

01_ 언리얼 시작하기

<제목 차례>

01_ 언리얼 시작하기	1
1. 개요	2
2. 언리얼 엔진 설치하기	3
3. 처음으로 언리얼 엔진 프로젝트 만들기	8

인천대학교 컴퓨터공학부 박종승
무단전재배포금지

1. 개요

언리얼 엔진은 게임을 제작하는 데에 사용되는 가장 강력한 게임 엔진이다.

언리얼 엔진은 게임 개발자를 위한 여러 게임 개발 도구들로 구성되어 있다. 이 책은 언리얼 엔진에서 게임을 개발하기 위한 게임 개발자를 위해 작성되었다. 이 책에서는 블루프린트 이외의 언어에 대해서는 사용하지 않으므로 다른 프로그래밍 언어에 대해서는 지식이 전혀 없어도 된다.

언리얼 엔진은 1998년에 공개된 버전 1부터 시작하여 2022년에 공개된 버전 5에 이르고 있다. 우리는 앞으로 종종 언리얼 엔진을 UE 또는 언리얼로 줄여서 언급할 것이다. 버전 5를 함께 명시하는 경우에는 UE5로 언급한다. UE5는 UE4의 기반 구조를 그대로 계승하여 발전시킨 버전이어서 UE4의 개발자는 쉽게 UE5를 사용할 수 있다.

이 장에서는 언리얼 엔진을 시작하기 위해서 가장 먼저 해야 하는 준비 과정에 대해서 설명한다.

먼저 언리얼 엔진의 라이선스 정책에 대해서 알아보자. 언리얼 엔진의 라이선싱 옵션으로 두 가지 중에 하나를 선택하도록 한다. **퍼블리싱 라이선스** 또는 **크리에이터 라이선스** 중에서 하나를 선택한다. 일반적인 게임이나 판매용 인터랙티브 제품을 개발하는 목적인 경우에 **퍼블리싱 라이선스**를 선택하면 된다. 무료로 사용하면 되고 제품이 성공할 경우에만 로열티를 지불하면 된다. 구체적으로, 제품을 상품화하여 해당 제품의 총 수익이 1백만\$(USD)를 초과하면 총 수익의 5%를 로열티로 지불해야 한다.

내부용이나 무료 프로젝트를 제작하거나 상호작용이 없는 단방향 콘텐츠를 개발하는 경우에는 **크리에이터 라이선스**를 선택하면 된다. 영화 프로젝트, 건축, 시뮬레이션, 시각화와 같은 도급 및 컨설팅 프로젝트 등이 이에 해당된다. 이 라이선스에서는 양방향 상호작용적인 제품을 퍼블리싱하는 경우에는 사용할 수 없다. 이 라이선스는 무료로 사용하면 되고 로열티는 면제된다.

게임은 대표적인 양방향 상호작용형 콘텐츠이므로 우리는 **퍼블리싱 라이선스**에 해당한다.

<참고> 언리얼의 라이선스 정책에 대해서 더 자세한 내용은 다음의 링크를 참조하자.

<https://www.unrealengine.com/ko/faq>

이제, 언리얼 엔진을 설치하기 위한 컴퓨터 사양에 대해서 알아보자. 언리얼 엔진은 많은 컴퓨팅 자원을 사용하는 무거운 소프트웨어이다. 따라서 최신 컴퓨터 사양을 갖추는 것이 좋다. 특히 그래픽 카드는 최신 사양을 권장한다. 외장형 그래픽 카드를 장착한 데스크탑 컴퓨터를 사용하도록 하자. 내장형 그래픽 카드를 사용하는 컴퓨터나 노트북 컴퓨터에서는 구동이 매우 느릴 수 있으므로 사용하지 않는 것이 좋다.

<참고> 언리얼 엔진의 설치 사양에 대해서 더 자세한 내용은 다음의 링크를 참조하자.

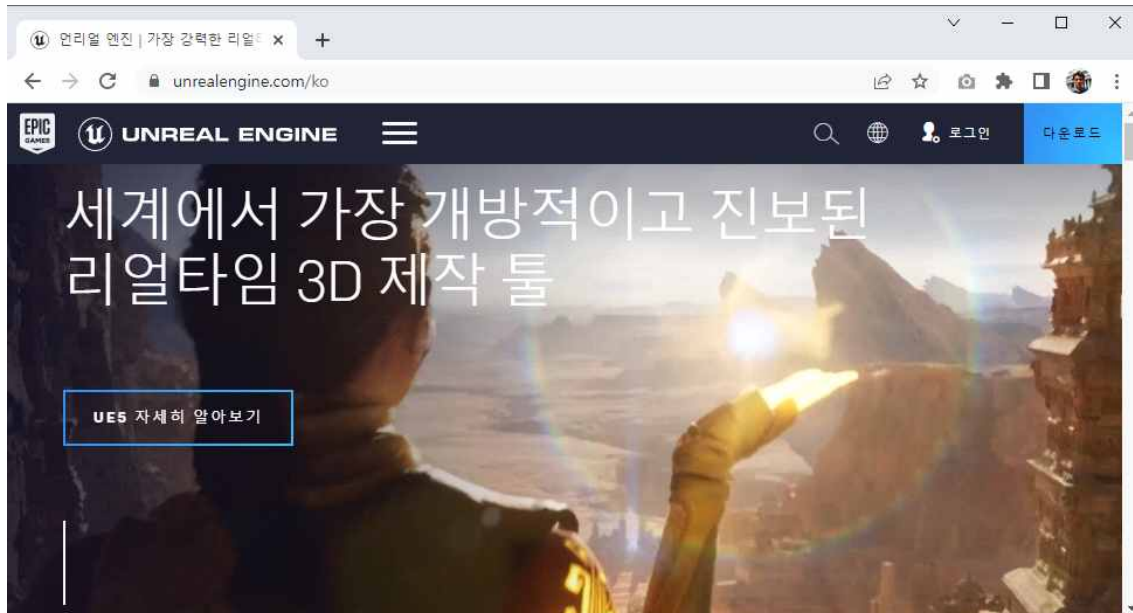
<https://docs.unrealengine.com/hardware-and-software-specifications-for-unreal-engine/>

