Unit 1:
Get Started in Unity

UCA Game Dev Course

2023

What is Unity?

https://www.youtube.com/watch?v=MuD7eLyfPMg

-> WHAT CAN YOU DO WITH IT?

Make games, simulations, films, apps, VR, AR, and more.

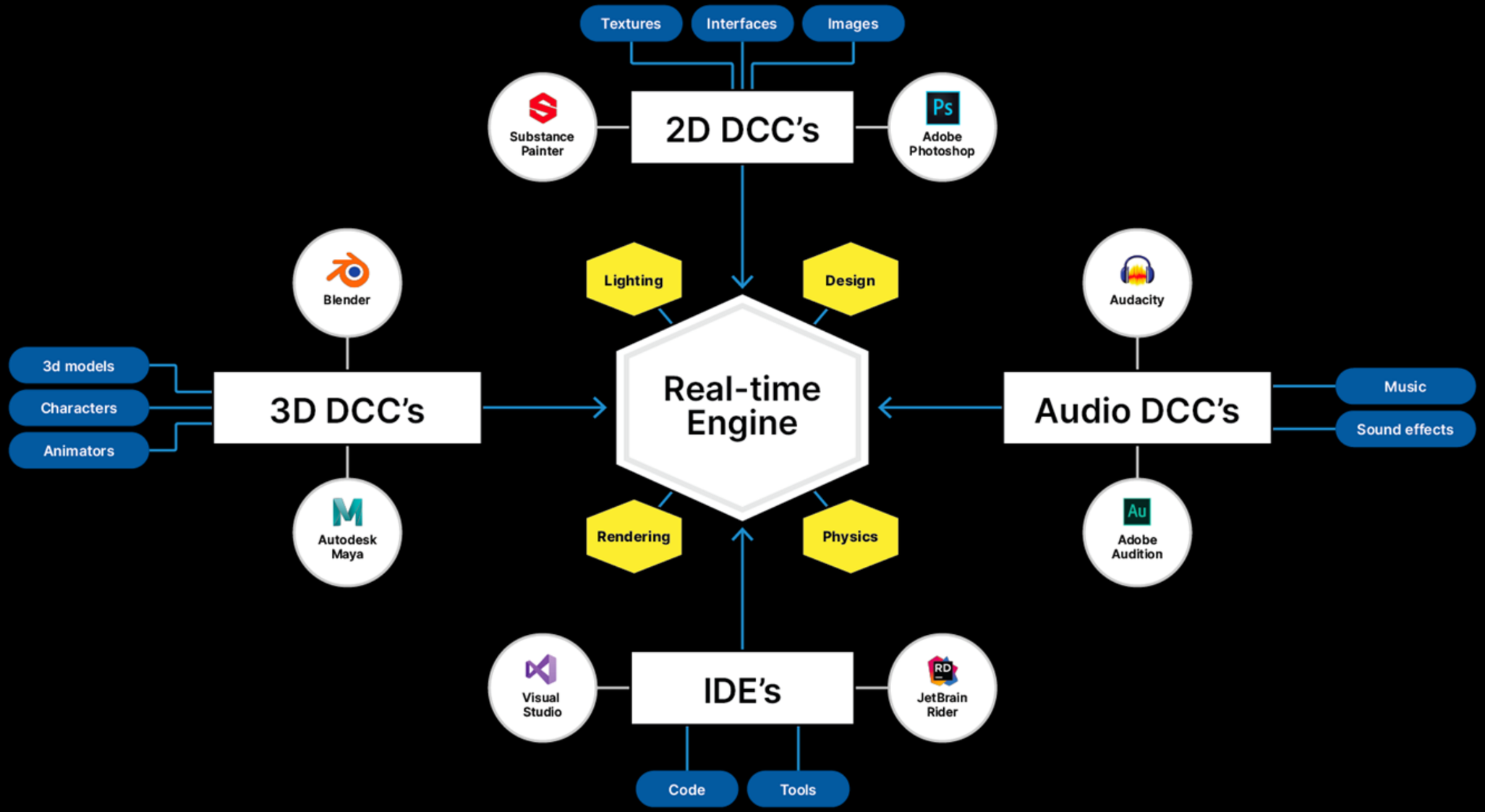
What is a Game Engine?

Unity began its life as a game engine but evolved to be a creative tool that's used by many different industries.
A **game engine** is the point of convergence for all aspects of creating a game. Games, like all applications, are made of smaller pieces like 3D models, scripts, and audio files. When put together, they create the full user experience.

http://drive.google.com/file/d/1xu2TdxSVPjHoxxomde-jRNkbEeeF5sOp/view

-> WHAT IS “REAL-TIME”?

Unity is a real-time 3D engine. What does “real-time” really mean?



Inside a game engine, you don't create assets. Instead, assets are created in specialized external programs called **Digital Content Creation (DCC)** tools. Many DCCs are integrated with Unity to ease the process of importing them.

25+

Platforms Supported

1M+

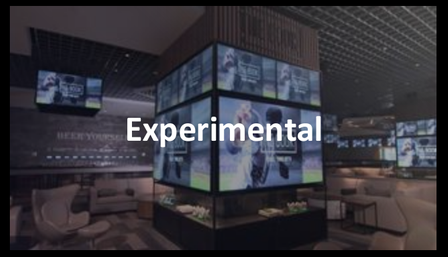
Monthly Active Developers

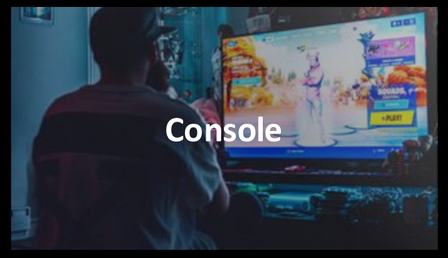
3B+

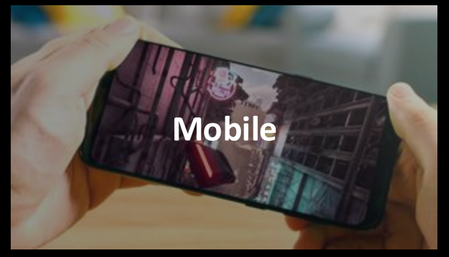
Devices Reached Monthly

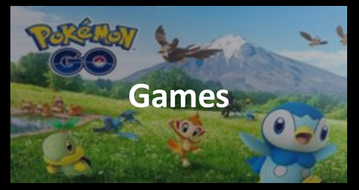
The World's Leading Real-time 3D creation Platform

