McGill

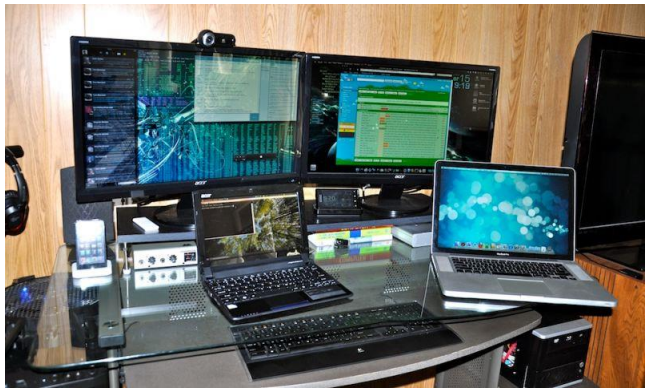
# Revision control

An introduction to Git and Github for ECSE students

Presented by Dirk Dubois

# What is our work environment

- Multiple people working on a software
  - E.g. two partners working on the same lab
- Multiple computers
  - Personal laptop, lab computer, tablet
- How do we share code, effectively work together, and track changes?

# How can we track our files

- Zip files and send them around by email
- Place them in a file syncing service, such as Dropbox, Google Drive, OneDrive
- Place them in a revision control repository, such as Git, mercurial (hg), subversion (SVN)

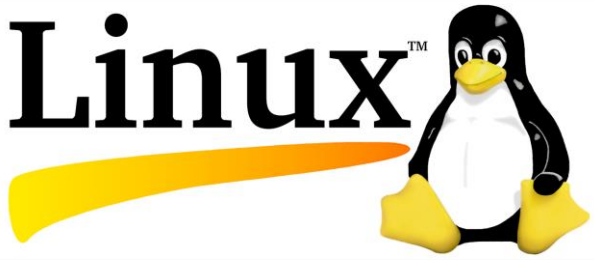
Difficult to track,
No history,
Unsure who has what

Ensures files are up to date
Doesn't keep a good history
Difficult to share
Difficult to revert changes

Designed to track code
Provides detailed history
Multiple people to work simultaneously

# The solution, Git as a repository


- Works on any platform
- Linux
- OS-X
- Windows
- Very popular tool, there are alternatives

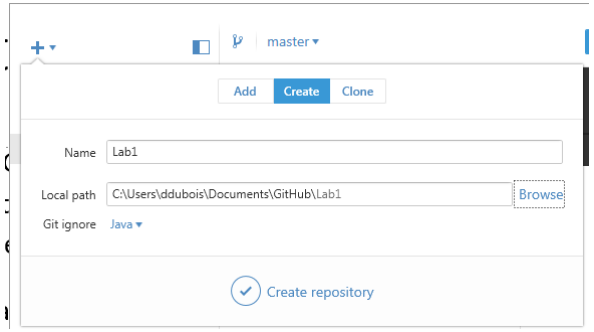
# What does git provide

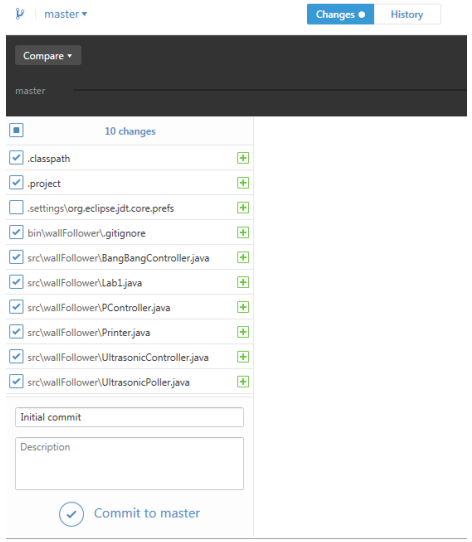
- Tracks history of changes
- Compare versions
- Communicate change
- Multiple workers
- Revert (go back) to old versions

# Creating your first repository

1. Open GitHub for desktop
2. Press the + and select add
3. Browse to the location of your project
4. Select a Git ignore for your programming language, i.e. Java
5. Press create
6. Add source files to the folder
    1. Unzip files provided for a lab, or start adding source files
    2. The location where you created your repo is also where your eclipse project will live
7. Add source files to the repo
    1. Select the changes tab
    2. Select all the source files, *.java, that should be added, include .classpath and .project from eclipse

