

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/lecture-software-engineering/>

**WITH
PLYMOUTH
UNIVERSITY**

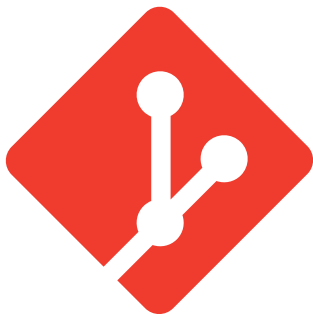
software engineering 101

(with a pinch of git on top)

03 Oct. 2017

Séverin Lemaignan

Centre for Robotics & Neural Systems
Plymouth University



BUILDING CODE

COMPILED VS NOT COMPILED

COMPILING CODE IN C++

```
/*  
 * "Hello, World!": A classic.  
 */  
  
#include <iostream>  
  
using namespace std;  
  
int main(void)  
{  
    cout << "Hello, World!" << endl;  
    return 0;  
}
```

COMPILING CODE IN C++

```
/*  
 * "Hello, World!": A classic.  
 */  
  
#include <iostream>  
  
using namespace std;  
  
int main(void)  
{  
    cout << "Hello, World!" << endl;  
    return 0;  
}  
  
$ g++ hello.cpp -ohello
```

COMPILING CODE IN C++

```
/*  
 * "Hello, World!": A classic.  
 */  
  
#include <iostream>  
  
using namespace std;  
  
int main(void)  
{  
    cout << "Hello, World!" << endl;  
    return 0;  
}  
  
$ g++ hello.cpp -ohello  
  
$ ./hello  
Hello, World!
```

COMPILING CODE IN C++: THE MAIN STAGES

1. Pre-processing
2. Compilation
3. Assembly
4. Linking

These four steps are transparently performed one after the other by your favourite compiler.

COMPILING CODE IN C++: PRE-PROCESSING

```
/*  
 * "Hello, World!": A classic.  
 */  
  
#include <iostream>  
  
using namespace std;  
  
int main(void)  
{  
    cout << "Hello, World!" << endl;  
    return 0;  
}
```

Pre-processor *directives* start with **#**

→ `#include <iostream>` is replaced by the content of that file.

COMPILING CODE IN C++: COMPILATION

```
$ g++ -S hello.cpp
```

```
main:
```

```
.LFB1493:
```

```
.cfi_startproc
```

```
pushq    %rbp
```

```
.cfi_def_cfa_offset 16
```

```
.cfi_offset 6, -16
```

```
movq     %rsp, %rbp
```

```
.cfi_def_cfa_register 6
```

```
leaq     .LC0(%rip), %rsi
```

```
leaq     _ZSt4cout(%rip), %rdi
```

```
call     _ZStlsISt11char_traitsIcEERSt13basic_ost
```

```
movq     %rax, %rdx
```

```
movq     _ZSt4endlIcSt11char_traitsIcEERSt13basic
```

```
movq     %rax, %rsi
```

```
movq     %rdx, %rdi
```

```
call     _ZNSolsEPFRSoS_E@PLT
```

COMPILING CODE IN C++: ASSEMBLY

```
$ g++ -s hello.cpp
$ hexdump a.out
00000000 457f 464c 0102 0001 0000 0000 0000 0000
00000010 0003 003e 0001 0000 07b0 0000 0000 0000
00000020 0040 0000 0000 0000 0000 1128 0000 0000
00000030 0000 0000 0040 0038 0009 0040 001b 001a
00000040 0006 0000 0005 0000 0040 0000 0000 0000
00000050 0040 0000 0000 0000 0040 0000 0000 0000
00000060 01f8 0000 0000 0000 01f8 0000 0000 0000
00000070 0008 0000 0000 0000 0003 0000 0004 0000
00000080 0238 0000 0000 0000 0238 0000 0000 0000
00000090 0238 0000 0000 0000 001c 0000 0000 0000
000000a0 001c 0000 0000 0000 0001 0000 0000 0000
000000b0 0001 0000 0005 0000 0000 0000 0000 0000
000000c0 0000 0000 0000 0000 0000 0000 0000 0000
000000d0 0b78 0000 0000 0000 0b78 0000 0000 0000
000000e0 0000 0020 0000 0000 0001 0000 0006 0000
```

...

COMPILING CODE IN C++: LINKING

The linker copies (and re-arrange) the machine code of the static dependencies (*static libraries*) into the executable.

That's what the `-l` flag is used for:

```
$ g++ cool_app.cpp -ocool_app -lcv_core -lcv_highgui -lcv_vi
```

LIBRARIES

A library is a collection of pre-compiled functions that might get called by an executable. *Libraries are not executable* by themselves.

Why libraries?

- to modularise your code
- to make it easier to reuse

LIBRARIES

A library is a collection of pre-compiled functions that might get called by an executable. *Libraries are not executable* by themselves.

Why libraries?

- to modularise your code
- to make it easier to reuse

Two main kinds:

- Static libraries, whose code is *copied* into the executable by the linker. Extensions: `.a`, `.lib`
- Dynamic libraries, whose code is *loaded by the operating system* at runtime. They are also called *shared libraries*. Extensions: `.so`, `.dll`, `.dylib`

STATIC VS DYNAMIC LIBRARIES

Take 5 min and try to list 2 advantages for the static libraries on one hand, and the dynamic libraries on the other hand.

STATIC VS DYNAMIC LIBRARIES

Advantages of static libraries:

- application can be certain that all its libraries are present
- libraries are the correct version (on Linux, distributions and package managers handle that for dynamic libraries)
- single executable: simpler distribution and installation
- only need to copy (and load into memory) the parts that are needed

STATIC VS DYNAMIC LIBRARIES

Advantages of static libraries:

- application can be certain that all its libraries are present
- libraries are the correct version (on Linux, distributions and package managers handle that for dynamic libraries)
- single executable: simpler distribution and installation
- only need to copy (and load into memory) the parts that are needed

Advantages of dynamic libraries:

- executables smaller because no need to copy the libraries' code
- prevent redundant code in the system
- allows the libraries to be easily updated to fix bugs and security flaws without updating each of the applications

HOW TO MAKE & USE LIBRARIES?

Code source of a pathfinding tool for our robots:

```
main.cpp
```

```
ui.cpp
```

```
pathfinding.cpp
```

```
$ g++ main.cpp ui.cpp pathfinding.cpp -opathfinding_ui
```

`pathfinding.cpp` contains the actual pathfinder, and might be useful for many other projects. How to turn it into a library?

HOW TO MAKE & USE LIBRARIES?

First, we need to extract the **API** of our library in a **public header** `pathfinding.hpp`:

```
#ifndef _PATHFINDING_HPP  
#define _PATHFINDING_HPP  
  
class Pathfinder {  
  
    Pathfinder(std::shared_ptr<const Map> map);  
    Path find(size_t goal_x, size_t goal_y);  
}  
  
#endif
```

The header contains the **declarations** of our classes, structures, functions, but not the **definitions** (the definitions are in `pathfinding.cpp`).

HOW TO MAKE & USE LIBRARIES?

Next, compile the library:

```
$ g++ -fPIC -shared pathfinding.cpp -olibpathfinding.so
```

HOW TO MAKE & USE LIBRARIES?

Finally, use it:

```
$ g++ main.cpp ui.cpp -lpathfinding -opathfinding_ui
```

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**.

EXAMPLE OF A CMAKE FILE

```
cmake_minimum_required(VERSION 2.8.3)
project(cmake_example)

find_package(OpenCV REQUIRED)

include_directories(include)

add_definitions(-std=c++11)

add_executable(run src/main.cpp)
target_link_libraries(run ${OpenCV_LIBRARIES})

install(TARGETS run
        RUNTIME DESTINATION bin
)

install(FILES
        config/app.conf
        DESTINATION etc
)
```

ORGANISING YOUR CODE

principle of least surprise

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

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?

EXAMPLE

```
my_proj/  
  main.cpp  
  ui.cpp  
  ui.hpp  
  pathfinding.cpp  
  pathfinding.hpp  
  ui.conf
```

EXAMPLE

```
my_proj/  
  src/  
    main.cpp  
    ui.cpp  
    ui.hpp  
    pathfinding.cpp  
  include/  
    pathfinding.hpp  
  etc/  
    ui.conf  
  README.md  
  CMakeLists.txt
```

EXAMPLE

When compiling the project, create a sub-directory `build` and perform an **out-of-tree** build:

```
$ mkdir build && cd build  
$ cmake ..  
$ make
```

EXAMPLE

```
build/  
    ... # lots of compilation artifacts  
src/  
    main.cpp  
    ui.cpp  
    ui.hpp  
    pathfinding.cpp  
include/  
    pathfinding.hpp  
etc/  
    ui.conf  
README.md  
CMakeLists.txt
```

The `build/` directory can be deleted at any point as it contains only generated files.

VERSIONING

SEMANTIC VERSIONING

Given a version number `MAJOR.MINOR.PATCH`, increment the:

- `MAJOR` version when you make incompatible API changes,
- `MINOR` version when you add functionality in a backwards-compatible manner, and
- `PATCH` version when you make backwards-compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the `MAJOR.MINOR.PATCH` format.

