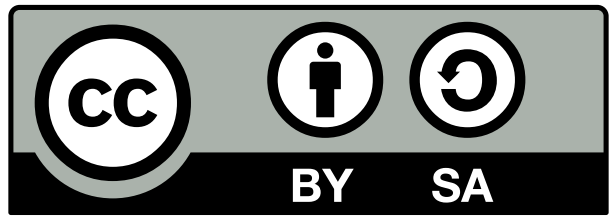


This presentation is released under the terms of the
Creative Commons Attribution-Share Alike license.

You are free to reuse it and modify it as much as you want as long as
(1) you mention me as being the original author,
(2) you re-share your presentation under the same terms.

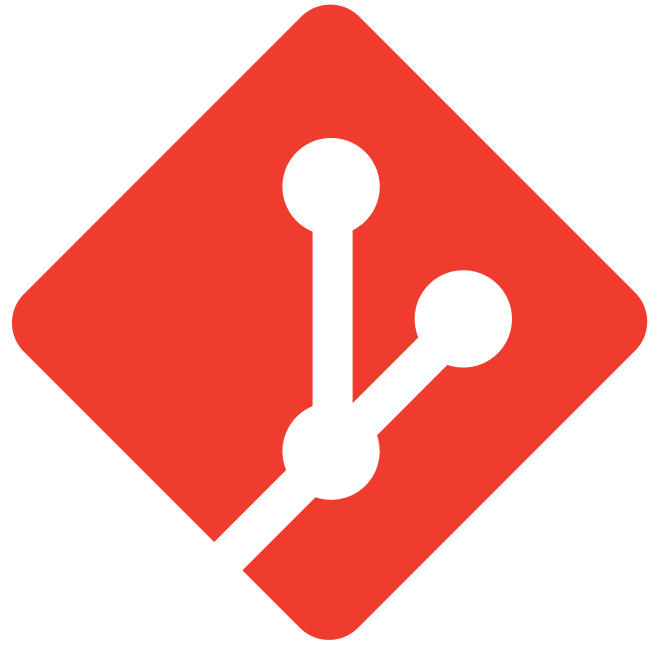
You can download the sources of this presentation here:

https://github.com/severin-lemaignan/git-presentation/

WITH PLYMOUTH UNIVERSITY

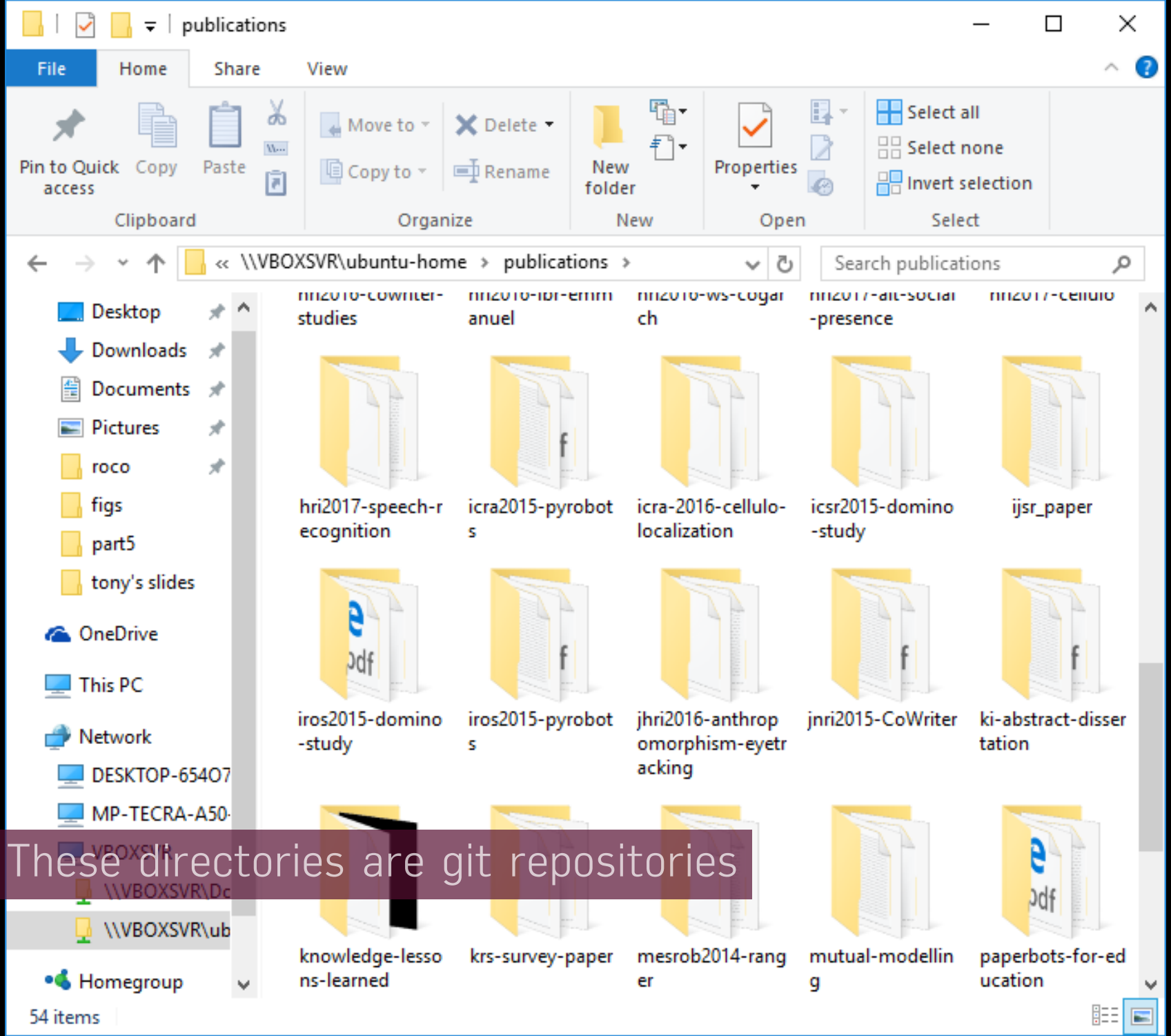
git
the basics

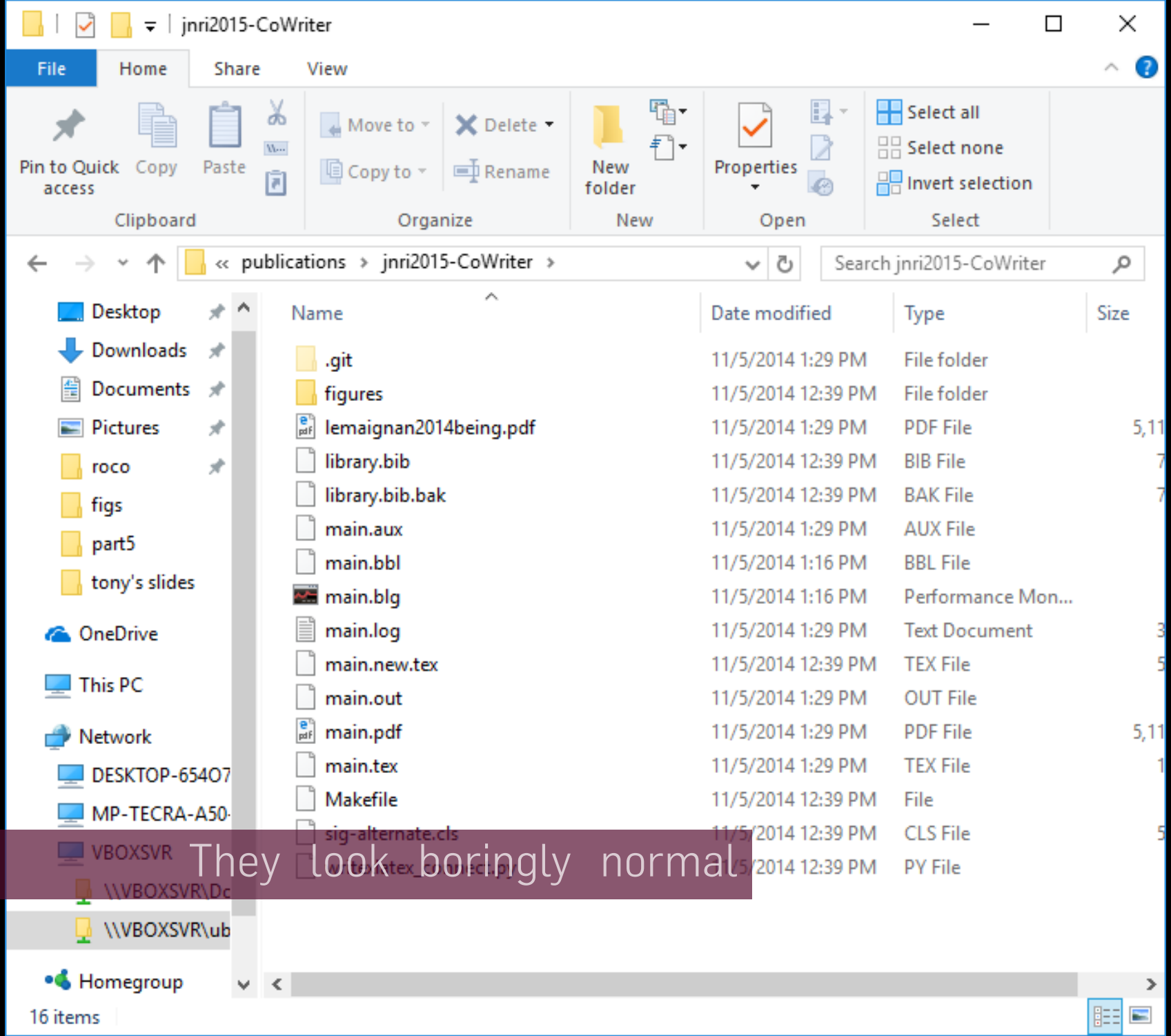
25 Jan. 2017



Séverin Lemaignan

Centre for Robotics & Neural Systems
Plymouth University

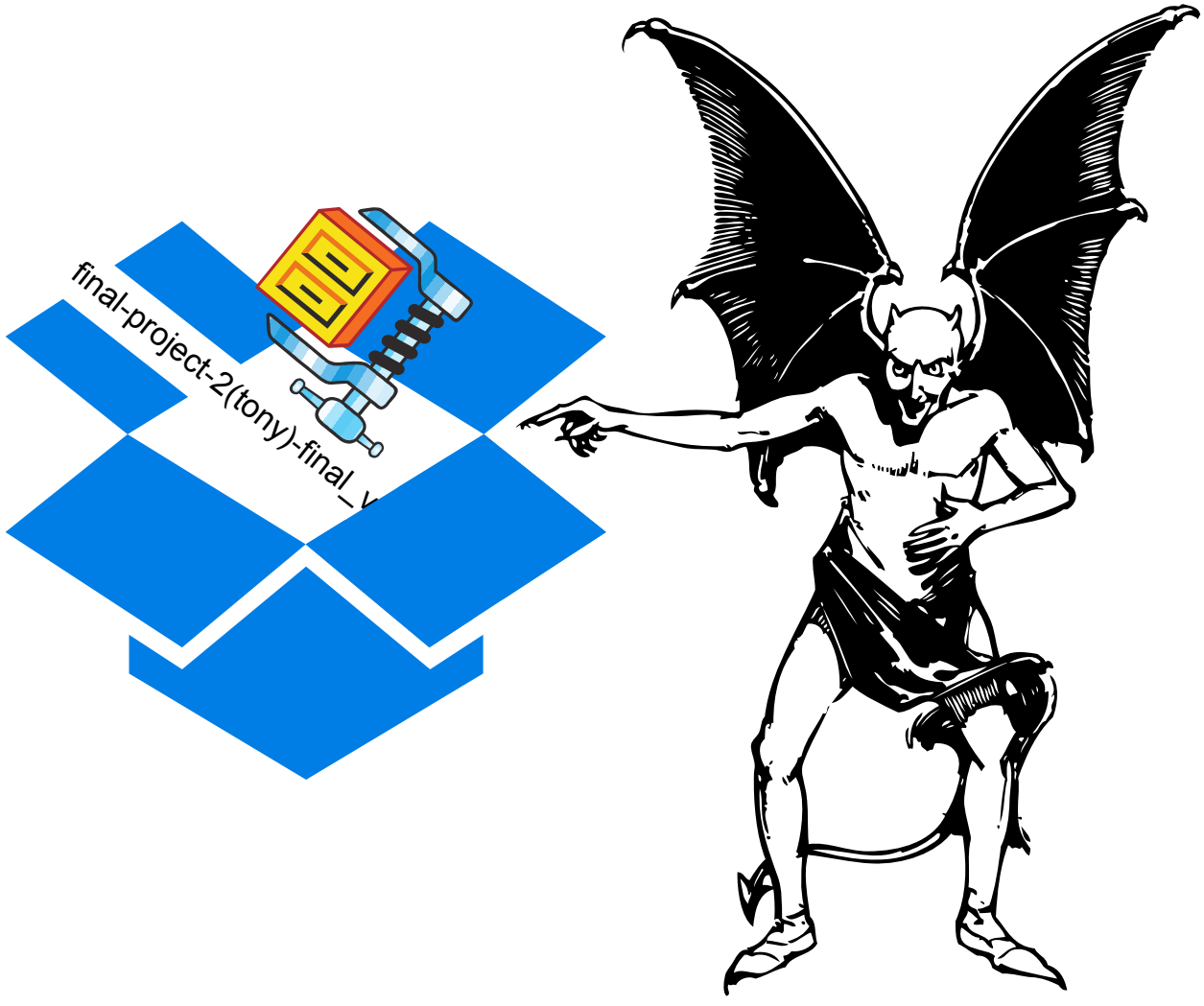

publications
File
Home
Share
View
Pin to Quick access
Copy
Paste
Clipboard
Move to
Copy to
Delete
Rename
Organize
New folder
New
Properties
Open
Select all
Select none
Invert selection
Select
\\VBOXSVR\ubuntu-home › publications ›
Search publications
Desktop
Downloads
Documents
Pictures
roco
figs
part5
tony's slides
OneDrive
This PC
Network
DESKTOP-654O7
MP-TECRA-A50
\\VBOXSVR\ub
Homegroup
54 items
studies
anuel
ch
-presence
hri2017-speech-recognition
icra2015-pyrobots
icra-2016-cellulo-localization
icsr2015-domino-study
ijsr_paper
iros2015-domino-study
iros2015-pyrobots
jhri2016-anthropomorphism-eyetracking
jnri2015-CoWriter
ki-abstract-dissertation
knowledge-lessons-learned
krs-survey-paper
mesrob2014-ranger
mutual-modelling
paperbots-for-education
These directories are git repositories



They look boringly normal

A few roundtrips later with colleagues...

final-project-2(tony)-final_v


final-project-2(tony)-final_v

facebook

Search

Home Profile Account

Get Started
Wall
Hidden Posts
Info
Listings
Photos
Dan's Welcome Page
Discussions
Edit

About Edit

7
people like this

Add to My Page's Favorites

Tony King B.

Real Estate · Toronto, Ontario Edit Info

Edit Page

Wall Tony King B. · Most Recent

Share: Status Photo Link Video

Write something...

Tony King B.
Paul, you can take my change below:

```
using namespace std;
using namespace cv;
HeadPoseEstimation::HeadPoseEstimation(const string& face_detection_model, float focalLength) :
        focalLength(focalLength),
        opticalCenterX(-1),
        opticalCenterY(-1)
{
        // Load face detection and pose estimation models.
        detector = get_frontal_face_detector();
        deserialize(face_detection_model) >> pose_model;
```

51 Impressions · 0% Feedback
Tuesday at 2:25pm via re2social · Like · Comment

Tony King B.
SVN is really cool, but I like Facebook better!

51 Impressions · 0% Feedback
Tuesday at 2:25pm via re2social · Like · Comment

Admins (4) [?] See All

Use Facebook
Promote with an Ad
View Insights
Suggest to Friends

You

Tony King B. likes this.

Quick Tips

Get more people to like your Page with Facebook Ads today!

Get More Connections

Sample Ad

The text of your ad will go here.

Like · JP Zeni likes this.

facebook
Search
Home
Profile
Account
Edit Page
Get Started
Wall
Hidden Posts
Info
Listings
Photos
Dan's Welcome Page
Discussions
Edit
About
Edit
7
people like this
Add to My Page's Favorites
See All
Sample Ad
The text of your ad will go here.
Like · JP Zeni likes this.
51 Impressions · 0% Feedback
Tuesday at 2:25pm via re2social · Like · Comment

We can do better!

We can do better!

git is essentially about recording the history of files

We can do better!

git is essentially about recording the history of files

(and who did what)

We can do better!

git is essentially about recording the history of files

(and who did what)

(and sharing as well)

VERSIONING

WHY VERSIONING?

- The history of your development/document
- Compare the current code with an older version
- Roll-back to previous versions
- Experiment without losing anything
- Trace who did what (at the level of the line of code)
- Annotate your workflow (important milestones, etc)
- Avoid catastrophes!

ATOMIC COMMITS

The single most important concept (because it requires to think about development/writing in terms of **functional units**):

Atomic commit

A (typically small) commit that represent a **single, coherent & complete** functional change.

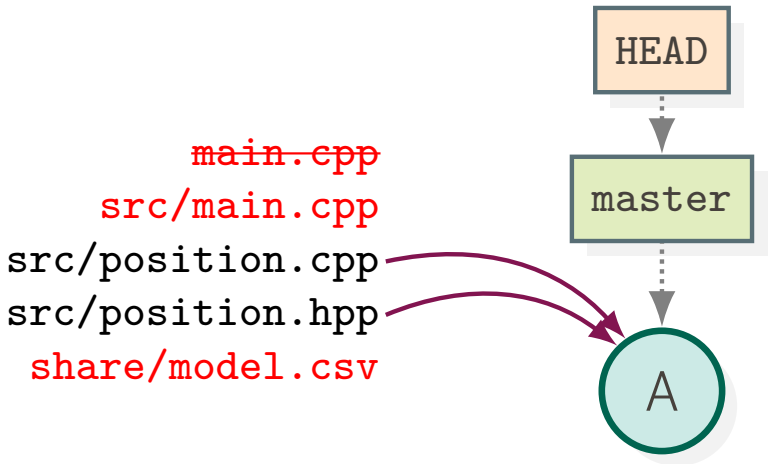
ATOMIC COMMITS

The single most important concept (because it requires to think about development/writing in terms of **functional units**):

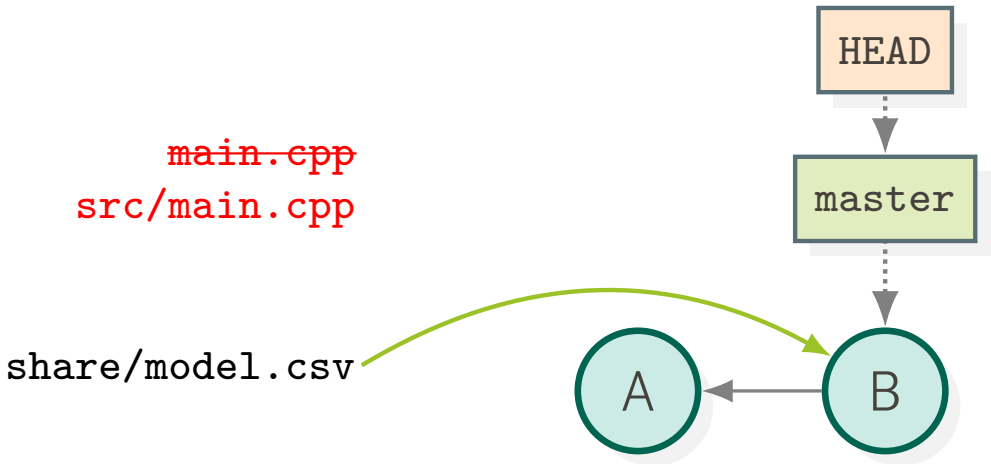
Atomic commit

- Easy to understand the change
- Debugging made easy (`git bisect`)
- Collaboration made easy (less, smaller conflict)
- Easy to write a useful commit message

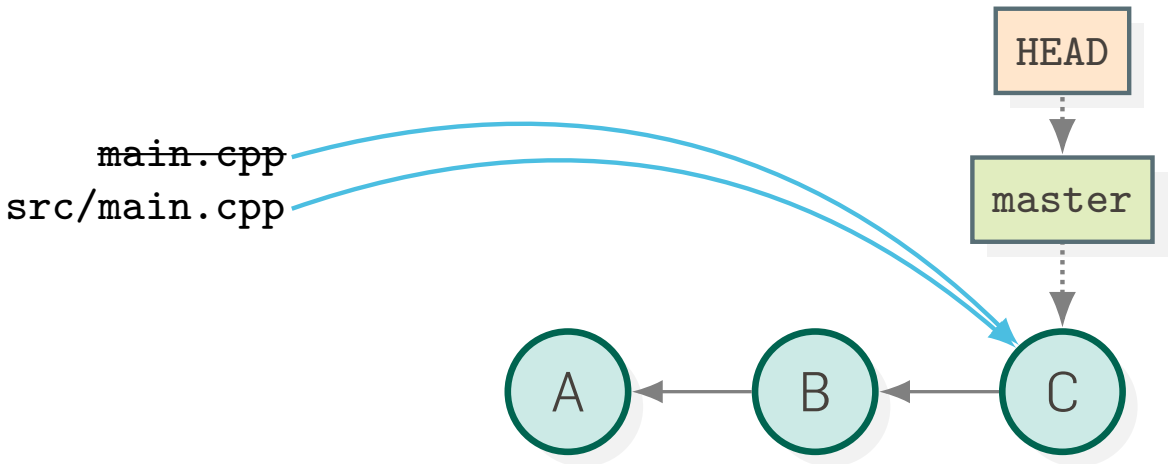
~~main.cpp~~
src/main.cpp
src/position.cpp
src/position.hpp
share/model.csv




```
git add src/position.*
git commit -m"Fix computation of position (float->double)"
```



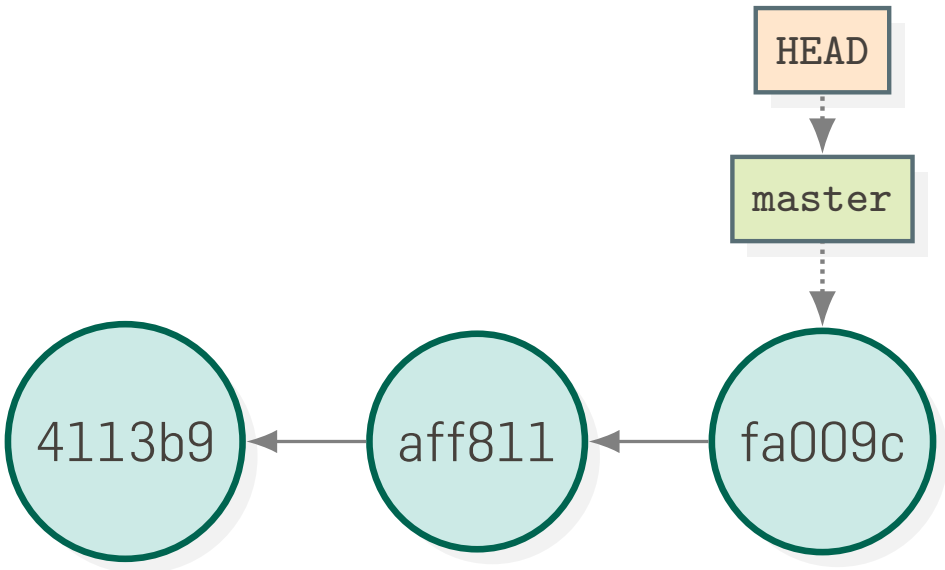

```
git add share/model.csv
git commit -m"Re-trained model with 52 more participants"
```




```
git rm main.cpp
git add src/main.cpp
git commit -m"Move main.cpp to src/"
```

LOG

```
$ git log
commit fa009cd7fca05b0b61170b20cf76a5f72b8843c2
Author: Severin Lemaignan <severin.lemaignan@plymouth.ac.uk>
Date:   Wed Feb 10 16:48:22 2016 +0000

    Move main.cpp to src/

commit aff81119459d9193c09effef1c150c4f7eac08dc
Author: Severin Lemaignan <severin.lemaignan@plymouth.ac.uk>
Date:   Wed Feb 10 16:48:02 2016 +0000

    Re-trained model with 52 more participants

commit 4113b9b6e6bbc8de532ad90153e0059cb5819de7
Author: Severin Lemaignan <severin.lemaignan@plymouth.ac.uk>
Date:   Wed Feb 10 16:47:46 2016 +0000

    Fix computation of position (float->double)
```


HEAD
master
4113b9
aff811
fa009c

THE STAGING AREA

But why do we have to manually tell Git what files to add or remove?

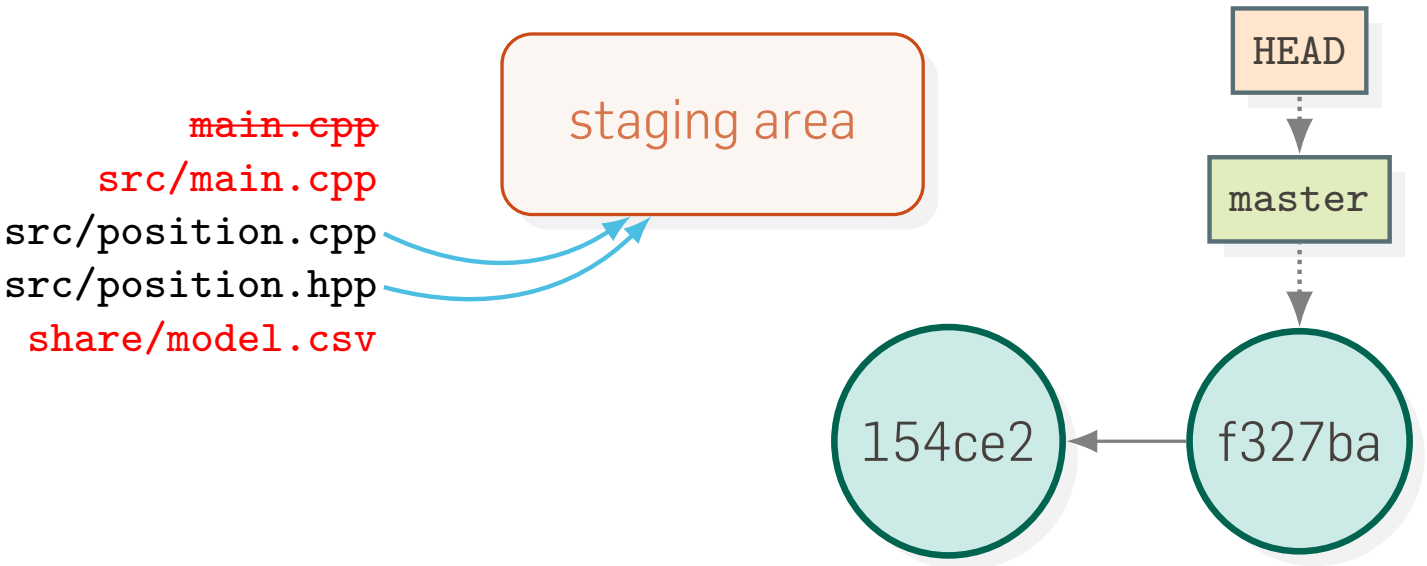
THE STAGING AREA

No “commit all changes” by default (well, you can, actually...)

$\Rightarrow$ Help thinking in terms of atomic commits!

THE STAGING AREA

Preparing a commit consists in filling the **staging area** (or **index**) with the list of changes:



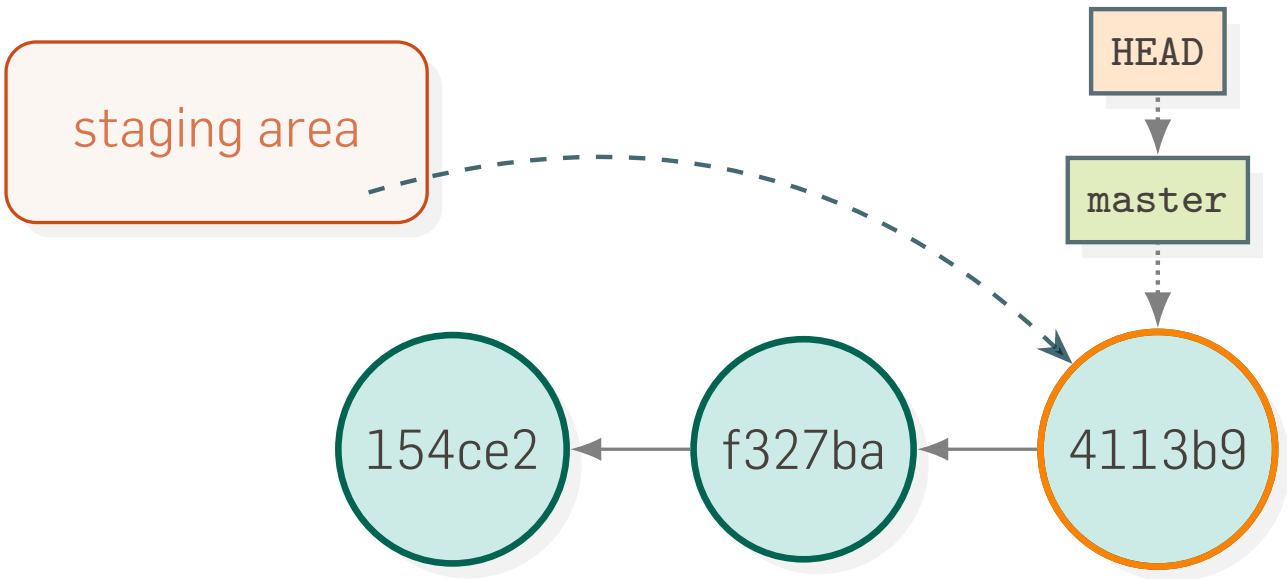

```
git add
git rm
git add -p
...
```

THE STAGING AREA

Preparing a commit consists in filling the **staging area** (or **index**) with the list of changes:

~~main.cpp~~
src/main.cpp

share/model.csv




```
git commit
```

TO SUMMARIZE...

