

# 포트폴리오

박철우

git: [https://github.com/iristopvt/TeamProject\\_PCW](https://github.com/iristopvt/TeamProject_PCW)





# 팀프로젝트

총 3명

프로젝트시작 2024 .08.12

1

2

프로젝트종료 2024.08.30

# 프로젝트 기여 내용

1. 입력 설정과 콜리전 시스템
2. 캐릭터(기사)
3. 아이템
4. UI시스템

# 1. 입력 설정과 콜리전 시스템

Object Channels

오브젝트 & 트레이스 채널을 포함해서 커스텀 채널은 18 개까지 가능합니다. 프로젝트의 오브젝트 타입 목록입니다. 게임에 사용되는 오브젝트 타입을 삭제하는 경우, 해당 타입의 모든 사용이 '월드 스텝'에서 삭제됩니다.

이름

기본 반응

Projectile

블록

Monster

블록

MyCharacter

블록

MyItem

무시

NPC

블록

Trace Channels

오브젝트 & 트레이스 채널을 포함해서 커스텀 채널은 18 개까지 가능합니다. 프로젝트의 오브젝트 타입 목록입니다. 게임에서 사용되는 트레이스 채널을 삭제하는 경우, 트레이스의 작동방식은 정의된 대로 작동합니다.

이름

기본 반응

Attack

무시

프리셋

프로젝트 프로파일은 어느 것이든 변경 가능합니다. 참고로 프로파일을 변경하면, 콜리전 작동방식이 변경될 수 있습니다. 현재 존재하는 (사용된) 콜리전 프로파일을 변경할 때는 주의하세요.

이름

콜리전

오브젝트 타입

NoCollision

콜리전 없음

WorldStatic

BlockAll

콜리전 켜짐 (쿼리 및 피직스)

WorldStatic

OverlapAll

쿼리만 (피직스 콜리전 없음)

WorldStatic

BlockAllDynamic

콜리전 켜짐 (쿼리 및 피직스)

WorldDynamic

OverlapAllDynamic

쿼리만 (피직스 콜리전 없음)

WorldDynamic

IgnoreOnlyPawn

쿼리만 (피직스 콜리전 없음)

WorldDynamic

OverlapOnlyPawn

쿼리만 (피직스 콜리전 없음)

WorldDynamic

Pawn

콜리전 켜짐 (쿼리 및 피직스)

Pawn

Spectator

쿼리만 (피직스 콜리전 없음)

Pawn

CharacterMesh

쿼리만 (피직스 콜리전 없음)

Pawn

PhysicsActor

콜리전 켜짐 (쿼리 및 피직스)

PhysicsBody

Destructible

콜리전 켜짐 (쿼리 및 피직스)

Destructible

InvisibleWall

콜리전 켜짐 (쿼리 및 피직스)

WorldStatic

InvisibleWallDynamic

콜리전 켜짐 (쿼리 및 피직스)

WorldDynamic

Trigger

쿼리만 (피직스 콜리전 없음)

WorldDynamic

Ragdoll

콜리전 켜짐 (쿼리 및 피직스)

PhysicsBody

Vehicle

콜리전 켜짐 (쿼리 및 피직스)

Vehicle

UI

쿼리만 (피직스 콜리전 없음)

WorldDynamic

Arrow

콜리전 켜짐 (쿼리 및 피직스)

Projectile

Monster

콜리전 켜짐 (쿼리 및 피직스)

Monster

MyCharacter

콜리전 켜짐 (쿼리 및 피직스)

MyCharacter

MyItem

쿼리만 (피직스 콜리전 없음)

MyItem

</

<콜리전>

매핑		
매핑 + 삭제		
▼	Move_IA	↶ ↷ + 삭제
▶	키보드 W	▼ 삭제
▶	키보드 S	▼ 삭제
▶	키보드 A	▼ 삭제
▶	키보드 D	▼ 삭제
▼	Look_IA	↶ ↷ + 삭제
▶	마우스 XY 2D 축	▼ 삭제
▼	Jump_IA	↶ ↷ + 삭제
▶	키보드 스페이스 바	▼ 삭제
▼	Attck_IA	↶ ↷ + 삭제
▶	마우스 왼쪽 마우스 버튼	▼ 삭제
▼	Drop_IA	↶ ↷ + 삭제
▶	키보드 G	▼ 삭제
▼	Inven_IA	↶ ↷ + 삭제
▶	키보드 I	▼ 삭제
▼	Mouse_IA	↶ ↷ + 삭제
▶	키보드 왼쪽 Ctrl	▼ 삭제
▼	Interaction_IA	↶ ↷ + 삭제
▶	키보드 F	▼ 삭제

