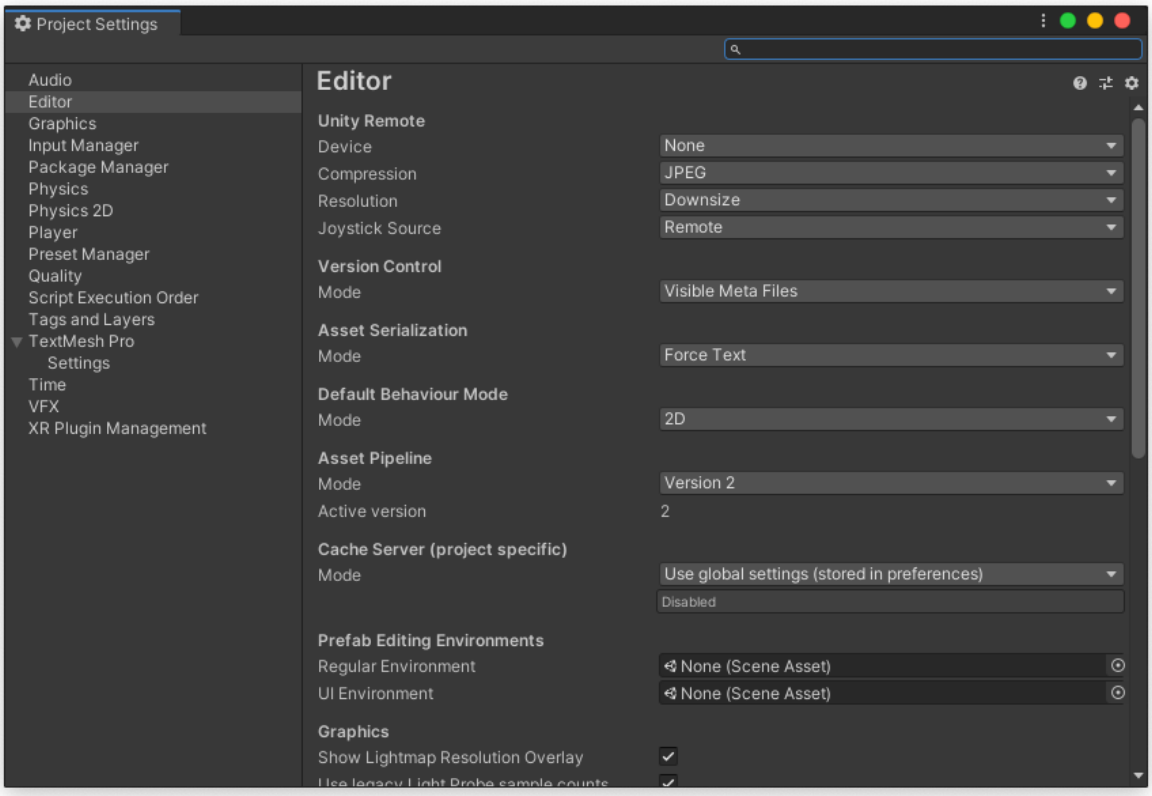
Unity Development Setup Tutorial

Basic Project Setup for Git

Unity projects can be extremely complicated, and are often only created with the Unity editor in mind. However, there are a few tweaks that we can make that allow a multi-person team to maintain a VCS for a project.

Configure Unity Settings

We need to ensure the project is configured to expose all files and serialize objects in a way that Git can understand. To do this, go to `Edit → Project Settings`, and enable `Visible Meta Files` and `Force Text`, seen in the screenshot below. Additionally, changing line endings to `Unix` will allow consistency if your team works across platforms.



Make sure to save the project for these settings to take effect.

Ignore unnecessary files

I recommend using the github/gitignore repo to create a gitignore for your projects, downloading the `Unity.gitignore` file, along with any playform specific ignores

- https://github.com/github/gitignore/blob/master/Unity.gitignore (Required)
- https://github.com/github/gitignore/blob/master/Global/Windows.gitignore (Recommended)
- https://github.com/github/gitignore/blob/master/Global/macOS.gitignore (Recommended)
- https://github.com/github/gitignore/blob/master/Global/JetBrains.gitignore (Optional)
- https://github.com/github/gitignore/blob/master/Global/VisualStudioCode.gitignore (Optional)
- etc...

Add the contents of as many of these files as necessary to the project's `.gitignore` file to prevent yourself from adding bloat and potentially breaking files to the repository.

```
$ curl https://raw.githubusercontent.com/github/gitignore/master/Unity.gitignore > .gitignore
$ curl https://raw.githubusercontent.com/github/gitignore/master/Global/macOS.gitignore >> .gitignore
```

Git LFS

Git Large File Storage (LFS) will allow large files in your repo to be stored separately, reducing the size of repos and allowing faster cloning and fetching.

