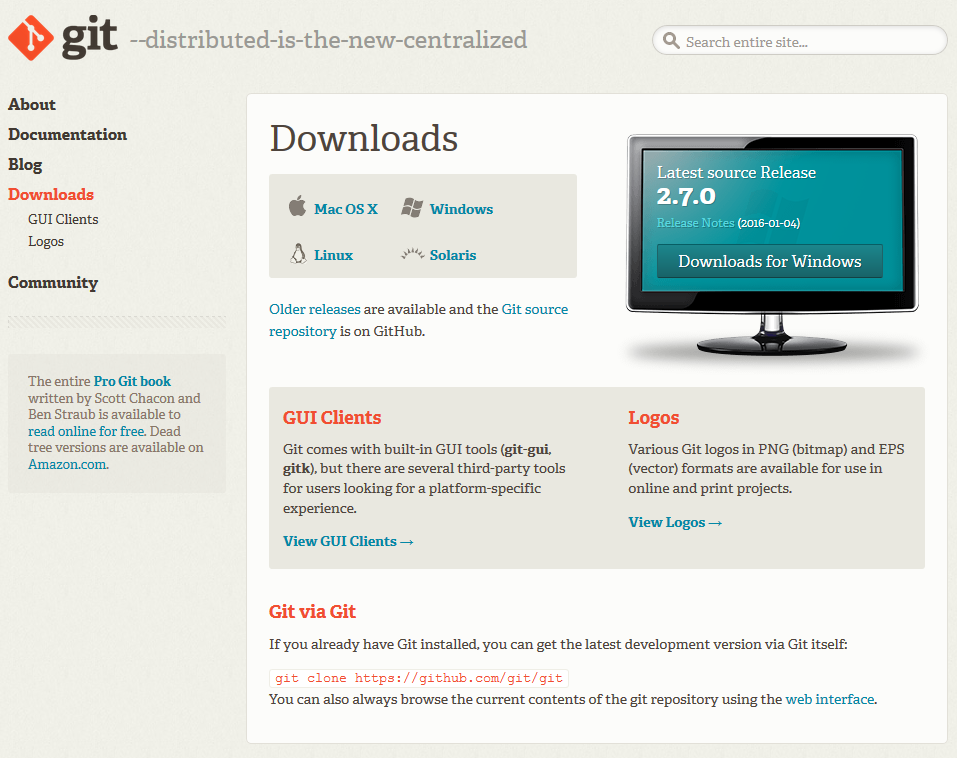
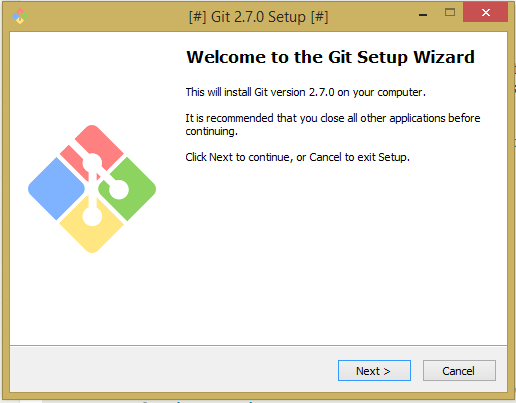
**HOW TO USE GIT-BASH**

**How to install git**

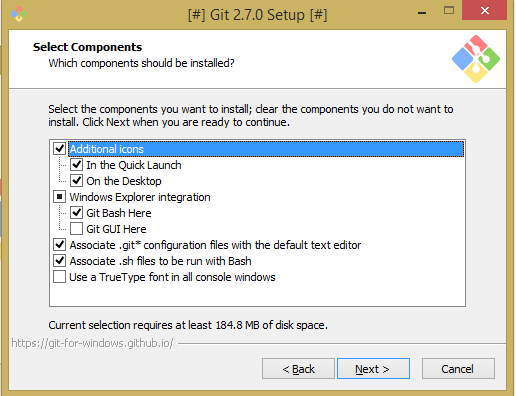
To install git, the command-line tool required for Robotics programming, visit the git website at: [*http://git-scm.com/download*](http://git-scm.com/download)



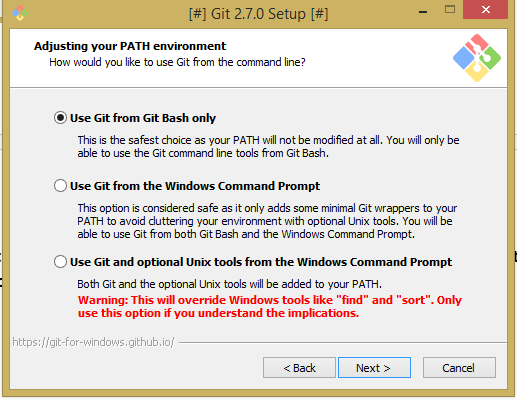
Once there, download the installer that matches your operating system. After it’s downloaded, you will be greeted with an installer.



