담당하는 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	ProjectileClass	Projectile	public
	발사체 타입		
구분	Name	Type	Visibility
	Description		
Operations	Fire	void	public
	캐릭터가 공격할 때 호출되는 함수이다.		
	공격 범위 내 피격 대상들에게 데미지를 전달한다.		

Projectile			
Class			
Description	발사체의 범위 판정 기능을 담당하는 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	Damage	float	protected
	발사체 피격 데미지량		
구분	Name	Type	Visibility
	Description		
Operations	Destroyed	void	public
	발사체 피격 시 호출되는 함수이다.		
	피격된 발사체를 파괴하고 이펙트와 사운드 효과를 출력한다.		
	OnHit	void	protected
	발사체 피격 시 호출되는 함수이다.		
	발사체 피격 범위 내 피격 정보에 대한 내용을 처리한다.		

ProjectileRocket			
Class			
Description	로켓 타입 발사체의 범위 판정 기능을 담당하는 클래스		
구분	Name	Type	Visibility
	Description		
Operations	OnHit	void	protected
	로켓 타입 발사체 피격 시 호출되는 함수이다. 로켓 타입 발사체 피격 범위 내 피격 정보에 대한 내용을 처리한다.		

## 2.5 Car

### 2.5.1 Car의 기능

Car는 게임 진행에 있어 플레이어가 이용할 수 있는 차량에 대한 기능들을 수행한다. 플레이어가 차량에 탑승하고 내리기 위한 상호작용 기능부터 차량을 운전한 상태에서도 플레이어가 이동하고 공격하는 등 플레이어 기능들을 수행할 수 있도록 차량 탑승 중 가능한 플레이어 기능들을 수행한다.

