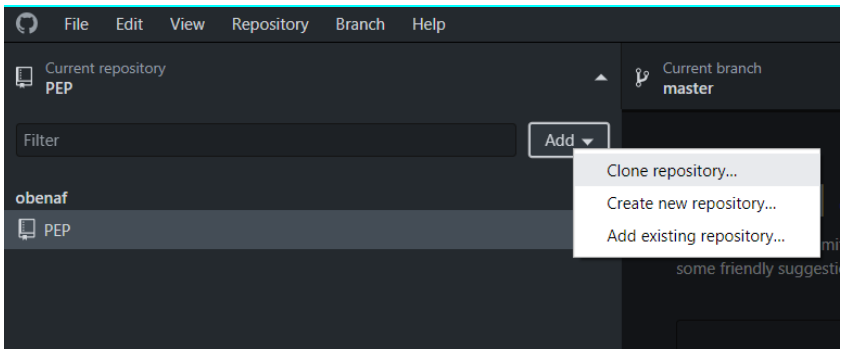
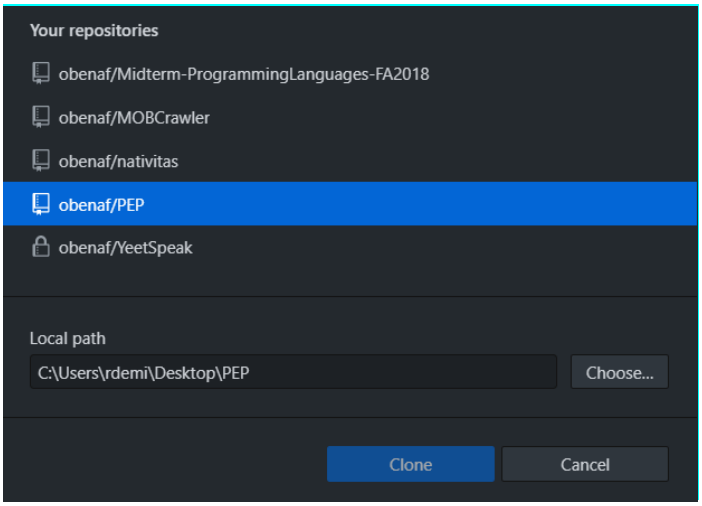
Git Guide, Using GitHub

Installation:

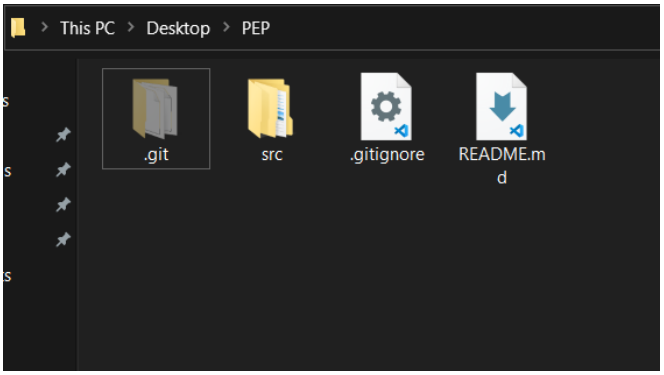
To get GitHub up and running, first download the GitHub app here. It is compatible with Windows, Mac, and Linux. Once running, sign into your GitHub account. Then click the top left banner titled current repository, then select to clone a repository as shown.



You can then choose from a list of active repositories, or add it by the url of the repositories webpage. Then select a path for the files to be stored. This folder and its children will be the files updated by git.

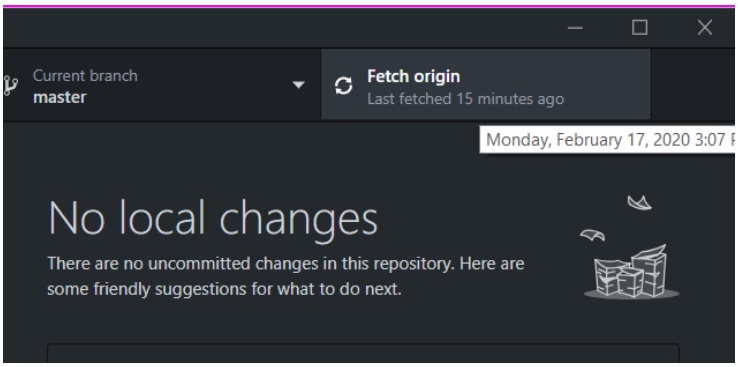


Any file you or another user add to this folder will be added to the project. The .gitignore contains a list of file extensions such as editor preferences that will not be added or changed on the project.



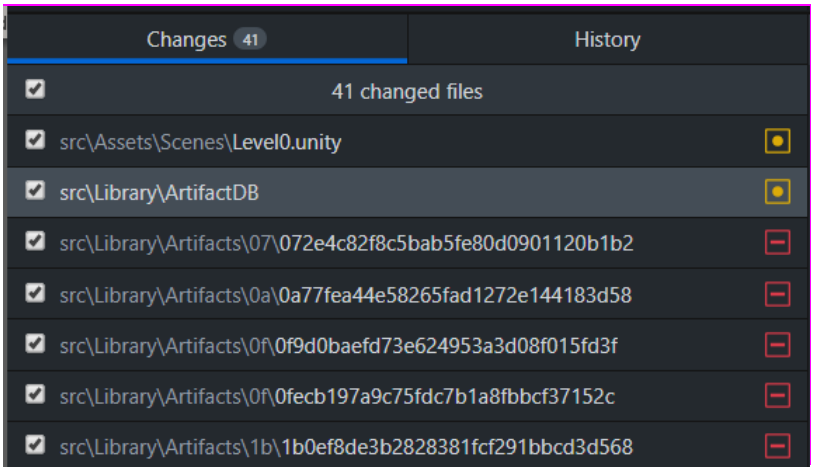
Workflow:

We will be using a linear workflow and will not be using branches. Instead each user will have a named folder for their work, that will communicate with other members code.

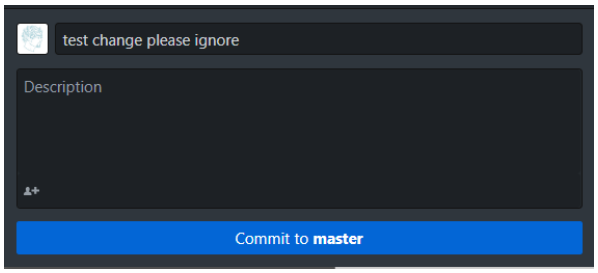


Before making changes, always fetch the origin. This syncs your folder with the most up to date project folder. This ensures that you don't work on something that may have already been changed or modified.

Then, to start working simply open the project in Unity and begin working and coding! Make sure you push your changes often and after major updates so you don't lose work, and keep the project as recent as possible.

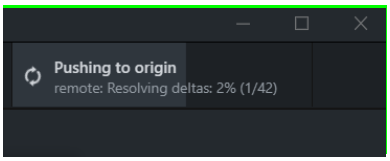


After making some changes, the changes window in the GitHub app will look something like this. Orange dots mean files that have been changed, red minus mean files that have been removed, and green squares mean new files. You can select each file and ensure that it has the appropriate changes made. Small edits in unity can cause many file changes, such as the removed Artifacts we see here. Don't worry, this is normal, especially when editing maps and project settings. Once everything looks good we are ready to push to the repo! **Warning: Make sure you edit Unity Editor before pushing! There are certain files only created and used when it is open that will not allow you to push.**



Then summarize your changes in the title, *this is required*, and add a description if you made significant or complex edits. Then simply press 'Commit to master'.

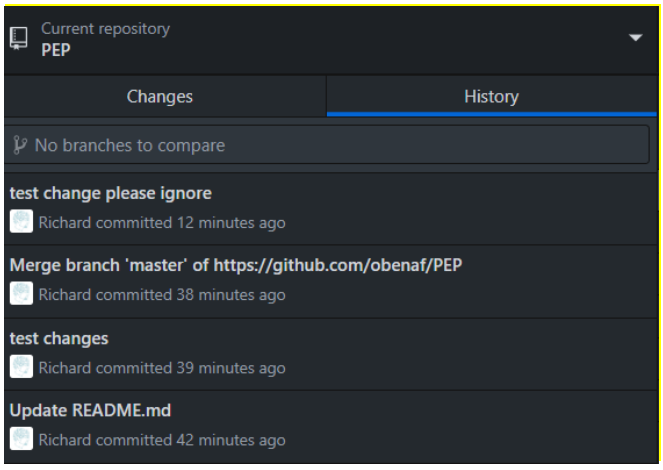
GitHub will then process your changes, if anyone else has been making changes to the same files you may have a **merge conflict**. Don't worry, this should rarely happen with our file system, but when it does you can open it in your favorite text editor and GitHub will show you what you changed and what they changed. You may need to reach out and discuss the differences with the other person. Sometimes there will be minor changes in project files such as the Artifacts we saw earlier. Generally, it is safe to choose to keep the 'master' changes. If you are worried, contact me or whomever made the change.



Then the same banner you used to pull new changes earlier will ask you to push to origin. This finalizes your changes and adds them to the GitHub repo. Congratulations, your changes are now live!

History and Recovery:

If something goes horribly wrong and the project is no longer working properly, don't worry! *Yet.* There are ways to revert to previous iterations.



Simply right click a previous commit, (that we know is stable) and click 'revert this commit'. This will change the project back to the previous iteration. If you aren't sure what you should revert to, reach out to the team and we can figure it out together.