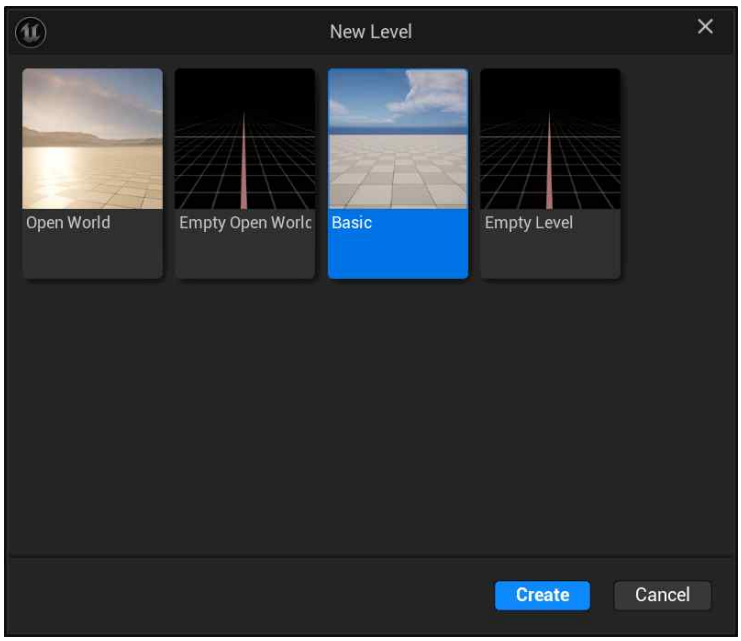
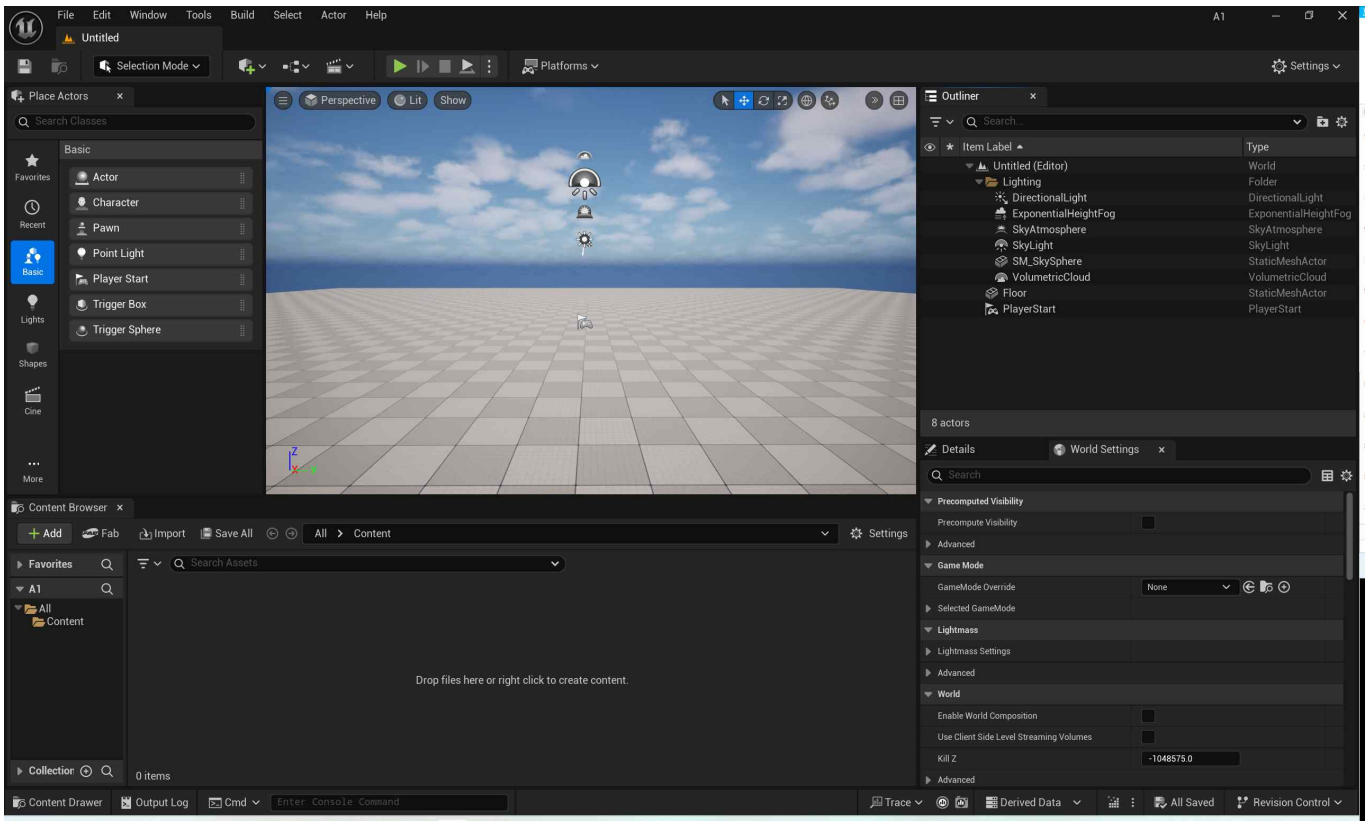
02_언리얼 프로젝트

1. 언리얼 프로젝트

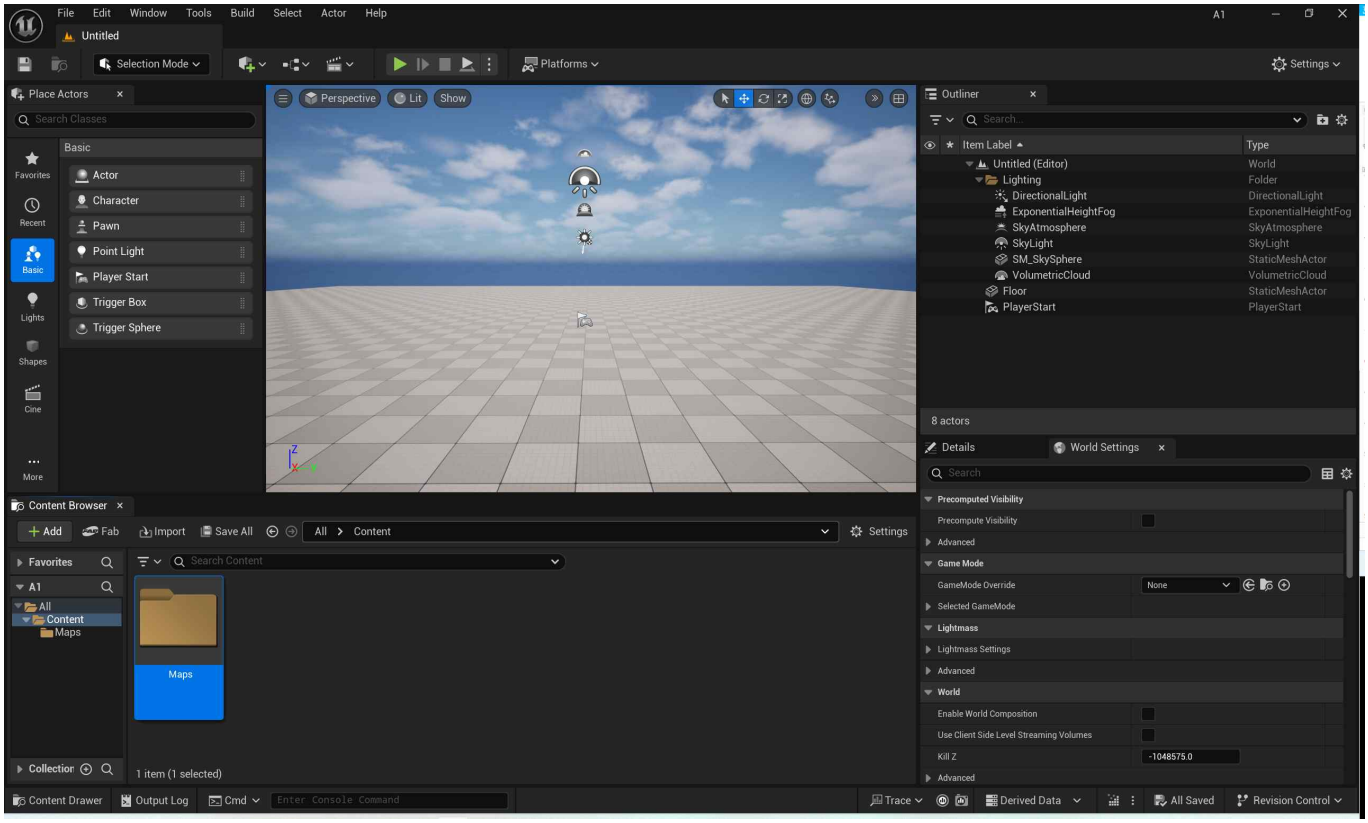
1) 새로운 레벨 만들기

- Unity의 Scene에 해당하는 Level을 생성하도록 하자.
- [**File**] - [**New Level...**] 또는 Ctrl + N 단축키를 눌러서 새로운 Level을 생성하자.
- **Basic Level**을 선택하여 Create 하자.

