M2 - Version Control

Purpose

When working in larger teams, it becomes imperative to manage the source code as multiple developers work on it. Version control allows multiple developers to work synchronously on a project, saving time and money. There are two primary types of version control systems in use: central repository and distributed. In the past version control was always based on a central repository versioning system called Subversion. Subversion is still used a few places, but this semester, for M2, we will use a distributed versioning system called Git. This milestone will introduce you to the basics of Git.

Git is described in http://git-scm.com/doc

There is a nice interactive tutorial for git here: https://try.github.io/levels/1/challenges/1

For central project hosting, you are required to use https://github.gatech.edu/

Task

1. Setup

Setup Git on your machine with the desired client (git for command line (**Recommended**), TortiseGit for file manager integration, the git software from GitHub or EGit for Eclipse integration). Note that for this semester if you are using NetBeans, Android Studio, or IntelliJ, those already have Git support out of the box -- although you may need to install the binary distribution of git for the IDE to use.

Initialize a new Git repository. Be sure that your teammates can access the repository also. You will need to work together to complete some portions of this lab. This step is not required if using GitHub hosting.

Download the M2 resource file (M2.zip) and unzip it.

Import the unzipped files (add, commit and push) as a new project named M2 (Only one person has to do this). Add the directory to GitHub.

Create a branch at this point called original. The purpose of this branch is to retain an original unmodified set of code.

Each team member should clone out the project M2.

BE SURE THAT original REMAINS AS THE UNMODIFIED VERSION and that changes to the repository are done on the main (master) branch.

2. Edit and commit files as detailed below.

If you examine the src directory, you will see the files Person1.java Person2.java Person3.java Person4.java Person5.java.

Each person on the team should choose ONE of these so that everyone is working on a different file.

Each person should create a branch named for their name (bob, sally, buzz…). Change to this branch.

Now examine your individual file and complete the required TODO. The Javadoc comment will explain what must be done.

Commit your changes and push back to Git. Merge your code from your development branch into the master branch.

3. Add and remove files

Each team member should add a text file on the master branch to the top-level directory labeled readme.pn.txt where the pn would be p1, p2, p3, p4 or p5 based upon which person you are for the lab. The contents of the file should include your name and email. Each team member should delete the text file useless.pn.txt where pn is p1,p2, p3, p4 or p5 based on your team member number. Do NOT delete the wrong files!!

Make sure all your commits have messages. Useful commit messages are essential for proper use of version control.

Roll back changes by viewing (checkout) your original branch from an earlier step. Verify that none of the changes you have made are in the original branch project version you checked out.

Criteria

Your M2 will be graded as a group upon the following criteria:

- M2 imported into git
- Changes to PersonN file committed and pushed to master
- Readme file added
- useless.txt file deleted
- All commits have messages
- Individual branch created and used
- Merges some changes into master branch
- "original" branch created and has no changes
- Your repository is located on Gatech’s GitHub