

유니티(Unity)를 활용한

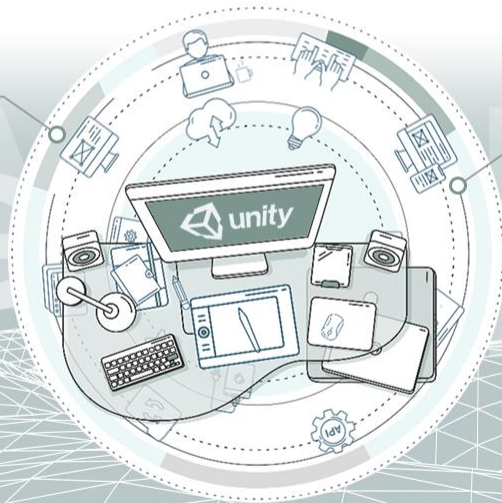
# 그래픽스 프로그래밍

05

Game Engine Architecture &  
Get Started with Unity

Geometry

Animation



PRO

GRAMMING

1

# Game Engine



PROGRAMMING

# Origins of Computer Game Engines

---

- » Game engines arose in the mid-1990s
- » Doom by id provided a separation between :
  - core game components (such as the rendering system, collision detection system, audio system)
  - art assets (models, textures, animations)
  - rules of play
- » Quakes, Unreal, and Unreal Tournament (all FPS games) were designed with the separation in mind
  - Sold licenses to their engine and tools
  - So of you may have done modding using these tools.

## Origins of Computer Game Engines

---

- » It became generic enough that it was possible to implement a wide variety of very different games based on a common core set of components, the game engine (Unity3D and Unreal Engine 4).

## Some Current Game Engines

---

### » Quake family

- Used to create many games
- Has lineage that extends to modern games like Medal of Honor
- Quake and Quake II engines source code are freely available

### » Unreal Engine

- Now at UE4
- Very rich tool set - Kismet
- Large developers network
- Good licensing model - good for small developers

## More Game Engines

### » Unity

- Very feature rich
- Uses Javascript or C# for scripting
- Large community support
- Great for cross-platform development

### » Source Engine

- Games like Half-life 2 and its sequels, Team Fortress 2, and Portal
- Very powerful with good graphics capabilities and a good toolset

### » DICE's Frostbite

- Used to create games like Battlefield 4
- FrostEd - asset creation tool

## Even More Game Engines

### » CryEngine

- Originally developed as a demo for Nvidia
- Used to develop numerous games - starting with Far Cry

### » Sony PhyreEngine

- Uses to create games for the Sony platforms
- Numerous titles have been written with this engine

### » Microsoft XNA and MonoGame

- Based on C# - easy to use
- Used for Xbox and PC games
- Not longer supported - replaced by MonoGame

## 2D Game Engines

- » Designed for non-programmers to build apps for Android and iPhone
- » Examples include
  - Multimedia Fusion 2
  - Game Salad Creator
  - Scratch