### 2.5.2 Car의 클래스 다이어그램

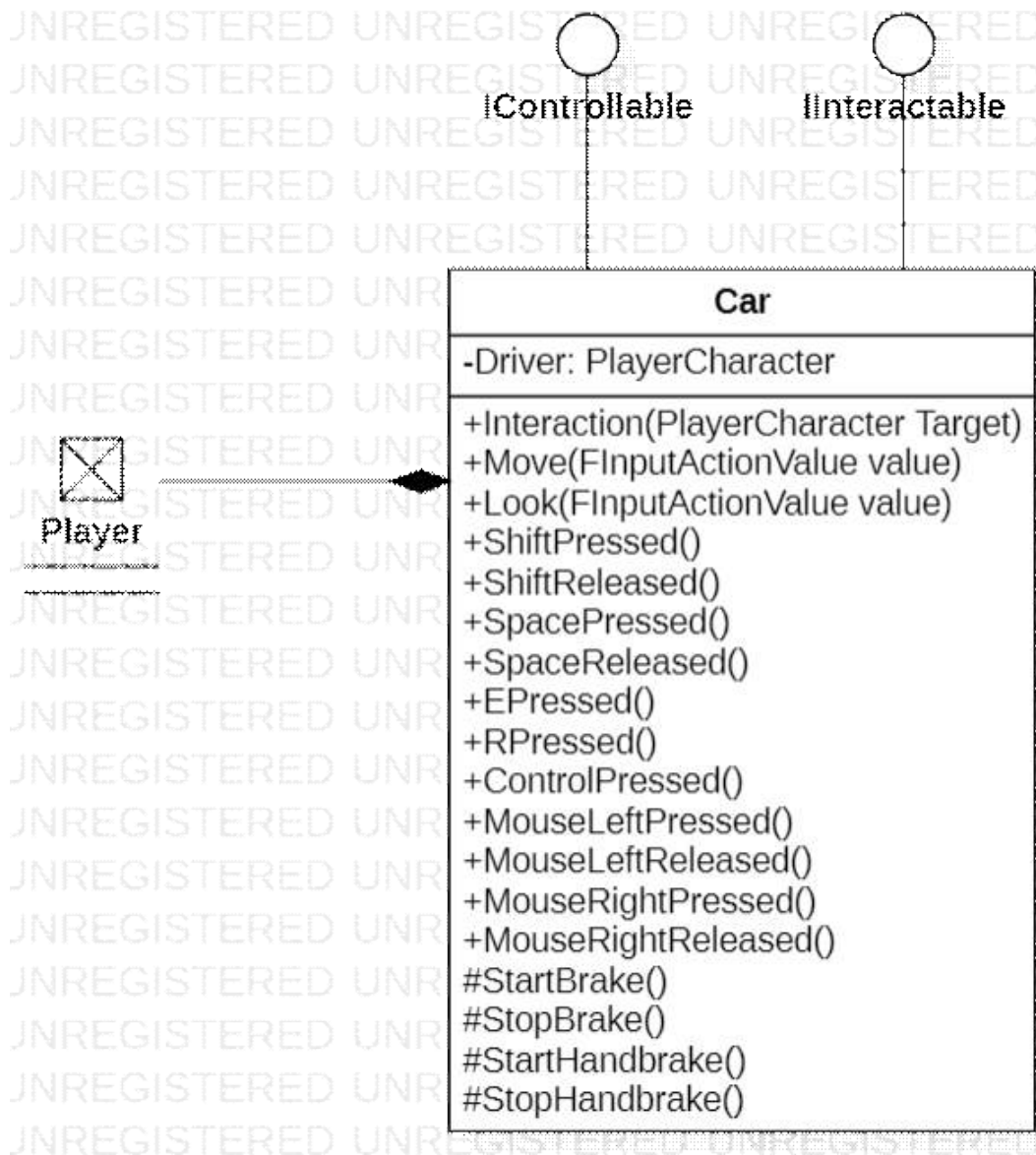


그림 15 Car의 클래스 다이어그램

Car			
Class	차량 기능을 담당하는 클래스		
Description			
구분	Name	Type	Visibility
	Description		
Attribute	Driver	PlayerCharacter	private
	차량을 운전하는 캐릭터		
구분	Name	Type	Visibility
	Description		
Operations	Interaction	void	public
	플레이어 캐릭터가 차량과 상호작용할 때 호출되는 함수이다. 플레이어 캐릭터의 정보를 받아 해당 캐릭터를 차량의 운전자로 설정하고 플레이어 컨트롤러와 상태를 차량에 맞게 적용하여 플레이어가 차량을 움직일 수 있도록 한다.		
	Move	void	public
	차량을 움직일 때 호출되는 함수이다. 차량 움직임 입력값을 받아 해당 움직임 입력값에 해당하는 방향, 각도, 위치로 차량을 이동시킨다.		
	Look	void	public
	차량 시야를 움직일 때 호출되는 함수이다. 차량 시야 입력값을 받아 해당 시야 입력값에 해당하는 방향, 각도로 시야를 이동시킨다.		
	ShiftPressed	void	public
	키보드 Shift키를 누르면 호출되는 함수이다. Shift키가 눌리고 있는 동안 차량 이동 속도를 높인다.		
	ShiftReleased	void	public
	키보드 Shift키를 누르고 있다가 떼면 호출되는 함수이다. Shift키를 누르고 있는 동안 높아진 차량 이동 속도를 다시 원래대로 낮춘다.		
	SpacePressed	void	public
	키보드 Space키를 누르면 호출되는 함수이다. Space키가 눌리고 있는 동안 차량을 바운스시켜 장애물을 뛰어넘을 수 있도록 한다.		
	SpaceReleased	void	public
	키보드 Space키를 누르고 있다가 떼면 호출되는 함수이다. 차량의 바운스 동작을 멈추도록 한다.		
	EPressed	void	public
	키보드 E키를 누르면 호출되는 함수이다. 차량에 탑승한 캐릭터가 소지한 2개의 장비 중 현재 사용하고 있는 장비와 뒤에 장비한 장비를 교체한다.		
	RPressed	void	public
	키보드 R키를 누르면 호출되는 함수이다. 현재 사용하고 있는 장비가 총기류일 경우 총기를 재장전한다.		
	ControlPressed	void	public
	키보드의 Control키를 누르면 호출되는 함수이다. 차량의 시점을 변경한다. 3인칭 시점이라면 1인칭 시점으로, 1인칭 시점이라면 3인칭 시점으로 변환한다.		