The first time...

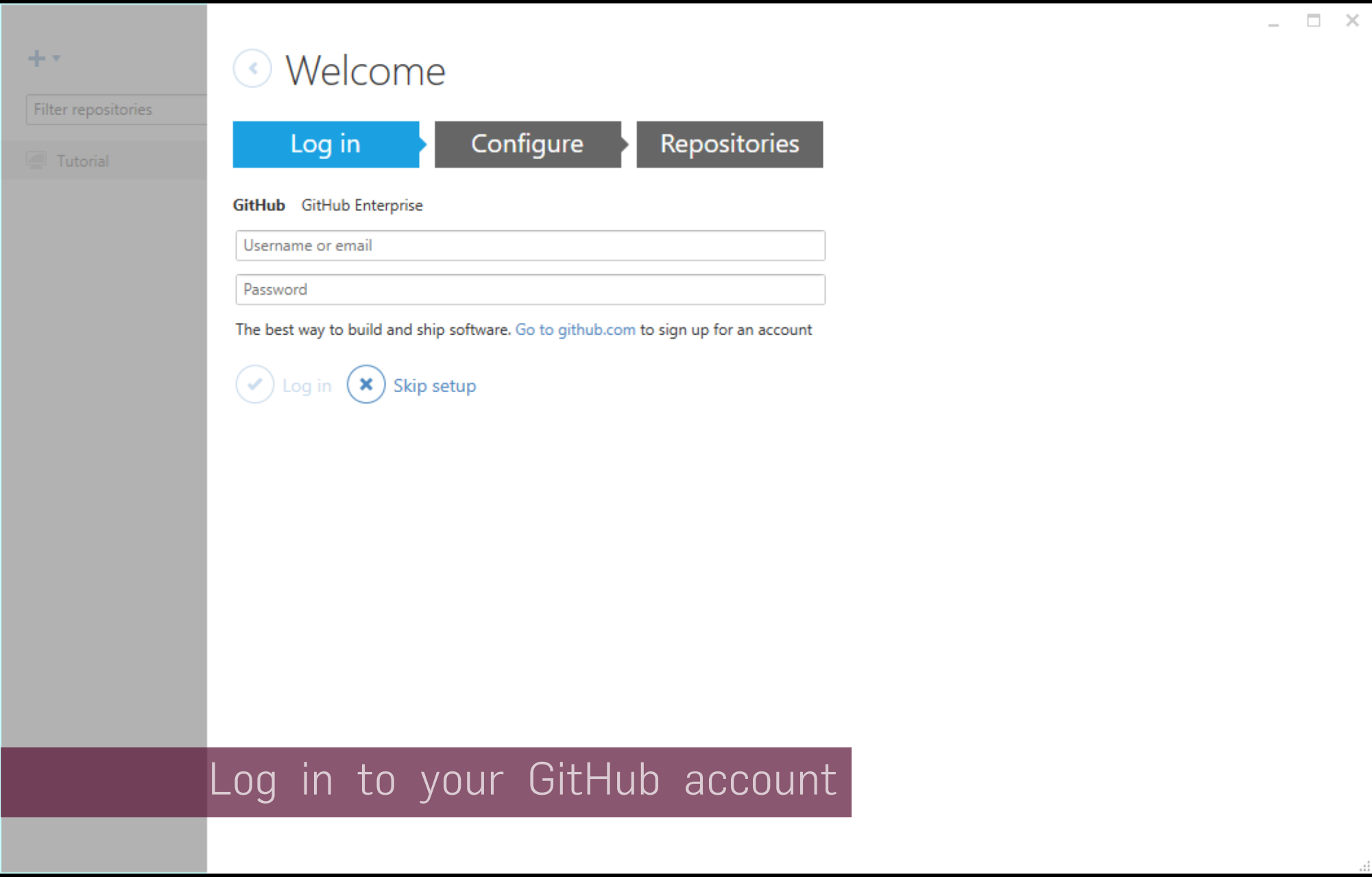
```
$ mkdir my_repo && cd my_repo
$ git init
```

Then...

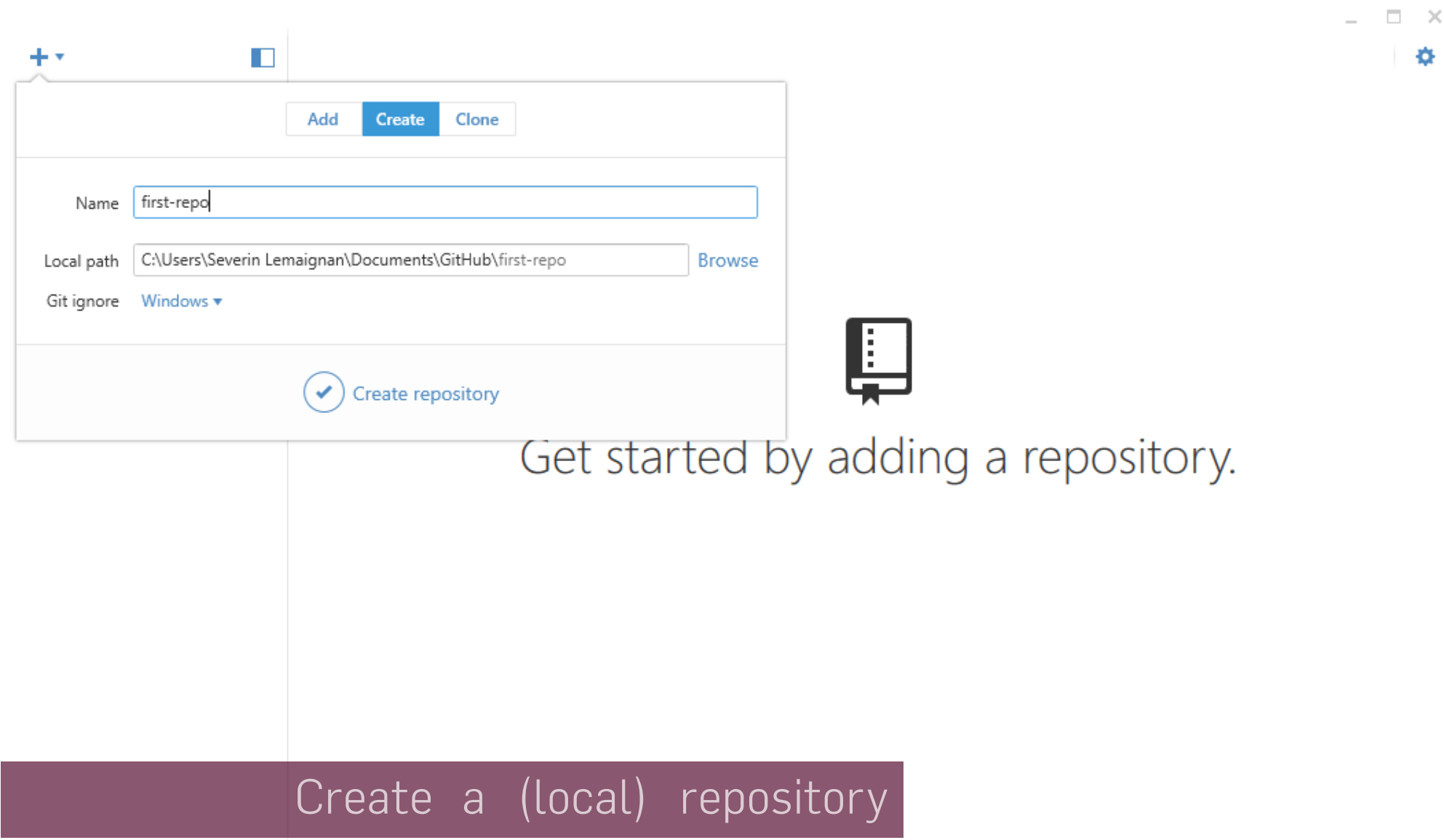
```
# make some changes...
$ git add <files>
$ git commit -m"<commit message>"
# make some changes...
$ git add <files>
$ git commit -m"<other commit message>"
# That's it!
```

Viewed from a GUI (macOS & Windows)
GitHub Desktop Walkthrough

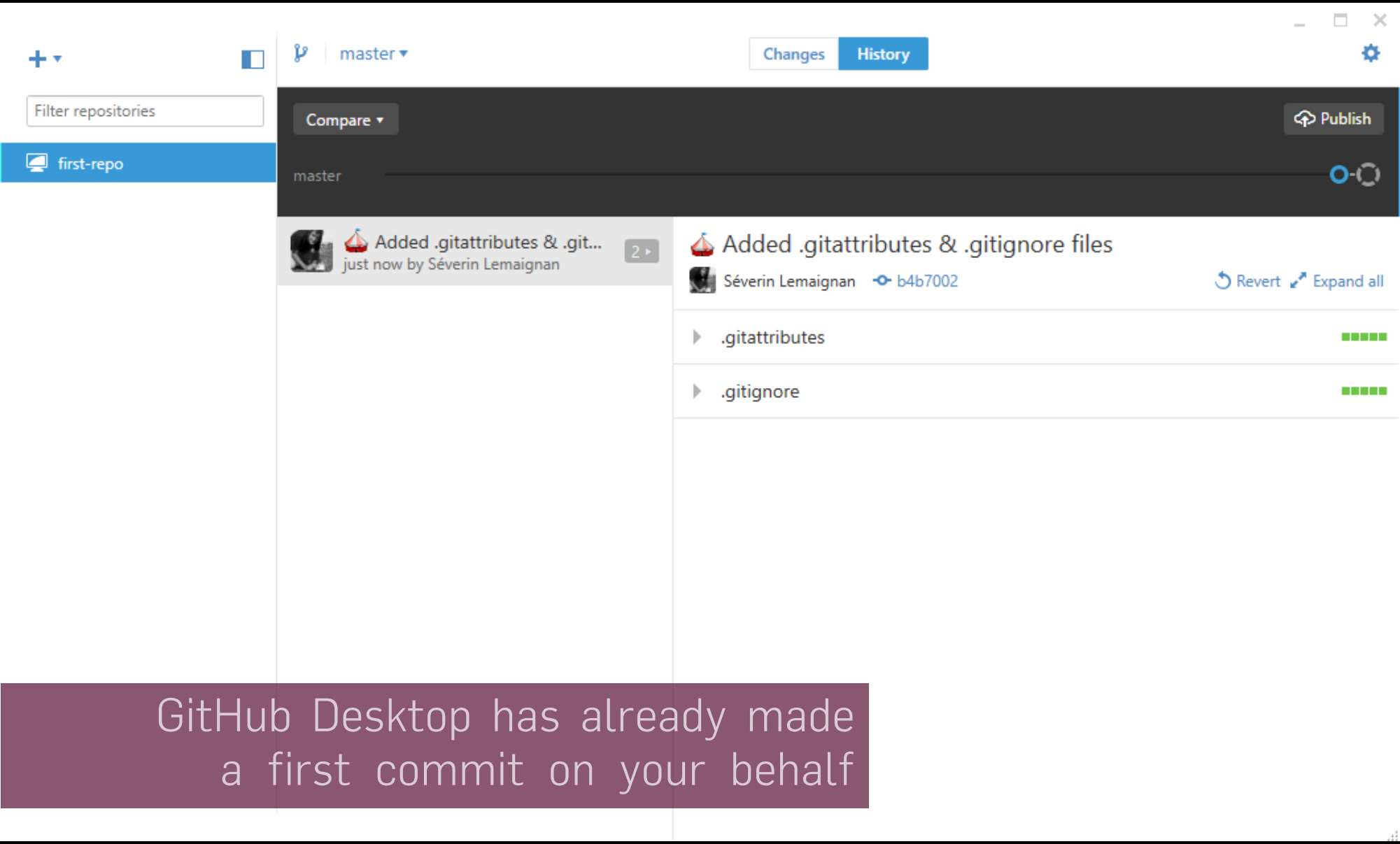
https://desktop.github.com/



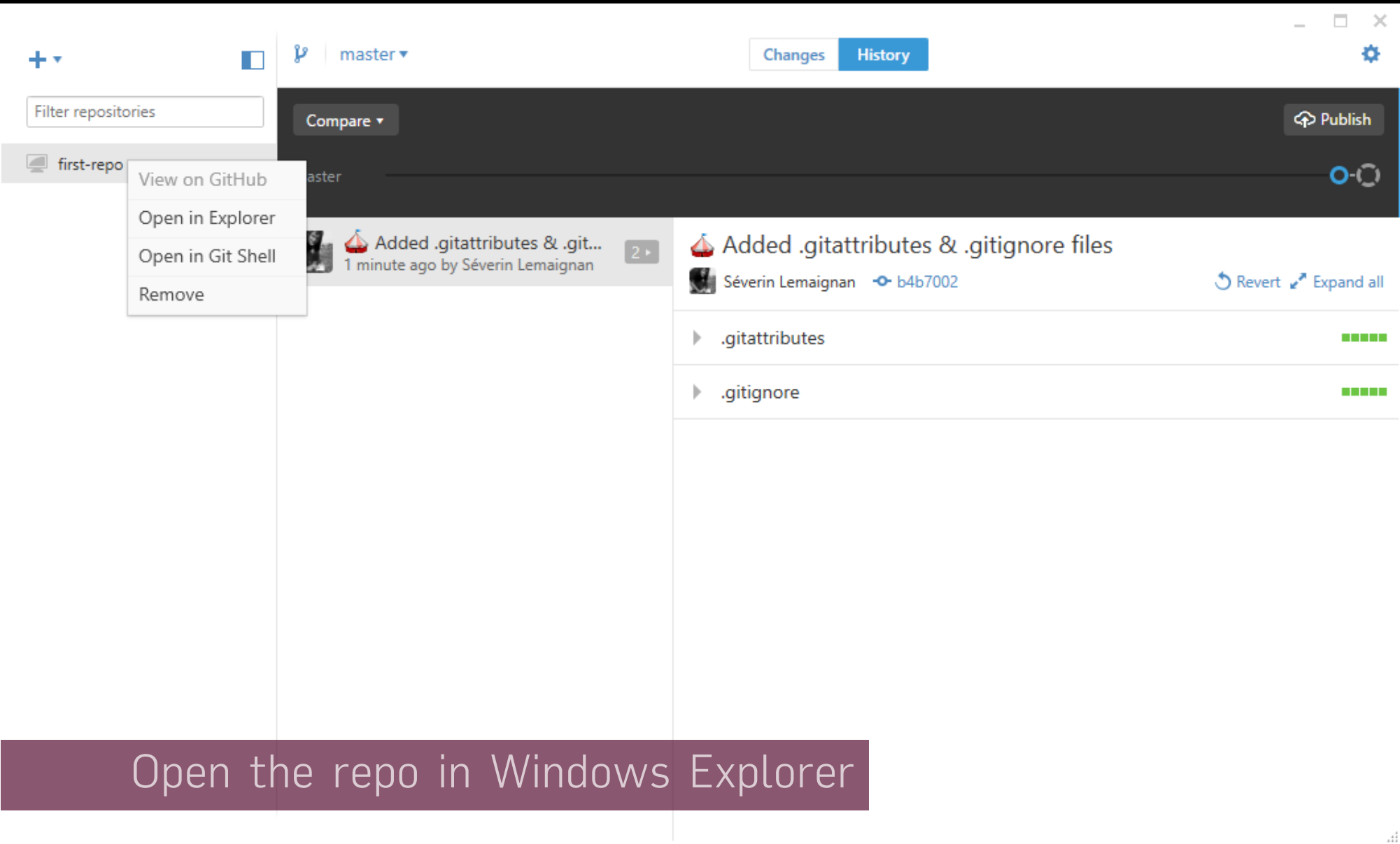
Log in to your GitHub account



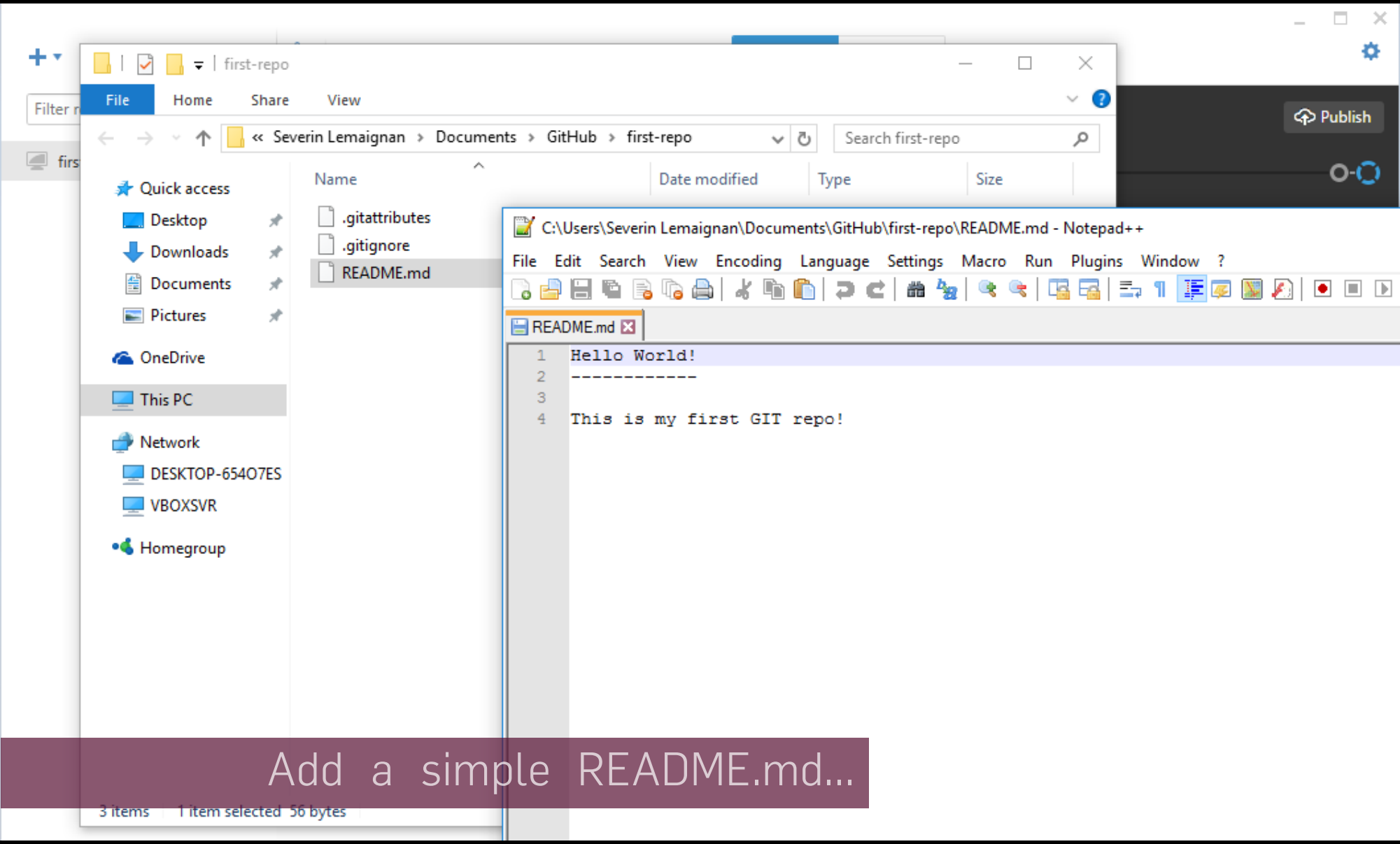
Create a (local) repository

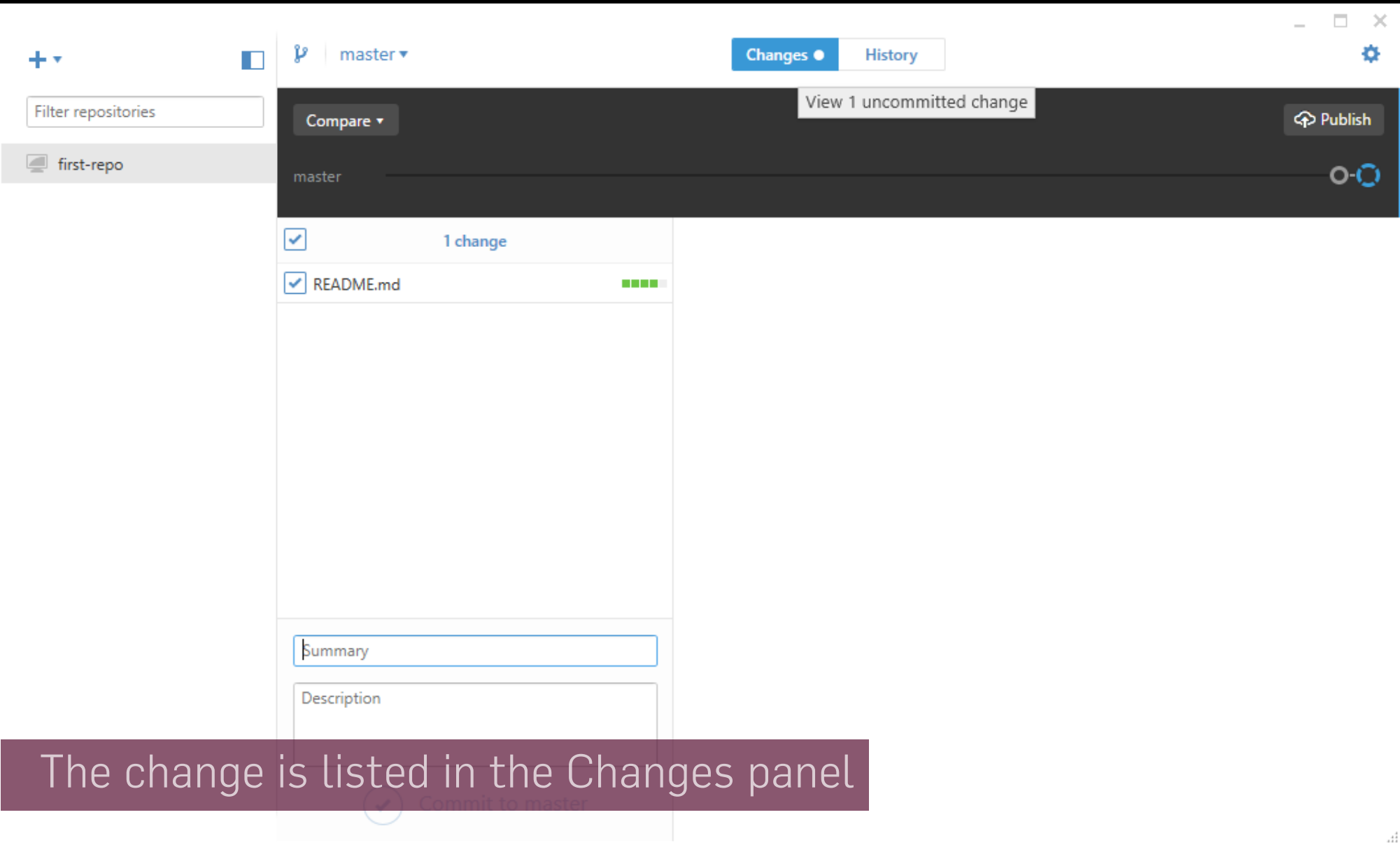


GitHub Desktop has already made a first commit on your behalf

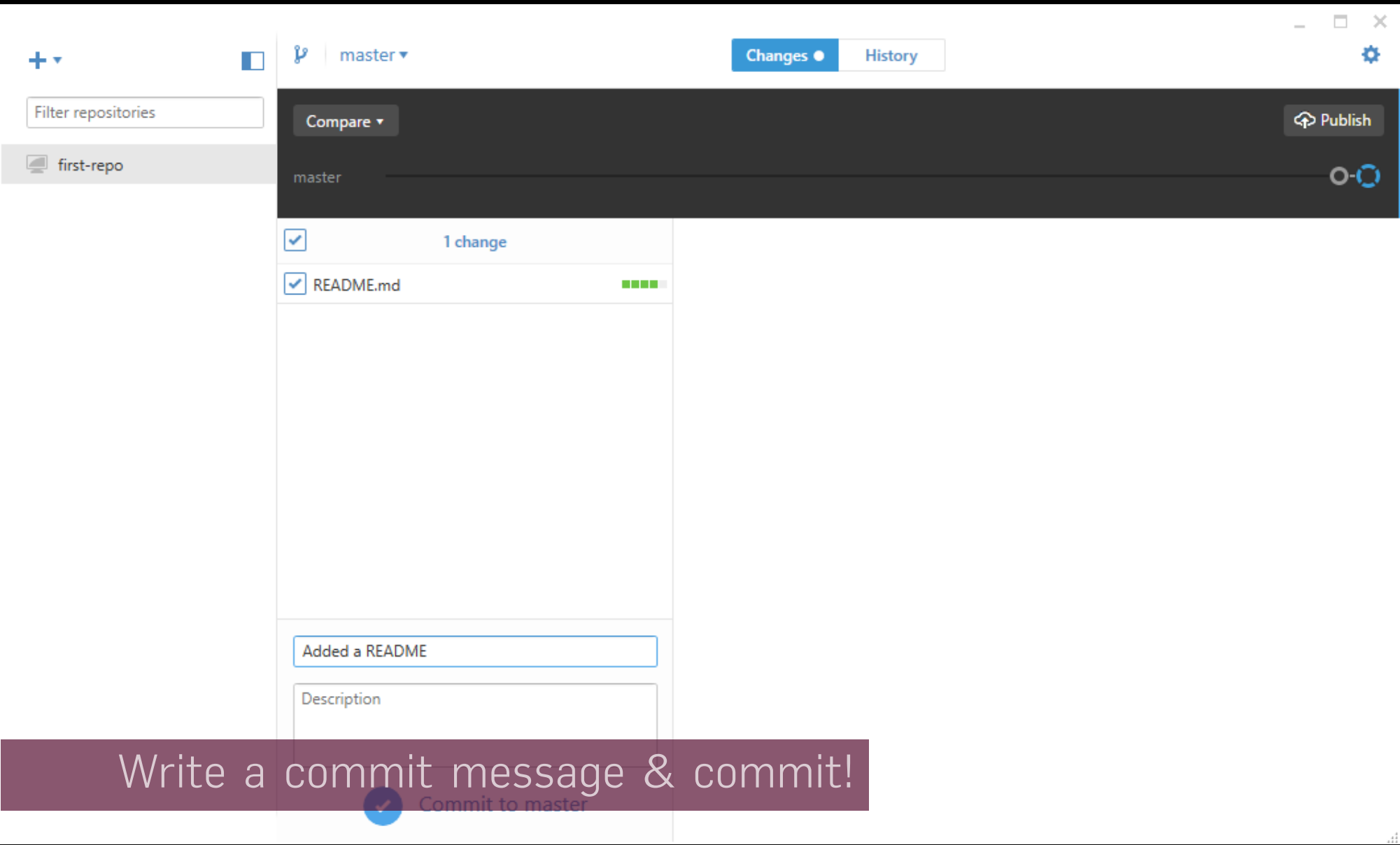


Open the repo in Windows Explorer

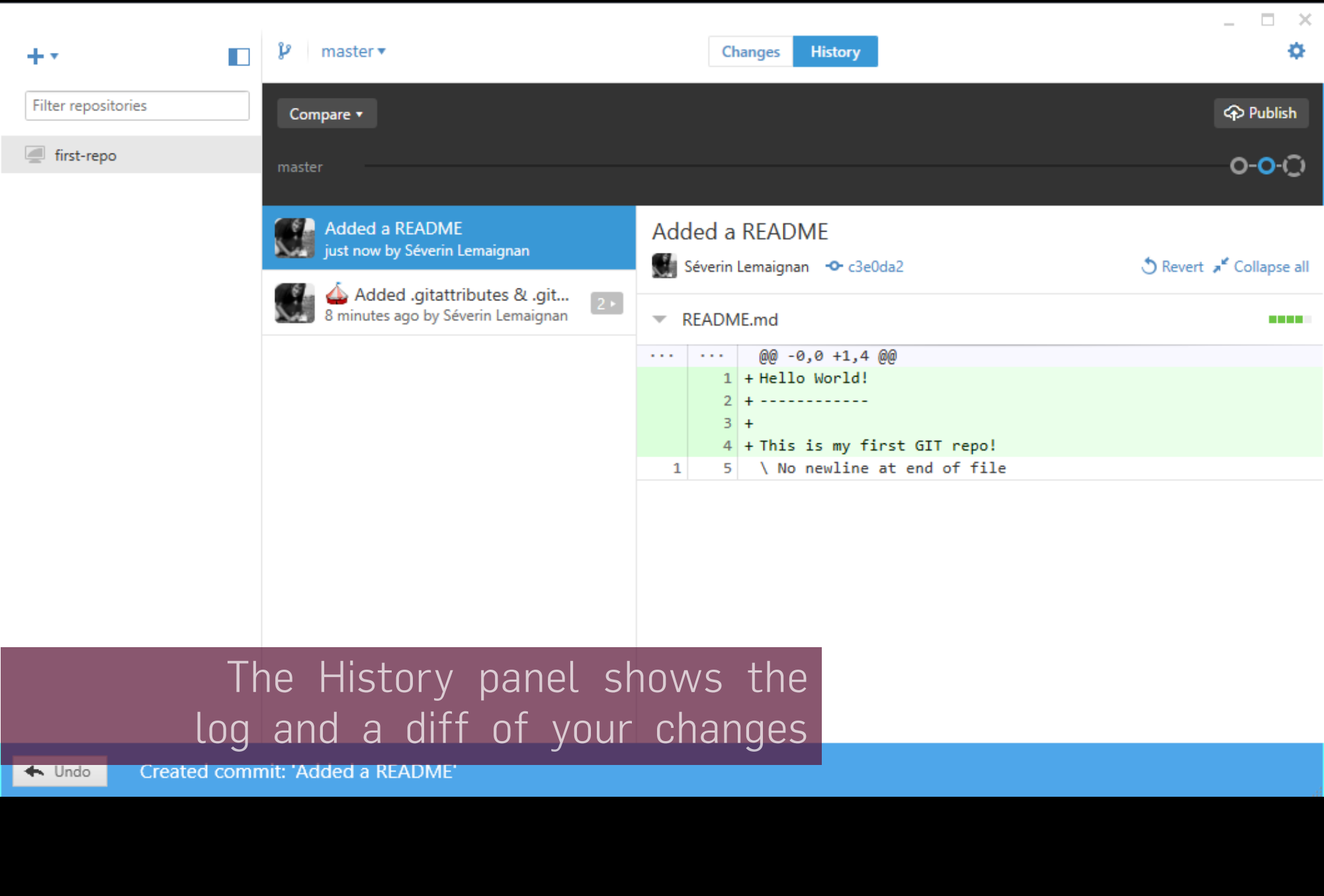

first-repo
File
Home
Share
View
« Severin Lemaignan › Documents › GitHub › first-repo
Search first-repo
Quick access
Desktop
Downloads
Documents
Pictures
OneDrive
This PC
Network
DESKTOP-654O7ES
VBOXSVR
Homegroup
Name
Date modified
Type
Size
.gitattributes
.gitignore
README.md
3 items
1 item selected 56 bytes
C:\Users\Severin Lemaignan\Documents\GitHub\first-repo\README.md - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
README.md
1 Hello World!
2 ------------
3
4 This is my first GIT repo!
Publish
Add a simple README.md...



