Git Workshop

Fall 2020

Jack Leightcap[1,2] Connor Northway[2]

[1]IEEE – nuieeeofficers@gmail.com

[2]Wireless Club – nuwirelessclub@gmail.com

December 7, 2020

Intro: Workshop Structure

Structure:

1. Background: how does `Git` work?
2. Example: managing homework
3. Example: larger project
4. Hands-on: solve `Git` 'puzzles' in browser

General:

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

Intro: Teaching Git

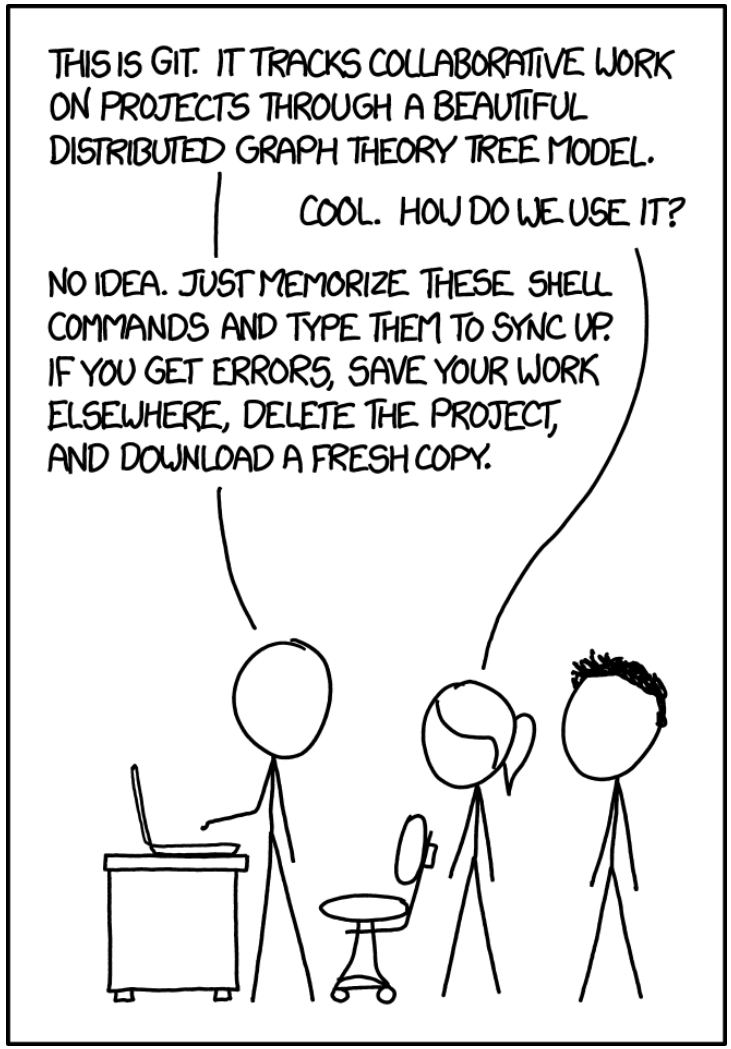

Figure: XKCD 1597 'Git' — CC BY-NC 2.5

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`

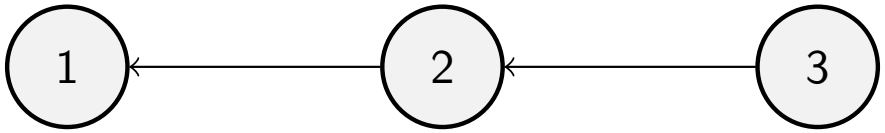

Figure: Linear History

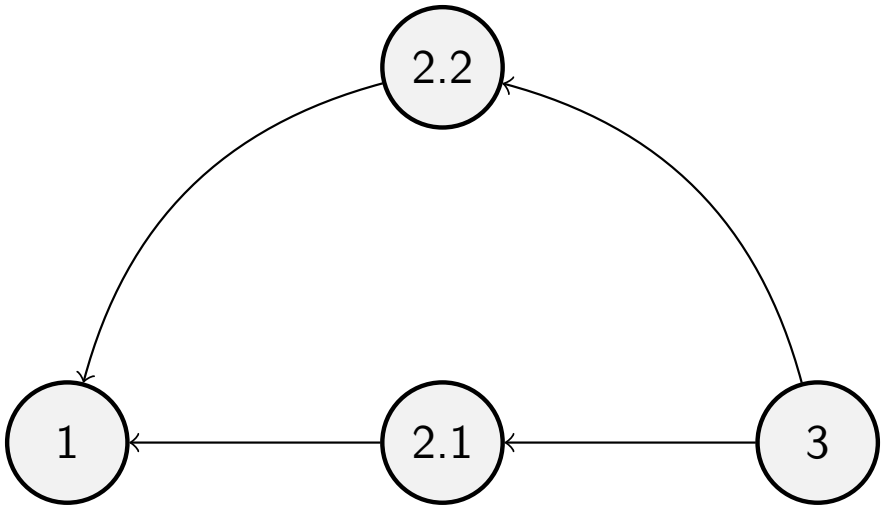

Figure: Branched History

Example: Managing Local Files

EXAMPLE: Homework Files

1. *Init*: make a file
2. *Add/Commit*: save file
3. *Branch/Checkout*: two new features
4. *Merge*: combine those new features
5. *Restore/Checkout*: restore save

Background: Git vs. GitHub?

