Lab 2: A GitLab Tutorial

We will be using a system called GitLab to provide the means to transfer the work you do on one computer in lab to other computers and finally to transfer your final product to us for grading. This week, before you start working on the actual programming project for the week, we want to take some time to do a little experimenting with GitLab to help you all get familiar with the system.

Some Background
GitLab is designed to help groups of individuals collaborate on the creation of electronic documents. In many cases, the documents are kept on a central server. In many cases the documents are program code.

Each member of the group can download a copy of the shared documents and edit them on their own machine. The process of downloading from the server for the first time is called checkout or cloning. When a member of the group is ready to share the changes made locally with others, the changes are committed and pushed back to the server. When someone wants to include changes that others have pushed to the server, that individual asks git to update to local copy (also called pulling the shared copy).

We will use GitLab in a slightly different way. Rather than providing the means for you to collaborate with other students in the class, GitLab will provide a way to transfer the current copy of your work on a given lab project from one computer to another. There are two common situations where this will be helpful.

First, if you have not completed your work during your scheduled lab period and prefer to complete the work on your own computer, GitLab will enable you to download a current copy of your work from to your machine.

Even if you choose to do all your work in our lab, it may be necessary to use GitLab. Suppose you start your lab work on a computer in the larger of our two labs but don’t finish it during lab period. When you return to complete the work, another course might be holding a scheduled lab in the large lab. The only machines available for you to use might be in the smaller lab. You will discover that if you move to a different machine in our lab, your work does not automatically move with you. Fortunately, you can explicitly use GitLab to download your work to another machine (as long as you didn’t forget to commit and push your work at the end of your previous work session).

The only surprise is that the procedure for downloading your work from the server to a computer is different depending on whether or not you are downloading a particular lab project to the computer for the first time. If it is the first time, you will “checkout” the project from the GitLab server (evolene.cs.williams.edu). Once you have done a checkout, if you return to the machine for an additional work session, you will ask the server to “update” the project. The advantage of making this distinction is that updates are easier than checkouts because GitLab can use information stored within the project it has already placed on the machine rather than requiring you to type in all the details.

A Little Practice
Given this bit of background, let’s practice the process of moving your code from one machine to another while making changes on the various machines.

• Even if you really liked the location of the machine you worked on during last week’s lab, deliberately find a different machine and log in.
• Launch BlueJ.
• BlueJ will start out with an empty project much as it did at the beginning of last week’s lab. (If BlueJ displays your work from last week, then you must be using the same computer as last week. Please move to another computer and try again.)
• In a web browser, visit the page of the course web site that explains how to work with GitLab. (Start by visiting http://www.cs.williams.edu, click on the Labs tab, scroll down to the section on “Using GitLab” and click on the link at the bottom of this section.)
• Follow the instructions for “Checking Out Your Code”, use “lab1” as the project name you enter at the end of these instructions.

The instructions you just followed from the course web page should seem very similar to the instruction you followed at the start of Part II of the lab handout last week. When you checked out the lab 1 project last week, it was an empty project. At the end of that lab, however, you committed and pushed the code you typed in during lab so that it replaced the empty project that had been stored on the GitLab server at the start of last week’s lab. This week’s checkout downloads the version you pushed at the end of last week’s lab.
Now, we want you to make a simple change to your LoginWindow program from last week.

- Double-click on the icon for LoginWindow in the BlueJ project window on your screen. A window in which you can edit the text will appear.
- Edit the LoginWindow class by changing the text displayed in the button from “Authenticate” to “Login”.
- Press the compile button and then run the program to make sure your change works. (To run the program just press the mouse on the LoginWindow icon while holding down the control key and select “new LoginWindow()” from the menu that appears.)
- Now, just as you did at the end of last week’s lab, click the “Commit/Push” button in the BlueJ project window. (You will probably have to click on the little upward pointing gray arrow below the “Compile” button in the BlueJ project window to make the “Commit/Push” button visible.)
- Follow the instructions under the heading “Committing Your Code” on the web page about using Git we had you open at the beginning of this handout.

Now, the version of your program that creates a button labeled with “Login” is stored both on the machine you have been using today and on our GitLab server evolene.cs.williams.edu. What about the other machine’s in our lab. In particular, what version of your project is stored on the machine you used last week? Let’s find out.

- Log out of the machine you have been using.
- As soon as it is free, log in to the machine you used last week. Be patient if someone else is still finishing the preceding steps. If you don’t remember which machine you used last week, ask Tom to show you the picture he took of you during last week’s lab. The background of the picture may jog your memory.
- Open BlueJ.
- Run the LoginWindow program on this machine to see the button’s label.

The button should say “Authenticate”. Even though you changed your program on another machine in the lab and pushed the updated version to GitLab, the machine you used last week still holds the old version of your code.

Now, use your imagination a bit. Suppose that instead of using a different machine in our lab, you had used your own computer to make some changes to your program but didn’t quite finish the job. Then, to finish the lab with help from the TAs, you came back to TCL and logged into the same machine you had previously used. You would want to see the changes that you had made on your own machine, but even if you had remembered to commit and push those changes to GitLab, they won’t automatically be incorporated in the version of your code on the lab machine. This is basically the same situation we have created today by having you modify your code using a lab machine you didn’t use last week.

You might think of fixing this by checking out the project again as we showed you above. This, however, will result in having many projects holding different versions of your work on a single machine. Instead, you can tell BlueJ to simply “Update” the existing project to make it consistent with the latest version pushed to GitLab. To do this:

- Follow the short instructions under the section “Updating Your Code” on the web page describing GitLab that is part of our course web site.

**Summary**

This exercise is designed to familiarize you with how to use GitLab. All you really have to remember is:

- The first time you do work for a particular lab on a new machine (either one of our lab machines or your own machine), you will use “Checkout” to get the latest version of your work on the project onto that machine.
- At the end of any work session (or even more often), you should commit and push your work to the GitLab server.
- If you resume work on a lab using a machine you have already used to work on the lab but is different from the machine you used during your last work session, you will tell BlueJ to “Update” the project at the start of your work.

In addition, when we grade your work, we will grade the latest version uploaded to GitLab. So, always remember to commit and push the version you want graded when you are finished.