

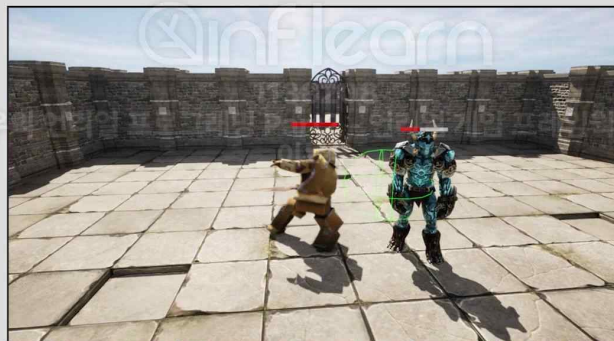
언리얼 프로그래밍 Part2-10

제목: 게임 데이터 관리

**강의 내용 :

강의 내용

데이터 기반의 캐릭터와 기믹 시스템의 제작



이번 강의에서는 데이터를 기반으로 기존에 우리가 제작했던 캐릭터와

**강의에서 다루는 게임프레임워크 요소

강의에서 다루는 게임프레임워크 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 게임프레임워크의 요소입니다

**강의 목표

강의 목표

- 게임 데이터를 관리하는 싱글톤 객체의 등록
- 엑셀 데이터 및 INI 파일을 활용한 게임 데이터의 관리
- 액터의 초기화를 위한 지연 생성 기능의 이해와 활용

먼저 게임 데이터를 관리하는

*** 데이터 기반의 게임 시스템 구축

**엑셀 파일 준비

-엑셀 파일을 csv파일로 저장하면 언리얼 엔진에서 읽을 수 있게됨.

-이를 위해서 먼저 언리얼 엔진에서 엑셀 파일 내에 메인 열(속성)을 가진 구조체를 먼저 선언해줘야 한다.

**엑셀 데이터의 임포트

엑셀 데이터의 임포트

- DataAsset과 유사하게 FTableRowBase를 상속받은 구조체를 선언
- 엑셀의 Name 컬럼을 제외한 컬럼과 동일하게 UPROPERTY 속성을 선언
- 엑셀 데이터를 csv로 익스포트한 후 언리얼 엔진에 임포트



Row Name	Max Hp	Attack	Attack Range	Attack Speed	Movement Speed
CLVL1	100.000000	100.000000	40.000000	1.000000	400.000000
CLVL2	150.000000	120.000000	40.000000	1.000000	400.000000
CLVL3	200.000000	140.000000	40.000000	1.100000	400.000000
CLVL4	250.000000	160.000000	40.000000	1.150000	400.000000
CLVL5	300.000000	180.000000	40.000000	1.200000	400.000000
CLVL6	350.000000	200.000000	40.000000	1.250000	400.000000
CLVL7	400.000000	220.000000	40.000000	1.300000	400.000000
CLVL8	450.000000	240.000000	40.000000	1.350000	400.000000
CLVL9	500.000000	260.000000	40.000000	1.400000	400.000000
CLVL10	550.000000	280.000000	40.000000	1.450000	400.000000

앞서 살펴본 CSV 파일을

-앞서 구축한 csv파일의 경우 언리얼 엔진에서 import가 가능하도록 기본적으로 기능을 제공합니다.

-이것을 사용하기 위해서는 FTableRowBase를 상속받은 구조체가 필요함

-이 구조체를 선언할 때 주의할 점은 엑셀 데이터 테이블의 컬럼에는 Name이라는 이름의 컬럼이 들어가야 한다.[이 값이 키값이 됨]

-이 컬럼을 제외한 나머지 데이터들을 우리들이 구조체를 선언해주면 된다.[당연히 우리가 제어 가능하도록 UPROPERTY 선언]

*실습

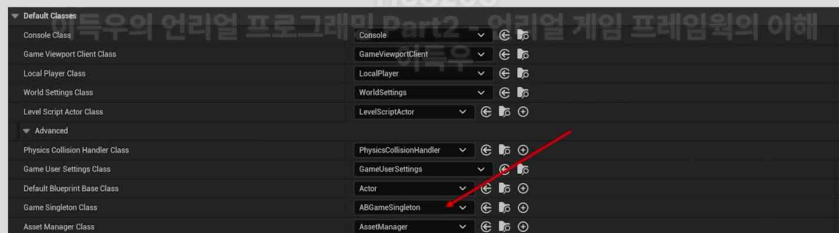
-데이터 테이블이 임포트가 완료되면 이것을 관리할 수 있는 별도의 클래스를 생성해보자

-이러한 클래스는 단 하나의 인스턴스에서 관리하는게 좋다.[싱글톤 형태로 객체를 만들어서 등록해보자]

