

언리얼 프로그래밍 Part2-1

제목:언리얼 엔진 게임 제작 기초

**강의 내용 : 예제 게임을 활용한 언리얼 엔진의 게임 프레임워크의 소개

**강의 목표 :

강의 목표

- 언리얼 게임 프레임워크를 활용한 게임 제작 방식의 이해
- 강의 예제에 사용되는 프로젝트와 기본 C++ 클래스 생성
- 향후 운영될 C++ 프로젝트 운영 규칙의 이해

inflearn
1138208
이득우의 언리얼 프로그래밍 Part2 - 언리얼 게임 프레임워크의 이해
이득우

언리얼 엔진이 제공하는 게임 프레임워크를 바탕으로 해서

**게임 콘텐츠의 구조

게임 콘텐츠의 구조

- 게임 제작을 위해 언리얼 엔진은 자체적으로 설계한 프레임워크를 제공함.
- 이를 게임플레이 프레임워크(Gameplay Framework)이라고 부름. 줄여서 게임 프레임워크
- 언리얼 게임 프레임워크의 각 구성 요소를 파악하고, 이를 확장하면서 게임을 제작하는 것을 권장



이 게임이라는 콘텐츠를 체계적으로 분석하는 것이 쉬운 일이 아니네요

-우리가 제작할 게임 콘텐츠는 다양한 구성요소가 모여서 만들어 지는데 모두 언리얼 엔진에서 제공을 한다 따라서 최대한 있는 것을 가져다 쓰면 쉽게 게임을 만드는 것이 가능하다.

**1강 예제 게임 콘텐츠의 구성

1강 예제 게임 콘텐츠의 구성

- 1강 예제 게임 콘텐츠에 사용된 게임플레이 프레임워크의 기능 목록

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		

-우리가 살펴본 예제는 전체 기능중 민트색 색칠된것처럼 일부만 사용하여 구현함 이중 각각의 주요 요소를 살펴볼것임.

**월드(World)

월드(World)

- 게임 콘텐츠를 담기 위해 제공되는 가상의 공간
- 월드는 시간, 트랜스폼, 틱을 서비스로 제공한다.
- 월드 세팅이라는 콘텐츠 제작을 위한 기본 환경 설정을 제공한다.
- 월드의 기본 단위는 액터(Actor)로 정의되며, 액터 클래스는 언제나 접두사 A를 사용한다.

게임	월드	모드	상태	1133 리얼프로그래밍Par 이동 모션 액션 위젯
기믹	트리거	스폰	물리	
플레이어	입력	카메라	HUD	상태
폰	이동	모션	액션	위젯
데이터	애셋	테이블	설정	저장
인공지능	길찾기	BT		

월드의 역할

- 게임 콘텐츠를 담는 공간 (Transform)
- 게임 콘텐츠의 진행 (Tick)
- 게임 흐름을 위한 시간 (Time)
- 기본 환경 설정 (World Setting)
- 월드의 단위 구성 요소 (Actor)

월드라는 것은 게임 콘텐츠를 담기 위해서

-월드는 좀 특별한 언리얼 오브젝트인데 시간, 트랜스폼, 틱이라고 하는 독특한 기능을 서비스로 제공함.

-월드가 제공하는 기본 서비스를 바탕으로 동작하는 기본 콘텐츠 단위가 있는데 이것을 Actor라고 한다.(액터 클래스는 접두사A를 통해 클래스명 지정해줘야함)

**게임모드(Game Mode)

게임 모드(Game Mode)

- 게임 규칙을 지정하고 게임을 판정하는 최고 관리자 액터. 형태가 없다.
- 언리얼 엔진에서 하나의 게임에는 반드시 하나의 게임 모드만 존재한다.
- 게임 모드에서 입장할 사용자의 규격을 지정할 수 있음.
- 멀티플레이어 게임에서 판정을 처리하는 절대적 권위의 심판

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		

게임 모드의 역할

- 플레이어 입장 관리
- 게임의 최종 승패 관리
- 게임의 규칙 관리
- 게임 데이터 검증 (멀티플레이 게임)

게임 모드는 게임 규칙을 지정하고 게임을 판정하는 최고 관리자 액터입니다

**기믹(Gimmick)

기믹(Gimmick)

- 게임 진행을 위한 이벤트를 발생시키는 사물 액터
- 주로 이벤트 발생을 위한 충돌 영역을 설정하는데, 이를 트리거(Trigger)라고 한다.
- 트리거를 통해 캐릭터와 상호 작용하고, 월드에 액터를 스폰해 콘텐츠를 전개한다.

