



# 게임엔진프로그래밍응용

---

## 1. 유니티 엔진

청강문화산업대학교 게임콘텐츠스쿨

반 경 진

# 코로나 유의사항

# 코로나 유의 사항

- 2023학년도 1학기 방역 및 학사운영 방안  
<https://www.ck.ac.kr/archives/193175>
- 2023학년도 1학기 국가공휴일 및 대학 행사 수업 대체 일정 공지  
<https://www.ck.ac.kr/archives/193109>

# 온라인 수업 저작권 유의 사항

# 온라인 수업 저작권 유의 사항

## 온라인수업 저작권 유의사항 안내



**강의 저작물을 다운로드, 캡처하여  
교외로 유출하는 행위는  
불 법 입 니 다**

저작권자의 허락 없이 저작물을 복제, 공중송신 또는 배포하는 것은  
저작권 침해에 해당하며 저작권법에 처벌받을 수 있습니다.

강의 동영상과 자료 일체는 교수 및 학교의 저작물로서 저작권이 보호됩니다.  
수업자료를 무단 복제 또는 배포, 전송 시 민형사상 책임을 질 수 있습니다.

# Index

- 1 유니티의 역사
- 2 유니티 엔진 설치 및 개발환경 구성
- 3 유니티 인터페이스 둘러보기
- 4 유니티 엔진이 동작하는 원리

# 유니티의 역사

# 유니티의 역사

- 2004년 덴마크 코펜하겐 게임회사 설립
- 2005년 “GooBall” 출시
- 2005년 6월 Unity Engine v1.0.0
- 2006년 Apple Design Award 수상
- 2010년 모바일 및 콘솔 플랫폼 지원
- 2016년 12월 Unity 2017 버전 배포
- 2022년 3월 11일 2021.2.15 LTS



# 유니티 엔진의 특징

- 쉽고 빠른 개발
- C#과 자바스크립트 지원
- 3D 맥스와 유사한 인터페이스
- 게임 로직에 집중
- 다양한 IDE 지원
- 에셋 스토어
- 풍부한 개발자 풀

# 유니티 엔진의 장점

- WYSIWYG
- 간편한 빌드
- 낮은 요구 사양
- 에셋 스토어
- 넓은 사용자 층과 그로 인한 풍부한 관련자료
- 저렴한 라이선스 비용

# 유니티 엔진의 단점

- 빈약한 고급 기능
- 제한적인 멀티스레드 기능
- GC 스파이크 문제
- 소스 코드 비공개
- 보안성 문제

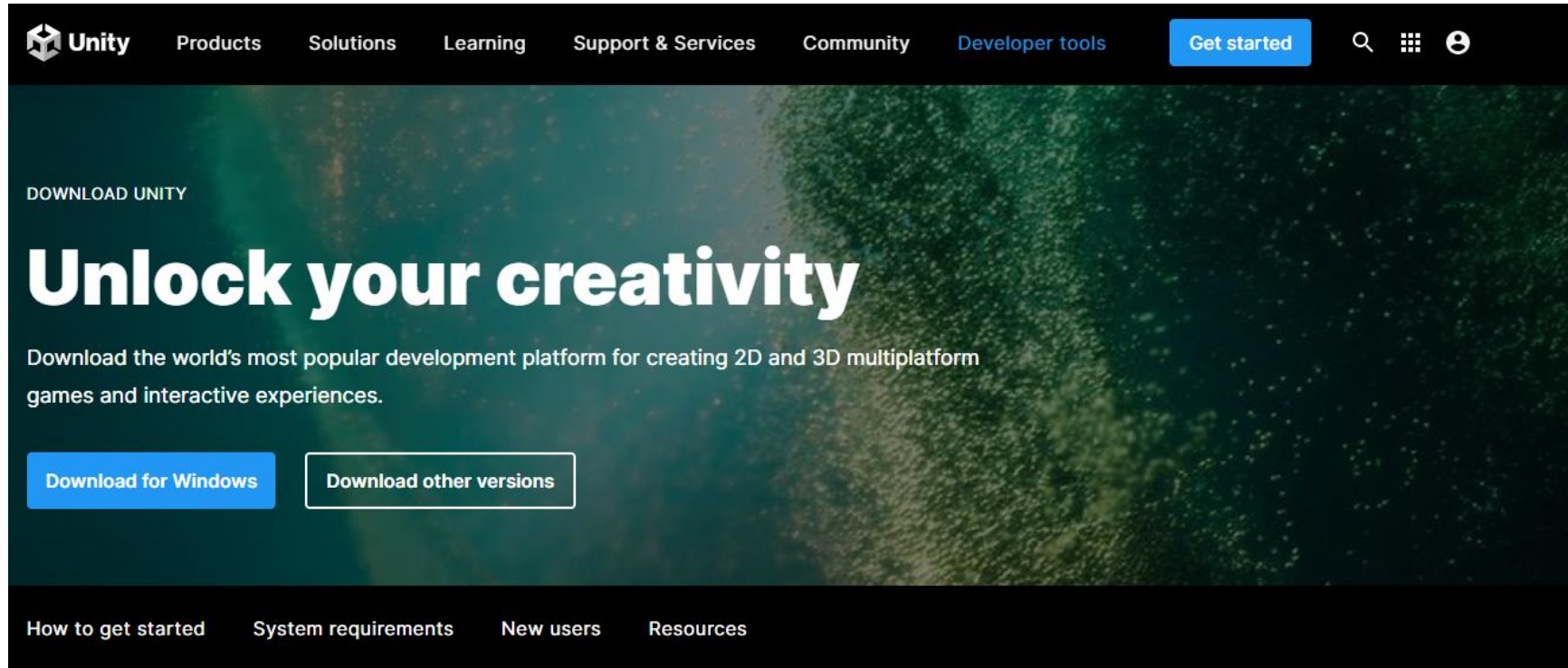
# 유니티 엔진 설치 및 개발환경 구성

# 유니티 엔진 설치

- Unity Hub
- 실습 시 사용할 버전 - Unity 2021.3 LTS

# 유니티 엔진 설치




- <https://unity.com/download>



**Create with Unity in three steps**

# 유니티 엔진 설치

- <https://unity3d.com/get-unity/download/archive>

 [Products](#) [Solutions](#) [Case Studies](#) [Learning](#) [Asset Store](#) [Support & Services](#)  

## Unity download archive



From this page you can download the previous versions of Unity for both Unity Personal and Pro (if you have a Pro license, enter in your key when prompted after installation). Please note that we don't support downgrading a project to an older editor version. However, you can import projects into a new editor version. We advise you to back up your project before converting and check the console log for any errors or warnings after importing.

**Long Term Support releases**

The LTS stream is for users who wish to continue to develop and ship their games/content and stay on a stable version for an extended period.

[Download LTS releases](#)

[Unity 2021.x](#) [Unity 2020.x](#) [Unity 2019.x](#) [Unity 2018.x](#) [Unity 2017.x](#) [Unity 5.x](#) [Unity 4.x](#) [Unity 3.x](#)

Unity 2021.2.15 11 Mar, 2022	 Unity Hub	Downloads (Win) ▼	Downloads (Mac) ▼	Downloads (Linux) ▼	Release notes
Unity 2021.2.14 3 Mar, 2022	 Unity Hub	Downloads (Win) ▼	Downloads (Mac) ▼	Downloads (Linux) ▼	Release notes

# 유니티 유저 메뉴얼

- <https://docs.unity3d.com/Manual/index.html>

Unity Documentation

Manual Scripting API

Search manual...

unity3d.com →

Version: 2021.3 ▾

Language : English ▾

Unity Manual

+

 Unity User Manual 2021.3 (LTS)

•

 New in Unity 2021 LTS

+

 Packages and feature sets

+

 Working in Unity

+

 Asset Workflow

+

 Input

+

 2D

+

 Graphics

+

 World building

+

 Physics

+

 Scripting

+

 Multiplayer and Networking

+

 Audio

+

 Video overview

+

 Animation

+

 Create user interfaces (UI)

+

 Navigation and Pathfinding

+

 Unity Services

+

 XR

+

 Unity's Asset Store

+

 Platform development

+

 Unity Search

+

 Legacy Topics

•

 Glossary

Unity User Manual 2021.3 (LTS)

←

→

Unity User Manual 2021.3 (LTS)

Use the Unity Editor to create 2D and 3D games, apps, and experiences. Download the Editor at [unity3d.com](https://unity3d.com).

The Unity User Manual helps you learn how to use the Unity Editor and its associated services. You can read it from start to finish, or use it as a reference.

If it's your first time using Unity, take a look at the introductory documentation on [Working with Unity](#), and see the [Unity Tutorials](#).

New

Packages

Best Practice guides

• Features introduced in 2021.3: [What's New](#)

• Upgrading Unity projects from older versions of Unity: [Upgrade Guide](#)

• Learn about packages: [Working with the Package Manager](#)

• Find documentation for a specific package: [Verified packages](#) and [Preview packages](#)

• Learn how to build a custom package: [Creating custom packages](#)

• Best practices from Unity Support engineers: [Advance best practice guides](#)

Unity User Manual sections



Working in Unity

A complete introduction to the Unity Editor.



Unity 2D

All of the Unity Editor's 2D-specific features including gameplay, sprites and physics.



Graphics

The visual aspects of the Unity Editor including cameras and lighting.



Physics

Simulation of 3D motion, mass, gravity and collisions.



Networking

How to implement Multiplayer and networking.



Scripting

Programming your games by using scripting in the Unity Editor.



Audio



Animation



UI



Navigation



Unity services

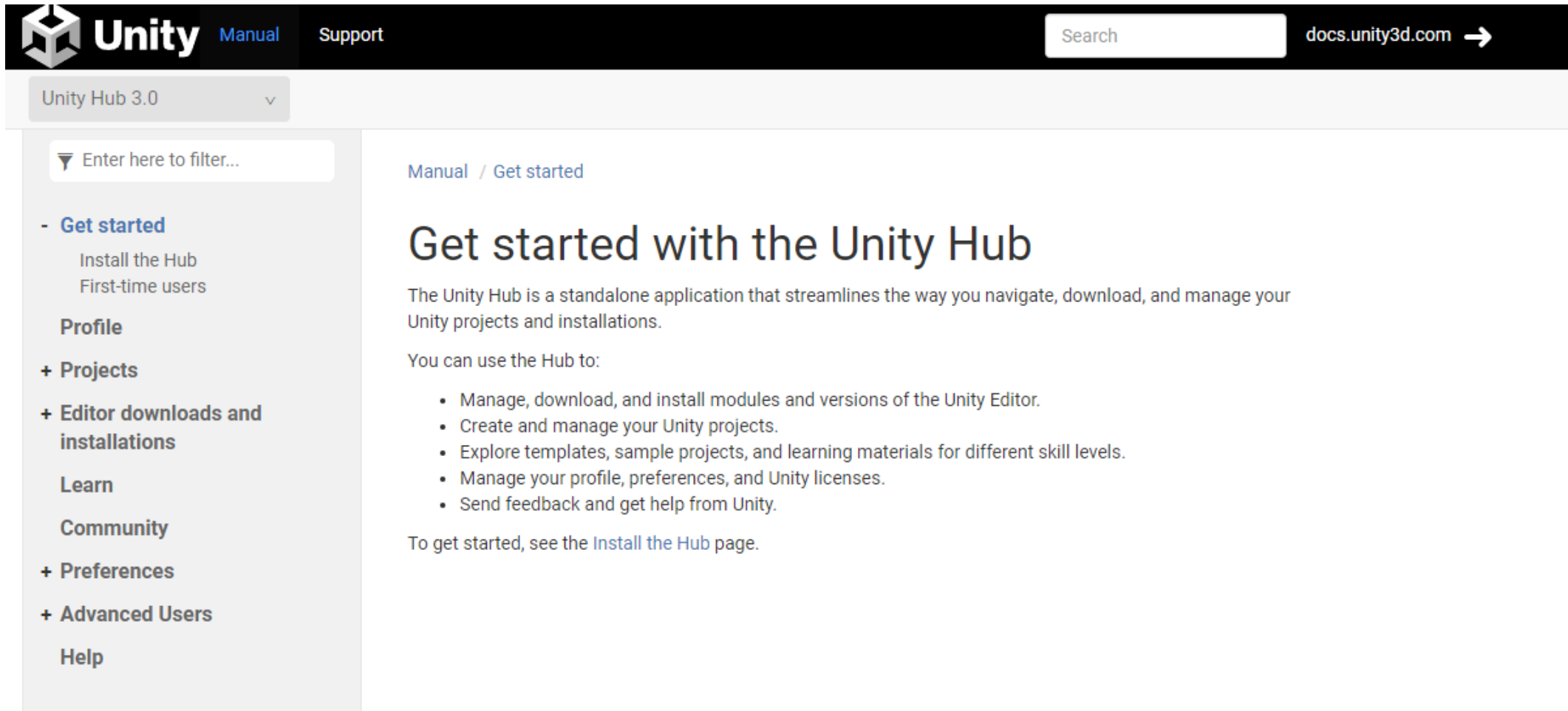


Virtual reality



# 유니티 허브

- <https://docs.unity3d.com/hub/manual/index.html>



The screenshot shows the Unity Hub manual page. The top navigation bar is black with the Unity logo, 'Manual' link, 'Support' link, a search bar, and the URL 'docs.unity3d.com' with a right arrow. Below this is a light gray bar with 'Unity Hub 3.0' and a dropdown arrow. The left sidebar is light gray and contains a search bar with the text 'Enter here to filter...'. Below the search bar are several menu items: '- Get started' (with sub-items 'Install the Hub' and 'First-time users'), 'Profile', '+ Projects', '+ Editor downloads and installations', 'Learn', 'Community', '+ Preferences', '+ Advanced Users', and 'Help'. The main content area is white and has a breadcrumb 'Manual / Get started'. The title 'Get started with the Unity Hub' is in a large, bold font. Below the title is a paragraph: 'The Unity Hub is a standalone application that streamlines the way you navigate, download, and manage your Unity projects and installations.' This is followed by a section 'You can use the Hub to:' with a bulleted list: 'Manage, download, and install modules and versions of the Unity Editor.', 'Create and manage your Unity projects.', 'Explore templates, sample projects, and learning materials for different skill levels.', 'Manage your profile, preferences, and Unity licenses.', and 'Send feedback and get help from Unity.' At the bottom of the main content area, it says 'To get started, see the [Install the Hub](#) page.'

Unity Manual Support Search docs.unity3d.com →

Unity Hub 3.0

Enter here to filter...

- Get started  
Install the Hub  
First-time users

Profile

+ Projects

+ Editor downloads and installations

Learn

Community

+ Preferences

+ Advanced Users

Help

Manual / Get started

## Get started with the Unity Hub

The Unity Hub is a standalone application that streamlines the way you navigate, download, and manage your Unity projects and installations.

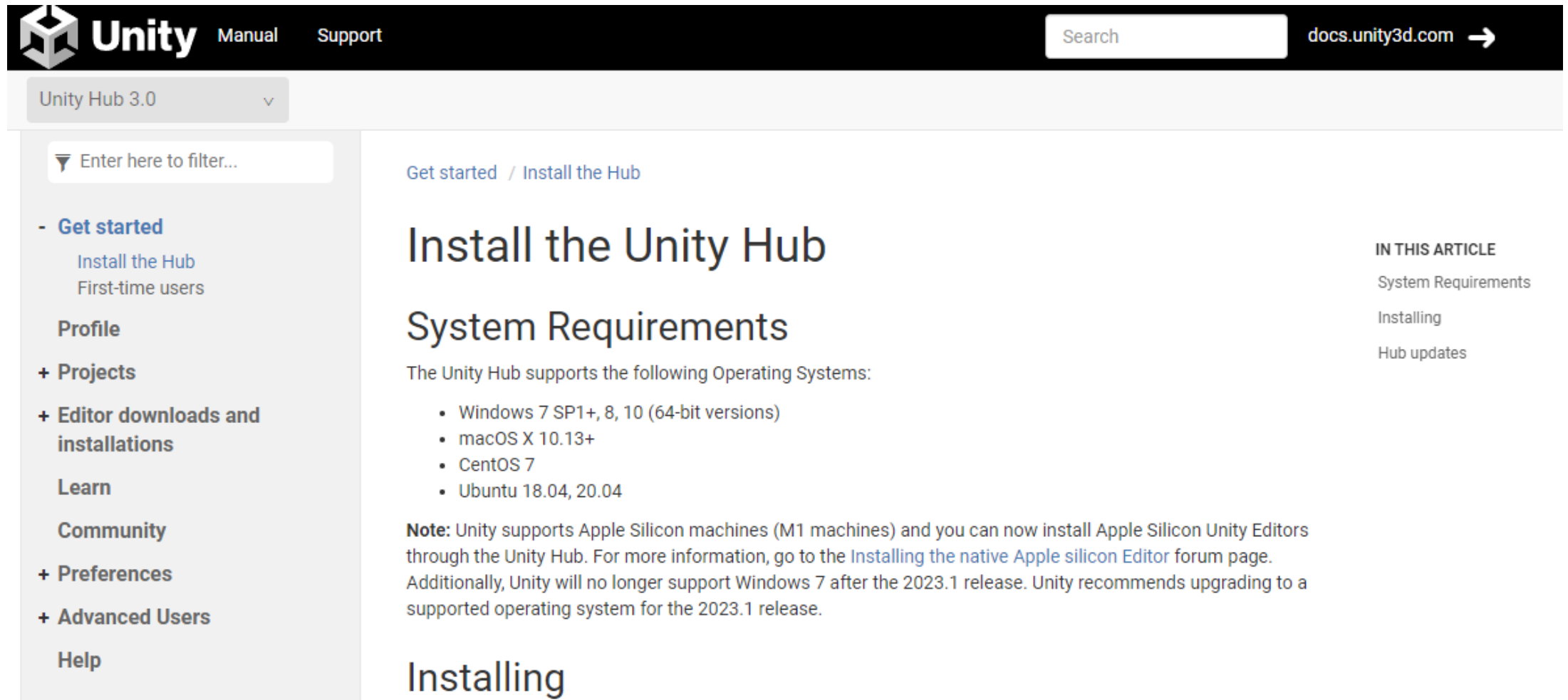
You can use the Hub to:

- Manage, download, and install modules and versions of the Unity Editor.
- Create and manage your Unity projects.
- Explore templates, sample projects, and learning materials for different skill levels.
- Manage your profile, preferences, and Unity licenses.
- Send feedback and get help from Unity.

To get started, see the [Install the Hub](#) page.

# 유니티 허브

- <https://docs.unity3d.com/hub/manual/index.html>



The screenshot shows the Unity Hub 3.0 manual page. The top navigation bar includes the Unity logo, 'Manual', 'Support', a search bar, and the URL 'docs.unity3d.com' with a right arrow. Below the navigation bar, a dropdown menu shows 'Unity Hub 3.0'. The left sidebar contains a search bar and a list of navigation items: 'Get started' (with sub-items 'Install the Hub' and 'First-time users'), 'Profile', '+ Projects', '+ Editor downloads and installations', 'Learn', 'Community', '+ Preferences', '+ Advanced Users', and 'Help'. The main content area has a breadcrumb 'Get started / Install the Hub' and a title 'Install the Unity Hub'. Below the title is the section 'System Requirements' with the text 'The Unity Hub supports the following Operating Systems:' followed by a bulleted list: 'Windows 7 SP1+, 8, 10 (64-bit versions)', 'macOS X 10.13+', 'CentOS 7', and 'Ubuntu 18.04, 20.04'. A 'Note' section follows, stating that Unity supports Apple Silicon machines (M1 machines) and that Windows 7 support will end after the 2023.1 release. The right sidebar, titled 'IN THIS ARTICLE', lists 'System Requirements', 'Installing', and 'Hub updates'. The bottom of the page shows the start of the 'Installing' section.

Unity Manual Support Search docs.unity3d.com →

Unity Hub 3.0

Enter here to filter...

