

유니티 엔진 설치

[유니티 허브, 유니티 에디터 설치 가이드] + GitHub을 이용한 버전관리

학과 : 소프트웨어학과

학번 : 2019975070

이름 : 한재훈



목차

- 게임 엔진 이란?
- 유니티 엔진
- 유니티 허브
- 유니티 에디터
- Git을 이용한 버전 관리(추가)

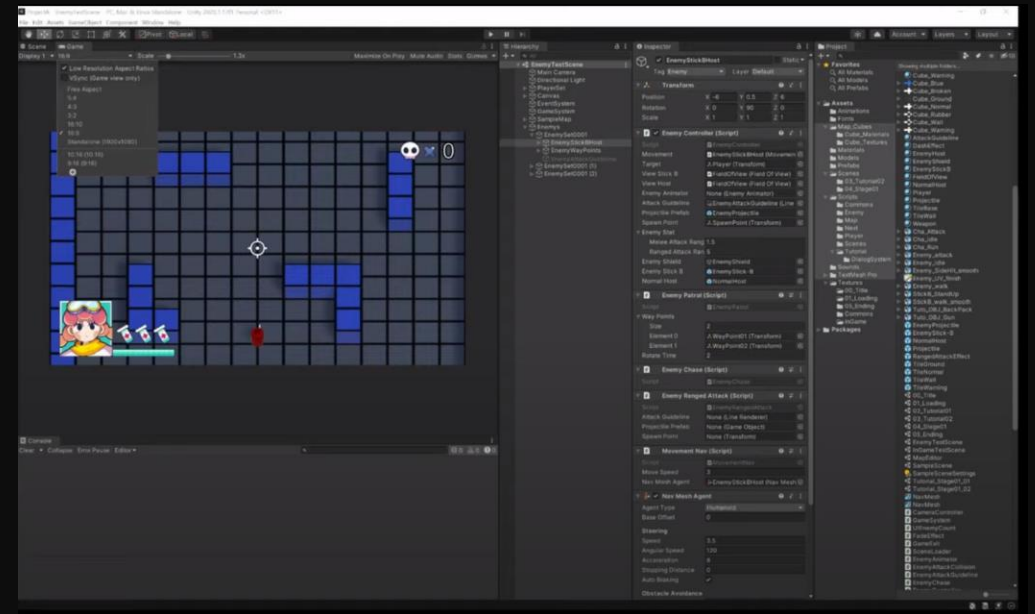
게임 엔진 이란?

- 게임 개발에 범용적으로 필요한 기능들을 제공하는 소프트웨어
 - Ex) Unity, Unreal Engine
-



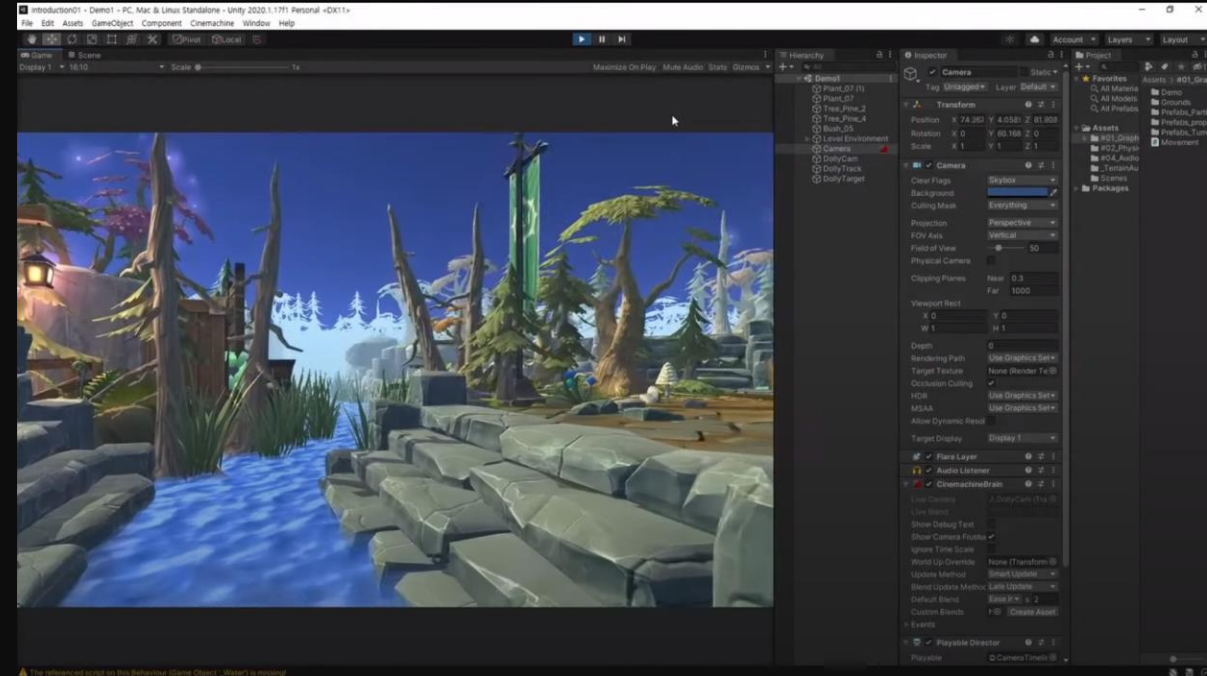
그래픽 사용자 인터페이스 (Graphical User Interface, GUI)

- 개발자가 게임 개발을 위해 사용하는 게임 엔진을 좀 더 쉽게 제어할 수 있도록 그래픽 기반의 사용자 인터페이스 제공



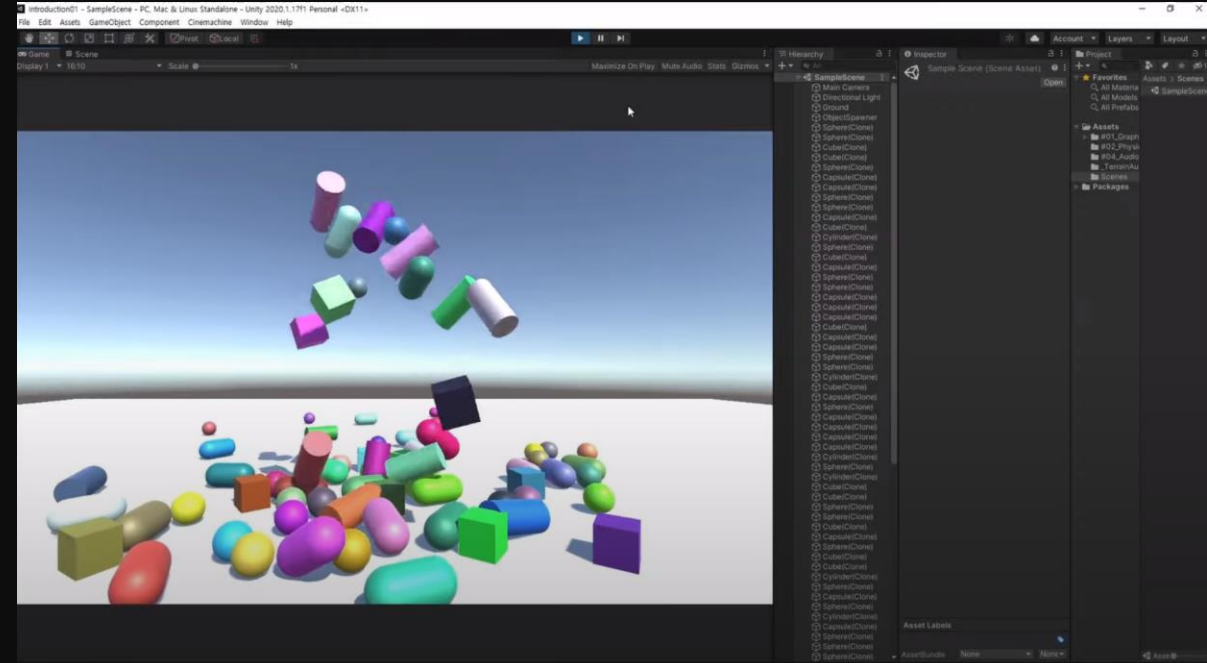
그래픽스(Graphics), 렌더링(Rendering)

- 2D 이미지, 3D 모델(애니메이션)을 화면에 출력하는 렌더링 과정
- 게임의 장면 표현에 필요한 여러 그래픽 요소 구현
 - 텍스처 매핑, 범프 매핑, 조명, 그림자, 반사, 투명, 셰이딩



물리(Physics)

- 게임 내에 존재하는 오브젝트가 물리적인 영향을 받았을 때에 대한 처리
 - 중력의 영향으로 떨어지거나 무언가에 부딪혔을때의 처리등



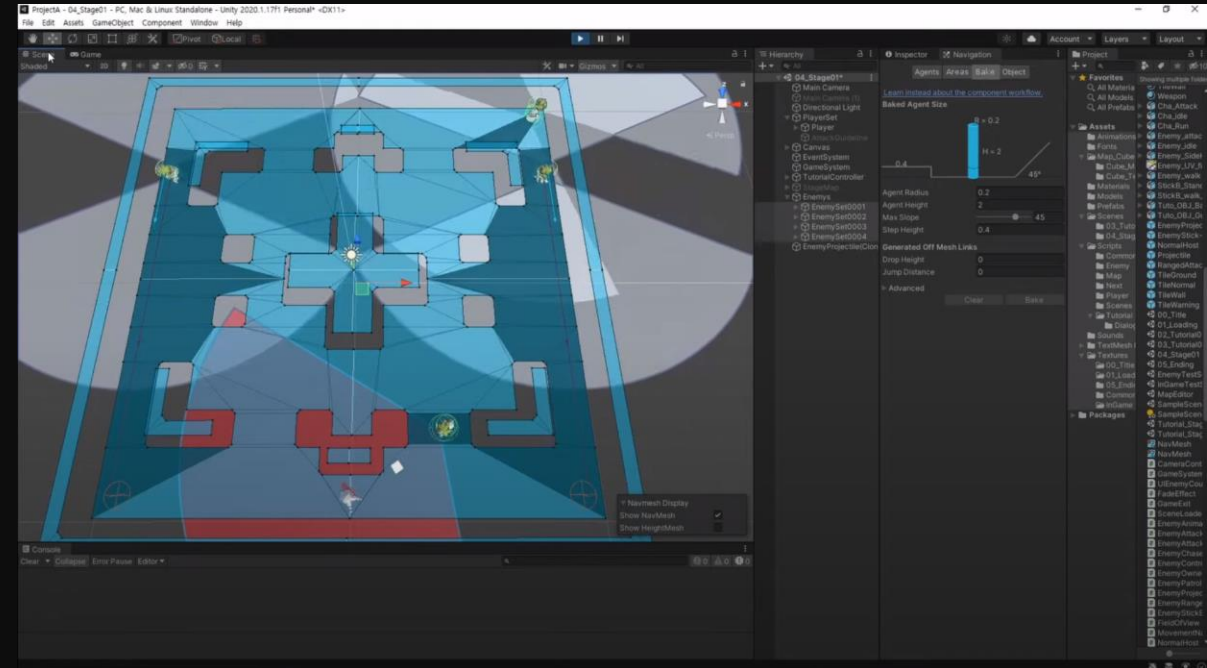
오디오(Audio)

- 게임에서 사용되는 BGM, 효과음을 재생
- 실시간 믹싱 및 마스터링, 믹서 계층 구조, 스냅샷 등의 다양한 사운드 라이브러리를 사용할 수 있게 제공



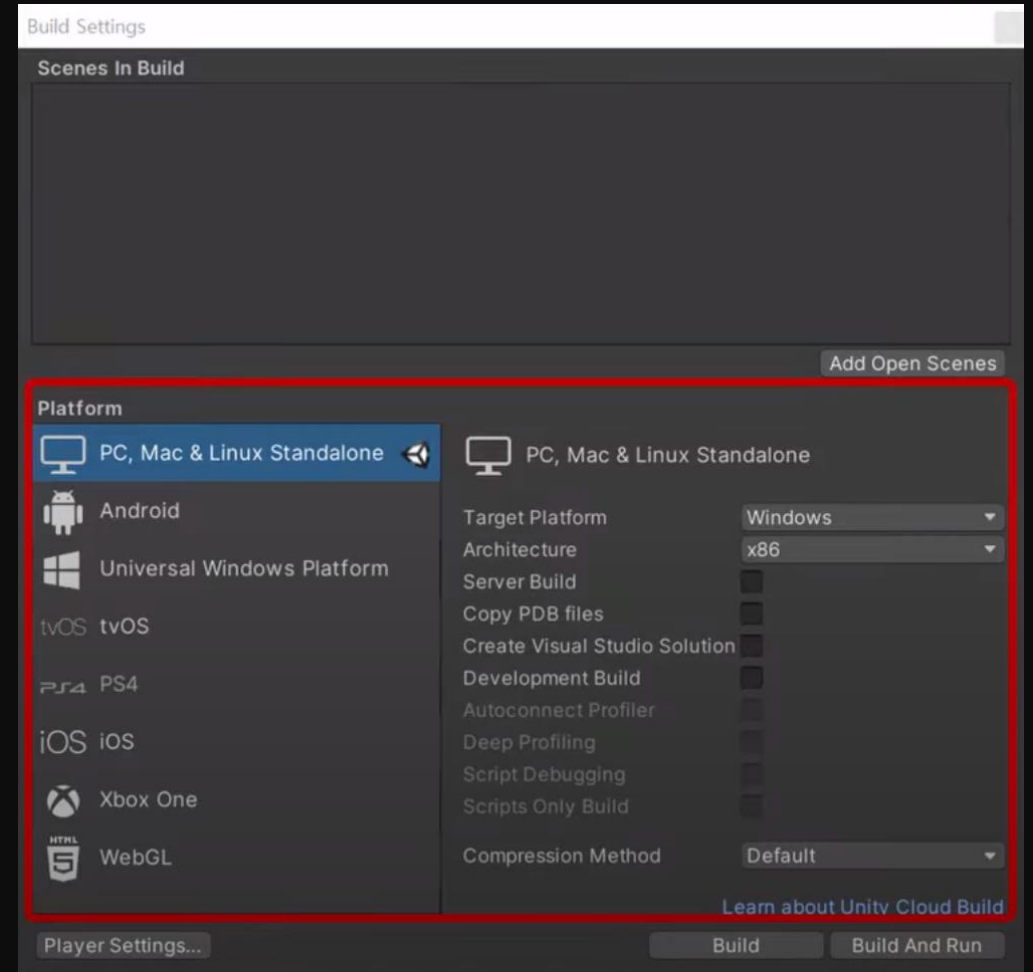
인공지능(Artificial Intelligence)

- 게임에 사용되는 여러 인공지능 라이브러리를 제공
- 유한 상태 기계(FSM), 경로 탐색, 장애물 회피, 지형 분석, 트리거 시스템 등



다중 플랫폼(Multi Platform)

코드에 약간의 수정 또는 수정 없이 게임
콘솔과 PC, Mobile 등 다양한 플랫폼에서
실행 가능하도록 도와주는 기능



그외

협업 시스템 (Collaborate), 클라우드 서비스,
In-App 결제, 광고

Audio

Editor

Graphics

Input Manager

Package Manager

Physics

Physics 2D

Player

Preset Manager

Quality

Script Execution Order

Services

Ads

Analytics

Cloud Build

Cloud Diagnostics

Collaborate

In-App Purchasing

Tags and Layers

TextMesh Pro

Time

Timeline

Version Control

Audio

Editor

Graphics

Input Manager

Package Manager

Physics

Physics 2D

Player

Preset Manager

Quality

Script Execution Order

Services

Ads

Analytics

Cloud Build

Cloud Diagnostics

Collaborate

In-App Purchasing

Tags and Layers

TextMesh Pro

Time

Timeline

Version Control

Audio

Editor

Graphics

Input Manager

Package Manager

Physics

Physics 2D

Player

Preset Manager

Quality

Script Execution Order

Services

Ads

Analytics

Cloud Build

Cloud Diagnostics

Collaborate

In-App Purchasing

Tags and Layers

TextMesh Pro

Time

Timeline

Version Control

Cloud Build

Build games faster

Turn on Cloud Build to upload and share builds with your team.

Get Unity Teams Advanced to Build in the Cloud and easily compile, deploy, and test your games automatically.

Start Using Cloud Build

Learn More

Collaborate

Create together seamlessly

Quickly and easily access your project from anywhere. Sync and share with the entire team.

Ask the organization owner or manager for access to Unity Teams to use the Collaborate service.

Learn More

Cloud Diagnostics

Discover app errors and collect user feedback.

Crashes and exceptions

Crashes and exceptions aggregates your exception data, so you can optimize your game faster.

