

Basics of Unity



Abstract

Unity is a popular video game engine that is widely used in educational contexts due to its simplicity and stability. For these reasons, the projects of this course have been implemented in the Unity engine. While Unity is a much simpler engine when compared to other physics simulators available, it still features a wide variety of advanced capabilities which make it more than suitable for the vast majority of potential tasks.

This document will not offer a deep dive into many advanced Unity topics, and will instead focus on offering new users quick insights into how they can interact with the Unity Editor while working on their projects. All of the projects may be about programming control code, however students will likely find that learning a few tricks in the Unity Editor will help them enormously when it comes to debugging their projects.

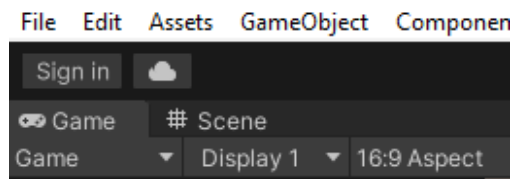
Editor Layout

Consider the following screenshot of the Unity Editor which has been overlaid with labels:



Thus we can see that the layout of the Unity Editor is divided into five sections, which can be described as follows:

- **Menu Bar:** A typical menu bar like you would see in most applications. Only the “File” and “Edit” buttons have the potential to be used in these projects.
- **Render Windows:** A section of the Editor which displays 3D renderings of the environment (called *the scene*) that the Ego vehicle is moving through. This section is composed of two windows. The *scene window* shows what the environment looks like if it were viewed by an outside observer. You can move through and interact with the environment using this window at any time. The *game window* shows what the camera on the Ego vehicle is seeing, but only when the simulation is running (before the simulation begins, the window will be empty).
 - **Note on window positions for newly installed Unity instances:** Sometimes when Unity is newly installed on a computer, it doesn’t separate the game and scene windows like in the above screenshot. Instead, Unity will only display one window at a time, and you must click on the window name in the shared window toolbar, as shown below:



To fix this problem, click on the “Game” window or “Scene” window tabs, then drag the tab away from the toolbar while holding down the mouse button. This will cause the window you grabbed to pop out of the Editor.

Move your mouse around a bit until it pops back into a new position such that both the game and scene windows can be seen simultaneously. You don’t have to do this, but it is *highly* recommended that you do.

- **Hierarchy:** The hierarchy displays a list of all the entities which are active in the scene. You will not need to interact with this section unless you are using debugging methods as described in the Debugging Helper documentation.
- **Inspector:** This panel displays the properties of an entity in Unity. To see the properties of an entity, click on the name of the entity in Hierarchy. You will not need to interact with this section unless you are using debugging methods as described in the Debugging Helper documentation.
- **Console Window:** The console window is where information can be printed by your code during the debugging process. The console window will also warn you if your code can’t compile, or if there is a runtime error caused by a bug in your code.
 - **Note on the ‘Project Window’:** There is a “Project” tab next to the console window. If you click it, you will see Unity’s default file explorer. This window is not required.

Navigating With the Scene Window

The Scene Window in the rendering section of the layout is one of the best tools at your disposal for understanding what is going on in the environment. As such it is recommended that you familiarize yourself with some of the basic controls for navigation in the Scene Window.

First, you will need to open a scene by clicking on “File” in the menu bar, then clicking on “Open Scene”, and then selecting the scene you want to open. This is described in more detail in the Quick Start Guide which is located on the Github page for the AVL (or inside the README.md text file).

Once a scene is open, you can perform the following navigation tasks using their associated controls:

- **Center an Object**
 - To center the Scene Window on an object in the environment (such as the Ego vehicle), click on this object in either the Scene Window or in the Hierarchy, and then press the “F” button.
- **Pan**
 - To pan the Scene Window, click and drag the center mouse button.
- **Orbit**
 - To orbit the Scene Window around it’s focal point, press the “Alt” button on the keyboard, and then click and drag the left mouse button.
 - Whenever you center on an object with “F”, the focal point becomes the object.
 - Whenever you pan the focal point moves as well.
- **Zoom**
 - To zoom in or out from the focal point, scroll with the center mouse button.
- **Undo**
 - If you accidentally change the scene via the hierarchy and want to undo your changes, use Ctrl + Z.

There are many ways to navigate around the Unity environment – the prior list is simply some of the most basic methods. One strategy to consider for moving to look at a specific destination is as follows:

- Look either left or right of your desired location using orbit such that the view of the Scene Window is perpendicular to the necessary direction of travel.
- Pan either left or right (depending on which direction your looked), and move the view of the Scene Window until the desired destination is seen.