The change is listed in the Changes panel



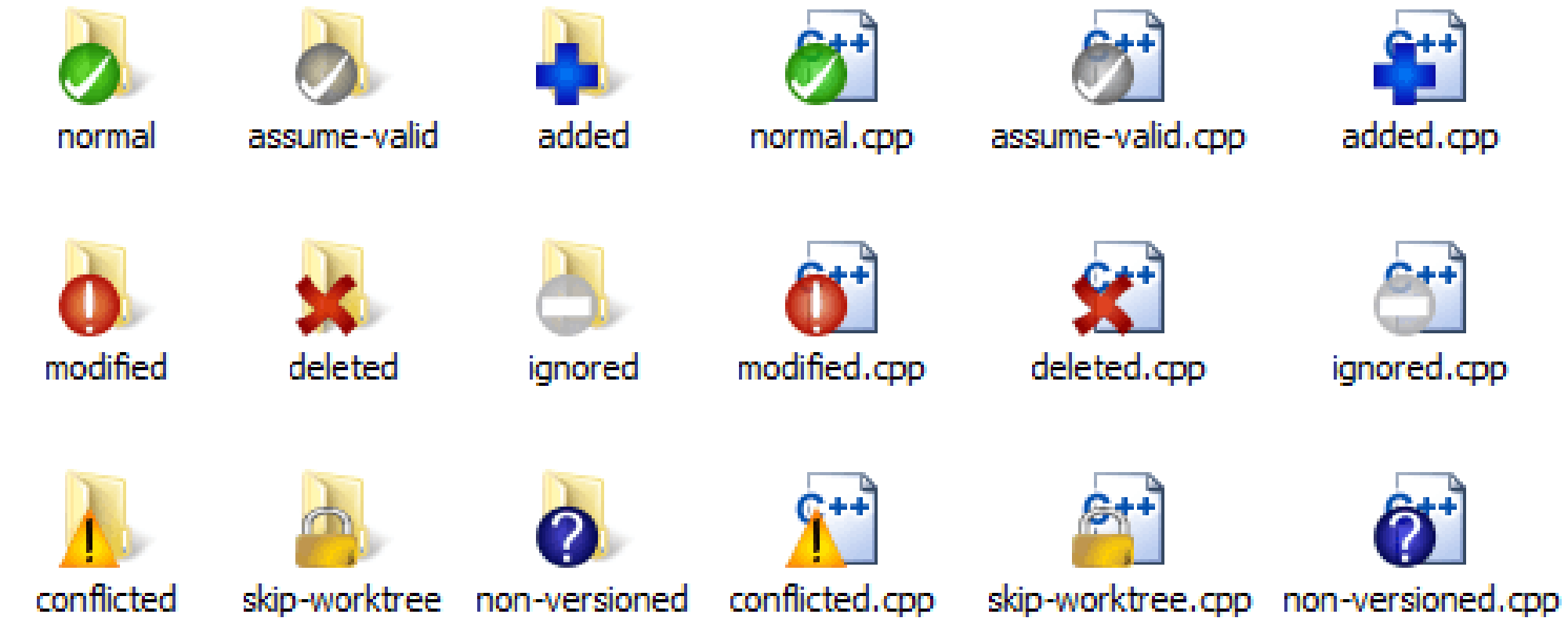
Write a commit message & commit!


Filter repositories
first-repo
master
Changes
History
Compare
Publish
master
Added a README
just now by Séverin Lemaignan
Added .gitattributes & .git...
8 minutes ago by Séverin Lemaignan
2
Added a README
Séverin Lemaignan
c3e0da2
Revert
Collapse all
README.md
@@ -0,0 +1,4 @@
+ Hello World!
+ ------------
+
+ This is my first GIT repo!
\ No newline at end of file
The History panel shows the log and a diff of your changes
Undo
Created commit: 'Added a README'

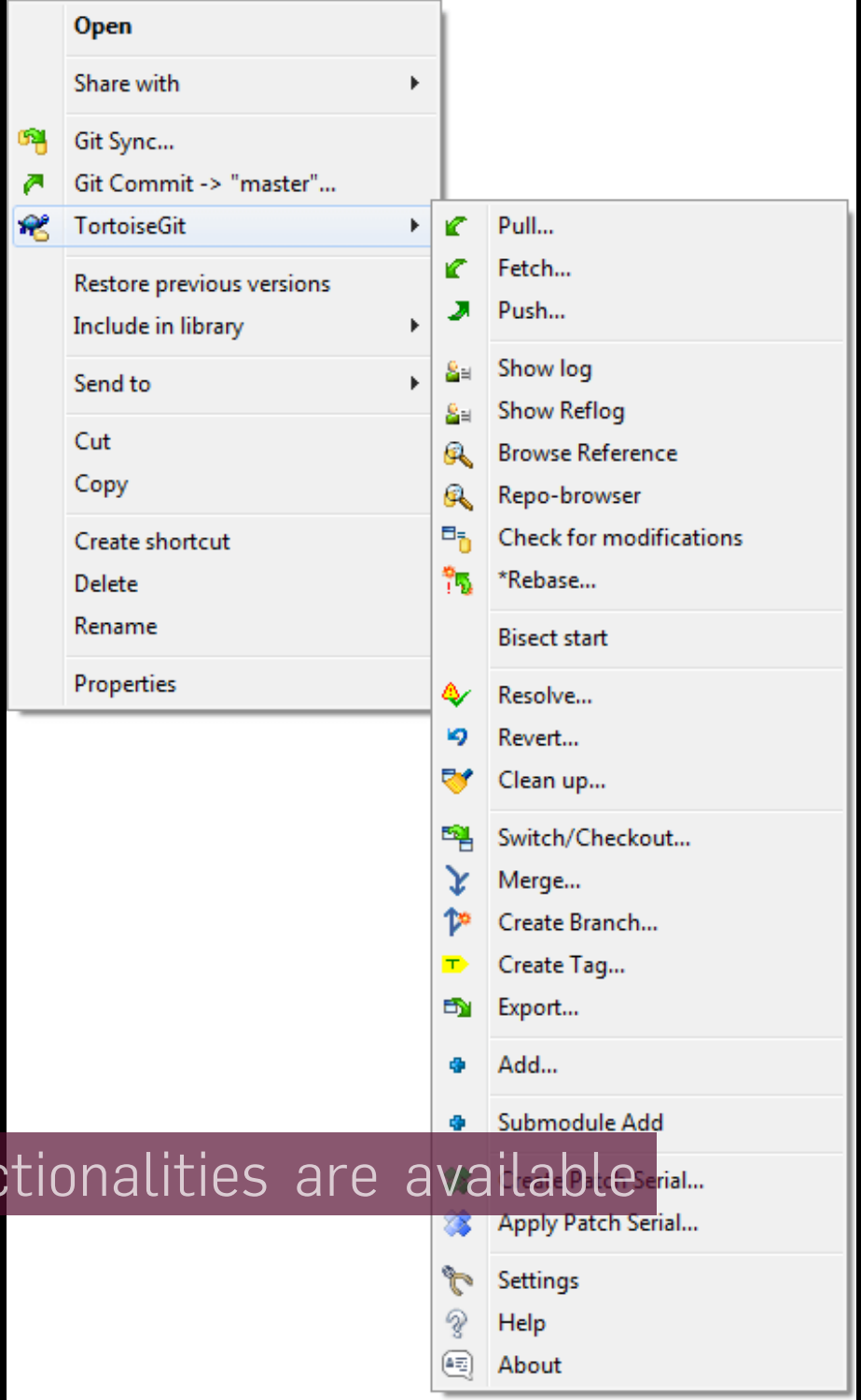
Viewed from a GUI
Tortoise GIT

https://tortoisegit.org/

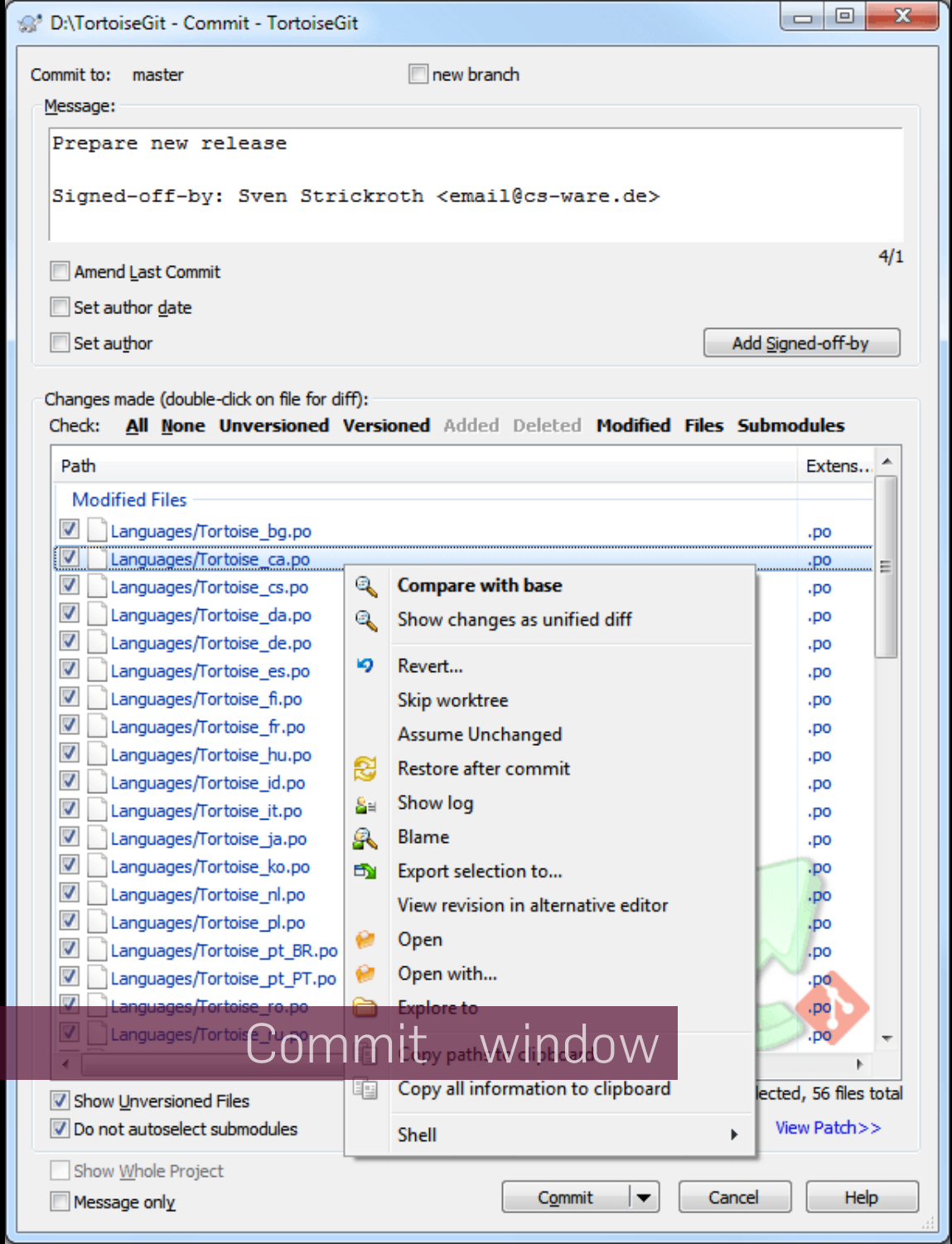
Direct interaction in the Windows explorer



Files' status appear as icons



All the functionalities are available


D:\TortoiseGit - Commit - TortoiseGit
Commit to: master
new branch
Message:
Prepare new release
Signed-off-by: Sven Strickroth <email@cs-ware.de>
4/1
Amend Last Commit
Set author date
Set author
Add Signed-off-by
Changes made (double-click on file for diff):
Check: All None Unversioned Versioned Added Deleted Modified Files Submodules
Path
Extens...
Modified Files
Languages/Tortoise_bg.po
Languages/Tortoise_ca.po
Languages/Tortoise_cs.po
Languages/Tortoise_da.po
Languages/Tortoise_de.po
Languages/Tortoise_es.po
Languages/Tortoise_fi.po
Languages/Tortoise_fr.po
Languages/Tortoise_hu.po
Languages/Tortoise_id.po
Languages/Tortoise_it.po
Languages/Tortoise_ja.po
Languages/Tortoise_ko.po
Languages/Tortoise_nl.po
Languages/Tortoise_pl.po
Languages/Tortoise_pt_BR.po
Languages/Tortoise_pt_PT.po
Languages/Tortoise_ro.po
Languages/Tortoise_ru.po
.po
Compare with base
Show changes as unified diff
Revert...
Skip worktree
Assume Unchanged
Restore after commit
Show log
Blame
Export selection to...
View revision in alternative editor
Open
Open with...
Explore to
Copy path to clipboard
Copy all information to clipboard
Shell
Commit window
Show Unversioned Files
Do not autoselect submodules
lected, 56 files total
View Patch>>
Show Whole Project
Message only
Commit
Cancel
Help

WHAT SHOULD BE TRACKED?

Short answer: **everything you care about in your project**

WHAT SHOULD BE TRACKED?

Short answer: **everything you care about in your project**

(you can left out temporary files, automatically generated files, etc)

WHAT SHOULD BE TRACKED?

Short answer: **everything you care about in your project**

(you can left out temporary files, automatically generated files, etc)

However, versioning is **less useful for binary files**:

- no line-by-line tracking of changes
- every single change creates a whole copy: repo size might grow quickly!

Binary files include images, archives (zip files), **PDF**, **most office document (docx/xlsx/pptx)**

WHAT SHOULD BE TRACKED?

Short answer: **everything you care about in your project**

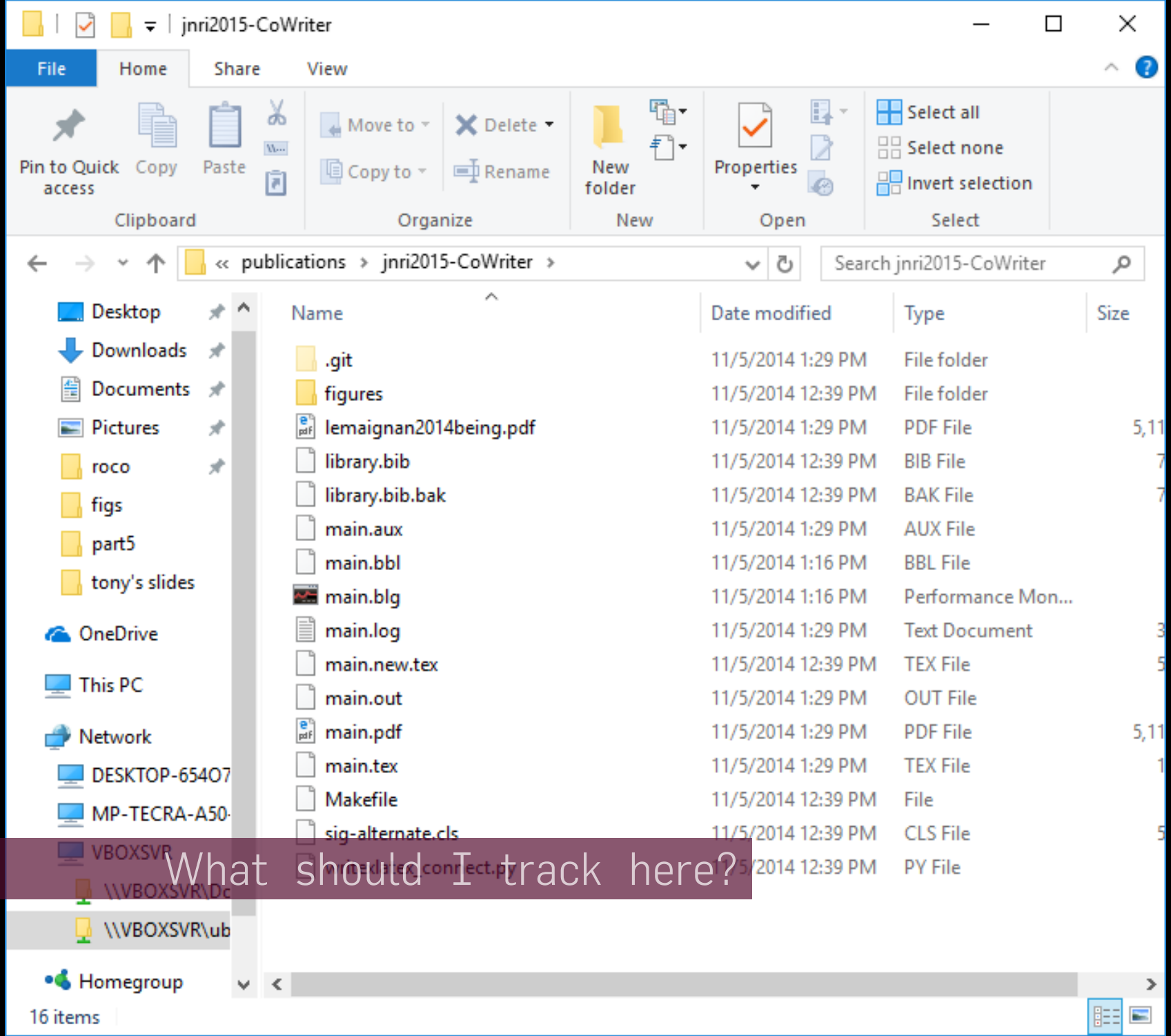
(you can left out temporary files, automatically generated files, etc)

However, versioning is **less useful for binary files**:

- no line-by-line tracking of changes
- every single change creates a whole copy: repo size might grow quickly!

Binary files include images, archives (zip files), **PDF**, **most office document (docx/xlsx/pptx)**

For documents, you might want to consider alternative like markdown.

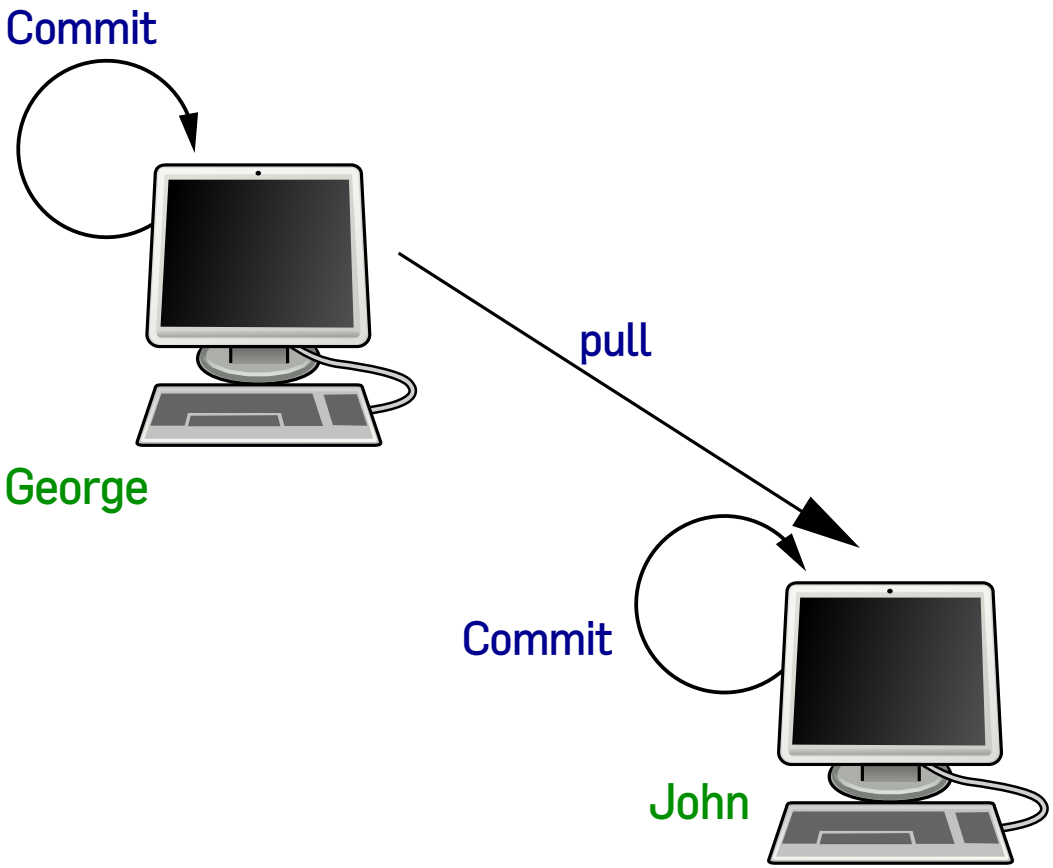

jnri2015-CoWriter
File
Home
Share
View
Pin to Quick access
Copy
Paste
Clipboard
Move to
Copy to
Delete
Rename
Organize
New folder
New
Properties
Open
Select all
Select none
Invert selection
Select
« publications › jnri2015-CoWriter ›
Search jnri2015-CoWriter
Desktop
Downloads
Documents
Pictures
roco
figs
part5
tony's slides
OneDrive
This PC
Network
DESKTOP-654O7
MP-TECRA-A50
VBOXSVR
\\VBOXSVR\Dc
\\VBOXSVR\ub
Homegroup
Name
Date modified
Type
Size
.git 11/5/2014 1:29 PM File folder
figures 11/5/2014 12:39 PM File folder
lemaignan2014being.pdf 11/5/2014 1:29 PM PDF File 5,11
library.bib 11/5/2014 12:39 PM BIB File
library.bib.bak 11/5/2014 12:39 PM BAK File
main.aux 11/5/2014 1:29 PM AUX File
main.bbl 11/5/2014 1:16 PM BBL File
main.blg 11/5/2014 1:16 PM Performance Mon...
main.log 11/5/2014 1:29 PM Text Document
main.new.tex 11/5/2014 12:39 PM TEX File
main.out 11/5/2014 1:29 PM OUT File
main.pdf 11/5/2014 1:29 PM PDF File 5,11
main.tex 11/5/2014 1:29 PM TEX File
Makefile 11/5/2014 12:39 PM File
sig-alternate.cls 11/5/2014 12:39 PM CLS File
PY File
What should I track here?
16 items

COLLABORATING

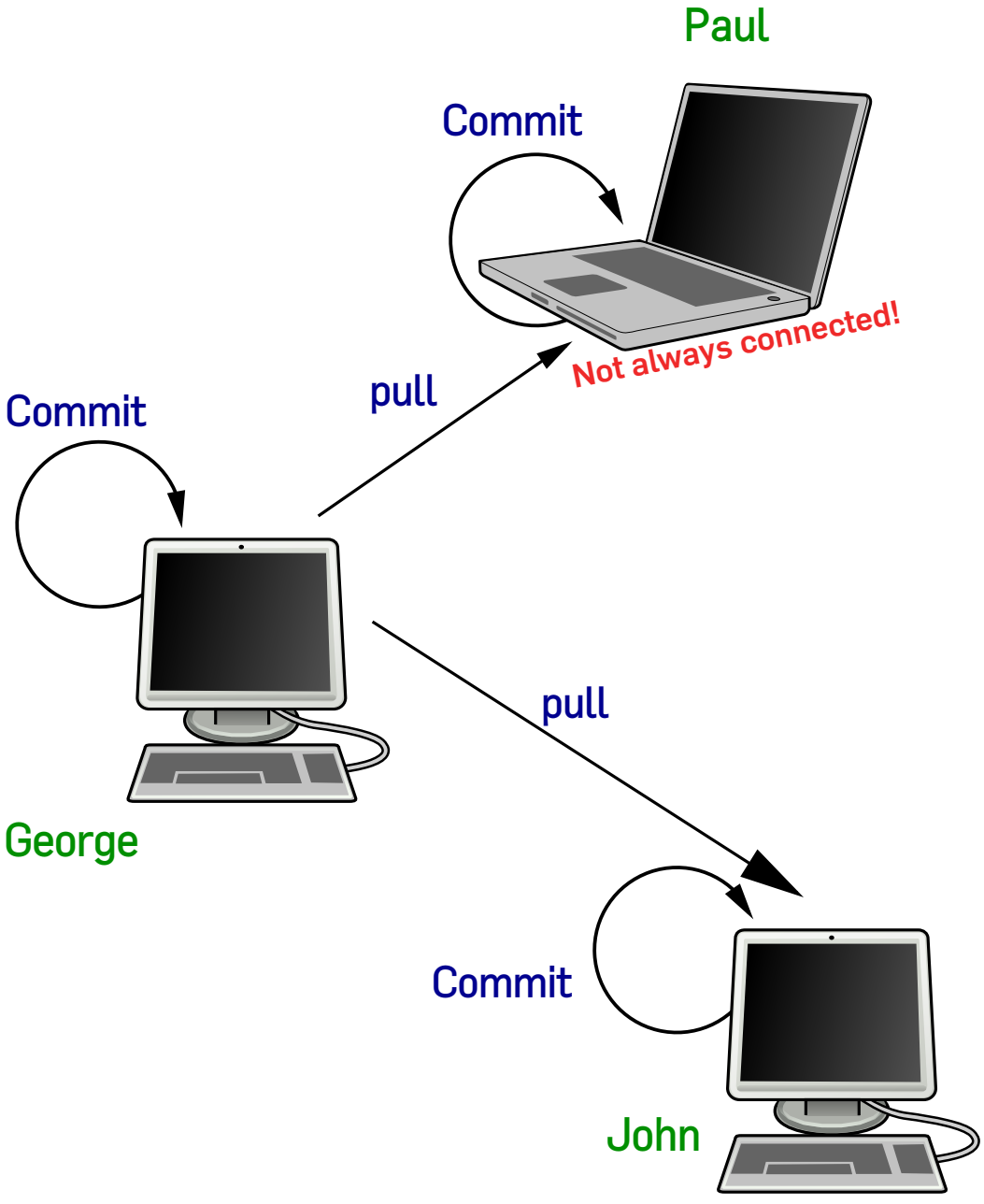
Commit

George

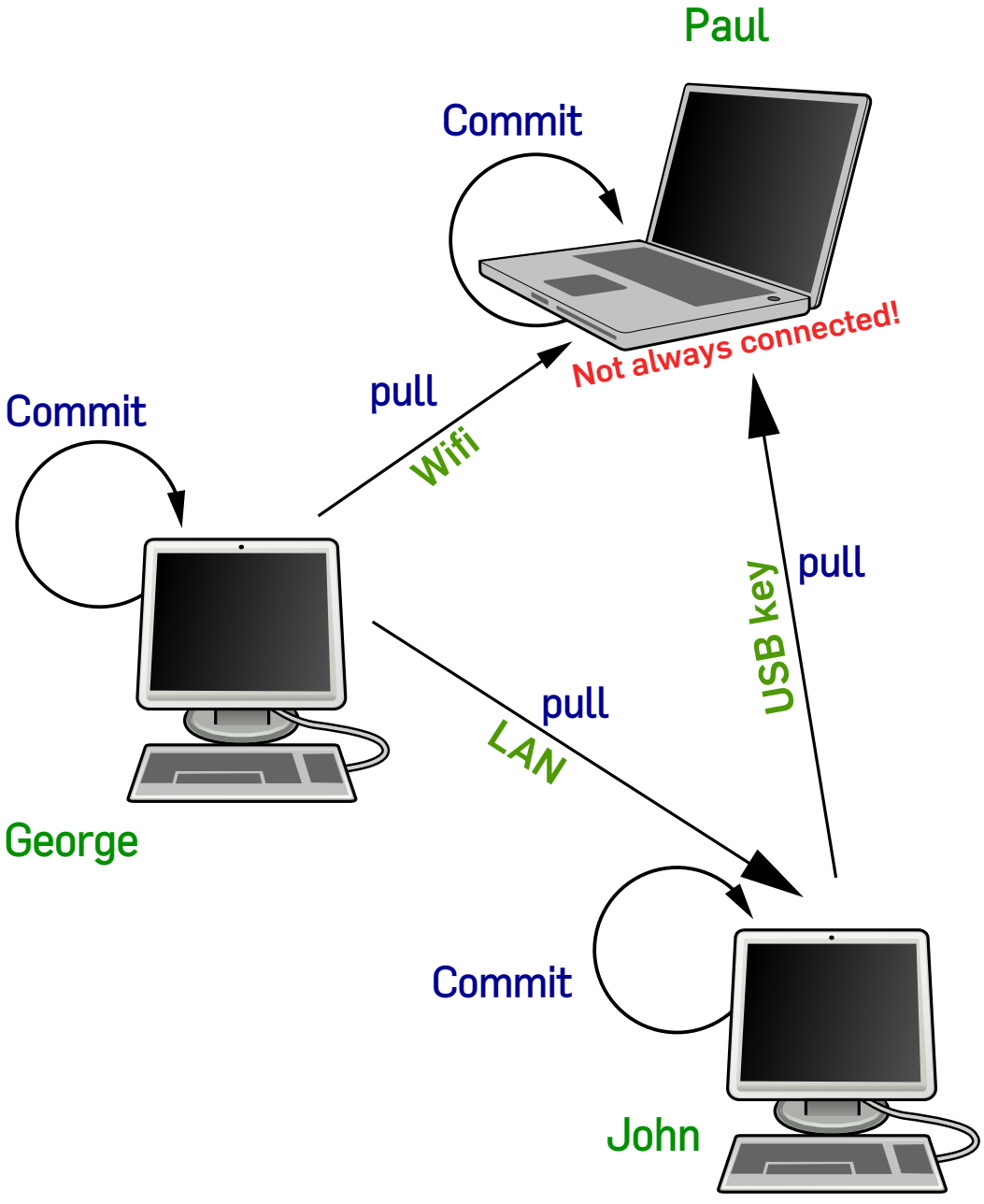
Based on a figure by M. Herrb, CC-BY-SA 3.0



