

Get Started with Unity

Fall 2023

9/14/2023

Kyoung Shin Park
Computer Engineering
Dankook University

Game Engine

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.
- ❑ 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

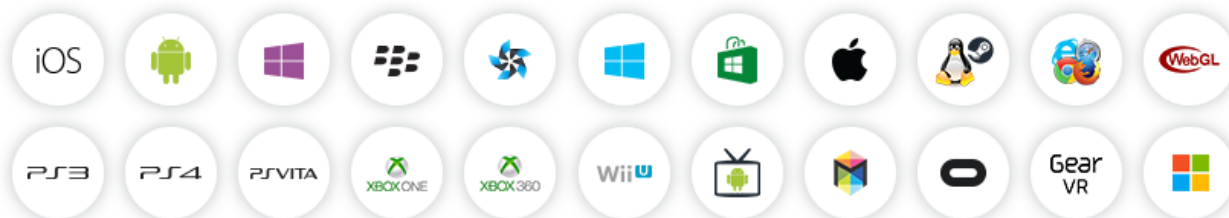
- ❑ GameDev Academy Best Game Engines for 2023
 - <https://gamedevacademy.org/best-game-engines/>
- ❑ Top 10 Game Engines – Ulab - SumDU
 - <https://ulab.sumdu.edu.ua/top-10-game-engines>
- ❑ Examples include
 - Unity
 - Unreal
 - Godot
 - Phaser
 - GameMaker
 - CryEngine
 - AppGameKit
 - RPG Maker
 - Amazon Lumberyard
 -

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/>

 **BUILD ONCE DEPLOY ANYWHERE**




Visual Studio Code

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn [Download](#)

[Version 1.82](#) is now available! Read about the new features and fixes from August.


Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



[Download](#) **Windows**
Windows 10, 11


User Installer	x64	x86	Arm64
System Installer	x64	x86	Arm64
.zip	x64	x86	Arm64
CLI	x64	x86	Arm64



[Download](#) **.deb**
Debian, Ubuntu

[Download](#) **.rpm**
Red Hat, Fedora, SUSE

.deb	x64	Arm32	Arm64
.rpm	x64	Arm32	Arm64
.tar.gz	x64	Arm32	Arm64
Snap	Snap Store		
CLI	x64	Arm32	Arm64



[Download](#) **Mac**
macOS 10.11+

.zip	Intel chip	Apple silicon	Universal
CLI	Intel chip	Apple silicon	

<https://code.visualstudio.com/download>

By downloading and using Visual Studio Code, you agree to the [license terms](#) and [privacy](#)

Download Unity Hub

Start Your Creative Projects and I 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**

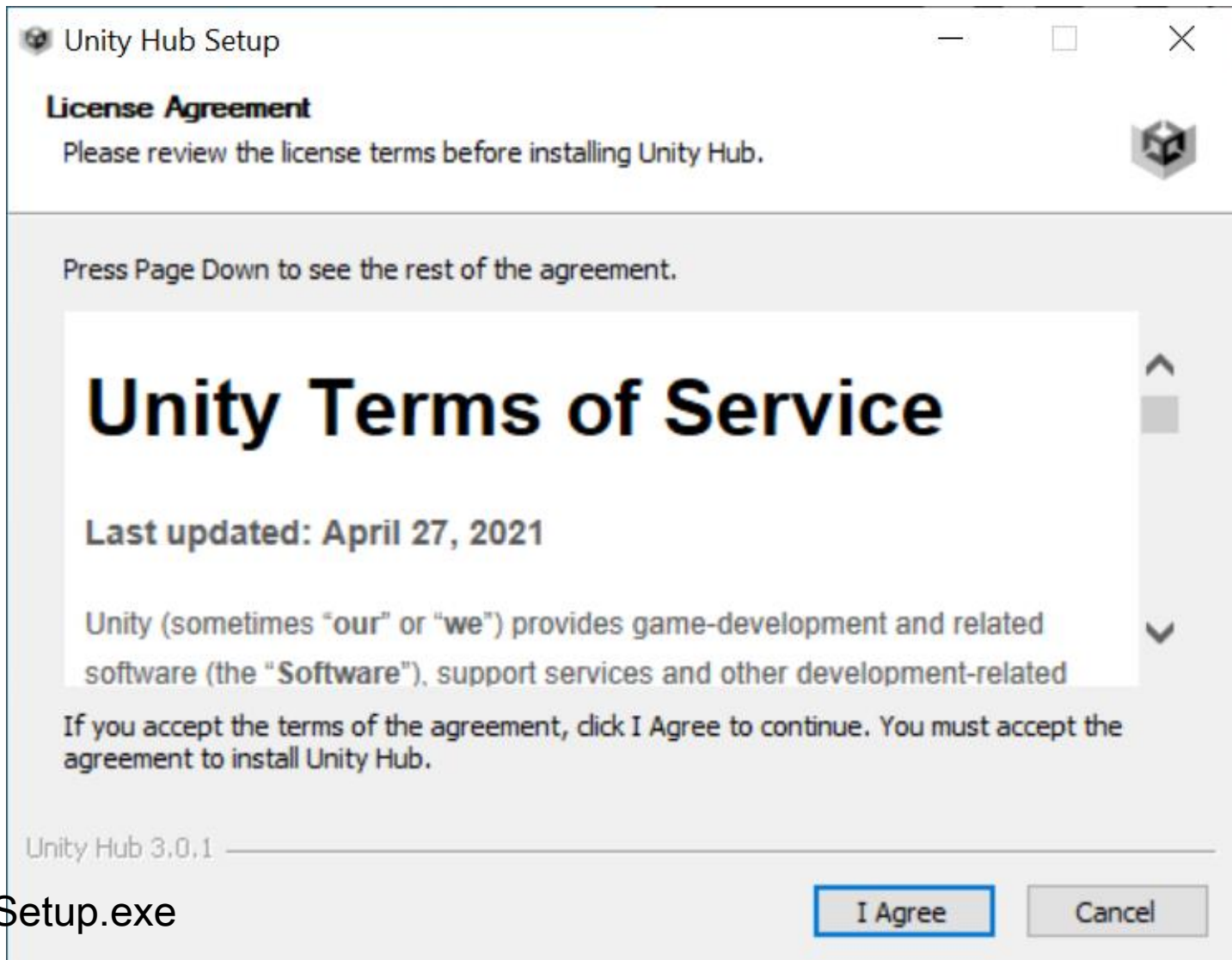
Install the latest version of Unity, an older release, or a beta featuring the latest in-development features.

[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.

