

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**

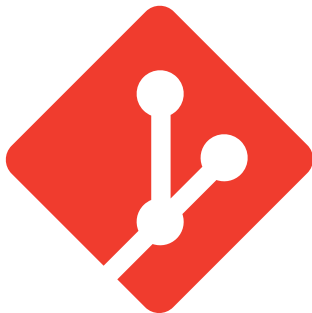
git

the basics

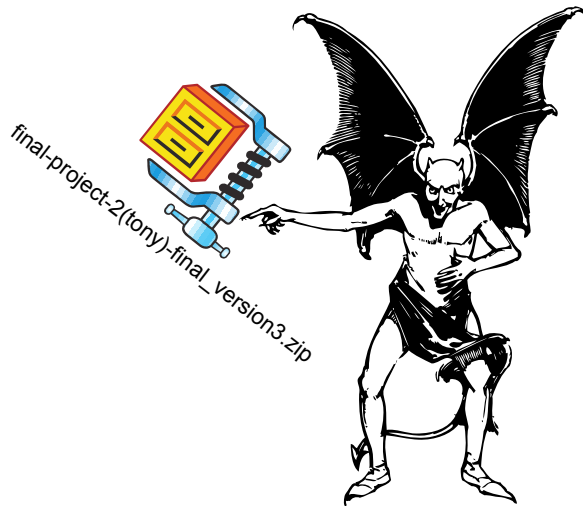
24 Jan. 2017

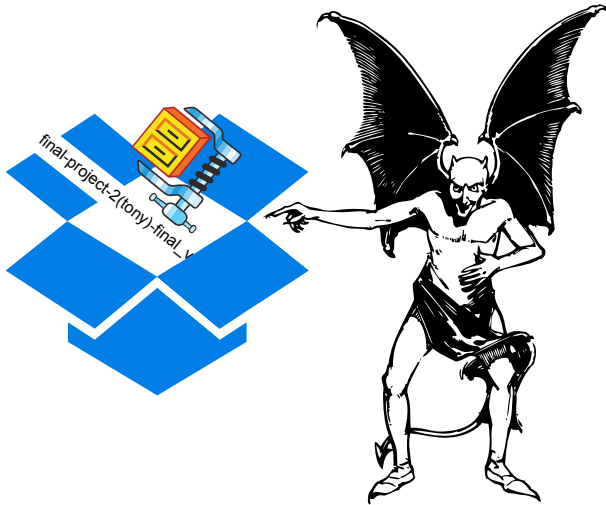
Séverin Lemaignan

Centre for Robotics & Neural Systems
Plymouth University



FIRST THINGS FIRST: WHY BOTHER?







Get Started

Wall

Hidden Posts

- Info
- Listings
- Photos
- Dan's Welcome Page
- Discussions
- Edit

About

7
people like this

Add to My Page's Favorites

Tony King B.

Real Estate · Toronto, Ontario

Edit Page

Wall

Tony King B. · Most Recent

Shares: 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.

SVH 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.



Tony King B.

Real Estate · Toronto

Edit Page

Edit Page

Most Recent

Admins (4)

See All

Get Started

Wall

Hidden Posts

Info

Listings

Photos

Dan's Welcome Page

Discussions

Edit

About

Edit



Tony King B.

SVH is really cool, but I like Facebook!

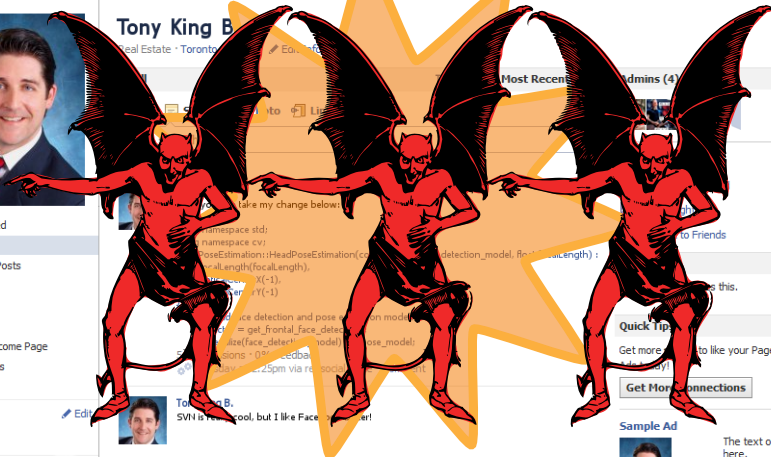
51 Impressions · 0% Feedback

Tuesday at 2:25pm via re2social · Like · Comment

7

people like this

Add to My Page's Favorites



Quick Tip

Get more likes to like your Page with Facebook

Get More Connections

Sample Ad



The text of your ad will go here.

Like · JP Zeni likes this.

WHY CODE 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!

CODE VERSIONING

ATOMIC COMMITS

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

Atomic commit

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

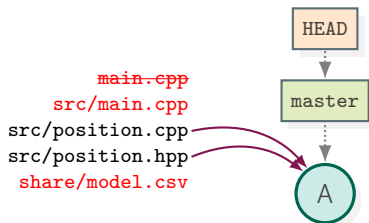
ATOMIC COMMITS

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

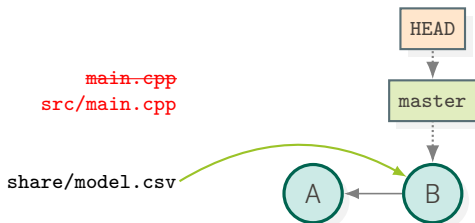
Atomic commit

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

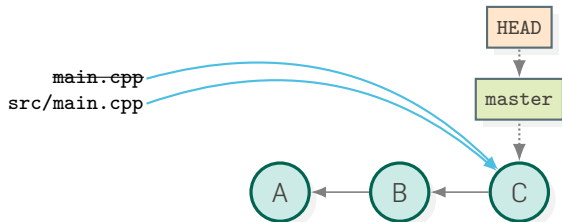
```
main.cpp  
src/main.cpp  
src/position.cpp  
src/position.hpp  
share/model.csv
```



```
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 add src/main.*  
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
```

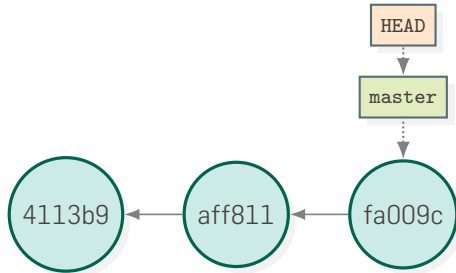
```
    Re-trained model with 52 more participants
```

```
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)
```

THE STAGING AREA

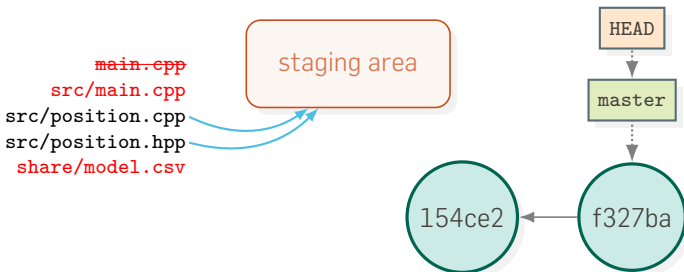
But why do we have to manually tell Git what files to add or remove?

THE STAGING AREA

No “commit all changes” by default (well, you can, actually...)
⇒ Help thinking in terms of atomic commits!