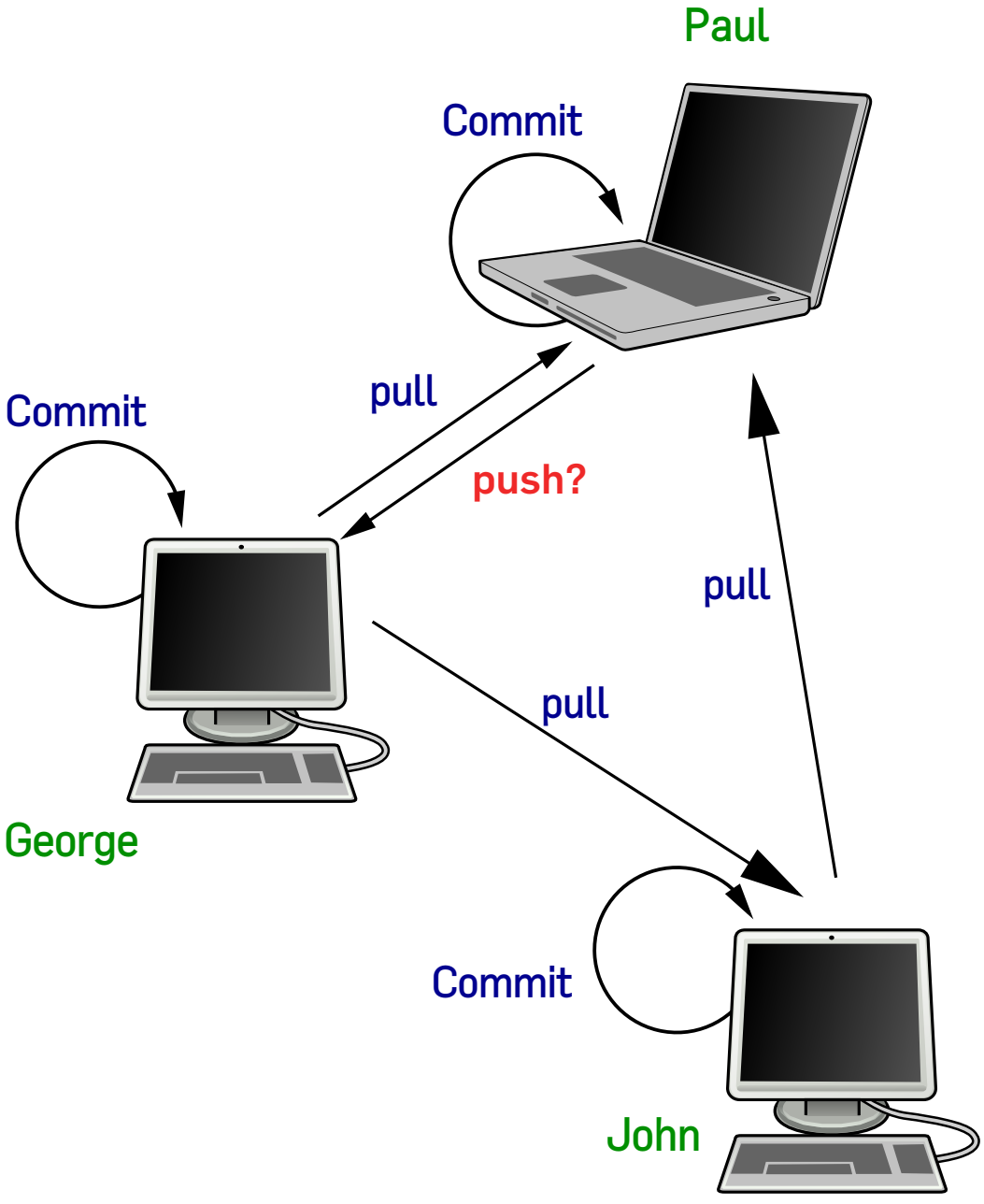
Based on a figure by M. Herrb, CC-BY-SA 3.0



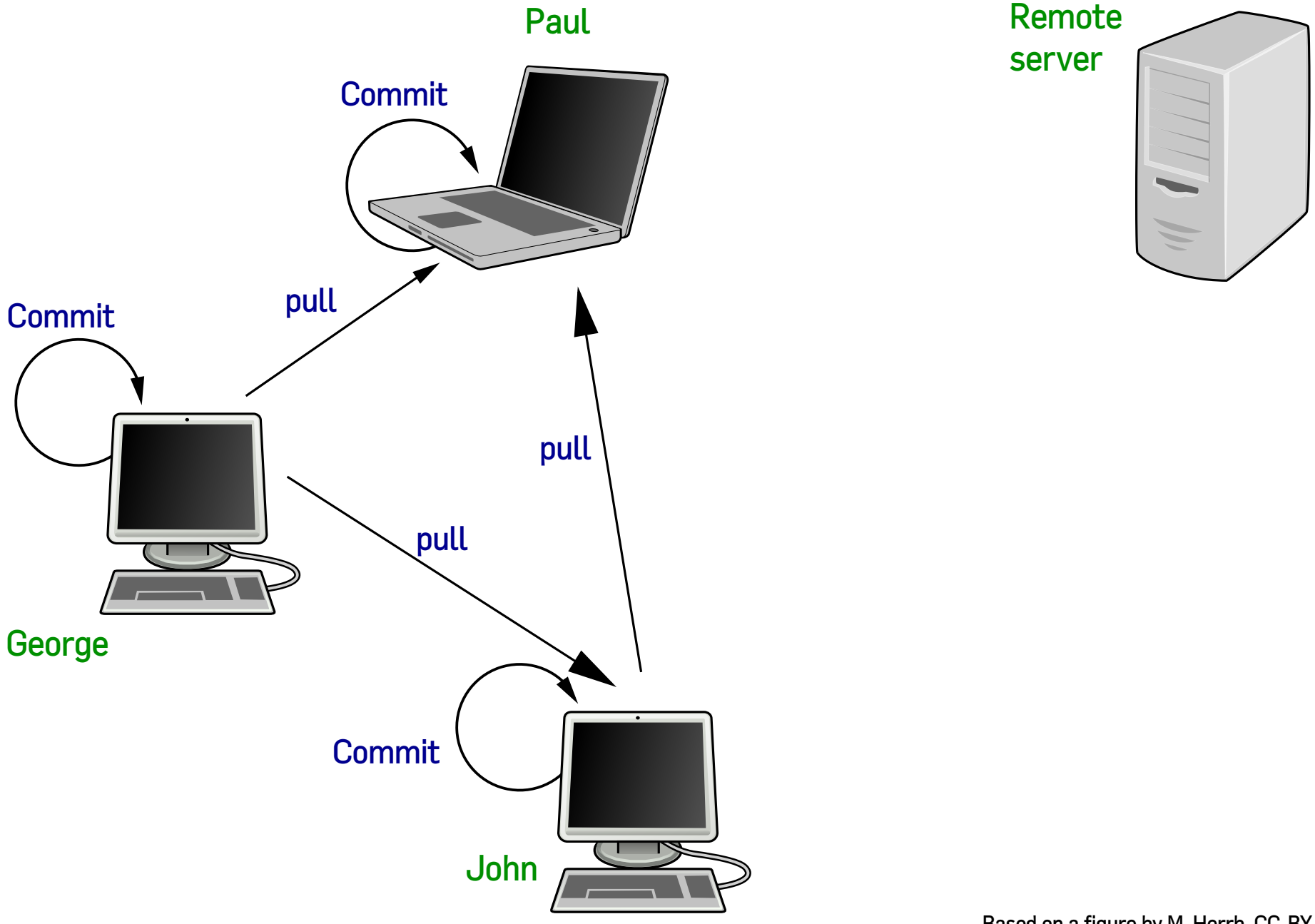
Based on a figure by M. Herrb, CC-BY-SA 3.0



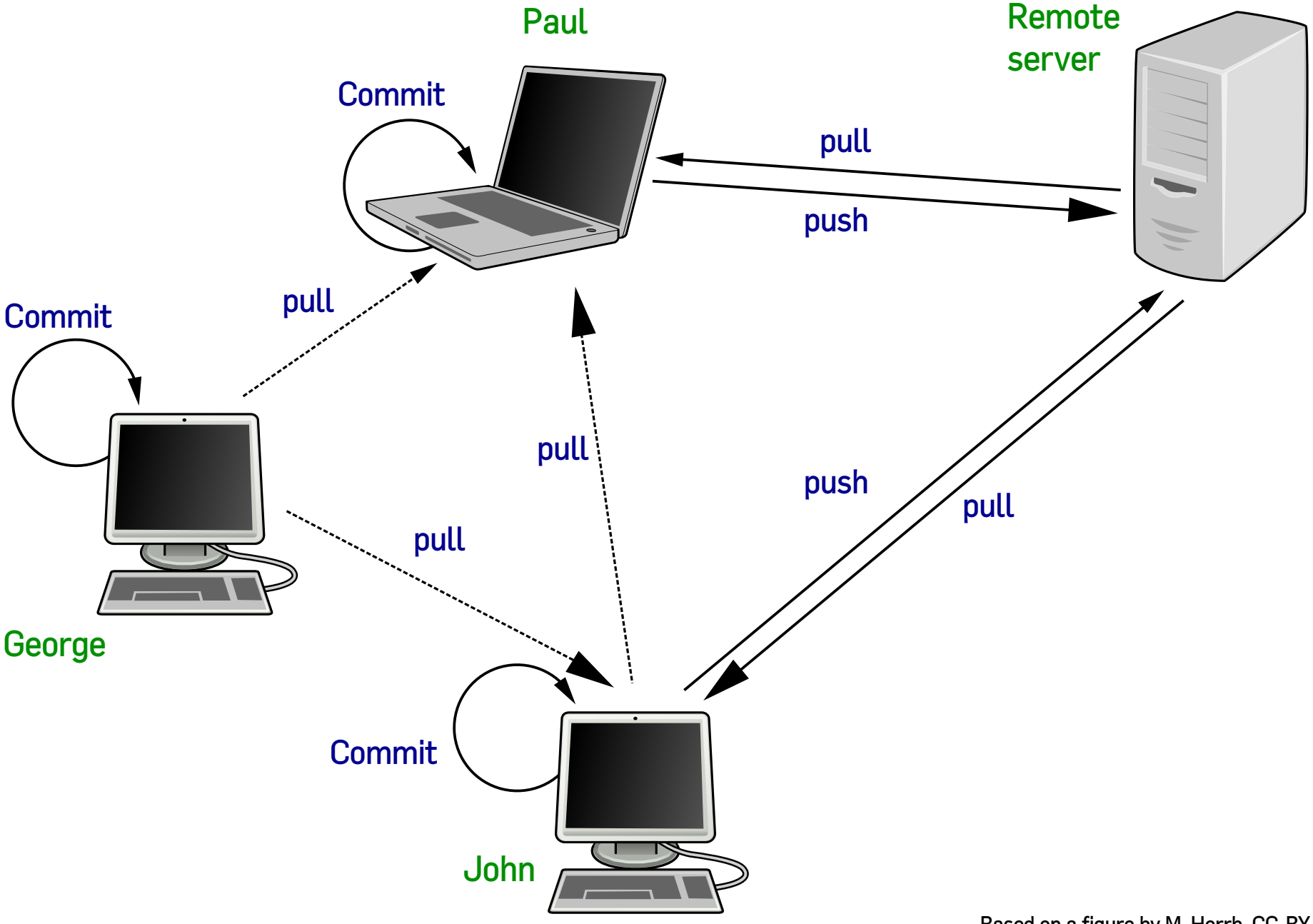
Based on a figure by M. Herrb, CC-BY-SA 3.0



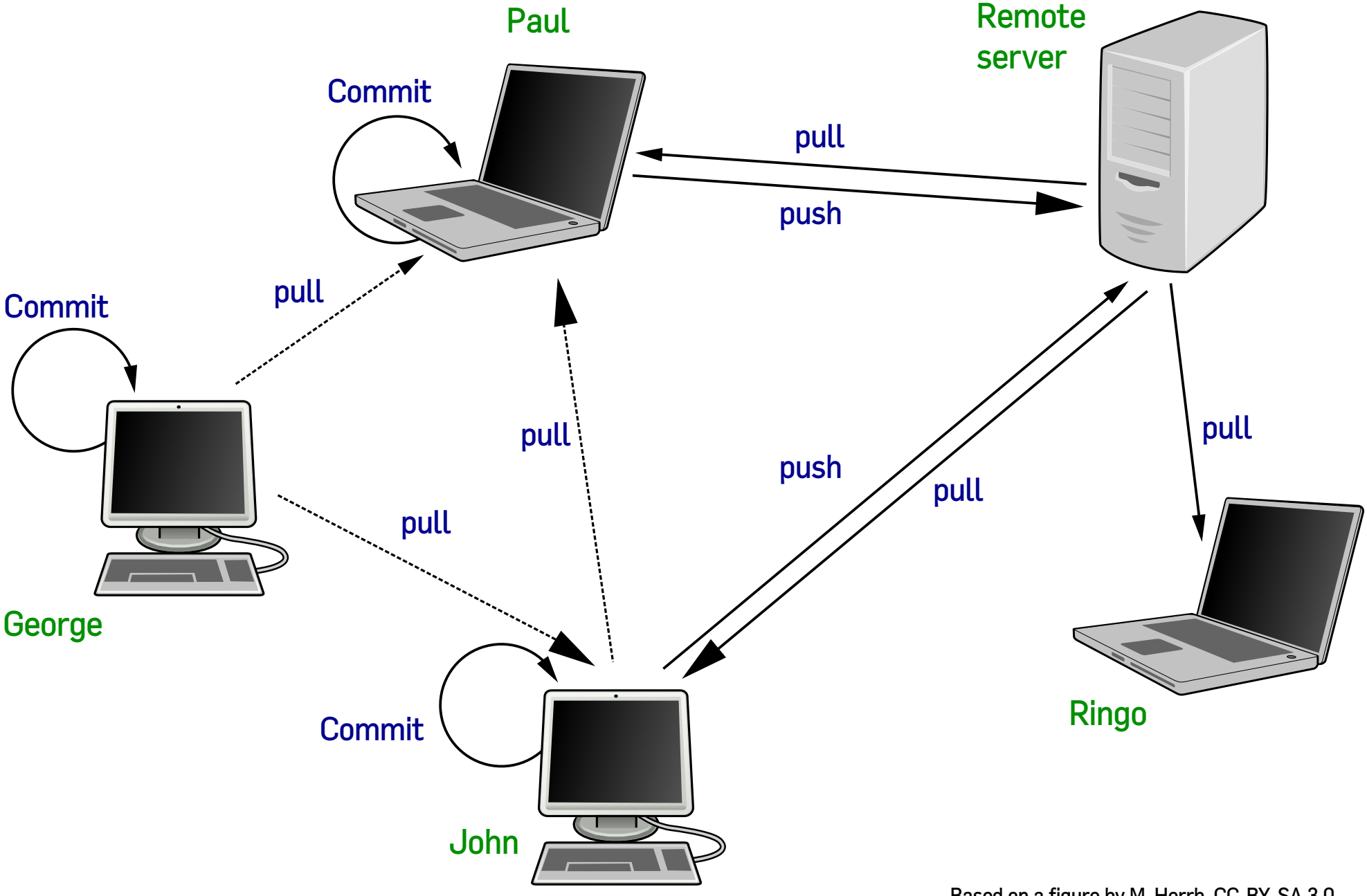
Based on a figure by M. Herrb, CC-BY-SA 3.0



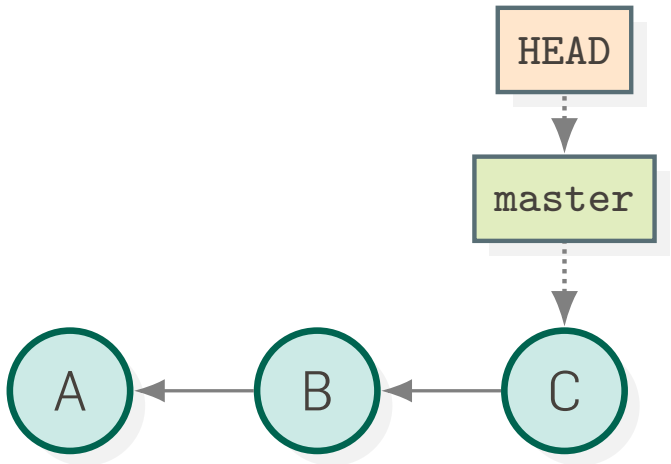
Based on a figure by M. Herrb, CC-BY-SA 3.0

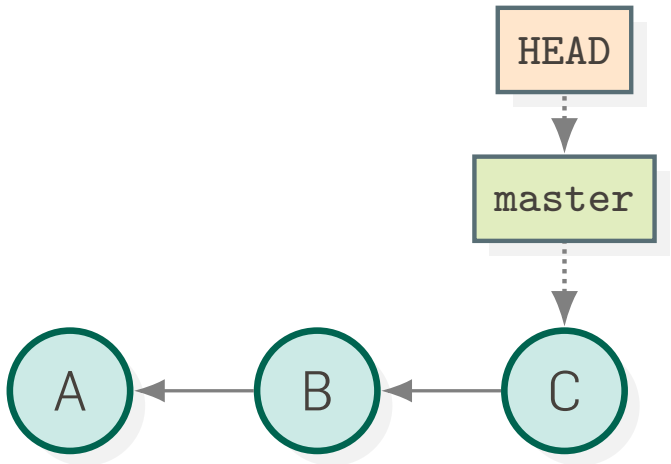


Based on a figure by M. Herrb, CC-BY-SA 3.0

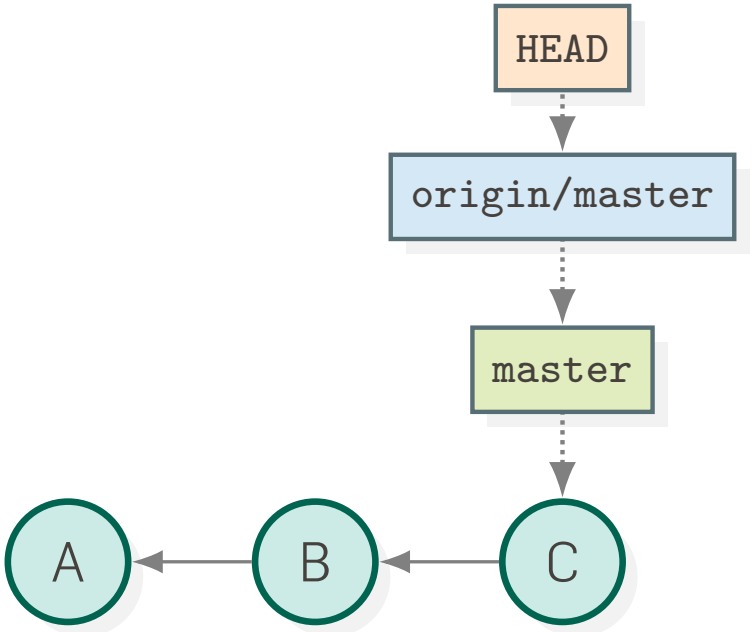


Based on a figure by M. Herrb, CC-BY-SA 3.0


HEAD
master
A
B
C

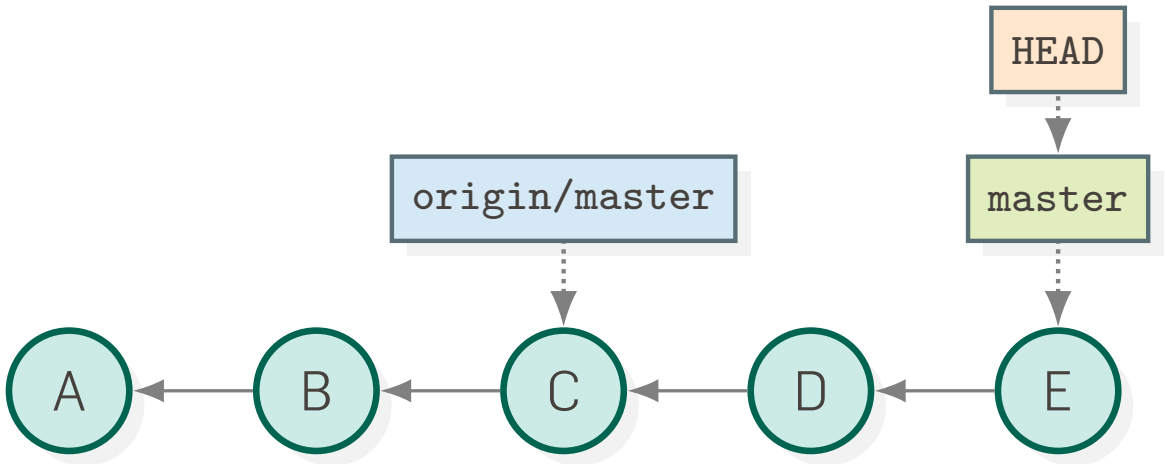


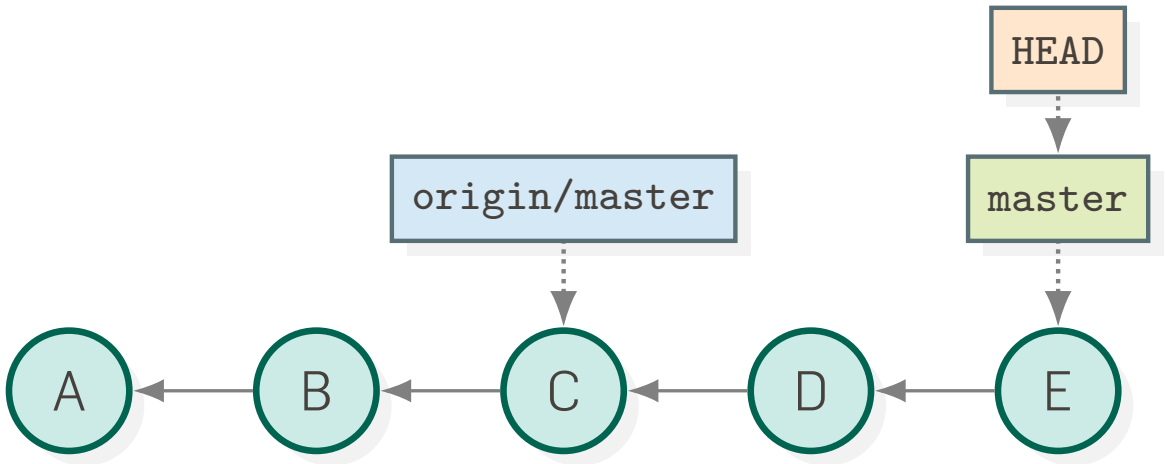

```
git remote add origin git@github.com:user/repo.git
git remote add john-usb E:\john_repo
git remote add ftp-origin ftp://host.xz/path/to/repo.git/
...
```



`git push origin master`

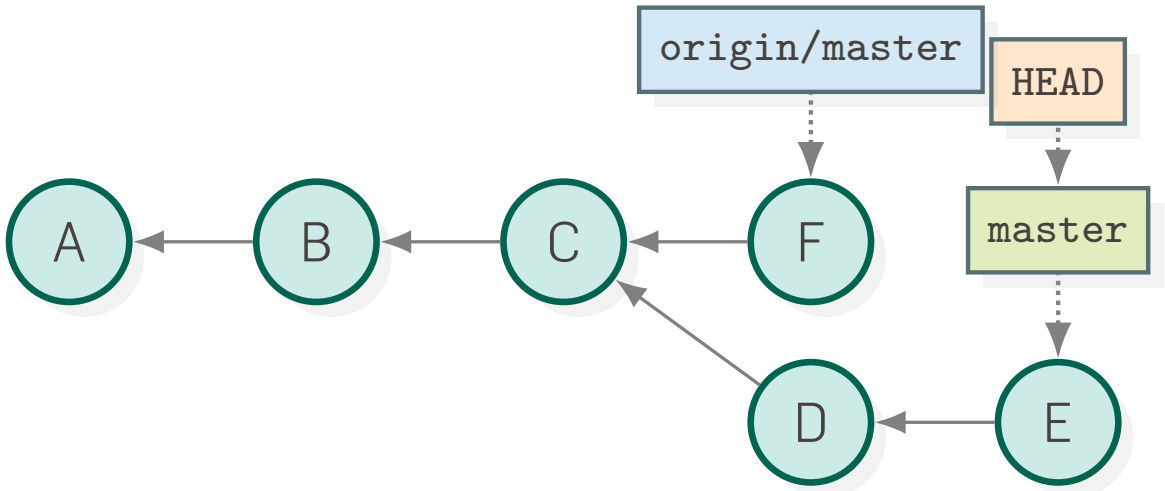
(or simply `git push`)

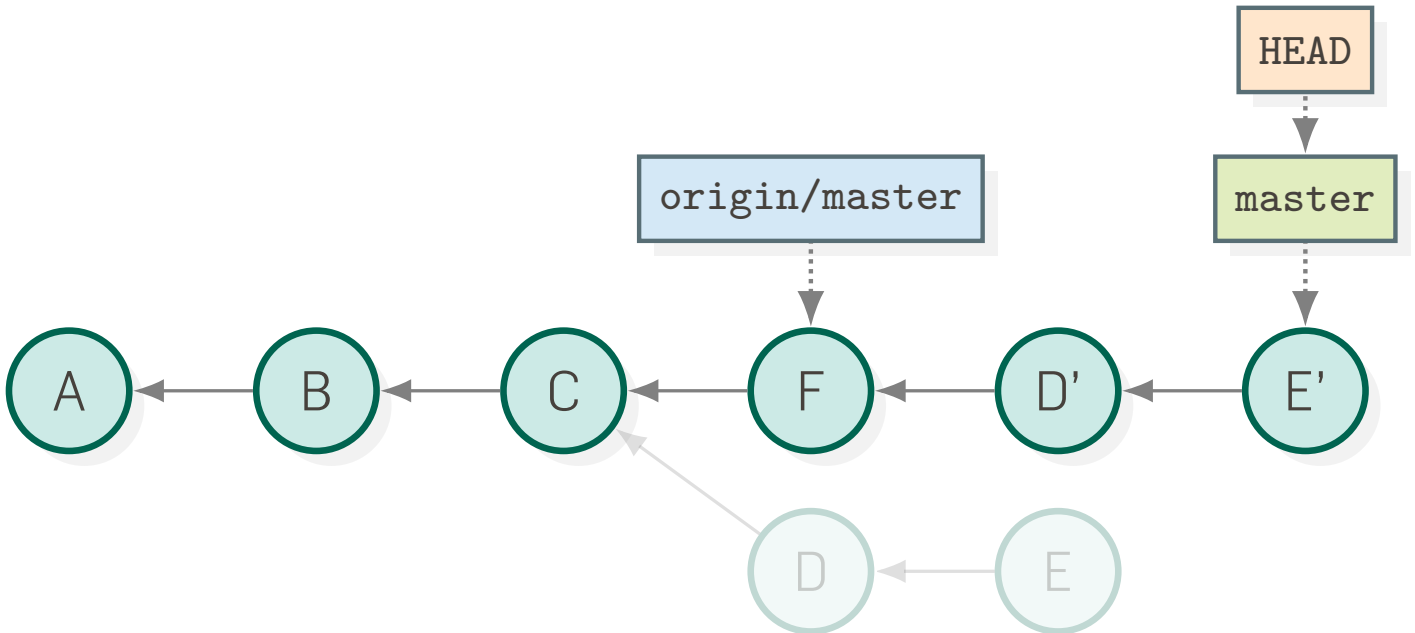

HEAD
origin/master
master
A
B
C
D
E



What happened on our remote? Let's have a look...

