언리얼 프로그래밍 Part2-8

제목: 아이템 시스템

**강의 내용

강의 내용

캐릭터에 다양한 종류의 아이템을 제공하는 시스템을 구현

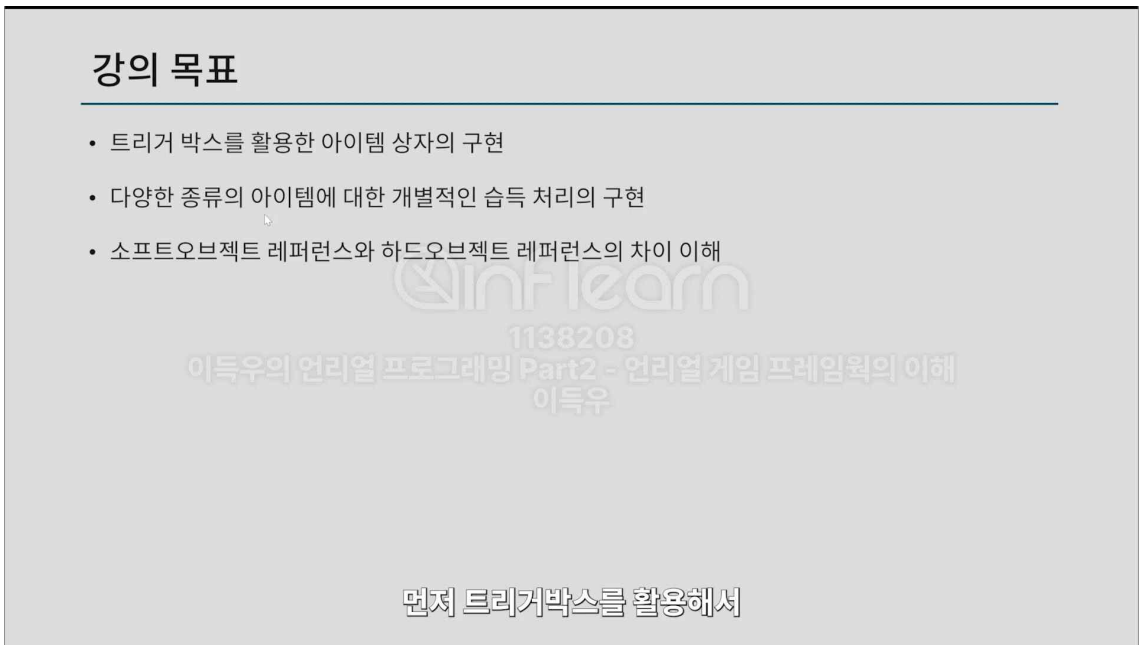
**강의에서 다루는 게임 프레임웍 요소

강의에서 다루는 게임프레임웍 요소

게임	월드	모드	상태	
기믹	트리거	스폰	물리	
플레이어	입력	카메라	HUD	상태
폰	이동	모션	액션	위젯
데이터	애셋	테이블	설정	저장
인공지능	길찾기	BT		

이번 강의에서 다루는 게임프레임워크 요소입니다

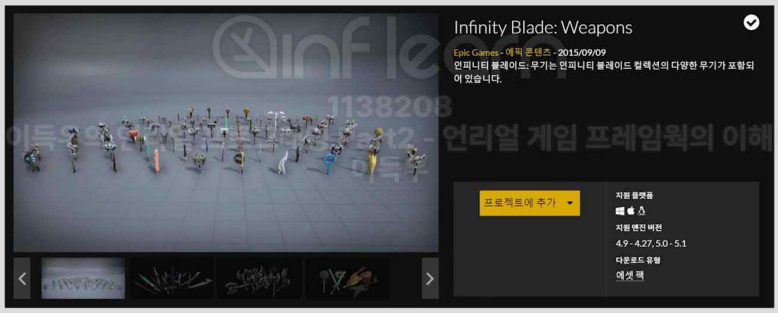
**강의 목표


강의 목표
• 트리거 박스를 활용한 아이템 상자의 구현
• 다양한 종류의 아이템에 대한 개별적인 습득 처리의 구현
• 소프트오브젝트 레퍼런스와 하드오브젝트 레퍼런스의 차이 이해
1138208
이득우의 언리얼 프로그래밍 Part2 - 언리얼 게임 프레임웍의 이해
이득우
먼저 트리거박스를 활용해서

