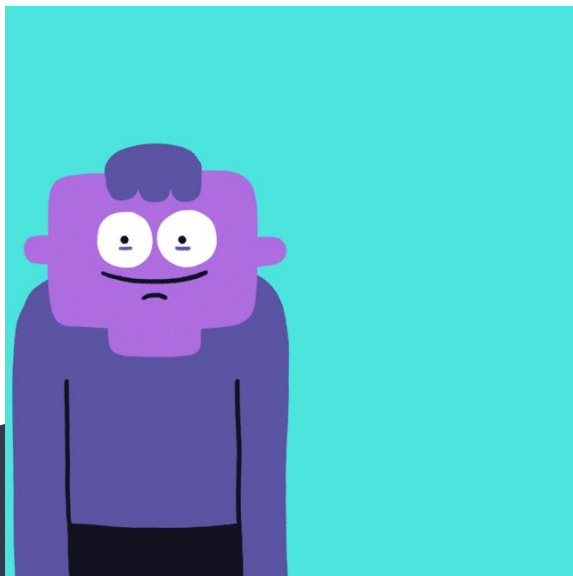


Making a Private Clone of a Public Repo



a way to set up the public Controlbook repository for private student development

By Carson Moon

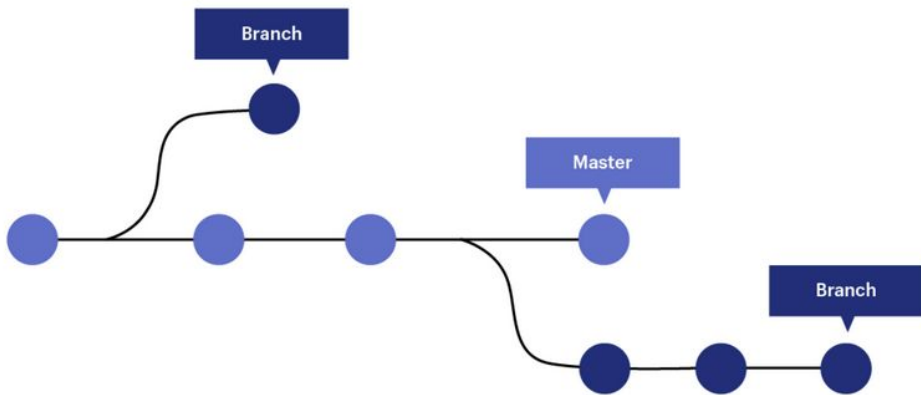
You might want to use Git for your own development

This isn't an intro guide on how to use Git, but here's a few things to maybe persuade you to use it:

1. Git is a version control system. It's like Google Drive, but for coding and it's way smart at managing and tracking changes made to projects, especially when lots of people are contributing to it simultaneously. It's also still useful for personal projects.
2. For example, if the professors made some changes to the Controlbook repo during the semester, merging those changes manually with your work done so far could potentially be a mess. Merging is one of the big reasons Git exists!
3. Last semester a student's laptop went kaput and they lost all their code. Don't be that person.
4. Everyone who codes professionally uses it, so you should too.

The Problem: the Controlbook repository is a public repo

Usually, when you want to work on and modify a repo found online, you just "*branch*" or "*fork*" it, but GitHub doesn't allow private branches or forks of public repos. And since Controlbook is a public repo, we can't just branch or fork like usual.



So what can you do??

Essentially, you (and the Honor Code) don't want some random joe finding your code solutions online and using it.

A. Start by making a new private repo online

Make your private repo online on github.com/new

1. Give it a name like "control_systems"
2. *Ensure it is private!*
3. That's it! You're all done! Just kidding, click "create repository" now.

By the way, if you don't have Git downloaded on your computer, you'll need to do that before continuing:

- <https://www.git-scm.com/downloads> (the Git download page)
- <https://desktop.github.com/> (a handy visualizer for GitHub that I like using, which also downloads Git for you too)

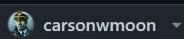
Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *



Repository name *

control_systems

✔ control_systems is available.