# Best Game Engines

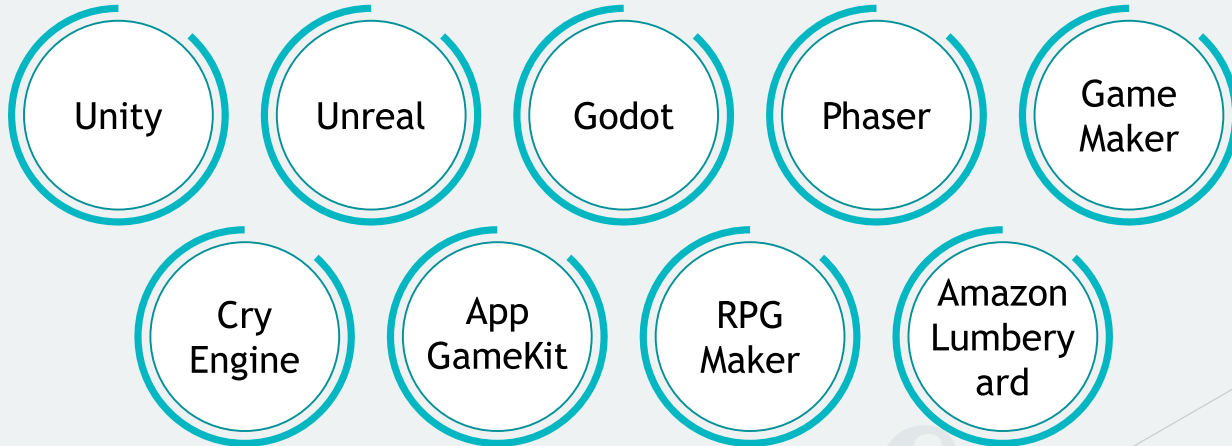
- » GameDev Academy Best Game Engines

  - ▶ <https://gamedevacademy.org/best-game-engines/>

- » Top 10 Game Engines - Ulab - SumDU

  - ▶ <https://ulab.sumdu.edu.ua/top-10-game-engines>

- » Examples include



Game Engines

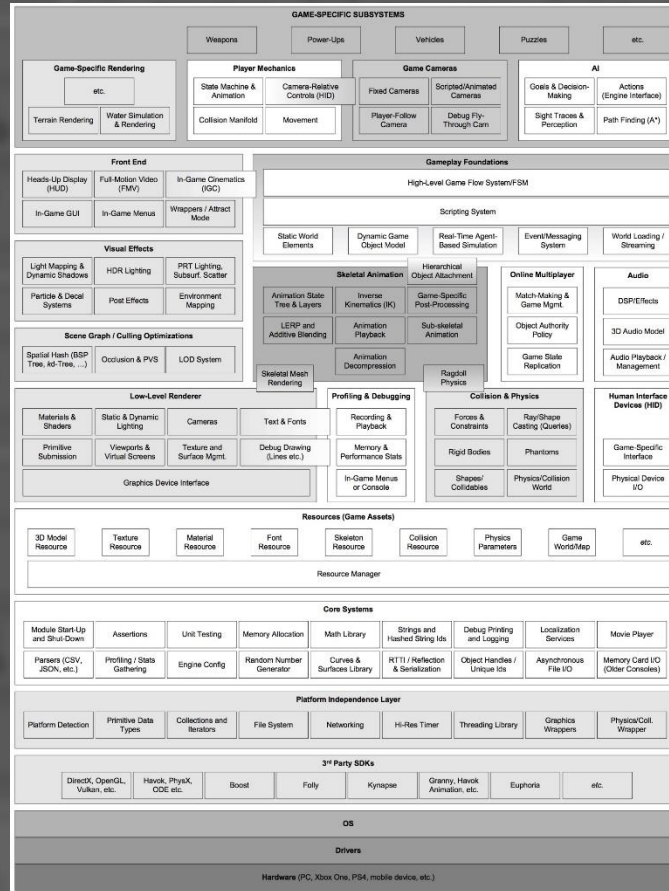
2

## Game Engine Architecture

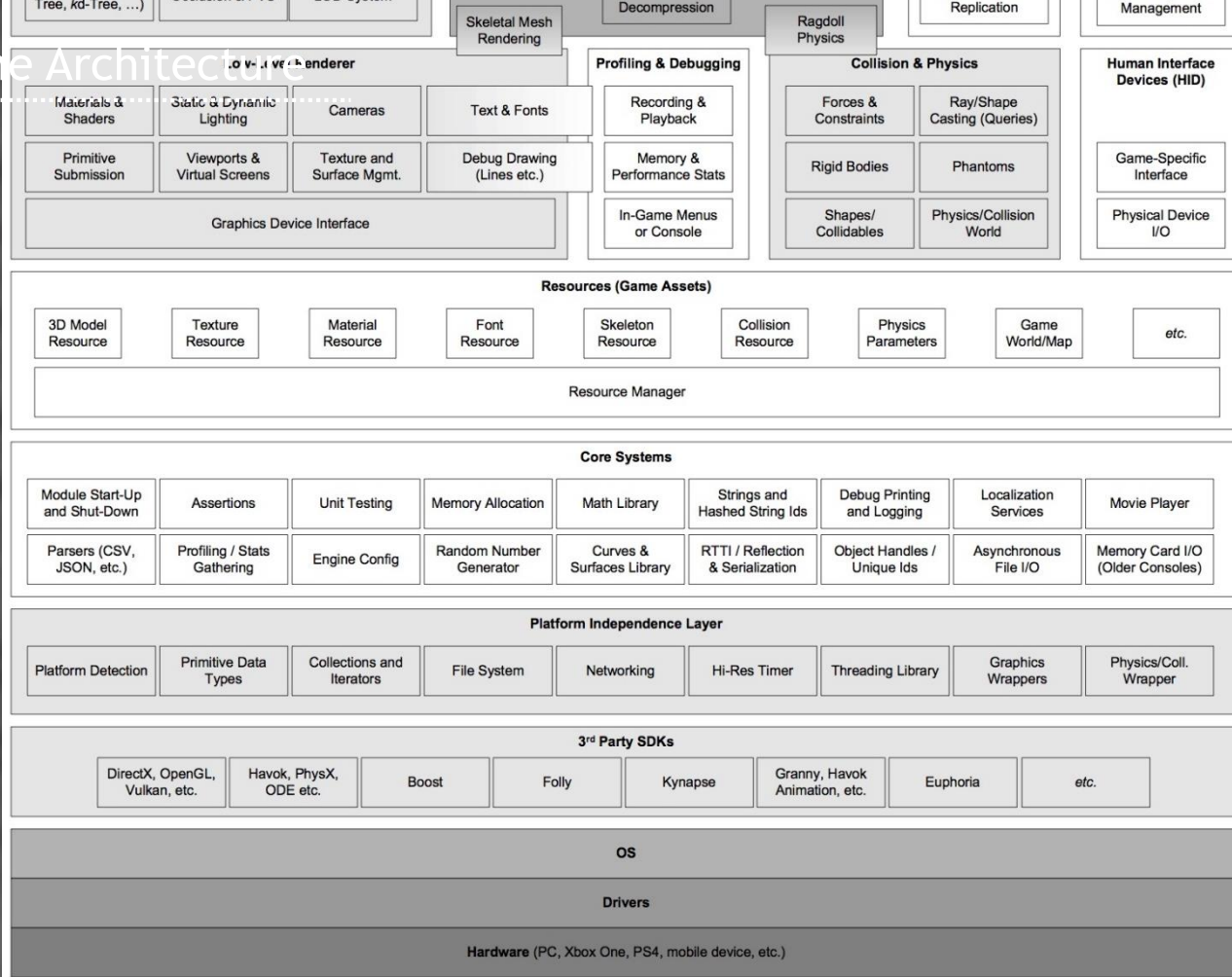


PROGRAMMING

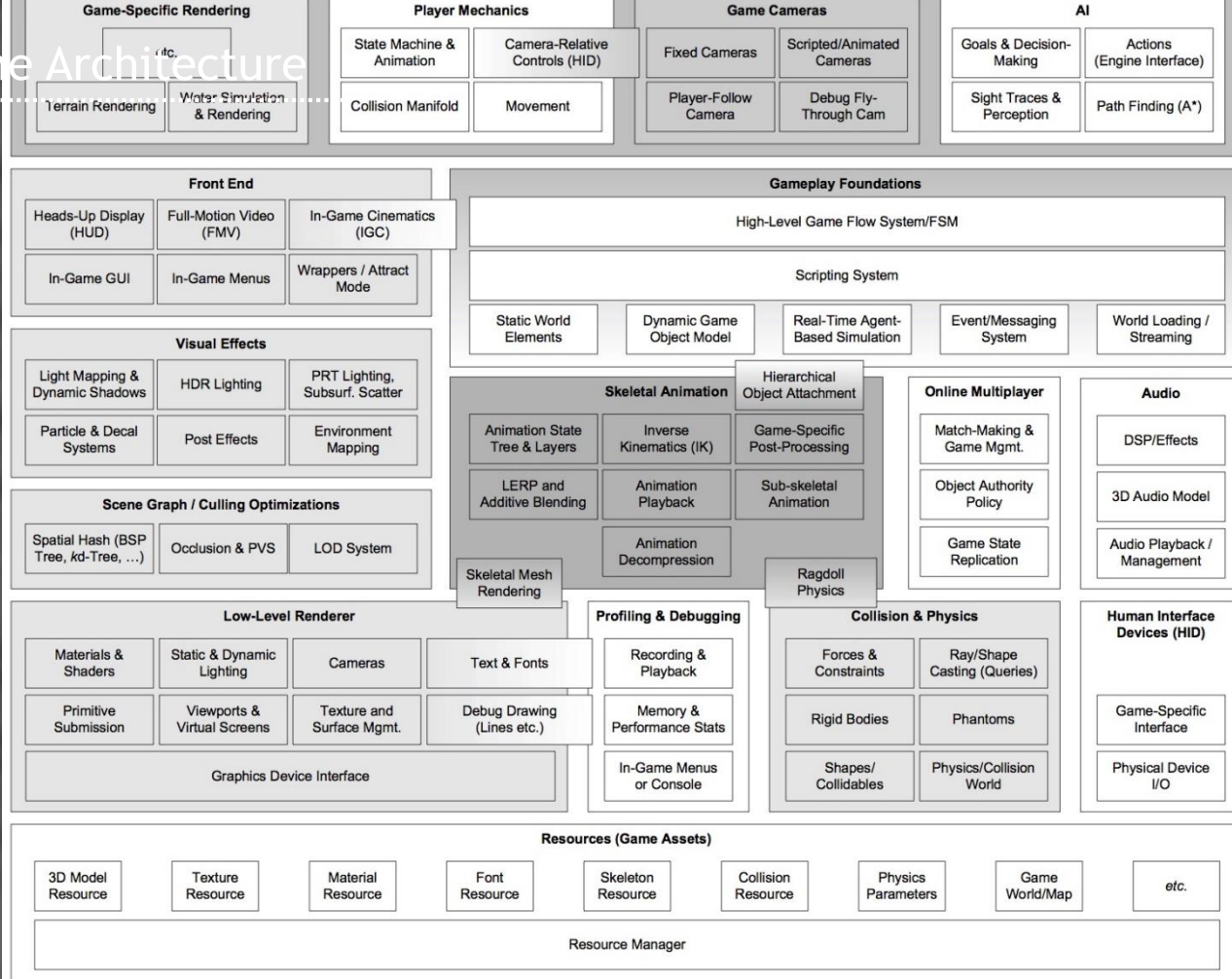
# Game Engine Architecture



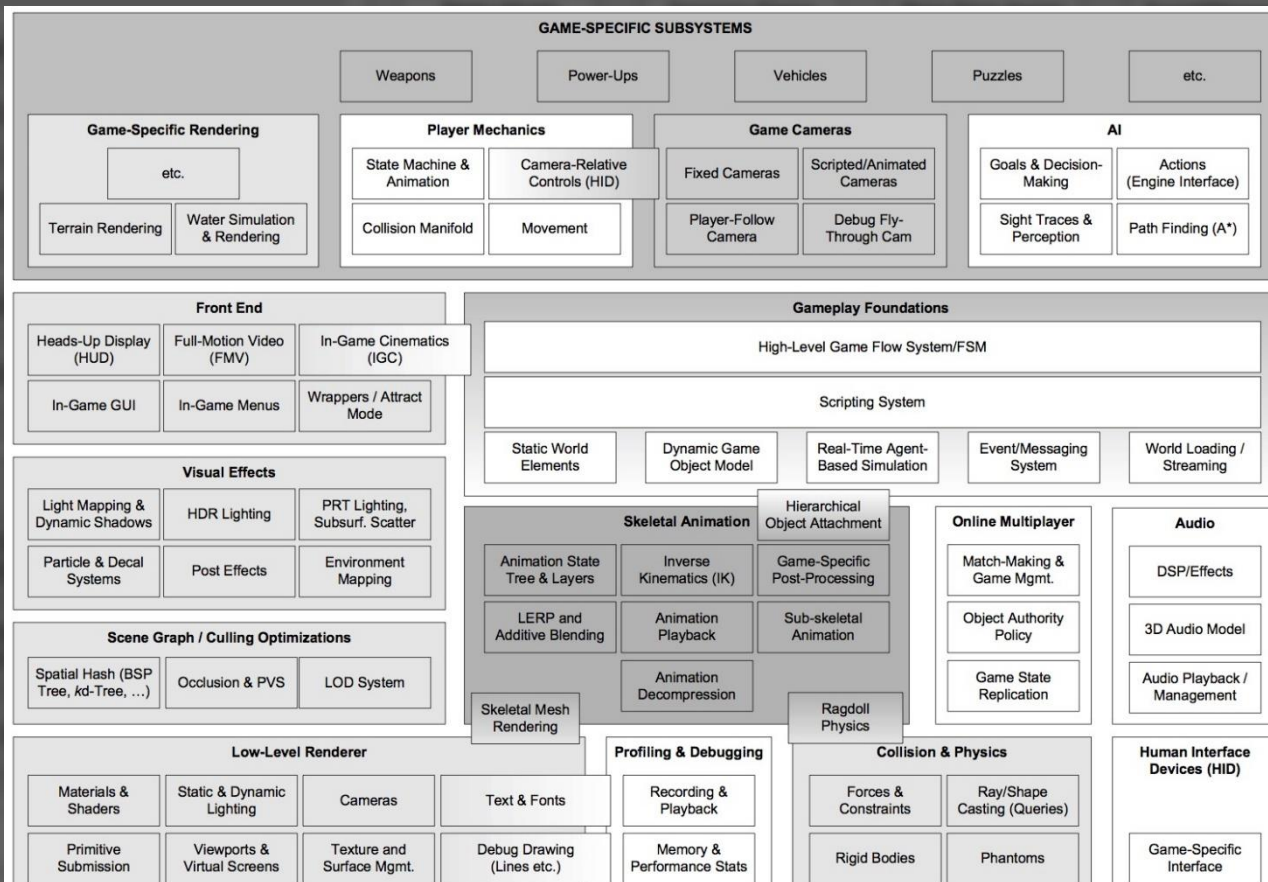
# Game Engine Architecture



# Game Engine Architecture

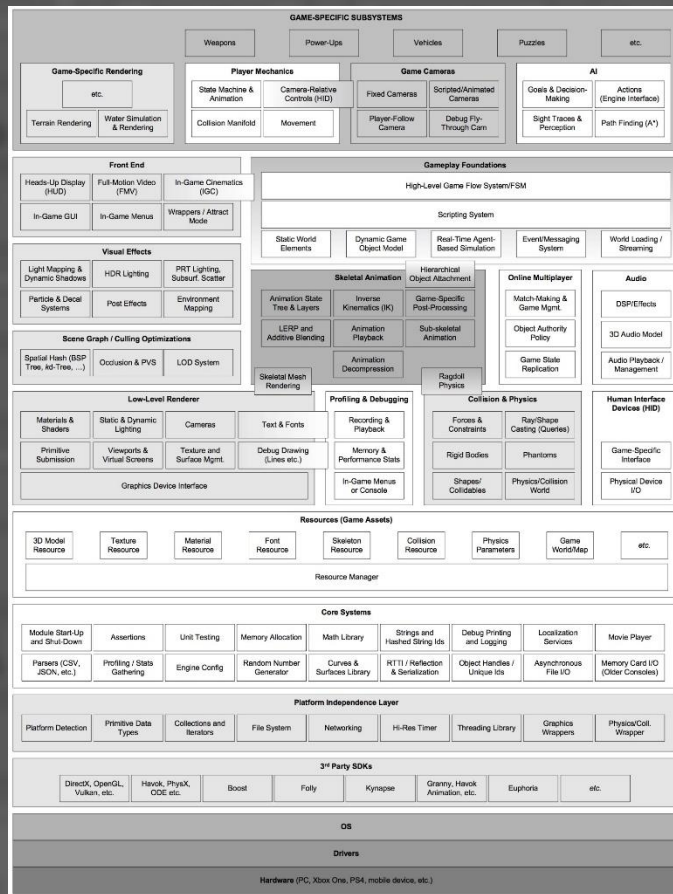


# Game Engine Architecture





# Game Engine Architecture



# Game Engine Architecture

---

- » Low level components
- » 3rd Party SDKs
- » Platform independence layer
- » Core systems
- » Resources manager
- » Rendering engine
- » Profiling/Debugging
- » Collisions and Physics
- » Animation
- » Human Interface Devices
- » Audio
- » Gameplay foundation system

Game Engine Architecture



# Low Level Components

## » Hardware

- This is the system that the game is to run on

## » Device Drivers

- Shield the OS and upper layers from low level device communications details

## » Operating System

- Handles the execution and interruption of multiple programs on a single machine
- Very thin on a console

- » PC (Windows, Linux, Mac OS), game consoles (Xbox, Playstation, Wii), mobile devices (hand-held game consoles, tablets, smart phones)

## 3rd Party SDKs

---

### » Data Structure and Algorithms

- STL - C++ standard template library data structures, strings, stream-based I/O
- Boost - powerful data structures and algorithms

### » Graphics

- OpenGL, DirectX

### » Collisions and Physics

- Havok, PhysX, ODE, Bullet



## 3rd Party SDKs

---

- » Character Animation
  - ▶ Granny
- » Artificial Intelligence
  - ▶ Kynapse



## Platform Independence Layer

---

- » Allows the engine to be developed without the concern of the underlying platform
- » Provides wrappers to common target specific operations
- » Include things like primitive types, network, file systems, etc.

# Core System



- » Assertions - error checking code
- » Memory Management
- » Math library - vector and matrix math, numeric integrators
- » Debugging aids
- » Parsers and Serializers (e.g., CSV, xml-based import and export)
- » File I/O
- » Video playback



## Resource Manager

- » Provides a unified interface for accessing game assets
  - Large graphics programs involve accessing various resources, such as geometric models for characters and buildings, texture images for coloring these geometric models, maps representing the game's world.
- » The level of complexity is dictated by need
  - Often the game programmers must do resource loading directly
  - Engines like UT do unpackaging and complex manipulation of assets in the engine



## Rendering Engine

---

- » Low-level Renderer
- » Scene graph management
- » Visual effects
- » Front end



## Low-Level Renderer

- » Focuses on rendering primitives as quickly and richly as possible
  - Does not consider visibility
- » Graphics Device Interface
  - Access and enumerate the graphics devices
  - Initialize the Graphics Device (GC)
  - Setup buffering



## Low-Level Renderer

### » Others

- Representation of the geometric primitives
- Abstraction of the camera interface
- Material system
- Dynamic lighting system
- Text and fonts





## Scene Graph

- » Limits the number of primitives submitted for rendering
- » Uses frustum culling - remove things outside of the visible screen
- » Spatial subdivision
  - ▶ BSP, quadtree, octree, kd-tree



# Visual Effects

- » Particle systems

- ▶ which are used for rendering smoke, water, fire, explosions

- » Decal systems

- ▶ for painting bullet holes, damage scratches, powder marks from explosions, foot prints, etc

- » Light mapping

- » Dynamic shadows

- » Full screen post effects





## Front End

---

- » Head-Up Display (HUD)
- » Menus
- » GUI for character manipulation
- » Full-motion video (FMV) for cut scenes





## Profile & Debug

---

- » Code timing
- » Display stats on the screen
- » Dumping performance stats
- » Determining memory usage
- » Dumping memory usage
- » Record and playback game events
- » Print statement output control



# Collisions and Physics

- » Usually rigid body dynamics
- » Physics engine creation is its own unique undertaking
- » Many companies use available 3rd party libraries
  - Havok
  - PhysX
  - ODE
  - Bullet



# Animation

- » Five types of animation are used
  - Sprite/texture animation
  - Rigid body hierarchy animation
  - Skeletal animation
  - Vertex animation
  - Morphing
- » Skeletal animations still the most popular



## HID (Human Interface Device)

---

- » Keyboard and mouse abstractions
- » Joypads
- » Specialized controllers
- » Massages raw data into useful information





## Audio

- » Often overlooked until the end
- » Varies in sophistication based on need
- » Many games use existing tools
  - XACT
  - Screamer



## Multiplayer/Networking

### » Four main flavors

Single screen	multiple players on the same screen
Split-screen multiplayer	multiple perspectives on the same screen
Networked multiplayer	multiple computers networked together
Massive multiplayer online games	run in a central server

### » Difficult to convert single to multiplayer, easy to do the opposite

# Gameplay Foundation System

---

- » Most everything that makes the game a game
- » World loading
- » Game object model
- » Static world elements
- » Real-time agent simulations



## Event System

- » Objects need to communicate with one another
- » Easiest to handle this through a common system
- » Objects send messages that are routed to the event handler





## Scripting System

---

- » Allows for the creation on new game logic without recompiling
- » Speeds software development considerably



# Artificial Intelligence Foundations

---

- » Provides AI building blocks
  - Path planning
  - Navigation mesh generation
  - Object avoidance
- » Autodesk has a middleware called Gameware that provides many of these features



## Game-specific Subsystems

---

- » All of the specific stuff needed for a game
- » This layer could be considered outside of the game engine itself



3

## Get Started with Unity



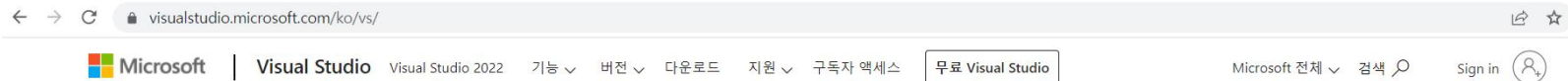


# Unity

- » Unity3D is a widely-used cross-platform game development system.
- » It consists of a game engine and an integrated development environment (IDE).
- » It can be used to develop games for many different platforms, PCs, consoles, mobile devices and deployment on the Web.
- » Tutorials at <https://learn.unity.com/>

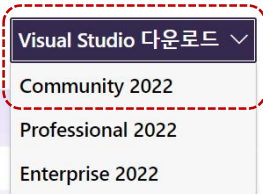


# Visual Studio 2022 Community



## 더 빠르게 코딩 더 스마트하게 작업

Studio 2022를 사용하여 미래 만들기



아이디어는 최고의 도구를 가질 자격이 있습니다.