THE STAGING AREA

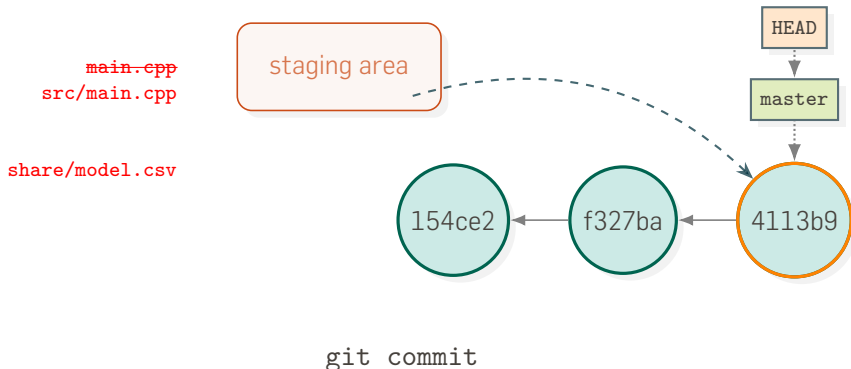
Preparing a commit consists in filling the **staging area** (or **index**) with the list of changes:



```
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:



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/>



Filter repositories

Tutorial



Welcome

Log in

Configure

Repositories

GitHub GitHub Enterprise

The best way to build and ship software. [Go to github.com](https://github.com) to sign up for an account



Log in



Skip setup

Log in to your GitHub account

Add

Create

Clone

Name

first-repo

Local path

C:\Users\Severin Lemaignan\Documents\GitHub\first-repo

Browse

Git ignore

Windows

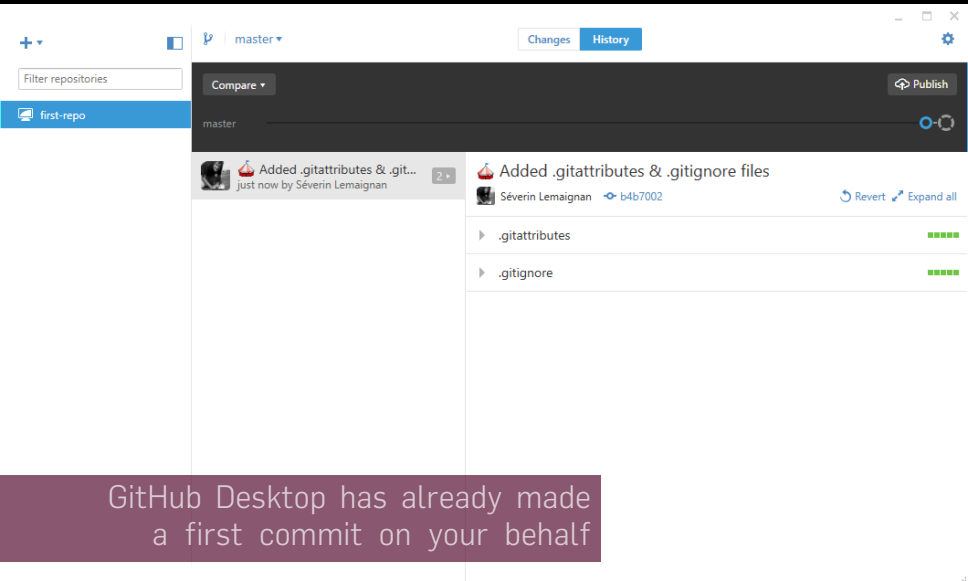
✓

Create repository



Get started by adding a repository.

Create a (local) repository



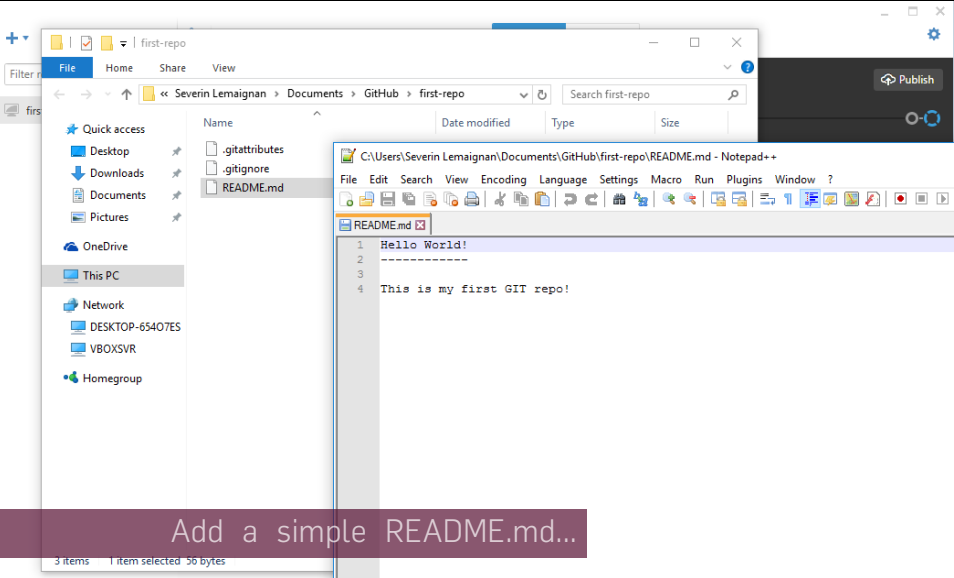
Visual Studio interface showing a Git repository named "first-repo". The "master" branch is selected. The "Changes" tab is active, displaying a commit history. A context menu is open over the repository name, offering options: "View on GitHub", "Open in Explorer", "Open in Git Shell", and "Remove".

The commit history shows a commit by Séverin Lemaignan (b4b7002) titled "Added .gitattributes & .gitignore files". The commit message is "Added .gitattributes & .gitignore files". The commit is dated "1 minute ago".

The commit details show two files added:

- .gitattributes
- .gitignore

A purple banner at the bottom of the image reads: "Open the repo in Windows Explorer".



+

Filter repositories

first-repo

master

Compare

View 1 uncommitted change

Publish

