Git and GitHub

Git is a *source code management system* (otherwise known as revision or version control). These systems are designed to solve a lot of the problems associated with working on large software projects in a team of developers.

You create a repository for your code (usually hosted on a server somewhere for large teams, or on your own machine if it's just you working on a project), and can *commit* code updates to the repository. Each time you commit some code to the repo you are saving a snapshot of the project at that time. You can revert back to previous revisions if desired, and any changes between revisions are tracked.

GitHub is an online platform based around Git. It was originally aimed at providing server space for open source projects. With a free GitHub account you can host and contribute to an unlimited number of public repositories. This means that anyone can see (and download) your code. You can also add any number of contributors (people with write access) to your repositories. For a monthly fee, you can also create private repositories, and GitHub offers an enterprise version for companies who wish to use GitHub internally.

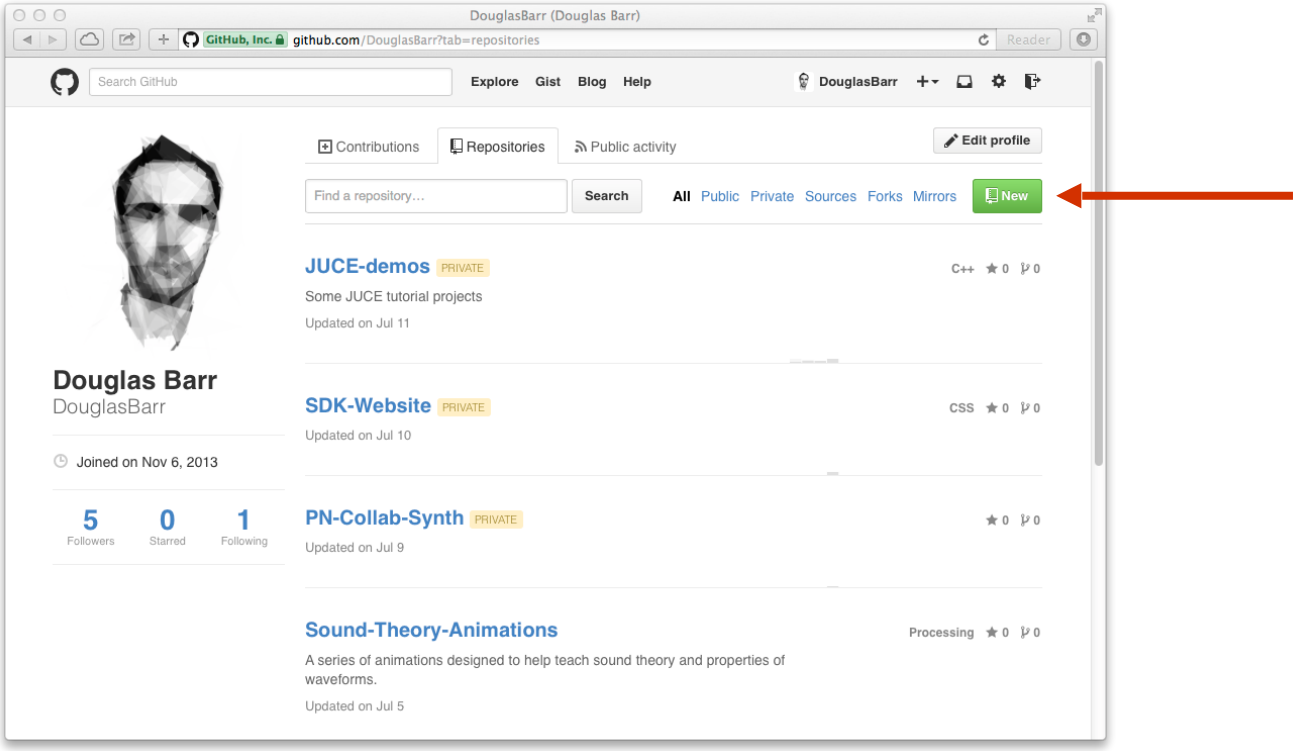
Most employers will expect you to know how to use some form of version control system. They will also expect to be able to see examples of your work, and see that you have contributed to collaborative projects. GitHub has become the expected platform to show off these areas. Going for a job without some kind of online portfolio will put you at a significant disadvantage compared to candidates who have an active, interesting GitHub account.