***프로젝트 준비

인피니티 블레이드 웨폰

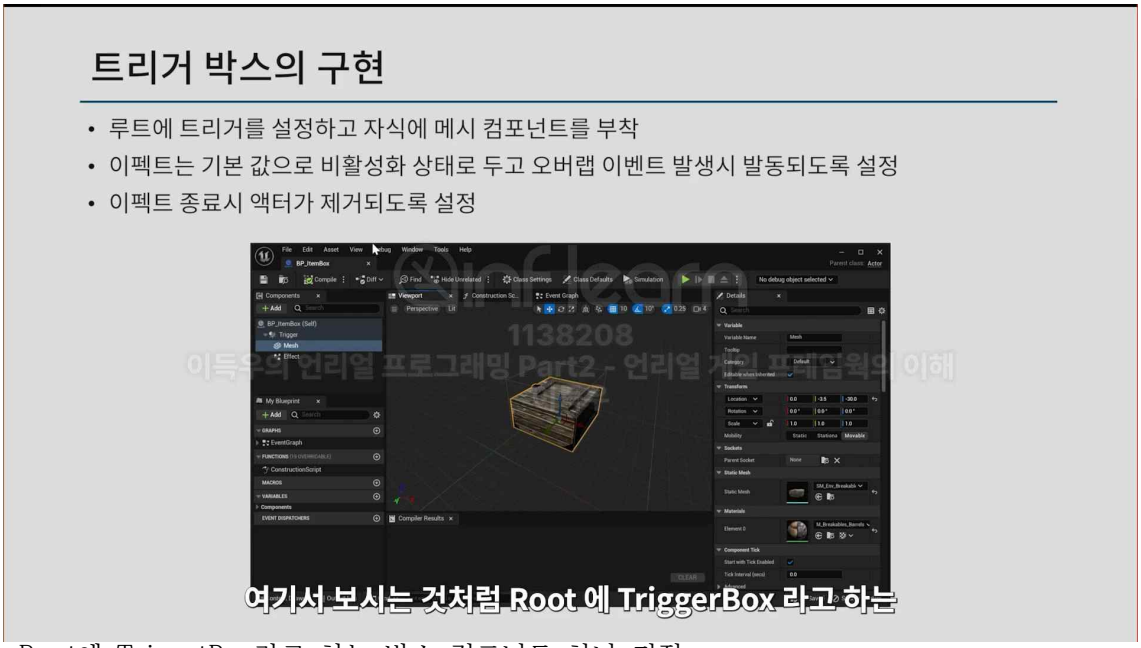
- 마켓플레이스에서 제공하는 인피니티 블레이드 웨폰 애셋을 현재 프로젝트에 추가
- Infinity Blade: Weapons



