Git Exercises

Weijie Tang, s2511342@bi.no

1. Clone the repository for the class: https://github.com/BI-DS/GRA-4152

a. Explore the version history by visualizing it as a graph.

b. When was the last time README.md was modified?

c. What was the commit message associated with the last modification to the README.md? (Hint: use git blame and git show).

git clone https://github.com/BI-DS/GRA-4152.git
cd GRA-4152

```
Last login: Sun Sep  7 16:48:42 on ttys000
(base) tang@Mac ~ % git clone https://github.com/BI-DS/GRA-4152.git
Cloning into 'GRA-4152'...
remote: Enumerating objects: 109, done.
remote: Counting objects: 100% (61/61), done.
remote: Compressing objects: 100% (51/51), done.
remote: Total 109 (delta 21), reused 48 (delta 10), pack-reused 48 (from 1)
Receiving objects: 100% (109/109), 25.58 KiB | 2.33 MiB/s, done.
Resolving deltas: 100% (30/30), done.
(base) tang@Mac ~ % cd GRA-4152
(base) tang@Mac GRA-4152 %
```

git log --all --graph –oneline

```
(base) tang@Mac GRA-4152 % git log --all --graph --oneline
* bcf7650 (HEAD -> master, origin/master, origin/HEAD) warpbreaks dataset
* dc730e8 adding colab notebooks
* e5c1c97 adding material lec 10
* ff60de7 adding material lecture 9
* e6ac538 material lecture 7
* b412adb adding material lecture 6
* 7e0089c adding material for lecture 6
* c26010d adding material for lec 5
* 1a4a267 adding material lecture 4
* 9222545 adding material for lecture 3
* 8b4efee adding material for lecture 3
* 71f261f added honor code
* a610fc6 adding 1 asnyc exercise from lecture 1
* 0f6036b adding 1 asnyc exercise from lecture 1
* ae45ae7 removing git_exercise.py file
* 84ed53d adding file for git exercise in lecture 2
* 37ac00f adding some py files for lecture 1
* 0fb7842 adding instructions for UML
* 20b8851 initial commit
```

git log -1 -- README.md

```
(base) tang@Mac GRA-4152 % git log -1 -- README.md
commit 71f261f8dbb09c828dfd2be1ad664a14b1fbc498
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date:   Wed Aug 31 09:59:14 2022 +0200

    added honor code
```

git blame README.md

```
(base) tang@Mac GRA-4152 % git blame README.md
^20b8851 (rogelioandrade 2022-08-19 08:48:04 +0200  1) # GRA-4152
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  2) This repository contains different materi
als used throughout the course, e.g. examples shown in lectures, suggestd solutions for homework
, problems discussed in tutorial sessions, etc. You should follow this repository frequently, as
 different materials will become available as we cover th
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  3)
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  4) ## Packages
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  5) Unified Modeling Language (UML) is a tool
 to visualize the design, or architecture, of (complex) software systems. Just like classes in O
OP. We can generate UML diagrams for `Python` classes using the library `pylint`, which uses `gr
aphviz` to generate `png` or `pdf` files showing the architecture of a given class.
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  6) ```bash
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  7) pip install pylint
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  8) sudo apt install graphviz
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200  9) pyreverse -o png <your code>.py
0fb7842d (rogelioandrade 2022-08-22 08:50:07 +0200 10) ```
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 11)
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 12) ## Honor Code
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 13) You are free to form study groups and may
 discuss homework in groups. However, each student must write down the solutions and code from s
cratch independently and must understand the solution well enough. It is a honor code violation
to copy, refer to, or look at written or code solutions from a previous year or solutions posted
 online (inspired by the Stanford Honor Code).
71f261f8 (rogelioandrade 2022-08-31 09:59:14 +0200 14)
```

git show 71f261f8

```
(base) tang@Mac GRA-4152 % git show 71f261f8
commit 71f261f8dbb09c828dfd2be1ad664a14b1fbc498
Author: rogelioandrade <rogelio.a.mancisidor@bi.no>
Date:   Wed Aug 31 09:59:14 2022 +0200

    added honor code

diff --git a/README.md b/README.md
index a7359ae..a404da3 100644
--- a/README.md
+++ b/README.md
@@ -8,3 +8,7 @@ pip install pylint
 sudo apt install graphviz
 pyreverse -o png <your code>.py
 ```
+
+## Honor Code
+You are free to form study groups and may discuss homework in groups. However, each student mus
t write down the solutions and code from scratch independently and must understand the solution
well enough. It is a honor code violation to copy, refer to, or look at written or code solution
s from a previous year or solutions posted online (inspired by the Stanford Honor Code).
+
```

2. One common mistake when learning Git is to commit large files that should not be managed by Git or adding sensitive information.

a. Add a .gitignore file to your portfolio code repository (https://github.com/SXXXXXXX/GRA4152) and exclude files and/or folders. You might need to create a foo.py file to be excluded.

