

## 영상 링크

<https://www.youtube.com/watch?v=fVY9flridMo>

## 소스 코드 링크

[https://github.com/bossmaru/KimAhYoung\\_Portfolio/tree/49a10bd2c2df0a997dc14c5bcd660916f2de8133/00\\_Projects/00\\_Unreal5](https://github.com/bossmaru/KimAhYoung_Portfolio/tree/49a10bd2c2df0a997dc14c5bcd660916f2de8133/00_Projects/00_Unreal5)

## 제작 기간 및 팀 구성

- 24.08.12 - 24.08.30 (약 18일)
- 팀원 2명

## 역할 분담

### □ 김아영 (본인)

- MyGameInstance, MyPlayerManager, MyUIManager 구현
- PlayerSelectionUI 구현
- Knight, Archer 구현 및 관리
  - Knight: 근거리 공격 / Archer: 원거리 공격
- MyPlayer 이동, 공격, 인벤토리 관리 등의 입력 액션 처리
- Creature 애니메이션 및 공격 함수 구현
- Boss Monster 어그로 시스템, 어그로 UI, AI, 애니메이션 구현
- Monster 사망 시 아이템 드랍 기능 구현
- NPC 애니메이션 및 상호작용(충돌) 구현
- VFX 적용, 최종 맵 배치

### □ 팀원 1

- 기본 몬스터 AI 구현
- 인벤토리 UI, 상점 UI 및 골드 UI 구현
- Item 관련 함수 구현
- 보스 몬스터 AI 기능 개선 작업
- HpBar, 상점 UI, 골드 UI 구현
- 보스 몬스터 AI를 보완하고 개선
- 최종 영상 제작

## 주요 기능

### □ Player Selection



### □ Inventory



## □ Inventory



## □ Item Drop / Use





□ NPC / Store



□ HP Bar





## □ Reward System



## □ Get Gold Reward









□ Aggro System



□ Aggro System



---

## 주요 컴포넌트 설명

### □ AI 시스템

- **AMyAIController**: Behavior Tree를 사용하여 AI의 행동을 제어.
- **BTask\_SetRandomPos, BTask\_SetTargetPos**: AI가 랜덤 위치를 설정하거나 타겟 위치를 설정하는 태스크를 수행.
- **BService\_FindTarget, BService\_PrimeSetting**: 타겟을 탐색하거나 프라임 위치를 설정하는 서비스.
- **BTDeco\_CanAttack, BTDeco\_IsBossDead, BTDeco\_IsDamagedByBoss**: AI 행동에 대한 다양한 조건을 설정하는 데코레이터.

### □ 보스 몬스터 시스템

- **BossMonster**: 보스 몬스터의 행동을 정의한 클래스. 보스는 고유의 공격 패턴과 AI를 가짐.
- **BService\_FindBoss, BTask\_AggroAttack**: 보스 몬스터의 어그로 시스템이 특정 대상에게 집중 공격을 수행하도록 함.
- **BTDeco\_IsBossDead**: 보스 몬스터의 상태를 확인하고 죽음 여부를 결정하는 데코레이터.

### □ 캐릭터 클래스

- **AMyPlayer, AKnight, AArcher, AMyMonster, AMyNPC**: 다양한 캐릭터 클래스가 있으며, 각 캐릭터는 고유한 공격 방식과 애니메이션을 가지고 있음.
- **전투 시스템**: AttackHit 함수와 애니메이션 몽타주를 사용하여 공격을 처리.
- **UMyAnimInstance**: 캐릭터 애니메이션을 관리하고, 공격 및 데미지 처리 시 몽타주를 재생.

### □ 인벤토리 시스템

- **UMyInventoryUI**: 플레이어가 수집한 아이템을 관리할 수 있는 인벤토리 UI. 아이템을 추가하거나 삭제 가능
  - **UMyInventoryComponent**: 인벤토리에 아이템을 추가하거나 드롭할 수 있는 기능을 제공.
  - **AMyItem**: 게임 내 아이템을 관리하는 클래스.
-



---

#### □ 상점 시스템

- **UMyStoreUI**: 플레이어가 NPC와 상호작용하여 아이템을 거래할 수 있는 상점 UI.
- **AMyNPC**: 상점과 같은 상호작용을 제공하는 NPC 클래스.

#### □ UI 관리 시스템

- **AMyUIManager**: 인벤토리, 상점, 플레이어 선택 등 다양한 UI를 관리하고 토글할 수 있는 기능을 제공.
- **UPlayerSelectionUI**: 게임 시작 시 플레이어가 선택할 캐릭터를 선택하는 UI.
- **MyHpBar, UI\_AggroInfo, UI\_StatUp**: 게임 진행 중 캐릭터의 스탯과 어그로 정보를 표시하는 UI.

#### □ 스탯 관리 시스템

- **UMyStatComponent**: 각 캐릭터의 스탯(체력, 공격력 등)을 관리합니다. 스탯 변경 시 UI 업데이트 및 이벤트를 처리.

#### □ 플레이어 관리 시스템

- **AMyPlayerManager**: 게임 내에서 플레이어의 타입(Knight 또는 Archer)을 설정하고, 선택된 플레이어에 따라 캐릭터를 관리.

#### □ VFX 및 사운드 시스템

- **VFX\_Manager**: 다양한 시각 효과를 관리하며, 게임 내 이벤트에 따라 VFX를 출력.
  - **SoundManager**: 사운드를 관리하며, 게임 내 효과음 및 배경음을 처리.
-