



언리얼 엔진4 용어 정리

- Project
 - 각각의 게임을 이루는 콘텐츠와 코드 전부가 들어있는 독립적인 유닛(디렉터리)
 - 파일과 관련된 .uproject 파일을 가리키기도 하지만 서로와 공존하는(?) 별개의 파일이 2개 존재
 - .uproject : 파일을 생성하고 열고 저장하는 데 사용되는 레퍼런스 파일
 - Project : 프로젝트에 관련된 모든 파일과 폴더
 - 프로젝트는 동시에 몇 개를 만들어 유지하면서 개발 가능
- Object
 - 언리얼 엔진의 기본 구성 요소
 - 게임 에셋에 대한 본질적인 “내부적” 함수성이 다수 포함
 - 언리얼 엔진4의 모든 것이 object를 상속함(함수성을 이어 받음)
 - UObject는 모든 오브젝트의 베이스 클래스
 - 가비지컬렉션, 언리얼 에디터로의 변수노출을 위한 메타데이터 지원, 로드 & 저장을 위한 시리얼라이제이션 등의 기능 구현
- Class
 - 게임 제작에 사용되는 특정 액터 또는 오브젝트의 작동방식 및 프로퍼티 정의
 - 계층구조식이라 상속받아서 만들어진 클래스가 그 정보를 또 자손에게 물려주기 가능
 - C++ 또는 blueprint로 생성 가능
- Actor

- 레벨에 배치할 수 있는 오브젝트
- 이동, 회전, 스케일과 같은 3D transform을 지원하는 범용 클래스
- 게임플레이 코드(C++, blueprint)를 통해 생성 및 소멸 가능
- AActor는 모든 액터의 베이스 클래스

- Component
 - 액터에 추가시킬 수 있는 함수성 조각
 - 독립적으로 존재 x But 액터에 추가시키면 자립적인 함수성 조각

- Pawn
 - Actor의 서브클래스
 - 게임 내 아바타 또는 페르소나 역할
 - 플레이 불가능한 캐릭터(NPC)형태로 게임의 AI 제어 받기도 가능
 - pawn이 제어를 받을 때는 (Posses) 빙의된 것으로 간주, 제어 안 받을 땐 (Unpossess) 빙의되지 않은 걸로 간주

- Characters
 - Pawn의 서브 클래스
 - 플레이어 캐릭터에 쓰임
 - 서브클래스
 - 콜리전 셋업
 - 이족 보행을 위한 입력키 바인딩
 - 플레이어가 제어하는 이동에 대한 부가 코드
 - etc,,

- PlayerController
 - 플레이어 입력을 받아 게임 내 상호작용으로 변환하는 데 사용되는 베이스 클래스
 - 모든 게임에는 최소 하나의 플레이어 컨트롤러 존재

- 종종 게임 내 플레이어의 표상으로 폰 또는 캐릭터를 빙의
- 멀티 플레이어 게임에서 PlayerController가 주요 네트워크 상호작용 지점이 되기도 함
- AIController
 - 게임 내 NPC의 표상으로 Pawn을 빙의
 - 기본적으로 Pawn과 Character는 구체적인 PlayerController에 빙의되거나 별도의 AIController를 만들어주지 않으면, 기본 AIController 가짐
- Brush
 - 레벨 지오메트리(BSP) 및 게임플레이 볼륨을 정의하기 위해 레벨에 매치하는 3D 볼륨
 - BSP 브러시를 사용하여 게임플레이 테스트용 레벨 프로토타입 또는 기본적인 윤곽 작업 진행
 - 볼륨에 붙는 이펙트에 따른 여러가지 용도
 - 블로킹 볼륨 : 보이지는 않지만 액터의 통과를 막는 데 사용되는 볼륨
 - 페인 코징 볼륨 : 겹치는 액터에게 시간에 따라 서서히 데미지를 입히는 볼륨
 - 트리거 볼륨 : 액터가 드나들 때 이벤트 유발용으로 사용되는 볼륨
- Level
 - 게임플레이에 사용자 정의된 영역
 - Level 안에 액터를 배치하고 트랜스폼을 적용한 뒤 프로퍼티를 편집함으로 레벨을 생성, 확인, 변경함
 - 에디터에서 각 Level은 별도의 .umap 파일로 저장되어 가끔 map이라고도 일컬어짐
- World
 - 로드된 레벨 목록이 들어있음
 - 레벨 스트리밍 및 동적인 액터 스폰을 처리함

- 월드와의 직접 상호작용이 필수는 아니지만, 게임 구조 내 구체적인 언급 지점을 제공하는 데 도움이 된다(이해 못 함)

- GameMode

- 플레이 중인 게임의 규칙 설정을 담당
 - 게임 참가 방식, 게임 일시정지 가능 여부, 레벨 전환, 승리 조건과 같은 게임 전용 작동방식
- 프로젝트 세팅에서 기본 게임 모드 설정 가능, 레벨 단위로 덮어쓰기 가능
- 레벨당 항상 단 하나의 게임모드가 존재
- 멀티 게임에서는 게임 모드는 서버에서만 존재, 규칙은 리플리케이션을 통해 접속된 각 클라이언트에 전송

- PlayerState

- 사람 플레이어 또는 플레이어를 흉내내는 봇과 같은 게임내 참여자의 상태
- 게임 월드의 일부로 존재하는 Player가 아닌 AI의 경우 PlayerState가 없다
- ex) 플레이어 이름, 점수, 그 현재 레벨 또는 생명력 등등
- 멀티 게임에서 모든 플레이어에 대한 PlayerState는 모든 머신에 존재하며 (PlayerController와 다름) 서버에서 클라이언트로 프로퍼티를 리플리케이션 하여 동기화 상태 유지 가능