

[몬스터 만들기 & 배치 & 드랍]

- 몬스터 만들기

1. 몬스터 정하기

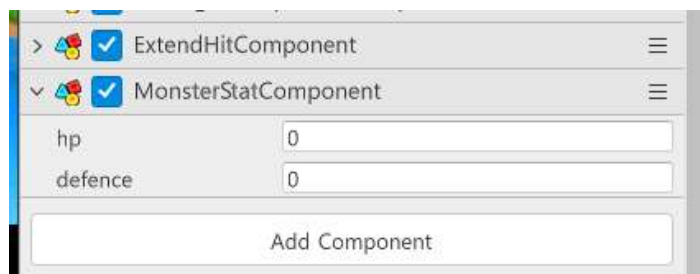
=> 예시로는 초록버섯 monster 879



1-1 가져온 몬스터가 “MoveMonster” 인지 확인 (StaticMonster, ChaseMonster 는 사용 불가능)



1-2 ExtendHitComponent와 MonsterStatComponent가 들어있는지 확인 (AIWanderComponent도 확인)

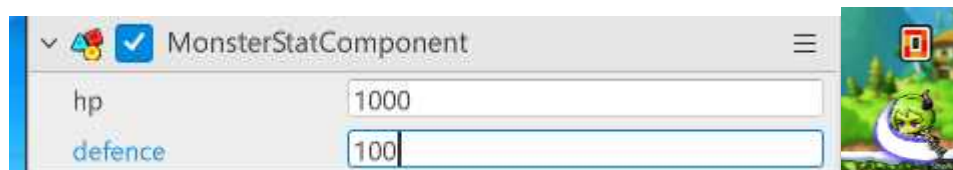


2. 체력 & 방어력 설정 (현재 플레이어의 공격력은 500-> 추후 구현 예정)

=> 예시로는 체력 1000 방어력 100으로 설정 (공격 공식은 체력-(플레이어 공격력-방어력))

=> 3대 맞으면 죽은 초록버섯 완성

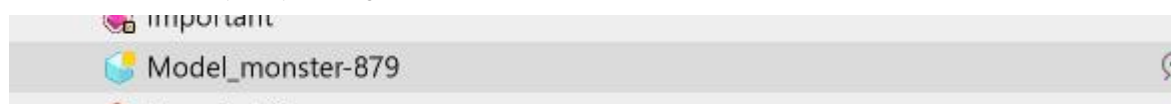
=> 만약 공격력보다 방어력이 높으면 0 return



3. 방금 만든 개체를 Create Model From Entity로 Workspace에 생성



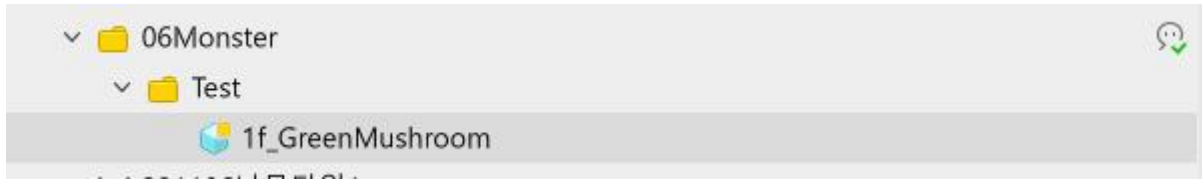
Model_monster-(번호)로 저장됨



저장된 Model을 (사냥터이름_몬스터이름)으로 06Monster에다가 옮김

=> 예) 1f_GreenMushroom : 1층 초록버섯

=> 사막맵의 빨간슬라임 (폴더) Desert 밑의 Desert_Slime으로 저장(통일성을 위해)