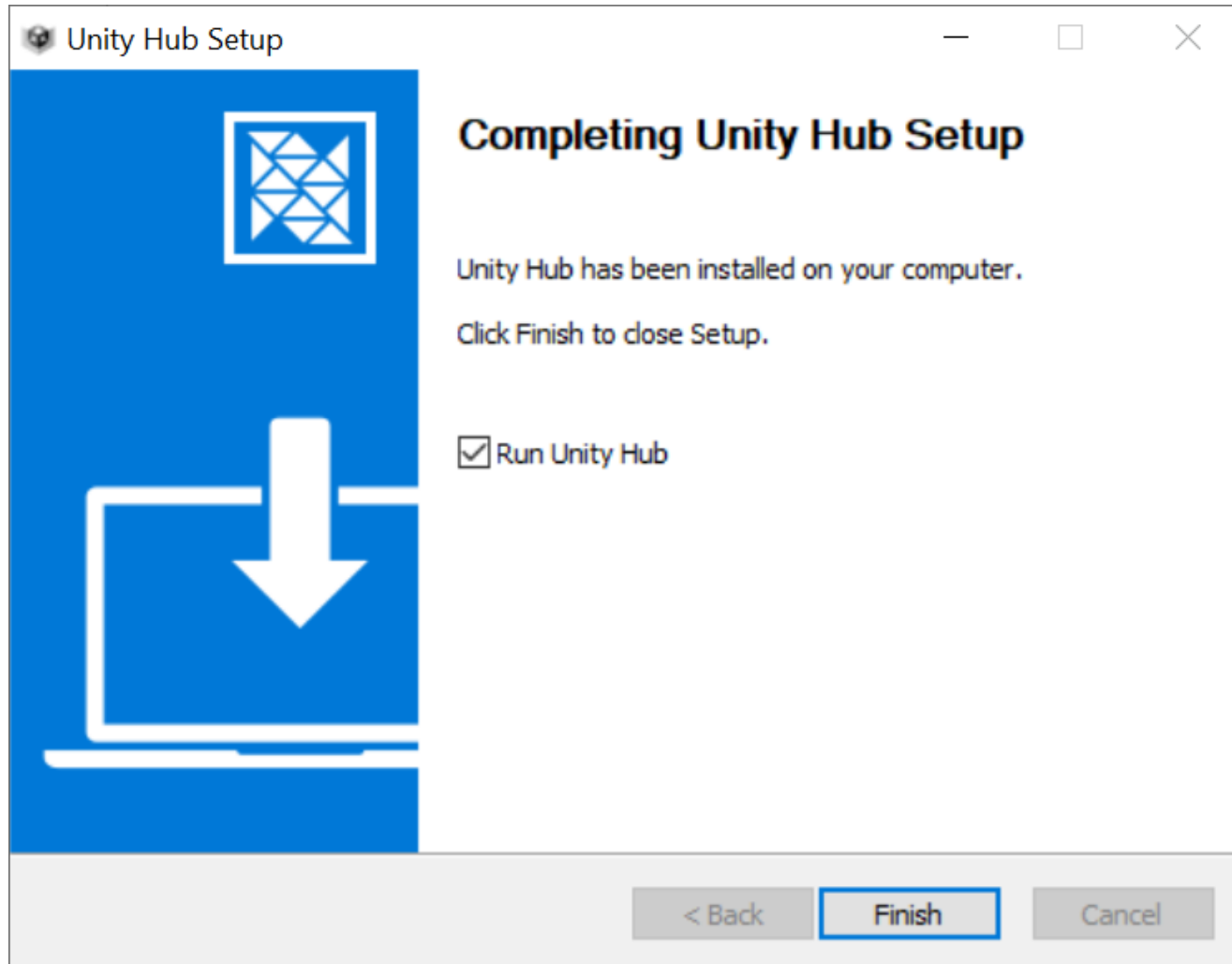
<https://unity.com/download>

Install Unity Hub

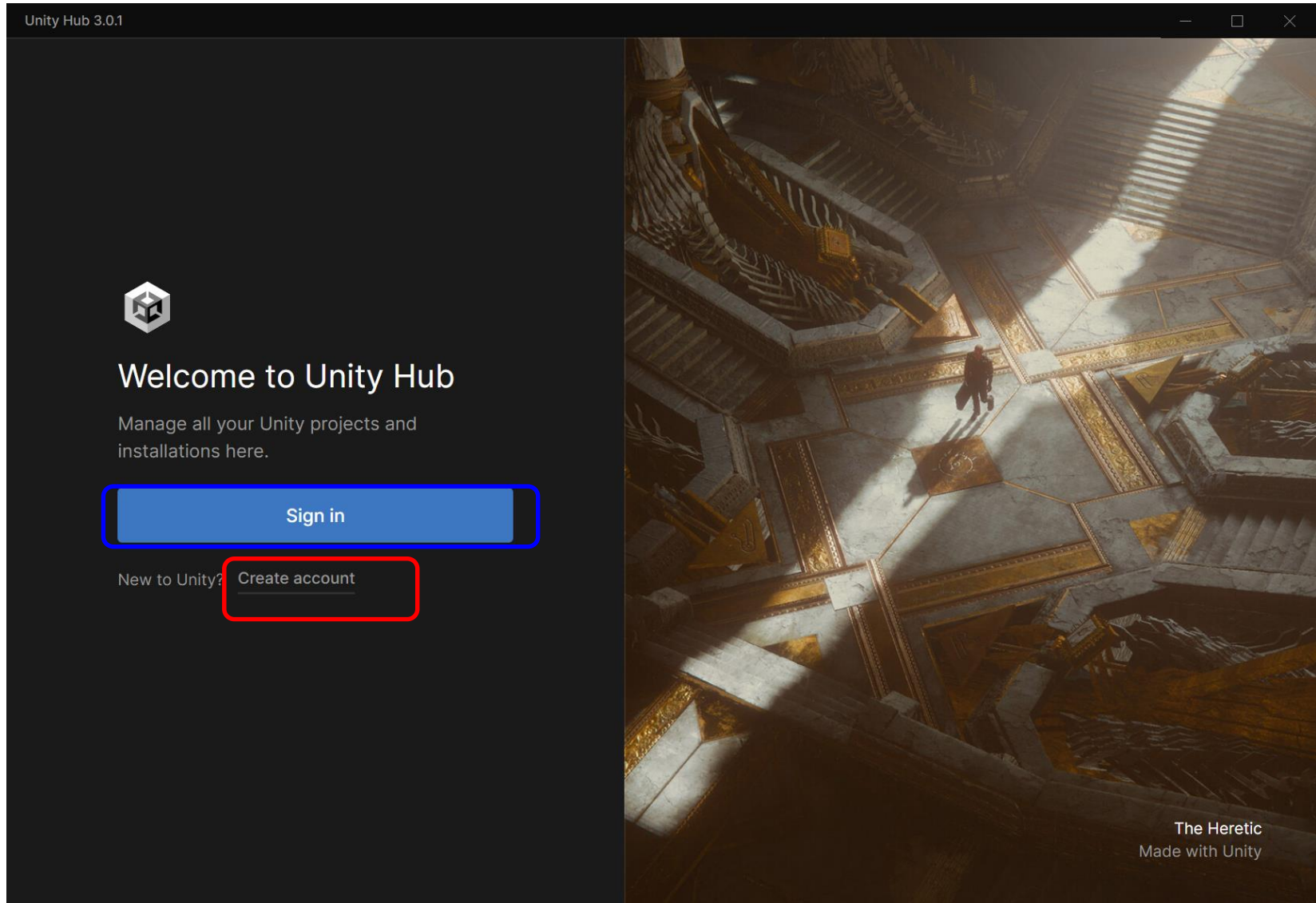


UnityHubSetup.exe

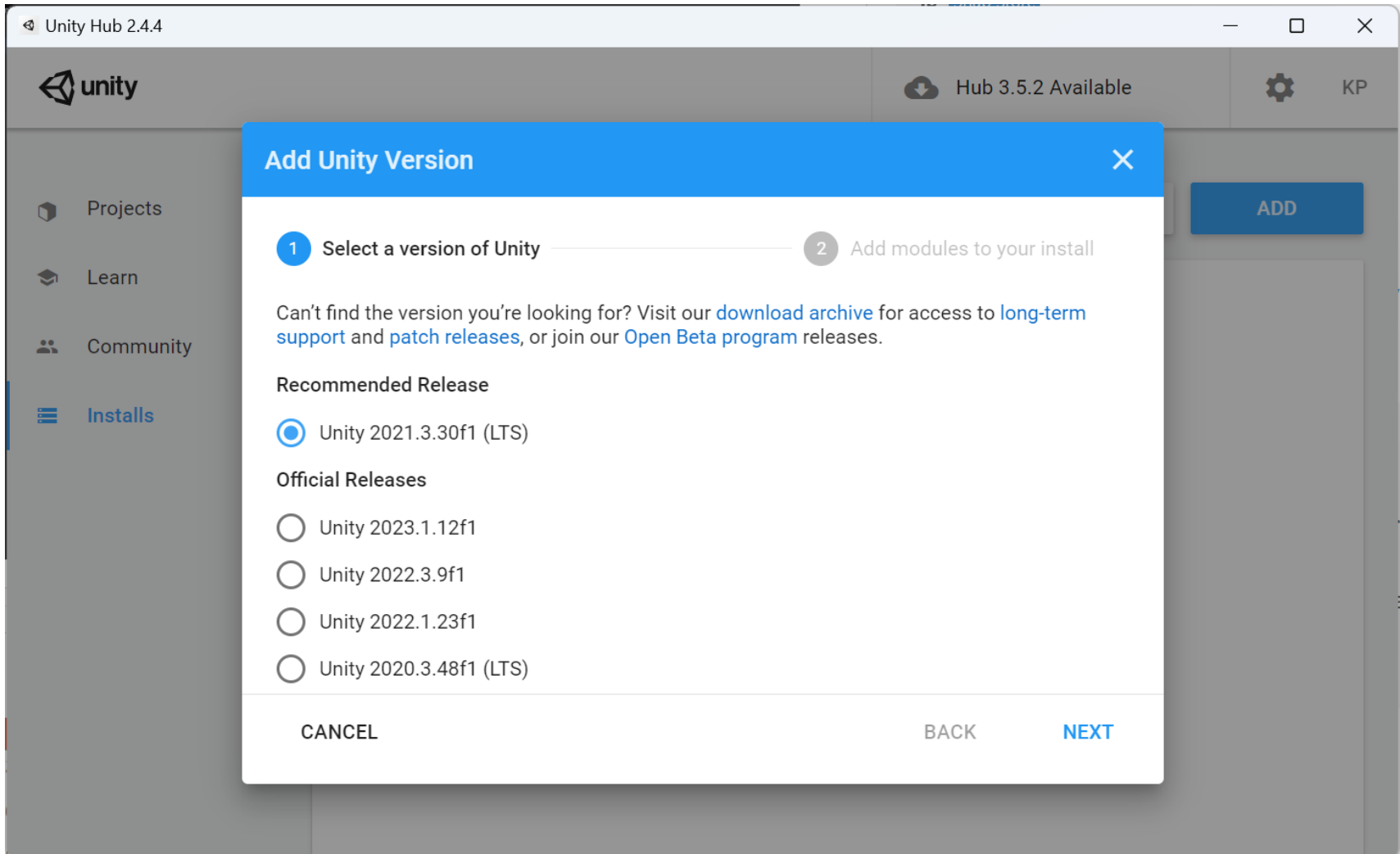
Run Unity Hub



Sign in (or Create account)



Install Unity



Install 2021.3.30f1 (LTS)

Install Unity



Hub 3.5.2 Available



Add Unity Version



<input type="checkbox"/>	Mac Build Support (Mono)	344.3 MB	1.9 GB
<input type="checkbox"/>	Mac Dedicated Server Build Support	342.3 MB	1.9 GB
<input checked="" type="checkbox"/>	Universal Windows Platform Build Support	294.8 MB	2.1 GB
<input type="checkbox"/>	WebGL Build Support	346.1 MB	1.7 GB
<input type="checkbox"/>	Windows Build Support (IL2CPP)	323.8 MB	1.9 GB
<input type="checkbox"/>	Windows Dedicated Server Build Support	643.3 MB	3.8 GB
Documentation			
<input checked="" type="checkbox"/>	Documentation	329.6 MB	672.3 MB
Language packs (Preview)			
<input type="checkbox"/>	日本語	2.4 MB	2.4 MB
<input checked="" type="checkbox"/>	한국어	2.2 MB	2.2 MB