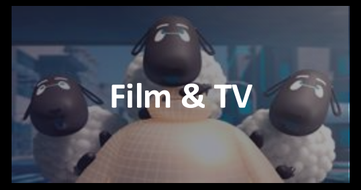
Across platforms and sectors

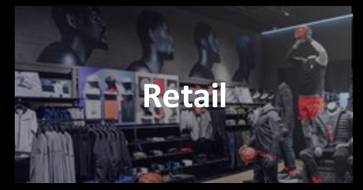


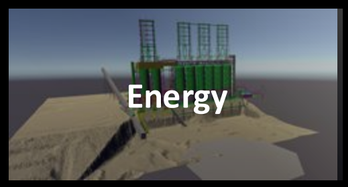


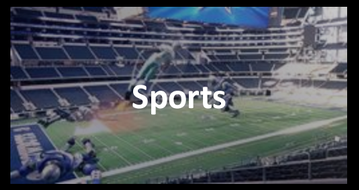


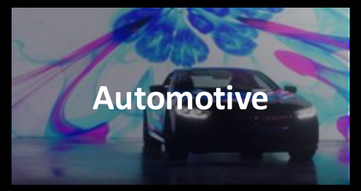


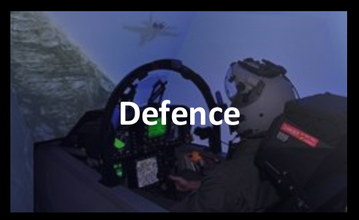


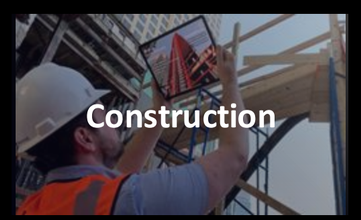


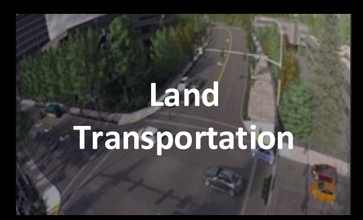


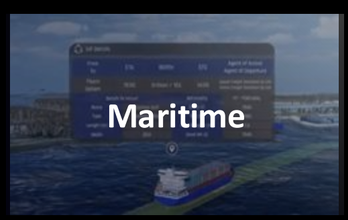