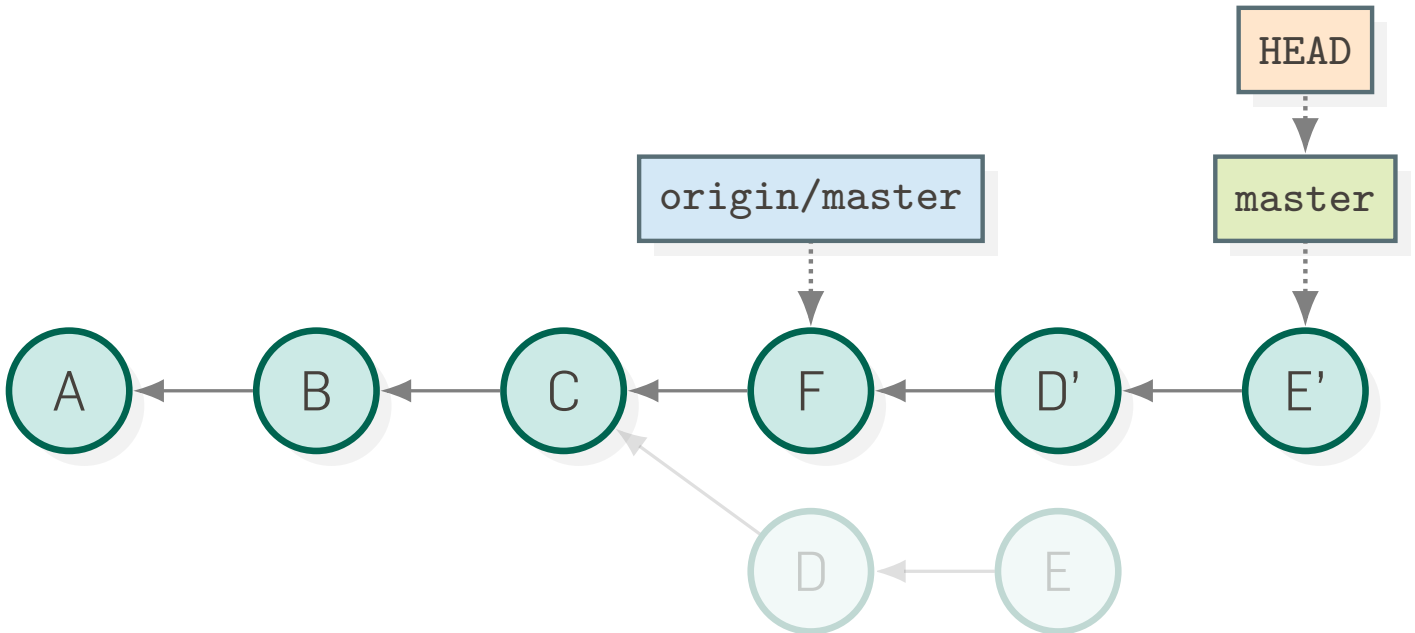
```
git fetch origin
```


origin/master
HEAD
A
B
C
F
master
D
E

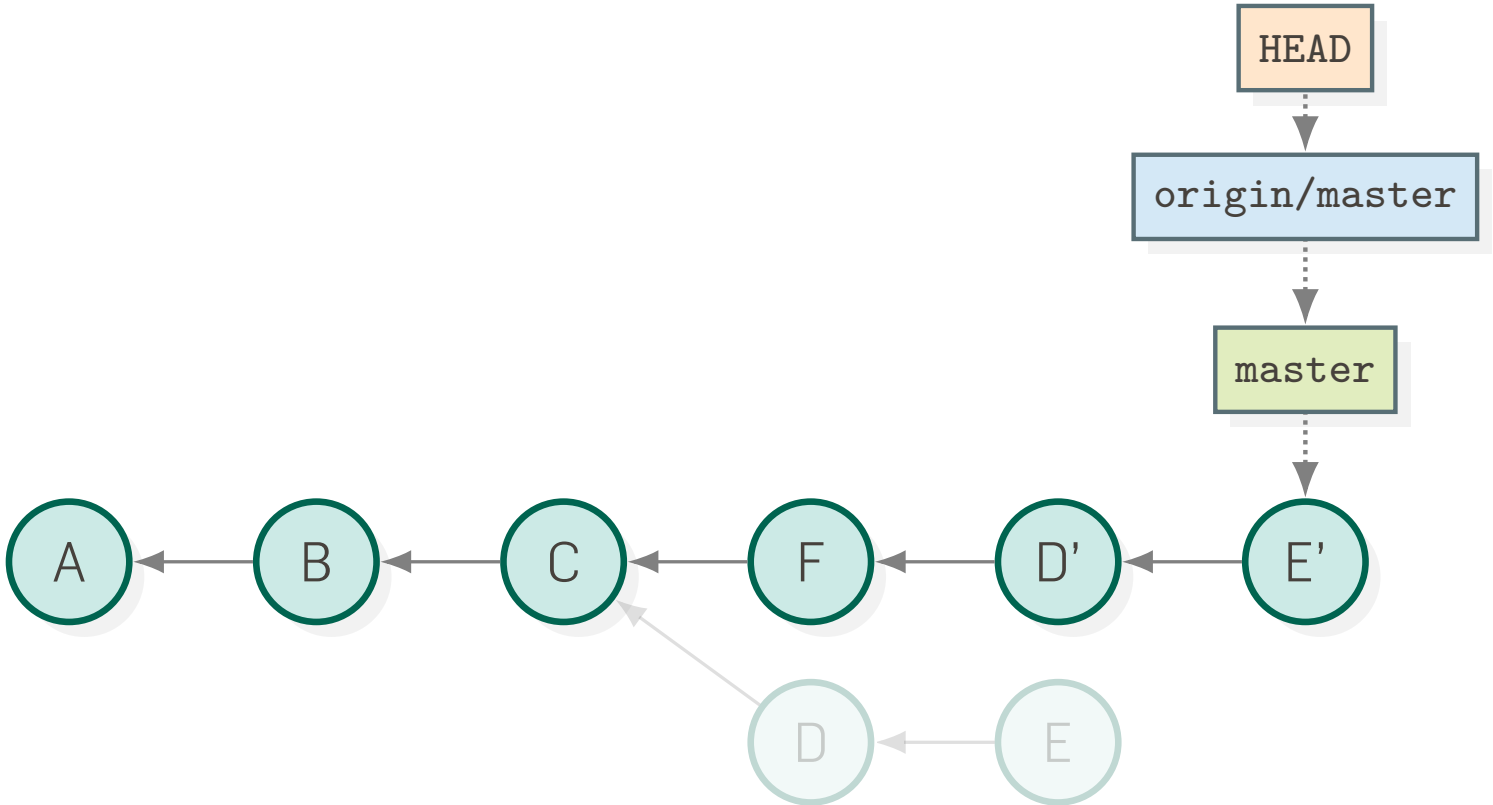


`git rebase origin/master`

(but you don't need it, because...)




```
git pull --rebase
```



`git push`

TO SUMMARIZE...

The first time...

```
$ git clone <url>
# for instance,
# git clone https://github.com/user/repo.git
```

Then...

```
$ cd <repo>
# make some changes...
$ git add <files>
$ git commit -m"<commit message>"
# ...
# when you want to share:
$ git pull --rebase # any changes on the remote?
$ git push
```

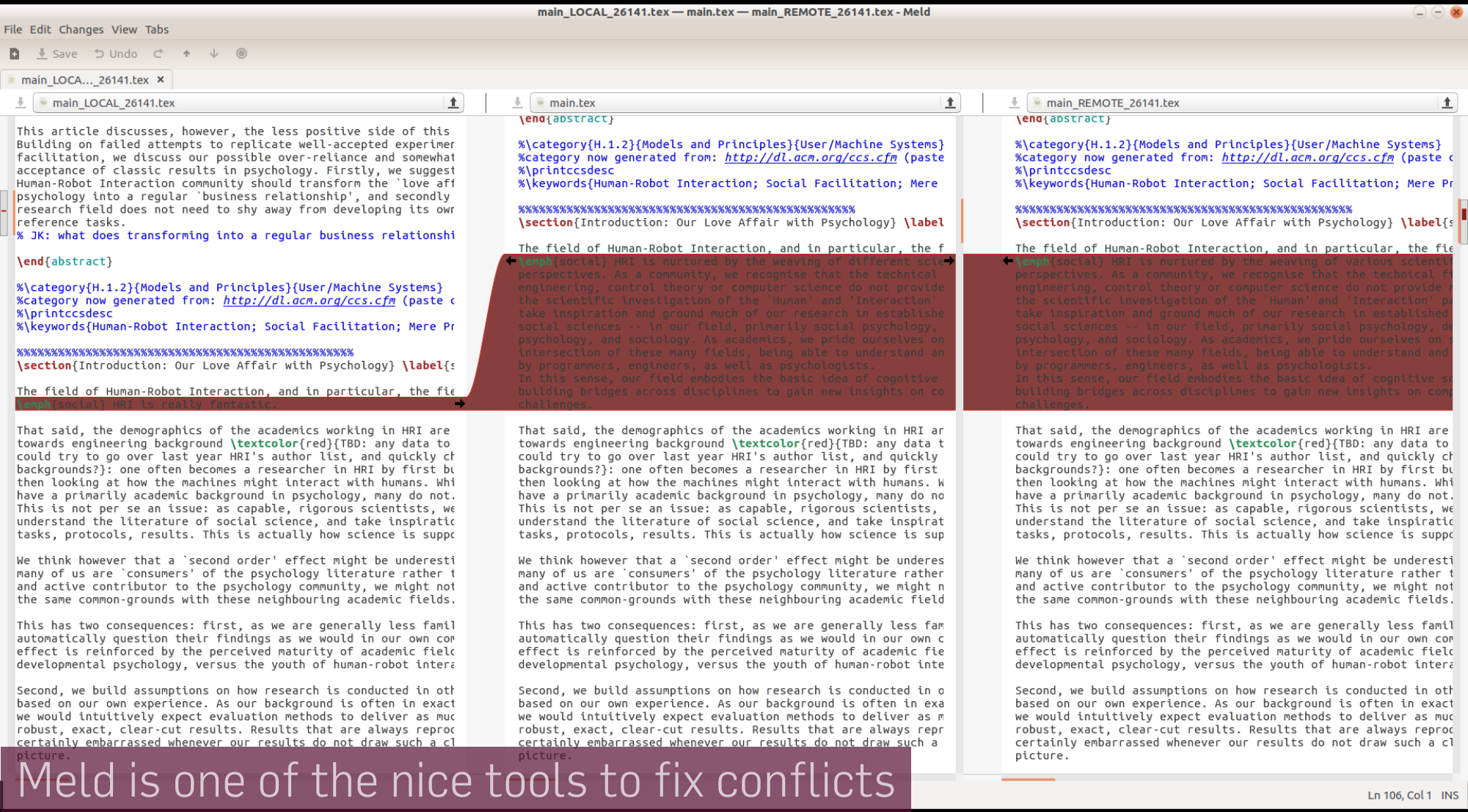
THE DREADFUL CONFLICTS

THE DREADFUL CONFLICT

While peacefully editing your last (great) paper...

```
$ git pull --rebase john master
First, rewinding head to replay your work on top of it...
Applying: Better terminology
Using index info to reconstruct a base tree...
M       main.tex
Falling back to patching base and 3-way merge...
Auto-merging main.tex
CONFLICT (content): Merge conflict in main.tex
error: Failed to merge in the changes.
Patch failed at 0001 Better terminology
The copy of the patch that failed is found in: .git/rebase-a

When you have resolved this problem, run "git rebase --conti
If you prefer to skip this patch, run "git rebase --skip" in
To check out the original branch and stop rebasing, run "git
```

```
$ git pull --rebase john master
# conflict!
$ git mergetool
```



Meld is one of the nice tools to fix conflicts

SOCIAL CODING: GITHUB WORKFLOW

morse-simulator / morse

Unwatch 28 Unstar 145 Fork 74

The Modular OpenRobots Simulation Engine http://morse-simulator.github.io/ — Edit

4,174 commits | 10 branches | 53 releases | 30 contributors

Branch: master New pull request | New file Find file HTTPS https://github.com/morse Download ZIP

adegroote [doc] Document more middleware addition — Latest commit 38fa4af 3 days ago

addons	[builder] fix few occurrences of removed method	a year ago
bin	[bin/morse_sync] Make sure to call it with the same python exec than ...	6 days ago
bindings	[bindings] Prepare for 1.4	7 days ago
config	[cmake] python 3.5 exists since September 2015	a month ago
data	[human] Removes all the code and support for the legacy human avatar	a month ago
doc	[doc] Document more middleware addition	3 days ago
examples	[builder] Rename Environment.set_simulator_frequency in Environemnt.s...	8 days ago
src	[mw/mavlink] Make sure to match only message of the 'good' type	3 days ago
testing	[builder] Rename Environment.set_simulator_frequency in Environemnt.s...	8 days ago
tools	[human] Removes all the code and support for the legacy human avatar	a month ago
.gitignore	add scene.*blend and eclipse files to gitignore	3 years ago
.mailmap	Added a mailmap to group variants of dev emails	3 years ago
.travis.bash	[travis] Upgrade the architecture for trusty	4 months ago
.travis.yml	[travis] Upgrade the architecture for trusty	4 months ago
AUTHORS	[doc] Fix several typos in credits	7 days ago
CITATION	[doc] Added a CITATION file	3 years ago
CMakeLists.txt	[bin/morse_sync] Make sure to call it with the same python exec than ...	6 days ago

GitHub

MakeHuman

ACTIONS

- Clone
- Compare
- Fork

NAVIGATION

- Overview
- Source
- Commits
- Branches
- Pull requests 1
- Downloads

Séverin Lemaignan / MakeHuman

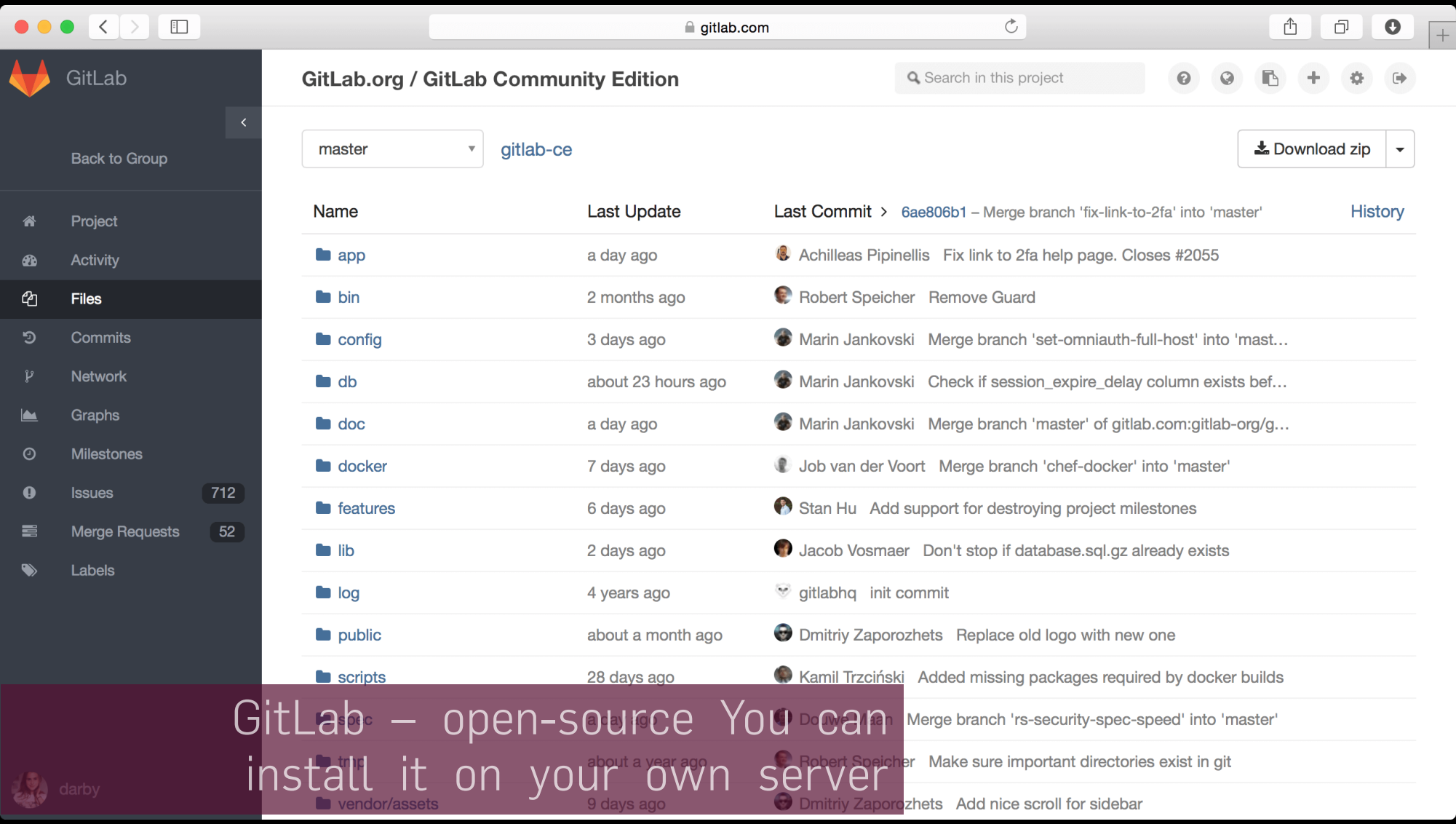
Source

default | MakeHuman /

blendertools			
buildscripts			
docs			
makehuman			
maketarget-standalone			
.hgeol	23 B	2014-02-03	Ensure use of LF native line endings for all text files, to avoid careless windows developers changing the line endings.
.hgignore	574 B	2014-03-18	merge with stable
.hgtags	47 B	2014-03-15	Cleanup hgtags
README	1.5 KB	2014-03-23	Add url to development tracker for dev status to readme

```
MakeHuman
=========

Makehuman is a completely free, innovative and professional software for the
modelling of 3-Dimensional humanoid characters. This is the official source
repository of the MakeHuman project.

Official website: http://www.makehuman.org
Development status: http://bugtracker.makehuman.org

License
-------

MakeHuman's source code and its mesh data is distributed freely under the AGPL3
license (see license.txt). Content created using the MakeHuman application is
released under the liberal CC0 license. For more details, refer to these pages:

* https://www.makehuman.org/doc/node/the_makehuman_application.html
* https://www.makehuman.org/doc/node/makehuman_mesh_licen[illegible]

Licenses for dependencies are included in the licenses folder.

Instructions
------------
```

BitBucket


gitlab.com
GitLab
GitLab.org / GitLab Community Edition
Search in this project
Back to Group
Project
Activity
Files
Commits
Network
Graphs
Milestones
Issues 712
Merge Requests 52
Labels
master
gitlab-ce
Download zip
Name
Last Update
Last Commit
6ae806b1 – Merge branch 'fix-link-to-2fa' into 'master'
History
app a day ago Achilleas Pipinellis Fix link to 2fa help page. Closes #2055
bin 2 months ago Robert Speicher Remove Guard
config 3 days ago Marin Jankovski Merge branch 'set-omniauth-full-host' into 'mast...
db about 23 hours ago Marin Jankovski Check if session_expire_delay column exists bef...
doc a day ago Marin Jankovski Merge branch 'master' of gitlab.com:gitlab-org/g...
docker 7 days ago Job van der Voort Merge branch 'chef-docker' into 'master'
features 6 days ago Stan Hu Add support for destroying project milestones
lib 2 days ago Jacob Vosmaer Don't stop if database.sql.gz already exists
log 4 years ago gitlabhq init commit
public about a month ago Dmitriy Zaporozhets Replace old logo with new one
scripts 28 days ago Kamil Trzciński Added missing packages required by docker builds
spec Merge branch 'rs-security-spec-speed' into 'master'
tmp about a year ago Robert Speicher Make sure important directories exist in git
vendor/assets 9 days ago Dmitriy Zaporozhets Add nice scroll for sidebar
darby
GitLab – open-source You can install it on your own server

morse-simulator / morse

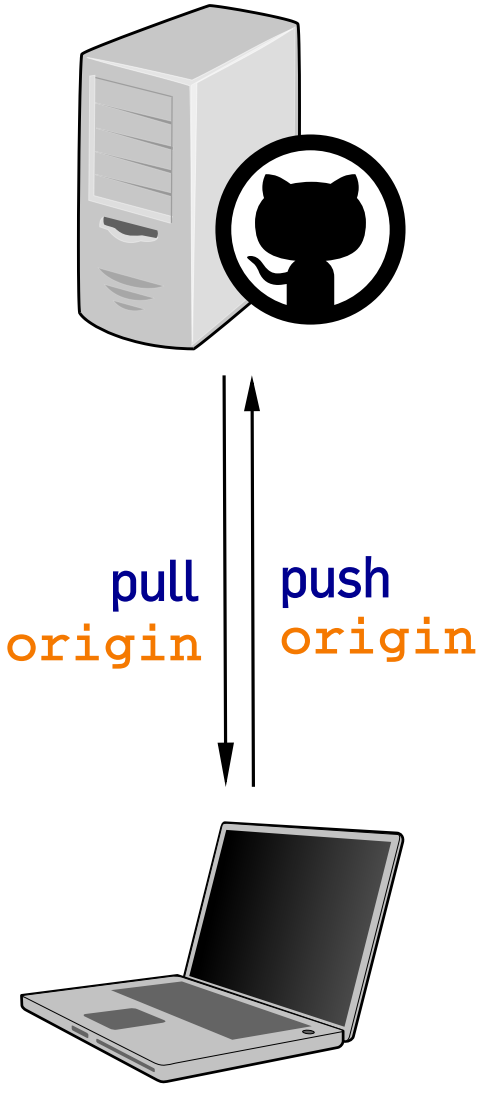
The Modular OpenRobots Simulation Engine http://morse-simulator.github.io/ — Edit

4,174 commits **10** branches **53** releases **30** contributors

Branch: **master** New pull request New file Find file HTTPS https://github.com/morse Download ZIP

adegroote [doc] Document more middleware addition Latest commit `38fa4af` 3 days ago

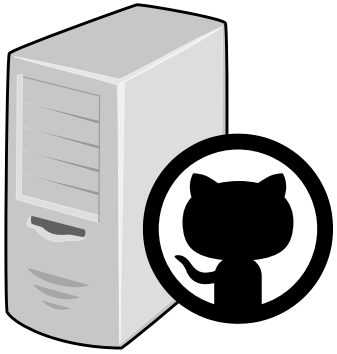
addons	[builder] fix few occurrences of removed method	a year ago
bin	[bin/morse_sync] Make sure to call it with the same python exec than …	6 days ago
bindings	[bindings] Prepare for 1.4	7 days ago
config	[cmake] python 3.5 exists since September 2015	a month ago
data	[human] Removes all the code and support for the legacy human avatar	a month ago
doc	[doc] Document more middleware addition	3 days ago
examples	[builder] Rename Environment.set_simulator_frequency in Environemnt.s…	8 days ago
src	[mw/mavlink] Make sure to match only message of the 'good' type	3 days ago
testing	[builder] Rename Environment.set_simulator_frequency in Environemnt.s…	8 days ago
tools	[human] Removes all the code and support for the legacy human avatar	a month ago
.gitignore	add scene.*blend and eclipse files to gitignore	3 years ago
.mailmap	Added a mailmap to group variants of dev emails	3 years ago
.travis.bash	[travis] Upgrade the architecture for trusty	4 months ago
.travis.yml	[travis] Upgrade the architecture for trusty	4 months ago
AUTHORS	[doc] Fix several typos in credits	7 days ago
CITATION	[doc] Added a CITATION file	3 years ago
CMakeLists.txt	[bin/morse_sync] Make sure to call it with the same python exec than …	6 days ago

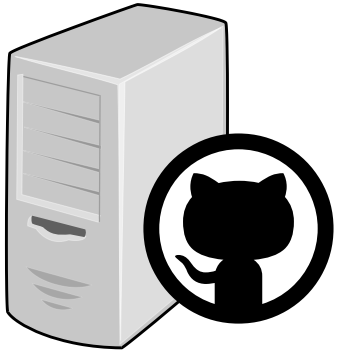

github.com/john/cool_app
pull
origin
push
origin
John

FORK
(a clone, really)

