

HO1: Setup and introduction to Unity

Objectives

- [Set up Git and GitHub](#) and become comfortable with their basic functionalities:
 - Creating a repository
 - Ignoring files
 - Committing changes locally
 - Pushing your commits to a remote (GitHub)
- Install the latest [Unity](#) LTS version (Unity 6)
- *Access the [Unity student plan](#) (optional)*
- Install an IDE ([Rider](#), [Visual Studio](#), or [VSCode](#))
- Install [Meta Quest Developer Hub](#)
- Complete missions 1 to 4 from the [Unity Essentials Pathway](#)

Introduction

Welcome to the Virtual Reality class!

Over the next five weeks, you'll complete hands-on exercises to learn VR development in Unity. These exercises are designed to be compatible with any OS; if you encounter issues, please let us know.

Please note that these exercises are not meant to be fully guided at every step. For instance, this week provides tool setup instructions and download links, but you'll need to install and familiarize yourself with them on your own. We expect you to utilize online documentation as needed, and you can always ask questions during exercise sessions.

It is crucial to get comfortable with Git, your IDE, and Unity as quickly as possible since upcoming weeks will build incrementally on these base concepts.

Set up Git and GitHub

In software and game development, version control is essential for tracking changes to your code and collaborating efficiently with others. Version control systems, like Git, keep a history of all changes made to a project, allowing developers to revert to previous versions, understand the evolution of the codebase, and avoid losing valuable work. Learning version control early will be invaluable throughout this course and in any collaborative development environment, making it easier to manage projects, collaborate effectively, and keep your work organized and safe.

In this class, we require you using Git, the most popular version control system. You must become familiar with Git and GitHub because you will submit your project as a GitHub repository.

If you're new to Git, we recommend using online resources, starting with GitHub's official [Getting started with Git](#) tutorial, especially the [Set up Git](#) section. If you're unfamiliar with Git, you may want to use [Github Desktop](#) as a user-friendly interface.

By the end of this week, you should be confident in:

- Creating a repository
- Ignoring some files
- Committing your changes
- Pushing your commits to a remote (GitHub)

For practice, we recommend creating a new repository, pushing it to GitHub, and adding a simple Unity project with just a cube. Note that Unity projects are particular about version control: if multiple users modify the same prefab or scene, irrecoverable conflicts may arise due to Unity's file format. To avoid this, team members should modify their own scenes/prefabs, and then one person can manually combine them into a final scene/prefab if needed.

A [standard .gitignore](#) file exists for Unity projects; you should use it to keep your repository clean. Additionally, since Unity projects can be large, you might need [Git LFS \(Large File Storage\)](#) to manage large files efficiently.

Install Unity

To create your VR projects, you will use Unity, a game engine offering a wide range of pre-built features and one of the simplest workflows for VR development.

Follow the installation instructions [here](#) and ensure you select the latest Unity LTS version (Unity 6). We cannot provide support for any other versions of Unity. Make sure to install the [Android Build Support](#) module as well.

For additional free 3D assets, consider signing up for the [Unity student plan](#).

Install an IDE

We suggest using [Rider](#) for its seamless integration with Unity. As an EPFL student, you can access Rider for free.

Alternatively, you can use Visual Studio or Visual Studio Code, though please note that support for these IDEs will not be provided.

Install Meta Quest Developer Hub (MQDH)

Meta Quest Developer Hub (MQDH) is an application that allows developers to manage their devices, debug their apps, and upload their apps on their headset. Other alternatives exist but we recommend using this official tool.

- Create **one Meta Developer account** for your team.
Note: you may need to verify your account by enabling Two-Factor Authentication using your phone number
- Download and install the **Meta Quest Developer Hub**.
 - You may be asked to set your ADB path. Set the one from the Unity version you downloaded earlier.
 - Log in using your Meta Developer credentials.

Once you receive your headset, you'll be able to upload your game directly from here.

Introduction to Unity

To get started with Unity, we recommend completing missions 1 to 4 of the official **Unity Essentials Pathway**.

Don't hesitate to look up additional tutorials online.

Note: Make sure you understand the core concepts but do not spend too much time going through every video. You'll get other chances to use and practice the basics of Unity and you can always go back to some tutorials later if needed.