TP 4 : GIT

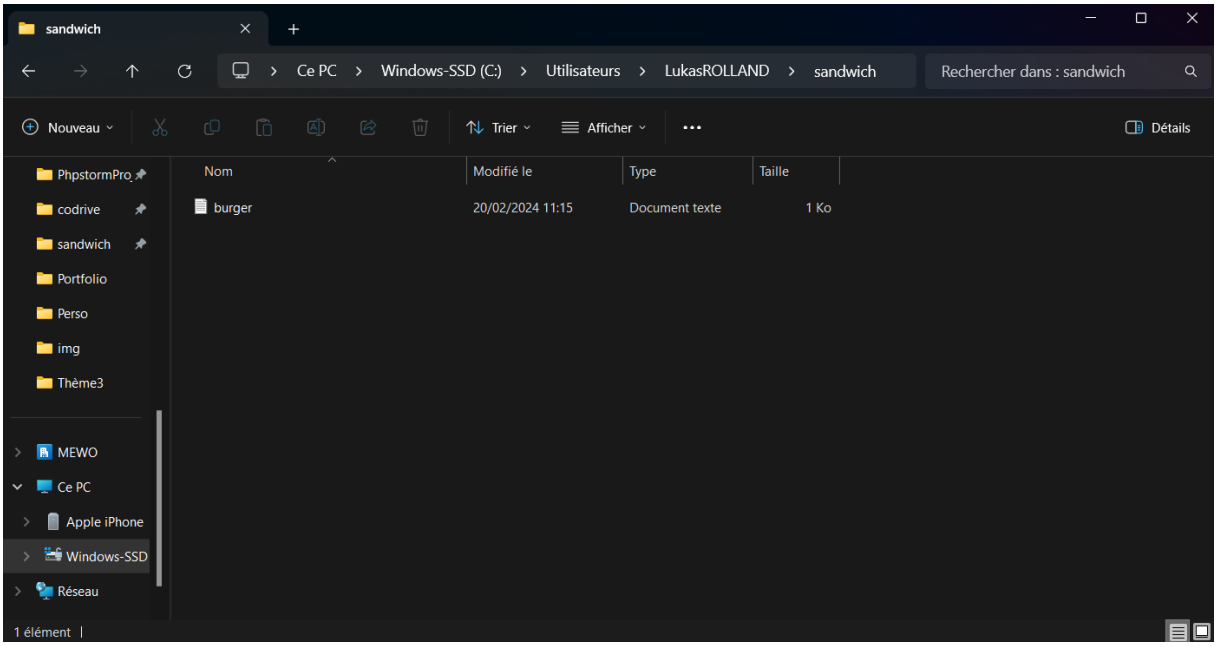
Question 2.1. *Initialiser un nouveau dépôt* Git *dans un répertoire* sandwich*, et créez le fichier* burger.txt *qui contient la liste des ingrédients d'un burger, un ingrédient par ligne.*

J'ai créé un nouveau dépôt Git dans un répertoire appelé sandwich grâce à la commande :

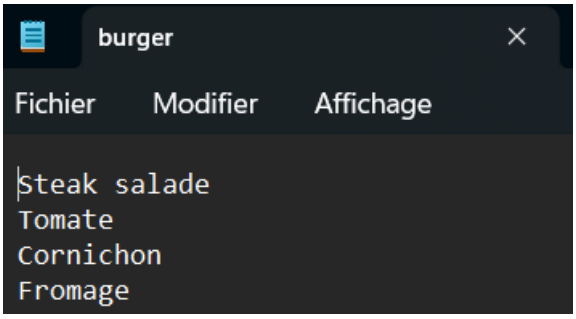
Git init sandwich

```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~
$ git init sandwich
Initialized empty Git repository in C:/Users/LukasROLLAND/sandwich/.git/
```

Puis j'ai créé un fichier.txt qui s'appelle burger :