4. 여기까지 한 후, 모델을 씬 창에 넣고(1f_GreenMushroom으로 저장한 것) 공격이 들어가는지, 죽으면 사라지는 지 확인



=> 몬스터 만들기 끝

- 몬스터 배치하기

1. 씬창에서 tab을 눌러 몬스터를 소환하고자 하는 곳의 좌표 확인

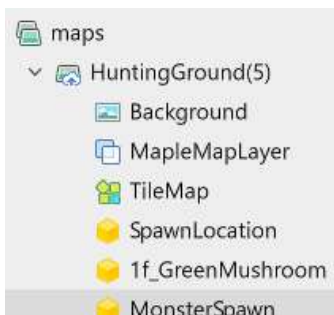


=> 약 (-5,0.2)부터 (6.8,0.2) 사이에 몬스터를 소환할 예정

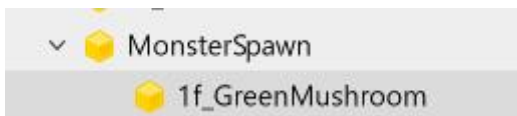
=> Y값은 동일해야함

2 스포너 만들기

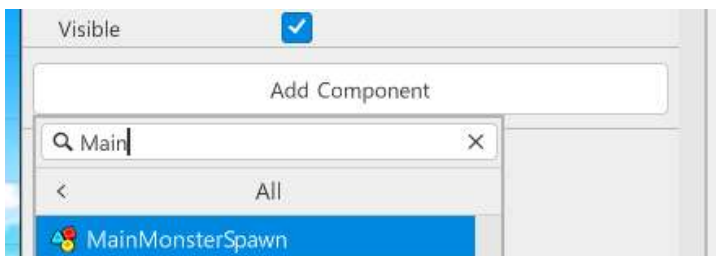
maps/(만들고 있는 맵) 아래에 Create Empty로 MonsterSpawn 생성 (이름 고정, 만약에 벌써 만들어져 있다면 그거 사용)

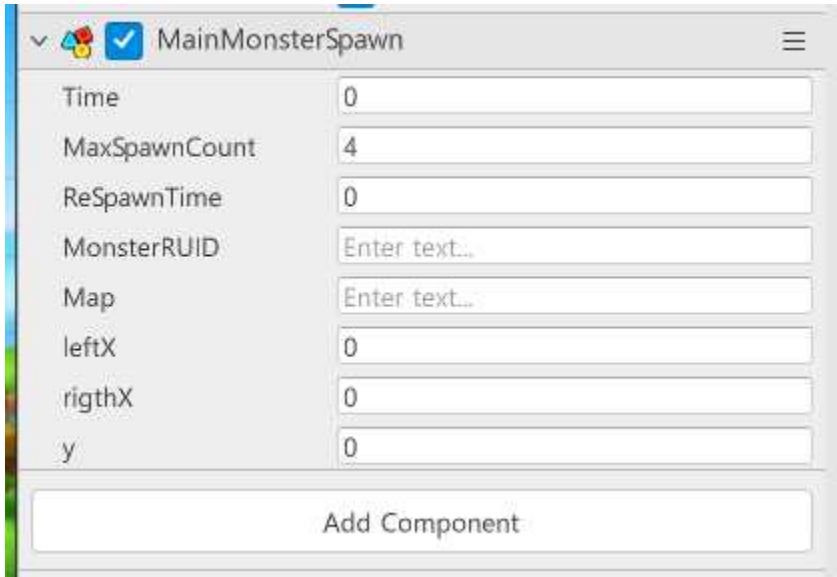


만든 MonsterSpawn 밑에 (Model명과 똑같은 빈 게임 오브젝트 생성) 다시 빈게임 오브젝트 생성



빈 게임 오브젝트에 (1f_GreenMushroom) MainMonsterSpawn 컴포넌트 추가





컴포넌트 설명

Time : [건들 필요 없음]

MaxSpawnCount : 해당 지역 최대로 스폰할 몬스터의 수 (기본값 : 4)