2. 언리얼 엔진 설치하기

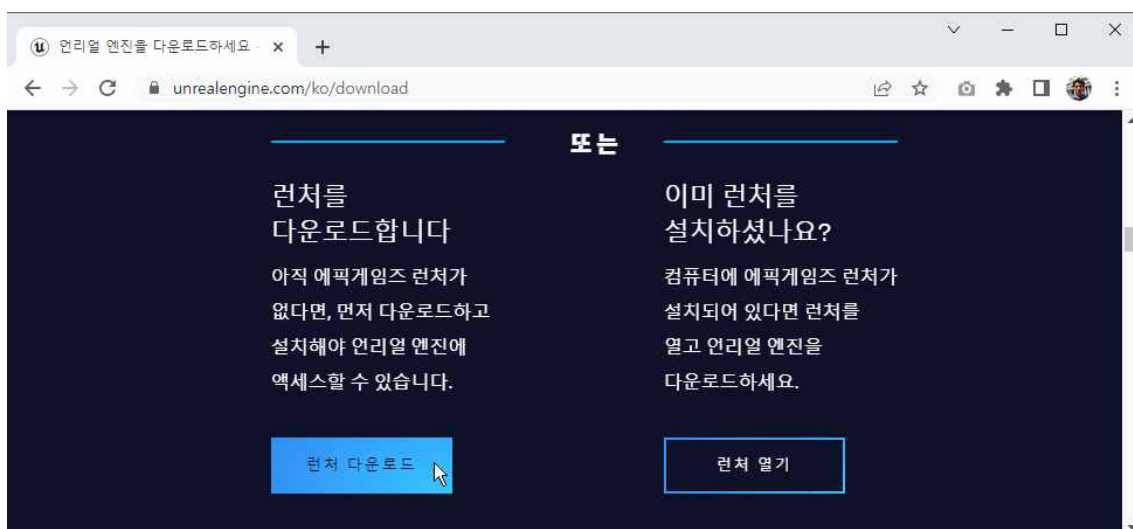
이 절에서는 언리얼 엔진을 설치하는 과정을 설명한다.
이제부터 언리얼 엔진을 설치하는 과정에 대해서 알아보자.

1. 먼저 웹 브라우저로 다음 웹사이트를 방문하자.

<https://www.unrealengine.com/>



2. 우측 상단의 **다운로드** 버튼을 클릭하자. 클릭 후에는 에픽게임즈 런처를 다운로드할 수 있는 아래 그림과 같은 화면으로 이동한다. **런처 다운로드** 버튼을 클릭하자. 그리고 **에픽게임즈 런처**를 다운받아 설치하자.

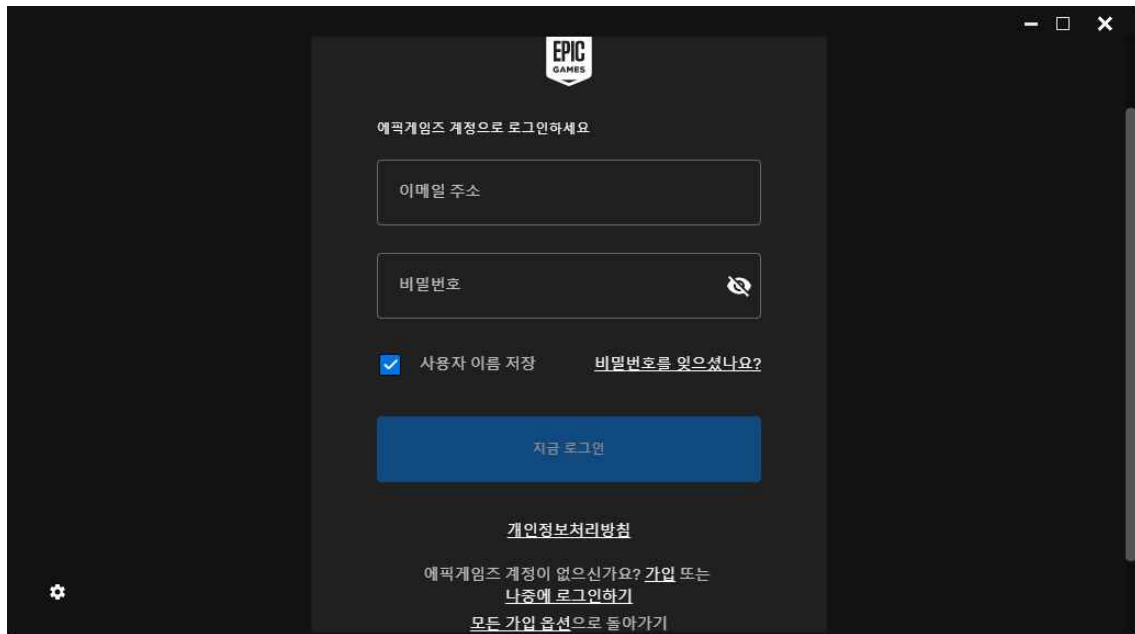


3. **에픽게임즈 런처**가 설치된 후에 이 프로그램을 실행하자. 바탕화면의 **Epic Games Launcher** 아이콘을 더블클릭하면 실행할 수 있다. 처음 실행하는 경우에 로그인을 요구한다. 로그인은 에픽게임즈 계정

