강건

UE5_Inventory 포트폴리오

목차

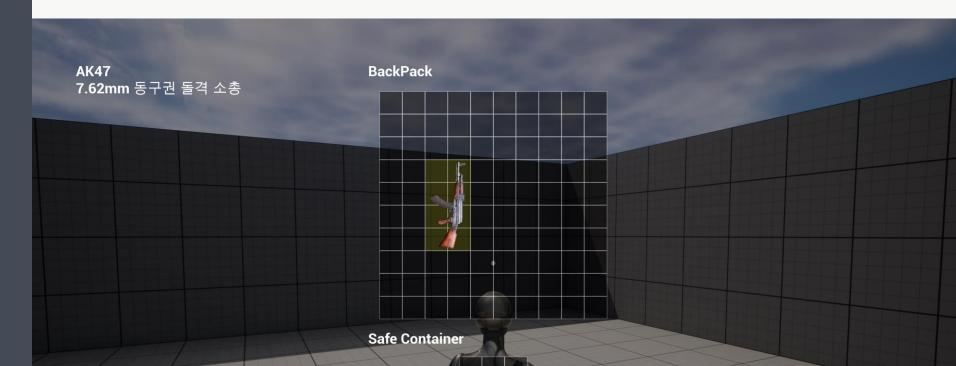
● 개요

링크

● 프로그래밍 구조

개요

- ◆ Unreal Engine : 5.0.3
- ◆ 타르코프와 같은 생존게임을 즐겨서 유사하게 제작하였습니다.
- ◆ 주요 업무에 해당하는 인벤토리 제작을 도전했습니다.
- ◆ '획득 버리기 이동 아이템 회전' 기능을 중점으로 구현하였습니다.



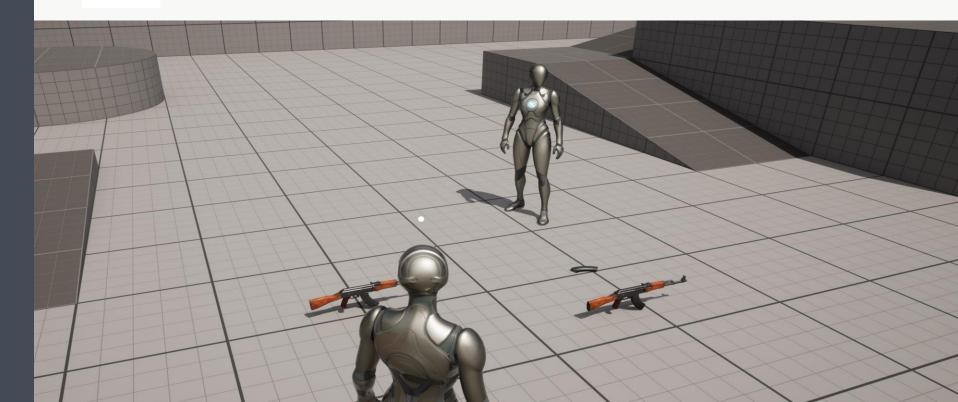




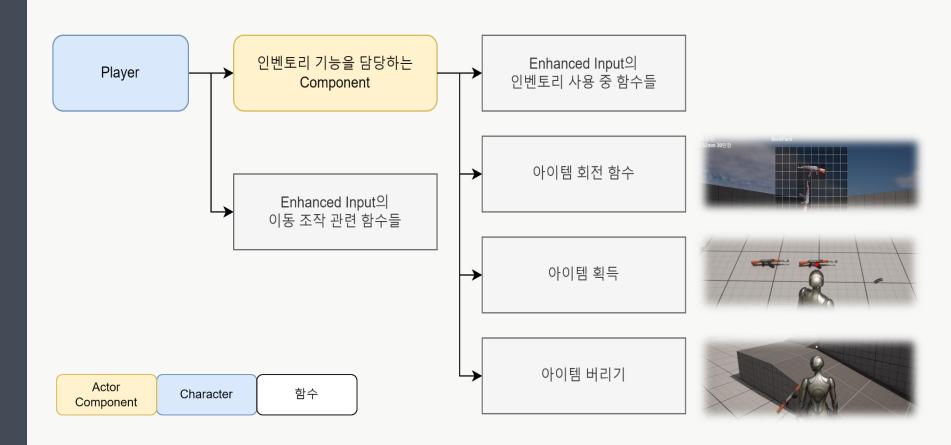
https://github.com/KitchenGun/UE5_Inventory



