Vershon control

(using git)

Danny Awty-Carroll

May 15, 2018

Sections

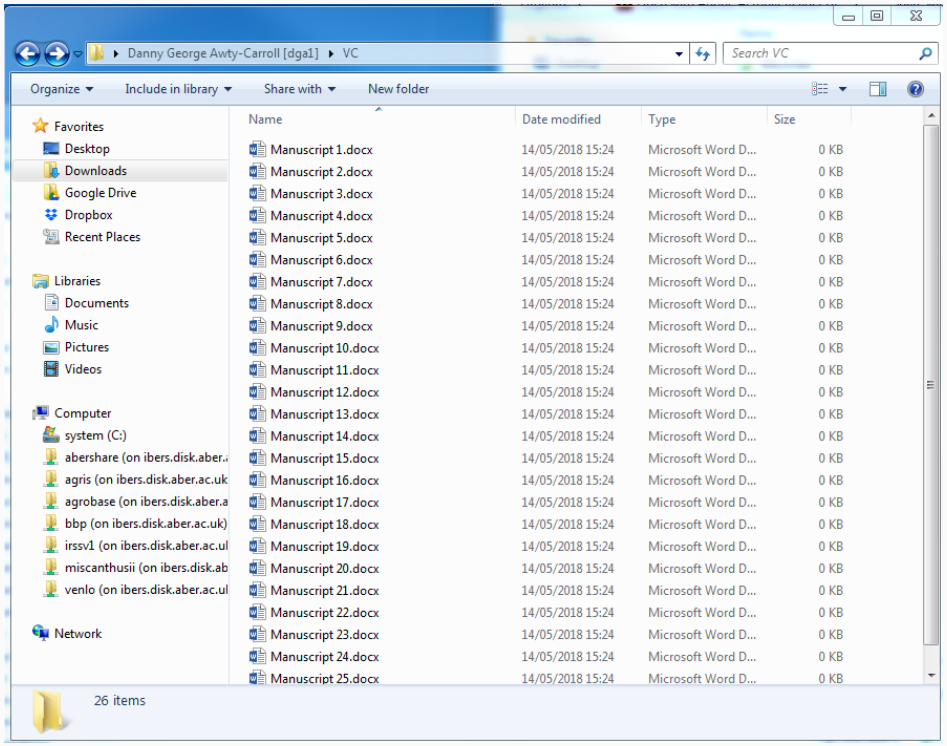
This will focus on using git in windows with a UI

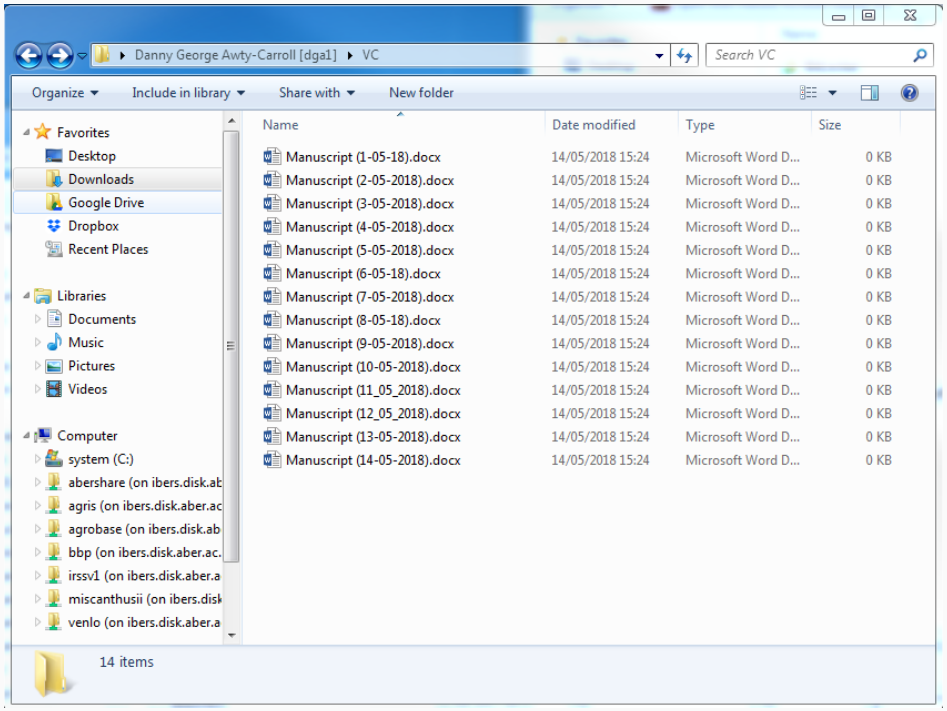
Why version control?

What is version control

This is just the manigment of vershons of a document.

One document through time.





All of us use some version control

Where things get complicated

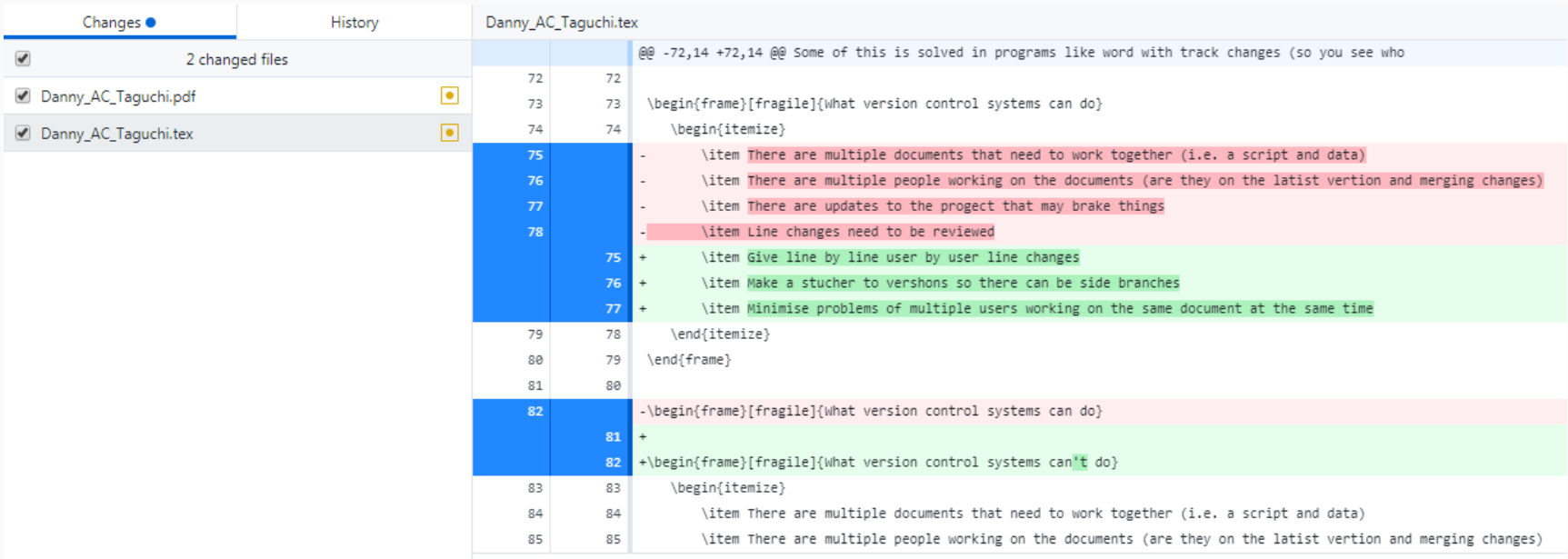
Numbering or dating documents works OK but can fall down when

- 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 latist vertion and merging changes)
- There are updates to the progect that may brake 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)

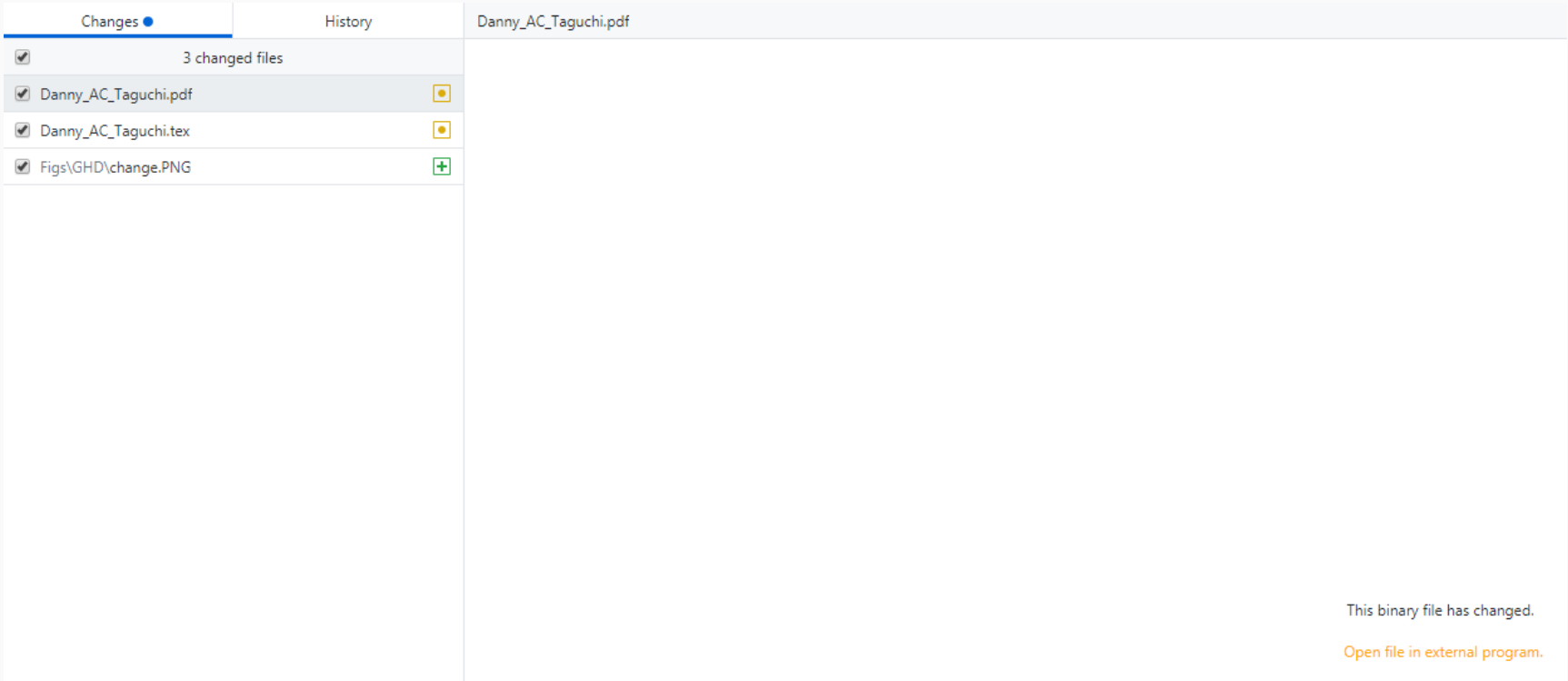
What version control systems can do

- Give line by line user by user line changes
- Make a stucher to vershons so there can be side branches
- Minimise problems of multiple users working on the same document at the same time



What version control systems can't do

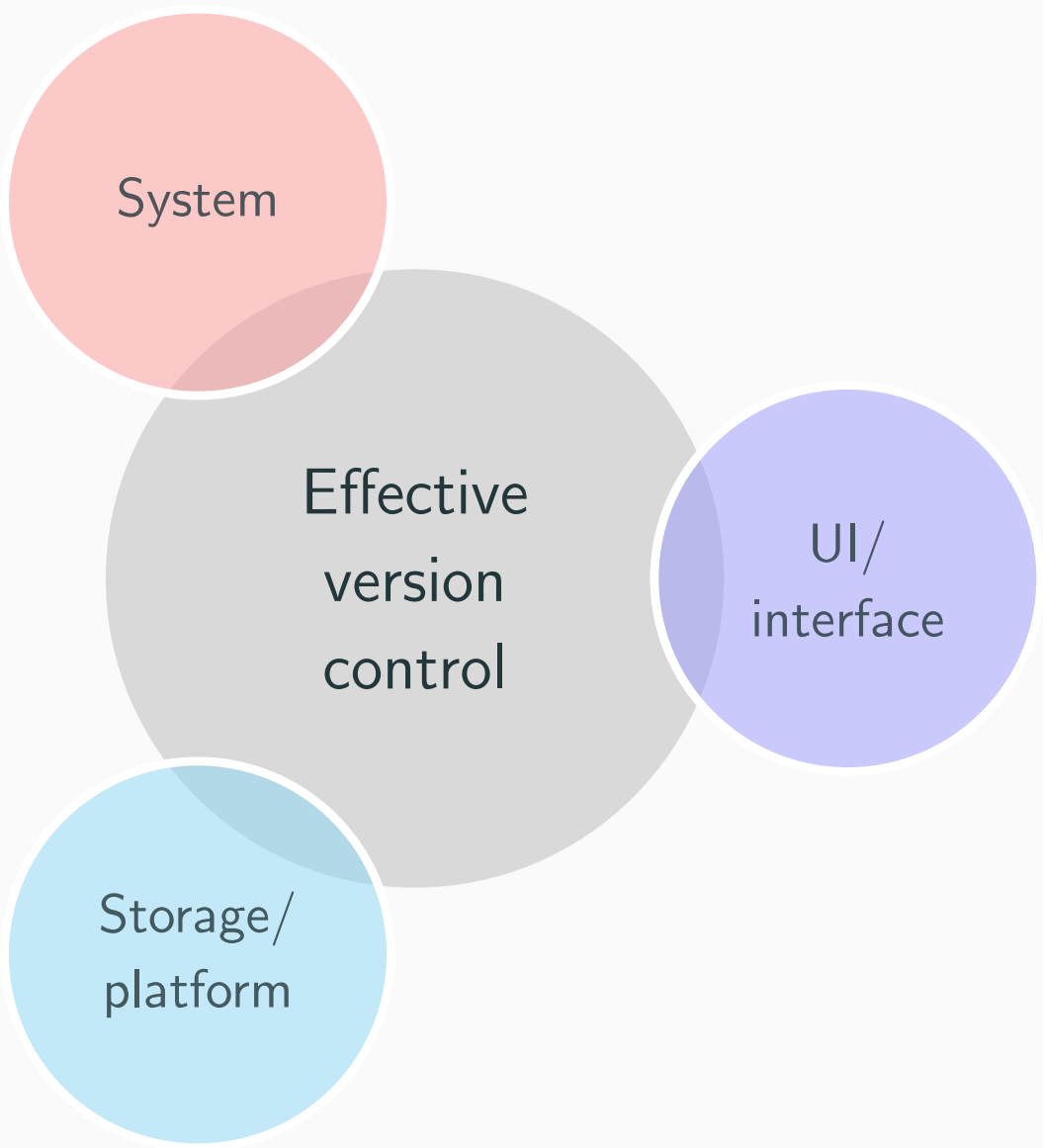
- Add much extra control to binary files (not plan text)
- Back up in real time



Version control system

What version control system?

