



🎉 New! Share publicly

Chat



Unity Game Development: Beginner to Intermediate

7 sources

These YouTube transcripts offer comprehensive tutorials on **game development using Unity**, a popular game engine. **Jason Weimann (GameDev)** provides a beginner-friendly approach, starting with **Unity installation**, **basic object manipulation** like scaling and positioning, and introducing **fundamental programming concepts** such as **rigidbodies** and **colliders** for physics interactions. The tutorial progresses to **creating simple animations** for characters, **implementing enemy behavior** including destruction on collision, **incorporating particle systems** for visual effects, and **managing multiple game levels**. **Code Monkey** presents a more intermediate course, focusing on building a "Builder Defender" game with an emphasis on **clean code practices** and **advanced Unity features** such as the **new input system**, **Shader Graph**, and **Cinemachine** for camera control. This course covers **complex interactions**, **UI development** including **progress bars** and **main menus**, and **sound management**. **Sorcerer** also offers a beginner tutorial, covering similar ground by demonstrating **scene setup**, **component understanding** (mesh filter, mesh renderer, colliders, rigidbodies), **creating trigger zones** to execute code on entry, and **implementing player respawn points**. Both **Jason Weimann (GameDev)** and **Sorcerer** illustrate the creation of simple 2D or 3D games with basic movement and interaction, while **Code Monkey** expands into more sophisticated game mechanics and development workflows.

Can you give me 3 sources to help me develop a game on Unity

Here are three sources from the provided materials that can help you develop a game in Unity:

- **"How to Make a Game - Unity Beginner Tutorial" by Jason Weimann (GameDev):**

- This tutorial is designed for **complete beginners**, even those who have never coded or considered building a game before. By the end of the video, you will know how to build your own game.

- It covers the essential steps: getting set up, **what tools are needed (Unity and Unity Hub)** and **how to install them**, and how to **pull art into a game easily**.

- The tutorial simplifies **coding in C#**, making it understandable even for a ten-year-old.

- You will learn to **manage Unity projects and installations using Unity Hub**, install specific Unity versions, and create a new project, specifically using the **2D template** for a 2D game.
- It guides you through setting up your **Unity editor layout to default** for consistency.
- Key concepts covered include creating sprites, adjusting their position and scale using transform tools, finding **free art resources** from sites like OpenGameArt.org, and importing assets into your project (either by dragging from an extracted zip or copying into the project's assets folder).
- The tutorial demonstrates how to **implement physics using Rigidbody2D** to make objects fall and how to **set up BoxCollider2D for collision detection** without necessarily matching the visual sprite.
- It delves into C# scripting, explaining how to create a C# script, the meaning of using statements, public classes, and MonoBehaviour inheritance, which allows a script to be attached as a component to game objects.
- Important Unity callback methods like Awake (called on initialization), Start (called before the first frame update), Update (called once per frame for continuous logic), OnMouseDown (for mouse input), and OnMouseUp (for mouse release) are explained.
- You'll learn to **control object positions (transform.position) based on mouse input** and how to **launch objects by adding force to their Rigidbody2D** using vectors.
- The concept of a **"launch power" variable exposed in the Inspector** (using [SerializeField]) is introduced for easier tweaking by designers without code changes.
- The tutorial also covers **scene management for creating multiple levels** and transitioning between them, using **particle systems for visual effects** like a "poof" cloud, and creating **prefabs** for reusable game objects.
- It also briefly introduces **Cinemachine for camera control** and helps create an enemy script to handle collisions and destruction.
- **"Learn Unity Beginner/Intermediate 2025 (FREE COMPLETE Course - Unity Tutorial)" by Code Monkey:**
 - This is a **free, complete course** designed for both **beginners and intermediate users**, teaching how to build a complex game from start to finish with a focus on **production-quality, easy-to-understand, clean code**. It explicitly states it will **not teach bad practices**.
 - The course covers a wide array of topics, including selecting a Unity version (recommending LTS versions), **Unity interface basics**, Visual Studio setup, and **coding style and naming rules** (e.g., PascalCase for functions, camelCase for parameters/fields, avoiding magic numbers, being explicit with accessors).
 - It guides you through importing assets, **setting up post-processing effects** like tone mapping, saturation, contrast, Bloom, and Vignette to enhance visuals.
 - A significant portion is dedicated to building a **character controller**, including keyboard input using the Legacy Input Manager and then **refactoring to the New Input System**, which makes **Gamepad joystick support super easy**.
 - It emphasizes **separating visuals from logic** by having empty parent game objects for logic and child game objects for visuals.
 - **Animation basics** and the Animator component are covered, along with the **Cinemachine package for advanced camera control**.