# Visual Studio 2022 Community

The screenshot shows the Visual Studio Installer window for Visual Studio Community 2022 (version 17.1.0). The 'Installation locations' tab is active, displaying a grid of workload options. Two workloads are selected with blue checkmarks: '.NET desktop development' and 'Game development with Unity'. The 'Game development with Unity' workload is highlighted with a red dashed border. To the right, the 'Installation details' pane shows a tree view of included and optional components, with 'Unity Hub' listed under the optional components and also highlighted with a red dashed border. At the bottom, the location is set to 'C:\Program Files\Microsoft Visual Studio\2022\Community', and the total space required is 6.84 GB. An 'Install' button is visible at the bottom right.

Installing — Visual Studio Community 2022 — 17.1.0

Workloads Individual components Language packs Installation locations

**Mobile development with .NET**   
Build cross-platform applications for iOS, Android or Windows using Xamarin. This includes a preview of the ....

**Desktop development with C++**   
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

**Mobile development with C++**   
Build cross-platform applications for iOS, Android or Windows using C++.

**Gaming (2)**

**Game development with Unity**   
Create 2D and 3D games with Unity, a powerful cross-platform development environment.

**Game development with C++**   
Use the full power of C++ to build professional games powered by DirectX, Unreal, or Cocos2d.

**.NET desktop development**   
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

**Universal Windows Platform development**   
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

**Installation details**

- ▶ Visual Studio core editor
- ▶ .NET desktop development
- ▶ Game development with Unity
  - ▼ Included
    - ✓ Visual Studio Tools for Unity
    - ✓ C# and Visual Basic
  - ▼ Optional
    - Unity Hub

Location  
C:\Program Files\Microsoft Visual Studio\2022\Community [Change...](#)

By continuing, you agree to the [license](#) for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the [3rd Party Notices](#) or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 6.84 GB

[Install while downloading](#) [Install](#)

# Visual Studio 2022 Community

The screenshot shows the Visual Studio Installer window. At the top, the title is "Visual Studio Installer". Below the title, there are two tabs: "Installed" (selected) and "Available". The main content area is divided into two columns. The left column lists installed versions of Visual Studio Community. The right column contains "Developer News" with several news items and a "View more" link. At the bottom right, there is a "Need help?" section with links to "Microsoft Developer Community" and "Visual Studio Support", and the "Installer version 3.1.2196.8931".

Visual Studio Installer

Installed Available

**Visual Studio Community 2022**  
17.1.0  
Powerful IDE, free for students, open-source contributors, and individuals  
[Release notes](#)

**Visual Studio Community 2019**  
16.11.10  
Powerful IDE, free for students, open-source contributors, and individuals  
[Release notes](#)

Modify  
Launch  
More ▾

Modify  
Launch  
More ▾

**Developer News**

[Visual Studio 2022 Launch videos available on-demand](#)  
Visual Studio 2022 Launch videos are now available...  
2021년 11월 9일 화요일

[Visual Studio 2022 is now available](#)  
We've reached general availability for Visual Studi...  
2021년 11월 8일 월요일

[Announcing .NET 6 – The Fastest .NET Yet](#)  
.NET 6 is now available. It is easier to use, runs fas...  
2021년 11월 8일 월요일

[View more Microsoft developer news...](#)

Need help? Check out the [Microsoft Developer Community](#) or reach us via [Visual Studio Support](#).

Installer version 3.1.2196.8931

# Download Unity Hub

Start Your Creative Projects and : x +

unity.com/download

Unity Products Solutions Learning More

Get started

DOWNLOAD UNITY

## Unlock your creativity

Download the world's most popular development platform for creating 2D and 3D multiplatform games and interactive experiences.

Download for Windows

Download other versions

### Create with Unity in three steps

- 1. Download the Unity Hub**

Follow the instructions onscreen for guidance through the installation process and setup.

[Download for Windows](#)  
[Download for Mac](#)  
[Instructions for Linux](#)
- 2. Choose your Unity version**

Install the latest version of Unity, an older release, or a beta featuring the latest in-development features.

[Visit the download archive](#)
- 3. Start your project**

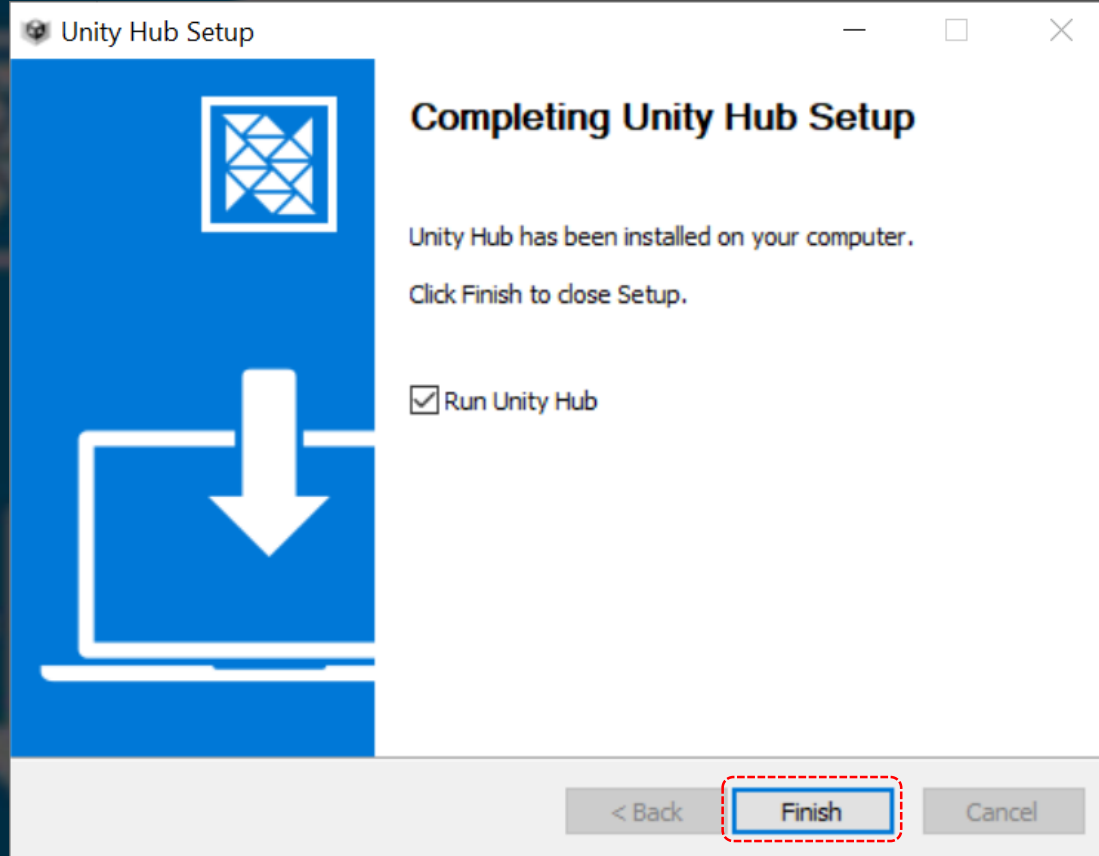
Begin creating from scratch, or pick a template to get your first project up and running quickly. Access tutorial videos designed to support creators, from beginners to experts.