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



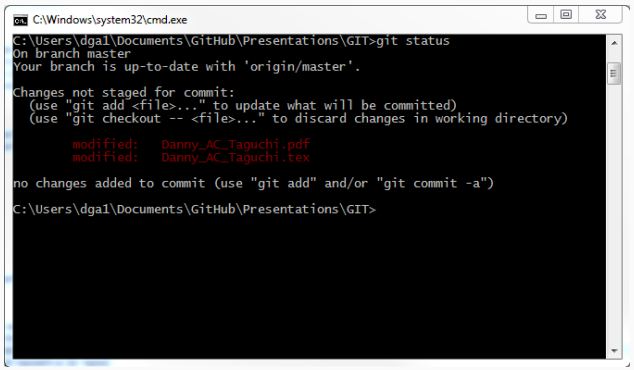
What version control system?

- There are two populer version control systems (git SVN)
- We will cover git as it is the most populer, the one I know and easyist 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 progects cauld reposetrys (or repos)
- It then has a tree struchure to manige the vershons

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 gray with colour branches coming of and then being merged back

Key comards?

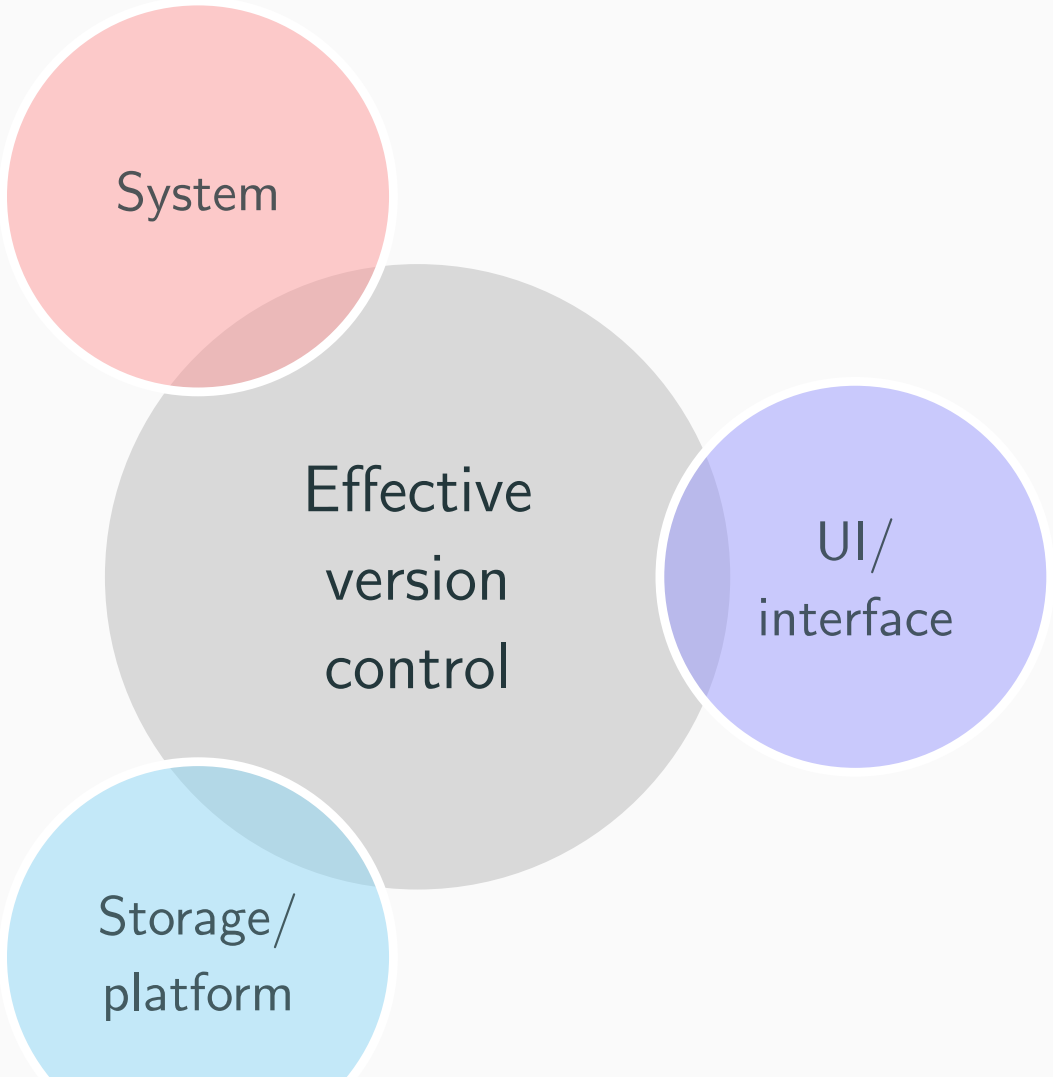
In the terminal interface there are some useful commards:

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

Platform

Which platform?

You can just use git with the treminal on your PC or you can store work on an online reposetry mainging site, the most populer of these are GitHub and Bitbucket



Why not just store the repostry on dropbox?

You can and it would be baked up and sherable, but there are problems:

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

Which platform?

Bitbucket

- Alows unlimetd public or privet repostorys
- Charges per colcolaberaters over 5 on the repostorys
- 1GB stroige for large flies all repostorys, 5GB if using a .ac email

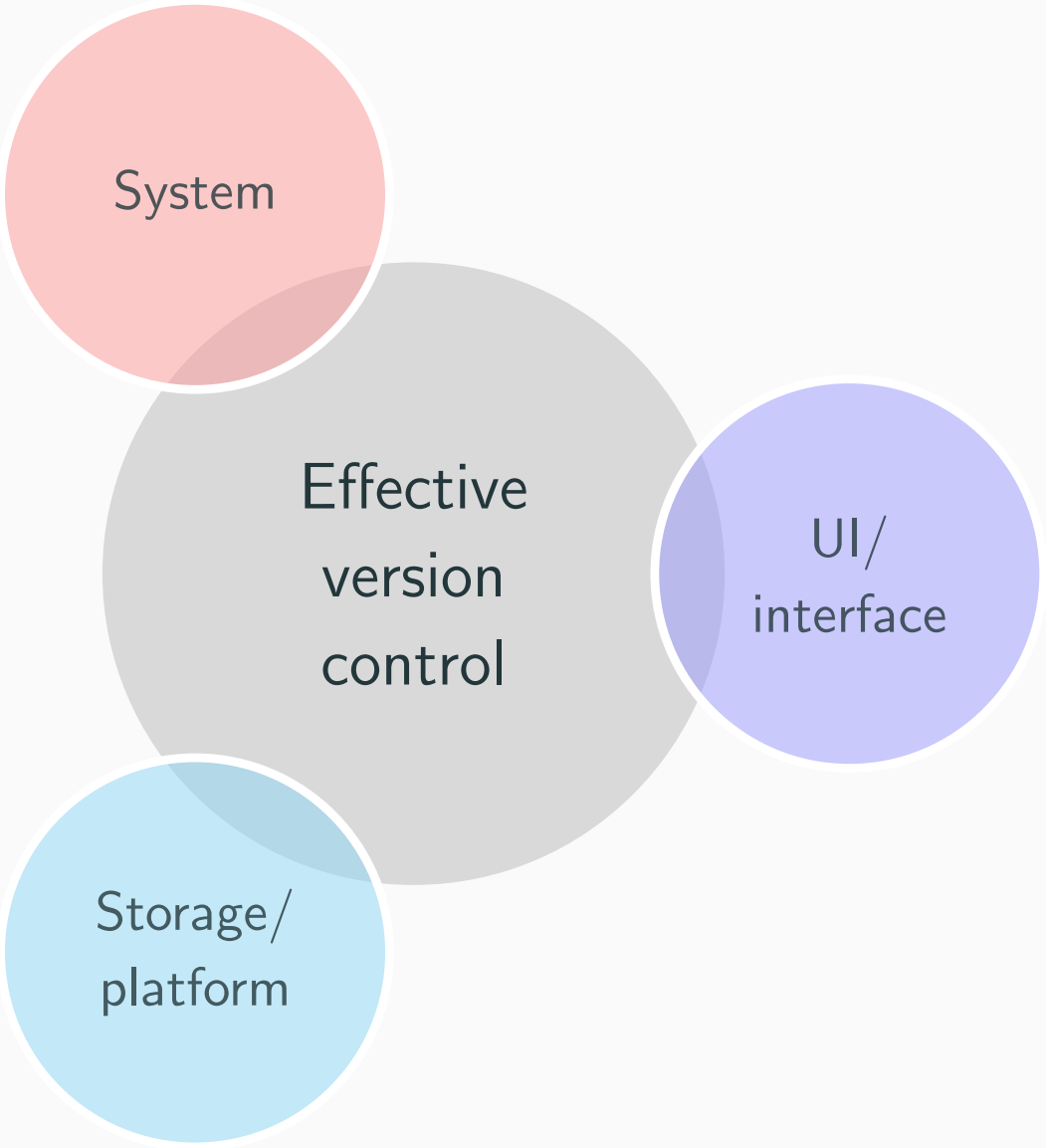
GitHub

- Has unlimeted public repostorys
- With unlimeted collaborators
- Pay for privet repostorys
- However as a student or academic you can sign up for free privet repostorys
- Github is about $\times 4$ more populer than Bitbucket

UI

Which UI?

You can just use git with the treminal but there is a frendlyer interfase

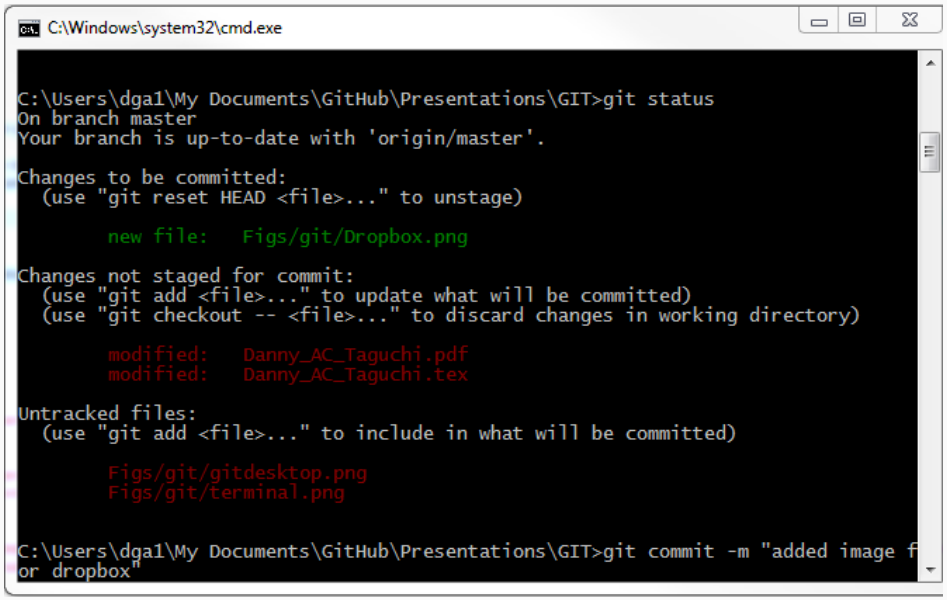


Which UI?

The Terminal

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

- If you are not using it enyway or are happy using it a graphical UI is nice
- You don't need to remember commands or reposty names

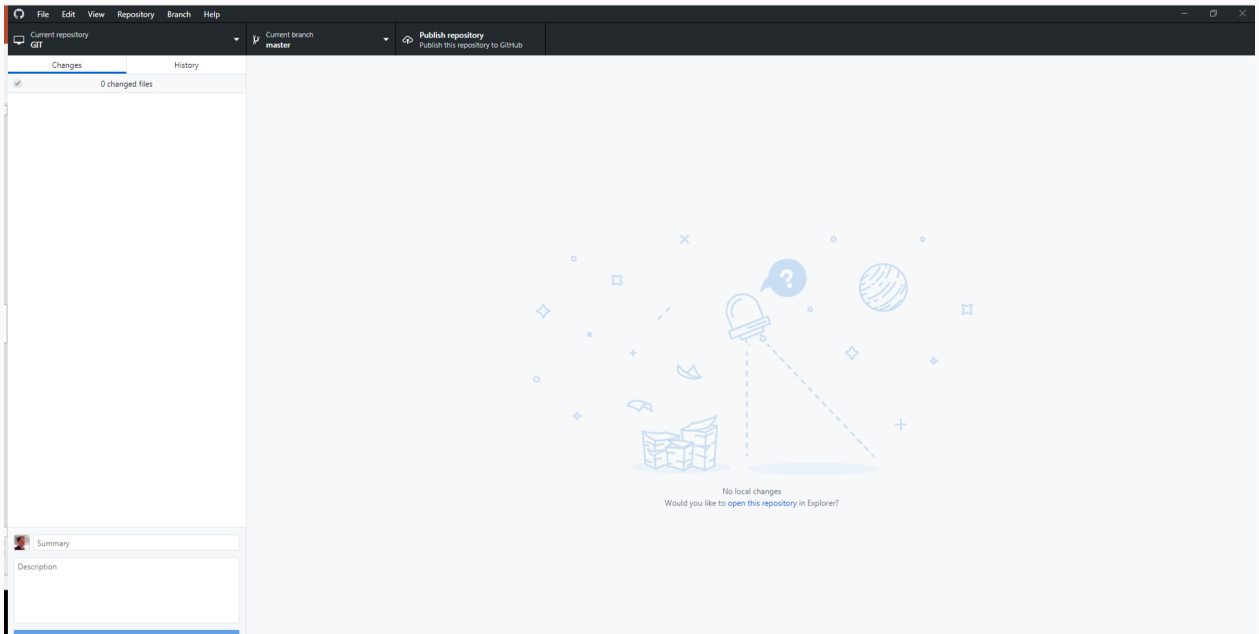


Which UI?