`github.com/john/cool_app`

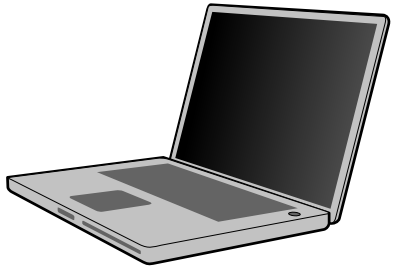
`github.com/paul/cool_app`



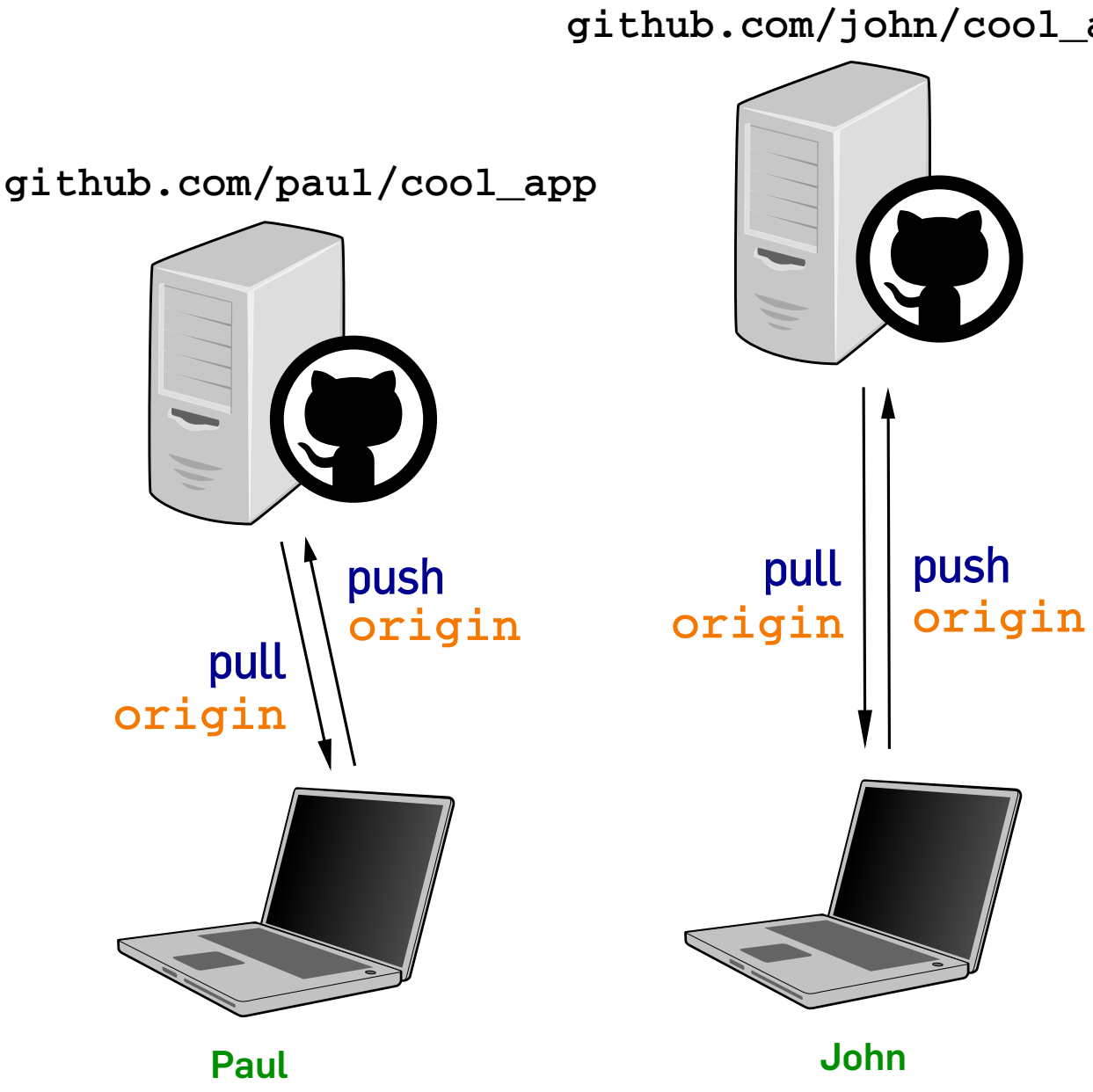


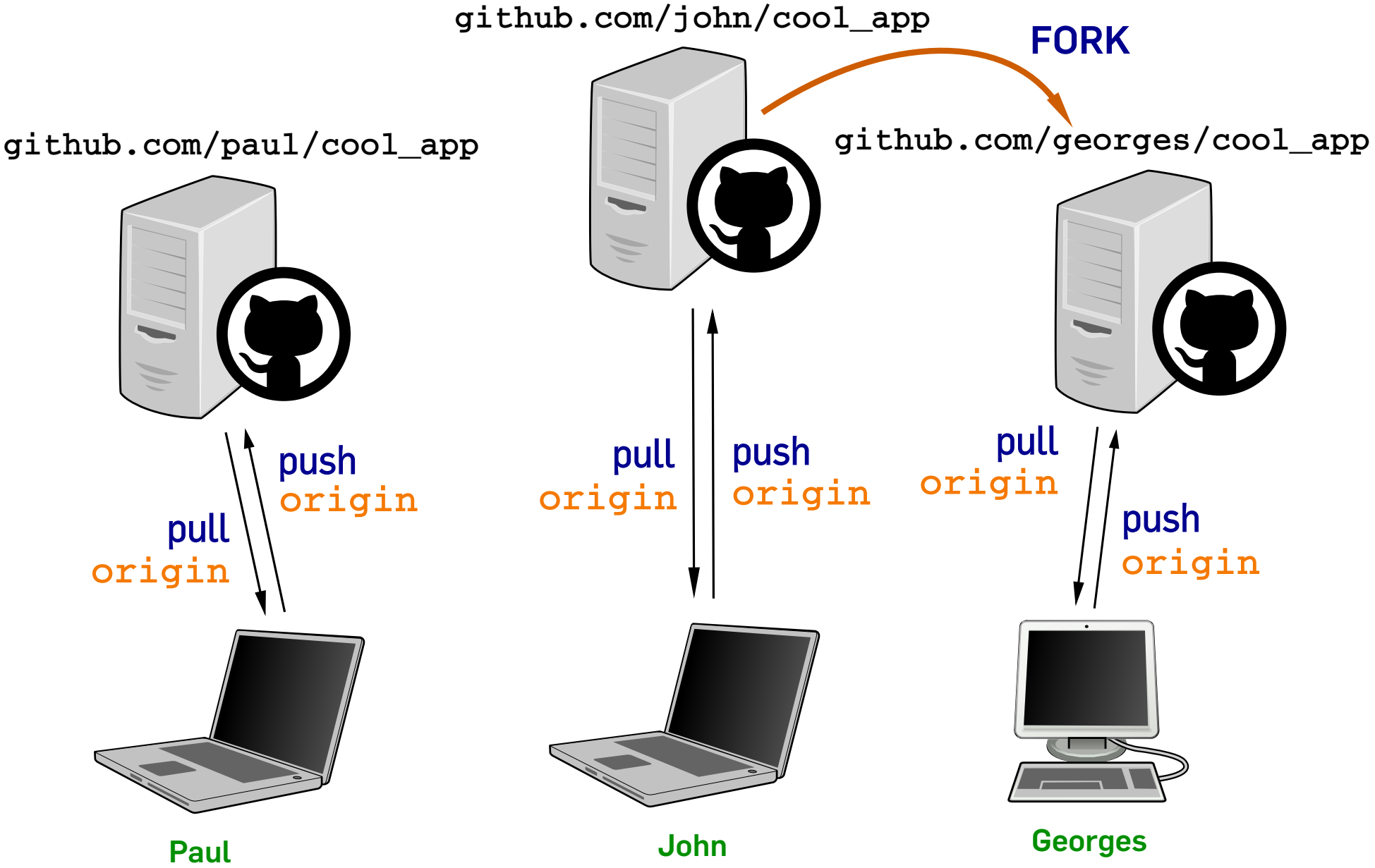
pull `origin`

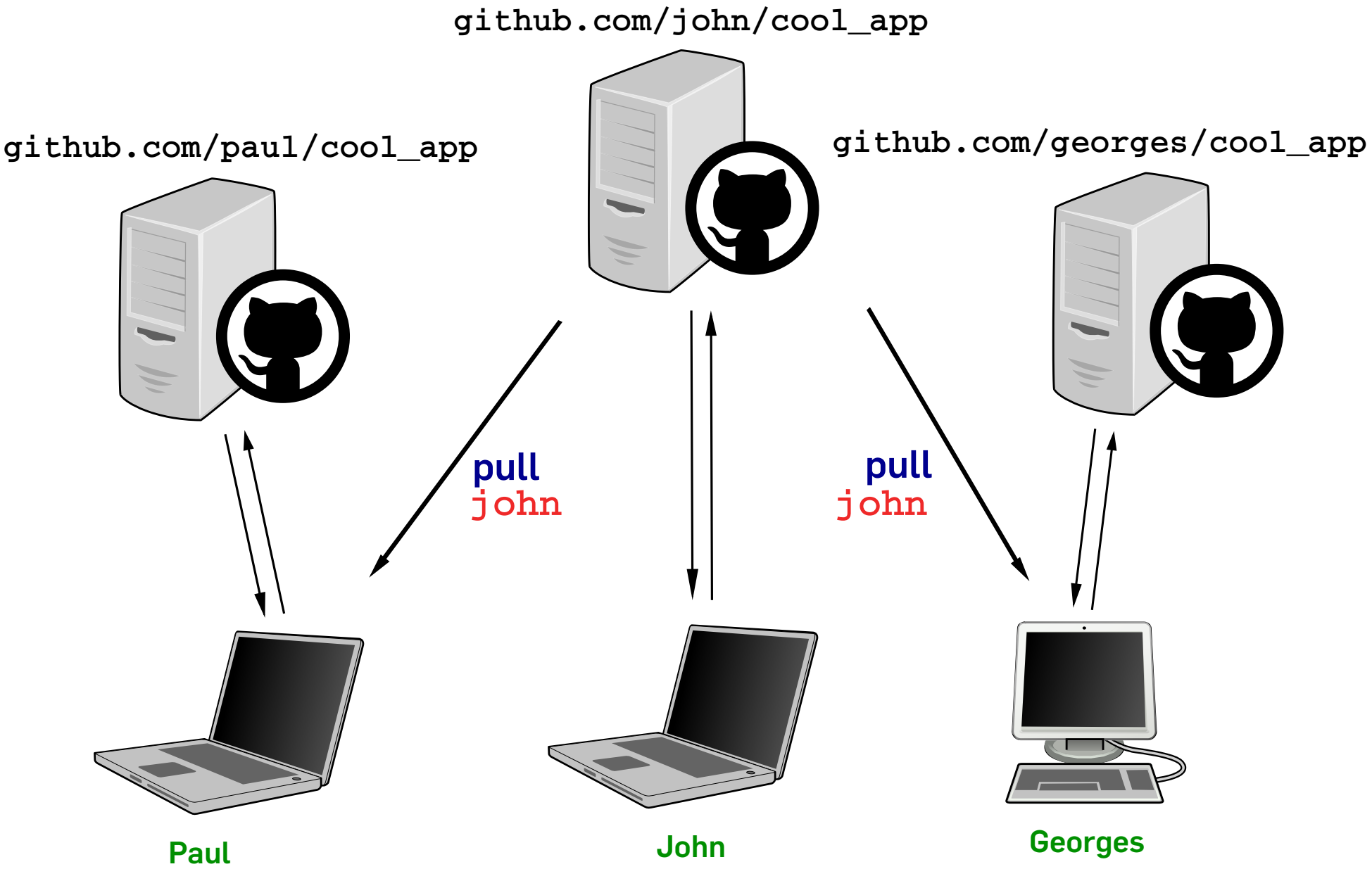
push `origin`

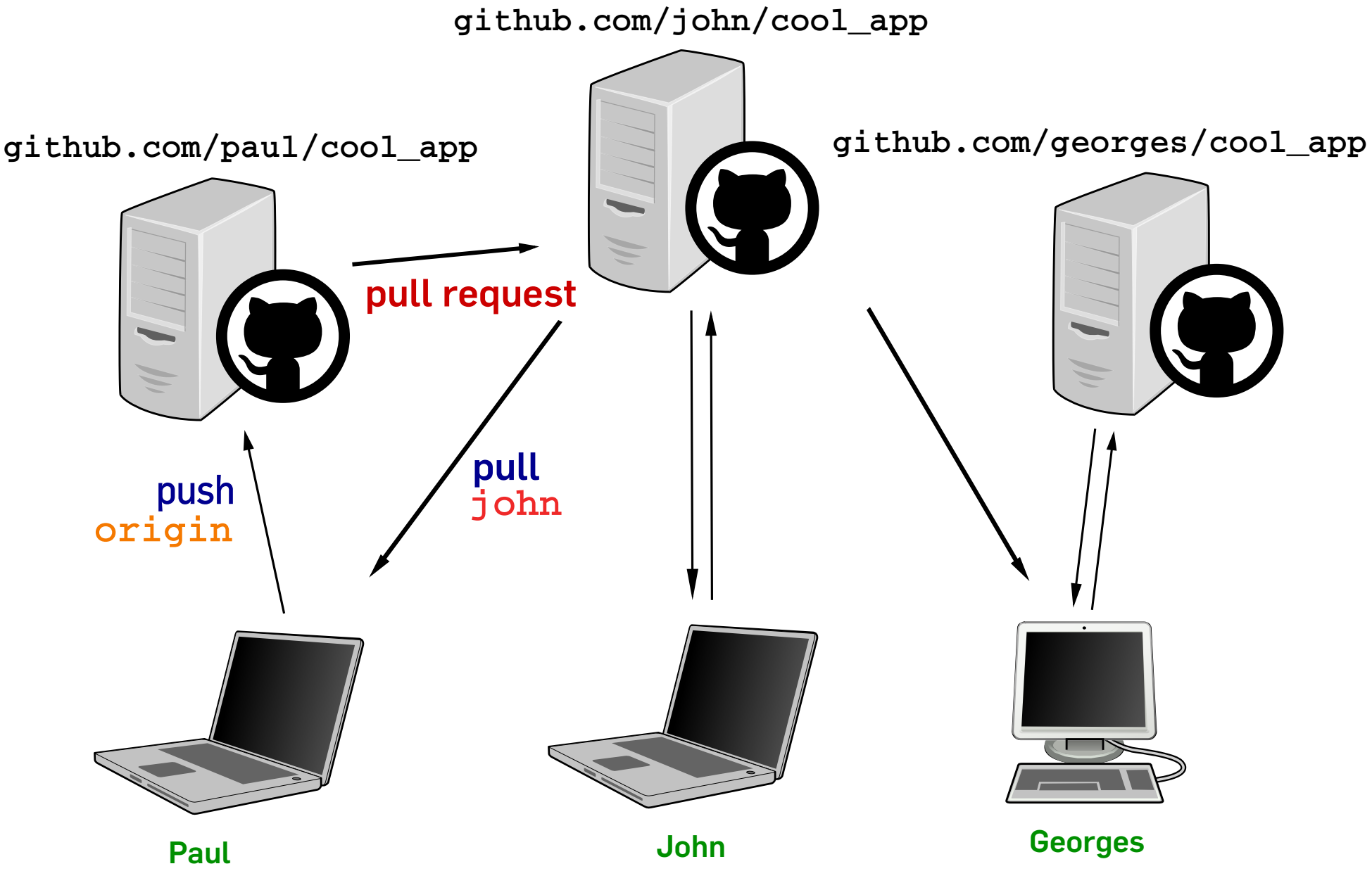


John

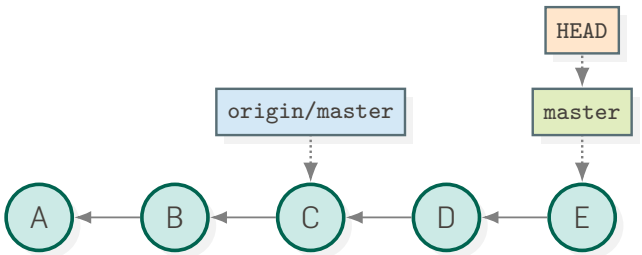

github.com/john/cool_app
github.com/paul/cool_app
push
origin
pull
origin
pull
origin
push
origin
Paul
John

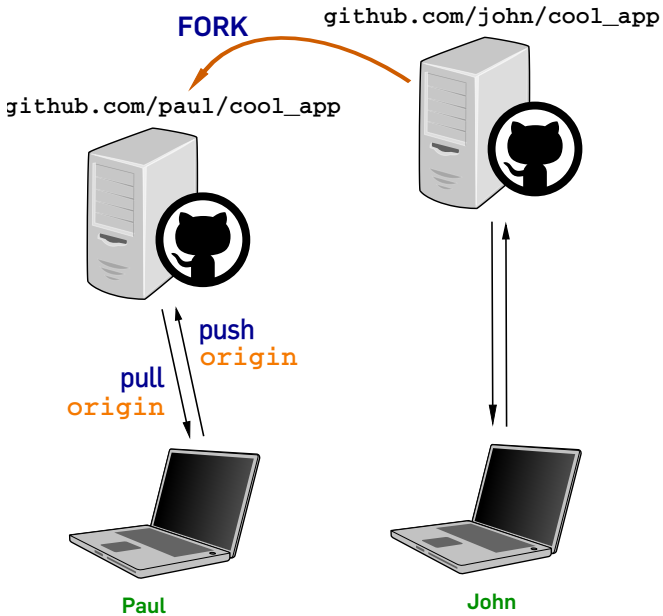

github.com/john/cool_app
FORK
github.com/paul/cool_app
github.com/georges/cool_app
push
origin
pull
origin
pull
origin
push
origin
pull
origin
push
origin
Paul
John
Georges


github.com/john/cool_app
github.com/paul/cool_app
github.com/georges/cool_app
pull
john
pull
john
Paul
John
Georges


github.com/john/cool_app
github.com/paul/cool_app
github.com/georges/cool_app
pull request
push
origin
pull
john
Paul
John
Georges

WHAT HAPPENED EXACTLY?



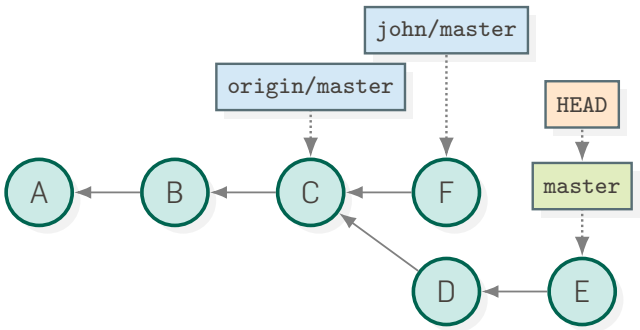


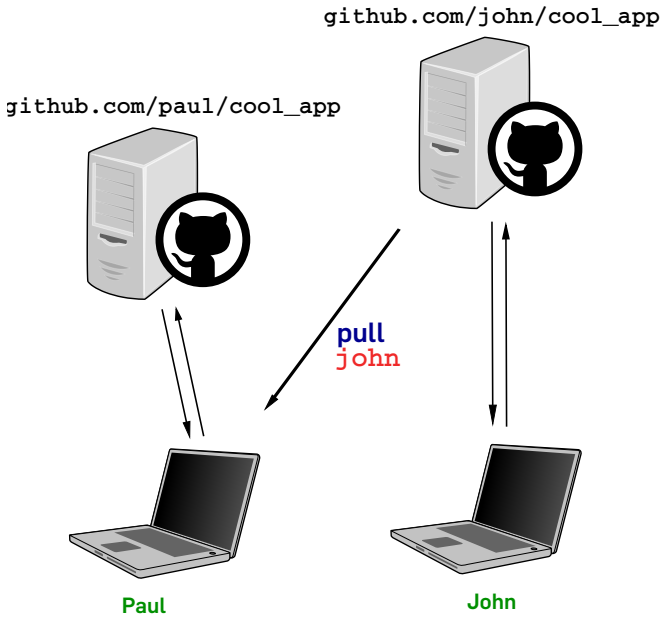
After forking on GitHub, Paul runs

`git clone https://github.com/paul/cool_app.git`

and he adds few local commits

WHAT HAPPENED EXACTLY?



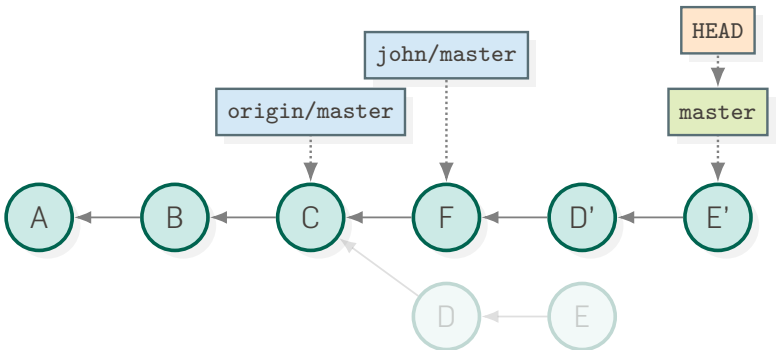


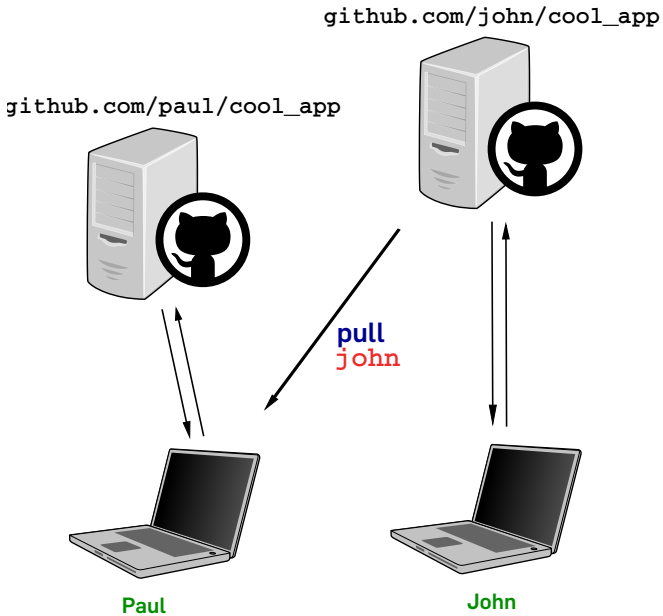
He would like to propose his changes to John

First, he needs to get the latest changes from John:

```
git add remote john https://github.com/john/cool_app.git
git fetch john
```

WHAT HAPPENED EXACTLY?



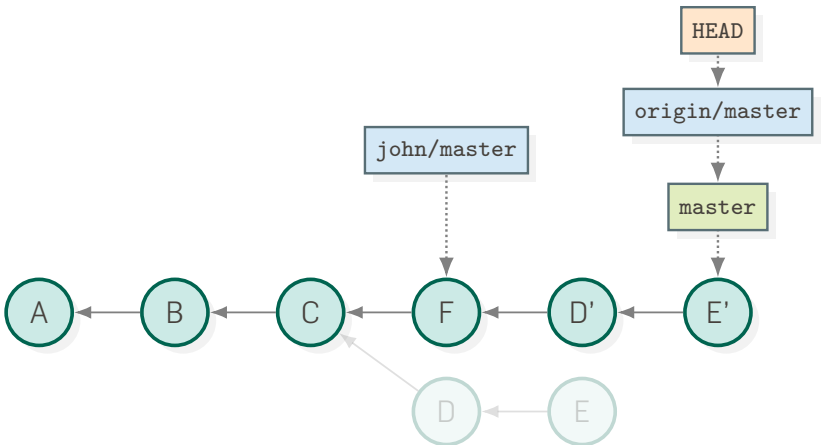


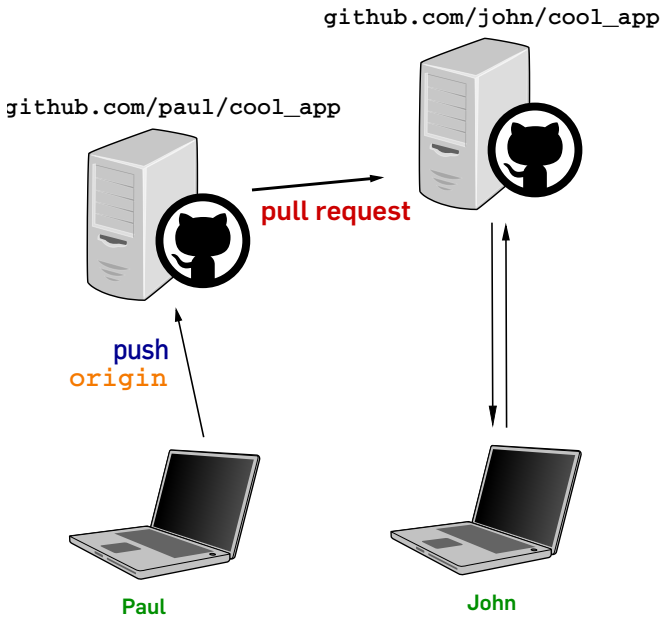
Paul rebases his `master` branch on John's one:

`git rebase john/master`

(actually, Paul would simply run `git pull --rebase john master`)

WHAT HAPPENED EXACTLY?





He pushes his commits to his own GitHub account:

`git push`

...and finally press the “Create a pull request” button in GitHub.

(what happens next on John's side is a story for another day :-)
But to make it short, he can press "Merge pull request" on his GitHub account if he is happy with the pull-request!)

FULLY STANDALONE REPO!
FULLY STANDALONE REPO!
FULLY STANDALONE REPO!
FULLY STANDALONE REPO!
FULLY STANDALONE REPO!
FULLY STANDALONE REPO!
Paul
John
Georges

THE ONE SLIDE TO REMEMBER

GIT CHEAT SHEET

To start...
...from scratch: `git init`
...from existing repo: `git clone <url>`

Prepare commits:
`git add`
`git rm`
`git add -p` (partial files)

Commit:
`git commit`

Create branch:
`git checkout -b <branch>`

Jump between branches:
`git checkout <branch>`

“Import” another branch:
`git rebase <other_branch>`

Add a remote source:
`git remote add <name> <url>`

What’s new on a remote?
`git pull <remote> <branch>`
(`git pull` alone ≡ `git pull origin master`)

Share stuff on a remote:
`git push <remote> <branch>`
(`git push` alone ≡ `git push origin master`)

Repo state
`git status`

Repo history
`git log`

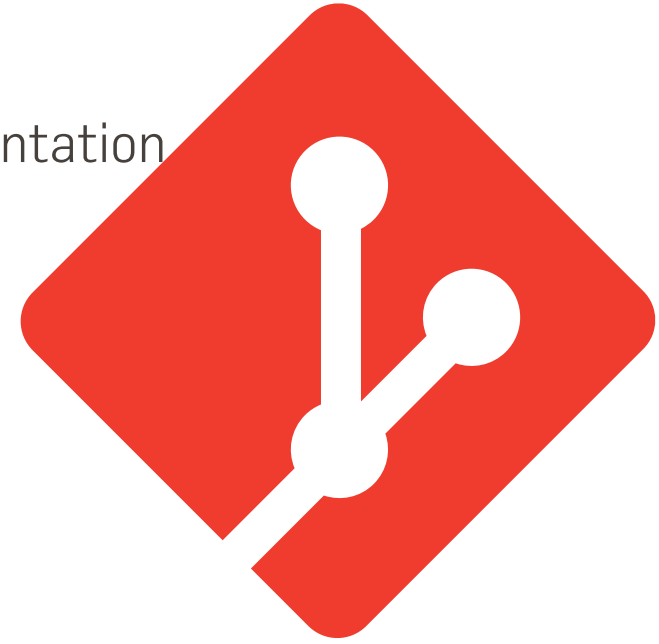
Who did what?
`git blame`

I’ve lost everythg!
`git reflog`

That’s all, folks!

Questions:
Portland Square A216 or **severin.lemaignan@plymouth.ac.uk**

Slides: github.com/severin-lemaignan/git-presentation



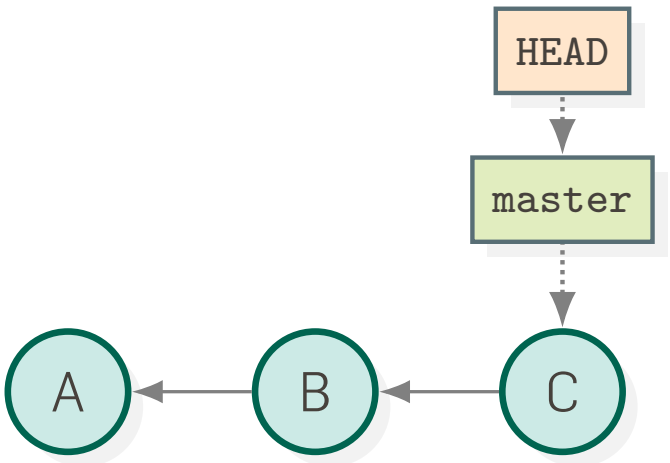
SUPPLEMENTARY MATERIAL

6. Working with branches

7. Etiquette of social coding 101

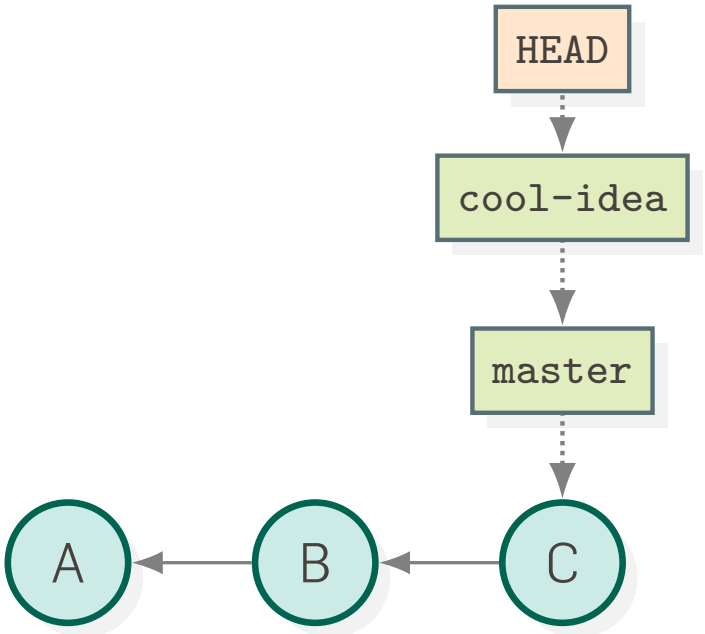
WORKING WITH BRANCHES

BRANCHES



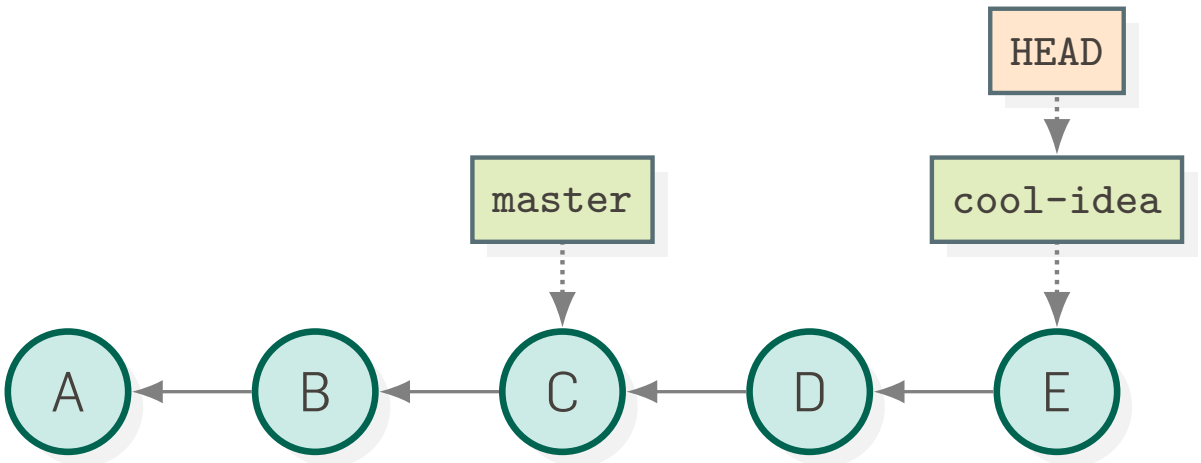
What if...?