<입력>

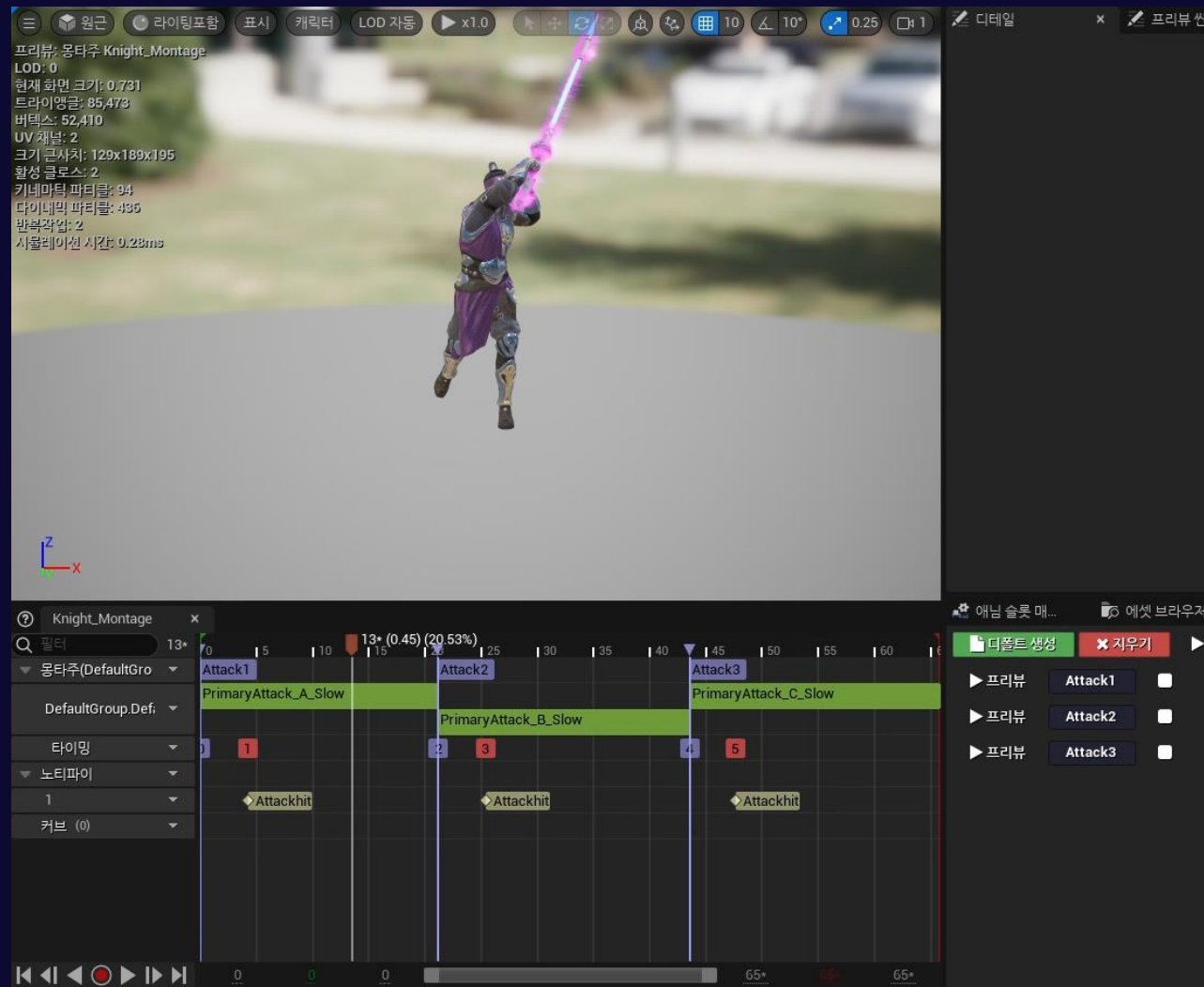
<상세내용>

-입력  
Attack의 경우 누를때마다  
콤보공격이 차례대로 나가게  
구현하였으며,

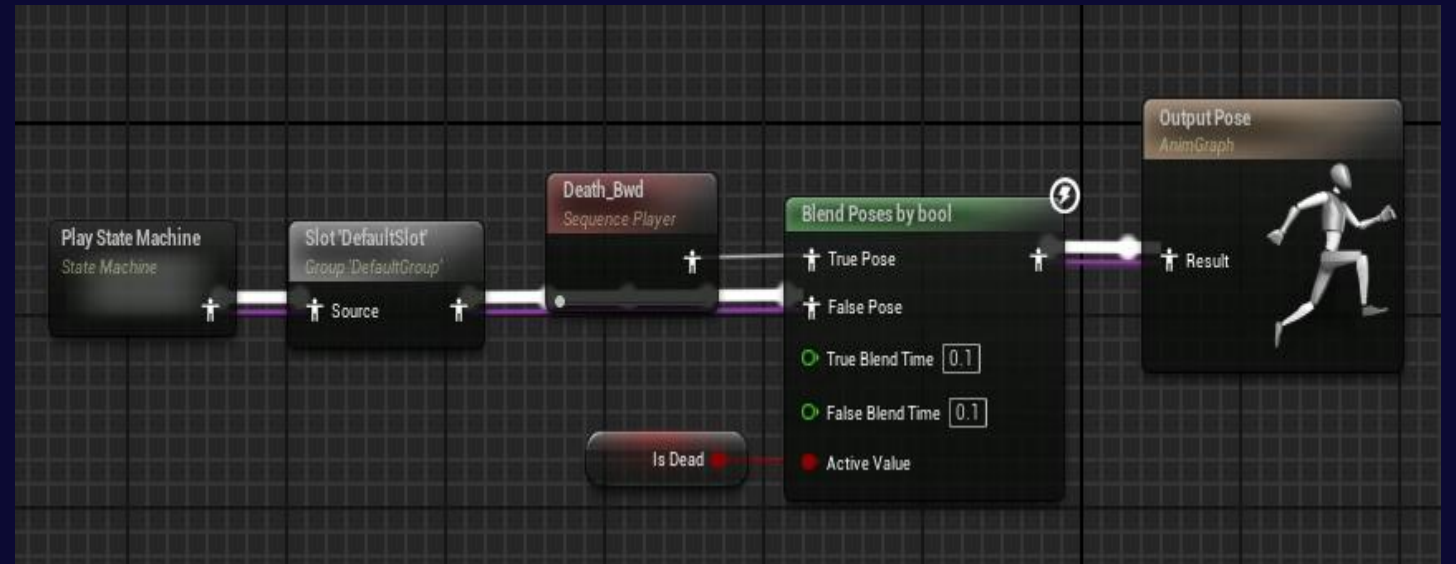
Mouse와 Inven의 경우  
눌렀을때 활성화되고  
다시한번 누를때  
비활성화되게 처리하였습니다

Interaction는 NPC에게 다가가  
해당버튼을 눌렀을때 상호작용을  
시켜주며 현재 구현되어있는  
StoreNPC에게 다가가 해당 키를  
눌렀을때 StoreUI를 화면에  
출력되게 만들었습니다.

## 2. 캐릭터(기사)



<몽타주>



<애니메이션 블루프린트>

### <상세내용>

-언리얼엔진의 애니메이션몽타주를 이용해  
기사의 콤보공격 을 만들었으며 노티파이를 설정해 공격을 실행하면  
차례대로 공격몽타주가 실행되게 설정하였으며,  
이후 기사의 애니메이션을 만들었습니다.



### 3. 아이템

```
USTRUCT(BlueprintType)
struct FItemData : public FTableRowBase
{
    GENERATED_BODY()

    UPROPERTY(EditAnywhere)
    FString ItemName;

    UPROPERTY(EditAnywhere)
    int32 Price;

    UPROPERTY(EditAnywhere)
    int32 Atk;

    UPROPERTY(EditAnywhere)
    UTexture2D* ItemTexture;

    UPROPERTY(EditAnywhere)
    USkeletalMesh* ItemMesh;

