

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 Ian Howard and Séverin Lemaignan as being the original authors,
- (2) you re-share your presentation under the same terms.

You can download the sources of this presentation here:
github.com/severin-lemaignan/module-introduction-sensors-actuators

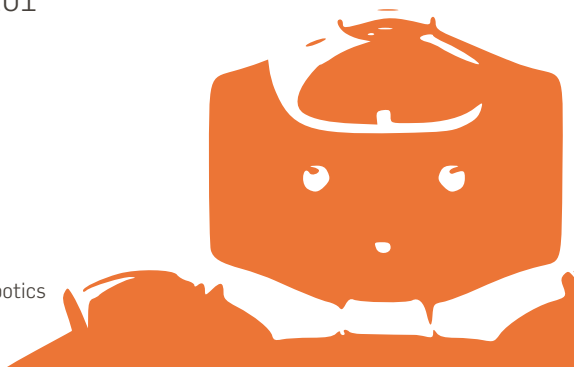
ROC0222

Intro to Sensors and Actuators

Software Engineering 101

Séverin Lemaignan

Centre for Neural Systems and Robotics
Plymouth University



LECTURE'S OBJECTIVES

In 2 hours time, you should know:

- What does “compiling code” *really* means
- What **JIT** (Just-In-Time) compilation means
- The difference between a `dll` and a `lib`
- (and what are `dlls`)
- What is **CMake**
- How to organise your code on your hard-drive
- What *Filesystem Standard Hierarchy* means
- What a *ROS package* is
- The differences between the GPL, MIT, BSD, ... licenses
- ...and plenty of things about GIT!

BUILDING CODE

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     _ZNSo1sERSt13basic_ostreamIcSt11char_traitsIcE
```

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 0000 01f8 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
  
```

That's what the `-1` flag is used for:

COMPILED VS NOT COMPILED

Multiple **execution models**:

- Compiled languages (eg C, C++, Ada...)
- Interpreted languages (eg: ...?)

COMPILED VS NOT COMPILED

Multiple **execution models**:

- Compiled languages (eg C, C++, Ada...)
- Interpreted languages (eg: ...?)
- → most 'interpreted languages' are actually 'JITed'

Just-In-Time compilation: the interpreter generates an efficient **intermediate representation** (commonly called **bytecode**) that is executed (and often stored).

Very common execution model: Python, C#, Javascript...

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

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: CMAKELISTS.TXT

```

cmake_minimum_required(VERSION 2.8.3)
project(cmake_example)

find_package(OpenCV REQUIRED) # one external dependency

# First, the library
add_library(pathfinding pathfinding.cpp)

install(TARGET pathfinding
        LIBRARY DESTINATION lib
)

# then, the executable, which depends on the library's target
include_directories(include)
add_executable(pathfinding_ui pathfinding main.cpp ui.cpp)
target_link_libraries(pathfinding_ui ${OpenCV_LIBRARIES})

install(TARGET pathfinding_ui
        RUNTIME DESTINATION bin
)

```

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/ | <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?

EXAMPLE 1

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

EXAMPLE 1

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

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

EXAMPLE 2: YOU TAKE OVER AN EXISTING PROJECT

```
joe@doe:/usr/robot-planning$ ls -alh
drwxr-xr-x  2 joe  root    4.0K Sep 22 10:40 .
drwxr-xr-x 12 root  root    4.0K Sep 22 10:36 ..
-rwxrwxr-x  1 joe  joe     8.8K Sep 22 10:40 pathfinding_ui
-rw-rw-r--  1 joe  joe     2.1K Sep 22 10:39 compile.sh
-rw-rw-r--  1 joe  joe     1.8K Sep 22 10:39 compile.bat
-rw-rw-r--  1 joe  joe      134 Sep 22 10:39 readme-first.txt
-rw-rw-r--  1 joe  joe     895 Sep 21 21:38 main.cpp
-rw-rw-r--  1 joe  joe     1.2K Sep 21 20:27 main.hpp
-rw-rw-r--  1 joe  joe     5.8K Sep 22 10:40 main.o
-rw-rw-r--  1 joe  joe      50 Sep 19 10:36 main.ini
-rw-rw-r--  1 joe  joe     2.3K Sep 20 09:31 pathfinding.cpp
-rw-rw-r--  1 joe  joe      230 Sep 20 10:32 pathfinding.hpp
-rw-rw-r--  1 joe  joe     4.3K Sep 22 10:40 pathfinding.o
-rw-rw-r--  1 joe  joe     7.3K Sep 21 10:40 pathfinding.so
-rw-rw-r--  1 joe  joe     6.7K Sep 20 11:13 core.cpp
-rw-rw-r--  1 joe  joe     7.1K Sep 22 10:40 core.o
-rw-rw-r--  1 joe  joe     6.1K Sep 22 10:40 core.a
-rw-rw-r--  1 joe  joe     6.0K Sep 21 16:22 core.lib
```

EXAMPLE 2: YOU TAKE OVER AN EXISTING PROJECT

Points that can be improved:

- Developing in `usr/` is a bad practice
- Rename files to be more descriptive
- Change the layout to follow the FHS (eg `main.ini` to `etc/main.ini`)
- Perform out-of-tree compilation
- Use a buildsystem (like CMake) instead of relying on ad-hoc scripts
- Add a **README** and a **LICENSE**
- Public headers should be moved to `include/`

ORGANISING YOUR CODE: ROS PACKAGES

When working with ROS, you can use `catkin_create_pkg` to quickly generate a project skeleton + `CMakeLists.txt`:

```
> catkin_create_pkg my_ros_node std_msgs rospy roscpp
> ls -lh my_ros_node
-rw-r--r-- 1 joe joe 6.7K Nov 28 15:46 CMakeLists.txt
drwxr-xr-x 3 joe joe 4.0K Nov 28 15:46 include
-rw-r--r-- 1 joe joe 2.1K Nov 28 15:46 package.xml
drwxr-xr-x 2 joe joe 4.0K Nov 28 15:46 src
```

```
catkin_create_pkg <package_name> [depend1] [depend2] [depend3]
```

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?

SOFTWARE 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).

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*

10 min break

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.

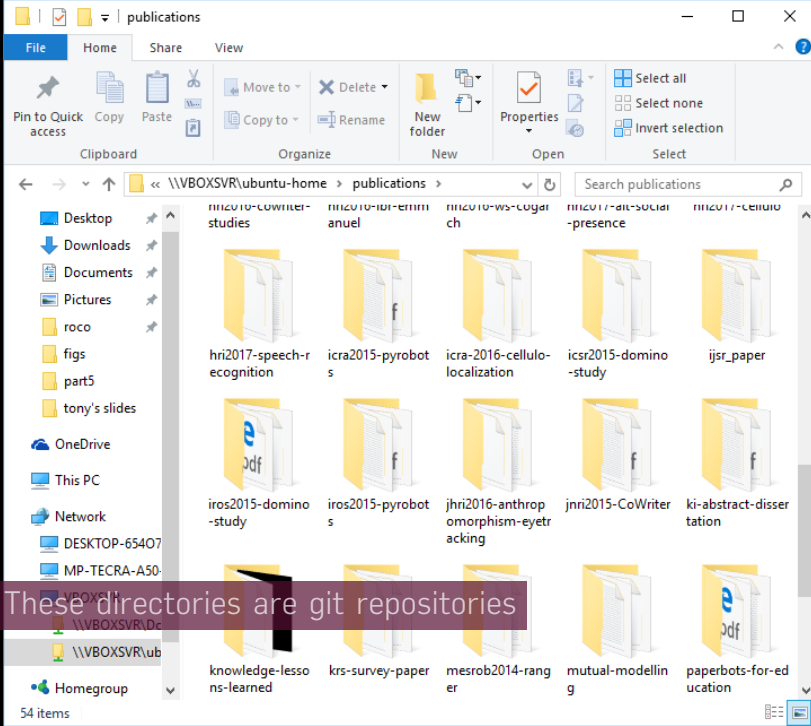
Today, I want to share our results. In addition, we're announcing the next steps in our GVFS journey for customers, including expanded open sourcing to start taking contributions and improving how it works for us at Microsoft, as well as for partners and customers.