Source: [semver website](#)

SEMANTIC VERSIONING

You are the maintainer of `cool_app`, that depends on OpenCV 2.4.11.

The OpenCV project releases a new version, what should you do...

- ...if the new version is 2.4.12?

SEMANTIC VERSIONING

You are the maintainer of `cool_app`, that depends on OpenCV 2.4.11.

The OpenCV project releases a new version, what should you do...

- ...if the new version is 2.4.12?
- ...if the new version is 2.5.0?

SEMANTIC VERSIONING

You are the maintainer of `cool_app`, that depends on OpenCV 2.4.11.

The OpenCV project releases a new version, what should you do...

- ...if the new version is 2.4.12?
- ...if the new version is 2.5.0?
- ...if the new version is 2.9.0?

SEMANTIC VERSIONING

You are the maintainer of `cool_app`, that depends on OpenCV 2.4.11.

The OpenCV project releases a new version, what should you do...

- ...if the new version is 2.4.12?
- ...if the new version is 2.5.0?
- ...if the new version is 2.9.0?
- ...if the new version is 3.0.0-beta?

SEMANTIC VERSIONING

You are the maintainer of `cool_app`, that depends on OpenCV 2.4.11.

The OpenCV project releases a new version, what should you do...

- ...if the new version is 2.4.12?
- ...if the new version is 2.5.0?
- ...if the new version is 2.9.0?
- ...if the new version is 3.0.0-beta?
- ...if the new version is 3.0.0?

SOFTAWRE LICENSES, OPEN-SOURCE, FREE SOFTWARE

SOFTWARE LICENSES

- **no license** \Rightarrow default copyright laws apply. You retain all rights to your source code; nobody else may reproduce, distribute, or create derivative works from your work.



SOFTWARE LICENSES

- **no license** \Rightarrow default copyright laws apply. You 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



SOFTWARE LICENSES

- **no license** \Rightarrow default copyright laws apply. You 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



SOFTWARE LICENSES

- **no license** ⇒ default copyright laws apply. You 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.

SOFTWARE LICENSES

- **no license** \Rightarrow default copyright laws apply. You 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/>

WHAT IF YOU WANT TO USE A GPL LIBRARY?

There is a legal dispute to know whether merely *linking* with a library result in a ***derivative work*** (which would then have to be licensed as GPL).

WHAT IF YOU WANT TO USE A GPL LIBRARY?

There is a legal dispute to know whether merely *linking* with a library result in a *derivative work* (which would then have to be licensed as GPL).

The LGPL (*Lesser GPL*) explicitly allows the usage of the library without putting restrictions on the licensing of the resulting executable.

Open-source vs Free software?

"When we call software "free," we mean that it respects the users' essential freedoms: the freedom to run it, to study and change it, and to redistribute copies with or without changes. This is a matter of freedom, not price, so think of "free speech," not "free beer."

"Open source is a development methodology; free software is a social movement"

Source:GNU website

GIT

Re: Kernel SCM saga..

From: Linus Torvalds

Date: Thu Apr 07 2005 - 23:41:58 EST

- **Next message:** [Evgeniy Polyakov: "Re: \[Fwd: Re: connector is missing in 2.6.12-rc2-mm1\]"](#)
 - **Previous message:** [David S. Miller: "Re: \[Fwd: Re: connector is missing in 2.6.12-rc2-mm1\]"](#)
 - **In reply to:** [Chris Wedgwood: "Re: Kernel SCM saga.."](#)
 - **Next in thread:** [kfogel: "Re: Kernel SCM saga.."](#)
 - **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)
-

On Thu, 7 Apr 2005, Chris Wedgwood wrote:

>

> *I'm playing with monotone right now. Superficially it looks like it*
> *has tons of gee-whiz neato stuff... however, it's *agonizingly* slow.*
> *I mean glacial. A heavily sedated sloth with no legs is probably*
> *faster.*

Yes. The silly thing is, at least in my local tests it doesn't actually seem to be `_doing_` anything while it's slow (there are no system calls except for a few memory allocations and de-allocations). It seems to have some exponential function on the number of pathnames involved etc.

I'm hoping they can fix it, though. The basic notions do not sound wrong.

In the meantime (and because monotone really `_is_` that slow), here's a quick challenge for you, and any crazy hacker out there: if you want to play with something `_really_` nasty (but also very `_very_` fast), take a look at kernel.org/pub/linux/kernel/people/torvalds/.

First one to send me the changelog tree of sparse-git (and a tool to commit and push/pull further changes) gets a gold star, and an honorable mention. I've put a hell of a lot of clues in there (*).

I've worked on it (and little else) for the last two days. Time for somebody else to tell me I'm crazy.

Linus

(*) It should be easier than it sounds. The database is designed so that you can do the equivalent of a nonmerging (ie pure superset) push/pull with just plain rsync, so replication really should be that easy (if somewhat bandwidth-intensive due to the whole-file format)

Brian Harrys blog

Everything you want to know about Visual Studio ALM and Farming

The largest Git repo on the planet

05/24/2017 by [Brian Harry MS](#) // [59 Comments](#)



It's been 3 months since I first wrote about [our efforts to scale Git to extremely large projects and teams](#) with an effort we called "Git Virtual File System". As a reminder, GVFS, together with a set of enhancements to Git, enables Git to scale to VERY large repos by virtualizing both the .git folder and the working directory. Rather than download the entire repo and checkout all the files, it dynamically downloads only the portions you need based on what you use.

A lot has happened and I wanted to give you an update. Three months ago, GVFS was still a dream. I don't mean it didn't exist – we had a concrete implementation, but rather, it was unproven. We had validated on some big repos but we hadn't rolled it out to any meaningful number of engineers so we had only conviction that it was going to work. Now we have proof.

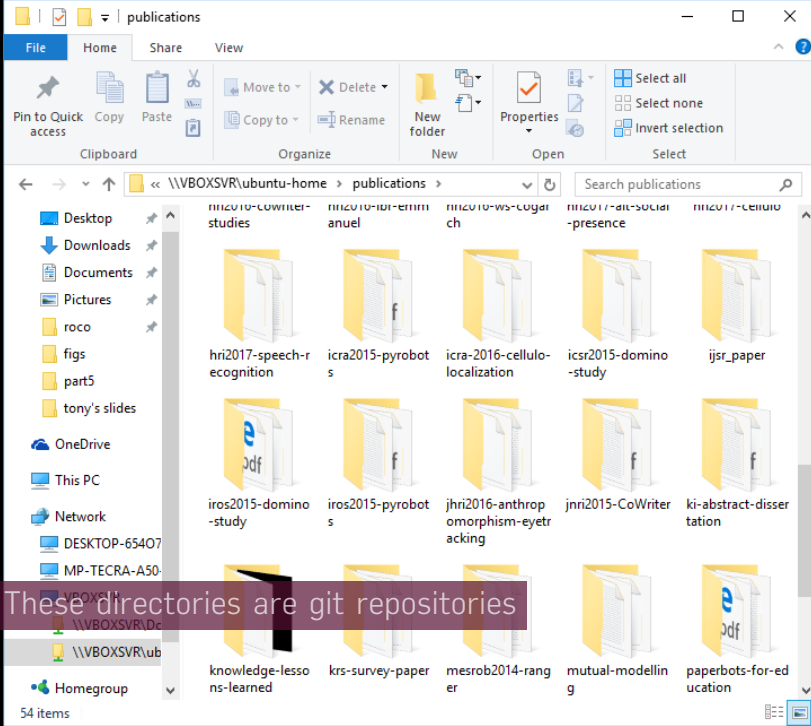
Today, I want to share our results. In addition, we're announcing the next steps in our GVFS journey for customers, including expanded open sourcing to start taking contributions and improving how it works for us at Microsoft, as well as for partners and customers.

Windows is live on Git

Over the past 3 months, we have largely completed the rollout of Git/GVFS to the Windows team at Microsoft.

As a refresher, the Windows code base is approximately 3.5M files and, when checked in to a Git repo, results in a repo of about 300GB. Further, the Windows team is about 4,000 engineers and the engineering system produces 1,760 daily "lab builds" across 440 branches in addition to thousands of pull request validation builds. All 3 of the dimensions (file count, repo size and activity), independently, provide daunting scaling challenges and taken together they make it unbelievably challenging to create a great experience. Before the move to Git, in Source Depot, it was spread across 40+ depots and we had a tool to manage operations that spanned them.

As of my writing 3 months ago, we had all the code in one Git repo, a few hundred engineers using it and a small fraction (<10%) of the daily build load. Since then, we have rolled out in waves across the engineering team.