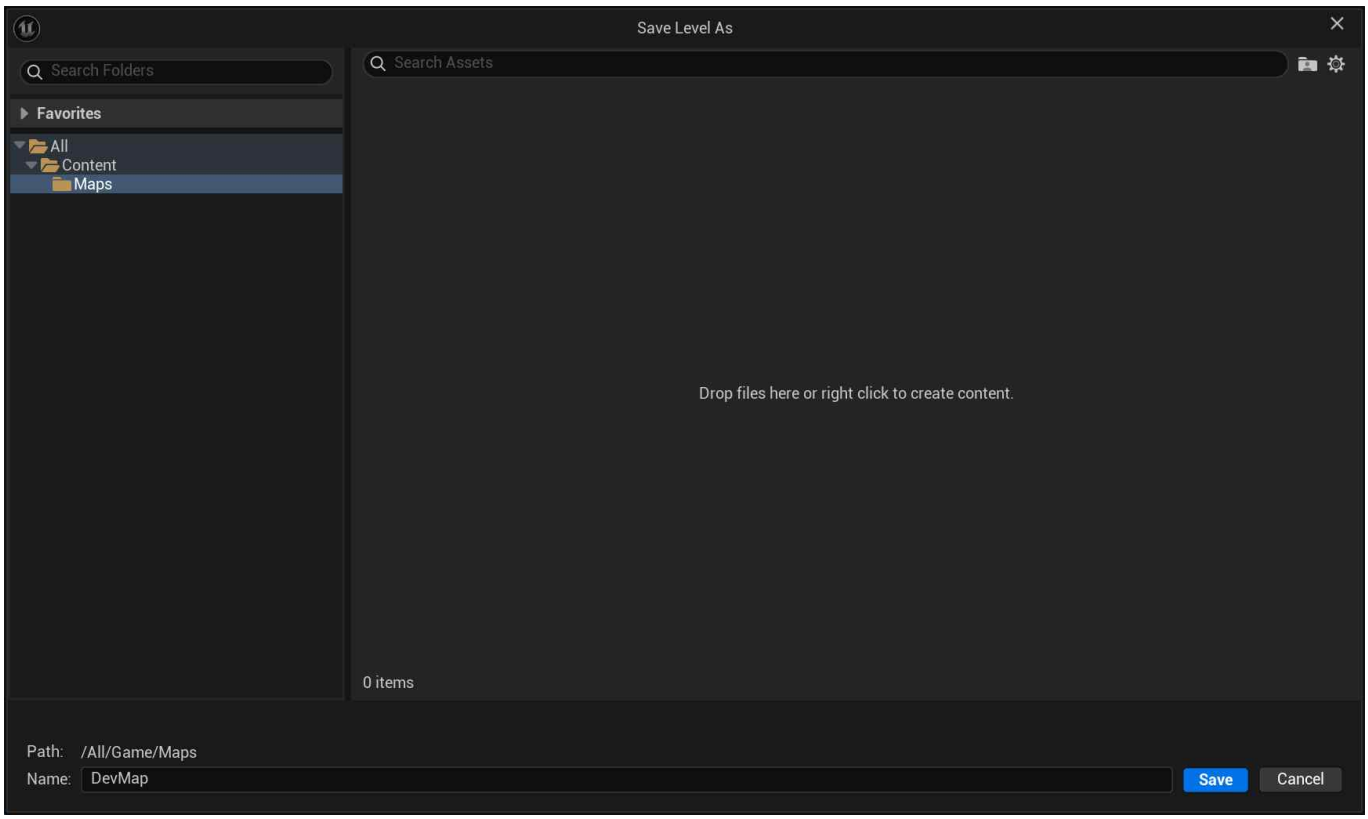




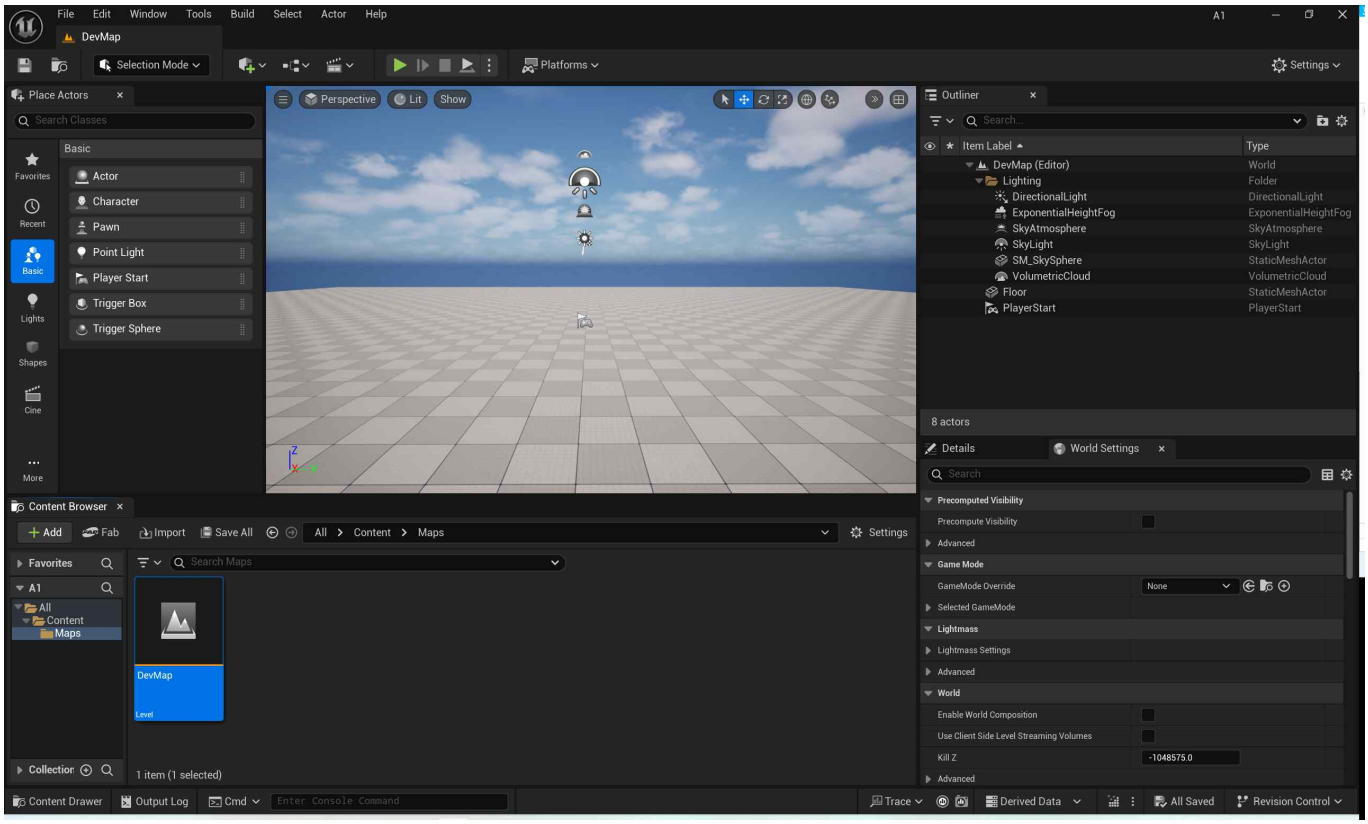
- 새롭게 추가한 맵을 저장해보자.
- 먼저 Content 폴더 하단에 새로운 폴더를 하나 추가하고 이름은 **Maps**로 설정하자.



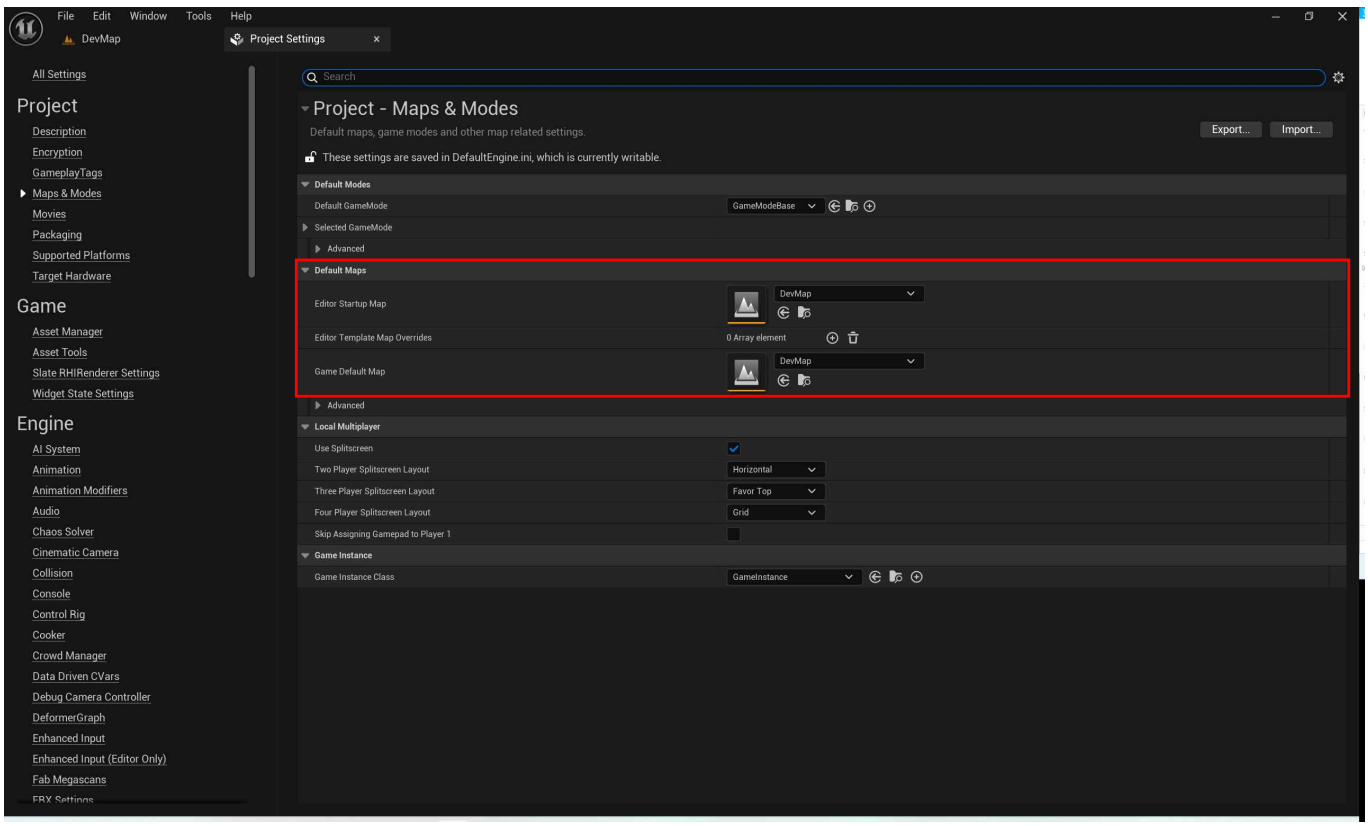
- Maps 폴더에 Basic Level을 저장하자.
- [**File**] - [**Save Current Level**]을 선택하거나 Ctrl + S 단축키를 눌러서 저장하자.
- Level의 이름은 **DevMap**으로 설정하자.



- Maps폴더에 DevMap이 추가된 것을 확인할 수 있다.



- 언리얼 에디터를 새로 시작하거나 게임을 시작했을 때, 해당 DevMap을 기본 Level로 설정해보자.
- [**Edit**] - [**Project Settings...**]를 열어서 **Project 카테고리**의 **Maps & Modes**를 선택하자.
- Project의 Maps & Modes의 **Editor Startup Map**과 **GameDefault Map**을 **DevMap**으로 설정하자.

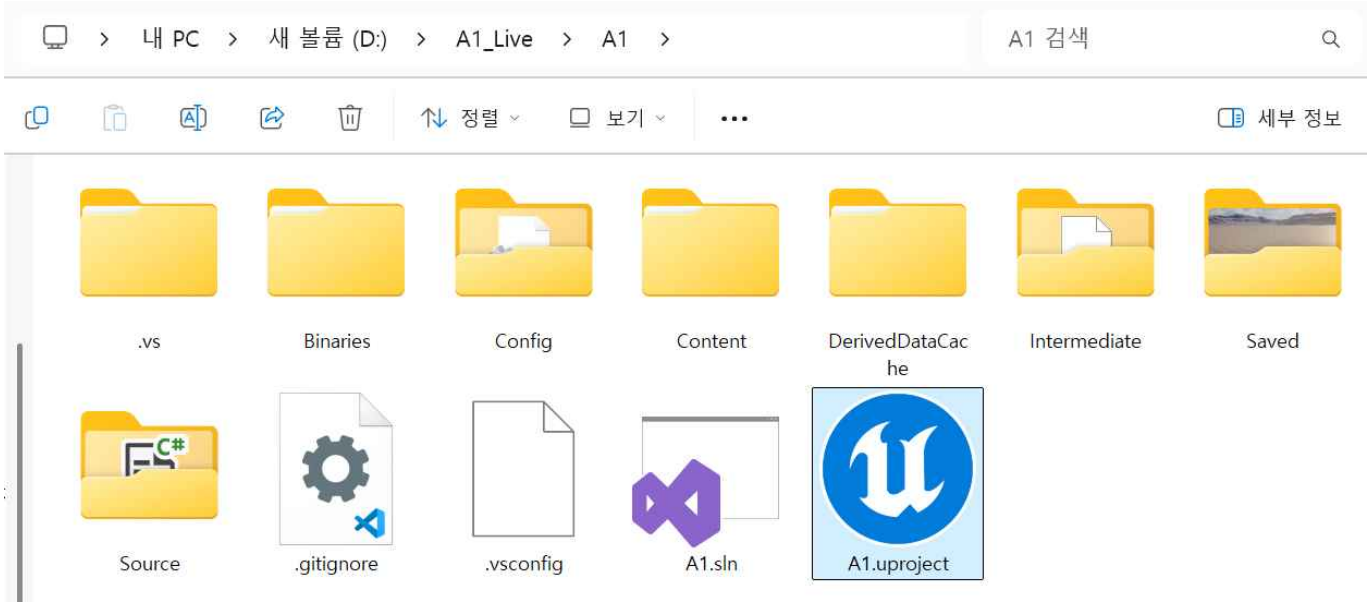


2. 언리얼 프로젝트 폴더

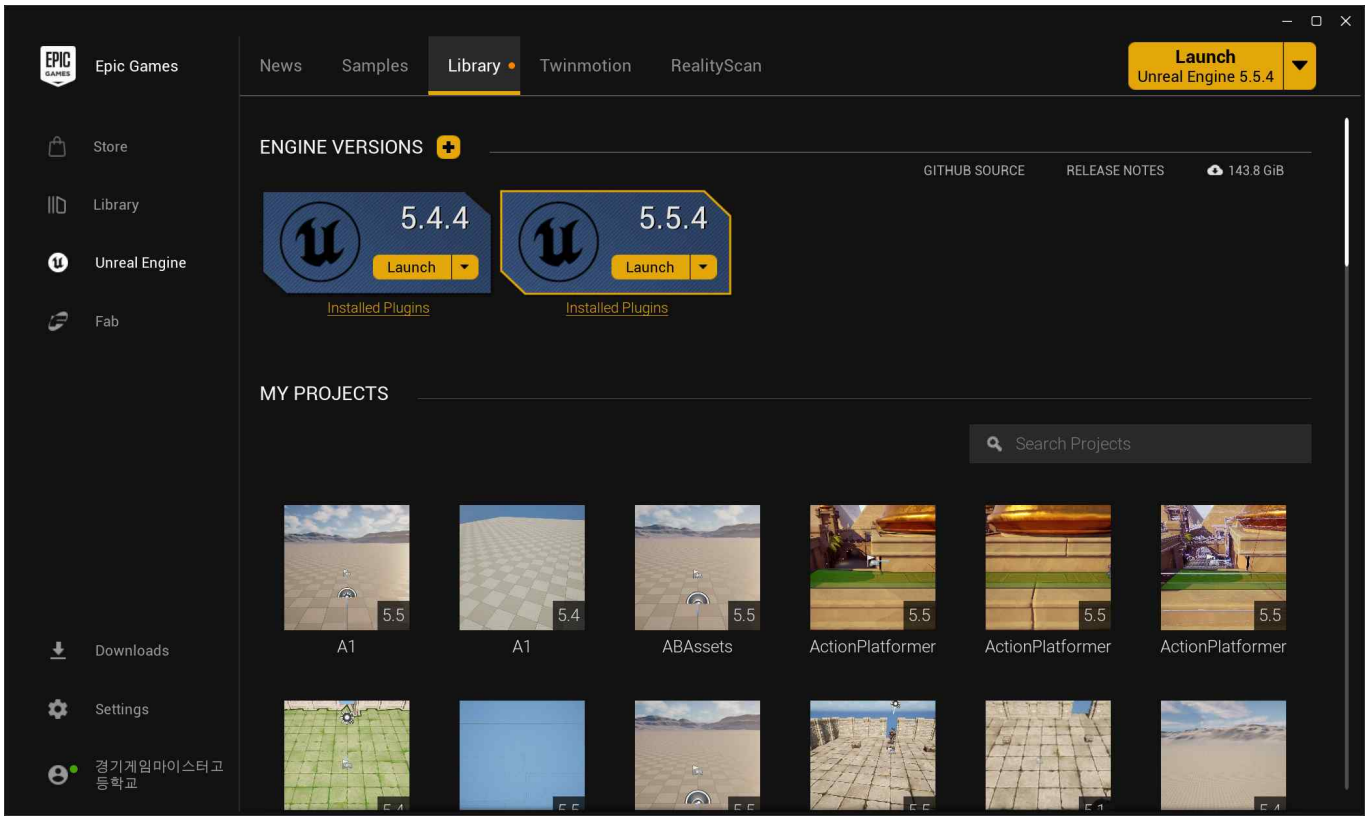
1) 언리얼 프로젝트 폴더 살펴보기

- A1_Live 폴더의 A1 폴더가 언리얼 프로젝트 폴더이다.
- 해당 폴더에서 언리얼 에디터를 켜는 방법을 살펴보자.

