Nick Zahlis

2/26/19

Summary:

Today’s lesson gave us a much more in depth explanation of how we can use git to collaborate with others. Github seems to be a very useful resource that will help us organize our files for our next project. I would say that my understanding of git so far is a 3/4 because although I understand all of the commands, I still don’t have a complete understanding of how merge conflicts work. I think that using it for everyday use during our next project will clear up this misunderstanding though.

2/26/19 git pt.2

* *Goals:*
* *Setup a remote repository*
* *Learn how to push our local files to the remote server*
* *Understand branches and how to use them*
* *Understand how to merge different branches*
* *Learn what a merge conflict is nad how to resolve it*
* A remote repository is a copy of our project that is stored “in the cloud”
* Its where we backup/share our work
* Accessible anywhere with an internet connection
* Git push tells git to upload all changes to server
* Git push isn’t needed after every commit, it will upload all commits since last push
* Branches are smaller bits extending from a tree trunk
* Branches represent different versions of our code
* Branches allows us to work on code fixes/features without breaking what is already probably working
* fixes/new features should always start on a branch
* The master branch is the trunk of the code tree and should only contain clean code ready for deployment (use on the web)
* Git branch <name> tells git to maintain a new copy of our code with the given name
* Git branch without a name will list the available branches and display an asterisk next to the one that we are currently working on
* Git checkout <branch> tells git to switch our working folder to the branch name specified
* Git tracks files in the branches independently
* The **merge** command is used to combine branches
* First chechout the branch that you want to merge **To**
* Then merge one branch into the other (mobile to master)
* Git merge mobile
* A merge conflict is where a file has changed in both of the branches you are trying to combine and git can’t determine what you want to keep
* During a merge conflict, git is asking for help because it is confused