master

1 change

✓ README.md

Summary

Description

The change is listed in the Changes panel

+

Filter repositories

first-repo

master

Changes

History

Compare

Publish

master

1 change

✓

README.md

■■■■■

Added a README

Description

Commit to master

Write a commit message & commit!

+

Filter repositories

first-repo

master

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 @@

1 + Hello World!

2 + -----

3 +

4 + This is my first GIT repo!

1 5 \ 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



normal



assume-valid



added



normal.cpp



assume-valid.cpp



added.cpp



modified



deleted



ignored



modified.cpp



deleted.cpp



ignored.cpp



conflicted



skip-worktree



non-versioned



conflicted.cpp

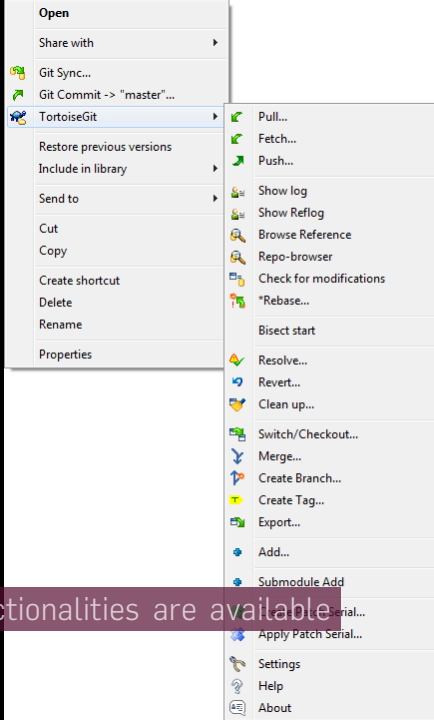


skip-worktree.cpp

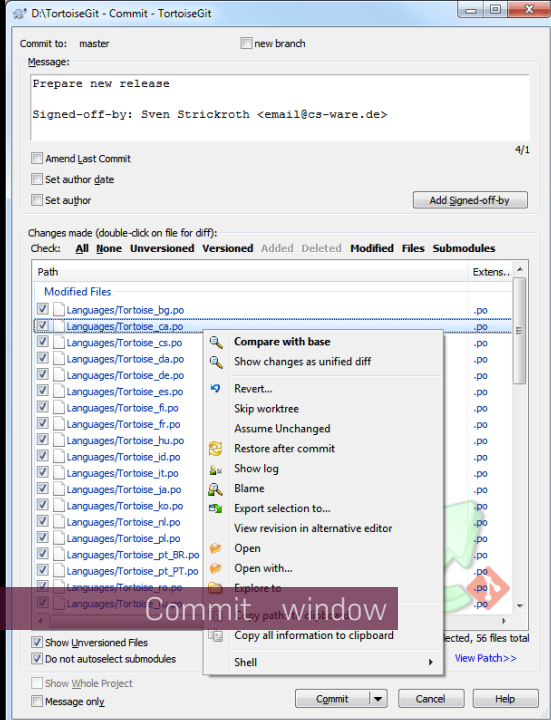


non-versioned.cpp

Files' status appear as icons



All the functionalities are available

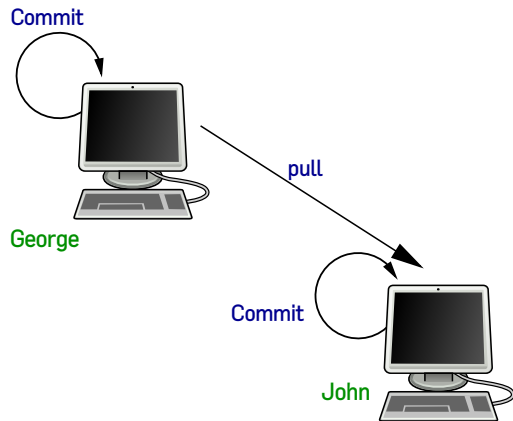


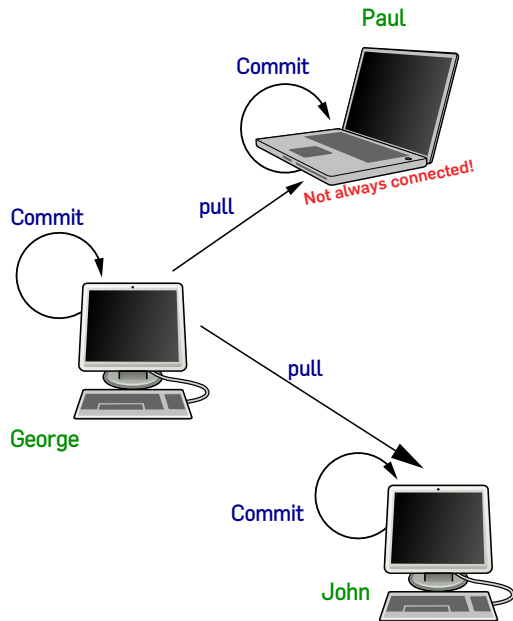
COLLABORATING

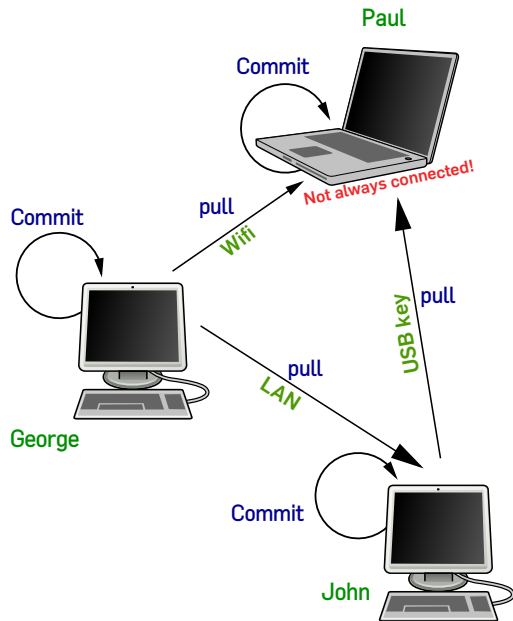
Commit

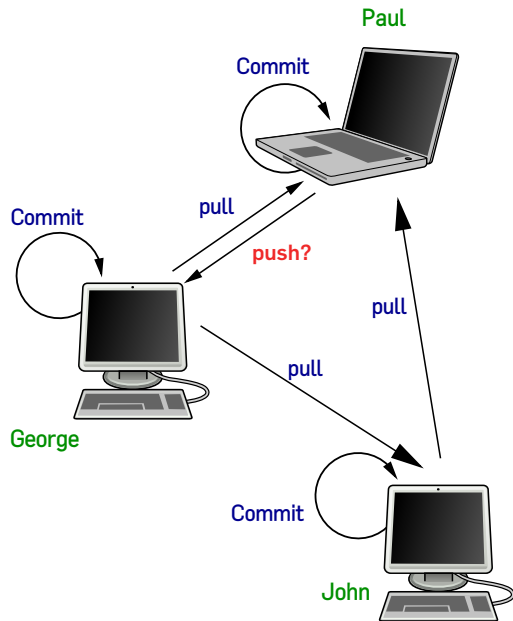


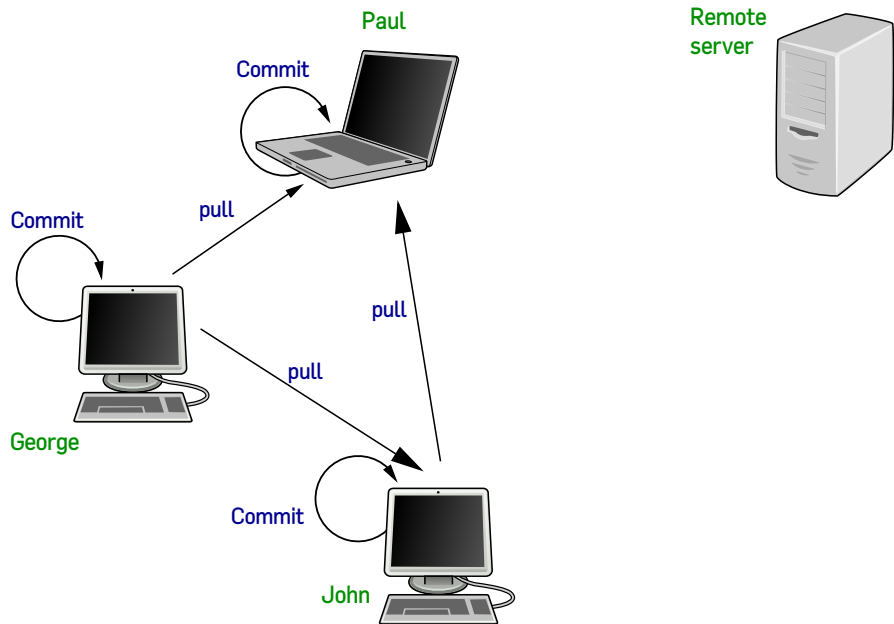
George



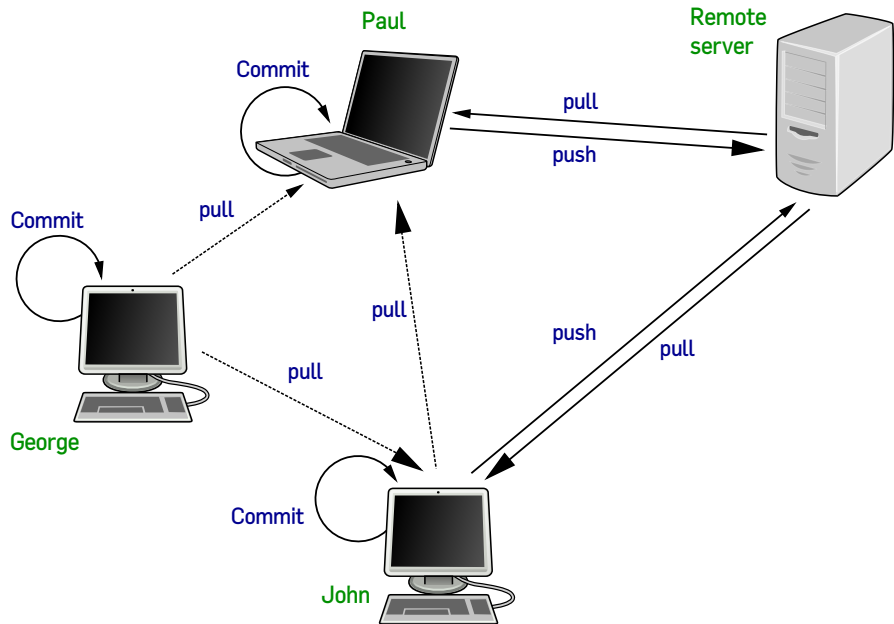




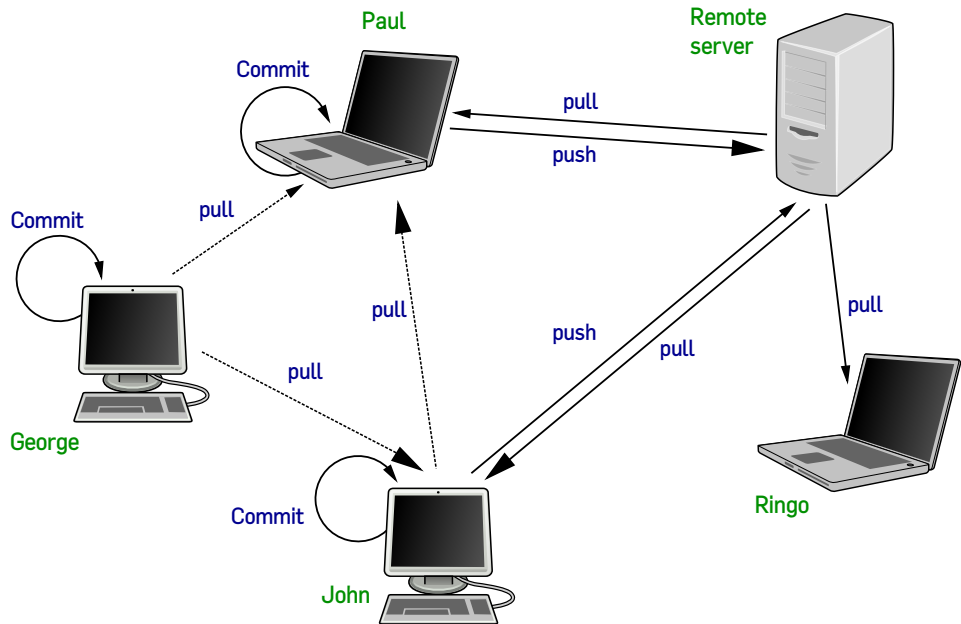




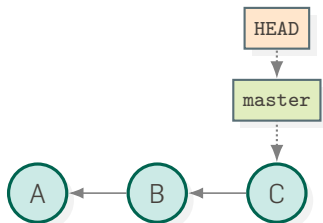
Based on a figure by M. Herrb, CC-BY-SA 3.0

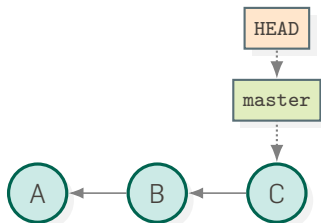


Based on a figure by M. Herrb, CC-BY-SA 3.0

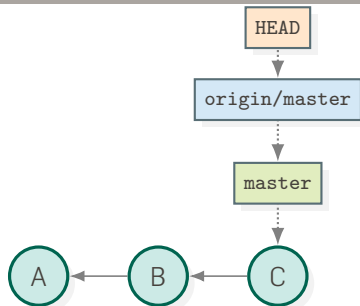


Based on a figure by M. Herrb, CC-BY-SA 3.0

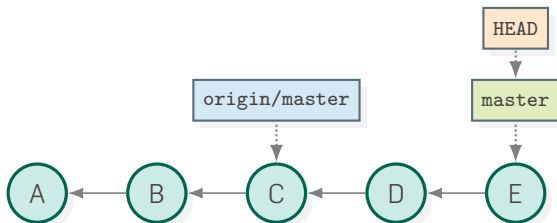


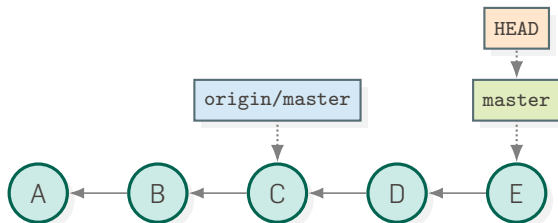


