NHERI SIMCENTER PROGRAMMING BOOTCAMP

JULY 30 THROUGH AUGUST 3, 2018, AT UC BERKELEY'S RICHMOND FIELD STATION

- **Using GitHub**

REVISION CONTROL / SHARED DEVELOPMENT

- Problem:
    - You develop software:
        - Using office computer and home computer
        - Want to try a new feature without breaking the code
        - Want to incorporate help from colleagues; share with advisor; …
        - Need to reuse portions in other projects
        - Want to keep backups of all changes
        - Need to review changes
- Solution:
    - Use a Revision control system

REVISION CONTROL: GITHUB

- GitHub
  - Get your free account on http://github.com

- Basic Idea (simplified):
  - Store change sets (do not copy files or directory trees)
  - Preserve and make every stage of your development accessible
  - Store those change sets on the cloud (backup of everything ☺)
- Key Features:
  - Automated merging (push/pull/fetch/merge)
  - Branching: try something without breaking the main development
  - Integrate or share changes by others without manual work

REVISION CONTROL: GITHUB

REVISION CONTROL: GITHUB

Project

Files And Folders

GITHUB (SIMPLE)

On GitHub.com

master

On your computer

master

clone

Push

Pull

Commit #1   Commit #2

REVISION CONTROL: GITHUB

Project

Clone
(fetch a
local copy)

Files
And
Folders

REVISION CONTROL: GITHUB

- What goes on GitHub?
    - Everything needed to build/compile your code
        - .c, .cpp, .h, .pro, .tex
        - documentation

- What does NOT go onto GitHub?
    - Everything the compiler generates
        - .o, .aux, .log, .pdf

GITHUB DESKTOP
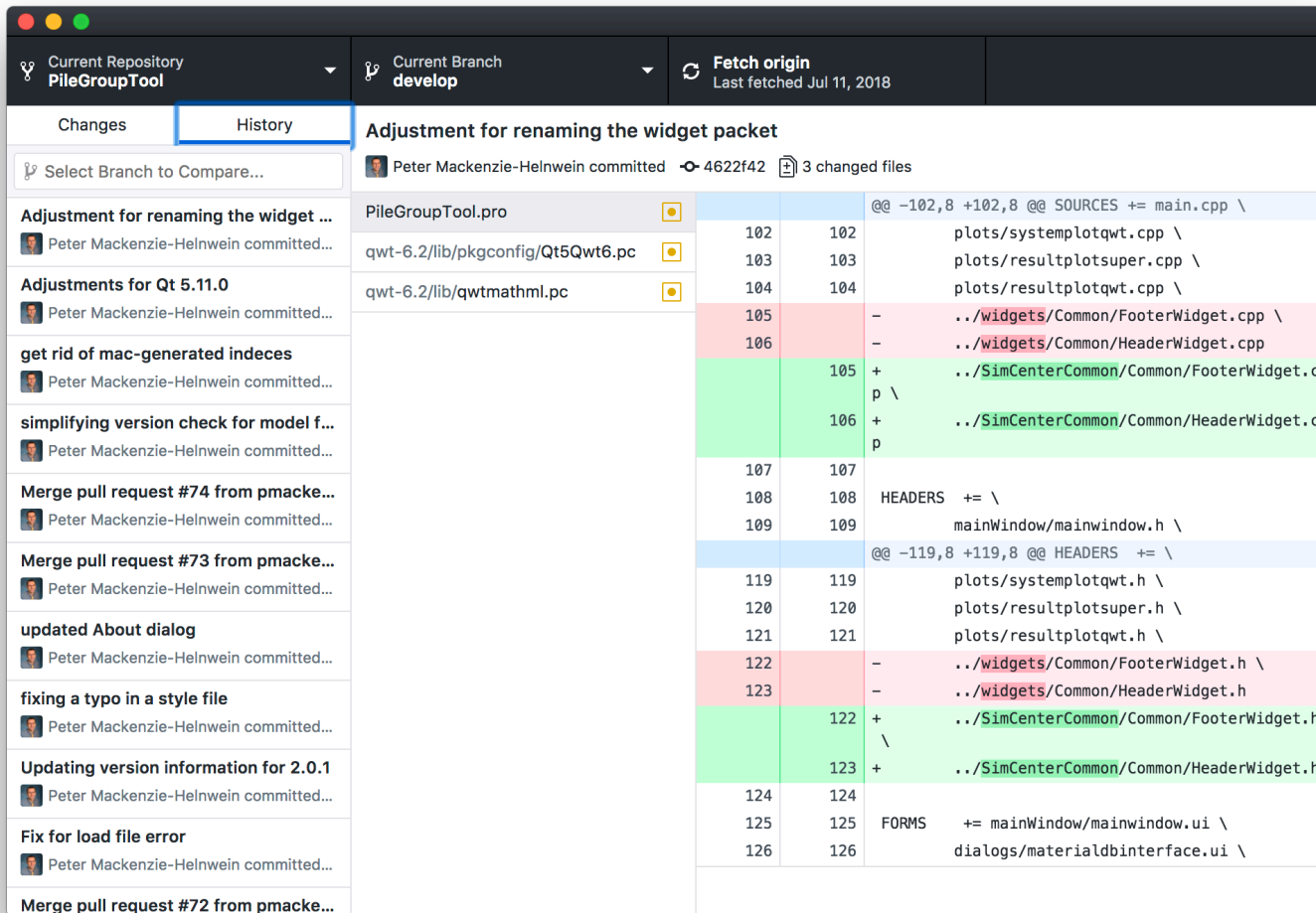
Performs most functions to communicate between your local computer and github.com