① uproject 파일 실행하기

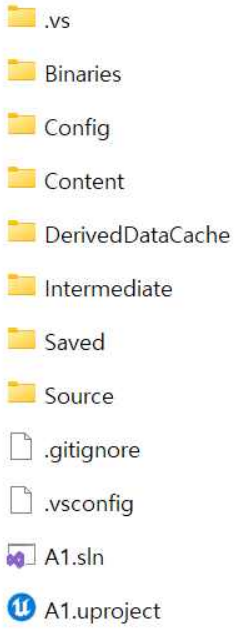


② 에픽 게임즈 런처에서 실행하기

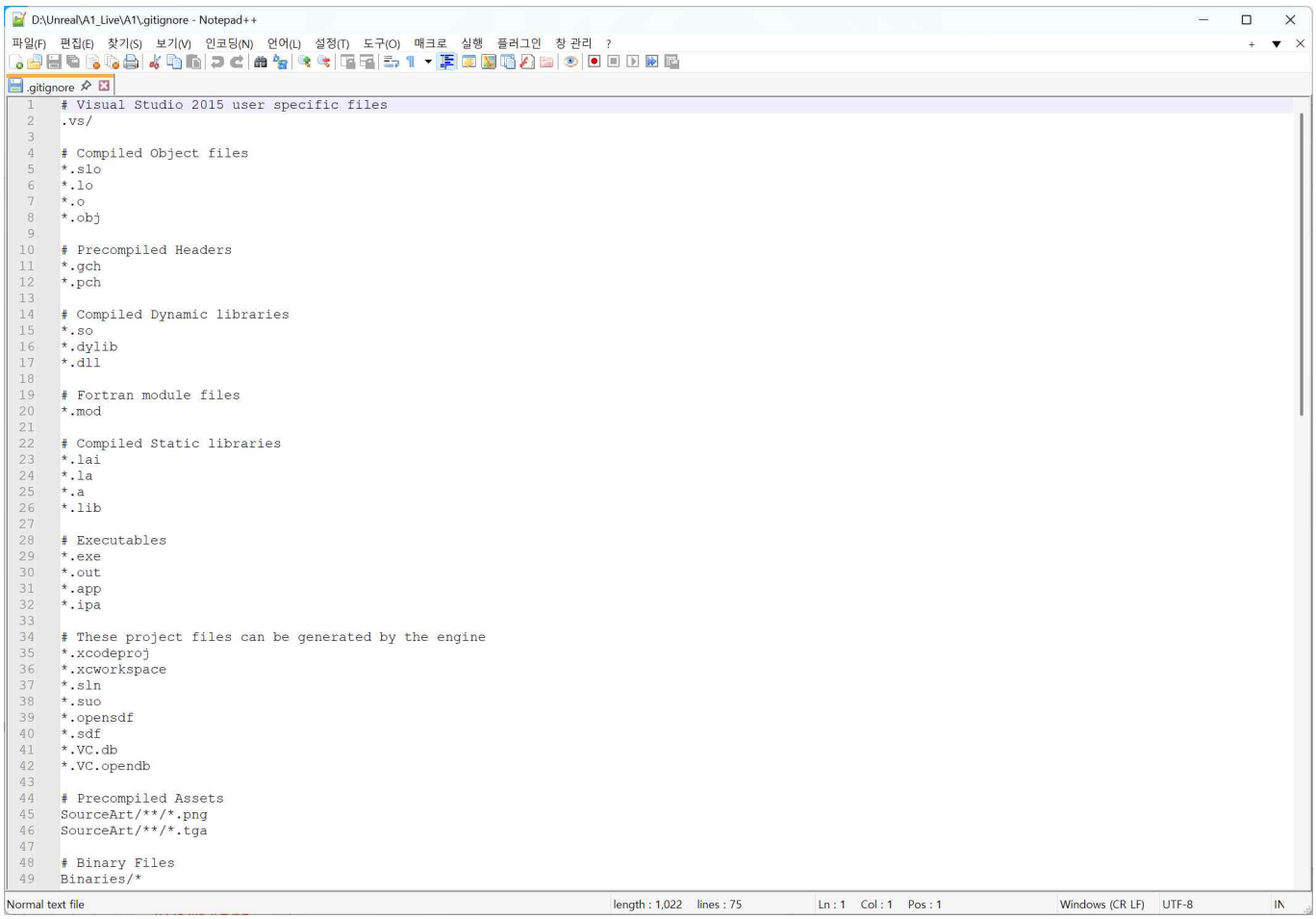


2) gitignore 살펴보기

- git에 커밋을 할 때, **A1 폴더**에 있는 모든 파일이 업로드 대상은 아니다.



- .gitignore를 열어보면 업로드 예외 파일 및 폴더를 확인할 수 있다.
- .vs, Binaries, DerivedDataCache, Intermediate, Saved 등 폴더는 언리얼 에디터와 게임 빌드를 위해서 임시로 생성되는 파일과 폴더이기 때문에 소스 컨트롤 대상이 아니다.



① 프로젝트의 중요 파일 및 폴더 (지우면 안됨)

파일 및 폴더	설명
.uproject 파일	▪ 언리얼 에디터 구동 시 해당 프로젝트에 대한 정보를 포함하고 있는 JSON 파일 언리얼 에디터를 구동시키는 연결 파일
Config 폴더	▪ 프로젝트의 언리얼 환경설정 파일을 담고 있는 폴더 ▪ 지워도 에디터 실행에는 문제가 되지 않지만 만약 프로젝트를 작업한 상태에서 지웠다면 설정해놓은 설정값들은 없어지고 언리얼 엔진에서 제공하는 기본 세팅으로 적용
Content 폴더	▪ 언리얼 에디터에서 사용하는 .uasset들이 들어있는 폴더
Source 폴더	▪ C++ 프로젝트 전용 ▪ 프로젝트의 모듈 소스와 빌드 파이프라인 C# 소스파일 등이 들어있는 폴더

② 임시 파일 및 폴더 (지워도 됨)

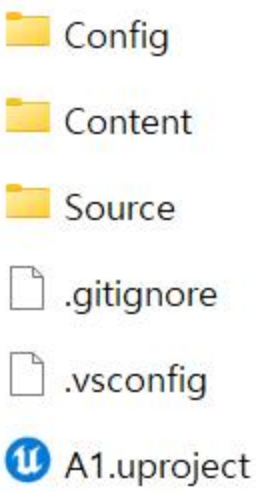
파일 및 폴더	설명
.sln	▪ Visual Studio Solution 파일 ▪ .uproject 파일 선택 후 마우스 우클릭 → 팝업 메뉴에서 'Generate Visual Studio project files'를 선택하면 생성되는 C++ 프로젝트에서만 볼 수 있는 파일 ▪ Visual Studio에서 작업하기 위해서는 필요함
.vs 폴더	▪ C++ 프로젝트에서 .sln 파일과 마찬가지로 Generate 과정에서 생성되는 폴더 ▪ 폴더 자체는 숨김 설정되어 있으므로 윈도우 탐색기의 폴더 옵션에서 숨김 파일 보기를 설정해야만 볼 수 있음 ▪ 해당 프로젝트를 Visual Studio에서 실행할 때 초기화 및 데이터 구조 등을 기록하여 추후 솔루션 파일을 실행할 때 불러오기 과정을 단축하기 위한 용도
Binaries 폴더	▪ C++ 컴파일 과정을 거쳐 생성된 프로젝트에 대한 에디터 전용 DLL 파일들이 있는 폴더 ▪ 이 폴더가 없으면 .uproject를 실행시킬 때마다 에디터 실행 시 'Missing <프로젝트명> Modules' 메시지 팝업을 띄움. Yes를 누르면 컴파일 과정을 거쳐 다시 파일과 폴더를 생성하고, 에디터가 실행됨
DerivedDataCache 폴더	▪ 주로 Assets의 쉐이더 컴파일 한 정보가 저장 됨 ▪ DDC 폴더를 삭제 한 후 편집기를 실행하면 쉐이더 컴파일이 진행됨 ▪ 프로젝트를 그대로 전달할 때 DDC 포함하면 쉐이더 컴파일이 일어나지 않음
Intermediate 폴더	▪ 언리얼 엔진 라이브러리 및 해당 프로젝트 소스 코드에 대한 빌드 과정을 거치면 생성되는 파일들이 들어있는 폴더 ▪ [언리얼 문서 - 디렉터리 구조 설명 참조] 엔진이나 게임 빌드 도중 생성된 임시 파일이 들어 있으며, 게임 디렉터리에서 셰이더는 Intermediate 디렉터리에 저장됨

③ 기타 폴더(지워도 되지만 지우지 않는게 유리)

파일 및 폴더	설명
Saved 폴더	▪ 프로젝트 작업 시 임시로 저장되는 파일들이 보관되는 장소로, 자동 저장, 스크린샷, 빌드 파일, 백업 임시 파일, 로그, SaveGames 등등 많은 데이터가 축적되는 장소 ▪ 언리얼 에디터 사용 시 작업의 효율성을 높이거나 복구작업, 문제 확인 등의 활용성이 높은 폴더 ▪ 다만 작업이 길어지고 패키징까지 하면 용량을 꽤 많이 차지함 (패키징 백업본 + Cooking 과정에서의 엔진 및 프로젝트 리소스 Cooked 데이터 등등)

3) 필요 없는 폴더 제거한 후 다시 생성하기

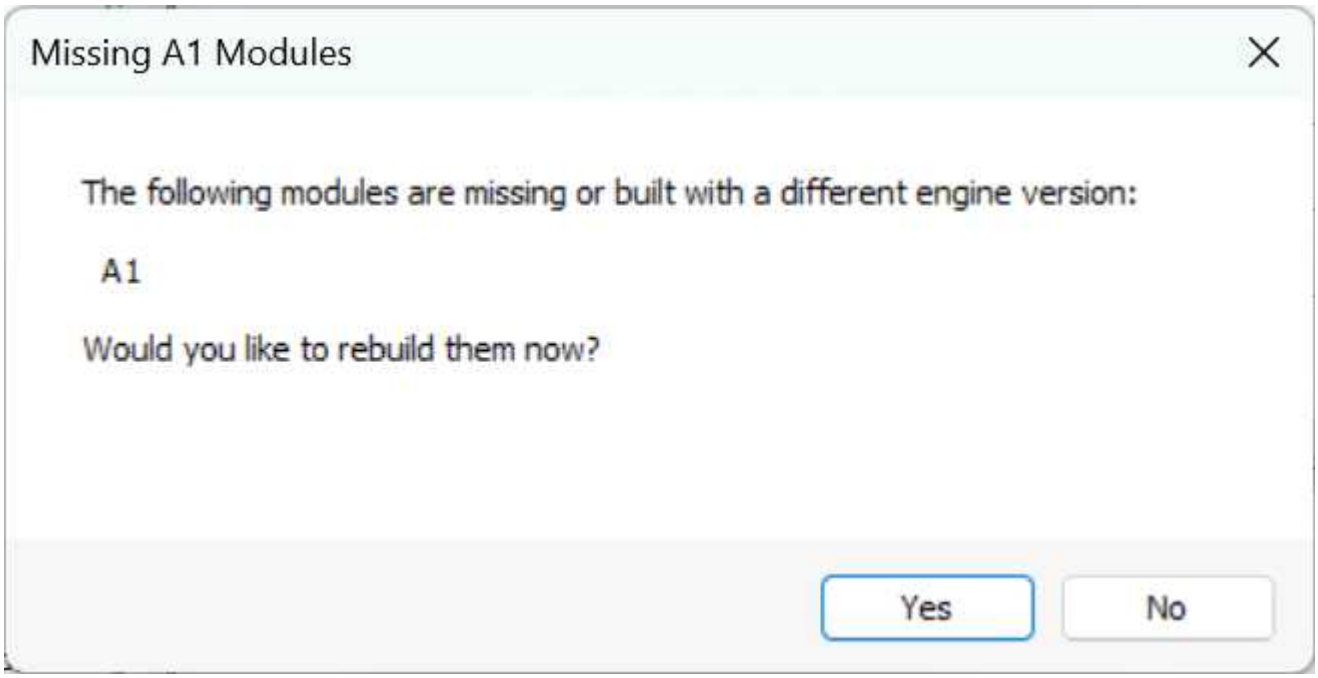
- 언리얼 프로젝트를 실행하기 위해서 반드시 필요한 파일만 남기고 제거해 보도록 하자.
- 프로젝트를 세트로 만들어 배포할 때는 아래의 데이터만 포함시켜서 전달하면 된다.