```
git remote add origin git@github.com:user/repo.git
git remote add john-usb E:\john_repo
git remote add ftp-origin ftp://host.xz/path/to/repo.git/
...
```



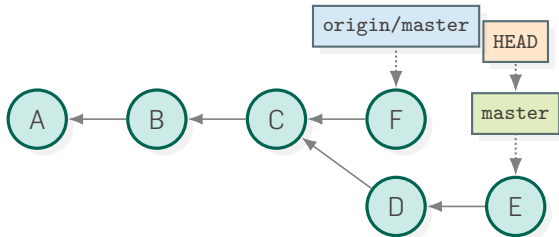
```
git push origin master  
(or simply git push)
```

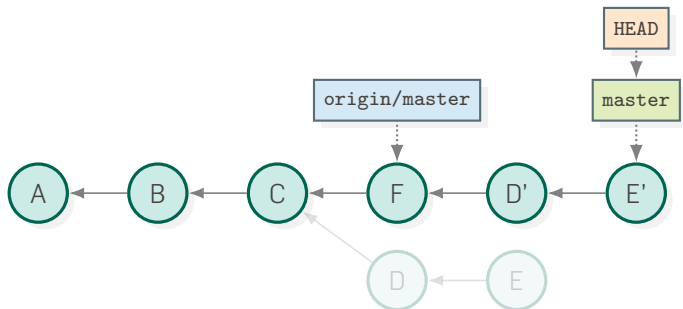





What happened on our remote? Let's have a look...

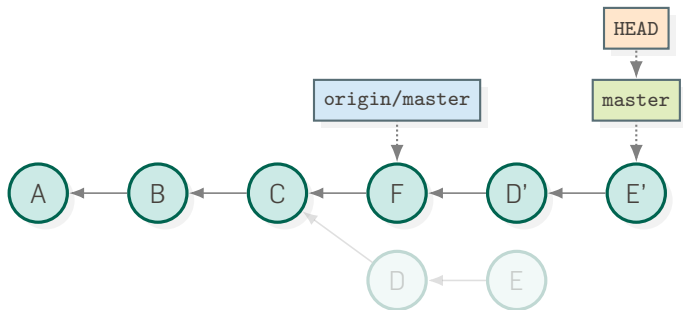
```
git fetch origin
```



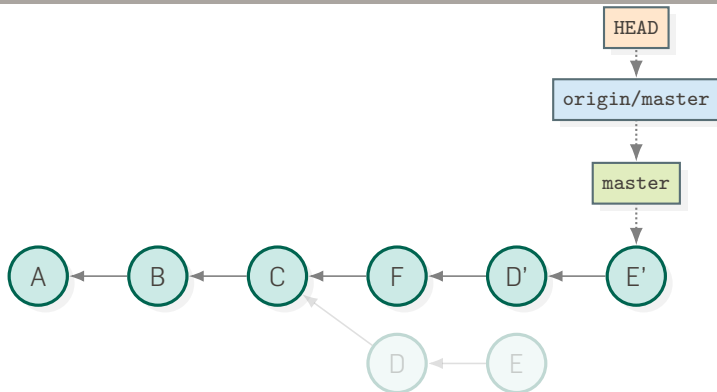