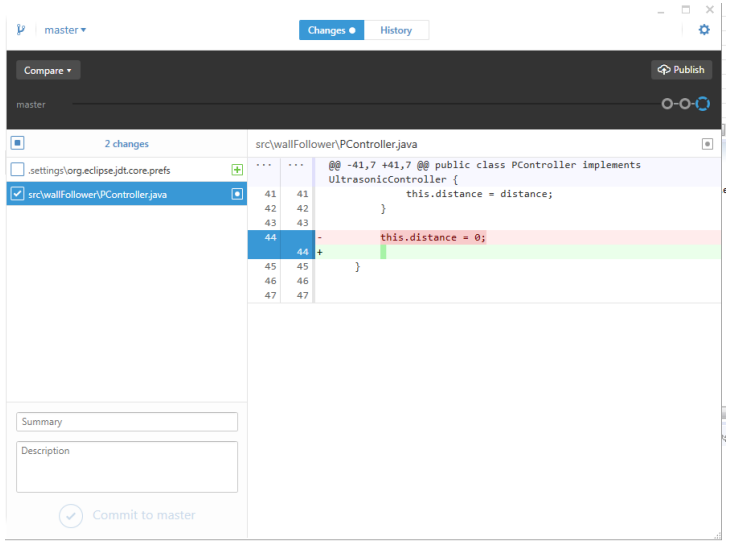
You only need to perform this action once per project

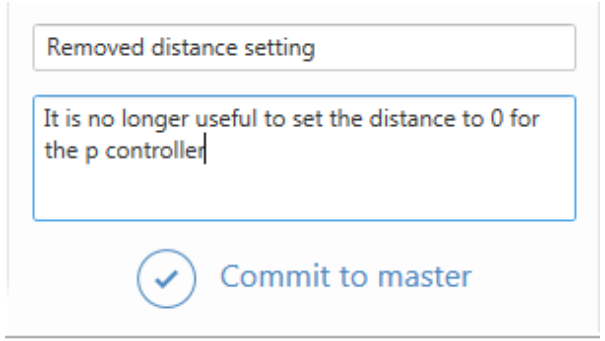


# Committing changes

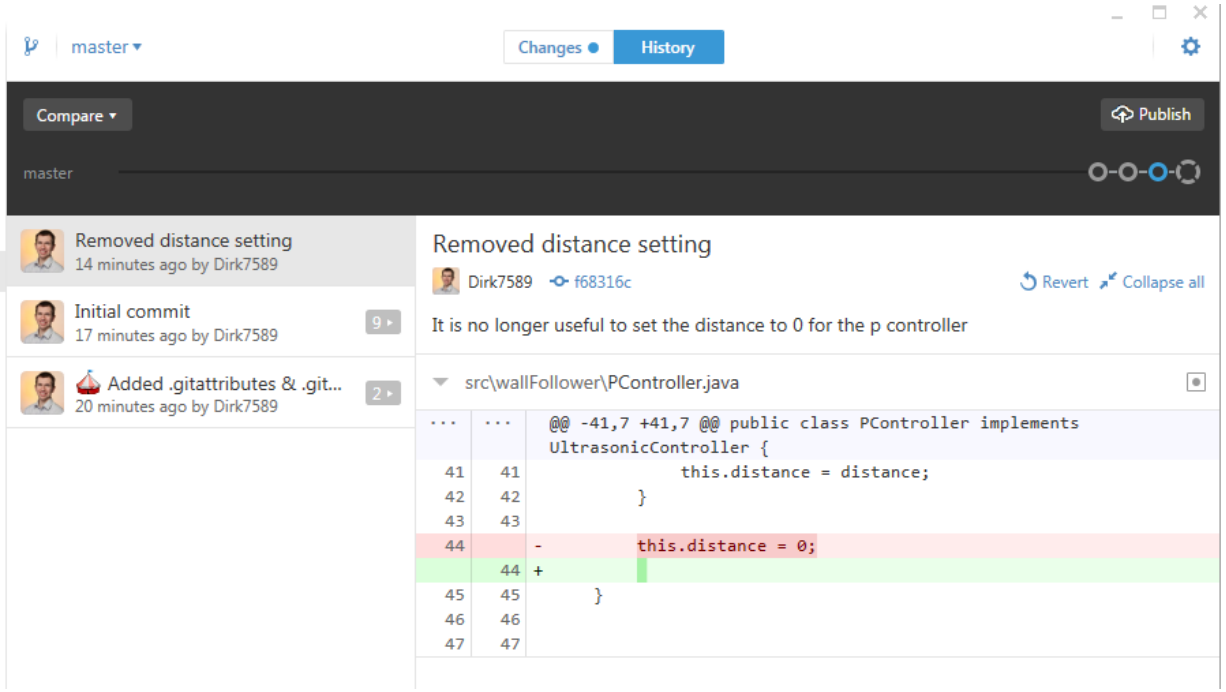
- When you change a source file, Git tracks this change
- You can see this change in the GitHub Desktop tool
- To commit a change (add it to the repository), select the file, fill out the comment, and press commit