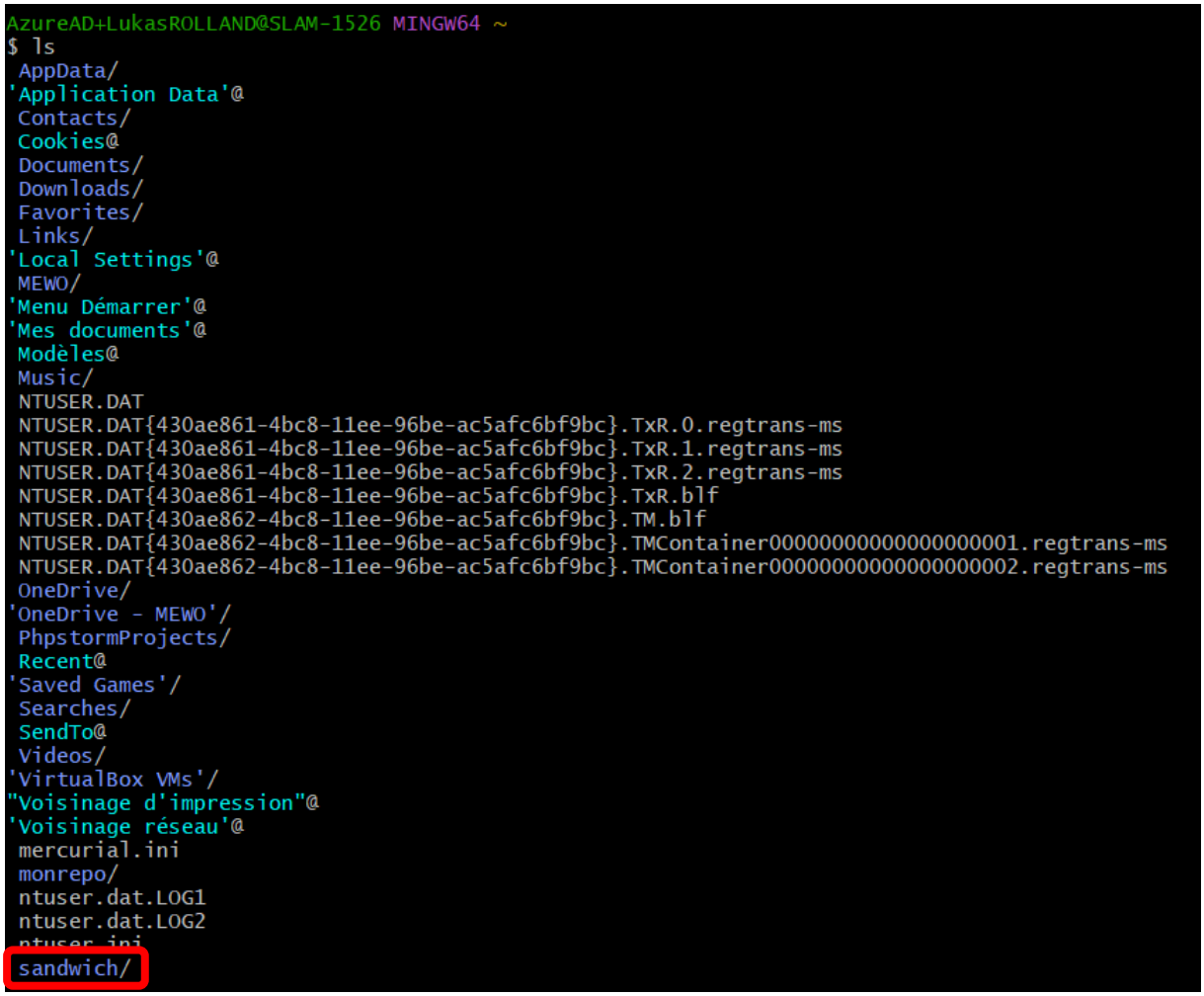


Et qui contient une liste d'ingrédients



Question 2.2. *Vérifiez avec* git status *l'état dans lequel se trouve votre dépôt. Vos modifications (l'ajout du fichier* burger.txt*) devraient être présentes seulement dans la copie de travail.*

J'ai d'abord utilisé la commande ls pour voir les fichiers présents dans le répertoire ou je suis. J'ai bien mon répertoire sandwich