Supported Platforms

Android iOS Linux Mac PC WebGL Windows 8 Universal Windows 10 Universal

User Reporting

To use user reporting you need to download the SDK.

Download the SDK

Supported Platforms

Android iOS Linux Mac PC WebGL Windows 8 Universal Windows 10 Universal

유니티 엔진 (Unity Engine)

- 개발사 : Unity Technologies
- 정식 출시 : 2005년 6월 8일, 맥 전용
개발 툴 추후 윈도우 지원
- 지원 언어 : C#



다양한 활용 분야

콘솔 / PC 패키지 게임, 모바일 게임, VR 게임,
게임 외에도 다양한 분야에서 수많은 작품이
배포되어 사용되고 있다.

다양한 활용 분야

게임 개발(2D/3D, PC or Mobile) 뿐만 아니라
VR/AR, 영화, 애니메이션 자동차 및 교통
분야, 설계/엔지니어링 및 건축 등 다양한
분야에서 활용 가능

유통 및 제조

유통 및 제조 업계 환경에
Unity를 활용하여 혁신을 확보하세요.



Unity Gaming Services

멀티플레이어, 백엔드, 플레이어 참여 유도, 분석,
수익화 등과 관련한 Unity Gaming Services로
게임을 성공작으로 만드세요.

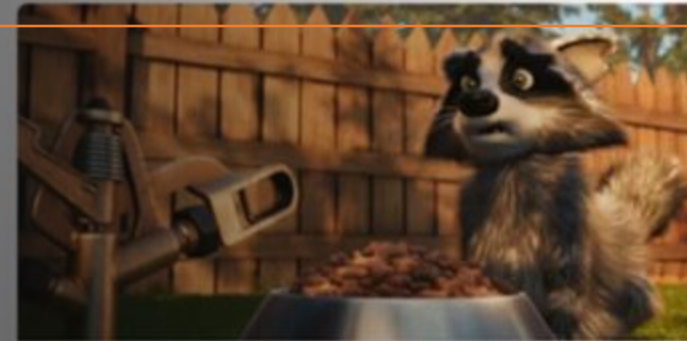
[자세히 알아보기](#)



기업용 Accelerator

기업이 더 빠르게 혁신할 수
있는 맞춤형 기술 솔루션입니다.

[자세히 알아보기](#)



영화, 애니메이션 및 시네마틱

영화 및 애니메이션 프로젝트에서 더 자유롭게 예
술적 영감을 표현하고 제작 기간을 단축하세요.

[자세히 알아보기](#)



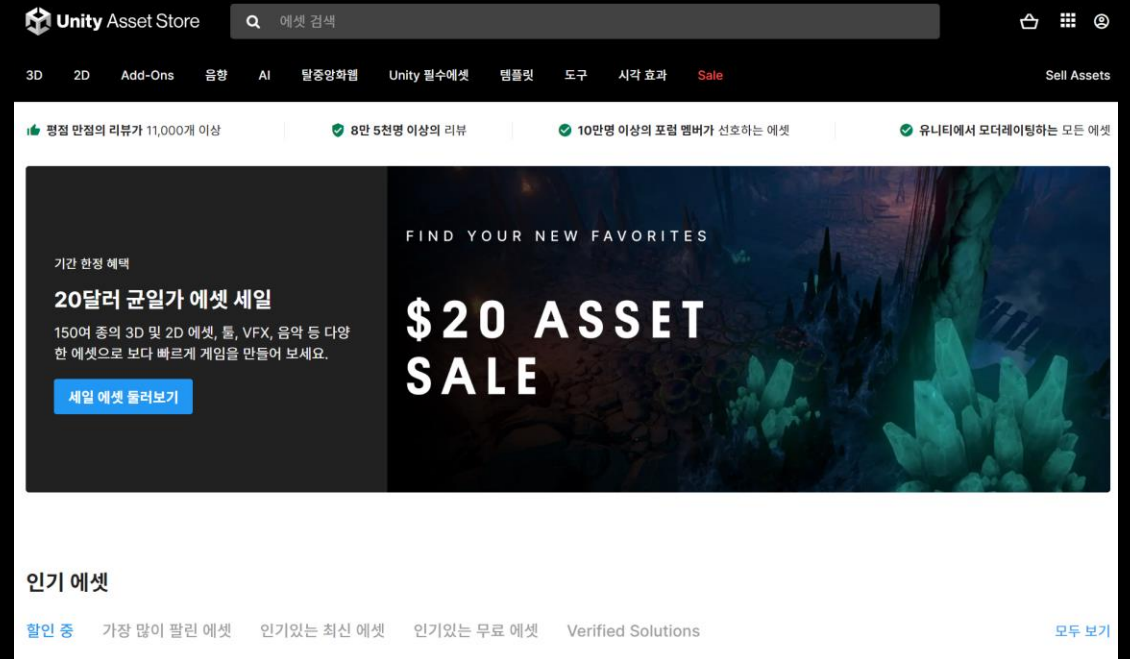
정부 및 항공우주

유니티가 제공하는 데이터 시
뮬레이션, AI 및 머신러닝 훈련, 확
률론적 시뮬레이션 및 훈련 솔루션

[자세히 알아보기](#)

에셋 스토어 (Asset Store)

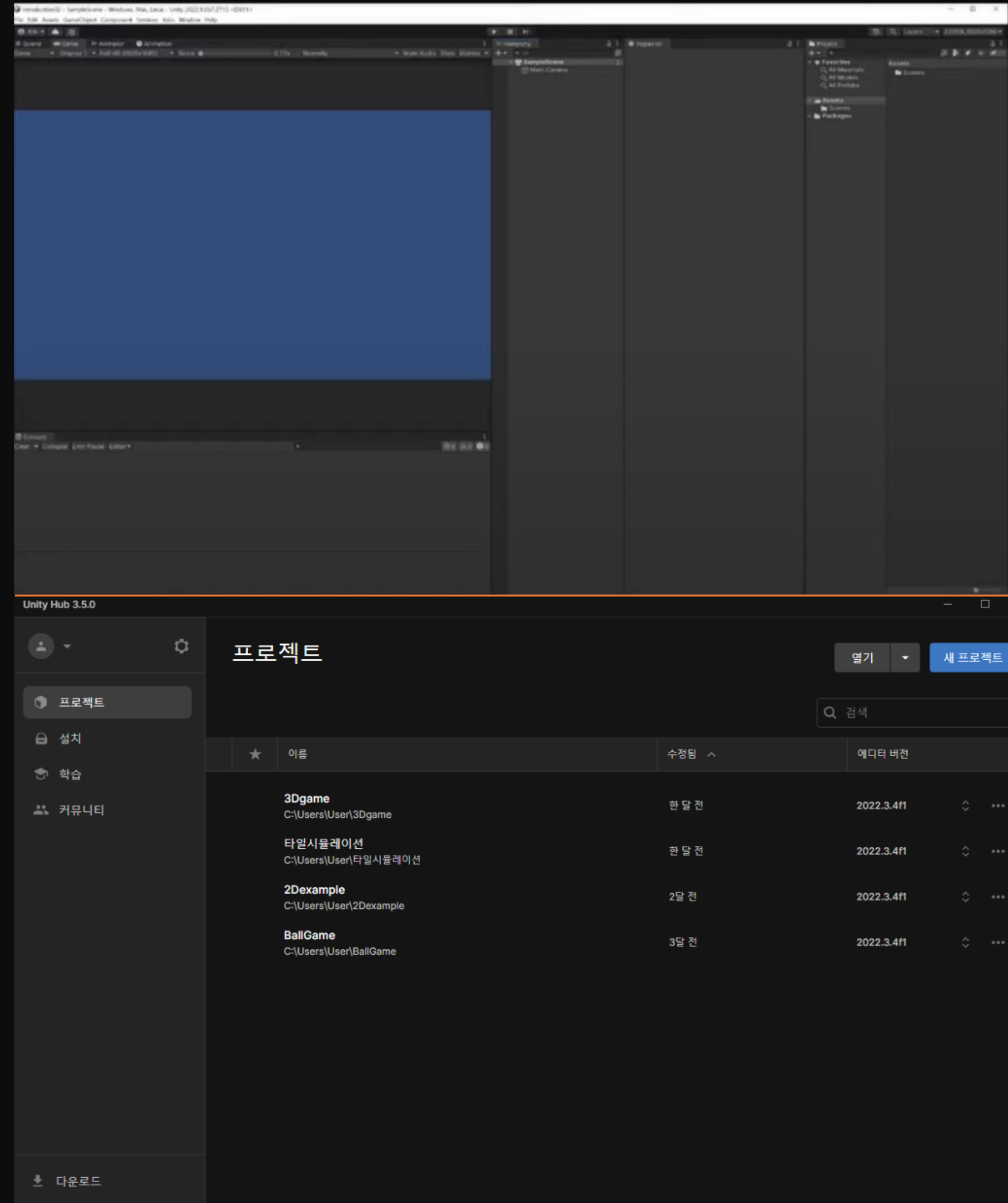
- 게임 개발에 필요한 콘텐츠와 리소스를 거래
- 2D/3D 그래픽 리소스, 사운드, 이펙트, 완성된 게임 프로젝트 거래
- 구매 뿐만 아니라 판매자가 되어 게임 외적인 수익 창출 가능



유니티 엔진 (Unity Engine)

- 유니티 허브 (Unity Hub) : 유니티 에디터 버전 관리 프로그램
- 유니티 에디터(Unity Editor) : 유니티 게임 개발 소프트웨어

❖ 위쪽이 유니티 에디터 밑에가 허브





유니티 허브(Unity Hub) 다운로드

유니티 홈페이지(<https://unity.com/kr>) 접속 후 "다운로드" 버튼 클릭

UNITY 다운로드

창의력을 발휘할 수 있는 공간

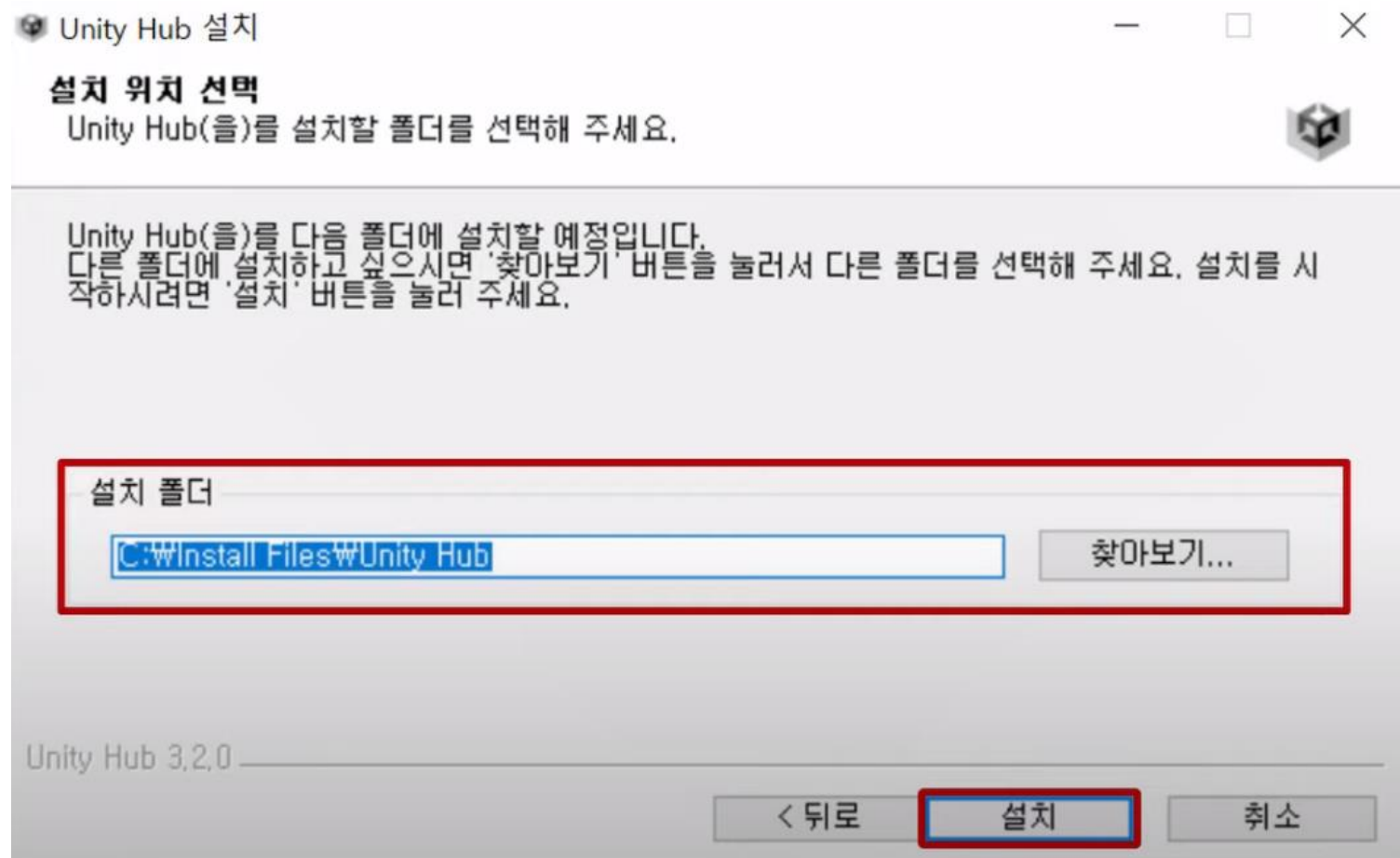
전 세계적으로 가장 큰 인기를 누리는 2D/3D 멀티플랫폼 게임 및 인터랙티브 경험 개발 플랫폼을 다운로드하세요.