File Home Share View

Clipboard: Pin to Quick access, Copy, Paste

Organize: Move to, Delete, Copy to, Rename

New: New folder

Open: Properties

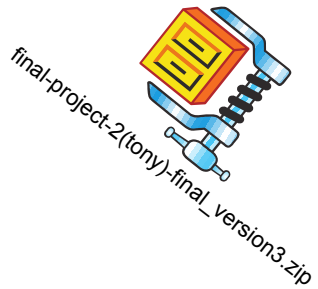
Select: Select all, Select none, Invert selection

← → ↑ ↓ publications > jnnr2015-CoWriter > Search jnnr2015-CoWriter

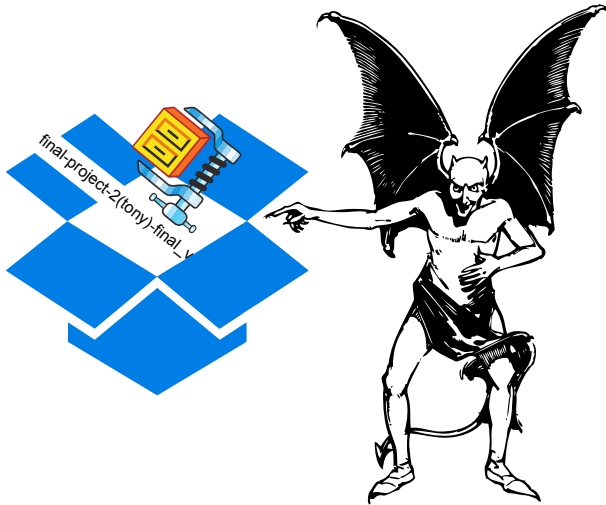
| | Name | Date modified | Type | Size |
|---------------|------------------------|--------------------|--------------------|------|
| Desktop | .git | 11/5/2014 1:29 PM | File folder | |
| Downloads | figures | 11/5/2014 12:39 PM | File folder | |
| Documents | lemaignan2014being.pdf | 11/5/2014 1:29 PM | PDF File | 5,11 |
| Pictures | library.bib | 11/5/2014 12:39 PM | BIB File | 7 |
| roco | library.bib.bak | 11/5/2014 12:39 PM | BAK File | 7 |
| figs | main.aux | 11/5/2014 1:29 PM | AUX File | |
| part5 | main.bbl | 11/5/2014 1:16 PM | BBL File | |
| tony's slides | main.blg | 11/5/2014 1:16 PM | Performance Mon... | |
| OneDrive | main.log | 11/5/2014 1:29 PM | Text Document | 3 |
| This PC | main.new.tex | 11/5/2014 12:39 PM | TEX File | 5 |
| Network | main.out | 11/5/2014 1:29 PM | OUT File | |
| DESKTOP-65407 | main.pdf | 11/5/2014 1:29 PM | PDF File | 5,11 |
| MP-TECRA-A50 | main.tex | 11/5/2014 1:29 PM | TEX File | 1 |
| VBOXSVR | Makefile | 11/5/2014 12:39 PM | File | |
| \\VBOXSVR\De | sig-alternate.cls | 11/5/2014 12:39 PM | CLS File | 5 |
| \\VBOXSVR\ub | sig-alternate.cls | 11/5/2014 12:39 PM | PY File | |

They look boringly normal

A few roundtrips later with colleagues...









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.

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!

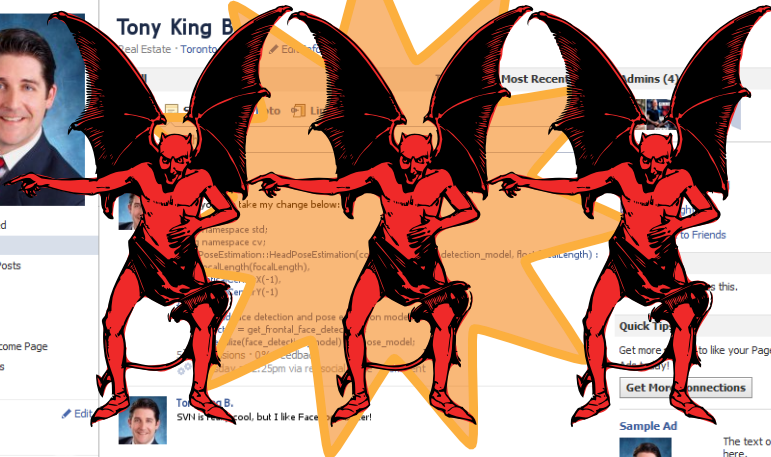
7

people like this

Add to My Page's Favorites

51 Impressions · 0% Feedback

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



Quick Tip

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.

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.

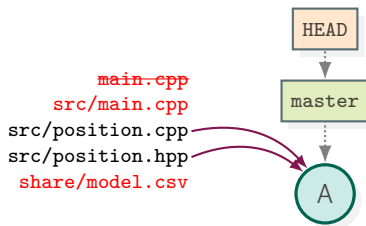
ATOMIC COMMITS

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

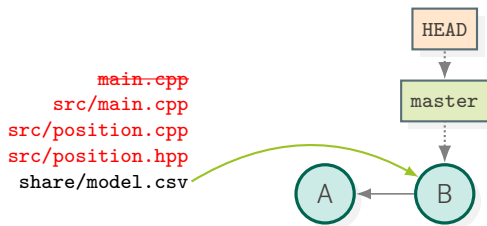
Atomic commit

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

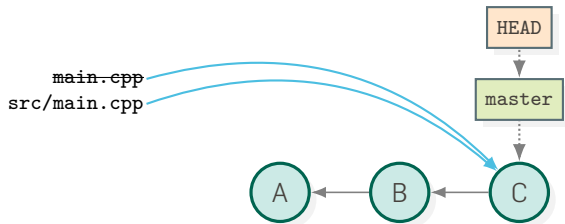
```
main.cpp  
src/main.cpp  
src/position.cpp  
src/position.hpp  
share/model.csv
```



```
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
```

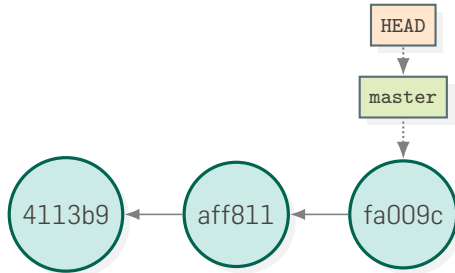
```
Re-trained model with 52 more participants
```

```
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)
```



THE STAGING AREA

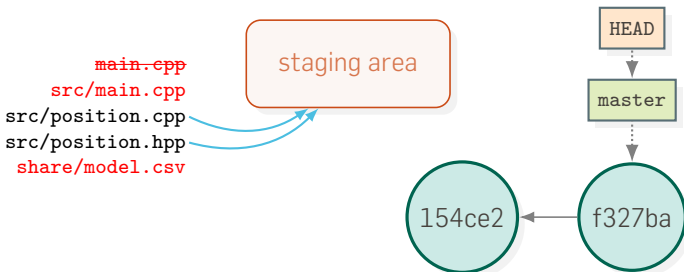
But why do we have to manually tell Git what files to add or remove?