- Get started
  - Install the Hub
  - First-time users
- Profile
- + Projects
- + Editor downloads and installations
- Learn
- Community
- + Preferences
- + Advanced Users
- Help

Get started / Install the Hub

## Install the Unity Hub

### System Requirements

The Unity Hub supports the following Operating Systems:

- Windows 7 SP1+, 8, 10 (64-bit versions)
- macOS X 10.13+
- CentOS 7
- Ubuntu 18.04, 20.04

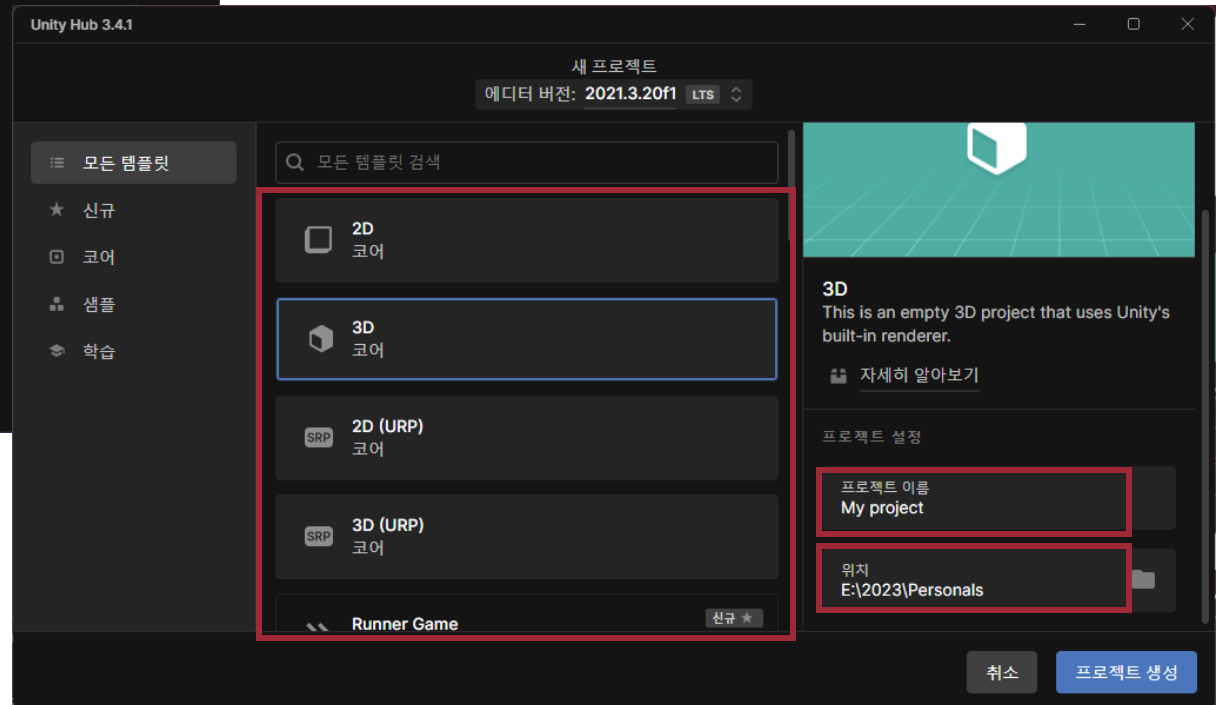
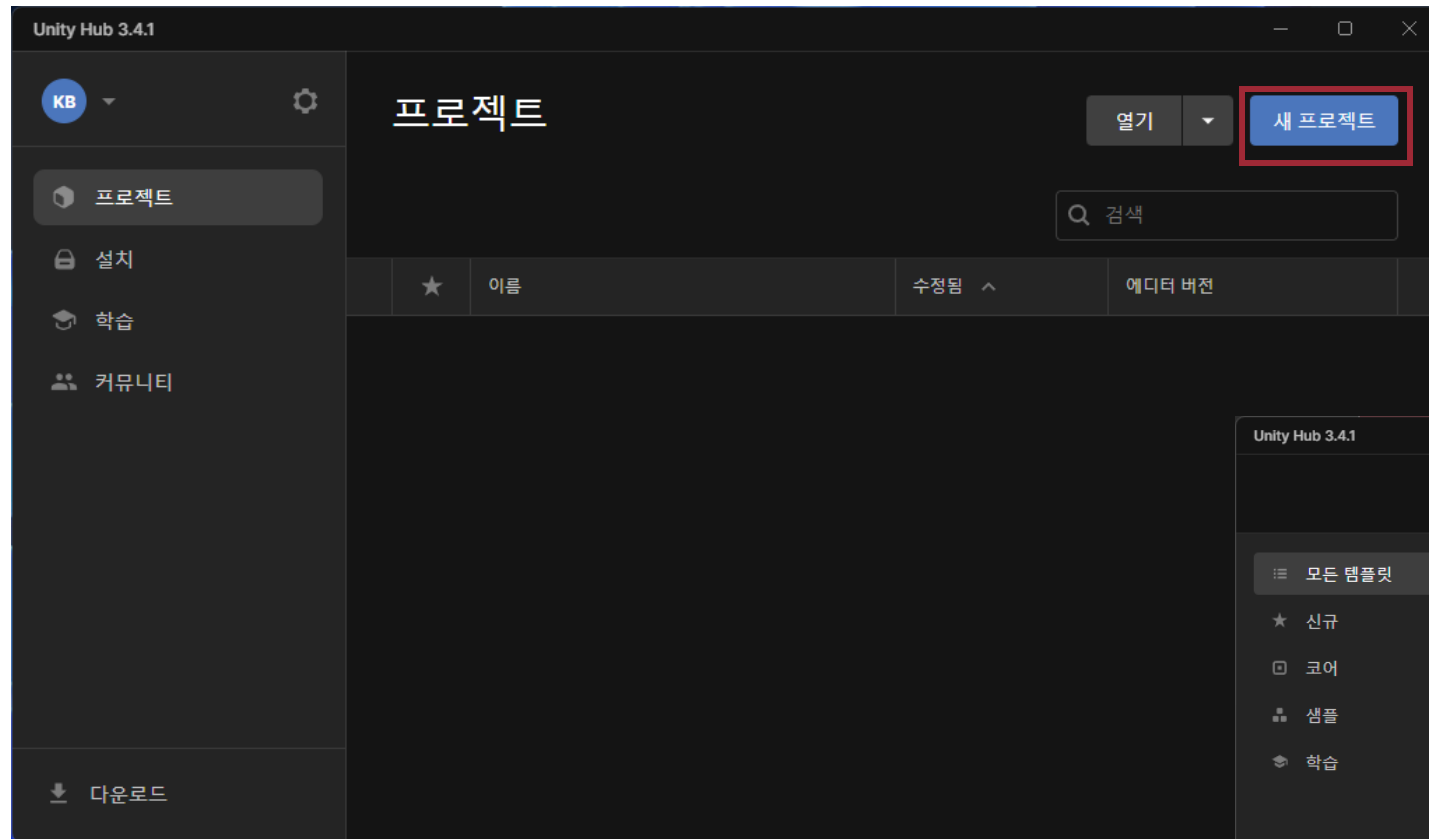
**Note:** Unity supports Apple Silicon machines (M1 machines) and you can now install Apple Silicon Unity Editors through the Unity Hub. For more information, go to the [Installing the native Apple silicon Editor](#) forum page. Additionally, Unity will no longer support Windows 7 after the 2023.1 release. Unity recommends upgrading to a supported operating system for the 2023.1 release.