Git Desktop

The easyist to use and most populer UI is git desktop:

- This is made by github and intigrates very well with github or local (on pc) reposotrys
- It will work with other hosting platforms but less easly
- Is the easyist to use
- Will install git for widows
  `https://git-scm.com/download/win`

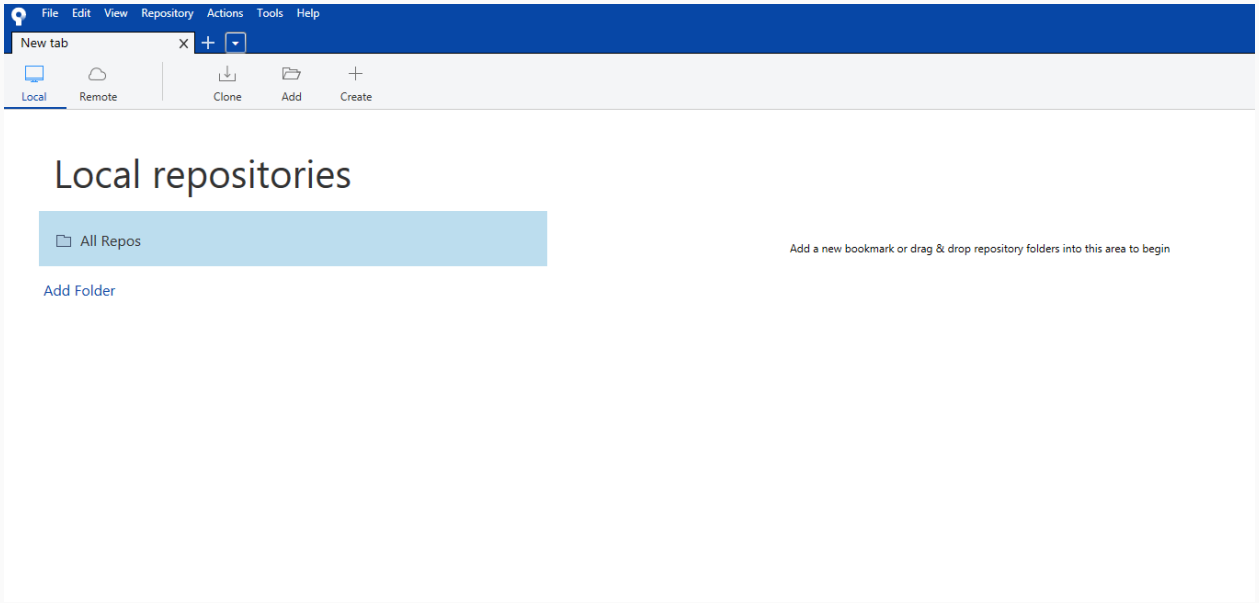


Which UI?

Sourcetree

The hosting platform indipendet:

- Sourcetree is uned by atlassian who ouns Bitbucket
- It will work with other hosting platforms easly
- It has more complex controlls
- Will install git for widows
  `https://git-scm.com/download/win`



