

# 0 구역 (ZeroSector)

📅 develop period	@2025년 3월 11일 → 2025년 6월 1일
⋮ roll	<span>기획</span> <span>팀장</span> <span>프로그래밍</span> <span>프로젝트 매니저</span>
⋮ category	<span>언리얼 게임</span> <span>팀프로젝트</span>
⌵ size	<span>Large</span>
⋮ FrameWork	<span>Unreal</span>
⋮ 언어	<span>C++</span>

## 개요

게임 설명

스토리

게임 목표

기본 조작

## 개발

본인파트

플레이어

전투 시스템

## 개요



- 전남대학교 4학년 1학기 캡스톤디자인 프로젝트를 통해 만든 게임
- 현재 개발 진행 중인 프로젝트라 설명이 미흡한 점 정말 죄송합니다. 아래 깃허브에 올라가 있는 이슈 및 코드를 봐주시면 감사하겠습니다.

전체 코드는 다음 깃에 올라가 있습니다.

<https://github.com/SingABrro/ZeroSectorProject>

## 게임 설명

게임의 스테이지는 크게 낮과 밤 두가지로 나뉘어져 있습니다.

낮에는 추리를 하는 맵으로 단서를 찾고 NPC들과 대화를 통해 진짜 치료제를 개발할 수 있는 연구원을 찾아갑니다.

밤에는 좀비와 전투를 하는 맵으로 주무기를 선택후 전투 스테이지로 이동하여 밀려오는 좀비 웨이브를 견뎌내고 모든 좀비를 섬멸해야 합니다.

## 스토리

생화학 무기로 활용하기 위해 좀비 바이러스를 연구하던 어느 군 소속 연구소. 바이러스에 '숙주의 경질화' 라는 치명적인 결함이 발견되어 프로젝트 존폐가 논의되던 상황.갑작스레 발현된 실험체들의 공격성에 연구소 구역이 습격을 받고, 확산 방지를 위해 각 섹터를 격리하게 된다.

연구소장은 상부에 지원을 요청하나 상부는 이를 빌미로 연구소를 통째로 은폐하기로 결정하고, 연구소의 폭격을 지시한다. 연구원들의 숙소가 있는 A구역에서 불침번을 서던 주인공(플레이어)은 A구역까지 넘어온 좀비를 마주치게 된다.좀비를 처치하고 나니 소란에 잠에서 깬 연구원S에게 설명을 듣는다. 치료제 제조법을 연구한 연구원을 찾기 시작한다.

## 게임 목표

- 치료제 제작법을 알고 있는 연구원을 찾아서 탈출하기
  - 낮: 연구원 추리 + 무기 강화
  - 밤: 시나리오 진행도에 따른 구역 점령(좀비 처치)

## 기본 조작

기본 조작은 다음과 같습니다.

- 키보드

W, A, S, D	상하좌우 방향 이동
WA, WD, SA, SD	대각선 방향 이동
Q	무기 교체
R	장전
F	상호작용
Tab	개인 수첩
Esc	메뉴 선택 창

- 마우스 (밤 한정)