THE STAGING AREA

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

THE STAGING AREA

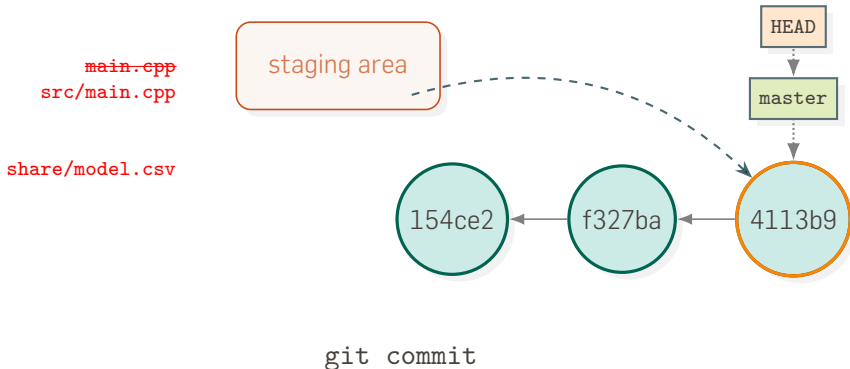
Preparing a commit consists in filling the **staging area** (or **index**) with the list of changes:



```
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

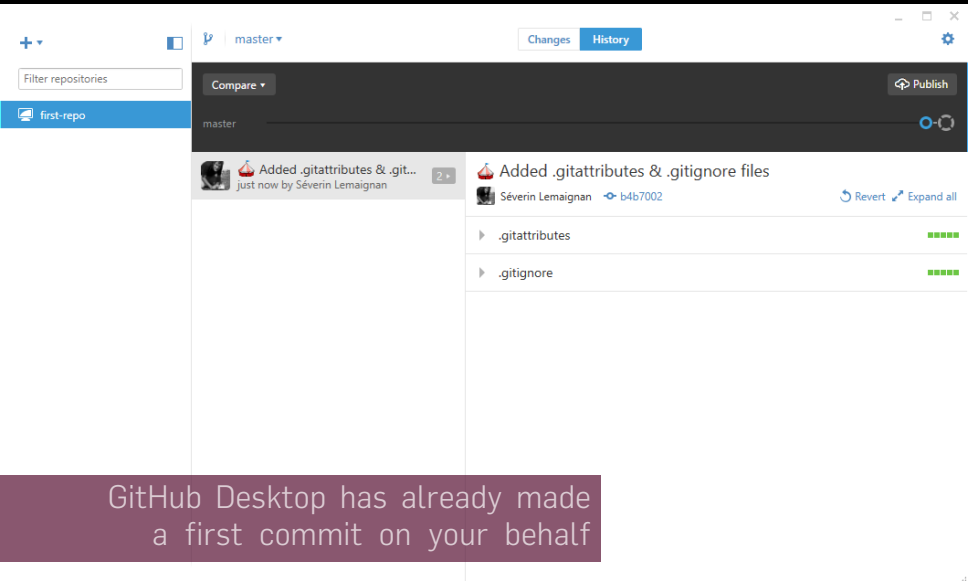
Git ignore Windows

✓ Create repository



Get started by adding a repository.

Create a (local) repository

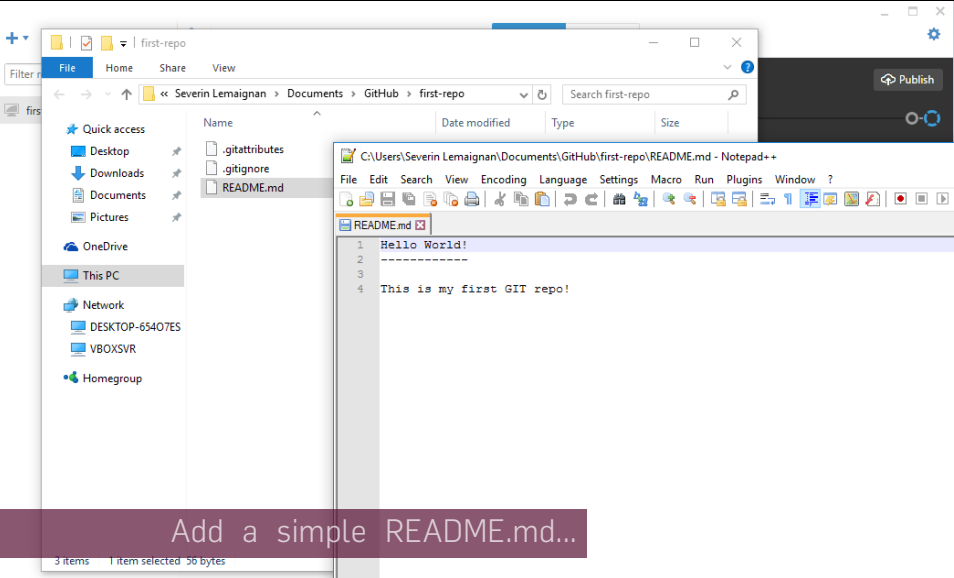


Visual Studio interface showing a Git repository named "first-repo". The "master" branch is selected. The "Changes" tab is active, displaying a commit history. A context menu is open over the repository name, offering options: "View on GitHub", "Open in Explorer", "Open in Git Shell", and "Remove".

The commit history shows a commit by Séverin Lemaignan (b4b7002) titled "Added .gitattributes & .gitignore files". The diff view shows changes to the following files:

- .gitattributes
- .gitignore

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



Add a simple README.md...

The screenshot displays the GitHub web interface for a repository named 'first-repo'. The 'Changes' tab is selected, showing a comparison of the 'master' branch. A dark banner at the top right indicates 'View 1 uncommitted change' and includes a 'Publish' button. Below this, a list of changes shows '1 change' with a checkbox and 'README.md' with a green progress bar. At the bottom, there are input fields for 'Summary' and 'Description'. A dark purple banner at the very bottom of the image contains the text: 'The change is listed in the Changes panel'.

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

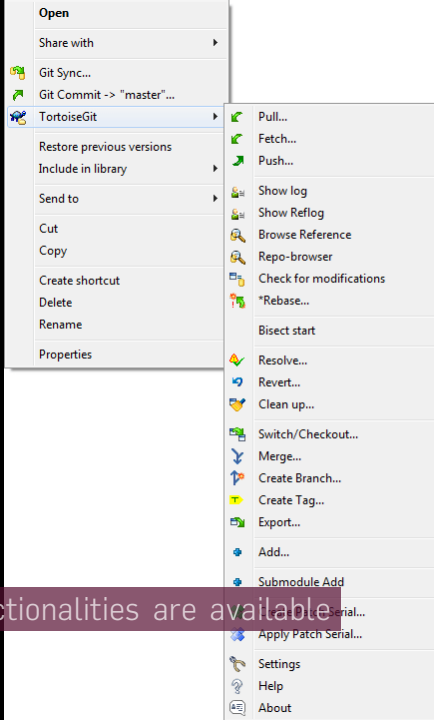


skip-worktree.cpp



non-versioned.cpp

Files' status appear as icons



All the functionalities are available

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 | |
| <input checked="" type="checkbox"/> Languages/Tortoise_bg.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_ca.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_cs.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_da.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_de.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_es.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_fi.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_fr.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_hu.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_id.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_it.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_ja.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_ko.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_nl.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_pl.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_pt_BR.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_pt_PT.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_ro.po | .po |
| <input checked="" type="checkbox"/> Languages/Tortoise_ru | .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

☒ Show Unversioned Files☒ Do not autoselect submodules☐ Show Whole Project☐ Message only

Commit

Cancel

Help

Commit window

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.

File Home Share View

Clipboard: Pin to Quick access, Copy, Paste

Organize: Move to, Delete, Copy to, Rename

New: New folder

Open: Properties

Select: Select all, Select none, Invert selection

← → ↑ ↓ publications > jnnr2015-CoWriter > Search jnnr2015-CoWriter

| | Name | Date modified | Type | Size |
|---------------|------------------------|--------------------|--------------------|------|
| Desktop | .git | 11/5/2014 1:29 PM | File folder | |
| Downloads | figures | 11/5/2014 12:39 PM | File folder | |
| Documents | lemaignan2014being.pdf | 11/5/2014 1:29 PM | PDF File | 5,11 |
| Pictures | library.bib | 11/5/2014 12:39 PM | BIB File | 7 |
| roco | library.bib.bak | 11/5/2014 12:39 PM | BAK File | 7 |
| figs | main.aux | 11/5/2014 1:29 PM | AUX File | |
| part5 | main.bbl | 11/5/2014 1:16 PM | BBL File | |
| tony's slides | main.blg | 11/5/2014 1:16 PM | Performance Mon... | |
| OneDrive | main.log | 11/5/2014 1:29 PM | Text Document | 3 |
| This PC | main.new.tex | 11/5/2014 12:39 PM | TEX File | 5 |
| Network | main.out | 11/5/2014 1:29 PM | OUT File | |
| DESKTOP-65407 | main.pdf | 11/5/2014 1:29 PM | PDF File | 5,11 |
| MP-TECRA-A50 | main.tex | 11/5/2014 1:29 PM | TEX File | 1 |
| VBOXSVR | Makefile | 11/5/2014 12:39 PM | File | |
| \\VBOXSVR\Dr | sig-alternate.cls | 11/5/2014 12:39 PM | CLS File | 5 |
| \\VBOXSVR\ub | sig-alternate.cls | 11/5/2014 12:39 PM | PY File | |
| Homegroup | | | | |

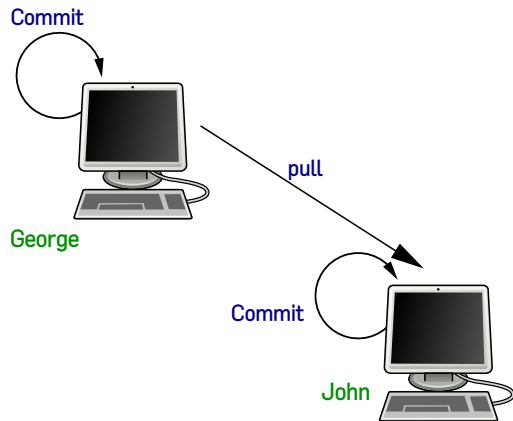
What should I track here?

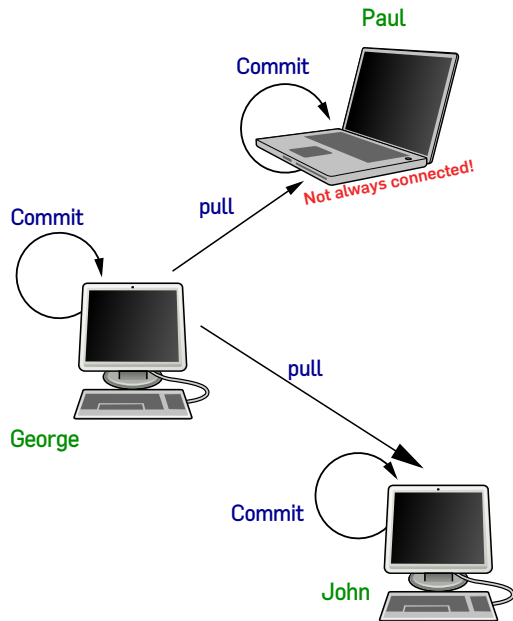
COLLABORATING

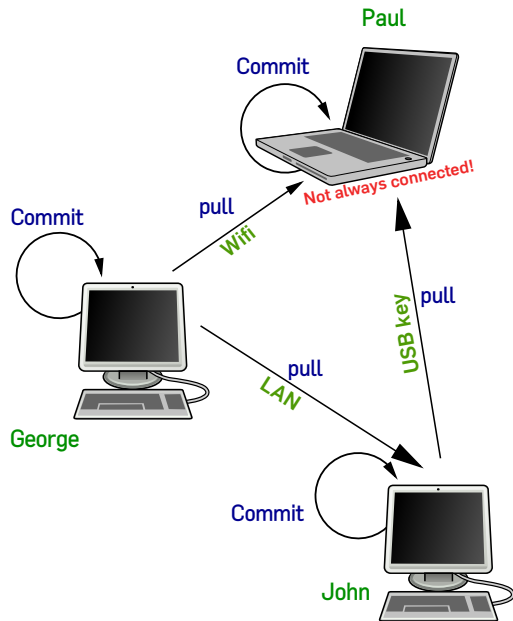
Commit

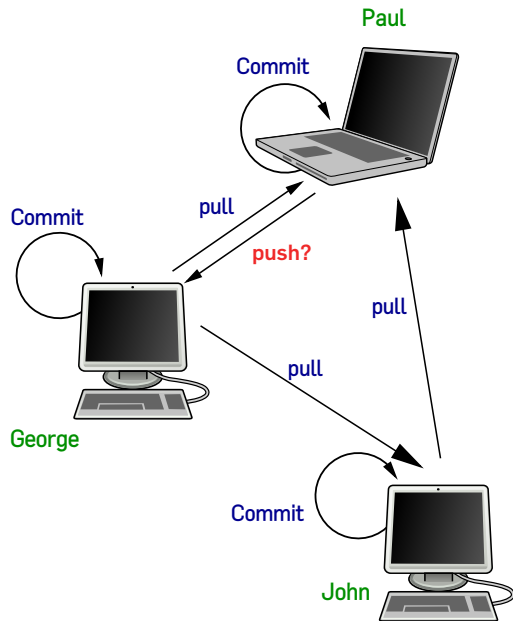


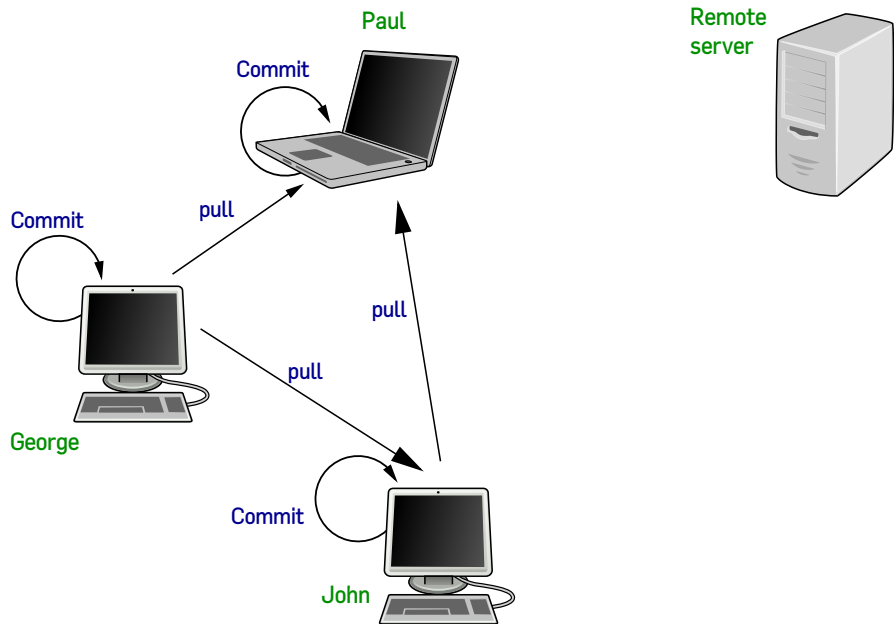
George



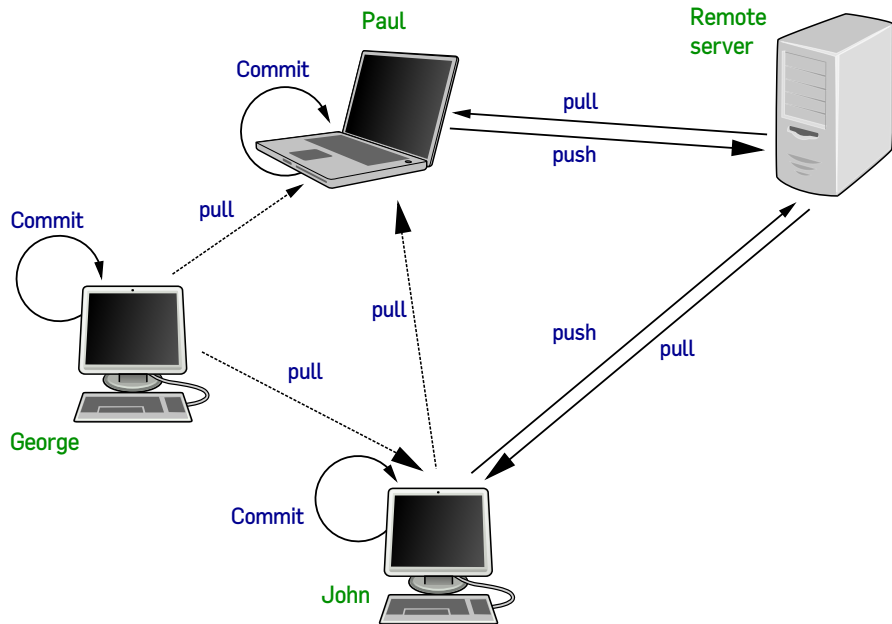




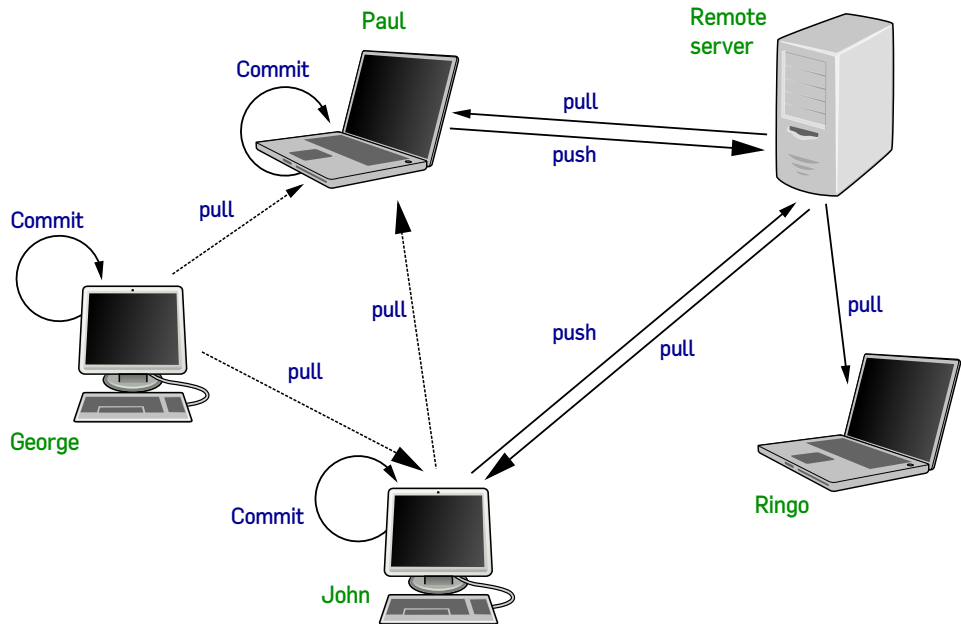




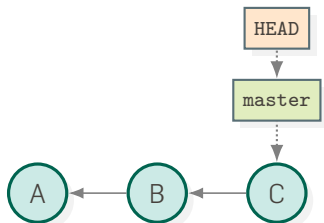
Based on a figure by M. Herrb, CC-BY-SA 3.0

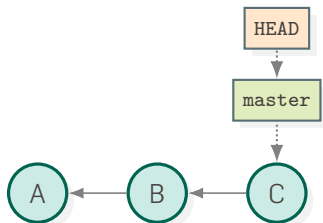


Based on a figure by M. Herrb, CC-BY-SA 3.0

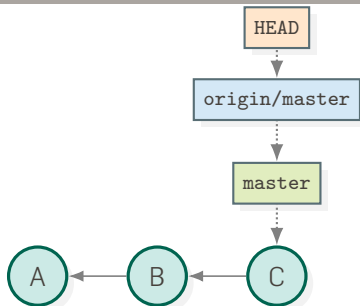


Based on a figure by M. Herrb, CC-BY-SA 3.0



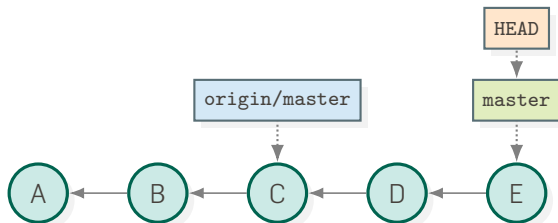


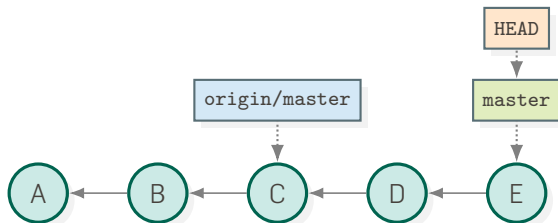
```
git remote add origin git@github.com:user/repo.git
git remote add john-usb E:\john_repo
git remote add ftp-origin ftp://host.xz/path/to/repo.git/
...
```