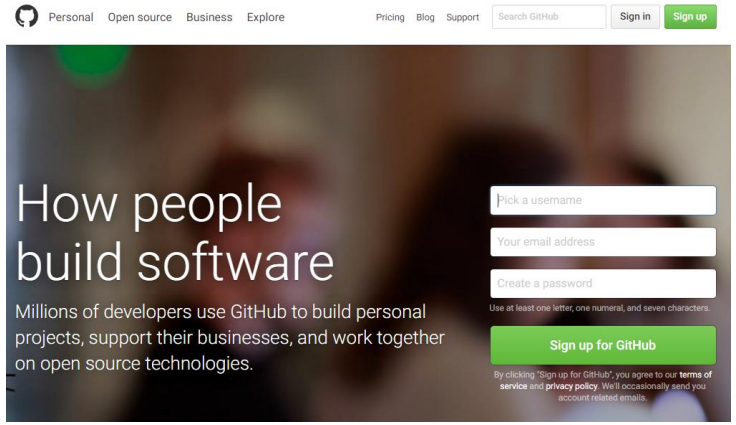
# Looking at history of changes

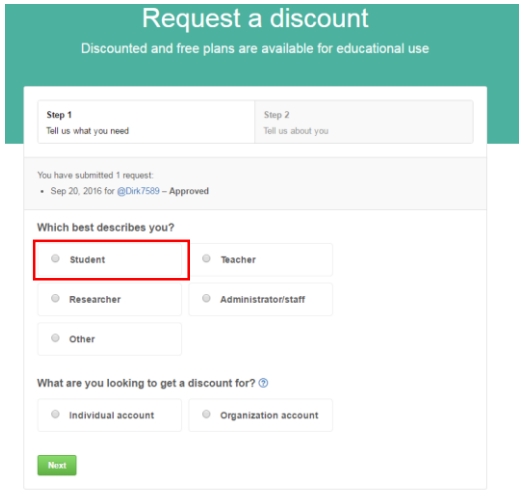
- Go to the history tab
- You can see a list of commits
- You can perform a diff to see all the changes made between those commits
- You can also revert changes from a commit using the revert button