```
git rebase origin/master
```

(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-ap
```

When you have resolved this problem, run `"git rebase --continue"`

If you prefer to skip this patch, run `"git rebase --skip"` instead.

To check out the original branch and stop rebasing, run `"git checkout john"`

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



SOCIAL CODING: GITHUB WORKFLOW



This repository Search

Pull requests Issues Gist



morse-simulator / morse

Unwatch 28

Unstar 145

Fork 74

Code

Issues 69

Pull requests 2

Wiki

Pulse

Graphs

Settings

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.mini	[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
- Downloads

1

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_license.html

licenses for dependencies are included in the licenses folder.

Instructions

BitBucket

<<

GitLab

Back to Group

Project

Activity

Files

Commits

Network

Graphs

Milestones

Issues712

Merge Requests52

Labels

gitlab.com


GitLab.org / GitLab Community Edition

Search in this project

Download zip

mastergitlab-ce

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 Spelcher 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 Trzcinski Added missing packages required by docker builds	
vendor/assets	9 days ago	Robert Spelcher Merge branch 'rs-security-spec-speed' into 'master'	
	about a year ago	Robert Spelcher Make sure important directories exist in git	
		Dmitriy Zaporozhets Add nice scroll for sidebar	

darby

GitLab – open-source You can install it on your own server



This repository Search

Pull requests Issues Gist



morse-simulator / morse

Unwatch 28

Unstar 145

Fork 74

Code

Issues 69

Pull requests 2

Wiki

Pulse

Graphs

Settings

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

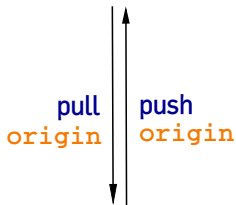
<https://github.com/morse>



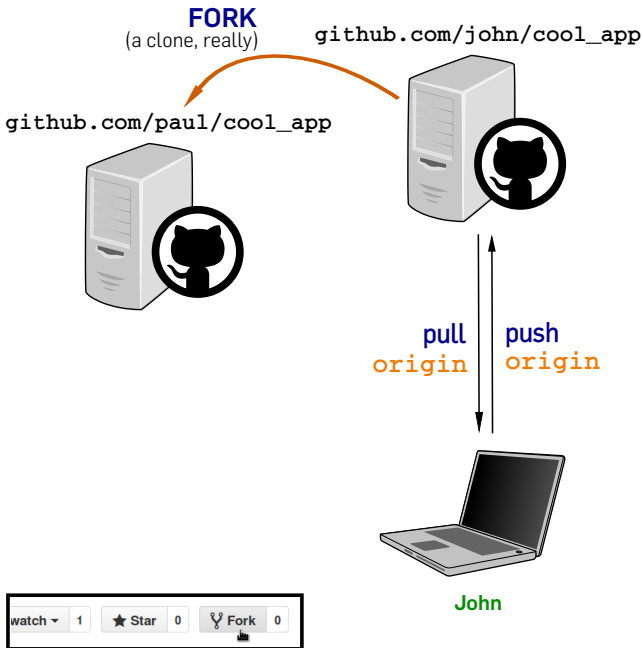
Download ZIP

