

# Introduction to Unity and VRChat Creator Companion

<https://www.youtube.com/embed/uLi52YrrDmY?t=25s>

Have you ever wanted to make your own VRChat world but never knew how to start?

Creating an interior in Unity isn't as hard as you might think!

This beginner course covers everything one should need to know when building a scene, including many best practices. It will be accessible to everyone, including those who have never touched 3D before. We will only be using one software: Unity; this way, you will be able to create scenes fast and efficiently without having to worry about learning or exporting from completely different software such as Blender.

First you have to install the Unity hub, which makes it easy to work with multiple versions of Unity; so when it inevitably updates, the hub makes it easier to download new versions and upgrade or migrate projects. It is also a requirement for creating VRChat projects.

If you're just using Unity without VRChat, it's preferable to download the latest LTS version of Unity. Otherwise, skip that and download the VRChat Creator Companion (either the download page after you login to the VRChat website or from the direct link in the description), which will install the correct version of Unity for you if you have the Unity Hub installed.

In the Unity Hub you'll need to log into your Unity account, or create one if you don't have one already. You need a license to use the software but the personal license is free and perpetual. If you're having trouble with this part, then in the preferences of Unity Hub go to Licenses to fix it. I only have a professional license because I'm using the student version of Unity.

If you're not creating a VRChat project go ahead and create a new Unity project. URP is preferable for most people starting new projects but this series will cover the Built-in render pipeline (because that's what VRChat uses) which only has small differences, mostly in the materials and shaders. I'll do my best to point out these differences when they come up. I would not recommend starting with the high definition render pipeline because that one is significantly more complex.

If you're making a VRChat project, the first thing you need is (not a Steam or Oculus account but) a VRChat account of at least New User rank, the blue one, otherwise you won't be able to upload anything!

After you've gotten the Unity Hub and Creator Companion installed, create a new UdonSharp project. This is a world project that allows for C# scripts that use VRChat's Udon programming language. Even if you're not gonna touch it yourself, it will allow you to import a bunch of cool toys the community has created; I'll touch on that in a future video.

Here's a tip: I recommend putting all of your Unity projects in a folder on your larger hard drive if you have multiple, because they can get pretty big!

Once all that is finished, Unity will take a bit to set everything up and then it will open! You should now see the default layout of Unity when you first create a project.

If you want to change Unity between light and dark modes, you can go to Edit>Preferences>General>Editor Theme.

The largest and most prominent window is the scene view; this is a 3D view of your Unity scene. You can use the mouse wheel to zoom in and out. You can use the middle mouse button to pan (or press Q to use the left mouse button instead), and if you hold ALT and left click you can pivot. You can look around by holding the right mouse button and moving your mouse, and while holding, you can also navigate the scene by using the WASD keys, as well as E and Q for ascending and descending.

There should be two objects in the default scene. You can see them in the Hierarchy tab, which is on the left side of the screen by default. If you click on the Main Camera, it will highlight in the editor, a 3-axis gizmo will appear on the object in the scene tab, and information will appear in the rightmost tab, the Inspector. The inspector shows all the information of an object, and each collapsable section is called a component. Every object will have a transform component, which denotes its X Y and Z position, rotation and scale. If you edit the value in the Inspector, the object updates in the Scene view, and vice versa.

Understanding gameobjects is key to understanding Unity. In Unity, everything is a gameobject, and every gameobject can be filled with components to make things happen. Put another way, gameobjects are essentially just containers for different components which can actually do things. For example, the main camera is not a camera object; it is a gameobject with a camera component. The directional light is not a light object; it is a gameobject with a light component. A character in a video game is not a character object but a gameobject with a skinned mesh renderer component and a character controller component and a bunch of other components and scripts that make it behave like a playable video game character.

Only when you start to think in components can you truly master Unity.

At the bottom of the screen should be the Project view, which shows all of the game assets in the project, many of which can be dragged into the scene.

Unity scenes are like separate levels or worlds: we can create multiple different scenes in a single Unity project but a VRChat world is confined to a single scene.

You can rearrange these panels however you want. For the sake of conformity and ease of learning, I will be keeping the default layout for this series unless I explicitly open a new tab

required for the course, which I will make clear. If you ever want to reset to the default layout, you can simply click the dropdown in the top right corner which should say Layout and select Default, or Window>Layouts>Default.

Now we will set up a test scene. First, we need to add a ground plane for the player to stand on. To add an object, go to the Hierarchy, right click and select 3D object>Plane (alternatively you can use the GameObject menu up top). Press F in the Scene tab with the object selected to snap to it. By default, there should be a gizmo on it in the center of the object that has 3 arrows; you can change between the gizmos with the keys W, E, R, T and Y. We want this plane to be in the center of our scene, also called the origin. To do this, make sure the plane is selected, go to the Inspector, and under the Transform component, click the 3 vertical dots, and click reset. This will reset the XYZ position and rotation of the object to 0,0,0 and the scale to 1,1,1.

If you are just making an environment in Unity and not VRChat, you can import and drag in your character controller of choice, such as Unity's first person controller template.

The rest of this lesson will be to ensure that it's possible to upload working worlds to VRChat.

If you made this project with the VRChat creator companion then you will see a menu in the menu bar called VRChat SDK. Before touching that go to the Project tab, instead of assets, click right below it where it says "packages", click the search bar and type "VRCWorld".

**REMEMBER!** This is in PACKAGES, not ASSETS!

Drag VRCWorld.prefab into the Hierarchy. Reset the transform to the world origin if it isn't there already.

Now, go to VRChat SDK>Show Control Panel, and a new window should pop up, which you can dock if you like. Log in and then select the build tab.

Under Local Testing, check force non VR. VRChat should launch, and if everything goes well then you should spawn in the scene on top of the plane!

I would recommend closing the VRCSDK window when not in use, as it has been known to slow Unity down. If the window is undocked, you can close it by clicking the x at the top right corner, or you can right click on the tab and select Close Tab.

VRWorldToolkit gives us a bunch of tools to help make our world more optimized and ensure it isn't broken; import it from the VRChat Creator Companion.

If you need help throughout this series, I would recommend asking in Unity's forums or the official VRChat Discord as you're way more likely to get a response (if you ask nicely and post relevant screenshots). Besides, the comments section isn't designed for answering technical questions.

That's it for this lesson, in the next one we'll start building our room!