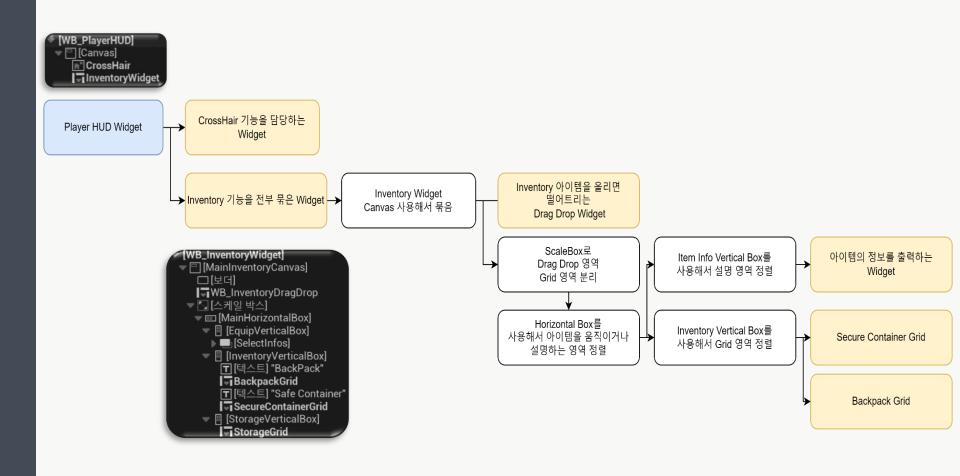
https://youtu.be/qXISRp6Ap7I



Player_Character(Actor Component)



Player_ Widget(Widget Blueprint 계층 구조)



Blueprint

Widget

Root

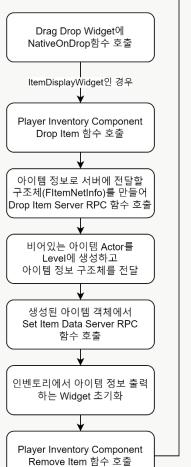
Widget

Item Add • Drop

아이템 획득 Flow Chart

Player에서 Interact 함수 아이템을 가리키는 경우 Player Inventory Component PickUp 함수 호출 Grid 우선순위에 맞춰서 빈공간 확인 Grid에 공간이 있는 경우 인벤토리에 위치한 동일 아이템 Add Item 호출 AddQuantity 호출하여 여러 개 보관 가능한 −보관 가능▶ 아이템 여부 확인 (EX: 총알) 수량 증가 보관 불가능 아이템을 보여줄 Item Display Widget 생성 아이템 위치와 정보를 InventoryComponent에 저장 Level에 배치된 아이템 삭제

아이템 버리기 Flow Chart







아이템 데이터 저장 방식

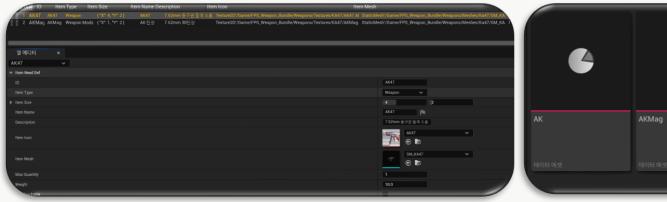
```
//그리드 저장 정보

TMap<EItemStorageType, TArray<UInventoryItemDisplay*>> ItemContainer; //ItemDisplay들 저장

TMap<EItemStorageType, TArray<UInventorySlot*>> ItemSlots; //각그리드의 Slot(WBP)의 정보들 저장

TMap<EItemStorageType, TMap<UInventorySlot*, bool>> ItemSlotMap; //각그리드의 Slot 공간 비었는지 확인용
```

◆ Map 형태의 자료구조를 사용하여서 아이템을 저장하도록 제작 시각적으로 보여주는 Widget을 따로 저장하고 아이템이 차지하는 슬롯에 대한 정보를 추가로 따로 처리했습니다.