### Installing

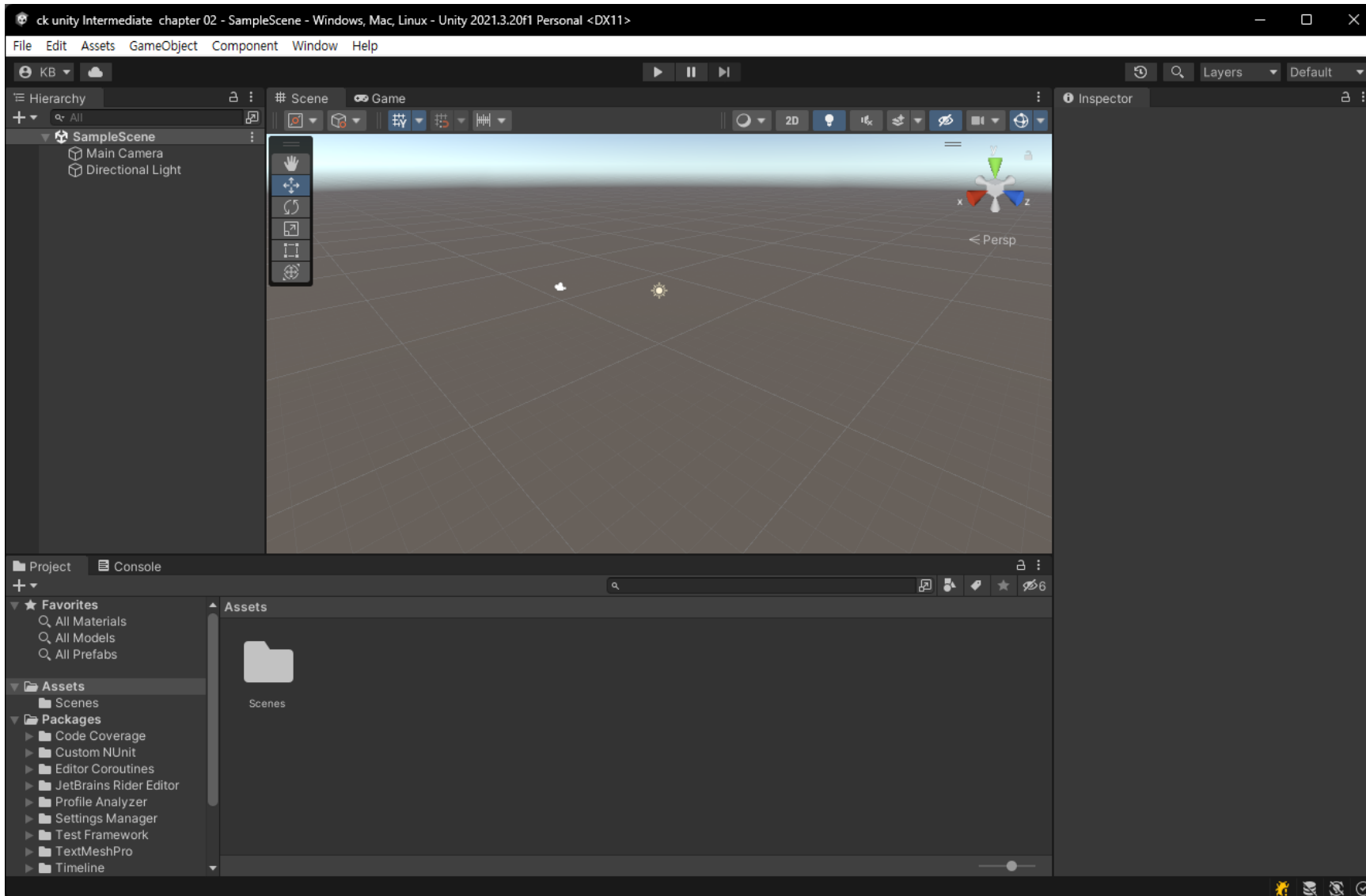
IN THIS ARTICLE

- System Requirements
- Installing
- Hub updates

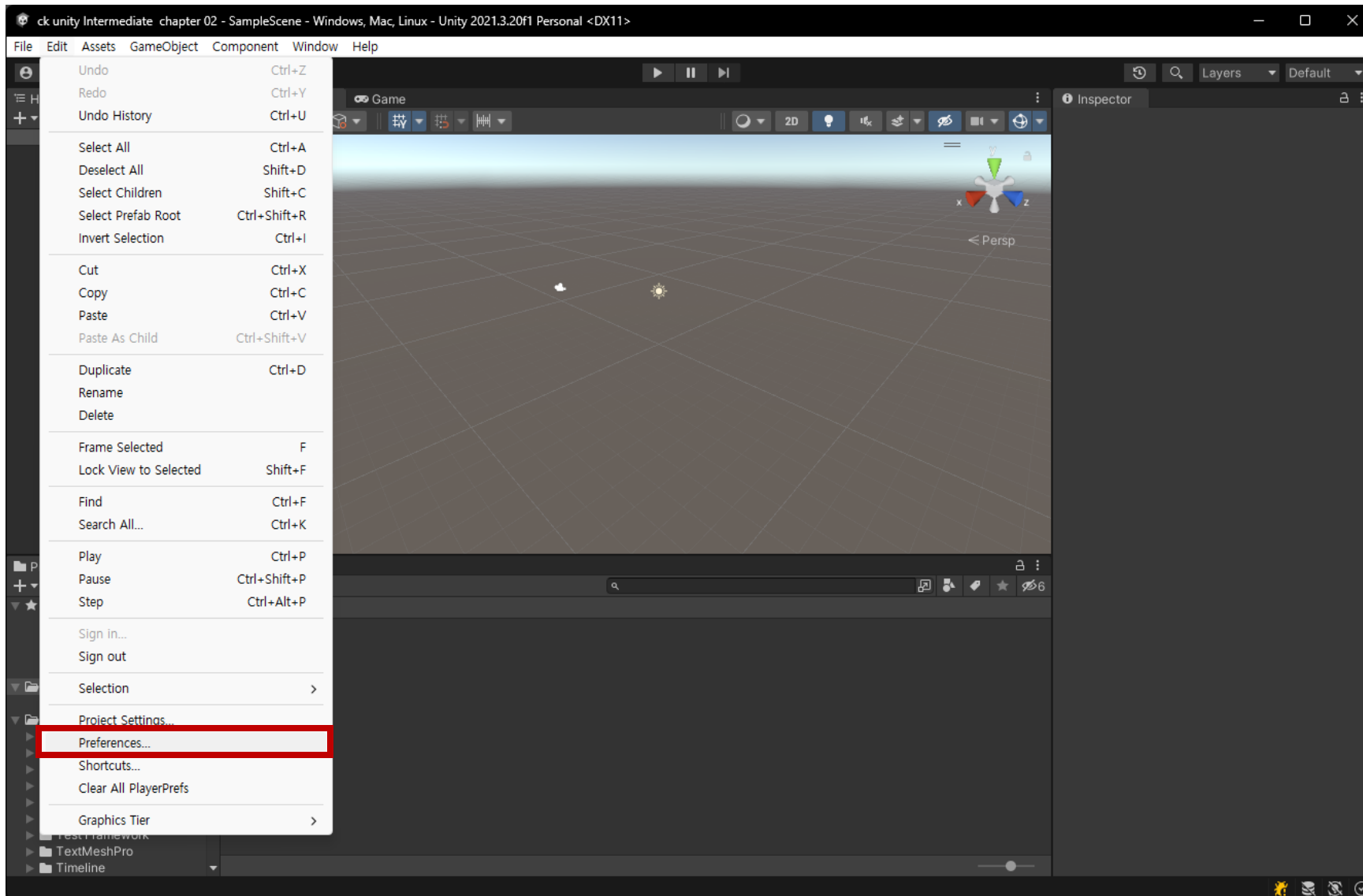
# 프로젝트 생성



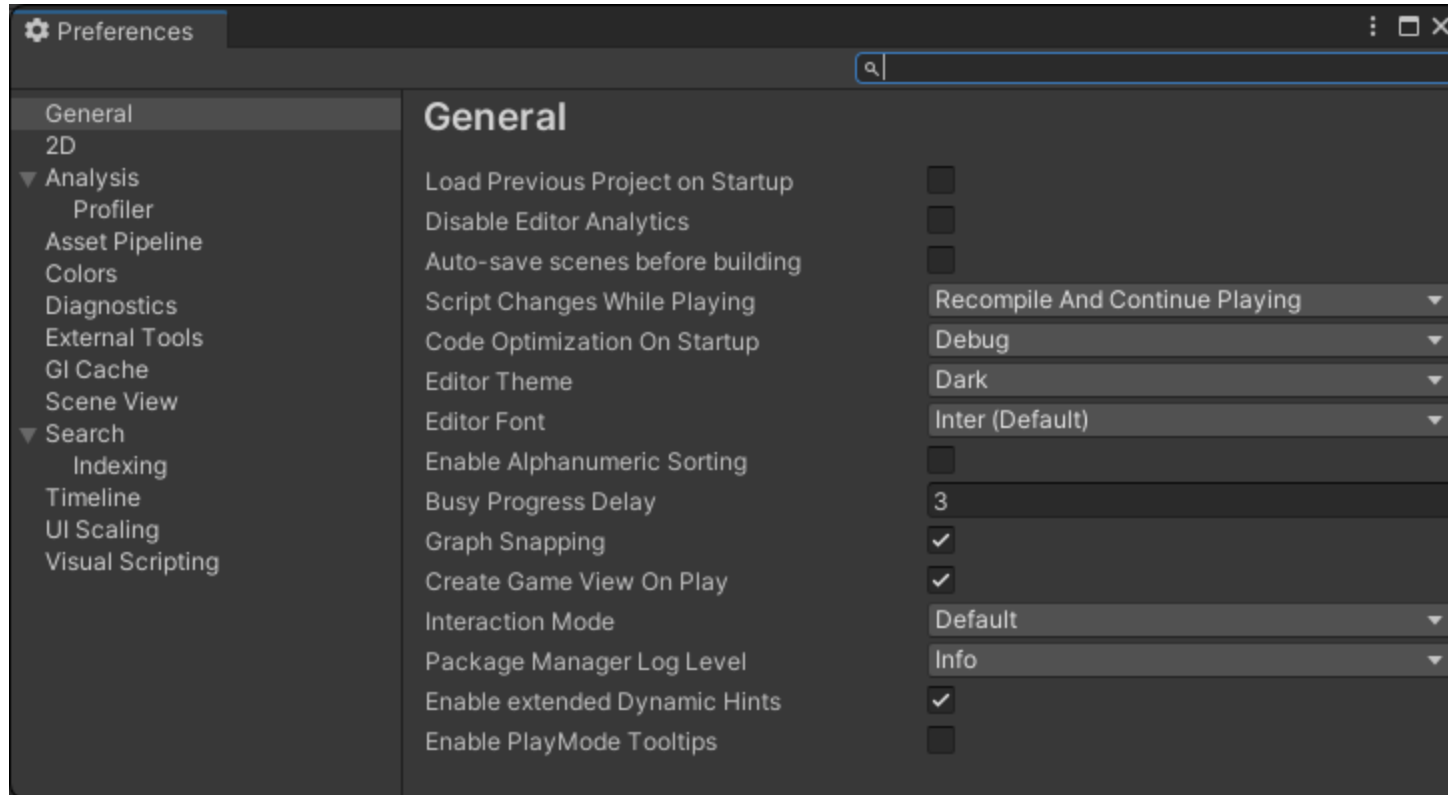
# 프로젝트 생성



# 개발 환경 구성



# 개발 환경 구성



## ■ 에디터 설정

General

2D

Analysis

Asset Pipeline

Color

Diagnostics

External Tools

GI Cache

Scene View

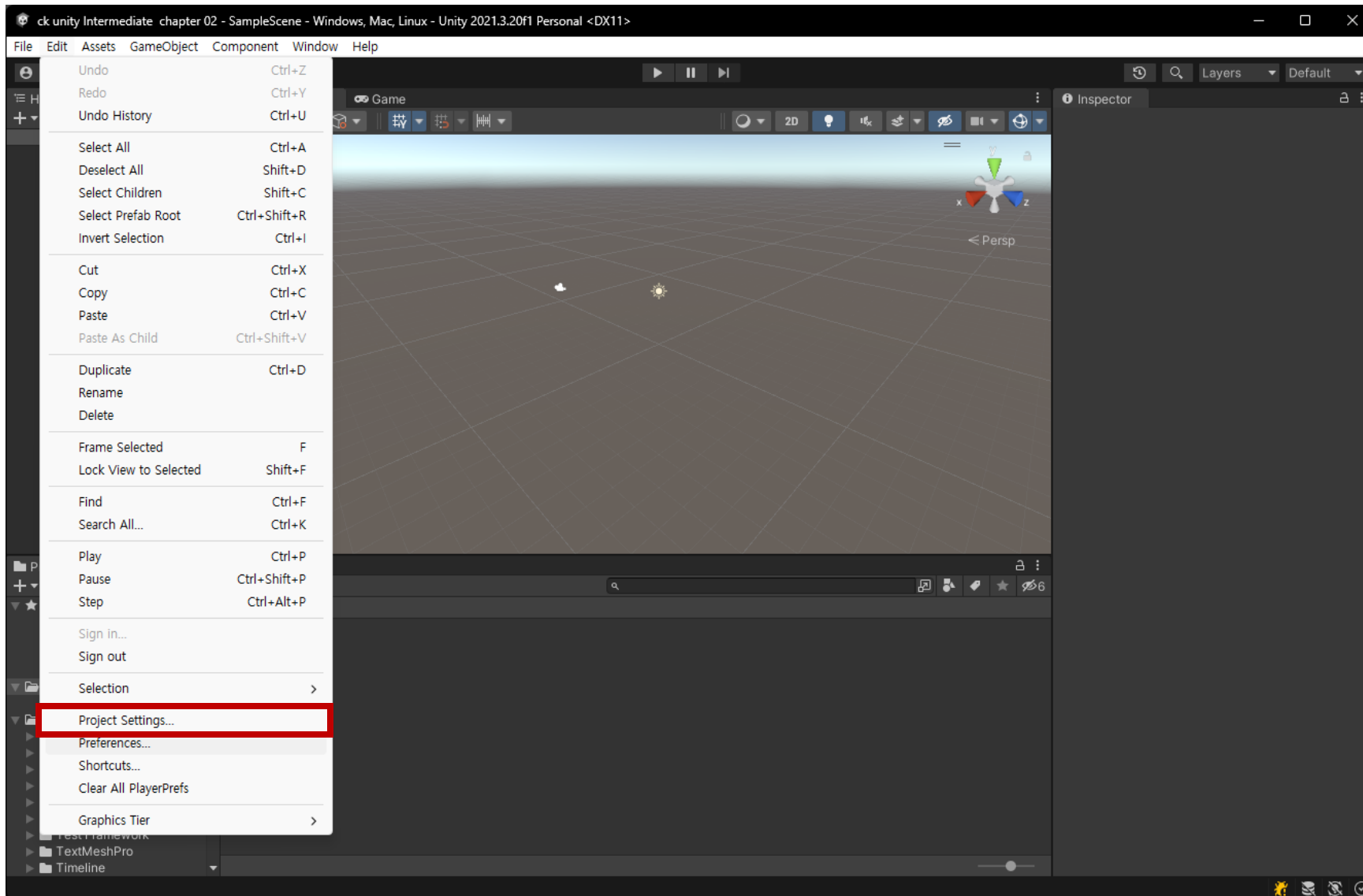
Search

Timeline

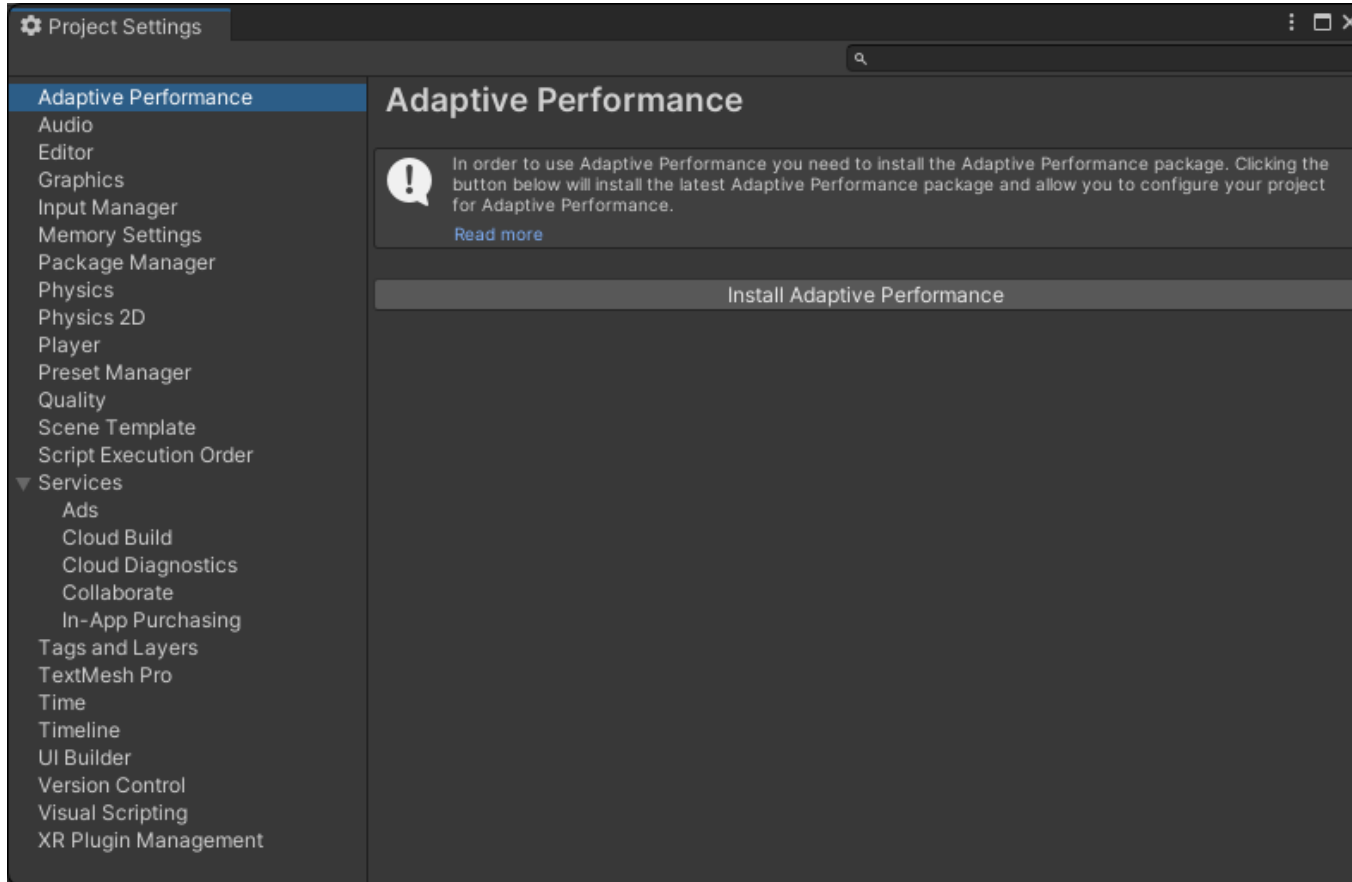
UI Scaling

Visual Scripting

# 개발 환경 구성



# 개발 환경 구성



## ■ 프로젝트 설정

Adaptive Performance

Audio

Editor

Graphics

Input Manager

Memory Settings

Package Manager

Physics

Physics 2D

Player

Preset Manager

Quality

Scene Template

Script Execution Order

Services

Tags and Layers

TextMesh Pro

Time

TimeLine

UI Builder

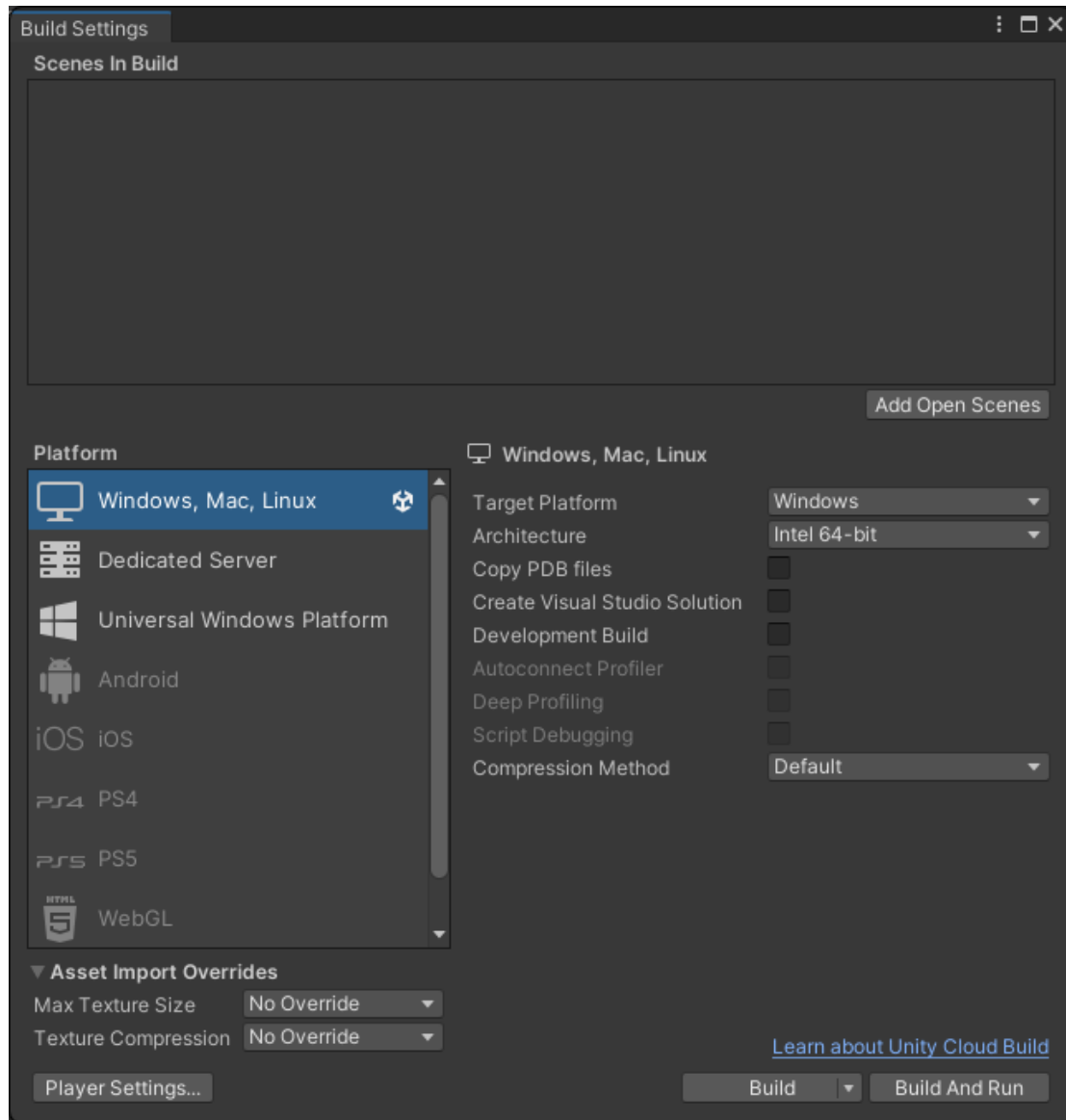
Version Control

Visual Scripting

XR Plugin Management



# 개발 환경 구성



- 빌드 설정

Scenes In Build

Platform

# 유니티 인터페이스 둘러보기

# 유니티 인터페이스



- A. The Toolbar
- B. The Hierarchy Window
- C. The Game View
- D. The Scene View
- E. Overlays
- F. The Inspector Window
- G. The Project Window
- H. The Status Bar

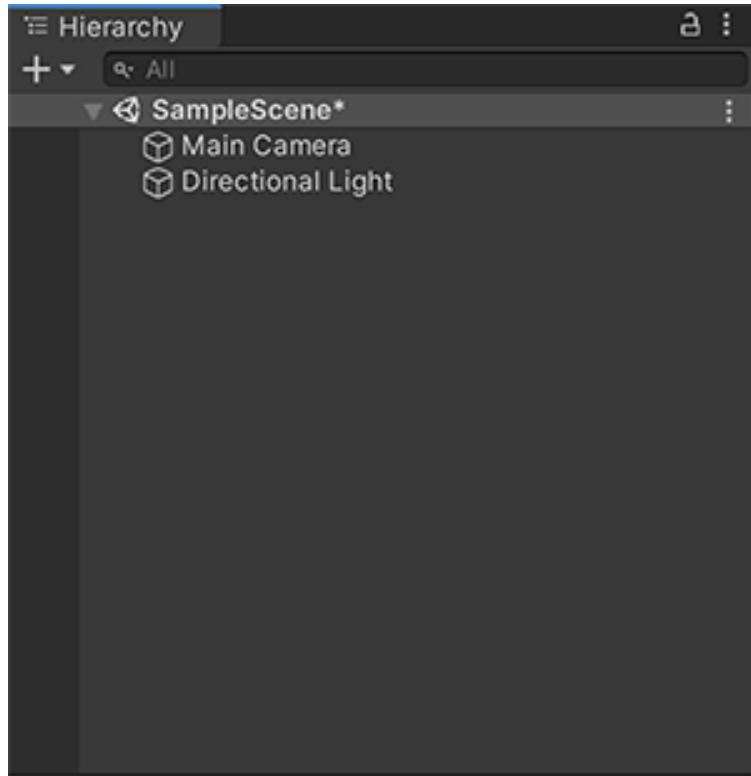
# The ToolBar



툴바는 Unity 에디터의 상단에 위치  
 툴바는 창이 아니며, Unity 인터페이스에서 재배열할 수 없는 유일한 부분

컨트롤:	설명:
	<b>Account</b> 드롭다운 메뉴에서 Unity 계정에 액세스합니다.
	Cloud 버튼을 클릭하여 <u>Unity Services</u> 창을 엽니다.
	<b>Collab</b> 드롭다운 메뉴에서 <u>Unity 콜라보레이트</u> 를 실행합니다.
	<u>게임 뷰</u> 에서 Play, Pause 및 Step 버튼을 사용합니다.
	에디터에서 수행한 작업은 <u>Undo History</u> 에서 확인하거나, 취소하거나 다시 실행할 수 있습니다.
	검색을 엽니다.
	<u>Layers</u> 드롭다운 메뉴에서 씬 뷰에 표시할 오브젝트를 관리합니다.
	<u>Layout</u> 드롭다운 메뉴에서 뷰 정렬을 변경하고 새 레이아웃을 저장하거나 기존 레이아웃을 로드할 수 있습니다.

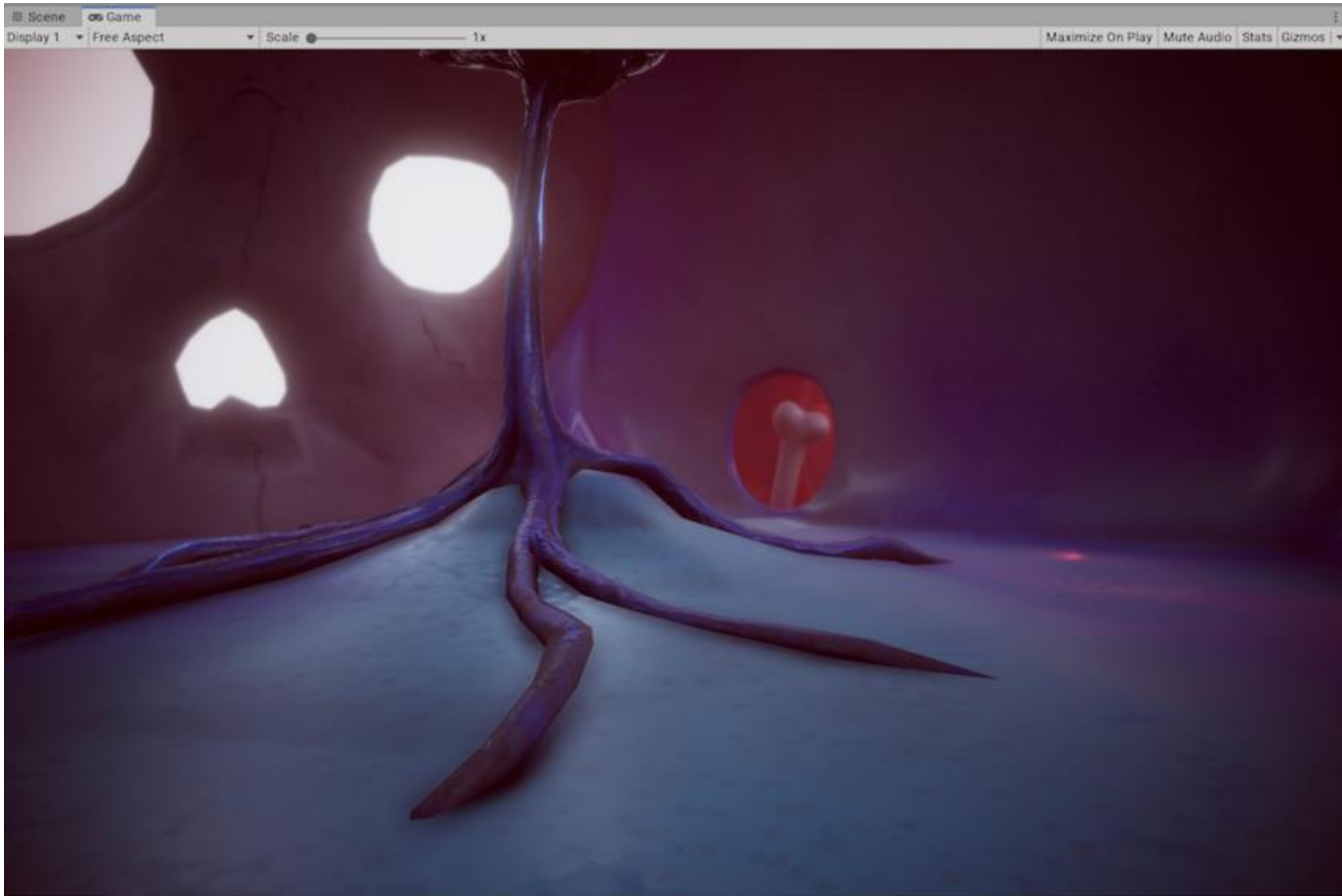
# The Hierarchy Window



- 부모-자식
- 게임 오브젝트 구성
  - 새 게임 오브젝트 생성
  - 자식 게임 오브젝트 가시성 토글
  - 자식 게임 오브젝트 생성
  - 부모 게임 오브젝트 생성
  - 형제 게임 오브젝트 생성
  - 게임 오브젝트 복제
  - 기본 부모로 게임 오브젝트 구성
- 씬의 가시성 및 선택 가능성 설정
- 게임 오브젝트 정렬
- 여러 씬 편집
- 오버라이드 표시기

계층 창은 모델, 카메라, 프리팹 등과 같은 모든 게임 오브젝트를 씬에 표시.  
계층 창을 사용하여 씬에 사용할 게임 오브젝트를 정렬하고 그룹화.  
씬 뷰에 게임 오브젝트를 추가하거나 제거하면 계층 창에서도 추가되거나 제거.  
또한 계층 창은 자체 게임 오브젝트가 들어 있는 다른 씬을 포함할 수 있음

# The Game View



게임 뷰는 애플리케이션의 카메라에서 렌더링되며, 최종적으로 퍼블리시된 애플리케이션을 보여줌

애플리케이션을 사용할 때 플레이어가 보는 것을 제어하려면 하나 이상의 카메라를 사용

카메라에 대한 자세한 내용은 카메라 컴포넌트 페이지를 참조

# The Game View

## 플레이 모드(Play mode)



툴바의 버튼을 사용하여 에디터 플레이 모드를 제어

플레이 모드에서는 모든 변경 사항이 임시적으로 적용되며 플레이 모드를 종료하면 바로 초기화.

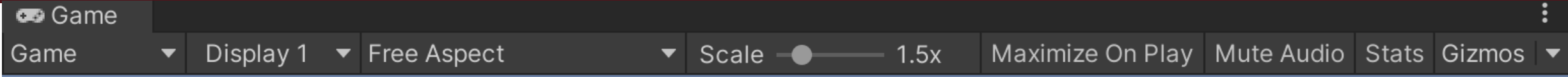
이러한 점을 쉽게 인식할 수 있도록 에디터의 UI가 어둡게 변합니다.