Great repository names are short and memorable. Need inspiration? How about **effective-barnacle** ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:



Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

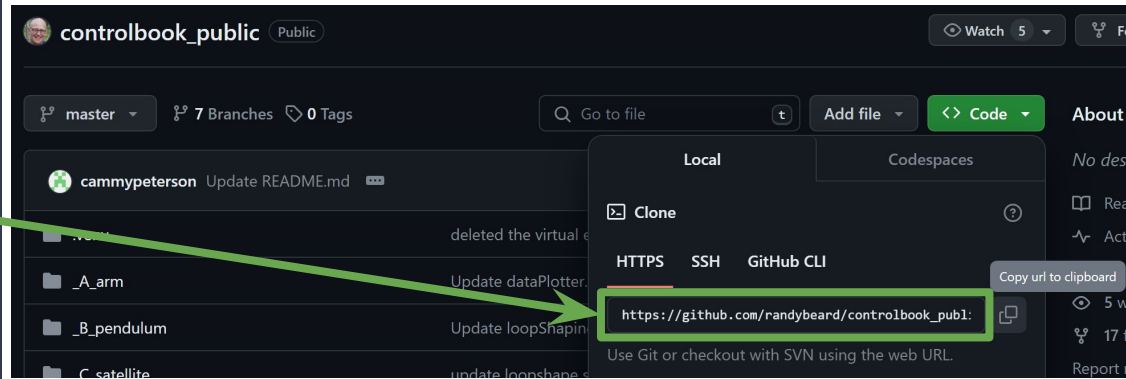
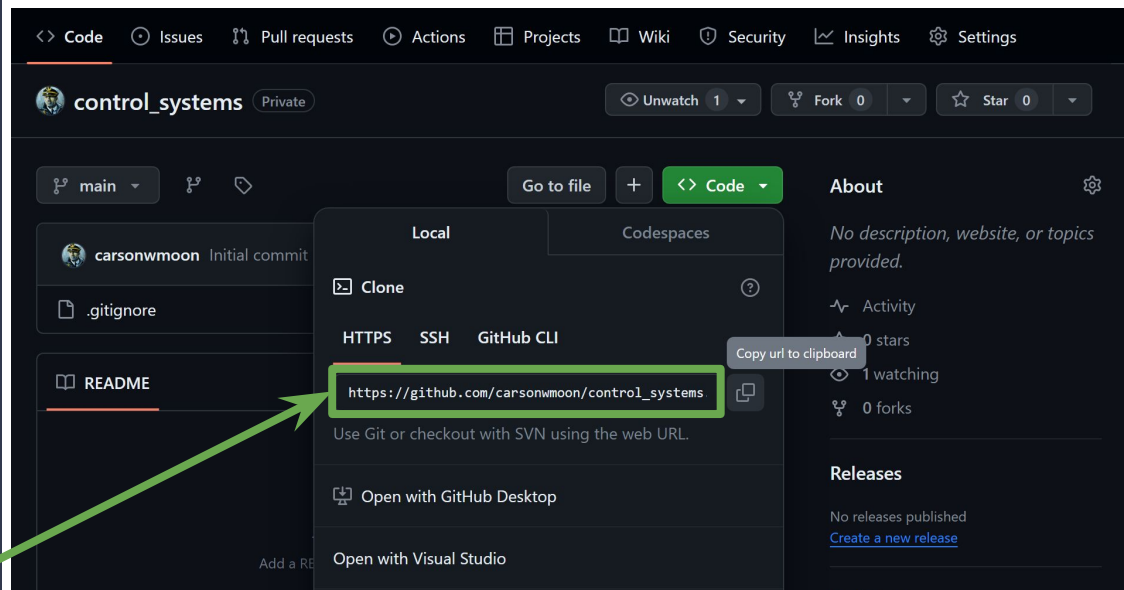


You are creating a private repository in your personal account.

