Lecture 2 - Git & Github

Programming for VR I

Patrick Mineault

Github

- You will learn things that you will immediately forget
- If you leave a trail, you don't have to memorize
- Github can be your super-memory
- The history of everything you've ever programmed

Source control

- Github hosts git repositories
- A repository: think of it as a filesystem with a history
- `git` is a source-control technology that was originally made by Linus Torvalds (who also wrote the Linux kernel)

Github as a lurker

- An almost magical cornucopia of things you can download
- Which things are “good to download”? Watch out for:
  - License
  - Language
  - Stars and forks
  - Last commit

Github as a contributor

- Exercise: let's sign up for Github!

Two workflows

- Clone a repo: make a local copy of a repo.
- Command line `git clone`
  `https://github.com/patrickmineault/programming-course.g`
- Try it!
- Q: can't I just download a zip file instead? Yes!

Create a repo and put stuff in it

- Create a repo, add stuff to it, and share it with the world.
- Let's create a repo on Github called `hello-world-python`
- Clone the repo
- Create a new file in the directory: `helloworld.py`
- Write our hello world example in there
- Commit it and push it

Key commands

- `git status`: What is the status of my repo?

```
$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be commi

    helloworld.py

nothing added to commit but untracked files present (use "g
```

Key commands

- `git add myfile.ext`: Add a file to be committed. You need to add uncommitted files (new files and modified) before you can commit them.

```
git add helloworld.py

$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   helloworld.py
```

Key commands

- `git commit -m "My message"`: commit *locally*

```
$ git commit -m "Life is fun"
[master (root-commit) 94583f3] Life is fun
 1 file changed, 1 insertion(+)
 create mode 100644 helloworld.py
```

- The message should tell, what this commit is, e.g. “added confobulator feature”.
- Now you have a commit.
- Now it’s time to upload to Github.

Key commands

- `git push`: push the current branch to the remote (Github).

Summary

- Three commands:
- `git add`
- `git commit -m "message"`
- `git push`
- Commit often, push often (e.g. once a class)
- When in doubt, search for “git cheat sheet”

Context about how programmers use git and github

- Programmers often like to work on different features at the same time - each branch can be its own feature. We're just going to work on the master branch, which is the default one.

```
$ git branch
* master
```

- You can fork other people's code and then merge it back.
- `git` has a lot of tools for managing multiple versions of the same code simultaneously. We'll go with a very linear workflow, but you can use a very nonlinear method (think of the plot of Back to the Future 2)
- By having a complete record of what happened, when something goes very wrong, you can figure out what went wrong and why.

Exercise

- Create a repo for the class.