### 게임 뷰 사용

**\*\* Game\*\*** 탭에서 시뮬레이터 뷰가 열려 있는 경우 왼쪽 상단의 드롭다운 메뉴를 사용하여 Game view를 클릭합니다.

메뉴에서 Window > General > Game으로 이동

# The Game View



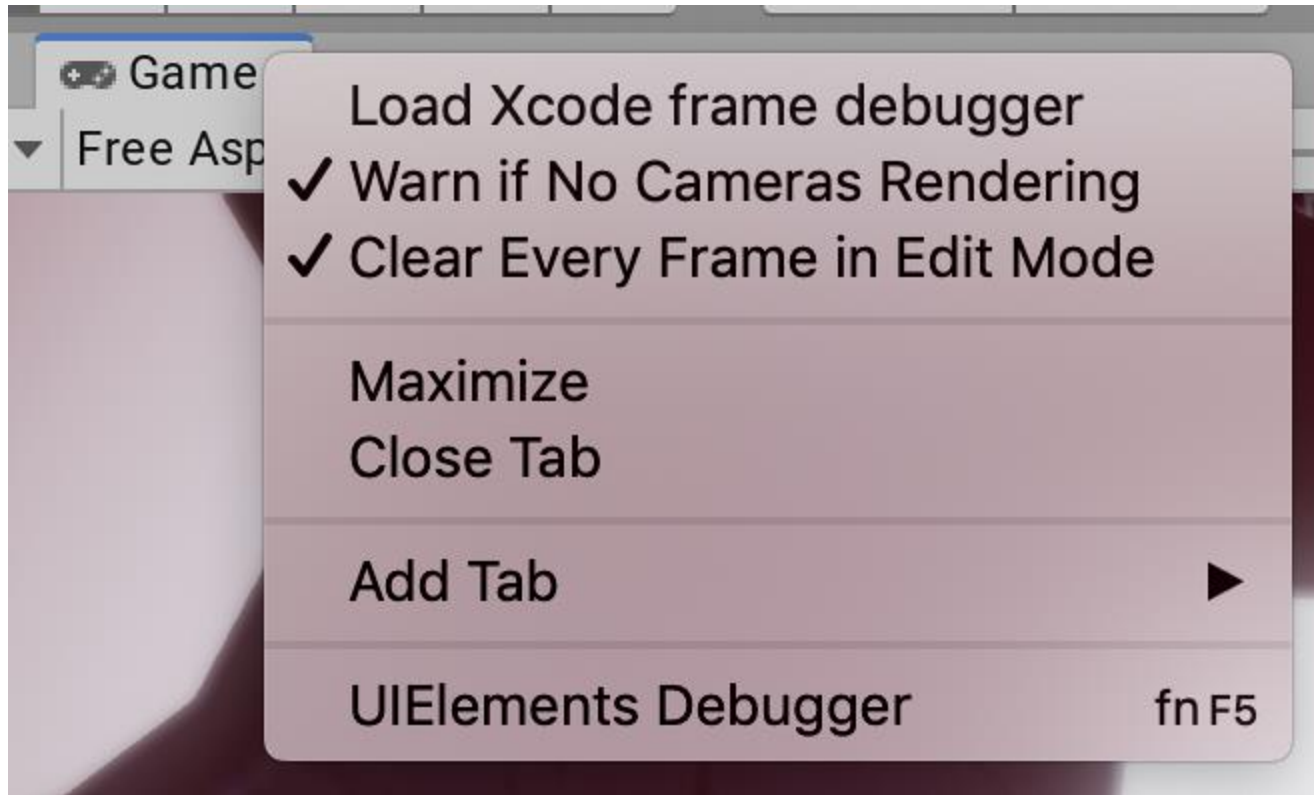
버튼	기능
Game/Simulator	클릭하면 드롭다운 메뉴에서 <b>게임 뷰</b> 또는 <b>시뮬레이터 뷰</b> 를 활성화합니다.
Display	씬에 카메라가 여러 대 있는 경우 카메라 리스트에서 이 버튼을 선택하여 원하는 카메라를 선택할 수 있습니다. 기본적으로 <b>Display 1</b> 로 설정되어 있습니다. <b>Target Display</b> 드롭다운 메뉴에서 카메라 모듈에 있는 카메라에 디스플레이를 할당할 수 있습니다.
Aspect	이 값을 선택하여 다른 종횡비를 가진 모니터에서 게임이 어떻게 보일지 테스트할 수 있습니다. 기본적으로 <b>Free Aspect</b> 로 설정되어 있습니다.
Low Resolution Aspect Ratios	구형 디스플레이의 픽셀 밀도를 에뮬레이트하려면 이 <b>Low Resolution Aspect Ratios</b> 를 활성화합니다. 이렇게 하면 종횡비를 선택했을 때 게임 뷰의 해상도가 줄어듭니다. 이 옵션은 레티나 디스플레이가 탑재되지 않은 기기에서 게임 뷰를 실행할 때 항상 활성화됩니다.
VSync (Game view only)	게임 뷰에 우선 순위를 부여하려면 <b>VSync (Game view only)</b> 를 활성화합니다. 이 옵션은 비디오를 녹화할 때 유용한 수직 동기화를 추가할 수 있습니다. Unity는 모니터 새롭고침 속도로 게임 뷰를 렌더링하려고 시도합니다(보장되지는 않음). 이 옵션이 활성화된 경우에도 다른 뷰를 숨기고 Unity가 렌더링하는 뷰 수를 줄이기 위해 플레이 모드에서 게임 뷰를 최대화하는 것이 좋습니다.
Scale 슬라이더	오른쪽으로 스크롤하여 게임 화면을 확대하고 화면 영역을 자세히 확인합니다. 이 슬라이더를 이용하면 게임 뷰 창의 크기에 비해 기기의 해상도가 높은 경우 전체 화면을 축소하여 볼 수 있습니다. 애플리케이션이 중지되었거나 일시 정지되었을 때에는 스크롤 휠이나 마우스 가운데 버튼을 사용하여 이 작업을 수행할 수도 있습니다.
Maximize on Play	클릭하면 활성화됩니다. 플레이 모드를 시작할 때 전체 화면 미리보기를 위해 게임 뷰를 최대화(에디터 창의 100%)하려면 이 옵션을 사용합니다.
Mute Audio	클릭하면 활성화됩니다. 플레이 모드를 시작할 때 애플리케이션 내 오디오를 음소거하려면 이 옵션을 사용합니다.
Stats	애플리케이션의 오디오 및 그래픽스에 대한 <b>렌더링 통계</b> 를 확인할 수 있는 통계 오버레이를 표시하려면 이 옵션을 클릭합니다. 이 옵션은 플레이 모드에서 애플리케이션의 성능을 모니터링하는 데 아주 유용합니다.
Gizmos	<b>기즈모</b> 를 표시하려면 이 옵션을 클릭합니다. 플레이 모드에서 특정 타입의 기즈모만 보려면 <b>Gizmos</b> 옆에 있는 드롭다운 화살표를 클릭하여 보려는 기즈모 타입을 활성화합니다.



# The Game View

## 고급 옵션

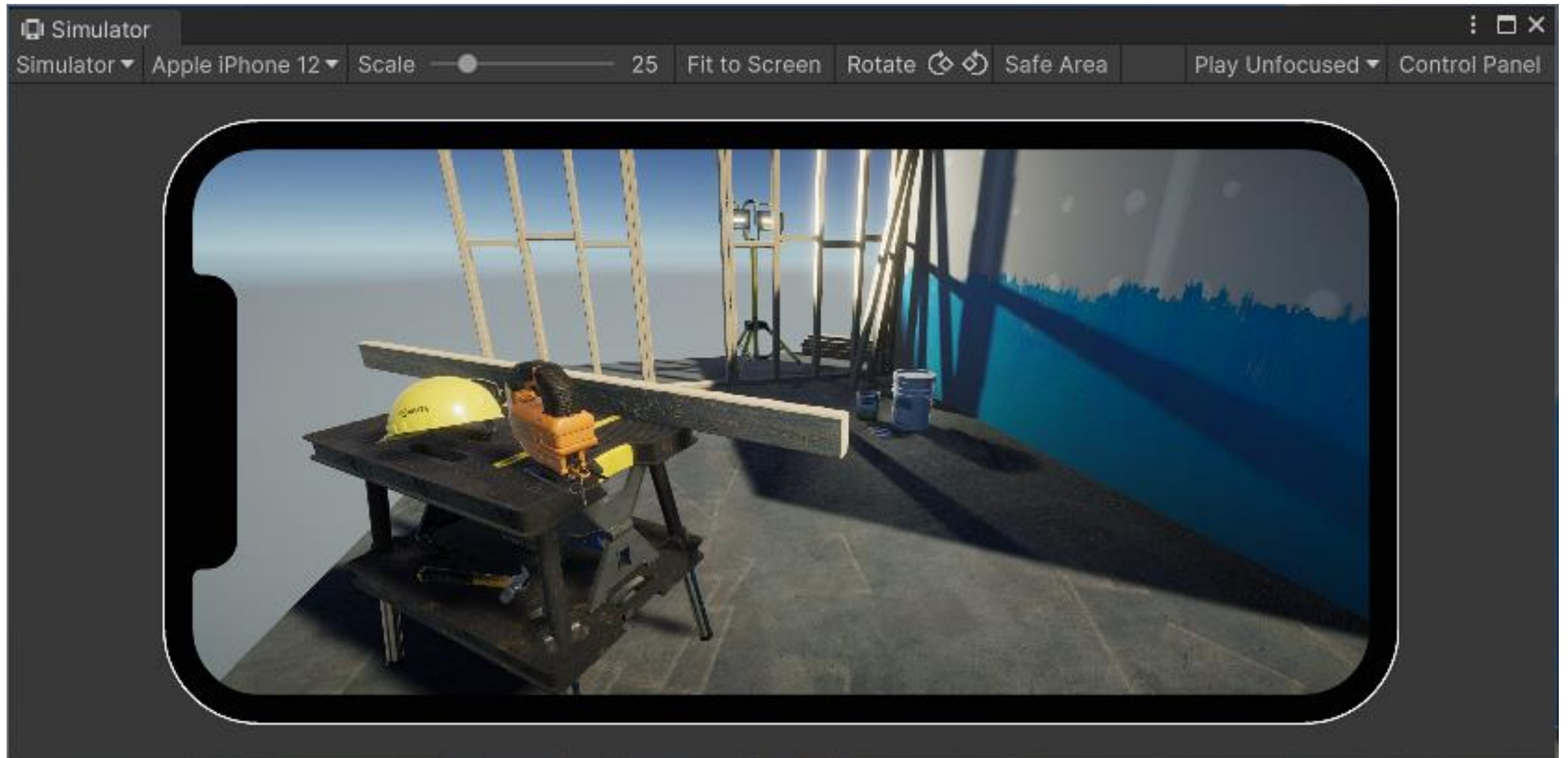
\*\* 게임 \*\* 탭에서 마우스 오른쪽 버튼을 클릭하여 고급 게임 뷰 옵션을 표시할 수 있습니다.



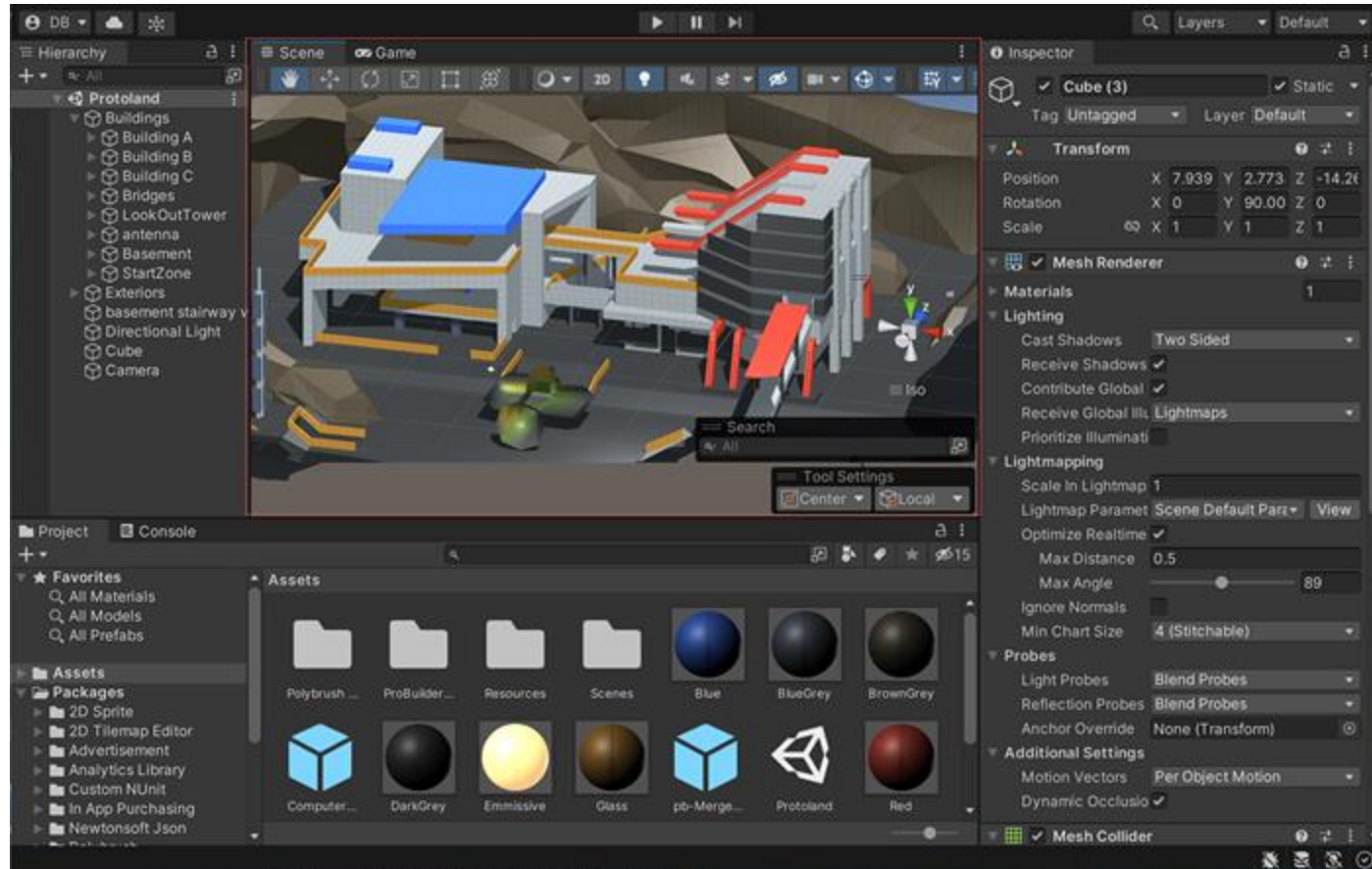
**Warn if No Cameras Rendering:** 이 옵션은 기본적으로 활성화되어 있습니다. 화면에 렌더링하는 카메라가 없을 경우 Unity가 경고 메시지를 표시합니다. 실수로 카메라를 삭제하거나 비활성화하는 등의 문제를 진단할 때 유용합니다. 애플리케이션 렌더링에 의도적으로 카메라를 사용하지 않는 경우가 아니라면 이 기능을 활성화 상태로 유지합니다.

**Clear Every Frame in Edit Mode:** 이 옵션은 기본적으로 활성화되어 있습니다. 애플리케이션이 실행되지 않을 때 Unity가 프레임마다 게임 뷰를 지웁니다. 이렇게 하면 애플리케이션을 설정할 때 얼룩 효과를 방지할 수 있습니다. 플레이 모드가 아닌 경우 이전 프레임의 콘텐츠에 의존하지 않는 한 이 설정을 활성화 상태로 유지합니다.

# The Simulator view



# The Scene View



## 씬 뷰

씬 뷰 는 제작 중인 월드를 표시하는 인터랙티브한 뷰이며, 씬, 캐릭터, 카메라, 광원, 게임 오브젝트의 모든 타입을 선택하고 배치하는데 사용됨.

# The Scene View

- 씬 뷰 내비게이션
- 게임 오브젝트 선택
- 오버레이
- 게임 오브젝트 위치 지정
- 커스텀 에디터 툴
- 게임 오브젝트 숨김 및 표시
- 씬 뷰 컨트롤 바
- 씬 뷰 카메라
- 기즈모 메뉴

# The Scene View

## 씬 뷰 내비게이션

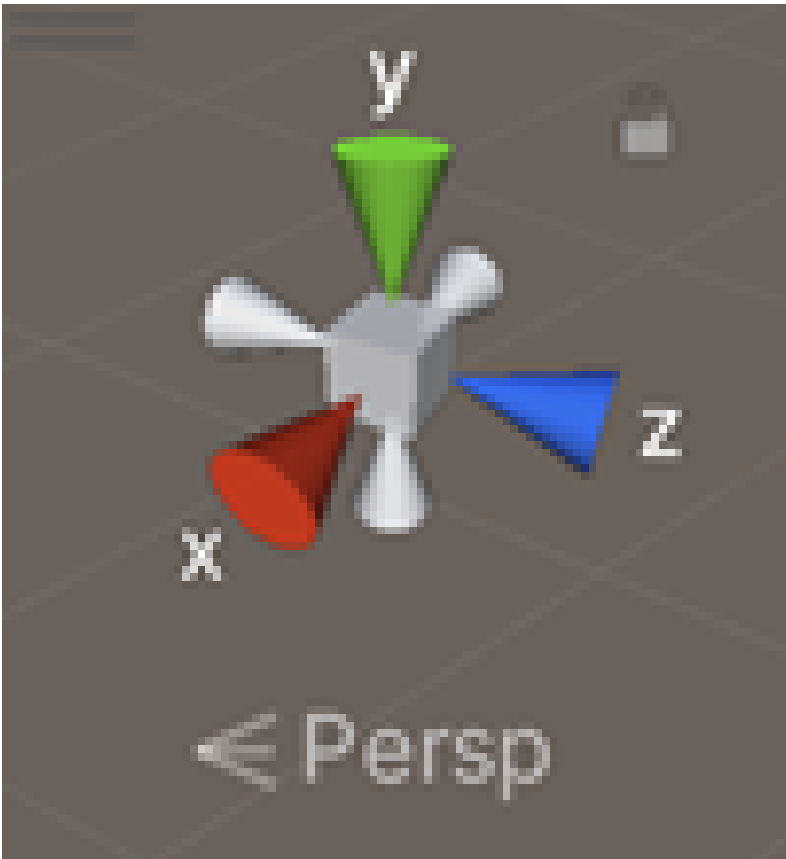
씬 뷰에는 효율적으로 이리저리 이동하는 데 유용한 내비게이션 컨트롤이 있습니다.

- 씬 기즈모
- 이동, 궤도 선회 및 확대/축소 툴
- 중심 맞추기 툴

# The Scene View

## 씬 기즈모

씬 기즈모는 씬 뷰에 나타나며, 씬 뷰 카메라의 현재 방향을 표시합니다. 이 기즈모로 시야각과 투사 모드를 변경

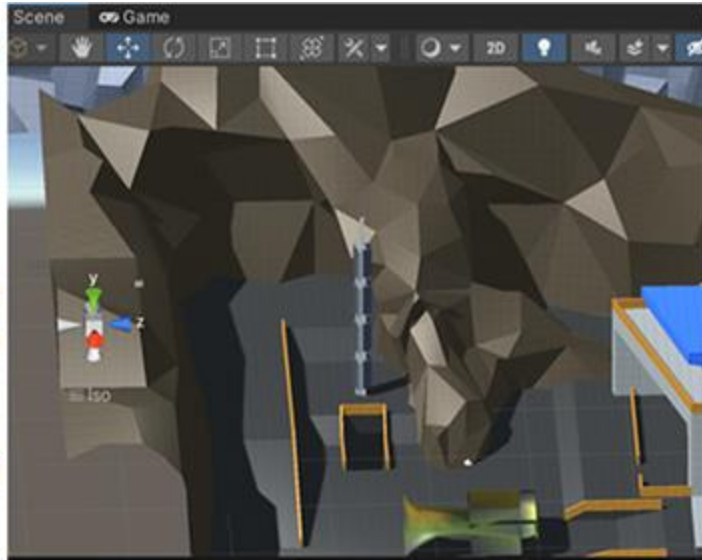


씬 기즈모에는 원뿔형 암이 큐브의 각 면에 있습니다. 맨 앞에 있는 암의 레이블은 X, Y, Z 입니다. 원뿔형 축 암을 클릭하여 씬 뷰 카메라를 암이 나타내는 축에 맞춥니다(예: 평면도, 좌면도, 정면도). 큐브를 마우스 오른쪽 버튼으로 클릭하여 시야각 리스트가 있는 메뉴도 표시할 수 있습니다. 기본 시야각으로 복원하려면 씬 기즈모를 마우스 오른쪽 버튼으로 클릭한 후 Free 를 클릭하십시오.

Perspective 도 켜다 켤 수 있습니다. 그러면 씬 뷰의 투사 모드가 Perspective 와 Orthographic(때로는 “아이소메트릭”이라고 함) 간에 전환됩니다. 이렇게 하려면 씬 기즈모의 중앙에 있는 큐브나 아래에 있는 텍스트를 클릭합니다. 직교 뷰는 원근이 없으며, 원뿔형 축 암 중 하나를 클릭할 때 함께 사용하여 전면도, 평면도 또는 측면도를 얻는 데 유용합니다

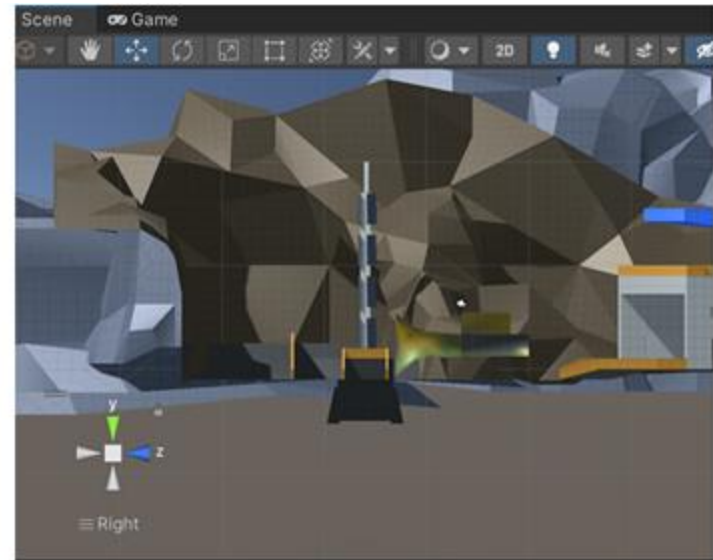
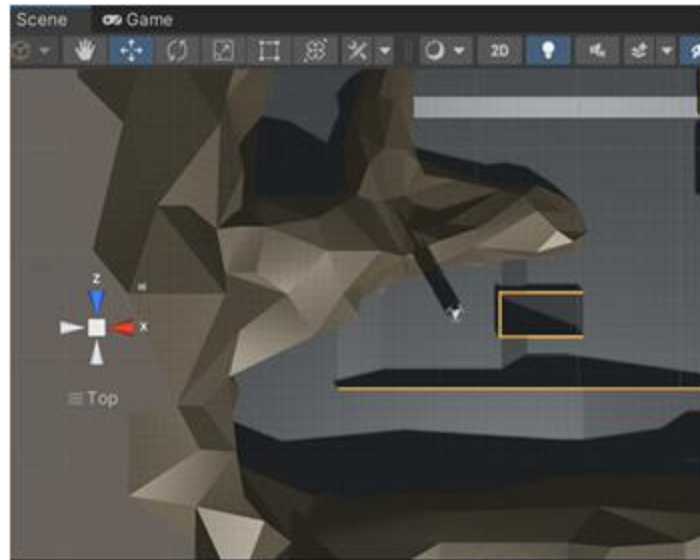


# The Scene View



원근 모드(왼쪽)와 직교 모드(오른쪽)로 표시한 씬

직교 모드에서 평면도 및 전면도로 표시한 동일 씬



# The Scene View

씬 뷰의 관점이 이상한 경우(아래 위 반전 또는 보기에 혼란스러운 각도), Shift 키를 누른 상태로 씬 기즈모 중앙에 있는 큐브를 클릭하여 씬을 측면과 약간 위쪽에서 보는 각도의 Perspective 뷰로 복원할 수 있습니다.

씬 기즈모의 오른쪽 상단에 있는 자물쇠를 클릭하여 씬 회전을 활성화하거나 비활성화합니다. 씬 회전을 비활성화한 후 마우스 오른쪽 버튼을 클릭하면 뷰가 회전되지 않고 패닝됩니다. 이 동작은 뷰 툴과 동일합니다.