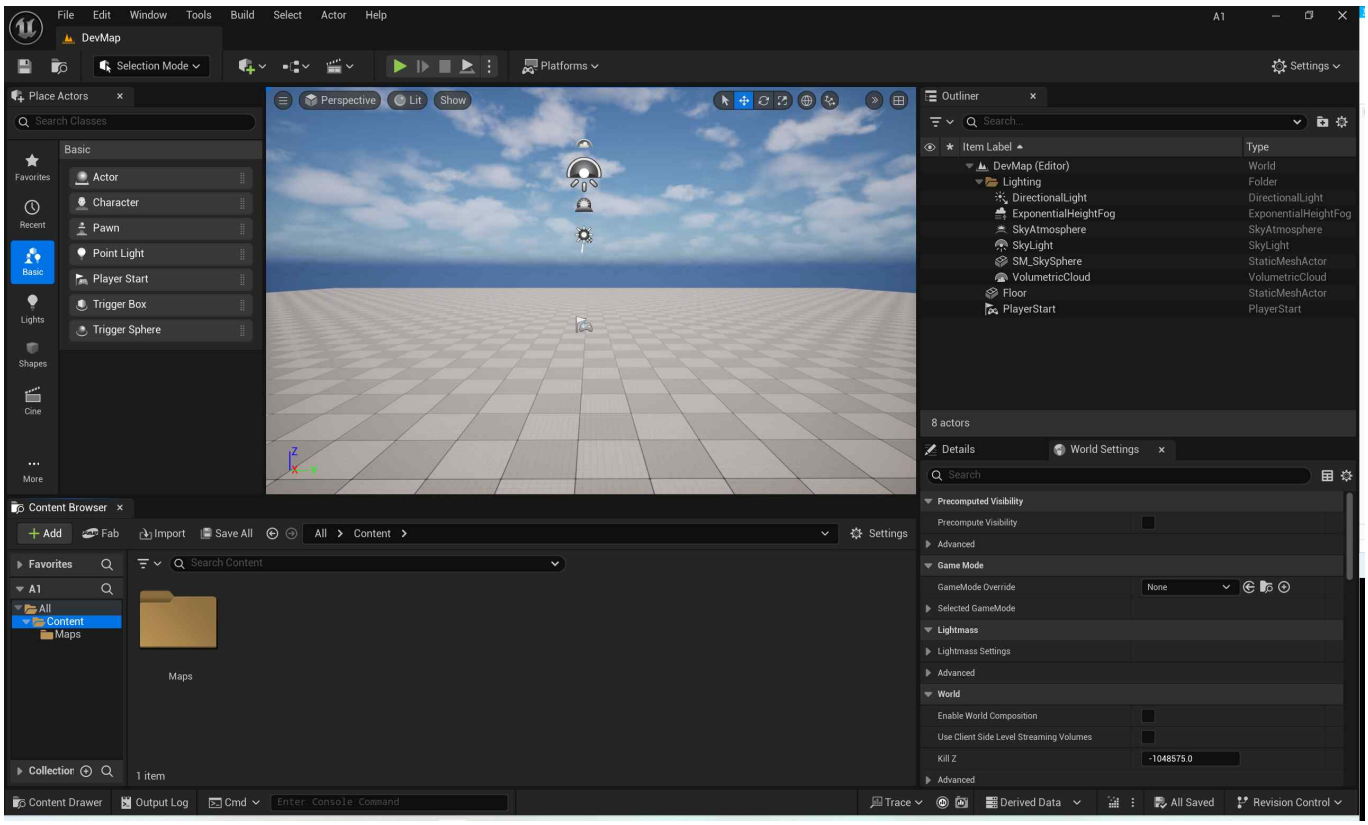
- 삭제한 임시 파일들을 복원하는 방법을 살펴보자.
- A1.uproject 파일을 선택한 후, 마우스 오른쪽 메뉴에서 추가 옵션 메뉴에서 Generate Visual Studio project files를 선택해 주어야 한다. 다시 복원 된다.

4) 언리얼 프로젝트 실행

- 언리얼 프로젝트를 다시 실행하기 위해서 A1.uproject를 실행하면 다음과 같은 에러 메시지가 나타난다.
- 임시 파일이 다 사라졌기 때문에, VS로 빌드한 모듈이 사라져서 그렇다.
- Yes를 선택하여 현재의 C++코드를 기반으로 모듈을 다시 생성하도록 하자.



- 언리얼 에디터가 켜지면서 이전 설정과 똑같은 DevMap이 열리는 것을 확인할 수 있다.

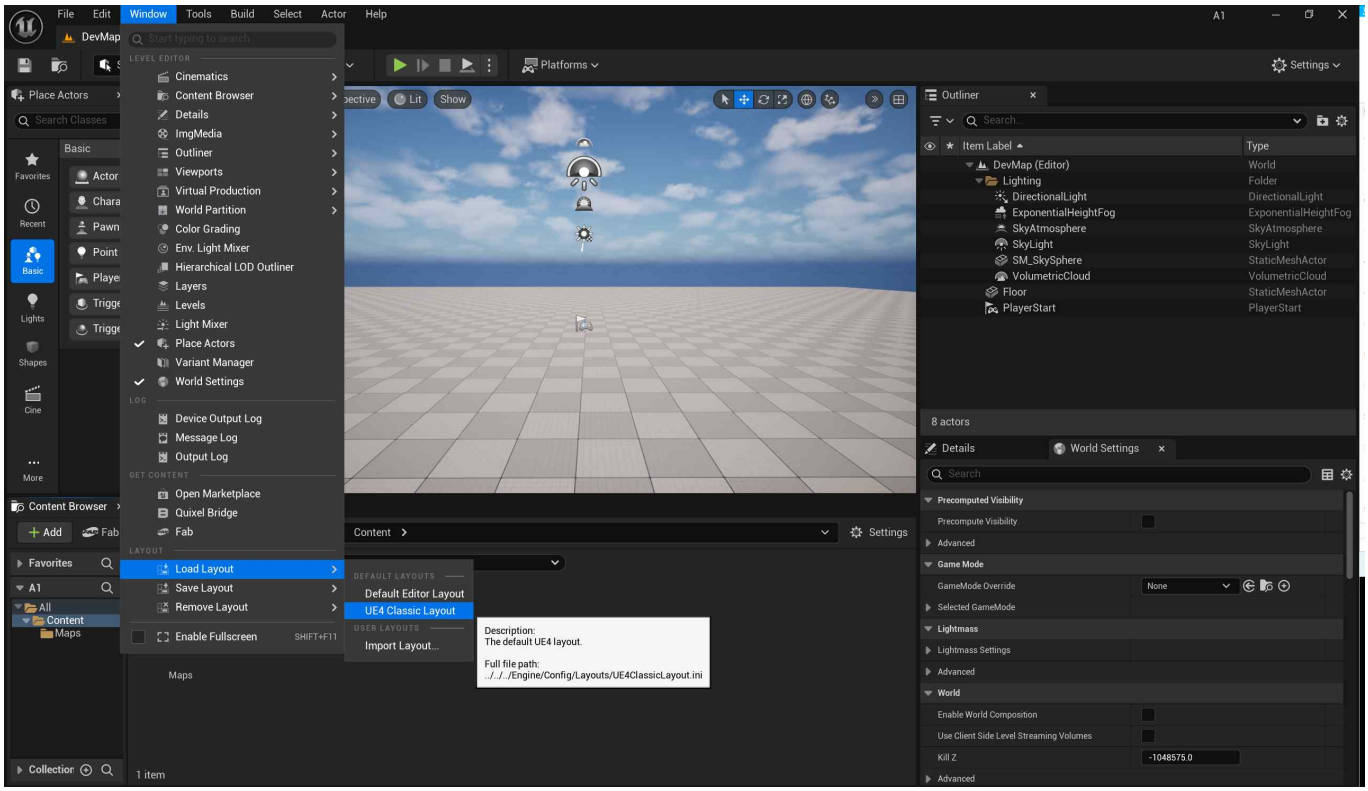


3. 언리얼 에디터

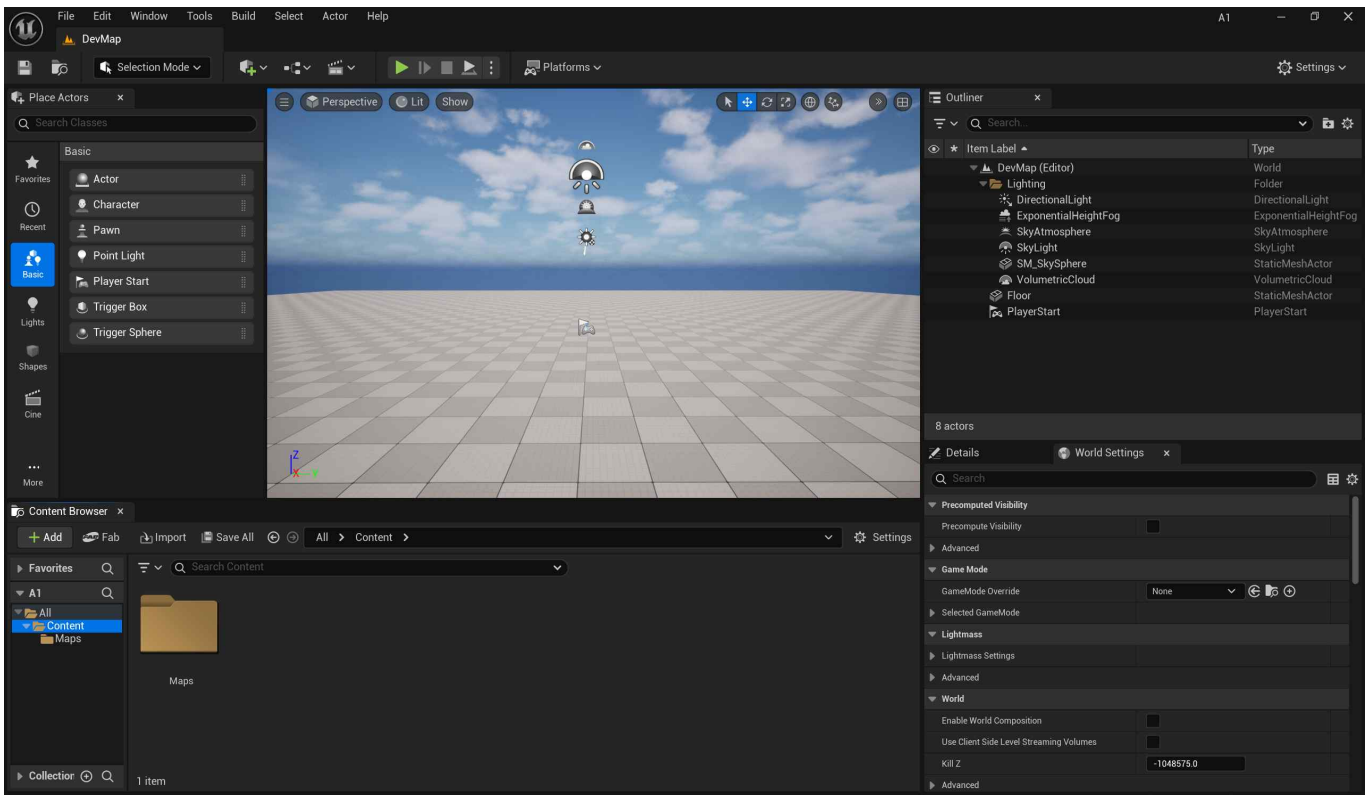
1) 언리얼 에디터 레이아웃

- 언리얼 에디터의 레이아웃을 설정해보자.