Windows용 다운로드

다른 Unity 플랜 보기

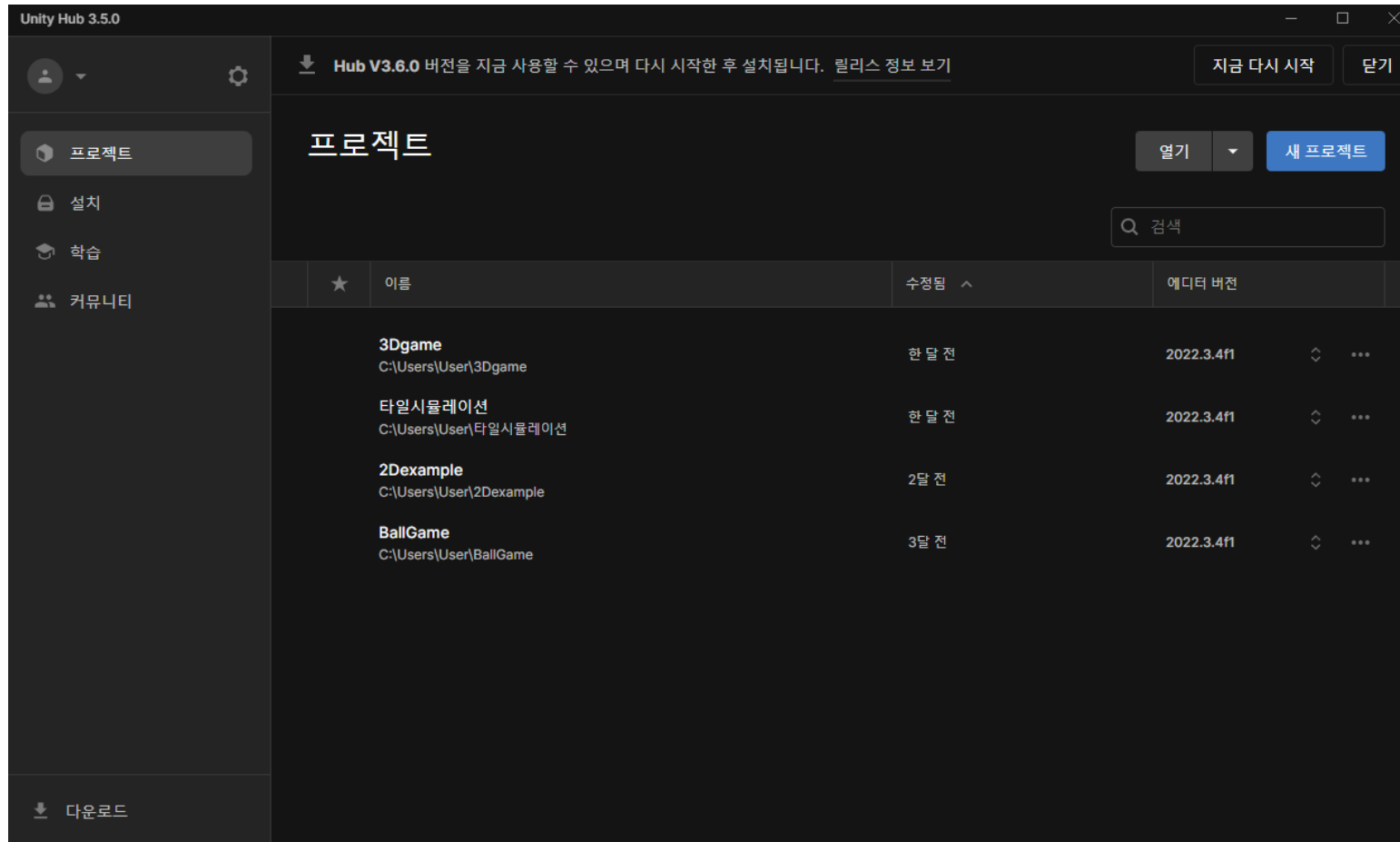
유니티 허브(Unity Hub) 다운로드

"Window용 다운로드" 버튼 선택



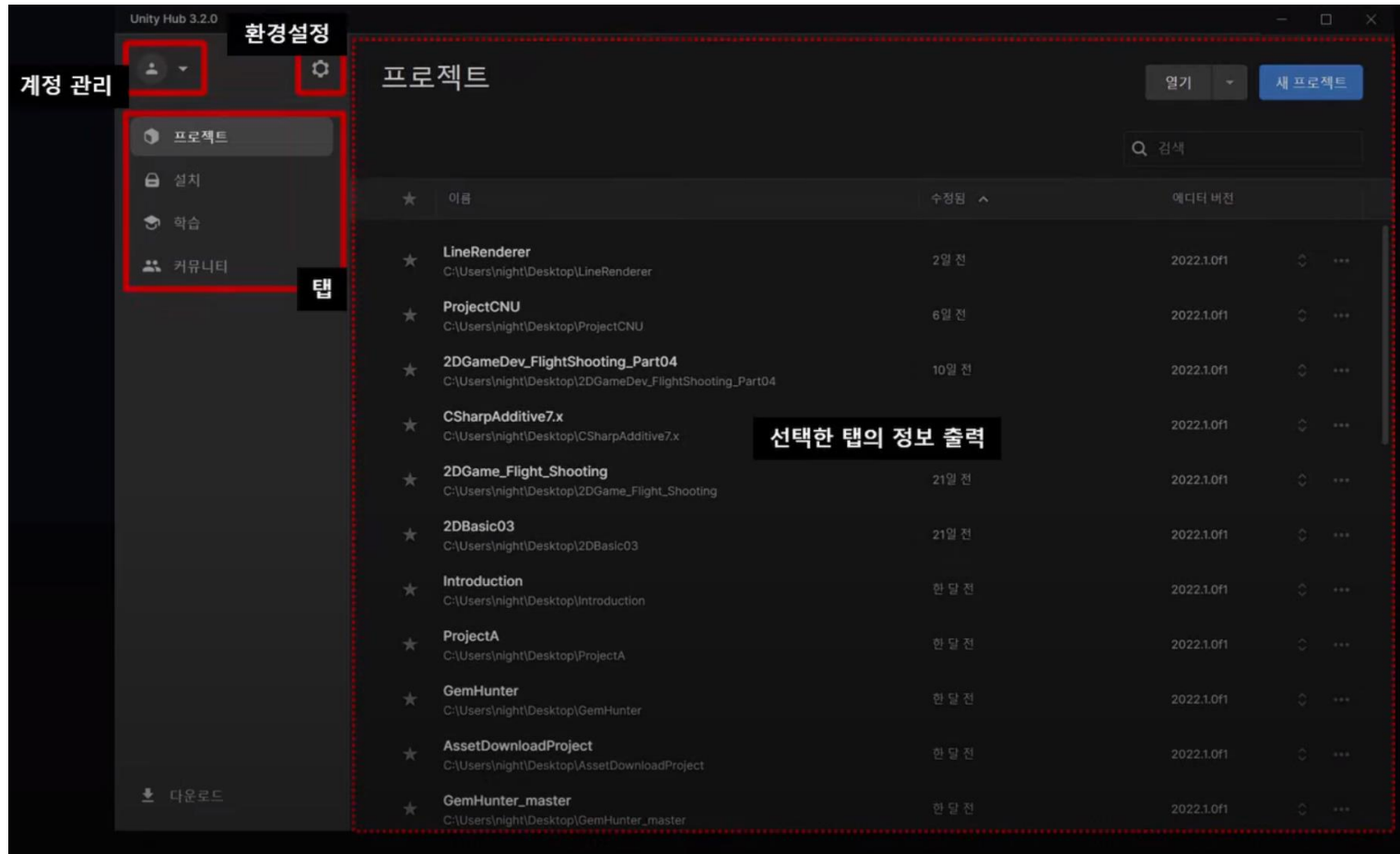
유니티 허브(Unity Hub) 설치

다운로드 한 유니티 허브 프로그램을 더블 클릭해서 실행한 후 설치



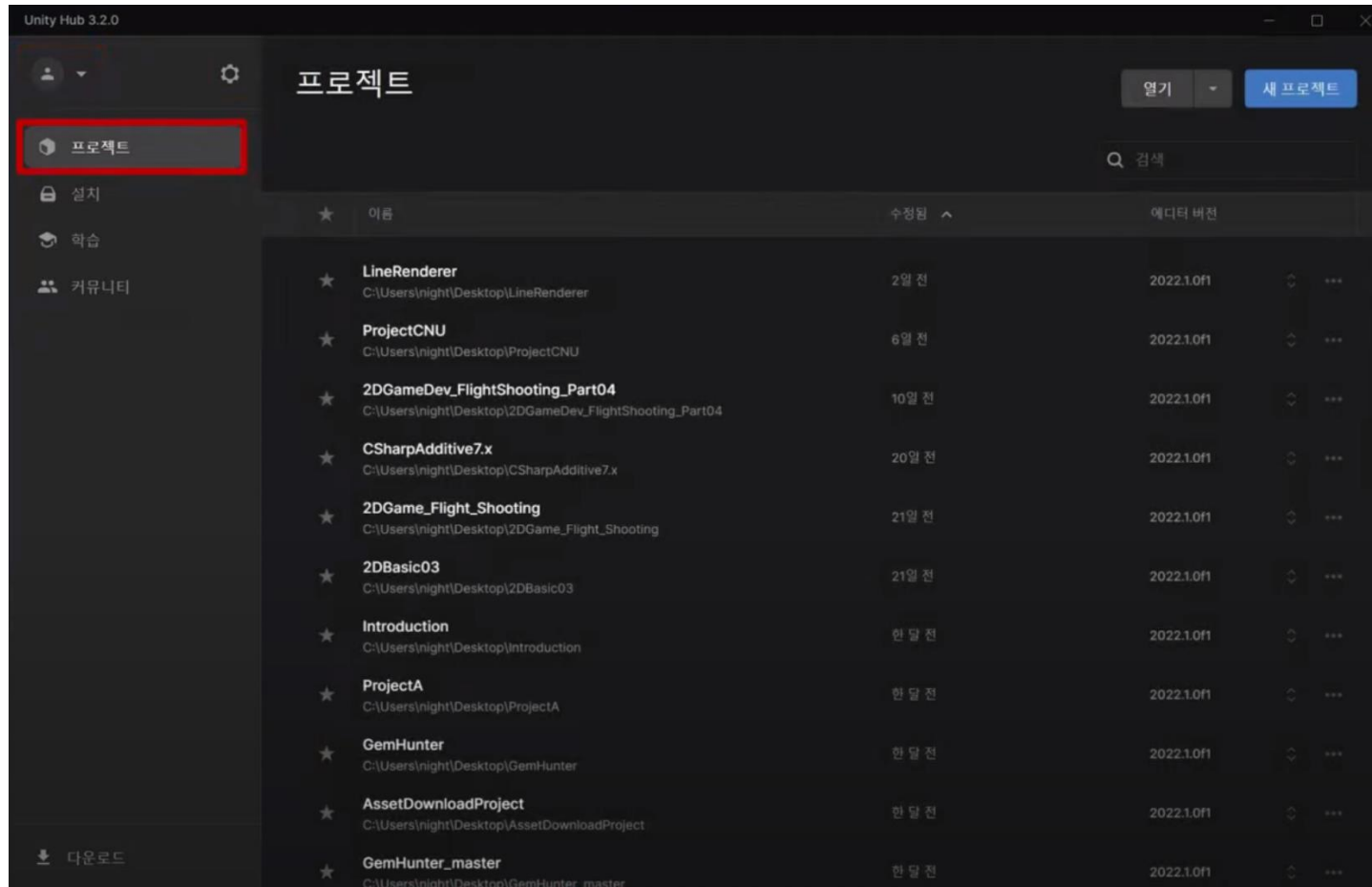
유니티 허브(Unity Hub)

원하는 버전의 유니티 에디터를 설치/실행할 수 있는 버전 관리 프로그램



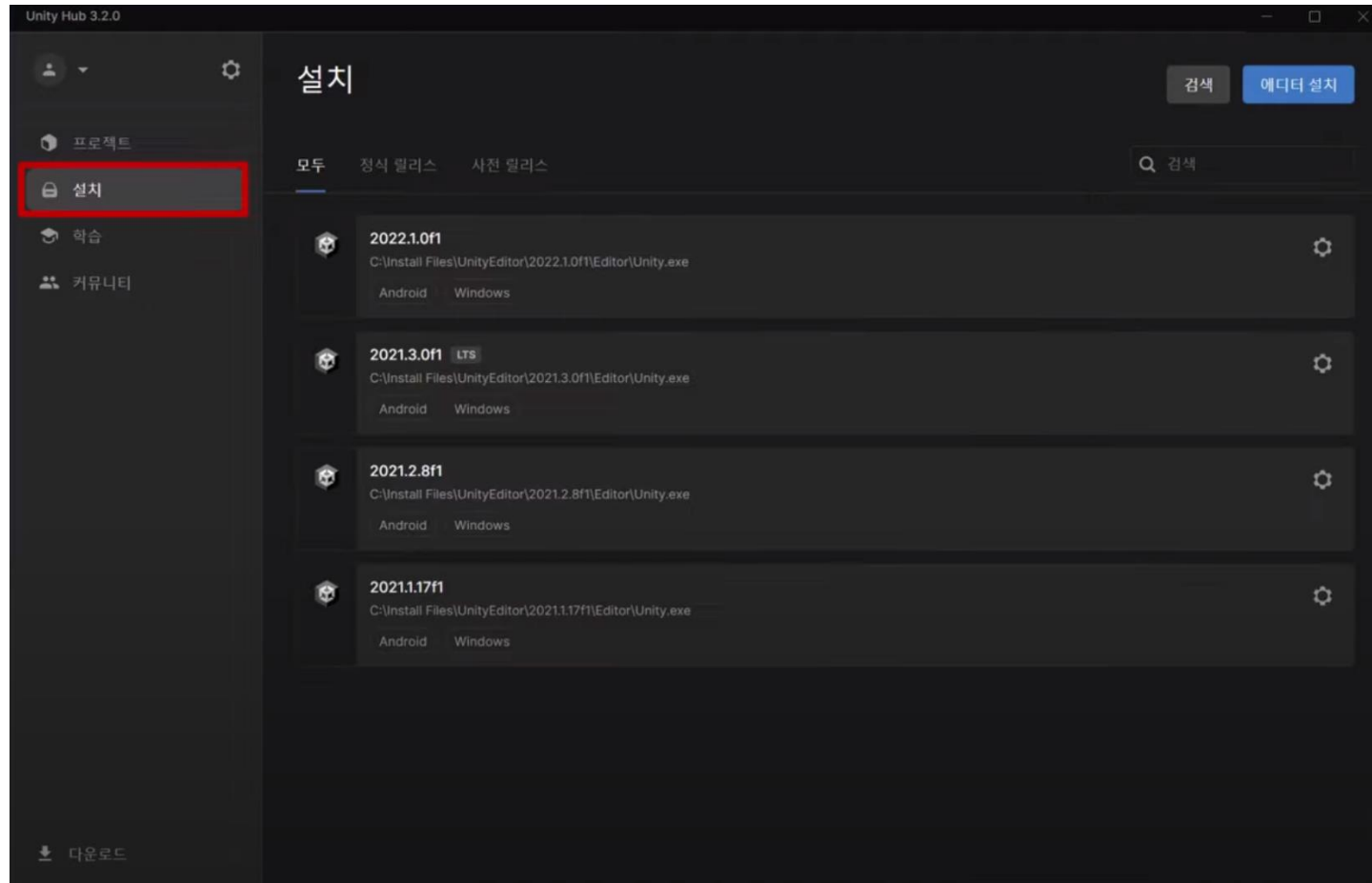
유니티 허브(Unity Hub)

원하는 버전의 유니티 에디터를 설치/실행할 수 있는 버전 관리 프로그램



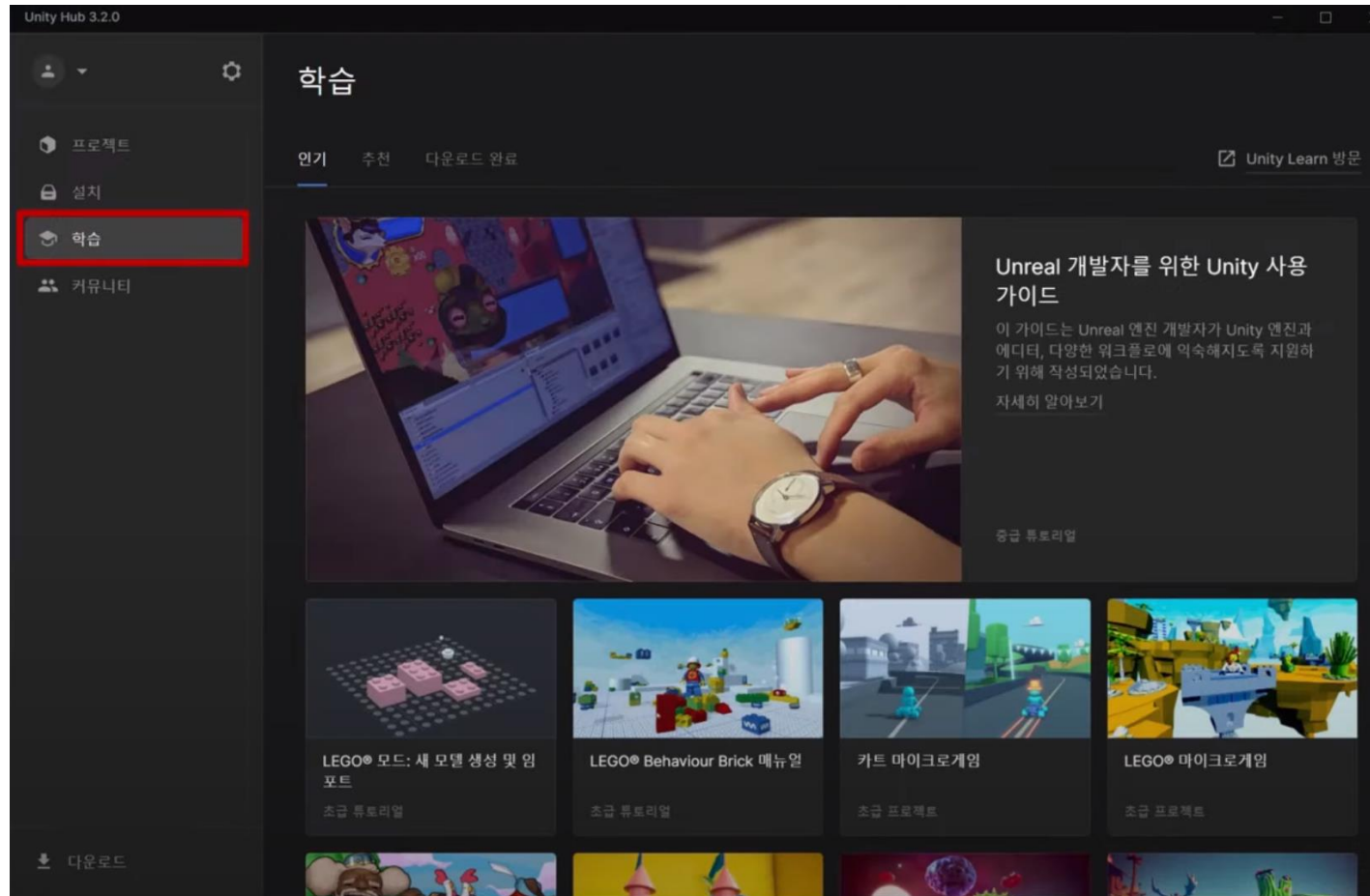
유니티 허브(Unity Hub)