Proceed through the installation until you reach the **Select Components** section.

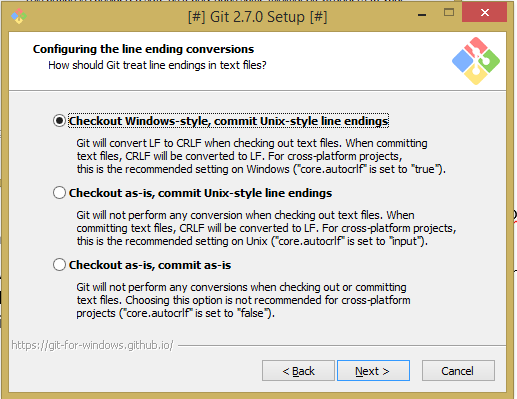


**Ensure the checkbox labelled, “*Git Bash Here”* is enabled! This is essential to be able to collaborate with the team.**

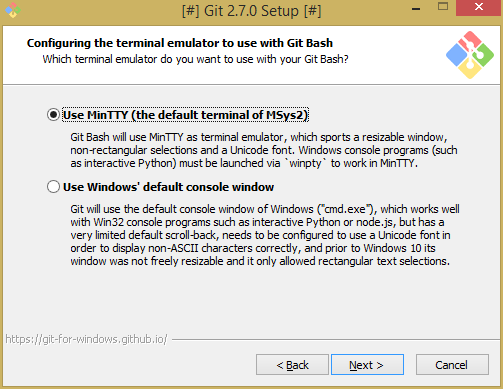


On the ***Adjusting your PATH environment***page, select the first radiobutton if you want to execute git commands through Git’s console.

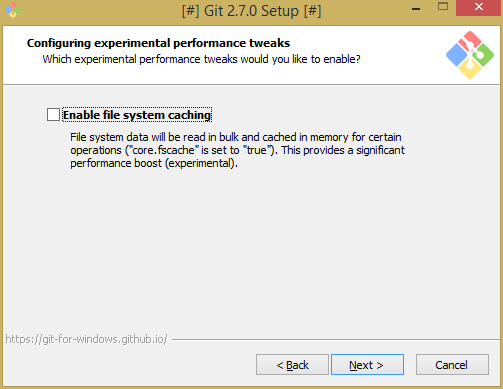
Otherwise, choose the second radiobutton if you want to execute git commands through the Windows Command Prompt (or Terminal, depending on the operating system you are installing it on)



For this section, just choose the first radiobutton.



For this section, you can choose to install MinTTY, a terminal emulator provided by Git. This terminal provides a number of extra features that the default Windows terminal does not have.



Disable file-system caching.

Clicking next will install Git.

**How to setup the git environment**

Once you have Git installed, begin by opening up your terminal of choice. To begin using Git, you have to configure a few settings beforehand. In your terminal, enter these commands:

git config --global user.email "your\_email@provider.com"

Use the email associated with your GitHub account. Next, enter this command:

git config --global user.name "Your Name"

Your name will be used to figure out who made what changes

After this, you must create a new folder in a directory, such as Documents. This directory will be the root of your local master branch for the git. Once you have created the folder, you will need to enter the directory from the terminal window. To do this, open your terminal window, and type:

pwd

This command prints the current working directory that your terminal is looking in. From there, you can access other directories by typing:

cd directory/

Once you are inside the directory you created, type the command:

git init

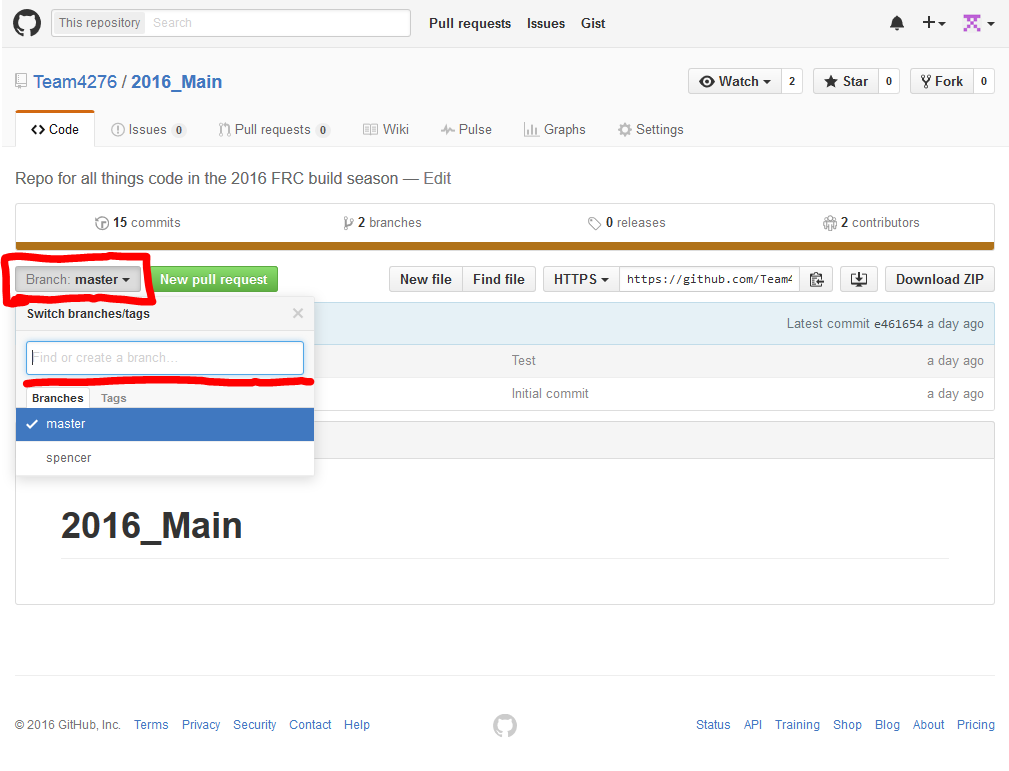
This command creates a blank git repository in that directory. Next, run this command:

git remote add origin <https://github.com/Team4276/2016_Main.git>

This code sets the Robotics’ git URL. Next, run this command:

git branch **branch\_name**

This command creates a branch for you to work with. You should call this branch your name, for example.



Lastly, visit the GitHub website, and create your own remote branch. Click on the Branch button, and type in your desired branch name in the text box. This will be used to store your changes before they are sent to the remote master branch.

**How to pull the most recent changes from the main repository**

To retrieve, or “pull” the most recent changes from the main repository, run this command in your terminal window:

git pull https://github.com/Team4276/2016\_Main.git

This command will pull all of the code from the GitHub’s remote repository, into your local master branch inside the folder.

**How to push changes**

*Before attempting to push changes, please ensure you have the most up-to-date source code, by pulling the repository beforehand.* To add changes that you’ve made to the remote repository, run these commands:

git add .

This command adds all files that you have modified locally into a “staging area”. This staging area is where your files will be prepared for sending. Next, run this command:

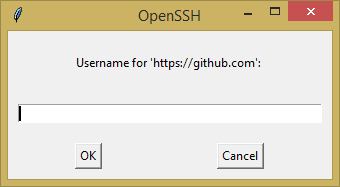
git commit -m "Description of commit"

This command tags all your staged files with that string, so people can see the changes you’ve made. Next, run this command:

git push origin HEAD:**branch\_name**

For the branch name, use the branch you created on the GitHub website.

You will be asked to enter your username and password, just enter them and click, “OK”.



After you have ensured the code works as desired, make sure you are in the master branch, by typing:

git checkout master

Lastly, merge your updated repository with the master remote repository with this command:

git merge origin/**your\_branch\_name**

**Useful terminal commands to know!**

cd *//Changes your current directory*

mkdir directory *//Creates a directory called “directory”*

ls *//Lists all files and folders in the current folder*

git status

git branch

rm -rf FolderName *//Removes folder “FolderName”*

rm testFile *//Removes file, “testFile”*