```
echo "print('This file should not be committed')" > foo.py
echo "foo.py" > .gitignore
git add .gitignore
git commit -m "Add .gitignore to exclude foo.py"
```

```
(base) tang@Mac GRA-4152 % git config --global user.name "S2511342"
(base) tang@Mac GRA-4152 % git config --global user.email "s2511342@bi.no"
(base) tang@Mac GRA-4152 % echo "print('This file should not be committed')" > foo.py
(base) tang@Mac GRA-4152 % echo "foo.py" > .gitignore
(base) tang@Mac GRA-4152 % git add .gitignore
(base) tang@Mac GRA-4152 % git commit -m "Add .gitignore to exclude foo.py"
[master 244725f] Add .gitignore to exclude foo.py
 1 file changed, 1 insertion(+), 14 deletions(-)
```
```

3. Clone some repository from GitHub and modify one of its existing files.

a. What happens when you type git stash?

b. What do you see when running git log --all --oneline?

c. Run git stash pop to undo what you did with git stash. In what scenario might this be useful?

d. List your current stashes and delete them with git stash drop <stash_id>

Now, modify a file and stash changes. Make a new modification to the same file, but this time commit those changes.

a. What happens if you type git stash pop and open the file that you have modified? What do you see in the file?

cd ~
git clone https://github.com/zbirnba1/quantative-finance
cd ~/quantative-finance

```
(base) tang@Mac GRA-4152 % cd ~
(base) tang@Mac ~ % git clone https://github.com/zbirnba1/quantative-finance
Cloning into 'quantative-finance'...
remote: Enumerating objects: 124, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 124 (delta 5), reused 9 (delta 3), pack-reused 109 (from 1)
Receiving objects: 100% (124/124), 89.39 KiB | 1.94 MiB/s, done.
Resolving deltas: 100% (52/52), done.
(base) tang@Mac ~ % cd ~/quantative-finance
(base) tang@Mac quantative-finance %
```

echo "My first change" >> README.md
git stash

```
(base) tang@Mac quantative-finance % echo "My first change" >> README.md
(base) tang@Mac quantative-finance % git stash
Saved working directory and index state WIP on master: 55c20fc updated requirements
```

git log --all –oneline

```
(base) tang@Mac quantative-finance % git log --all --oneline
4089440 (refs/stash) WIP on master: 55c20fc updated requirements
c6af52f index on master: 55c20fc updated requirements
dcf6aba (origin/dependabot/pip/pymongo-4.6.3) Bump pymongo from 3.4.0 to 4.6.3
55c20fc (HEAD -> master, origin/master, origin/HEAD) updated requirements
5b222b6 Updated setup file
05574b0 Updated setup file
fbc8d8e Updated setup file
021e606 Merge remote-tracking branch 'origin/master'
141d759 config file
05a990f Updated setup file
6b32e05 config file
ddca94b config file
8f487fd config file
0dd09ff config file
a5aacd7 config file
```

git stash pop

```
(base) tang@Mac quantative-finance % git stash pop
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (4089440e09b993aca4ed8df87a744c25c936e286)
```

git stash list
echo 'test' >> README.md
git stash
git stash list
git stash drop stash@{0}
git stash list

```
(base) tang@Mac quantative-finance % git stash list
(base) tang@Mac quantative-finance % echo 'test' >> README.md
(base) tang@Mac quantative-finance % git stash
Saved working directory and index state WIP on master: 55c20fc updated requirements
(base) tang@Mac quantative-finance % git stash list
stash@{0}: WIP on master: 55c20fc updated requirements
(base) tang@Mac quantative-finance % git stash drop stash@{0}
Dropped stash@{0} (0d59f03f7098b83912be31058b3eb65548e2c264)
(base) tang@Mac quantative-finance % git stash list
(base) tang@Mac quantative-finance %
```

echo '11111' >> README.md
git stash
echo '22222' >> README.md
git add README.md
git commit -m "22222"
git stash pop

```
(base) tang@Mac quantative-finance % echo '11111' >> README.md
(base) tang@Mac quantative-finance % git stash
Saved working directory and index state WIP on master: 55c20fc updated requirements
(base) tang@Mac quantative-finance % echo '22222' >> README.md
(base) tang@Mac quantative-finance % git add README.md
(base) tang@Mac quantative-finance % git commit -m "22222"
[master 81e6769] 22222
 1 file changed, 1 insertion(+)
(base) tang@Mac quantative-finance % git stash pop
```

cat README.md

```
<<<<<<< Updated upstream
22222
=======
11111
>>>>>>> Stashed changes
```

4. Create a new branch in your class repository (https://github.com/ SXXXXXXX/GRA4152) and call it my_test_branch.

a. Explore both branches, by switching back and forth.

b. Add a comment or a line in any file in the branch my_test_branch. Then add such a change and commit.

c. Finally, merge my_test_branch into master.

cd GRA-4152
git checkout -b my_test_branch

```
(base) tang@Mac ~ % cd GRA-4152
(base) tang@Mac GRA-4152 % git checkout -b my_test_branch
Switched to a new branch 'my_test_branch'
```

echo "# S2511342" >> README.md
git add README.md
git commit -m "Tang's git exercise 4b"

```
(base) tang@Mac GRA-4152 % echo "# S2511342" >> README.md
(base) tang@Mac GRA-4152 % git add README.md
(base) tang@Mac GRA-4152 % git commit -m "Tang's git exercise 4b"
[my_test_branch 4ebda10] Tang's git exercise 4b
 1 file changed, 1 insertion(+)
```

git checkout master
git merge my_test_branch

```
(base) tang@Mac GRA-4152 % git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
(base) tang@Mac GRA-4152 % git merge my_test_branch
Updating 244725f..4ebda10
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
```