Ensuite, j'ai utilisé la commande cd sandwich/ pour me rendre dans ce répertoire, puis un ls ou j'ai pus confirmer que mon fichier burger.txt était bien dedans. J'ai ensuite pus faire mon git add burger.txt pour que mon fichier soit présent dans l'index. Grâce au git status, on peut voir qu'il y a un nouveau document texte dans l'index qui s'appelle burger.

```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~
$ cd sandwich/

AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ ls
burger.txt

AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git add burger.txt

AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   burger.txt
```

Question 2.3. *Préparez* burger.txt *pour le commit avec* git add burger.txt*. Utilisez* git status *à nouveau pour vérifier que les modifications ont bien été placées dans l'index. Puis, utilisez* git diff --cached *pour observer les différences entre l'index est la dernière version présente dans l'historique de révision (qui est vide).*

Une fois le fichier dans l'index, j'ai utilisé la commande git diff –cached, qui me permettre de voir les différences entre l'index et les dernières versions présentes. Nous pouvons donc voir les lignes qui ont été rajouté par rapport a la version d'avant grâce au +.

```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git diff --cached
diff --git a/burger.txt b/burger.txt
new file mode 100644
index 0000000..39c0ee7
--- /dev/null
+++ b/burger.txt
@@ -0,0 +1,4 @@
+Steak salade
+Tomate
+Cornichon
+Fromage
```

Question 2.4. *Commitez votre modification avec* git commit -m "<votre_message_de_commit>"*. Le message entre guillemets doubles décrira la nature de votre modification (généralement ≤*65 *characteurs).*

J'ai utilisé la commande git commit -m « Ajout des ingredients pour la recette du sandwich » pour mettre à jour le répertoire, de plus, le message permet de savoir plus tard ce qu'on a changé.

```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git commit -m "Ajout des ingredients pour la recette du sandwich"
[master (root-commit) a9059de] Ajout des ingredients pour la recette du sandwich
 1 file changed, 4 insertions(+)
 create mode 100644 burger.txt
```

Question 2.5. *Exécutez à nouveau* git status*, pour vérifier que vos modifications ont bien été commités.*

Nous pouvons remarquer grâce au git status qu'il n'y a aucun changement.

```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Question 2.6. *Essayez à présent la commande* git log *pour afficher la liste des changements effectués dans ce dépôt; combien y en a-t-il? Quel est le numéro (un hash cryptographique en format SHA1) du dernier commit effectuer ?*

J'ai utilisé la commande git log, qui me permet de voir les derniers commit effectuer, nous pouvons donc voir le dernier que nous avons fait mais qui est aussi le seul. Nous pouvons aussi voir le numéro du dernier commit effectuer(encadré).

```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git log
commit a9059de7609f1c15023d445d582714a15330fc91 (HEAD -> master)
Author: LukasRolls <lukasrolland57@gmail.com>
Date:   Tue Feb 20 12:06:49 2024 +0100

    Ajout des ingredients pour la recette du sandwich
```

Question 2.7. *Créez quelques autres sandwiches* hot_dog.txt, jambon_beurre.txt*...et/ou modifiez les compositions de sandwiches déjà créés, en commitant chaque modification* séparément*. Chaque commit doit contenir une et une seule création ou modification de fichier. Effectuez au moins 5 modifications différentes (et donc 5 commits différents). À chaque étape essayez les commandes suivantes :*

— git diff avant git add *pour observer ce que vous allez ajouter à l'index;*

— git diff --cached après git add *pour observer ce que vous allez committer.*

Note : la commande git commit <file> a le même effet que git add <file> suivie de git commit.

J'ai créé un fichier qui s'appelle raclette