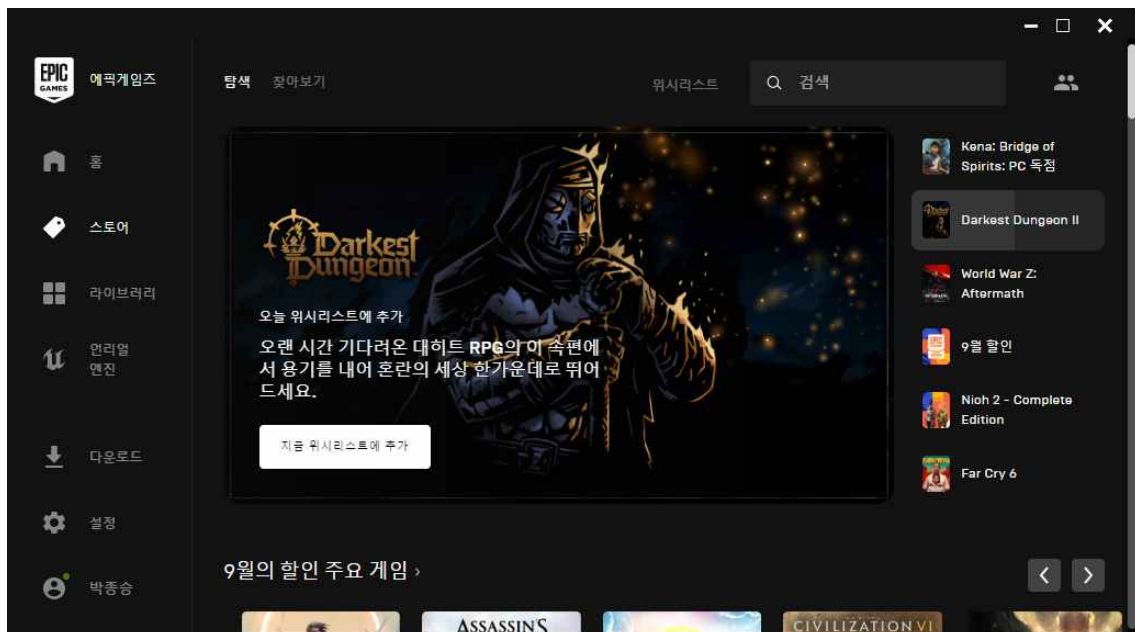
외의 다양한 종류의 계정으로도 로그인할 수 있다.

계정이 없는 경우에는 계정을 바로 생성할 수 있다. 계정을 생성하려면 화면 아래의 ‘가입’을 클릭하고 다음 화면에서 ‘이메일로 가입’을 진행하면 된다.

4. 에픽게임즈 계정이 준비되었으면 로그인하자. ‘에픽게임즈 로그인’을 클릭하면 아래와 같은 로그인 화면으로 이동한다.



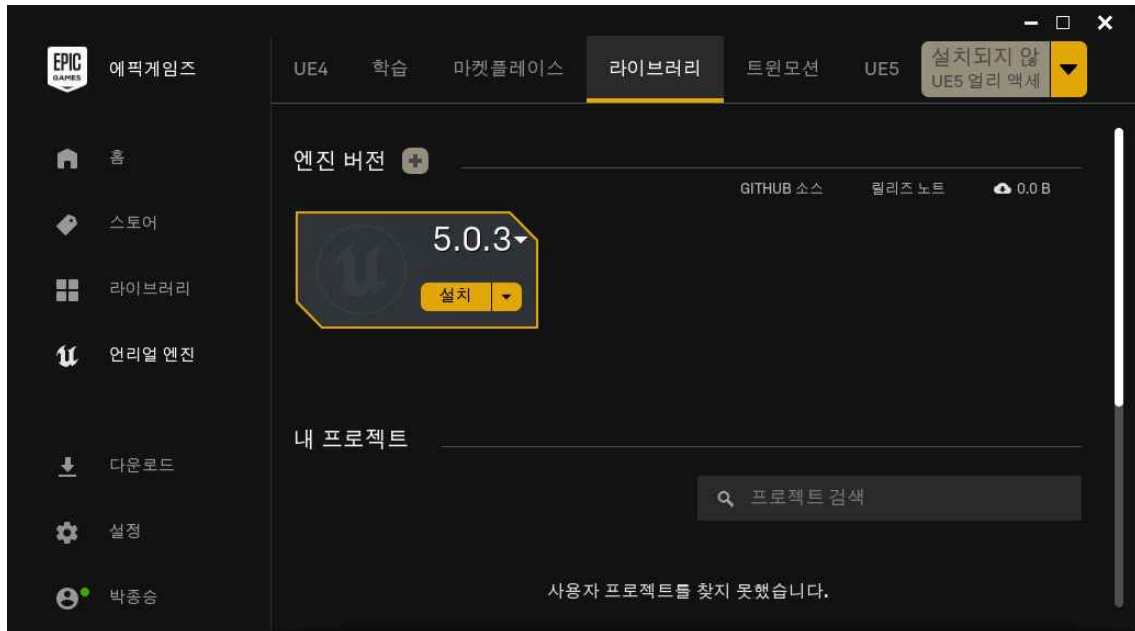
5. 로그인한 후에는 에픽게임즈 런처 메인 화면으로 이동한다.



