CSCI 3901 Lab 4: Git

Fall 2021

Objective

In this lab, you will explore the version control system called “Git”.

Working alone or in a group of 2, you will apply your knowledge of version control, notably Git, to manage a set of files for a project.

You will be given an initial configuration of Git for a project and will be asked to effect a set of changes to the files, stored in Git, to achieve a target configuration for your Git repository.

You will be asked to submit some future assignments and labs through Git, so it is important that you develop the skills to use the tool effectively.

Preparation

- Depending on your IDE, you may need to download git from https://git-scm.com/downloads and install it. You certainly need to do it for IntelliJ (then go under the “preferences” menu item, expand the “Version Control” list, select “Git”, and enter into the “path to Git executable” box the path to the git.exe file that was just installed for you).

Resources

- There are many Git tutorials online that can help you. Some examples are
  - https://git-scm.com/docs
  - https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html
  - https://education.github.com/git-cheat-sheet-education.pdf

Procedure

Set-up

1. Clone your own repository from the CS git lab using the command
   - git clone https://git.cs.dal.ca/courses/2021-fall/csci3901/lab4/xxx.git where you should replace xxx with your CS user ID (for bluenose).
2. Retrieve the file lab4.tar from Brightspace and unarchive the file. You should see a directory with subdirectories v6, v7, v8, v8.1, v9, and v10.

Lab steps

Part 1 – History

Using the “git status”, “git log”, and “git diff” commands, answer the following questions:

1. How many constants were added between versions 4 and 5?
2. In which version was the method “main” separated into its own file?
3. How many versions of the code have been checked in? How can you tell?
4. In which version did the external documentation file appear in the repository?

Resources: status, diff, log

Part 2 – Basic operations

Add the changes from the directory v6 to the repository and commit the changes to the central repository. Include an appropriate message for the update.

Do the same again with the files from the v7 and v8 directories and commit each of the changes to the central repository so that they appear as distinct checkpoints in the repository.

Resources: add, commit, push

Part 3 – Branch

Create a new branch called “new_feature”. Your new branch should be off your repository’s master branch.

Add the changes from directory v8.1 to this new branch and commit the changes.

Add the changes from the directory v9 to your master branch, not the “new feature” branch. Commit the changes.

Add the changes from the directory v10 to your master branch and commit the changes.

Finally, merge the changes from your “new_feature” branch into your original branch and commit the changes.

The output to assess your outcome of these operations will be the git structures themselves that appear in the central repository.

Use Git to answer the following questions:

5. Which imports were added between versions 9 and 10?

Resources: branch, checkout, add, commit, push, diff

Broadening questions

1. When working alone on a project, how frequently should you commit your code to a version control system? Explain why.
2. When working in a team, how frequently should you commit your code to a version control system? Explain why.
3. Why might you create branches for your project in your version control system?

Reporting

1. In one file, list
    - The members of your team, if you did the work in a group of 2.
    - The answers to the Broadening questions.
2. Generate a PDF from the document.
3. Submit the PDF in the main lab4 directory of your git repository. Ensure that all your version changes and the PDF file made it to the central repository by visiting git.cs.dal.ca with your web browser, navigating to your repository, and checking for the files.
4. Complete the update in the central repository by 11:59pm Halifax time on Friday, October 8, 2021.

Assessment

The assessment will be on a letter grade and will reflect how well you replicated the requisite git repository and your answers to the broadening questions.

Summary of version changes:

v1: Basic input management
v2: Write enough code to run through the whole file and print it to the screen
v3: Include formatting for the metadata section of the file. Still print the body as basic text.
v4: Split main() from the rest of the code. Include code to split the body into paragraphs / lists
v5: Include translation of pair-based formatting characters, notably bold, italicize, and underline
v6: Handle implicit closing of tabs at the end of a paragraph
v7: Include recognition of list items.
v8: Handle the single-word formatting
v8.1: Change "main" to convert multiple files, prompting the user for a file name each time
v9: Clean up the code to remove unused methods and be ready for others
v10: Update the code to work on bluenose.cs.dal.ca; needed a new way to initialize my maps