2D Mode에서는 씬 기즈모가 나타나지 않습니다. XY 평면을 수직으로 바라보는 뷰가 유일한 옵션이기 때문입니다.

## Mac 트랙패드 제스처

트랙패드가 있는 Mac에서는 두 손가락으로 드래그하여 뷰를 확대/축소할 수 있습니다.

세 손가락을 사용하여 Scene Gizmo의 암을 클릭하는 효과도 시뮬레이션할 수 있습니다. 상하좌우로 드래그하여 씬 뷰 카메라를 해당 방향에 맞추면 됩니다.



# The Scene View

## 씬 뷰 이동, 궤도 선회, 확대/축소

이동, 궤도 선회, 확대/축소는 씬 뷰 내비게이션의 핵심 동작입니다. Unity는 이러한 동작을 수행하는 여러 가지 방법을 제공하여 접근성을 극대화합니다.

- 화살표 이동
- 뷰 톨
- 플라이스루 모드
- 카메라 속도
- 동작 단축키



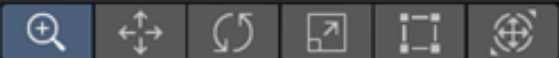
## 화살표 이동

Arrow Keys 를 사용하여 씬에서 걸어다니듯이 이리저리 움직일 수 있습니다. 위아래 화살표를 누르면 카메라가 향한 방향에서 앞과 뒤로 이동합니다. 왼쪽 및 오른쪽 화살표를 누르면 뷰가 옆으로 이동합니다. Shift 를 화살표와 함께 누르면 더 빨리 이동합니다.

# The Scene View

## 뷰 툴

뷰 툴을 선택하면(단축키: Q) 아래와 같은 마우스 컨트롤을 사용할 수 있습니다.

컨트롤:	설명:
Move	 클릭하고 드래그하여 카메라를 이동합니다.
Orbit	 <b>Alt</b> (Windows) 또는 <b>Option</b> (macOS)를 누른 상태에서 마우스 왼쪽 버튼을 클릭하고 드래그하면 카메라가 현재 피벗 포인트 주위의 궤도를 따라 회전합니다.  이 옵션은 뷰가 직교이므로 2D 모드에서는 사용할 수 없습니다.
Zoom	 <b>Alt</b> (Windows) 또는 <b>Option</b> (macOS) 키를 누른 상태로 마우스 오른쪽 버튼을 클릭하고 드래그하여 씬 뷰를 확대/축소합니다.  macOS에서는 <b>Control</b> 키를 누른 상태에서 마우스 왼쪽 버튼을 클릭하고 드래그할 수도 있습니다.

Shift를 계속 누르면 이동 속도와 확대/축소 속도가 증가합니다.

# The Scene View

## 플라이스루 모드

Flythrough Mode 를 사용하여 씬 뷰를 여러 게임에서 탐색하는 방법과 유사하게 1인칭 시점으로 날아다니면서 탐색할 수 있습니다.

오른쪽 마우스 버튼을 클릭한 상태로 계속 누릅니다.

마우스를 사용하여 뷰를 이리저리 움직이고 WASD 키를 사용하여 좌우와 앞뒤로 이동하고 Q 및 E 키를 사용하여 상하로 이동합니다.

Shift 를 계속 누르고 있으면 더 빠르게 이동합니다.

플라이스루 모드는 Perspective Mode 에서 사용합니다. Orthographic Mode 에서는 오른쪽 마우스 버튼을 누른 상태에서 마우스를 움직이면 카메라가 궤도를 선회합니다.

2D 모드에서는 Flythrough 모드를 사용할 수 없습니다. 대신 마우스 오른쪽 버튼을 누른 상태에서 마우스를 움직이면 씬 뷰가 이리저리 이동합니다.

## 카메라 속도

씬 뷰 내 카메라의 현재 속도를 변경하려면 툴바의 카메라 아이콘을 클릭하십시오.

Flythrough mode 에서는 씬 뷰에서 이동하는 동안 카메라 속도를 변경할 수 있습니다. 이렇게 하려면 마우스 스크롤 휠을 사용하거나 트랙패드에서 손가락 두 개로 드래그하십시오.

# The Scene View

## 동작 단축키

이러한 컨트롤을 선택된 변환 툴에 관계 없이 사용할 수 있어 더 효율적입니다. 사용하는 마우스 또는 트랙패드에 따라 컨트롤의 편의성이 달라집니다.

액션	3버튼 마우스	2버튼 마우스 또는 트랙패드	하나의 마우스 버튼 또는 트랙패드만 있는 Mac
Move	<b>Alt</b> 키를 누른 상태로 마우스 가운데 버튼을 클릭한 후 드래그	<b>Alt+Control</b> 키를 누른 상태로 마우스 왼쪽 버튼을 클릭한 후 드래그	<b>Alt+Command</b> 키를 누른 상태로 마우스 왼쪽 버튼을 클릭한 후 드래그
Orbit (2D 모드에서 사용 불가)	<b>Alt</b> 키를 누른 상태로 마우스 왼쪽 버튼을 클릭한 후 드래그	<b>Alt</b> 키를 누른 상태로 마우스 왼쪽 버튼을 클릭한 후 드래그	<b>Alt</b> 키를 누른 상태로 마우스 왼쪽 버튼을 클릭한 후 드래그
Zoom	스크롤 휠을 사용하거나, <b>Alt</b> 키를 누른 상태로 마우스 오른쪽 버튼을 클릭한 후 드래그	<b>Alt</b> 키를 누른 상태로 마우스 오른쪽 버튼을 클릭한 후 드래그	두 손가락 스와이프 방식을 사용하여 스크롤 인/아웃하거나 <b>Alt+Control</b> 키를 누른 상태로 마우스 왼쪽 버튼을 클릭한 후 드래그
**Change speed ** (플라이스루 모드에서만 제공됨)	이동하는 동안 스크롤 휠을 사용합니다.	이동하는 동안 손가락 두 개로 드래그합니다.	이동하는 동안 손가락 두 개로 드래그합니다.

# The Scene View

## 게임 오브젝트에 뷰의 중심 맞추기

게임 오브젝트에 씬 뷰의 중심을 맞추려면 계층 구조에서 게임 오브젝트를 선택한 후 씬 뷰 위로 마우스를 옮기고 F 키를 누르십시오. 게임 오브젝트가 이미 선택되어 있을 때 F 키를 누르면 피벗 포인트를 확대합니다. 이 기능은 메뉴 바의 Edit > Frame Selected 에서도 찾을 수 있습니다.

게임 오브젝트가 움직일 때 뷰를 게임 오브젝트에 고정하려면 Shift+F 키를 누르십시오. 이 기능은 메뉴 바의 Edit > Lock View to Selected 에서도 찾을 수 있습니다.

# The Scene View

## 게임 오브젝트 선택

씬 뷰 또는 계층 창에서 단일 게임 오브젝트를 선택할 수 있습니다. 또한 한 번에 여러 개의 게임 오브젝트를 선택할 수도 있습니다.

Unity는 씬 뷰에서 선택된 게임 오브젝트와 해당 게임 오브젝트의 자식을 강조 표시합니다. 선택 항목의 기본 윤곽선 컬러는 주황색이며, 자식의 윤곽선 컬러는 파란색입니다. 선택한 게임 오브젝트의 와이어프레임을 다른 컬러로 강조 표시하도록 선택할 수도 있습니다. 이러한 윤곽선 강조 표시 컬러는 Unity 환경 설정(macOS의 경우: Unity > Preferences, Windows의 경우: Edit > Preference)에서 변경할 수 있습니다.

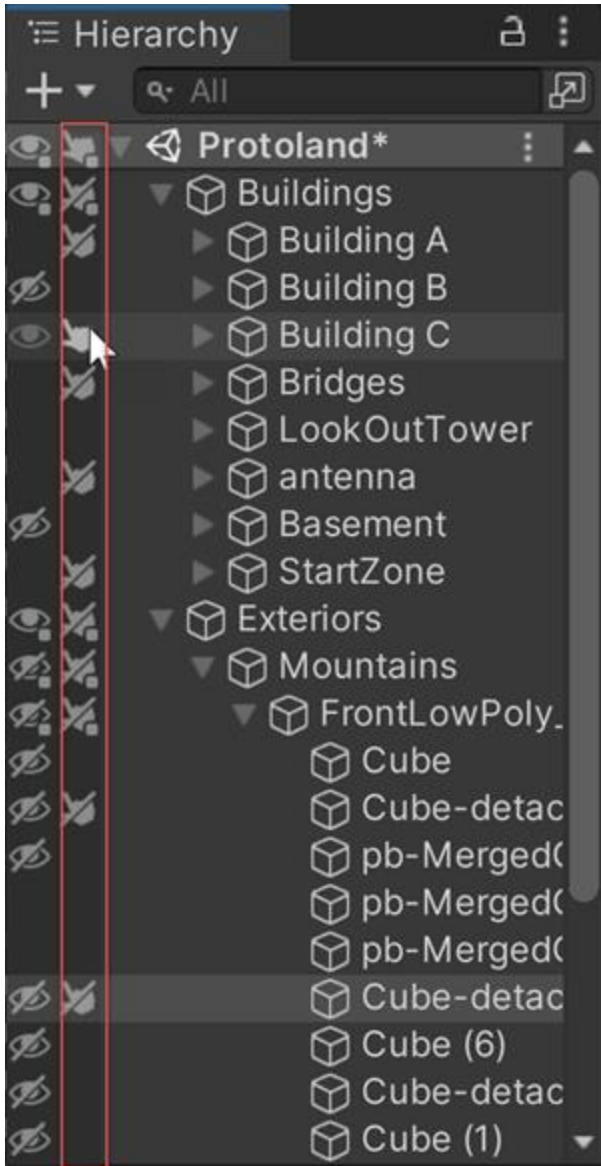
## 단일 게임 오브젝트를 선택

씬 뷰에서 클릭합니다. 오버랩된 게임 오브젝트 간의 공통 공간을 반복적으로 클릭하면 선택 항목이 순환됩니다. 계층 창에서 이름을 클릭합니다.

## 여러 개의 게임 오브젝트를 선택 또는 선택 해제

여러 개의 게임 오브젝트 주위를 드래그하여 사각형을 그립니다. 그러면 해당 경계 상자 안의 모든 항목이 선택됩니다. Shift 키를 누른 상태로 씬의 게임 오브젝트를 클릭합니다. Ctrl(Windows) 또는 Command(macOS) 키를 사용하여 선택 항목에서 게임 오브젝트를 추가하거나 제거할 수도 있습니다.

# The Scene View

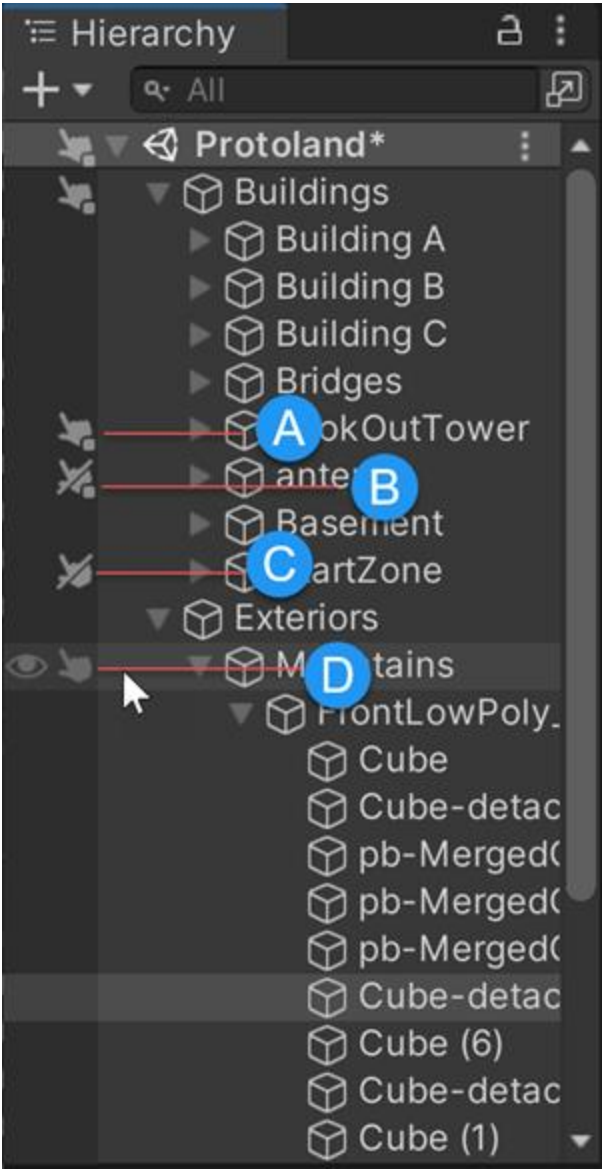






## 씬 선택 컨트롤

씬 선택 컨트롤을 토글하여 에디터 작업 시 선택 가능한 항목을 표시할 수 있습니다. 기본적으로 모든 항목을 선택할 수 있지만, 클릭 시 Unity가 선택 항목에 추가하지 않는 씬 항목을 선택할 수 있습니다. 예를 들어 10,000개 이상의 오브젝트가 있는 대형 씬에서 작업하는 경우 실수로 편집하는 것을 방지하기 위해 특정 게임 오브젝트가 선택되지 않도록 일시적으로 차단할 수 있습니다.

항목을 선택 불가로 표시하면 Unity는 씬 뷰에서 해당 항목을 선택하거나 편집할 수 없어도 해당 항목을 렌더링합니다. 선택 가능성 상태는 에디터에서만, 그리고 설정된 프로젝트의 사용자에게 대해서만 유지

# The Scene View



A		게임 오브젝트를 선택할 수 있지만, 해당 자식 중 일부를 선택할 수 없습니다.
B		게임 오브젝트를 선택할 수 없지만, 해당 자식 중 일부를 선택할 수 있습니다.
C		게임 오브젝트와 해당 자식을 둘 다 선택할 수 없습니다.
D		게임 오브젝트와 해당 자식을 선택할 수 있습니다. 이 아이콘은 게임 오브젝트에 커서를 올릴 때에만 나타납니다.



# The Scene View

## 오버레이

Unity 저작 툴은 씬 뷰 창에서 오버레이 패널로 사용하여 접근 가능성을 높이고 워크플로를 향상할 수 있습니다. 씬 뷰 창에서 표시되는 오버레이의 종류와 위치를 커스터마이징할 수 있으며 커스터마이징된 오버레이 설정을 프리셋으로 저장하여 다시 사용하거나 공유할 수 있습니다. 또한 커스텀 오버레이를 생성할 수 있습니다.

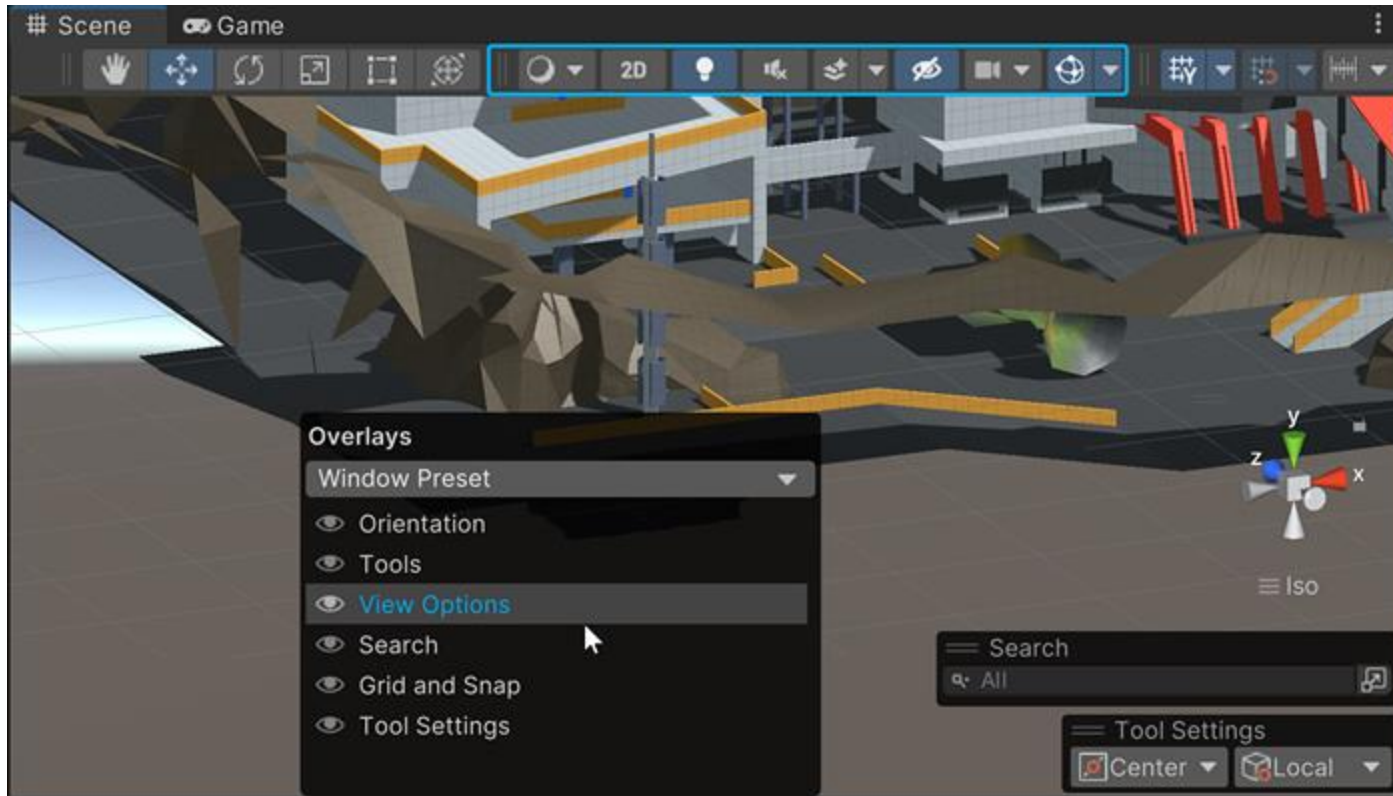


A	툴
B	뷰 옵션
C	<u>그리드와 스냅 툴바</u>
D	방향
E	검색
F	툴 설정

# The Scene View

## 오버레이 표시 및 숨김

Overlays 메뉴를 열려면 씬 뷰 에서 아무 곳이나 클릭하여 스페이스바를 선택합니다.  
표시하거나 숨기려면 Overlay를 클릭합니다. 오버레이가 이미 표시되어 있는 경우, 눈 모양 아이콘이 해당 오버레이 왼쪽에 표시됩니다. 표시된 옵션에 마우스를 올리면 씬 뷰 에서 오버레이가 파란색으로 강조됩니다.



## 모든 오버레이 숨김

백틱(`) 키를 눌러서 숨김/표시를 위해 설정된 모든 오버레이를 토글합니다.

# The Scene View

## 오버레이 위치 지정

핸들(=)을 클릭한 상태로 또는 오버레이의 헤더를 클릭하여 씬 뷰 에서 원하는 위치로 드래그합니다.

참고: 씬 뷰 에서 플로팅한 오버레이를 오버랩할 수 있습니다. 오버레이를 클릭하여 오버레이를 앞으로 가져옵니다.

## 오버레이 도킹

오버레이의 핸들(=)을 클릭한 상태로 또는 오버레이의 헤더를 클릭하여 씬 뷰 의 테두리 위로 드래그하고 테두리가 파란색으로 강조되면 릴리스합니다. 오버레이는 강조 표시된 위치에 자동으로 도킹됩니다.

## 오버레이를 이미 도킹된 다른 오버레이가 있는 위치로 이동

원하는 위치가 파란색으로 강조 표시될 때까지 오버레이를 드래그한 후 릴리스합니다. 오버레이는 자동으로 재배열됩니다.

참고: 방향과 같이 툴바가 아닌 오버레이는 도킹되면 축소됩니다. 아래쪽 화살표를 클릭하여 오버레이 옵션에 액세스하십시오.

## 오버레이 축소 및 확장

오버레이를 축소하려면 핸들(=)을 마우스 오른쪽 버튼을 클릭하거나 오버레이의 테두리를 클릭하여 Collapse를 선택합니다. 확장하려면 마우스 오른쪽 버튼을 클릭하여 Expand를 선택합니다.

## 오버레이 스내핑

오버레이의 핸들(=)을 클릭한 채 씬 뷰 의 코너로 드래그하여 파란색으로 강조된 사각형이 표시되면 릴리스합니다.

# The Scene View

## 오버레이 방향 변경

핸들(=)을 마우스 오른쪽 버튼으로 클릭하여 Horizontal 또는 Vertical을 선택합니다.  
오버레이가 플로팅하거나 씬 뷰 에 스내핑되면 Panel을 선택하여 오버레이를 이름이 지정된 패널로 변경합니다.

## 오버레이 설정 저장

1. 스페이스바를 눌러 오버레이 메뉴를 엽니다.
2. 드롭다운 메뉴에서 Save Preset을 선택합니다.
3. 프리셋에 이름을 지정하고 Save를 클릭합니다. 프리셋은 드롭다운 메뉴에 저장됩니다.