Using git

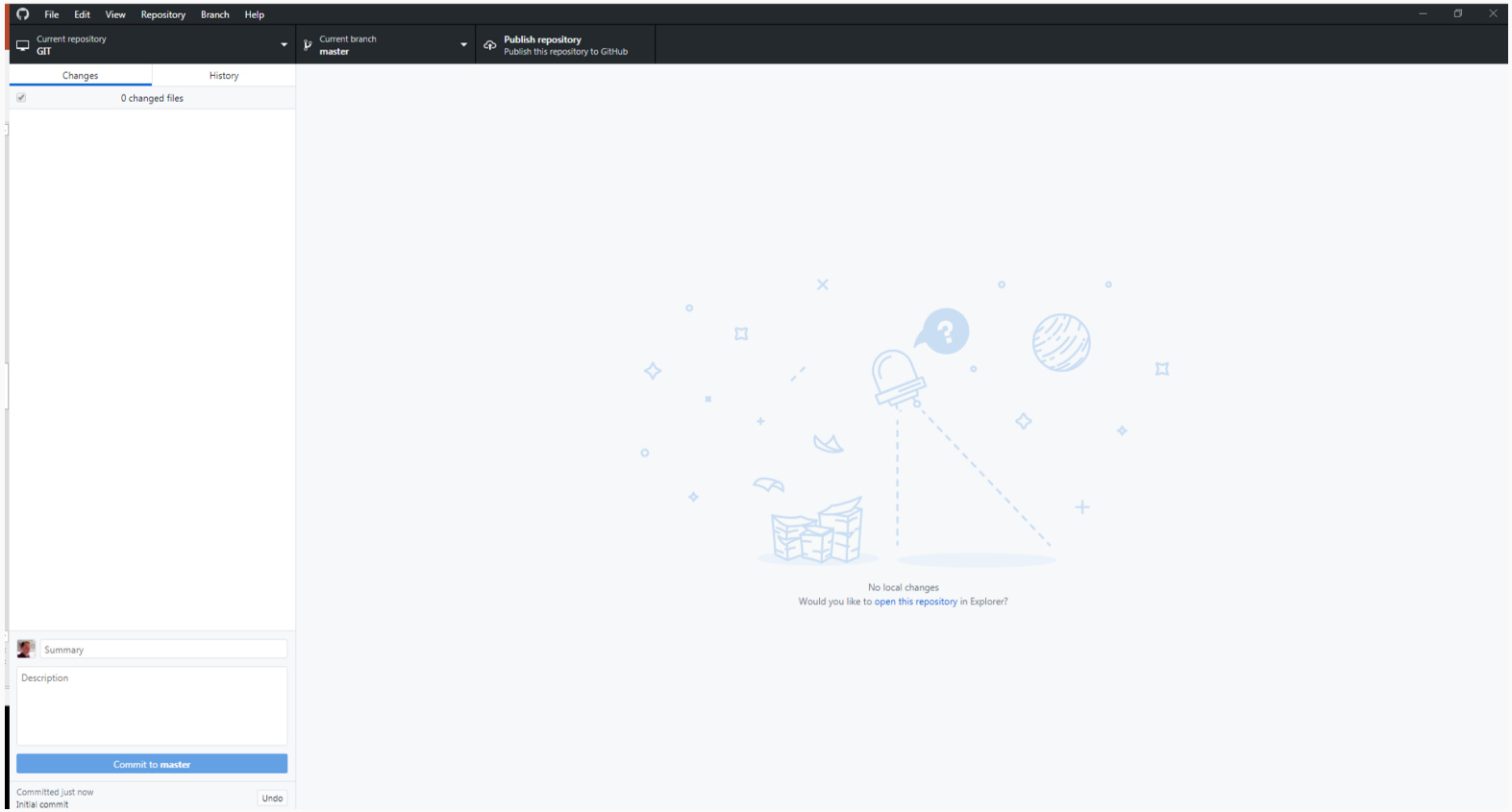
How to use git

This will assume use of githithub and gitdesktop

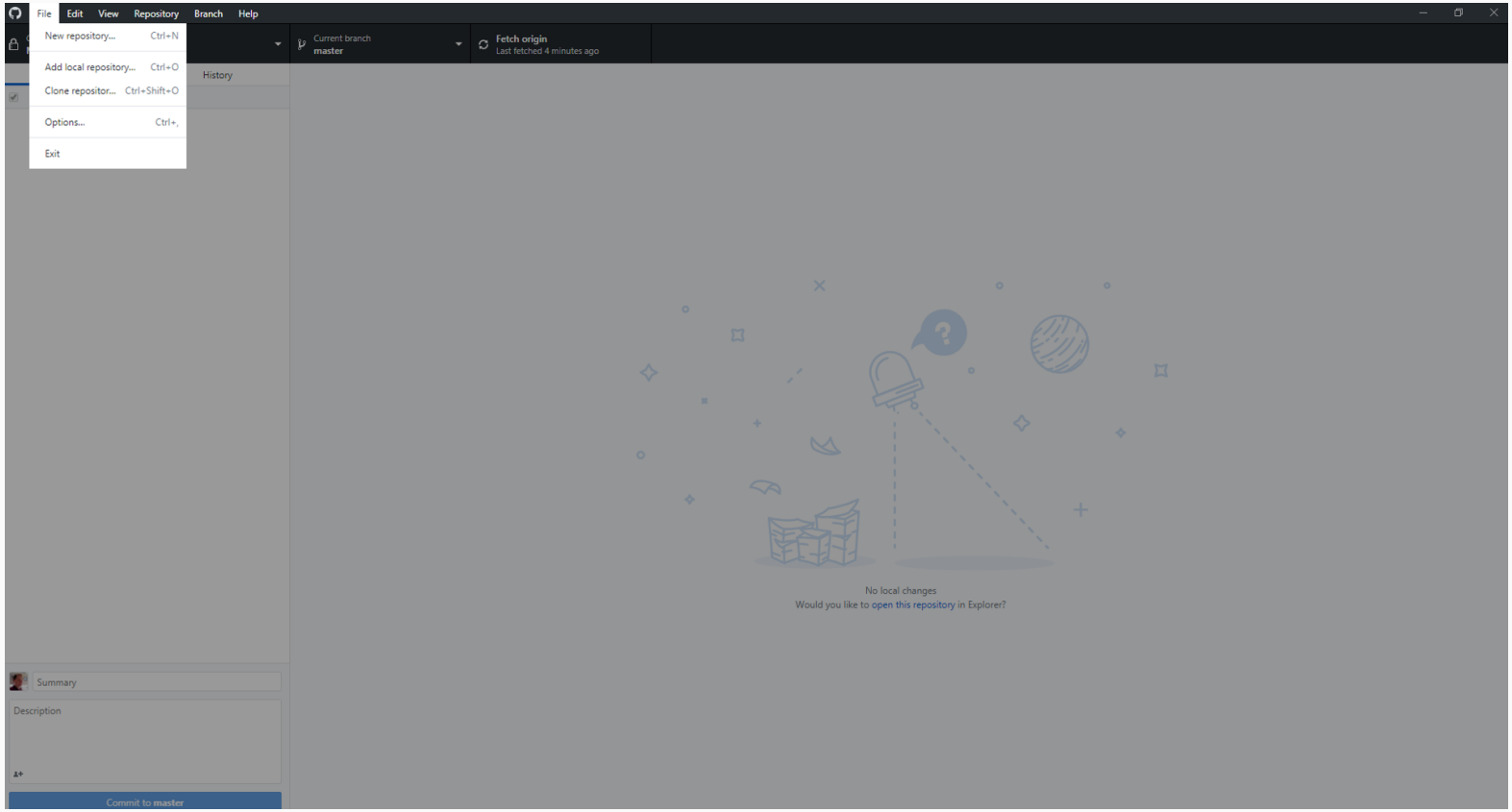
`https://desktop.github.com`

Following is a bref dmonstration of the interface

Adding a new repostry

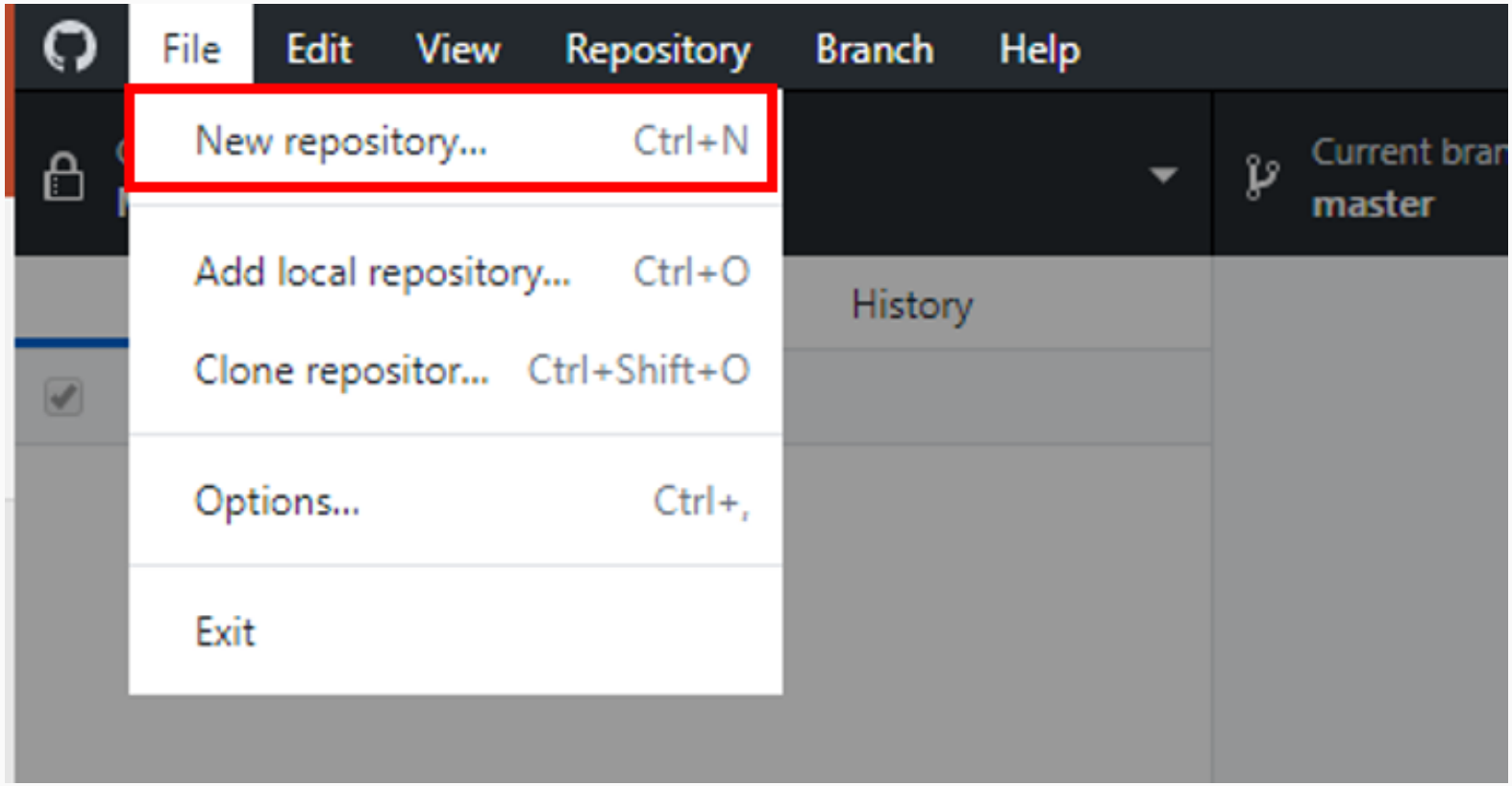


Adding a new repostry



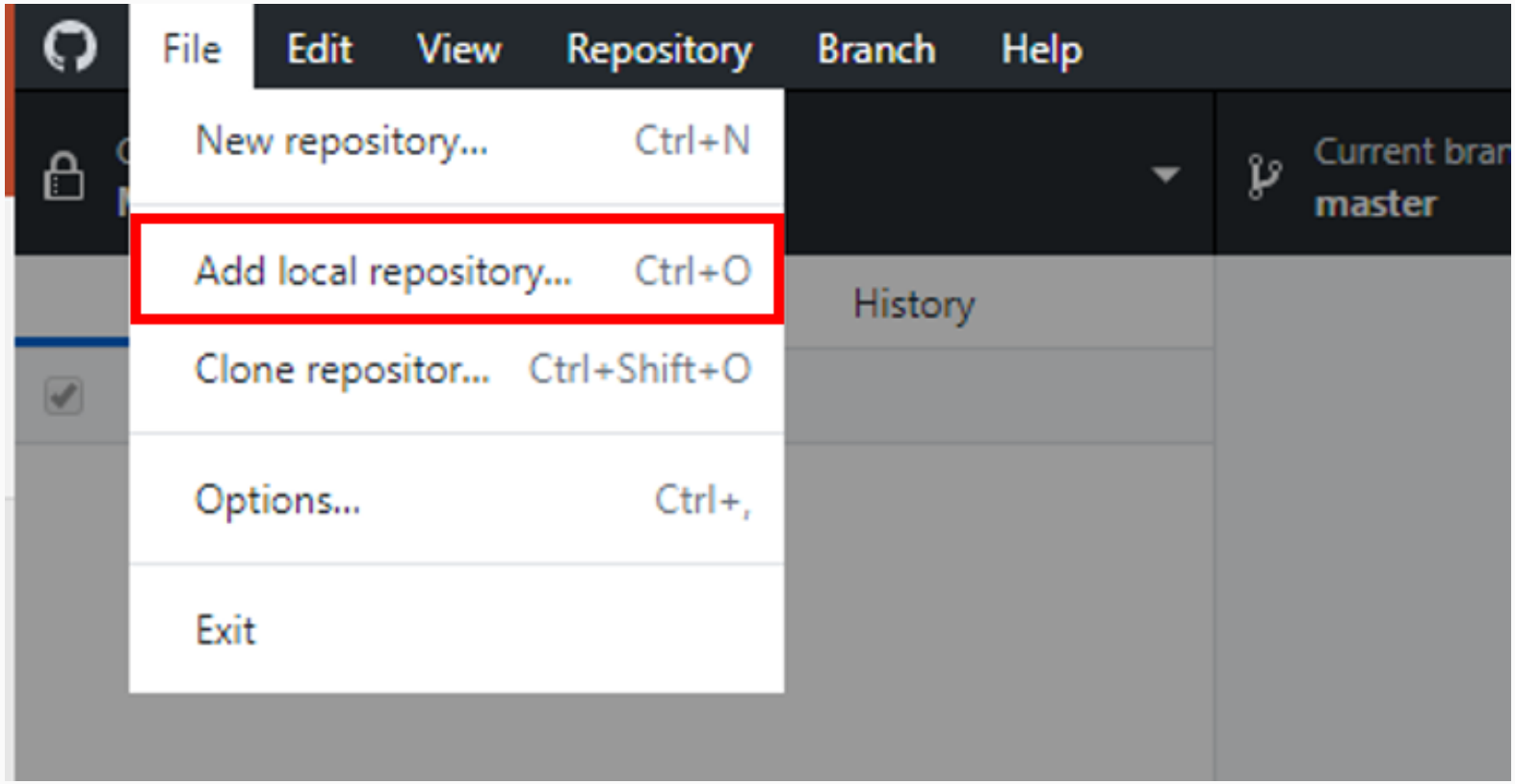
Adding a new repostry

This makes a totally new reposi-tory (with no '.git' folder)



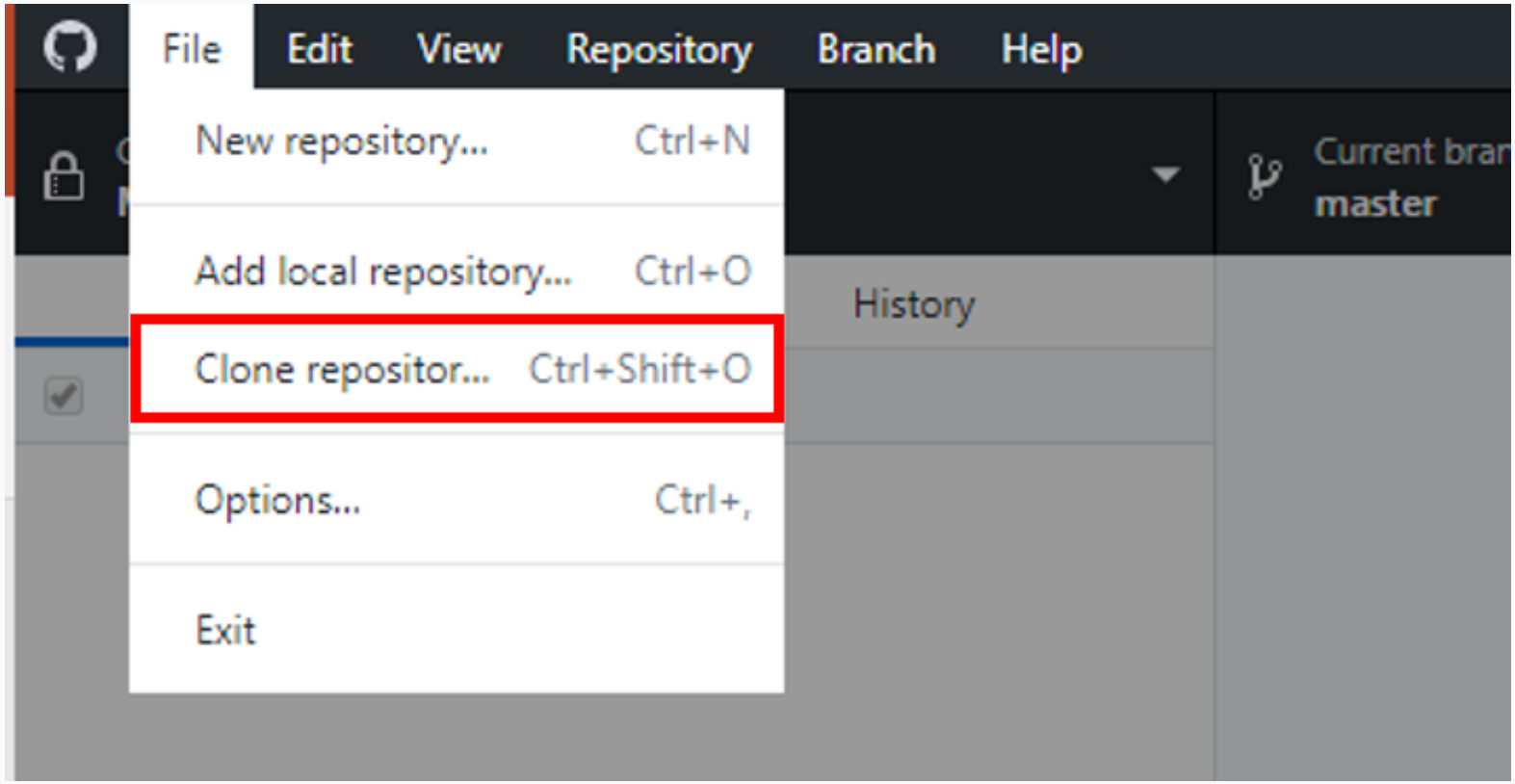
Adding a new repostry

addes an existing reposi-tory (with a '.git' folder)



