Git Repository Guide

# Summary

Git is a sub versioning system, which saves all of our files and every previous version of those files so that we can backtrack to previous versions.

Everyone keeps a **working copy** (a **git clone**) of the files on their own computer. When you use git, it will first check for changes you have made to your own **working copy**, you can then **commit** to those changes to indicate that you want to apply them to the **master copy** on the server.

When you make a **commit**, you can write a brief description of what you have changed so that we have a record of who has been changing what. When you are ready to apply the changes you have made you **push** them to the server.

To update your **working copy** with all of the **commits**, which have been **pushed** by the other users, you use the **pull** function.

# Using a Git client

Git can be utilized via the command line but it is much easier to use a Git client. Funnily enough, many of the university computers, including the macs in the orchard do not have any git clients installed however you should be able to install these apps to your file store. Here is a list of clients: <http://git-scm.com/downloads/guis> I have been using SourceTree for mac.

If you are using the github client for Mac or Windows, the initial setup is as follows:

1. The program should prompt you for a username or password when you first start the program, if it doesn’t, go to the preferences and there will be a section for login information under accounts. Login using your github username and password.
2. You should see your account appear in the list on the left under “Github.com” select your account.
3. You should now see a list of the repositories that you can push to, select the repo labeled warslett/Walking-Tour-Project and click “clone to computer”.
4. Enter the directory that you want to clone to and click “clone”. You will now have a working copy of the project.
5. Double click on the project to view its history, you should see a list of all the recent commits to the project.

# Pulling down updates

Before you start work you should always pull down all new commits. In the github app you can do this by going to the repository menu and clicking pull.

# Committing changes

To publish changes you must first commit them and then push them. If you push before you have made commits then nothing will happen. When you have finished working on something, go back to the github app, click on the “Changes” tab and you will see a list of changes. Check the boxes next to the changes that you would like to include in the commit and write a brief description of what you have changed. When you are done, click “commit”.

It is not good practice to commit lots of unrelated changes together. For example, if you have been working on both android source code and documentation, it would make sense to have one commit for the documentation and one for the android so that if we needed to backtrack on the source code, you could do so without reversing the changes to the documentation.

# Pushing Commits

When you have committed to all of your changes you need to push them so that the master copy on the server gets updated. To do this, go to the repository menu and click “Push”.