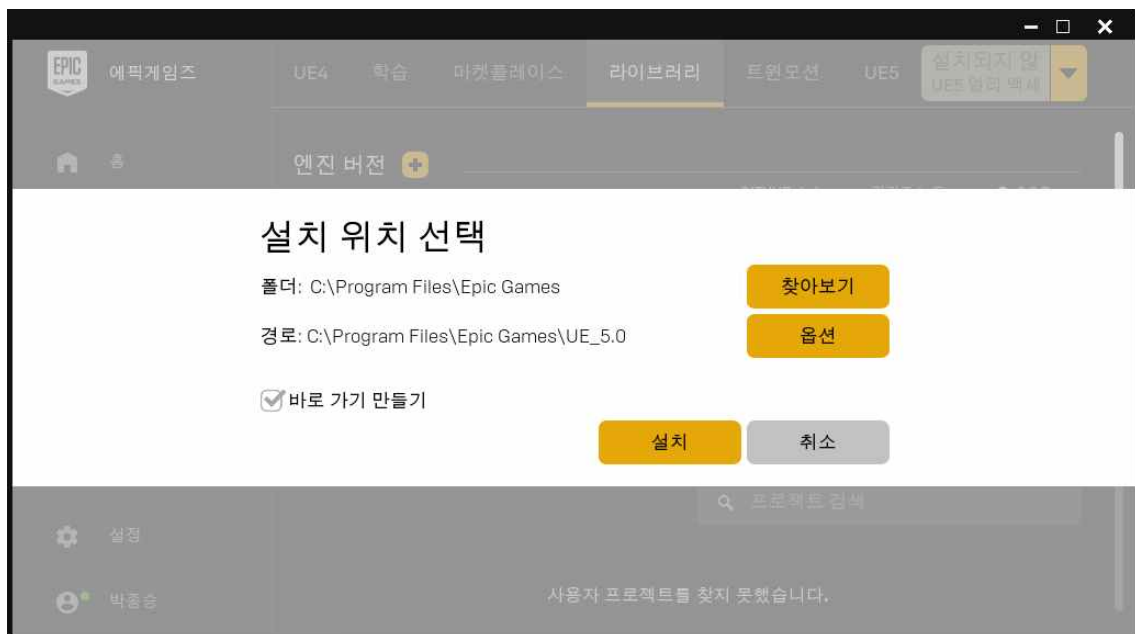
왼쪽과 상단의 메뉴를 클릭하여 다양한 정보를 확인할 수 있다. 우리는 왼쪽의 **엔리얼 엔진** 메뉴를 클릭하자.

6. 그다음에, 상단의 **라이브러리** 메뉴를 클릭하자. 아래에는 설치된 엔리얼 엔진 버전이 나열된다.

엔진 버전 오른쪽의 ‘+’ 표시를 클릭하면 설치할 언리얼 엔진의 특정 버전을 선택할 수 있다. 우리는 5.0의 가장 마지막 버전을 선택하고 **설치** 버튼을 클릭하자. 참고로, 이 교재를 작성하는 시점에서의 마지막 버전은 5.0.3이다.



7. 설치 위치를 묻는 대화상자가 나타난다. 바로 **설치** 버튼을 클릭하자. 설치가 시작되면 32GB 이상의 큰 용량의 엔진을 다운로드하기 시작한다. 따라서 설치에 시간이 상당히 걸릴 것이다.



8. 설치가 완료되면 엔진을 실행해보자. 에픽게임즈 런처의 **라이브러리** 메뉴에서 **엔진 버전** 아래에 있는 해당 엔진 버전 아이콘의 **설치** 버튼이 설치된 후에는 **실행** 버튼으로 바뀐다. 이 **실행** 버튼을 클릭하면 설치한 엔진이 실행된다.

이미 만들어진 언리얼 엔진 프로젝트가 있다면 **내 프로젝트** 아래에 프로젝트 아이콘이 보일 것이다. 그 아이콘을 클릭하면 엔진이 실행되며 해당 프로젝트를 로드한다.

9. 바탕화면의 바로가기 아이콘을 통해 에픽게임즈 런처를 실행하지 않고 엔진을 바로 실행할 수 있다.

만약 바탕화면에 바로가기 아이콘이 없다면 다음과 같이 바로가기 아이콘을 만들 수 있다.



엔진 아이콘의 **실행** 버튼의 오른쪽 화살표 버튼을 클릭하면 팝업 메뉴가 나타난다. 여기에서 **바로가기 생성**을 실행하자. 이제 바탕화면에 언리얼 엔진 아이콘이 생성될 것이다.

이제 이 언리얼 엔진 아이콘을 더블클릭하면 런처를 통하지 않고 바로 엔진을 실행할 수 있다. 우리는 앞으로 이 언리얼 엔진 아이콘을 더블클릭하여 런처를 통하지 않고 바로 엔진을 실행할 것이다.

이미 생성한 프로젝트가 있다면 그 폴더로 이동해서 확장자가 **.uproject**인 프로젝트 파일을 찾아서 이를 더블클릭하면 엔진이 실행되며 해당 프로젝트가 로드된다.

이제까지 언리얼 엔진의 설치에 대해서 알아보았다.

3. 처음으로 언리얼 엔진 프로젝트 만들기

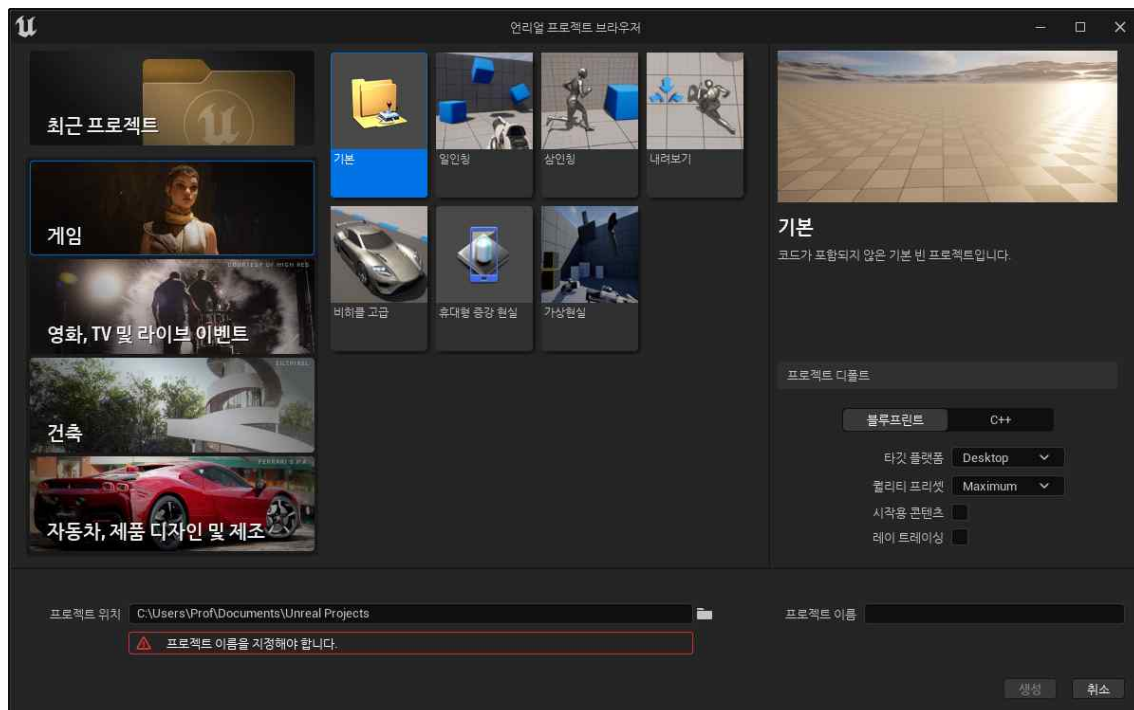
언리얼 엔진 프로젝트를 만드는 방법에 대해서 알아보자.

