This is a guide for non-programmers to using Github as a communication tool

Basics of Github

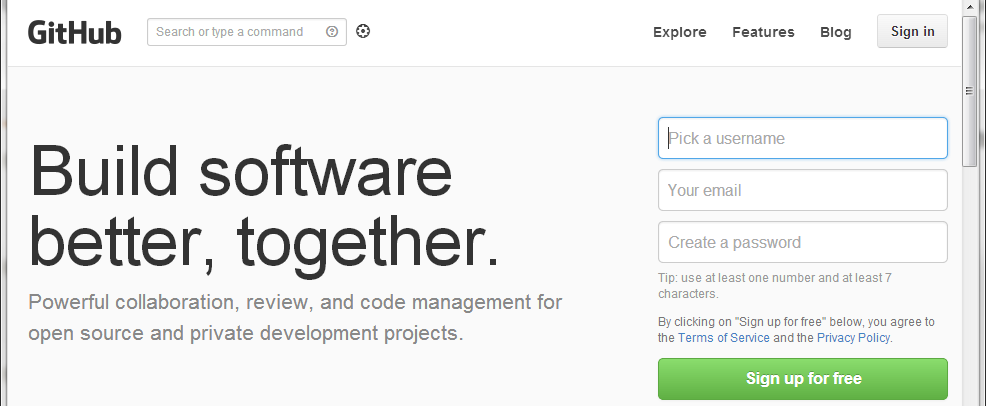
Github is a website that allows users to store any project (typically code) online and manage multiple users working on the same files. However, it also features three useful features for those not involved with working on the project directly.

1. Forum for posting ideas, issues, and any sort of discussion (deceptively called “issues”)
2. Wiki for safe-keeping of information (help guides, requirements list, etc.)
3. Ability to download every file in project for your own perusal

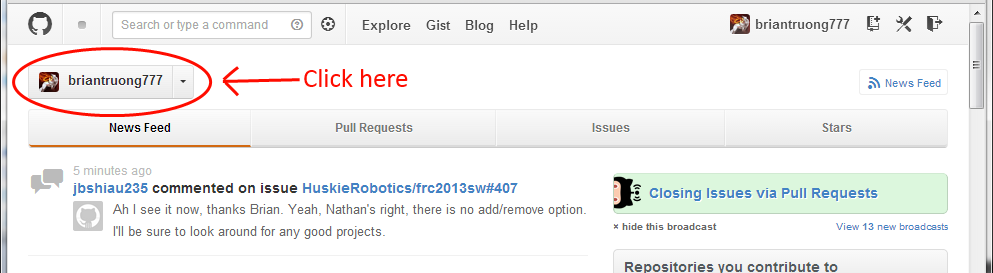
Before continuing, you should know that Github is continuously and rapidly developed resulting in interface changes occurring unexpectedly. These pictures are almost guaranteed to be out-of-date by the time you read this. I don’t think it is necessary (and I’m too lazy) to update the pictures every time Github changes since the basic operation is almost always the same anyways.

Setting up your own Github account

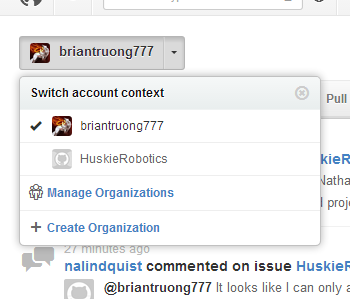
1. Make an account on the main page: [github.com](https://github.com/)