Create repository

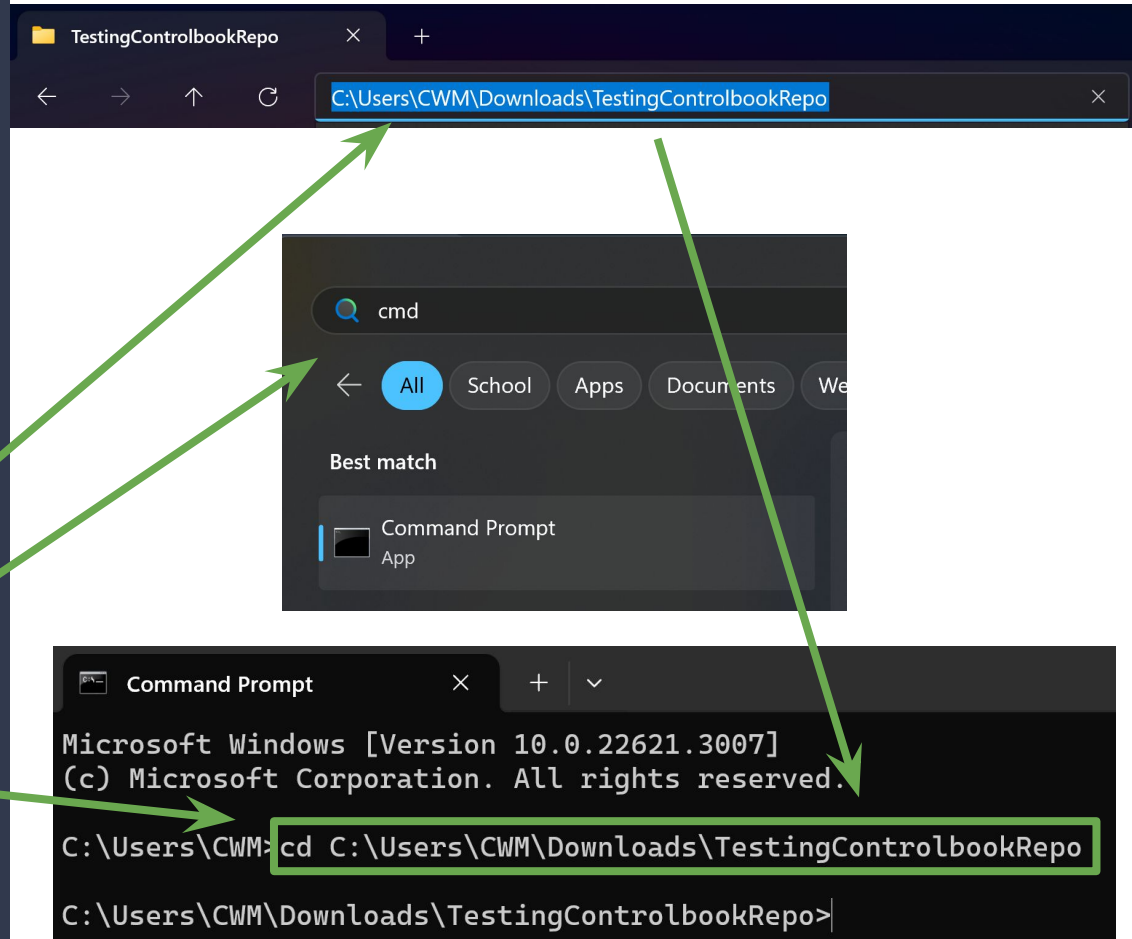
B. Find the HTTPS cloning URLs

1. The last step should've taken you to the main page of your private repo. Locate the HTTPS cloning URL. Then keep this tab open.
 - a. You'll use it soon, I promise.
2. While you're at it, open up <https://controlbook.byu.edu/> in a new tab because you'll need the URL from there too.



C. Open Command Prompt

1. Open file explorer to where you want to have your code on your computer and CTRL-C the file path.
2. Open your terminal.
 - a. search "CMD" on Windows
3. Type "cd path" where "path" is the file path from step C.1 above.



D. Clone it onto your computer and connect to remote upstream

1. Type "**git clone** privURL" where privURL is the URL from step **B.1**.
 - a. See, I told you you'd use it!
2. Type "**git remote add upstream** pubURL" where pubURL is from step **B.2**.
 - a. This connects your private repo to the public Controlbook repo
3. Now, when you type "**git remote -v**" it should show that it has a connection to both the private and public repos.

```
C:\Users\CWM\Downloads\TestingControlbookRepo>git clone https://github.com/carsonwmoon/control_systems.git
Cloning into 'control_systems'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

```
C:\Users\CWM\Downloads\TestingControlbookRepo>
```

```
C:\Users\CWM\Downloads\TestingControlbookRepo\control_systems>
```

```
git remote add upstream https://github.com/randybeard/controlbook_public.git
```

```
C:\Users\CWM\Downloads\TestingControlbookRepo\control_systems>git remote -v
origin https://github.com/carsonwmoon/control_systems.git (fetch)
origin https://github.com/carsonwmoon/control_systems.git (push)
upstream https://github.com/randybeard/controlbook_public.git (fetch)
upstream https://github.com/randybeard/controlbook_public.git (push)
```


E. Git merge!

1. Since the main branch of the public repo is called "master", type "**git fetch upstream master**"
 - a. This loads from online the "master" branch of the public repo we named upstream.
2. To put that content into our private repo, type "**git merge --allow-unrelated-histories upstream/master**"
3. Finally, type "**git push**"

