Use Git for the Projects

April 15, 2019

1 Introduction

Git is a distributed version-control system for tracking changes in source code during software development.
Before you start this project, you can refer https://git-scm.com/book/en/v2 for the usage of git.
Also, you should create a free https://github.com/ account for your own

2 Add a new SSH

Secure Shell (SSH) is a cryptographic network protocol for operating network services securely over an unsecured network. Typical applications include remote command-line login and remote command execution, but any network service can be secured with SSH.
For this step, please follow https://help.github.com/en/articles/connecting-to-github-with-ssh to learn about SSH and add a new SSH key to your GitHub accnout.

3 Create a repo

You should create a private repository to share your work with your partner and work together. Please follow https://help.github.com/en/articles/create-a-repo to create a **private** repository. Then, follow https://help.github.com/en/articles/inviting-collaborators-to-a-personal-repository to invite your partner to the private repository. Note that this repository will only be visible to you and your partner.

4 Clone

After you've created your private repository, it's time for you to have a copy of it on your local environment. SSH URLs provide access to a Git repository via SSH, a secure protocol. In the private repository you just created, click the green button "Clone or download", then click Use SSH and copy the URL. After that, on your local machine, cd to a directory where you prefer to put the project in, use command:

```
git clone URL(replace the URL term with the URL you just copied)
```

Then you will see a local repository being created on your machine, you should work within it.

5 Get the project file

Download the zip file from the project website, uncompress it to the local repository you just created. Then cd into it and use command:

```
git add -A

git commit -am ”proj2.1”

git push origin master
```

After this, you should see the project files appear on your Github repository. And you can start working on it, remember to share your work with your partner.

6 Important Information: Please Read!

- You are allowed to use any of Logisim's built-in blocks for all parts of this project.
- Save frequently and commit frequently! Try to save your code in Logisim every 5 minutes or so, and commit every time you produce a new feature, even if it is small.
- Don't move around the given inputs and outputs in your circuit; this could lead to issues with the autograder.
- Because the files you are working on are actually XML files, they're quite difficult to merge properly. **Do not work on the project in two places and attempt to merge your changes!** If you are working separately from your partner, make sure that only one person is working on the project at any given time. We highly recommend `https://purplebooth.co.uk/blog/2018/05/26/pair-programming-a-practical-guide/` on this project; understanding the nuts and bolts should help you experience your magical software-meets-hardware-I-actually-know-how-a-computer-works-now moment (and will prepare you to tackle datapath problems on exams).

7 Submission

Your should submit a ”proj2-1.tar” file to Autolab. And it should contain a bare repository. Here’s the guide: Use command:

```
git clone −−bare URL(replace the URL term with the URL you of your private
repository) .git
```

```
tar czvf proj2-1.tar .git
```

Then you can submit this .tar file to autolab.