MouseLeftPressed	void	public
마우스 왼쪽 버튼을 누르면 호출되는 함수이다. 마우스 왼쪽 버튼을 누르고 있는 동안 현재 사용하고 있는 장비의 공격 함수를 실행한다.		
MouseLeftReleased	void	public
마우스 왼쪽 버튼을 누르고 있다가 떴을 때 호출되는 함수이다. 마우스 왼쪽 버튼을 누르고 있는 동안 실행되던 장비의 공격 함수의 실행을 멈춘다.		
MouseRightPressed	void	public
마우스 오른쪽 버튼을 누르면 호출되는 함수이다. 마우스 오른쪽 버튼을 누르고 있는 동안 조준 상태로 변경한다.		
MouseRightReleased	void	public
마우스 오른쪽 버튼을 누르고 있다가 떴을 때 호출되는 함수이다. 마우스 오른쪽 버튼을 누르고 있는 동안 유지되던 캐릭터 조준 상태를 해제한다.		
StartBrake	void	protected
차량 브레이크 작동 시 호출되는 함수이다. 브레이크 입력 액션을 입력으로 받아 차량 브레이크 등을 점등한다.		
StopBrake	void	protected
차량 브레이크 작동 종료 시 호출되는 함수이다. 브레이크 입력 액션을 입력으로 받아 차량 브레이크 등을 소등한다.		
StartHandbrake	void	protected
차량 핸드 브레이크 작동 시 호출되는 함수이다. 핸드 브레이크 입력 액션을 입력으로 받아 차량 브레이크 등을 점등한다.		
StopHandbrake	void	protected
차량 핸드 브레이크 작동 종료 시 호출되는 함수이다. 핸드 브레이크 입력 액션을 입력으로 받아 차량 브레이크 등을 소등한다.		

## 2.6 Component

### 2.6.1 Component의 기능

Component(이하 컴포넌트)는 플레이어 캐릭터에 부가적으로 붙어 플레이어 캐릭터가 다른 요소들을 사용할 수 있게끔 다른 요소들과 연결하는 기능을 한다. 무기를 장비하고 물자와 같은 다른 요소들과 상호작용하며 플레이어 상태에 따른 이펙트를 주는 등 플레이어 캐릭터에 부가적인 기능들에 대한 역할들을 수행한다.