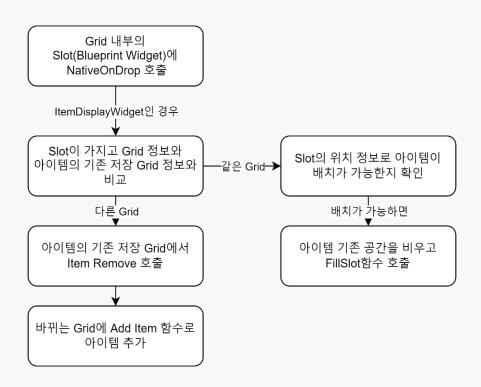


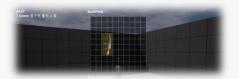


◆ 아이템 정보 저장 Data Table과 Data Asset을 활용하여서 제작 Data Table을 사용해서 Game Instance 객체에서 ID를 통해 검색해서 아이템 정보를 받아오고 Data Asset을 사용해서 빠르게 아이템을 생성할 수 있도록 제작하였습니다.

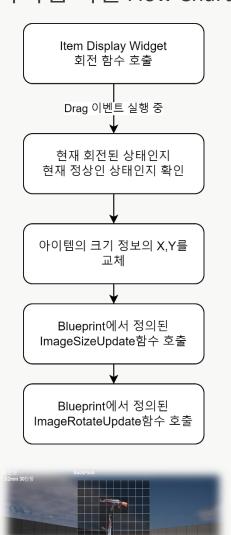
Item Move • Rotate

아이템 이동 Flow Chart

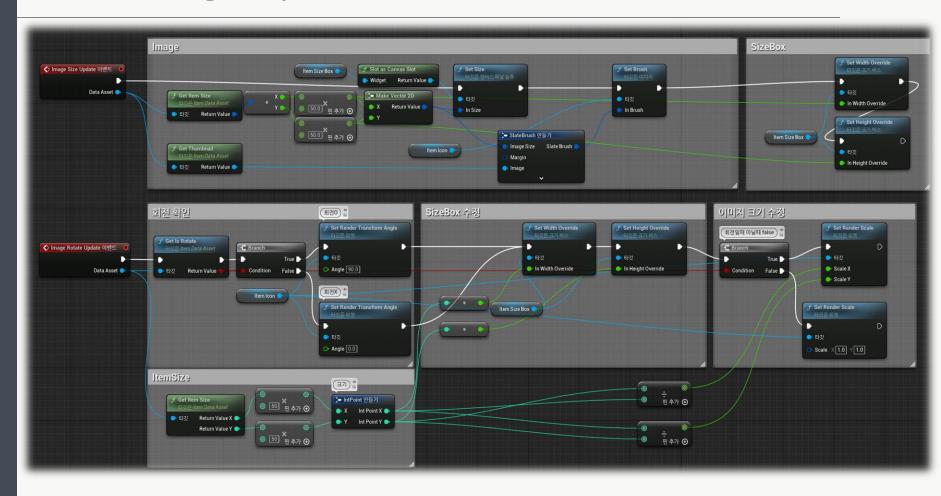




아이템 회전 Flow Chart



Item Image Update



- ◆ 아이템을 획득하면 C++에서 Image Size Update 이벤트를 호출하여 이미지를 Grid 크기에 맞게 수정합니다.
- ◆ 이미지가 회전할 경우 Image Rotate Update를 호출하여서 변화된 Size Box에 맞게 이미지의 비율을 수정합니다.