BRANCHES

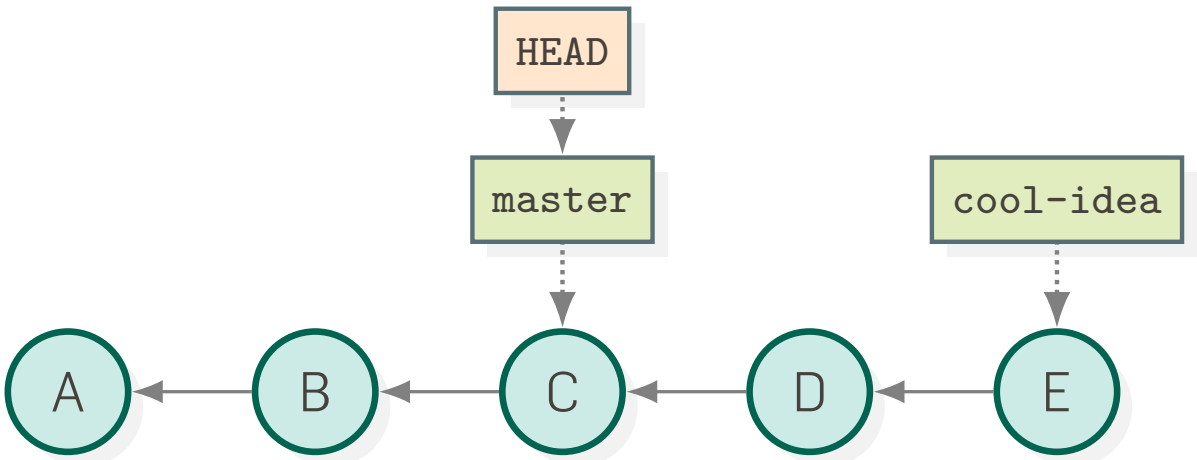



```
git checkout -b cool-idea
```

BRANCHES



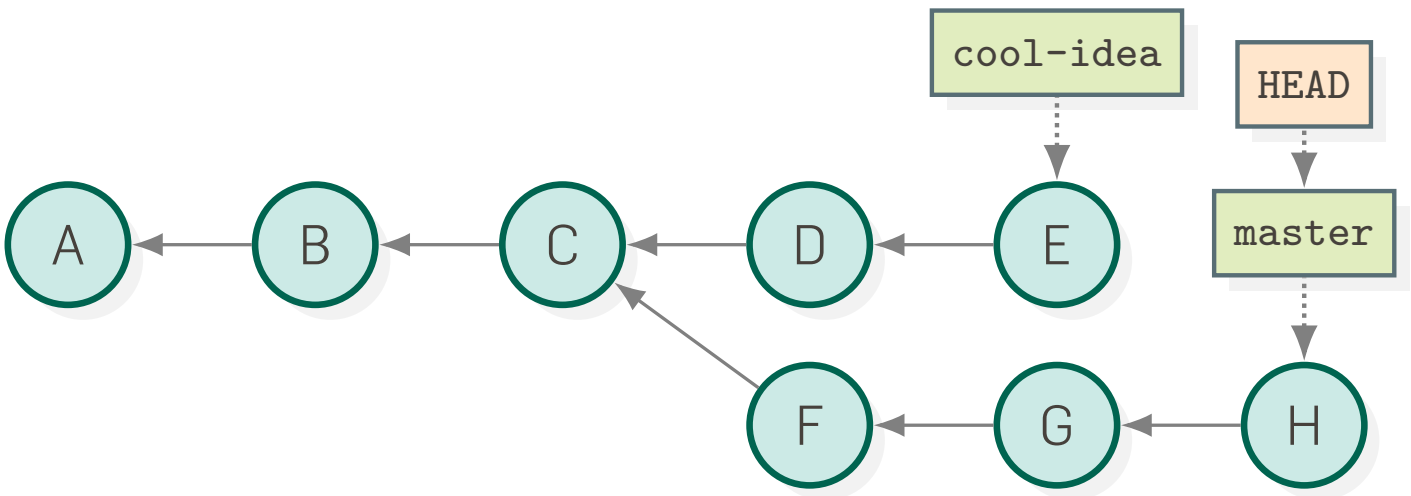
BRANCHES



Let go back to serious stuff!

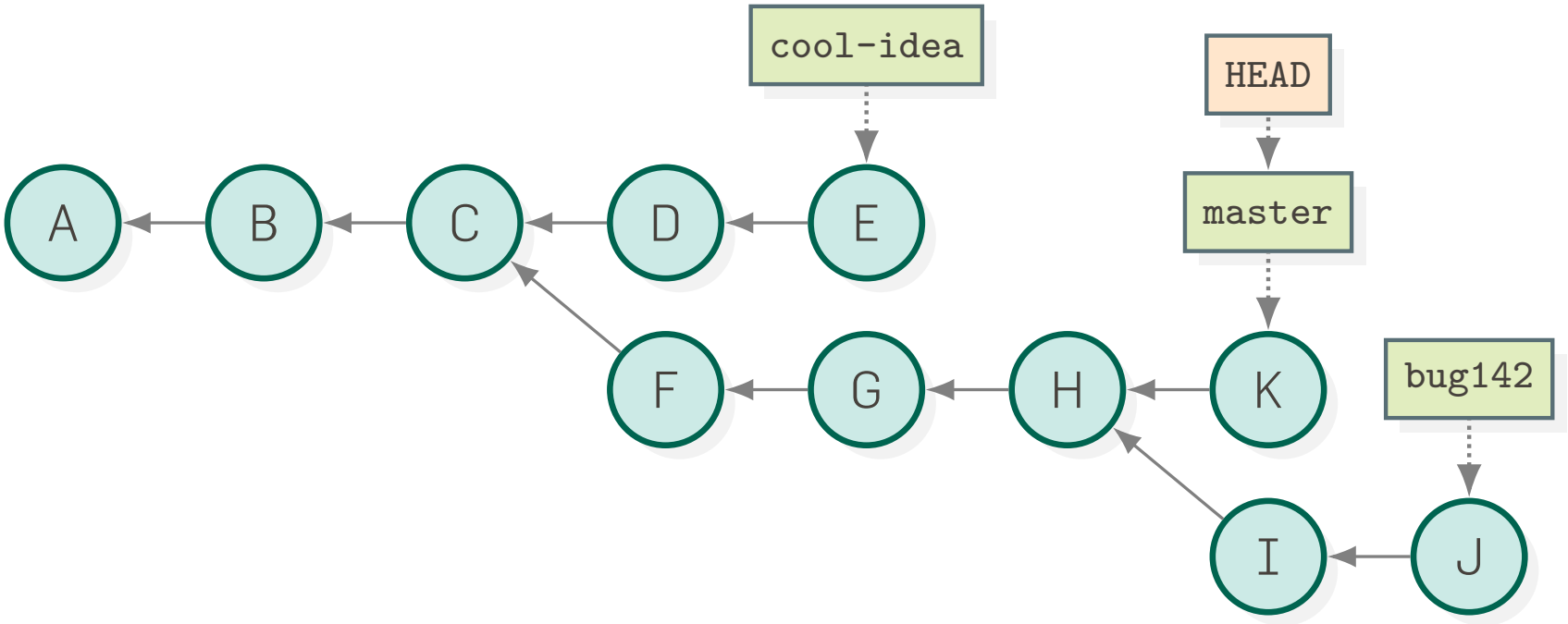
```
git checkout master
```

BRANCHES



The branch name is an alias for the tip of the current branch

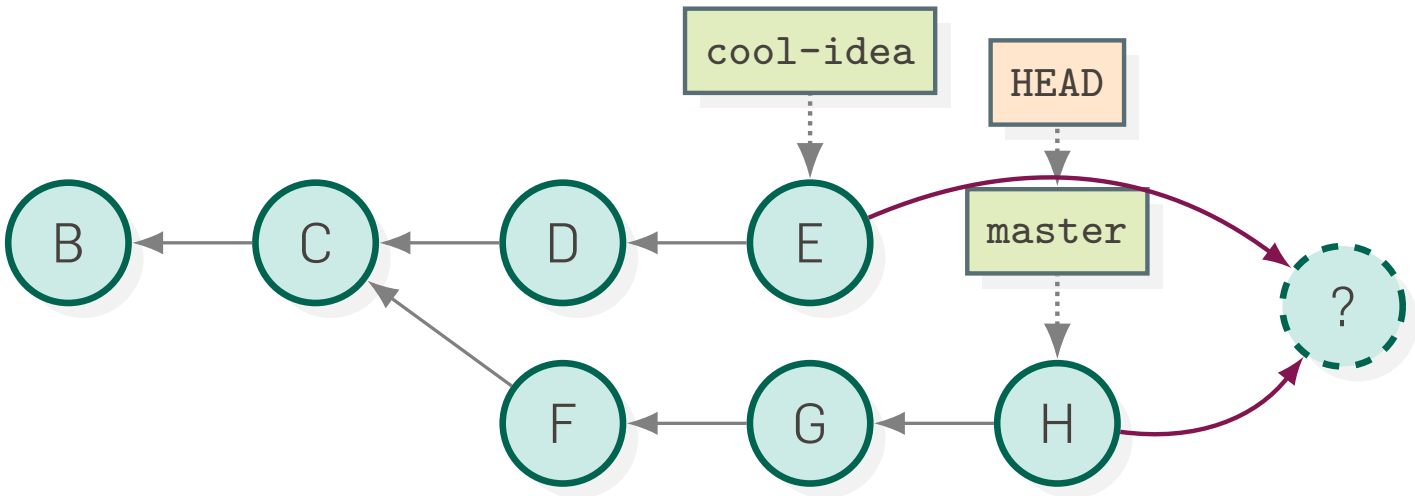
BRANCHES



⇒ branches are very cheap

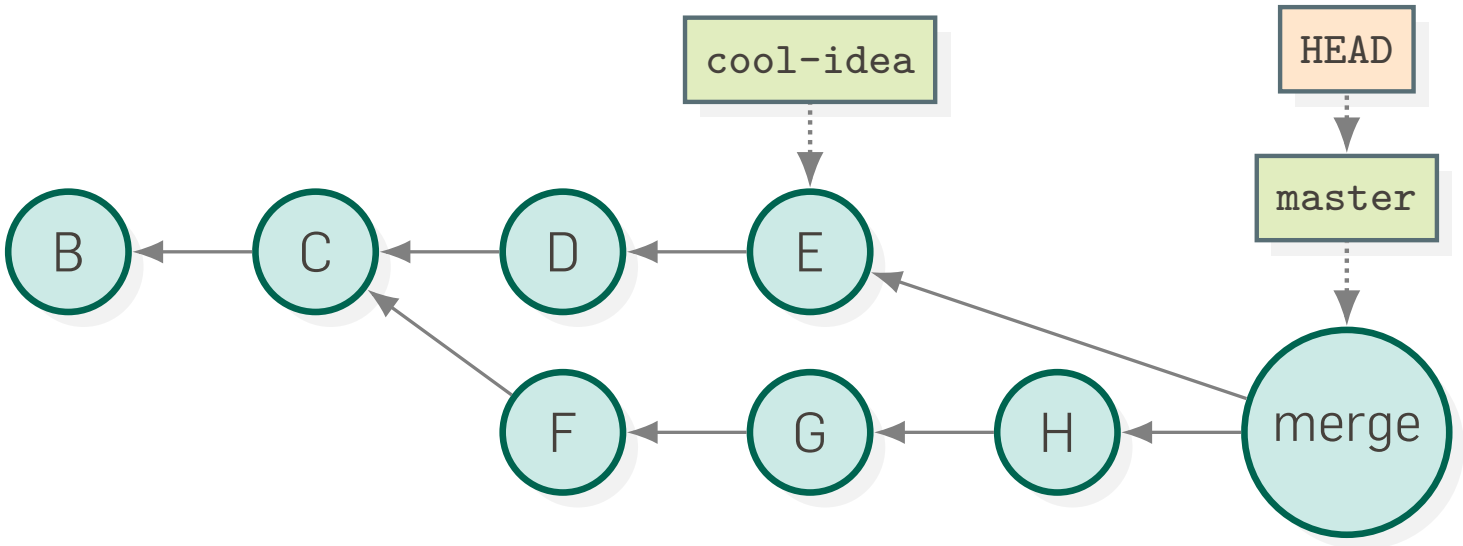
+10 of them at a given time it not uncommon

MERGING BRANCHES



Two options: **merging** and **rebasing**

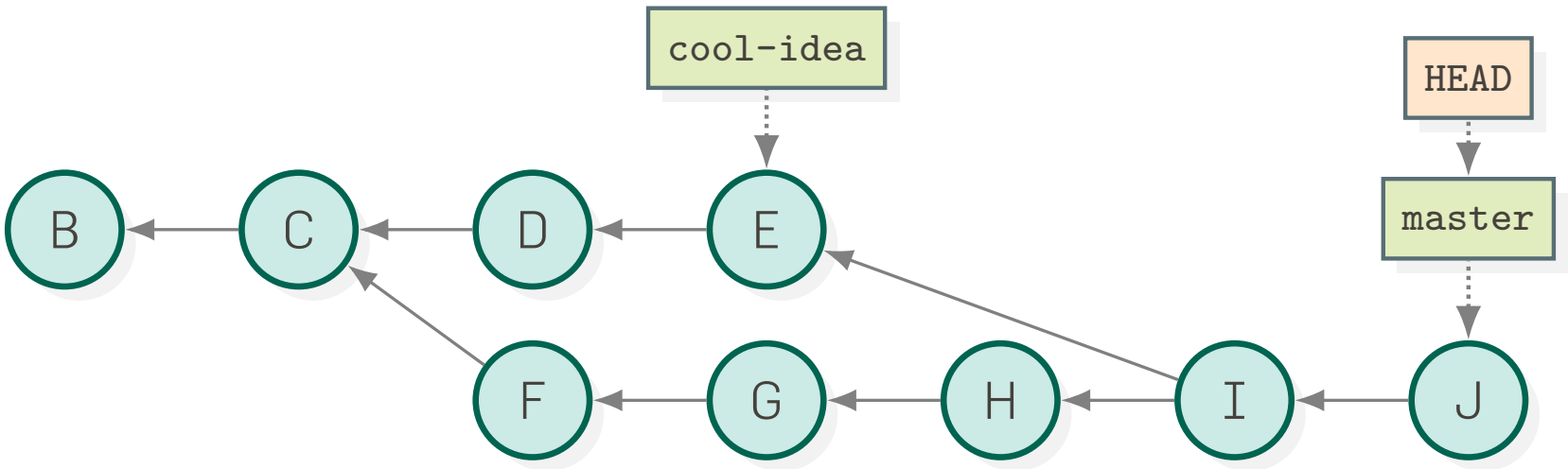
MERGING BRANCHES



Merging

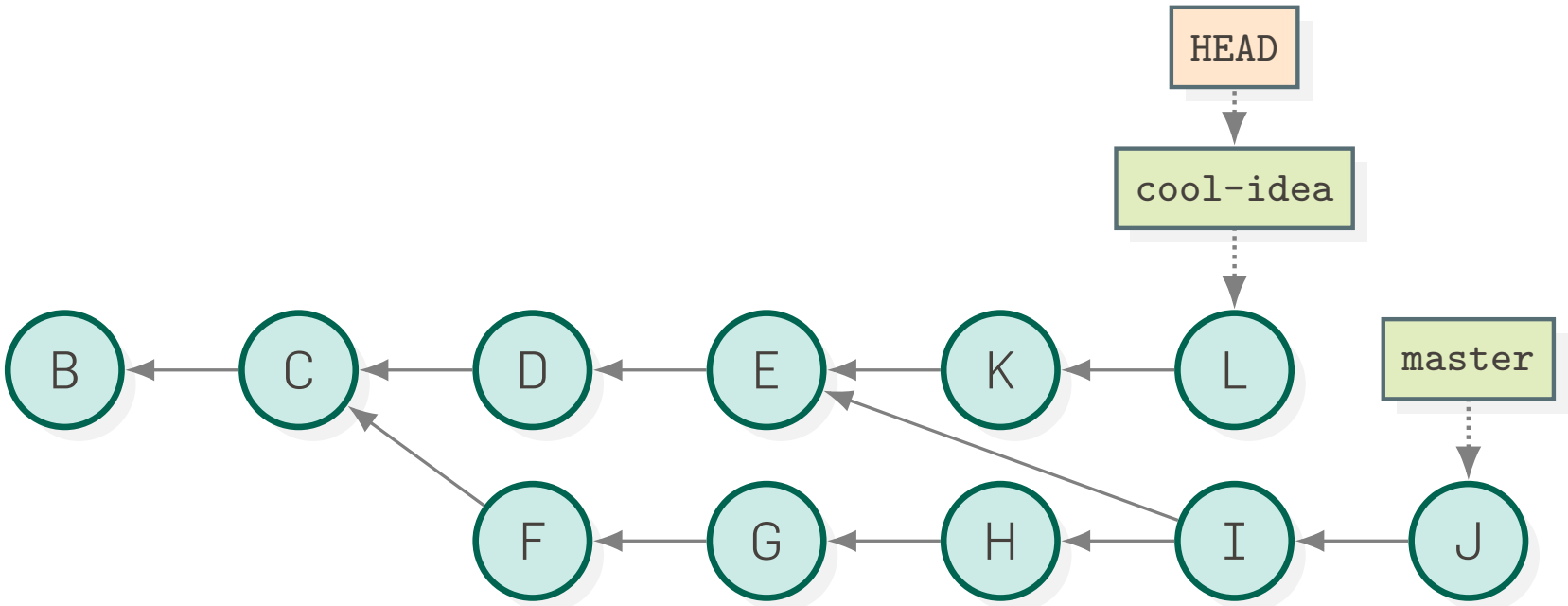
```
git merge cool-idea
```

MERGING BRANCHES



git commit

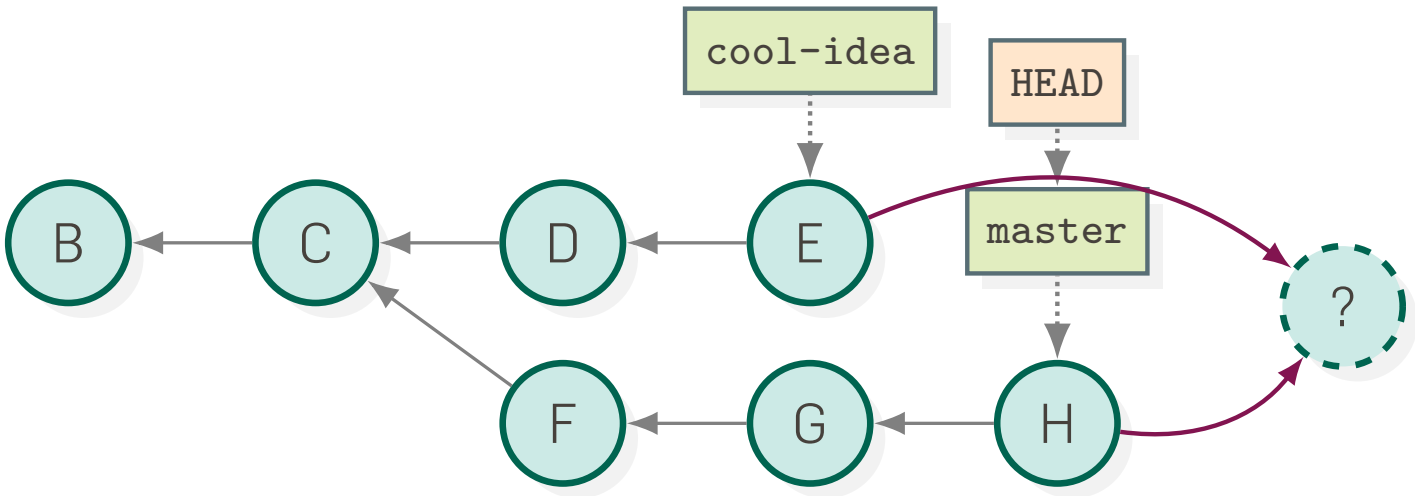
MERGING BRANCHES



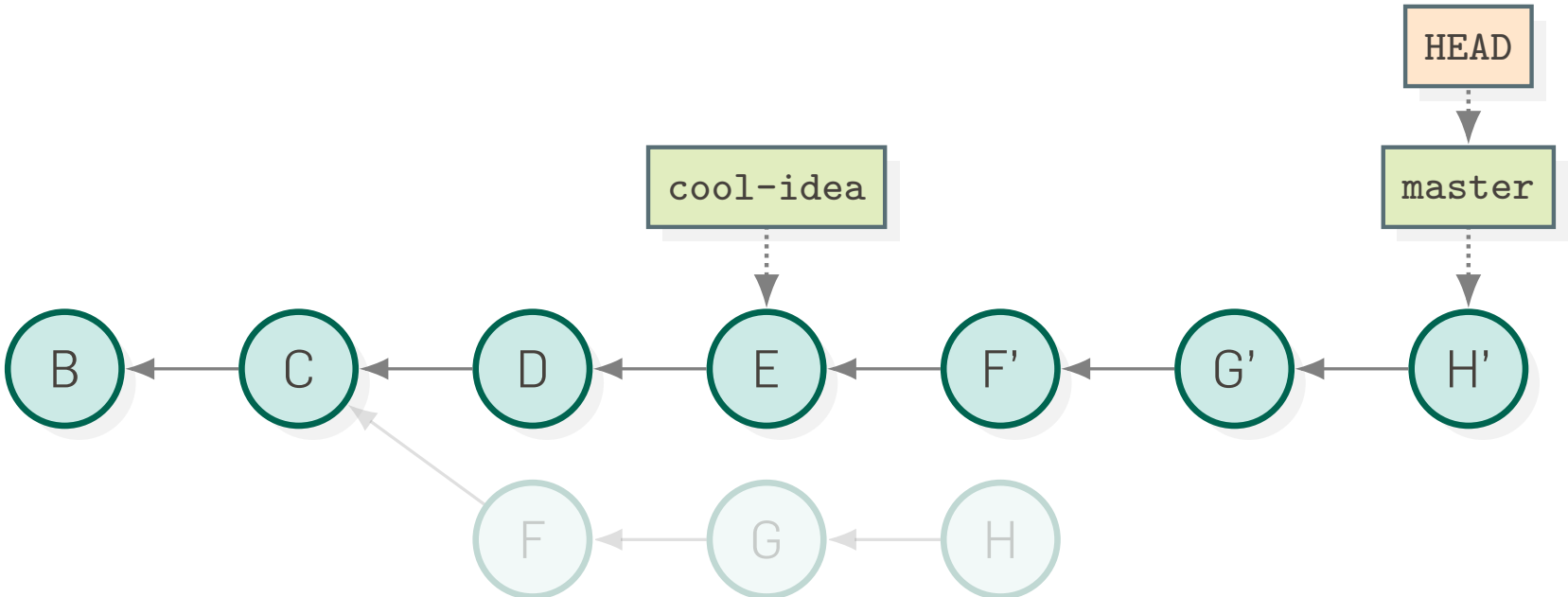

```
git checkout cool-idea
git commit
```

...etc.