Guides you through the process and enforces comments on commit ☺

GITHUB (2 COMPUTERS)

On your computer #1

master

Commit #3

Commit #4

Pull

push

On GitHub.com

master

On your computer #2

clone

Push

pull

master

Commit #1

Commit #2

REVISION CONTROL: GITHUB

NHERI-SimCenter / PileGroupTool
alborzgh / QtPile
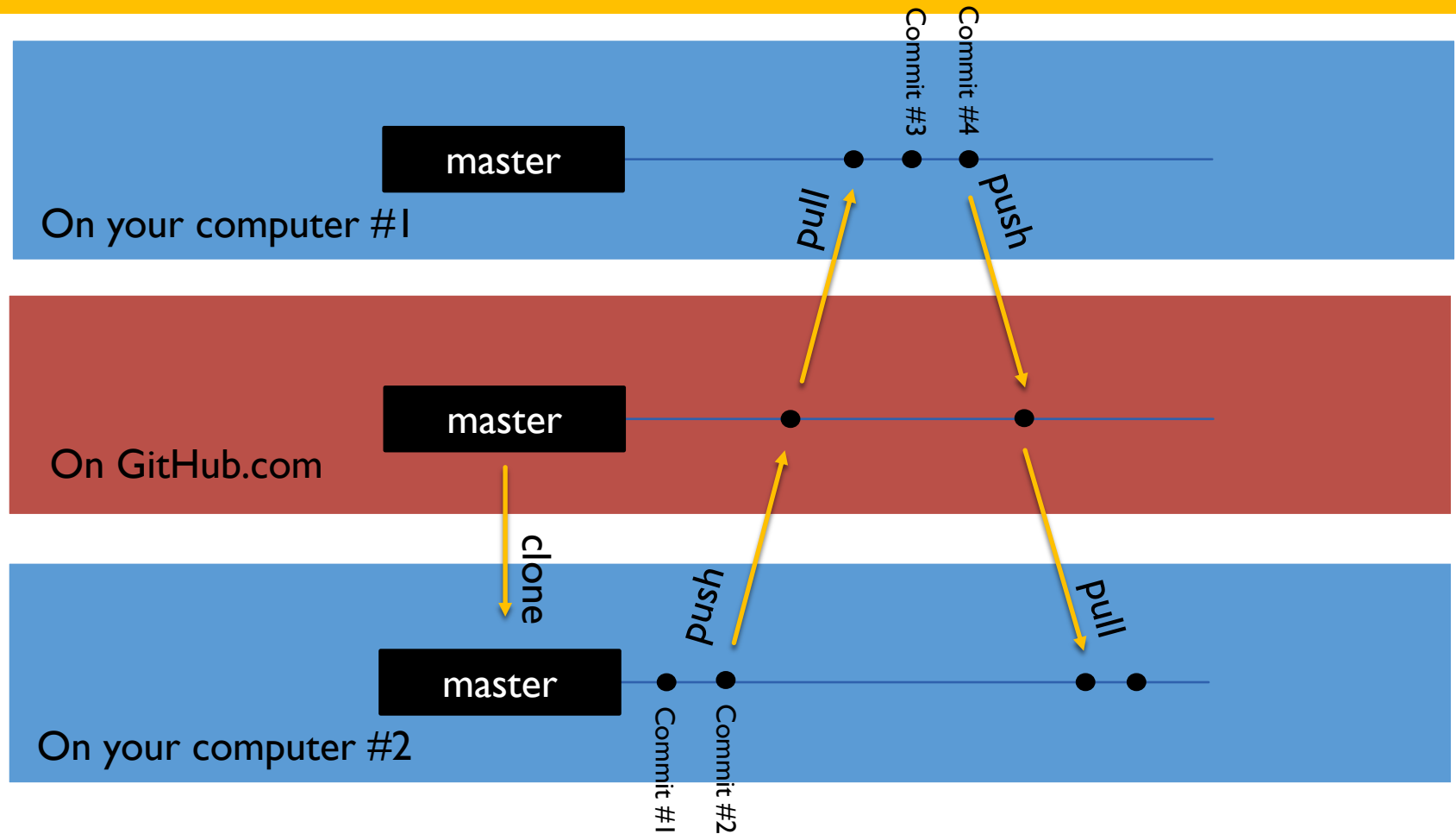aschellenberg74 / QtPile
chrshffmn / QtPile
fmckenna / QtPile
m4xwellNo4h / QtPile
ming91915 / PileGroupTool
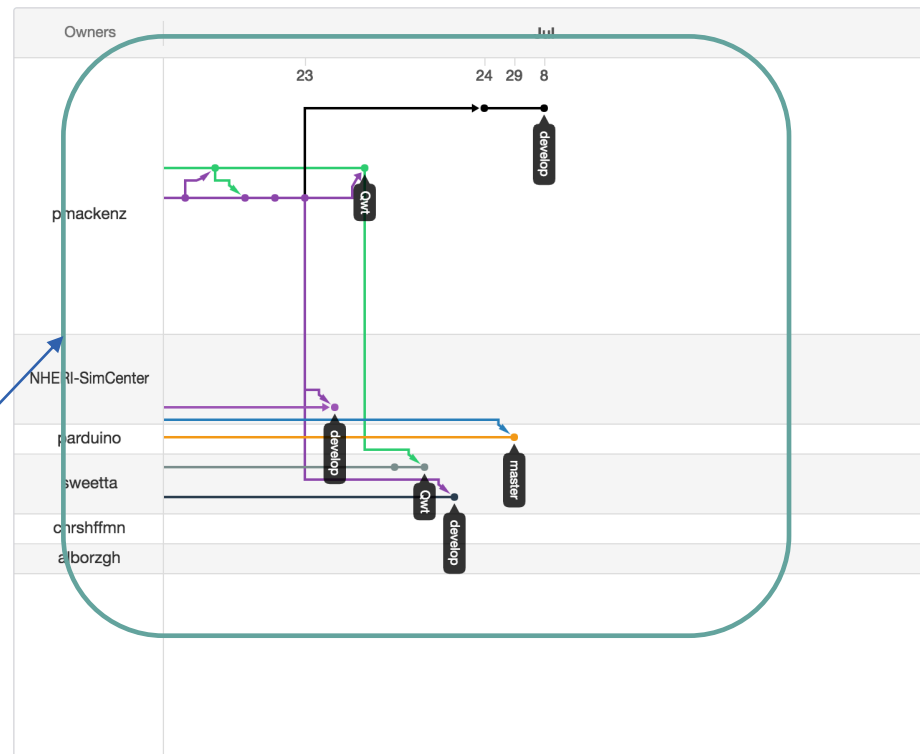parduino / PileGroupTool
pmackenz / PileGroupTool
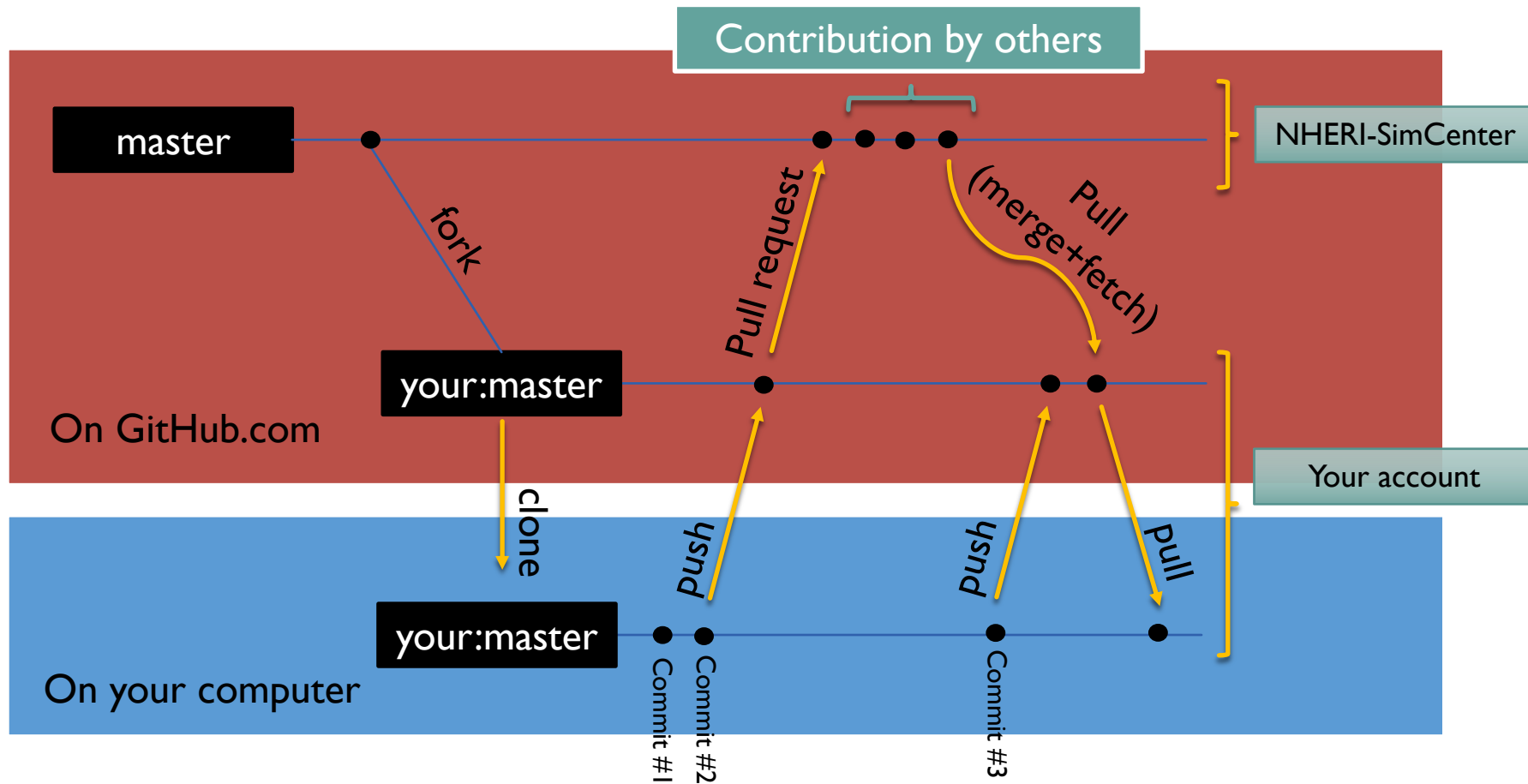QingleCheng / PileGroupTool
sweetta / PileGroupTool
xiaogh98 / PileGroupTool