# Github, a central storage of Git repositories

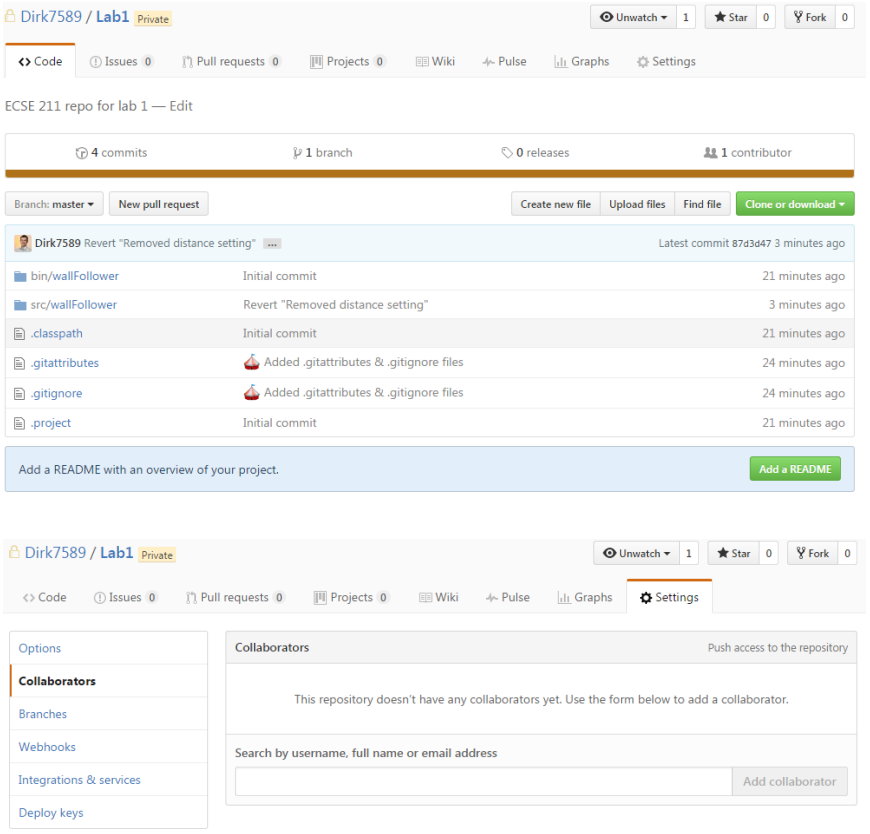
- Provides a cloud server where you can store your code and collaborate on your project
  - Holds your code so that you can share it with others
  - Provides an easy to use web interface so that you can see your code
  - Additional features:
    - Issue tracking
    - Wiki
- Request a student account for private repositories:
  - https://education.github.com/discount_requests/new
  - Use your McGill student email
- Alternative solutions:
  - Bitbucket: https://bitbucket.org
    - Similar to Github



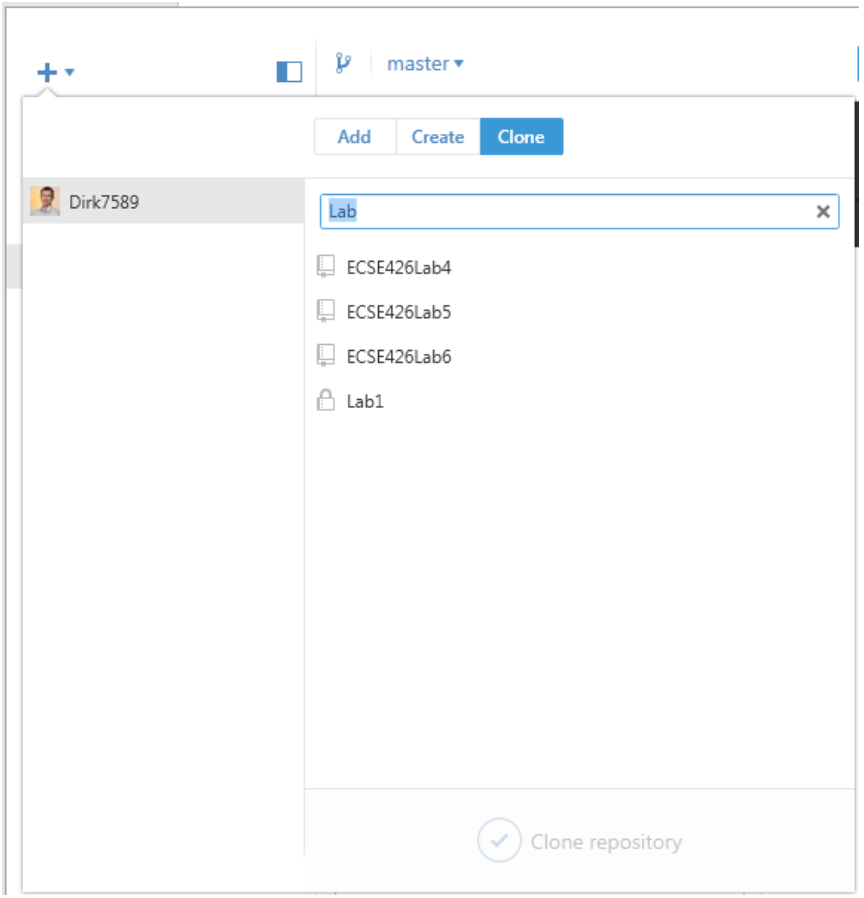
# Adding a repo to Github and collaborating