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

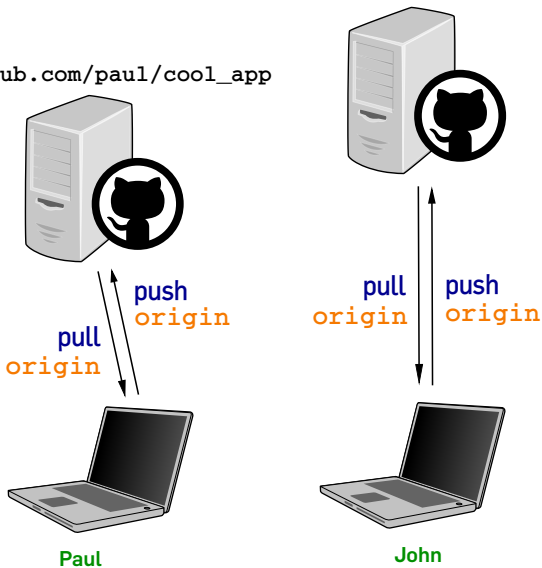


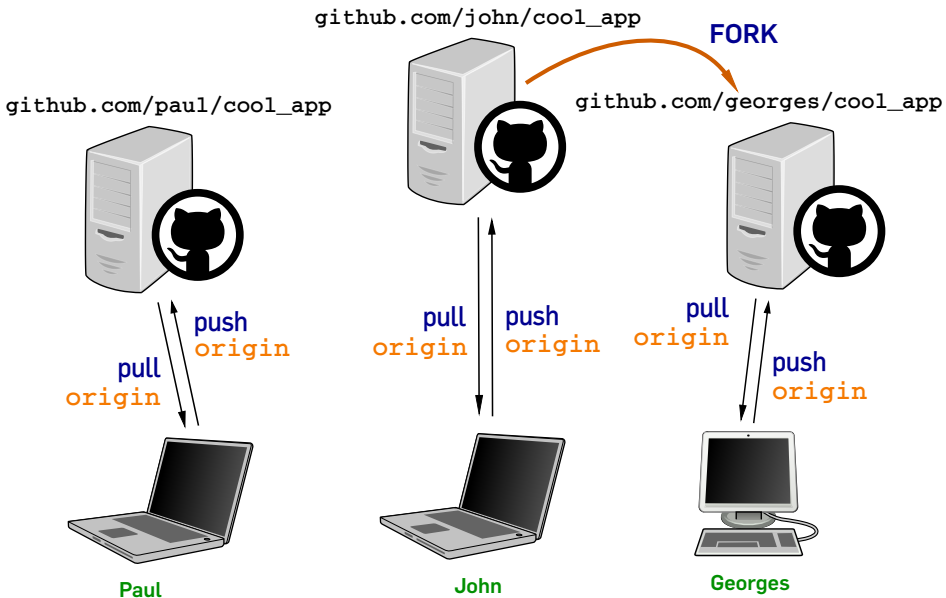
John



github.com/john/cool_app

github.com/paul/cool_app

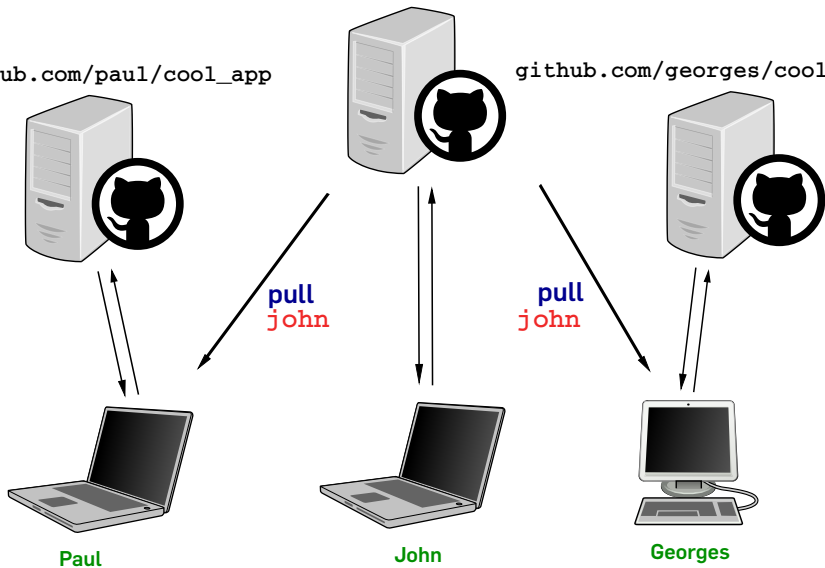


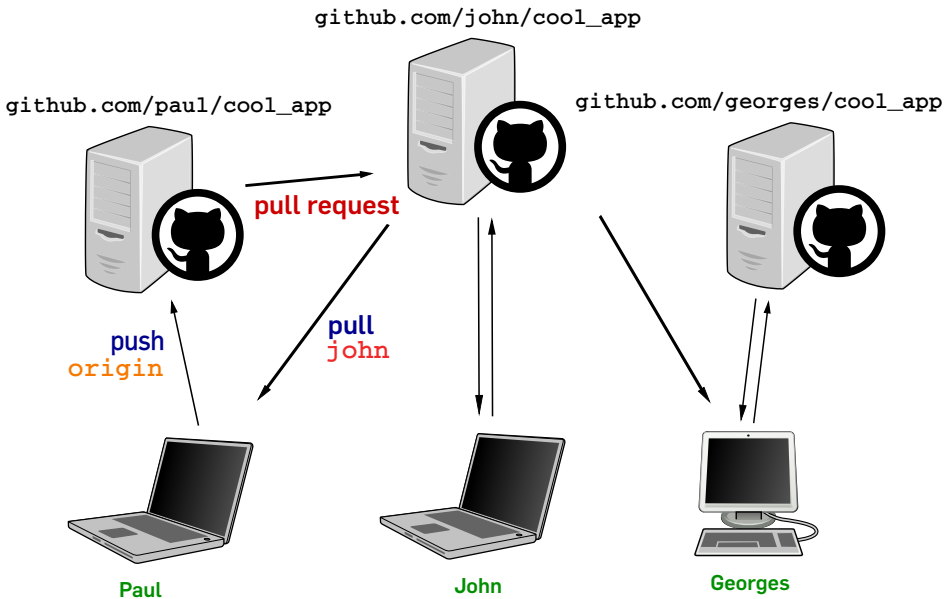


github.com/john/cool_app

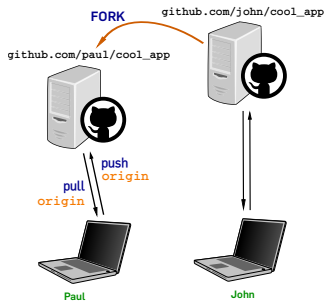
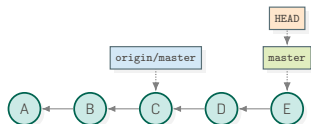
github.com/paul/cool_app

github.com/georges/cool_app



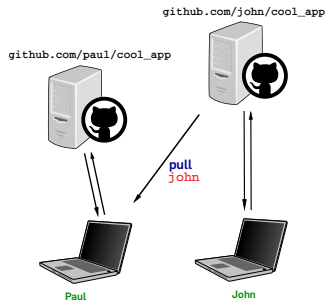
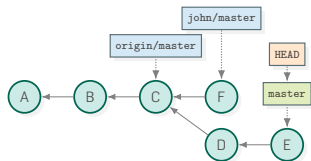


WHAT HAPPENED EXACTLY?



After forking on GitHub, Paul runs
`git clone https://github.com/paul/cool_app.git`
and he adds few local commits

WHAT HAPPENED EXACTLY?

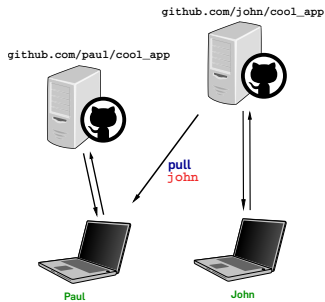
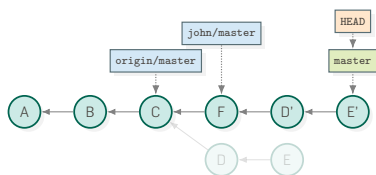


He would like to propose his changes to John

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?

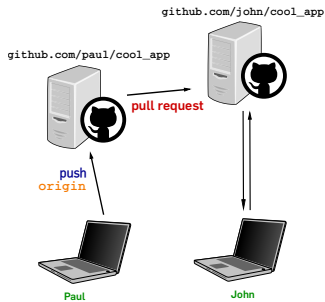
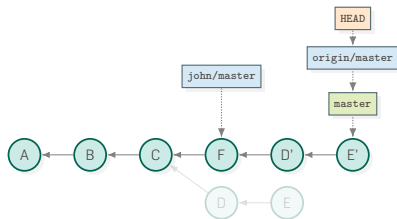


Paul rebases his `master` branch on John's one:

```
git rebase john/master
```

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

WHAT HAPPENED EXACTLY?

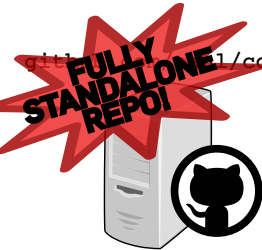


He pushes his commits to his own GitHub account:

```
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!)



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` \equiv `git pull origin master`)

Share stuff on a remote:

`git push <remote> <branch>`

(`git push alone` \equiv `git push origin master`)

Repo state

`git status`

Repo history

`git log`

Who did what?

`git blame`

I've lost everything!

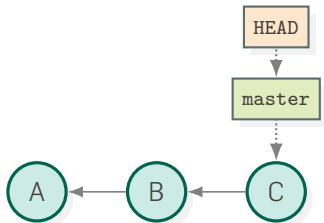
`git reflog`



That's all folks! The slides are
on-line: <https://github.com/severin-lemaignan/git-presentation>

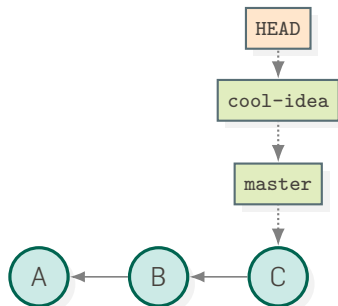
WORKING WITH BRANCHES

BRANCHES



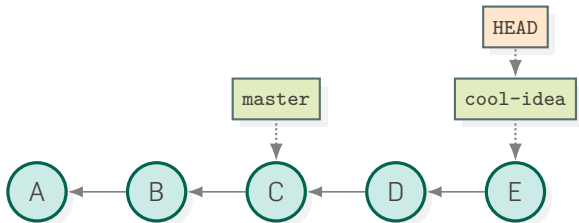
What if...?

BRANCHES

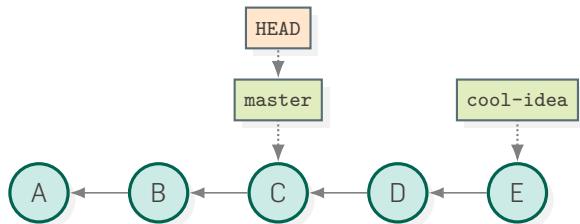