ReSpawnTime : 해당 지역에 몬스터가 죽었을 때 다시 리스폰될 시간

MonsterRUID : 해당 지역에 나올 몬스터

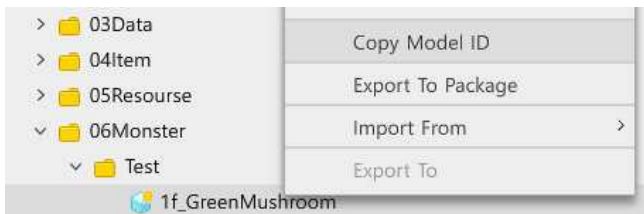
Map : 소환할 맵 경로

leftX : 스폰 지역 왼쪽 X 값

righthX : 스폰 지역 오른쪽 X 값

y : 스폰 지역 Y값

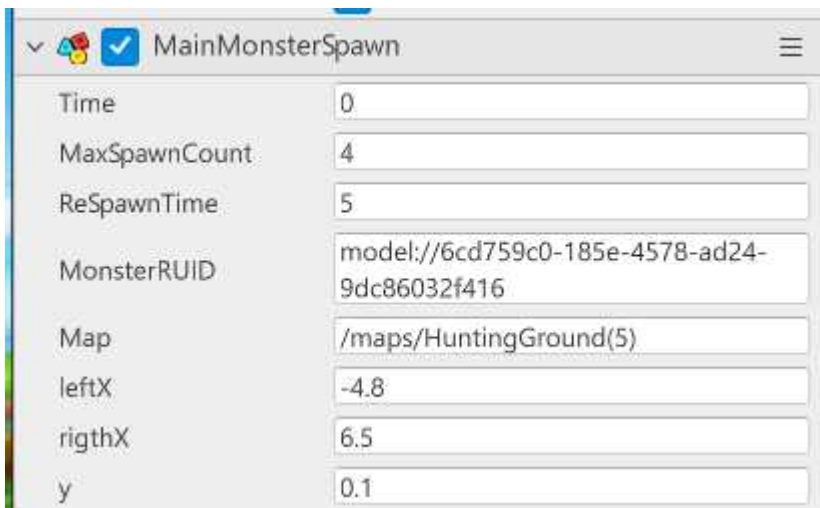
MonsterRUID 가져오는 법 : 스폰할 Modle 우클릭 Copy Model ID 후 붙여넣기



Map 경로 설정하는 법 : 맵 앞에 "/" 추가 ex) /maps/HuntingGround(5) 타이핑 (철자 고수)



완성본 예시



test

=> 아무것도 없는 빈 지역에서(MonsterSpawn) 시작



=> 시작한 후, 원하는 수만큼 스폰이 되는지 확인

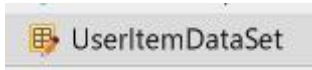


- 아이템 만들고 드랍하기

1. 개요

=> 예시로는 사과를 만들어 보자

=> 아이템은 무조건 UserItemDataSet에 저장해야 함



2. DataSet 저장(Name은 영어로 할 것)

ID	Name	IconRUID	Description	Divition	MaxStackCount	EffectAmount	Speed
24	doll	5ee3b6c1e23c4	땅에서 주운 인형	Important	1	0	0
25	Apple		Normal apple	Food	100	0	0

Name : 아이템 이름

IconRUID : 아이템 RUID (Model ID가 아닌, 이미지의 RUID를 사용해야 함)

Description : 아이템 설명(나중에 한글로 한 것 같긴 함)

Divition : 아이템 구분(Food-음식, Important-중요한 물건, sword-검 등등)

MaxStackCount : 최대 보유 개수

EffectAmount : 효과 (회복)

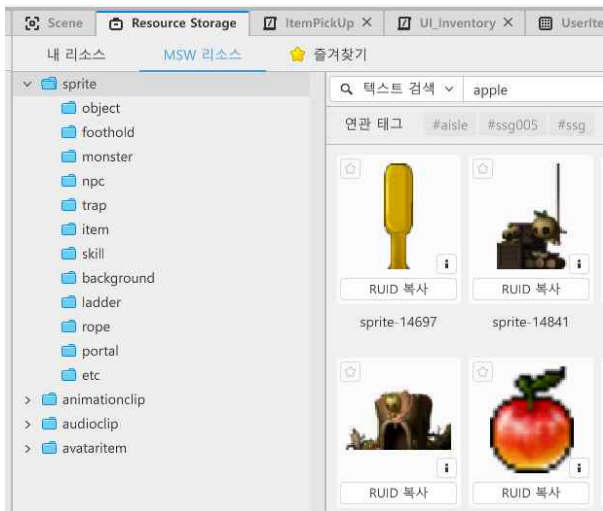
Speed : 이동 속도

...(필요할 때마다 열을 늘려서 사용)

3. 모델 만들기

=> Resource Storage - MSW리소스 - 검색 으로 원하는 아이템 찾기 또는 필요한 사진(jpg, png 등)

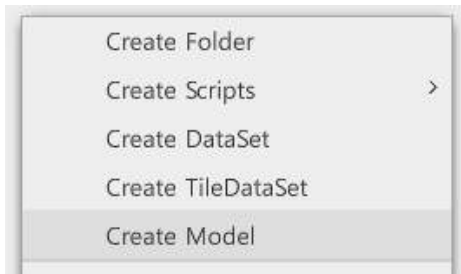
=> <https://maplestoryworlds-developers.nexon.com/ko/docs/?postId=66> (외부 사진 넣는 법)



