Contributors screenshot
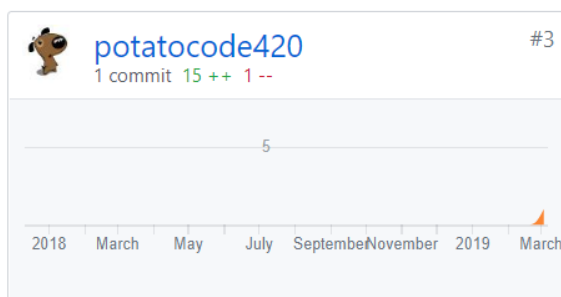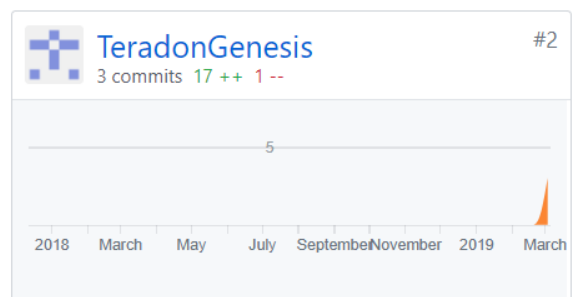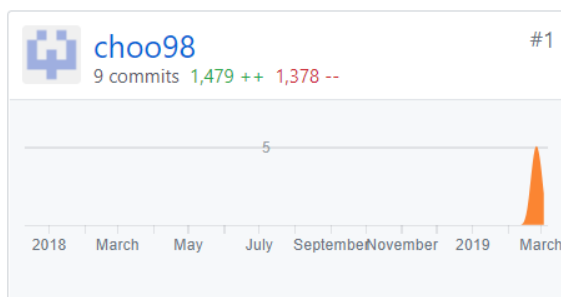
Figure(1): Choo98 is (me), TeradonGenesis is (Lim jia lok) and potatocode420 is (Marc)