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

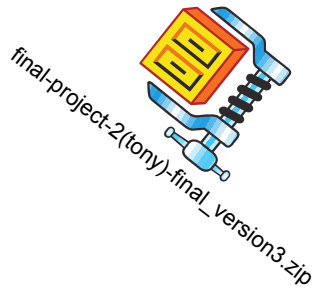
Select: Select all, Select none, Invert selection

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

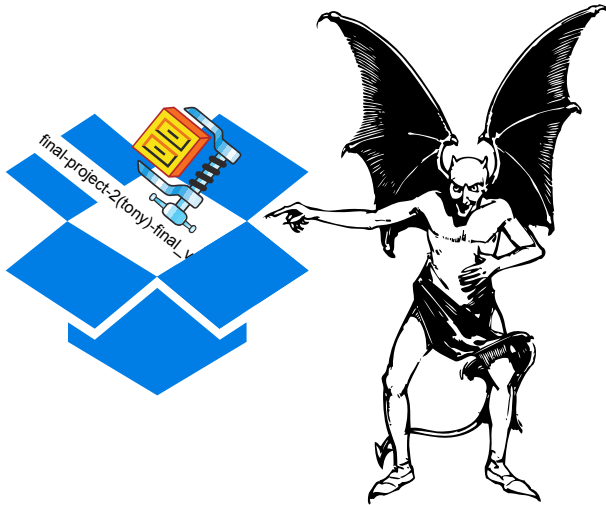
| | Name | Date modified | Type | Size |
|---------------|------------------------|--------------------|--------------------|------|
| Desktop | .git | 11/5/2014 1:29 PM | File folder | |
| Downloads | figures | 11/5/2014 12:39 PM | File folder | |
| Documents | lemaignan2014being.pdf | 11/5/2014 1:29 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 teammates...









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!

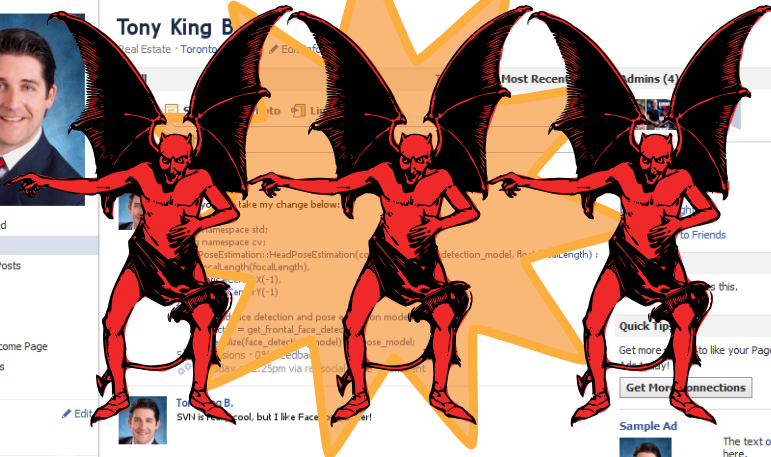
7

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)

CODE VERSIONING

WHY VERSIONING?

- The history of your development/document
- Compare the current code with an older version
- Roll-back to previous versions (think 'undo on steroids')
- 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.

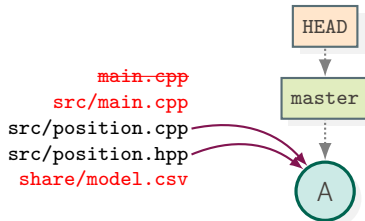
ATOMIC COMMITS

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

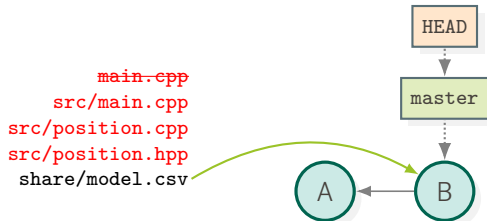
Atomic commit

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

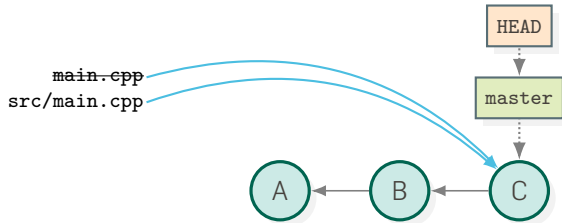
```
main.cpp
src/main.cpp
src/position.cpp
src/position.hpp
share/model.csv
```



```
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
```