This is just one of hundreds of navigation strategies. You will likely discover some of your own as you continue interacting with Unity.

Advanced Topic: Time Scaling

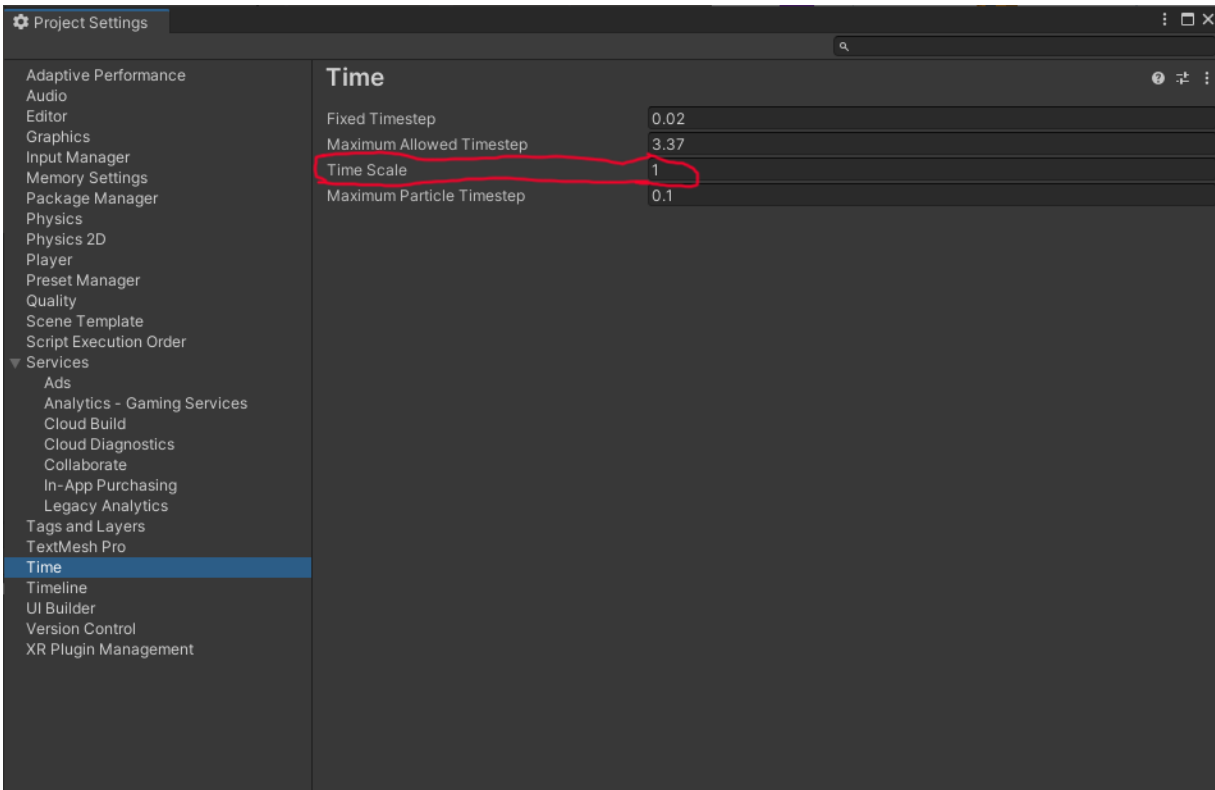
Time Scaling is an advanced topic in Unity which is not required for you to complete your project. However, it is a very useful concept which may speed up development of your project significantly if your control algorithms use strategies based around low-speed and high-precision.

The Unity Editor has a value called the “Time Scale” which influences how fast Unity is processing the simulation. When this Time Scale value is increased, the simulation runs much faster. Furthermore, this

increase does not come at the cost of physics accuracy – all objects are still moving at the same speed from Unity’s perspective, the simulator is simply running through the processing steps faster.

To change the Time Scale value, click on “Edit” in the tool bar, and then “Project Settings”, which will cause a window to appear. Within the Project Settings window, click on the “Time” menu option located in the left column.

Your Project Settings window should look like this:



You can thus modify the Time Scale value, which is highlighted in the above screenshot. A value of “1” causes the simulator to run in real-time, while a value of “2” causes the simulator to run at twice the speed of real time.