이미지 출처 : <https://unity.com/download>

# Install Unity Hub

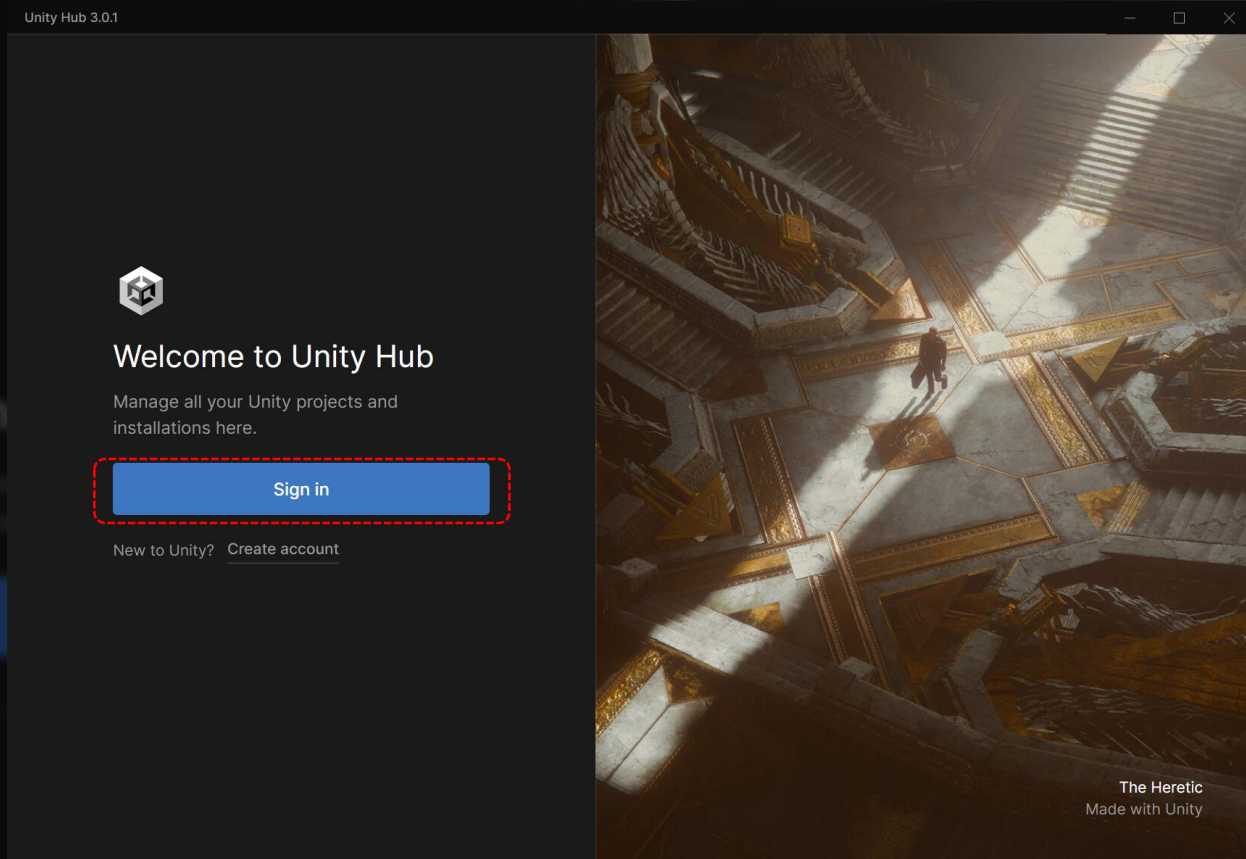


# Run Unity Hub



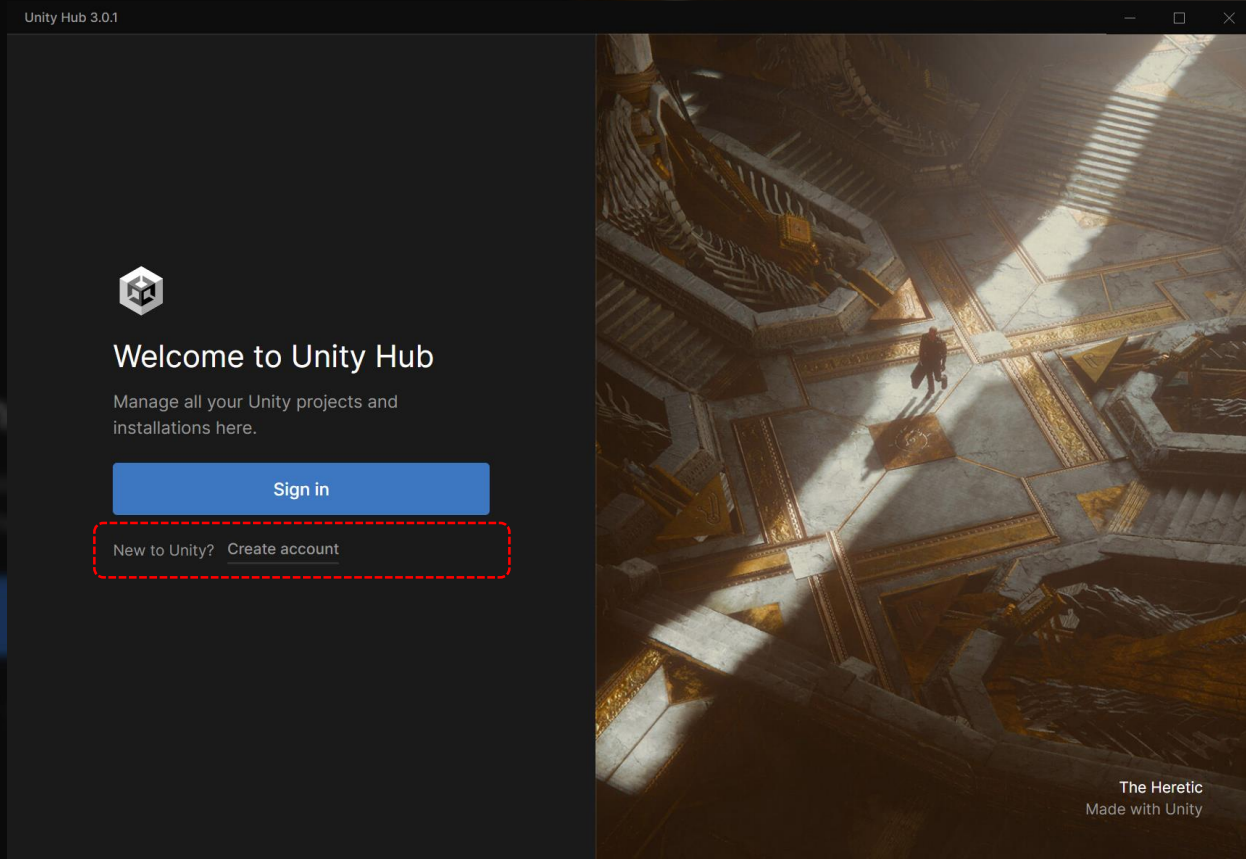


# Sign in (or Create account)





# Sign in (or Create account)



# Install Unity 2020.3.30.f1

Unity Hub 3.0.1

KP

## Installs

Locate Install Editor

Projects

Installs



Learn

Community


### Install Unity Editor

Official releases Pre-releases Archive

LONG TERM SUPPORT (LTS)

 <b>2020.3.30f1</b> LTS Recommended version	<b>Install</b>
 <b>2019.4.36f1</b> LTS	<b>Install</b>

OTHER VERSIONS

 <b>2021.2.14f1</b>	<b>Install</b>
--	----------------

[Beta program webpage](#)

Downloads

# Install Unity 2020.3.30.f1

Unity Hub 3.0.1

KP Installs Locate Install Editor

Projects Installs Learn Community

### Install Unity 2020.3.30.f1 LTS

Add modules Required: 7.18 GB Available: 1.47 TB

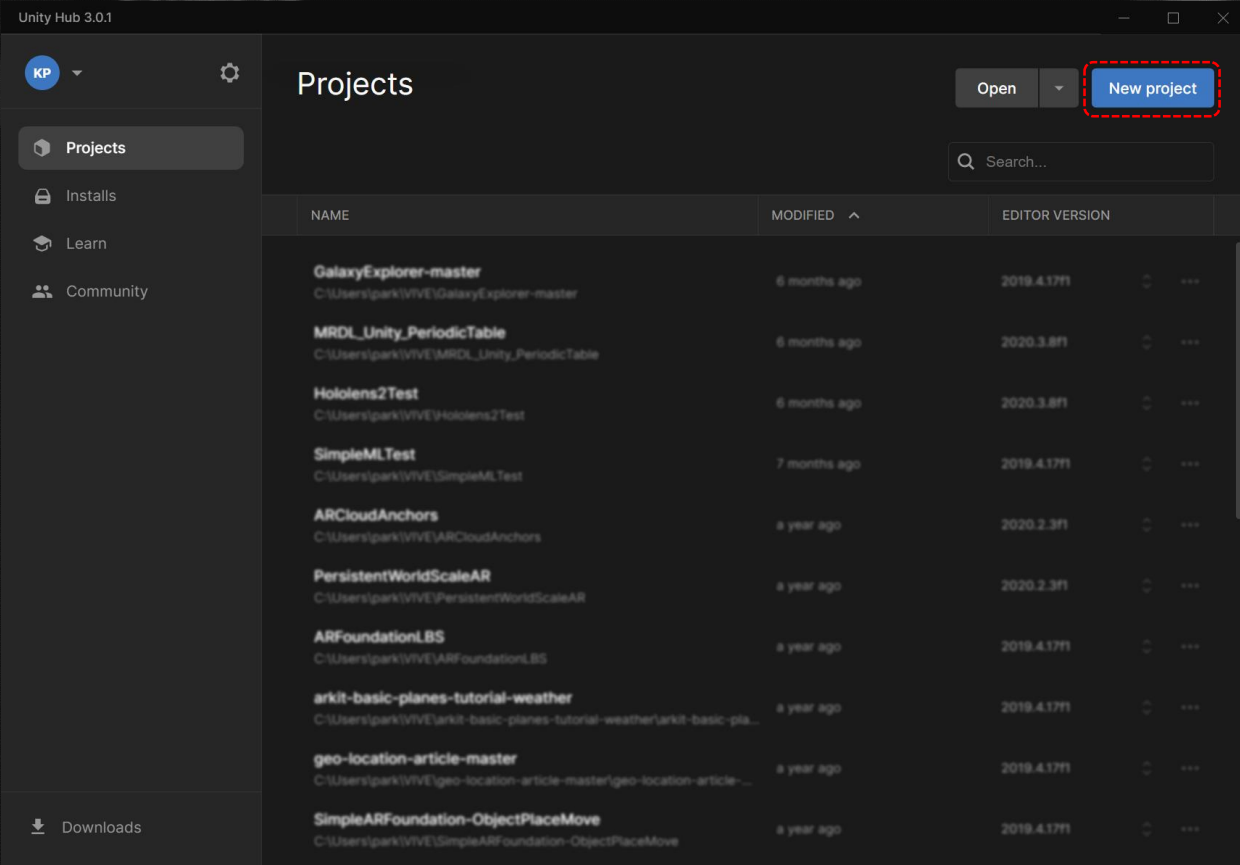
<input type="checkbox"/>	tvOS Build Support	374.12 MB	1.71 GB
<input type="checkbox"/>	Linux Build Support (IL2CPP)	98.22 MB	408.64 MB
<input type="checkbox"/>	Linux Build Support (Mono)	97.63 MB	402.36 MB
<input type="checkbox"/>	Mac Build Support (Mono)	305.1 MB	1.66 GB
<input checked="" type="checkbox"/>	Universal Windows Platform Build Support	260.91 MB	1.88 GB
<input type="checkbox"/>	WebGL Build Support	304.31 MB	1.07 GB
<input type="checkbox"/>	Windows Build Support (IL2CPP)	74.68 MB	355.78 MB
<input type="checkbox"/>	Lumin OS (Magic Leap) Build Support	153.6 MB	832.8 MB
▼	DOCUMENTATION	DOWNLOAD SIZE	SIZE ON DISK
<input checked="" type="checkbox"/>	Documentation	272.51 MB	555.92 MB

Back Install

Downloads

이미지 출처 :Unity

# Begin a Microgame



이미지 출처 :Unity

# Begin a Microgame

Unity Hub 3.0.1

New project  
Editor Version: 2020.3.30f1 LTS

All templates  
Core  
Sample  
**Learning**

**Learning Templates**  
Designed to be your first project, each Learning Template guides you through the basics of Unity Editor.

Search Learning templates

- LEGO® Microgame**  
Learning
- 2D Platformer Microgame**  
Learning
- Karting Microgame**  
Learning
- FPS Microgame**  
Learning

**FPS Microgame**  
Customize this first-person shooter game while learning the basics of Unity Editor. Preloaded with scenes, scripts, tutorials, and more.  
[Read more](#)