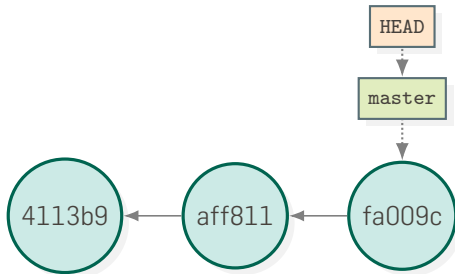
Move main.cpp to src/

```
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

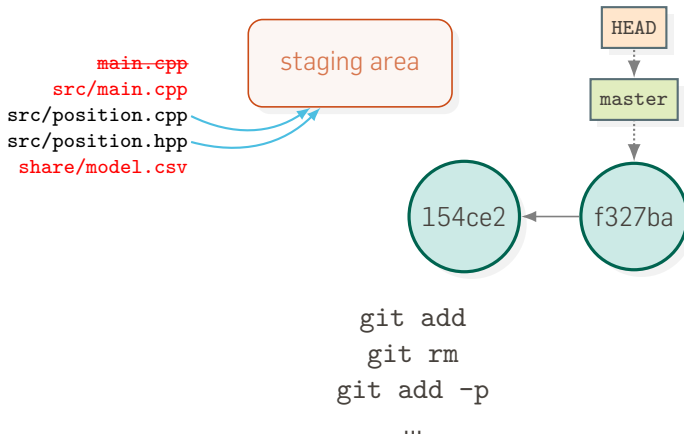
But why do we have to manually tell Git what files to add or remove?

THE STAGING AREA

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

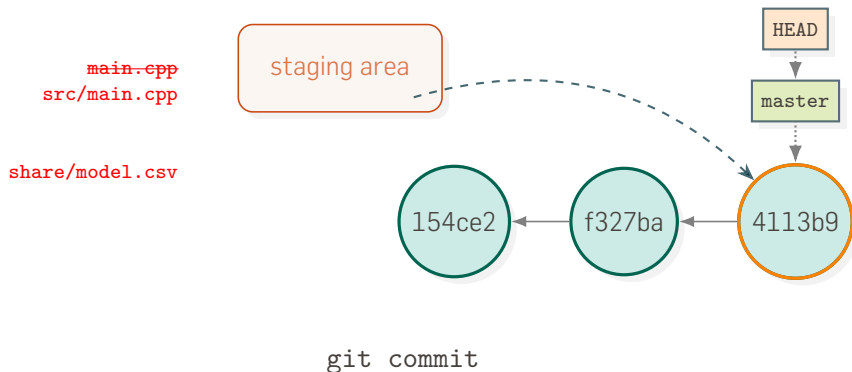
THE STAGING AREA

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



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

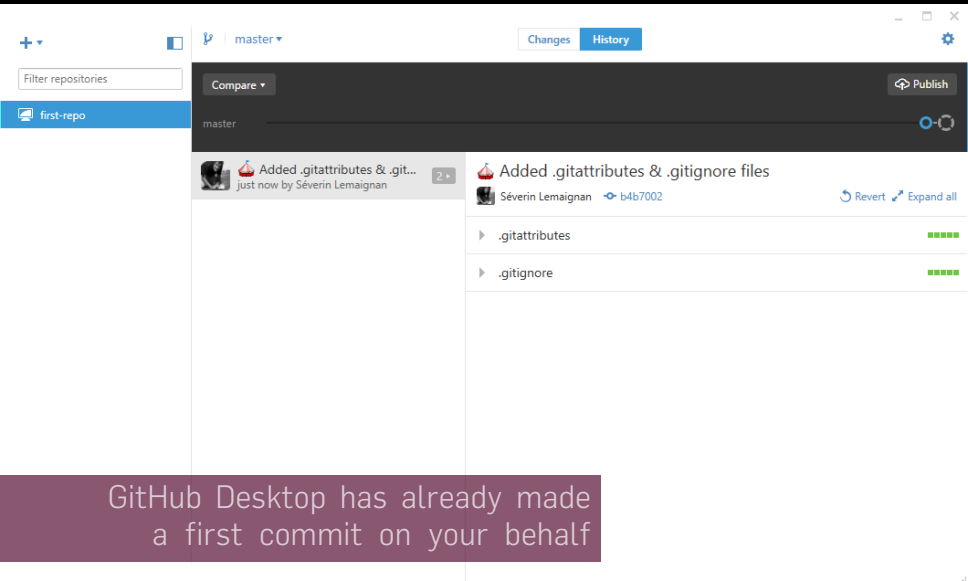
✓

Create repository

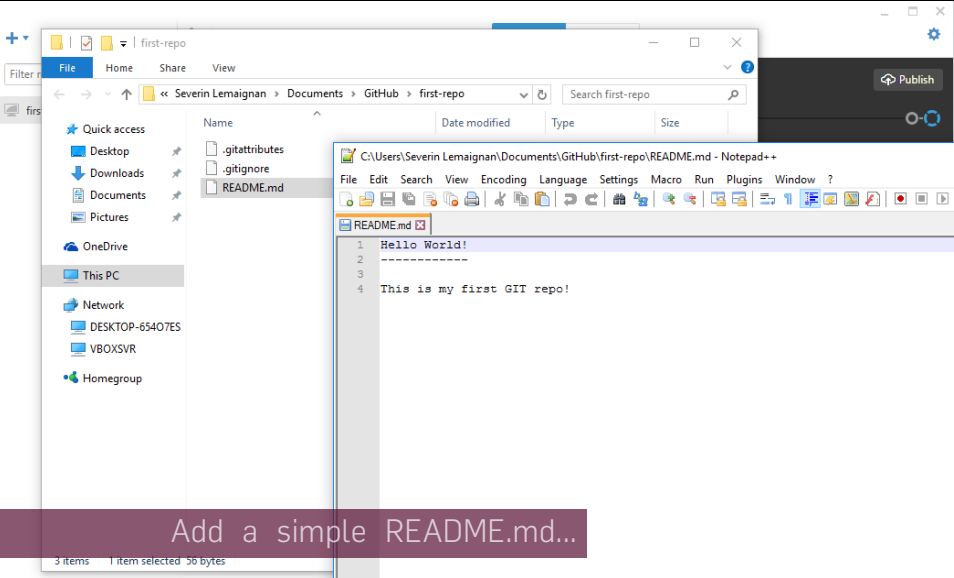


Get started by adding a repository.

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, showing 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 details show two files: ".gitattributes" and ".gitignore", both with green status indicators. A purple banner at the bottom of the image reads "Open the repo in Windows Explorer".



Add a simple README.md...

+

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

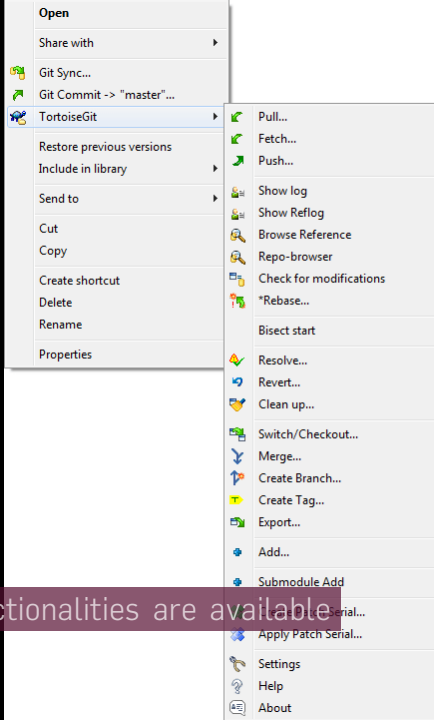


