

Git Workshop

Fall 2020

Jack Leightcap¹² Connor Northway²

¹IEEE – nuieeeofficers@gmail.com

²Wireless Club – nuwirelessclub@gmail.com

December 7, 2020

Intro: Workshop Structure

Structure:

- 1 Talk: how does Git work?
- 2 Example: managing homework
- 3 Hands-on: solve Git 'puzzles' in browser

General:

- Ask questions! (want to know more? something isn't clear?)
- Reach out: `neuwireless.slack.com`

Background: What is Git?



Figure: Git Logo — CC BY 3.0

- broadly; *a tool used to track changes to files and folders.*
- facilitates collaboration on software projects
- captures 'snapshots' of a project
- maintains metadata
 - what was changed
 - who was it changed by
 - when was it changed
 - messages associated with changes

Background: What is Git used for?

Group Applications (Industry, co-op)

- large software projects
- resolve conflicts when multiple people are editing the same things
- who wrote this!?

Personal Applications

- “I swear this worked 10 minutes ago. . .”
- find what broke something and when
- separate tasks; work on bug fixing is isolated from work on new feature
- can use for class work!

Background: Big Idea of Git

QUESTION: Intuition

based on this description of what Git does, can you imagine how you would go about making these 'snapshots' manually?



homework1-final-FINAL-
HAVEMERCY.cpp

Background: Big Idea of Git

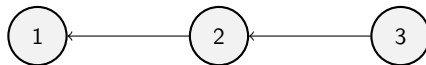


Figure: Linear History

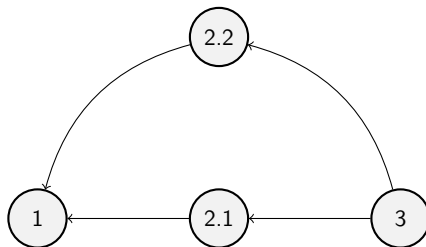


Figure: Branched History

Example: Managing Local Files

EXAMPLE: Homework Files

- ① *Init*: make some files, 'save' them
- ② *Commit*: changes, save those as well
- ③ *Checkout/Restore*: whoops, deleted something, restore it
- ④ *Branch*: new feature, make a new branch
- ⑤ *Merge*: done, merge back



Figure: XKCD 1597 'Git' — CC BY-NC 2.5

See Also

- <https://wyag.thb.lt/> – implement git from scratch in Python