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

기믹의 역할

- 월드 내 사물과 상호 작용을 수행
- 지정된 조건을 만족하면 이벤트를 발생
- 월드에 물체를 스폰(Spawn)시킬 수 있음

이벤트를 발생시켜주는 어떤 사물 액터를 의미합니다

-기믹은 주로 이벤트 발생을 위해 눈에 안보이는 충돌 영역을 설정하는데 이를 트리거라고 함,

**플레이어(Player)

플레이어(Player)

- 게임에 입장한 사용자 액터. 형태가 없다.
- 게임 모드의 로그인을 통해 사용자가 게임 월드에 입장하면 플레이어가 생성된다.
- 싱글 플레이 게임에는 0번 플레이어가 설정된다.
- 사용자와의 최종 커뮤니케이션을 담당한다. (예) 입력 장치의 해석, 화면 장치로의 출력)

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

플레이어의 역할

- 게임에 입장한 사용자와의 1:1 대응
- 사용자의 현재 상태 및 데이터를 관리
- 사용자 입력의 1차 처리 (필요시)
- 사용자 화면에 대응하는 카메라 관리
- 사용자 화면에 나타나는 HUD UI 관리

이것 역시 게임 모드와 마찬가지로 형태가 없습니다

-플레이어 또한 게임모드와 같이 형태가 없다.

-로그인이라고 하면 서버통신을 생각할 수도 있지만, 게임 모드가 의례적으로 플레이어를 생성하기 위해서 진행되는 절차라고 생각하자. --> 싱글플레이에서도 사용자가 입장에서 플레이어가 생성되는데 이때 0번이라고 하는 특수한 아이디가 배정됨. 이러한 플레이어는 사용자와 일대일로 대응되기 때문에 사용자와의 최종 커뮤니케이션을 담당한다.

-따라서 입력장치를 해석한다던가 화면장치로 출력하는 것을 플레이어가 담당한다.

**폰(Pawn)

폰(Pawn)

- 무형의 액터인 플레이어가 빙의해 조종하는 액터
- 길찾기를 사용할 수 있으며, 기믹 및 다른 폰과 상호작용한다.
- 폰 중에서 인간형 폰을 별도로 캐릭터라고 지칭한다.

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

폰의 역할

- 빙의(Possess)를 통해 플레이어와 연결
- 사용자 입력의 실제 처리
- 사용자 화면에 대응되는 카메라 설정
- 기믹과의 상호작용
- 현재 상태에 적합한 애니메이션 재생

앞서서 플레이어는 무형의 액터라고 설명드렸습니다

-앞서서 플레이어는 무형의 액터라고 했는데 언리얼 엔진의 경우 사용자와 1:1 로 대응되는 플레이어는 무형의 액터지만, 월드에 배치된 어떤 눈에 보이는 액터에 빙의하여 조종하는 개념을 가짐. (이 조종 당하는 액터를 Pawn이라고 한다.)

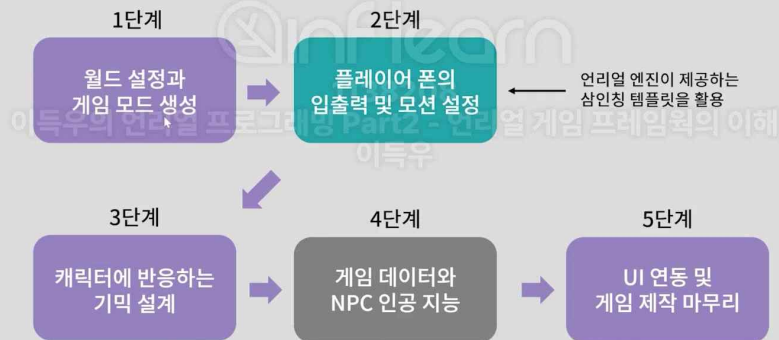
-Pawn은 의지를 가지고 있지 않고, 의지를 가진 플레이어가 빙의해서 조종하는대로 따라주는 사물을 의미한다. 이러한 폰은 길찾기를 사용할 수 있고, 기믹 및 다른 폰과 상호작용이 가능하다.

-인간형으로 구성된 폰을 캐릭터라고 언리얼 엔진에서 지칭한다.

**게임의 제작 과정

게임의 제작 과정

- 이번 과정에서 제작할 싱글플레이 게임 제작은 크게 총 다섯 단계로 나눌 수 있음
- 1강의 샘플 예제 게임은 블루프린트를 사용해 세 단계만 작업했음.