    UPROPERTY(EditAnywhere)
    int32 ItemId;

    FItemData()
    : ItemName(TEXT("Default"))
    , Price(0)
    , Atk(0)
    , ItemTexture(nullptr)
    , ItemMesh(nullptr)
    , ItemId(-1)
    {}
};
```



<ItemData>

1	HP Potion	HP Potion	3	0	/Script/Engine.Texture2D'/Game/Graphic/icon/Tex_reagent_12_b.Tex_reagent_12_b	/Script/Engine.SkeletalMesh'/Game/Kobo_Dungeon/Meshes/SM-Potion-03	1
2	Sword	Sword	5	50	/Script/Engine.Texture2D'/Game/Graphic/icon/Tex_tools_04_b.Tex_tools_04_b	/Script/Engine.SkeletalMesh'/Game/SKnight_modular/Skeleton_Knight_02	2
3	Armor	Armor	10	10	/Script/Engine.Texture2D'/Game/Graphic/icon/Tex_tools_12_b.Tex_tools_12_b	/Script/Engine.SkeletalMesh'/Game/SKnight_modular/Skeleton_Knight_03	3

열 에디터

HP Potion

HP Potion

Item Name	HP Potion
Price	3
Atk	0
Item Texture	 Tex_reagent_12_b
Item Mesh	 SM-Potion-03
Item Id	1