```
git checkout -b cool-idea
```

BRANCHES

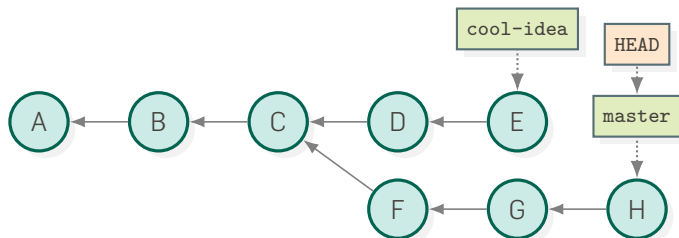


BRANCHES



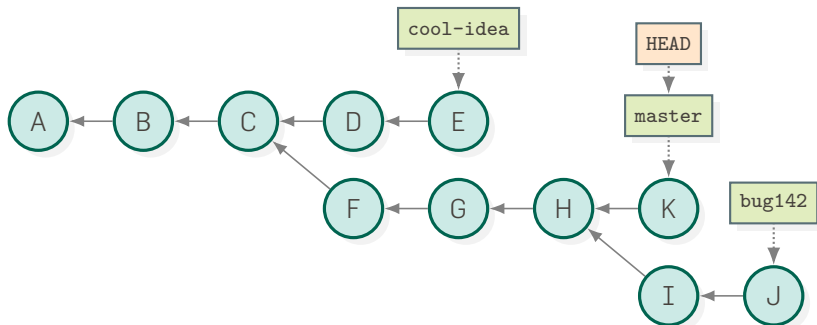
Let go back to serious stuff!
`git checkout master`

BRANCHES



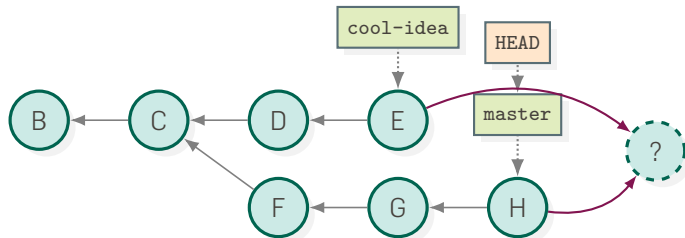
The branch name is an alias for the tip of the current branch

BRANCHES



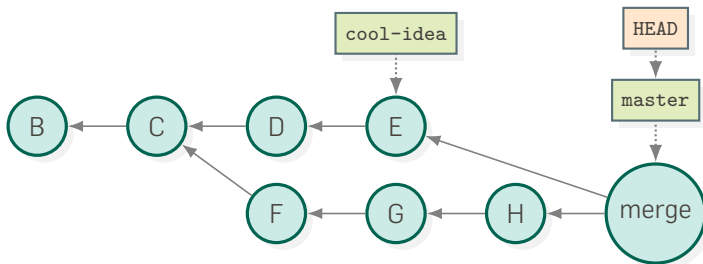
⇒ branches are very cheap
+10 of them at a given time it not uncommon

MERGING BRANCHES



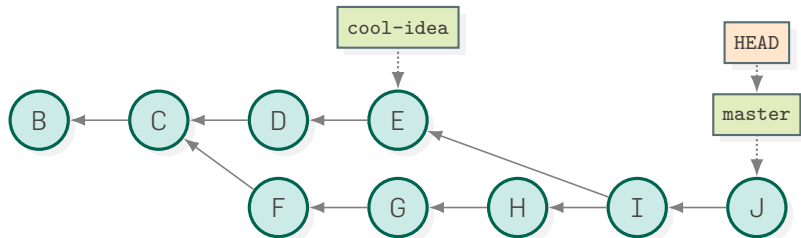
Two options: **merging** and **rebasing**

MERGING BRANCHES



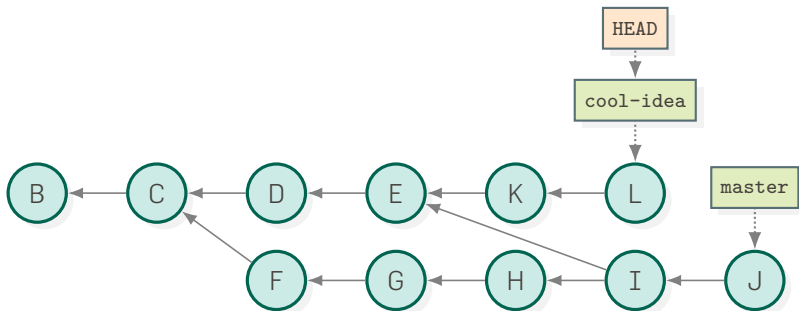
Merging
`git merge cool-idea`

MERGING BRANCHES



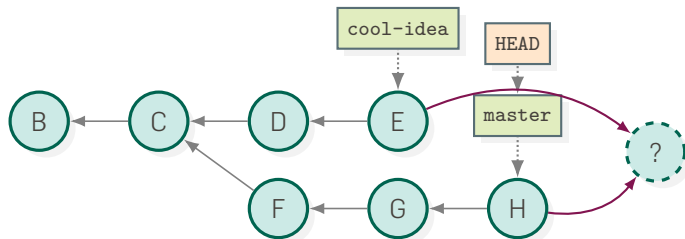
```
git commit
```

MERGING BRANCHES

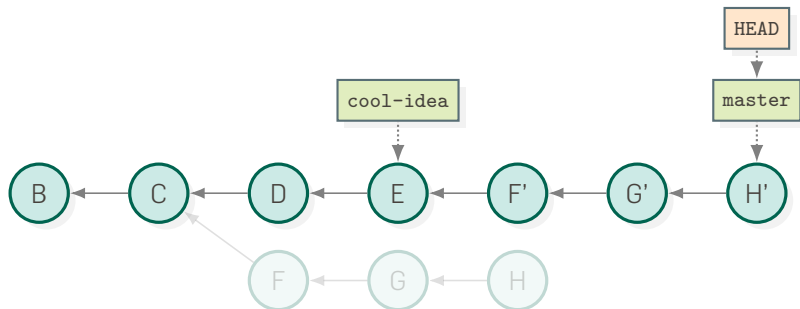