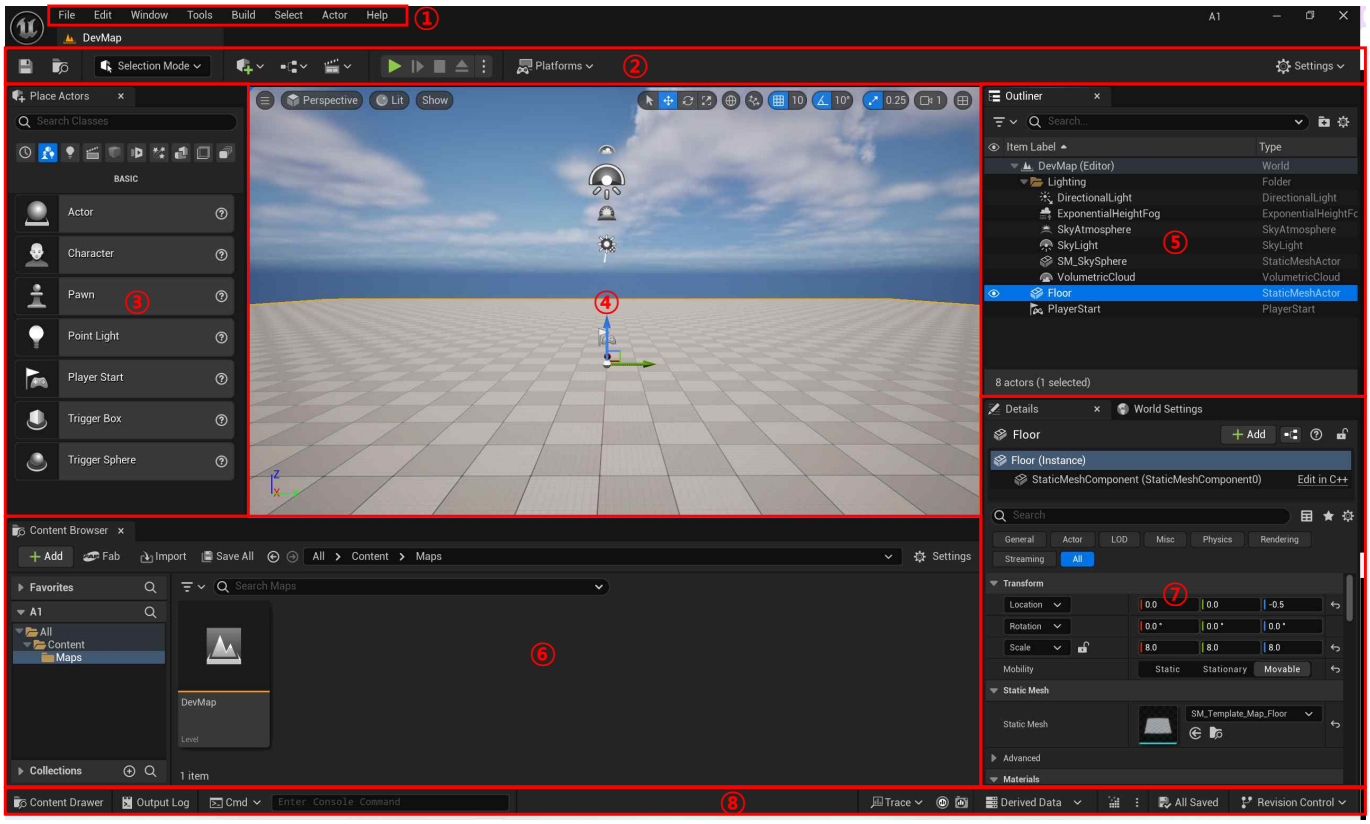
- [**Window**] - [**Load Layout**] - [**UE4 Classic Layout**]을 설정하도록 하자.



- 해당 화면이 가장 기본적인 레이아웃이다.



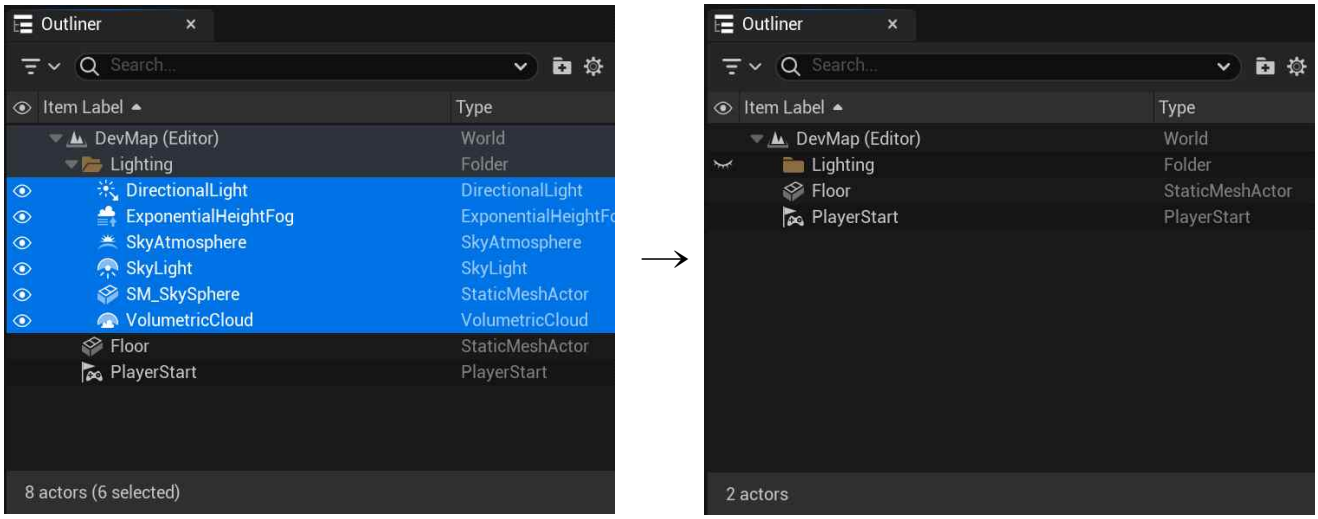
2) 기본 인터페이스



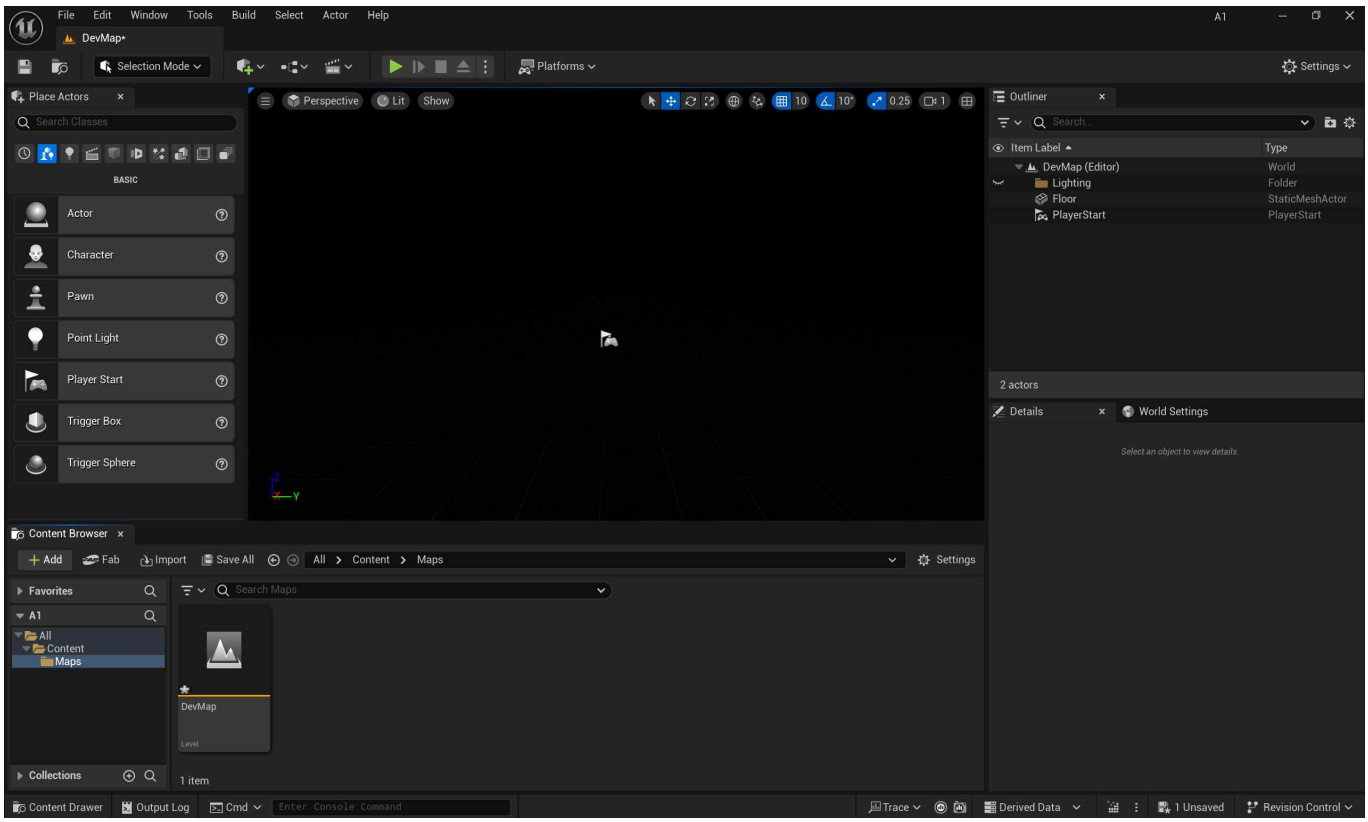
인터페이스	설명
① 메뉴 바	▪ 에디터의 메뉴 바 는 Windows 애플리케이션을 사용해 본 사람이라면 누구나 익숙할 것입니다. 에디터에서 레벨을 작업할 때 사용하는 일반 툴과 명령어를 여기서 이용할 수 있습니다.
② 툴바	▪ 툴바 패널에는 명령 그룹이 표시되어 자주 사용하는 툴과 작업에 빠르게 액세스할 수 있습니다.
③ 액터 배치	▪ "액터(Actors)"는 게임 개발이나 3D 시뮬레이션에서 사용되는 용어로, 레벨에 배치할 수 있는 모든 개체를 의미한다. ▪ 액터는 게임 환경을 만드는 데 필수적인 요소들로, 정적 메시(고정된 3D 모델), 사운드(소리 효과), 카메라, 플레이어 캐릭터(게임에서 사용자가 조종하는 캐릭터) 등을 포함한다.
④ 뷰포트	▪ 뷰포트 패널은 언리얼 엔진에서 제작 중인 월드를 보여주는 창입니다.
⑤ 아웃라이너	▪ 아웃라이너(Outliner) 패널은 씬 내 모든 액터를 계층형 트리 뷰로 표시합니다. 아웃라이너 에서 바로 액터를 선택하고 수정할 수 있습니다. ▪ 정보(Info) 드롭다운 메뉴를 사용하면 레벨, 레이어, ID 이름을 보여주는 추가 열을 표시할 수 있습니다.
⑥ 콘텐츠 브라우저	▪ 콘텐츠 브라우저(Content Browser) 는 언리얼 프로젝트 내의 콘텐츠 에셋을 생성, 임포트, 구성, 확인 및 관리하는 언리얼 에디터의 주요 영역입니다. ▪ 또한 콘텐츠 브라우저를 사용하여 콘텐츠 폴더를 관리하고 다음과 같은 특정 에셋 작업을 수행할 수도 있습니다.
⑦ 디테일	▪ 디테일 패널에는 뷰포트의 현재 선택한 항목에 대한 정보와 유틸리티, 기능이 포함되어 있습니다. ▪ 여기에는 액터의 이동, 회전, 스케일 조절을 위한 트랜스폼 편집 박스가 포함되어 있으며, 선택한 액터의 편집가능 프로퍼티가 전부 표시되고, 뷰포트에서 선택한 액터 유형에 맞는 추가 편집 기능을 쉽게 이용할 수 있습니다.
⑧ 하단 툴바	▪ 명령 콘솔(Command Console), 출력 로그(Output Log), 파생 데이터(Derived Data) 기능의 단축키가 포함되어 있습니다. 소스 컨트롤 상태도 표시됩니다.

3) 환경 설정

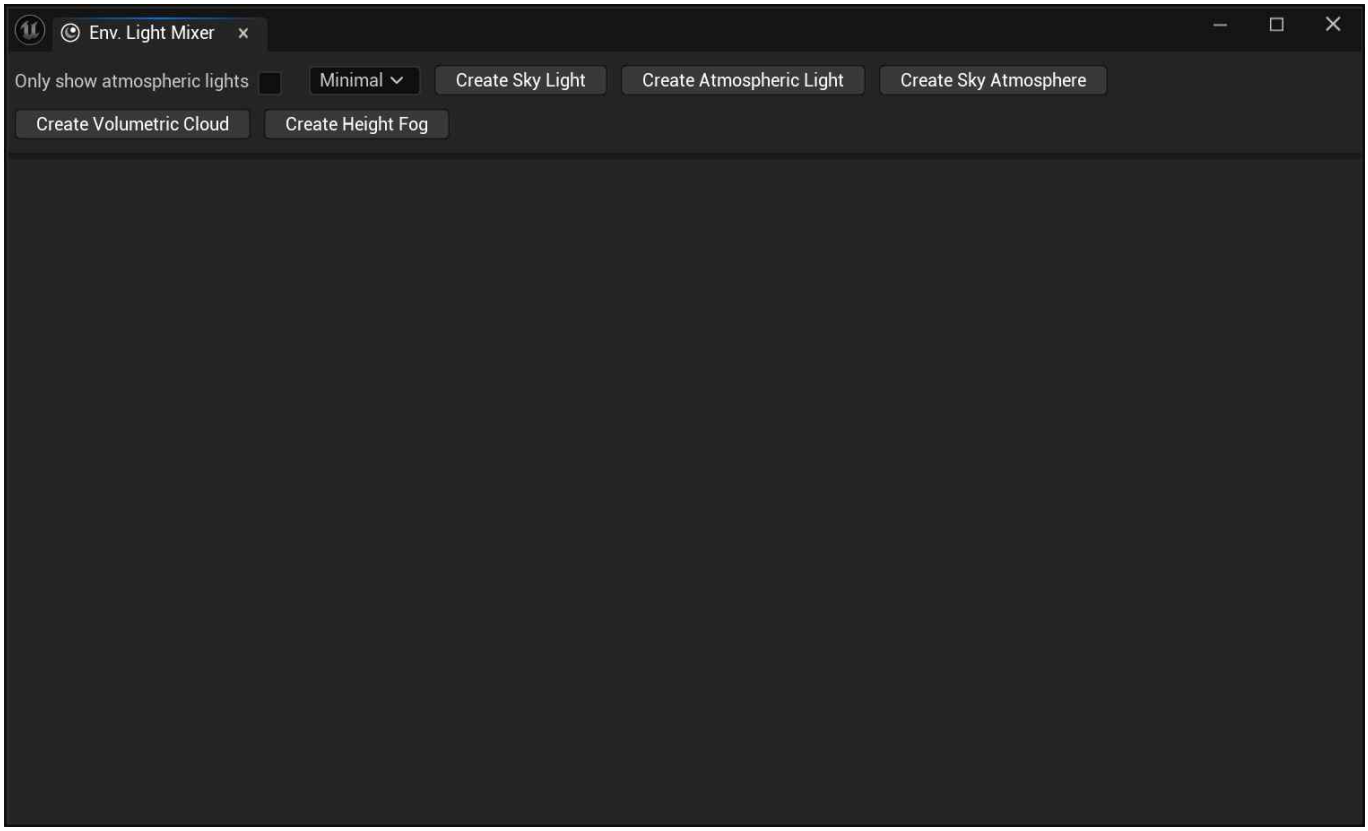
- 기존에 Level에 배치된 Lighting 정보를 삭제하고 다시 배치해보자.