- To push a repo to your Github, simply select the publish
  - You only need to publish the repo once. All subsequent times you will pull from this repo.
- You can then see your repository on Github
  - If you create a private repo only you and those you explicitly add can see your work (important for labs and project)
  - To add someone to your private repo in Github, select Settings, Collaborators, and add your partners username



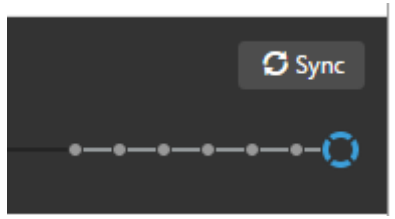
# Clone an existing repo

- Often, one person has created a repo and begun working and you want to contribute to the project. E.g. a partner in a lab group
- In this case we want to clone (get a local copy) of the existing project
- In Github desktop, you can see all your repos and all those repos that have been shared with you
- You can then commit, push and make changes as before



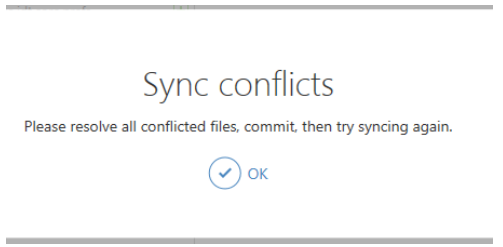
# Pushing and Pulling the repo

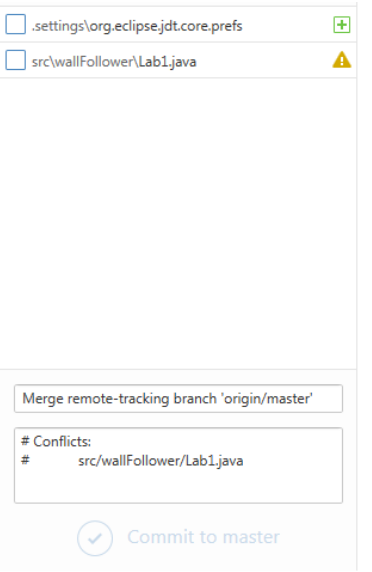
- Since each person is making changes on their own, i.e. commiting and altering files, we need to Push these changes to Github so that everyone can see them
- Once you have committed changes that you want others to see you press the sync button



# Resolving conflicts

- What happens when two people edit the same lines?
- We have to resolve these differences
  - Sometimes the tool can resolve these for us
  - Other times it requires manual intervention
  - You might have to open the file in eclipse and select which changes are the most appropriate




```
57          @SuppressWarnings("resource")                               // Because we don't bother to close
58  <<<<<<< HEAD
59          SensorModes ultraSonicSensor = new EV3UltrasonicSensor(usPort);     // usSensor is the instance
60          SampleProvider usDistance = ultraSonicSensor.getMode("Distance");   // usDistance provides sampl
61  =======
62          SensorModes ultraSensor = new EV3UltrasonicSensor(usPort);      // usSensor is the instance
63          SampleProvider usDistance = ultraSensor.getMode("Distance");    // usDistance provides samples f
64  >>>>>>> origin/master
65          float[] usData = new float[usDistance.sampleSize()];        // usData is the buffer in which dat
```

# Additional Resources

- Tutorials
  - Interactive git demo https://try.github.io/
    - Only takes 15 minutes and helps explain how to use git command line
  - Atlassian tutorial on git https://www.atlassian.com/git/tutorials/
    - What I presented today
- Tools:
  - Graphical tools for git:
    - Git for windows https://git-for-windows.github.io/
      - Simple, easy to use, has command line alternative
    - GitHub Desktop https://desktop.github.com/ (Demoed today)
      - Integrates with github, provides simple interface for git
    - SmartGit http://www.syntevo.com/smartgit/
      - Supports multiple platforms, full featured, free community addition
    - SourceTree https://www.sourcetreeapp.com/
      - Similar to SmartGit, provides similar features, also has a community addition