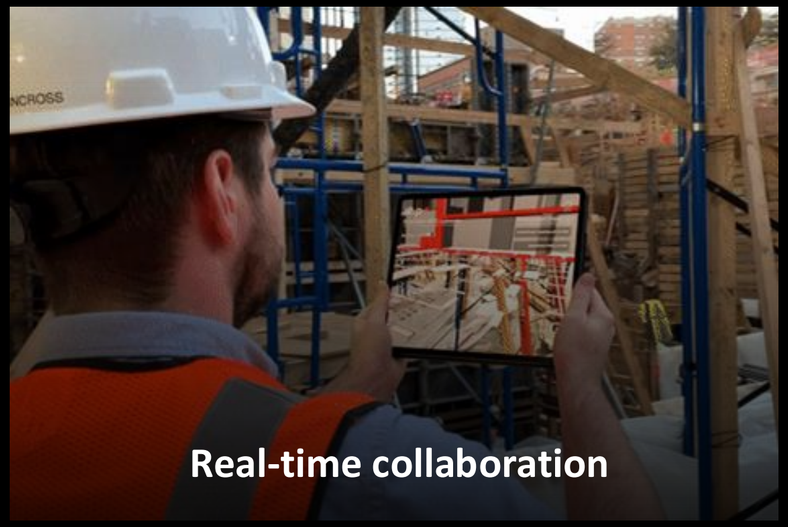


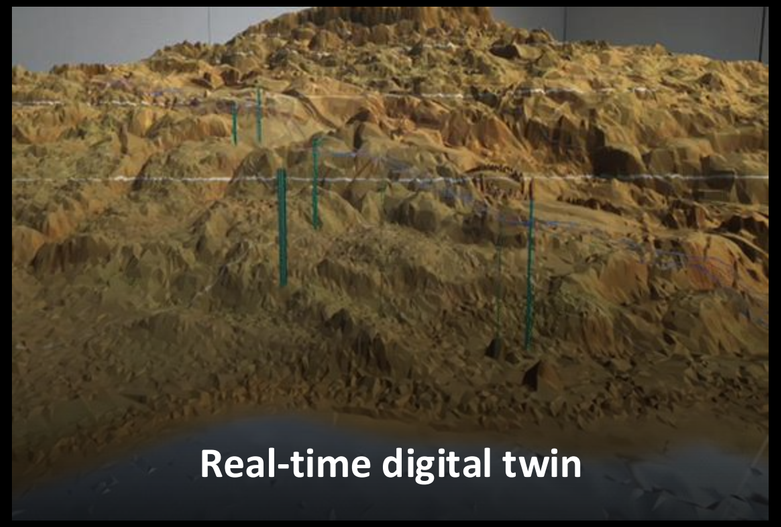


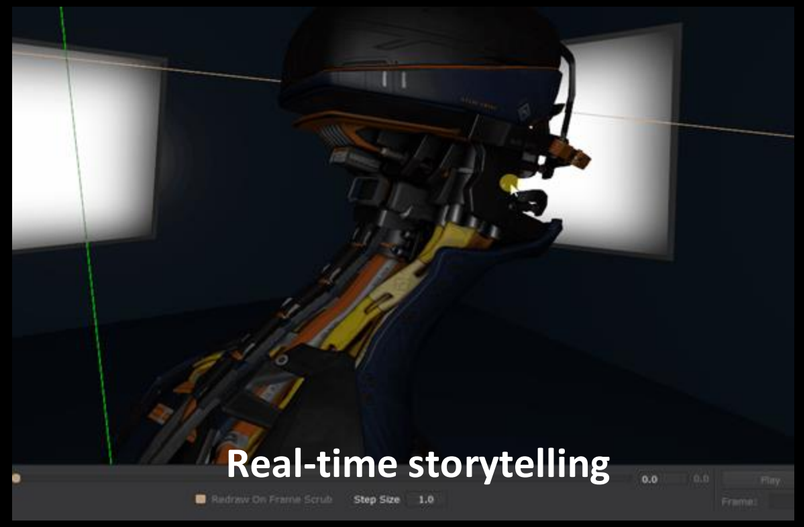


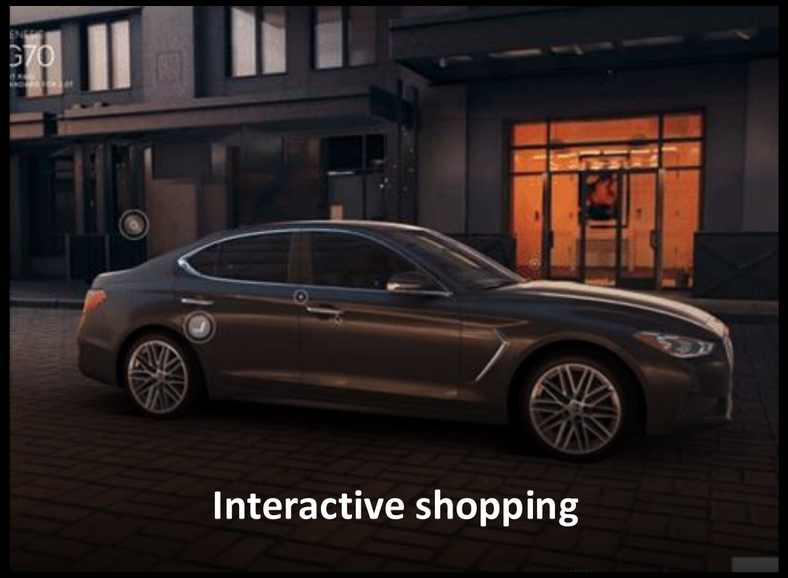


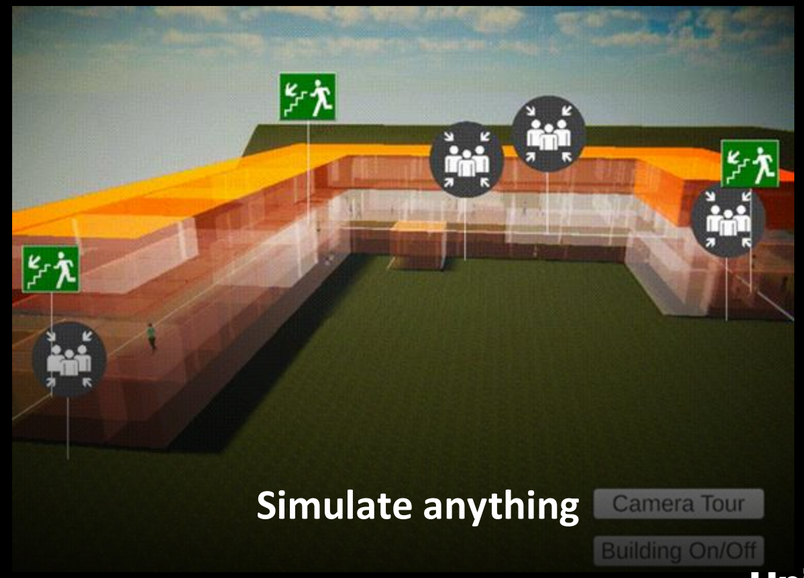
Real-time 3D technology use cases



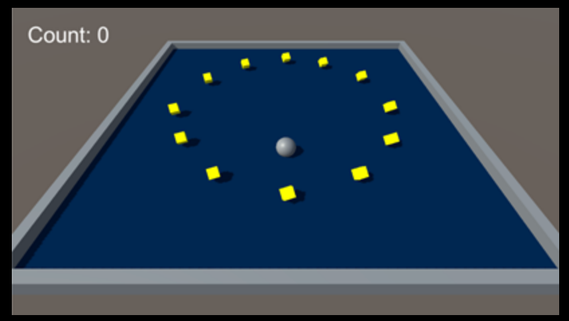






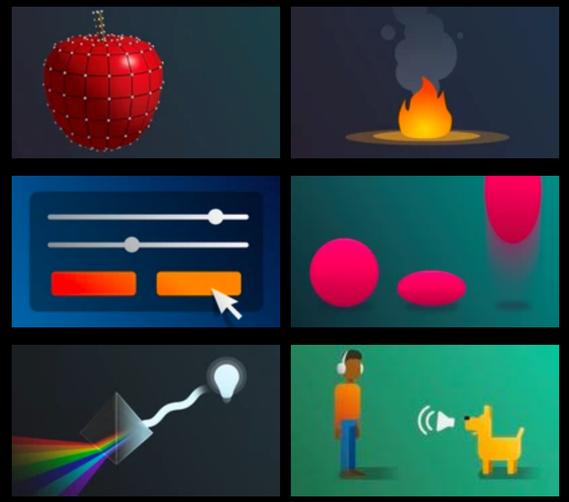


What will you do in this course?