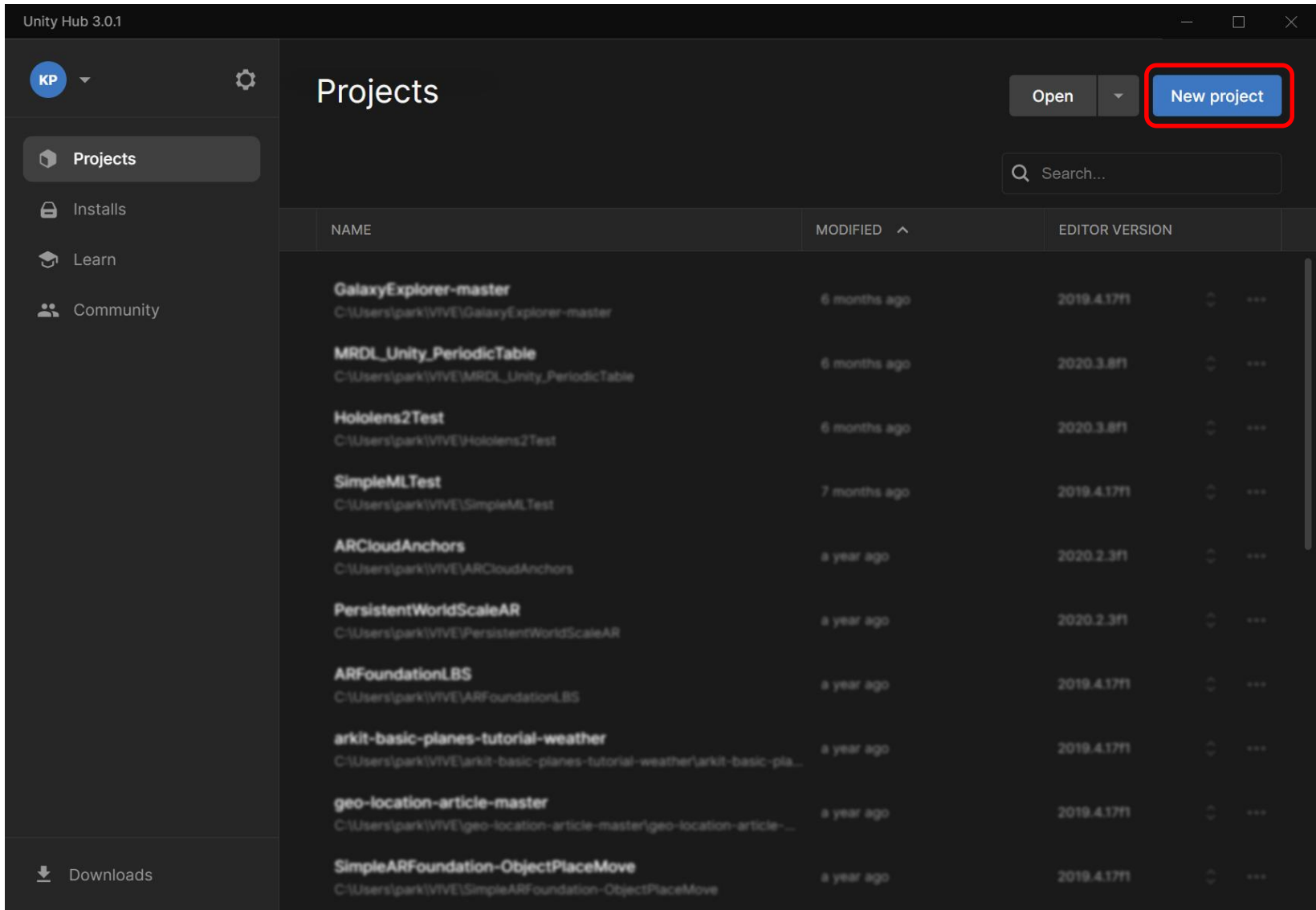
CANCEL

BACK

DONE

ADD

Begin a Microgame



The screenshot shows the Unity Hub 3.0.1 interface. The title bar reads "Unity Hub 3.0.1". On the left sidebar, there are navigation options: "Projects" (selected), "Installs", "Learn", and "Community". The main area is titled "Projects" and contains a table of project entries. In the top right corner of the main area, there are "Open" and "New project" buttons. The "New project" button is highlighted with a red rectangle. Below the buttons is a search bar labeled "Search...".

NAME	MODIFIED	EDITOR VERSION		
GalaxyExplorer-master C:\Users\park\WVE\GalaxyExplorer-master	6 months ago	2019.4.17f1	⌵	⋮
MRDL_Unity_PeriodicTable C:\Users\park\WVE\MRDL_Unity_PeriodicTable	6 months ago	2020.3.8f1	⌵	⋮
Hololens2Test C:\Users\park\WVE\Hololens2Test	6 months ago	2020.3.8f1	⌵	⋮
SimpleMLTest C:\Users\park\WVE\SimpleMLTest	7 months ago	2019.4.17f1	⌵	⋮
ARCloudAnchors C:\Users\park\WVE\ARCloudAnchors	a year ago	2020.2.3f1	⌵	⋮
PersistentWorldScaleAR C:\Users\park\WVE\PersistentWorldScaleAR	a year ago	2020.2.3f1	⌵	⋮
ARFoundationLBS C:\Users\park\WVE\ARFoundationLBS	a year ago	2019.4.17f1	⌵	⋮
arkit-basic-planes-tutorial-weather C:\Users\park\WVE\arkit-basic-planes-tutorial-weather\arkit-basic-pla...	a year ago	2019.4.17f1	⌵	⋮
geo-location-article-master C:\Users\park\WVE\geo-location-article-master\geo-location-article-...	a year ago	2019.4.17f1	⌵	⋮
SimpleARFoundation-ObjectPlaceMove C:\Users\park\WVE\SimpleARFoundation-ObjectPlaceMove	a year ago	2019.4.17f1	⌵	⋮

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.

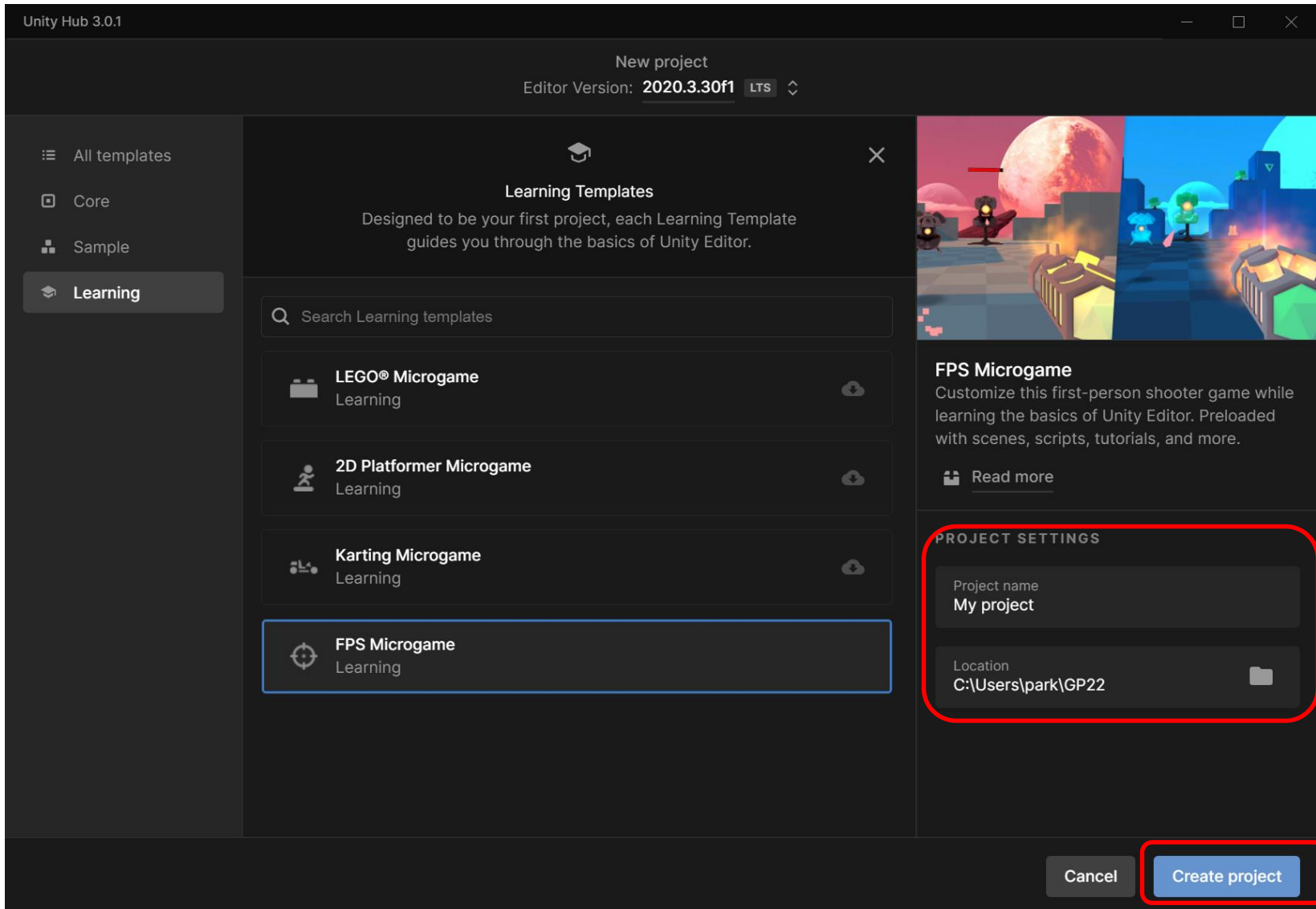
[Read more](#)