좌 클릭	탄 격발
우 클릭	정조준
마우스 휠	무기 교체

## 개발

git lfs를 사용해 작업물을 공유

## 본인파트

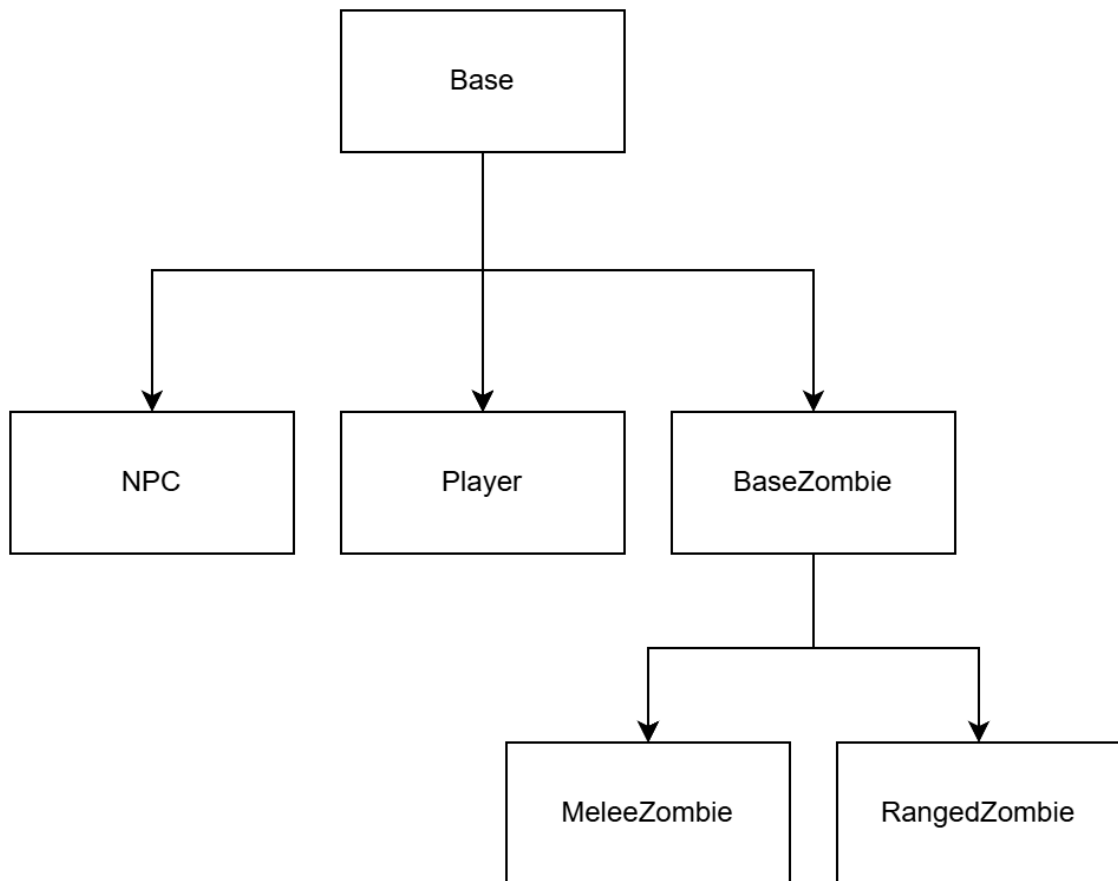
메인 메뉴, 개인 수첩 UI를 제외한 모든 파트

git branch 전략을 찾고 적용

- 플레이어
  - 기본 조작
  - 상호 작용
- 전투 시스템
  - 무기 개발 ( 권총, 라이플, 샷건 )
  - 무기 강화
- 게임 시스템
  - 낮과 밤
  - 게임 규칙
- 대화 시스템
  - NPC와 플레이어 간 대화
- 스텟 시스템
- 상호작용 시스템

- NPC 및 상호작용 가능한 액터 인식
- UI
  - HUD
  - 무기 강화 UI
  - Fade In & Out
- 몬스터
  - 좀비 AI
  - 좀비 공격
- 애니메이션
  - 캐릭터 애니메이션 블루프린트
  - 몬스터 애니메이션 블루프린트
  - 애니메이션 리타게팅
- Git Branch 전략
  - Trunk-Based