skip-worktree.cpp

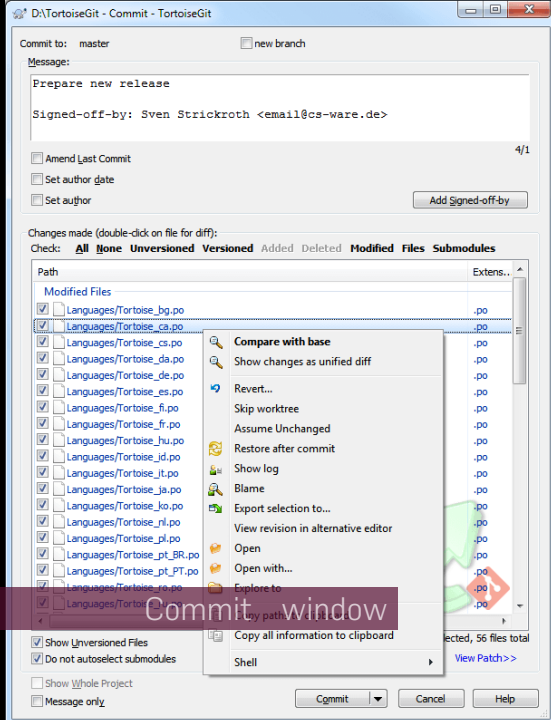


non-versioned.cpp

Files' status appear as icons



All the functionalities are available



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 | | | | |

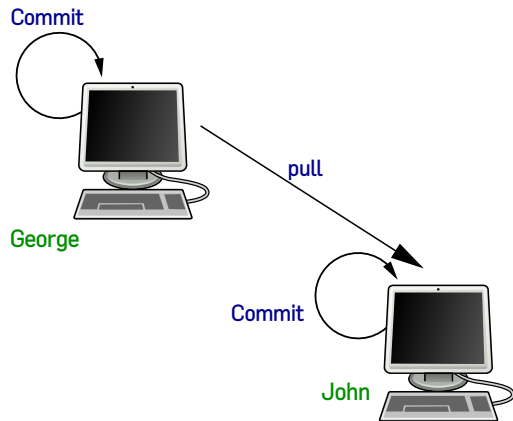
What should I track here?

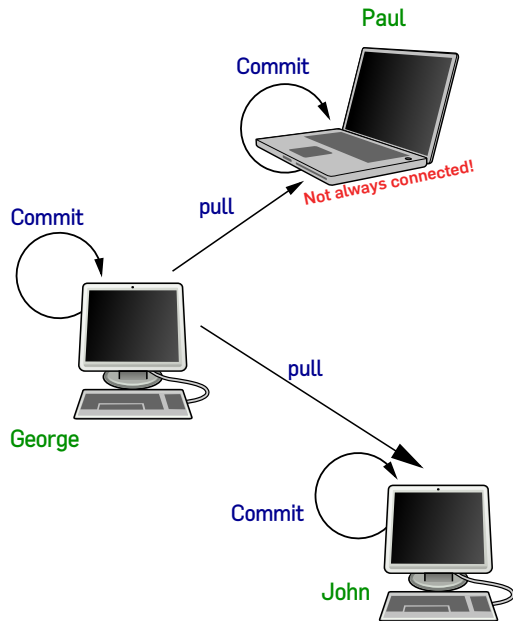
COLLABORATING

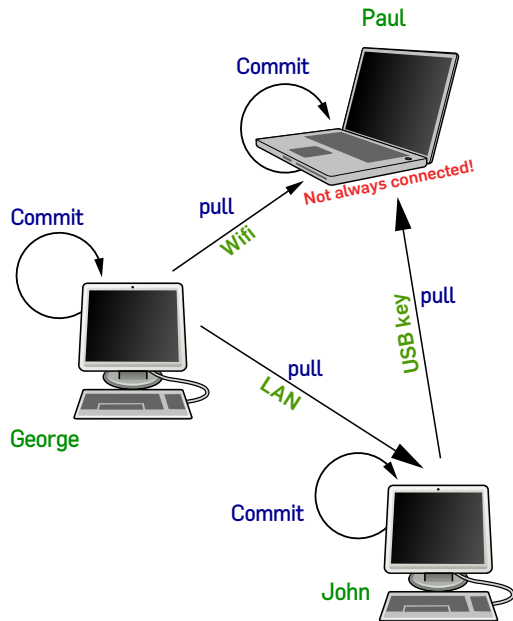
Commit

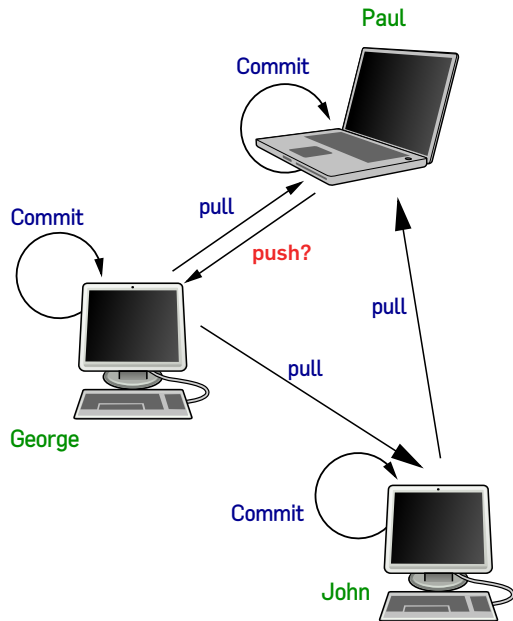


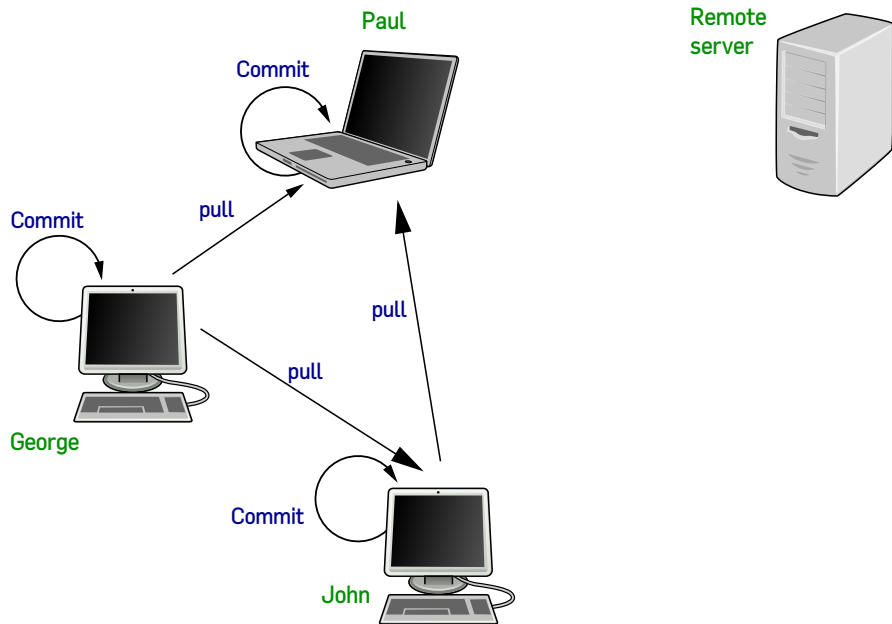
George



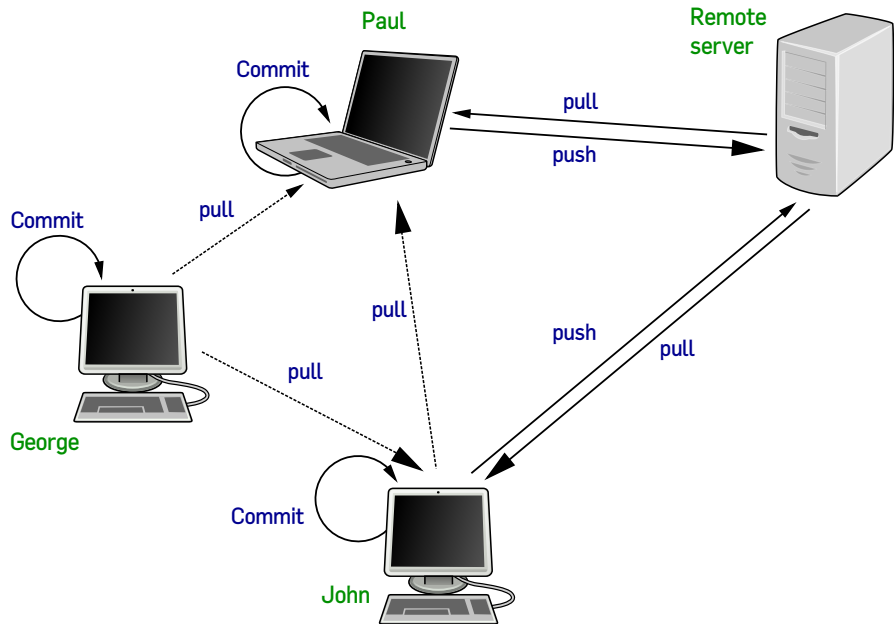


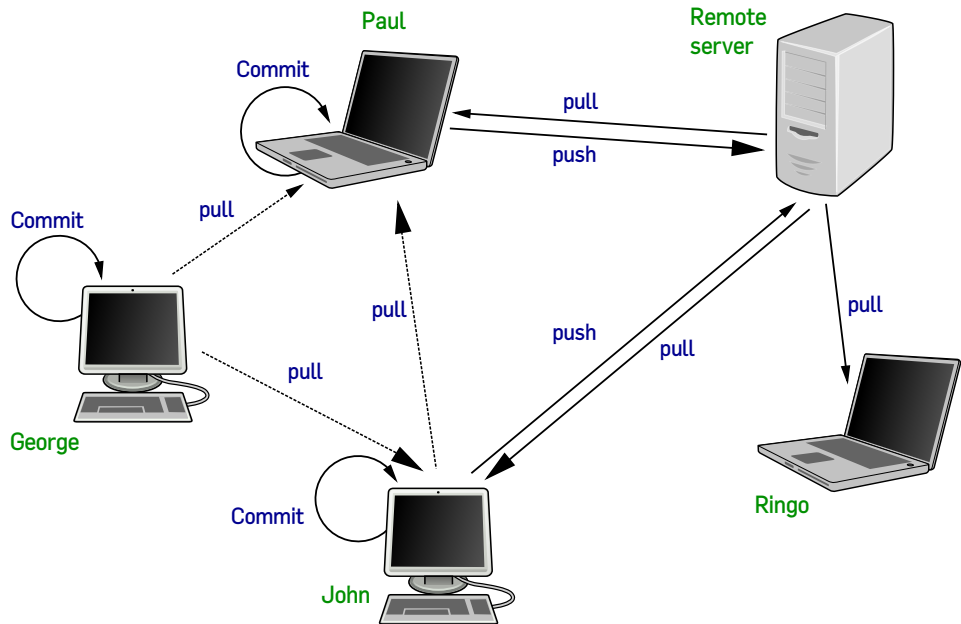




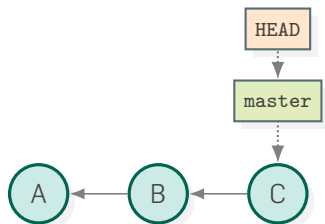


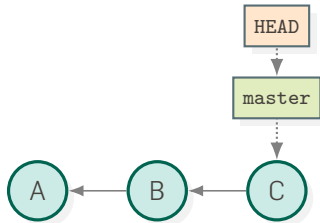
Based on a figure by M. Herrb, CC-BY-SA 3.0



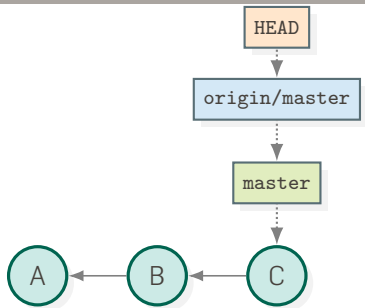


Based on a figure by M. Herrb, CC-BY-SA 3.0



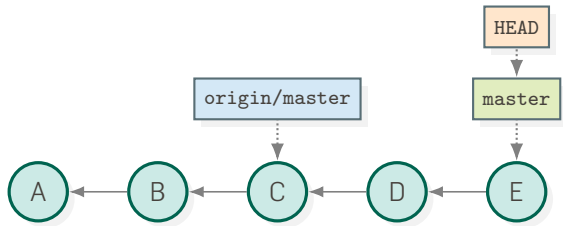


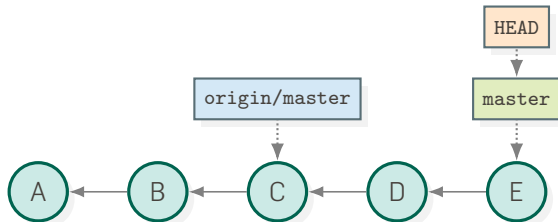
```
git remote add origin git@github.com:user/repo.git
git remote add john-usb E:\john_repo
git remote add ftp-origin ftp://host.xz/path/to/repo.git/
...
```



```
git push origin master
```

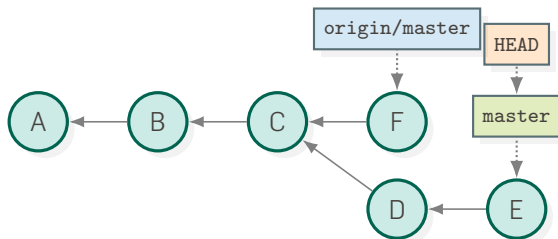
(or simply `git push`)

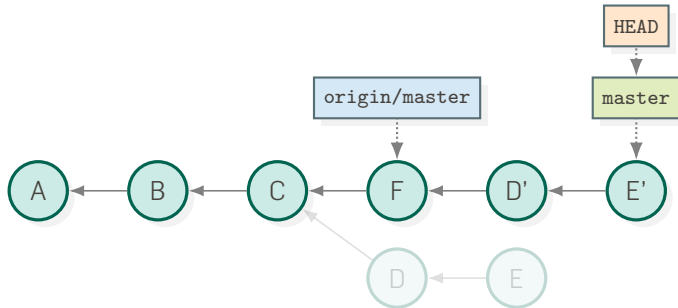




What happened on our remote? Let's have a look...

```
git fetch origin
```





```
git rebase origin/master
```

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



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) report...

```
> 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 lab-journal.md

Falling back to patching base and 3-way merge...

Auto-merging lab-journal.md

CONFLICT (content): Merge conflict in lab-journal.md

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-apply/patch

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

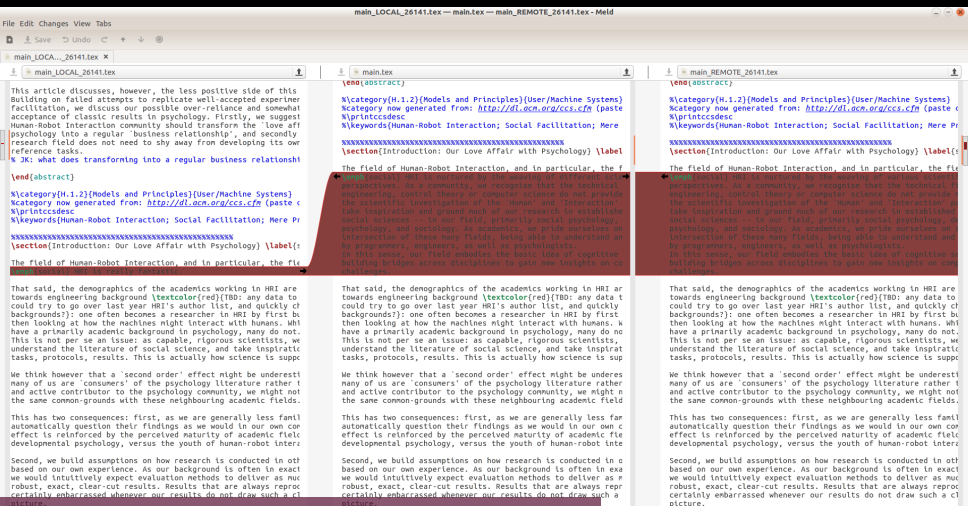
If you prefer to skip this patch, run **"git rebase --skip"** instead.

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

A conflict happens when two modifications of a given file overlap

Two persons can modify the same file at the same time, as long as they do not work on the same region of the file.

```
> 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

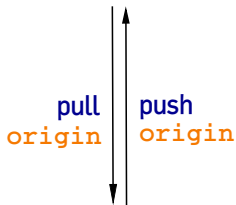
<https://github.com/morse>



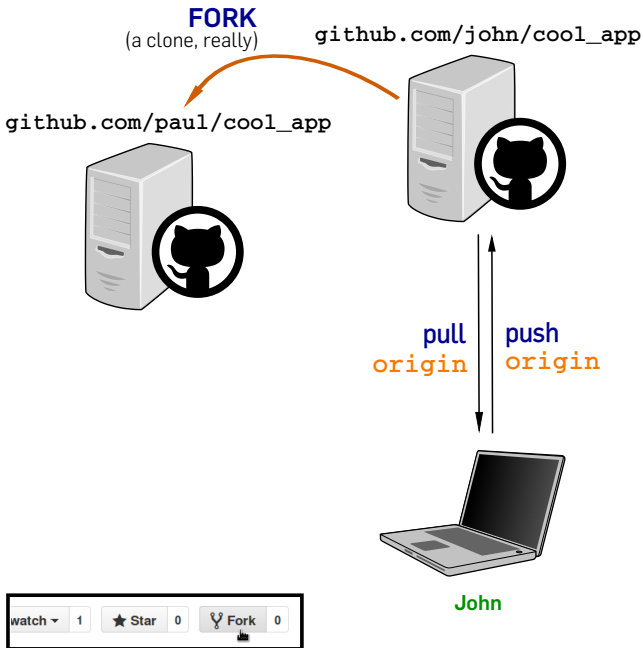
Download ZIP

| | |
|--|---|
| adeegroote [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

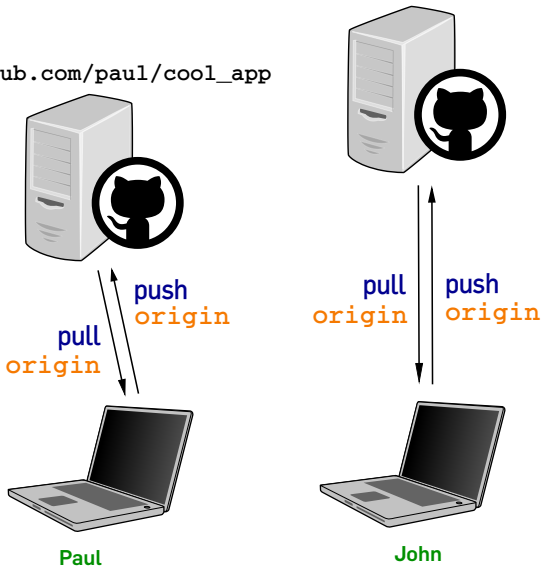


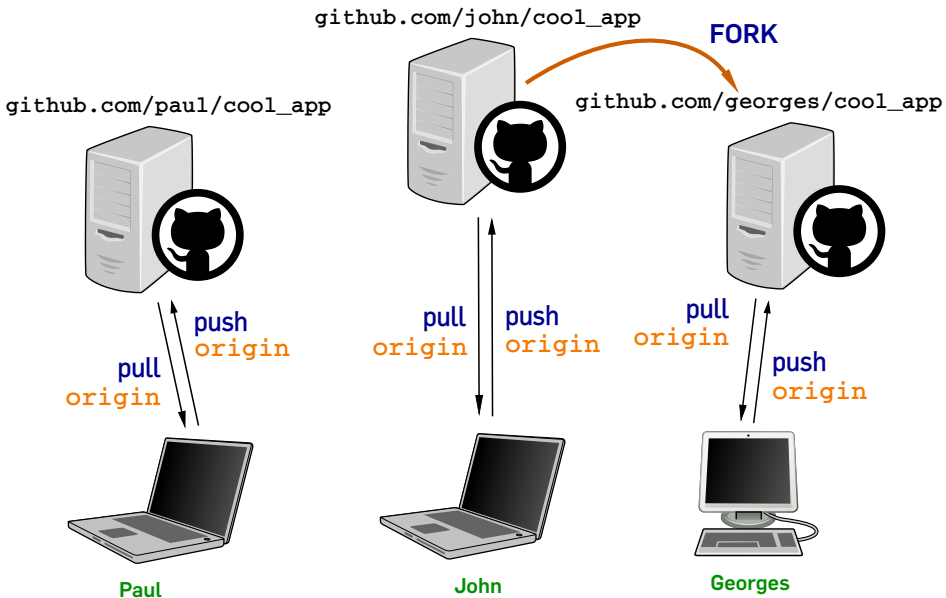
John



github.com/john/cool_app

github.com/paul/cool_app

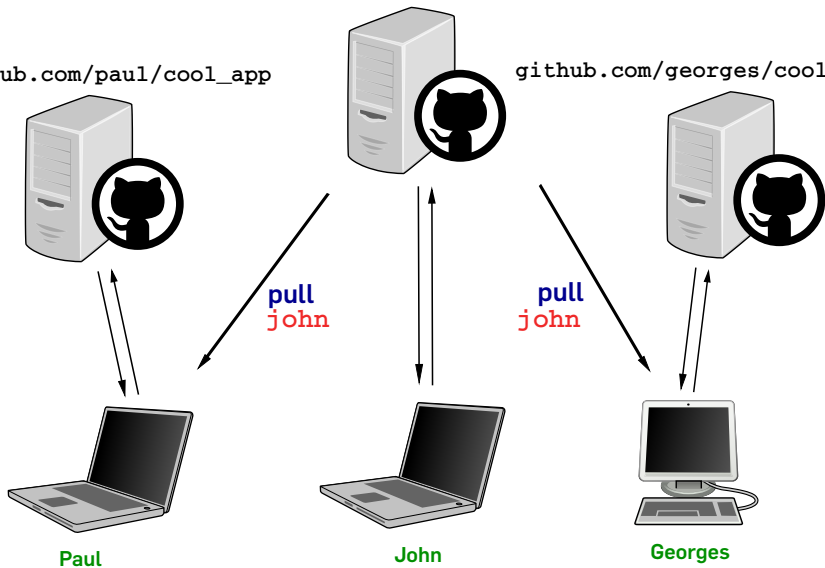


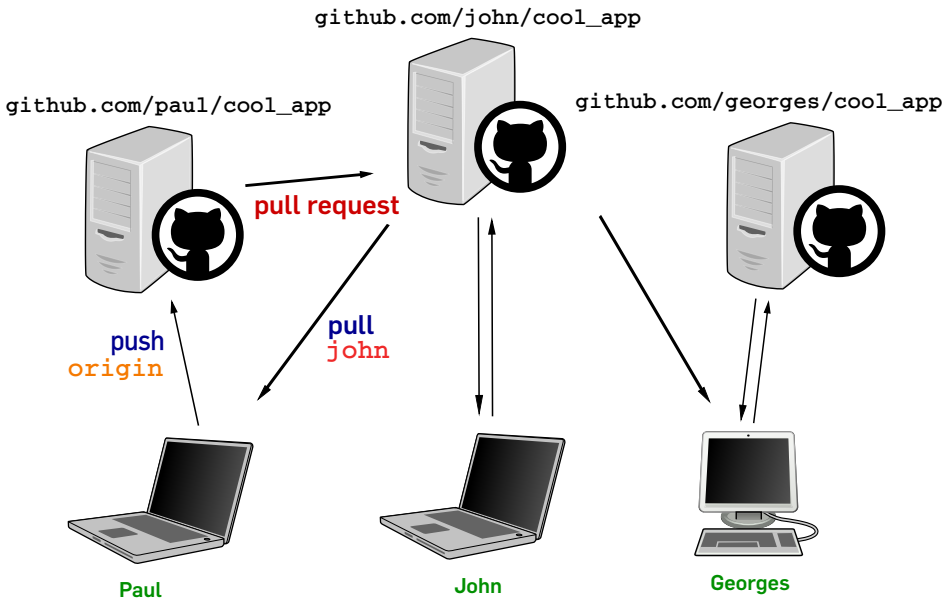


github.com/john/cool_app

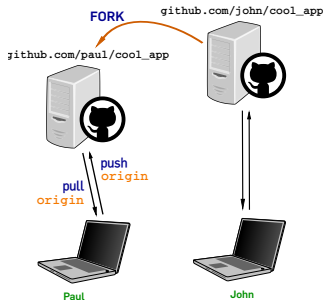
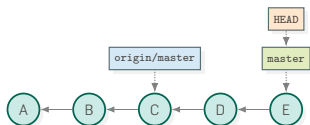
github.com/paul/cool_app

github.com/georges/cool_app



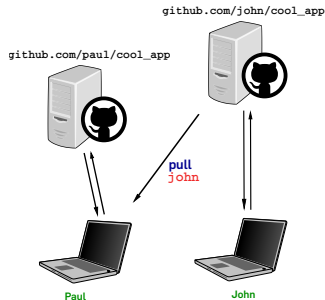
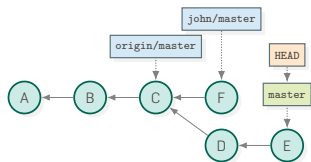


WHAT HAPPENED EXACTLY?



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

WHAT HAPPENED EXACTLY?

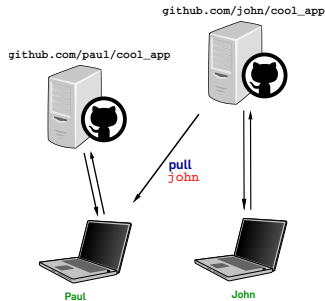
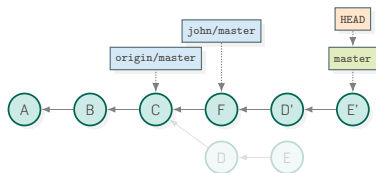


He would like to propose his changes to John

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?

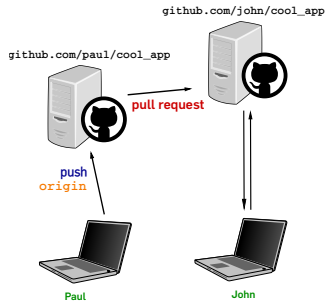
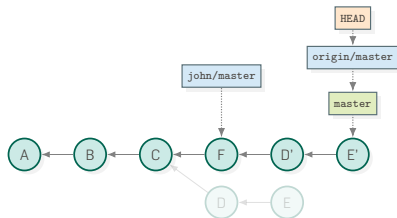


Paul rebases his master branch on John's one:

```
git rebase john/master
```

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

WHAT HAPPENED EXACTLY?

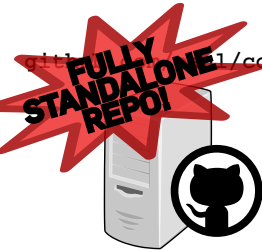


He pushes his commits to his own GitHub account:

```
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 everythg!

`git reflog`

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:

- o **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 DifferentialDriveVelocity, it 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

That's all, folks!

Questions:

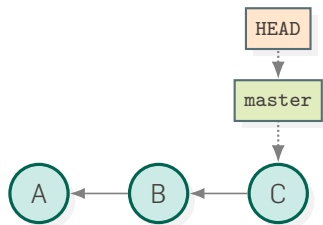
Portland Square B316 or **severin.lemaignan@plymouth.ac.uk**

Slides:

github.com/severin-lemaignan/module-introduction-sensors-actuators

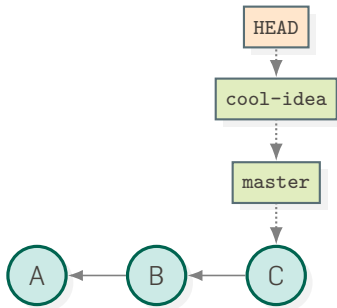
WORKING WITH BRANCHES

BRANCHES



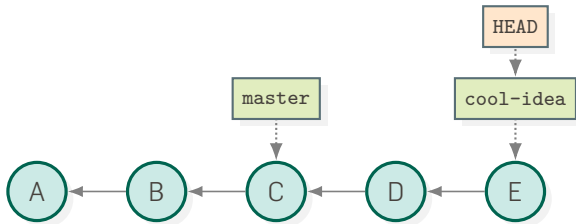
What if...?

BRANCHES

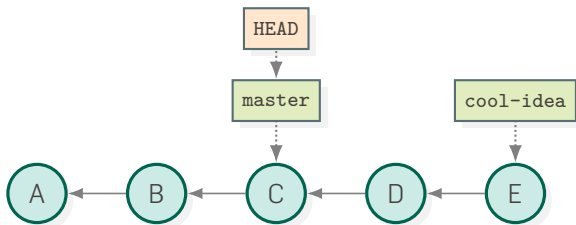


```
git checkout -b cool-idea
```

BRANCHES

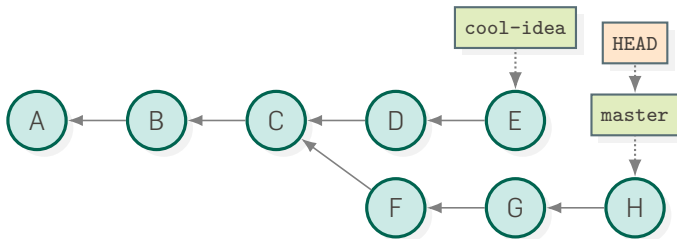


BRANCHES



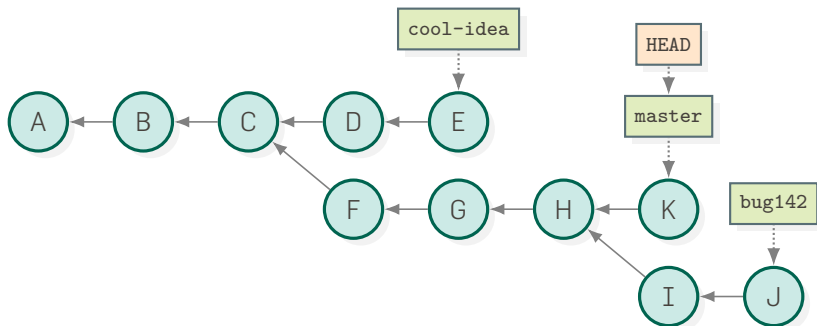
Let go back to serious stuff!
`git checkout master`

BRANCHES



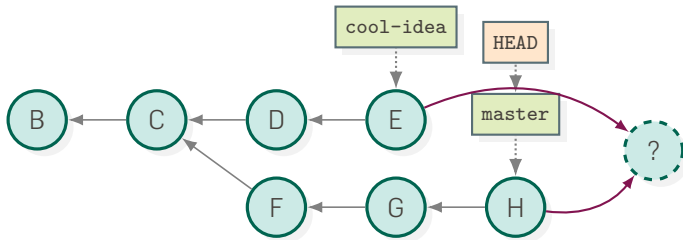
The branch name is an alias for the tip of the current branch

BRANCHES



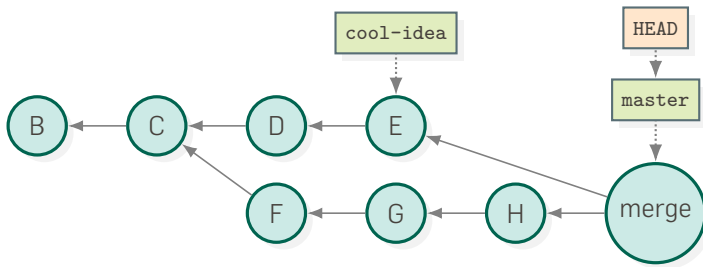
⇒ branches are very cheap
+10 of them at a given time it not uncommon

MERGING BRANCHES



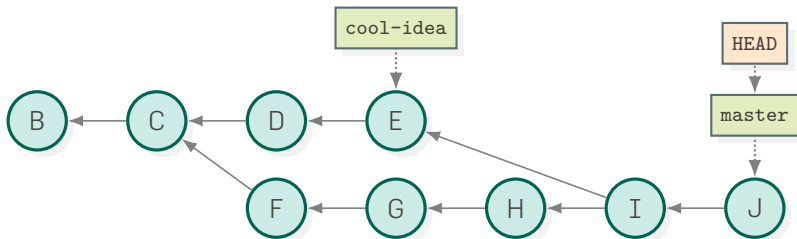
Two options: **merging** and **rebasing**

MERGING BRANCHES



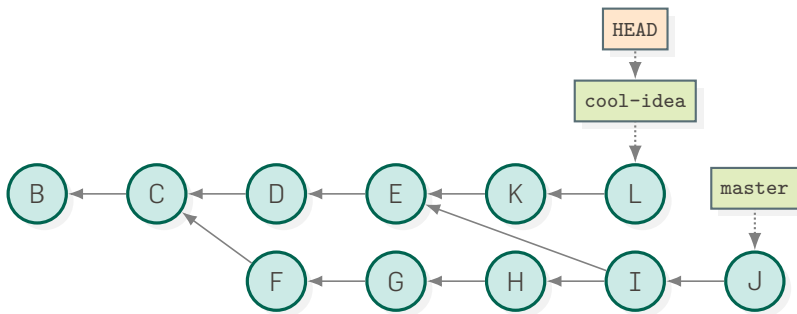
Merging
`git merge cool-idea`

MERGING BRANCHES



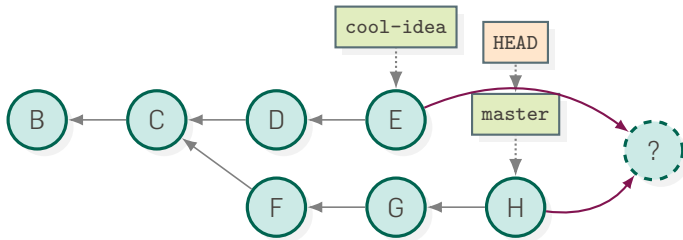
```
git commit
```

MERGING BRANCHES

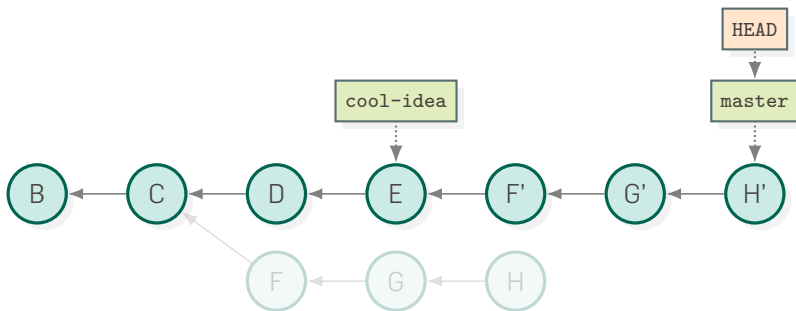


```
git checkout cool-idea  
git commit  
...etc.
```

REBASING BRANCHES



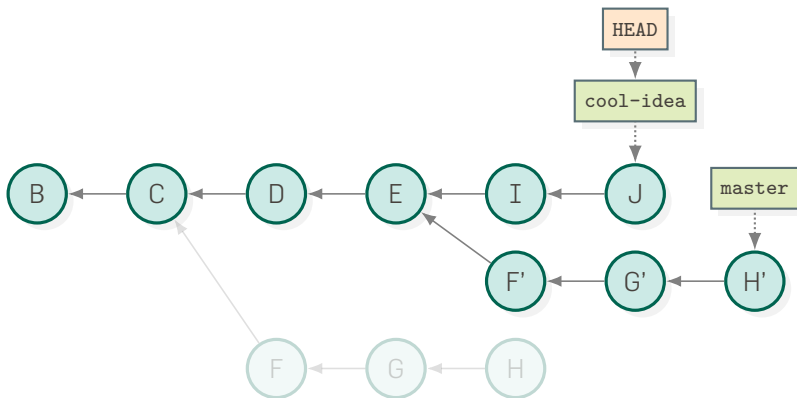
REBASING BRANCHES



Rebasing

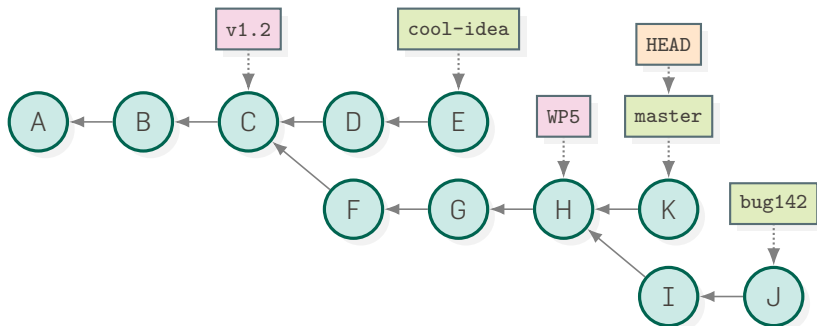
`git rebase cool-idea`

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

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 "Other branches" with `master` listed below. The main editor displays a C++ file named `main.cpp` with the following code:

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

The interface also shows a "Changes" tab, a "History" tab, and a "Publish" button. The commit message is "Switch to numerical coordinates" by Séverin Lemaignan (8f6ef0f). The commit is dated 9:51 AM.

We can jump between branches...

Filter repositories

first-repo

master

Changes

History

Update from numerical_coordinates

View branch

Publish

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

13 13

14 14

15 15

16 20

17 21

18 22

19 +

20 23

21 24

22 25

23 26

24 27

25 28

26 29

27 30

28 31

29 32

30 33

31 34

32 35

33 36

34 37

35 38

36 39

37 40

38 41

39 42

40 43

41 44

42 45

43 46

44 47

45 48

46 49

47 50

48 51

49 52

50 53

51 54

52 55

53 56

54 57

55 58

56 59

57 60

58 61

59 62

60 63

61 64

62 65

63 66

64 67

65 68

66 69

67 70

68 71

69 72

70 73

71 74

72 75

73 76

74 77

75 78

76 79

77 80

78 81

79 82

80 83

81 84

82 85

83 86

84 87

85 88

86 89

87 90

88 91

89 92

90 93

91 94

92 95

93 96

94 97

95 98

96 99

97 100

98 101

99 102

100 103

101 104

102 105

103 106

104 107

105 108

106 109

107 110

108 111

109 112

110 113

111 114

112 115

113 116

114 117

115 118

116 119

117 120

118 121

119 122

120 123

121 124

122 125

123 126

124 127

125 128

126 129

127 130

128 131

129 132

130 133

131 134

132 135

133 136

134 137

135 138

136 139

137 140

138 141

139 142

140 143

141 144

142 145

143 146

144 147

145 148

146 149

147 150

148 151

149 152

150 153

151 154

152 155

153 156

154 157

155 158

156 159

157 160

158 161

159 162

160 163

161 164

162 165

163 166

164 167

165 168

166 169

167 170

168 171

169 172

170 173

171 174

172 175

173 176

174 177

175 178

176 179

177 180

178 181

179 182

180 183

181 184

182 185

183 186

184 187

185 188

186 189

187 190

188 191

189 192

190 193

191 194

192 195

193 196

194 197

195 198

196 199

197 200

198 201

199 202

200 203

201 204

202 205

203 206

204 207

205 208

206 209

207 210

208 211

209 212

210 213

211 214

212 215

213 216

214 217

215 218

216 219

217 220

218 221

219 222

220 223

221 224

222 225

223 226

224 227

225 228

226 229

227 230

228 231

229 232

230 233

231 234

232 235

233 236

234 237

235 238

236 239

237 240

238 241

239 242

240 243

241 244

242 245

243 246

244 247

245 248

246 249

247 250

248 251

249 252

250 253

251 254

252 255

253 256

254 257

255 258

256 259

257 260

258 261

259 262

260 263

261 264

262 265

263 266

264 267

265 268

266 269

267 270

268 271

269 272

270 273

271 274

272 275

273 276

274 277

275 278

276 279

277 280

278 281

279 282

280 283

281 284

282 285

283 286

284 287

285 288

286 289

287 290

288 291

289 292

290 293

291 294

292 295

293 296

294 297

295 298

296 299

297 300

298 301

299 302

300 303

301 304

302 305

303 306

304 307

305 308

306 309

307 310

308 311

309 312

310 313

311 314

312 315

313 316

314 317

315 318

316 319

317 320

318 321

319 322

320 323

321 324

322 325

323 326

324 327

325 328

326 329

327 330

328 331

329 332

330 333

331 334

332 335

333 336

334 337

335 338

336 339

337 340

338 341

339 342

340 343

341 344

342 345

343 346

344 347

345 348

346 349

347 350

348 351

349 352

350 353

351 354

352 355

353 356

354 357

355 358

356 359

357 360

358 361

359 362

360 363

361 364

362 365

363 366

364 367

365 368

366 369

367 370

368 371

369 372

370 373

371 374

372 375

373 376

374 377

375 378

376 379

377 380

378 381

379 382

380 383

381 384

382 385

383 386

384 387

385 388

386 389

387 390

388 391

389 392

390 393

391 394

392 395

393 396

394 397

395 398

396 399

397 400

398 401

399 402

400 403

401 404

402 405

403 406

404 407

405 408

406 409

407 410

408 411

409 412

410 413

411 414

412 415

413 416

414 417

415 418

416 419

417 420

418 421

419 422

420 423

421 424

422 425

423 426

424 427

425 428

426 429

427 430

428 431

429 432

430 433

431 434

432 435

433 436

434 437

435 438

436 439

437 440

438 441

439 442

440 443

441 444

442 445

443 446

444 447

445 448

446 449

447 450

448 451

449 452

450 453

451 454

452 455

453 456

454 457

455 458

456 459

457 460

458 461

459 462

460 463

461 464

462 465

463 466

464 467

465 468

466 469

467 470

468 471

469 472

470 473

471 474

472 475

473 476

474 477

475 478

476 479

477 480

478 481

479 482

480 483

481 484

482 485

483 486

484 487

485 488

486 489

487 490

488 491

489 492

490 493

491 494

492 495

493 496

494 497

495 498

496 499

497 500

498 501

499 502

500 503

501 504

502 505

503 506

504 507

505 508

506 509

507 510

508 511

509 512

510 513

511 514

512 515

513 516

514 517

515 518

516 519

517 520

518 521

519 522

520 523

521 524

522 525

523 526

524 527

525 528

526 529

527 530

528 531

529 532

530 533

531 534

532 535

533 536

534 537

535 538

536 539

537 540

538 541

539 542

540 543

541 544

542 545

543 546

544 547

545 548

546 549

547 550

548 551

549 552

550 553

551 554

552 555

553 556

554 557

555 558

556 559

557 560

558 561

559 562

560 563

561 564

562 565

563 566

564 567

565 568

566 569

567 570

568 571

569 572

570 573

571 574

572 575

573 576

574 577

575 578

576 579

577 580

578 581

579 582

580 583

581 584

582 585

583 586

584 587

585 588

586 589

587 590

588 591

589 592

590 593

591 594

592 595

593 596

594 597

595 598

596 599

597 600

598 601

599 602

600 603

601 604

602 605

603 606

604 607

605 608

606 609

607 610

608 611

609 612

610 613

611 614

612 615

613 616

614 617

615 618

616 619

617 620

618 621

619 622

620 623

621 624

622 625

623 626

624 627

625 628

626 629

627 630

628 631

629 632

630 633

631 634

632 635

633 636

634 637

635 638

636 639

637 640

638 641

639 642

640 643

641 644

642 645

643 646

644 647

645 648

646 649

647 650

648 651

649 652

650 653

651 654

652 655

653 656

654 657

655 658

656 659

657 660

658 661

659 662

660 663

661 664

662 665

663 666

664 667

665 668

666 669

667 670

668 671

669 672

670 673

671 674

672 675

673 676

674 677

675 678

676 679

677 680

678 681

679 682

680 683

681 684

682 685

683 686

684 687

685 688

686 689

687 690

688 691

689 692

690 693

691 694

692 695

693 696

694 697

695 698

696 699

697 700

698 701

699 702

700 703

701 704

702 705

703 706

704 707

705 708

706 709

707 710

708 711

709 712

710 713

711 714

712 715

713 716

714 717

715 718

716 719

717 720

718 721

719 722

720 723

721 724

722 725

723 726

724 727

725 728

726 729

727 730

728 731

729 732

730 733

731 734

732 735

733 736

734 737

735 738

736 739

737 740

738 741

739 742

740 743

741 744

742 745

743 746

744 747

745 748

746 749

747 750

748 751

749 752

750 753

751 754

752 755

753 756

754 757

755 758

756 759

757 760

758 761

759 762

760 763

761 764

762 765

763 766

764 767

765 768

766 769

767 770

768 771

769 772

770 773

771 774

772 775

773 776

774 777

775 778

776 779

777 780

778 781

779 782

780 783

781 784

782 785

783 786

784 787

785 788

786 789

787 790

788 791

789 792

790 793

791 794

792 795

793 796

794 797

795 798

796 799

797 800

798 801

799 802

800 803

801 804

802 805

803 806

804 807

805 808

806 809

807 810

808 811

809 812

810 813

811 814

812 815

813 816

814 817

815 818

816 819

817 820

818 821

819 822

820 823

821 824

822 825

823 826

824 827

825 828

826 829

827 830

828 831

829 832

830 833

831 834

832 835

833 836

834 837

835 838

836 839

837 840

838 841

839 842

840 843

841 844

842 845

843 846

844 847

845 848

846 849

847 850

848 851

849 852

850 853

851 854

852 855

853 856

854 857

855 858

856 859

857 860

858 861

859 862

860 863

861 864

862 865

863 866

864 867

865 868

866 869

867 870

868 871

869 872

870 873

871 874

872 875

873 876

874 877

875 878

876 879

877 880

878 881

879 882

880 883

881 884

882 885

883 886

884 887

885 888

886 889

887 890

888 891

889 892

890 893

891 894

892 895

893 896

894 897

895 898

896 899

897 900

898 901

899 902

900 903

901 904

902 905

903 906

904 907

905 908

906 909

907 910

908 911

909 912

910 913

911 914

912 915

913 916

914 917

915 918

916 919

917 920

918 921

919 922

920 923

921 924

922 925

923 926

924 927

925 928

926 929

927 930

928 931

929 932

930 933

931 934

932 935

933 936

934 937

935 938

936 939

937 940

938 941

939 942

940 943

941 944

942 945

943 946

944 947

945 948

946 949

947 950

948 951

949 952

950 953

951 954

952 955

953 956

954 957

955 958

956 959

957 960

958 961

959 962

960 963

961 964

962 965

963 966

964 967

965 968

966 969

967 970

968 971

969 972

970 973

971 974

972 975

973 976

974 977

975 978

976 979

977 980

978 981

979 982

980 983

981 984

982 985

983 986

984 987

985 988

986 989

987 990

988 991

989 992

990 993

991 994

992 995

993 996

994 997

995 998

996 999

997 1000

998 1001

999 1002

1000 1003

1001 1004

1002 1005

1003 1006

1004 1007

1005 1008

1006 1009

1007 1010

1008 1011

1009 1012

1010 1013

1011 1014

1012 1015

1013 1016

1014 1017

1015 1018

1016 1019

1017 1020

1018 1021

1019 1022

1020 1023

1021 1024

1022 1025

1023 10



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    19    -    cout << " A B C " << endl;  
20    20    +    cout << " 1 2 3 " << endl;  
21    21        cout << " -----" << endl;  
22    22        cout << "1";
```

We switch back to numerical_co-
ordinates and merge in master



numerical_coordinates ▾

Changes ●

History



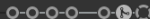
Filter repositories

first-repo

Compare ▾

Publish

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   }
37  +   done = true;
38  +
39  +   cout << endl << "Enter X coordinate (A, B or C):";
40  +   cin >> x_char;
40  +   x = (x_char == 'A' ? 0 : (x_char == 'B' ? 1 : 2));
```

The merge commit is reflected
in the history of the branch