별을 눌러 즐겨찾기를 하면 나중에 찾기 편함 & RUID 복사



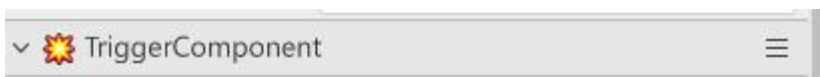
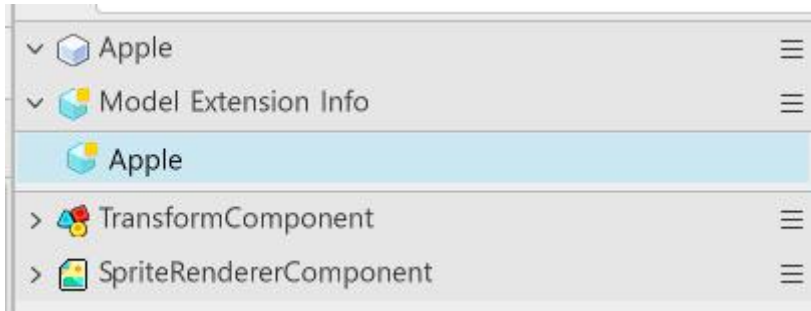
Workspace - MyDesk - 우클릭 - CreateModel 선택



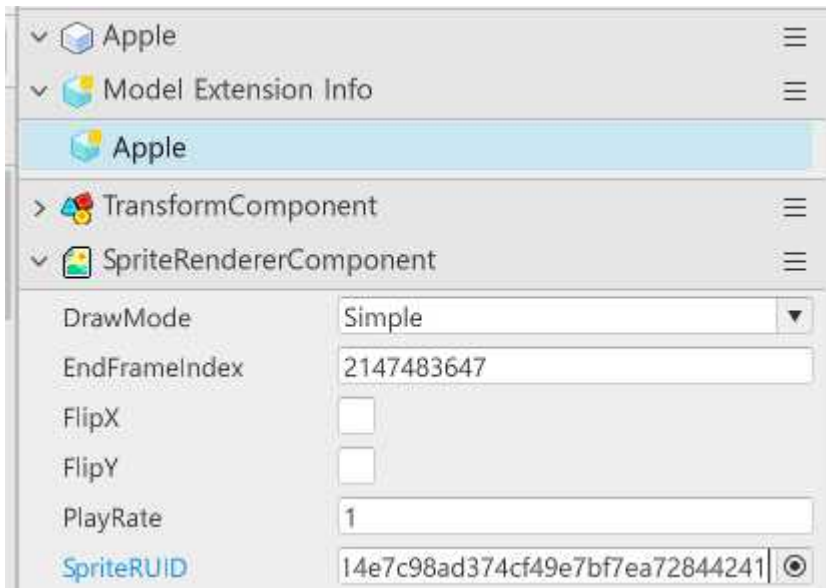
이름을 아이템 이름으로 바꾸고(DataSet의 Name과 통일)



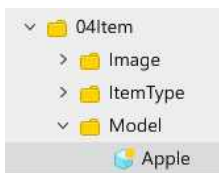
Property에 TransformComponent와 SpriteRendererComponent와 TriggerCompoenet 추가



SpriteRendererComponent의 SpriteRUID에 위에서 복사한 이미지 RUID 붙여넣기



만든 Model을 04Item - Model에 저장



DataSat의 IconRUID에 넣기

주의 : 이미지의 RUID를 넣어야 인벤토리에서 출력이 됨

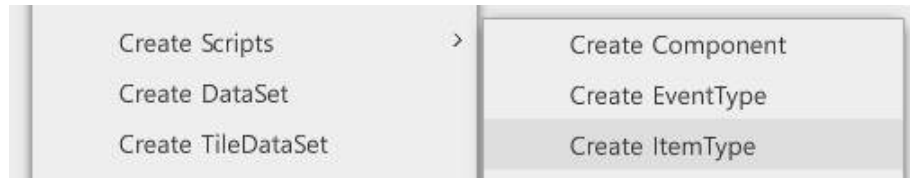
올은 예) 014e7c98ad374cf49e7bf7ea72844241

틀린 예) Model//014e7c98ad374cf49e7bf7ea72844241

25	Apple	7-311e8f259aed	Normal apple	Food	100	0	0
----	-------	----------------	--------------	------	-----	---	---

4. 아이템 분류 만들기 (분류가 생성되어 있으면 4번은 뛰어 넘어도 됨)