Setting up Git and GitHub

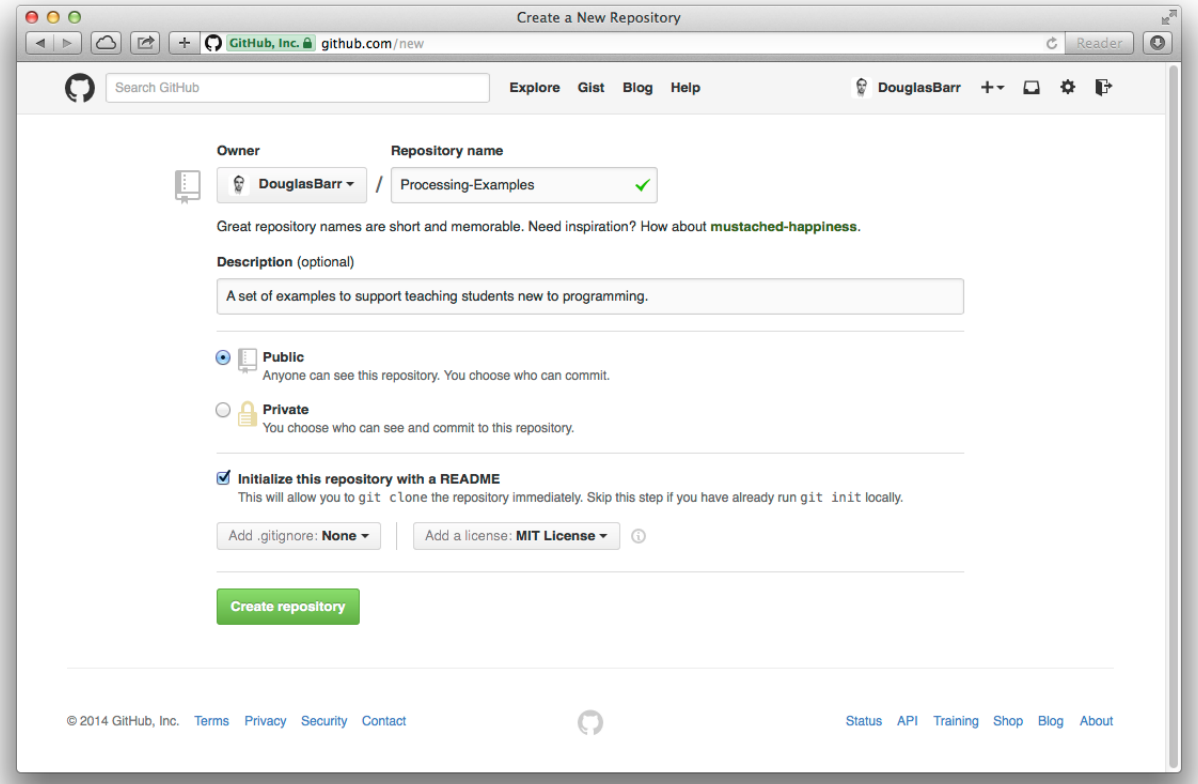
GitHub has an excellent set of tutorials hosted at https://help.github.com/articles/set-up-git. Following this should allow you to install Git and GitHub for Windows/Mac on your home computer. I would recommend using the client until you are familiar with Git, at which point you can start experimenting with the command line tools.

Git needs to be able to set up a secure connection with the GitHub servers. This sort of connection is not normally allowed on the college network, so a proxy server has been set up that will allow this. You will be provided with instructions on how to set this up

Once you have Git installed, you need to start creating repositories. You can create repositories locally, but if you haven't set up local access yet then you can create them in GitHub and then clone them to your computer. On your profile page just click the *New* button.