```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ echo patate fromage > raclette
```

En suite j'ai fait un git add puis un git commit. Puis j'ai répété l'opération plusieurs fois :

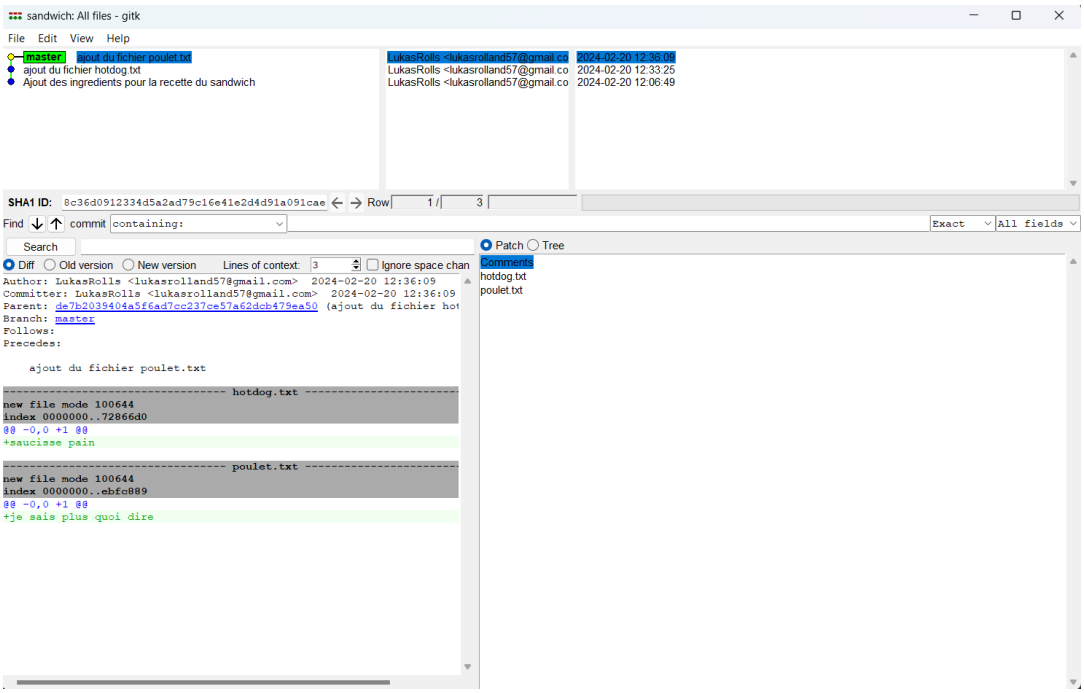
```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git log
commit 8c36d0912334d5a2ad79c16e41e2d4d91a091cae (HEAD -> master)
Author: LukasRolls <lukasrolland57@gmail.com>
Date:   Tue Feb 20 12:36:09 2024 +0100

    ajout du fichier poulet.txt

commit de7b2039404a5f6ad7cc237ce57a62dcb479ea50
Author: LukasRolls <lukasrolland57@gmail.com>
Date:   Tue Feb 20 12:33:25 2024 +0100

    ajout du fichier hotdog.txt

commit a9059de7609f1c15023d445d582714a15330fc91
Author: LukasRolls <lukasrolland57@gmail.com>
Date:   Tue Feb 20 12:06:49 2024 +0100

    Ajout des ingredients pour la recette du sandwich
```

Question 2.8. *Regardez à nouveau l'historique des modifications avec* git log *et vérifiez avec* git status *que vous avez tout commité.* Git *offre plusieurs interfaces, graphiques ou non, pour afficher l'historique. Essayez les commandes suivantes (*gitg *et* gitk *ne sont pas forcément installés) :*

— git log
— git log --graph --pretty=short
— gitg
— gitk

J'ai utilisé les différentes commandes pour voir les différents logs disponibles.

```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git log --graph --pretty=short
* commit 8c36d0912334d5a2ad79c16e41e2d4d91a091cae (HEAD -> master)
| Author: LukasRolls <lukasrolland57@gmail.com>
|
|     ajout du fichier poulet.txt
|
* commit de7b2039404a5f6ad7cc237ce57a62dcb479ea50
| Author: LukasRolls <lukasrolland57@gmail.com>
|
|     ajout du fichier hotdog.txt
|
* commit a9059de7609f1c15023d445d582714a15330fc91
  Author: LukasRolls <lukasrolland57@gmail.com>

      Ajout des ingredients pour la recette du sandwich

AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ gitg
bash: gitg: command not found

AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ gitk
```



Question 2.9. *Vous voulez changer d'avis entre les différents états de la Figure 1? Faites une modification d'un ou plusieurs sandwiches, ajoutez-la à l'index avec* git add *(vérifiez cet ajout avec* git status*), mais ne la commitez pas. Exécutez* git reset *sur le nom de fichier (ou les noms de fichiers) que vous avez préparés pour le commit; vérifiez avec* git status *le résultat.*

J'ai modifié mon fichier burger.txt en lui enlevant des lignes, ensuite, j'ai fait un git add pour ajouter le nouveau fichier. On peut voir que le fichier a été modifié grâce au git status. Puis j'ai fait un git reset burger.txt pour annuler le git add. Nous pouvons constater grâce au git status que le fichier n'est plus dans l'index.