**최종 예제 콘텐츠의 구성

최종 예제에서 확장할 내용

- 기본 템플릿을 사용하지 않고 직접 제작
- 두 가지의 카메라 모드를 제공 (솔더뷰, 탑뷰)
- 액션 입력을 통한 콤보 공격의 구현
- 아이템 애셋을 활용한 무기 상자의 구현
- NPC 레벨 정보를 스프레드시트로 관리
- 플레이어와 전투하는 NPC 인공지능의 구현
- UI 연동과 게임의 마무리

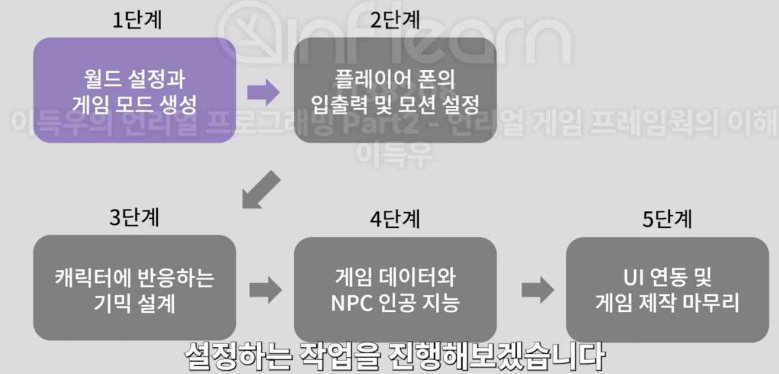


최종 예제에서 확장할 내용을 조금 더 정리를 해봤는데요

**프로젝트의 시작

프로젝트의 시작

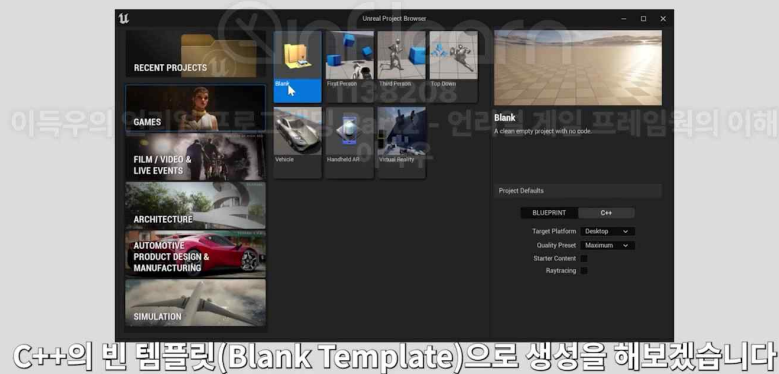
- 신규 프로젝트 생성과 1단계 작업을 위한 준비



**프로젝트 생성

프로젝트 생성

- C++ Blank 템플릿으로 생성
- 프로젝트 이름은 ArenaBattle
 - 주 게임 모듈 이름은 자동으로 ArenaBattle로 지정됨.



**프로젝트 소스코드 폴더 규칙

프로젝트 소스코드 폴더 규칙

- 게임플레이를 구성하는 요소별로 폴더를 제작해 관리
 - **Game** : 게임 모드와 게임 설정에 관련된 클래스 (이번 강의에서 생성)
 - **Gimmick** : 기믹 배경 물체에 관련된 클래스
 - **Player** : 플레이어에 관련된 클래스 (이번 강의에서 생성)
 - **Input** : 플레이어 입력에 관련된 클래스
 - **Character** : 기본 캐릭터와 PC 및 NPC에 관련된 클래스 (이번 강의에서 생성)
 - **Action** : 캐릭터 액션에 관련된 클래스
 - **Item** : 캐릭터 아이템에 관련된 클래스
 - **Stat** : 캐릭터 스탯에 관련된 클래스
 - **AI** : NPC 인공지능에 관련된 클래스
 - **UI** : UI에 관련된 클래스
 - **Physics** : 물리 충돌체 설정에 관련된 전처리기
 - **Interface** : 인터페이스 클래스를 모아둔 폴더

이번 프로젝트의 소스 코드 폴더 규칙입니다

-게임과 플레이어 캐릭터 세 가지의 폴더만 생성해보자

**클래스 생성 규칙

클래스 생성 규칙

- 모든 클래스는 AB(ArenaBattle의 약자)의 접두사로 시작함
- 폴더마다 다른 폴더와의 의존성을 최소화
- 다른 폴더의 클래스에 접근할 때 가급적 인터페이스를 통해 접근
- 헤더 경로를 찾기 쉽도록 모듈.Build.cs 파일에 모듈의 폴더를 인클루드 경로에 설정
 - `PublicIncludePaths.AddRange(new string[] { "모듈이름" });` 추가

가급적이면 다른 폴더의 헤더를 직접 참조하지 않도록 구성하는 것이 바람직

우리 프로젝트는 AB라고 하는 접두사로 시작하도록 하겠습니다

****최초 생성할 클래스**

최초 생성할 클래스

- 게임을 구성하는 기본 골격의 설정
- 캐릭터는 PC와 NPC를 구분해 구현할 예정
- 게임 모드에서는 입장할 플레이어의 규격을 지정
(AABPlayerController, AABCharacterPlayer)



