What is Github?

Github is a platform where developers can host their code in places called **repositories**, which can be forked, modified, and contributed to in the true spirit of open source! Github is used for a variety of things, from simply storing code to even hosting websites. Repositories can be shared between teams in a secure manner, making source code management extremely easy. Github also facilitates community involvement, as **issues** can be used to track bugs and allow users to communicate with each others in a forum-like setting in order to solve their problems together.

Github was first developed in 2008 by Chris Wanstrath, P. J. Hyett, Tom Preston-Werner using Ruby on Rails, and currently has its headquarters in San Fransisco. Within a year, 46,000 repositories were created with many being forked and merged by other users to create their own projects. By 2011, Github was hosting 2 million repositories, more than SourceForge and Google Code, its main competitors at the time, and in 2013, Guthub finally hit a staggering 10 million repositories.

Github has plans for free, professional, and enterprise accounts, which range in pricing from free to a few dollars per month per user. Free accounts are most often associated with open source projects, which strive to make every part of their code free to the world, while professional and enterprise accounts are often used by businesses. All accounts can use unlimited numbers of both public and private repositories, but free plan accounts are limited to having only three collaborators on each private repository.

Besides hosting source code, Github can be used to create documentation for projects in the form of README.md files and Wikis, both of which provide creator-written information on how to utilize the project. It also tracks commits, merges, and pull requests, allowing a user to see the development of the code over time as well as to rollback the code if it stops functioning. This allows for quick and easy version control, making sure that the best possible code is publically available.

Github also has a static web hosting service referred to as Github Pages, which can be used to host blogs, documentation, or even entire books. The pages are stored under the domain .github.io, and Jekyll is used to serve the website. Another service provided by Github is Gist, which stores small snippets of code that are easily shared, and can become tiny projects of their own.

Github has developed the projects Atom and Electron, the former of which is a free and open-source text and source code editor, with a clean interface and simple usage. The latter is an open source framework that can serve websites as desktop applications, and is used by several companies, including Discord and Spotify.

Using Github

Github can be used from either the command line, the website, or the desktop app. In order to use the command line interface (CLI), you need to download git, a system for tracking changes in source code and implementing version control. In order to use the desktop app, you need to download it from the Github website. The desktop app and website are easier to use, but the CLI has more capabilities. This tutorial will use the CLI.

CLI

To create a Git repository in the CLI, create an empty folder for the project using *mkdir* and cd into it. Then, run *git init* to initialize the repository.

The command *touch <filename>* can be used to create new files, and *git add <filename>* can be used to add those files. The command git status can be run to check if any untracked files exist, which can be added using *git add <filename>*. Once the files have been added, run *git commit -m "message"* to commit the files with a message that describes the action.

After a repository has been committed to, git branches can be used to move between different versions of a project *git checkout -b < branchname>* in order to create a new branch and move onto it. The command *git branch* can then be run in order to verify that the branch has been created. By using branches, you can modify and test your code while being able to rollback the repository to previous versions where it is confirmed to work.

In order to upload the repository to Github, you need to click the New Repository button, after which you name and initialize the repository in Github. To add the existing repository on the computer to the Github repository, run git remote add origin *https://github.com/<username>/ <repositoryname>.git*, and then *git push -u origin master* in order to push the repository to the master branch, which is the one public on Github. Branches can be pushed to Github using *git push origin <branchname>*. Pull requests can be made on the website to alert the owners of a repository that you intend to change the repository, which they can approve and merge the changes.

Finally, *git pull origin master* can be used to get the most recent changes made.

Proposal

I propose that the submissions be stored in Github as follows:

gci2019 ← the main folder of the repository

- student1 ← username of one of the students
    - README.md file containing a short introduction from the student and their opinions on GCI and their experience working with liquid galaxy, with links to their own github or other coding websites if applicable
    - task1 ← standalone task, like the arduino ones that are not multi-part
        - task1 project files
    - task2 ← multi-part task, like canvas or citytour, then use the common name
        - task2 project files from the last step and the final versions files from earlier steps
    - and so on
    - Videos
        - all videos showing demos uploaded by the student across all tasks
    - Apps
        - .apk files of any android apps developed by the student across all tasks
    - Arduino
        - .ino files of any arduino programs developed by the student across all tasks
    - Bash
        - .sh files of any bash scripts developed by the student across all tasks
- student2 ← username of another student
- and so on
- file containing table of values with username as columns and tasks as rows to show which students did which task so that it is easier to find examples of a task