이 절에서는 처음으로 언리얼 엔진 프로젝트를 만드는 과정을 설명한다.

1. 새 언리얼 프로젝트 **Pfirst**를 만들자.

바탕화면에 있는 언리얼 엔진 아이콘을 더블클릭하여 언리얼 엔진을 실행하자. 에픽게임즈 런처에서 설치한 엔진 아이콘을 클릭하여 실행해도 된다.

실행하게 되면 먼저 **언리얼 프로젝트 브라우저**가 보일 것이다.



왼쪽의 **최근 프로젝트** 탭을 클릭하면 이미 생성된 프로젝트를 선택할 수 있다.

최근 프로젝트 아래의 탭들은 새 프로젝트를 만드는 경우에 만들 프로젝트의 종류에 따라서 관련 템플릿을 보여주고 있다. 게임 프로젝트를 위한 **게임** 탭이 있고 그 아래에는 영화나 건축 등을 위한 탭들이 있다.

게임 탭을 클릭하자. 중간에는 게임을 위한 템플릿 목록이 나열되고 우측에는 프로젝트 생성을 위한 옵션이 보인다. 템플릿 목록에서는 가장 기본적인 **기본** 템플릿을 선택하자.

2. 언리얼 프로젝트 브라우저의 오른쪽에서는 상단에는 현재 선택된 템플릿을 설명하는 그림이 보인다. 그 아래의 **프로젝트 디폴트** 영역에서 생성을 위한 옵션들이 나열된다.

먼저 블루프린트 또는 C++ 중에서 프로젝트에서 어떤 언어를 주로 사용할 지의 옵션이 있다. 블루프린트는 언리얼 엔진에서 사용하는 비주얼 스크립트 언어이다. 프로젝트에서 블루프린트만 사용할 것이라면 **블루프린트**를 선택하면 된다. 블루프린트와 더불어 C++도 사용할 것이라면 **C++**를 선택하면 된다. 그러나 블루프린트를 선택한 경우라도 나중에 C++ 코드를 추가할 수 있고 C++를 선택한 경우라도 나중에 블루프린트를 추가할 수 있다.

우리는 앞으로 블루프린트만 사용할 것이므로 **블루프린트**를 선택하자.

3. 프로젝트 디폴트의 두번째 옵션인 타겟 플랫폼에서는 Desktop 또는 Mobile 중에서 하나를 선택할 수 있다. 우리는 디폴트인 Desktop에 그대로 두자.

퀄리티 프리셋에서는 Maximum 또는 Scalable 중에서 하나를 선택할 수 있다. 우리는 디폴트인 Maximum에 그대로 두자.

타겟 플랫폼과 퀄리티 프리셋은 나중에 프로젝트 세팅 창에서 언제든지 바꿀 수 있다.

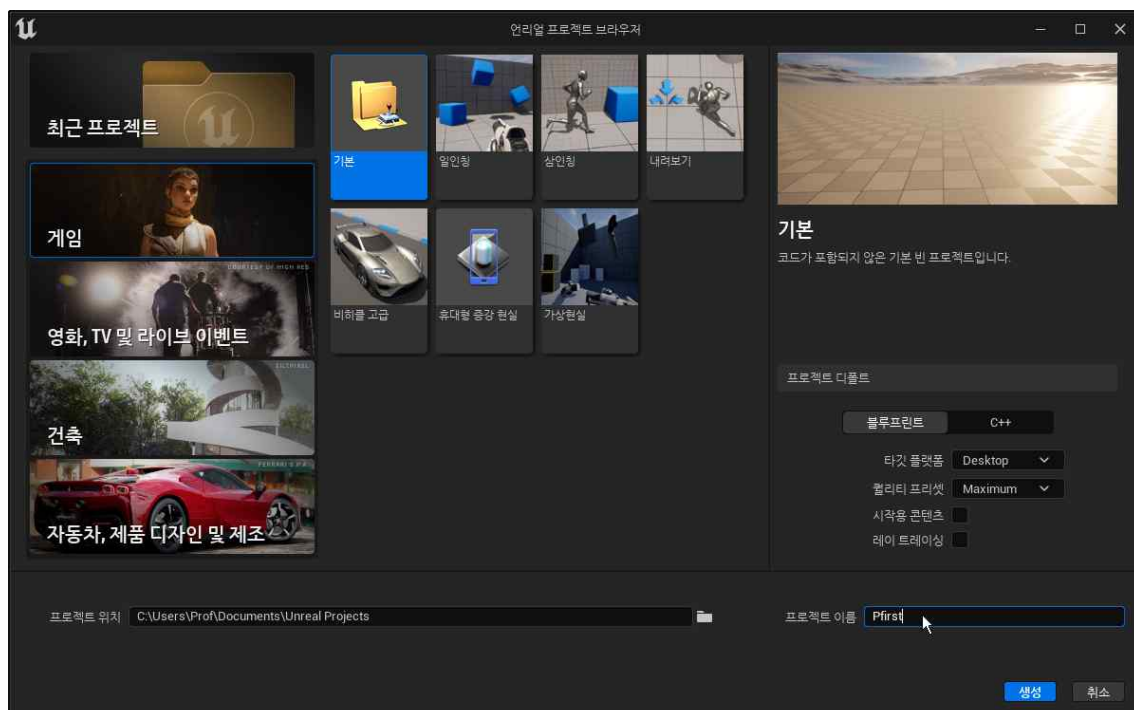
4. 그다음 옵션인 시작용 콘텐츠는 프로젝트에 시작용 콘텐츠를 포함할지에 대한 선택 옵션이다. 여기서 시작용 콘텐츠 체크박스를 체크 해제하여 시작용 콘텐츠가 포함되지 않도록 하자. 앞으로의 다른 모든 프로젝트에서도 시작용 콘텐츠는 특별한 언급이 없으면 포함되지 않도록 하자.