```
C:\Users\CWM\Downloads\TestingControlbookRepo\control_systems>git fetch upstream master
remote: Enumerating objects: 13538, done.
remote: Counting objects: 100% (921/921), done.
remote: Compressing objects: 100% (437/437), done.
remote: Total 13538 (delta 553), reused 734 (delta 466), pack-reused 12617
Receiving objects: 100% (13538/13538), 197.23 MiB | 8.67 MiB/s, done.
Resolving deltas: 100% (5396/5396), done.
From https://github.com/randybeard/controlbook_public
* branch          master      -> FETCH_HEAD
* [new branch]    master      -> upstream/master
```

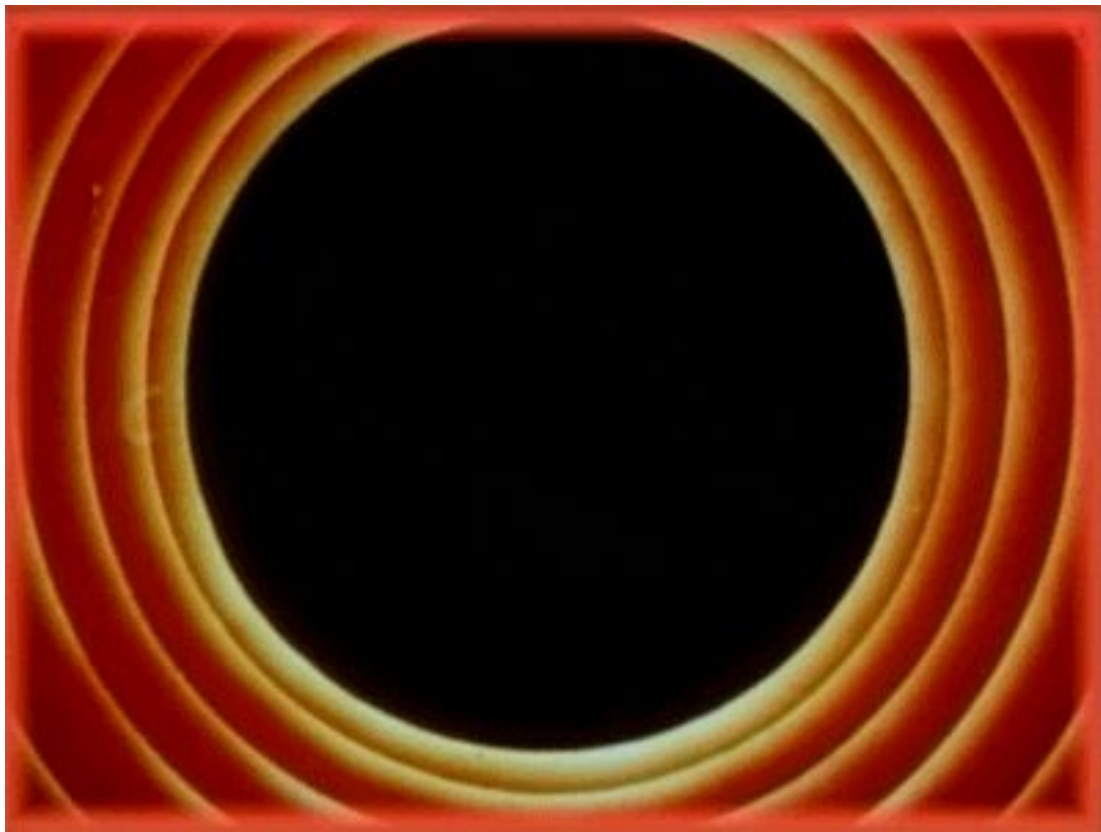
```
create mode 100755 vtol_path_planner/python/VTOLParam.py
create mode 100755 vtol_path_planner/python/VTOLSim_path_planner.py
create mode 100755 vtol_path_planner/python/ctrlTrajectoryFollower.py
create mode 100755 vtol_path_planner/python/dataPlotter.py
create mode 100755 vtol_path_planner/python/pathPlanner.py
create mode 100755 vtol_path_planner/python/signalGenerator.py
```

```
C:\Users\CWM\Downloads\TestingControlbookRepo\control_systems>git merge --allow-unrelated-histories upstream/master
```

F. for respect

1. You did it! If you check your private repo online and in the file explorer, they should both contain the same folders and files as the public repo!
2. At this point, use your repo like normal by adding, committing, and pushing as you please.
3. If changes occur on the public repo, just follow the steps on E. again and you should be good.

Big thanks to "Amir Saniyan" on <https://stackoverflow.com/questions/7983204/having-a-private-branch-of-a-public-repo-on-github> for his answer on how to perform these steps.



If you want to learn more about Git...

learngitbranching.js.org/

This website is how I learned Git. The visualizations and level-based tutorials are amazing. Feel free to check it out. It took me about 2 hours to do all the levels, but I found it incredibly helpful.