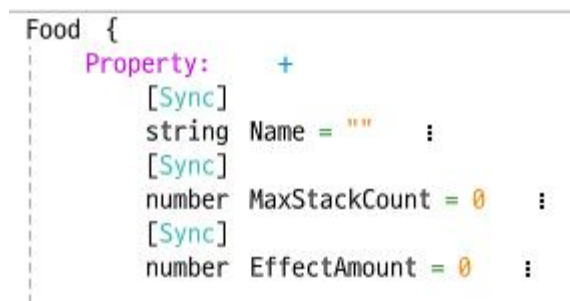
MyDesk 우클릭 Create Scripts - Create ItemType 클릭



사과이므로 Food 생성 (UserDataSet - Divistion이 Food 라면 더 이상 만들지 않아도 됨)



Food에서 Property +를 눌러 타입을 만듭



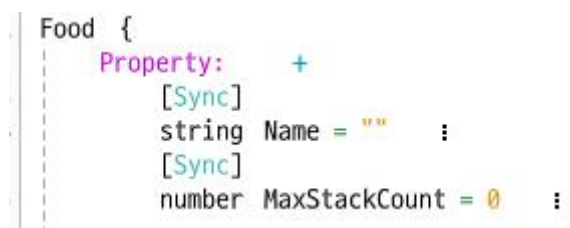
※ 여기서의 Property는 DataSet의 열 이름과 반드시 같아야 함

string Name = "" (음식의 이름이 들어갈 곳 ex) 사과)

number MaxStackCount = 0 (음식의 최대 보유 개수가 들어갈 곳 ex) 100)

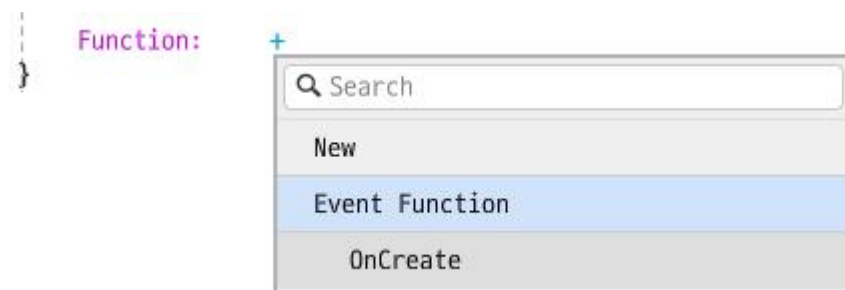
number EffectAmount = 0 (EffectAmount는 포션에서 회복 수치의 개념이므로, Food에 포함 되서는 안됨)

Food의 Property의 완성본 (참고 : string은 뒤의 값이 “문자”가 올 경우 ex) 사과, apple / number는 뒤의 값이 “숫자”가 올 경우 ex) 0, 50, 100임)



4-1 DataSet이랑 연결하기

Food의 Function에서 +를 누르고 OnCreate 선택




```

Function: +
void OnCreate ( ) :
{
    self.Name = self.ItemTableData:GetItem("Name")
    self.MaxStackCount = tonumber(self.ItemTableData:GetItem("MaxStackCount"))
}

```

self.Name = self.ItemTableData:GetItem("Name")

self.MaxStackCount = tonumber(self.ItemTableData:GetItem("MaxStackCount"))

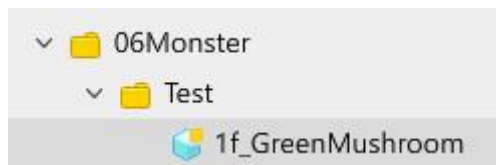
OnCreate에 붙여 넣기 (참고로 위의 두 개는 어떤 분류이건 필요하기 때문에 무조건 붙여넣음)
추가하는 값이

string 일 경우 : self.(property의 변수 값) = self.ItemTableData:GetItem("(DataSet의 변수값과 같은 값)")