Network screenshot
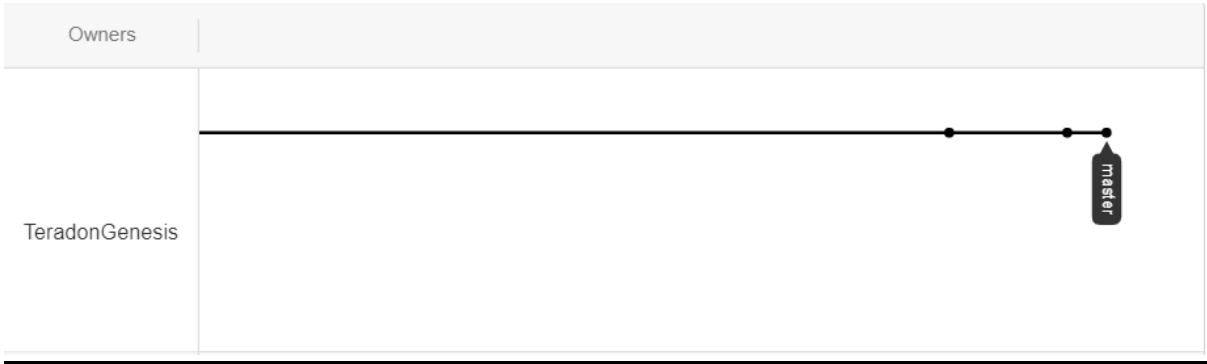
Figure(2): a screenshot of our network
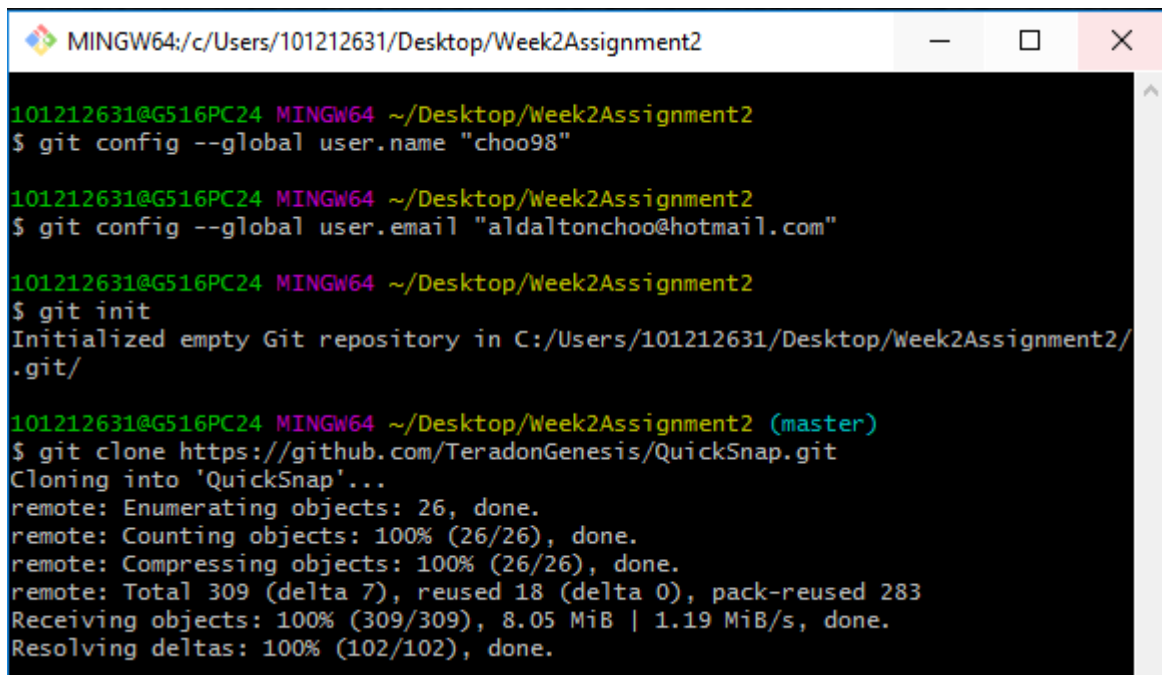
Overview process of using Git for revision control