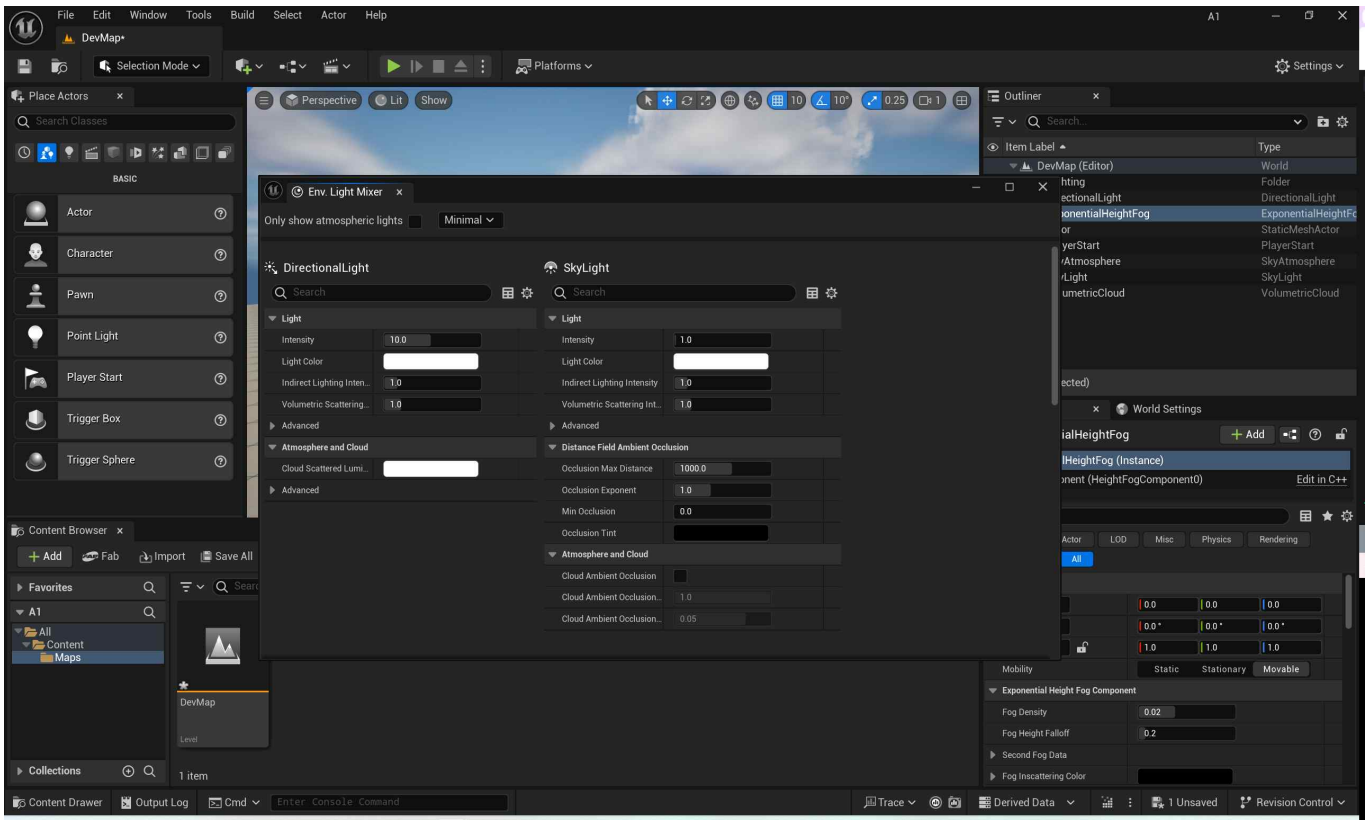
- 조명이 사라져서 화면이 어두워 진다.



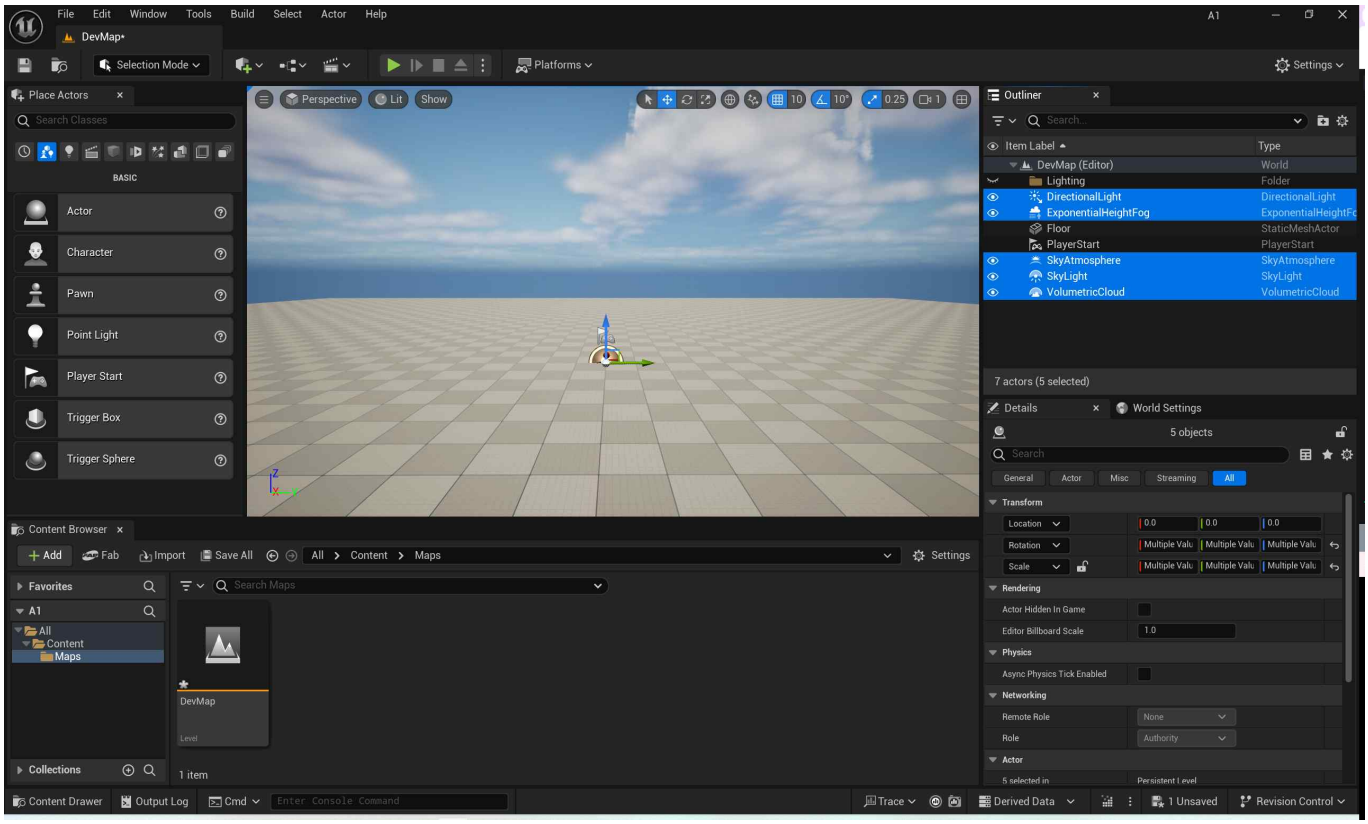
- Level에 사용할 라이트를 설정해보자.
- [**Window**] - [**Env Light Mixer**]를 실행하자.



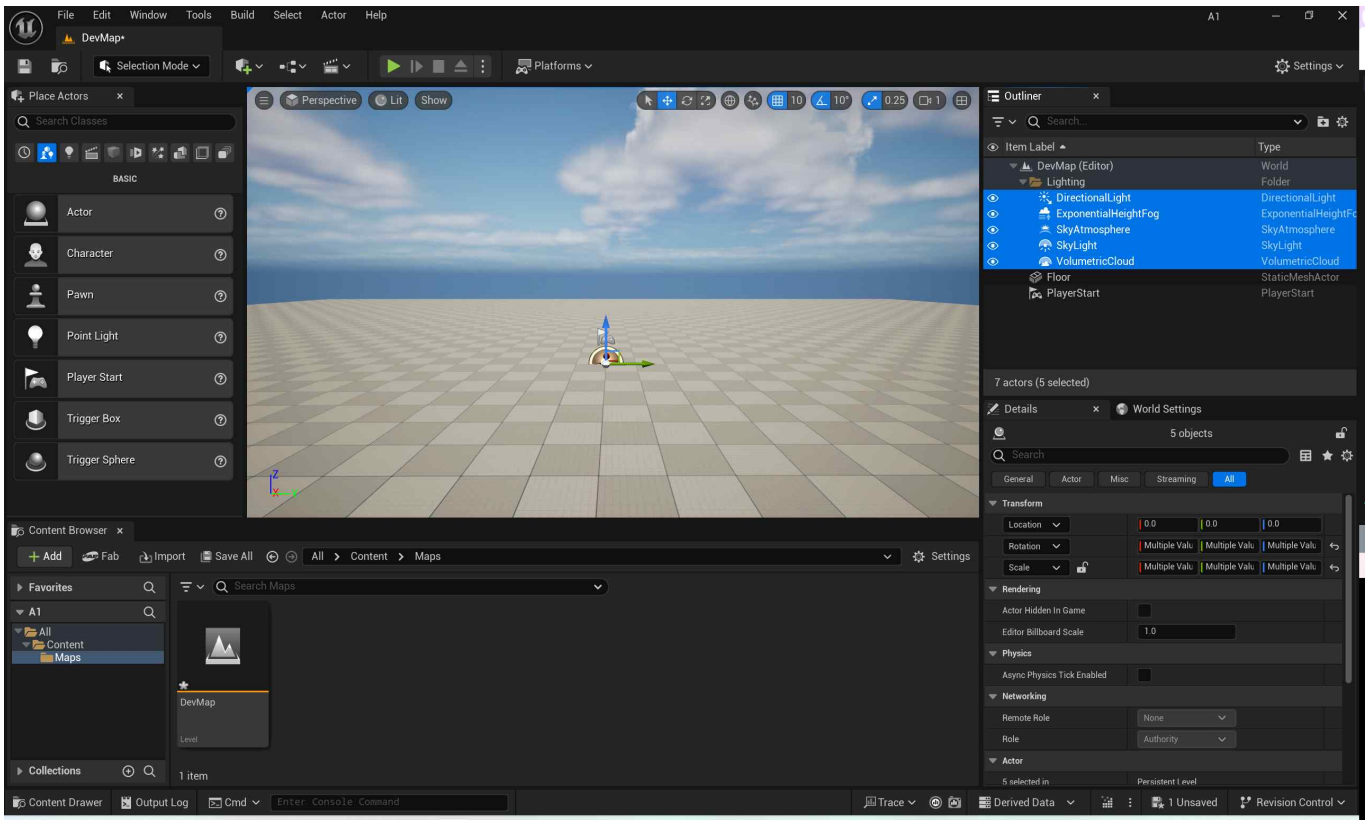
- 라이트의 프리셋이다.
- 모든 라이트와 스카이 박스 및 안개까지 추가해보자.
- 다시 뷰포트에 환경 설정이 복원된 것을 확인할 수 있다.



- 추가한 라이트를 다시 Lighting 폴더 하단으로 이동시키자.