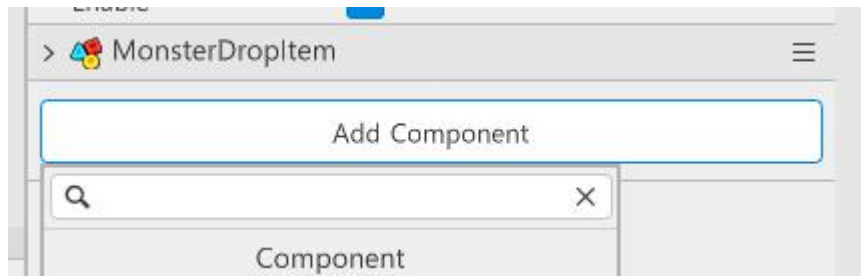
number 일 경우 : self.(property의 변수 값) = tonumber(self.ItemTableData:GetItem("(DataSet의 변수값과 같은 값)"))

5. 몬스터에서 아이템 튀어나오게 하기

※주의※ 1 에서만든, 몬스터 Model 에서 진행

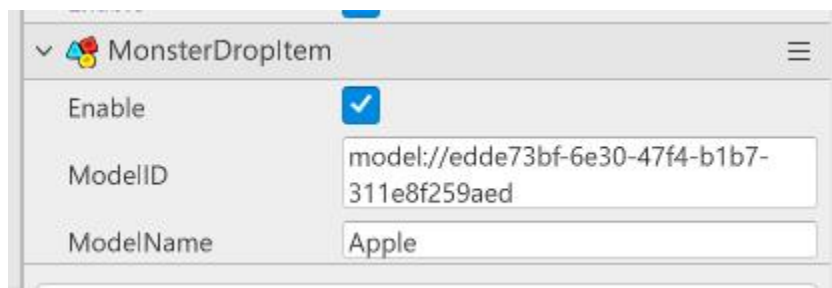


Add Component 를 눌러 MonsterDropItem 을 추가



※주의※ 아이템, 아이템 ID 를 Copy 해서 진행

드랍하고 싶은 아이템의 ID 와 이름을 넣음



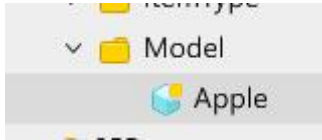
몬스터가 죽으면 아이템이 스폰되는지 확인



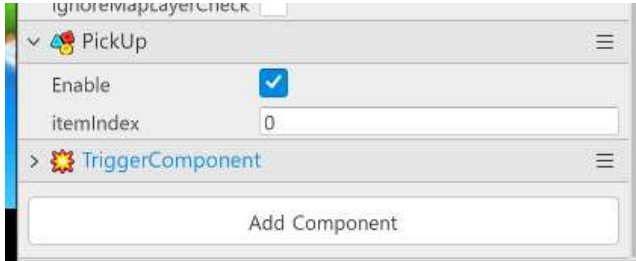
- 아이템을 인벤토리에 저장

1. 스크립트 넣기

아이템 Model 에서 시작



Add Component 로 Pickup Component 등록



2. 스크립트 수정

아이템의 이름과 타입 명을 Table 형식으로 저장

```
--UserItemDataSet에 추가되어 있는 아이템들의 이름과 아이템 타입 명을 Table로 만들어 줍니다.  
local itemNames = {  
  {"doll",Important},  
  {"Apple",Food}  
}
```

만약 3 번째로 Sword 를 저장하고 싶다면,

{"Apple",Food}, -- ,를 찍어주고

{"Sword",Weapon}

}

이런식으로 테이블을 늘리면 됨

3. 인덱스 수정

```
--UserItemDataSet에 추가되어 있는 아이템들의 이름과 아이템 타입 명을 Table로 만들어 줍니다.  
local itemNames = {  
  {"doll",Important},  
  {"Apple",Food}  
}
```

이 상태에서 doll 의 인덱스는 1, Apple 의 인덱스는 2 임

만약 위와 같이 Sword 를 Table(스크립트 안의)에 저장했다면 Sword 의 인덱스는 3 이됨

지금 예시에서는 사과를 소환하고 싶으니, 사과는 2 번 인덱스임

PickUp 의 itemIndex 에 2 번을 넣음



완성



+ 추가

현재 인벤토리에 계속해서 아이템이 쌓이므로, 임시적으로 아이템을 삭제해줄 “Model”인 “Garbage” Scene 에 깔아 놓고, 달으면 인벤토리에 있는 모든 아이템 삭제

