GitBot

Final Project

Bjorn Borgonia

CSCI 2912

Professor Edward Randall Souza

August 18, 2010

GitBot

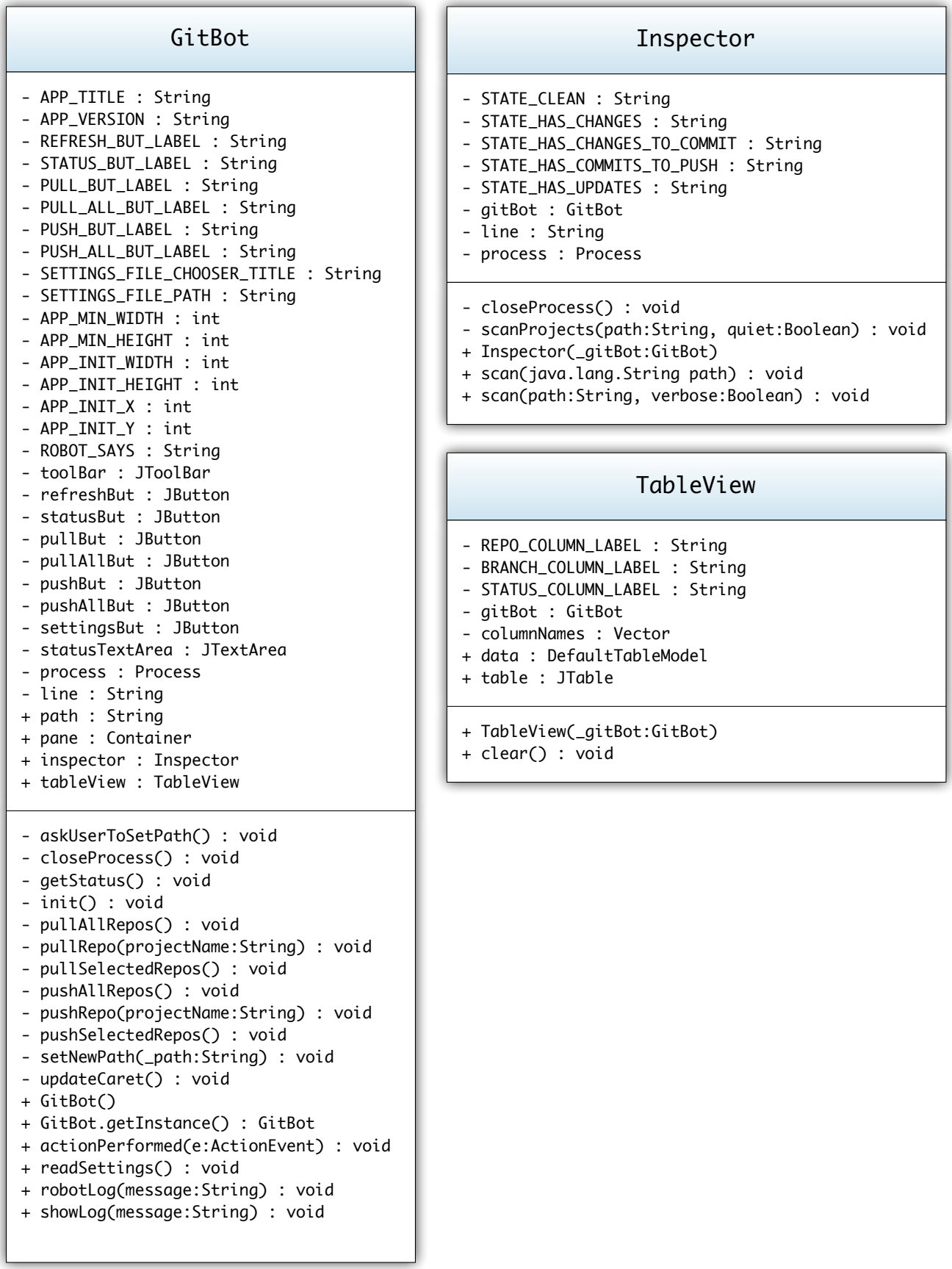
CSCI 2912 Final Project

Git is a great source code management system with useful tools to manipulate and track your code base. The problem is that there is currently is no application for managing various Git repositories available for any platform. There are occasions when programmers might forget to push their commits to the git server for access at a later time.

An example is a user doing work at the office and then leaving before remembering to push changes to the server. When the user gets home and wants to continue working, or if a colleague wants to continue working on the same project, the commits that were not pushed will not be available, so options of continuing work on the project are limited.

To solve this problem, I'll utilize some built in java swing libraries to make a user interface for instant visual feedback to the user on the status of their git repositories. There is no official git api yet. To the best of my knowledge, the only way I can accomplish this task that is to access a shell from java, run git commands using the shell, and parse the output. The additional technologies that I've employed are, the ‘Singleton’ design pattern, the java.io.Runtime class and java.lang.Process class to give me access to the system shell, git tool to run git commands, java.io.BufferedReader class and String class to capture and parse the output of the git commands, and the jar command to create an executable java archive for distribution. I've also asked my girlfriend to create a few graphics for the splash screen and Xdock:icon, we both used Adobe Illustrator CS4 to accomplish this. The splash screen is necessary because for users with many git repositories (like myself), the initial startup of GitBot will take some patience while it fetches new data and updates the status of each repository. I have future plans to experiment with POSIX's nohup command to see if it will speed up this process.