Simple game

+



Creative Core
materials, vfx, UI, animation, lighting, audio

apply to your game

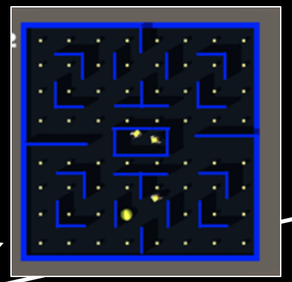
Custom game

Design Document

GAME DESIGN DOCUMENT By: Unity Technologies

1 - Introduction

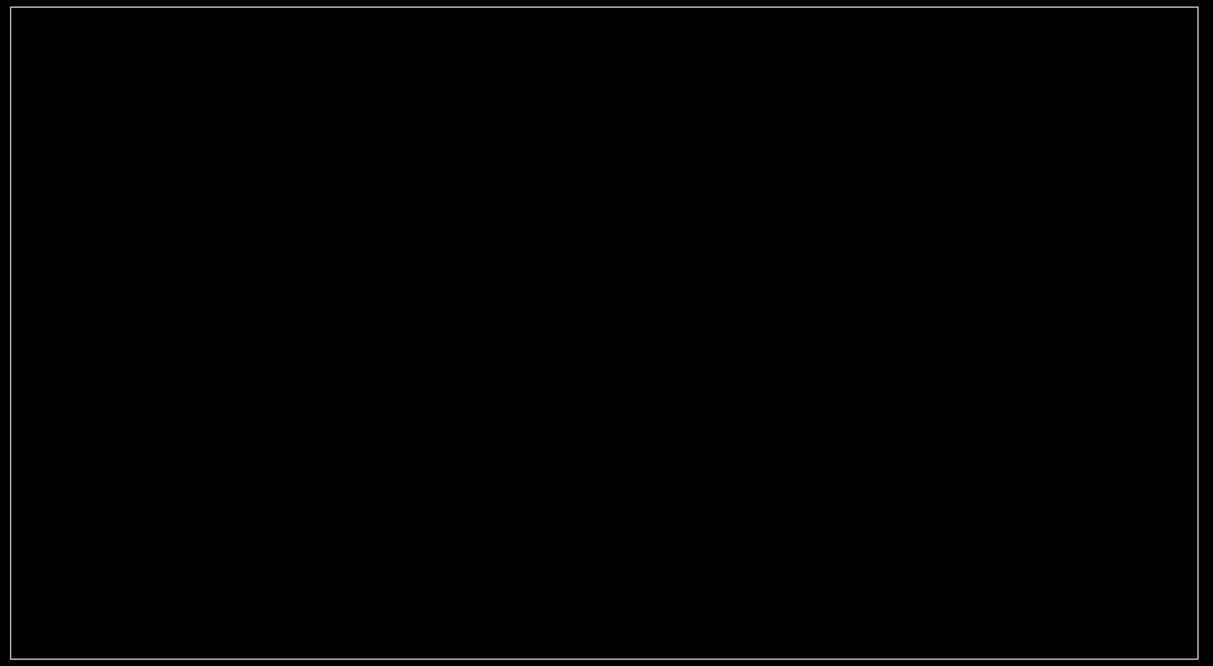
Working title:	Pinchy Crab
Concept:	A beach ball is rolling around on the beach trying to avoid getting pinched and popped by a grumpy crab. The ball is trying to collect treasures along the way.
Genre:	Casual game
Target Audience:	5-10 year-old casual gamers.
Target Platform:	Desktop | Mobile | Web



What will you do in this unit?

Navigate the Unity Editor:

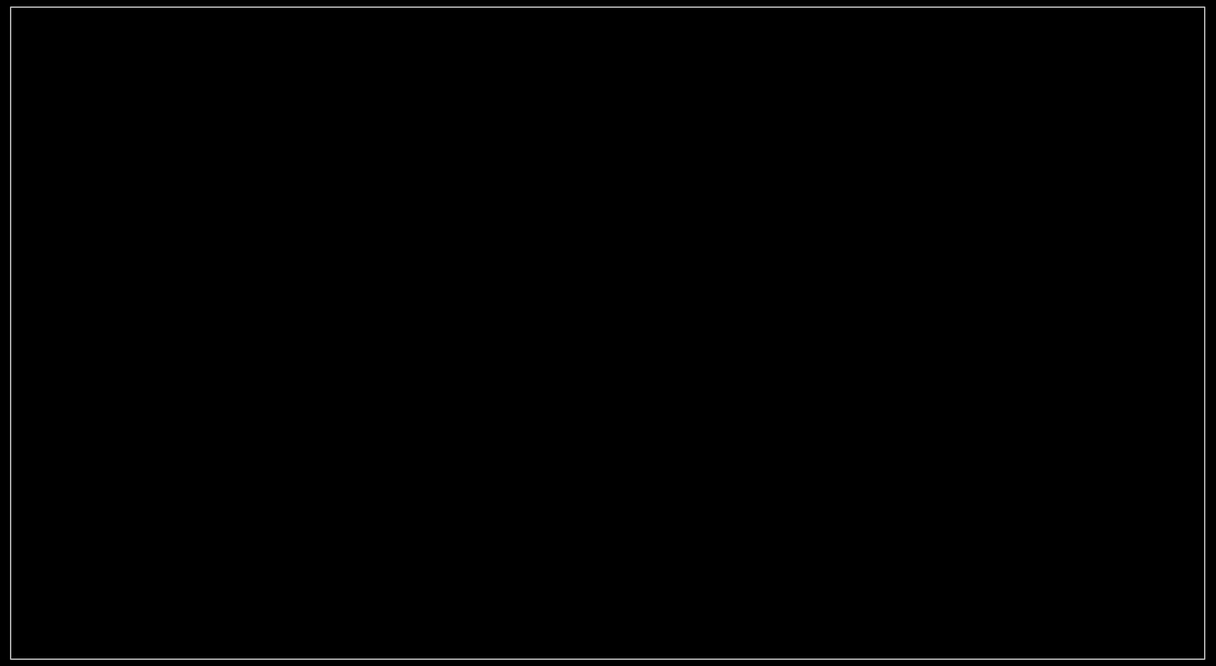
→ You will create a new project, navigate Unity's interface, experiment with 3D objects and materials, integrate Unity's physics system, and learn how to use prefabs.