REBASING BRANCHES



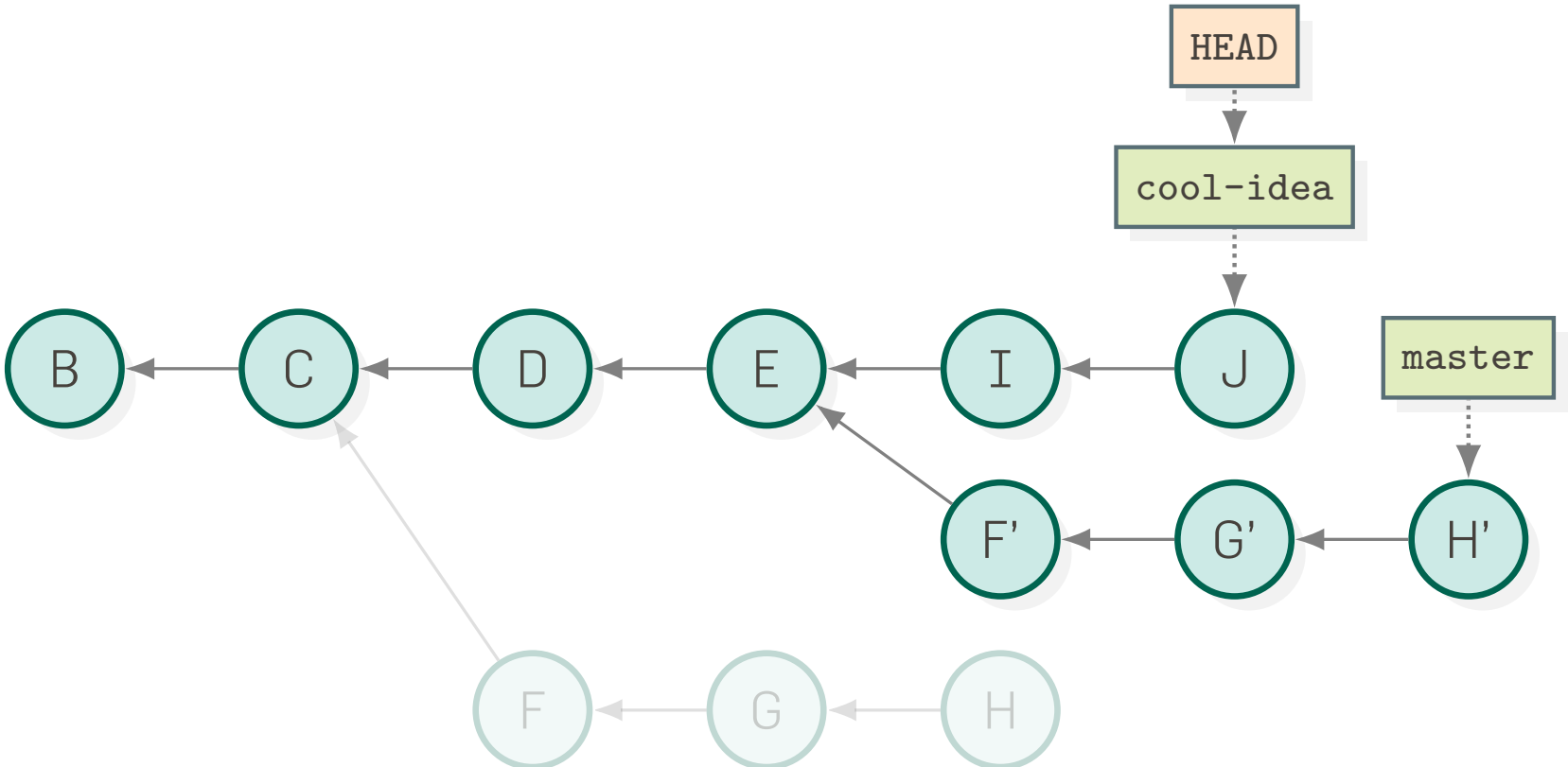
REBASING BRANCHES



Rebasing

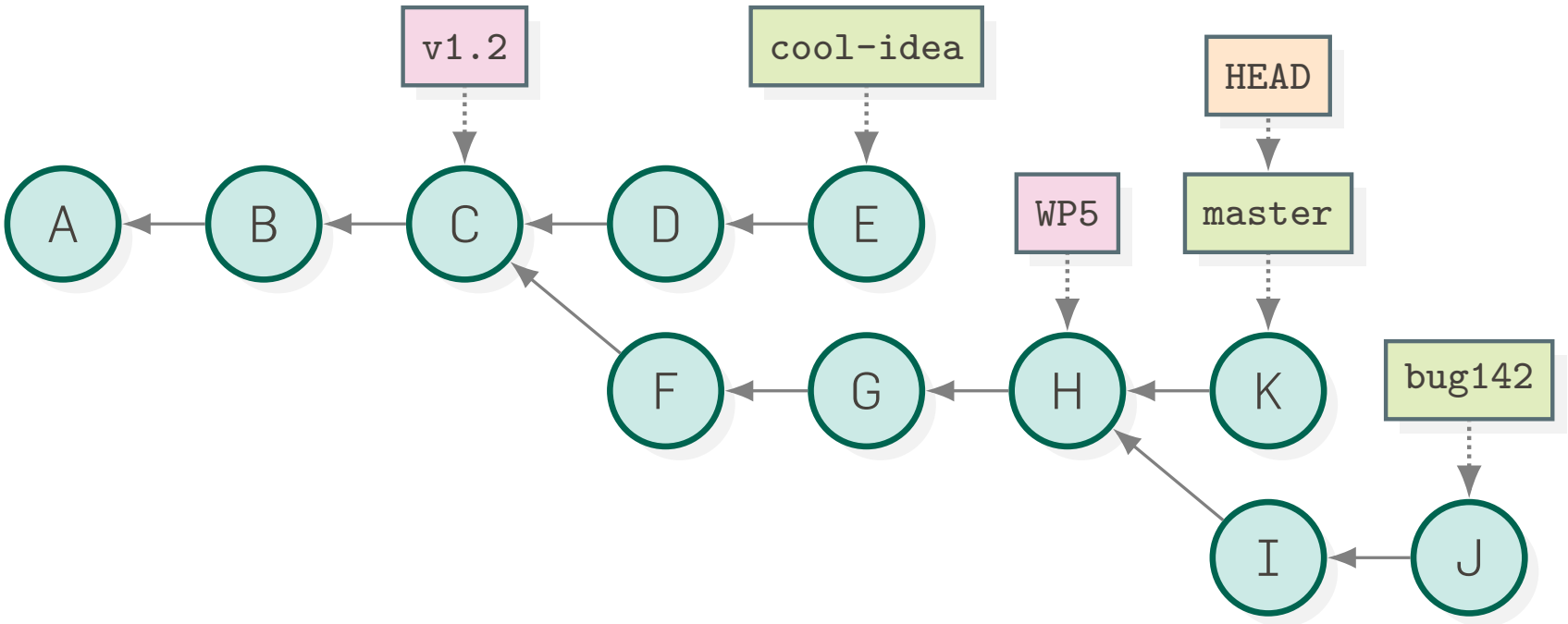
```
git rebase cool-idea
```

REBASING BRANCHES




```
git checkout cool-idea
git commit
```

MORE COMMIT ALIASES: TAGS



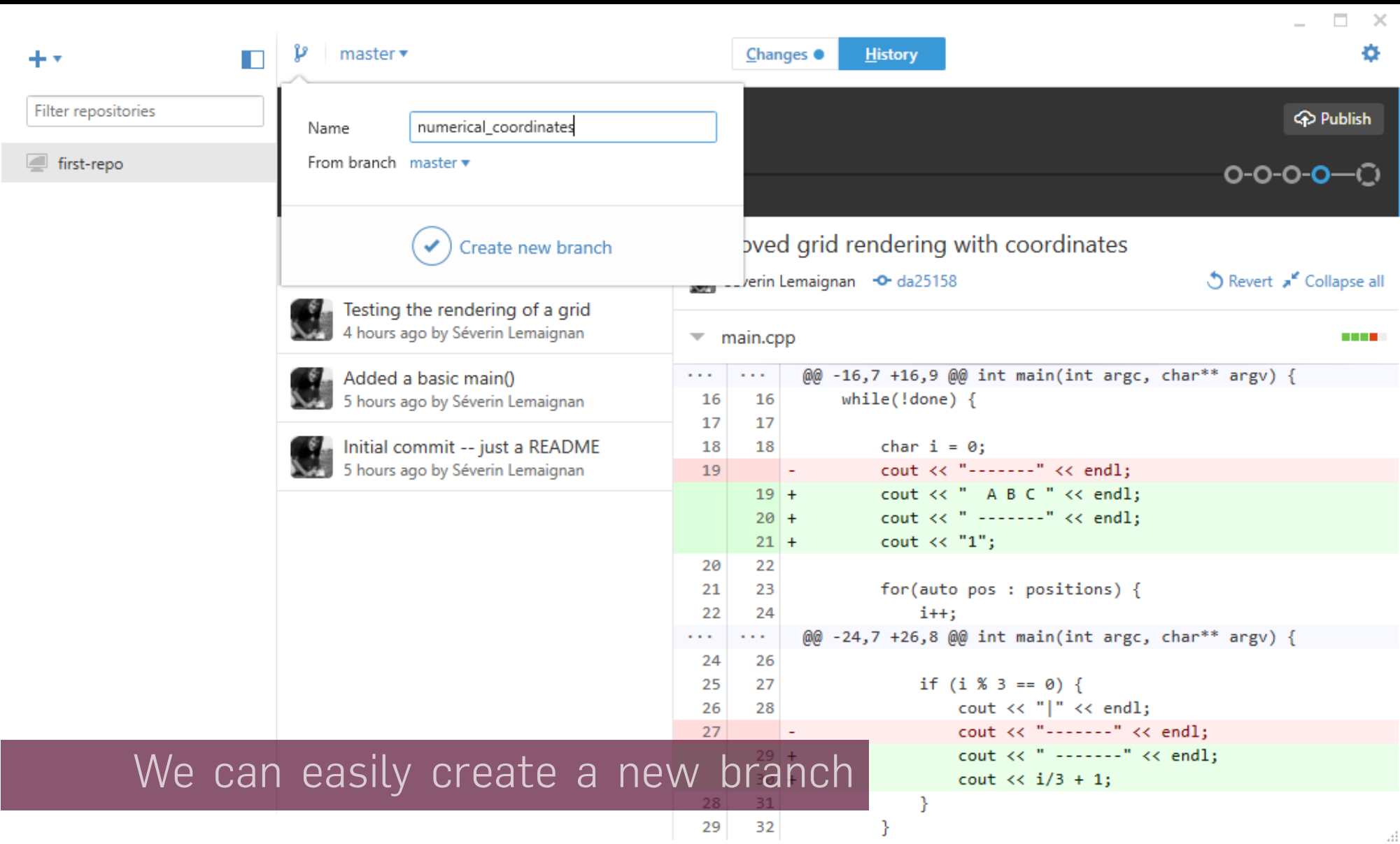
Label important commits/milestones

```
git tag v1.2
git tag WP5
```

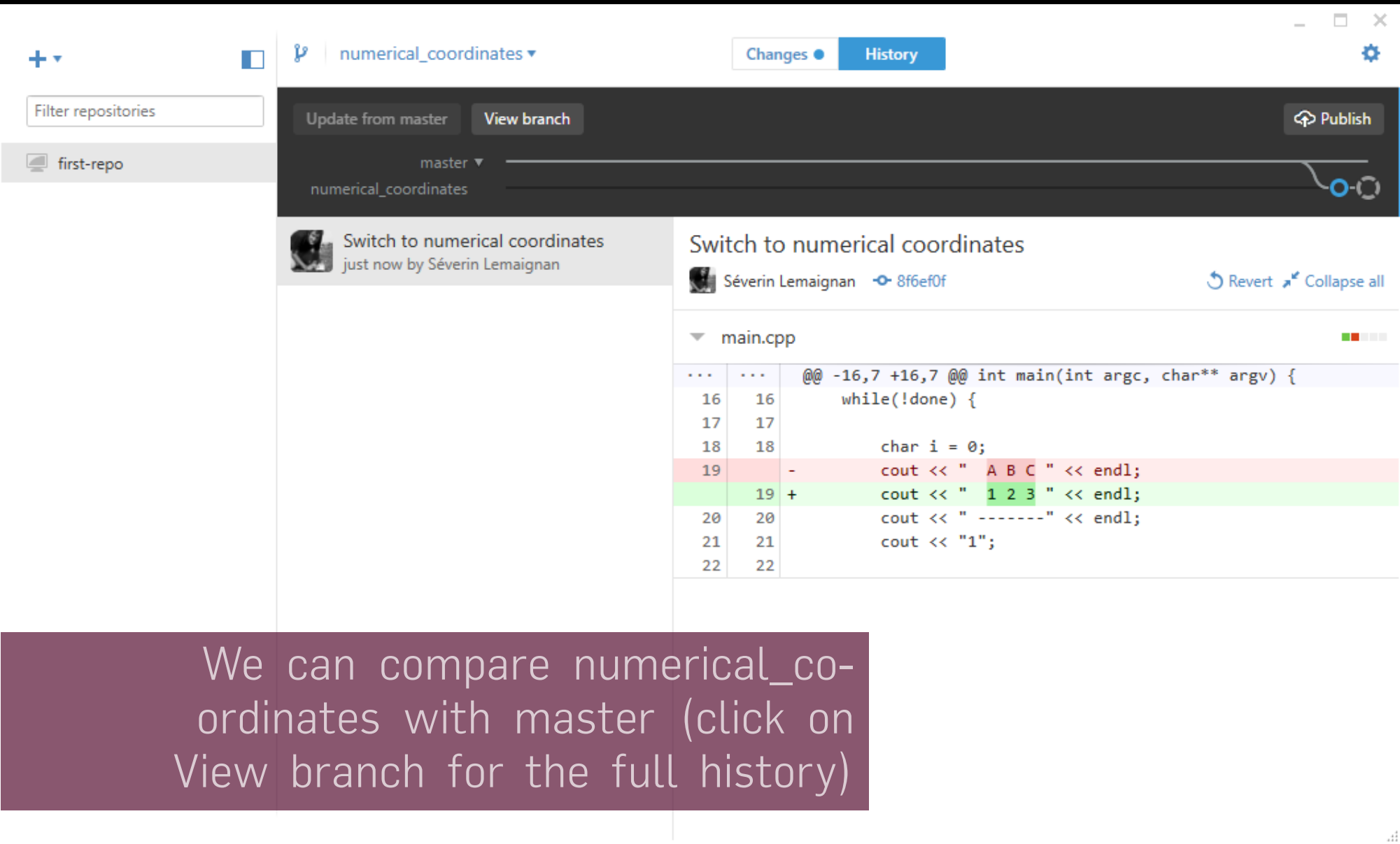
TO SUMMARIZE...

```
# where are we?
$ git branch
master
# make some changes...
$ git add <files> && git commit -m"<commit message>"
# start working on something new?
$ git checkout -b new-idea
$ git branch
new-idea
# work in that branch for a while
$ git add <files> && git commit -m"<commit message>"
# back to master
$ git checkout master
#...
# rebase master on new-idea: new-idea is now in master
$ git rebase new-idea
```

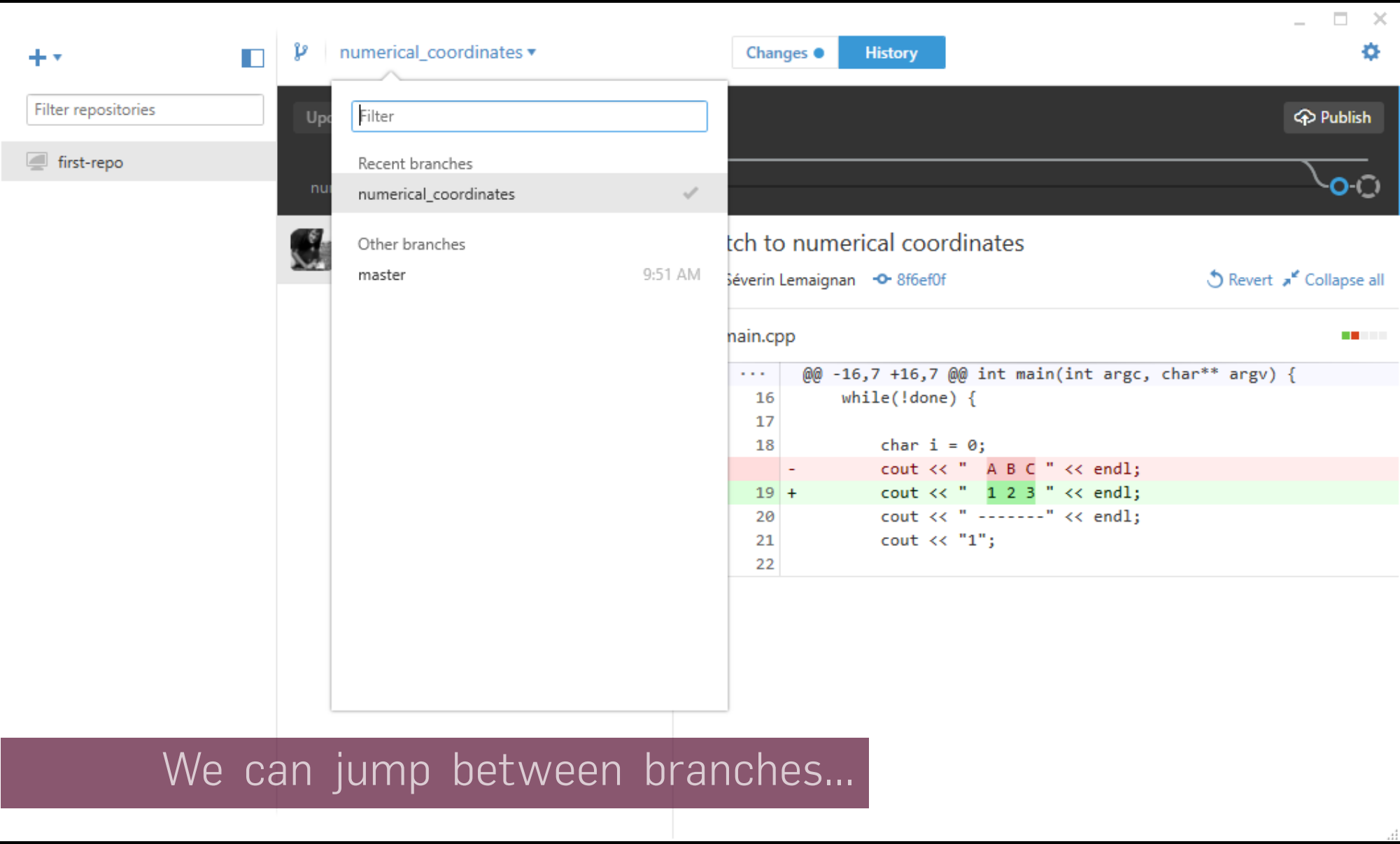
Viewed from a GUI...



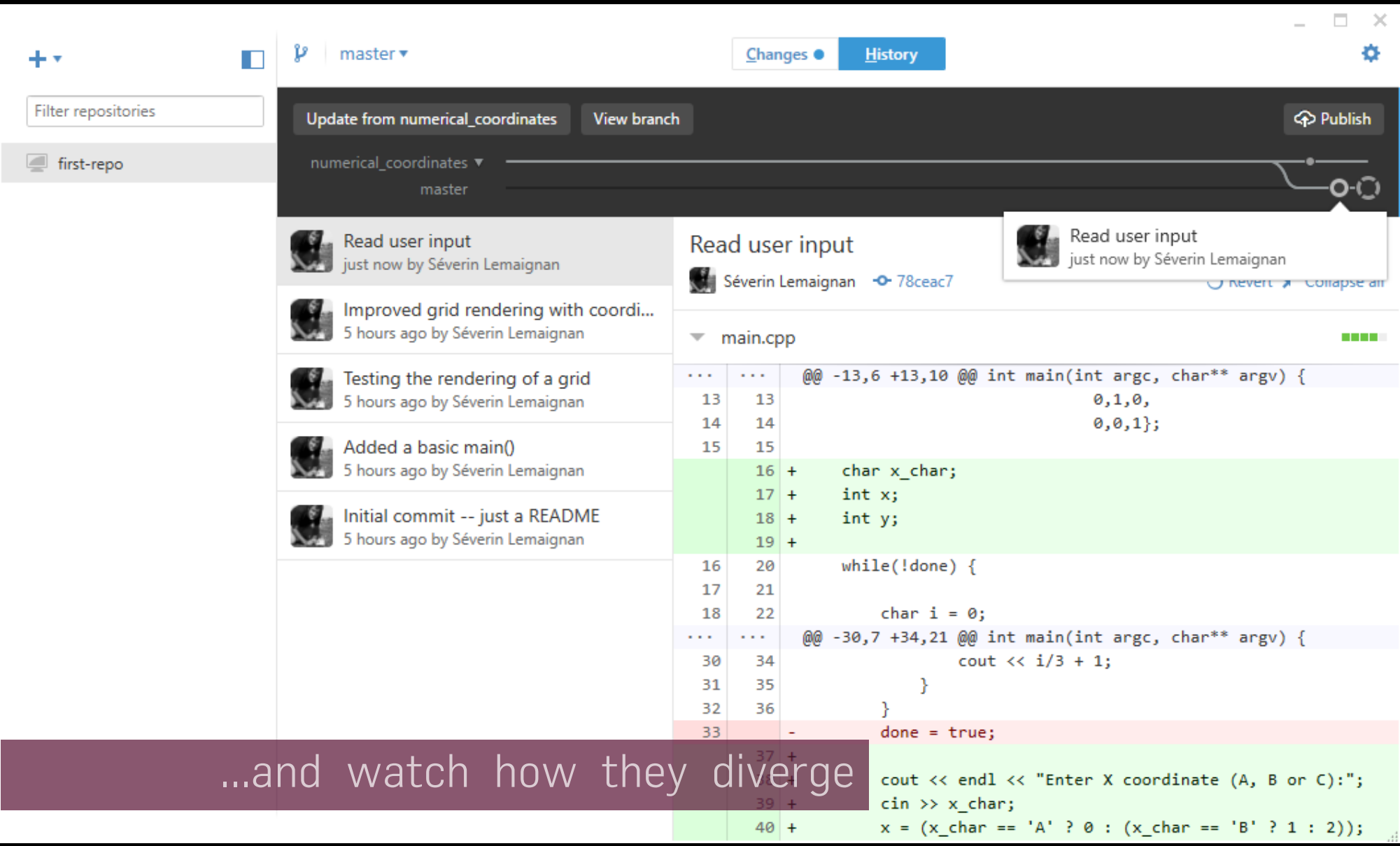
We can easily create a new branch



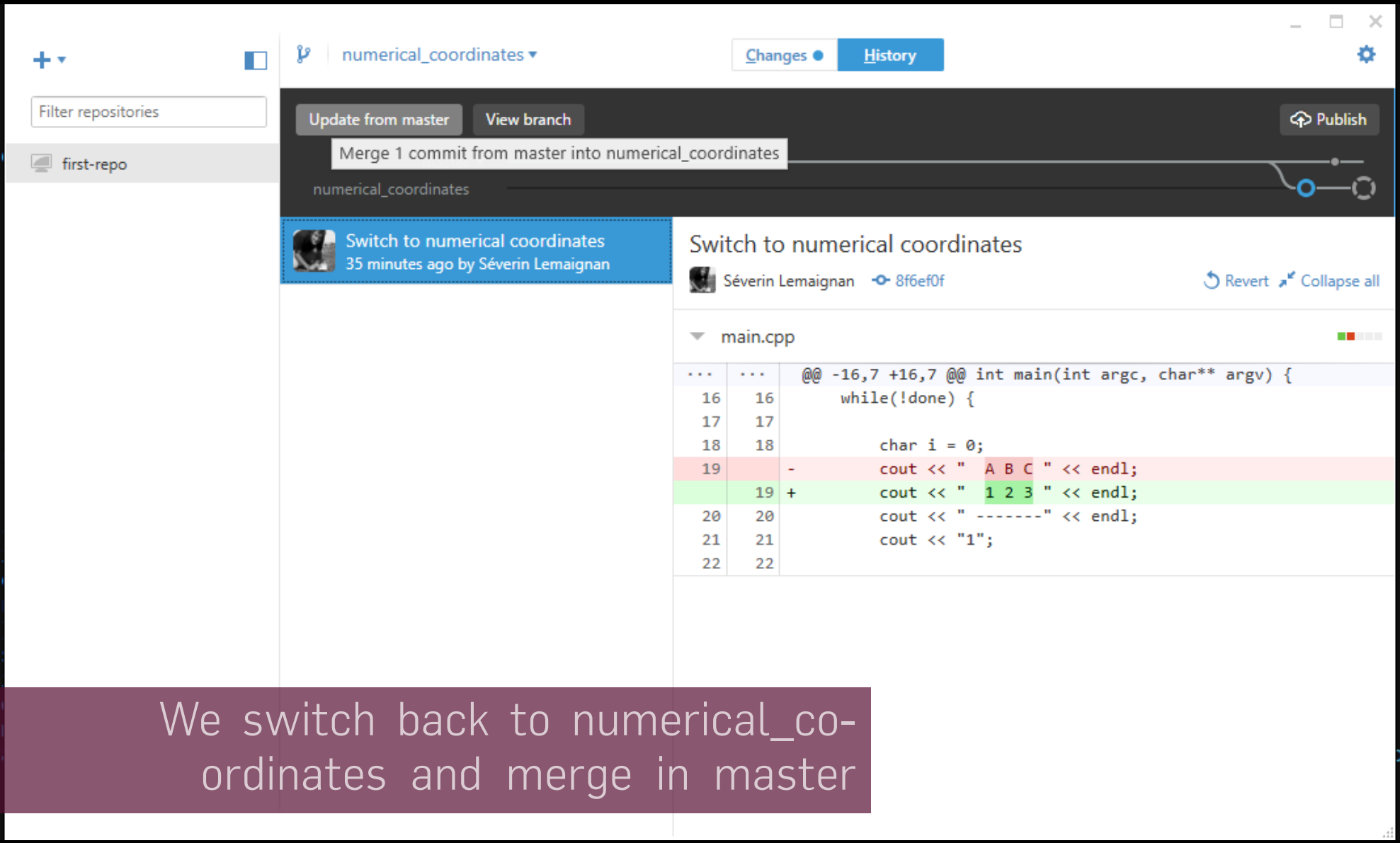
We can compare numerical_coordinates with master (click on View branch for the full history)



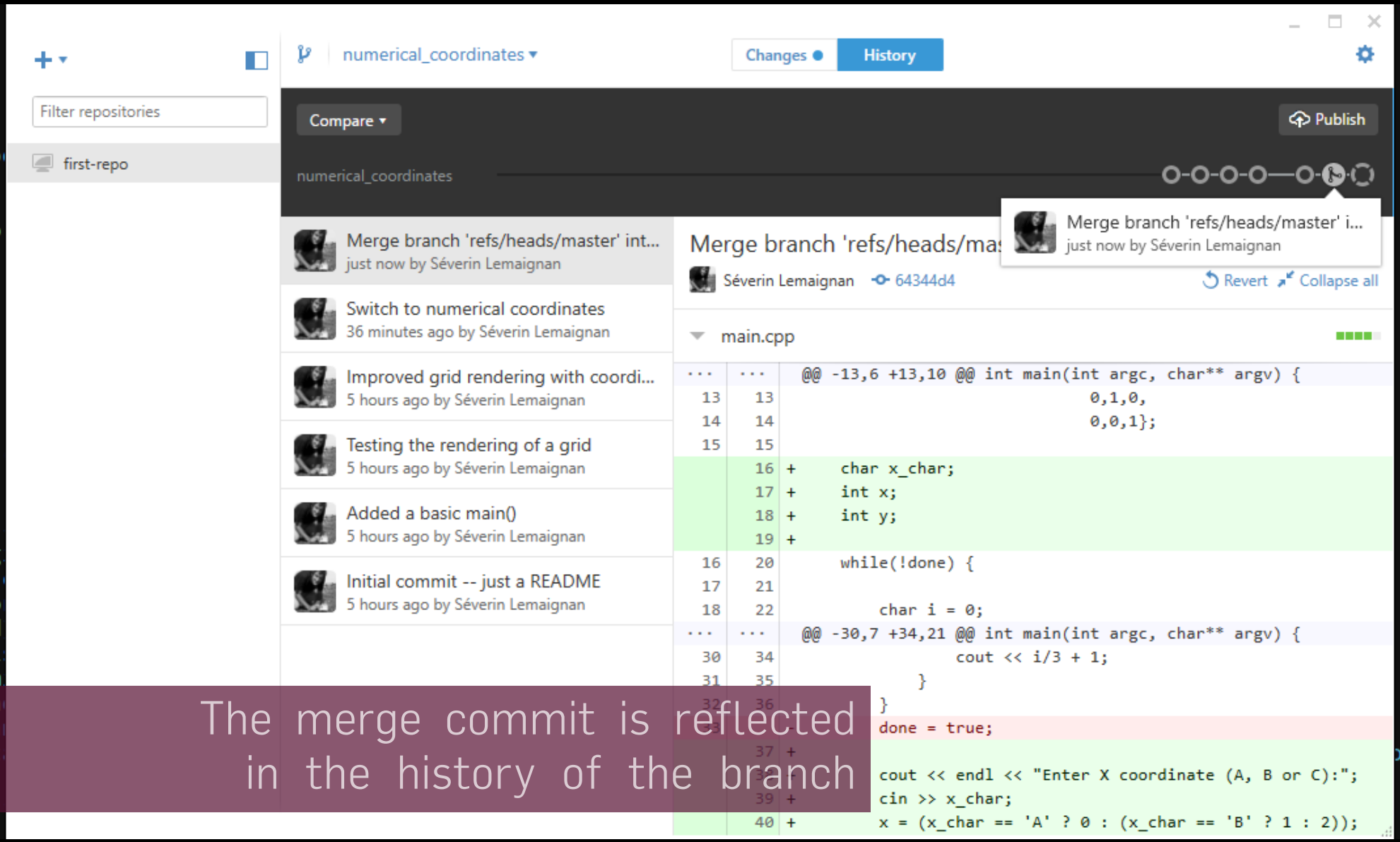
We can jump between branches...



...and watch how they diverge



We switch back to numerical_coordinates and merge in master



The merge commit is reflected in the history of the branch

ETIQUETTE OF SOCIAL CODING 101

principle of least surprise

Make people feel at home when they interact with your project!

one repo = one thing

make plenty of repos!

REPOSITORY LAYOUT

Try to follow as much as possible the **Filesystem Hierarchy Standard** (FHS). Mainly:

```
src/          # source
include/      # *public* headers
etc/          # configuration files
share/        # data
doc/          # documentation
README
LICENSE
```

NO build artifacts!!

no binaries (except possibly in `share/`)

REPOSITORY LAYOUT

Try to follow as much as possible the **Filesystem Hierarchy Standard** (FHS). Mainly:

```
src/          # source
include/      # *public* headers
etc/          # configuration files
share/        # data
doc/          # documentation
README
LICENSE
```

`README` (or better, use markdown: `README.md`): what is the project about? who is the target audience? how to install? how to get started?

LICENSE

- **no license** ⇒ default copyright laws apply. You (or probably UoP) retain all rights to your source code; nobody else may reproduce, distribute, or create derivative works from your work.
- **Permissive licenses**: others do essentially whatever they want with your code, as long as they give your attribution. Examples: MIT, BSD
- **Copyleft licenses**: Derivative work must be made available under the same terms as the original work (***viral licenses***). Example: GPL

You always keep the author rights!

⇒ you can change the license at any time.

LICENSE

- **no license** ⇒ default copyright laws apply. You (or probably UoP) retain all rights to your source code; nobody else may reproduce, distribute, or create derivative works from your work.
- **Permissive licenses**: others do essentially whatever they want with your code, as long as they give your attribution. Examples: MIT, BSD
- **Copyleft licenses**: Derivative work must be made available under the same terms as the original work (***viral licenses***). Example: GPL

Check **http://choosealicense.com/**
and discuss that with your supervisor

BUILD SYSTEM

Use and provide a build system!

- Windows-only $\Rightarrow$ a Visual Studio solution is ok
- MacOS-only $\Rightarrow$ a XCode project is ok

In all other cases, go for a cross-platform build system like **CMake**.

COMMIT HYGIENE

“Show me the project history, I’ll tell you what coder you are”

- **Commit often!** Push when needed (or at the end of day)

Because commits are local (ie, private), **do commit often**: **mistakes are ok** as you can fix them before sharing with others.

COMMIT HYGIENE

“Show me the project history, I’ll tell you what coder you are”

- Write useful messages (no “`Fixed bug`” or “`New file`”)
- First line of commit messages < 72 characters

COMMIT HYGIENE

“Show me the project history, I’ll tell you what coder you are”

- Tag important commits!

Notably, GitHub (amongst others) interpret tags as **releases** of your code.

A FEW COOL GITHUB STUFF TO FINISH

Besides bugtracking, project homepages and wikis,GitHub integrates with many third-party services & tools:

- **Travis CI** or **AppVeyor** for continuous integration

[sensors] Added an 'encoders' level to the velocity sensor #541

Edit

Open **severin-lemaignan** wants to merge 1 commit into `morse-simulator:master` from `severin-lemaignan:encoders`

Conversation 4 | Commits 1 | Files changed 3 | +134 −3

severin-lemaignan commented on 29 May 2014 — The Modular OpenRobots Simulation Engine member

This new abstraction level for the velocity sensor that returns encoder ticks instead of linear/angluar speeds

[...]

Add more commits by pushing to the **encoders** branch on **severin-lemaignan/morse**.

All checks have failed
1 errored check — Hide all checks

✕ **continuous-integration/travis-ci** — The Travis CI build could not complete du… Details

This branch has conflicts that must be resolved
Use the command line to resolve conflicts before continuing.

Merge pull request or view command line instructions.

Labels
None yet

Milestone

Notifications
Unsubscribe
You're receiving notifications because you authored the thread.

2 participants

Lock conversation

A FEW COOL STUFF TO FINISH

+ GitHub integrates with many external services & tools:

  - **Travis CI** or **AppVeyor** for continuous integration
  - **zenodo**: associate a DOI to your repository
  - **ReadTheDocs**: generate and publish on-line documentation