### 2.6.2 Component의 클래스 다이어그램

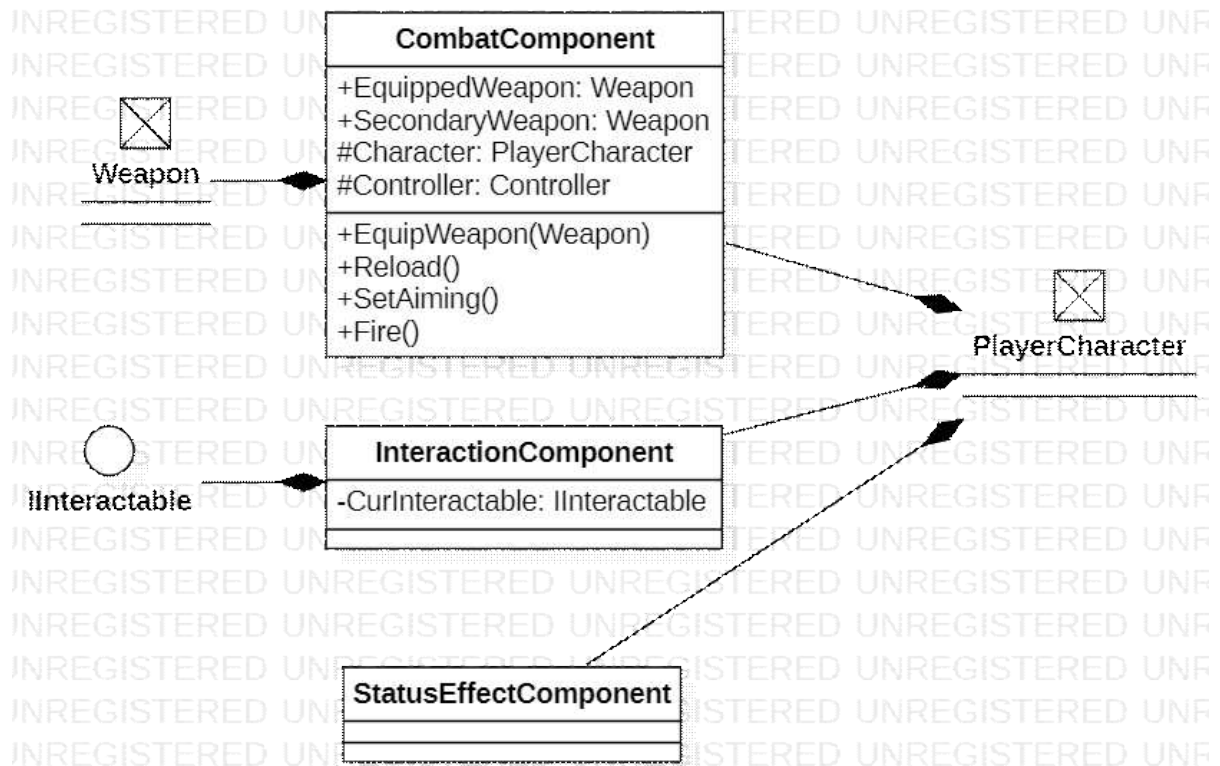


그림 16 Component의 클래스 다이어그램

CombatComponent			
Class			
Description	캐릭터가 장비하고 있는 무기 기능을 위한 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	EquippedWeapon	Weapon	public
	플레이어가 현재 사용하고 있는 무기		
	SecondaryWeapon	Weapon	public
	플레이어가 보조 역할로 장비하고 있는 무기		
	Character	PlayerCharacter	protected
	플레이어 캐릭터		
	Controller	Controller	protected
	캐릭터 컨트롤러		
구분	Name	Type	Visibility
	Description		
Operations	EquipWeapon	void	public
	무기를 장비할 때 호출되는 함수이다.		
	장비할 무기를 입력으로 받아 플레이어 캐릭터에 해당 무기를 장비한다.		
	Reload	void	public
	총기 재장전을 할 때 호출되는 함수이다.		
	플레이어 캐릭터가 장비한 총기류 무기를 재장전한다.		
	SetAiming	void	public
	총기 조준 사격을 할 때 호출되는 함수이다.		
	플레이어 캐릭터가 총기를 조준 모드 화면으로 전환한다.		
	Fire	void	public
	캐릭터가 공격할 때 호출되는 함수이다.		
	피격 대상들의 정보를 입력으로 받아 해당 대상들에게 데미지를 전달한다.		

InteractionComponent			
Class			
Description	캐릭터 상호작용을 위한 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	CurlInteractable	IInteractable	public
	현재 상호작용하고 있는 대상		

StatusEffectComponent	
Class	
Description	캐릭터 상태 표시를 위한 클래스

## 2.7 GameMode

### 2.7.1 GameMode의 기능

GameMode는 말 그대로 게임 플레이 시작 시 선택되는 게임 모드와 관련한 기능들을 위한 클래스이다. 플레이어가 선택한 게임 모드에 따라 추가 및 삭제되는 요소들을 세팅하여 게임이 해당 모드에 맞게 플레이될 수 있도록 한다.