```
git push origin master
```

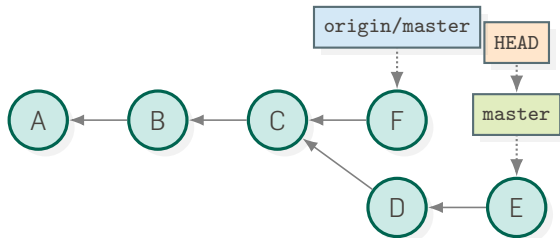
(or simply `git push`)

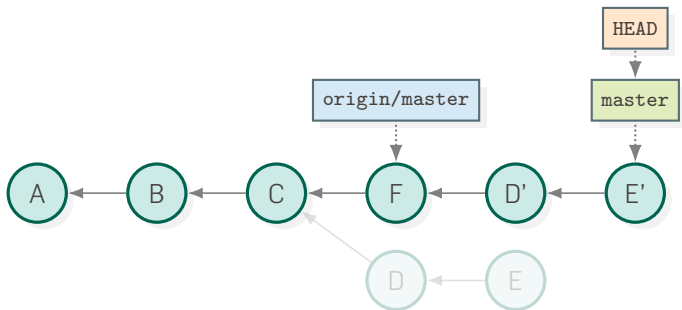




What happened on our remote? Let's have a look...

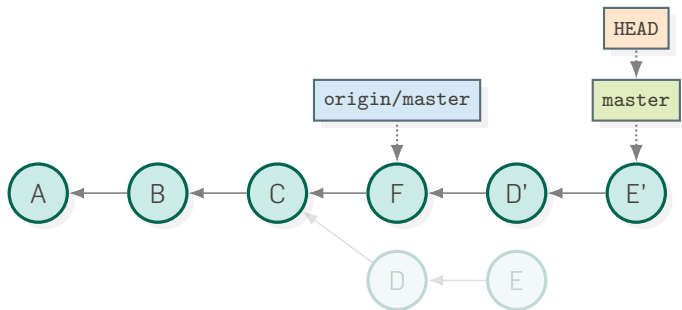
```
git fetch origin
```



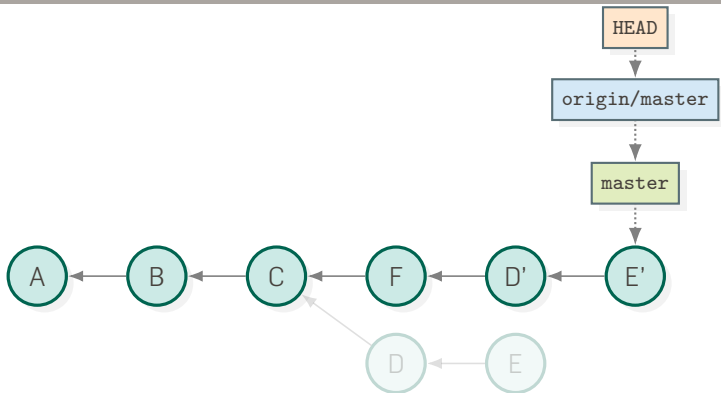


