Using Git for our Project

Tools

The following tools will help with git, and some are required. I would recommend downloading the following:

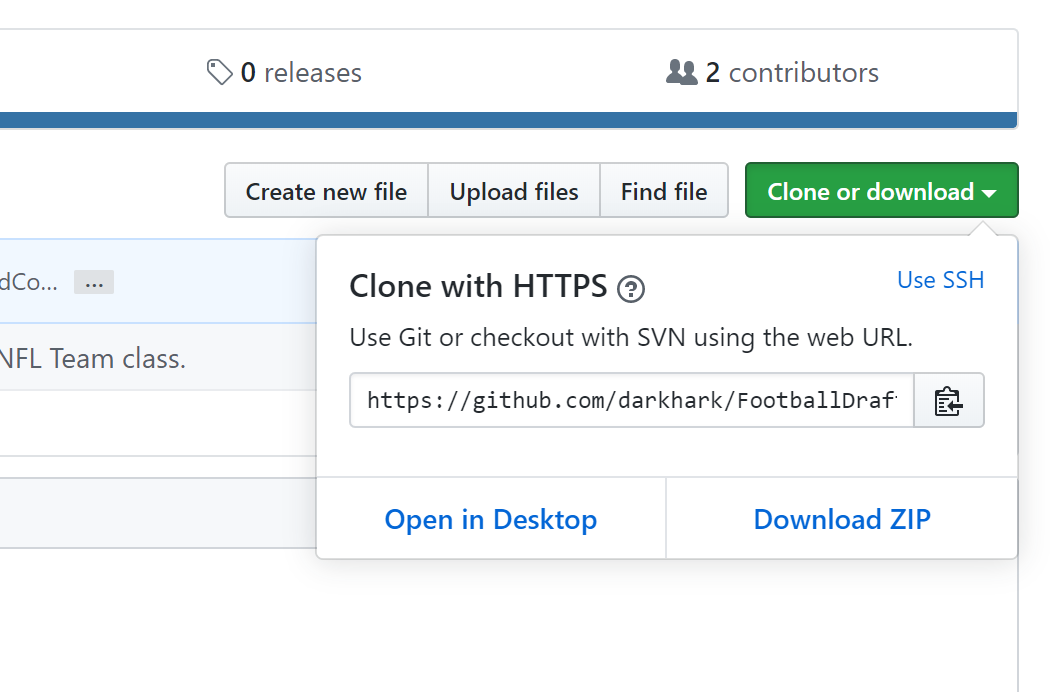
1. Git Bash – For Windows only. Mac is Linux based so you can just run all the commands in the Mac’s terminal.
2. Git Extensions
3. PyCharm from JetBrains
4. An account with JetBrains so you can connect to YouTrack, an issue tracking web app.

“Git”ing Started

First, you’ll need to create a GitHub account and accept the invite to the git project, <https://github.com/darkhark/FootballDraftAnalysis>.

After you’ve accepted the invite, you need to clone the repository. Cloning the repository downloads the repository (folder) onto your machine so you can work on the project.

Open the project in GitHub. Click clone or download on the right of the page.



Decide where you want to save the project to. For me, I save it in C:\Users\joshu\Documents\CNT5805\FinalProject. Open Git Bash or the Mac terminal. The terminal and git bash may open in different places. Use cd <directory\_name> to change to the desired directory. For me, git bash starts at C:\Users\joshu\ so I use

cd Documents/CNT5805/FinalProject

to get to my project. Always use / not \ for both git bash and the terminal. Now that you’re at the desired location for the project, type

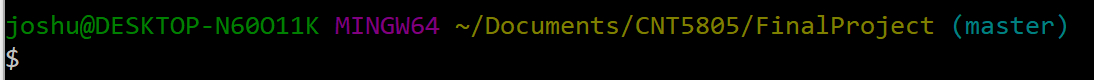
git clone <project\_link>

This will now download the project into that directory.

Understanding Git

I would suggest looking into git flow to understand the topic more, but I’ll give a general overview.

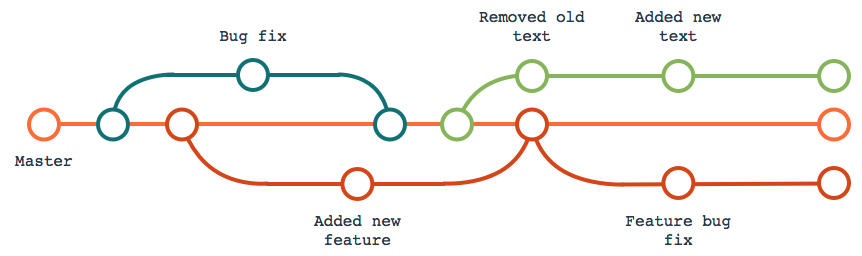
In git, there should always be at least one branch that’s considered the stable branch. This branch will only take in code that has been reviewed to reduce the risk of something fatal being passed in. Our stable branch is master and will be the first branch you see in git bash (not sure if terminal shows it).