- Key programming concepts include **Collision detection** using `Physics.CapsuleCast` for accurate character shape collision, which is preferred over `Raycast` for player bodies.
- It introduces **C# interfaces** for flexible object interaction (e.g., allowing players or counters to hold kitchen objects) and **C# events** for loose coupling and efficient notification when actions occur (e.g., `OnInteractAction`, `OnPlayerGrabObject`).
- The course highlights the use of **Scriptable Objects for defining object types** and related data (e.g., kitchen objects, recipes).
- It explains **prefab variants** for creating reusable object hierarchies with shared base properties.
- You will learn about **World canvases** for in-game UI elements like progress bars and how to make objects always look at the camera.
- Complex counter logic is built, including a **stove counter with a state machine** for cooking and burning, and a cutting counter with progress bars.
- It also covers adding **music and sound effects** using `AudioSource` and `AudioMixer`.
- Finally, it guides through implementing **game start and end states** with a countdown, a **main menu with scene loading**, a pause window, and an **options menu with rebindable keys** and audio level control.
- **"Unity Full Beginner Tutorial | Make Your First Game" by Sorcerer:**
 - This tutorial focuses on helping **absolute beginners create their very first game from scratch** in Unity.
 - It starts with the basics: **installing Unity Hub and a Unity editor version**, and setting up **Visual Studio 2022 for game development with Unity**.
 - You'll learn to **create a new 3D core project**.
 - It introduces the **Unity editor interface**, specifically the **Hierarchy** (list of objects in a scene) and **Scene view** (visualization of objects).
 - The tutorial shows how to **place primitive 3D objects** (like cubes, spheres, planes) and how to **navigate the scene view** using mouse and keyboard.
 - It explains the **Inspector window**, where you view an object's **components** like `Transform` (position, rotation, scale), `Mesh Filter` (defines shape), and `Mesh Renderer` (makes mesh visible).
 - You will learn how to **add color to objects using Materials** by creating them in the Project window and applying them to objects.
 - Core physics concepts are covered: **BoxCollider for collision detection** (visualized by a green outline) and **Rigidbody to enable physics and gravity** for an object.
 - The tutorial introduces **Triggers** for running code when an object enters a specific area without causing physical collision.
 - You will write your first C# script, understanding the basic methods like `Start` (runs once at the beginning) and `Update` (runs every frame).
 - The `OnTriggerEnter` method is explained as the way to execute code when something enters a trigger.
 - You'll learn to **manipulate an object's position programmatically** using `transform.position` and `Vector3`.
 - The concept of **float for decimal numbers in C#** (e.g., `2.5f`) is introduced.

- It demonstrates how to **make a ball roll using `Rigidbody.AddForce`** and how to **get player input using `Input.GetAxis`** for horizontal and vertical movement (WASD or arrow keys).
- You'll learn to use a **[`SerializeField`] attribute** to expose private variables to the Inspector for easier modification.
- The tutorial also covers **encapsulation** using a public property with a private setter (Lambda expression) to access variables from other scripts.
- It shows how to **set up a respawn point using an empty `GameObject`** whose transform position can be referenced in code, allowing for quick level iteration.
- The process of **creating environmental walls** from cubes and organizing them under an empty `GameObject`.
- The benefit of **prefabs** is demonstrated by creating a "trap" prefab, where changing one instance updates all of them, acting as a template for reusable objects.
- Basic **camera following logic** is implemented by calculating an offset from the player's position and applying it in `LateUpdate`.
- Finally, the tutorial concludes by showing how to add a "win" condition where the ball is launched when it hits a "goal" trigger.

The information you gave me, give me the links you got this information from.

Based on the information provided, here are the three YouTube sources the excerpts were taken from:

- **"How to Make a Game - Unity Beginner Tutorial"** uploaded on the YouTube channel **"Jason Weimann (GameDev)"**.
- **"Learn Unity Beginner/Intermediate 2025 (FREE COMPLETE Course - Unity Tutorial)"** uploaded on the YouTube channel **"Code Monkey"**.
- **"Unity Full Beginner Tutorial | Make Your First Game"** uploaded on the YouTube channel **"Sorcerer"**.

You can find these videos by searching for their exact titles and channel names on YouTube.

Can you list my sources?

Certainly, I can list the sources you provided.

The information I've used to respond to your queries comes from the following three YouTube video transcripts:

- **"How to Make a Game - Unity Beginner Tutorial"** uploaded on the YouTube channel **"Jason Weimann (GameDev)"**.
- **"Learn Unity Beginner/Intermediate 2025 (FREE COMPLETE Course - Unity Tutorial)"** uploaded on the YouTube channel **"Code Monkey"**.
- **"Unity Full Beginner Tutorial | Make Your First Game"** uploaded on the YouTube channel **"Sorcerer"**.

Never mind I change my mind I'm going to do my project on Godot

Okay, I understand you've changed your mind and decided to do your project on **Godot**.

Based on our conversation history and the sources you provided, all the information I have is specifically focused on **Unity** game development. The tutorials and courses I referenced, such as "How to Make a Game - Unity Beginner Tutorial" by Jason Weimann, "Learn Unity Beginner/Intermediate 2025" by Code Monkey, and "Unity Full Beginner Tutorial | Make Your First Game" by Sorcerer, are all designed to teach game development within the Unity engine,

covering its tools, interface, scripting in C#, and various Unity-specific systems like physics (Rigidbody2D, BoxCollider2D), animation (Animator, Animation window), scene management, UI (World canvases), and camera control (Cinemachine).

Unfortunately, the provided sources do not contain any information about the Godot engine. Therefore, I cannot offer direct advice or relevant resources for developing a game using Godot based on this material.

3 sources

NotebookLM can be inaccurate; please double check its responses.

Reply ready.