The following is a UML diagram of the current version of GitBot (v0.1 dbfeac...)



Operating Instructions

There are a few options available to obtain and install GitBot.

- Clone the GitBot project from github: git clone git@github.com:theRemix/GitBot.git
- Fork GitBot and clone your own at http://github.com/theRemix/GitBot
- Download the source from here: http://github.com/theRemix/GitBot/archives/master

Once downloaded, you may run GitBot with the java command:

```
java -Xdock:name="GitBot" -Xdock:icon=icon_128x128.png -splash:splash.png 'GitBot'
```

or by using the JAR file:

```
java -jar GitBot.jar
```

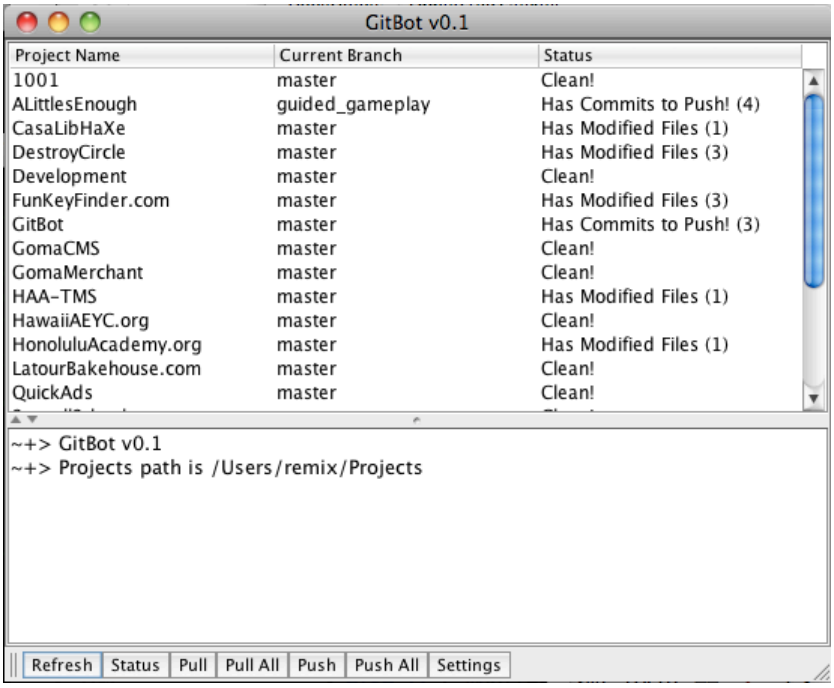
On first startup, a File System browser will appear and ask you to choose the directory that contains your git projects (not the working git project). Click “Open” to make your selection. GitBot will update all git projects in this directory, if you have a lot of git project directories, this process may take a while. Once it’s finished scanning, the top half of the GitBot interface will be populated with all your git projects in the selected directory in tabular format. You can now work with these projects by selecting row(s) and clicking the “Status”, “Pull”, or “Push” buttons. You can refresh all your projects by clicking the “Refresh” button. You can ‘git push’ all of the listed git projects by clicking the “Push All” button, and run ‘git pull’ on all of the listed git projects by clicking “Pull All”. You can switch your projects directory (the directory that contains all your git projects) by clicking the “Settings” button to open a File System browser window, selecting the directory you want, and clicking the “Open” button.

Description of files in this project

- GitBot.java - The main class source file for GitBot. This class creates an instance of GitBot, sets up the ui using the javax.swing package, manages the settings configuration file, and handles interactive events.
- Inspector.java - The “Inspector” class source file for GitBot. This class handles scanning git projects in your projects directory, opens a shell and runs git commands.
- TableView.java - The “TableView” class source file for GitBot. This class creates a JTable object and DefaultTableModel object and has a single function, clear(), which clears the table’s data.
- README.mkd - The standard README file written in markdown format for github.
- manifest - The jar manifest file used to create GitBot.jar
- GitBot.jar - The executable java archive for GitHub
- icon_128x128.png - The Xdock:icon image file
- icon.ai - The Xdock:icon image source file in Adobe Illustrator format
- splash.png - The splash screen image for GitBot
- splash-black.png - An alternative splash screen image for GitBot
- splash-translucent.png - An alternative splash screen image for GitBot
- all class files - java compiled bytecode files used to create GitBot.jar
- doc - The directory that contains some documentation and the output destination for javadoc.
- doc/CSCI 2912 Final Project.pages - This document, used for getting an “A” in my CSCI 2919 final project.
- doc/CSCI 2912 Final Project.pdf - This document in pdf format
- doc/GitBot UML.pdf - The UML Diagram for GitBot in pdf format
- doc/GitBot UML.graffle - The UML Diagram source file used to create GitBot UML.pdf

Screenshots

The Splash Screen



The dock icon in the OSX dock

The main interface on init GitBot showing the project “GitBot” selected and “ahead of master by 6 commits”