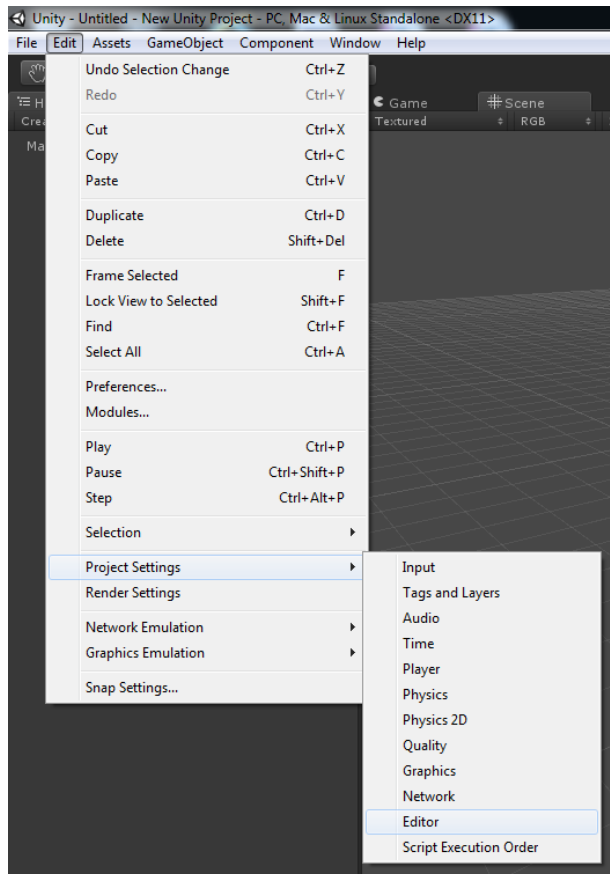


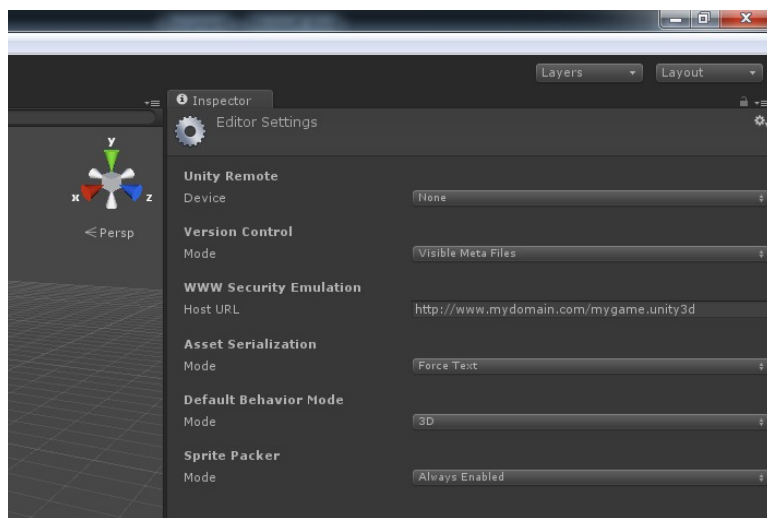
# Using GitHub with Unity

## Setting up a project

To make the Unity project function without script errors while using GitHub you have to change a few of the file settings in the project. First open the editor settings inspector as below.



There are two settings in Editor Settings that need to be changed to be able to use the project with GitHub. Version Control should be set to **Visible Meta Files** and Asset Serialization should be set to **Force Text**.



Meta files need to be set to visible so that GitHub recognises them in the project folder, without the Meta files you would lose all of the links between objects in the scene. Forcing the Asset Serialization to Text allows for the version control of the assets to be controlled by GitHub as it works on a line by line editing method that wouldn't be possible by using binary objects.

### Scenes

Version Control of Unity scenes while using GitHub is the main source of issues with the project, due to the way unity serializes its objects the scene files between two versions of the same project will be drastically different and impossible to merge properly.

For this reason below are a few pointers for using scenes without any issues.

- [ Have a master build scene that nobody works in but contains the most up to date **working** versions of all of your systems
- [ Have an individual scene for each of the developers that only they are working in.
- [ **Prefab everything!** Prefabs are the best way of ensuring that links between objects stay while syncing them over GitHub.
- [ When merging back in to the master build scene drag and drop your prefab objects in to the Hierarchy window, they should maintain all their object links.

These methods are not entirely error proof but they will help in preventing some of the most common errors created by using GitHub with Unity.

### Gitignore

The default Unity GitIgnore file provided by GitHub misses out a few crucial files that are known to cause errors with script and scene versioning. In the repo (where you found this document) is a file called 'unity\_gitignore.txt', this is more complete version of the GitIgnore file. Replace the existing one with the contents of this file (or if it doesn't already exist, create a file named **.gitignore** in your main repo directory and paste the ignore details in to the file) and commit it.

These methods are not flawless but they have fixed a lot of issues that have been experienced while using Unity and GitHub together.