이를 위해서 마켓플레이스에서 제공하는 무료 에셋을

***트리거 박스의 설정

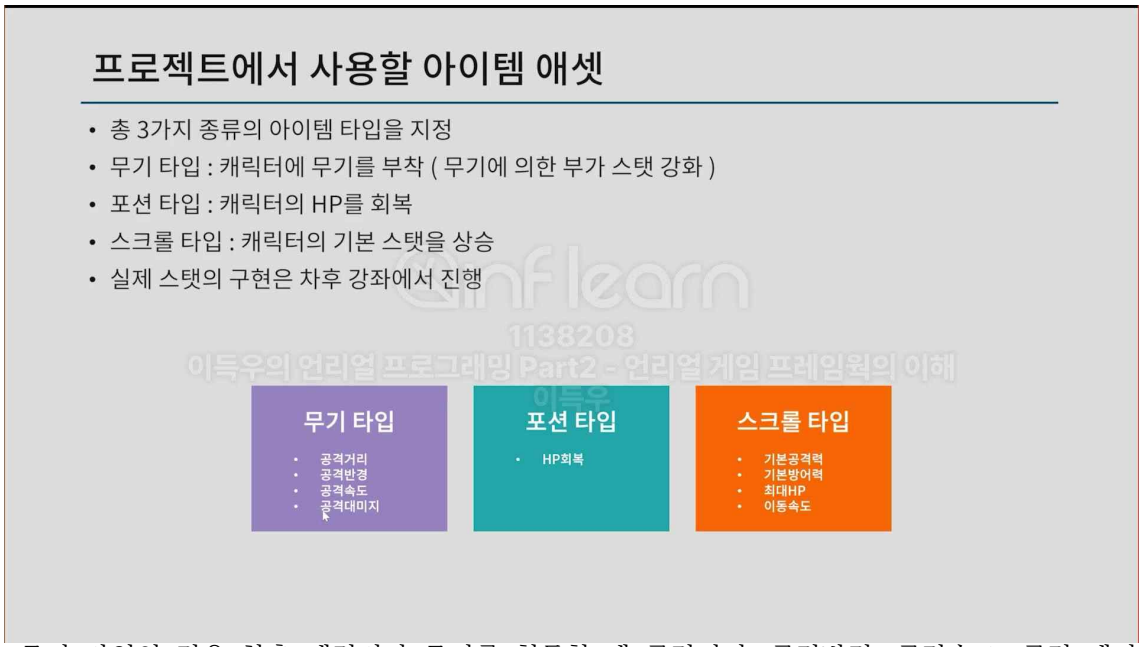
**트리거 박스의 구현



-Root에 TriggetBox라고 하는 박스 컴포넌트 하나 지정
-자식에 Mesh Component를 부착하여 상자의 모습을 띄도록 함
-상자에 닿았을 때 발동하는 이펙트 컴포넌트를 장착함.(이 이펙트는 기본값이 비활성 이지만, 캐릭터의 캡슐 영역이 상자의 트리거 영역에 겹치면 오버랩 이벤트가 발동되며 이벤트가 발동되도록 설정)
-발동된 이펙트가 종료되면 엑터가 스스로 제거되도록 설정.
-먼저 액터에 대한 c++클래스를 만들고 진행

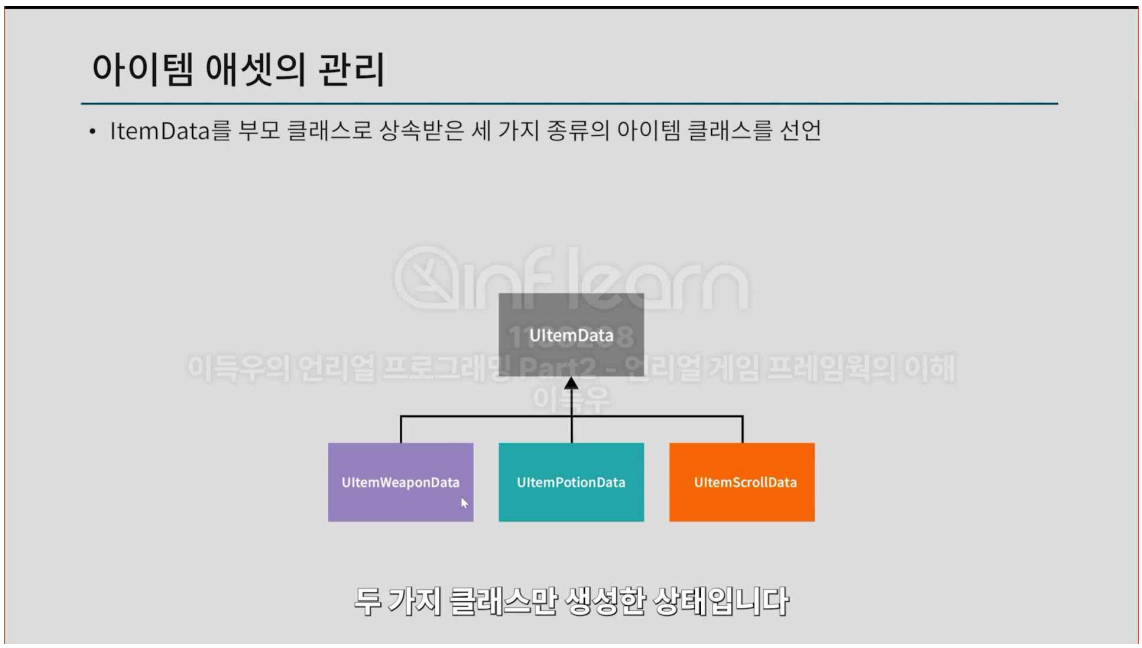
***아이템 에셋의 설계

**프로젝트에서 사용할 아이템 에셋



-무기 타입의 경우 향후 캐릭터가 무기를 획득할 때 공격거리, 공격반경, 공격속도, 공격 대미지라는 4가지 스탯에 추가적인 영향을 주도록 설계할거임(이번 강의에서 이 스탯에 대해서는 다루지 않을 예정이다.)
-포션타입은 HP회복
-스크롤 타입은 기본 스탯 정보를 올림.