<ItemDataTable>

#### <상세내용>

-아이템의 정보를 구조체로 정의하고  
구조체를 기반으로 데이터 테이블을  
생성했습니다

데이터 테이블을 이용하여  
아이템 정보를 불러오고 이를  
UI에 반영하도록 구현했습니다

### 3. UI 시스템

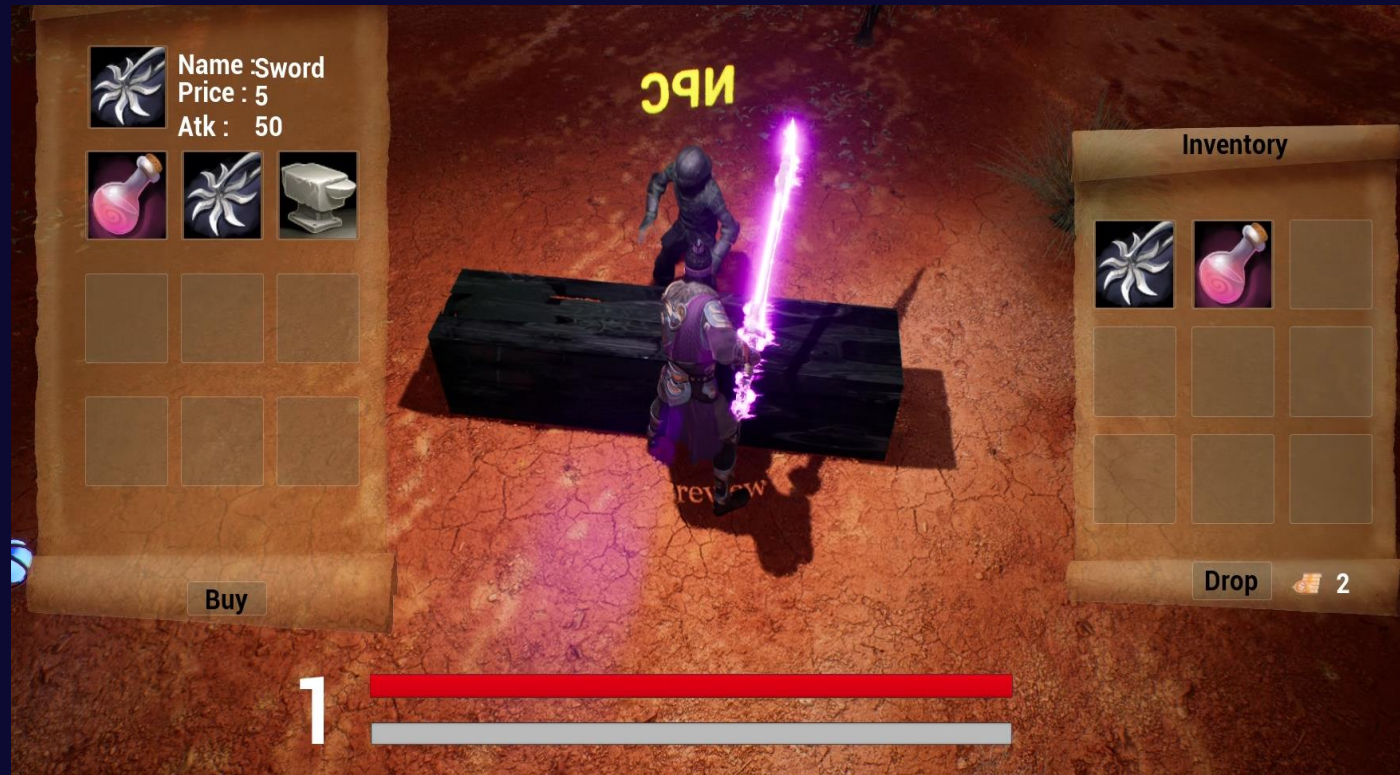


<UI>



<Inventory UI>





<Store Buy>

## <상세내용>

-HPBar와 ExpBar를 StatDataTable과 연결하여,  
해당 레벨에 맞는 HP와 EXP를 자동으로 설정할 수 있도록 구현했습니다.

-inventory  
-필드에 있는 아이템과 충돌 시,  
해당 아이템이 인벤토리에 추가되도록 구현하였으며,  
아이템의 이미지는 ItemDataTable에서 가져와 인벤토리 UI에 표시하도록 하였습니다.

-Store  
-Store에서 아이템을 클릭하면 클릭한 아이템의 정보를 확인할수 있게 하였으며,  
이후 Buy 버튼을 클릭하면 해당 아이템의 정보를 가져와 플레이어의 인벤토리에  
추가되도록 하였습니다.