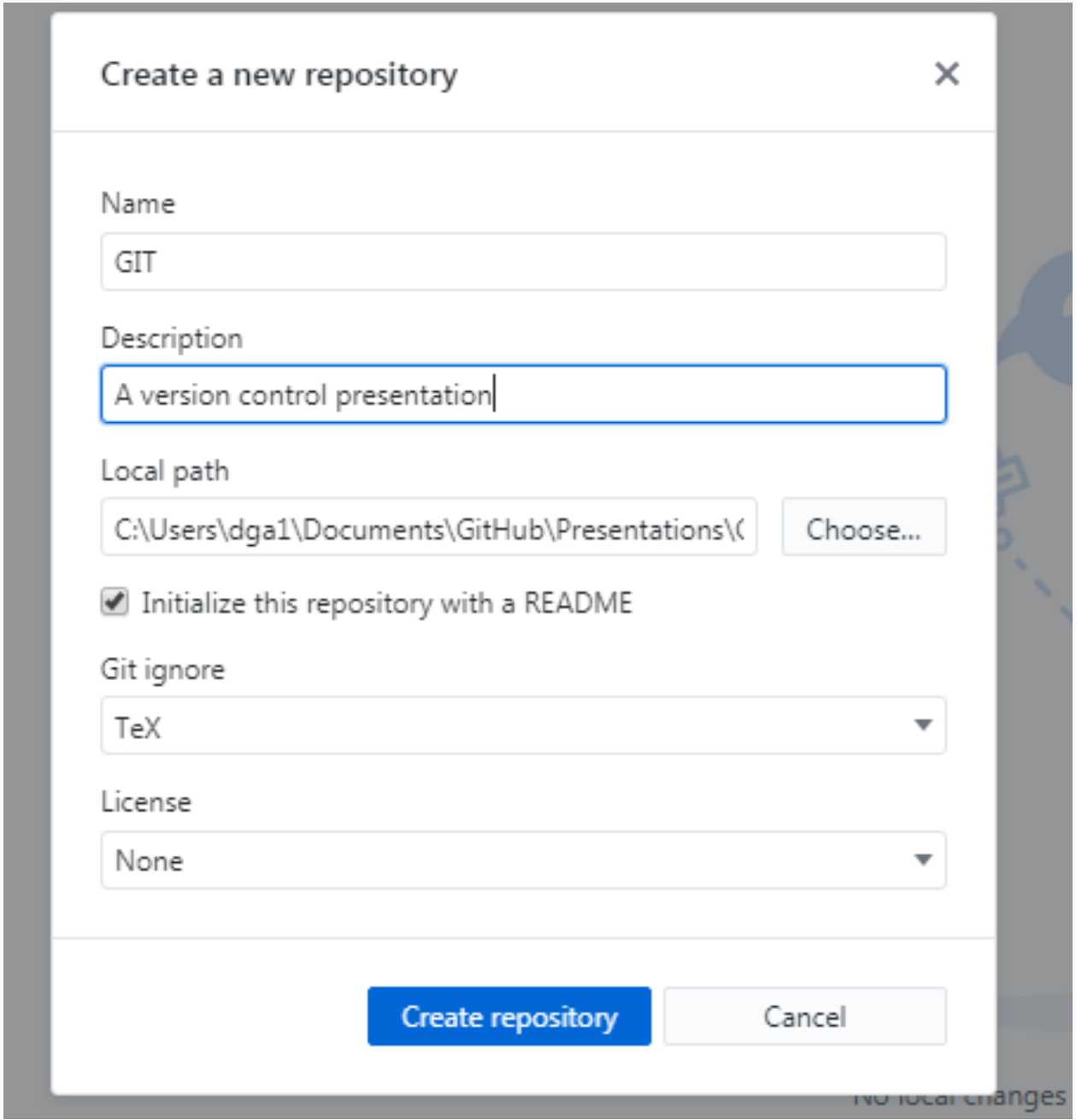
Adding a new repostry

Copys an online reposi-tory to the PC

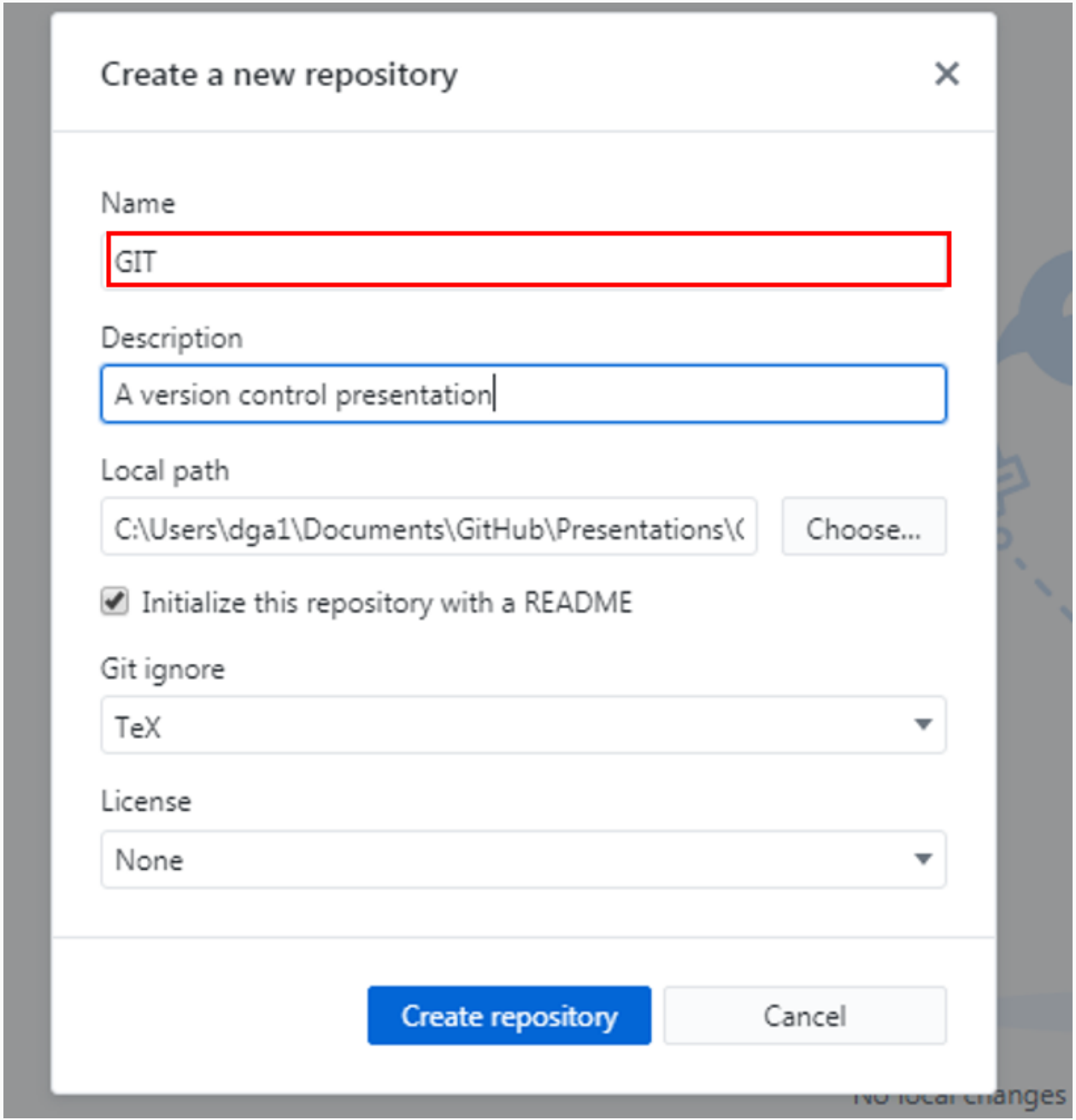


Adding a new repostry

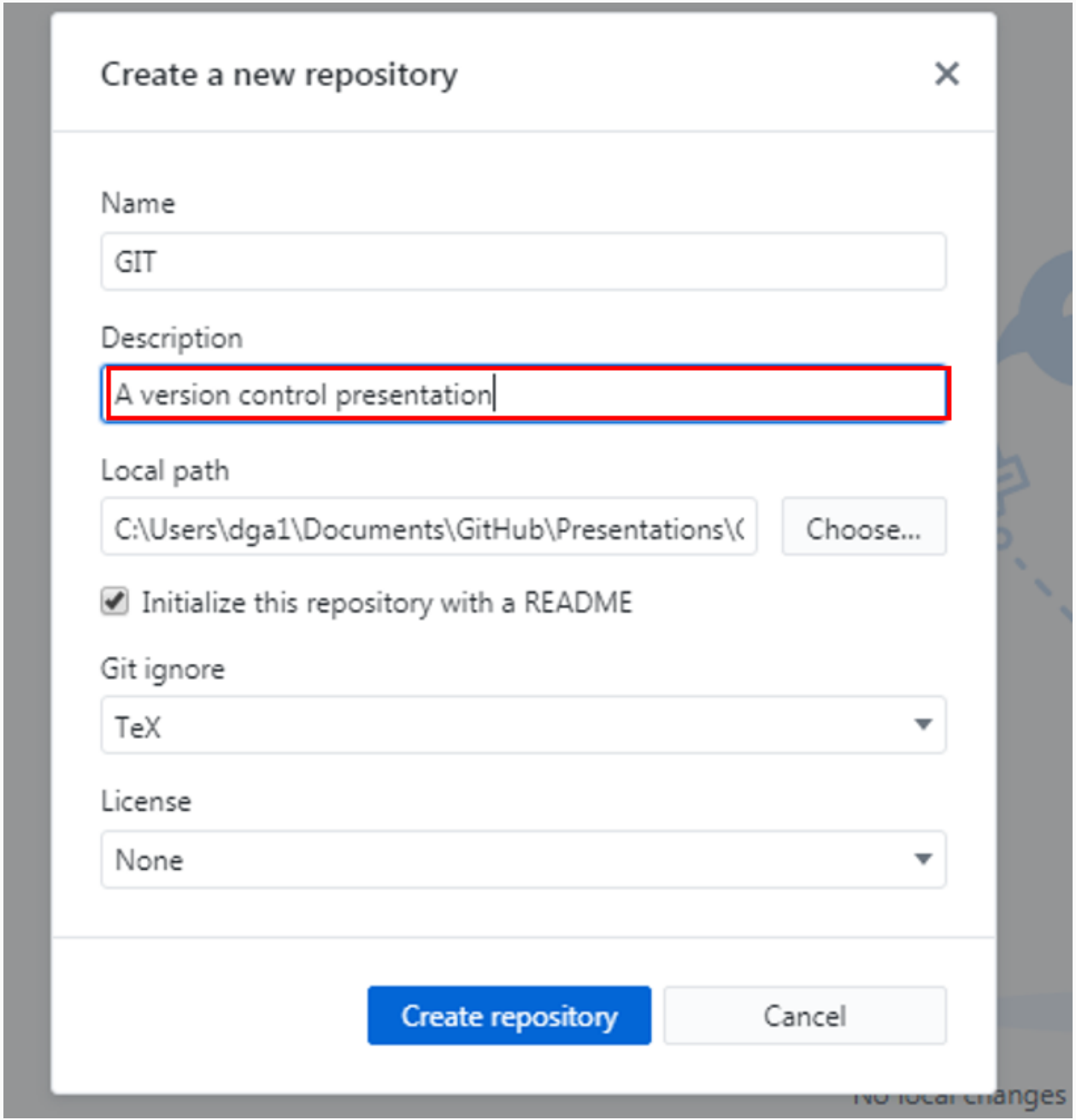
new repository menu



Adding a new repostry

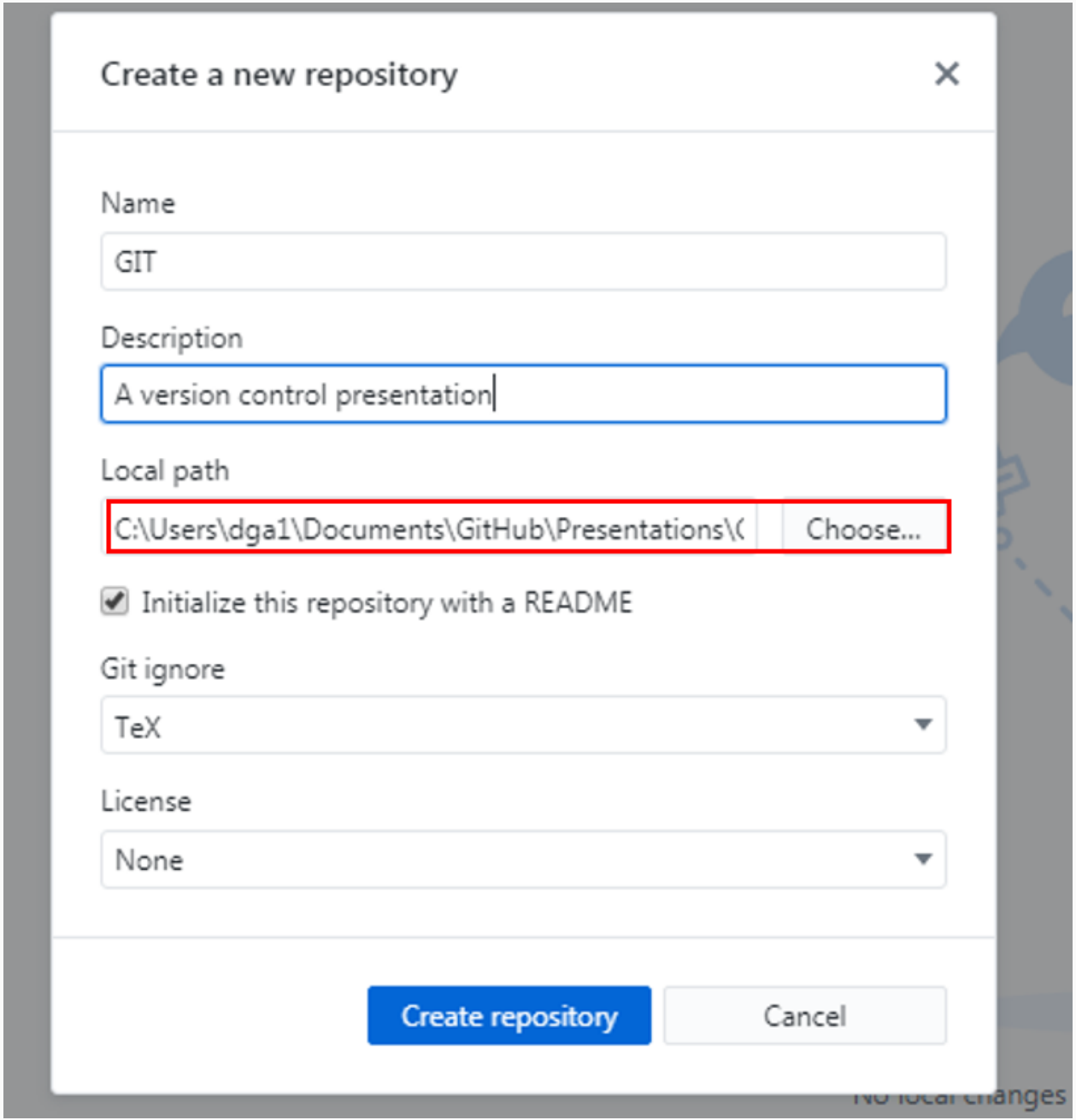


Adding a new repostry



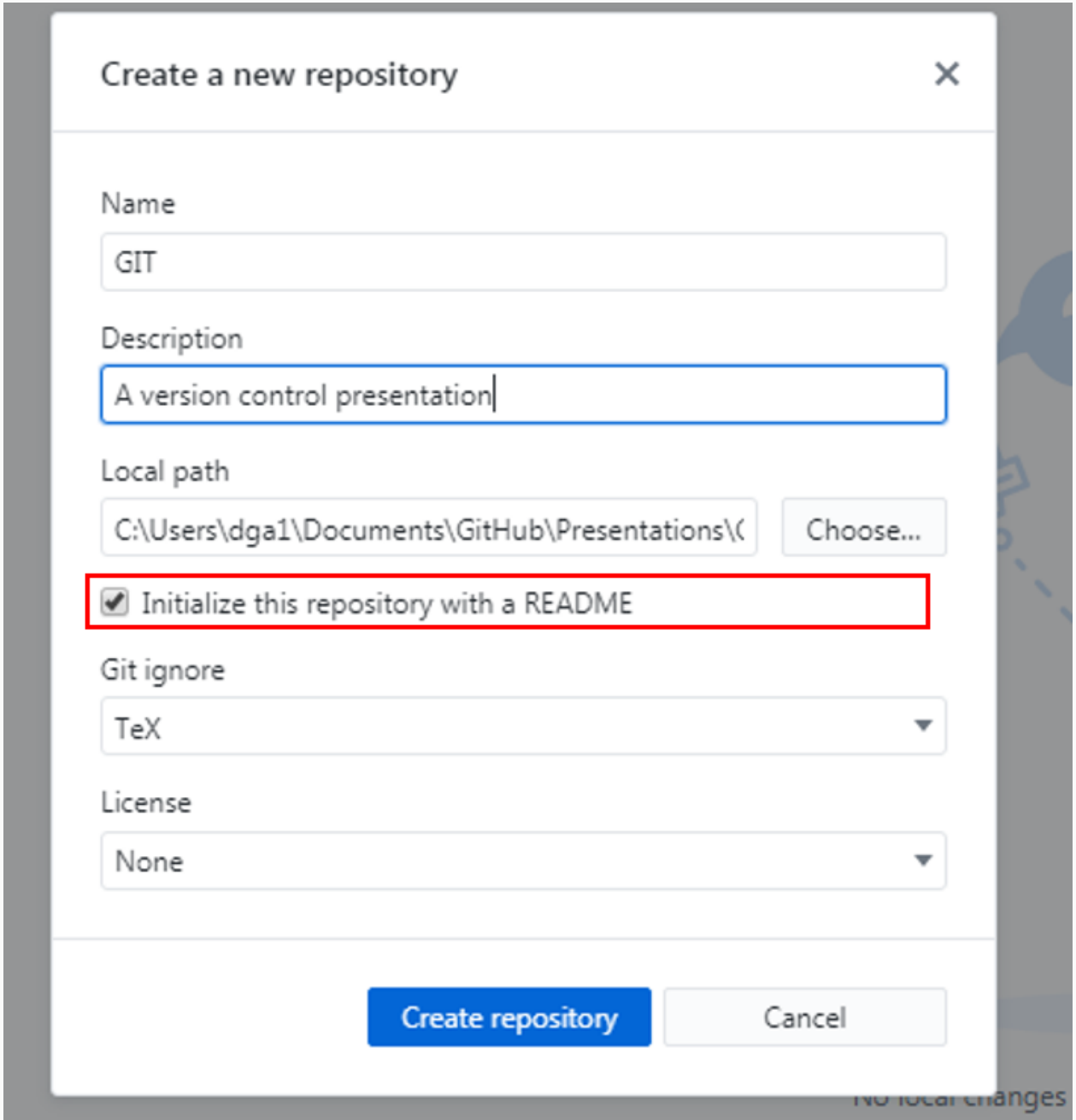
Adding a new repostry

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



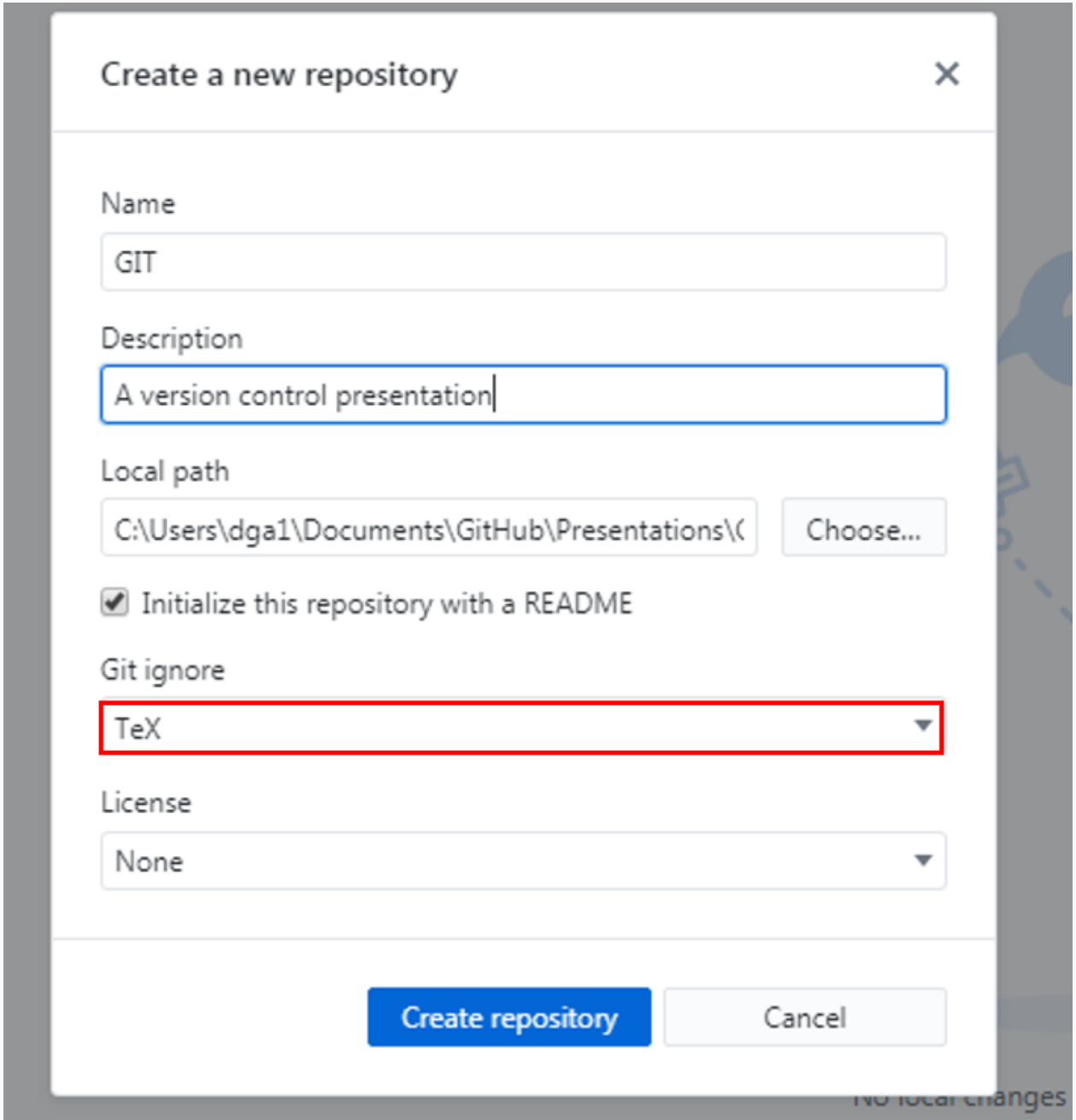
Adding a new repostry

This will inishalise the repositry with a readme containing the discription.