참고: 오버레이 프리셋을 이런 방식으로 저장하는 경우 다른 프로젝트로 로드할 수 있는 파일을 생성하지는 않지만 저장된 프리셋은 해당 컴퓨터 상의 다른 Unity 프로젝트에서는 사용할 수 있습니다.

오버레이 설정은 에디터 레이아웃을 저장할 때에도 저장됩니다.

## 오버레이 설정 전환

1. 스페이스바를 눌러 오버레이 메뉴를 엽니다.
2. 드롭다운 리스트에서 오버레이 프리셋을 선택합니다. 씬 뷰 에 오버레이 설정이 나타납니다.

# The Scene View

## 오버레이 설정 импорт 및 익스포트

1. 스페이스바를 눌러 오버레이 메뉴를 엽니다.
2. 드롭다운 메뉴에서 Load Preset From File을 선택합니다.
3. 프리셋인 .wpr 파일을 찾아 Open을 클릭합니다. 해당 프리셋은 드롭다운 메뉴에 추가됩니다.

오버레이 설정을 익스포트하려면 .wpr 파일로 저장한 후 다음과 같이 다른 프로젝트로 импорт합니다.

1. 스페이스바를 눌러 오버레이 메뉴를 엽니다.
2. 드롭다운 메뉴에서 Save Preset to File을 선택합니다.
3. 프리셋에 이름을 지정하고 Save를 클릭합니다. 해당 프리셋은 폴더로 드롭다운 메뉴에 저장됩니다.

# The Scene View

## 커스텀 오버레이 생성

### 패널 오버레이

1. 모든 오버레이는 반드시 오버레이 기본 클래스를 상속해야 하며 `CreatePanelContent()` 메서드를 구현해야 합니다. 이는 as-is를 사용하거나 툴바 요소를 추가할 수 있는 기본 패널을 생성합니다.
2. 에디터 폴더에 새로운 C# 스크립트를 생성하여 이름을 지정합니다.
3. 새로운 스크립트를 엽니다.
4. 메인 중괄호 사이에 있는 보일러플레이트 콘텐츠를 제거하고 `UnityEditor.Overlays` 네임스페이스 안에 `Overlay` 클래스를 구현합니다.
5. `CreatePanelContent` 함수를 오버라이드하고 콘텐츠를 시각적 요소에 추가합니다.
6. `Overlay` 속성을 클래스에 추가하고 이 오버레이를 사용할 수 있는 창의 타입을 지정합니다. `EditorWindow`를 타입으로 지정하면 오버레이를 모든 에디터 창에서 사용할 수 있으며 `SceneView`를 지정하면 오버레이를 씬 뷰에서만 사용할 수 있습니다.
7. 이름, ID, 표시 이름을 추가합니다.
8. 선택 사항: `Icon` 속성을 `Overlay` 클래스에 추가하여 압축 모드에서 사용할 아이콘을 지정합니다. 아무 아이콘도 지정되어 있지 않은 경우 기본적으로 시스템은 오버레이 이름의 처음 두 글자(또는 처음 두 단어의 처음 두 글자)를 사용하십시오.

# The Scene View

```
using UnityEditor;
using UnityEditor.Overlays;
using UnityEngine.UIElements;

[Overlay(typeof(SceneView), "My Custom Toolbar", true)]
public class MyToolBarOverlay : Overlay
{
    public override VisualElement CreatePanelContent()
    {
        var root = new VisualElement() { name = "My Toolbar Root" };
        root.Add(new Label() { text = "Hello" });
        return root;
    }
}
```

# The Scene View

**툴바 오버레이**는 툴바 아이템을 유지하고 EditorToolbarElement 컬렉션으로 구성된 컨테이너입니다.

툴바 오버레이에는 빌트인 수평, 버티컬, 패널 레이아웃이 있습니다. ToolbarOverlay는 파라미터가 없는 생성자를 구현하며 EditorToolbarElementAttribute ID를 전달합니다. 패널 오버레이와는 달리 콘텐츠는 요소의 스트립을 형성하기 위해 수집된 스탠드얼론 조각으로 정의됩니다.

1. 패널 오버레이와 마찬가지로 C# 스크립트를 생성하고 에디터 폴더 내에 저장하여 시작한 다음 스크립트를 열고 편집합니다.
2. 툴바 요소를 추가합니다.
3. 툴바 요소를 오버레이 생성자에 추가합니다.
4. 패널 오버레이를 추가하고 툴바 요소를 사용하여 구현합니다.

툴바 오버레이를 생성하는 경우 다음을 수행하십시오.

- [EditorToolbarElement(Identifier, EditorWindowType)]을 사용하여 ToolbarOverlay 구현에서 사용하기 위해 툴바 요소를 등록합니다.
- 모든 오버레이는 반드시 OverlayAttribute로 태그되어야 합니다.
- 툴바 오버레이는 반드시 ToolbarOverlay를 상속하고 파라미터가 없는 생성자를 구현해야 합니다.
- 툴바 콘텐츠는 기본 생성자에 전달되는 문자열 ID로 채워져 있습니다.
- ID는 EditorToolbarElementAttribute로 정의됩니다.
- Icon 속성은 오버레이가 축소되면 볼 수 있는 아이콘을 정의합니다. 아이콘이 제공되지 않는 경우 오버레이 이름의 처음 두 글자(또는 처음 두 단어의 처음 두 글자)가 사용됩니다.



# The Scene View

오버레이 안에서 ToolbarOverlay 특정 요소를 구현할 때는 다음을 수행하십시오.

- IAccessContainerWindow 인터페이스는 툴바 전용이므로 요소는 해당 컨텍스트를 인식하지 못합니다. DropdownToggleExample에서 요소를 토글해도 아무 작업도 하지 않습니다.
- 툴바 요소는 오버레이에서 스타일링을 전달하지 않습니다. 시각 요소의 경우 UIElement 스타일링을 사용하십시오.

# The Scene View

```
using UnityEditor;
using UnityEditor.Overlays;
using UnityEditor.Toolbars;
using UnityEngine;
using UnityEngine.UIElements;

[EditorToolbarElement(id, typeof(SceneView))]
class DropdownExample : EditorToolbarDropdown
{
    public const string id = "ExampleToolbar/Dropdown";
    static string dropChoice = null;

    public DropdownExample()
    {
        text = "Axis";
        clicked += ShowDropdown;
    }

    void ShowDropdown()
    {
        var menu = new GenericMenu();
        menu.AddItem(new GUIContent("X"), dropChoice == "X", () => { text = "X"; dropChoice = "X"; });
    }
}
```

# The Scene View

```
        menu.AddItem(new GUIContent("Y"), dropChoice == "Y", () => { text = "Y"; dropChoice = "Y"; });
        menu.AddItem(new GUIContent("Z"), dropChoice == "Z", () => { text = "Z"; dropChoice = "Z"; });
        menu.ShowAsContext();
    }
}

[EditorToolbarElement(id, typeof(SceneView))]
class ToggleExample : EditorToolbarToggle
{
    public const string id = "ExampleToolbar/Toggle";

    public ToggleExample()
    {
        text = "Toggle OFF";
        this.RegisterValueChangedCallback(Test);
    }

    void Test(ChangeEvent<bool> evt)
    {
        if (evt.newValue)
        {
            Debug.Log("ON");
        }
    }
}
```

# The Scene View

```
        text = "Toggle ON";
    }
    else
    {
        Debug.Log("OFF");
        text = "Toggle OFF";
    }
}

[EditorToolbarElement(id, typeof(SceneView))]
class DropdownToggleExample : EditorToolbarDropdownToggle, IAccessContainerWindow
{
    public const string id = "ExampleToolbar/DropdownToggle";
    public EditorWindow containerWindow { get; set; }
    static int colorIndex = 0;
    static readonly Color[] colors = new Color[] { Color.red, Color.green, Color.cyan };

    public DropdownToggleExample()
    {
        text = "Color Bar";
        tooltip = "Display a color rectangle in the top left of the Scene view." +
            "Toggle on or off, and open the dropdown to change the color.";
    }
}
```

# The Scene View

```
dropdownClicked += ShowColorMenu;
SceneView.duringSceneGui += DrawColorSwatch;
}

void DrawColorSwatch(SceneView view)
{
    if (view != containerWindow || !value)
    {
        return;
    }

    Handles.BeginGUI();
    GUI.color = colors[colorIndex];
    GUI.DrawTexture(new Rect(8, 8, 120, 24), Texture2D.whiteTexture);
    GUI.color = Color.white;
    Handles.EndGUI();
}

void ShowColorMenu()
{
    var menu = new GenericMenu();
    menu.AddItem(new GUIContent("Red"), colorIndex == 0, () => colorIndex = 0);
    menu.AddItem(new GUIContent("Green"), colorIndex == 1, () => colorIndex = 1);
}
```

# The Scene View

```
        menu.AddItem(new GUIContent("Blue"), colorIndex == 2, () => colorIndex = 2);
        menu.ShowAsContext();
    }

    [EditorToolbarElement(id, typeof(SceneView))]
    class CreateCube : EditorToolbarButton
    {
        public const string id = "ExampleToolbar/Button";

        public CreateCube()
        {
            text = "Create Cube";
            icon = AssetDatabase.LoadAssetAtPath<Texture2D>("Assets/CreateCubeIcon.png");
            tooltip = "Instantiate a cube in the scene.";
            clicked += OnClick;
        }

        void OnClick()
        {
            var newObj = GameObject.CreatePrimitive(PrimitiveType.Cube).transform;
            Undo.RegisterCreatedObjectUndo(newObj.gameObject, "Create Cube");
        }
    }
}
```

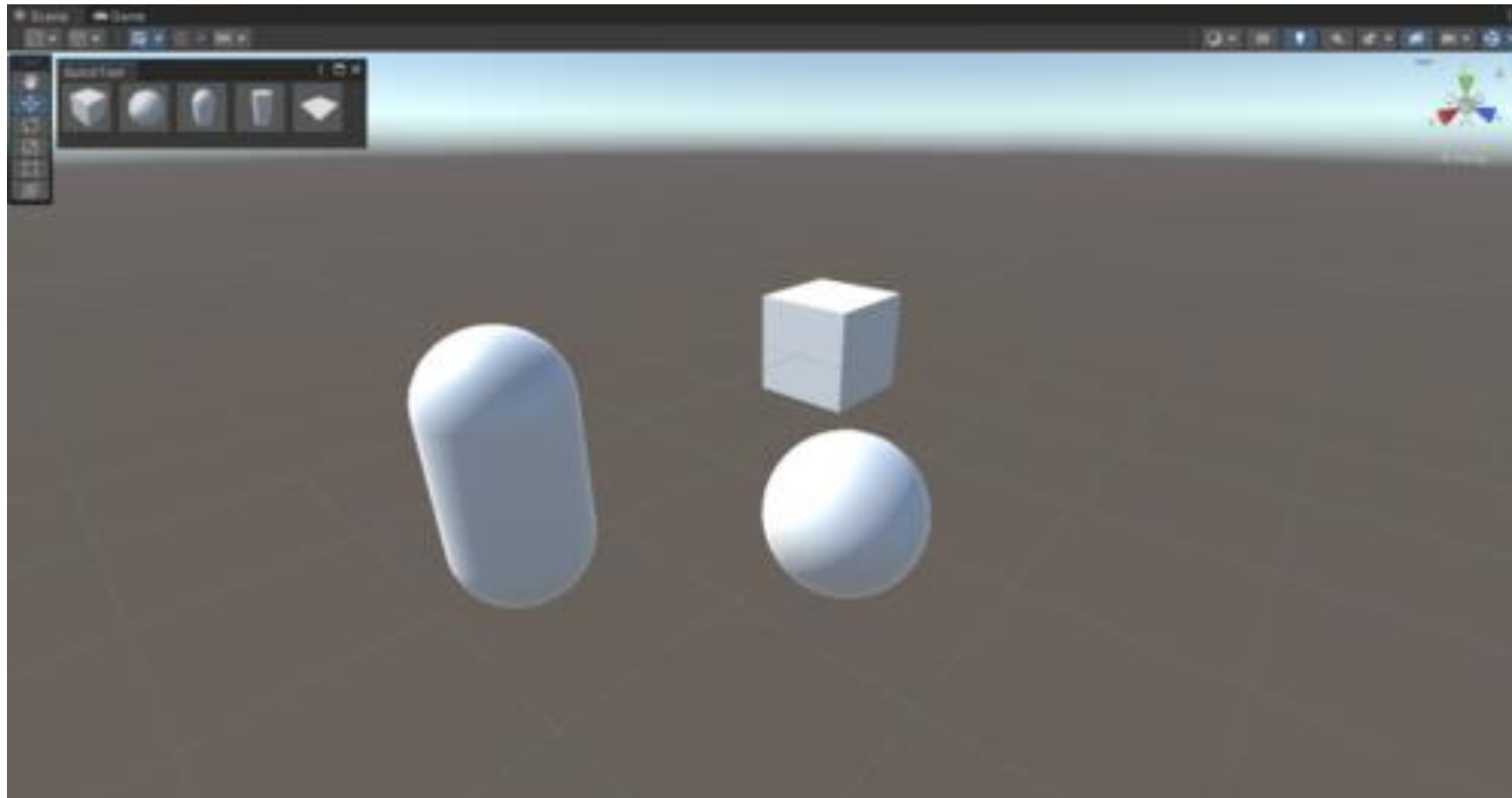
# The Scene View

```
[Overlay(typeof(SceneView), "ElementToolbars Example")]
[Icon("Assets/unity.png")]
public class EditorToolbarExample : ToolbarOverlay
{
    EditorToolbarExample() : base(
        CreateCube.id,
        ToggleExample.id,
        DropdownExample.id,
        DropdownToggleExample.id
    )
    {
    }
}
```

# The Scene View

UI Toolkit 을 사용한 커스텀 에디터 윈도우 제작 실습

<https://learn.unity.com/tutorial/ui-toolkit-first-steps#61df0f23edbc2a2bf49579a2>



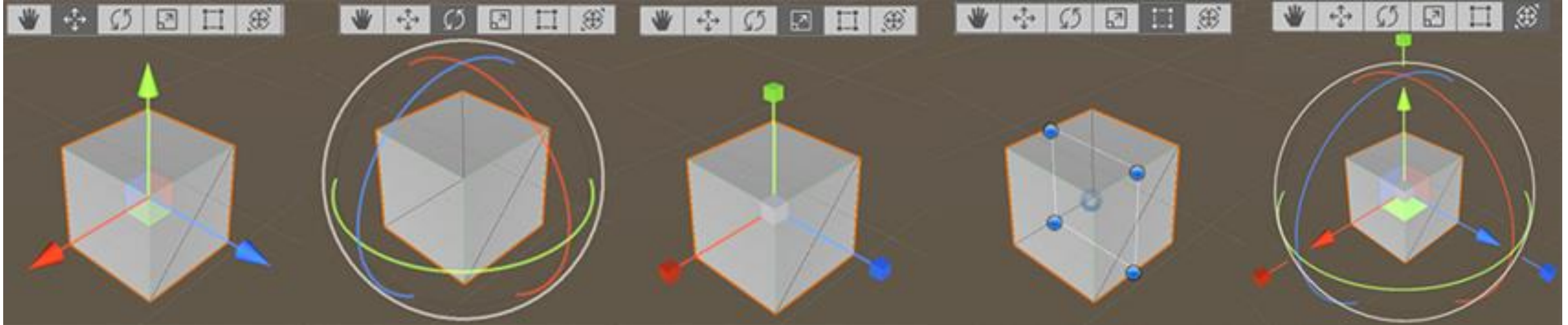


# The Scene View

## 게임 오브젝트 위치 지정

게임 오브젝트의 Transform 컴포넌트를 수정하려면 마우스를 사용하여 기즈모 축을 조작하거나, 인스펙터에서 Transform 컴포넌트의 숫자 필드에 값을 직접 입력하십시오.

또는 단축키를 사용하여 각 Transform 모드를 선택할 수 있습니다. Move를 선택하려면 W 키를, Rotate를 선택하려면 E 키를, Scale을 선택하려면 R 키를, RectTransform을 선택하려면 T 키를, Transform을 선택하려면 Y 키를 누르거나 씬 뷰에서 툴 오버레이를 사용합니다.



# The Scene View

## 이동

Move 기즈모 가운데에는 단일 평면 안에서 게임 오브젝트를 드래그하는 데 사용할 수 있는 작은 삼각형이 세 개 있습니다. 한 축이 가만히 정지된 상태에서 두 축을 한꺼번에 이동할 수 있습니다. Shift 키를 누른 채로 기즈모의 가운데를 클릭하고 드래그하면 기즈모 가운데가 납작한 사각형으로 변합니다. 납작한 사각형은 씬 뷰 카메라가 향하는 방향을 기준으로 게임 오브젝트를 평면에서 이리저리 움직일 수 있음을 나타냅니다.

## 회전

Rotate 도구를 선택한 상태에서 게임 오브젝트 주위에 나타나는 와이어프레임 구체 기즈모의 축을 클릭하고 드래그하여 오브젝트의 회전을 변경합니다. 이동 기즈모와 마찬가지로, 마지막으로 변경한 축이 노란색으로 표시됩니다. 빨간색, 녹색, 파란색 원이 이동 모드에서 나타나는 빨간색, 녹색, 파란색 축 주위를 회전하고 있다고 생각할 수 있습니다. 빨간색이 X축, 녹색이 Y축, 파란색이 Z축을 나타냅니다. 마지막으로, 맨 바깥쪽 원을 사용하여 씬 뷰의 Z축 주위로 게임 오브젝트를 회전합니다. 화면 공간에서 회전하는 것이라고 생각하면 됩니다.

## 스케일

Scale 도구를 사용하여 기즈모 가운데에서 큐브를 클릭하고 드래그하여 게임 오브젝트를 모든 축에서 균일하게 다시 스케일할 수 있습니다. 각 축에 대해 개별적으로 스케일할 수도 있지만, 자식 게임 오브젝트가 있을 때에는 이상하게 보일 수 있으므로 주의해야 합니다.

# The Scene View

## 사각 트랜스폼

RectTransform은 스프라이트 또는 UI 요소와 같은 2D 요소의 위치를 지정할 때 많이 사용되지만, 3D 게임 오브젝트를 조작할 때에도 효과적으로 활용할 수 있습니다. 이동, 스케일, 회전이 하나의 기즈모에 합쳐져 있습니다.

사각형 기즈모 안을 클릭하고 드래그하여 게임 오브젝트를 이동합니다.

사각형 기즈모의 모서리나 가장자리를 클릭하고 드래그하여 게임 오브젝트를 스케일합니다.

가장자리를 드래그하여 한 축을 따라 게임 오브젝트를 스케일합니다.

모서리를 드래그하여 두 축을 따라 게임 오브젝트를 스케일합니다.

게임 오브젝트를 회전하려면 커서를 사각형 모서리 바깥에 가까이 놓습니다. 커서가 변경되어 회전 아이콘이 표시됩니다. 이 영역에서 클릭하고 드래그하여 게임 오브젝트를 회전합니다.

2D 모드에서는 기즈모를 사용하여 씬의 Z축을 변경할 수 없습니다. 하지만 특정 스크립트 기법에서는 Z축을 다른 용도로 활용할 수도 있으므로 인스펙터에서 Transform 컴포넌트를 사용하여 Z축을 설정해도 됩니다.

## 트랜스폼

Transform 도구는 Move, Rotate, Scale 도구가 합쳐진 도구입니다. 이 도구의 기즈모에는 이동 및 회전 핸들이 있습니다. Tool Handle Rotation 을 Local 로 설정하면(아래 참조) 선택된 게임 오브젝트를 트랜스폼 도구의 핸들을 사용하여 스케일할 수도 있습니다.

# The Scene View

## 커스텀 툴

프로젝트가 커스텀 에디터 툴을 사용하는 경우 그중 일부가 게임 오브젝트의 위치 지정을 허용할 수도 있습니다.

씬 뷰 Tools 툴바 오버레이의 Editor Tools 드롭다운 메뉴에서 커스텀 툴에 액세스할 수 있습니다.

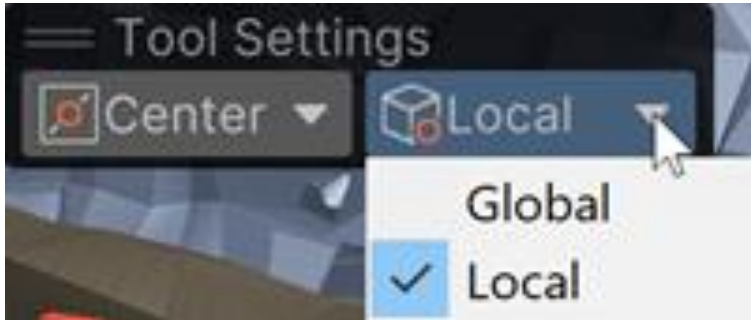


# The Scene View

기즈모 핸들 포지션 토글



# The Scene View



## 포지션

드롭다운 메뉴를 사용하여 Pivot 과 Center 사이를 전환합니다.