```
git checkout cool-idea
git commit
...etc.
```

REBASING BRANCHES

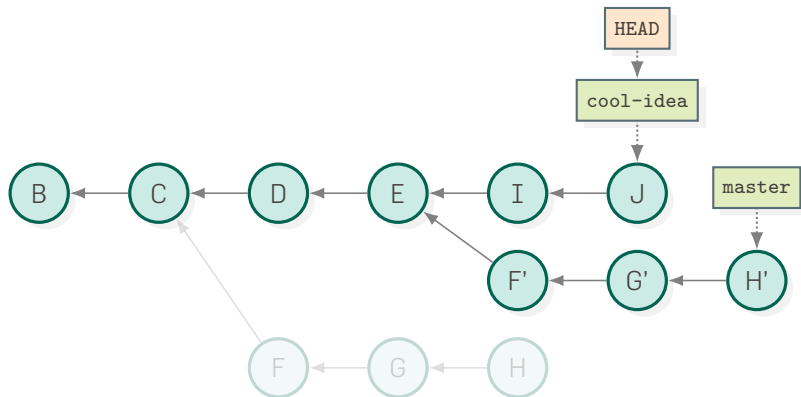


REBASING BRANCHES



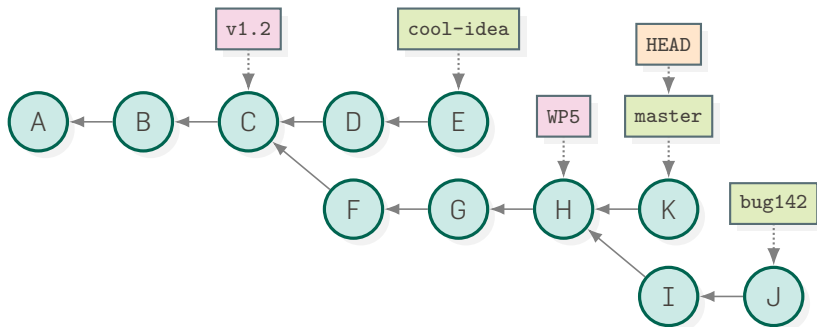
Rebasing
`git rebase cool-idea`

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...

Filter repositories

first-repo

master

numerical_coordinates

From branch master

Create new branch

Testing the rendering of a grid
4 hours ago by Séverin Lemaignan

Added a basic main()
5 hours ago by Séverin Lemaignan

Initial commit -- just a README
5 hours ago by Séverin Lemaignan

Changes

History

Publish

oved grid rendering with coordinates

Séverin Lemaignan da25158

Revert Collapse all

main.cpp

```
... @@ -16,7 +16,9 @@ int main(int argc, char** argv) {
16 16     while(!done) {
17 17
18 18         char i = 0;
19 - cout << "-----" << endl;
19 + cout << " A B C " << endl;
20 + cout << " -----" << endl;
21 + cout << "1";
20 22
21 23     for(auto pos : positions) {
22 24         i++;
... @@ -24,7 +26,8 @@ int main(int argc, char** argv) {
24 26
25 27         if (i % 3 == 0) {
26 28             cout << "|" << endl;
27 - cout << "-----" << endl;
27 + cout << " -----" << endl;
28 + cout << i/3 + 1;
28 31
29 32     }
```

We can easily create a new branch

+

Filter repositories

first-repo

numerical_coordinates

Changes

History

Update from master

View branch

Publish

master

numerical_coordinates

Switch to numerical coordinates
just now by Séverin Lemaignan

Switch to numerical coordinates

Séverin Lemaignan 8f6ef0f

Revert Collapse all

main.cpp

...	...	@@ -16,7 +16,7 @@ int main(int argc, char** argv) {
16	16	while(!done) {
17	17	
18	18	char i = 0;
19	-	cout << " A B C " << endl;
	19 +	cout << " 1 2 3 " << endl;
20	20	cout << " -----" << endl;
21	21	cout << "1";
22	22	

We can compare numerical_co-ordinates with master (click on View branch for the full history)

Visual Studio Code interface showing a repository named `numerical_coordinates`. The interface includes a sidebar with a "Filter" input, a list of repositories (including `first-repo`), and a dropdown menu for branches. The dropdown menu shows "Recent branches" with `numerical_coordinates` selected and `master` listed below. The main editor displays a C++ file named `main.cpp` with a `while` loop. The code snippet shows a change on line 19, where the output string is updated from `" A B C "` to `" 1 2 3 "`. The interface also features a "Changes" tab, a "History" tab, and a "Publish" button.

We can jump between branches...

+

Filter repositories

first-repo

master

ChangesHistory

Update from numerical_coordinatesView branchPublish

numerical_coordinates

master

Read user input

just now by Séverin Lemaignan

Improved grid rendering with coordi...

5 hours ago by Séverin Lemaignan

Testing the rendering of a grid

5 hours ago by Séverin Lemaignan

Added a basic main()

5 hours ago by Séverin Lemaignan

Initial commit -- just a README

5 hours ago by Séverin Lemaignan

Read user input

just now by Séverin Lemaignan

Revert

Commit

main.cpp

...13141516171819202122232425262728293031323334353637383940

@@ -13,6 +13,10 @@ int main(int argc, char** argv) {
0,1,0,
0,0,1};
+ char x_char;
+ int x;
+ int y;
+
while(!done) {
char i = 0;
@@ -30,7 +34,21 @@ int main(int argc, char** argv) {
cout << i/3 + 1;
}
}
- done = true;
+
+ cout << endl << "Enter X coordinate (A, B or C):";
+ cin >> x_char;
+ x = (x_char == 'A' ? 0 : (x_char == 'B' ? 1 : 2));

...and watch how they diverge



Filter repositories



first-repo



numerical_coordinates ▾

Changes ●

History



Update from master

View branch

Publish

Merge 1 commit from master into numerical_coordinates

numerical_coordinates



Switch to numerical_coordinates
35 minutes ago by Séverin Lemaignan

Switch to numerical_coordinates



Séverin Lemaignan 8f6ef0f

Revert Collapse all

main.cpp

```
...  ... @@ -16,7 +16,7 @@ int main(int argc, char** argv) {  
16 16     while(!done) {  
17 17  
18 18         char i = 0;  
19 -     cout << " A B C " << endl;  
19 +     cout << " 1 2 3 " << endl;  
20     cout << " -----" << endl;  
21     cout << "1";  
22 22 }
```

We switch back to numerical_co-
ordinates and merge in master



numerical_coordinates ▾

Changes ●

History



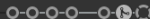
Filter repositories

Compare ▾

Publish

first-repo

numerical_coordinates



Merge branch 'refs/heads/master' into 'refs/heads/master' just now by Séverin Lemaignan



Switch to numerical_coordinates 36 minutes ago by Séverin Lemaignan



Improved grid rendering with coordinates 5 hours ago by Séverin Lemaignan



Testing the rendering of a grid 5 hours ago by Séverin Lemaignan



Added a basic main() 5 hours ago by Séverin Lemaignan



Initial commit -- just a README 5 hours ago by Séverin Lemaignan

Merge branch 'refs/heads/master' into 'refs/heads/master'



Merge branch 'refs/heads/master' into 'refs/heads/master' just now by Séverin Lemaignan



Séverin Lemaignan 64344d4

Revert Collapse all

main.cpp



```
...  ...  @@ -13,6 +13,10 @@ int main(int argc, char** argv) {
13  13      0,1,0,
14  14      0,0,1};
15  15
16  +   char x_char;
17  +   int x;
18  +   int y;
19  +
16  20   while(!done) {
17  21
18  22       char i = 0;
...  ...  @@ -30,7 +34,21 @@ int main(int argc, char** argv) {
30  34       cout << i/3 + 1;
31  35   }
32  36   }
33  +   done = true;
34  +
35  +   cout << endl << "Enter X coordinate (A, B or C):";
36  +   cin >> x_char;
37  +   x = (x_char == 'A' ? 0 : (x_char == 'B' ? 1 : 2));
40  +
```

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/	# <i>source</i>
include/	# <i>*public* headers</i>
etc/	# <i>configuration files</i>
share/	# <i>data</i>
doc/	# <i>documentation</i>
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/	<i># source</i>
include/	<i># *public* headers</i>
etc/	<i># configuration files</i>
share/	<i># data</i>
doc/	<i># documentation</i>
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** \Rightarrow 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!

\Rightarrow you can change the license at any time.

LICENSE

- **no license** \Rightarrow 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

 **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/angular speeds

Concerning DifferentialDrive, I ignored [...] ...

Labels

None yet

Milestone

No milestone

Notifications

🔔 Unsubscribe

You're receiving notifications because you authored the thread.

2 participants



🔒 Lock conversation

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](#).

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