Figure(3) Setting up the git repository

First we set the username and email by using the "git config –global user.name and user.email" command. Then we use the "git init" command to initialize the git in the specific folder that we want to create the project in. If we have an existing project in the Git Hub repository, we can use the "git clone [URL]" command to clone the project in the Git Hub repository into the specific folder as shown as the picture above.
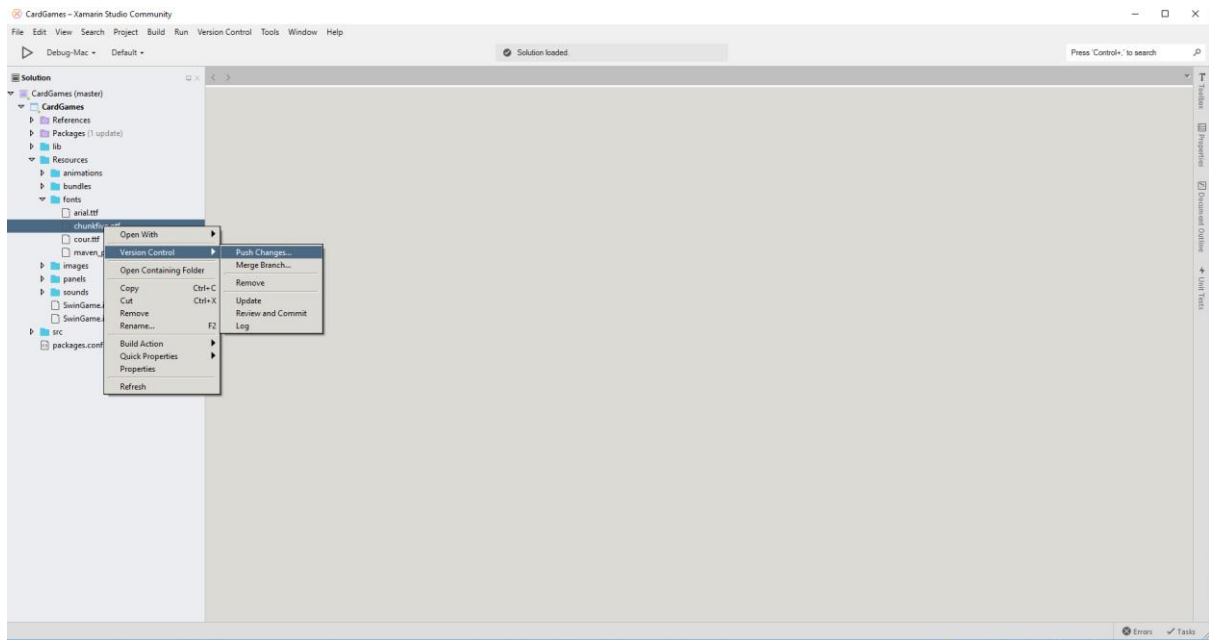
Figure(4) Pushing a single file Git Hub
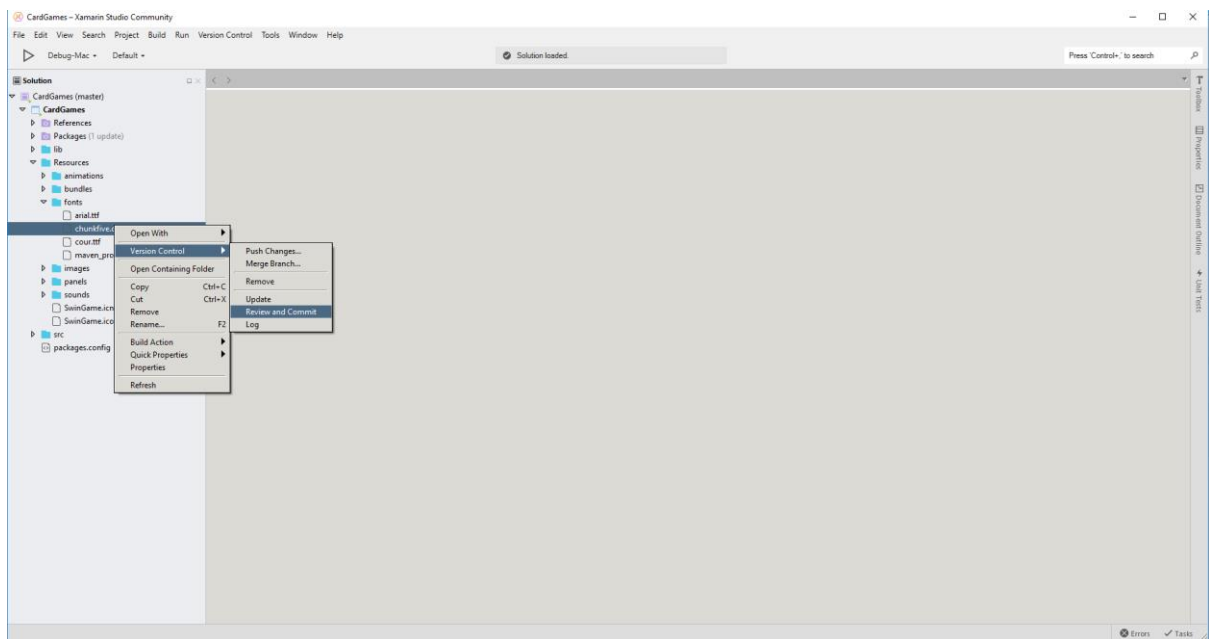
Figure(5): Reviewing the file and committing to Git Hub