- Pivot 은 Transform 컴포넌트에 정의된 대로 게임 오브젝트의 실제 피벗 포인트에 기즈모를 배치합니다.
- Center 는 선택된 게임 오브젝트를 기반으로 중앙 위치에 기즈모를 배치합니다.

## 회전

드롭다운 메뉴를 사용하여 Local 과 Global 사이를 전환합니다.

- Local 을 선택하면 기즈모가 계속 게임 오브젝트를 기준으로 회전합니다.
- Global 을 선택하면 기즈모가 월드 공간 방향으로 고정됩니다.



# The Scene View

## 스내핑

Unity는 다음의 세 가지 스내핑 타입을 제공합니다.

- 월드 그리드 스내핑: X, Y, Z축상에 표시된 그리드에 게임 오브젝트를 스냅하거나 X, Y 또는 Z축에서 증분식으로 게임 오브젝트를 변형합니다. 이 타입은 월드 또는 글로벌 핸들 방향을 사용하는 동안에만 사용할 수 있습니다.
- 표면 스내핑: 게임 오브젝트를 Collider의 교차점에 빠르게 스냅합니다.
- 버텍스 스내핑: 특정 메시의 모든 버텍스를 다른 메시의 버텍스 또는 표면의 포지션에 스냅합니다. 버텍스를 버텍스에, 버텍스를 표면에, 피벗을 버텍스에 스냅할 수 있습니다.

## 표면 스냅

Move 툴을 사용해 가운데에서 드래그하는 동안 Shift와 Control(Mac에서는 Command) 키를 누르고 있으면 게임 오브젝트를 Collider의 교차점에 빨리 스냅할 수 있습니다.

## 버텍스 스냅

버텍스 스냅을 사용하여 씬을 빨리 정리할 수 있습니다. 이렇게 하려면 메시에서 버텍스를 하나 선택한 후 다른 메시의 버텍스와 같은 포지션에 가져다 놓아야 합니다. 버텍스 스냅은 레이싱 게임에서 도로를 정확하게 정렬하거나 파워업 아이템을 메시의 버텍스에 정렬해서 배치하는 등의 용도로 사용할 수 있습니다.

# The Scene View

1. 조작할 메시를 선택하고 Move 툴이 활성화 상태인지 확인합니다.
2. V 키를 길게 눌러 버텍스 스내핑 모드를 활성화합니다.
3. 피벗 지점으로 사용할 메시의 버텍스 위로 커서를 이동합니다.
4. 커서가 원하는 버텍스 위에 놓이면 마우스 왼쪽 버튼을 누른 상태에서 메시지를 다른 메시의 버텍스로 끌어다 놓습니다.
5. 버텍스를 다른 메시의 표면에 스냅하려면 추가 후 Shift+Ctrl(Windows) 또는 Shift+Command(macOS) 키를 길게 누른 상태에서 스냅하려는 표면 위로 이동합니다.
6. 피벗을 다른 메시의 버텍스에 스냅하려면 추가 후 Ctrl(Windows) 또는 Command(macOS) 키를 길게 누른 상태에서 스냅하려는 버텍스로 커서를 옮깁니다.
7. 결과가 만족스러우면 마우스 버튼과 V 키에서 손을 뗍니다. Shift+V 로 이 기능을 켜거나 끌 수 있습니다.



# The Scene View

## 보기 회전

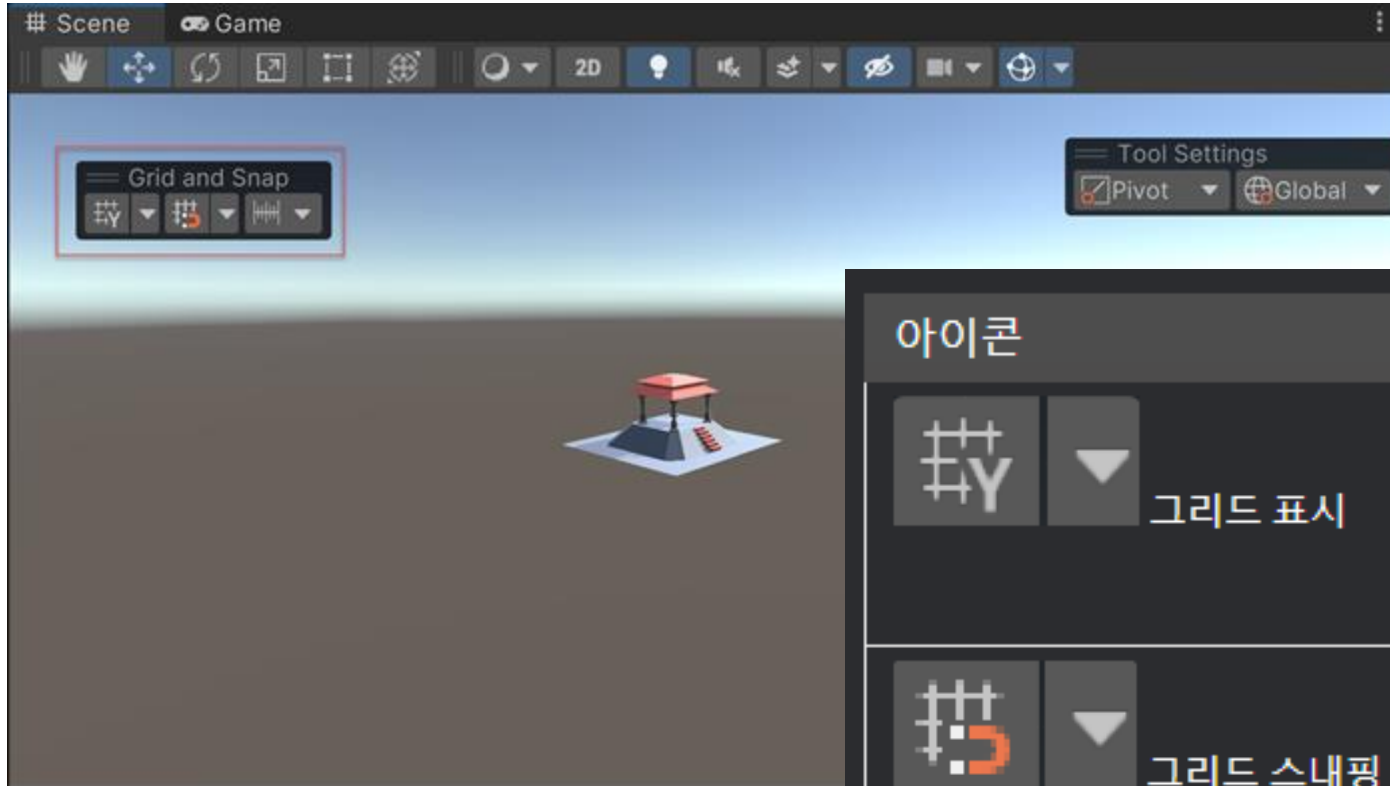
Rotate 툴을 사용하는 동안 Shift 와 Control(Mac에서는 Command)을 길게 눌러 게임 오브젝트를 Collider 표면의 한 점을 향해 회전합니다.







## 화면 공간 트랜스폼

Transform 도구를 사용하는 중에 Shift 키를 길게 누르면 화면 공간 모드가 활성화됩니다. 이 모드에서는 씬이 아닌 화면에 나타나는 게임 오브젝트를 이동하거나, 회전하거나, 스케일할 수 있습니다.

# The Scene View

## 그리드와 스냅 툴바 오버레이



아이콘	동작
  그리드 표시	<u>그리드 적용/해제</u> <u>그리드 축 변경</u> <u>그리드 불투명도 변경</u> <u>그리드를 기즈모 핸들로 이동 또는 그리드 원위치</u>
  그리드 스내핑	<u>그리드 스내핑 적용/해제</u> <u>그리드 크기 변경</u> <u>게임 오브젝트를 축에 정렬</u>
  스냅 인크리먼트	<u>기본 인크리먼트 스냅 값 변경</u>

# The Scene View


정렬, 스냅, 증분 이동

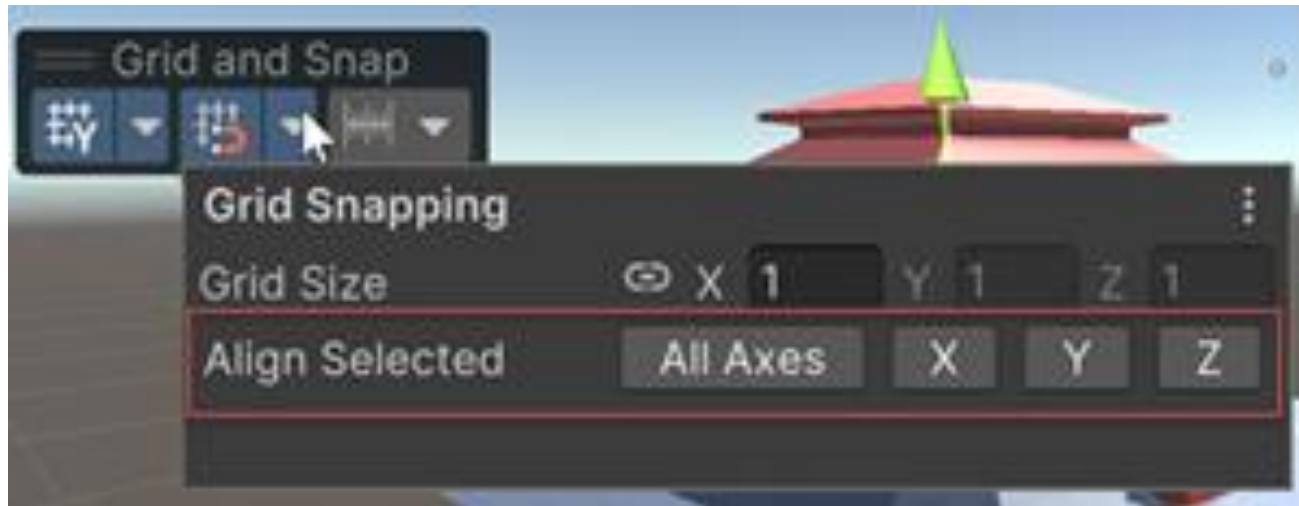
- 그리드에 맞게 정렬(밀기)
- 자동 스냅핑 활성화
- 증분식으로 이동, 회전 및 스케일

# The Scene View

## 그리드에 맞게 정렬(밀기)

특정 축의 그리드에서 가장 가까운 지점에 게임 오브젝트를 정렬

1. Grid and Snap 오버레이 툴바에서 그리드 스내핑 드롭다운 메뉴를 엽니다(  )
2. Align Selected 섹션에서 X 또는 Y, Z 버튼을 클릭하여 푸시하고자 하는 축과 일치시킵니다.




# The Scene View



## 그리드에 맞게 정렬(밀기)

모든 축에서 게임 오브젝트를 한 번에 정렬

1. 그리드에 정렬하려는 게임 오브젝트를 선택합니다.
2. Ctrl+W(Windows) 또는 Command+W(macOS) 단축키를 사용하여 모든 축에서 게임 오브젝트를 그리드로 밀니다.

또는 Grid and Snap 오버레이 툴바에서 그리드 표시 드롭다운 메뉴를 열어() Align Selected 섹션에서 All Axes를 클릭

## 자동 스내핑 활성화


1. Move 툴을 활성화하고 툴 설정 오버레이의 핸들 방향이 Global로 설정되어 있는지 확인합니다(Global).
2. Grid and Snap 툴바 오버레이에서 그리드 스내핑 아이콘()을 클릭합니다. 활성화되면 해당 아이콘은 파란색이 됩니다.

# The Scene View

## 증분식으로 이동, 회전 및 스케일

- 트랜스폼 기즈모 중 하나를 사용하는 동안 Ctrl 키(Windows) 또는 Command 키(macOS)를 길게 누릅니다.

## 기본 증분 스냅 값을 변경

- Grid and Snap 오버레이 툴바에서 스냅 인크리먼트 드롭다운 메뉴를 엽니다().
- Increment Snap 섹션에는 모든 축에 대해 동일한 값이나 다른 값을 설정하도록 해주는 여러 프로퍼티가 들어 있습니다.

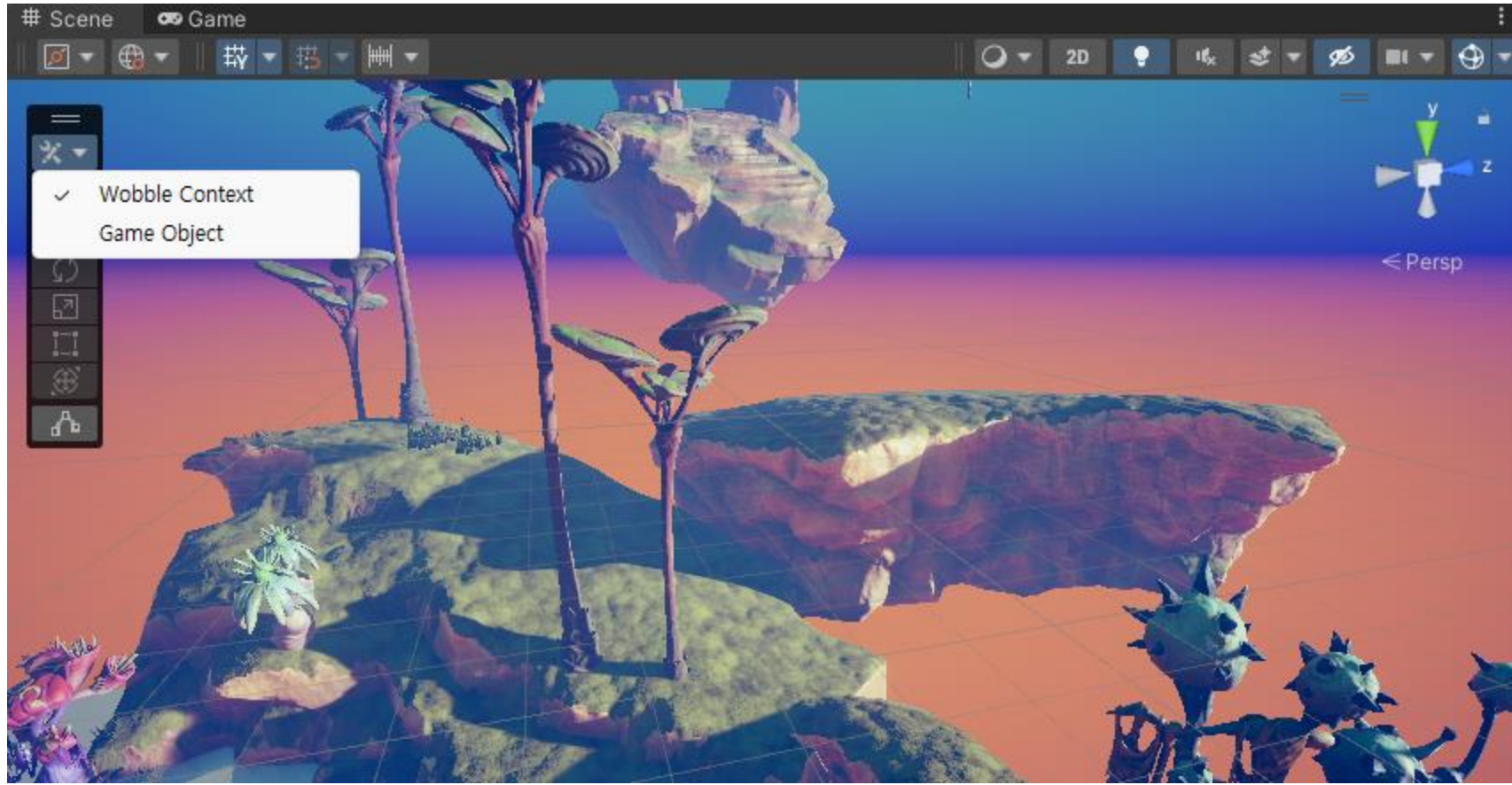


# The Scene View

액션	기본 단축키
<u>그리드 크기 증가</u>	Ctrl+](Windows) 또는 Command+](macOS)
<u>그리드 크기 감소</u>	Ctrl+[ (Windows) 또는 Command+[ (macOS)
<u>그리드를 뒤로 살짝 밀기</u>	Shift+[
<u>그리드를 앞으로 살짝 밀기</u>	Shift+]
<u>그리드로 밀기</u> (선택 항목을 그리드에 정렬)	Ctrl+\(Windows) 또는 Command+\(macOS)
<u>그리드 초기화</u>	(기본 단축키 없음)