NEVER WORK ON THE MASTER BRANCH!!!

If accidently do, it’s okay and I can help you move the code over to your own branch.

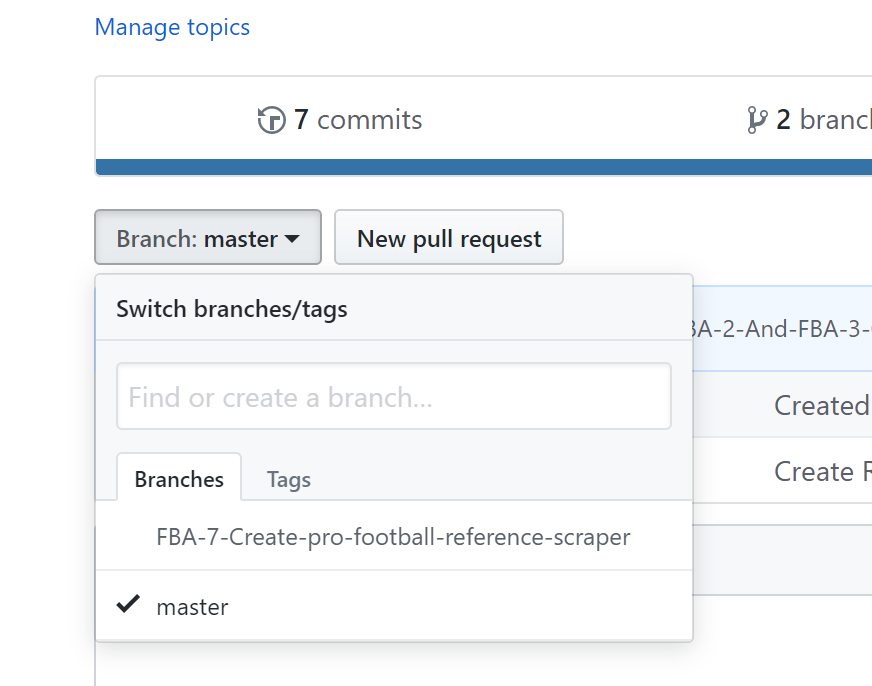
Any work you perform on the project should be done on your own git branch. As you work on your branch, you should aim to keep it up to date with master as other people merge their changes into master. When you’re finished with your work and ready to merge it into master, you create a pull request. The pull request is reviewed and as long as your changes are approved, they’ll be pulled into master. Hopefully the image below can help you visualize this.



What to do After Cloning

Go to YouTrack and look for an open issue. If you don’t see one, make one. The tickets allow us to see what everyone is working on or has worked on. It also allows for an easier branch naming methodology which leads to better communication.

Once you have an issue to work on, mark it as in progress. There are multiple ways to create a branch, but the easiest to remember probably involves GitHub. Go to GitHub and select the drop down for branches.