**PROJECT SETTINGS**

Project name  
My project

Location  
C:\Users\park\GP22

Cancel **Create project**

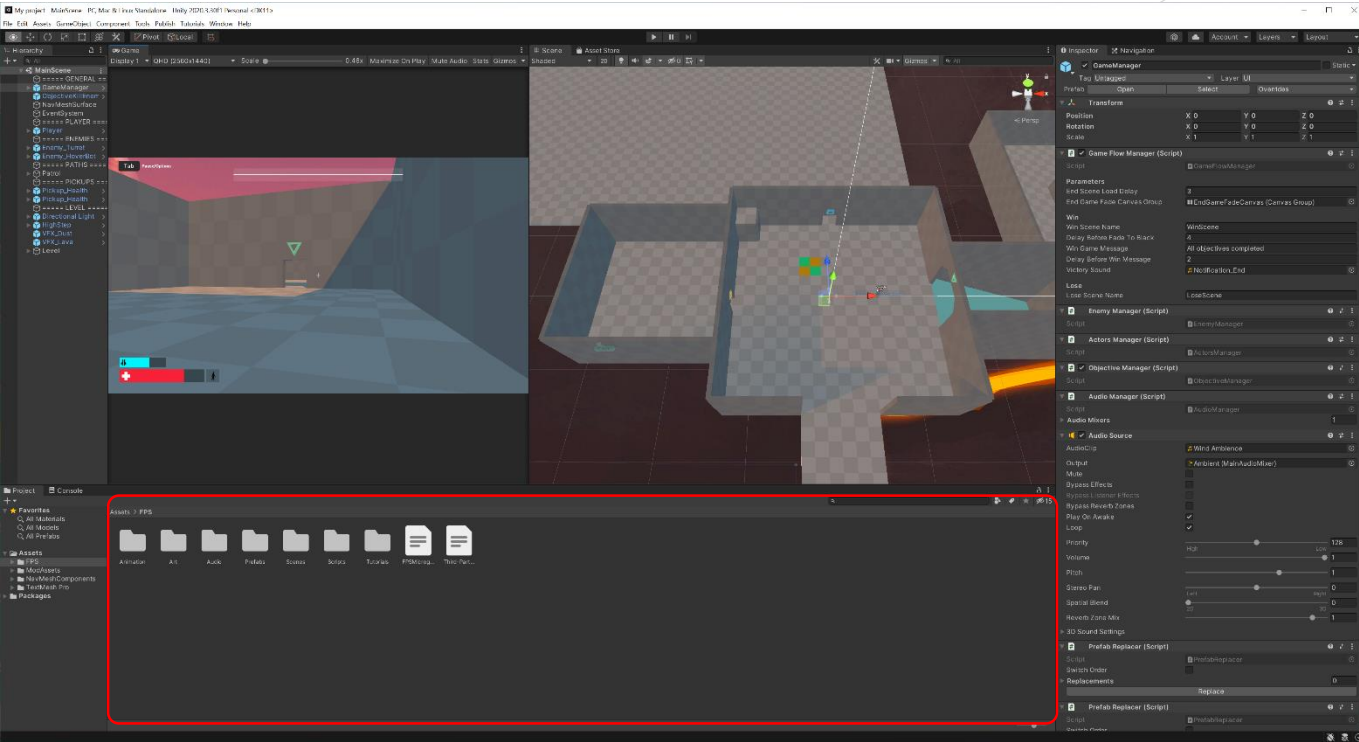


4

## Unity Interface



# Unity Interface

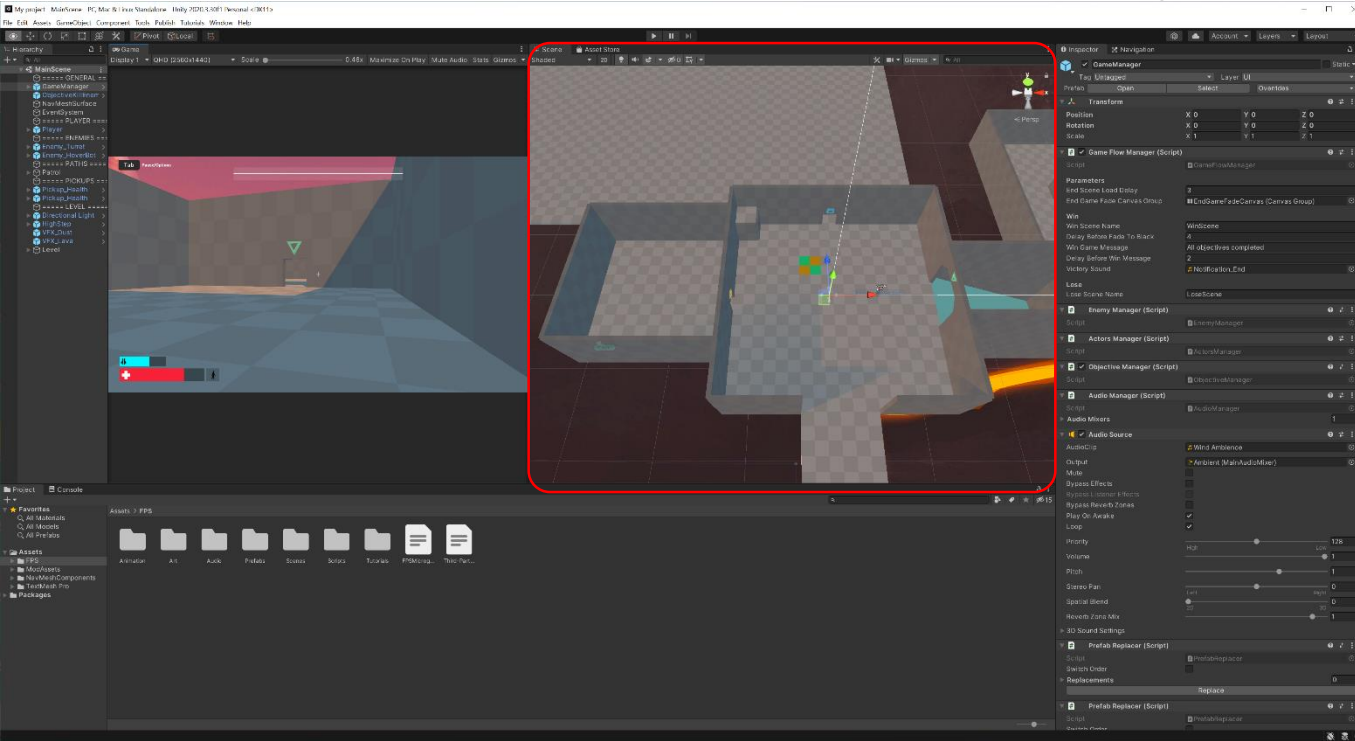


Project



이미지 출처 :Unity

# Unity Interface



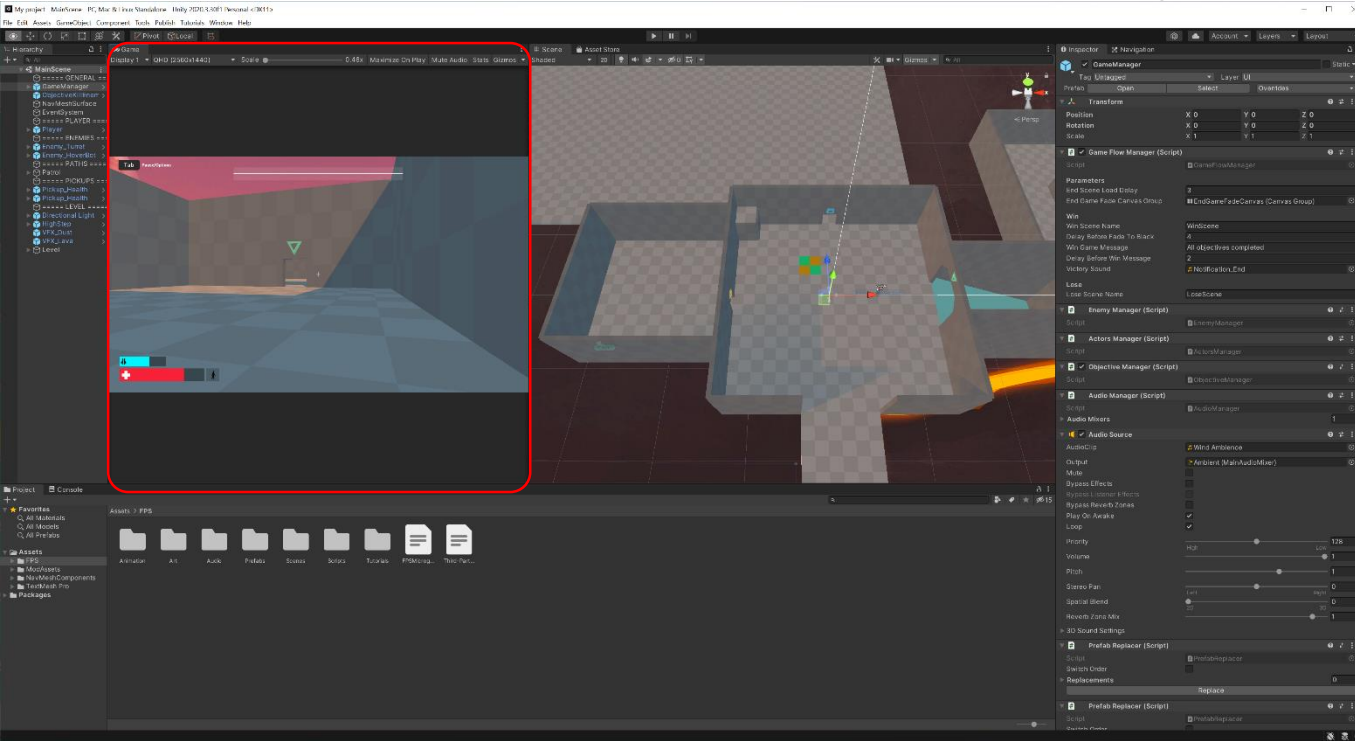
➤ Project

➤ Scene View





# Unity Interface



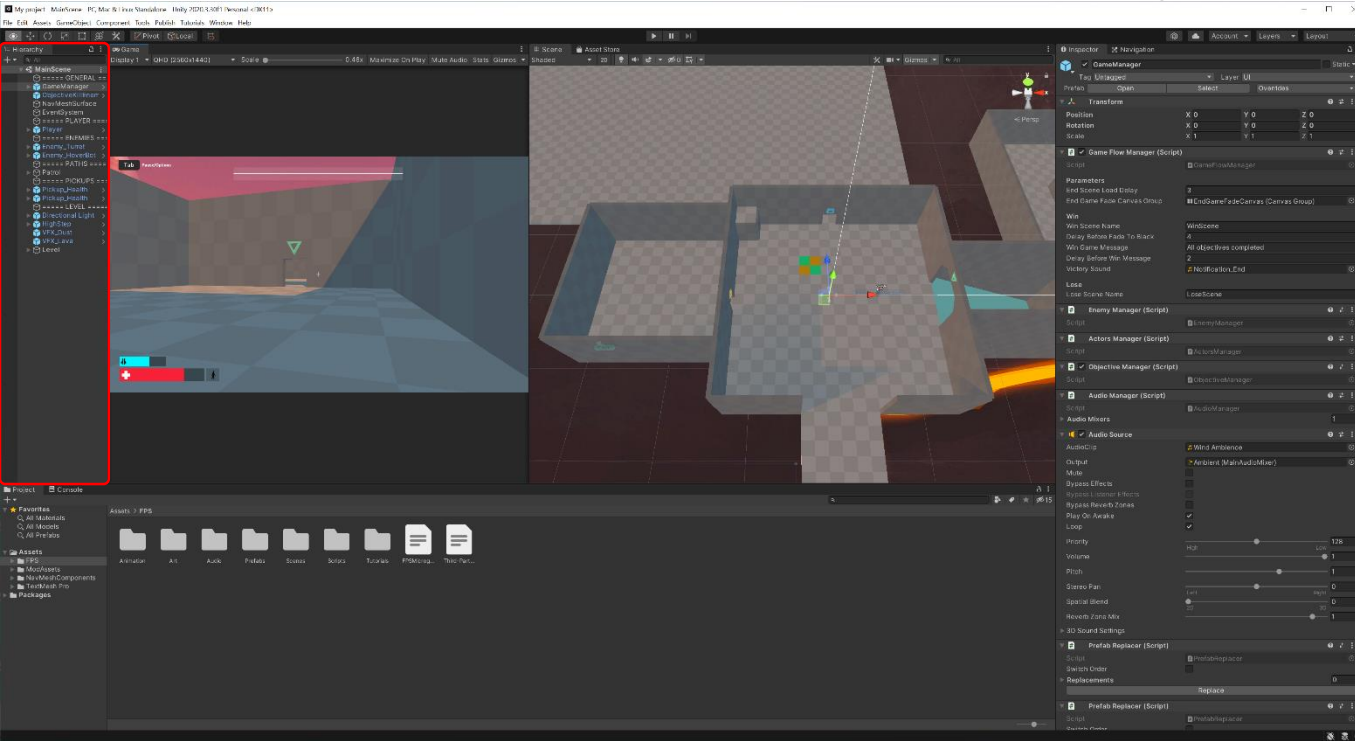
» Project

» Scene View

» Game View



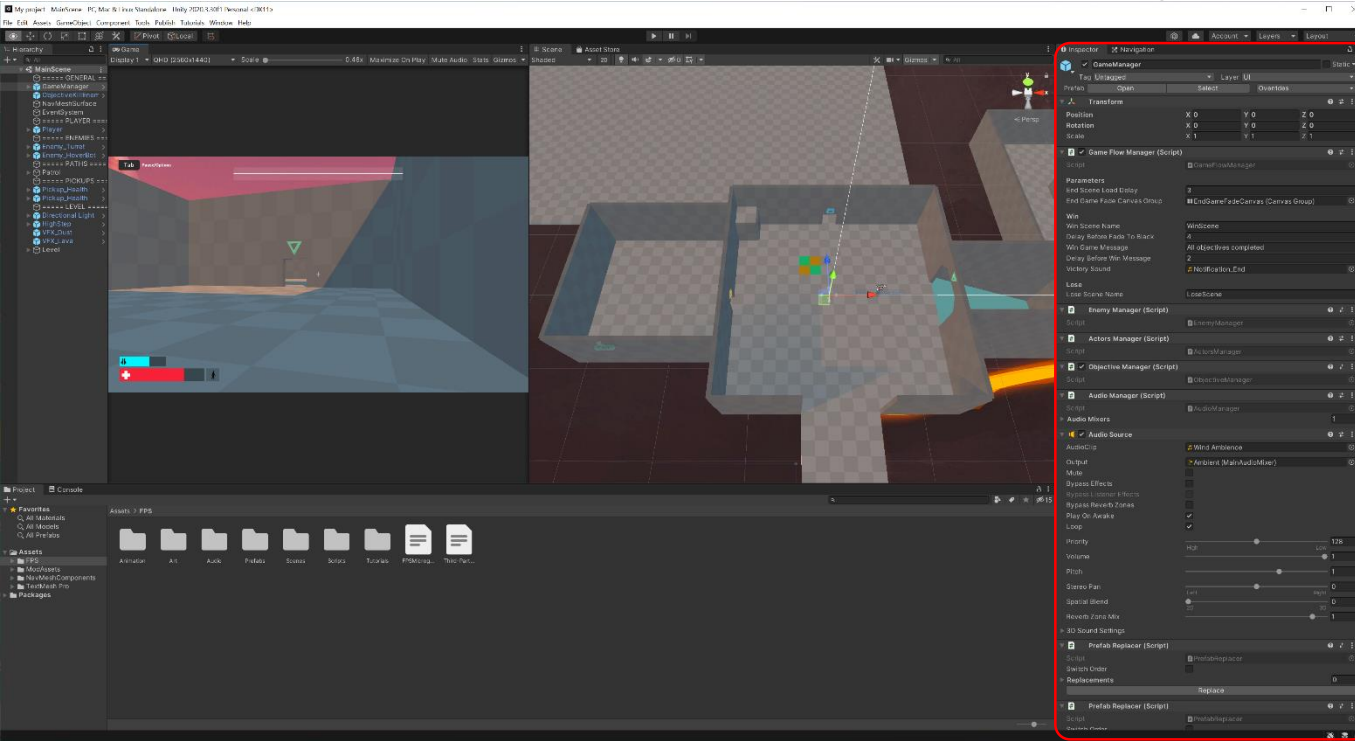
# Unity Interface



- ▶ Project
- ▶ Scene View
- ▶ Game View
- ▶ Hierarchy



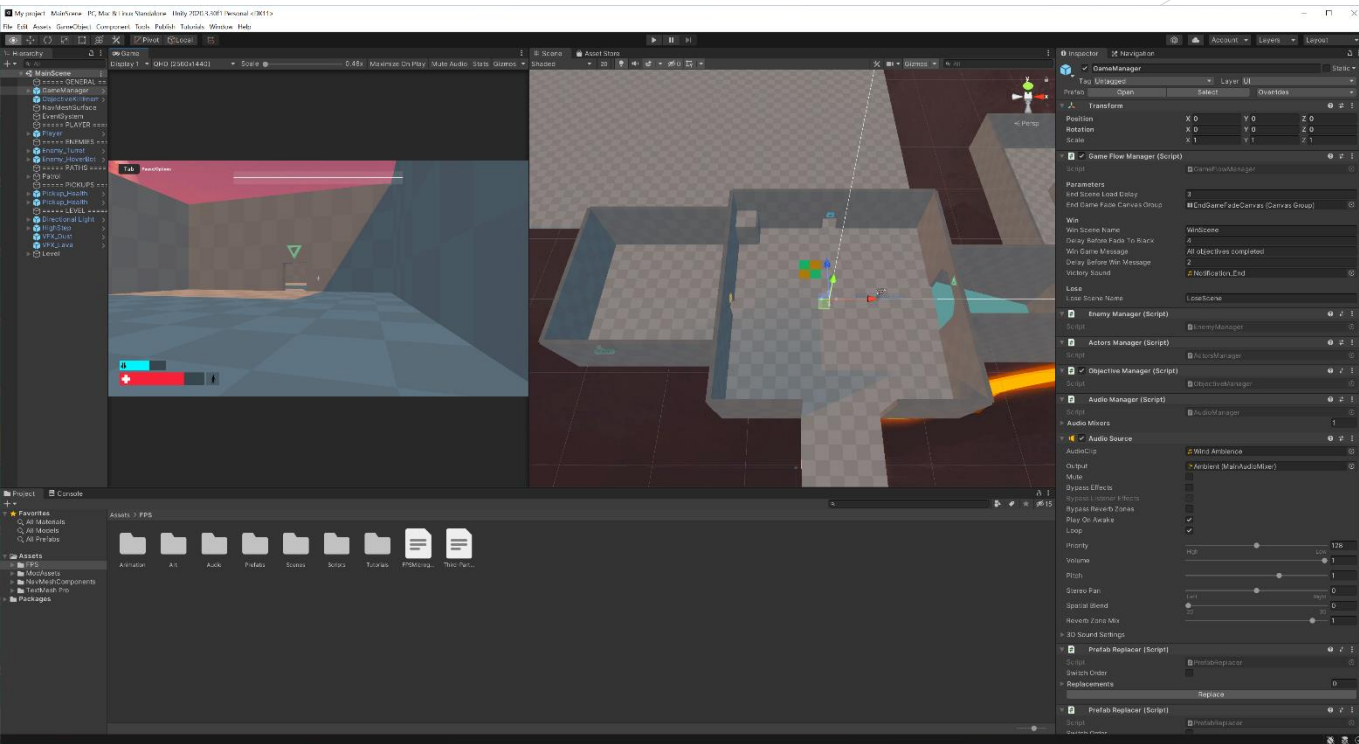
# Unity Interface



- » Project
- » Scene View
- » Game View
- » Hierarchy
- » Inspector



# Unity Interface

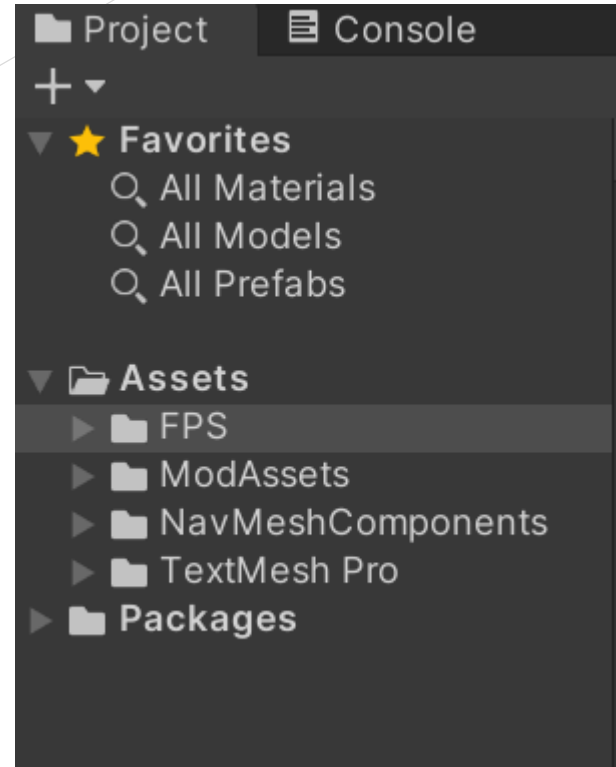


- » Project
- » Scene View
- » Game View
- » Hierarchy
- » Inspector
- » Console



## Project Window

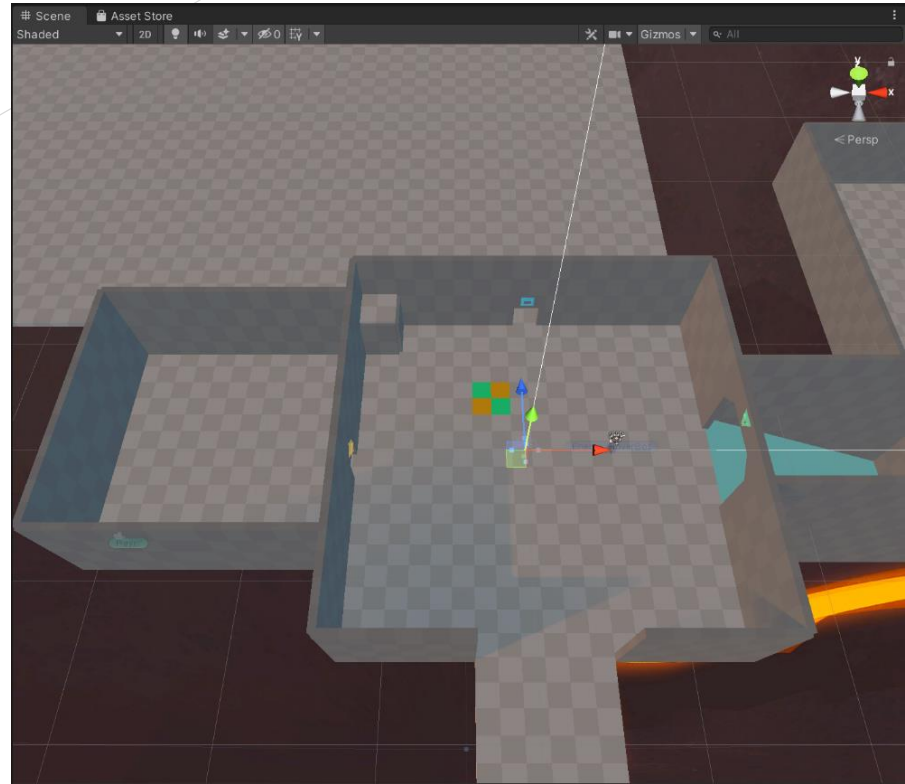