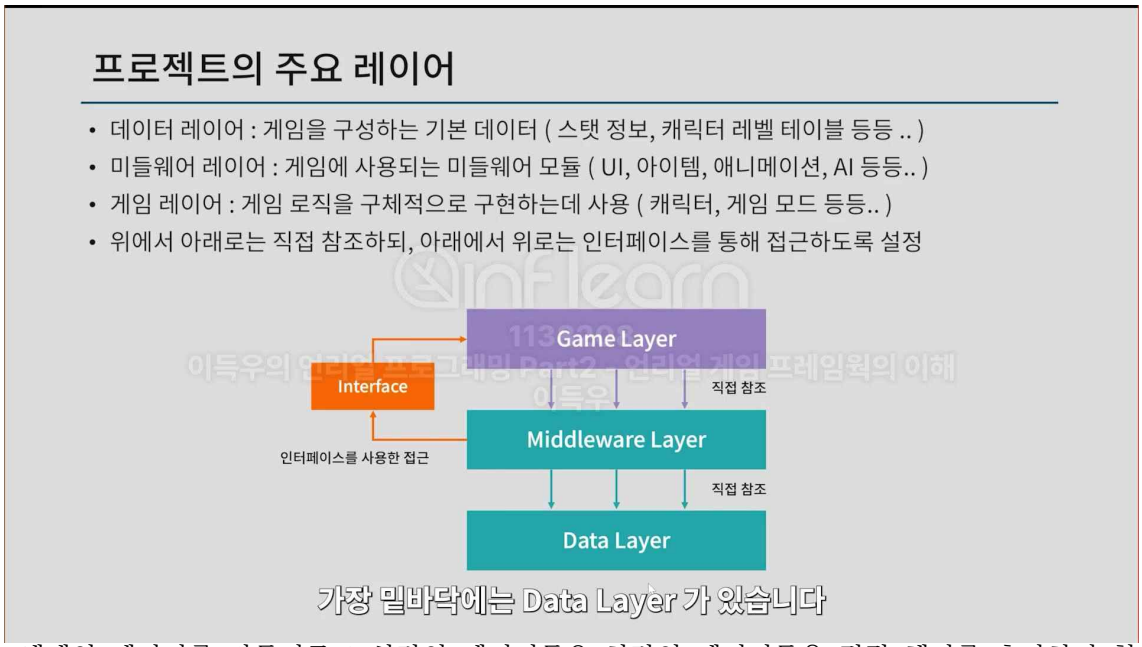
**아이템 에셋의 관리



***의존성 분리를 위한 설계 규칙

**만들어진 애셋들을 캐릭터에 전달할 때 캐릭터는 각각 습득한 아이템 종류별로 다른 처리를 진행해야 한다. 하지만 기획적으로 새로운 아이템이 추가가 될 수가 있는데 그럴 때마다 함수를 만들고 로직을 추가하는 것은 번거롭다. 이를 수월하게 진행할 수 있게 의존성을 최대한 분리된 상태에서 설계를 진행한다.