```
git rebase origin/master
```

(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"** in

To check out the original branch and stop rebasing, run **"git**

```
$ 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.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
- 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

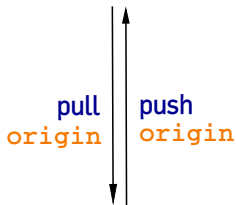
<https://github.com/morse>



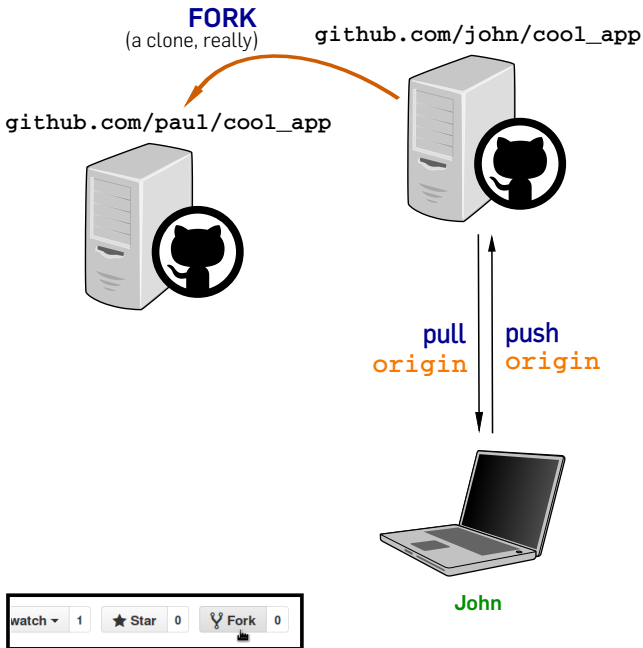
Download ZIP

| | | |
|-----------------|--------------------------------------------------------------------------|----------------------------------|
| 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