<참고> 시작용 콘텐츠는 언리얼 엔진 초보자들에게 유용한 애셋들을 모아둔 것으로 배우는 시기에는 매우 편리하게 사용할 수 있다. 그러나 파일 용량이 매우 크다. 시작용 콘텐츠를 추가하면 프로젝트 폴더 아래에서 Content 폴더 아래에 StarterContent라는 폴더가 생긴다. 이 폴더는 대략 270개의 파일로 포함하고 있고 용량이 630MB 이상이다. 배우는 단계에서는 유용할 수 있으나 나중에 어느 정도 학습 수준이 높아진 후에는 굳이 포함시킬 필요가 없다. 그리고, 프로젝트 생성 시에 시작용 콘텐츠를 포함시키지 않더라도 나중에 콘텐츠 브라우저를 사용하여 추가할 수도 있다. 우리는 기본적으로 시작용 콘텐츠가 포함되지 않도록 프로젝트를 생성할 것이다.

5. 그다음 옵션인 레이 트레이싱은 프로젝트에 실시간 레이 트레이싱을 지원할지의 여부를 선택한다. 레이 트레이싱은 계산량이 큰 고급 그래픽 기능이다. 이 고급 기능을 사용하지 않는 경우라면 체크하지 않고 기본값인 언체크 상태 그대로 두자.

6. 하단에는 프로젝트가 저장될 위치와 프로젝트 이름을 입력하기 위한 입력 상자가 있다.

프로젝트 위치는 그대로 두자. 프로젝트 이름은 Pfirst로 입력하자.



이제 생성 버튼을 클릭하자. 프로젝트 생성에는 약간의 시간이 걸리므로 생성이 완료될 때까지 잠시 기다려야 한다.

<참고> 이 교재에서 프로젝트 이름을 작명하는 방법에 대해서 알아보자. 프로젝트 이름은 접두사 P로 시작한다. 그 뒤에 프로젝트 내용과 관련된 하나 또는 두 개의 영단어를 모두 소문자로 붙인다. 이번 예제에서는 접두사 P와 첫 프로젝트라는 의미의 first를 붙여서 Pfirst라는 문자열을 프로젝트 이름으로 지정하였다. 이렇게 하는 이유는 프로젝트 이름이 너무 길어지지 않도록 하면서 동시에 대문자 P로 시작하는 하나의 단어처럼 보이도록 하는 목적이 있다.

이러한 작명 방법은 이 교재의 샘플 프로젝트 이름의 작명을 위해서 저자가 고안한 방법이다. 표준 방법이 아니므로 꼭 이렇게 따라할 필요는 전혀 없다.

7. 프로젝트가 생성된 후에는 자동으로 언리얼 에디터 창이 나타날 것이다.
 언리얼 에디터는 언리얼 엔진의 메인 창으로 아래와 같은 모습으로 나타날 것이다.

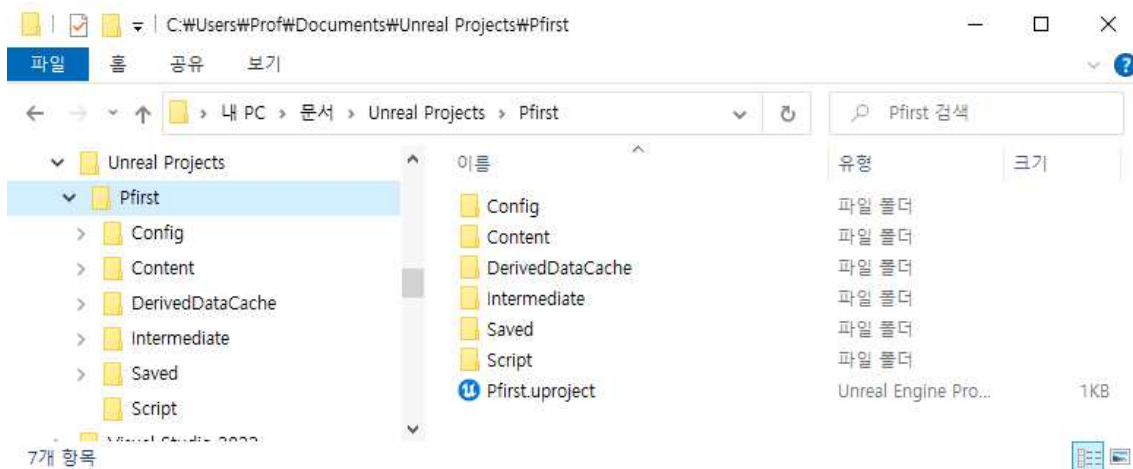


<참고> 만약 위의 그림과 같은 모습이 아닌 다른 모습이 보인다면, 직전의 단계에서, 언리얼 프로젝트 브라우저에서 프로젝트를 생성할 때에 **시작용 콘텐츠** 옵션을 체크해제하지 않고 체크된 채로 그대로 프로젝트를 생성했기 때문일 것이다. **시작용 콘텐츠**에 체크된 채로 프로젝트가 생성되면 **시작용 콘텐츠**에 포함되어 있는 레벨인 **Minimal_Default**라는 레벨이 로드된다.