플레이어 컨트롤러도 생성을 해 보겠습니다

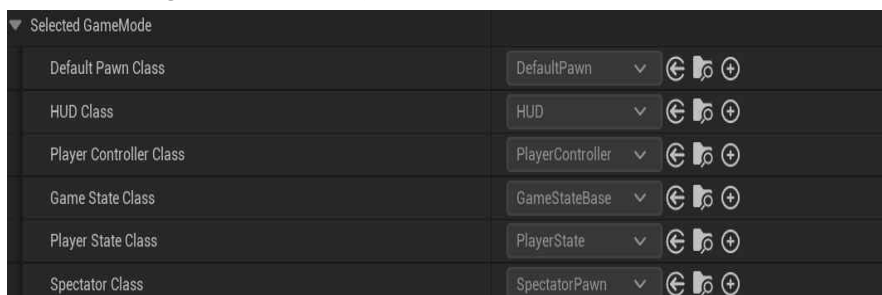
**실습

-언리얼엔진은 Maps라는곳에 월드와 레벨을 보통 저장한다.

-게임 모드 베이스를 ArenaBattle->Game폴더에 생성하면 빌드 에러가 나타날텐데 ArenaBattle.Build.cs에 한줄 추가해준다[PublicIncludePaths.AddRange(new string[] { "ArenaBattle" });]

-Step2라고 하는 레벨에서 우리가 3인칭 템플릿을 제공하는 게임모드를 지정하면 이 게임모드를 사용하여 캐릭터가 로그인하여 등장한다.[Step2레벨의 월드셋팅에서 게임모드를 BP_ThirdPersonCharacter로 하고 플레이 하면 마네키가 등장한다]

-WorldSetting Gamemode설명



(1)Default Pawn Class가 우리가 등장시킬 조종할 사물에 대한 캐릭터 사물에 대한 클래스를 지정하는 곳

(2)Player Controller Class는 언리얼에서 기본으로 제공하는 플레이어 컨트롤 클래스를 사용한다.

(3)게임모드를 ABGameMode로 변경하면 아무값도 안들어가 있기 때문에 언리얼에서 제공하는 기본 클래스를 사용하여 플레이어를 스폰하도록 설정이 되어 있다.

(4)ABGameMode에서 Default Pawn Class의 경우 BP_ThirdPersonCharacter로 변경을 하고 Player Controller Class의 경우에는 ABPlayerController로 변경을 할 것이다.

■ DefaultPawn을 BP_ThirdPersonCharacter로 변경하기 위해서는 BP_ThirdPersonCharacter가 c++ 클래스가 아니기 때문에 우리가 이 에셋으로부터 클래스 정보를 얻어와야 한다.

■ BP_ThirdPersonCharacter도 캐릭터를 상속받은 엄연한 언리얼 체계의 클래스 구조를 따르기 때문에 에셋으로부터 클래스 정보를 얻어올 수 있다.(References - Copy Reference)

■ 클래스 형에 대해서는 앞에 있는 파옴표 부분 `[/Script/Engine.Blueprint']`은 생략해 주고 에셋의 경로만 입력해준다. 그리고 클래스 정보만 가져올것이기에 마지막에 `'_C'`를 붙여준다.

(5)ABPlayerController를 사용하여 직접적으로 시작할 때 마우스가 뷰포트 안으로 들어가게 설정하기

(6) 우리가 만든 C++ 객체들은 Content Browser에 보이는 것처럼 계층구조 폴더 구조를 가지고 있지만, 독립적인 파일로 구성되어 있지 않지만, 고유한 경로를 가지고 있다. 따라서 Player 폴더의 ABPlayerController의 경우에도 레퍼런스[고유 주소값]을 가지고 있다. 이것을 똑같이 사용하여 ThirdPerson에서 사용했던것처럼 동일하게 사용하면 헤더를 include하지 않아도 클래스 정보를 얻어올 수가 있다.

(7)ABPlayerController도 비슷하게 해보자.==>`static ConstructorHelpers::FClassFinder<APlayerController> PlayerControllerClassRef(TEXT("/Script/ArenaBattle.ABPlayerController"))` 사용

**정리

언리얼 게임프레임웍의 이해

1. 게임 콘텐츠 구조의 이해
2. 게임 프레임워크의 구성 요소 살펴보기
3. 단순한 게임 구조에서 출발해 점점 복잡한 게임으로 확장하는 제작 과정을 이해
4. C++ 프로젝트 구성을 위한 기본 설정
5. 게임 구성 요소를 폴더별로 분류하고, 헤더 참조를 최소화하는 규칙 수립

먼저 게임 콘텐츠를 구성하는 여러가지