What's the difference between Git and GitHub?

Git

- tool independent of GitHub
- provides interface for services like GitHub

GitHub

- centralized location to store git repositories
- alternatives: GitLab, SourceHut, BitBucket, self hosting, ...
- additional project management utilities
- CI, CD, hosting, ...

Example: Large Project

EXAMPLE: Larger Project

- Development workflow
- More complex histories

```
[cnorthway@sff-f33 git_workshop]$ git l
* 4a4caa6 (HEAD -> master) hello world
[cnorthway@sff-f33 git_workshop]$ git checkout -b fizzbuzz
Switched to a new branch 'fizzbuzz'
```

Figure: Create and checkout 'fizzbuzz' feature branch

```
[cnorthway@sff-f33 git_workshop]$ git add main.c
[cnorthway@sff-f33 git_workshop]$ git status
On branch fizzbuzz
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   main.c

[cnorthway@sff-f33 git_workshop]$ git commit
```

Figure: Make some commits

```
[cnorthway@sff-f33 git_workshop]$ git l
* fc07086 (HEAD -> fizzbuzz) generalize output strings
* 908b67c generalize input numbers
* b8d5d8e implement basic fizzbuzz
* 4a4caa6 (master) hello world
```

Figure: Commits on feature branch, no other branches

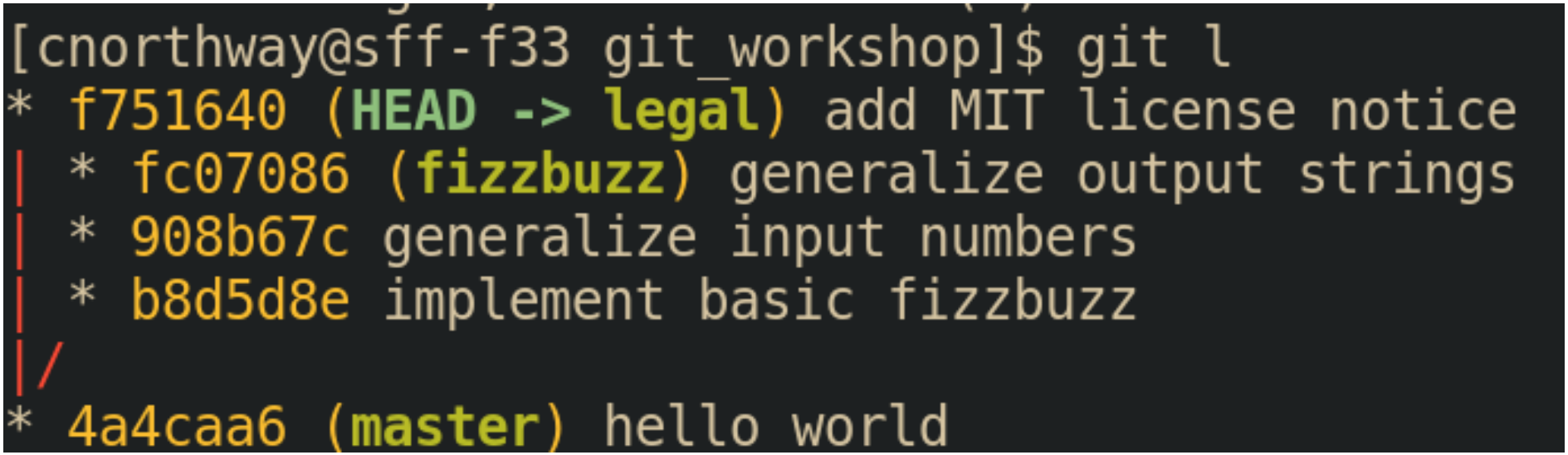

Figure: Legal team makes new branch from master, makes changes

```
[cnorthway@sff-f33 git_workshop]$ git l
* 046c675 (HEAD -> fizzbuzz) add friendly message
* fc07086 generalize output strings
* 908b67c generalize input numbers
* b8d5d8e implement basic fizzbuzz
| * f751640 (legal) add MIT license notice
|/
* 4a4caa6 (master) hello world
```