Create your first scene:

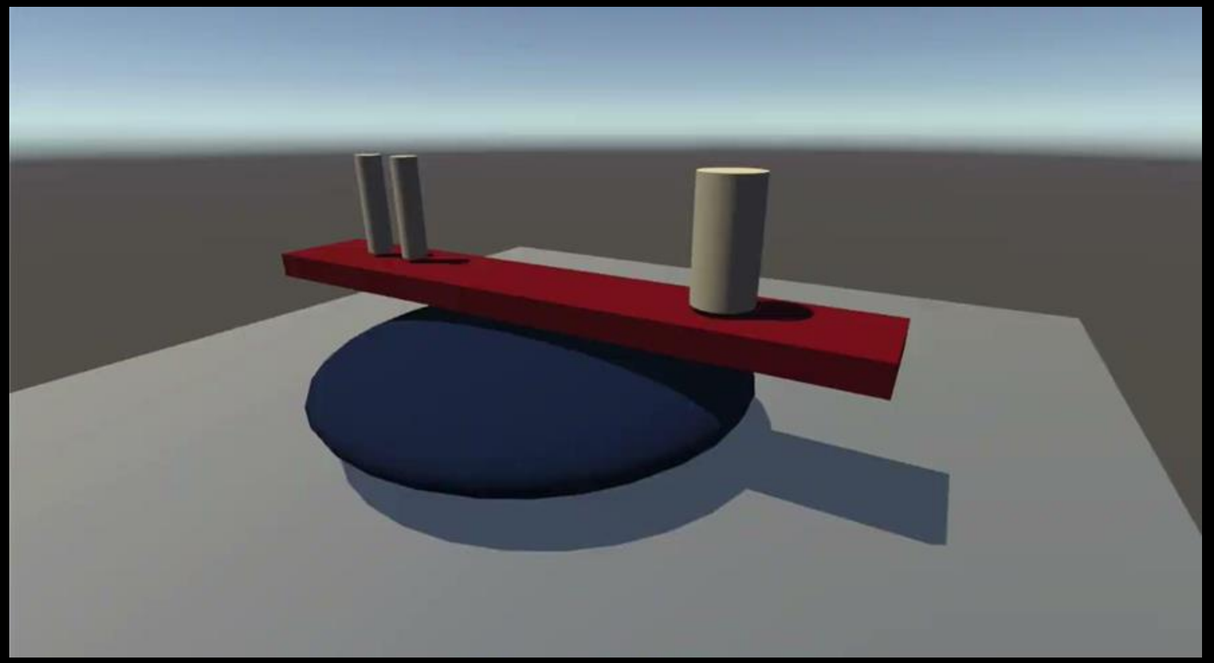
→ You'll create a unique scene that you designed from scratch, which you can explore with the character of your choice.

→ It could look something like this example — or it could look completely different!



Create a balanced primitive structure:

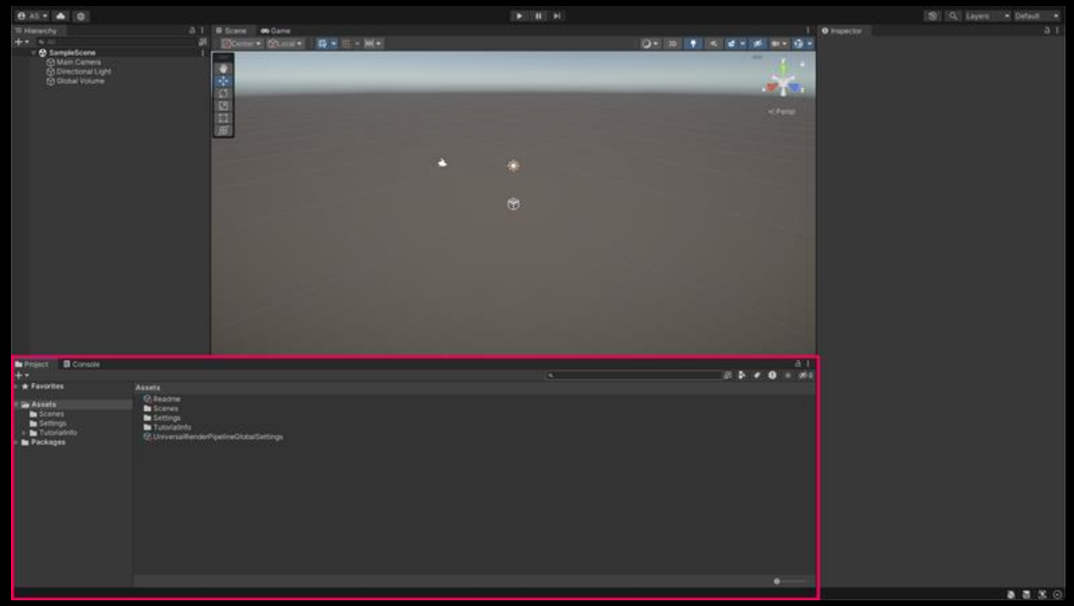
→ You will construct a sculpture from various Unity primitives, ensuring it remains stable with gravity enabled.



Key concepts for this unit

→ The **Project window** is typically found at the bottom of the screen, but it can be moved or docked elsewhere based on your personal layout preference.

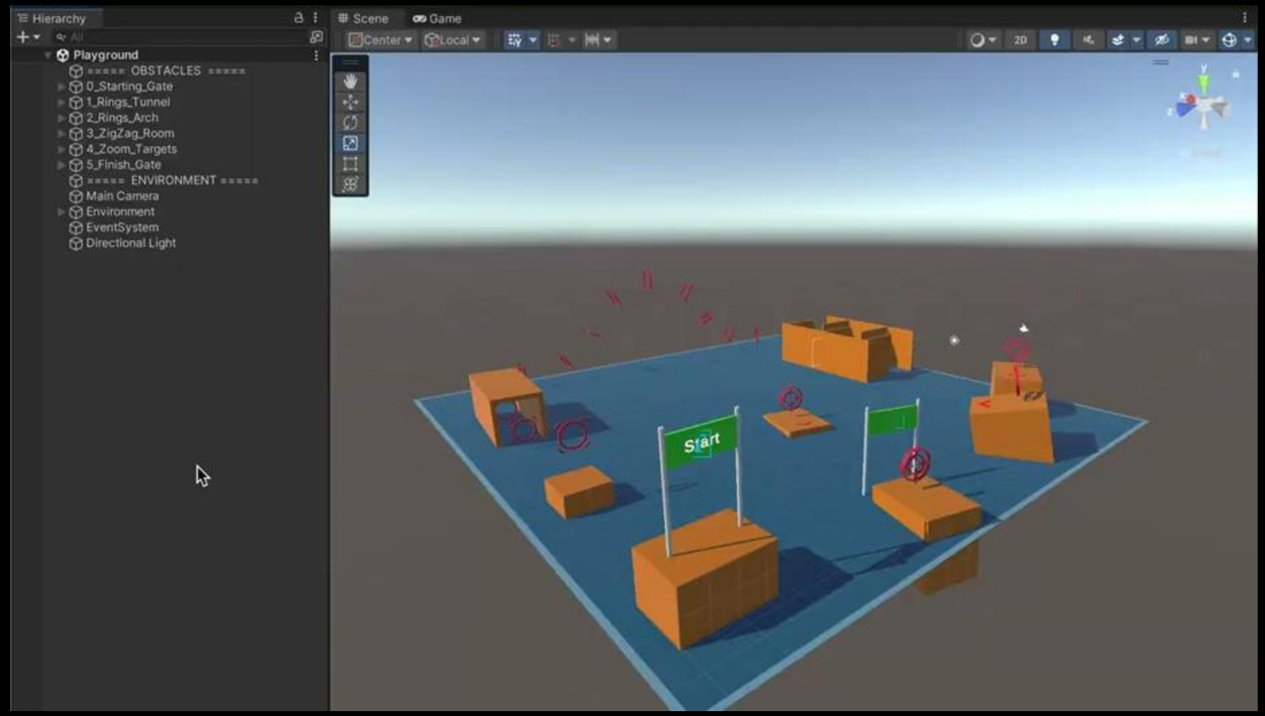
→ The Project window is essentially the file manager of your Unity project. It's where all the files you'll use in your application, such as scripts, textures, audio clips, and scenes, are listed and can be managed.