PROJECT SETTINGS

Project name
My project

Location
C:\Users\park\GP22

Cancel **Create project**

The image shows the Unity Hub 3.0.1 interface for creating a new project. The 'New project' dialog is open, displaying a list of 'Learning Templates'. The 'FPS Microgame' template is highlighted with a blue border. To the right, a preview of the FPS Microgame is shown, featuring a first-person view of a character in a futuristic environment with a large red moon. Below the preview, the 'PROJECT SETTINGS' section is visible, with a red box around it. The 'Project name' is set to 'My project' and the 'Location' is 'C:\Users\park\GP22'. At the bottom right, the 'Create project' button is highlighted with a red box.

Unity Interface

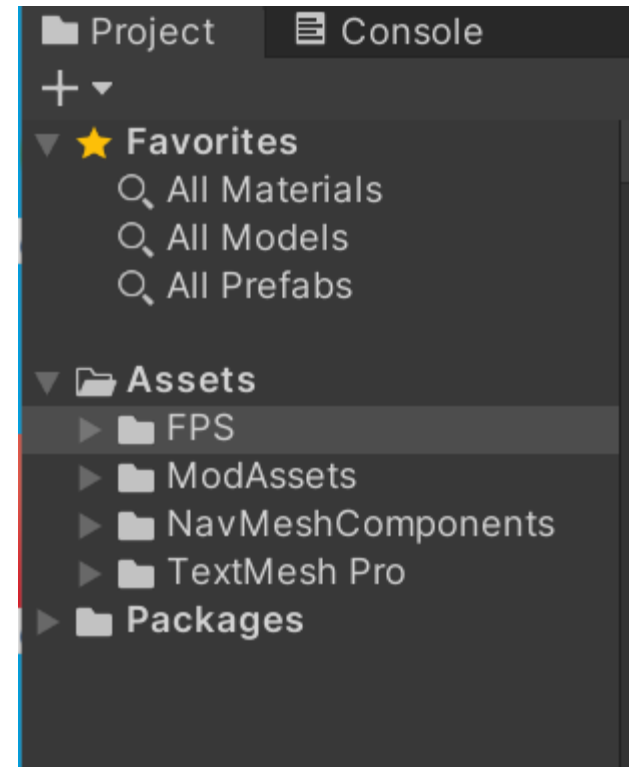
Unity Interface

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



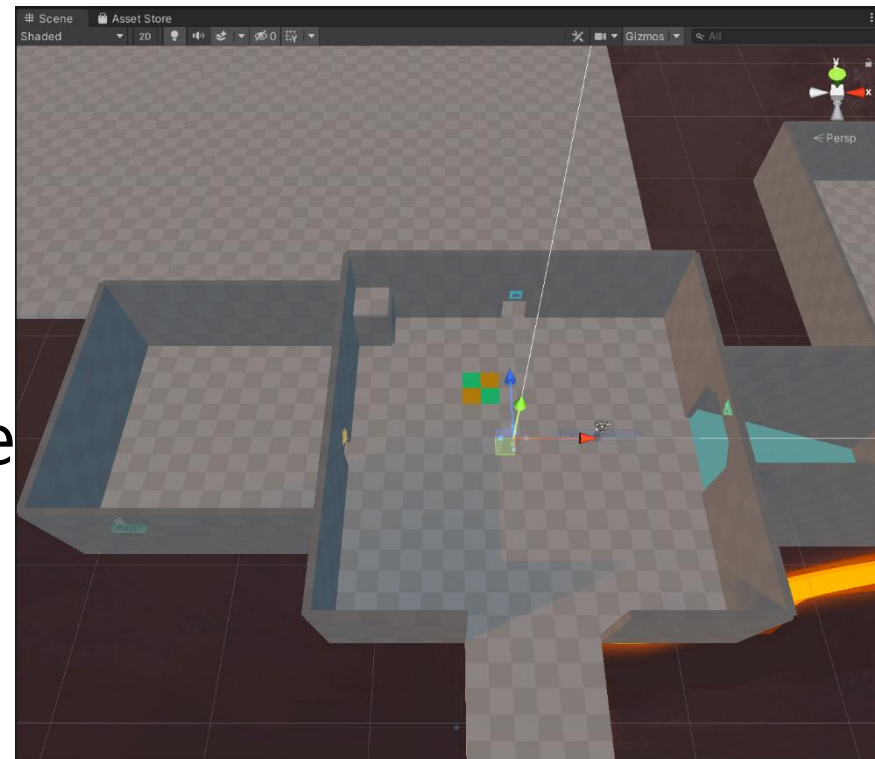
Project Window

- ❑ 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.).



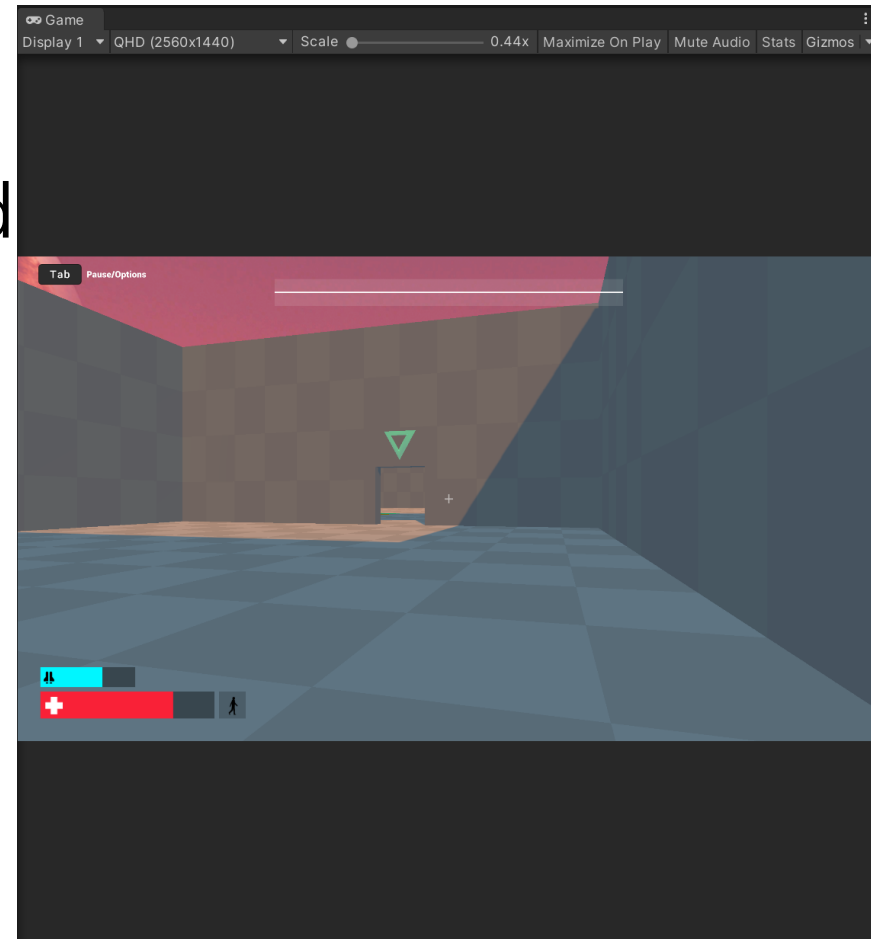
Scene View

- 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.