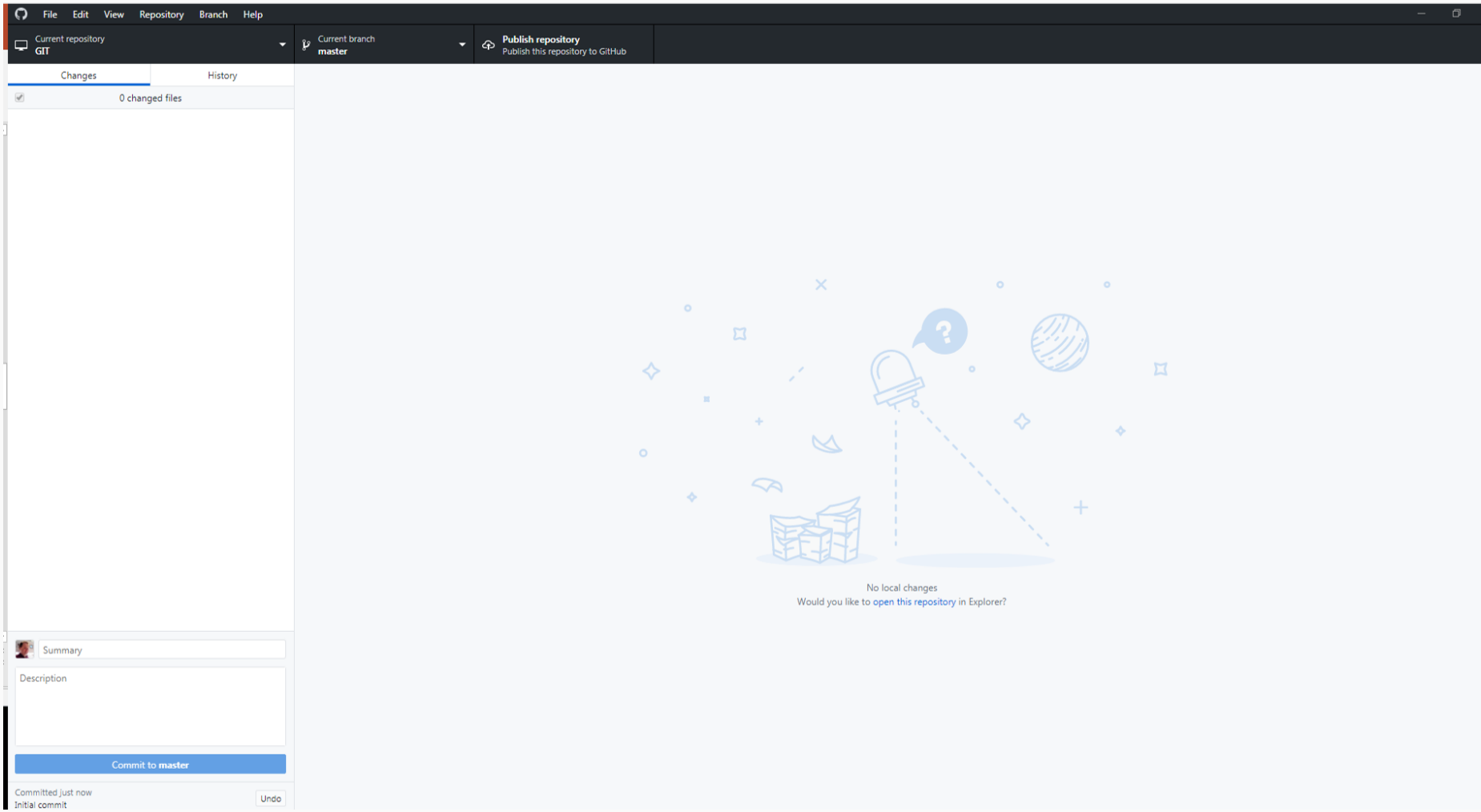


Adding a new repostry

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

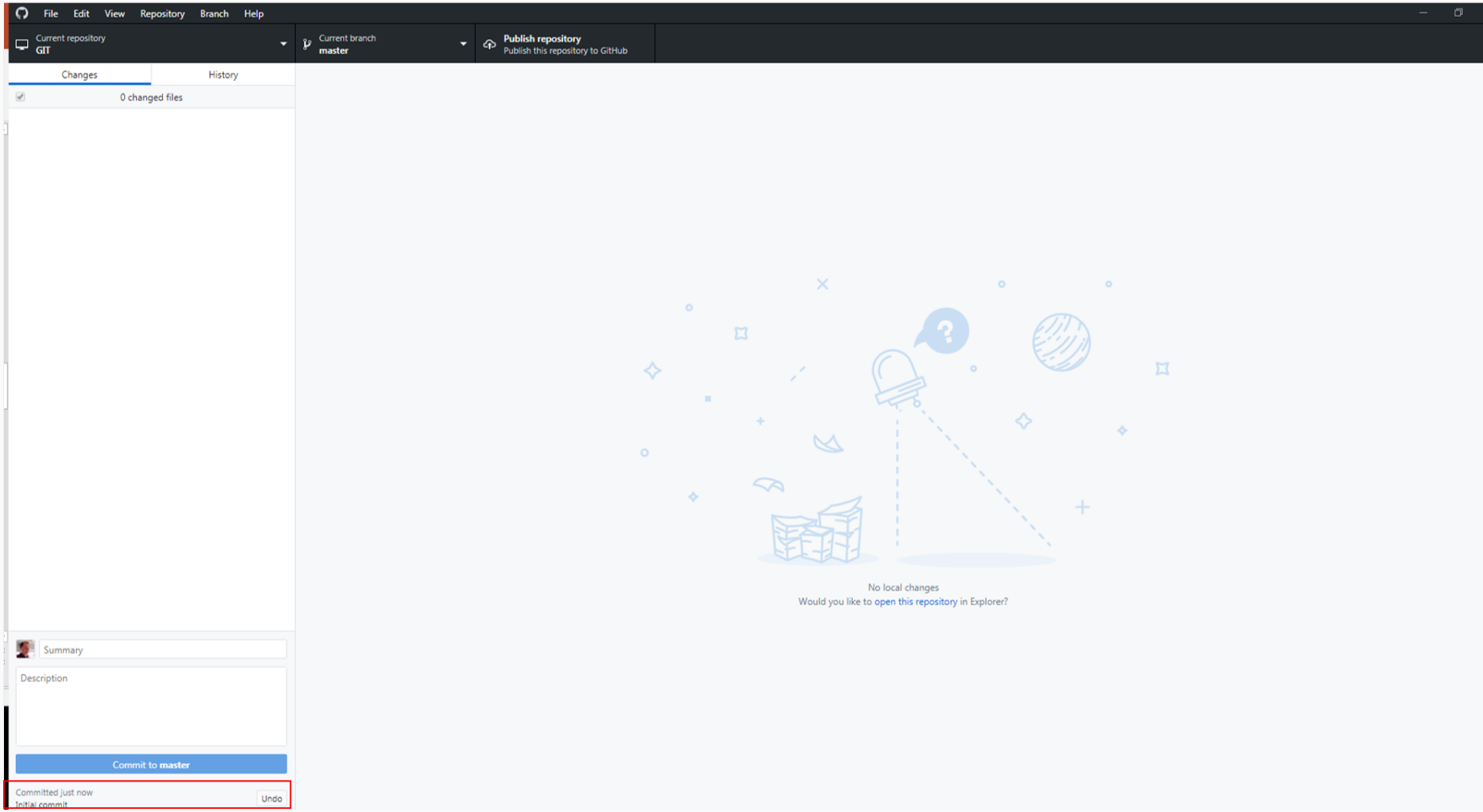


Using the repostry



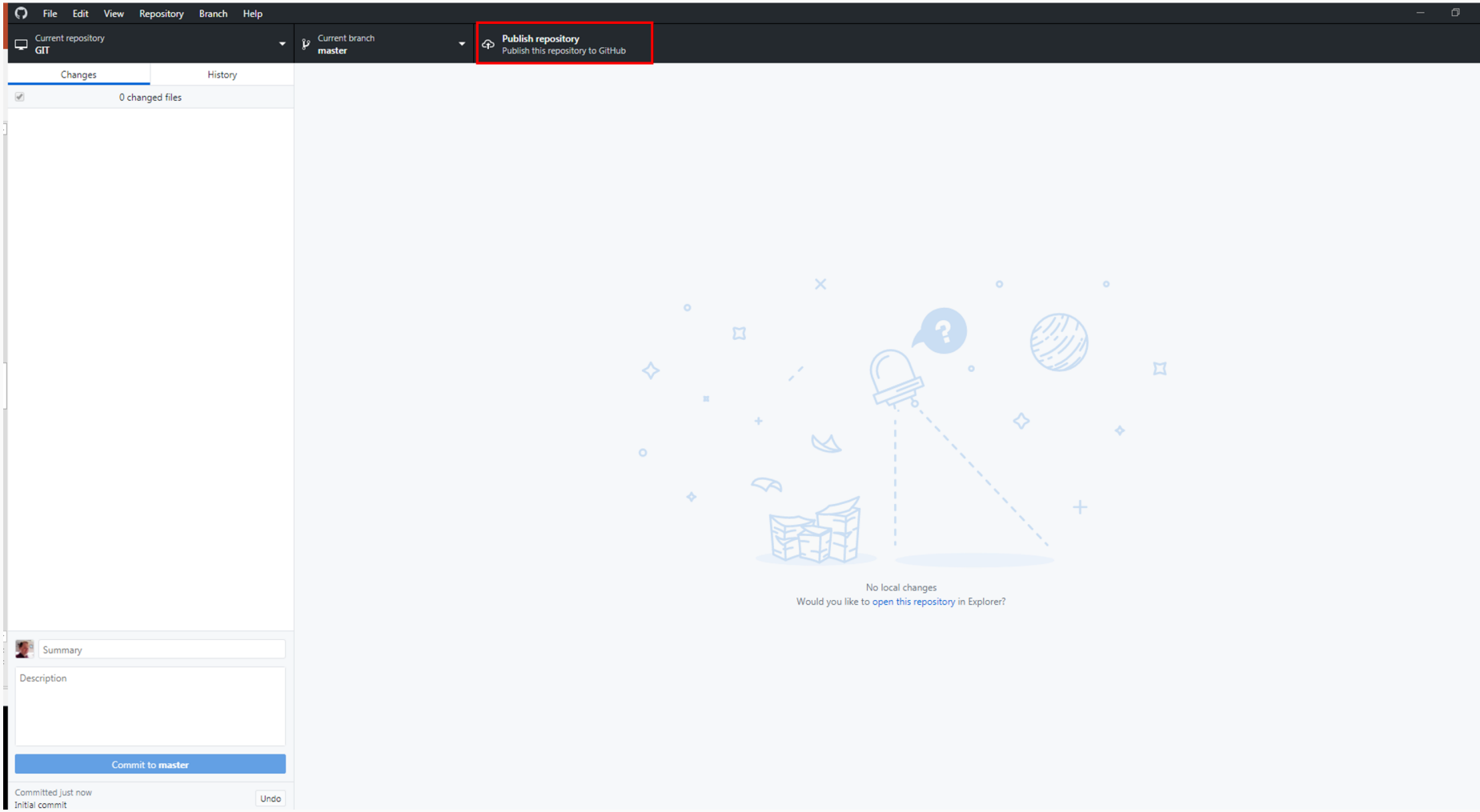
Using the repostry

Now we have a inshal commit of the readme done



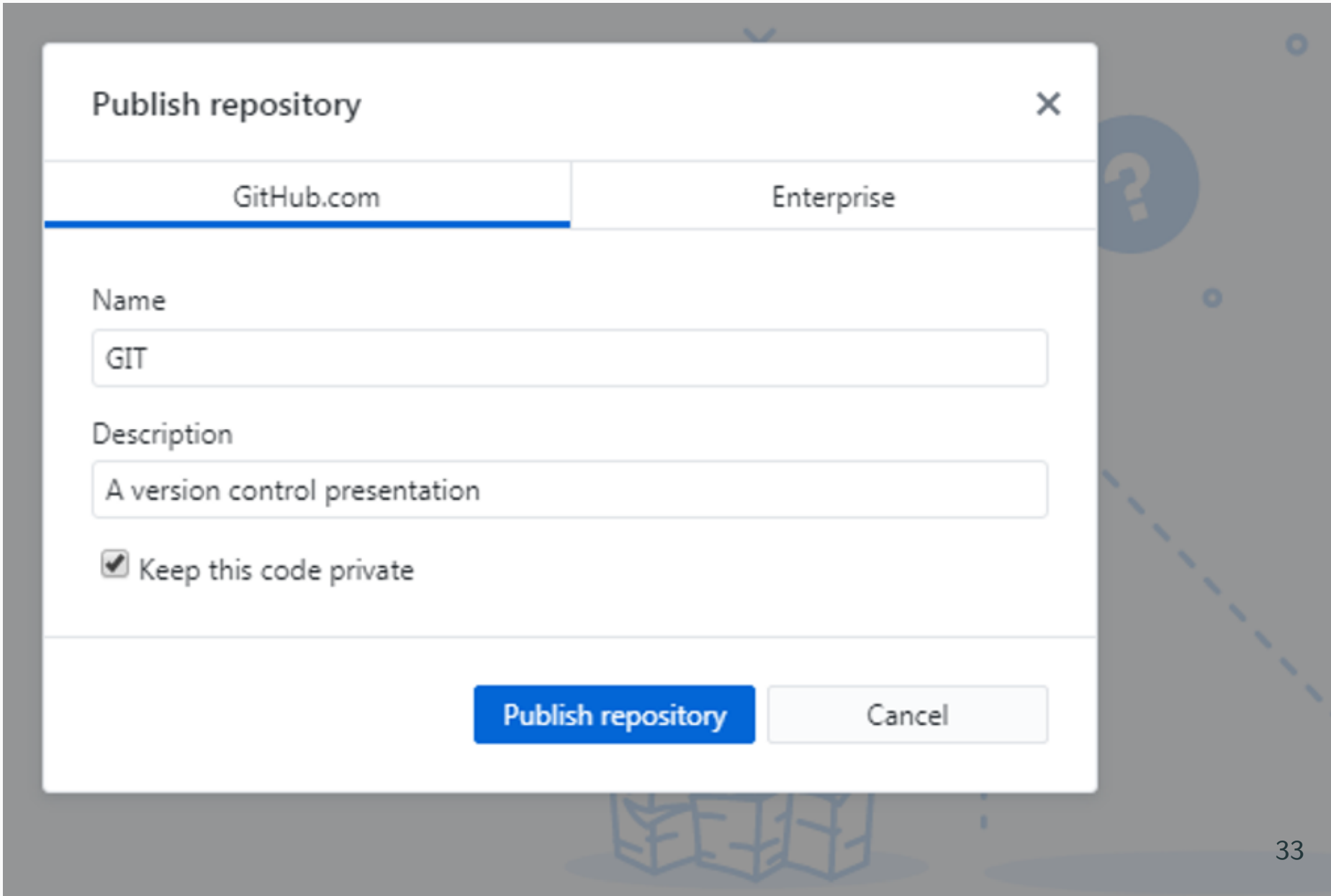
Using the repostry

We can carry on making local commits or or publish the repostry to github