→ The **Hierarchy** window lists all the objects in your current scene and is located on the leftmost side in the Default layout.

→ When you double–click on an object in the Hierarchy window, you select that object and the Scene view focuses on it, centering the object in the camera's perspective.

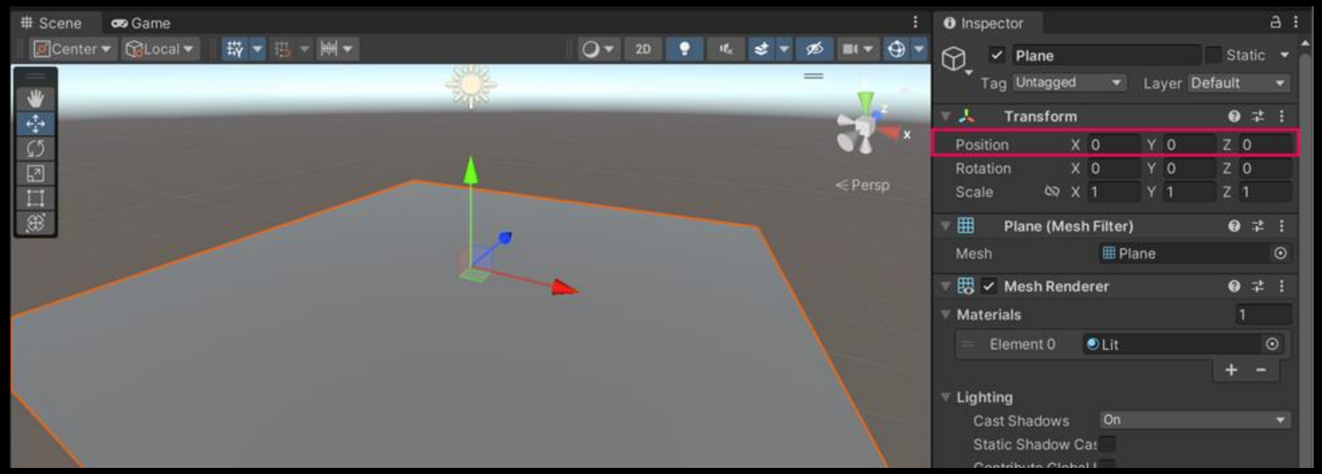
→ You can create **empty** objects in the Hierarchy as containers to group and **parent** related game objects for better organization.



→ The **Inspector window** displays detailed properties and settings of the currently selected GameObject.

→ A **component** is a modular piece of functionality or behavior that can be added to a GameObject.

→ The **Transform** component defines an object's position, rotation, and scale in the scene and is the only required component.



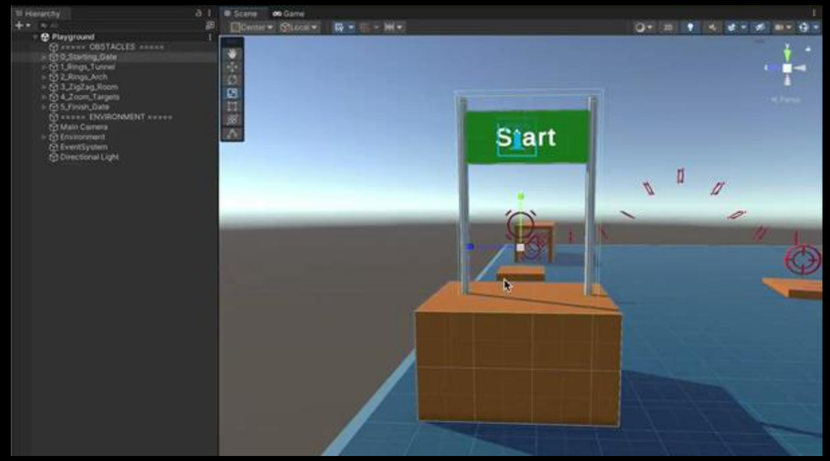
→ The **Scene View** is where you design, edit, and navigate your game's 3D environment.

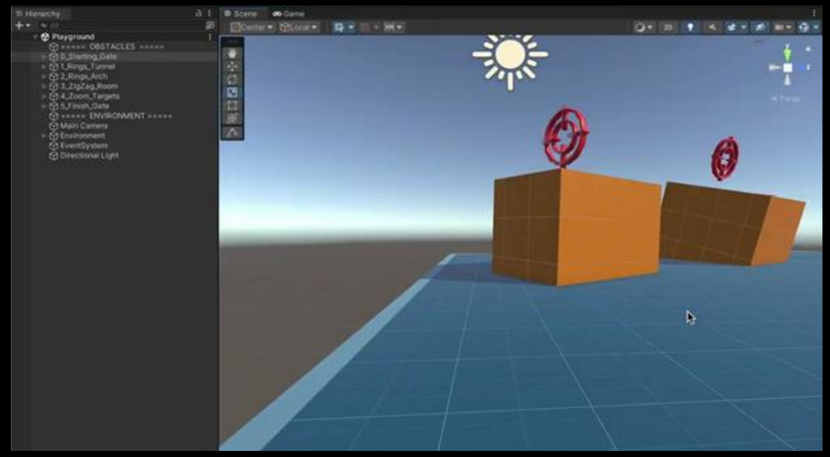
→ **Flythrough mode** lets you navigate the scene as if "flying" through it.