### 2.7.2 GameMode의 클래스 다이어그램

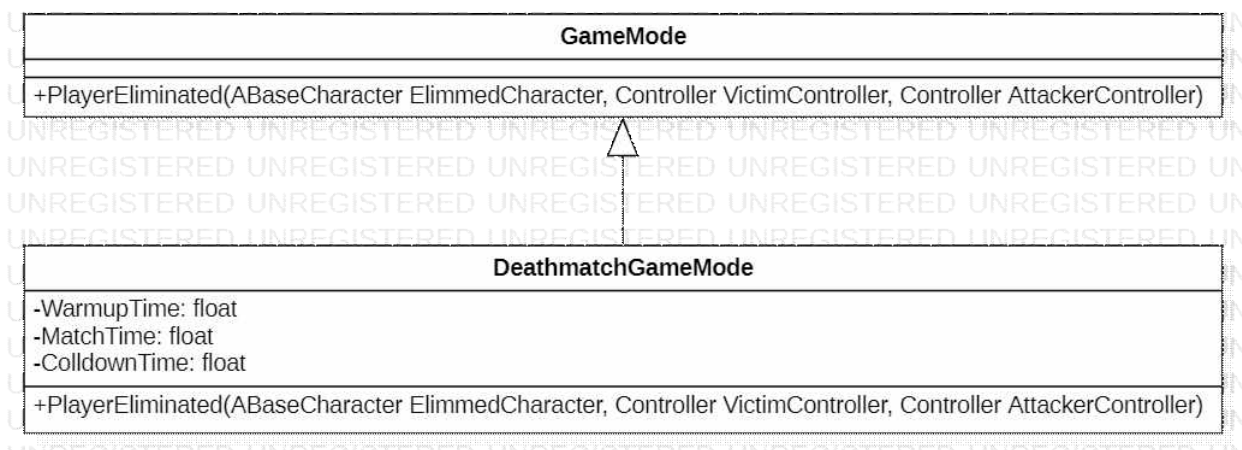


그림 17 GameMode의 클래스 다이어그램

GameMode			
Class	게임 모드를 관리하기 위한 클래스		
Description			
구분	Name	Type	Visibility
	Description		
Operations	PlayerEliminated	void	public
	플레이어가 제거될 때 호출되는 함수이다. 제거된 캐릭터와 해당 캐릭터의 컨트롤러, 제거한 플레이어의 컨트롤러를 입력으로 받아 제거된 캐릭터의 사망 처리를 수행한다.		



DeathmatchGameMode			
Class			
Description	데스매치 게임 모드를 관리하기 위한 클래스		
구분	Name	Type	Visibility
	Description		
Attribute	WarmupTime	float	private
	워밍업 시간		
	MatchTime	float	private
	매치 시간		
	CooldownTime	float	private
	휴식 시간		
구분	Name	Type	Visibility
	Description		
Operations	PlayerEliminated	void	public
	<p>플레이어가 제거될 때 호출되는 함수이다.</p> <p>제거된 캐릭터와 해당 캐릭터의 컨트롤러, 제거한 플레이어의 컨트롤러를 입력으로 받아 제거된 캐릭터의 사망 처리를 수행한다.</p>		

### 3. 수행 방법

#### 3.1 설계 및 개발 환경

본 프로젝트는 기획, 프로그래밍, 디자인 3가지 분야에서 프로젝트를 수행하여야 한다.

- 기획은 게임 전반적인 내용 및 플로우 구상과 더불어 세부 시스템 설계를 담당하며 프로그래밍 및 디자인 중앙에서 의견 및 일정 조율을 담당한다.
- 프로그래밍은 언리얼 5 엔진을 이용하여 C++와 블루프린트를 사용하여 실제 게임 시스템을 구현한다.
- 디자인은 UI 구조 배치 및 실제 사용되는 3D 모델링과 관련하여 UI 디자인을 담당하고 기타 사용될 에셋들과 관련된 작업을 수행한다.

#### 3.2 설계 및 개발 도구

개발 도구는 프로그래밍과 디자인 업무에 따라 다음과 같은 도구를 가진다.

- 프로그래밍 : -S/W : 언리얼 엔진 5.3.2, Visual Studio Code 1.88, Rider 2024.1
- 디자인 : - SW : Adobe, Figma

#### 3.3 테스트 방안

본 프로젝트 테스트에 있어 가장 중요한 요소는 서로 다른 환경 간 멀티 플레이에 있어 동기화가 얼마나 잘 되는가가 중요한 요소로 같은 환경에서 1차적으로 멀티 플레이 테스트를 거친 후 서로 다른 기기에서 멀티 플레이를 테스트하고 실제 4인 멀티 플레이를 테스트 플레이하는 것으로 테스트 범위를 점차 확장할 계획이다.

#### 4. 참조 자료

언리얼 엔진 5.3 문서

[https://dev.epicgames.com/documentation/ko-kr/unreal-engine/unreal-engine-5-3-documentation?application\\_version=5.3](https://dev.epicgames.com/documentation/ko-kr/unreal-engine/unreal-engine-5-3-documentation?application_version=5.3)