- » The project window contains all of the assets that are available for you to use.
- » Typically, these are organized into folders, for example, according to the asset type (models, materials, audio, prefabs, scripts, etc.).



이미지 출처 :Unity

## Scene View

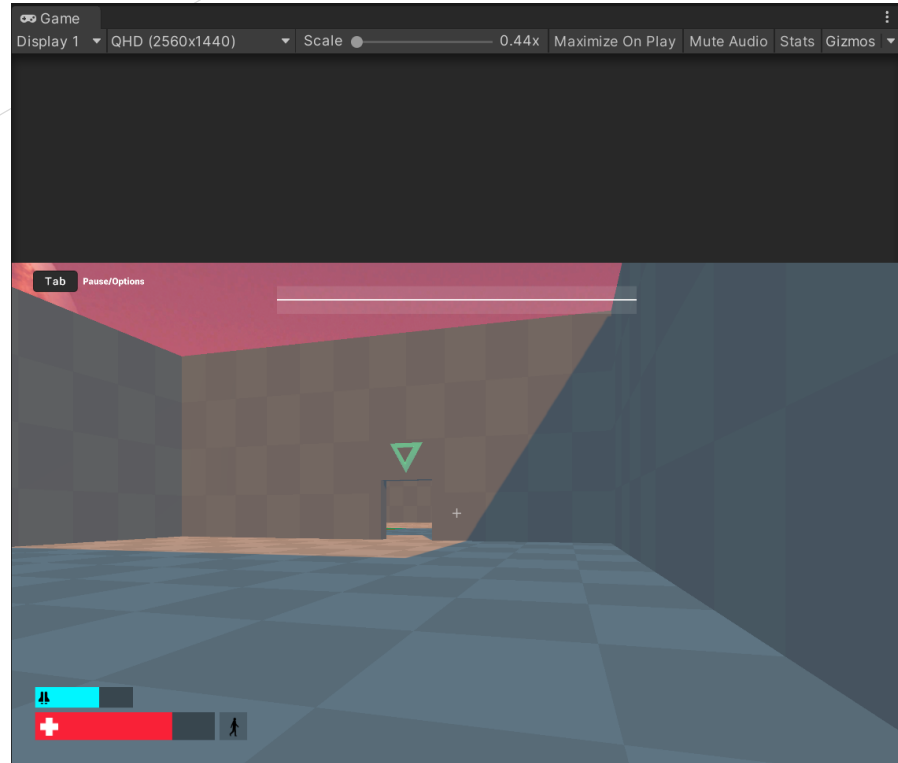
- » This scene window shows all the elements of the current scene.
- » Most editing of the scene is done through the scene view, because it provides access to low-level and hidden aspects of the objects.



이미지 출처 :Unity

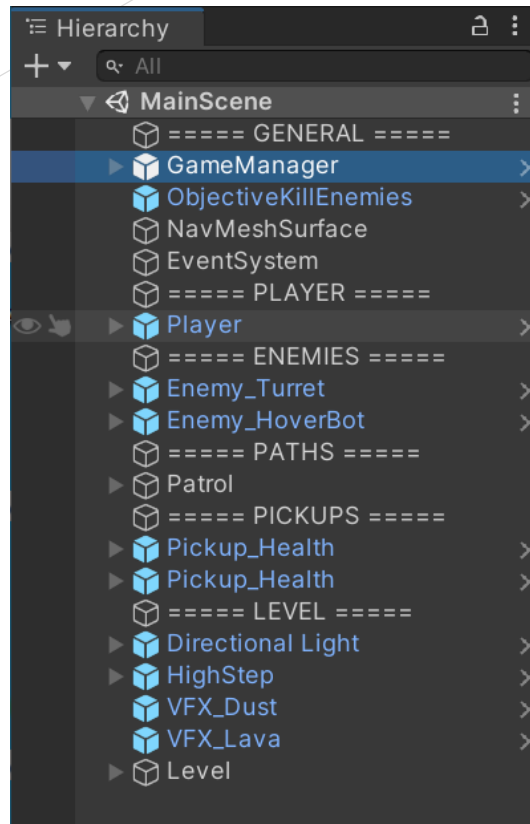
# Game View

- » This game window shows the elements of the scene as they would appear to the player.

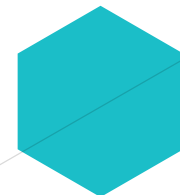


# Hierarchy

- » This window shows all the game objects that constitute the current scene.
- » Game objects are stored hierarchically in a tree structure.