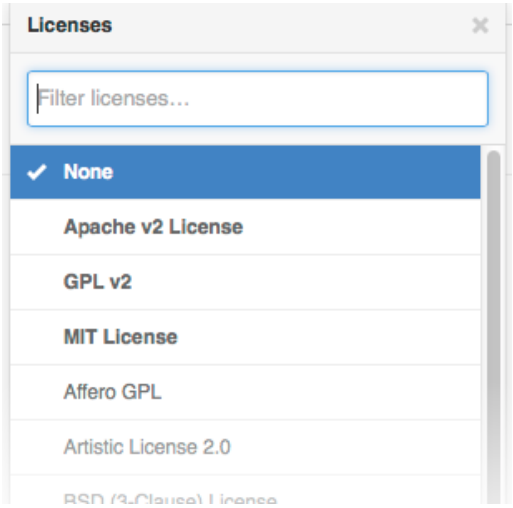


You should give your repo a relevant name and description, and should check the box to initialise the repo with a Readme file. If you have a free account your only privacy option will be public.



The Readme file is used to hold information on your project for people viewing it on GitHub. The contents of it will show up on the homepage for your repository. Readme files are written in Markdown, an increasingly popular markup language.

You should make sure that you are aware of the GitHub licensing policy, and choose an appropriate licence. There are a lot of licences available, and GitHub have created a website that will help you pick one.

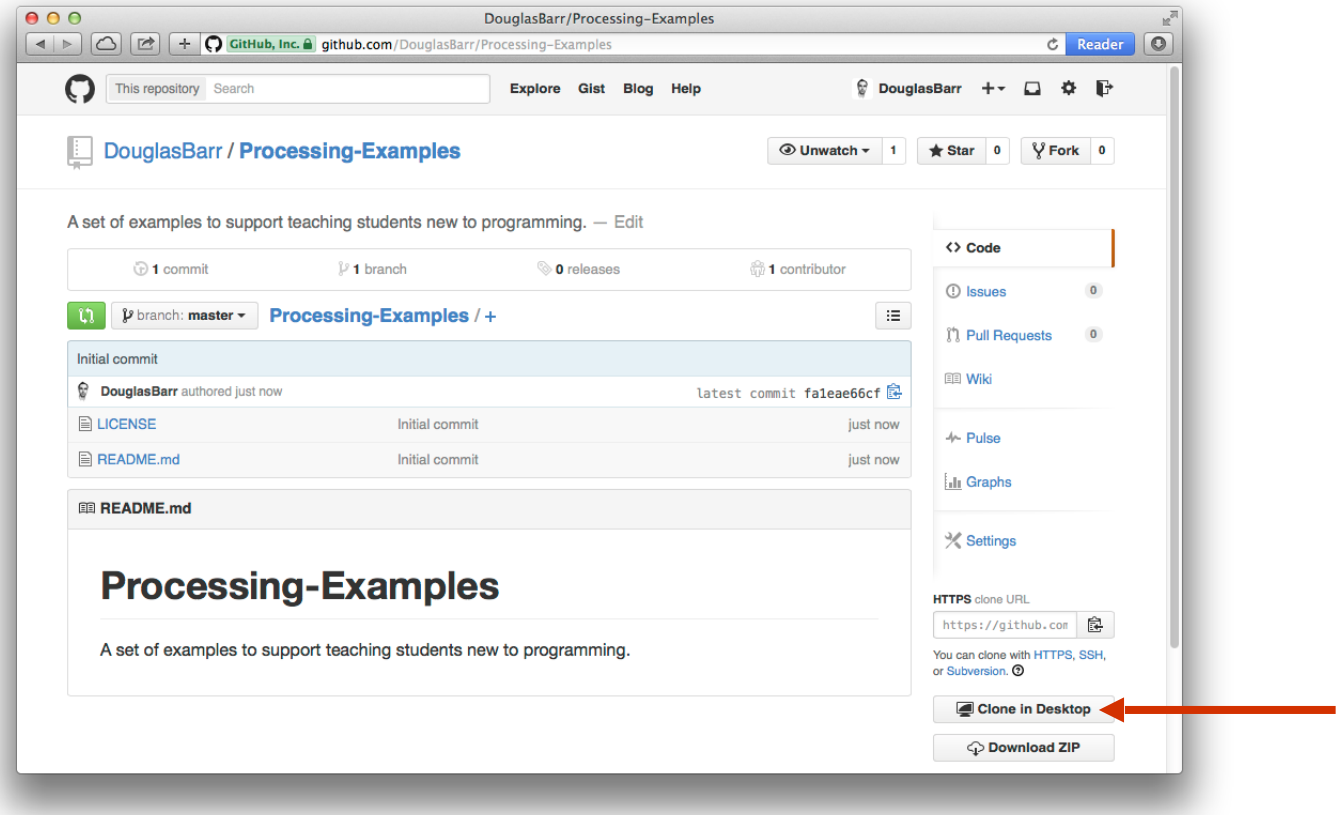


Unless you have very good reason not to, I'd suggest that you pick an open source license for your repos.