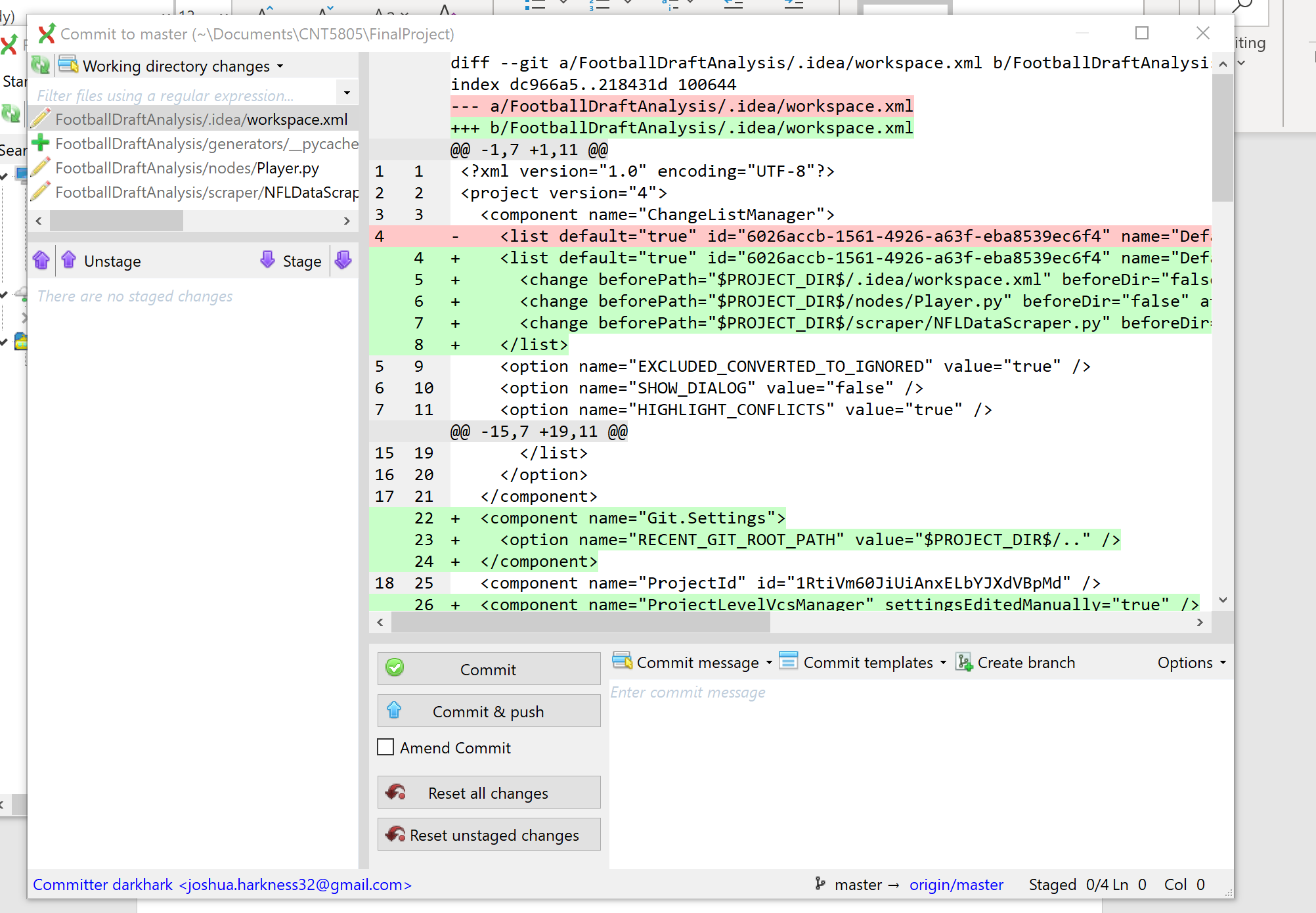
When creating the branch, start with the ticket number, then follow it up with a short definition of the task. Use – instead of spaces or CamelCase.

Now that the branch is created, go back to git bash or your terminal. While in the project directory, type “git pull”. This will pull down any changes from master (since you’re currently on master) including new branches. To access your branch, type

git checkout <your\_branch\_name>

How to Merge in Your Changes

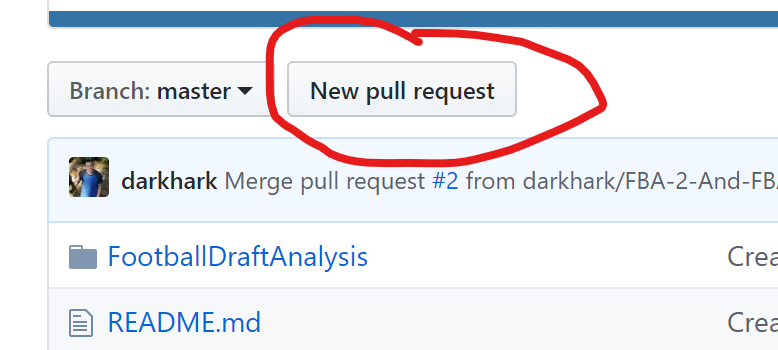
When you complete your task, you need to commit it. This is where Git Extensions comes in handy. Open git extensions and select the commands drop down. Click commit.



A new window should open. I’m still working on a way to ignore anything in the “.idea” folder, so for now just make sure not to stage any changes from that folder. Staging a change prepares it to be send to your remote branch. All the work you do on your machine is done on your local branch. Each change you want merged pushed onto your remote branch needs to be moved into the staged section.

On the right, the green shows the additions you made while the red shows the removals you made. On the bottom right is the message box. Before you commit anything to your remote branch, you should provide a short description of the work you did. This makes it easier for reviewers to understand why you made the changes you did.

Once you have a good description in the commit message box, select commit and push. All your work is now available for everyone to see if they checkout your branch. To merge it into master, go back to GitHub, make sure your branch is selected and select “New pull request.”



I think that should be it, but you may need to specify that you want to merg into the master branch.