****데이터를 관리할 싱글톤 클래스의 설정**

데이터를 관리할 싱글톤 클래스의 설정

- 언리얼 엔진에서 제공하는 싱글톤 클래스
 - 게임 인스턴스
 - 에셋 매니저
 - 게임 플레이 관련 액터 (게임 모드, 게임 스테이트)
 - 프로젝트에 싱글톤으로 등록한 언리얼 오브젝트
- 언리얼 오브젝트 생성자에서 사용하지 않도록 주의



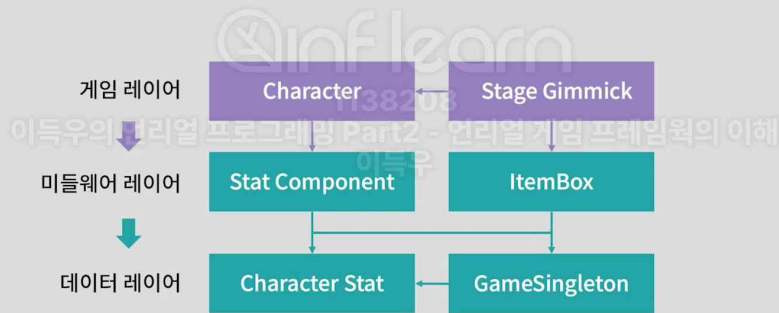
이번에는 데이터를 관리할 싱글톤 클래스를

- 언리얼 엔진에서 게임 관리를 위해 다양한 싱글톤 클래스 제공
- 싱글톤 클래스란 게임 내에서 단 하나의 인스턴스만 보장받는 클래스를 의미함.
- 대표적으로 파트1에서 게임 인스턴스 클래스를 볼 수 있음.(이전 강의에서 에셋 매니저 클래스)
- 초기에서 사용했던 게임 플레이에 관련된 액터들(게임 모드가 대표적)
- 게임 모드의 경우에는 전체 게임(멀티플레이를 포함)내에서 단 하나만 존재하는 굉장히 특이한 싱글톤 클래스임.
- 이번에 우리는 게임 클라이언트 내에서 데이터를 관리하는 용도로 사용할 것이기 때문에 게임 인스턴스를 사용해도 되지만 게임 인스턴스는 주로 어플리케이션의 데이터를 관리하는 용도로 사용 되기 때문에 게임 콘텐츠에 직접적인 영향을 미치는 새로운 클래스를 생성하고 이를 싱글톤으로 등록하자.
- 언리얼 엔진에서 이를 싱글톤으로 등록하는 방법은 프로젝트 세팅->일반설정->Default Class->Advanced 탭으로 가서 게임 싱글톤 클래스에 우리가 만든 클래스를 등록하면 엔진이 초기화 될 때 싱글톤을 GEngine이라고 하는 전역 변수에 자동으로 만들어 준다. 우리는 이것을 사용 하면 됨.
- Advanced 탭을 보면 AssetManager라는 값이 기본적으로 등록된 것을 볼 수가 있음. 그렇기에 우리가 이전 강의에서 에셋 매니저를 싱글톤으로 안전하게 사용이 가능 했다.

**프로젝트의 주요 레이어

프로젝트의 주요 레이어

- 게임 레이어 : 기믹과 NPC
- 미들웨어 레이어 : 캐릭터의 스탯 컴포넌트, 아이템 박스
- 데이터 레이어 : 스탯 데이터, 데이터 관리를 위한 싱글톤 클래스



우리가 이번 강의에서 진행할 프로젝트는

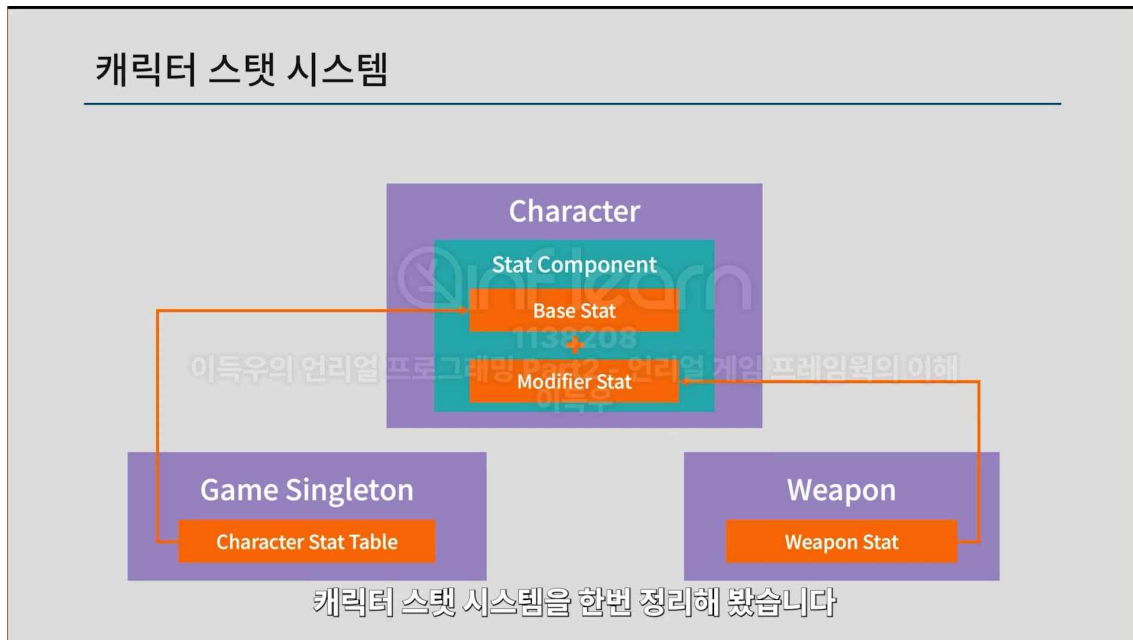
-게임 진행에 사용하는 플레이어 '캐릭터', 'NPC캐릭터', '스테이지 기믹'들은 최상단의 게임 레이어에 위치해 있다고 볼 수 있다.

-이러한 게임 레이어의 객체들은 미들웨어 레이어에 속한 '스탯 컴포넌트'와 '아이템 박스'들을 사용함.

-이러한 스탯 컴포넌트와 아이템 박스들은 이번 강의에서 제작한 데이터 레이어에 위치한 캐릭터 스탯이나 게임 싱글톤을 참조하도록 선언하였다.

-이렇게 위에서 아래로만 참조를 하고 밑에서는 위로참조 하지 못하게 전체적인 구조를 잡아봄.

**캐릭터 스탯 시스템



- 캐릭터가 소유한 스탯 컴포넌트는 스탯 데이터를 가지고 있다.
- 자신의 레벨에 따라서 게임 싱글톤으로부터 스탯정보를 가져와 베이스 스탯 멤버변수에 저장
을 함.
- 이러한 스탯은 Modifier Stat이라고 하는 또 하나의 스탯 정보를 가지고 있다. 이는 캐릭터
가 무기를 착용할 때 weapon으로부터 스탯정보를 받아 모디파이어 스탯에다가 저장을하고
캐릭터의 최종 스탯은 Base Stat +Modifier Stat을 더한 최종 값으로 공격 판정을 진행하도
록 설계한다.

**액터의 생성과 지연 생성의 프로세스



-우리가 일반적으로 `SpawnActor`라는 API를 통해 액터를 생성하면 액터와 액터가 가지고 있는 모든 컴포넌트에 `BeginPlay`라는 함수가 바로 호출이 됨.

-그래서 앞선 코드에서 `ABCharacterStatComponent.cpp`를 보면 `SpawnActor`가 수행되자마자 `SetLevelStat`과 `SetHp`함수가 호출이 됨. 근데 이때 `CurrentLevel`값은 기본값인 1을 가지고 있기 때문에 HP는 1레벨만큼 차게 됨.

-그다음 `ABStageGimmick.cpp`로 가면 기믹은 이미 `SpawnActor`를 호출하여 `BeginPlay`를 실행한 상태에서 `SetLevel`을 진행하였기 때문에 `CurrentLevel`은 반영이 안되고 `BaseStat`에만 반영이 된 것을 볼 수가 있습니다.

-이 문제를 해결하기 위하여 `SpawnActorDeferred`라고 하는 지연 생성함수를 제공하고 있다.

-액터를 지연생성하는 `SpawnActorDeferred`함수는 마지막에 `FinishSpawning` 함수를 호출해줘야지만 그제서야 `BeginPlay`가 호출된다. 따라서 우리가 `FinishSpawning` 함수를 호출하기 전에 초기화를 진행하면 `BeginPlay`로 호출하는 `SetHp`함수를 올바르게 실행할 수 있음.

-우리가 기존에 작업했던 모든 스폰코드는 초기값이 반영이 되어야 한다면 지연생성코드로 바꿔줘야 좋다.

****정리**

데이터 기반의 게임 시스템 구축

1. 외부 데이터 파일로부터 게임 데이터를 관리하는 다양한 방법의 학습
2. 데이터를 관리하는 싱글톤 클래스 생성 방법의 이해
3. 데이터 테이블 기반의 캐릭터 스탯 시스템 구축
4. 지연 생성을 활용한 액터 초기화 방법의 이해

이번 강의에서는 데이터를 기반으로