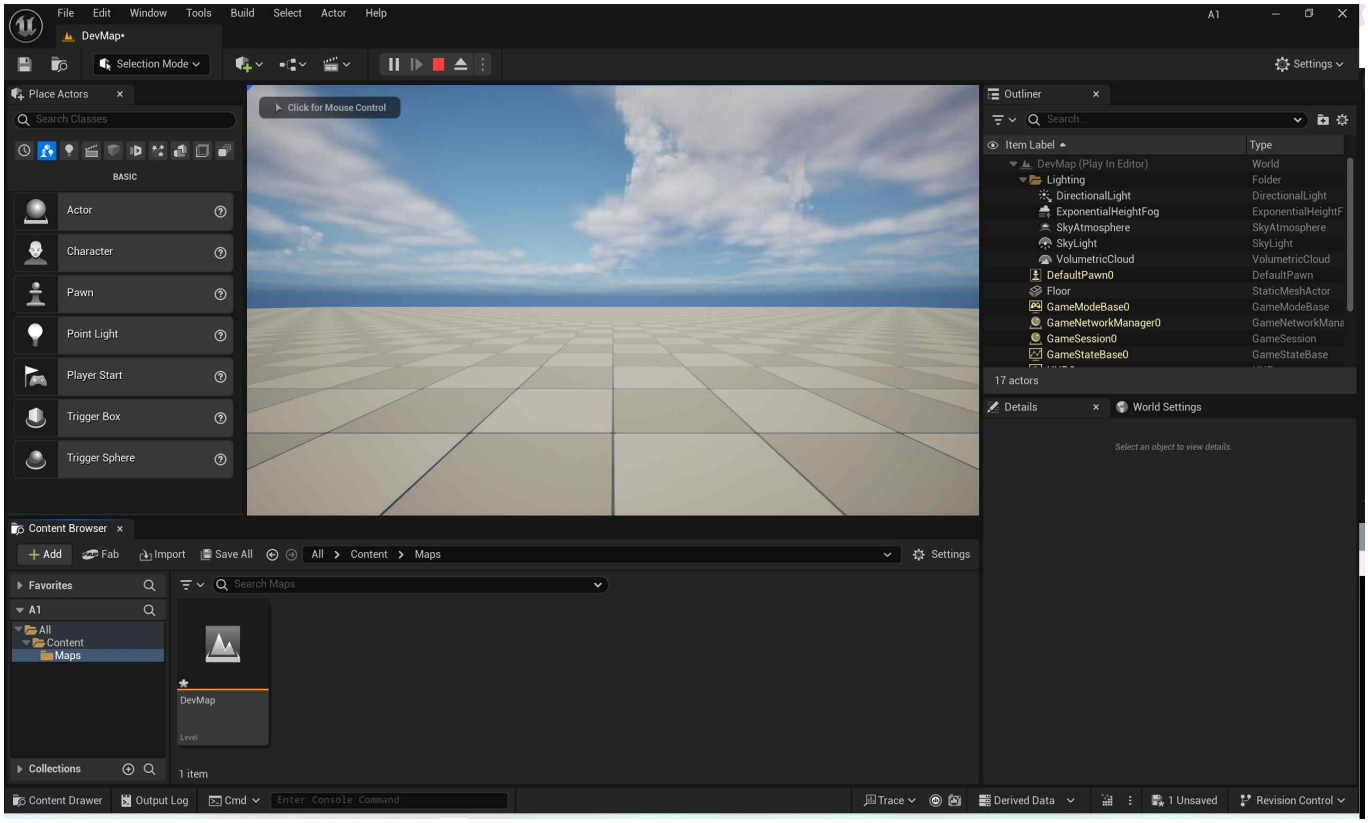


- Lighting 폴더에 정리된 것을 확인할 수 있다.

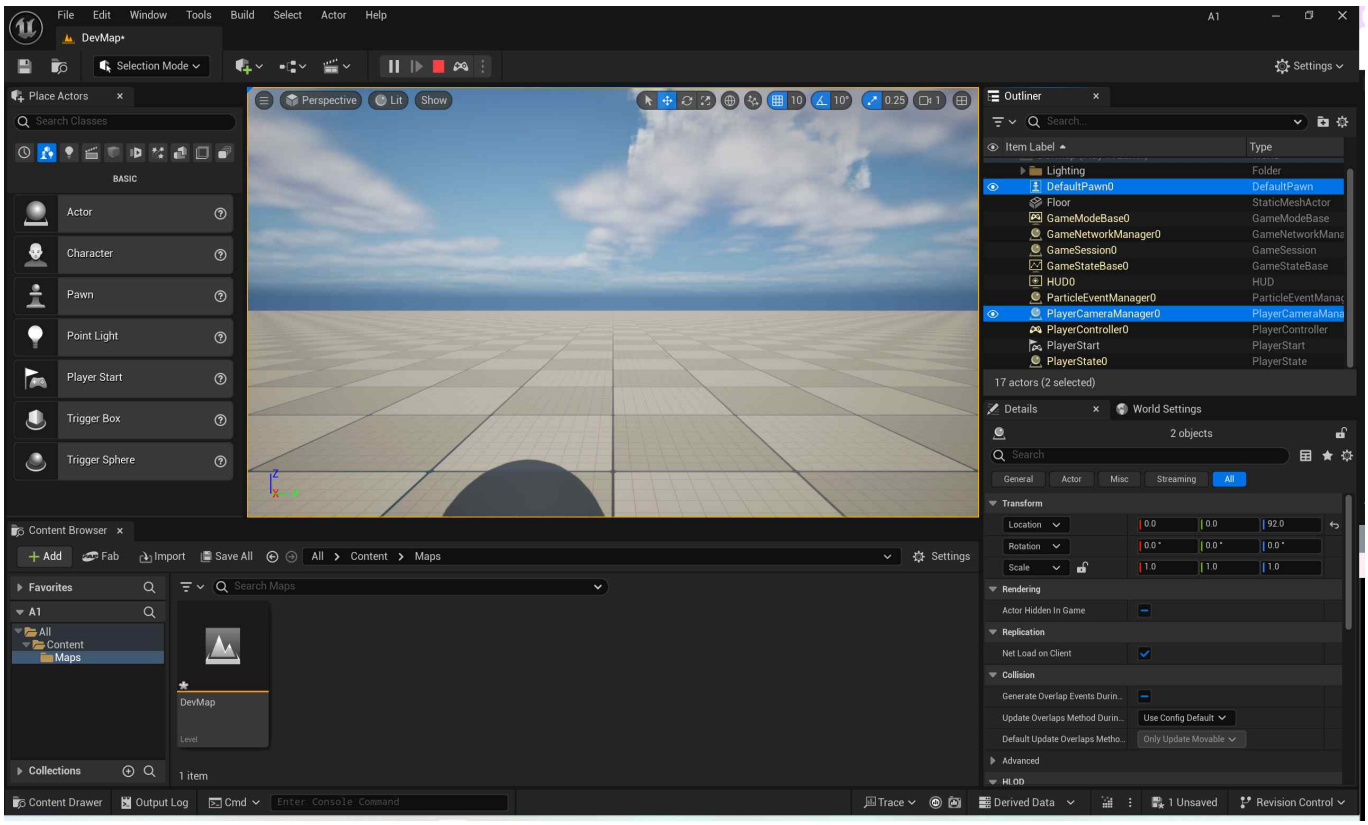


4) 게임 실행하기

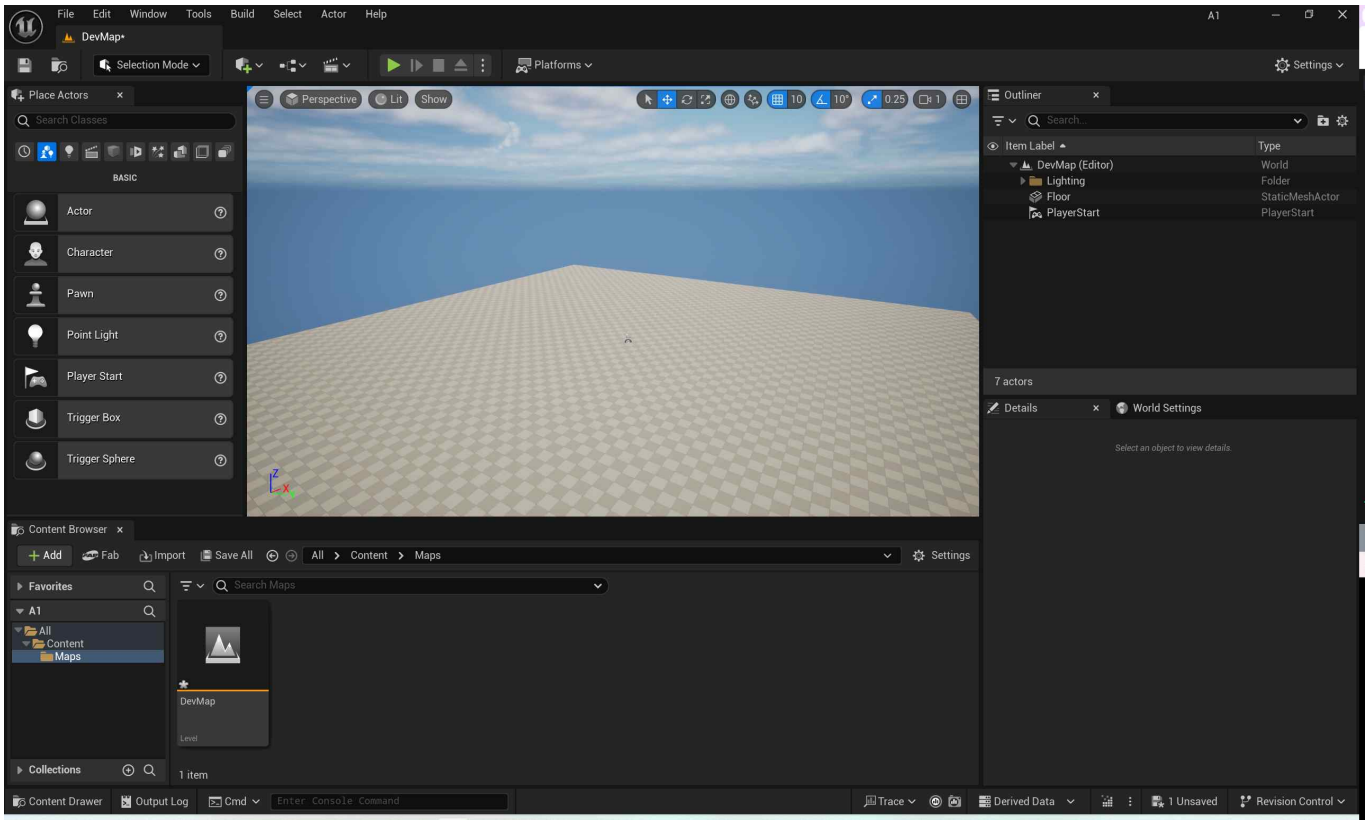
- 언리얼 에디터에서 게임을 실행 해보자. 재생 버튼(▶)을 눌러보자.



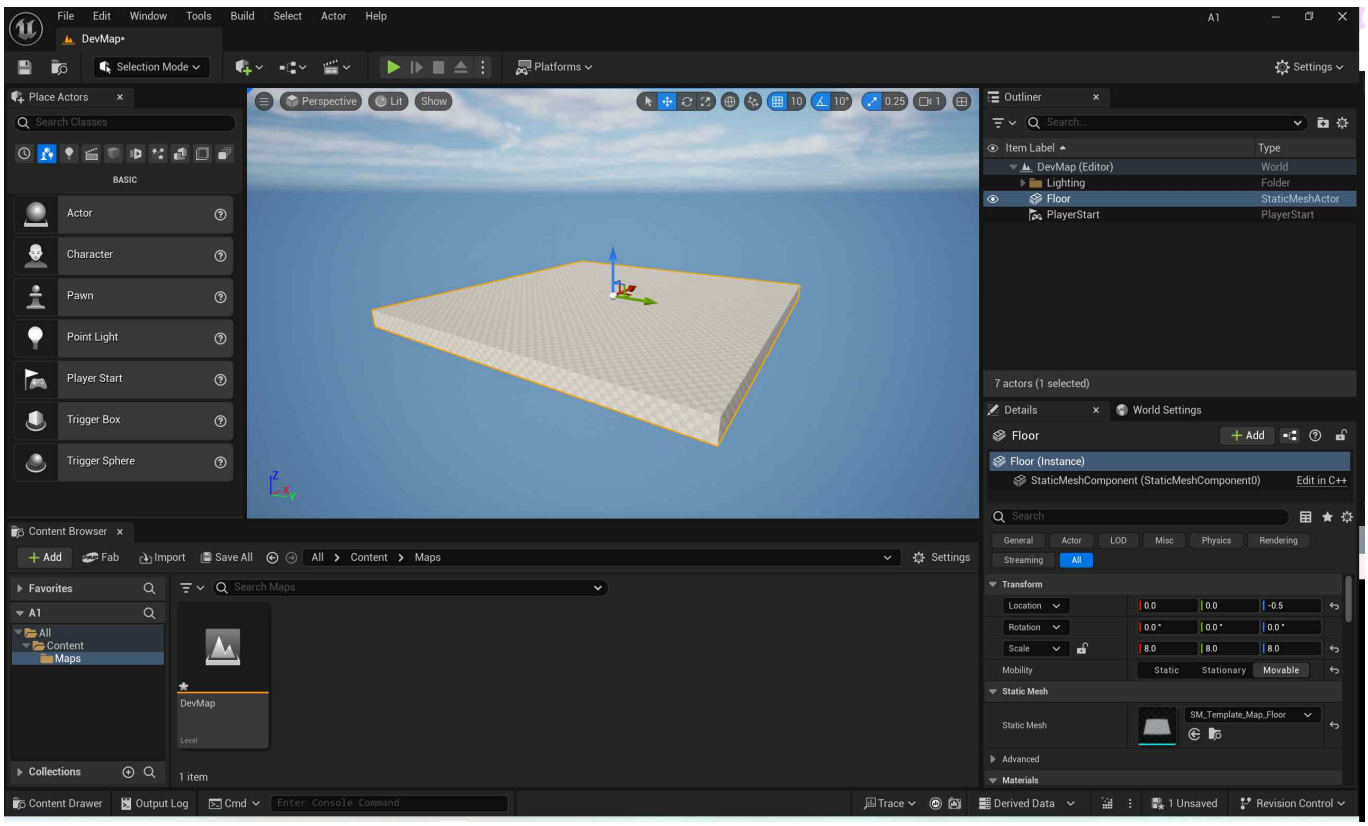
- 게임을 실행하면 언리얼 엔진에서 기본 값으로 설정한 캐릭터와 카메라가 호출되면서 게임 화면이 보이게 되고 이동이 가능해진다.
- 이때, 언리얼 에디터상에서 마우스 커서가 필요하면 단축키 F8을 누르면 된다.