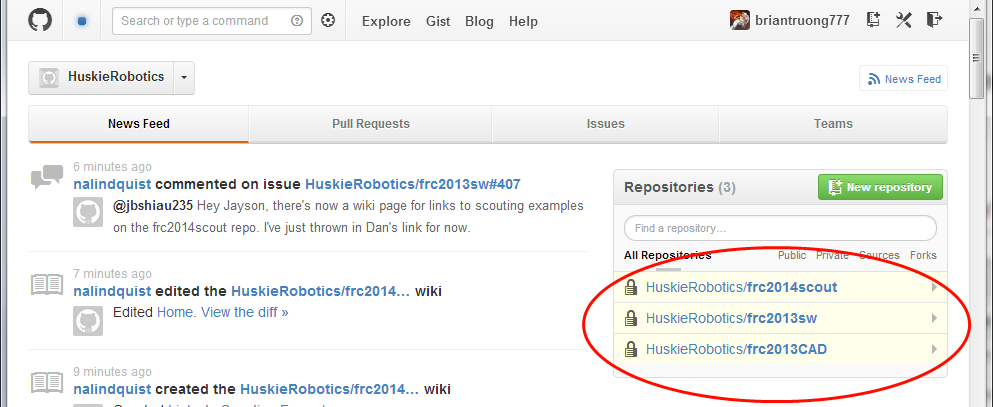
1. Bother your local admin programmer to add you to the Scouting Team on the Huskie Robotics organization. They will require your exact username. Then bother them again for kicks. Don’t bother them after that unless you want to suffer pain.
2. Go back to [github.com](https://github.com/) and log in again if you aren’t already logged in or refresh the page if you never left
3. Click on the top left button where your username appears to access the organizations you are involved in



1. Click on “HuskieRobotics”



1. You will now be able to see all the HuskieRobotics repositories (projects) you have access to. Chances are you will see only the scouting repository. Click on a repository to enter it.

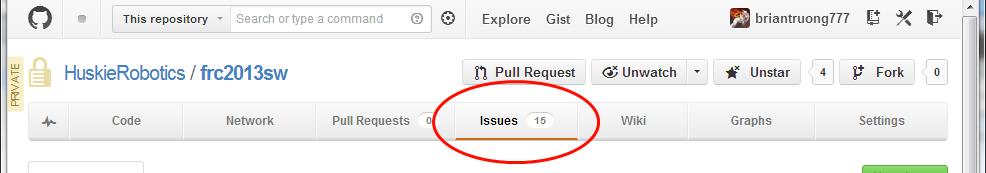


1. You are now finished setting up your account. Yay!

Using the discussion board (aka issues)

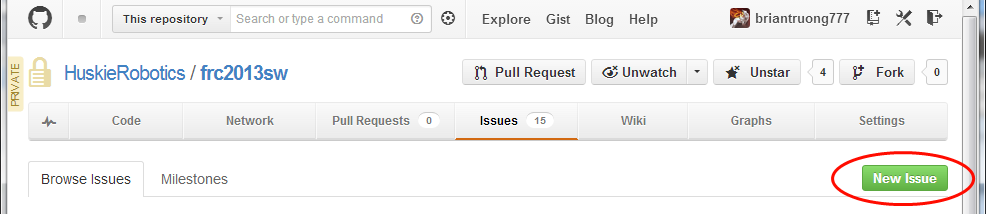
Although they were originally designed to manage issues in code, the issues section is really just a forum where issues (threads) can be opened, commented on, labeled, and finally closed (archived). Even if you don’t actually have an issue, feel free to make one whenever something needs to be discussed or you need to announce something to the group.

First of all, click on the issues button to access all the issues for the repository.