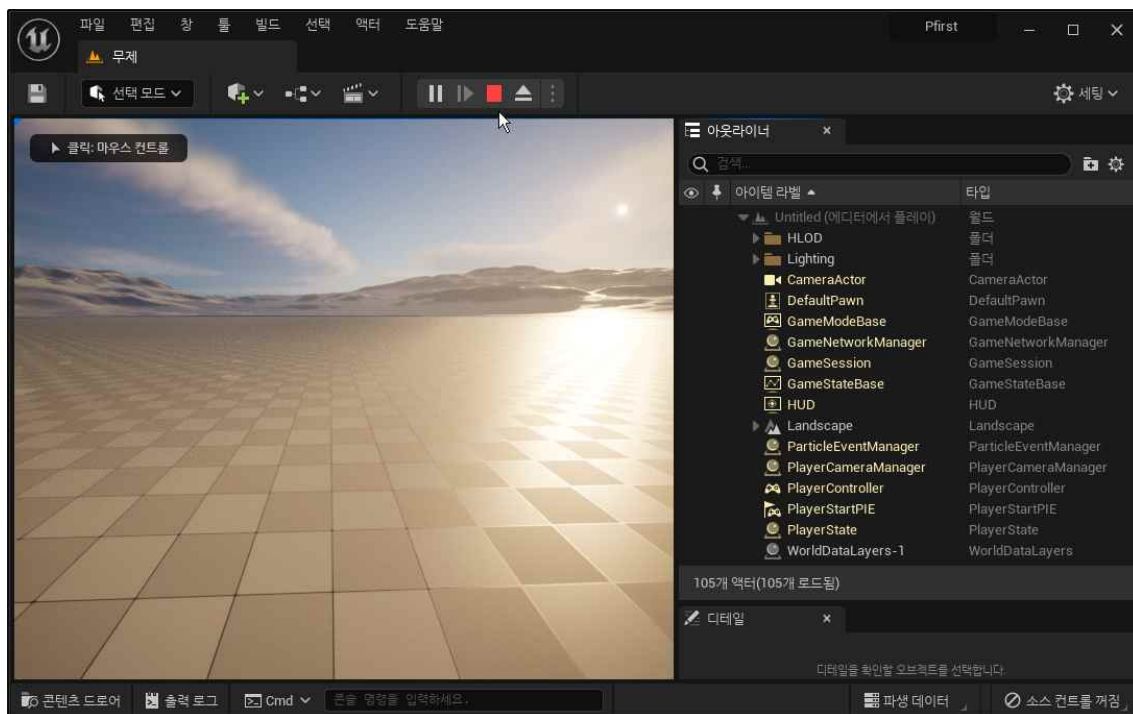
이 경우에는, 프로젝트를 삭제하고 동일한 이름으로 프로젝트를 다시 생성하자. 먼저, 파일 탐색기를 열고 사용자의 **문서** 폴더로 이동하자. 그다음에, **Unreal Projects** 폴더로 이동하자. 그 아래에 프로젝트 폴더인 **Pfirst** 폴더가 있을 것이다. 이 폴더를 삭제하자. 그다음, 바탕화면에 있는 언리얼 엔진 아이콘을 더블클릭하여 언리얼 엔진을 다시 실행하고, 프로젝트 생성 과정을 처음부터 다시 진행하면 된다.

8. 프로젝트가 생성된 폴더 구조를 살펴보자. 먼저, 윈도우의 파일 탐색기를 열자. 그다음, 사용자의 **문서** 폴더로 이동하고 그 아래의 **Unreal Projects** 폴더로 이동하자. 그 아래에, 프로젝트 폴더인 **Pfirst** 폴더가 있을 것이다.

프로젝트 폴더의 바로 아래에는 프로젝트 파일인 **Pfirst.uproject**가 있다. 그리고 여러 하위 폴더들이 생성되어 있다. 그리고 **Config**, **Content** 등의 여러 하위 폴더들이 생성되어 있다.