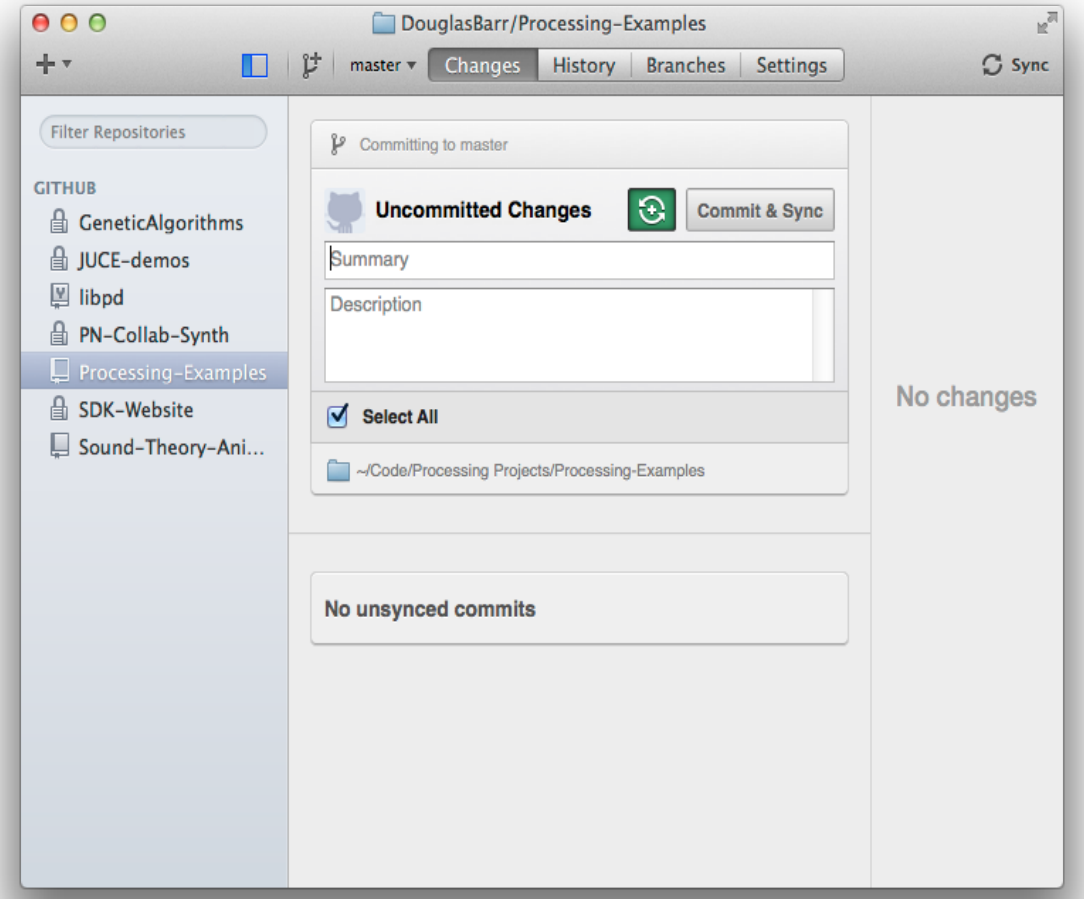
Firstly, because it's nice. And it's nice to be nice. Secondly, it shows people (and I mainly mean potential employers here) that you know about open source licences and that you have read enough to know the difference between the main licences available.

This might seem like a minor issue at this stage in your careers, but Intellectual Property is very important to most employers in software development. They want their employees to be as well read as possible in copyright/creative commons/open source issues. Being able to speak confidently and articulately about issues that affect the industry is always a good thing.

You should now have a repository. You can clone this in your desktop using the link in the sidebar.