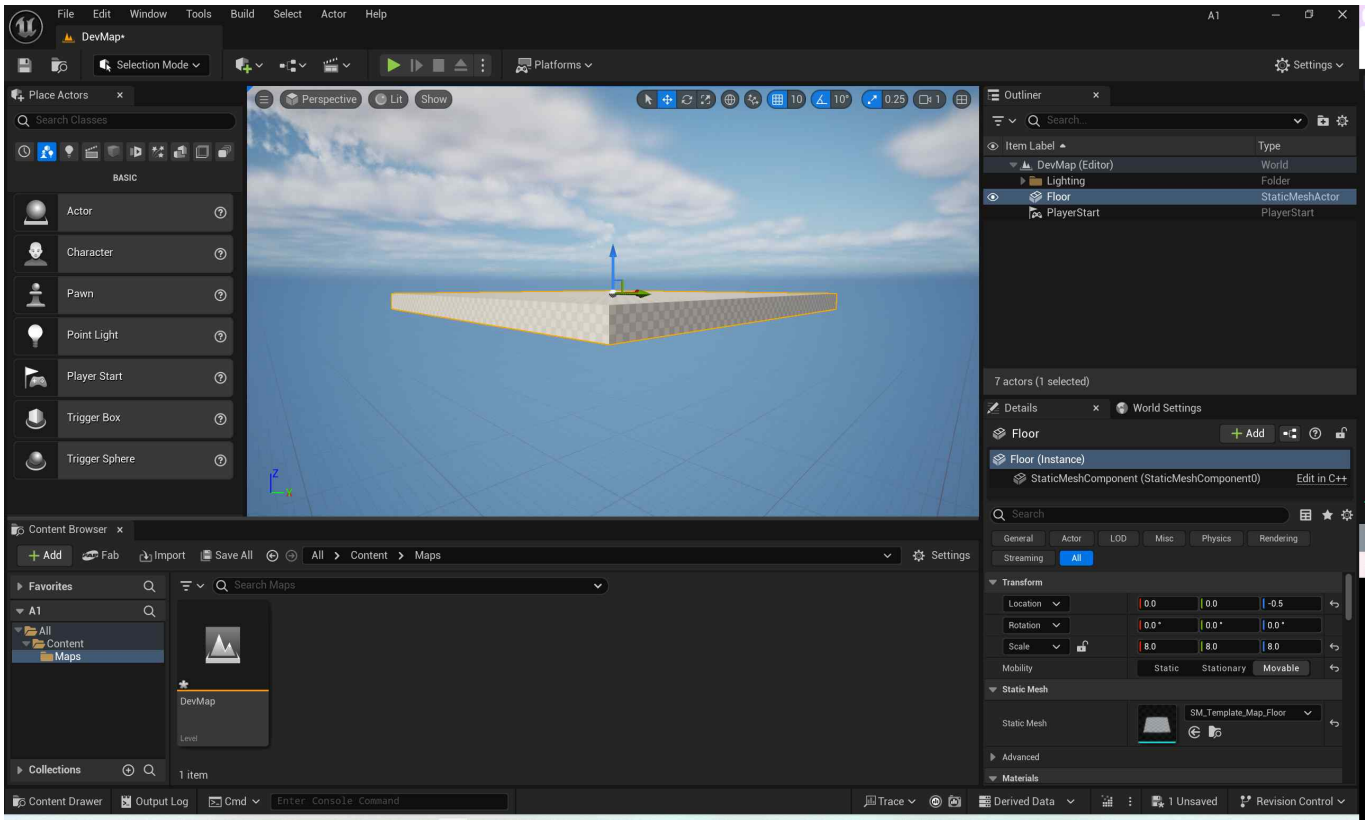
- 뷰포트 제어를 해보자.
- WASD를 이용하여 에디터상의 카메라의 위치를 이동시킬 수 있고, 마우스 오른쪽 버튼을 누른채로 이동시키면 카메라의 회전이 가능하다.



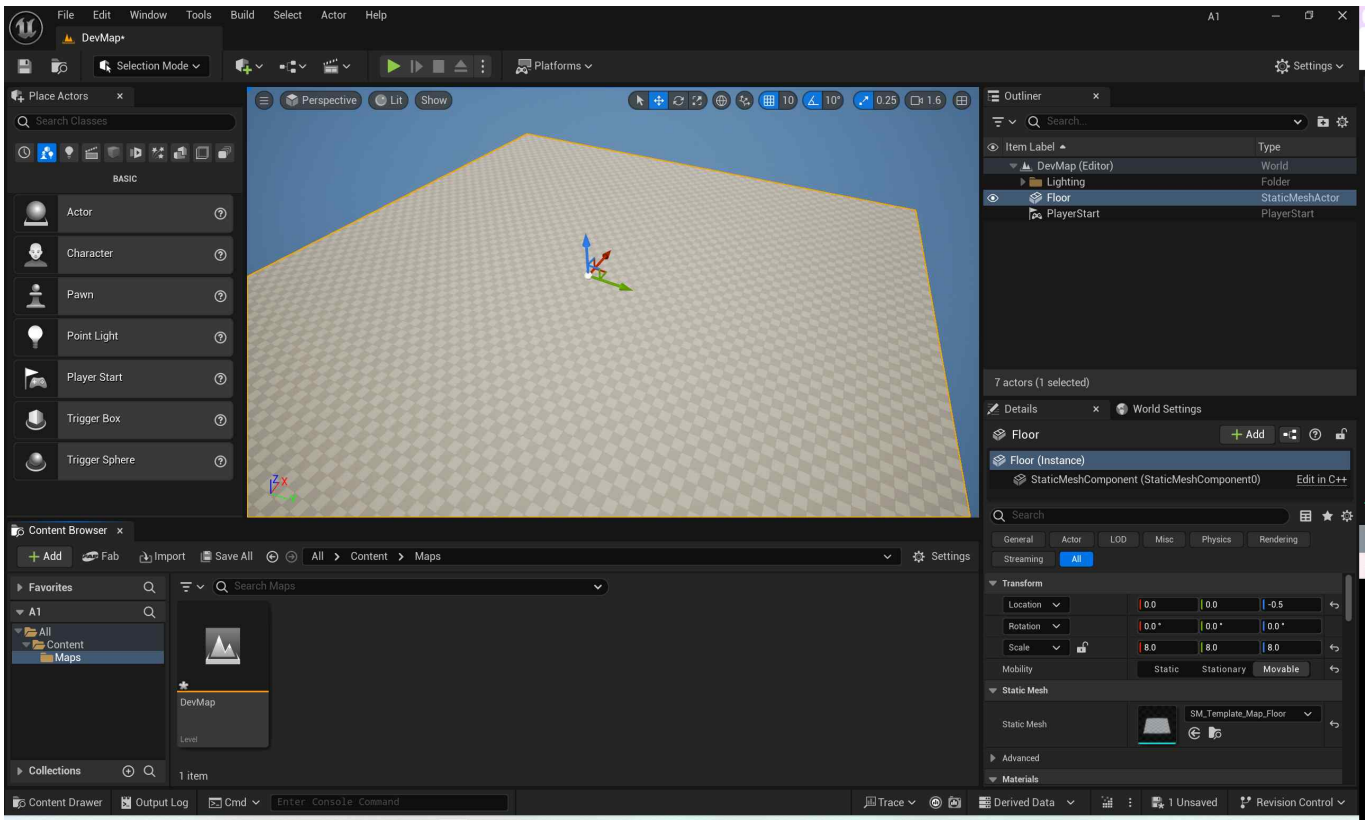
- 특정 오브젝트를 선택한 후, F키를 누르면 오브젝트를 포커싱하여 시야거리 안에 들어오게 한다.



- Alt 키를 누른 채, 마우스 왼쪽 버튼을 누르고 이동시키면 화면의 시점이 변경된다.

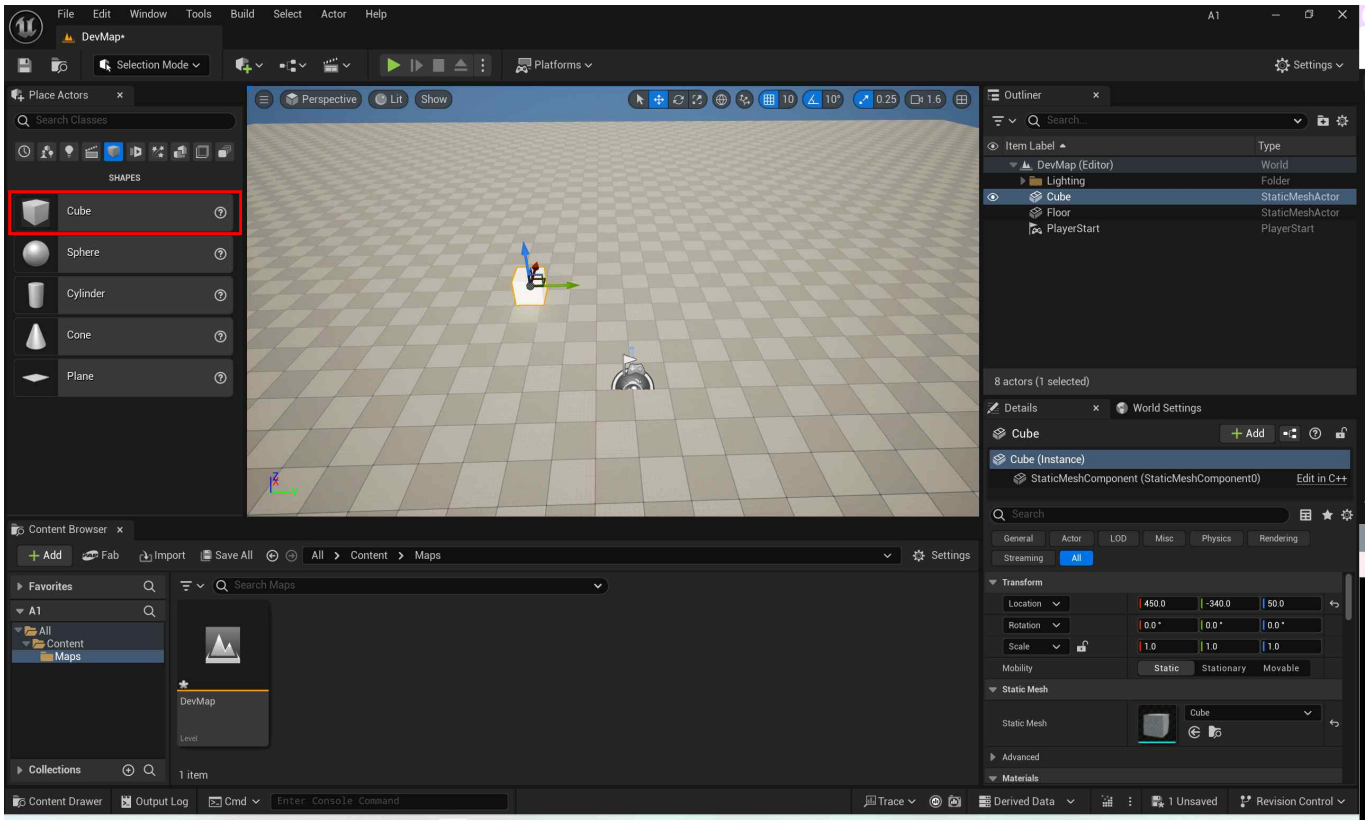


- 마우스 휠을 이용하여 확대, 축소를 하면 된다.



5) 액터 추가하기

- Level에 추가되는 오브젝트는 액터이다.
- 언리얼 엔진에서는 액터를 상속받은 오브젝트를 레벨에 배치할 수 있다.
- Place Actors의 Cube를 하나 끌어서 Level에 배치해보자.



- 선택, 이동, 회전, 크기를 조정해 보고, 스내핑과 카메라 이동 속도도 조절해보자.