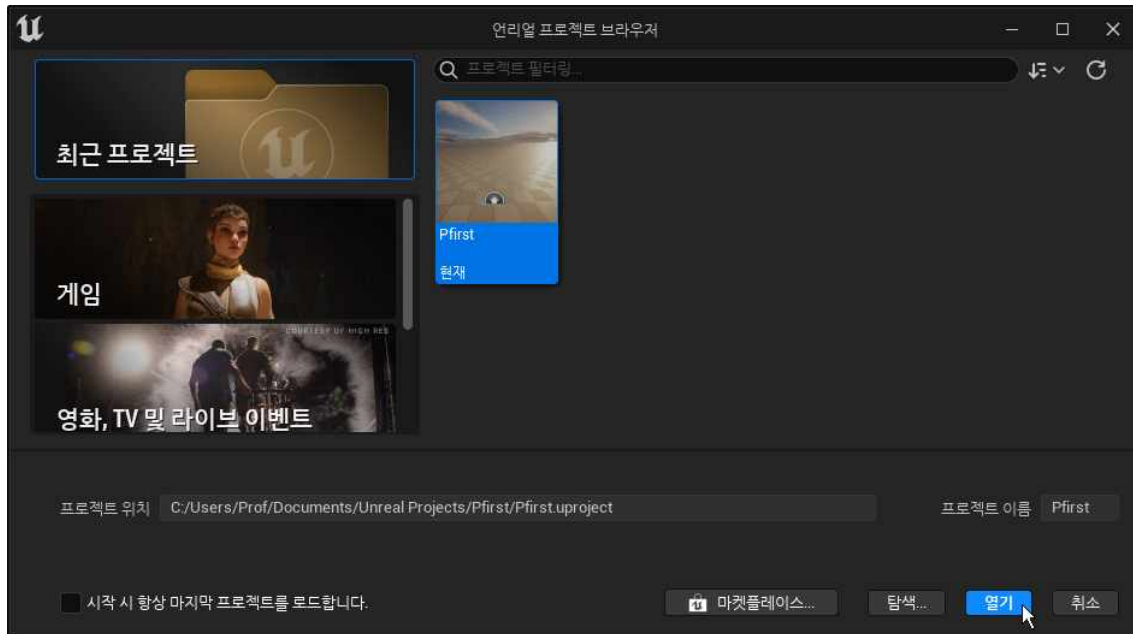
9. 게임을 플레이해보자. **언리얼 에디터**의 툴바에 있는 **플레이** 버튼을 클릭하면 게임이 실행된다. 디폴트 게임 실행모드에서는 뷰포트 내에서 게임이 실행될 것이다. 게임을 중지하려면 **ESC** 키를 누르거나 또는 툴바의 중지 버튼을 클릭하면 된다.



<참고> 게임이 플레이되면 화면 왼쪽 상단에 **클릭: 마우스 컨트롤**이라는 메시지가 표시된다. 이와 관련된 마우스 제어권에 대해서 알아보자. 게임이 플레이되더라도 마우스 제어는 여전히 에디터에 남아 있다. 즉 게임이 시작하더라도 에디터의 메뉴 등을 조작할 수 있는 상태로 남아있게 된다. 마우스 제어권을 게임플레이로 넘기기 위해서는 뷰포트를 클릭하면 된다. 뷰포트를 클릭하면 뷰포트 화면 왼쪽 상단에 **마우스 커서는 Shift+F1**이라는 메시지가 표시되며, 마우스 조작이 게임플레이로만 전달되고 에디터의 메뉴 등을 조작할 수 없게 된다. 만약 게임을 종료하지 않고 실행 도중에 마우스 제어권을 다시 에디터로 넘기려면 **Shift+F1** 키를 누르면 된다. 이렇게 마우스 제어권을 수시로 에디터와 게임플레이 사이를 오가게 할 수 있다.

10. 이미 생성된 프로젝트를 다시 로드하는 방법에 대해서 알아보자. 현재의 언리얼 에디터 창을 종료하자. 그다음, 다시 바탕화면에 있는 언리얼 엔진 실행 아이콘을

더블클릭해보자. **언리얼 프로젝트 브라우저**가 뜬다. 왼쪽의 **최근 프로젝트** 탭을 클릭하면 오른쪽에 기존의 프로젝트가 나열된다. 여기에서 **Pfirst**를 선택하고 창의 아래 부분에 있는 **열기** 버튼을 클릭하자. 언리얼 에디터 창이 뜰 것이다.



<참고> **언리얼 프로젝트 브라우저**에서 **최근 프로젝트** 탭이 선택된 경우 창의 왼쪽 아래에 ‘**시작 시 항상 마지막 프로젝트를 로드합니다.**’로 표시된 체크박스가 있다. 이것을 체크하면 다음부터는 언리얼 엔진을 실행할 경우 **언리얼 프로젝트 브라우저**가 뜨지 않고 바로 최근 작업한 프로젝트에 대한 언리얼 에디터 창이 열린다. 하나의 프로젝트만을 계속해서 작업하는 경우에는 **언리얼 프로젝트 브라우저**를 생략하도록 이 체크박스에 체크하는 것이 편리할 것이다.

그러나 우리는 새 프로젝트를 수시로 생성할 것이므로 체크하지 않도록 하자.

한편, 체크한 후에 원래대로 체크를 해제한 상태로 되돌리려면 언리얼 에디터의 메뉴바에서 **파일 » 프로젝트 열기** 메뉴를 선택하자. **프로젝트 열기** 창이 뜰 것이다. 이 창의 아래에서 ‘**시작 시 항상 마지막 프로젝트를 로드합니다.**’로 표시된 체크박스가 나타나면 여기서 체크를 해제하자. 다음부터는 다시 **언리얼 프로젝트 브라우저**가 표시될 것이다.

11. 프로젝트를 백업하거나 다른 컴퓨터로 옮기는 방법에 대해서 알아보자.

언리얼 프로젝트는 콘텐츠가 추가됨에 따라서 용량이 매우 커지게 되고 백업에도 어려움이 있다. 프로젝트 폴더 아래에는 프로젝트 파일인 **Projectname.uproject** 파일과 다음의 폴더들이 있다: **Config**, **Content**, **DerivedDataCache**, **Intermediate**, **Saved**.

이 중에서 **Config**, **Content**의 두 폴더와 **Projectname.uproject** 파일만 백업하면 된다. **Config** 폴더에는 프로젝트 설정 **.ini** 파일들이 있다. 이들은 프로젝트 세팅의 중요한 정보가 있으므로 삭제해서는 안된다. **Content** 폴더에는 프로젝트의 모든 콘텐츠 파일인 애셋 데이터 **.usset** 파일과 레벨 데이터 **.umap** 파일이 있다. 이들은 중요한 정보이고 삭제해서는 안된다.

그 외의 파일과 폴더는 삭제해도 된다. **DerivedDataCache**, **Intermediate**, **Saved** 폴더들은 프로젝트를 빌드하면 자동생성된다.

이제까지 우리의 첫 언리얼 엔진 프로젝트를 만들어보았다.