Figure: We add another commit

```
[cnorthway@sff-f33 git_workshop]$ git l
* 046c675 (fizzbuzz) add friendly message
* fc07086 generalize output strings
* 908b67c generalize input numbers
* b8d5d8e implement basic fizzbuzz
| * f751640 (HEAD -> legal) add MIT license notice
|/
* 4a4caa6 (master) hello world
[cnorthway@sff-f33 git_workshop]$ git checkout master
Switched to branch 'master'
[cnorthway@sff-f33 git_workshop]$ git merge legal
Updating 4a4caa6..f751640
Fast-forward
 main.c | 24 ++++++++++++++++++++++++
 1 file changed, 24 insertions(+)
[cnorthway@sff-f33 git_workshop]$ git l
* 046c675 (fizzbuzz) add friendly message
* fc07086 generalize output strings
* 908b67c generalize input numbers
* b8d5d8e implement basic fizzbuzz
| * f751640 (HEAD -> master, legal) add MIT license notice
|/
* 4a4caa6 hello world
```

Figure: Merge legal change to master

```
[cnorthway@sff-f33 git_workshop]$ git merge fizzbuzz
Auto-merging main.c
Merge made by the 'recursive' strategy.
 main.c | 23 ++++++++++++++++++++++-
 1 file changed, 22 insertions(+), 1 deletion(-)
```

Figure: Merge fizzbuzz – lucky, no conflicts!

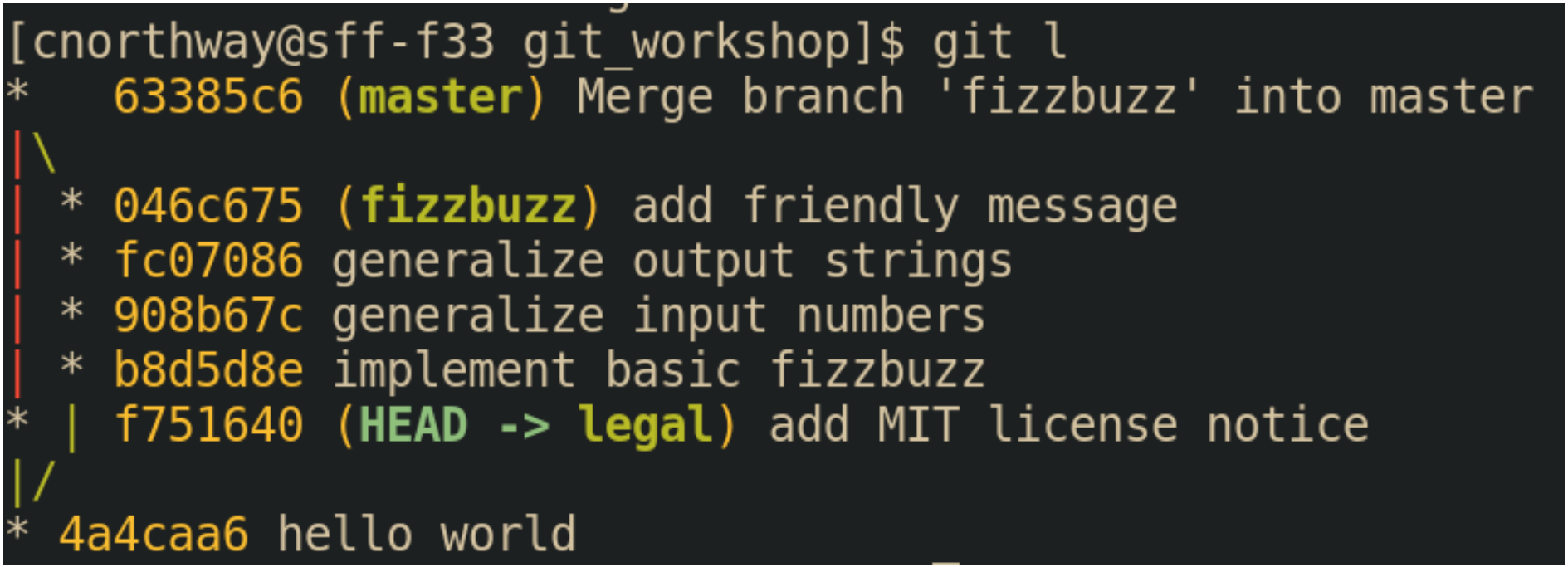

Figure: Result of merging fizzbuzz – merge commit with two parents

Hands-On: learngitbranching

HANDS-ON: learngitbranching

https://learngitbranching.js.org/

See Also

General

- MIT Missing Semester Lecture 6: Version Control (Git)
- Tim Pope: A Note About Git Commit Messages
- Implement git from scratch in Python
- How to contribute to code on GitHub

Git Documentation

- Git configuration documentation
- `.gitignore` documentation