If we want to push only a single file to the Git Hub repository, we right click on the specific file and click push changes as shown in figure 4. Next, we click review and commit so that the file will be uploaded to the Git Hub repository. The same process is repeated if we want to save our changes that we have done on the current project, but the only difference is clicking on the version control at the top. The process is shown in figure 6 and 7.
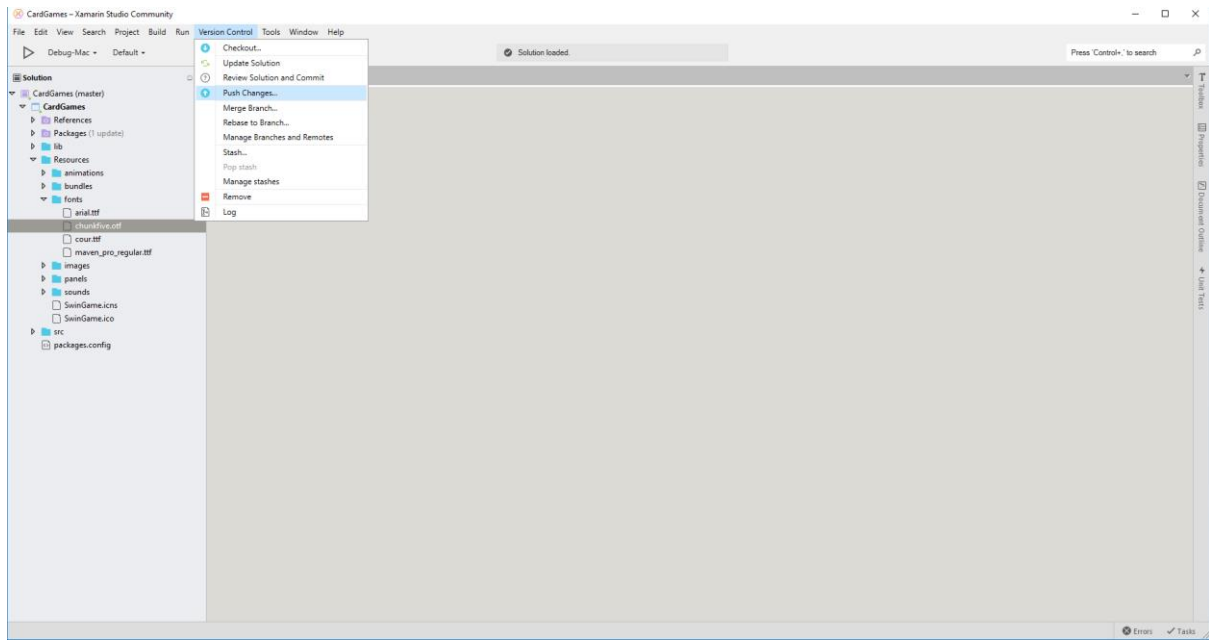
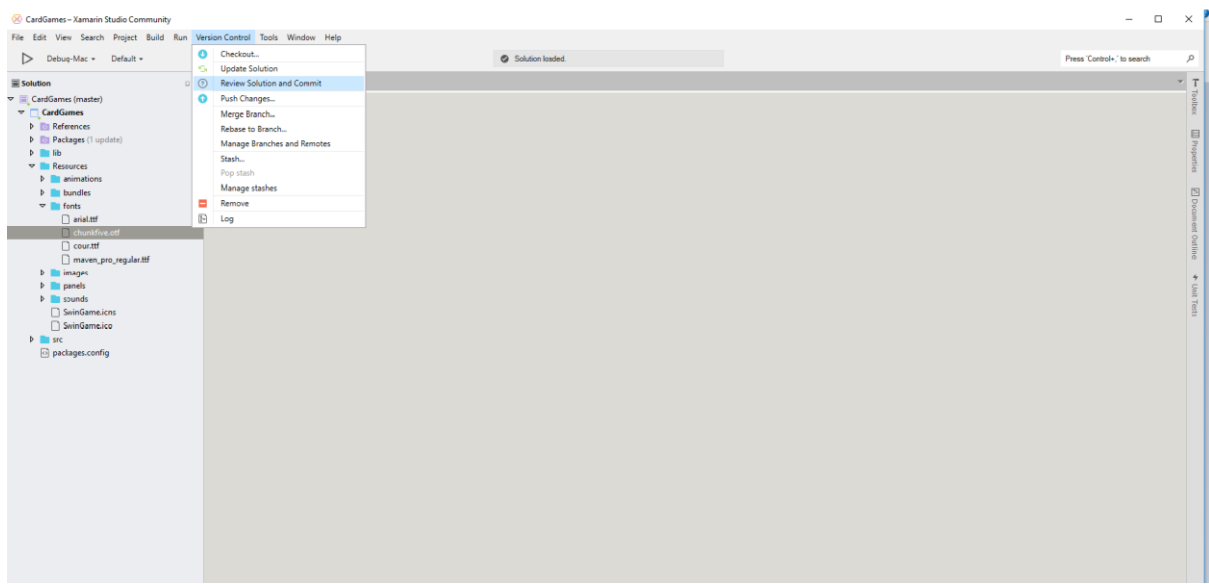Figure (6) pushing changes for whole file