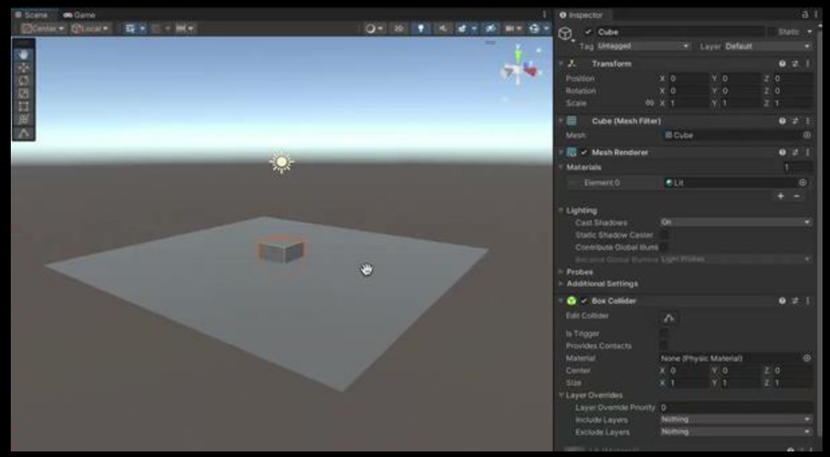
→ You can drag or use tools to change the position of game objects.

→ When you press **F**, the camera focuses on the selected object.

→ Holding **Alt** and dragging the mouse rotates the camera view around a point of interest.



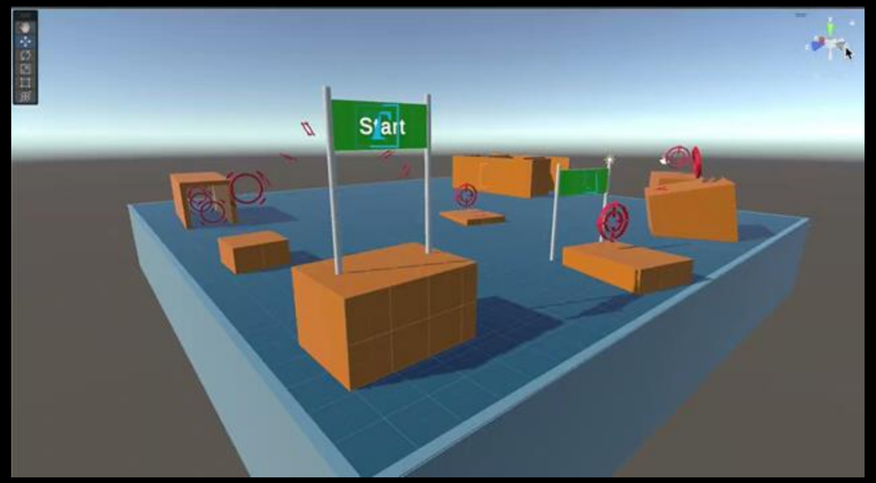




→ At the top-right corner of the Scene view is the **Scene gizmo**, which serves as a control center for adjusting your viewing angles and projection modes.

→ You can toggle between **Perspective** and **Orthographic** (also known as isometric) modes.

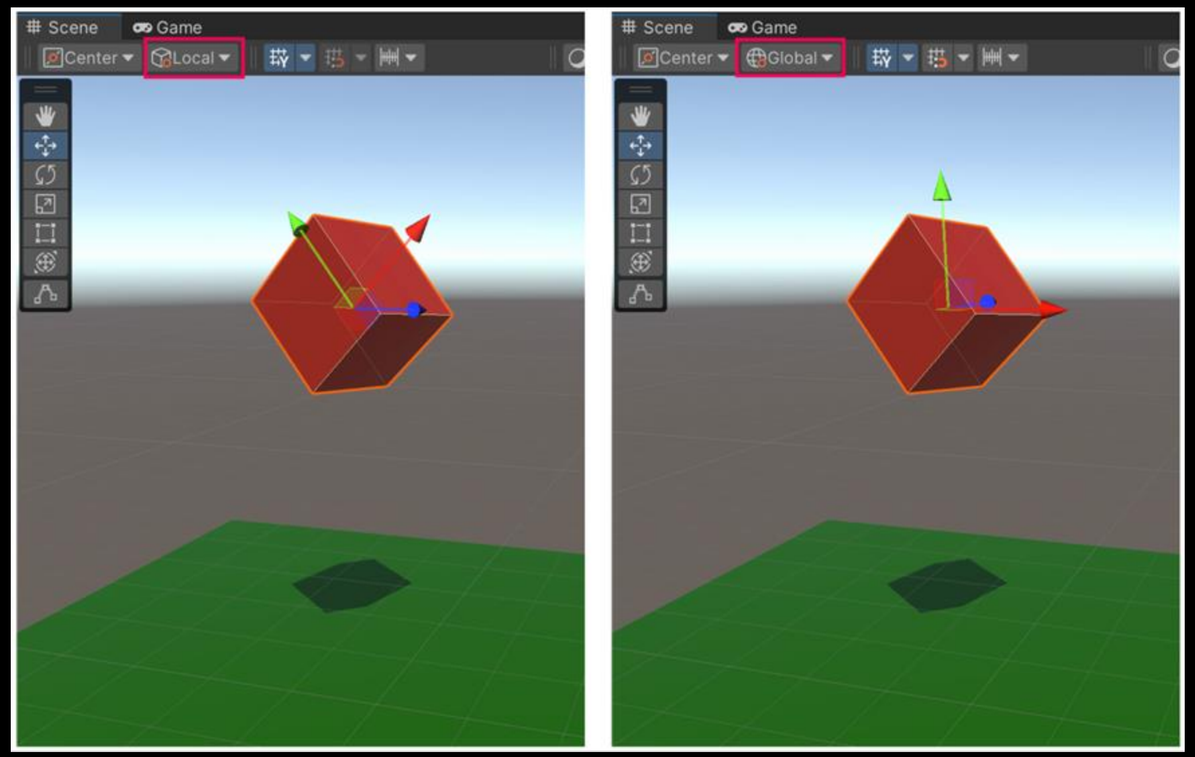
→ The Orthographic mode, devoid of perspective, pairs excellently with an axis arm click to give you an undistorted front, top, or side view of your scene.