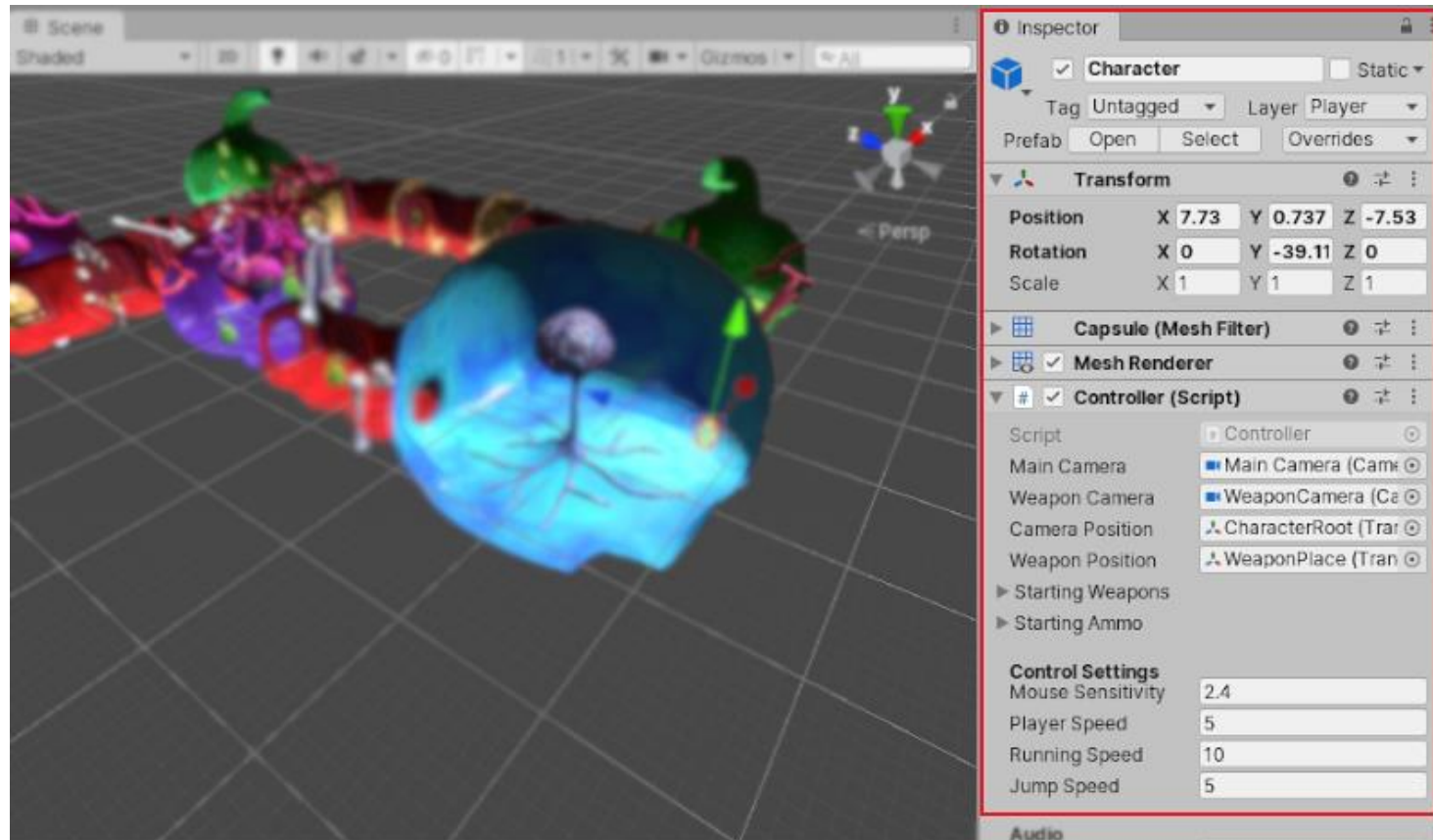
# 커스텀 에디터 툴 예제

<https://docs.unity3d.com/ScriptReference/EditorTools.EditorToolContext.html>





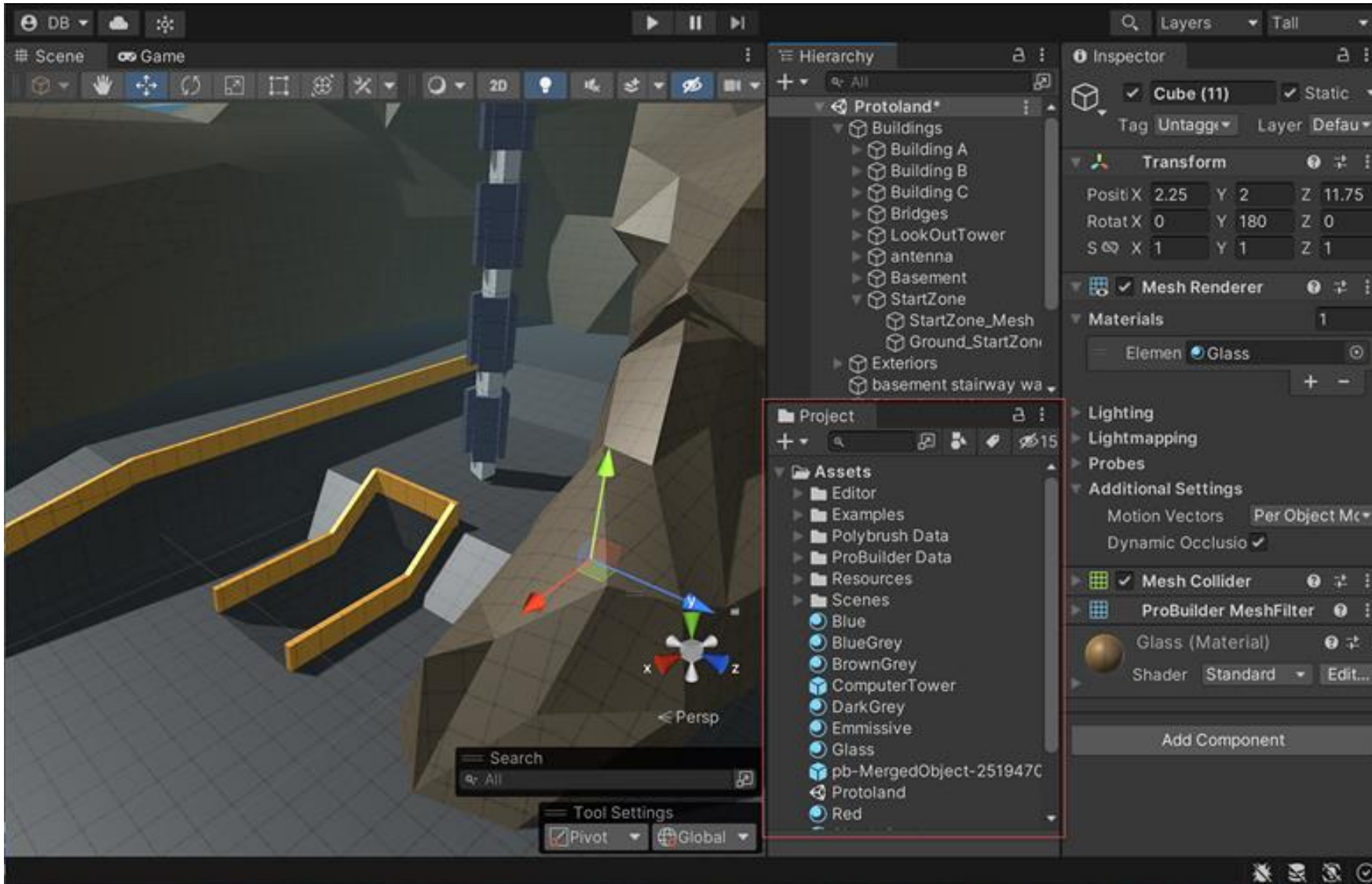
# The Inspector Window



## 인스펙터 창

인스펙터 창을 사용하면 게임 오브젝트, Unity 컴포넌트, 에셋, 머티리얼, 에디터 내 설정 및 환경 설정을 비롯하여 Unity 에디터의 거의 모든 요소에 대한 설정과 프로퍼티를 확인하고 편집할 수 있음.

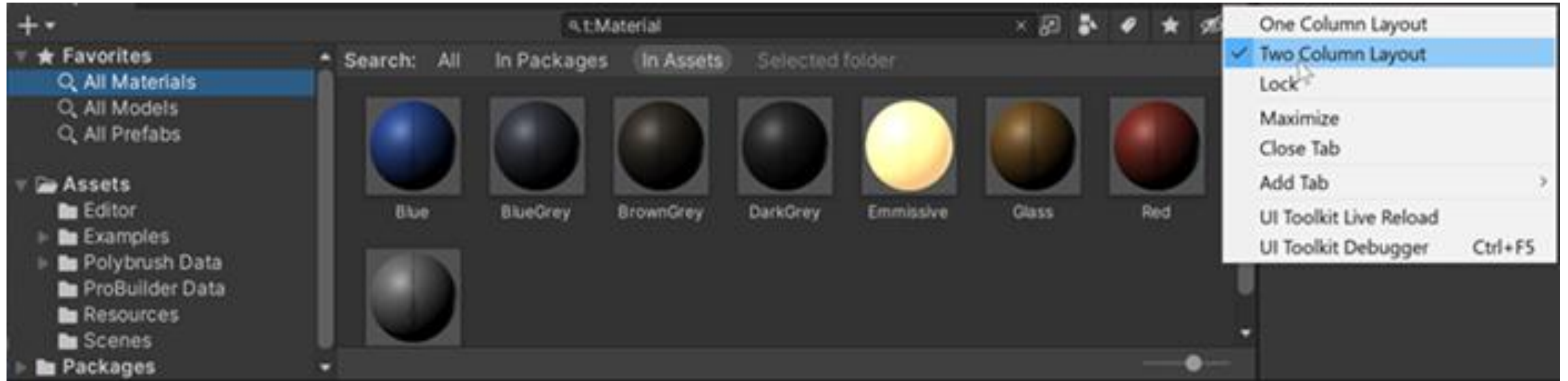
# The Project Window



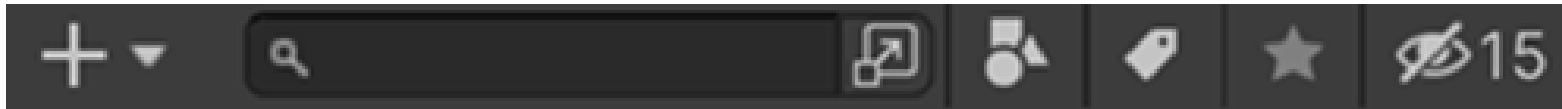
## 프로젝트 창

프로젝트 창은 프로젝트와 관련된 모든 파일을 표시하며 애플리케이션에서 에셋 및 기타 프로젝트 파일을 탐색하고 찾을 수 있는 주요 방법입니다. 기본적으로 새 프로젝트를 시작하면 이 창이 열립니다. 그러나 찾을 수 없거나 닫혀 있는 경우 Window > General > Project를 통해 열거나 Ctrl+5(macOS: Cmd+5)를 누릅니다.

# The Project Window



# The Project Window



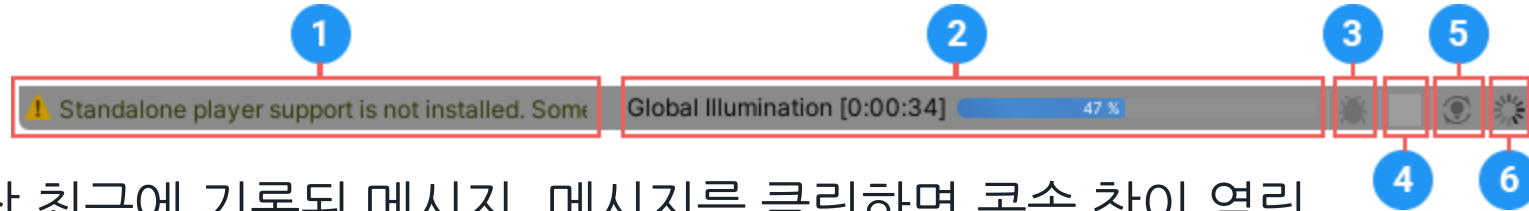
프로퍼티	설명
Create menu	현재 선택한 폴더에 추가할 수 있는 에셋 및 기타 하위 폴더 리스트를 표시합니다.
Search bar	검색 창을 사용하여 프로젝트 내에서 파일을 검색하십시오. 전체 프로젝트(All) 내에서, 프로젝트의 상위 레벨 폴더(개별적으로 나열됨)에서, 현재 선택한 폴더 또는 에셋 스토어에서 검색할 수 있습니다.
Open in Search	[Unity 검색](검색 개요) 툴을 열어 상세히 검색합니다.
Search by Type	이 프로퍼티를 선택하면 검색을 특정 타입(예: 메시, 프리팹, 씬)으로 제한할 수 있습니다.
Search by Label	이 프로퍼티를 선택하면 검색할 태그를 선택할 수 있습니다.
Save Search	왼쪽 패널의 <b>Favorites</b> 아래에 검색을 저장합니다.
Hidden packages count	이 프로퍼티를 선택하면 프로젝트 창의 패키지 가시성을 토글할 수 있습니다.

# The Project Window

단축키	기능
<b>F</b>	선택된 프레임. 즉, 선택된 에셋을 에셋이 포함된 폴더에 표시
<b>Tab</b>	선택 열을 첫 번째 열과 두 번째 열 간 전환(2열 레이아웃)
<b>Ctrl/Cmd + F</b>	검색 필드 선택
<b>Ctrl/Cmd + A</b>	리스트에 표시된 모든 항목 선택
<b>Ctrl/Cmd + D</b>	선택된 에셋 복제
<b>Delete</b>	다이얼로그를 표시하고 삭제(Win)
<b>Delete + Shift</b>	다이얼로그를 표시하지 않고 삭제(Win)
<b>Delete + Cmd</b>	다이얼로그를 표시하지 않고 삭제(OSX)
<b>Enter</b>	선택 항목 이름 바꾸기 시작(OSX)
<b>Cmd + 아래쪽 화살표</b>	선택된 에셋 열기(OSX)
<b>Cmd + 위쪽 화살표</b>	부모 폴더로 바로 이동(OSX, 2열 레이아웃)
<b>F2</b>	선택 항목 이름 바꾸기 시작(Win)
<b>Enter</b>	선택된 에셋 열기(Win)
<b>Backspace</b>	부모 폴더로 바로 이동(Win, 2열 레이아웃)
<b>오른쪽 화살표</b>	선택된 항목 확장(트리 뷰 및 검색 결과). 항목이 이미 확장된 경우 첫 번째 자식 항목이 선택됩니다.
<b>왼쪽 화살표</b>	선택된 항목 축소(트리 뷰 및 검색 결과). 항목이 이미 축소된 경우 부모 항목이 선택됩니다.
<b>Alt + 오른쪽 화살표</b>	에셋을 미리보기로 표시할 때 항목 확장
<b>Alt + 왼쪽 화살표</b>	에셋을 미리보기로 표시할 때 항목 축소

# The Status Bar

- 상태 표시줄은 다양한 유니티 프로세스에 대한 알림과 관련 툴 및 설정에 대한 빠른 액세스를 제공. 상태 표시줄은 이동하거나 재정렬 할 수 없음.

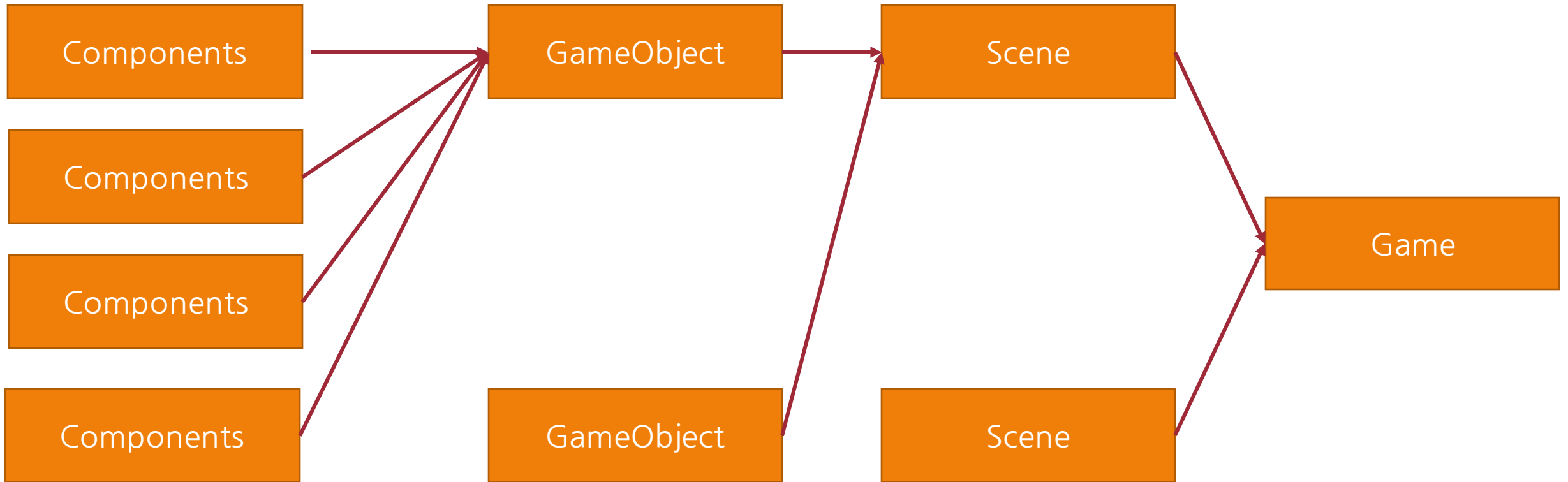


1. 콘솔 창에 가장 최근에 기록된 메시지. 메시지를 클릭하면 콘솔 창이 열림.
2. 다양한 비동기 작업(예: 셰이더 컴파일, 라이트맵 베이킹, 오클루전 컬링)에 대한 전역 진행 표시줄. 진행 표시줄을 클릭하면 백그라운드 작업 창이 열림. 여기에는 각 작업 및 하위 작업의 진행률이 표시
3. 현재 코드 최적화 모드. 코드 최적화 아이콘을 클릭하면 디버그 모드와 릴리스 모드 간에 전환
4. 캐시 서버 상태. 아이콘을 클릭하면 캐시 서버에 대한 추가 정보를 가져오고 끊어진 연결을 다시 연결
5. 전역 조명에 대한 자동 조명 생성 상태. 상태 텍스트를 클릭하면 라이팅 창이 열림. 조명을 수동으로 생성하거나 Unity가 자동으로 생성하는 경우에는 진행 표시줄이 메시지 표시됨
6. Unity가 C# 스크립트를 컴파일하거나 비동기 작업을 실행할 때 나타나는 작업 표시기(스피너).

# 유니티 엔진이 동작하는 원리



# 유니티 엔진이 동작하는 원리





# 유니티 엔진이 동작하는 원리

- 유니티 아키텍처
- 상속을 이용한 개발 방법
- 컴포넌트 패턴의 장점
- 컴포넌트와 게임 오브젝트의 관계
- MonoBehaviour
- 메시지 기반 방식의 원리

# 유니티 아키텍처

- 유니티의 .NET 개요
- 스크립트 백엔드
- 에디터에서 코드 재로드
- 스크립트 직렬화
- 스크립트 컴파일

# 상속을 이용한 개발 방법

- 상속

상속은 객체들 간의 관계를 구축하는 방법이다.

부모클래스의 메소드 혹은 필드를 자식 클래스에게 물려주는 것

기존 클래스에 기능을 추가하거나 재정의하여 새로운 클래스를 정의한다.



`class Monster`



`class Orc : Monster`



`class OrcChieftan : Orc`

# 상속을 이용한 개발 방법



class Monster

인공지능 기능  
애니메이션 기능  
공격과 방어 기능  
물리 기능  
기타 필수 기능



class Orc : Monster

초록색 피부  
오크의 애니메이션  
오크의 스킬  
그외 오크의 고유 기능



class OrcChieftan : Orc

대장 모자  
새로운 무기와 강력한 스킬  
그외 오크 대장의 고유 기능

# 상속을 이용한 개발 방법

- 상속의 장점

코드를 재사용

코드의 간결성

개발 시간 단축

유지보수 용이

다형성 구현

# 상속을 이용한 개발 방법

- 상속의 단점

상위 클래스에 강하게 결합한다.

다중 상속에 의한 문제 발생

유연성 및 확장성이 떨어짐

기존 코드를 재사용하기 힘든 경우가 발생

기획자가 새로운 오브젝트를 만들려면 프로그래머에게 의존해야 한다.

# 컴포넌트 패턴의 장점

컴포넌트 패턴 혹은 컴포지션 패턴이란 미리 만들어진 부품을 조립하여 완성된 오브젝트를 만드는 방식

- 컴포넌트의 장점

유연한 재사용

기획자의 프로그래머 의존도가 낮아짐

컴포넌트의 독립성 덕분에 추가와 삭제가 쉽다.

# 컴포넌트와 게임 오브젝트의 관계

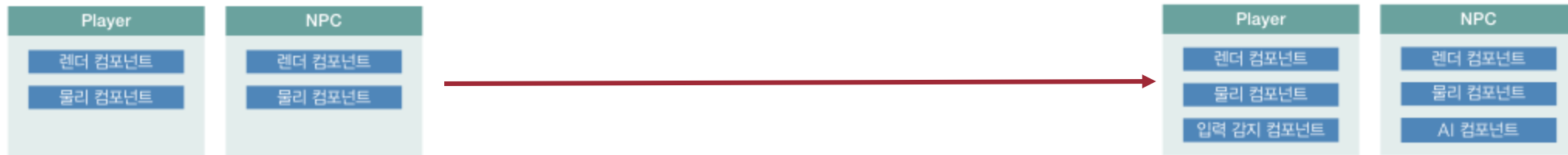
- 게임 오브젝트는 단순한 빈 껍데기

몇 가지 식별 기능과 자신에게 어떠한 컴포넌트가 조립되어 있는지 알 수 있는 기능을 제외하면 특별한 기능은 없음

- 컴포넌트는 스스로 동작하는 독립적인 부품

컴포넌트는 자신과 같은 게임 오브젝트에 추가된 다른 컴포넌트에 관심이 없음

컴포넌트의 기능은 컴포넌트 내부에 완성되어 있기 때문에 다른 컴포넌트에 의존 하지 않음(예외 상황 있음)





# MonoBehaviour

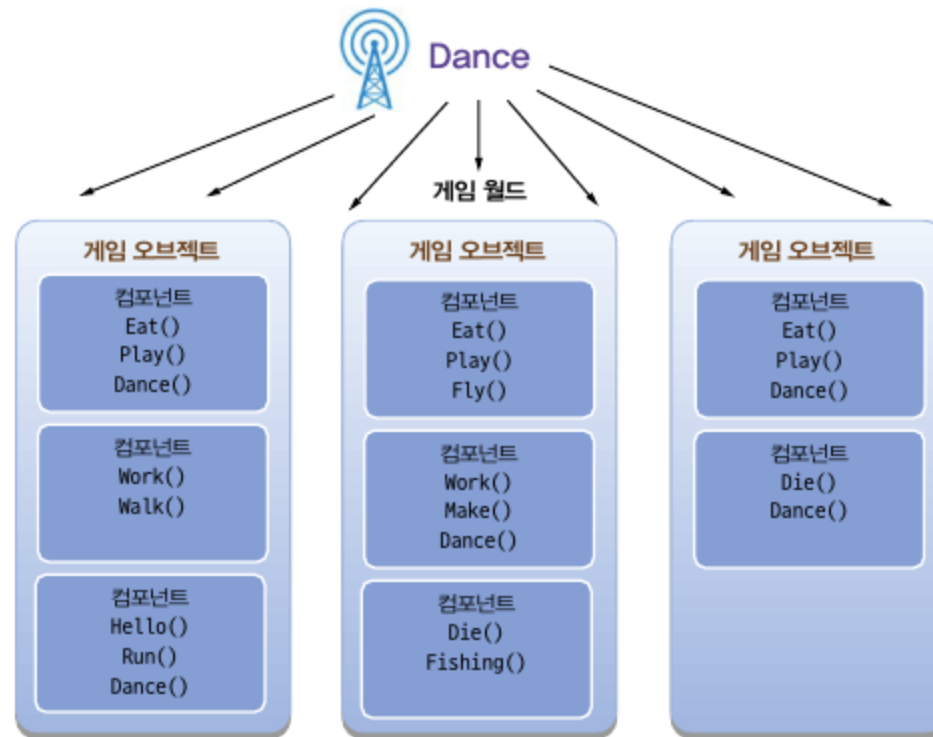
- 유니티의 모든 컴포넌트는 MonoBehaviour 클래스를 상속한다.
- MonoBehaviour 클래스는 유니티에서 미리 만들어 제공하는 클래스이며 컴포넌트에 필요한 기본 기능을 제공
- MonoBehaviour 클래스를 상속한 클래스는 게임 오브젝트에 컴포넌트로 추가될 수 있음
- MonoBehaviour 클래스를 상속해서 만든 컴포넌트는 유니티의 제어를 받게 됨



# 메시지 기반 방식의 원리

컴포넌트 구조에서는 '전체 방송'을 이용하여 컴포넌트의 특정 기능을 간접적으로 실행할 수 있음.

이러한 전체 방송을 '브로드캐스팅'이라 부름



# 메시지 기반 방식의 원리

- 메시지 방식의 특징

메시지를 보내는 쪽은 누가 받게 될지 신경 쓰지 않음

메시지를 받은 쪽은 누가 보냈는지 신경 쓰지 않음

메시지를 받았을 때 메시지에 명시된 기능을 가지고 있다면 실행, 없다면 무시

- 유니티 이벤트 메소드

Start(), Update(), OnTriggerEnter() ...