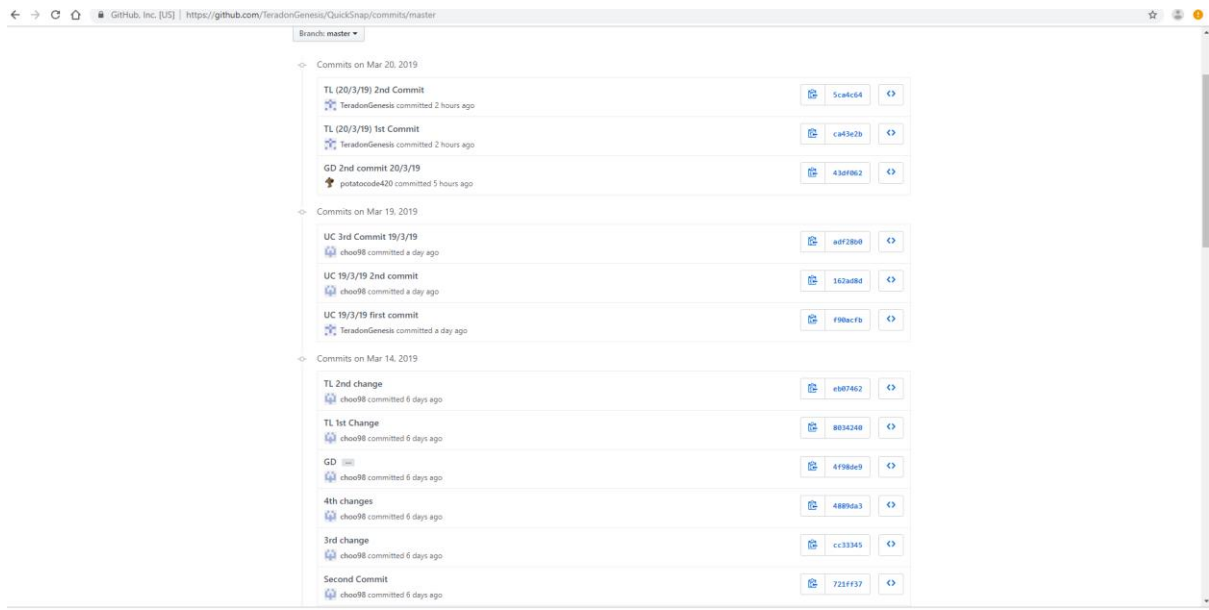Figure (7) Reviewing file before committing changes to Git Hub Repository

<u>Figure (8) reviewing commits history on Git Hub</u>

We can also review the history of commits that have been submitted through the Git Hub website. This makes back tracing much easier if there is anything wrong with the project.

Git Hub Reference Sheet

Setup

1) Git init: initialize an existing directory as a Git repository
2) Git clone [url]: retrieve a repository based on URL given.

Staging

1) Git status: shows status of modified files that is currently working on
2) Git add[file]: adds a file for the next commit.
3) Git commit -m "[Description]": commits your current project.

Branch and merging

1) Git branch: list out all your branches
2) Git branch [branch-name]: creates a new branch
3) Git merge [branch]: merges a specific branch into the current branch
4) Git log: shows all the commits in a branch's history.

Sharing and updating

1) Git remote add [alias] [url]: add a git url as an alias
2) Git fetch [alias]: fetches all branches based on the alias given
3) Git push [alias] [branch]: pushes local branch commits to the remote respiratory branch
4) Git pull: fetch all commits from remote repository branch.