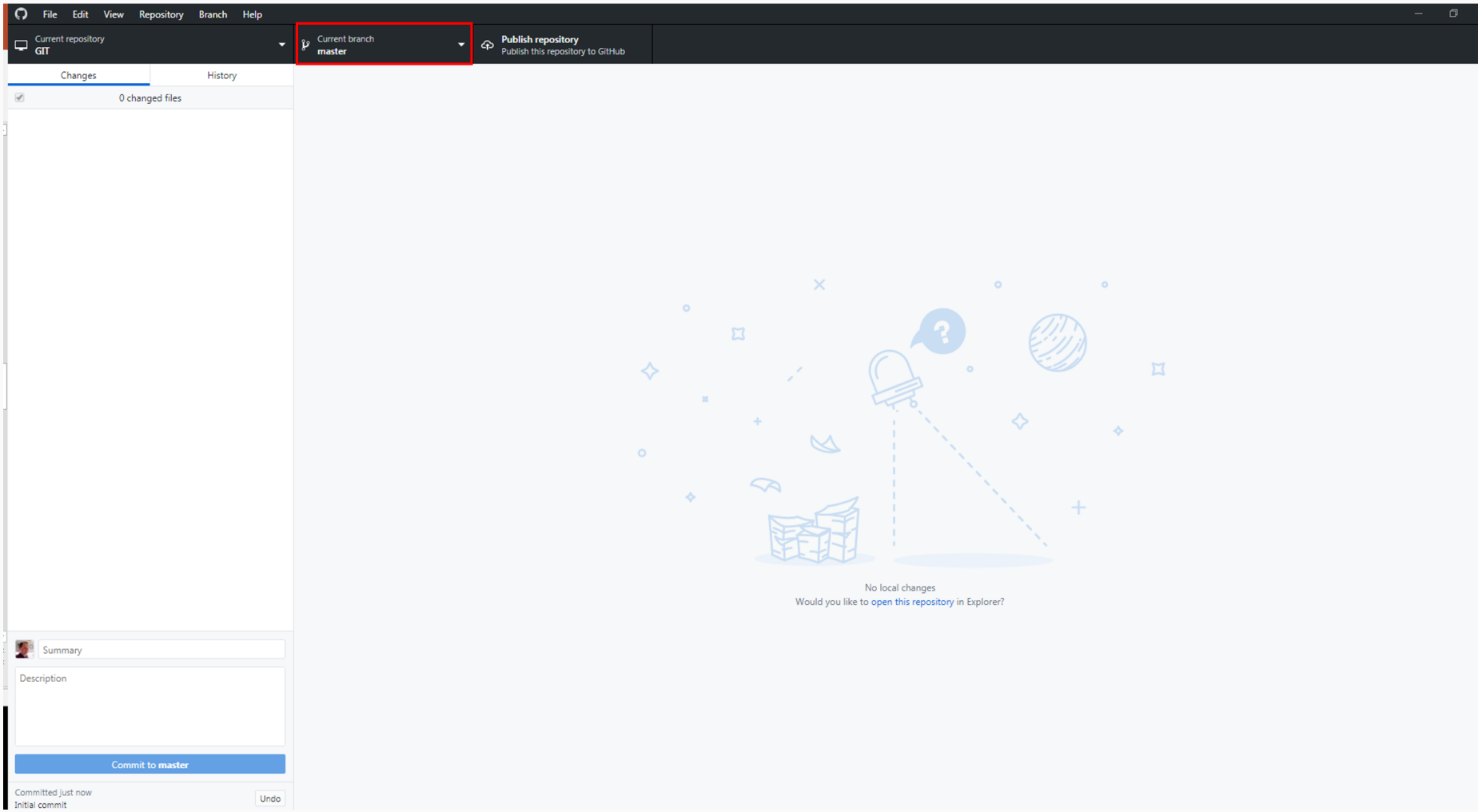
Using the repostry

We can publish to github



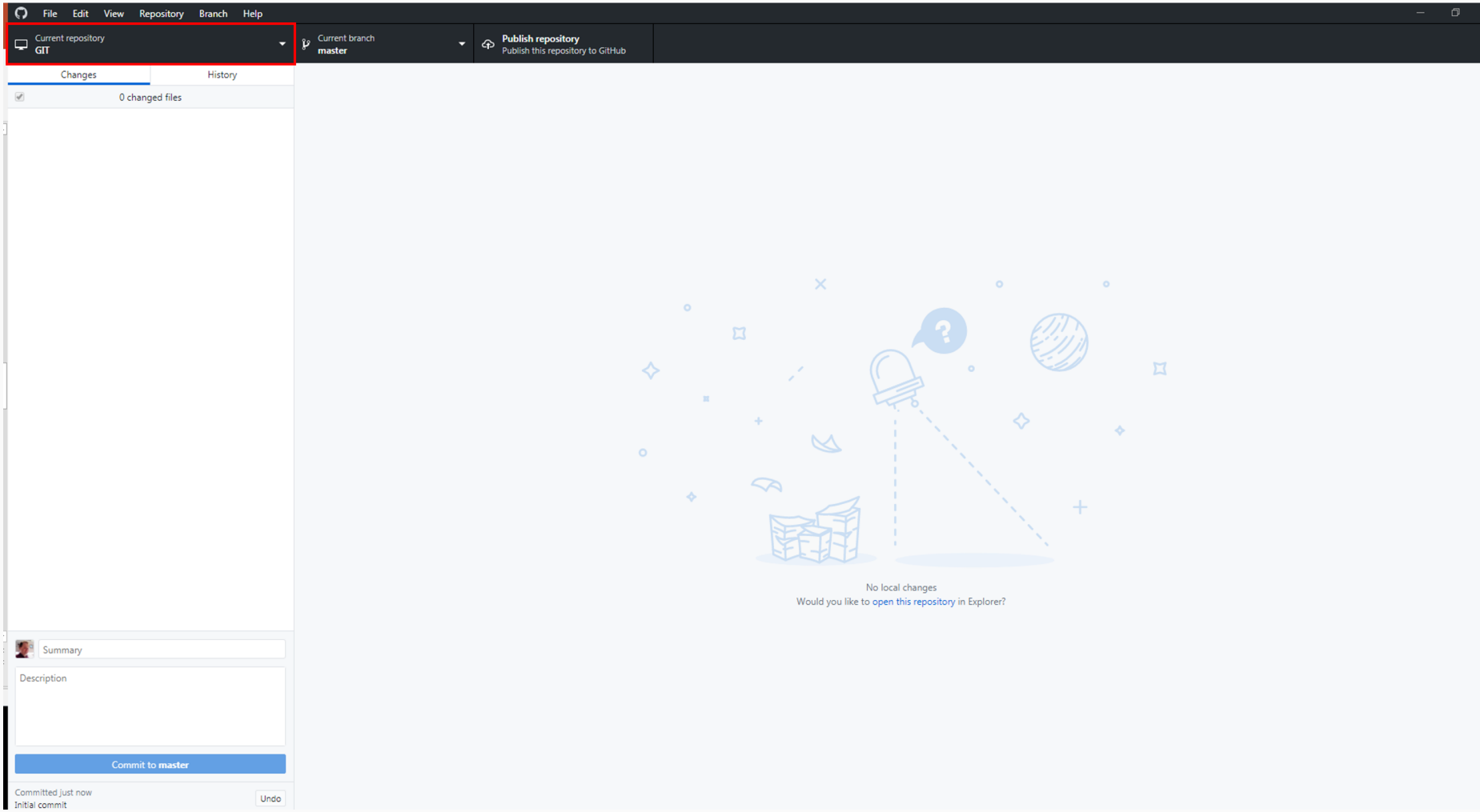
Using the repostry

Make branches



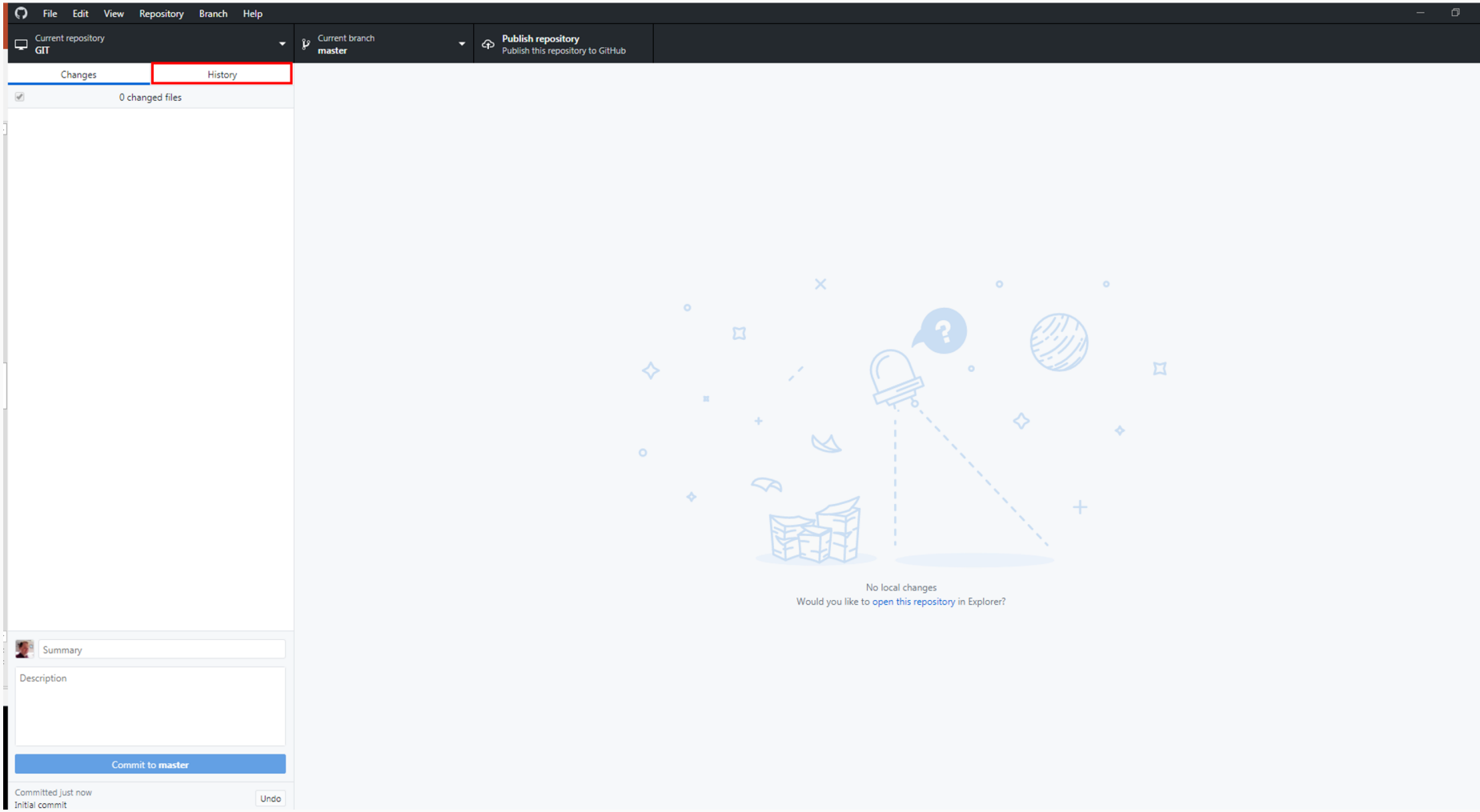
Using the repostry

swich to another reposotory



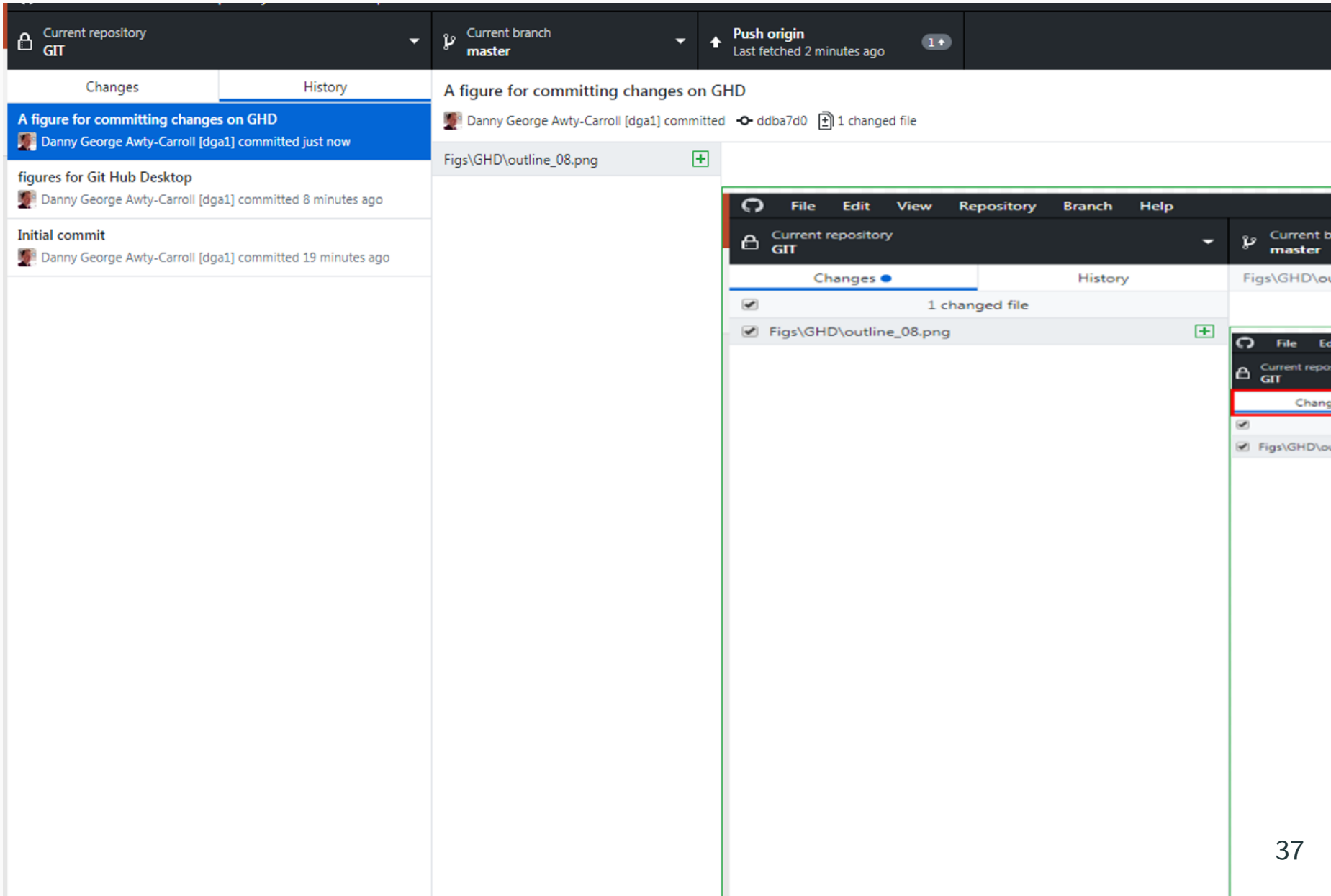
Using the repostry

look at the history of this repository



