CSC-450, Github Assignment

Introduction

Github (https://github.com) is a development platform and is probably the most popular web-based source code management system. Github uses uses *git* for revision control and source code management. In addition, Github implements collaborative features such as bug tracking and feature requests. Github is free to use for public repositories (a monthly charge is required for private repositories). Other source code management systems such as BitBucket (https://bitbucket.org) are available as well. Both command line *git* tools and a GUI tool called Github Desktop are available. We will be using the command line *git* tool in class.

Why use Github?

- Github is a version control system:
    - you can easily revert to a previous version of code, if needed
    - you can easily see how files change from one version to another
    - branches can be used to separate major versions of code (e.g., release and development versions; see figure below)
- Github makes it easy to work with existing code and to collaborate:
    - a use can (a) *fork* (copy) code from anyone's repository, (b) clone the code (i.e. make a local copy for development), (c) make changes to the code, (d) commit (record) the changes and (e) push (save) the changes to the remote repository (i.e., the respository on github.com).
    - if appropriate, the user can submit a *pull request*, which is a request to merge his/her code with the version that was originally forked.
    - pull requests can be automatically merged if there are no conflicts; otherwise conflicts are noted and must be manually resolved.

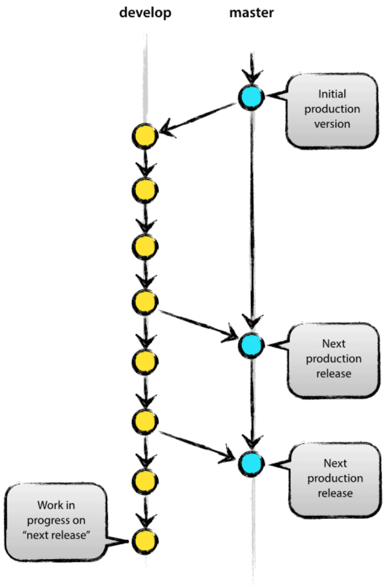
develop
master
Initial production version
Next production release
Next production release
Work in progress on "next release"

Figure 1. Example Github branches, source: http://nvie.com/posts/a-successful-git-branching-model/

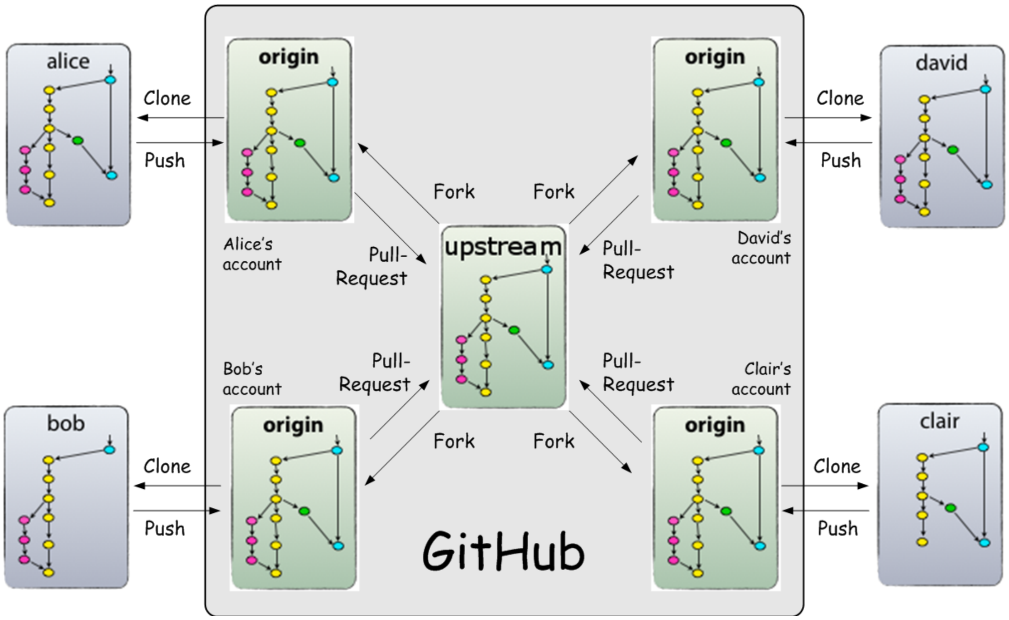

Figure 2. Collaboration on Github, source: http://www.dalescott.net/2012/09/14/using-gitflow-with-githubs-fork-pull-model/

Reproducible research

Reproducible research is fundamental to scientific research, because source code is necessary to verify published results. By making your software (and data) available, others will be able to reproduce (i.e., validate) your findings, and expand on what you have done, either by asking different questions about your data and/or modifying the code you have developed.

As part of the project requirement, you must make your source code and data available. We will use Github for this purpose (other resources for hosting your code/data can be used with permission). In addition to supporting your research project, having a Github account is a great way to promote the work you have done and will look great on your resume! Common *git* commands are listed below:

Table 1. Common *git* commands

Command	Description
git clone *url*	Clones a local copy of a repository
git status	Lists modified files and files not staged for commit
git commit –a	Commits all changes
git commit *file*	Commits changes for the specified file
git add *file*	Adds a file to the repository
git push	Pushes commits to the Github server
git log	Displays a log of previous commits
git checkout commit	Checks out files from a previous commit, allowing you to look back at old versions.

Exercise #1

1. Create a Github (https://github.com/) account if you do not already have one

2. Download and set up Git: https://help.github.com/articles/set-up-git/

3. Fork my *hello_world* program, available here, by clicking the *Fork* button on the top right of the page: https://github.com/gdancik/hello_world. This will create a new repository in your Github account with the name *hello_world*.

4. Go to **your** Github page, and using the Github shell, clone the forked repository by clicking the *Clone or download* and copying the 'clone with HTTPS' url. Then type the following command, where *url* is the clone URL, which will create a folder containing the repository on your computer.

   ```
   git clone url
   ```

5. Edit the main program (using a text editor of your choice) to change the values of *firstName* and *lastName*. Change the print statement so that the program outputs a greeting in the form, "Hello, my name is Garrett Dancik", where 'Garrett' is the value of *firstName* and 'Dancik' is the value of *lastName*.

6. At any point, you may get a list of changed files by typing the following command from within the repository folder:

   ```
   git status
   ```

7. To see the what changes are made, type the following, where *file* is the name of the file you want to examine:

   ```
   git diff file
   ```

8. Commit your changes by typing the following to commit *all* changes:

   ```
   git commit –a
   ```

   A text editor will open and you must leave a comment describing your changes.

9. To save the changes to the github server, type:

   ```
   git push
   ```

10. Submit a pull request by clicking on the appropriate links from the *hello_world* repository on your Github page.

Exercise #2

From your Github repository page, click the *New* button to create a new repository. You may choose to edit this repository directly in Github. You can also follow the steps to create a new repository on the command line. Your repository must have a README.md file that provides a description of your project. Now create a web page for your project by following the instructions for the "*Project site*" at the following link: https://pages.github.com/

When completed, send me an e-mail with the link to your Github repository page and your project web page.