**프로젝트의 주요 레이어



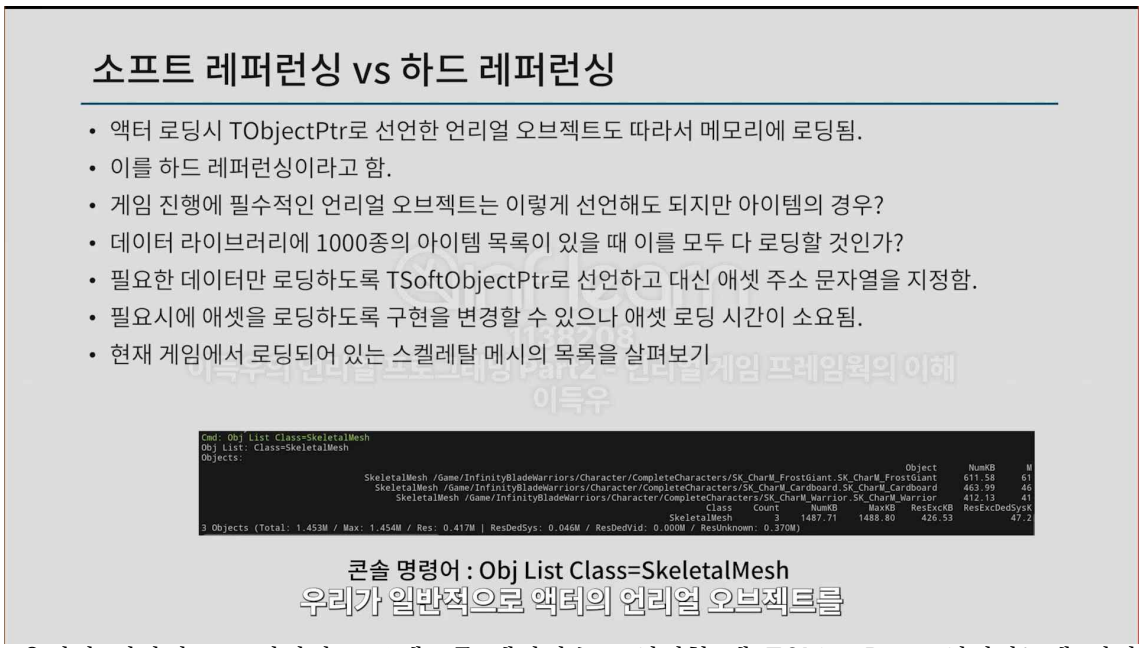
-세계의 레이어를 만들어주고 상단의 레이어들은 하단의 레이어들을 직접 헤더를 추가하여 참조할 수도 있지만, 반면 하단의 레이어들은 상단의 레이어를 직접 참조하지 않도록 설계 하자.
- 그런데 명령을 보내야 하는 경우가 있다. 예를 들면 내가 아이템을 먹었을 경우 캐릭터한테 해당 행동을 수행하라고 명령을 내려야 하는데 이때는 인터페이스를 사용하여 간접적으로 접근하도록 설정할 것이다.

-우리 프로젝트 기준으로 Item폴더의 경우 미들 웨어 이기 때문에 캐릭터 폴더의 헤더 정보를 인클루드 안하도록 규칙을 만들어줄 필요가 있다. 이렇게 해야 기능을 확장하는데 유연하게 대처가 가능하다.

-세가지 아이템을 먹었을 때 캐릭터에 설정된 세 가지 로직에 각각 바인딩 되어 호출되도록 최대한 의존성을 없앤 상태에서 구조를 구현해보자.

***소프트 레퍼런싱

**소프트 레퍼런싱 vs 하드 레퍼런싱



-우리가 일반적으로 언리얼 오브젝트를 멤버변수로 선언할 때 TObjectPtr로 선언하는데 이렇게 선언한 오브젝트는 메모리에 자동으로 액터가 로딩될 때 따라서 로딩된다 ==> 하드 레퍼런싱

-게임 진행에 필수적인 언리얼 오브젝트는 이렇게 따라서 로딩해도 되지만, 아이템의 경우 만약 데이터 라이브러리에 굉장히 많은 종류의 아이템 목록이 있는데 이것을 다 로딩하는건 메모리에 큰 부담이 될 수 있다.

-따라서 우리가 사용하는 필요한 데이터만 로딩하도록 멤버변수를 선언하되 로딩하지않게 TSoftObjectPtr로 선언해주면 에셋 대신에 에셋주소 문자열이 지정이 되면서 필요할 때만 에셋을 로딩할 수 있다.

**정리

아이템 상자와 캐릭터의 적용

1. 기믹 구현을 위한 트리거 액터의 설계
2. 데이터 애셋을 활용한 아이템 데이터 관리
3. 의존성 분리를 위한 설계 구현
4. 메모리 최적화를 위한 소프트 레퍼런싱의 구현