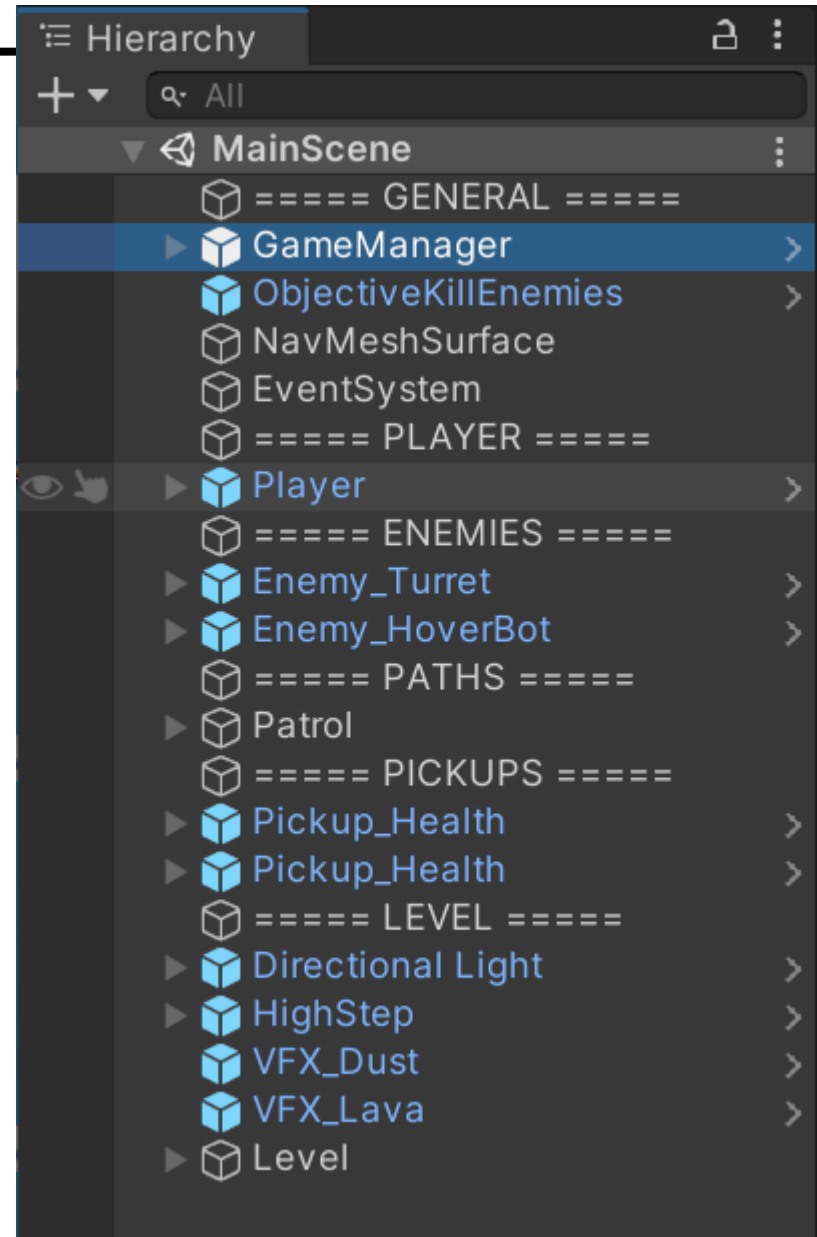
Game View

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



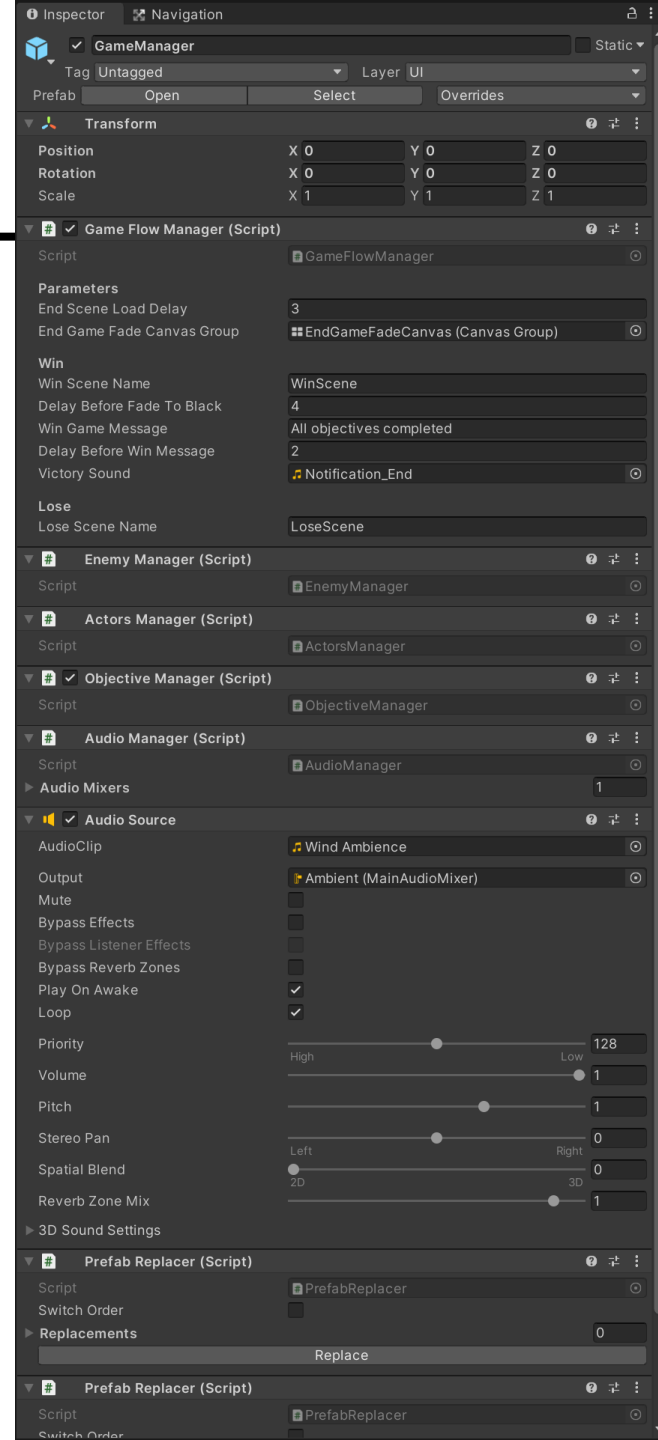
Hierarchy

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



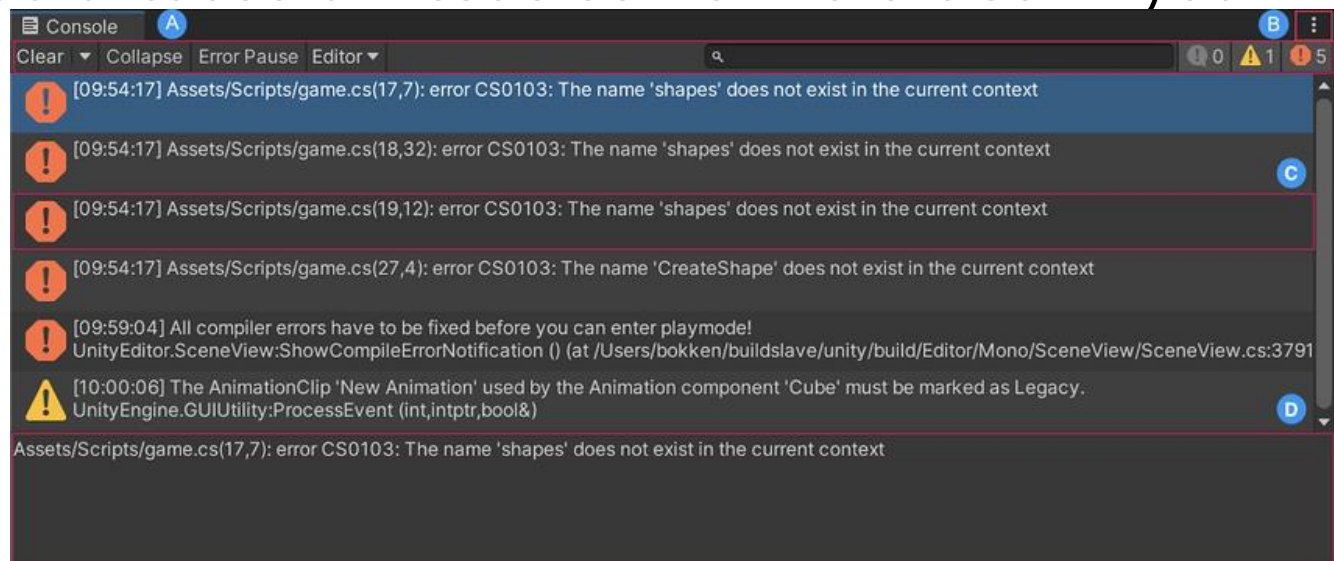
Inspector

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