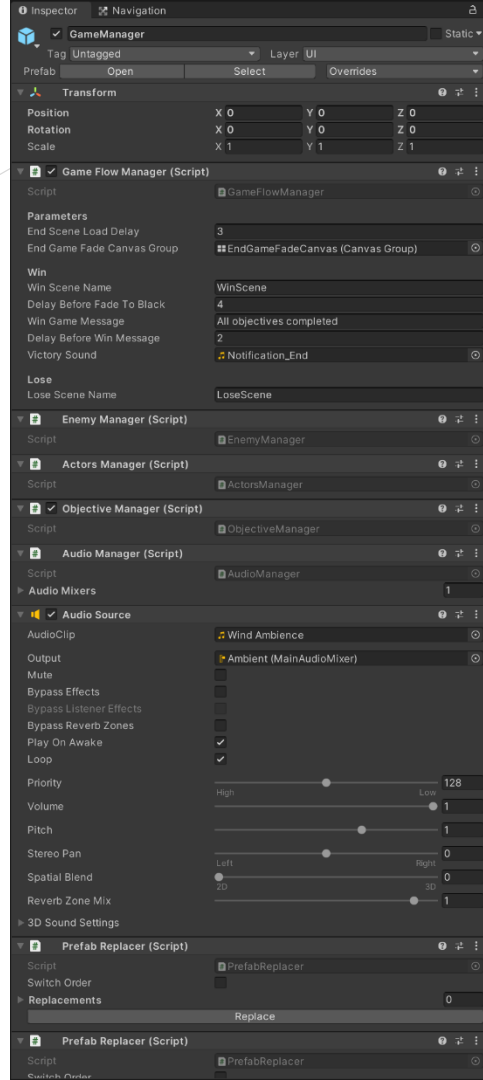
이미지 출처 : Unity





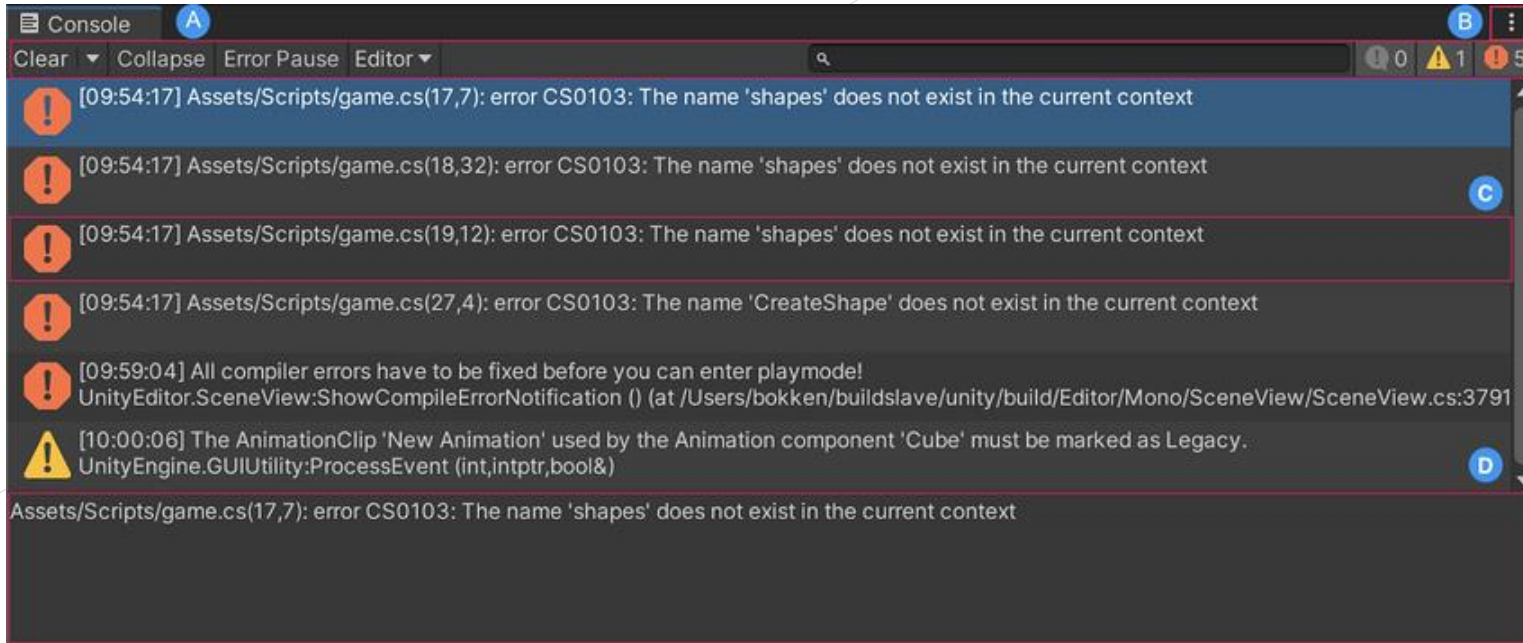
# Inspector

- » At any time there is an active game object (which the designer selects by clicking on the object or on its entry in the hierarchy).
- » This window provides all the component information associated with this object.



# Console

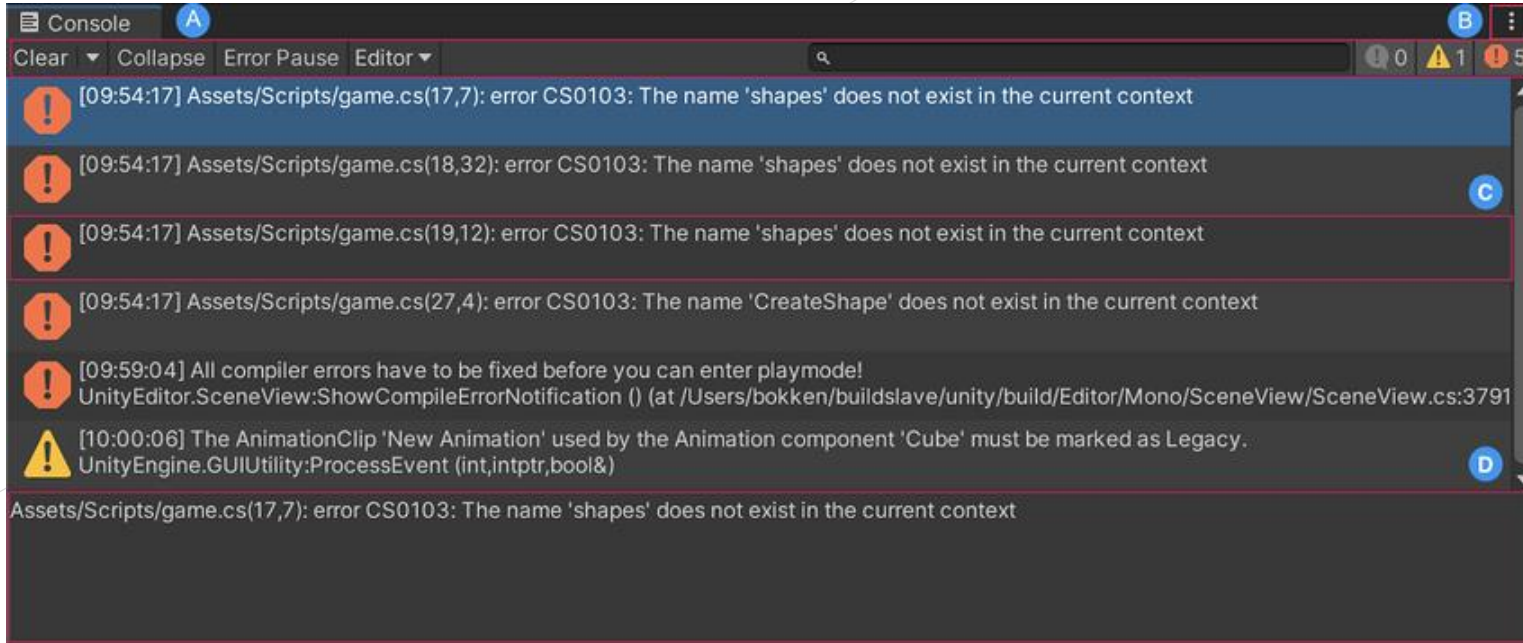
- » The console window displays errors, warnings, and other messages the Editor generates.



이미지 출처 :Unity

# Console

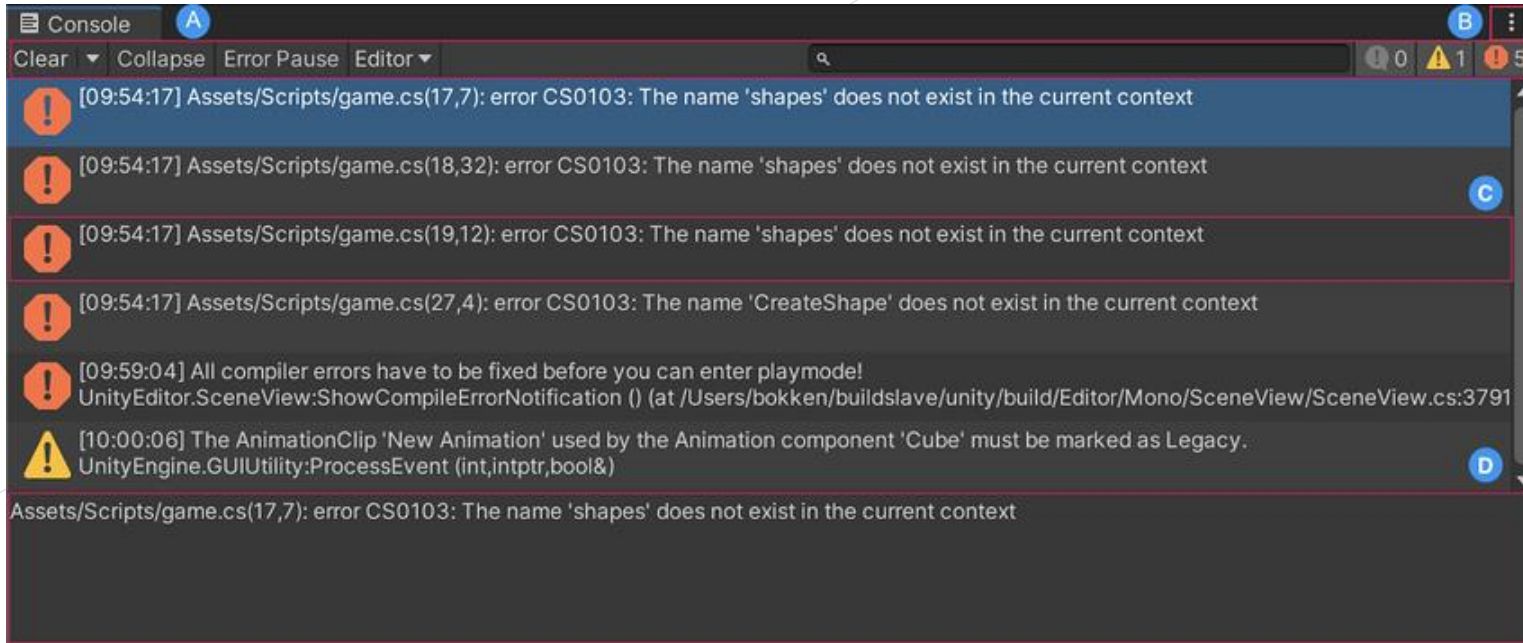
- » These errors and warnings help you find issues in your project, such as script compilation errors.



이미지 출처 :Unity

# Console

- » They also alert you to actions the Editor has taken automatically, such as replacing missing meta files, which could cause an issue somewhere else in your project



이미지 출처 :Unity

5

## Unity Basics

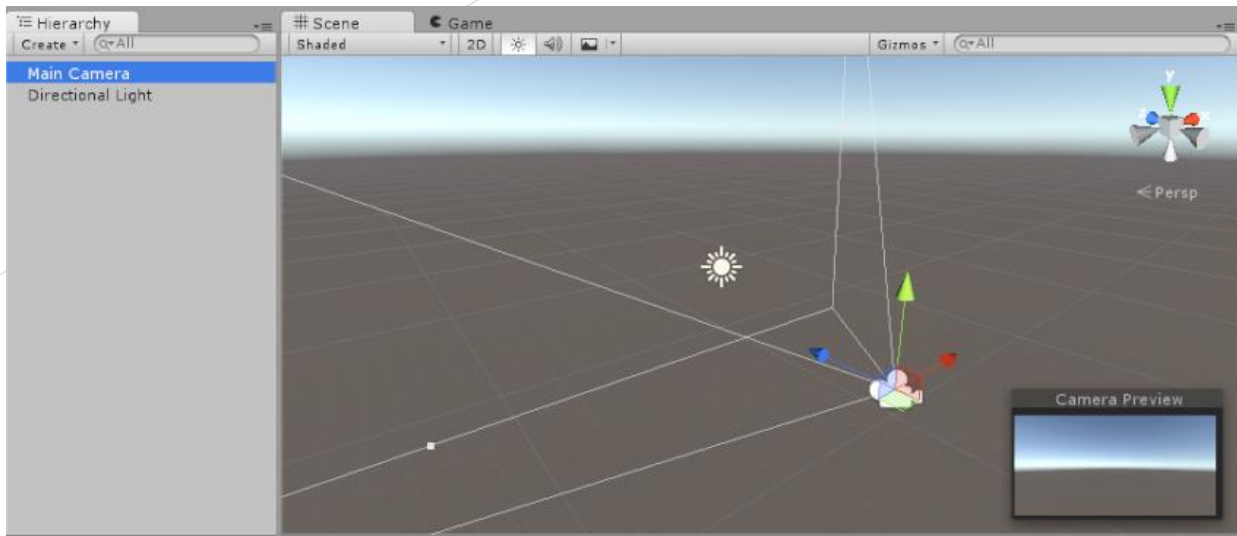


PROGRAMMING

# Scenes

## » Scenes

- ▶ A scene contains a collection of game objects that constitute the world that the player sees at any time.
- ▶ Below example shows a sample scene that contains only a Camera and a Light.