Developers

Development tree

GITHUB (MONITORED PROJECT)

Contribution by others

NHERI-SimCenter

master

fork

your:master

Pull request

Pull (merge+fetch)

On GitHub.com

Your account

clone

Push

Push

pull

your:master

Commit #1

Commit #2

Commit #3

On your computer

REVISION CONTROL: GITHUB

REVISION CONTROL: GITHUB

- Collaboration Guidelines

  - In the code: Comments, comments, comments

  - Commit frequently and sync frequently to avoid conflicts

  - Every commit deserves a proper description. NO commit without comments!

  - Only commit IF code compiles (or use private branch)

  - WHO BREAKS THE CODE HAS TO HANG A RUBBER CHICKEN OVER THE DOOR TILL BUG IS FIXED (blame game)

  - Unstable development belongs into a branch

  - Branches need to be reintegrated into main development once stable and compatible.

GITHUB.COM

- When to use the web interface?
    - Creating pull requests to the monitored project
    - Submitting and reviewing issues
    - Statistics
    - Viewing revision tree details

- Other options:
    - Use the git command line tool (most powerful but most complex)

EXERCISE

1. Create a private fork of
   https://github.com/NHERI-SimCenter/SimCenterBootcamp

2. Clone to your machine

3. Add a folder (your_name_without_spaces) to your branch

4. Add a README.txt file to this folder

5. Commit (locally)

6. Push to your fork on github.com