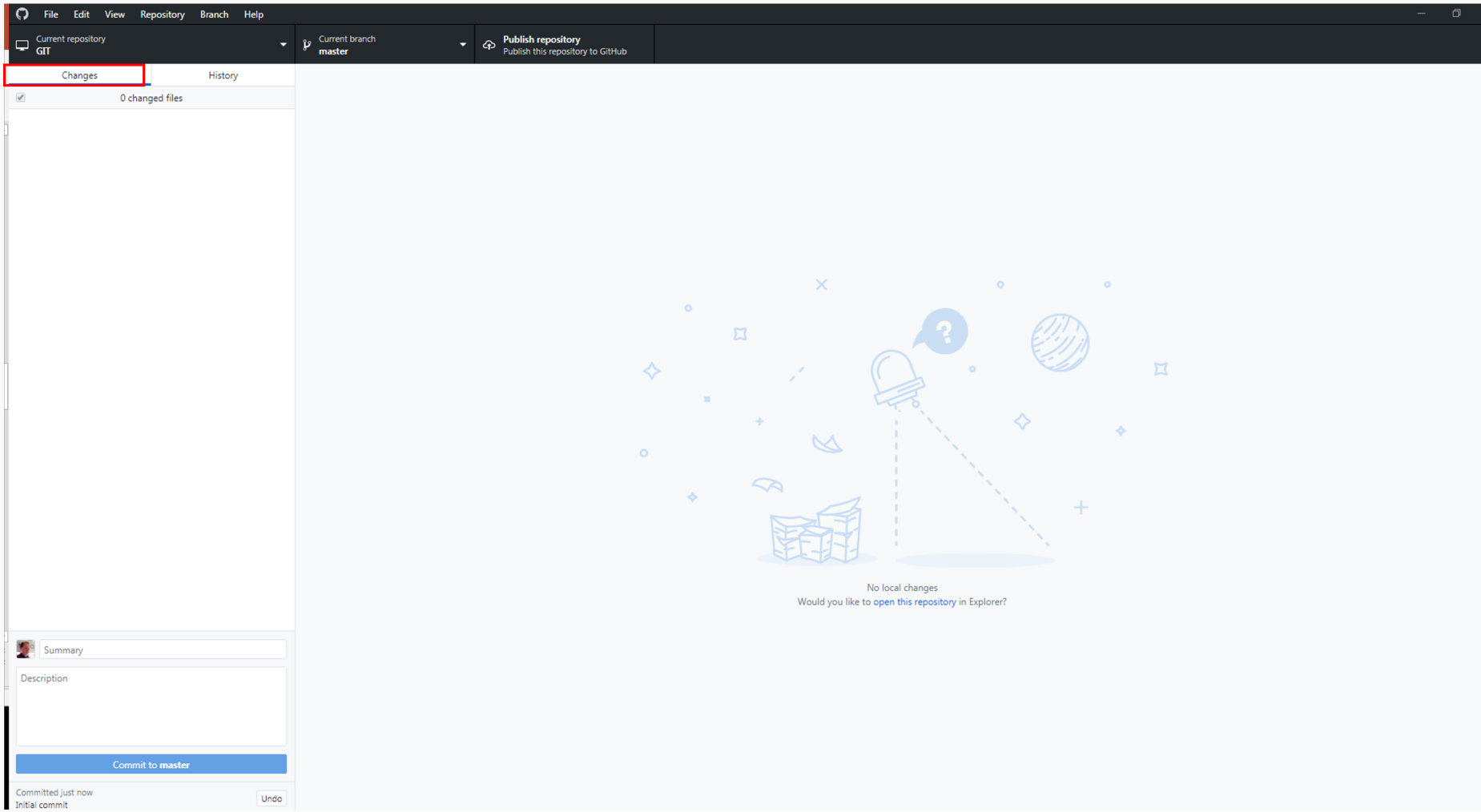
Using the repostry

look at the history of this repository



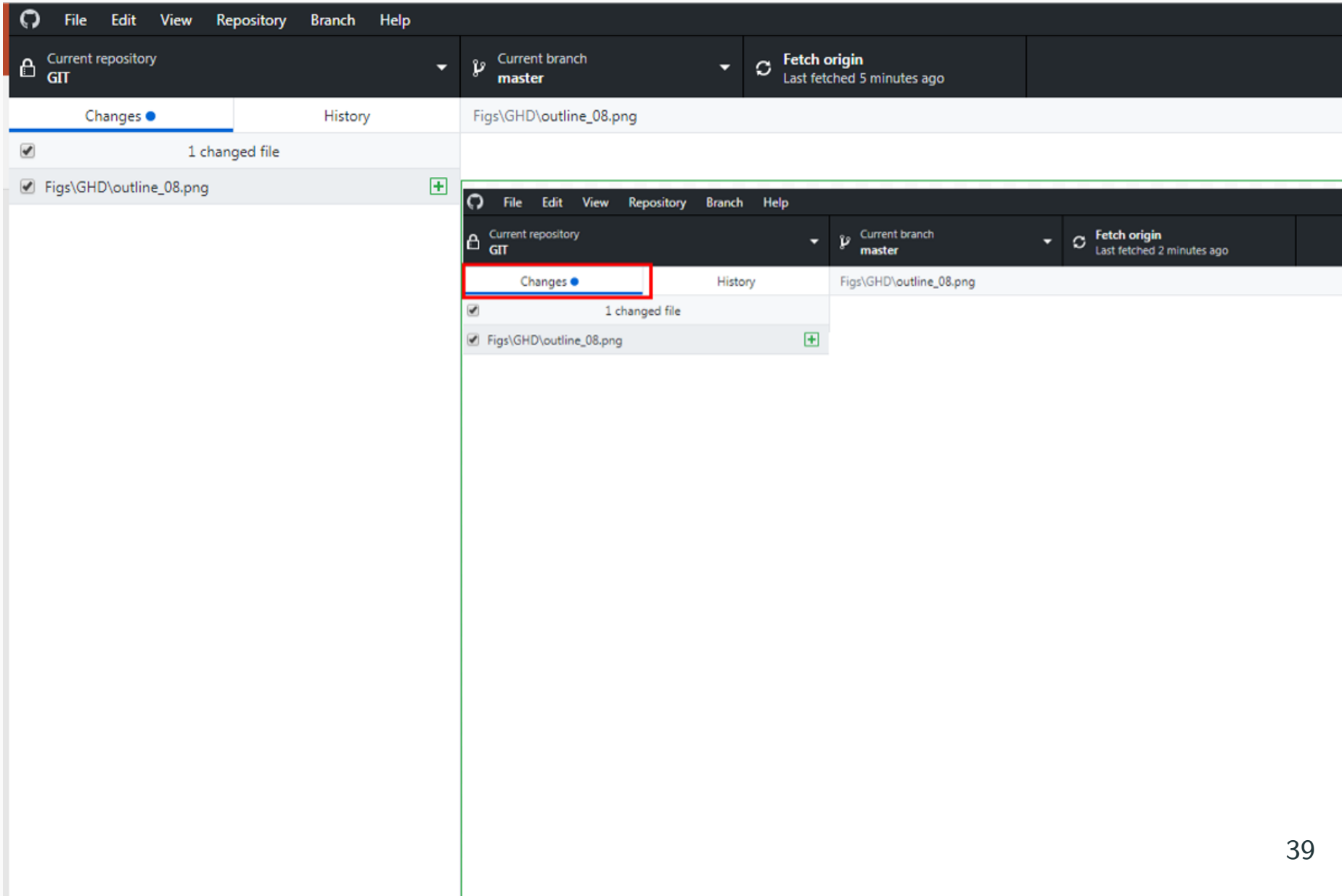
Using the repostry

we can carry on making local commits or or publish the repostry to github



Using the repostry

we can carry on making local commits or or publish the repostry to github




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)

References i

(xkcd)