You can use **Global** or **Local** coordinate systems.

→ **Global mode** allows consistent movement or rotation based on the world's axes.

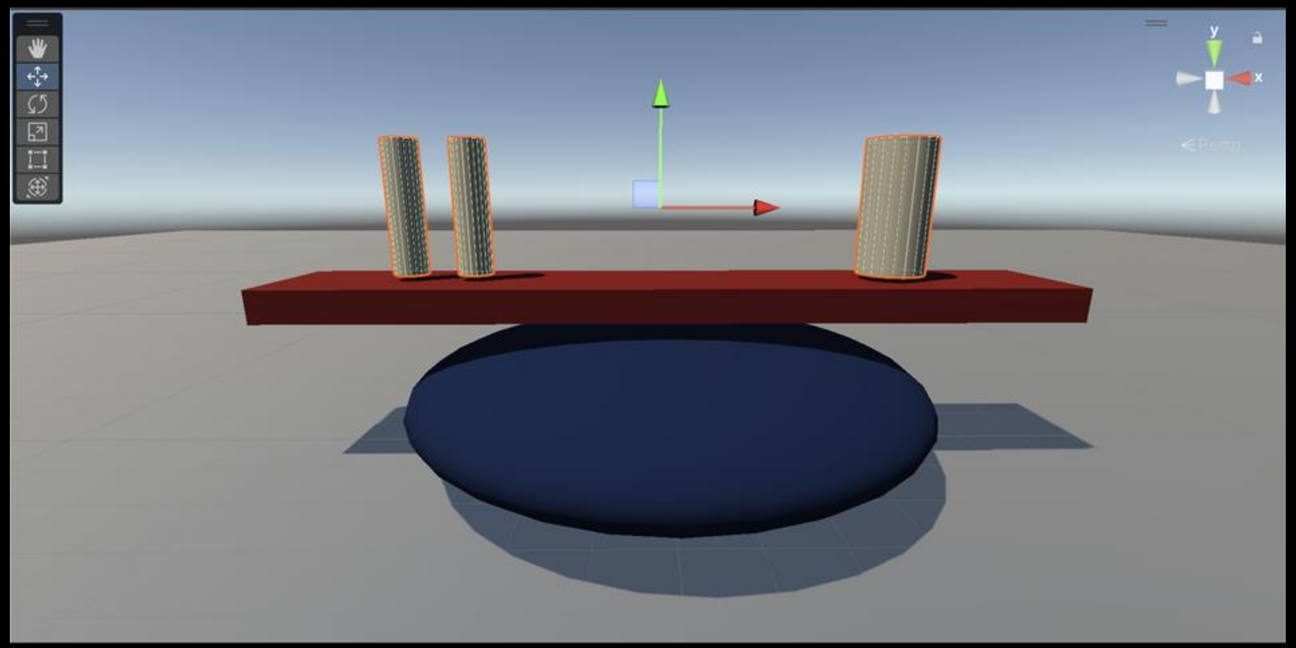
→ **Local mode** allows movement or rotation based on the selected object's own orientation.



→ The **Rigidbody** component allows game objects to be affected by physics, enabling movement, gravity, and force interactions.

→ **Collider** components define an object's physical shape for collision detection, ensuring it can interact with other colliders and the environment.

→ There are box colliders, sphere colliders, capsule colliders, and mesh colliders



Learning Objective Review

Exam objectives in the unit (1 of 2)

→ 14: Asset Management | Assets | Default GameObjects - Differentiate GameObjects by their appearance

→ 15: Asset Management | Assets | Default GameObjects - Identify GameObjects within a scene

→ 20: Asset Management | Assets | Scene File - Load a scene

→ 21: Asset Management | Assets | Scene File - Save a scene

→ 31: Editor Interface | Editor Customization | Layouts - Differentiate Unity editors

→ 34: Editor Interface | Views | Hierarchy - Explain the purpose of the Hierarchy Window

→ 35: Editor Interface | Views | Hierarchy - Differentiate the Hierarchy Window

→ 36: Editor Interface | Views | Hierarchy - Utilize the Hierarchy Window

→ 37: Editor Interface | Views | Hierarchy - Create empty GameObjects

→ 38: Editor Interface | Views | Hierarchy - Parent objects

→ 39: Editor Interface | Views | Inspector - Explain the functionality of the Inspector Window

→ 40: Editor Interface | Views | Inspector - Reset component

Exam objectives in the unit (2 of 2)

→ 41: Editor Interface | Views | Project - Explain the functionality of the Project View Window
→ 42: Editor Interface | Views | Project - Explain the purpose of the Project View Window
→ 45: Editor Interface | Views | Project - Focus the Scene View Camera
→ 46: Editor Interface | Views | Scene - Differentiate the Project View Window
→ 47: Editor Interface | Views | Scene - Use the Zoom Tool
→ 48: Editor Interface | Views | Scene - Orbit the camera
→ 49: Editor Interface | Views | Toolbar - Modify Gizmo's
→ 87: Physics | Colliders | 3D Capsule - Identify colliders by their appearance
→ 90: Physics | Colliders | 3D Capsule - Utilize colliders
→ 94: Physics | Rigid Bodies | Components - Explain Rigidbody's
→ 116: Project Management | Game Objects | Transform - Explain the function of GameObject components
→ 117: Project Management | Game Objects | Transform - Recognize GameObject components

Thank you

Unity®

UNITY.COM


Unity®