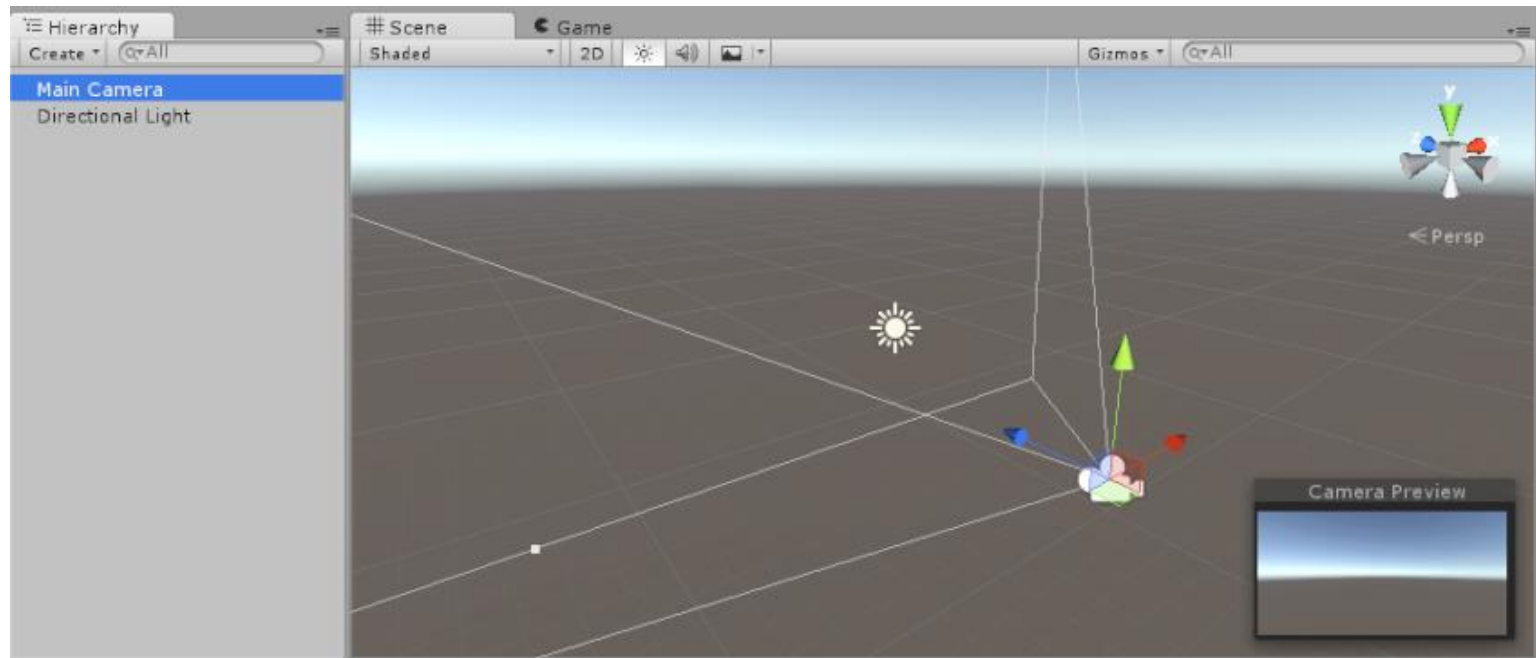


Unity Basics

Scene

□ Scenes

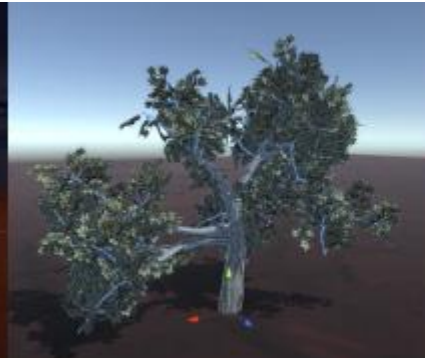
- A scene contains a collection of game objects that constitute the world that the player sees at any time.
- Below example shows a sample scene that contains only a **Camera** and a **Light**.



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.