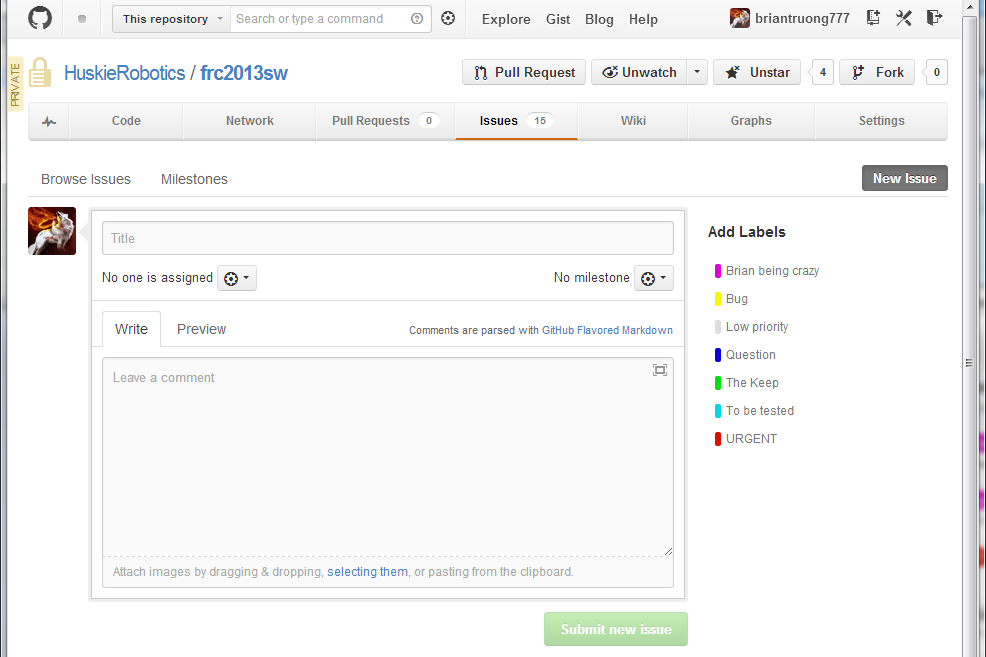


Opening an new issue

1. Click the “New Issue” button



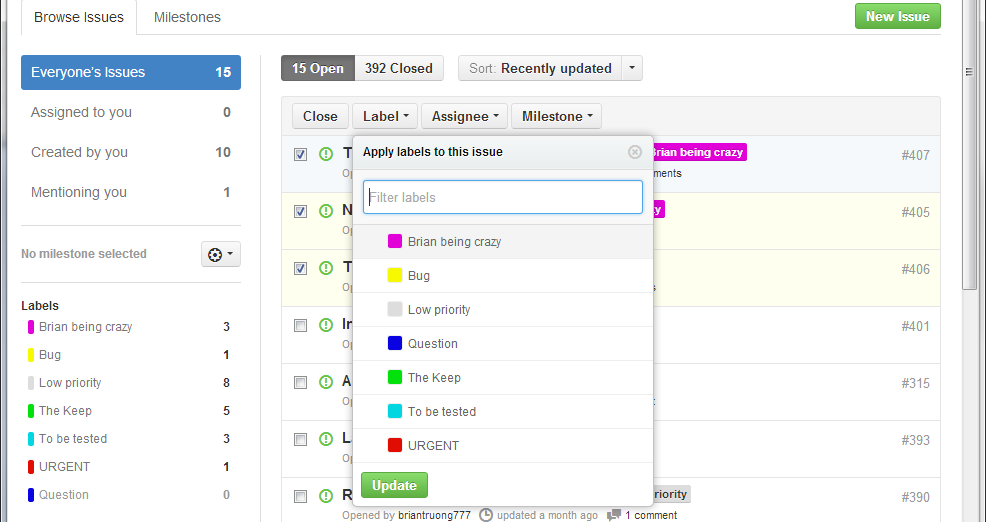
1. Fill in a title, a body, and add any labels you feel are appropriate. The body section supports Github flavor markdown (look it up only if you feel like it) which can allow for some pretty fancy formatting, but that’s something for another guide.



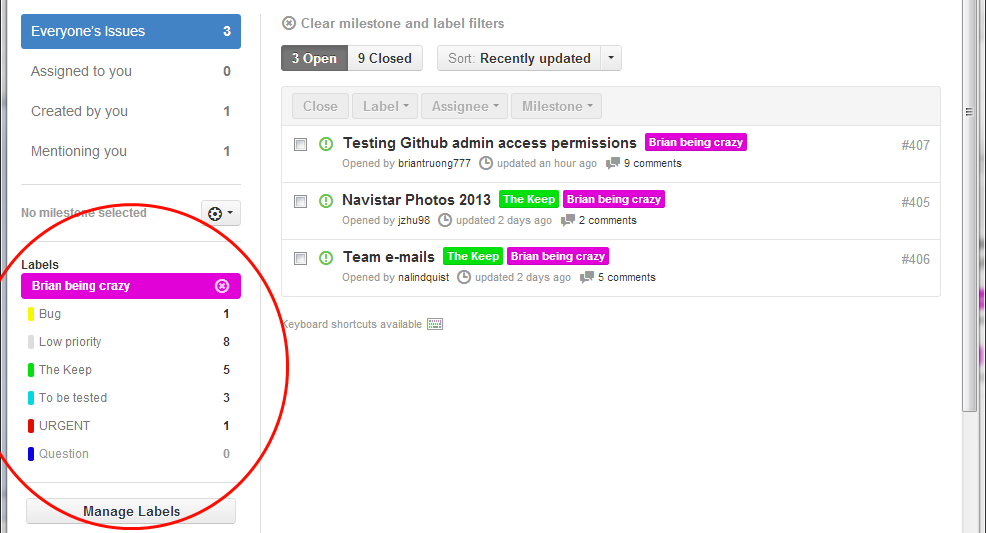
1. Submit your new issue and it will pop up on the issues list