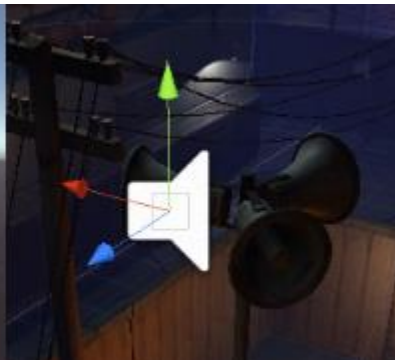
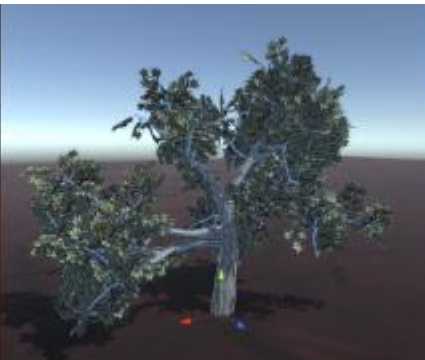
이미지 출처 :Unity



# GameObjects

## » GameObjects

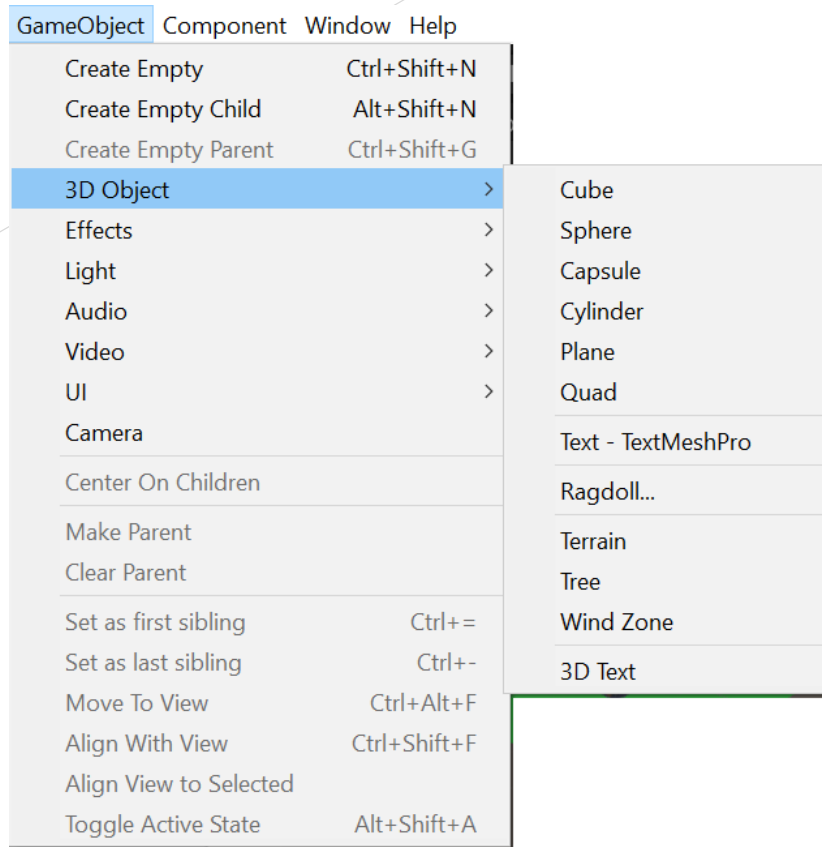
- ▶ The game objects are all the things that constitute your scene.
- ▶ GameObjects are the fundamental objects in Unity that represent characters, props and scenery. They do not accomplish much in themselves but they act as containers for Components, which implement the functionality.
- ▶ Four different types of GameObject  
: an animated character, a light, a tree, and an audio source



# GameObjects

## » GameObjects

- ▶ Empty
- ▶ 3D Object - Cube, Sphere,
- ▶ Light - Directional Light,
- ▶ Audio
- ▶ Video
- ▶ Effect - Particle System
- ▶ UI
- ▶ Camera





# Components

## » Components

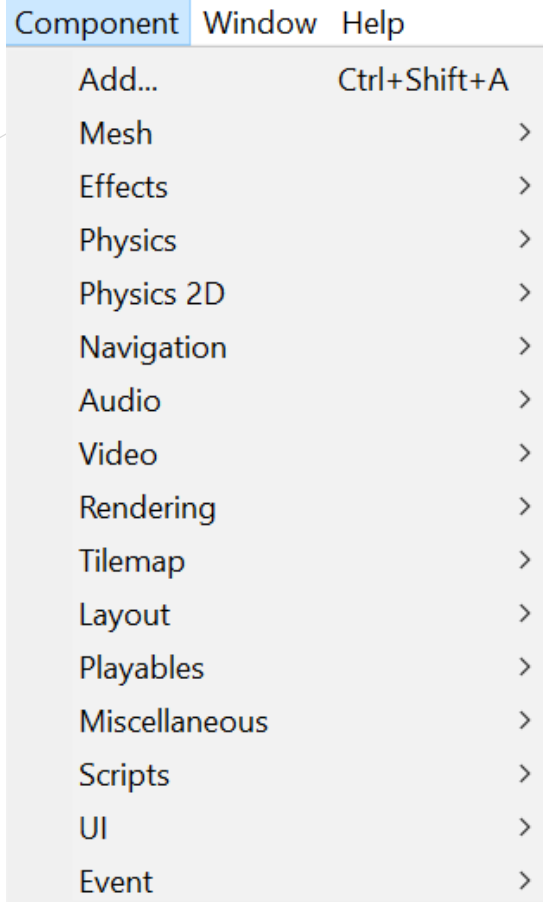
- Each GameObject is defined by a collection of associated elements, called Components.
- The set of components that are associated with a game object depend on the nature of object. For example, a light source object is associated with color and intensity of the light source. A camera object is associated with various properties of how the projection is computed (wide-angle or telephoto).
- The various components that are associated with a game object can be viewed and edited in the Inspector window.



# Components

## » Components

- ▶ Mesh
- ▶ Effects
- ▶ Physics
- ▶ Physics2D
- ▶ Navigation
- ▶ Audio
- ▶ Video
- ▶ Rendering
- ▶ Tilemap
- ▶ Layout
- ▶ Playables
- ▶ Miscellaneous
- ▶ Scripts
- ▶ UI
- ▶ Event



이미지 출처 :Unity

# Prefabs

## » Prefabs

- ▶ Unity's Prefab system allows you to create, configure, and store a GameObject complete with all its components, property values, and child GameObjects as a reusable Asset.
- ▶ The Prefab Asset acts as a template from which you can create new Prefab instances in the Scene.



# Assets

## » Assets

- ▶ An asset is any resource that will be used as part of an object's component.
- ▶ Examples include meshes (for defining the shapes of objects), materials (for defining shapes), physics materials (for defining physical properties like friction), and scripts (for defining behaviors).



# Scripts

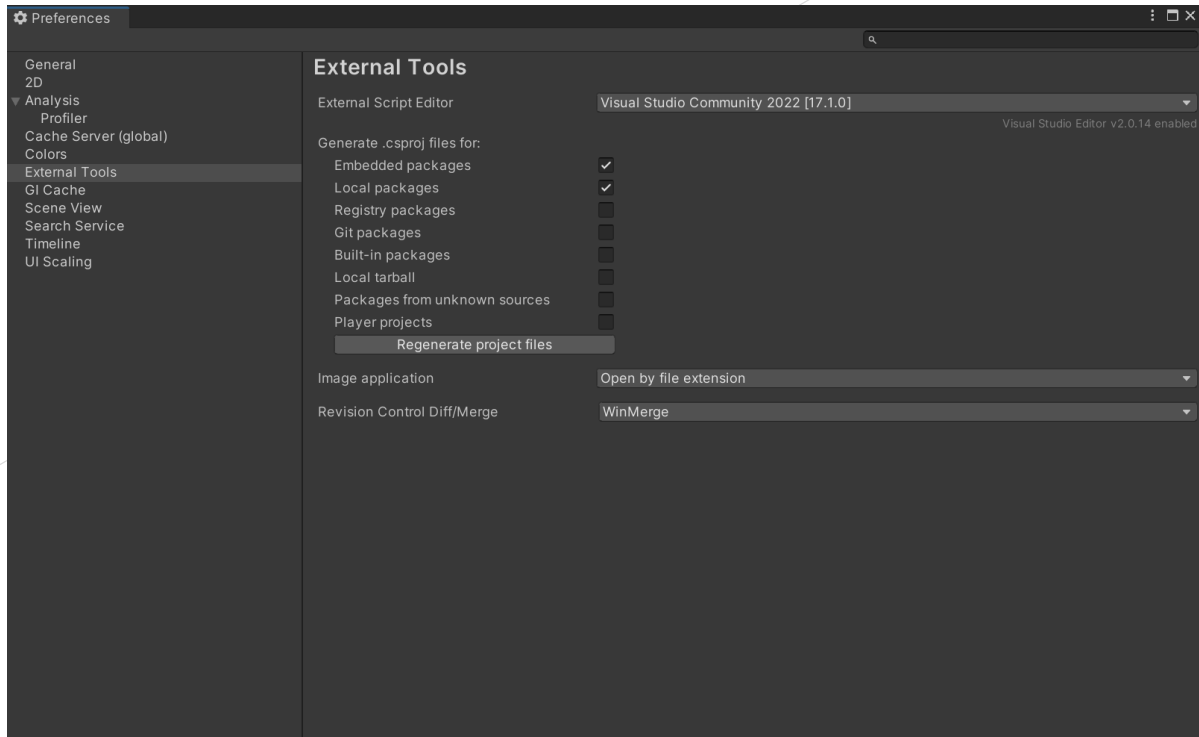
## » Scripts

- ▶ A script is a chunk of code that defines the behavior of game objects. Scripts are associated with game objects.
- ▶ There are various types of scripts classes, depending on the type of behavior being controlled.
- ▶ Because interactive game programming is event-driven, a typical script is composed as a collection of functions, each of which is invoked in response to a particular event. (e.g., A function may be invoked when this object collides with another object.)
- ▶ Typically, each of these functions performs some simple action (e.g., moving the game object, creating/destroying game objects, triggering events for other game objects), and then returns control to the system.



# IDE (Visual Studio 2022) for Script

Edit	
Undo	Ctrl+Z
Redo	Ctrl+Y
Select All	Ctrl+A
Deselect All	Shift+D
Select Children	Shift+C
Select Prefab Root	Ctrl+Shift+R
Invert Selection	Ctrl+I
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Paste As Child	Ctrl+Shift+V
Duplicate	Ctrl+D
Rename	
Delete	
Frame Selected	F
Lock View to Selected	Shift+F
Find	Ctrl+F
Play	Ctrl+P
Pause	Ctrl+Shift+P
Step	Ctrl+Alt+P
Sign in...	
Sign out	
Selection	>
Project Settings...	
Preferences...	
Shortcuts...	
Clear All PlayerPrefs	
Graphics Tier	>
Grid and Snap Settings...	



이미지 출처 :Unity



# Packages

## » Packages

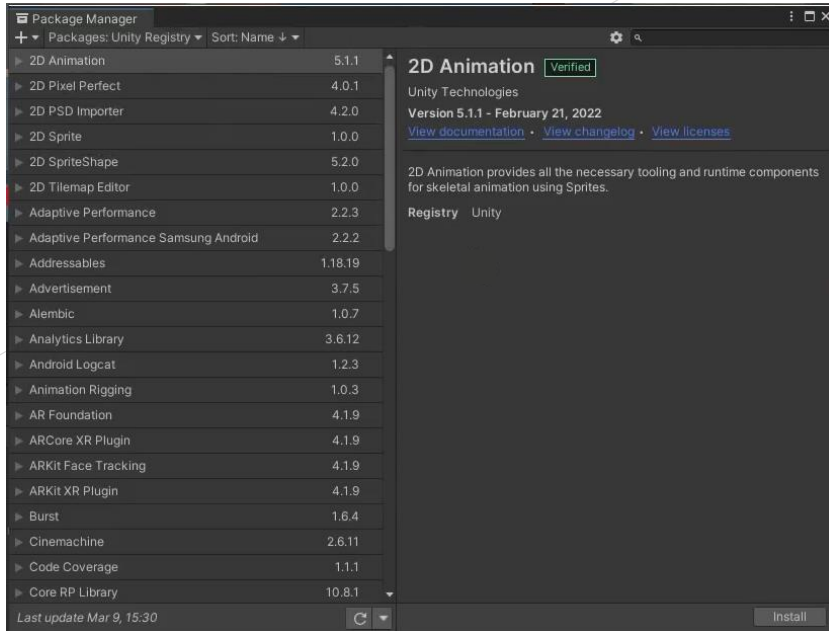
- A package is an aggregation of game objects and their associated meta-data.
- They are related objects (models, scripts, materials, etc.). Here are some examples
  - a collection of shaders for rendering water effects
  - particle systems for creating explosions
  - models of race cars for a racing game
  - models of trees and bushes to create a woodland scene



# Packages

## » Packages

- ▶ Unity provides a number standard packages for free, and when a new project is created, you can select the packages that you would like to have imported into your project.



이미지 출처 :Unity





# Reference

- » <https://www.gameenginebook.com/coursemat.html>
  - Lecture Slides prepared by Roger Mailler, Ph.D., Associate Professor of Computer Science, University of Tulsa
- » Unity Manual
  - <https://docs.unity3d.com/Manual/UnityOverview.html>
- » CMSC425 lecture2
  - <https://www.cs.umd.edu/class/fall2018/cmsc425/Lects/lect02-architecture.pdf>
- » CMSC425 lecture3
  - <https://www.cs.umd.edu/class/fall2018/cmsc425/Lects/lect03-unity.pdf>