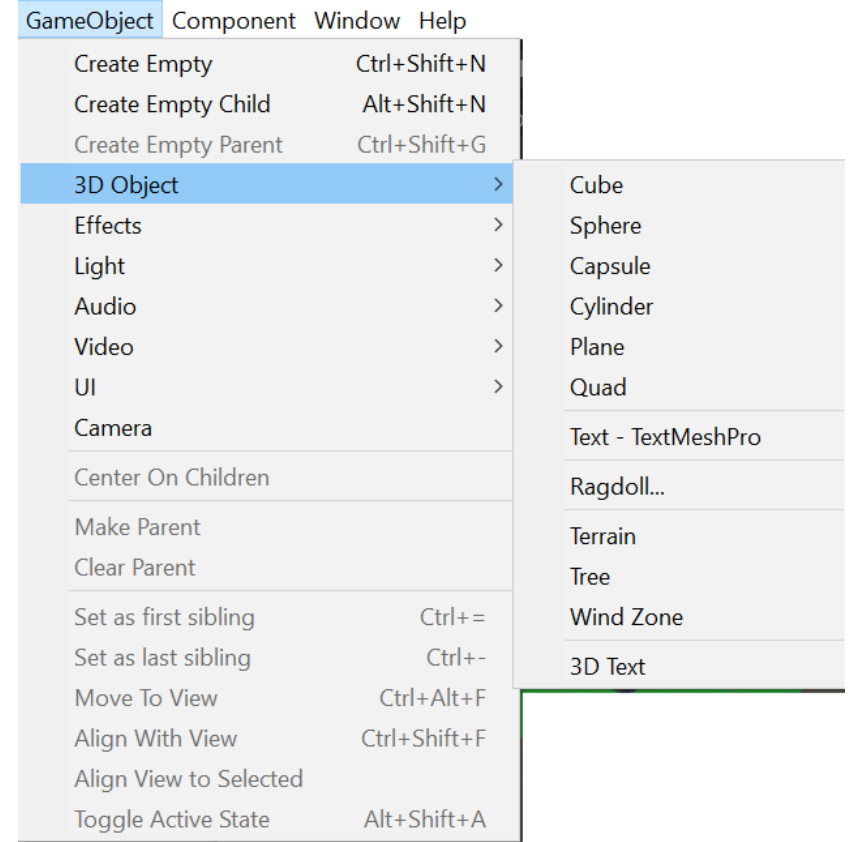


Four different types of GameObject: an animated character, a light, a tree, and an audio source

GameObjects

□ GameObject

- 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

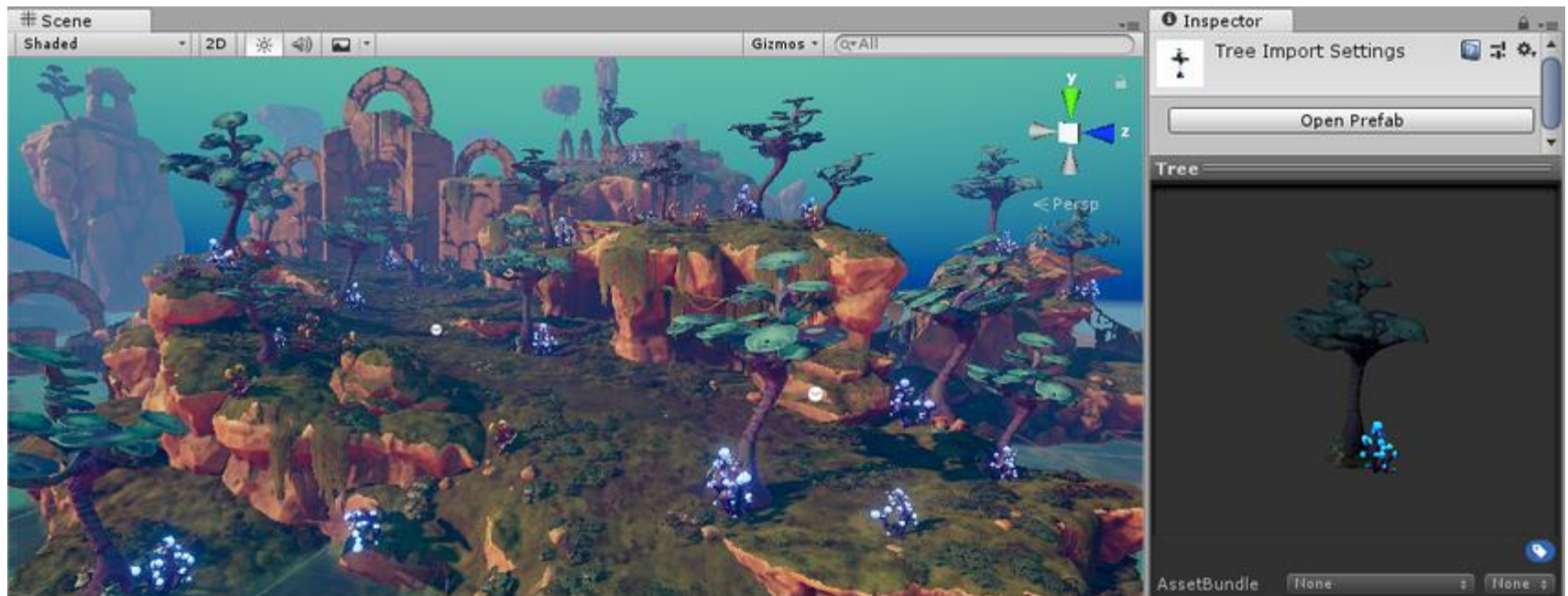
- Components
 - Mesh
 - Effects
 - Physics
 - Physics2D
 - Navigation
 - Audio
 - Video
 - Rendering
 - Tilemap
 - Layout
 - Playables
 - Miscellaneous
 - Scripts
 - UI
 - Event

Component	Window	Help
Add...		Ctrl+Shift+A
Mesh		>
Effects		>
Physics		>
Physics 2D		>
Navigation		>
Audio		>
Video		>
Rendering		>
Tilemap		>
Layout		>
Playables		>
Miscellaneous		>
Scripts		>
UI		>
Event		>

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) for Script

Edit Assets GameObject Component Window

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...	

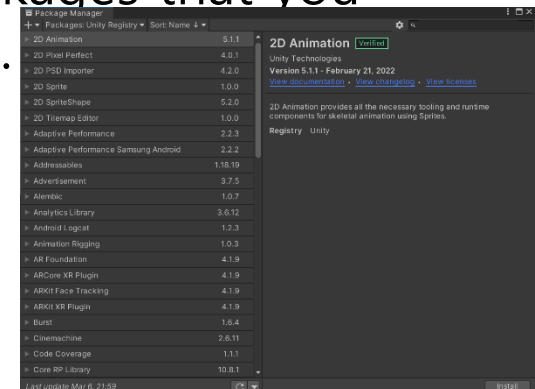
The screenshot shows the Visual Studio Preferences dialog, specifically the External Tools section. The left sidebar lists various preference categories, with 'External Tools' selected. The main area is titled 'External Tools' and contains the following settings:

- External Script Editor: Visual Studio Community 2022 [17.1.0] (Visual Studio Editor v2.0.14 enabled)
- Generate .csproj files for:
 - Embedded packages:
 - Local packages:
 - Registry packages:
 - Git packages:
 - Built-in packages:
 - Local tarball:
 - Packages from unknown sources:
 - Player projects:
- Regenerate project files: [Button]
- Image application: Open by file extension
- Revision Control Diff/Merge: WinMerge

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
- 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.



Reference

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