Basic issue stuff

* To “close” an issue doesn’t mean it is deleted, it just won’t show on the open issues list. Issues can be reopened and closed as many times as desired. Typically, issues are closed when they are solved or become irrelevant. You should always explain why you are closing an issue, but someone else can always reopen it and ~~complain~~ comment.
* After clicking on an issue, you can see the discussion in chronological order. Users can actually edit any comment, not just their own (beware of trolling). Scroll to the bottom to write a comment or close/reopen an issue.
* When browsing issues, you can change multiple issues by checkboxing them and then applying a new set of labels or closing all of them.



* You can choose to view only certain labeled issues by selecting them on the left.



* Right below the list of labels, the “Manage Labels” button allows you to…manage labels. This includes deleting them, changing their color, and renaming them. Right below this button, you can make a new label as well.
* You should be aware that there is a special type of issue called a “pull request”. These are related to branches/coding and basically indicate the developers want to merge a branch into the main branch. If you didn’t understand that, that’s fine since you can still read the comments the developers make.

Pull request issues can be identified by this icon:



Rather than the normal issue icon:



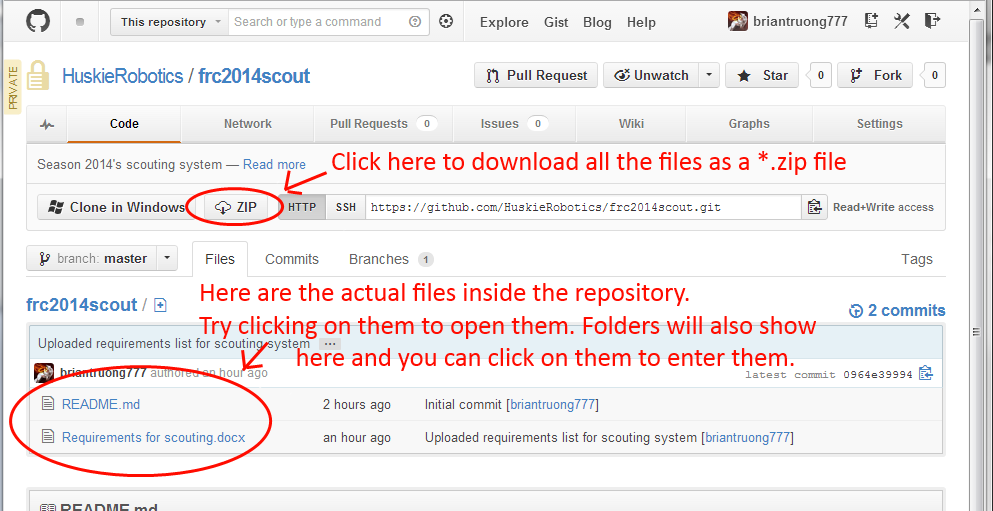
I’ve covered the most important elements of Github’s issue management, but there is definitely a lot more so feel free to explore.

Using the wiki

Like the “pages” feature in Infinite Canvas (if you are still using that), the wiki for Github’s equivalent. You can edit pages, make new ones, and link them together. When editing a page, you can choose what edit mode you want to use. The default (markdown) is somewhat unintuitive but powerful. It does not have “what you see is what you get” formatting. Unfortunately, teaching how exactly to make a nice wiki is beyond the scope of this guide.

Downloading files

You can either download a particular file, or download everything as a \*.zip file. To download a particular file, click on it, and then click on “Raw” to download.



After clicking on a particular file:

