Version control

(using git)

Danny Awty-Carroll

May 16, 2018

IBERS
Athrofa y Gwyddorau Biolegol, Amgylcheddol a Gwledig
Institute of Biological, Environmental and Rural Sciences

Sections

This will focus on using git in windows with a UI

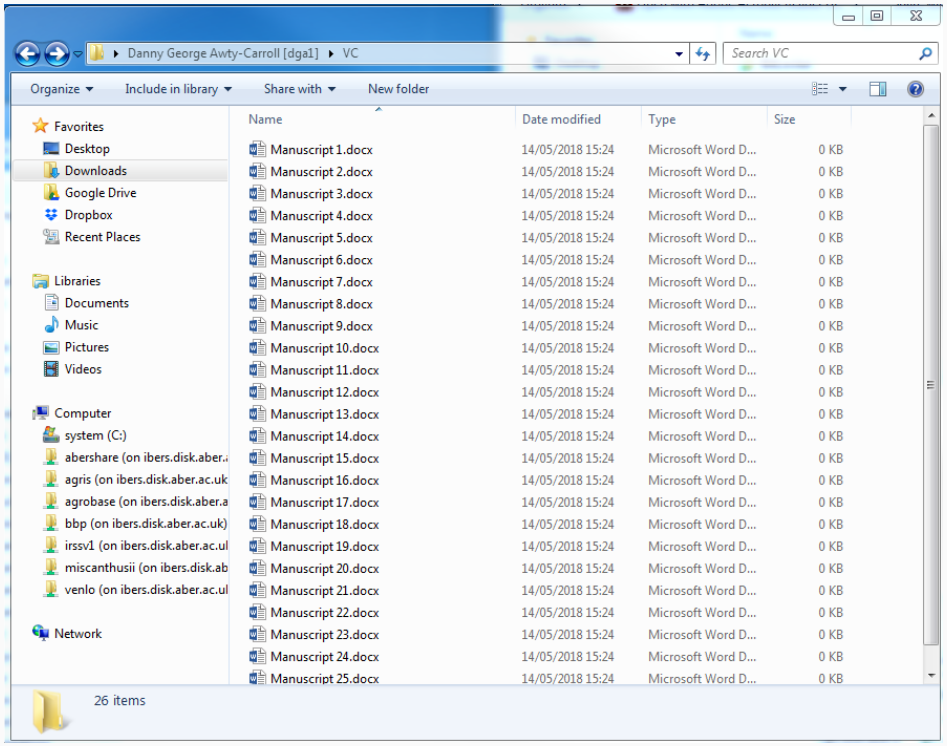
1. Why version control?
2. Version control system
3. Platform
4. UI
5. Using git
6. Questions & Demo

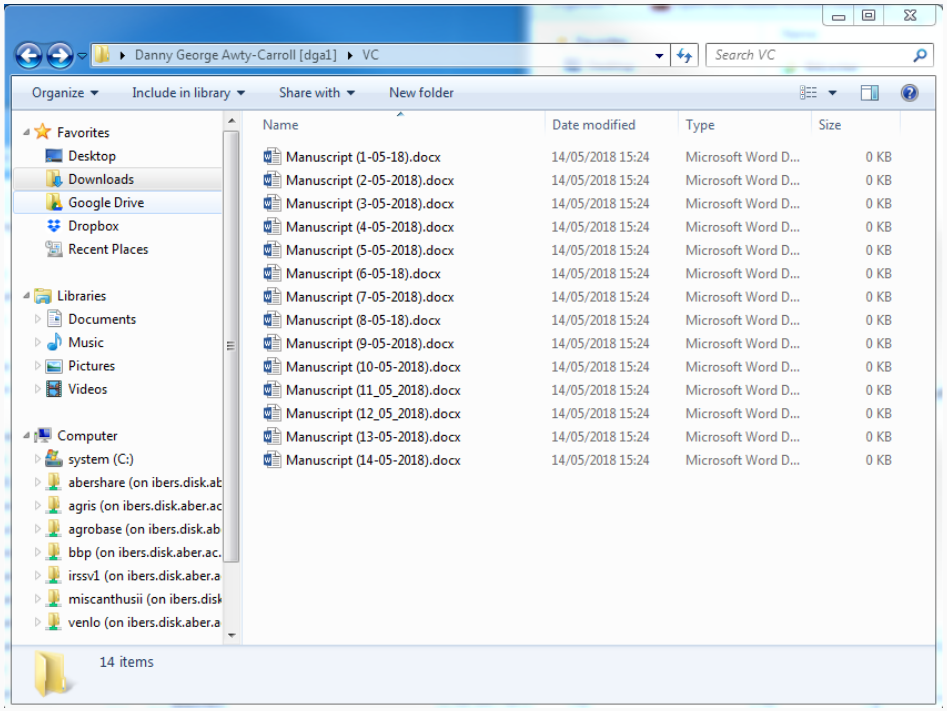
Why version control?

What is version control

This is just the management of versions of a document.

One document through time.





All of us use some version control

Where things get complicated

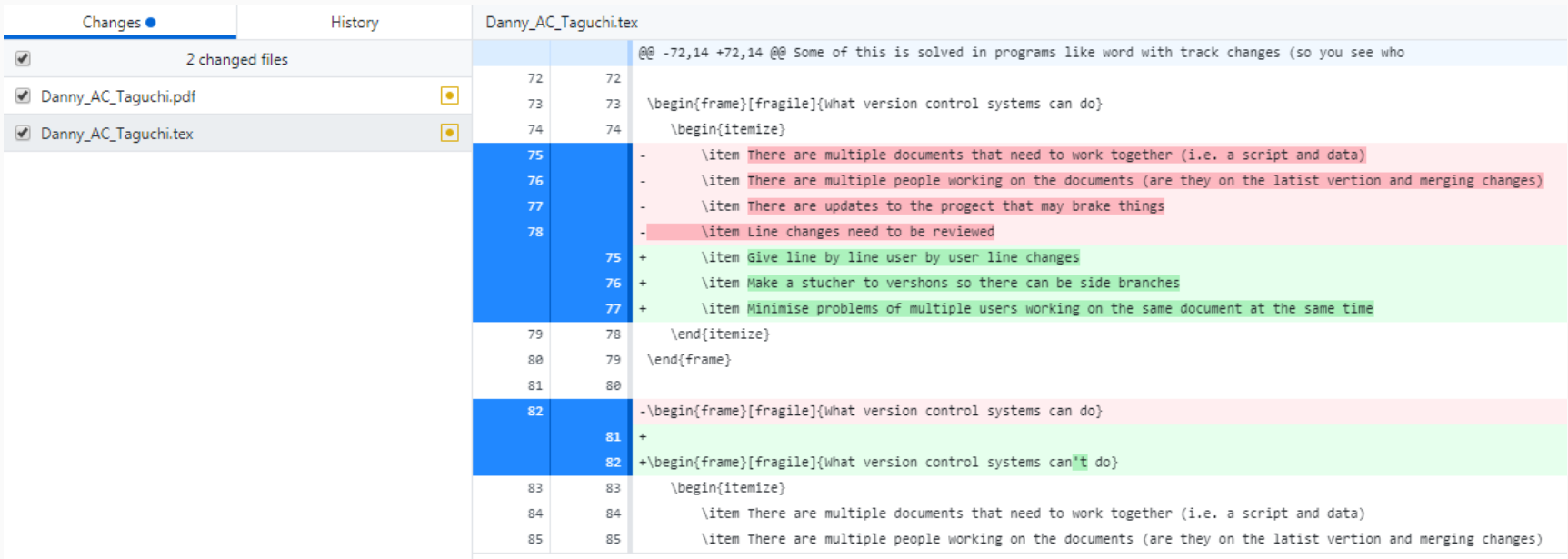
Numbering or dating documents works OK but can fall down when

- There are multiple documents that need to work together (i.e. a script and data)
- There are multiple people working on the documents (are they on the latest version and merging changes)
- There are updates to the project that may break things
- Line changes need to be reviewed

Some of this is solved in programs like word with track changes (so you see who altered what when)

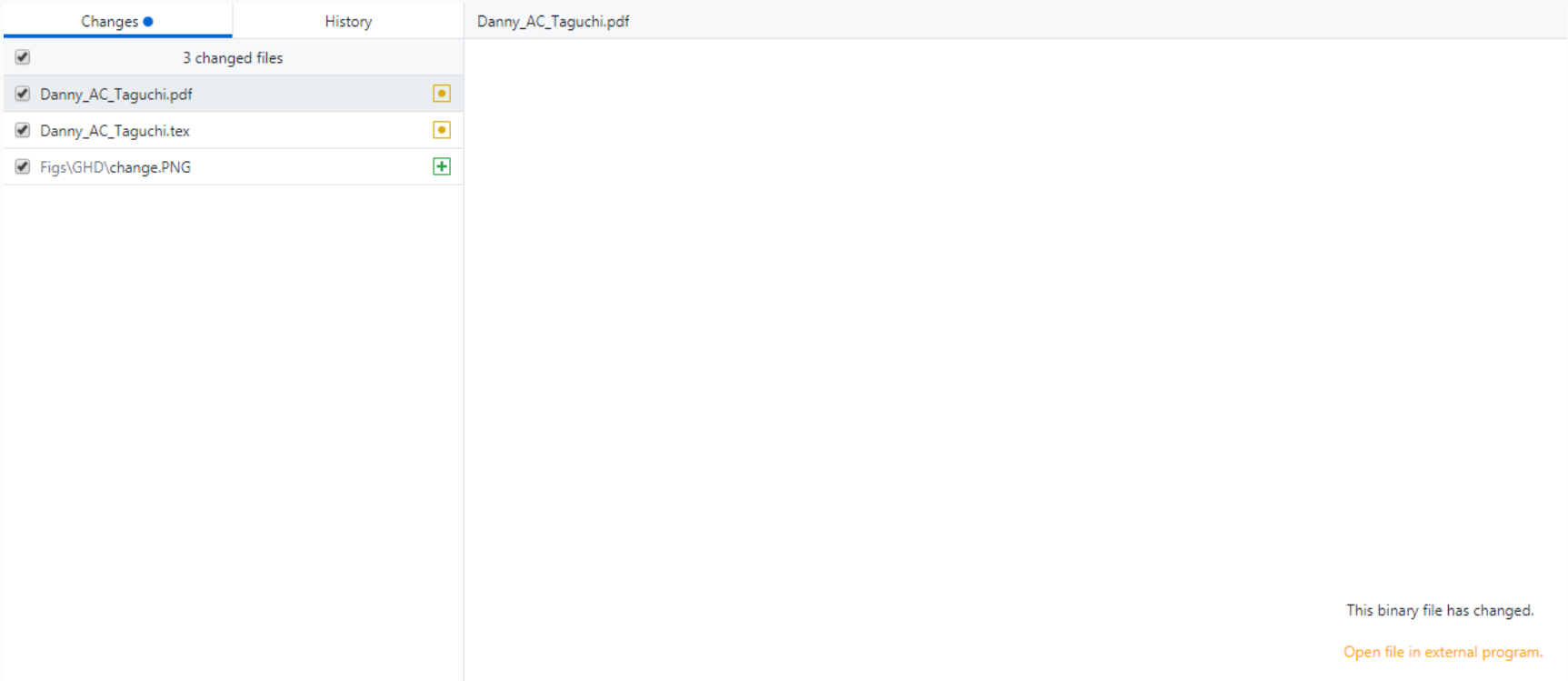
What version control systems can do

- Give line by line changes linked to a user (which are keeped forever)
- Makes a structure to the version control so there can be side branches (this is where the project can take a detour and be merged back in later)
- Minimise problems of multiple users working on the same document at the same time



What version control systems can't do

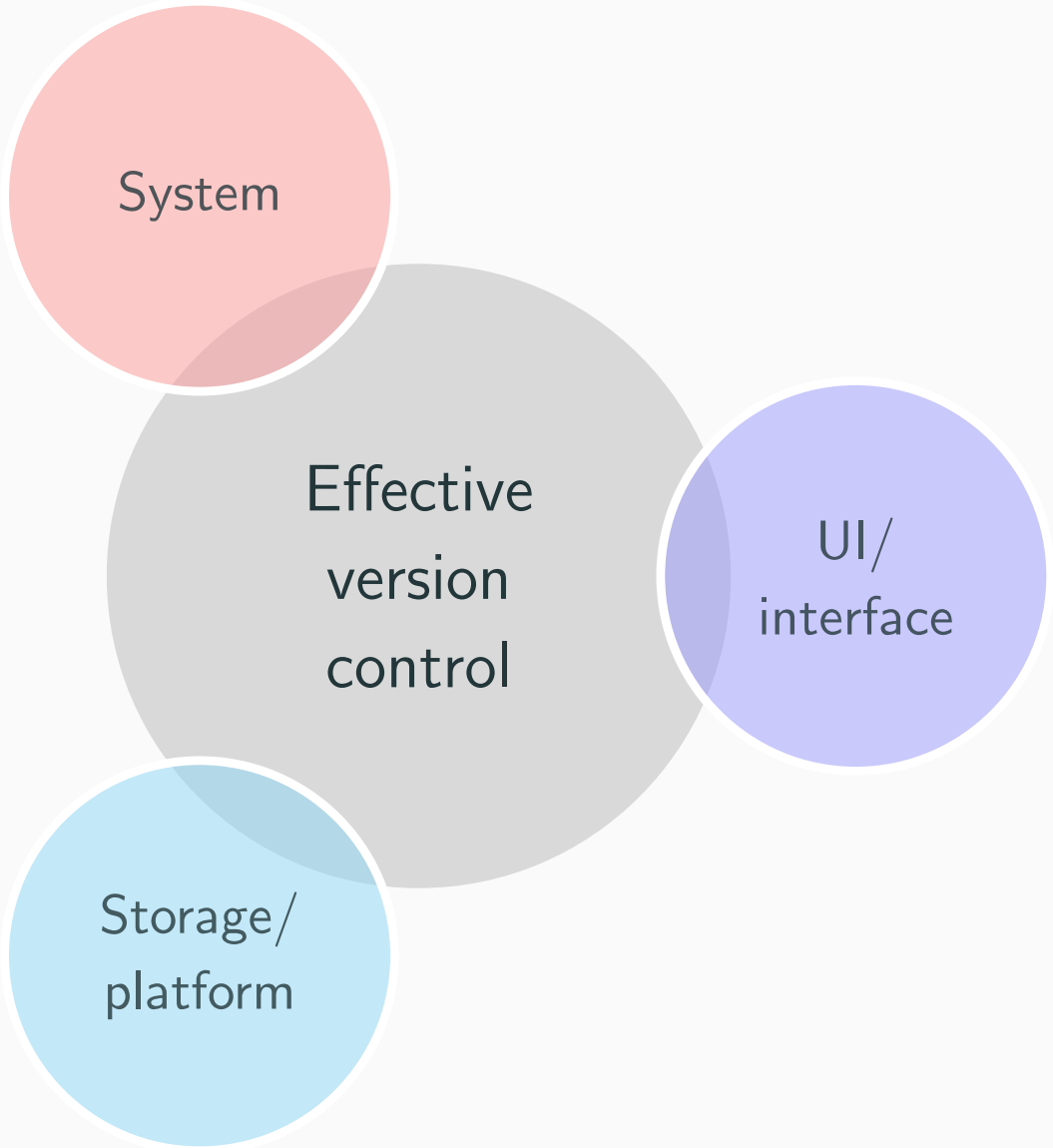
- Add much extra control to binary files (not plain text)
- Back up in real time



Version control system

What version control system?

The first thing about version control systems is which system (we will cover git)



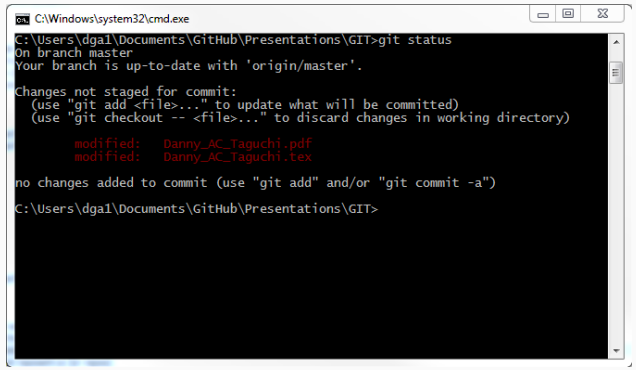
What version control system?

- There are two popular version control systems (git & SVN)
- We will cover git as it is the most popular, the one I know and easiest to use
- Git was developed in 2005 by Linus Torvalds

(Principal developer of Linux)

What is git?

- Git is just system of version control



- By default it is used through a terminal
- Git has projects with multiple documents called repositories (or repos)
- It then has a tree structure to manage the versions of the repository

Tree structure?

[master] 6c6faa5 My first commit - John Doe

[develop] 3e89ec8 Develop a feature - part 1 - John Doe

[develop] e188fa9 Develop a feature - part 2 - John Doe

[master] 665003d Fast bugfix - John Fixer

[myfeature] eaf618c New cool feature - John Feature

[master] 8f1e0e7 Merge branch `develop` into `master` - John Doe

[master] 6a3dacc Merge branch `myfeature` into `master` - John Doe

[master] abcdef0 Release of version 0.1 - John Releaser

The master branch is in grey with colour branches coming of and then being merged back

Key commands?

In the terminal interface there are some useful commands:

- *init* - initialise a repo
- *status* - tells you if you have files out of sync with the current version
- *commit* - adds commits to the current version - a change with a comment and an ID
- *add* - adds new files ready to commit
- *remove* - removes files ready to commit
- *push* - pushes commits to an online repo (only if using a git platform)
- *pull* - pulls commits from an online platform (only if using a git platform)

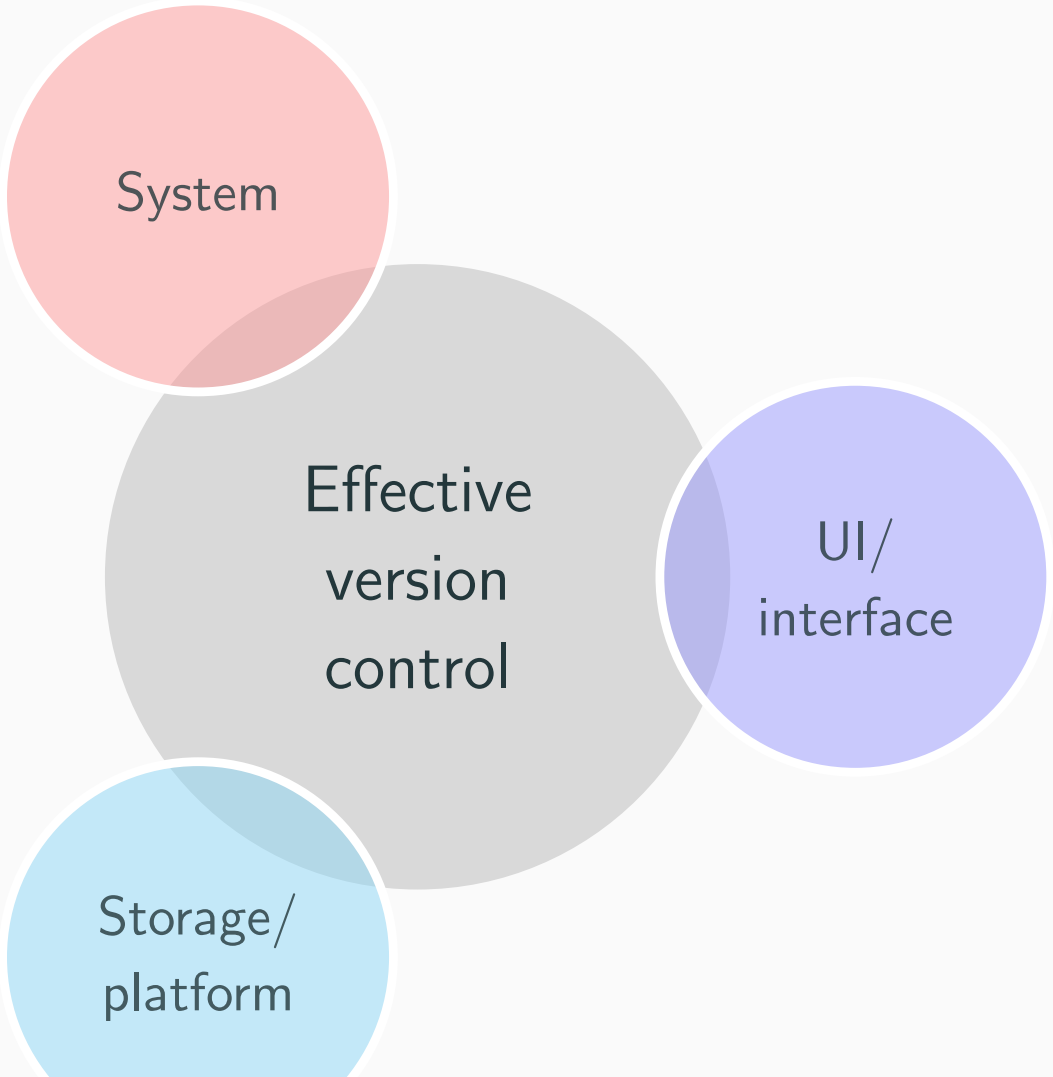

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

(xkcd)

Platform

Which platform?

You can just use git with the terminal on your PC or you can store work on an online repository managing platform, the most popular of these are GitHub and Bitbucket



Why not just store the repository on Dropbox?

You can and it would be backed up and sharable, but there are problems:

- You can run into Dropbox syncing and git version control clashes
- If you share the files with a collaborator they aren't identified separately to you
- You or a collaborator can completely mess up the git system (mostly by deleting the records in the '.git' folder)
- If you do you can't just re download

Which platform?

Bitbucket

- Allows unlimited public or private repositories
- Charges per collaborators over 5 on those repositories
- 1GB storage for **large** files all repositories, 5GB if using a .ac email

GitHub

- Has unlimited public repositories
- With unlimited collaborators
- Pay for private repositories
- However as a student or academic you can sign up for free private repositories
- GitHub is about $\times 4$ more popular than Bitbucket

IBERS Git?

IBERS don't have an internally hosted git system but:

The National Plant Phenomics Centre has a server with GitLab (like GitHub, but local.)

IBERS Git?

IBERS don't have an internally hosted git system but:

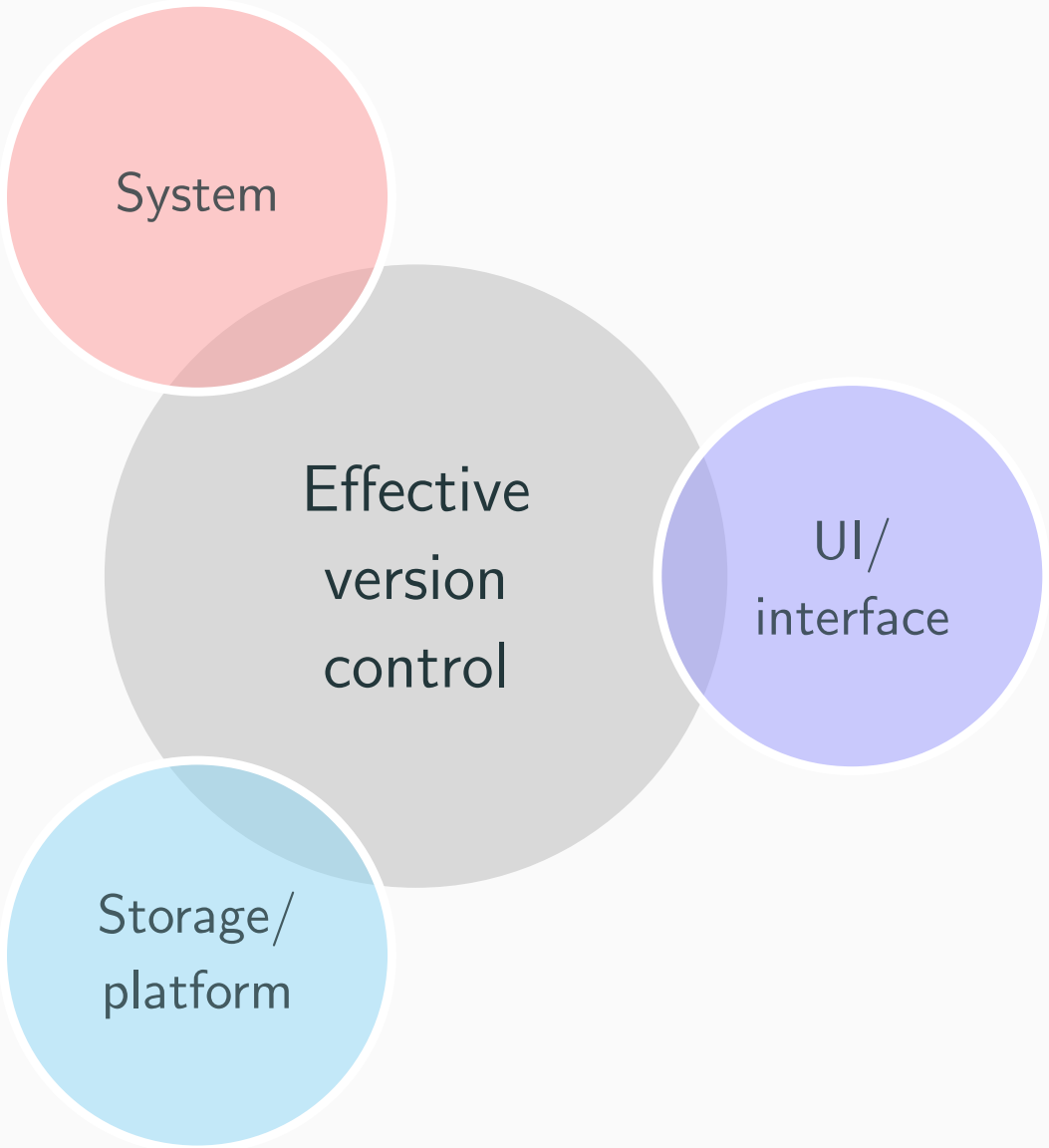
The National Plant Phenomics Centre has a server with GitLab (like GitHub, but local.)

If you are interested in using ths contact Max Friedersdorff (maf54@aber.ac.uk), the NPPC's Data Manager.

UI

Which UI?

You can just use git with the terminal but there is a friendlier interface

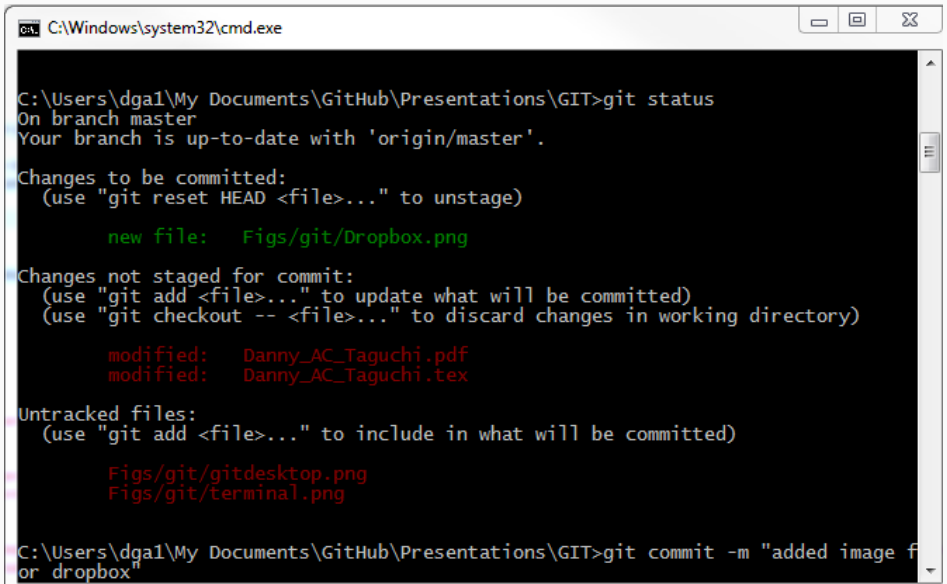


Which UI?

The Terminal

You can just use the terminal with the raw git commands as seen before... but:

- If you are not using the terminal anyway or would rather not a graphical UI is nice
- You don't need to remember commands or repo names
- You need to install git for widows `https://git-scm.com/download/win`

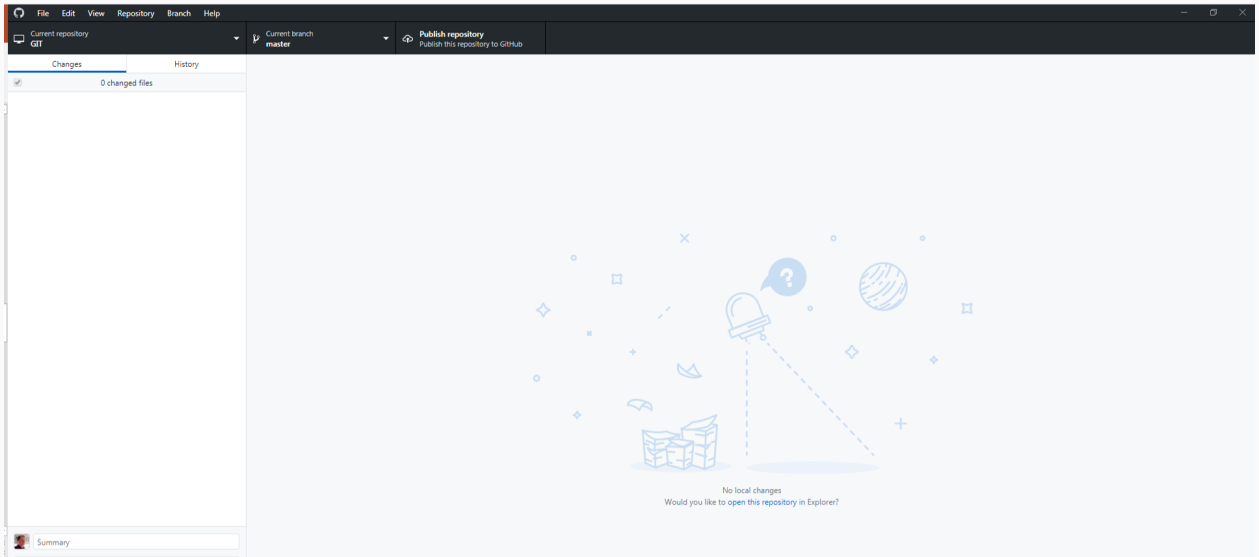


Which UI?

Git Desktop

The easiest to use and most popular UI is git desktop:

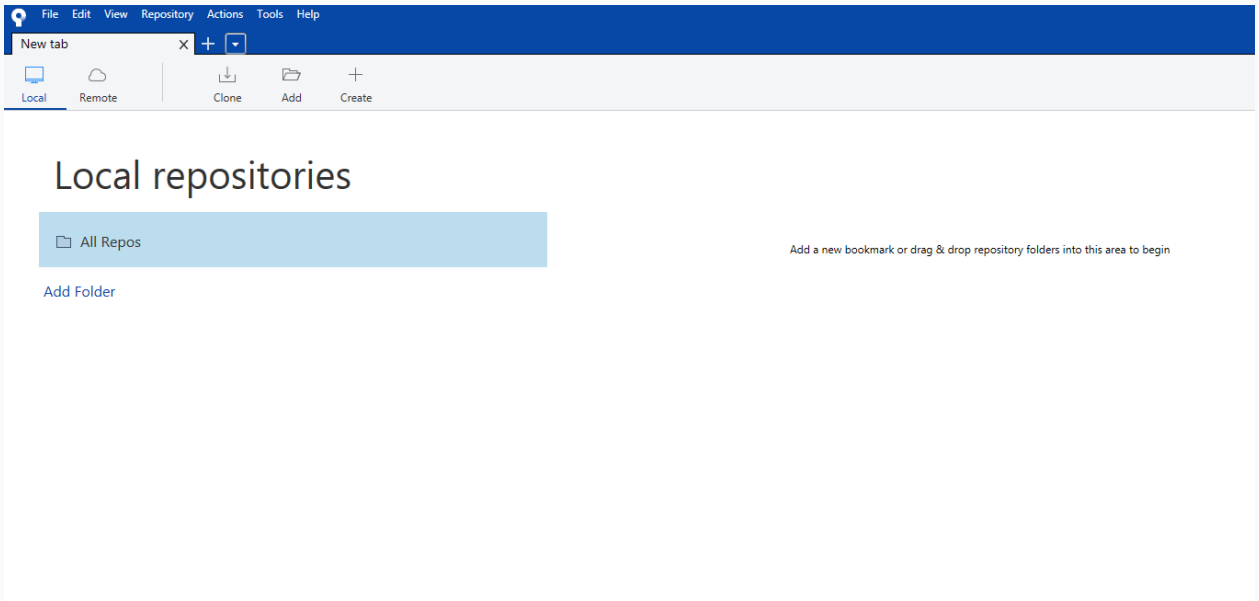
- This is made by GitHub and integrates very well with GitHub or local (on pc) repositories
- It will work with other hosting platforms but less easily
- Is the easiest to use!
- Will install git for windows `https://git-scm.com/download/win`



Which UI?

Sourcetree

- Sourcetree is owned by atlassian who ouns Bitbucket
- It will work with other hosting platforms easily
- It has more complex controls
- Will install git for windows
  `https://git-scm.com/download/win`



Using git

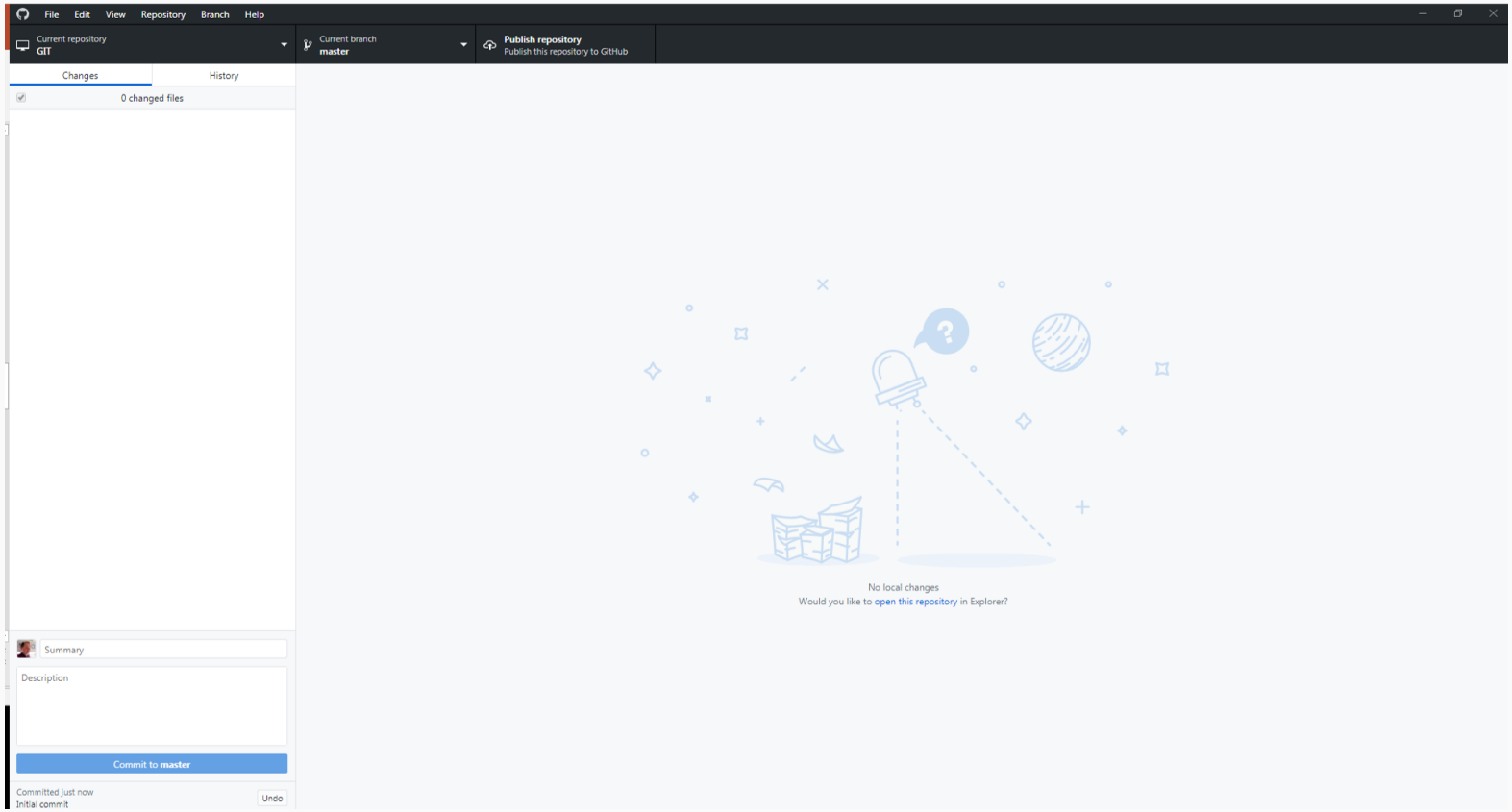
How to use git

This will assume use of GitHub and GitDesktop

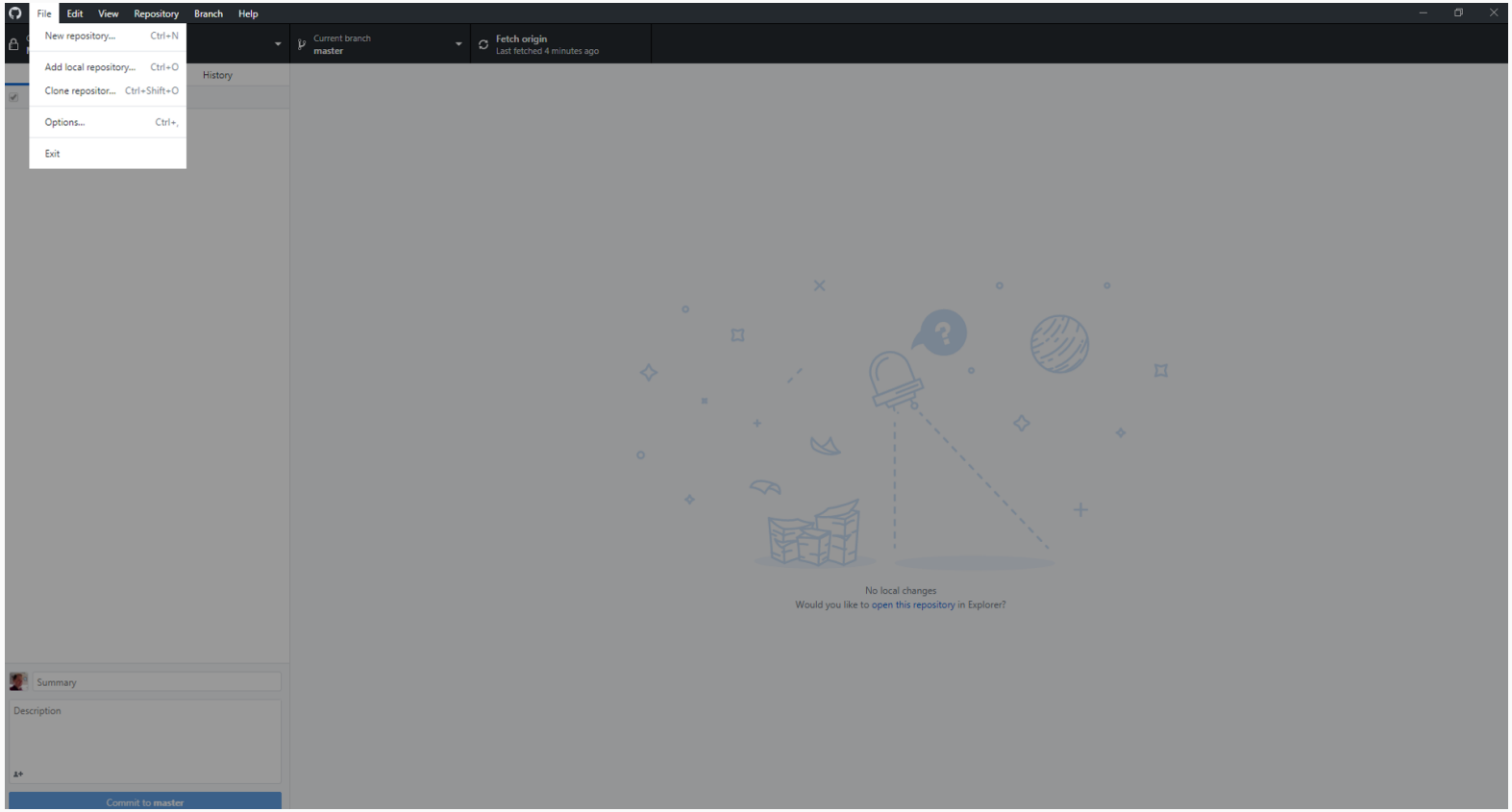
`https://desktop.github.com`

Following is a brief demonstration of the interface

Adding a new repository

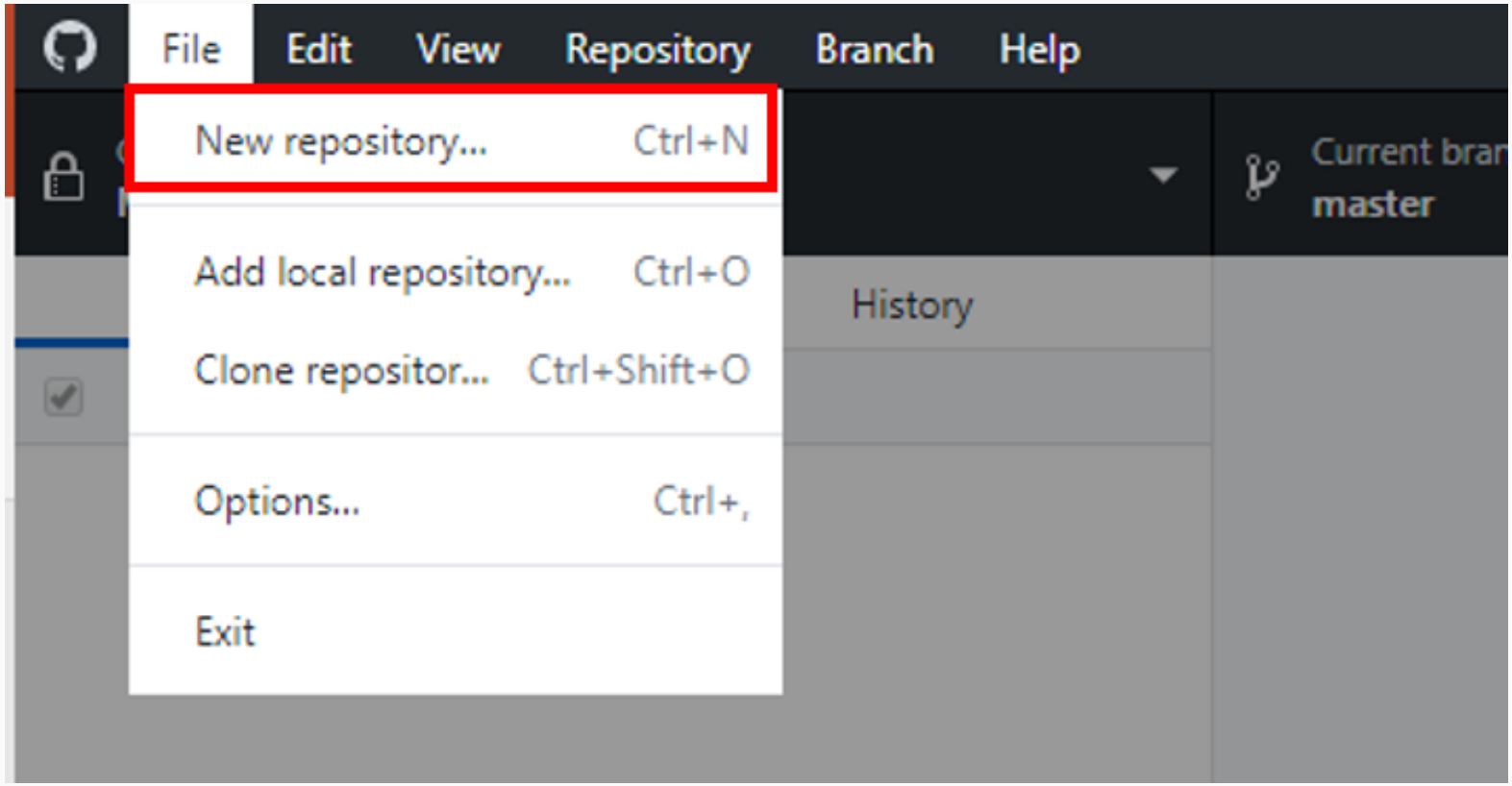


Adding a new repository



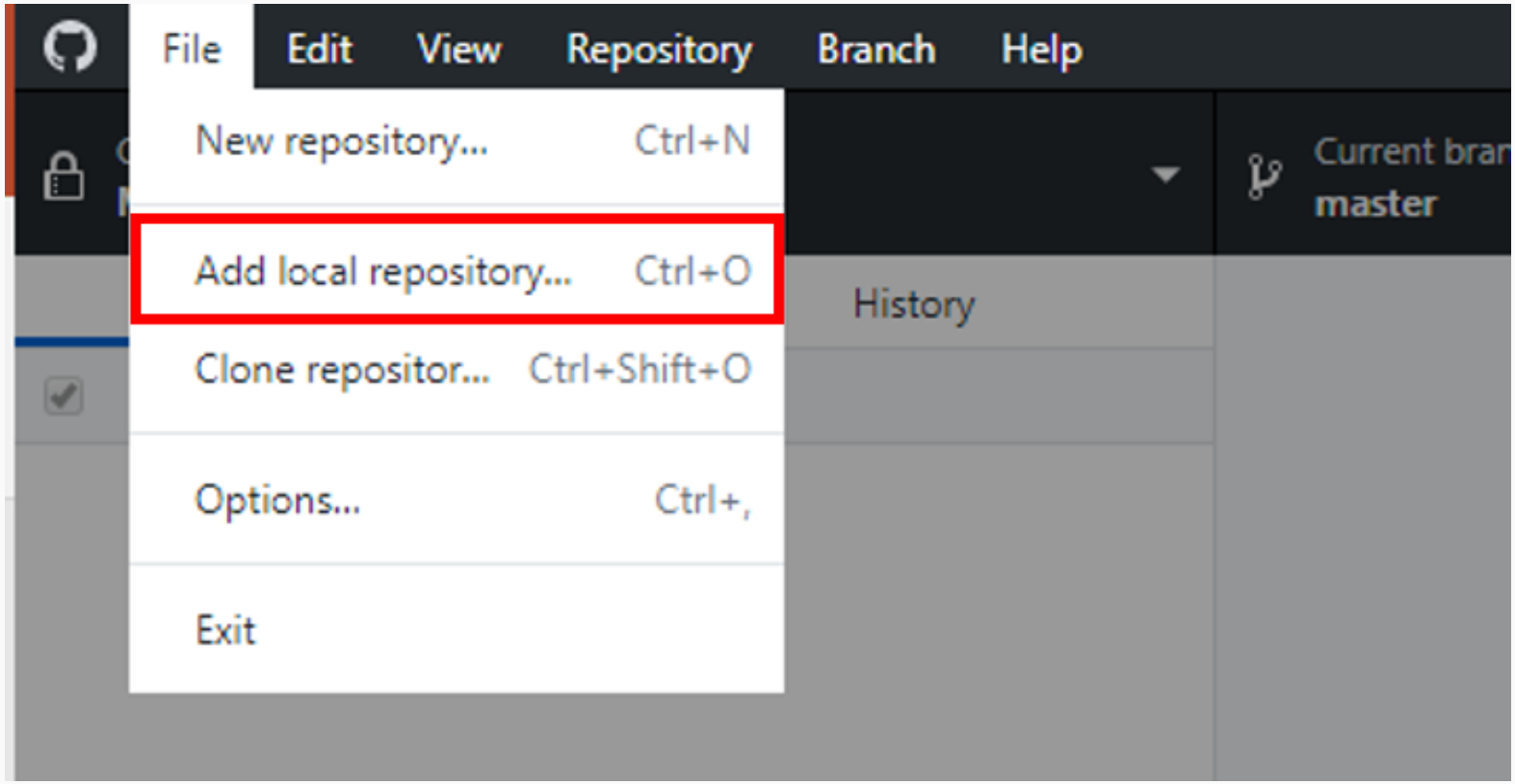
Adding a new repository

This makes a totally new repository (with no '.git' folder)



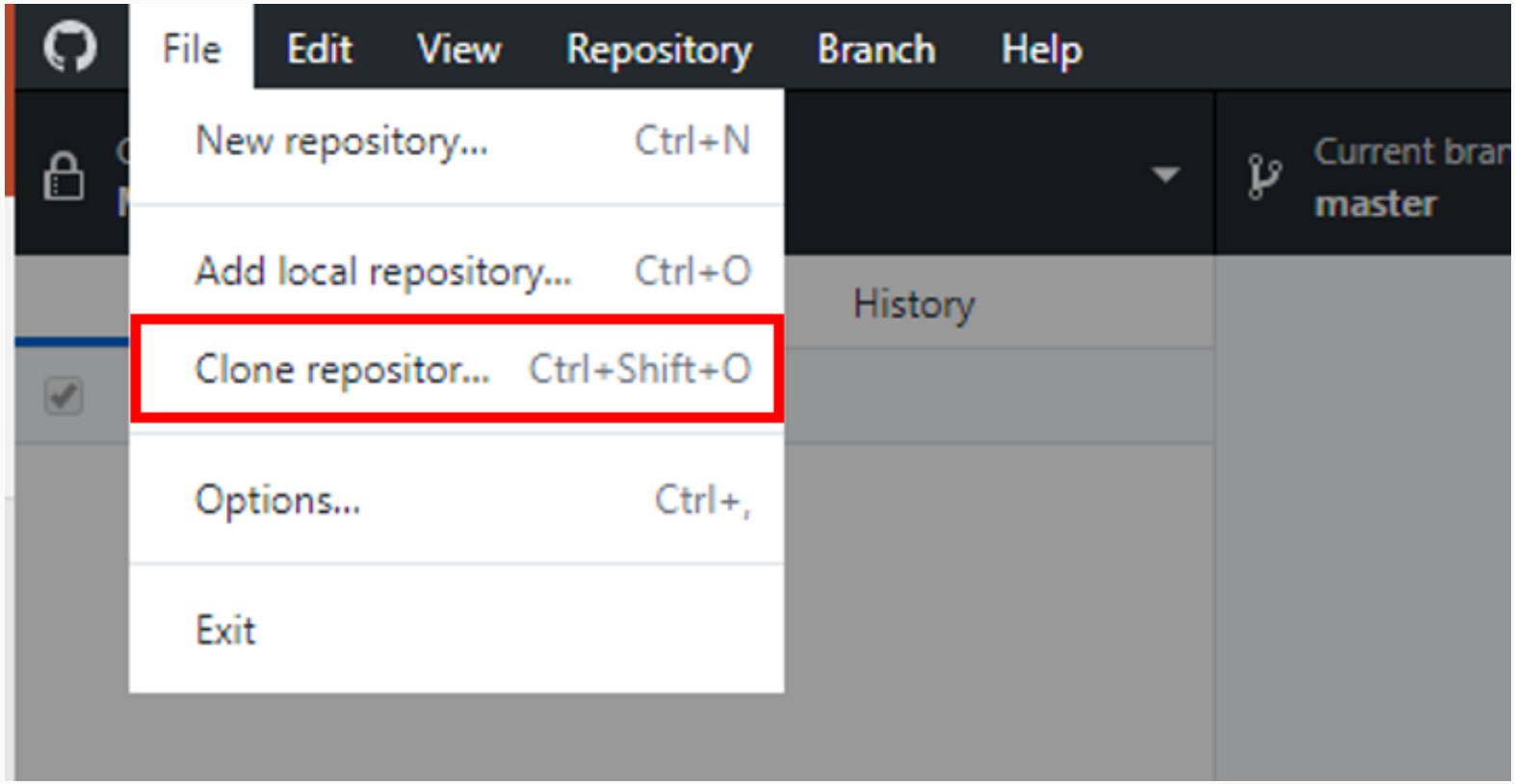
Adding a new repository

adds an existing repository (with a '.git' folder)



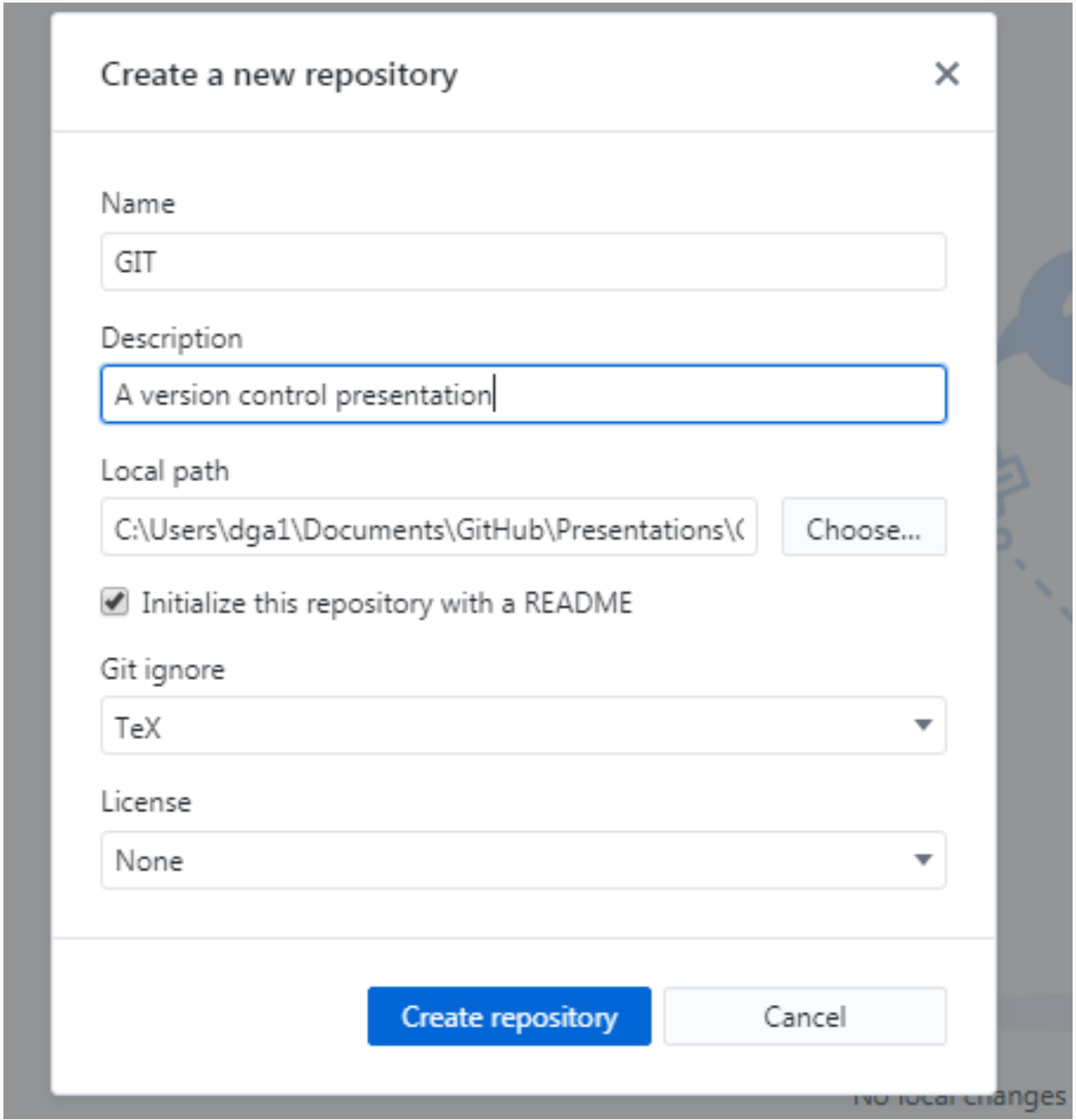
Adding a new repository

Copies an online repository to the PC

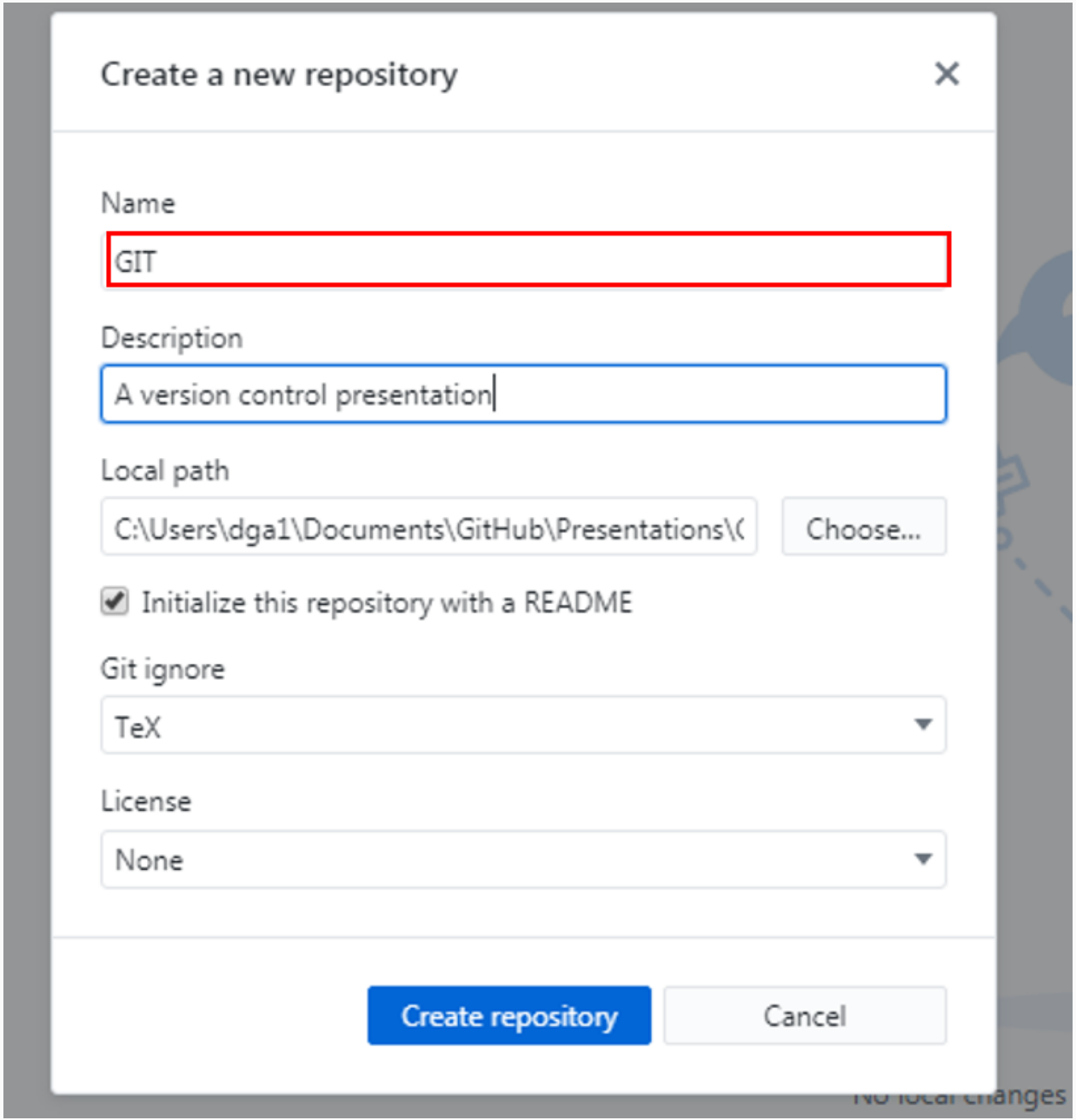


Adding a new repository

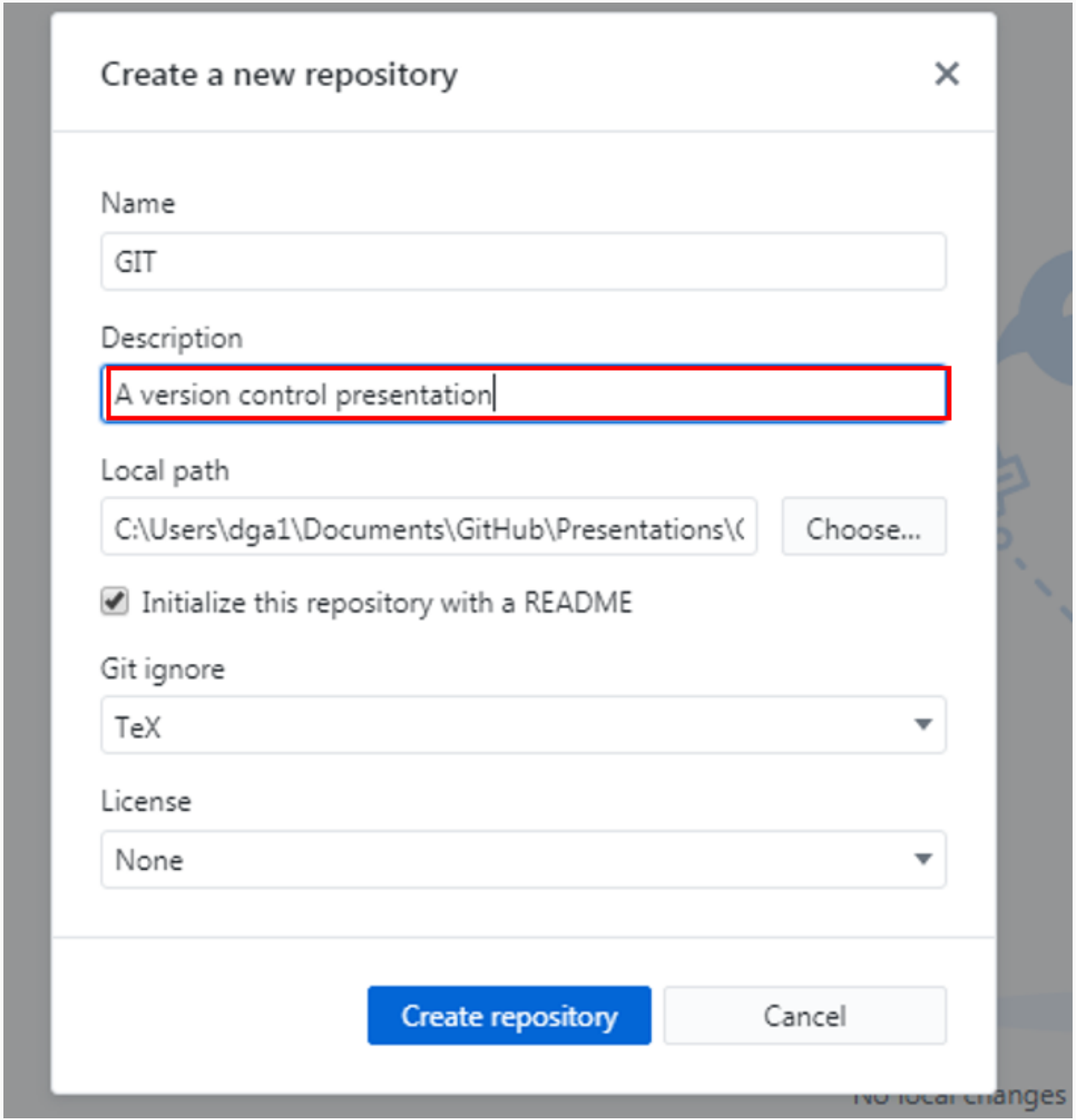
new repository menu



Adding a new repository

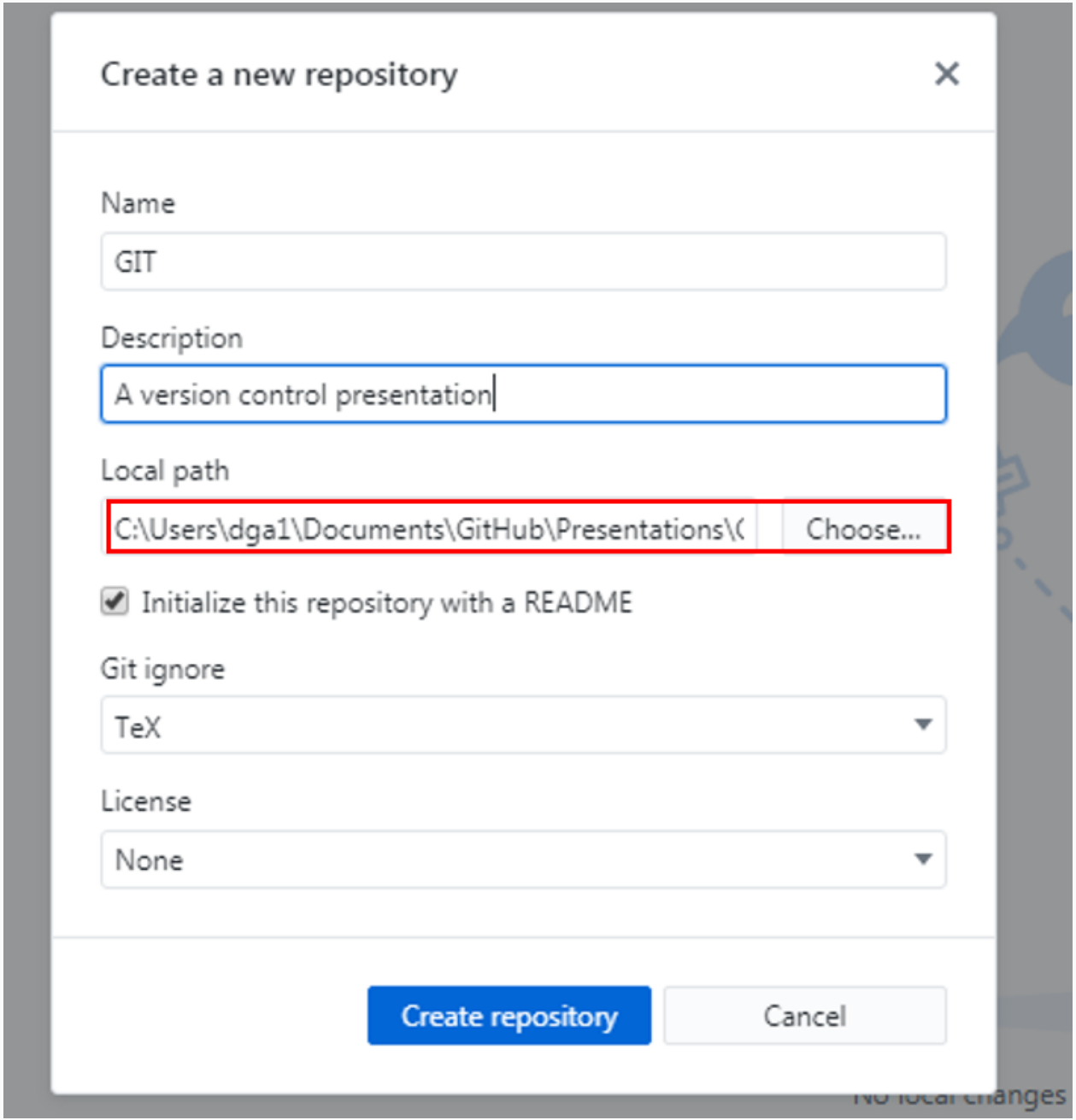


Adding a new repository



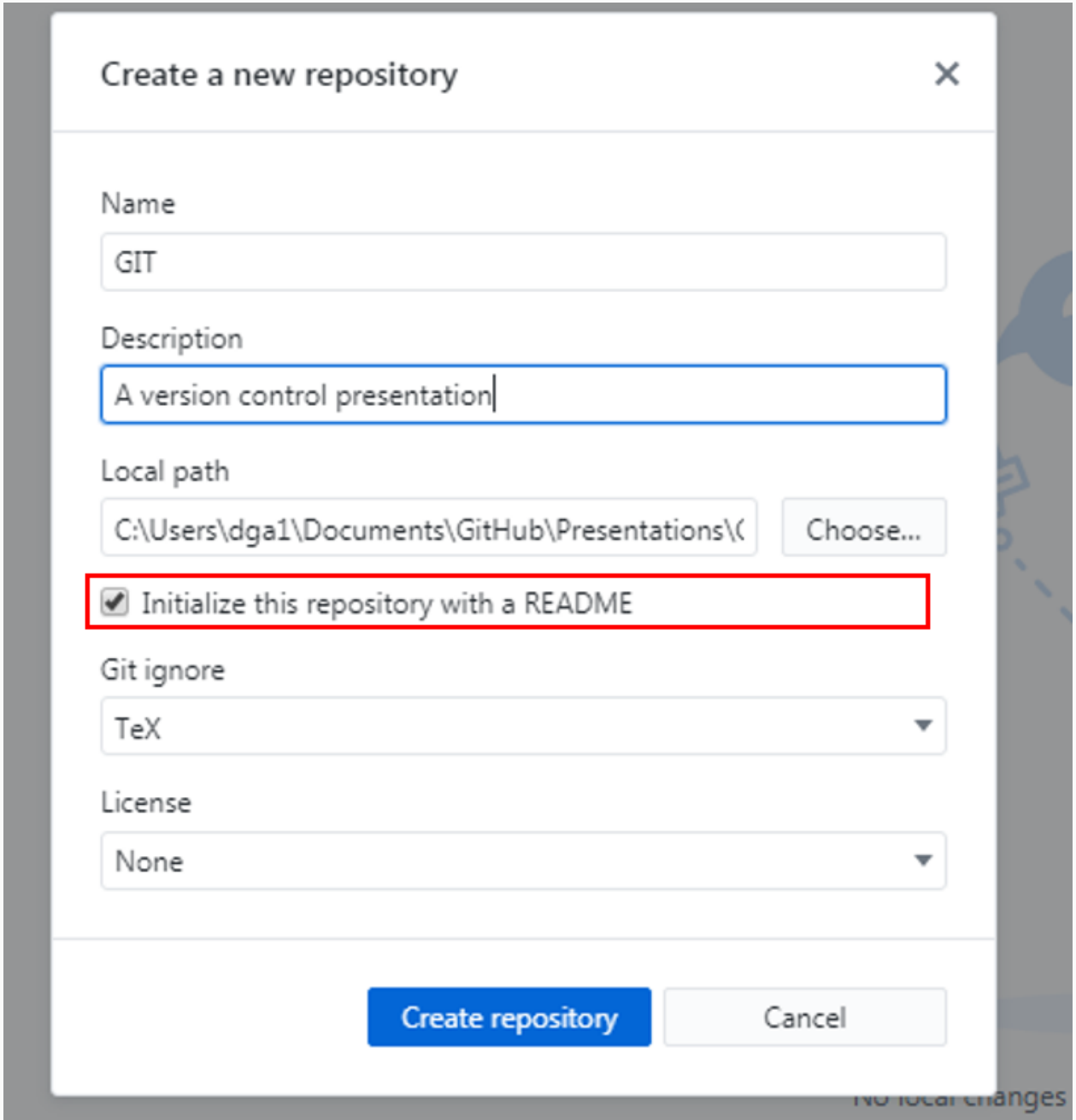
Adding a new repository

The folder **in which** to store the git repository folder.



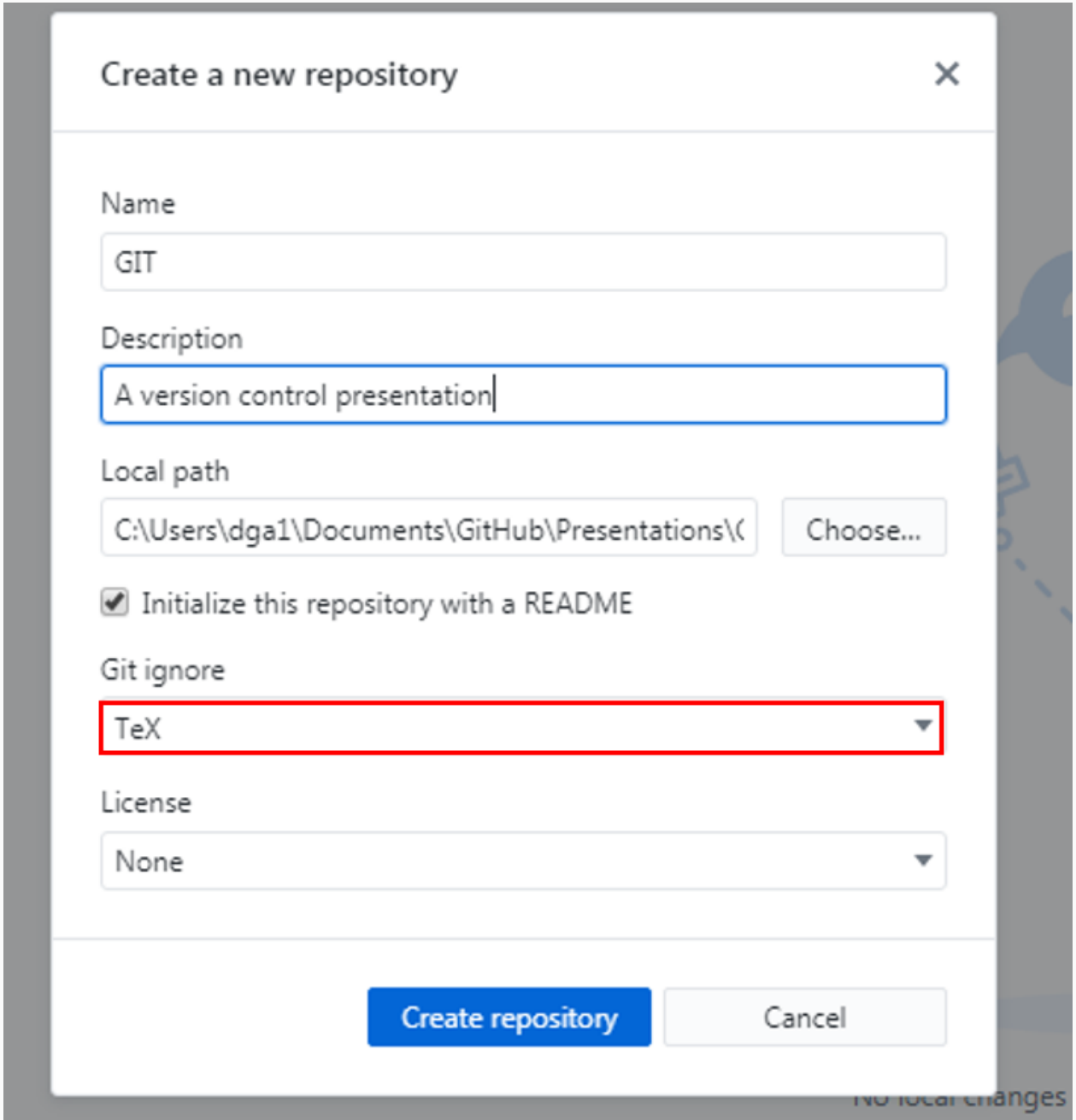
Adding a new repository

This will initialise the repository with a readme containing the description.

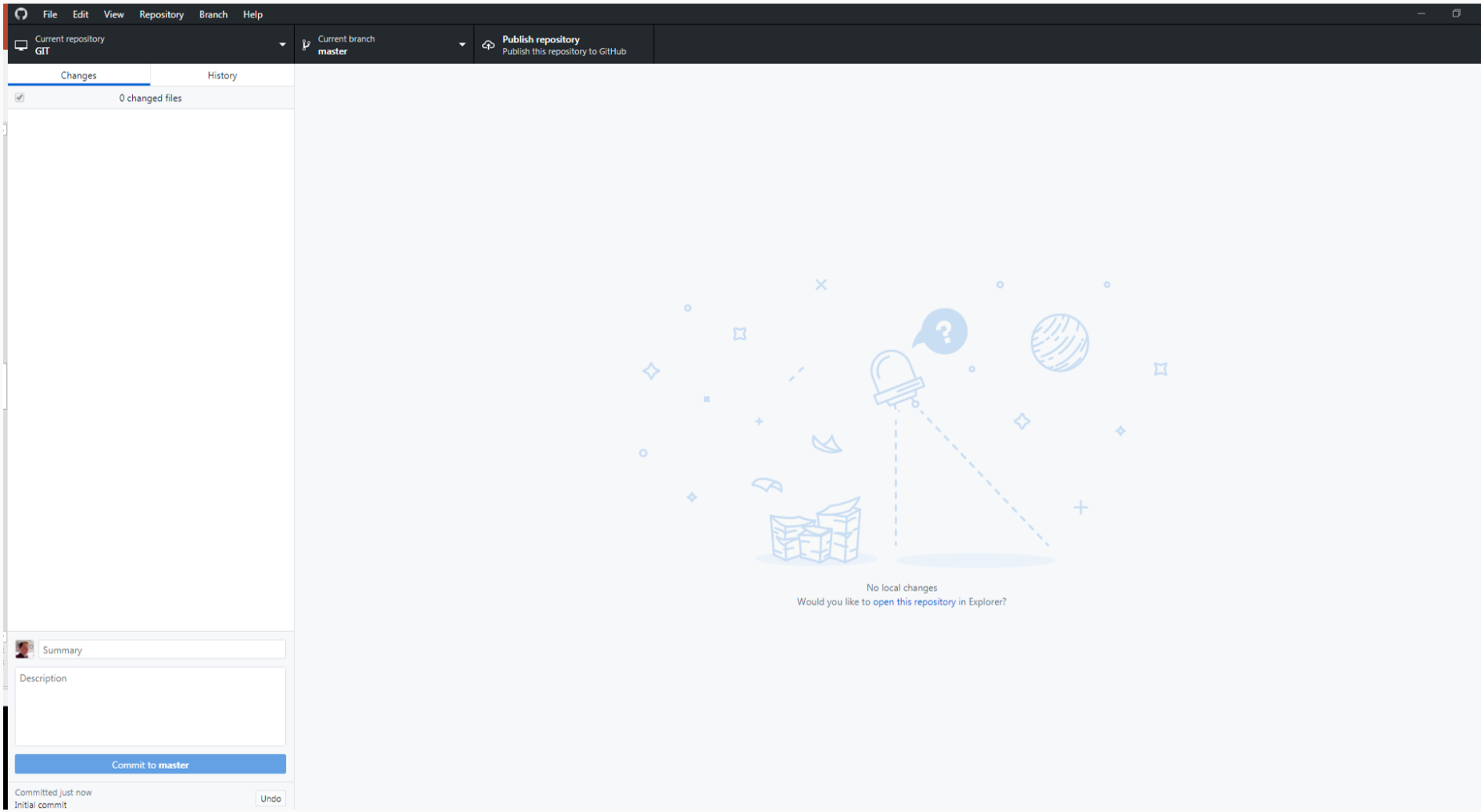


Adding a new repository

This is a file called '.gitignore' which lists files & folders to not version control

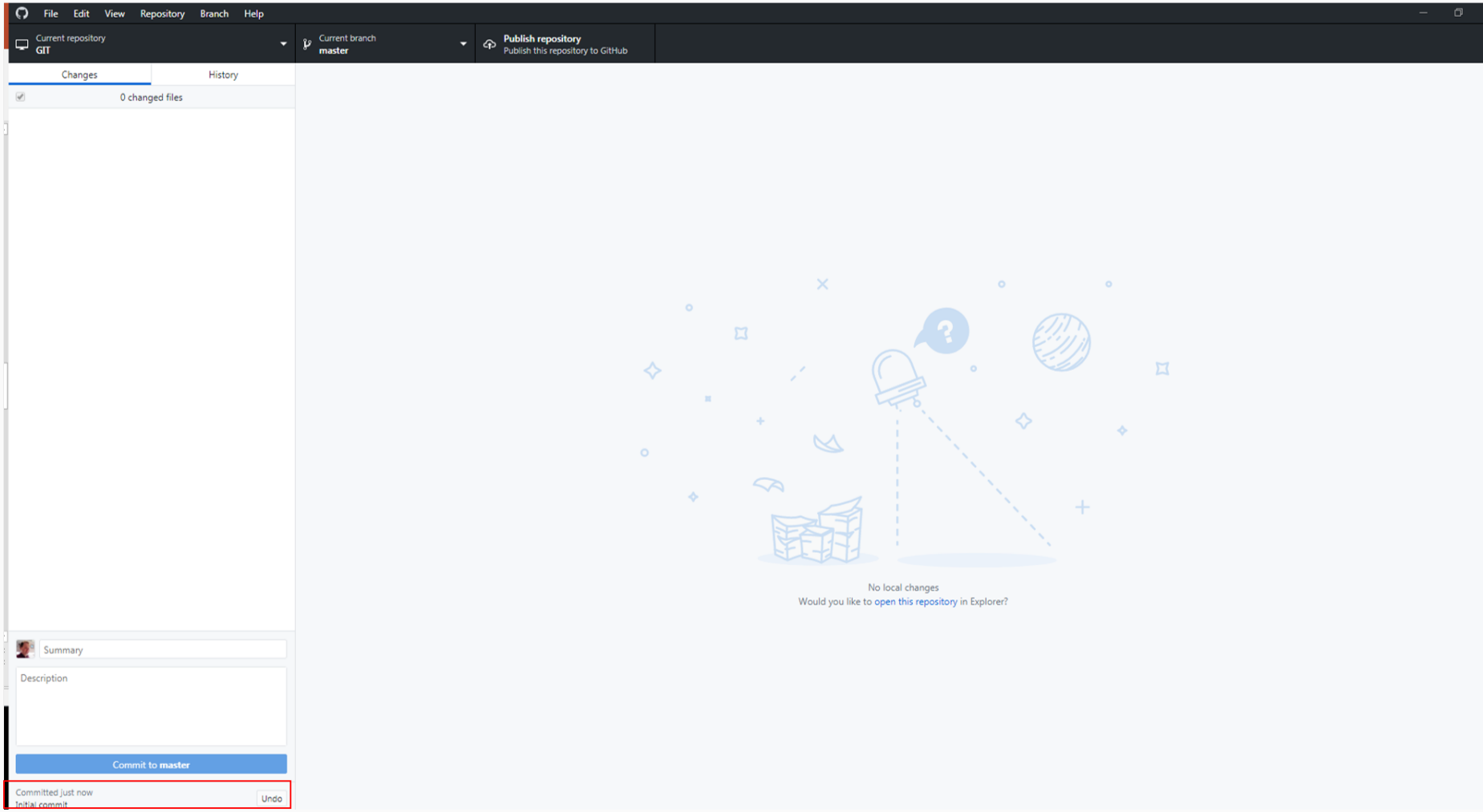


Using the repository



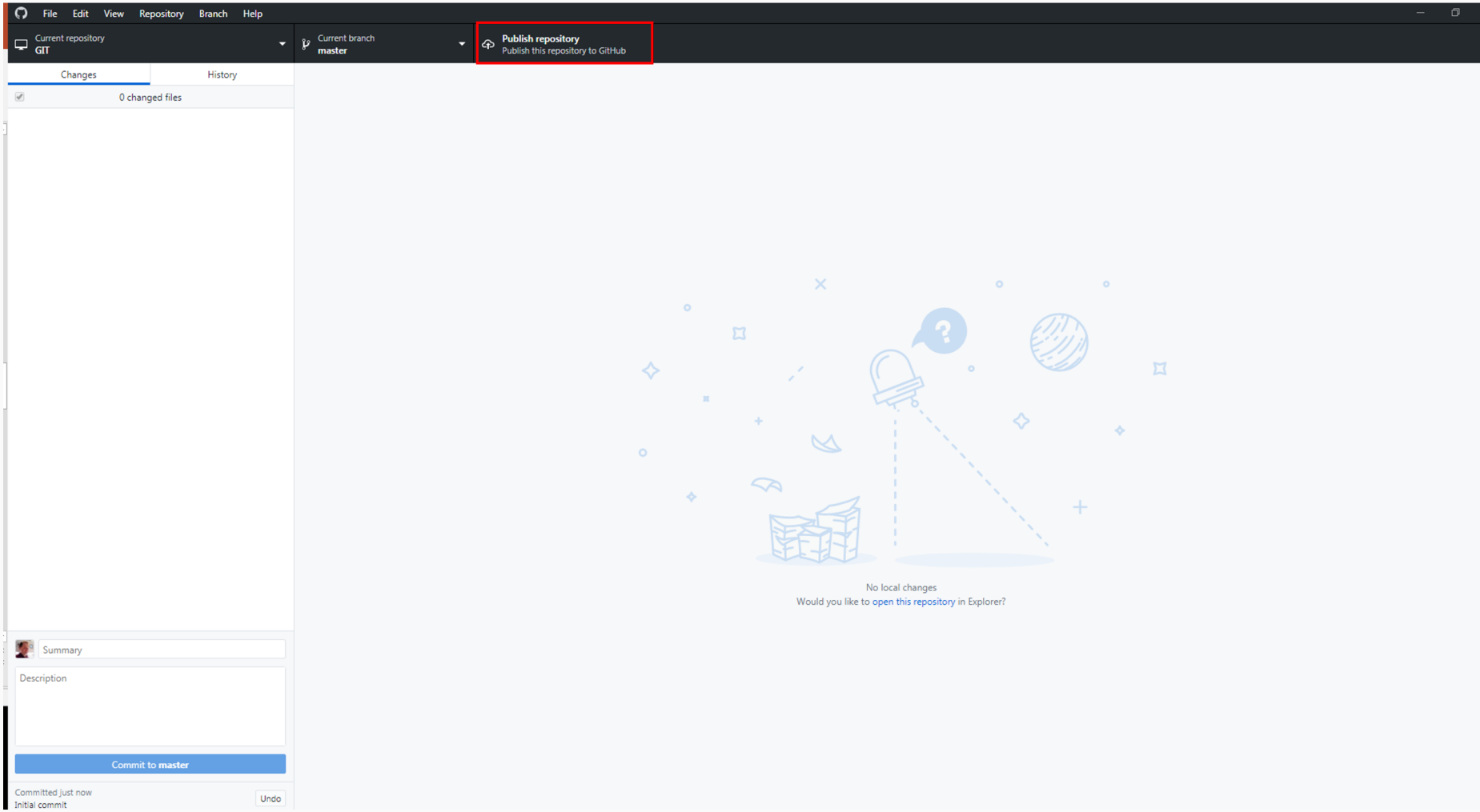
Using the repository

Now we have a initial commit of the readme done



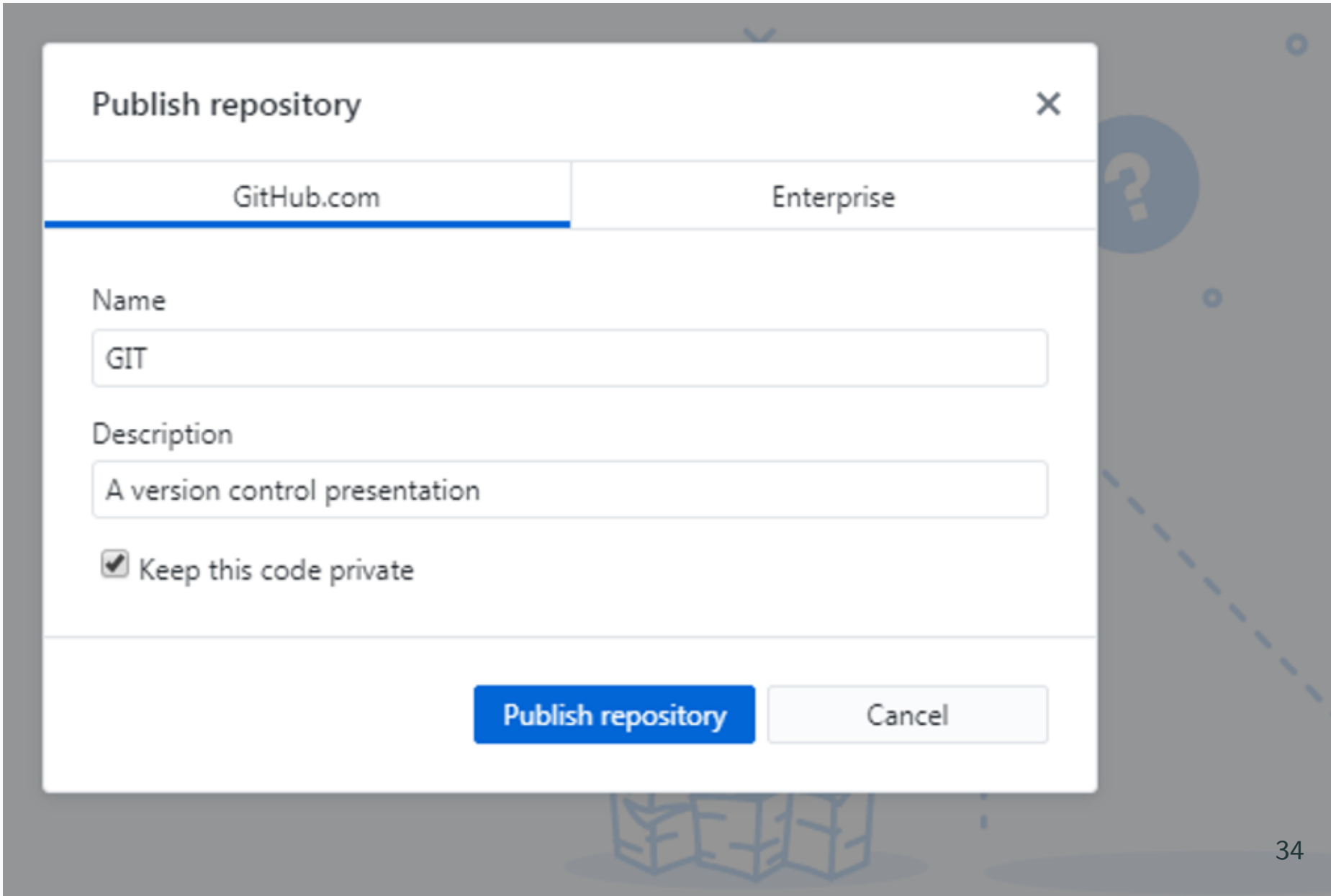
Using the repository

We can carry on making local commits or publish the repository to GitHub



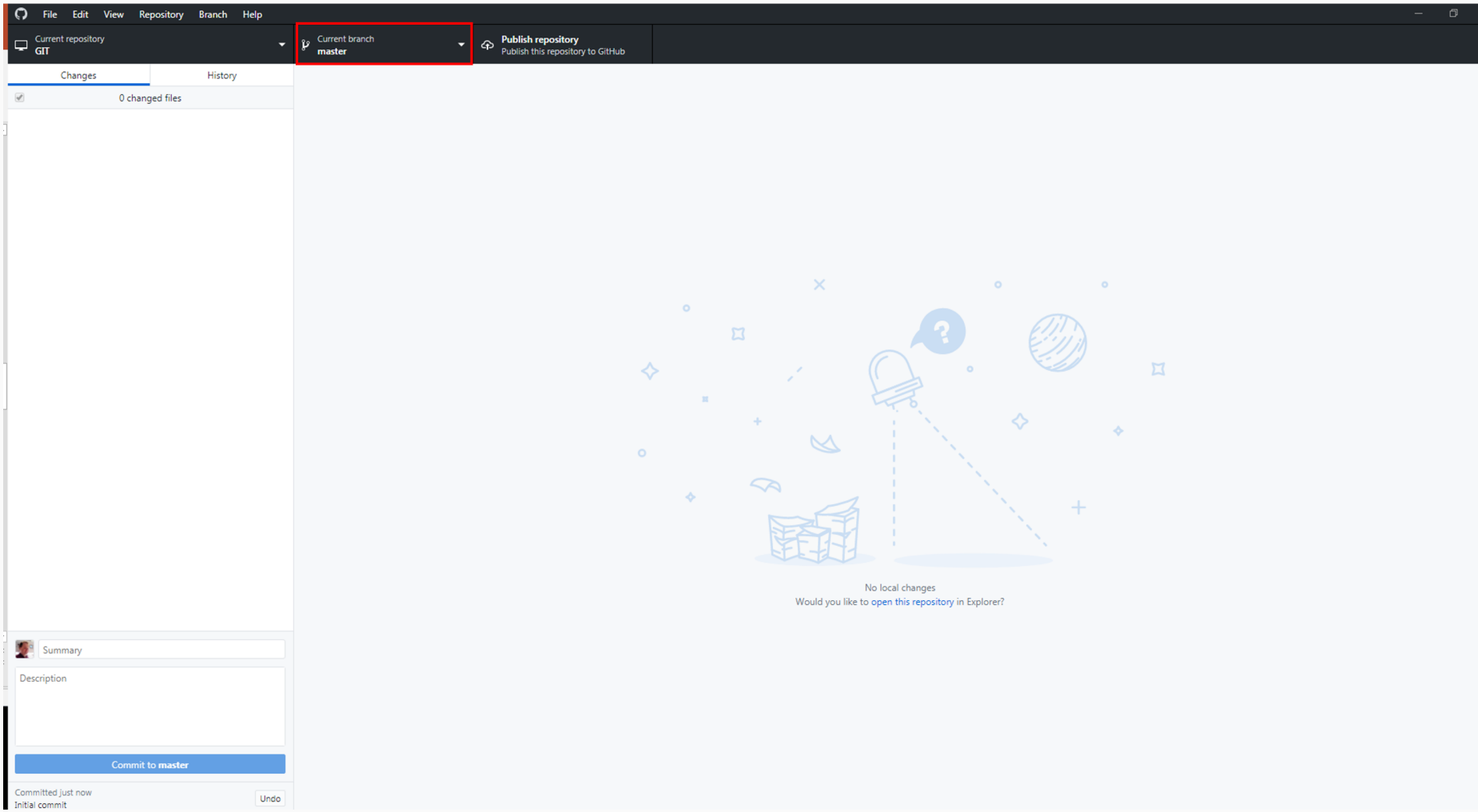
Using the repository

We can publish to GitHub



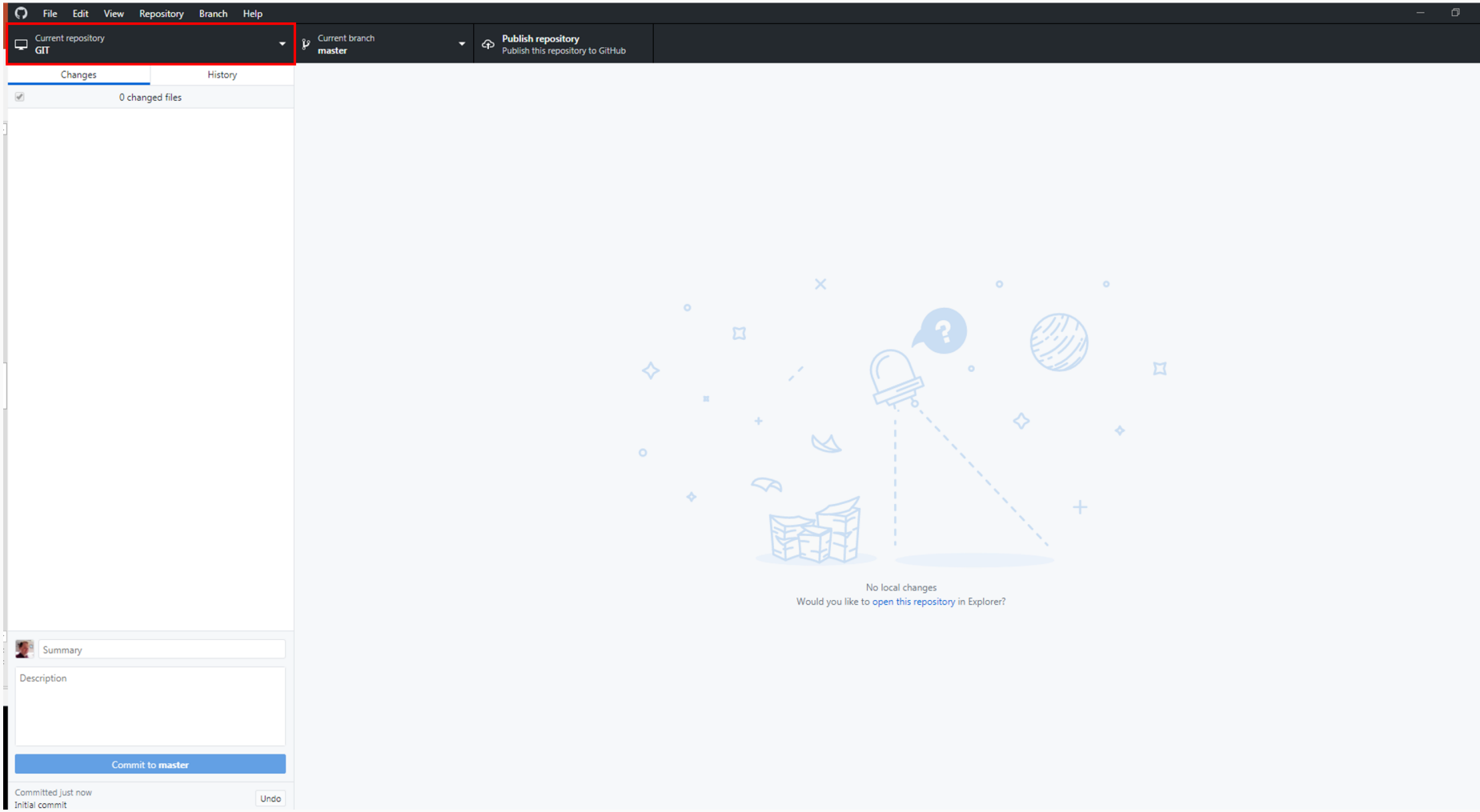
Using the repository

Make branches



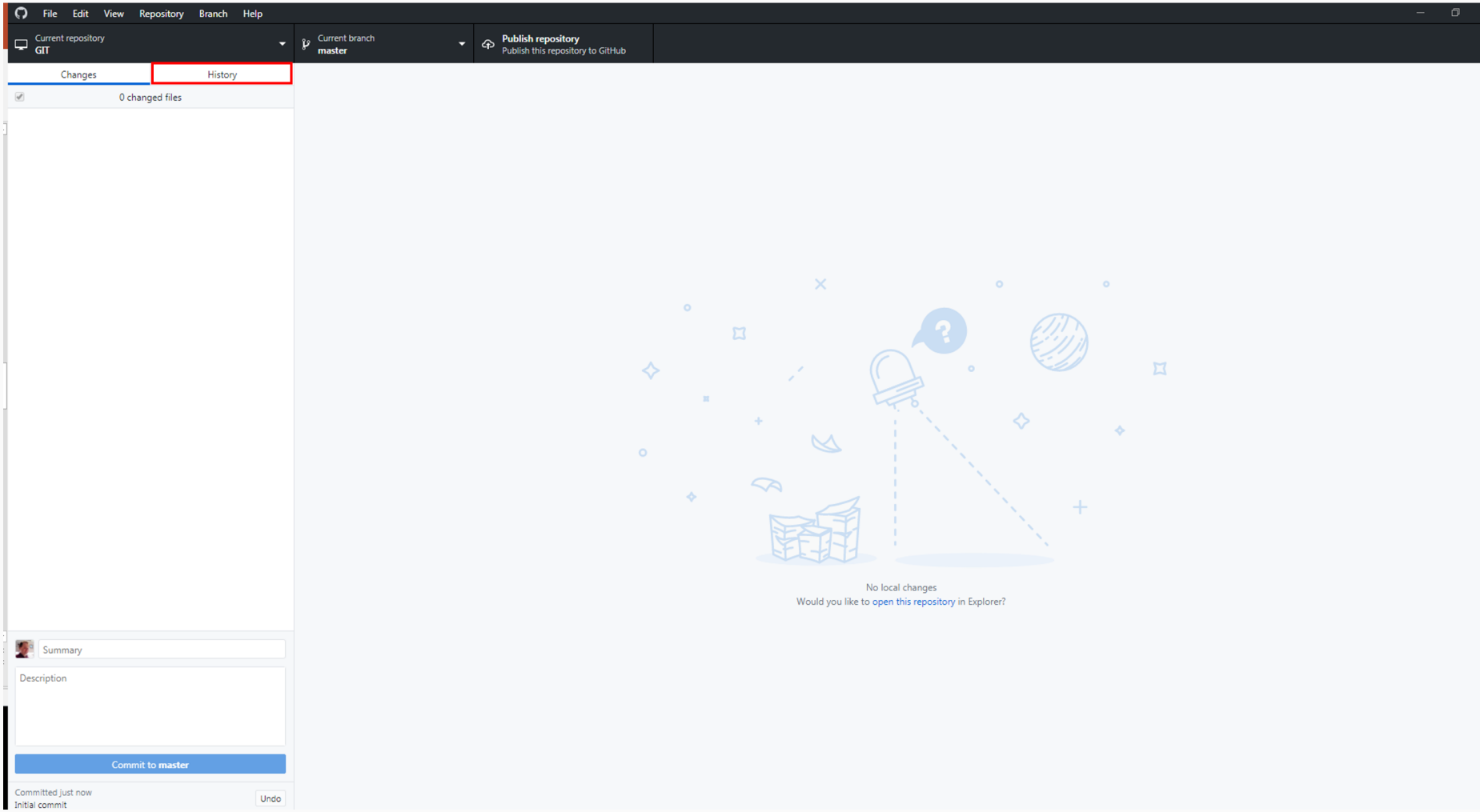
Using the repository

Switch to another repository



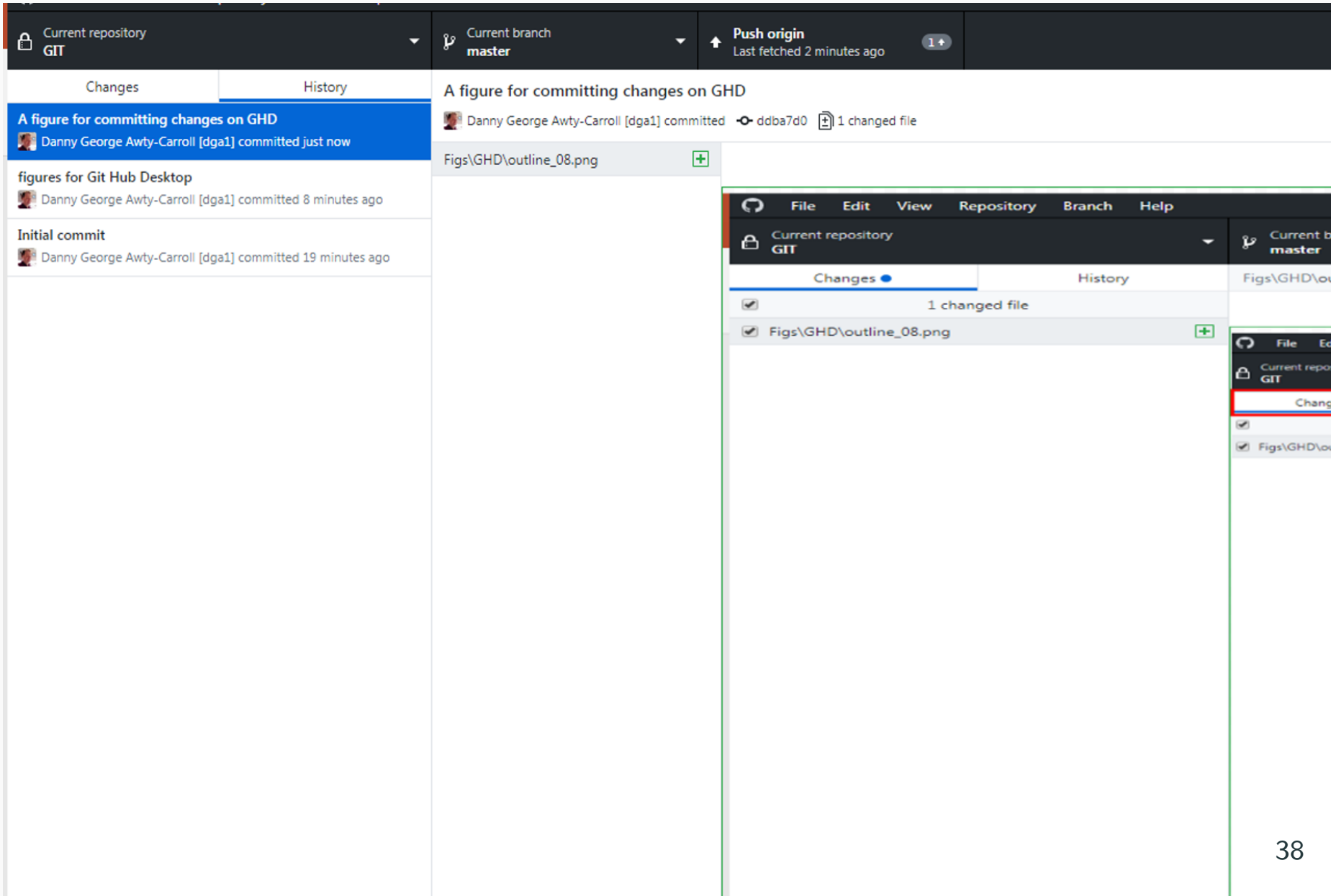
Using the repository

Look at the history of this repository



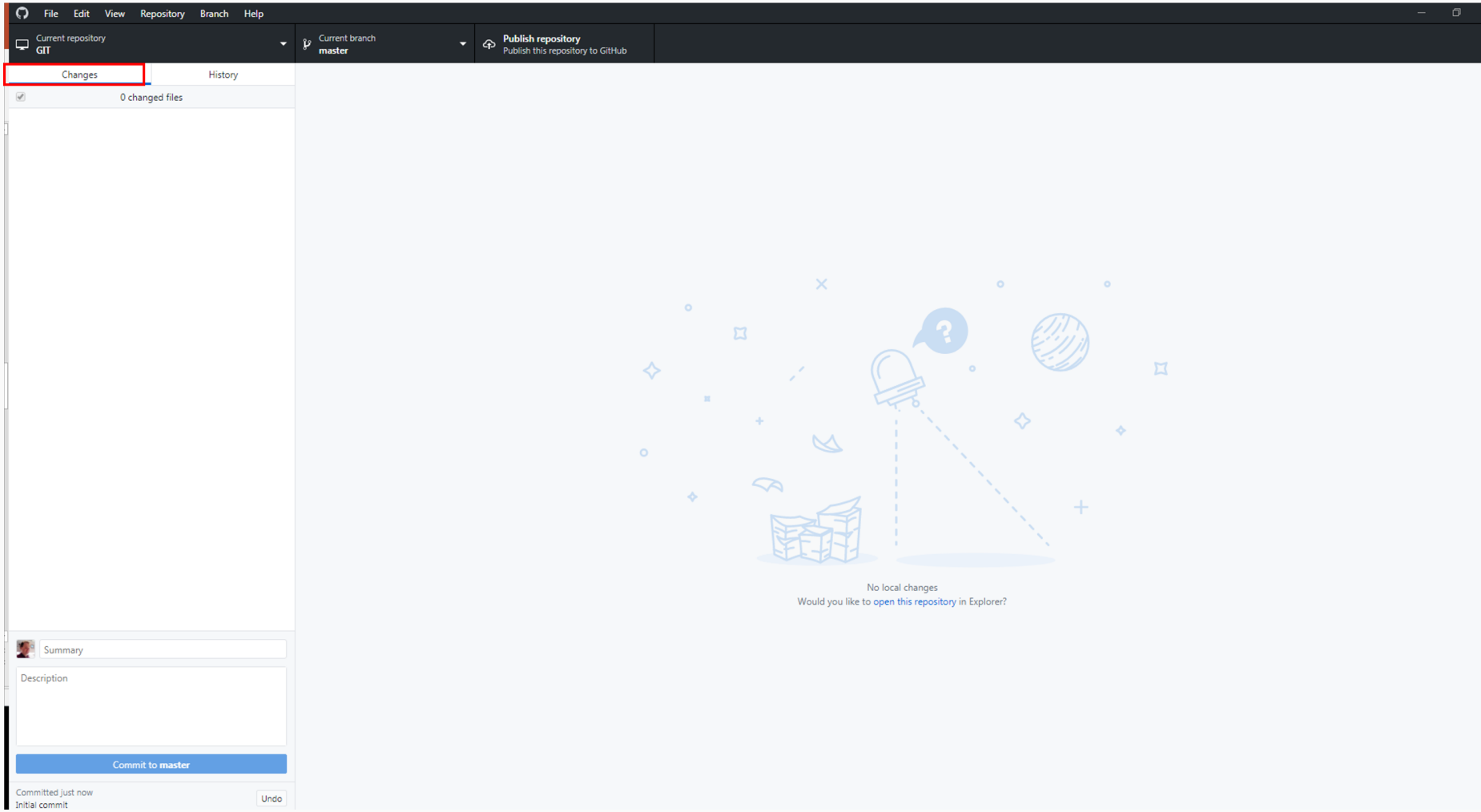
Using the repository

Look at the history of this repository



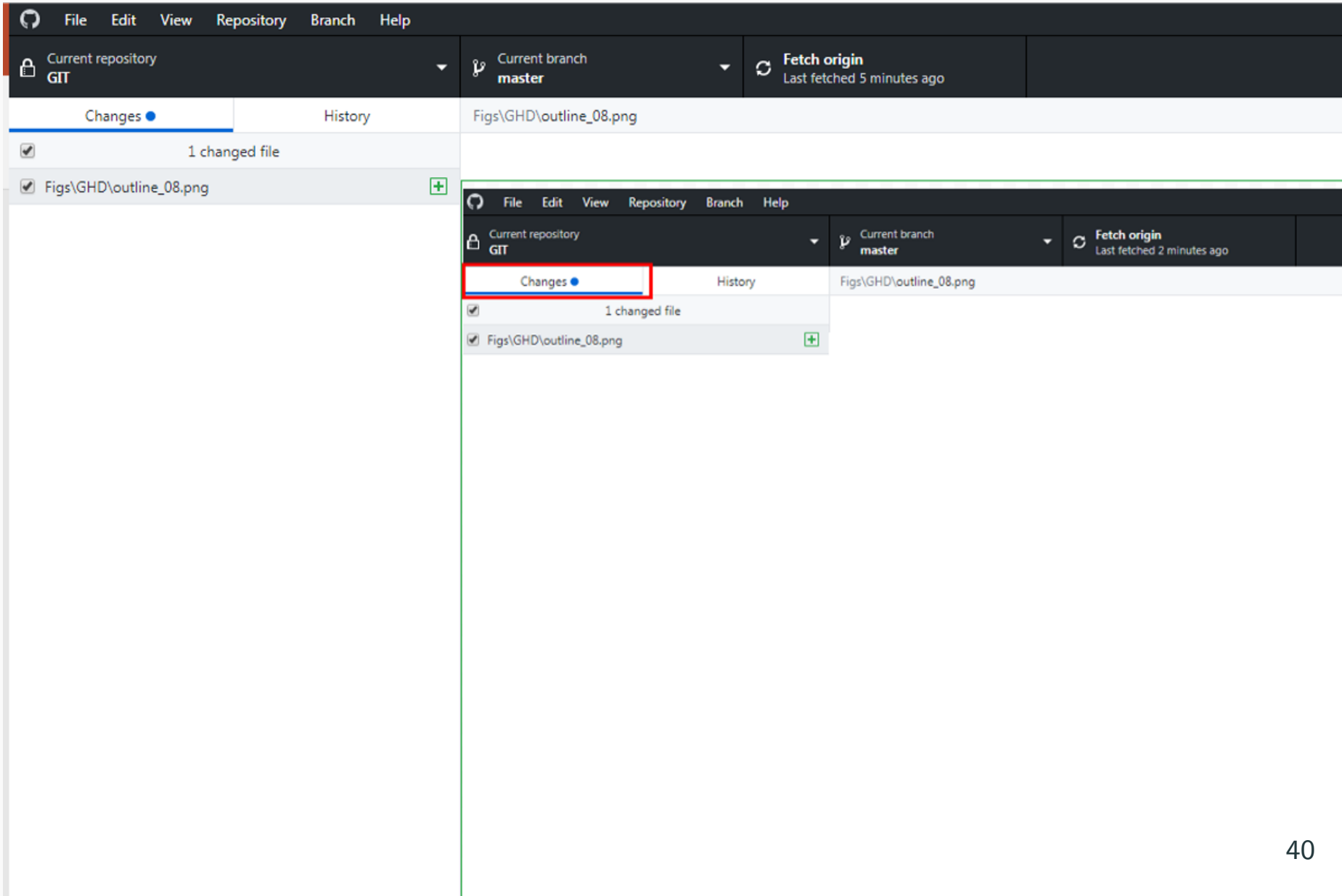
Using the repository

Or look at how the files have changed from the records (same as *git status*)



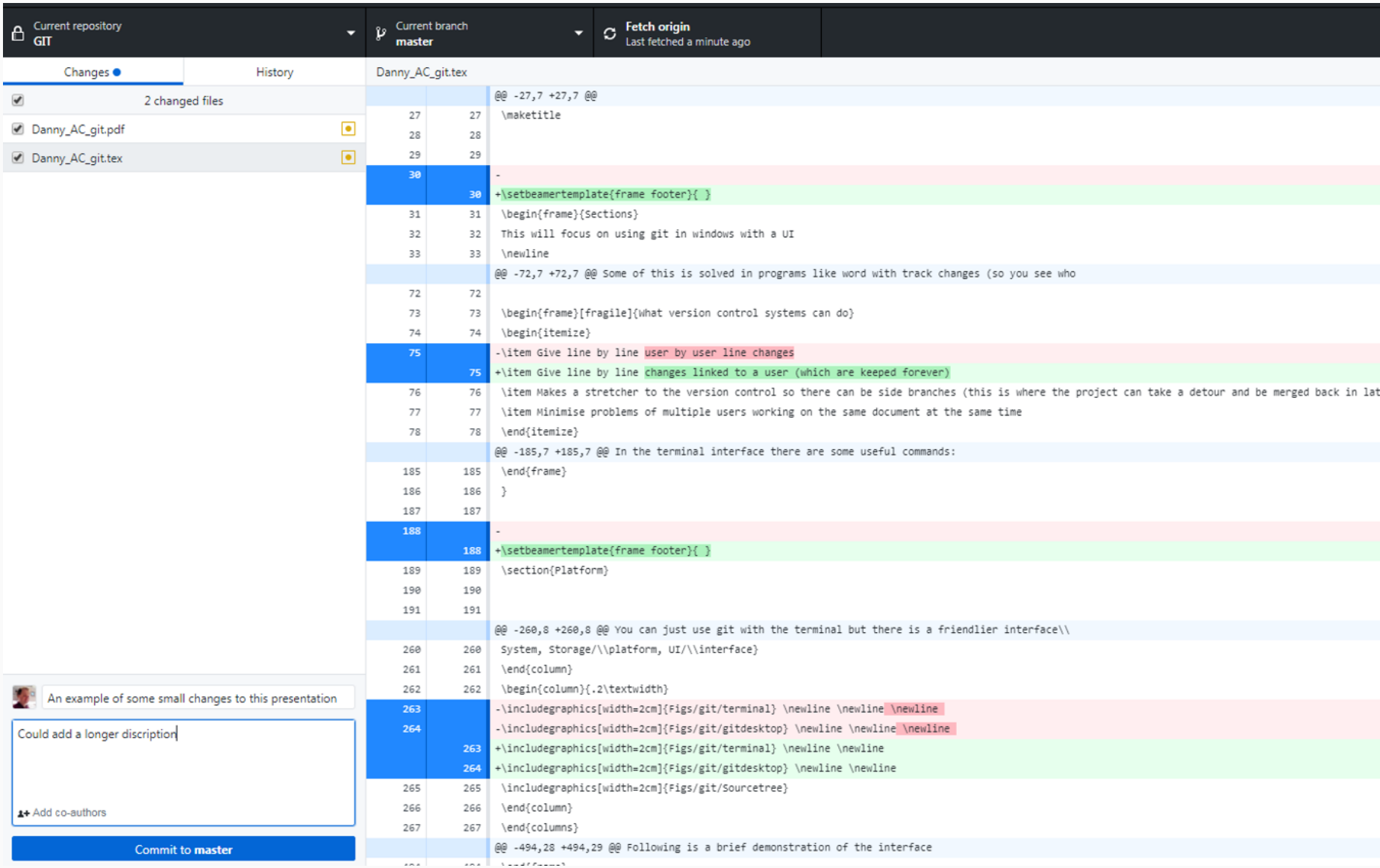
Using the repository

When there are changes you can make a new commit



Using the repository

An example of some small changes to this presentation



Questions & Demo


COMMENT
DATE
CREATED MAIN LOOP & TIMING CONTROL
14 HOURS AGO
ENABLED CONFIG FILE PARSING
9 HOURS AGO
MISC BUGFIXES
5 HOURS AGO
CODE ADDITIONS/EDITS
4 HOURS AGO
MORE CODE
4 HOURS AGO
HERE HAVE CODE
4 HOURS AGO
AAAAAAAA
3 HOURS AGO
ADKFJSLKDFJSDKLFJ
3 HOURS AGO
MY HANDS ARE TYPING WORDS
2 HOURS AGO
HAAAAAAAAANDS
2 HOURS AGO
AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

(xkcd)