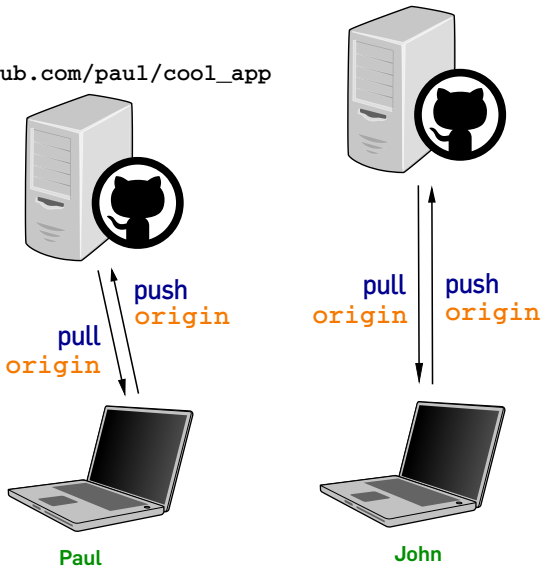


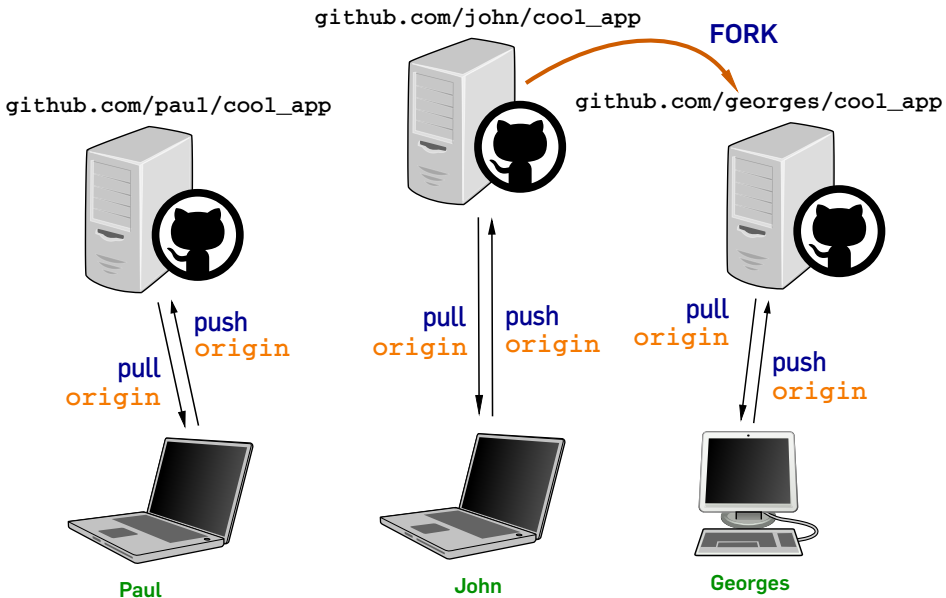
John



github.com/john/cool_app

github.com/paul/cool_app

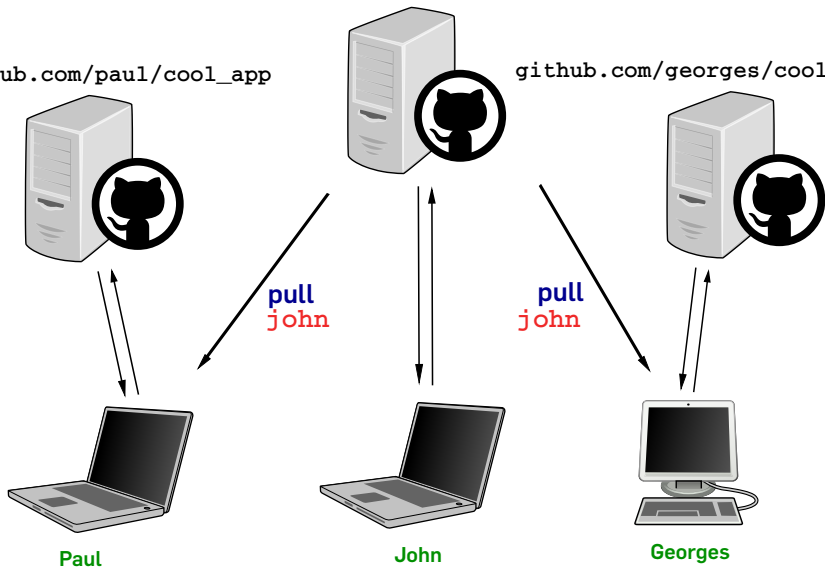


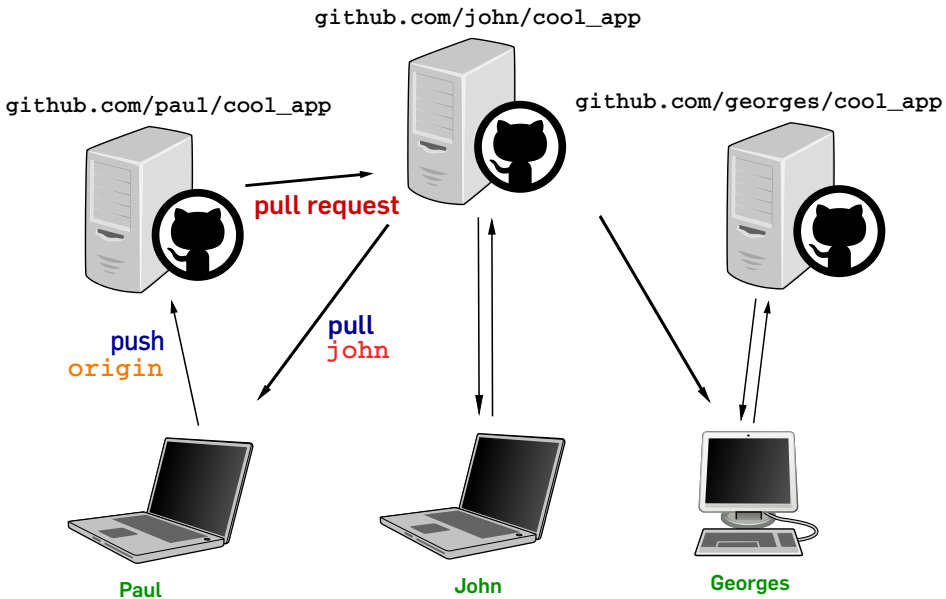


github.com/john/cool_app

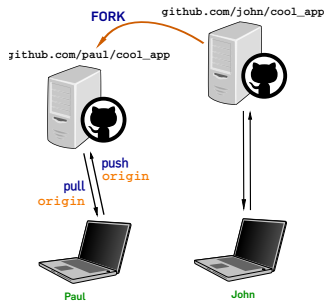
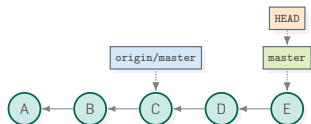
github.com/paul/cool_app

github.com/georges/cool_app





WHAT HAPPENED EXACTLY?

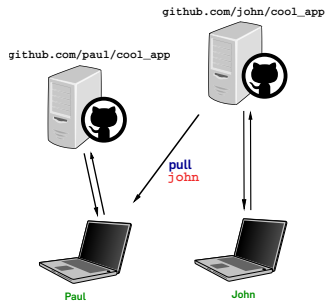
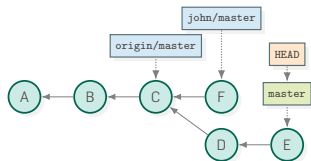


After forking on GitHub, Paul runs

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

and he adds few local commits

WHAT HAPPENED EXACTLY?

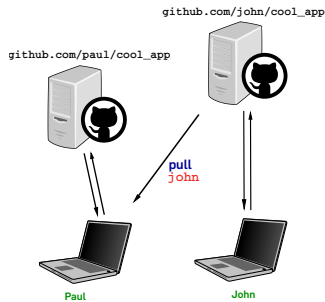
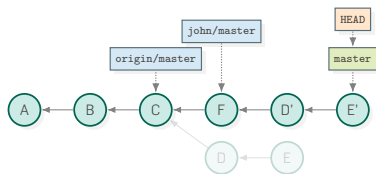


He would like to propose his changes to John

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?

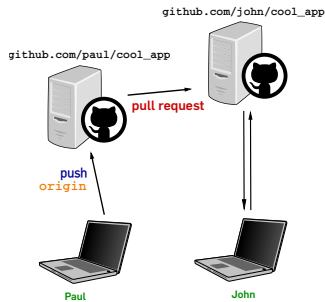
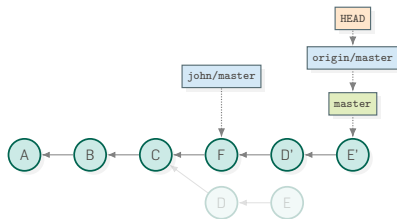


Paul rebases his `master` branch on John's one:

```
git rebase john/master
```

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

WHAT HAPPENED EXACTLY?

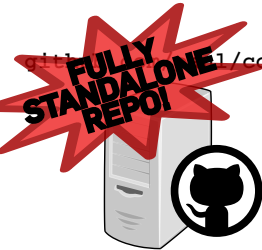


He pushes his commits to his own GitHub account:

```
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!

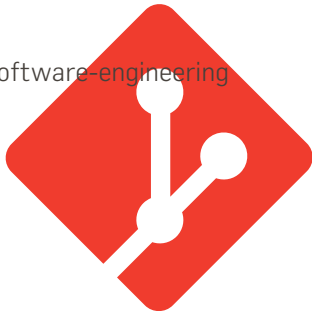
`git reflog`

That's all, folks!

Questions:

Portland Square B316 or **severin.lemaignan@plymouth.ac.uk**

Slides: github.com/severin-lemaignan/lecture-software-engineering

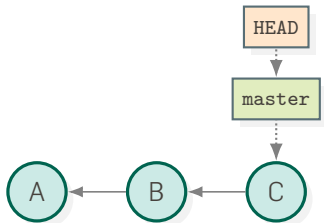


SUPPLEMENTARY MATERIAL

11. Working with branches

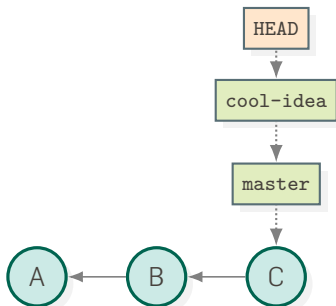
WORKING WITH BRANCHES

BRANCHES



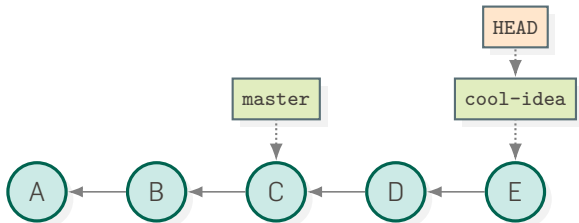
What if...?

BRANCHES

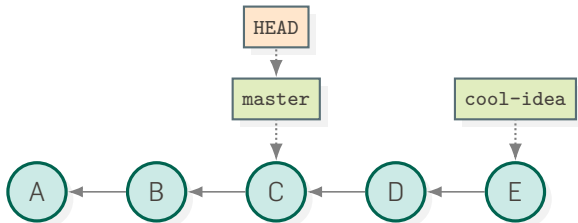


