GitHub and GitKracken work together as a team. I’ve never used either until now. At first, they both seemed really confusing. I thought to myself, “Well, why not just use google drop box to share your files?” But as I started using GitHub and GitKracken, it all made sense why we would use this in our DGM classes.

GitHub is the server of the whole system. It is the cloud of Git. This server is where your files are uploaded to. Here you can share these files with other users. We will get more into the sharing part later. GitKracken is the client directly installed to the hard drive of your computer. Within GitKracken, you can create a history of changes you’ve made to a specific file. The first step is to create a link between your GitHub account and the GitKracken client. Once that us done, you designate a folder on your hard drive that you can drop files into and those files will be shown in GitKracken, where it will keep a log of changes you’ve made to those files. This is called a repository. It is tracked on both GitHub and GitKracken.

When you open up a file that has been stored in your repository folder, you can make changes to it. As soon as you save those changes, GitKracken will tell you when there have been changes made to a file within a linked repository by showing ‘WIP’ on the viewport. This means you are ready to give detail to the changes logged and keep a history of those changes. The first step to adding these changes, begins with ‘staging’. Staging refers to preparing changes to a file to be logged in GitKracken. Once it is stages, you can write a description of what changes occurred when editing a file. For example, the file you create shows weather changes and updates, every hour or so you update the weather temperature, you save the file, then the ‘WIP’ appears in GetKracken. Then you stage the file, for each time you changed and saved the file, and Write a small summary of what changes you made.

The next step that follows staging, is to ‘commit’. When one commits, they are sure of the changes and are ready to publish those changes after staging them and describing the changes made. Committing logs the changes in GitKracken. Once one has commited, it is done locally, meaning it is only logged on the computer and not on the servers. That is when ‘push’ comes in to play. When you have made all the necessary changes, staged them and committed them, you then need to sync those changes with the GitHub serves. When you ‘push’ those commits, you are updating the GitHub server, you are keeping the server up to date with the changes you made to the repository. This allows those you’re sharing a repository with to receive the most up to date file.

When sharing a repository with another user, they need to ‘pull’ the updated file from GitHub onto GitKracken, which will save it locally to their synced repository. At that moment, they are able to make changes to that file, but only locally, until they stage and commit the changes, and push the commits to where an updated version is accessible.

Once you get the hang of it all, it is really simple to use and very convenient or sharing files and making a log of changes between users.