프로젝트 원하는 버전(설치되어 있는)으로 유니티 프로젝트 생성 및 열기



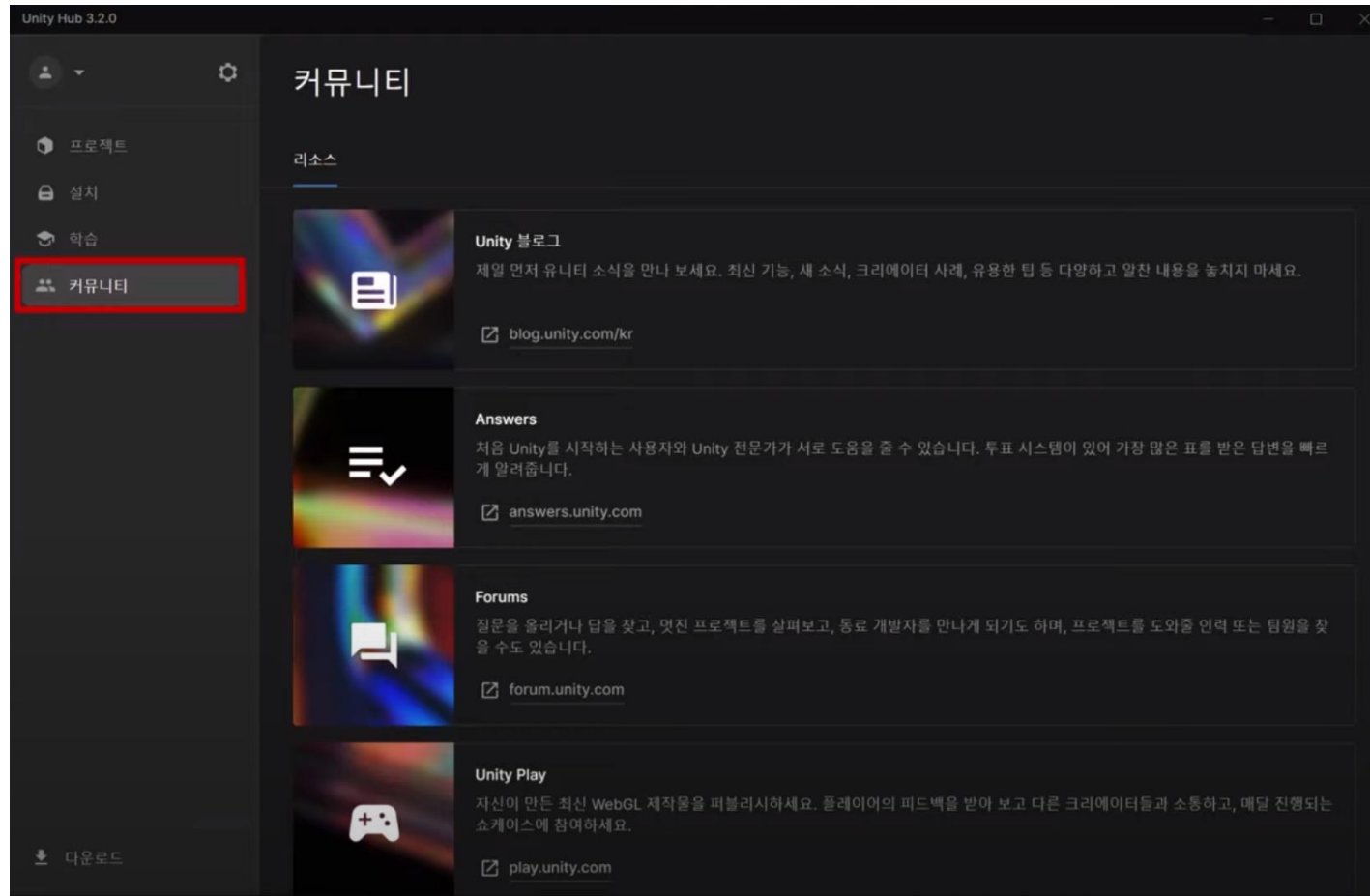
유니티 허브(Unity Hub)

설치 원하는 버전의 유니티 에디터(Unity Editor) 설치 및 삭제



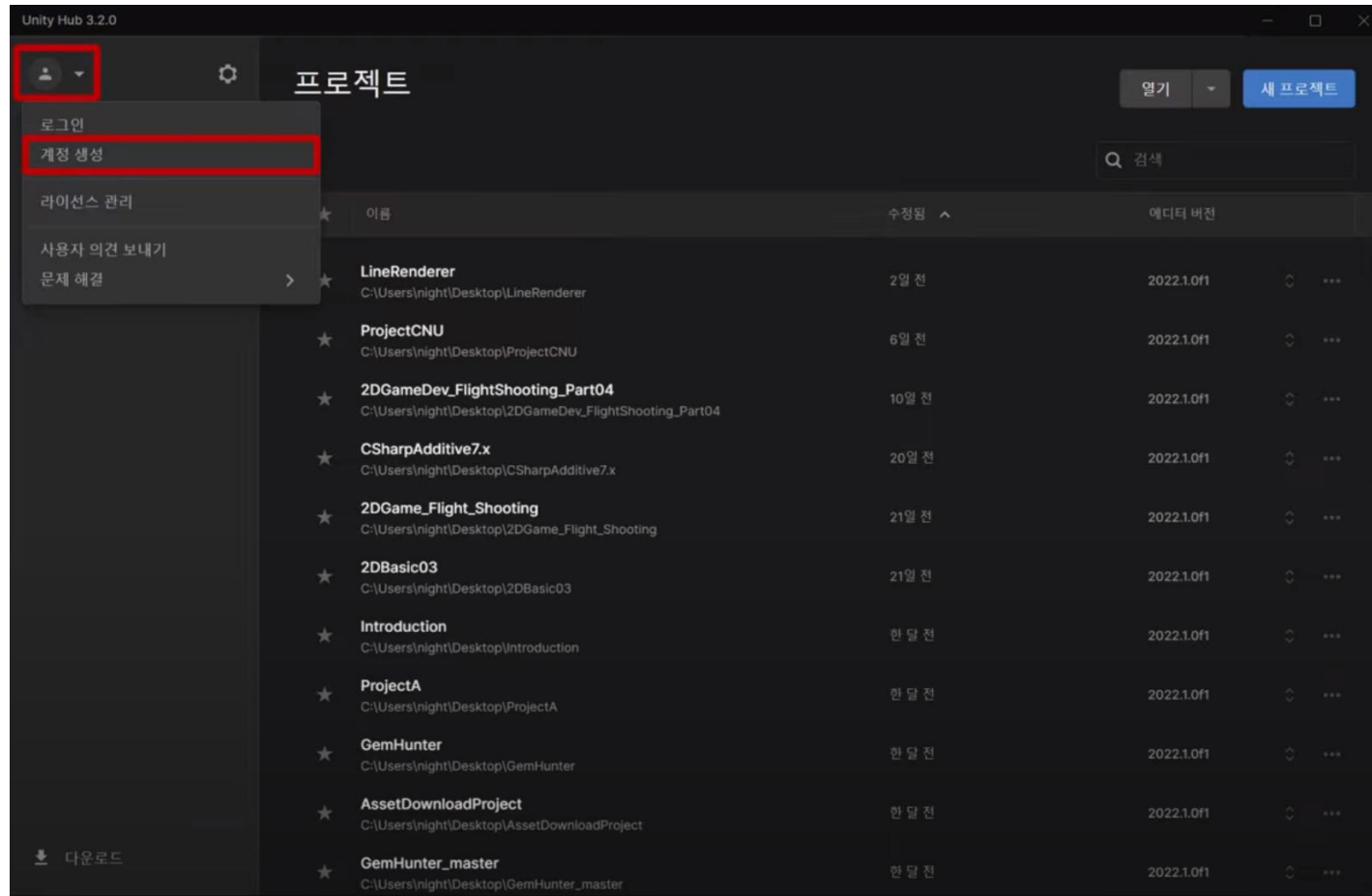
유니티 허브(Unity Hub)

학습 유니티에서 제공하는 유니티 콘텐츠 학습 자료 (튜토리얼, 프로젝트)



유니티 허브(Unity Hub)

커뮤니티 유니티 블로그 등 여러 정보를 얻을 수 있는 사이트 링크 목록



유니티 계정 생성

계정 관리 - 계정 생성



계속하려면 로그인해주세요

Unity ID가 없으시면 **새로 ID를 만드세요.**

유니티 계정 생성

이메일

암호

☒ 기억하기

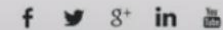
[암호를 잊으셨나요? Help](#)

로그인

OR

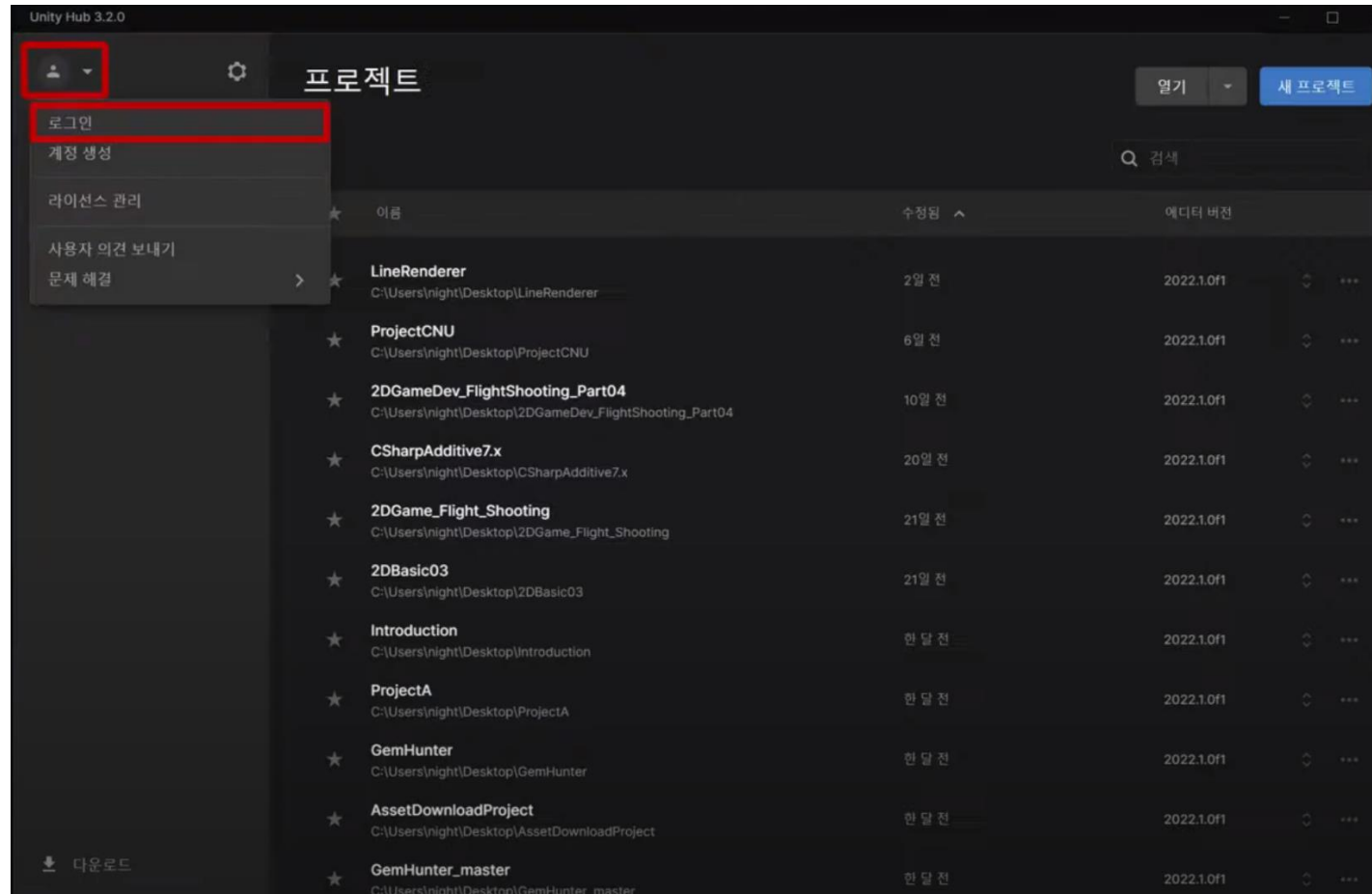


외부 계정 연동



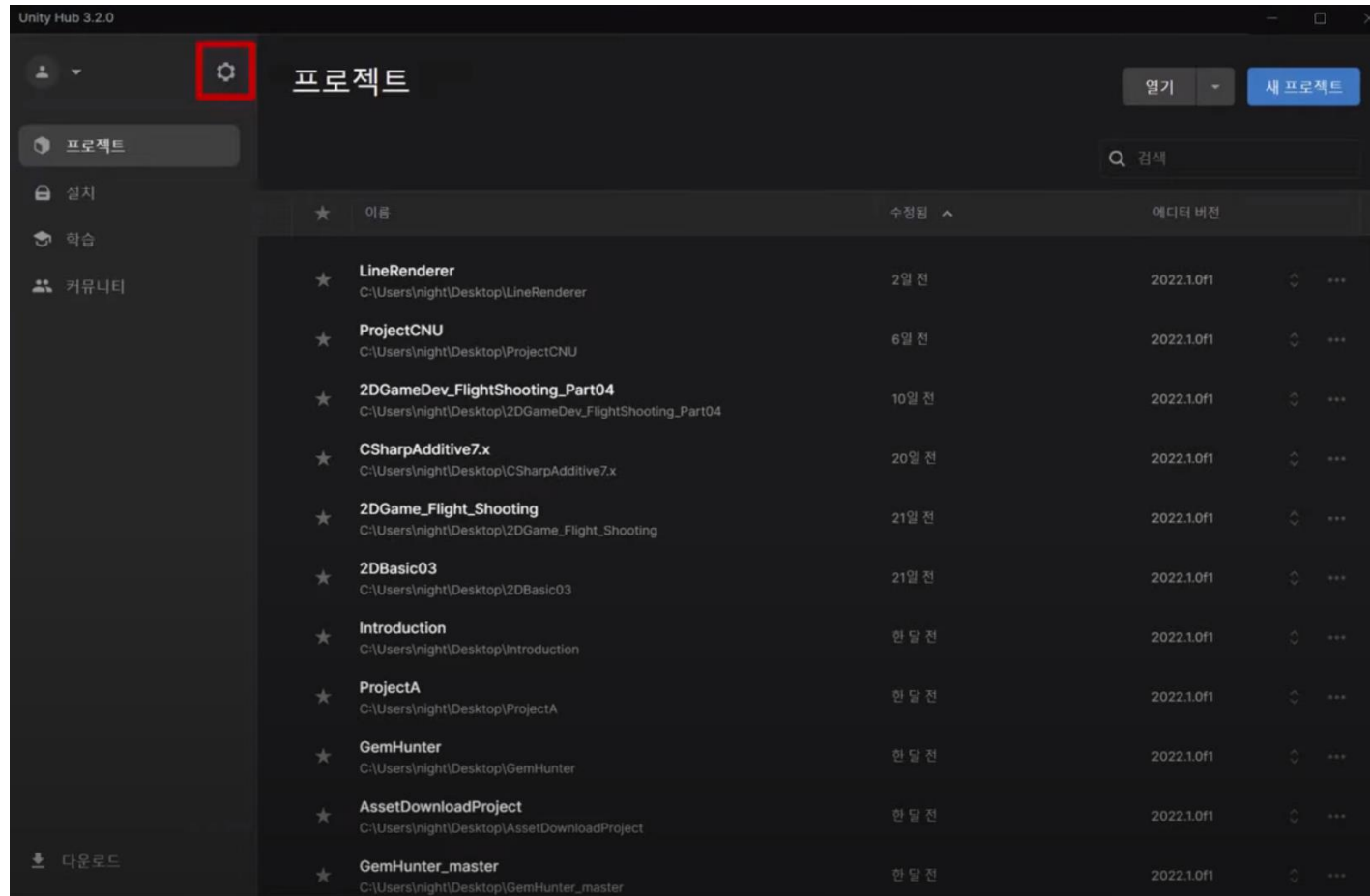
유니티 계정 생성

유니티 계정을 생성하거나 외부 계정 연동



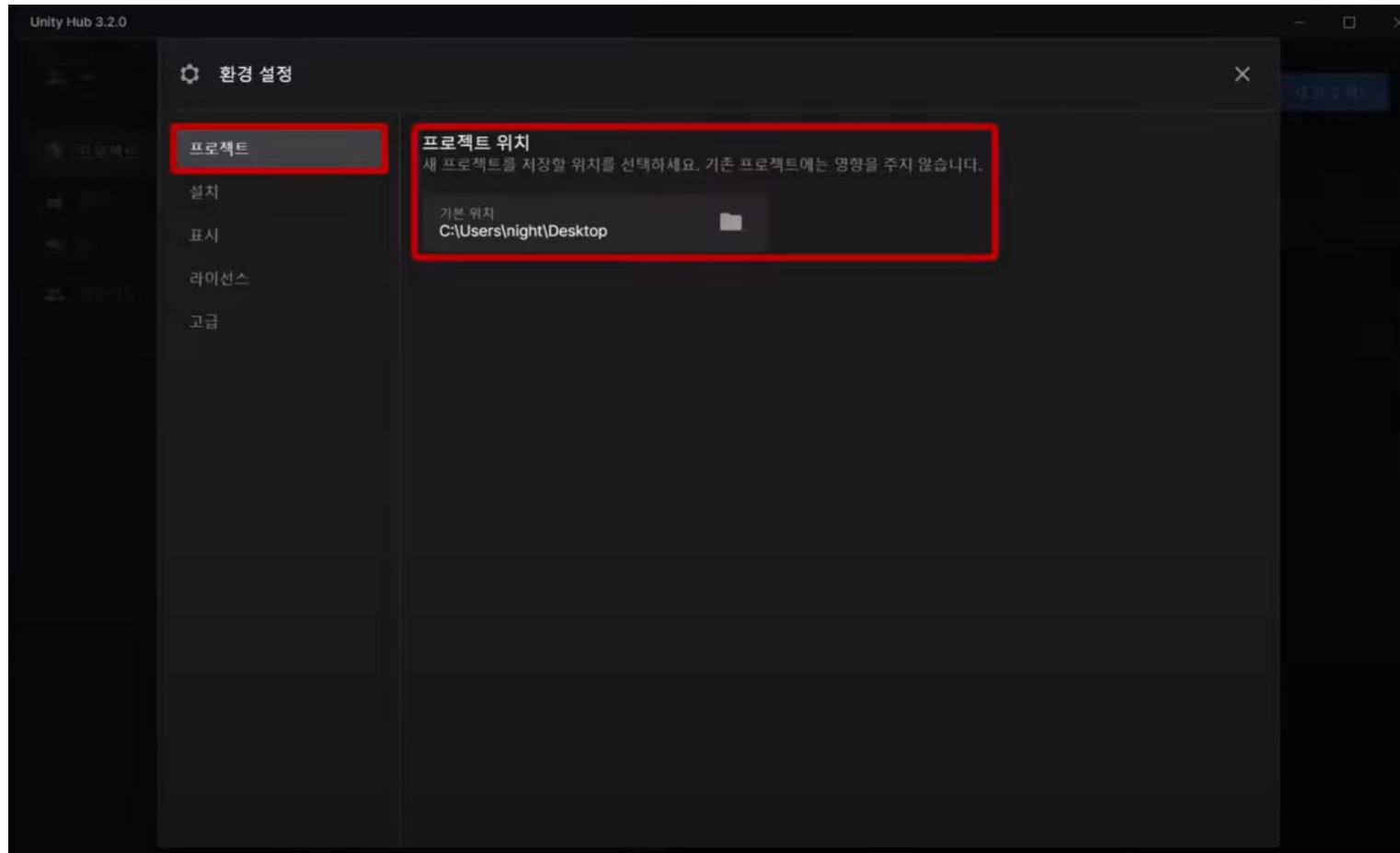
유니티 계정 로그인

계정 관리 - 로그인



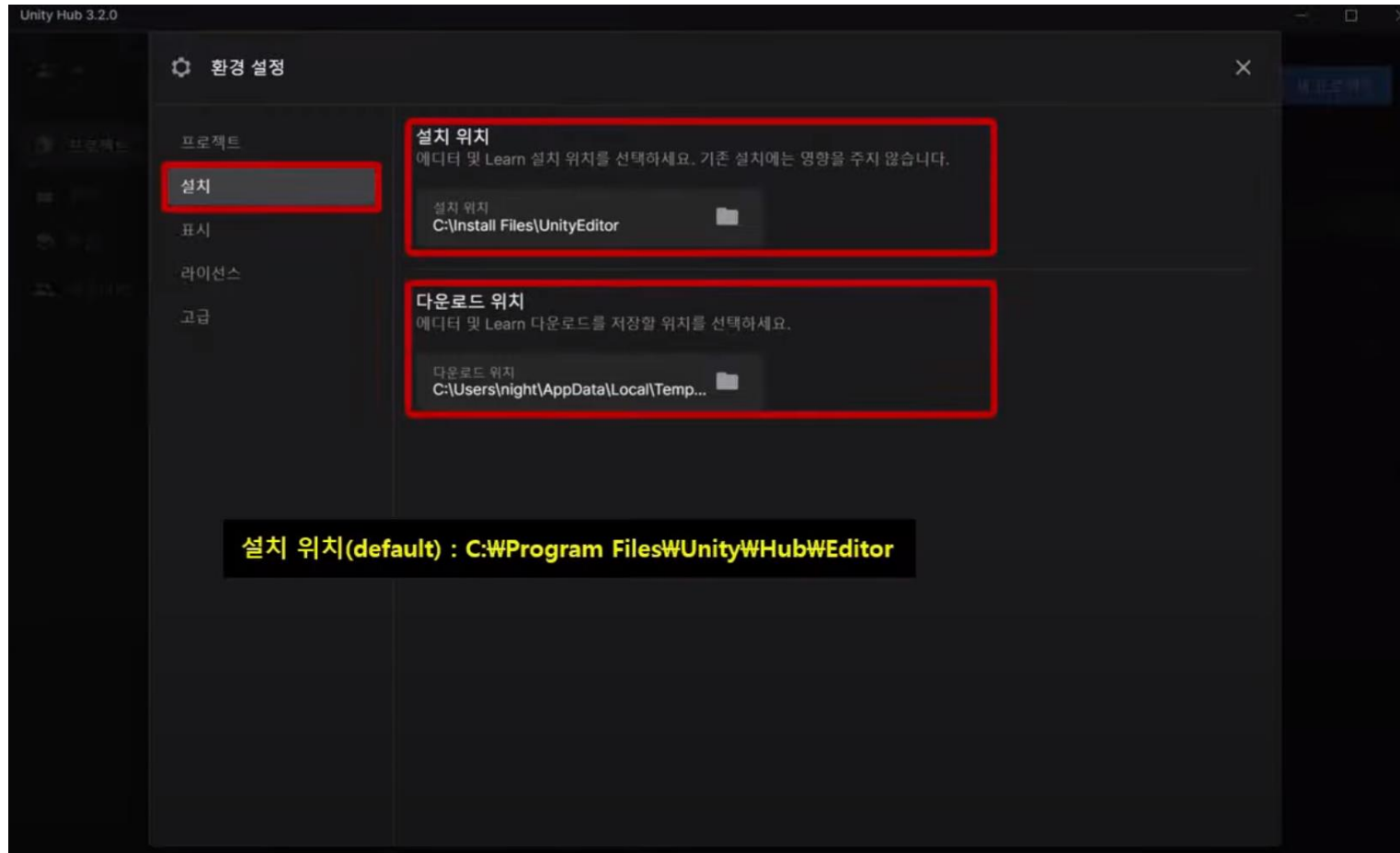
환경 설정

환경 설정 버튼을 눌러 환경 설정 창 열기



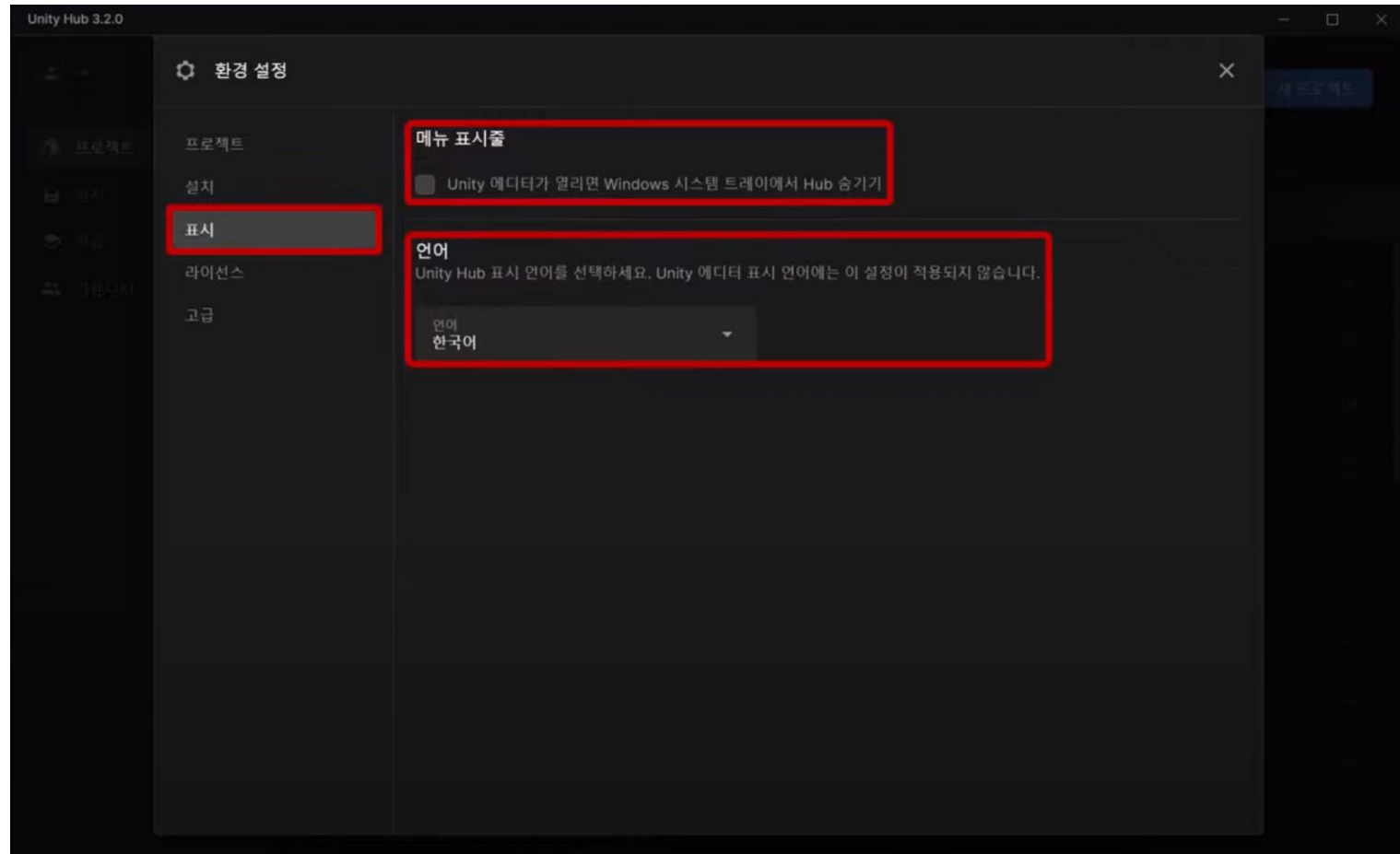
환경설정

새 프로젝트 저장 위치



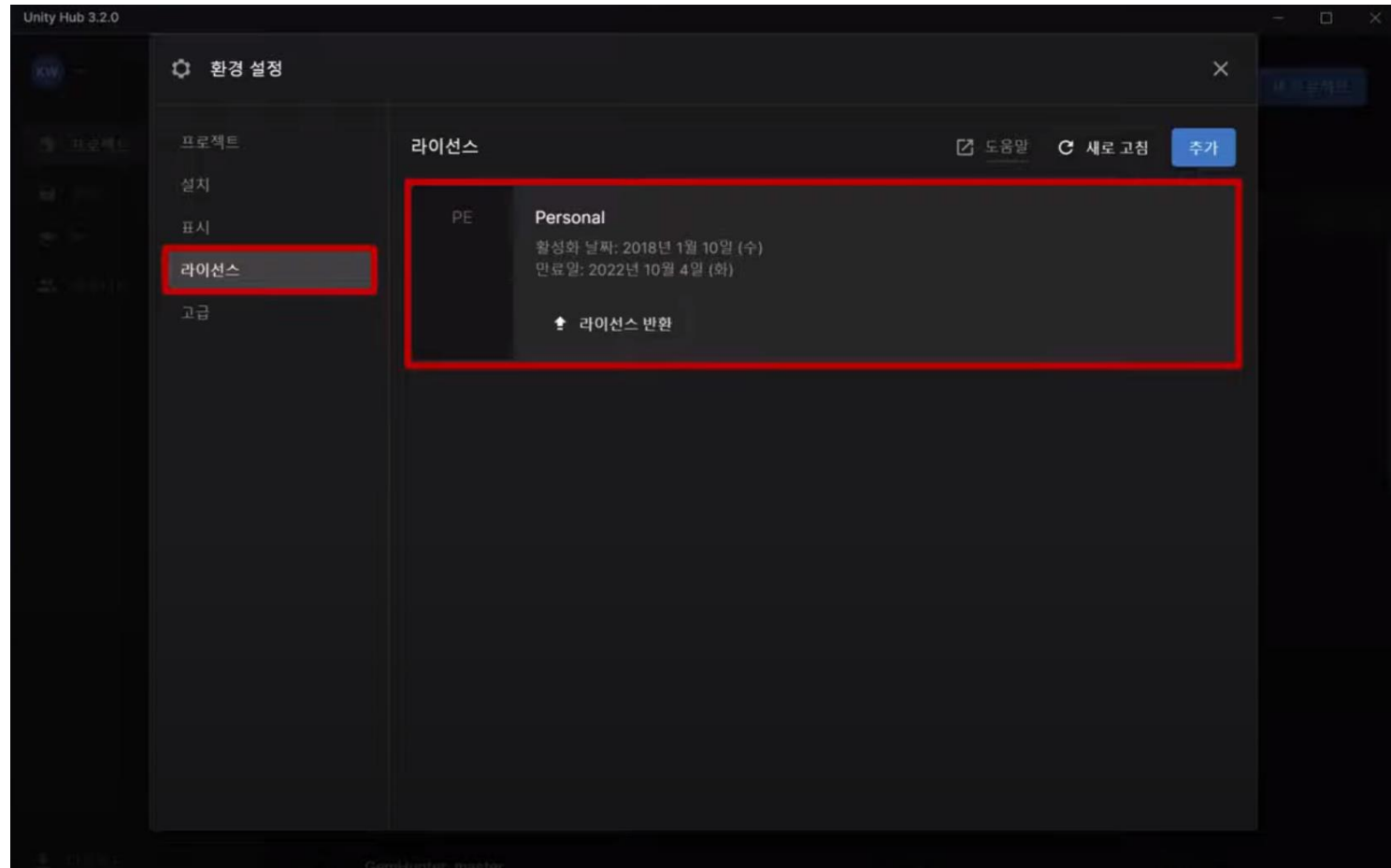
환경설정

유니티 에디터 다운로드 및 설치 위치



환경 설정

언어, 메뉴 표시줄 옵션



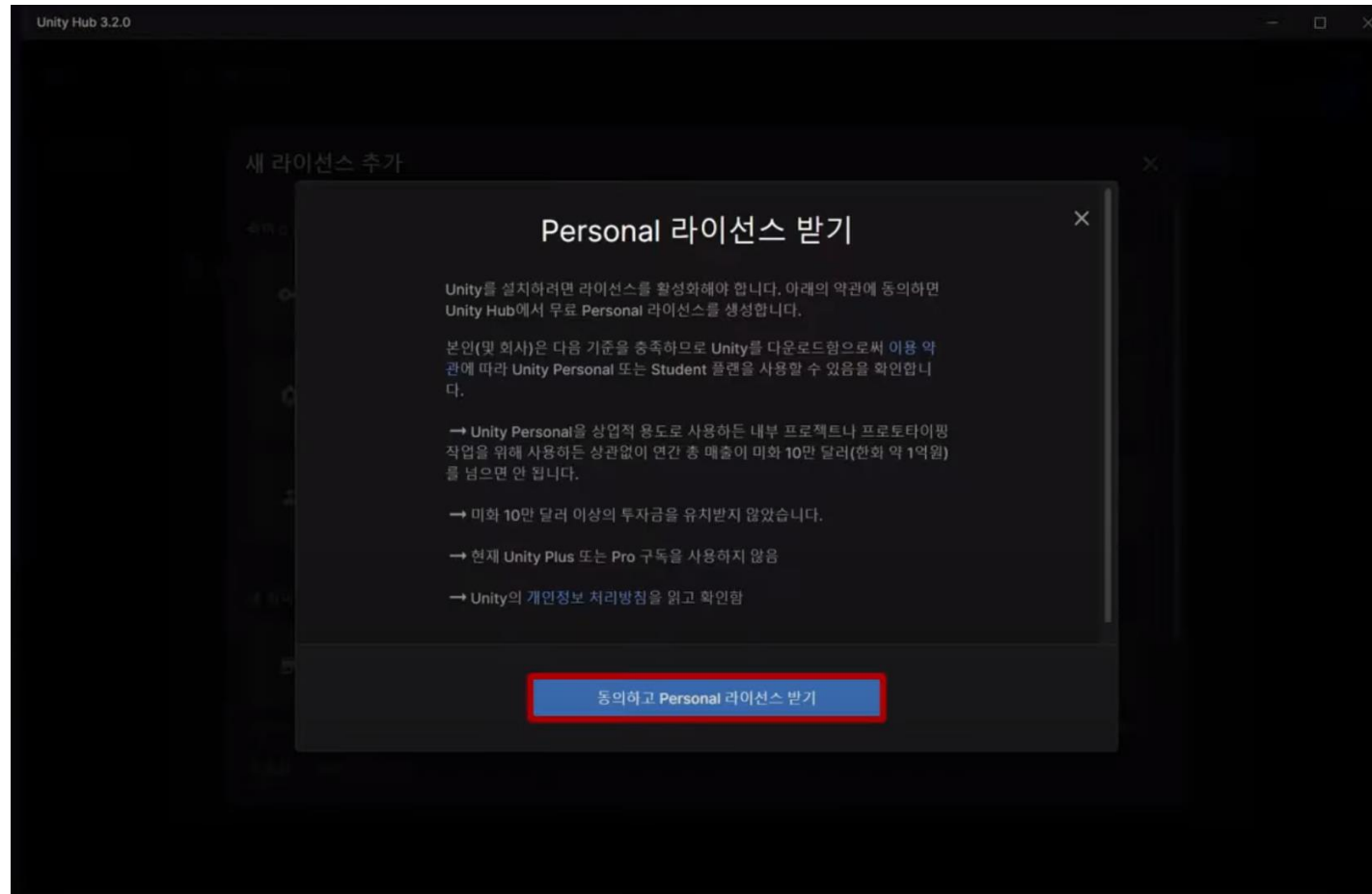
환경설정

라이선스 추가 및 확인 (라이선스가 있어야만 진행이 가능하므로 무료 라이선스인 학생이나 개인을 만들어줍니다.)



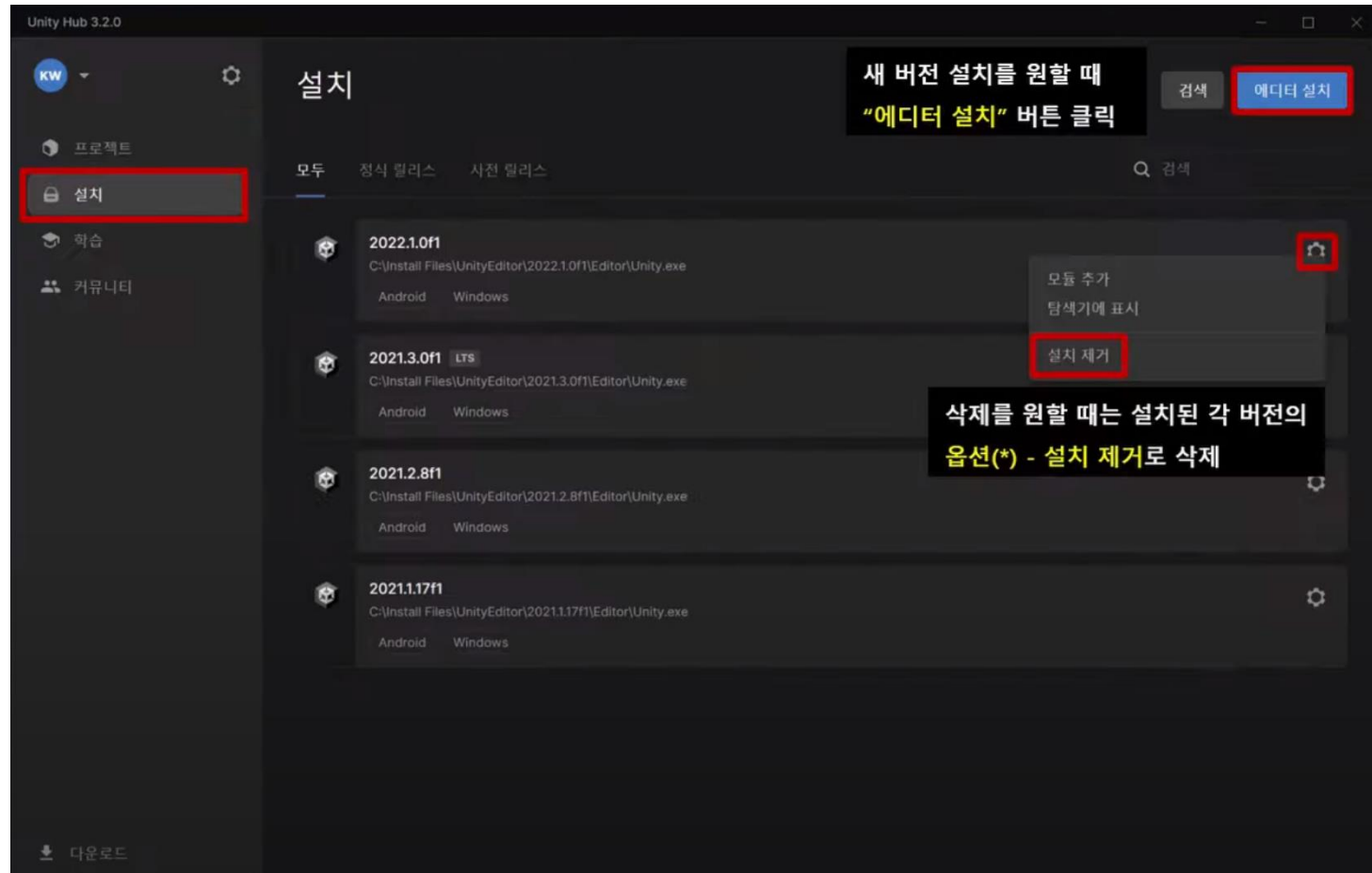
환경설정

라이선스 추가 및 확인



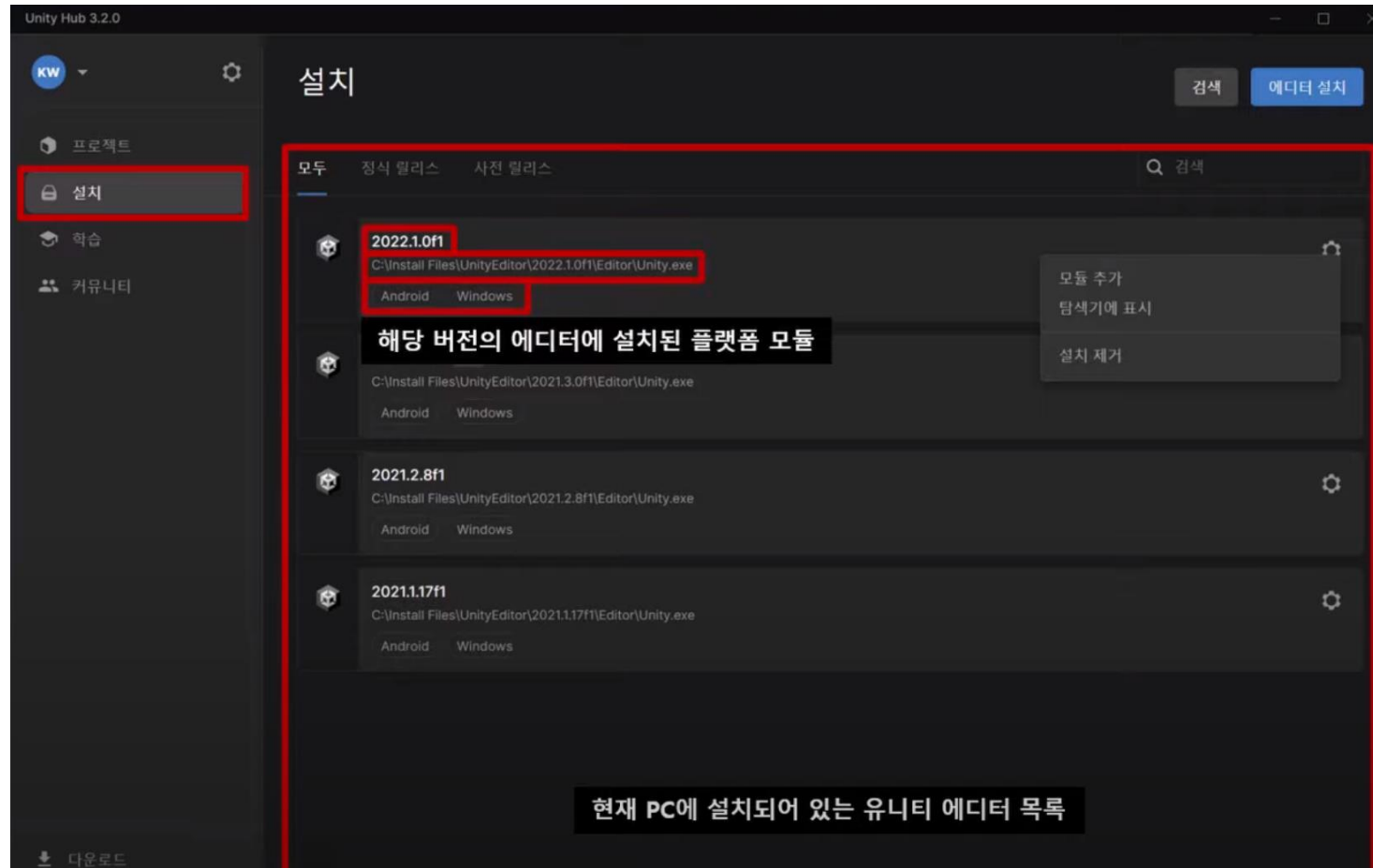
환경설정

라이선스 추가 및 확인



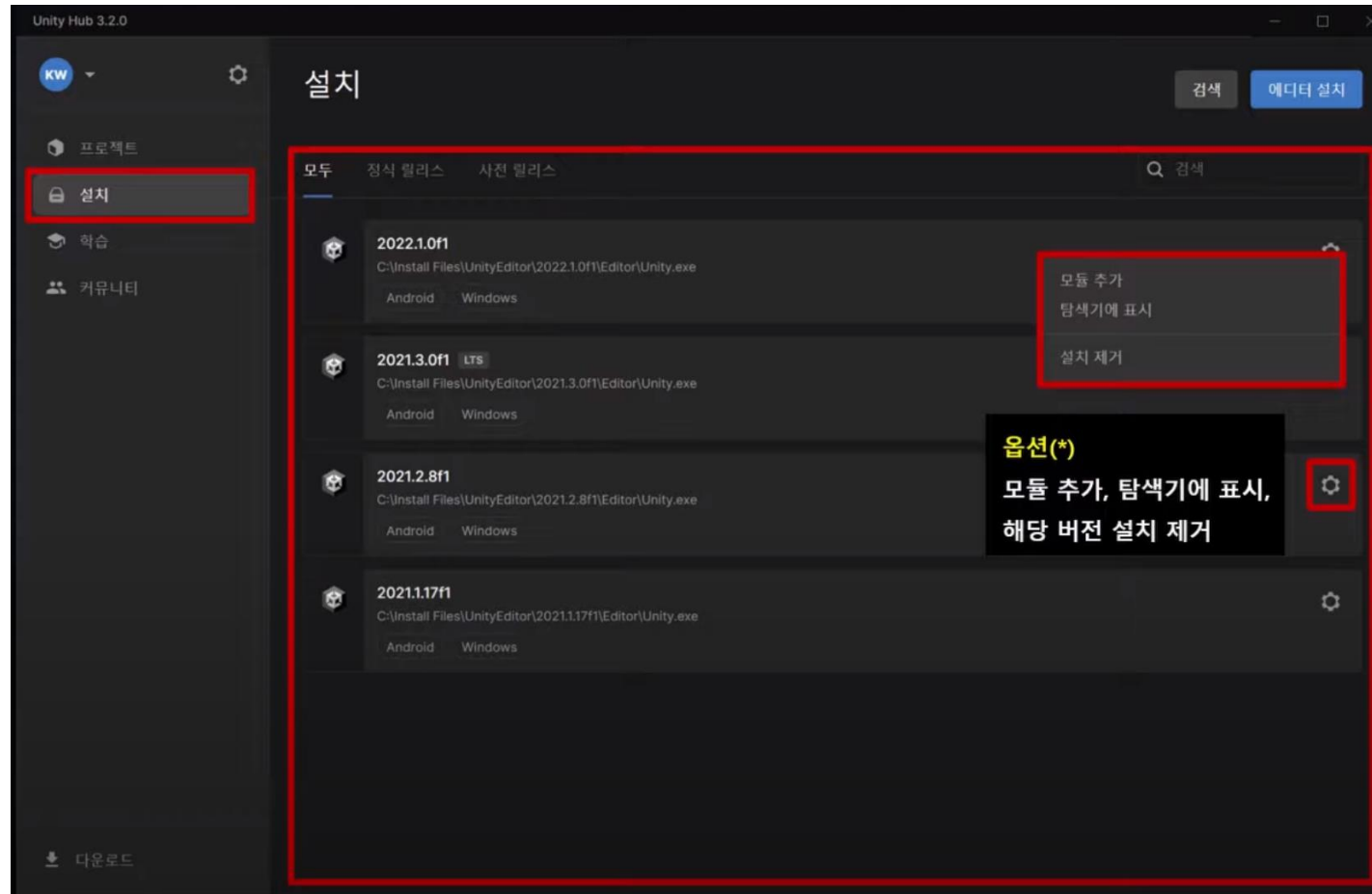
유니티 에디터 (Unity Editor)

유니티 에디터 설치 탭



유니티 에디터 (Unity Editor)

유니티 에디터 설치 탭



유니티 에디터 (Unity Editor)

유니티 에디터 설치 탭

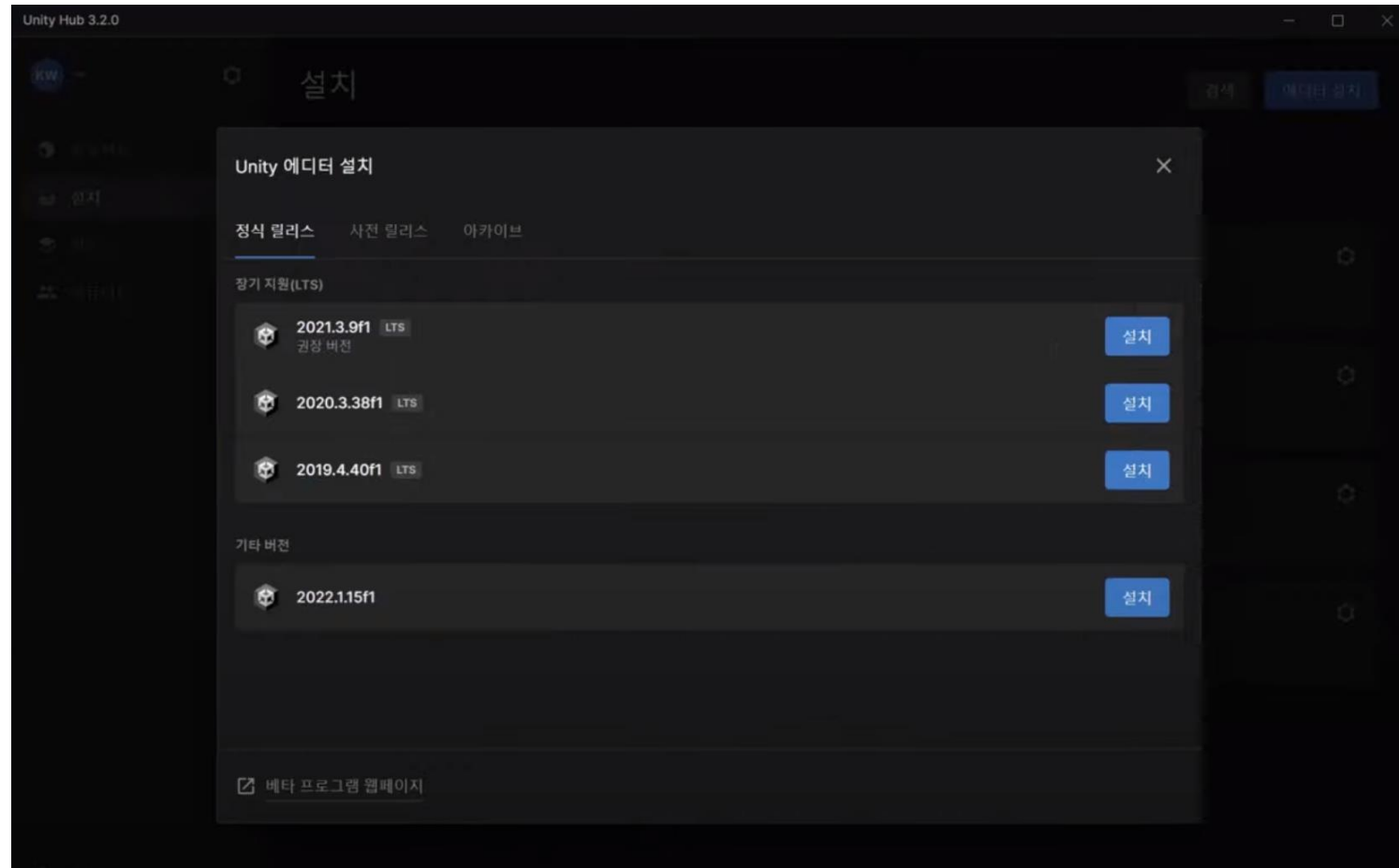
유니티 에디터 버전 설명

유니티의 버전은 A.B.C f1과 같이 관리됩니다.(ex. 2022.1.0f1) [a : 알파, b : 베타, f : 파이널]

A 숫자는 년도를 의미하며 엔진 기능이 대폭 업데이트 된다. [버전 간 호환이 원활하지 않을 수 있다.]

B 숫자가 다를 경우 버전 간 호환에 약간의 문제가 발생할 수 있다.

C 숫자는 미미한 추가 기능이나 Bug Fix와 같은은 수정이기 때문에 호환성에 문제가 없다.



Unity 에디터 설치 창

정식 릴리스 탭

Unity 에디터 설치 창

장기 지원 (LTS) 버전

정식 릴리스 : 현재 정식 버전 중 최신 버전과 장기 지원(LTS) 버전

장기 지원(LTS, Long Term Support) 버전은 가장 안정된 버전으로 실제 개발을 진행한 한다면 해당 버전을 설치 하는 것이 좋다.

Unity 에디터 설치 창

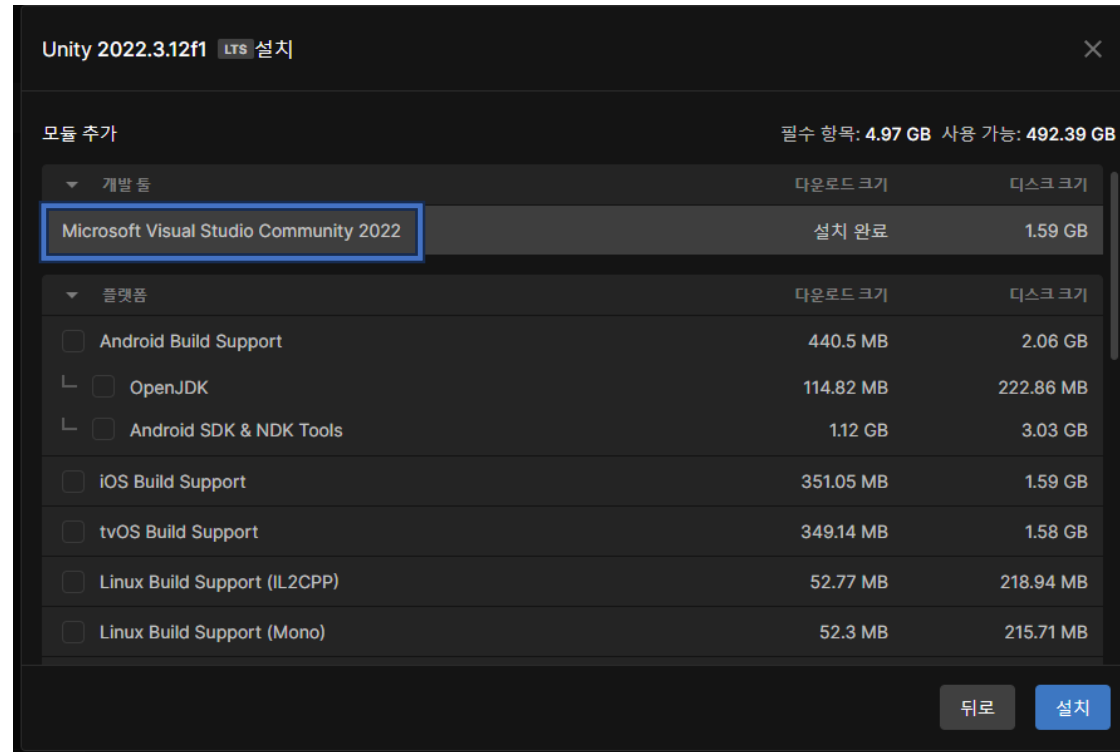
- 기타 버전
- 정식 출시(f, final) 버전 중 가장 최신 버전

Unity 에디터 설치 창

- 사전 릴리스 탭
- 사전 릴리스 : 알파(a. Alpha) , 베타(b, beta) 버전

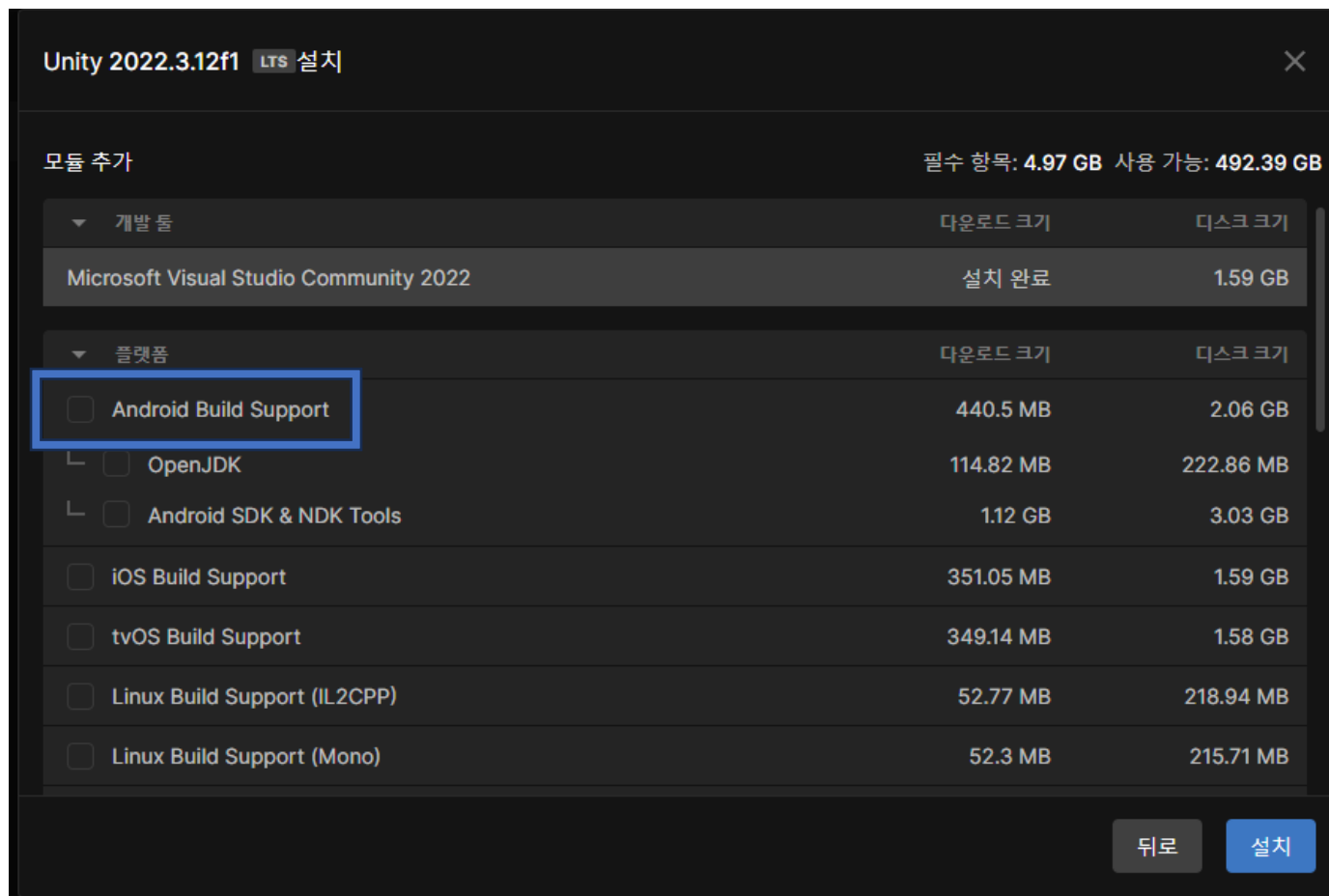
Unity 에디터 설치 창

- 아카이브 탭
- 아카이브 : 허브에 표시되지 않는 이전 버전 다운로드
- 유니티 허브는 각 버전의 최신 버전만 표시하기 때문에 예전 버전을 다운로드 할 때는 홈페이지로 이동



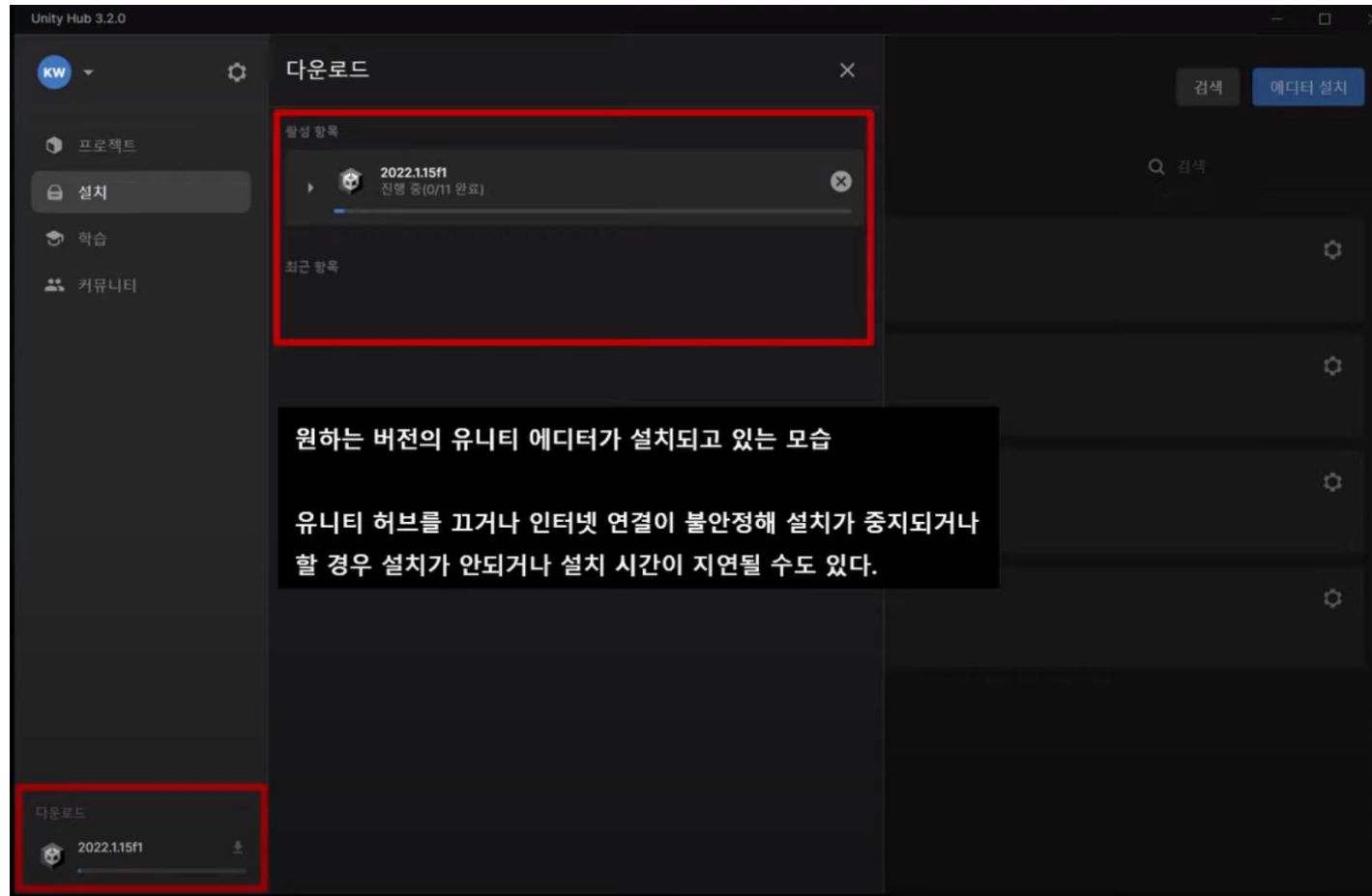
유니티 에디터 설치

- 코드 편집 툴 Visual Studio Community 2022 설치
- 코드를 편집하는 Visual Studio가 설치되어 있지 않을 경우 "Microsoft Visual Studio Community 2022" 체크



유니티 에디터 설치

Android Build Support 플랫폼 설치



유니티 에디터 설치

다운로드 탭에서 다운로드 및 설치 상태 확인

Git을 이용한 Unity 버전 관리

+ github repository 작성시 확인할점

Unity 프로젝트용 repository를 생성할 때 Unity 프로젝트 관리를 위한 "gitignore" 파일을 추가하여 줘야한다. 유니티에서는 에디터를 켤 때마다 내용이 바뀌거나 굳이 다른 개발자들과 공유하지 않아도 프로젝트를 진행하는 데 문제가 없는 파일들이 꽤 많이 존재합니다. 이러한 파일까지 다 올리면 용량량이 많이 차지할 뿐만 아니라, 필요한 파일을 찾아보기도 힘들어지기 때문에 .gitignore 파일에서 명시적으로 해당 파일은 remote repository에 업로드 하지 않습니다. Add. Ignore 버튼을 누르지 않더라도 git ignore 파일을 따로 다운받아 해당 remote repository에 업로드하면 동일하게 적용가능합니다.

Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * BankBoy22 / Repository name * Unity_project
✔ Unity_project is available.

Great repository names are short and memorable. Need inspiration? How about [glowing-octo-waddle](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

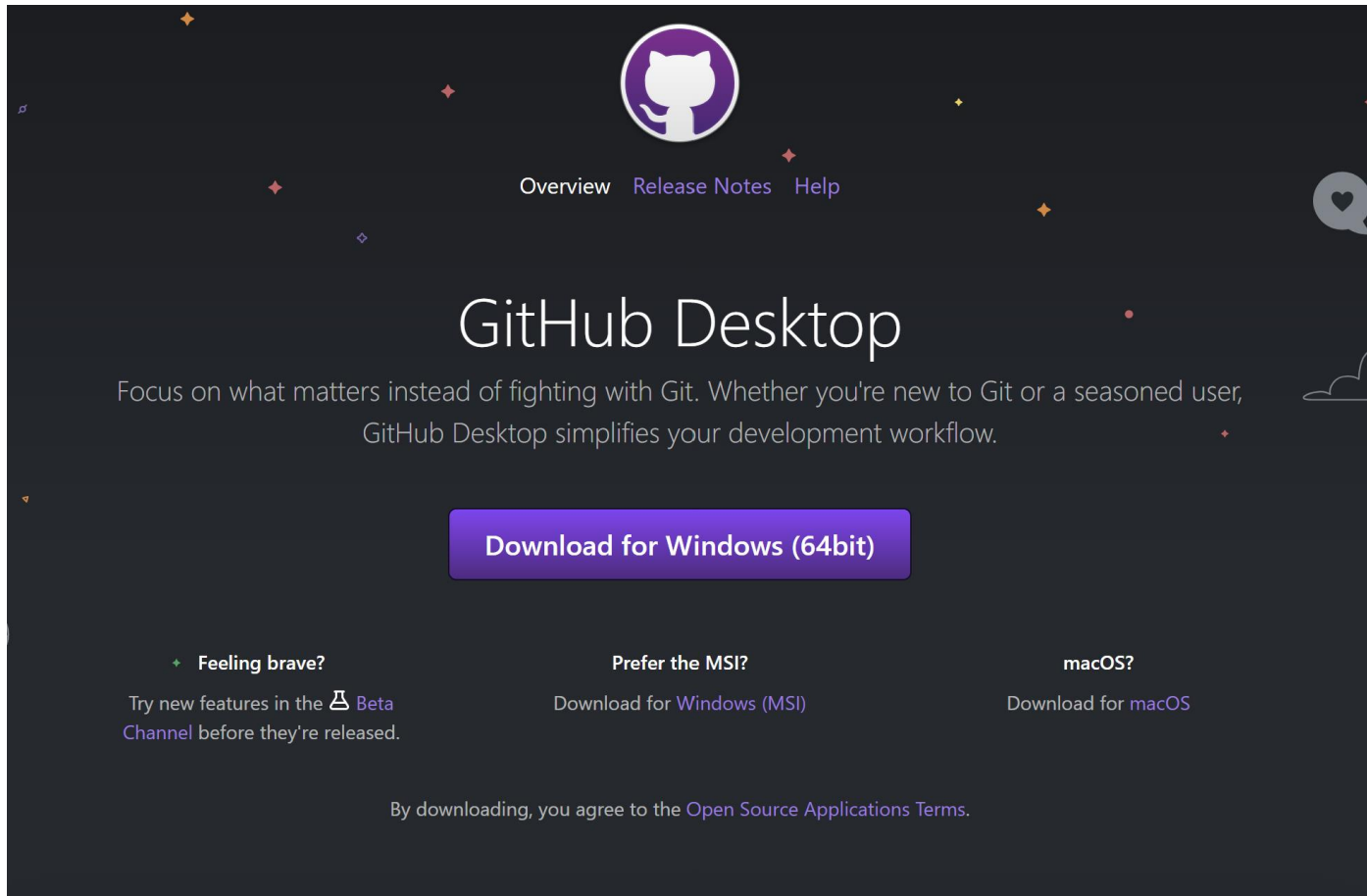
Add .gitignore
.gitignore template: Unity

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license
License: None

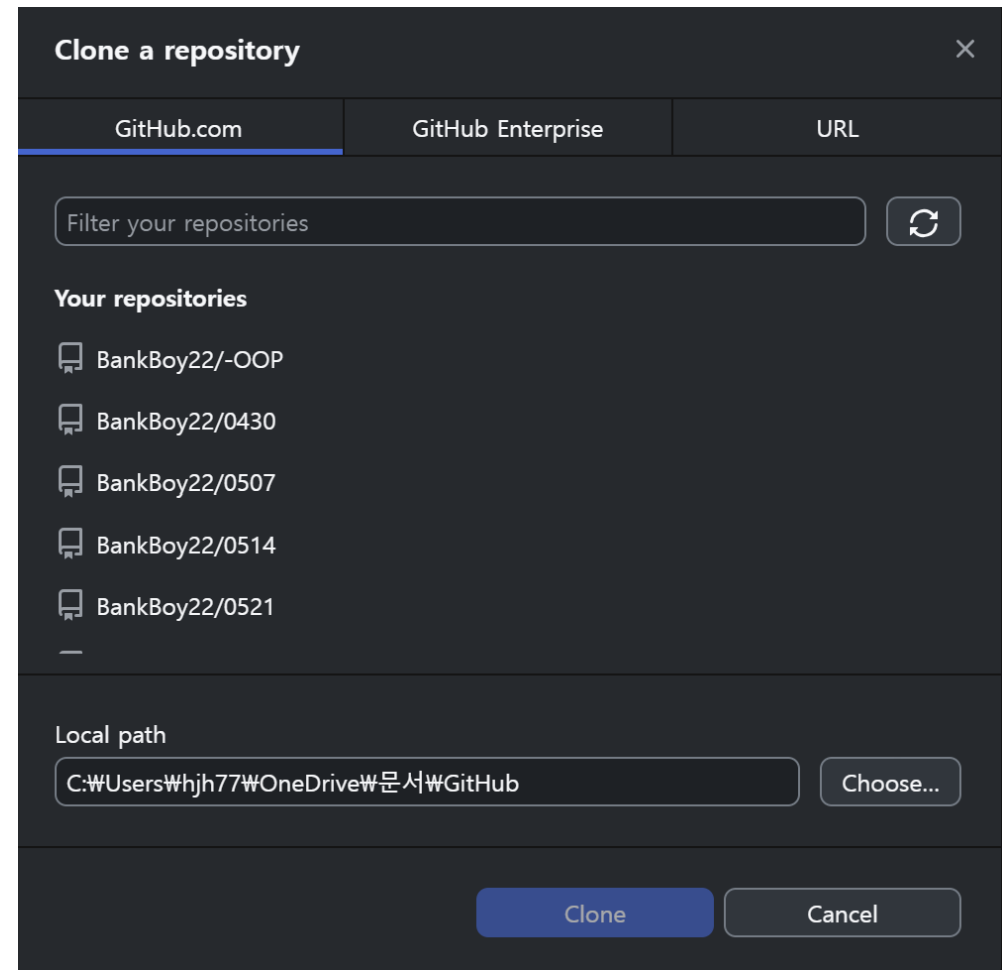
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

"Git Hub" 데스크탑 설치



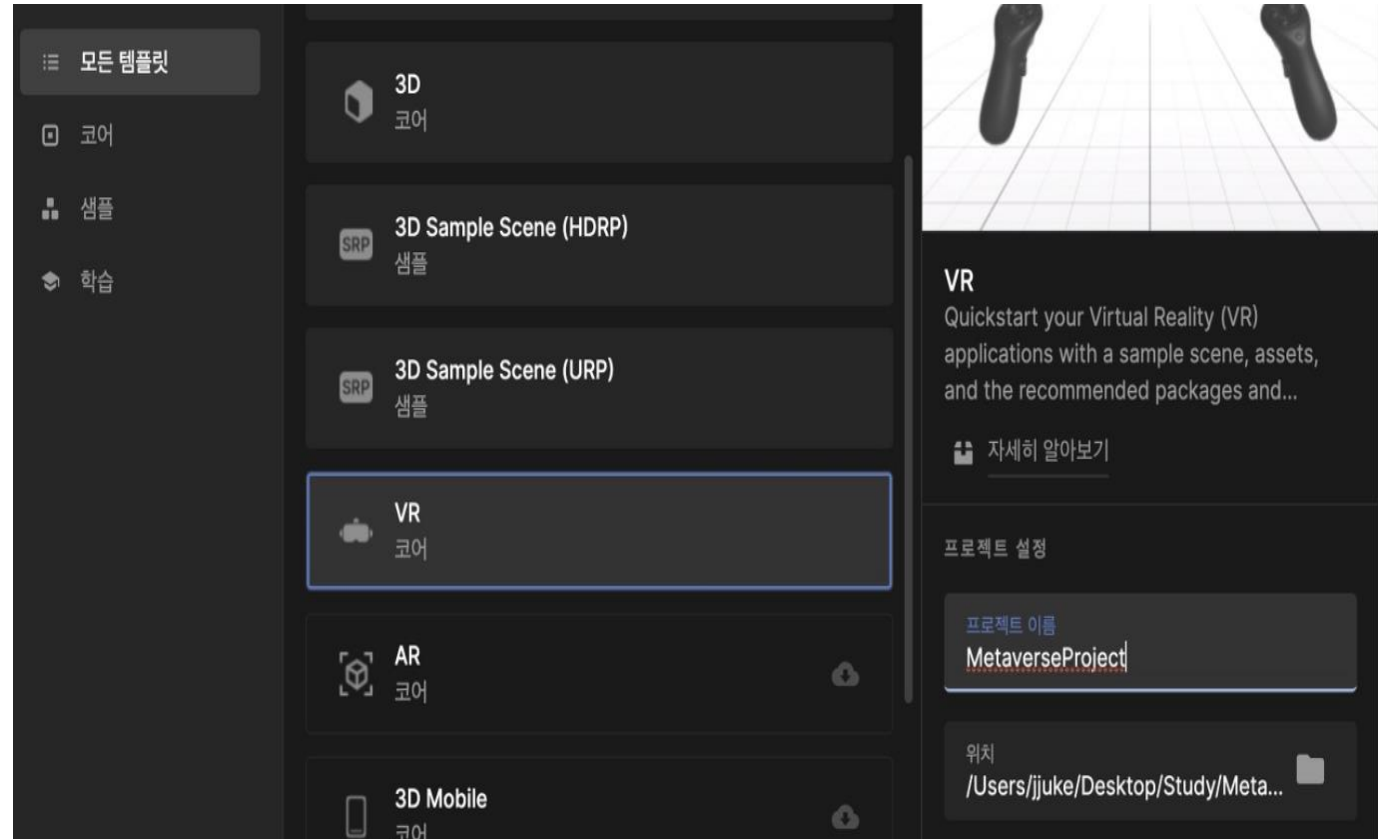
Local repository 생성

- Github Desktop을 열어서 Clone Repository를 통해 로컬 리포지토리를 생성해줍니다.



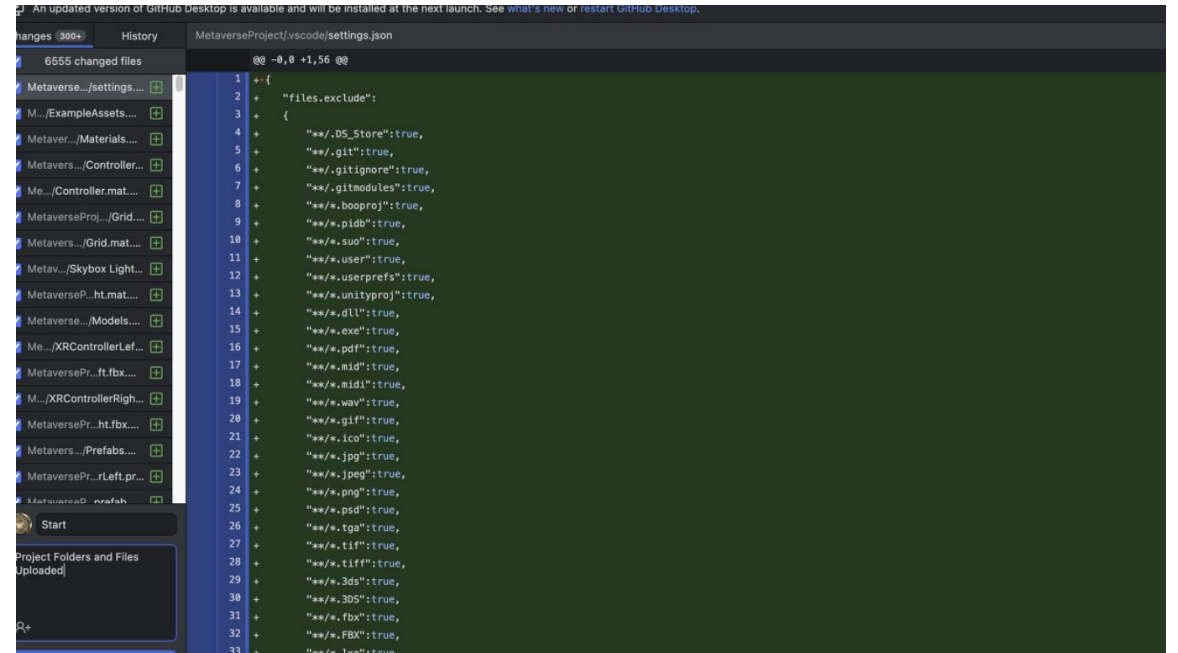
Unity 프로젝트 생성 및 푸시

- Unity Hub를 통해 새 프로젝트를 만듭니다. 이때 방금 생성한 Local Repository에 프로젝트 폴더를 만들어줍니다.



Unity 프로젝트 생성 및 푸시

- 그 이후, Github Desktop에서 새로 생성된 내용들을 Commit 및 Push 해줍니다.



깃허브의 기본 동작

Github Desktop으로 Gitdmf 관리 할때
기준입니다!

1) Commit

- 일반적으로는 변경된 파일에 대해서 통째로 Commit하고, 어떤 내용을 Commit 하는지 Summary와 Description을 작성하지만 일부 내용은 빼고 Commit 할 수도 있습니다.

2) Discard

- Discard는 자신의 변경사항을 지우고, 파일을 마지막으로 커밋했던 상태로 되돌리는 것입니다. Reset과의 차이점은 이미 푸시까지 된 상황에서 버전 자체를 되돌리는 것이지만, Discard는 수정된 내용에 대해서 되돌리는 것이다. Commit과 마찬가지로 줄 단위로 Discard가 가능하고, 파일 단위로도 가능합니다.

3) Push

- 푸시는 커밋된 내용을 Remote Repository에 업로드 하는 것입니다.

4) Pull

- Pull은 다른 사람이 작업한 내용, 즉 변경 사항을 Local Repository에 다운로드 하는 과정입니다. 수정한 내용들이 충돌이 발생할 수 있기 때문에, 내용을 수정하기 전에 항상 Pull을 실행하는 것을 습관화 해주는 것이 좋습니다.

5) Merge -1-

- Pull로 가져온 사항과 자신의 변경 사항이 서로 충돌하는 경우에, 어느 커밋을 적용할지 결정하는 동작입니다. 예를 들어, 'Script.cs'라는 파일에 대해서 개발자1은 Fuction1이라는 함수를 사용하여 파일을 수정하였는데, 개발자2가 Function2라는 함수를 사용하여 파일을 수정해두고 푸시까지 해버린 경우에 개발자 1이 commit후 push를 할때 새로운 커밋이 서버에 있으니 패치를 진행하라는 메시지가 뜹니다.

5) Merge -2-

- Fetch를 누르게 되면 pull을 통해 가져와야 하는 Commit이 존재한다고 알려주게 되고, 개발자2가 변경한 내용이 개발자1이 변경한 내용과 관련이 없는 부분이라면 Pull이 가능하다. 하지만 같은 내용을 서로 다르게 수정한 내용이 있다면, 아래와 같이 Conflict(충돌)가 발생하게 됩니다. 이럴 때 'Open in Visual Studio Studio Code' 버튼과 같은 Merge 도구를 이용하여 충돌이 발생한 위치를 알아보고, 서로 협의를 통해 하나의 변경 사항만 선택하여야 합니다. 새로 수정을 한 내용을 선택하려면 **Accept Current**를 선택하고, 먼저 푸시한 사람의 코드를 선택하면 **Accept Incoming Change**를 선택하면 됩니다.

Unity 프로젝트를 Github로 관리할 때 주의점

유니티 프로젝트를 깃허브로 관리하고,
협업을 진행할 때 주의점 입니다.

1)Unity의 버전 통일

- 가장 먼저, Unity의 버전을 통일해 주어야 한다. 호환이 되는것이 아니냐고 생각할 수 있겠지만, 유니티 프로젝트 버전을 관리할 때 가장 중요한 파일인 '.meta'파일의 내용이 유니티 에디터의 버전에 따라 달라질 수 있기 때문에 버전을 통일하는 일은 매우 중요합니다. 따라서 버전이 다르면 메타파일들이 계속해서 불필요하게 변경 및 Commit 되는 문제가 발생할 수 있습니다.

2) Asset 직렬화 방식

- Unity 엔진에서 Prefab이나 Scene등의 파일을 저장하는 방법은 '바이너리', '텍스트', 그리고 이 두가지를 섞어서 사용하는 혼합 방식이 존재합니다. 이 세가지 방식을 Asset 직렬화 방식이라고 한다. 최신 버전의 에디터에서는 기본적으로 텍스트를 사용하도록 설정되어 있는데, 만약 바이너리로 설정된 한명의 개발자가 있다면 깃헙에서 관리가 어려워지고, 충돌이 발생했을 때 Merge 과정이 매우 복잡해 집니다. (바이너리 파일은 사람이 읽어서 해석하기 어려운 내용입니다.) 하지만 텍스트의 경우 어느정도는 사람이 읽어서 해석할 수 있으므로, Asset 직렬화 방식은 '텍스트'로 통일해 주도록 합니다. 이를 바꾸기 위해서 **유니티 창 상단 메뉴 바에서 'Edit -> Project Settings -> Editor -> Asset Serialization Mode'를 'Force Text'를 선택해 줍니다.**

3) Version Control

- 유니티 엔진에서는 메타파일에 기록된 내용을 바탕으로 Scene에 포함된 오브젝트들을 관리한다. 그런데 'Edit -> Project Settings -> Version Control -> Version Control Mode'가 개발자마다 다를 경우, 개발자들 간에 풀 및 푸시를 할 때 문제가 발생할 수 있습니다. 따라서 별도의 다른 버전 컨트롤 프로그램을 사용하는 것이 아니라면 'Visible Meta Files'로 통일하여 사용하는걸 권장합니다.

출처

- <https://www.youtube.com/watch?v=MeHUEKGZQko&list=PLC2Tit6NyVieQ6vVq9HX9zEJKjPZ8QNcn>
- <https://www.youtube.com/watch?v=wBsSUBEUYYV4>