> 💡 Note that GitHub only offers 1GB of free LFS storage, whereas GitLab offers 10GB

Once set up, run `git lfs install`, then you can either individually track large files with `git fls track "*.file"`, or add files to the `.gitattributes` file manually.

Below is a recommended `gitattributes` file to use, including all binary files that Unity supports, and setting up some defaults for how GitHub shows diffs.

> https://gist.github.com/furgoose/137b69c18a8410f31a3f83e24764ff4f

Ensure that you've added `.gitattributes` to your repo, then all files that are added will automatically be managed using LFS, which you'll be able to recognise with the following in GitHub/GitLab

GitHub

GitLab

LFS Locking

Since version 2.0, LFS allows for locking of files (providing the server supports it), which provides a vital tool in preventing painful merges of Unity scenes etc.

```
$ git lfs lock Assets/Scenes/Tutorial.scene
# and
$ git lfs unlock Assets/Scenes/Tutorial.scene
```

GitHub for Unity Plugin

If using GitHub, there is an official plugin that can be dowloaded from the Asset Store or from https://unity.github.com. This willl offer helpful features to manage your repo from inside the Unity editor, including making commits, viewing locked files, etc.

Additional Resources

Working as a team

Resolving conflicts in Unity scenes with git and UnityYAMLMerge

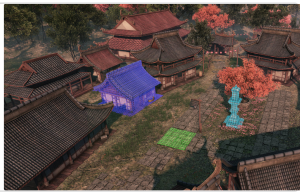
Tulenber · 1 May, 2020 · Intermediate · 8 min · 2019.3.11f1 · Since I did not manage to commit before my comrade, it's time to fulfill the promise given in the article about configuring git to work with Unity and figure out how to use UnityYAMLMerge to resolve

https://kihontekina.dev/posts/git_conflicts/

How to effectively collaborate with your team in Unity - KinematicSoup Technologies Inc.

We frequently see questions on the Unity forums and Reddit asking how Unity users are meant to collaborate on projects as a team. While there is no 'one size fits all' answer to a subject as complicated as this, we hope to provide a place to start.

https://www.kinematicsoup.com/news/how-to-effectively-collaborate-in-unity



Multi-Scene editing

Multi Scene Editing allows you to have multiple open in the editor simultaneously, and makes it easier to manage scenes at runtime. The ability to have multiple scenes open in the editor allows you to create large streaming worlds and improves the workflow when

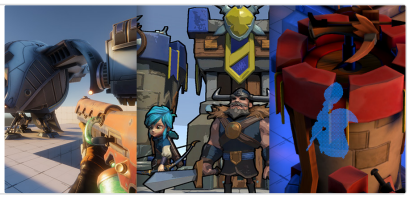
https://docs.unity3d.com/Manual/MultiSceneEditing.html

Advanced Unity techniques

Unity-Technologies/UniversalRenderingExamples

This project contains a collection of Custom Renderer examples. This will be updated as we refine the feature and add more options. Current support : 2019.3.0f6 or later with UniversalRP 7.2.0 Usage of the project Clone the repo/Download the zip down to your

https://github.com/Unity-Technologies/UniversalRenderingExamples

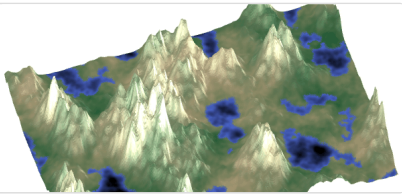


Advanced rendering techniques and examples

Making maps with noise functions

written Jul 2015, then updated Feb 2016, May 2017, Oct 2017, Jan 2018, Feb 2019, Jun 2019, Mar 2020, May 2020 One of the more popular pages on my site is about polygonal map generation [1]. Making those maps was a lot of work. I didn't start there.

https://www.redblobgames.com/maps/terrain-from-noise/

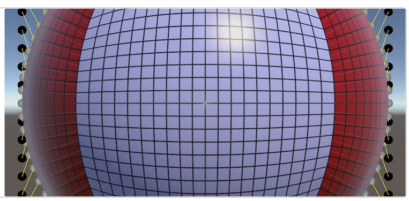


Procedurally generated terrain tutorial

Cube Sphere, a Unity C# Tutorial

Better Roundness Turn a cube into a sphere. Visualize the mapping in Unity. Critically examine the conversion. Use math to come up with a better approach. In this tutorial we'll create a sphere mesh based on a cube, then use mathematical reasoning to improve it. This

https://catlikecoding.com/unity/tutorials/cube-sphere/



Useful tutorial for creating deformable spheres in Unity

https://www.shadertoy.com/

Massive collection of shaders that can be adapted for use in Unity