```
git checkout -b cool-idea
```

BRANCHES

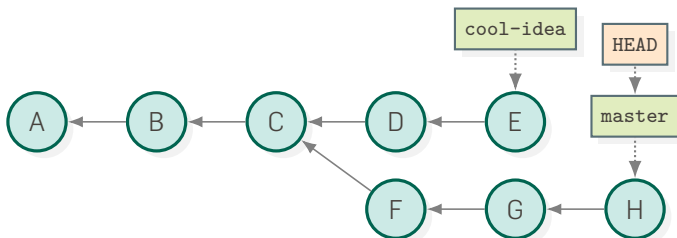


BRANCHES



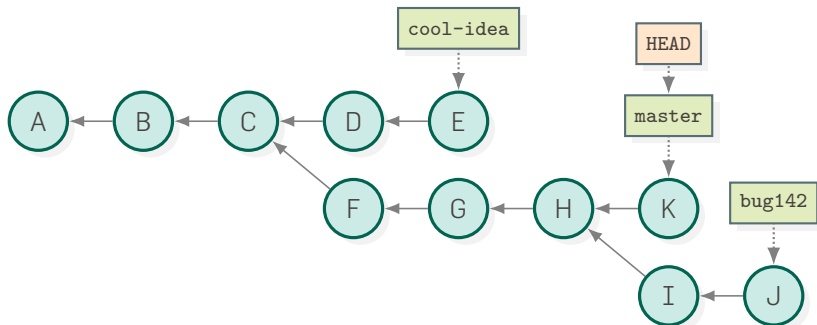
Let go back to serious stuff!
`git checkout master`

BRANCHES



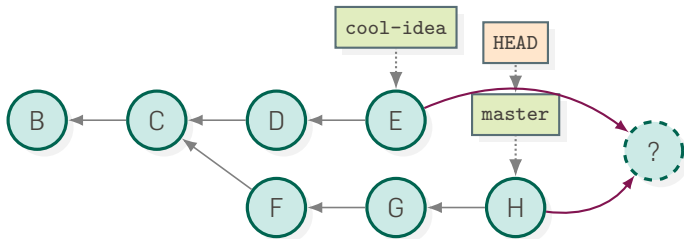
The branch name is an alias for the tip of the current branch

BRANCHES



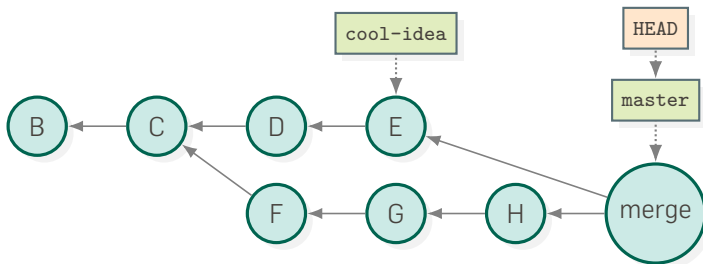
⇒ branches are very cheap
+10 of them at a given time it not uncommon

MERGING BRANCHES



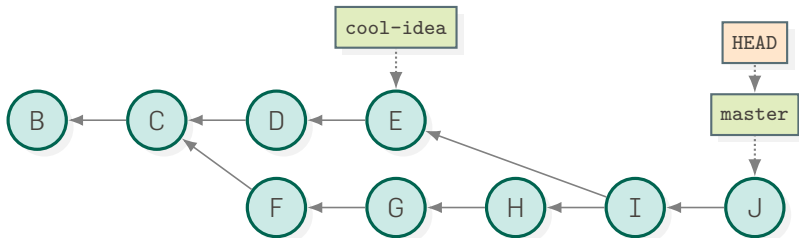
Two options: **merging** and **rebasing**

MERGING BRANCHES



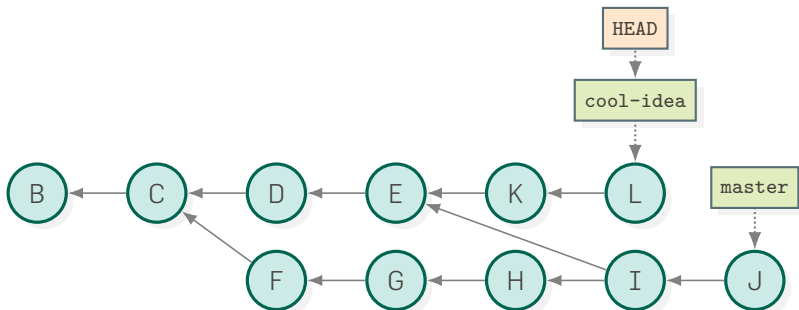
Merging
`git merge cool-idea`

MERGING BRANCHES



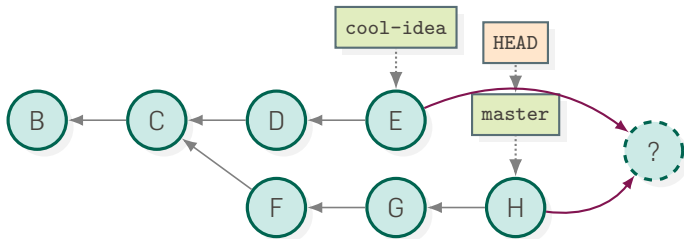
```
git commit
```

MERGING BRANCHES

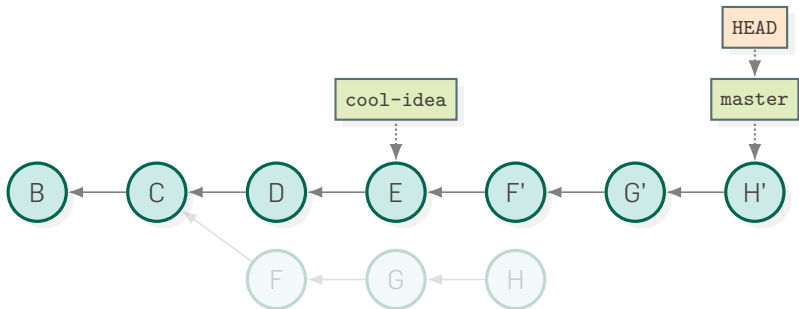


```
git checkout cool-idea  
git commit  
...etc.
```

REBASING BRANCHES

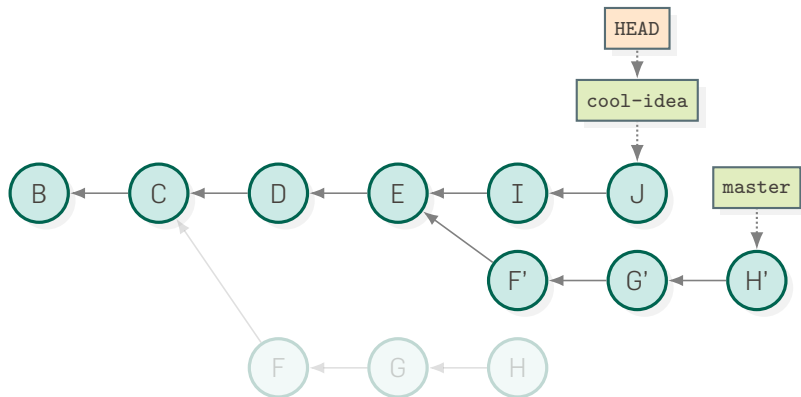


REBASING BRANCHES



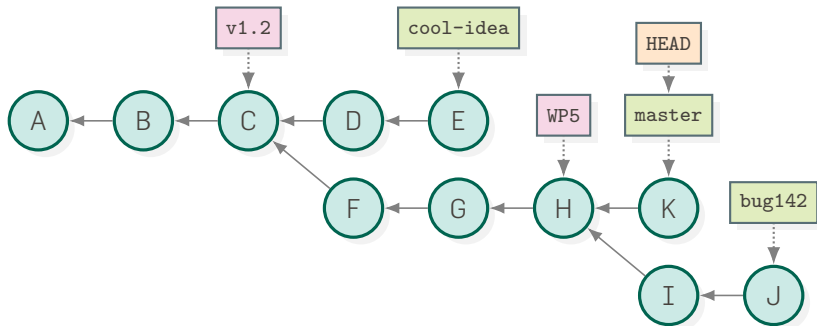
Rebasing
`git rebase cool-idea`

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

ChangesHistory

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

oved grid rendering with coordinates

Séverin Lemaignan da25158

RevertCollapse 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

Publish

Update from master

View branch

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 "Recent branches" (including `numerical_coordinates` and `master`), and a main editor area displaying a C++ file named `main.cpp`. The code in `main.cpp` shows a `while` loop with a `cout` statement. The current line (19) is highlighted in green, indicating a change from `cout << " A B C " << endl;` to `cout << " 1 2 3 " << endl;`. The interface also shows a "Changes" tab and a "History" tab, along with a "Publish" button.

We can jump between branches...

Filter repositories

first-repo

master

ChangesHistory

Update from numerical_coordinatesView branchPublish

numerical_coordinatesmaster

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

main.cpp

@@ -13,6 +13,10 @@ int main(int argc, char** argv) {
13130,1,0,
14140,0,1};
1515
16+ char x_char;
17+ int x;
18+ int y;
19+
1620 while(!done) {
1721 char i = 0;
1822
@@ -30,7 +34,21 @@ int main(int argc, char** argv) {
3034 cout << i/3 + 1;
3135 }
3236 }
33- done = true;
37+
38+ cout << endl << "Enter X coordinate (A, B or C):";
39+ cin >> x_char;
40+ 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



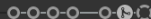
Filter repositories

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  +   cout << endl << "Enter X coordinate (A, B or C):";
35  +   cin >> x_char;
36  +   x = (x_char == 'A' ? 0 : (x_char == 'B' ? 1 : 2));
40  +
```

The merge commit is reflected
in the history of the branch

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.

one repo = one thing

make plenty of repos!

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