

Day 3 Class Notes

More with Git and GitHub

Git and GitHub Process

Before we start we need some basic terminology for git and GitHub. The first word is a *repository*. A repository is a folder that contains files and subfolders that have files that you wish to track the changes in. Repositories can be stored on your local machine (a *local repository* (repo)) or on a computer in the cloud at a site like GitHub (a *remote repository*).

Once you have a repository you will tell the git program which files you want to track. This is called adding the file to the repository. Once added, git will track the changes you make in the file.

Many people can have the same repository locally on their computer, but their versions may be different. Git has ways of checking to see if there is a **conflict** between different versions of the repository once people look to **merge** their versions.

Let's work with an example.

When making a change Once a change is complete, you should **commit** your changes to repository. On PyCharm you do this by clicking on the green check mark then adding a message about what you did and hitting the button labeled commit.

Make a new branch Sometimes you want to make changes free of other's changes. You want to work on your own area, or you want to keep a version of what you have done intact without. This is most easily done by making a new branch. Branches can be made using the PyCharm Git menu. Branches allow you have several versions of the same code on your machine. When you check out a new branch Git will update the tracked files to be in alignment with the most recent version of that branch.

Make commits in that branch You make commits in a branch the way you always make commits. You click on the checkmark or goto Git/Commit then type a short message about the commit ##### Make a pull-request

Once you have made changes in a branch that you want to tell other people that are working on the project that you have completed an important task. To do this you can create a pull request.

Cloning Cloning is a way of copying the project repository on to your own device.

Git Glossary

- **repository** - The highest level of a GitHub project.
- **branch** - A branch is a version of the project that can be updated, or it may be the main branch which contains all the completed changes.
- **commit** - A step in the completion of the project. This is a moment in the development of your project where you have completed a significant step (sort of like saving).
- **pull-request** - An announcement that an updated version of the code has been completed and needs to be incorporated into the main branch.
- **clone** - making a copy of a remote repository on your own computer.

Practice

1. Make a new project and clone the repo at: https://github.com/CS570-2022/Text_Integrator_Test
2. Create a new branch using your name
3. Create a file using your name and use the markdown ending, “.md” for the name. So make a file called “Jane_Doe.md”
4. In that file write what you like about robotics, and sign your name, and add a few line spaces at the end of the file.
5. Commit that change
6. Make a pull-request to the main branch.

Rolling back a commit Let’s say that you have been working on something and then find out you have been doing all wrong. You want to go back to some previous commit that you had in that branch. This can be done with the command:

```
git reset current~2 # this will take your code back two commits.
```

You can also do it in PyCharm by going to the Git menu and first choosing **Show Git Log**. Then find the code for the commit that you would like to return to. Control click on the commit that you would like to revert to, and choose **Reset Current Branch to Here...** You will be given a few options. Read through them, and pick the best one for you.

Seeing the network of branches **Show Git Log** also gives a sense of the ways in which the branches are working, and can be useful to understanding how progress is going on the project.

Seeing your progress Another way to see progress is by looking at the differences between files.

Looking at commits on GitHub Reading commits on GitHub can make