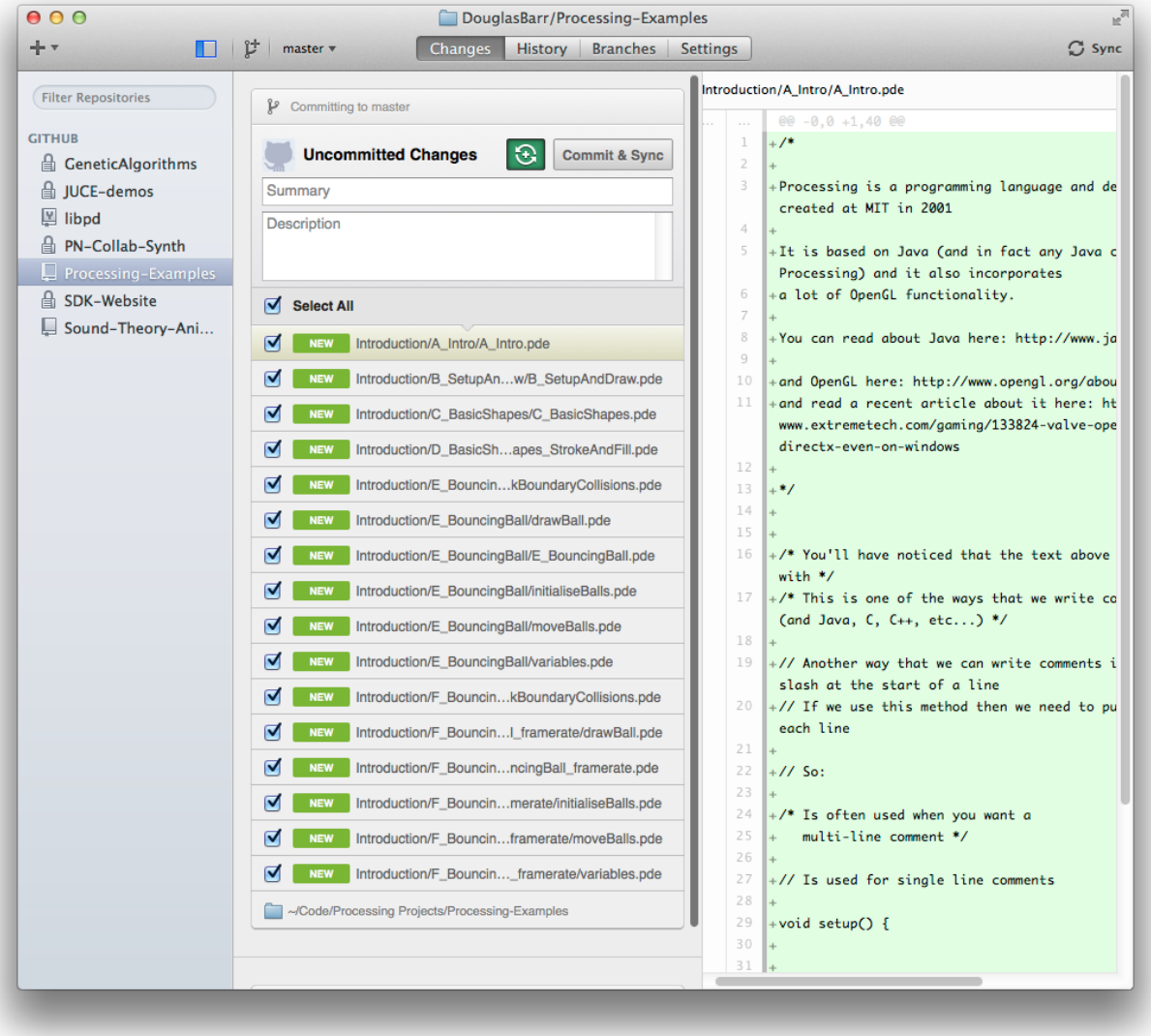


Clicking the *Clone in Desktop* button should automatically open your GitHub Desktop client.



Now that you have your Repo set up, you can make and commit changes to it. You can treat the local clone of your repository as you would any other folder.

Once you have made changes to your folder, you should see a list of *Uncommited Changes* in your GitHub client.



At this stage you can add a summary and description for the commit, and then click the commit and sync button. You should now see the committed changes on your online GitHub account.