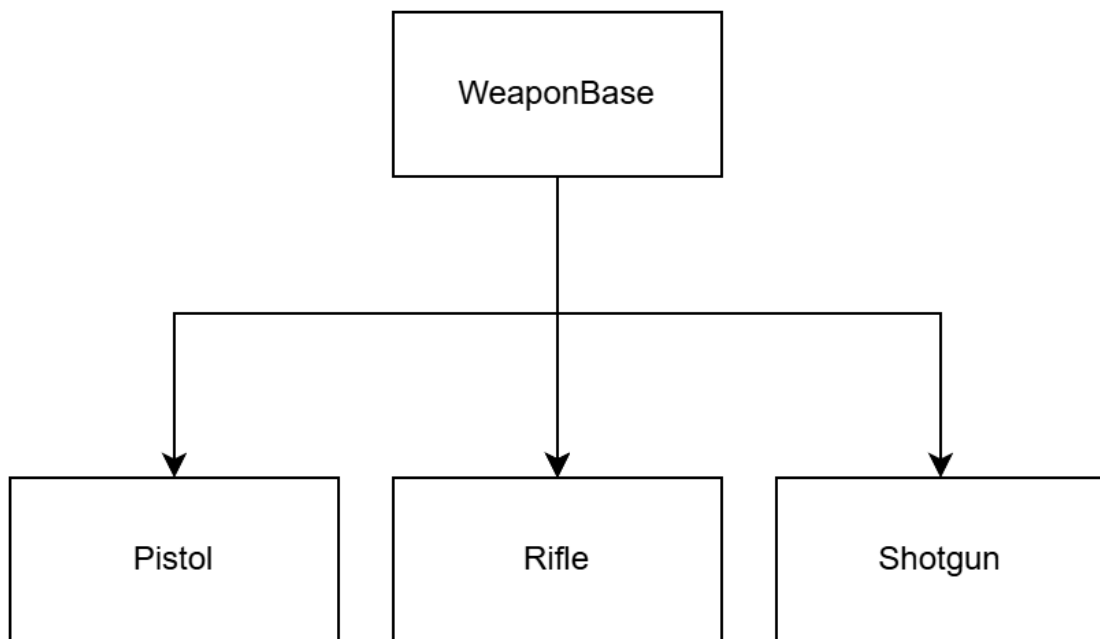
## 플레이어



- Character 클래스 상속 구조
- Base 클래스
  - IGenericTeamAgentInterface 를 구현해 NPC, Zombie, Player 팀을 구별
  - IZeroClassIdentifierInterface 를 구현해 각 객체마다 이름을 부여한 후 Map 컨테이너의 Key 값으로 이름 설정, Value 값으로 객체에 필요한 데이터 설정
- Player 클래스
  - Player 클래스 자체는 기능이 없으며, 각 기능별 컴포넌트를 구현해 Player 클래스에 부착함으로써 프로젝트 확장성, 디커플링을 확보
  - 낮과 밤에 사용하는 조작법이 다르기 때문에 낮 전용, 밤 전용 InputComponent를 생성
    - 이렇게 함으로써 낮에 필요한 데이터는 낮에만 메모리에 올라가 있으며, 밤에 필요한 데이터 또한 밤에만 메모리에 올라가게 할 수 있었음

- Player가 상호작용을 통해 생성되는 UI Widget 을 관리하는 UIComponent 생성
- Player의 스탯을 관리하는 StatComponent 생성
- 각 컴포넌트들은 델리게이트를 만들어 Player 클래스에서 델리게이트에 필요한 함수를 바인딩하여 통신

## 전투 시스템



- Weapon 클래스 상속 구조
- 샌드박스 패턴을 사용하여 구현
- 원래는 상위 클래스에서 protected 이상의 접근 지정을 하여 하위 클래스에 기능을 제공해야 하지만, 언리얼은 리플리케이션 기능을 통해 상위 클래스의 함수를 같이 호출할 수 있기 때문에, 상위 클래스에서 제공해야 하는 기능을 private으로 캡슐화 한 뒤 하위 클래스에서 재정의 할 가상 함수에서 호출함으로써 구현

```

void AZeroWeaponBase::Fire()
{
    if (bIsFire || CurrentAmmo <= 0) return;

```

```

    blsFire = true;

    switch (WeaponType)
    {
    case EWeaponType::EPistol:
        PistolFire();
        break;
    case EWeaponType::ERifle:
        RifleFire();
        break;
    case EWeaponType::EShotgun:
        ShotgunFire();
        break;
    default:
        return;
    }
}

```

- WeaponType을 통해 무기 종류 구별
- Fire() 가상 함수에서 현재 무기 종류에 따라 다른 공격을 호출함
- 각 무기의 스텟은 무기 액터가 생성된 뒤, 현재 무기 종류에 따라 DataTable에서 스텟을 가져와 적용
- 총기 반동, 탄창, 무기 대미지 적용 등 구현 완료