```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git add burger.txt

AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   burger.txt


AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git reset burger.txt
Unstaged changes after reset:
M       burger.txt

AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git status
On branch 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:   burger.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Question 2.10. *Votre modification a été « retirée » de l'index. Vous pouvez maintenant la jeter à la poubelle avec la commande* git checkout *sur le ou les noms des fichiers modifiés, qui récupère dans l'historique leurs versions correspondant au tout dernier commit. Essayez cette commande, et vérifiez avec* git status *qu'il n'y a maintenant plus aucune modification à commiter.*

J'ai fait un git checkout pour récupérer mon ancien fichier que j'ai commit. J'ai fait ensuite un git status pour vérifier que j'ai la dernière version de mon fichier burger.txt

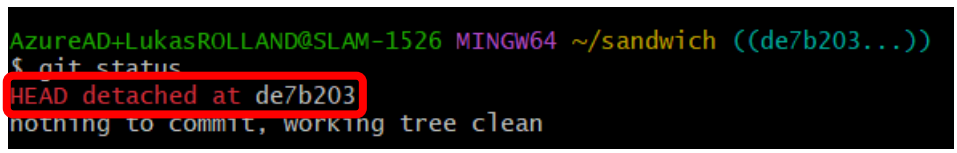
```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git checkout burger.txt
Updated 1 path from the index

AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Question 2.11. *Regardez l'historique de votre dépôt avec* git log *; choisissez dans la liste un commit (autre que le dernier). Exécutez* git checkout COMMITID *où* COMMITID *est le numéro de commit que vous avez choisi. Vérifiez que l'état de vos sandwiches est maintenant revenu en arrière, au moment du commit choisi. Que dit maintenant* git status*?*

J'ai tout d'abord fait un git log pour savoir l'ID d'un de mes derniers COMMIT. En suite j'ai pus utiliser la commande git checkout suivis de l'ID du COMMIT choisis.

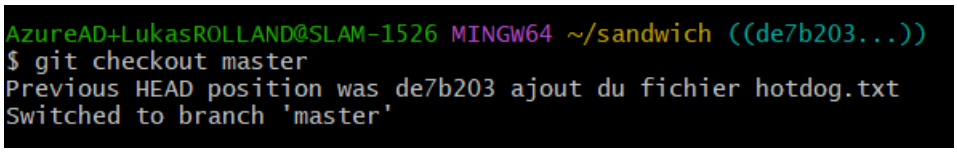
```
AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git log
commit 8c36d0912334d5a2ad79c16e41e2d4d91a091cae (HEAD -> master)
Author: LukasRolls <lukasrolland57@gmail.com>
Date:   Tue Feb 20 12:36:09 2024 +0100

    ajout du fichier poulet.txt

commit de7b2039404a5f6ad7cc237ce57a62dcb479ea50
Author: LukasRolls <lukasrolland57@gmail.com>
Date:   Tue Feb 20 12:33:25 2024 +0100

    ajout du fichier hotdog.txt

commit a9059de7609f1c15023d445d582714a15330fc91
Author: LukasRolls <lukasrolland57@gmail.com>
Date:   Tue Feb 20 12:06:49 2024 +0100

    Ajout des ingredients pour la recette du sandwich

AzureAD+LukasROLLAND@SLAM-1526 MINGW64 ~/sandwich (master)
$ git checkout de7b2039404a5f6ad7cc237ce57a62dcb479ea50
Note: switching to 'de7b2039404a5f6ad7cc237ce57a62dcb479ea50'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at de7b203 ajout du fichier hotdog.txt
```

Pour finir, j'ai utilisé la commande git status, nous pouvons voir que l'ID a changer et est bien celui demandé plutôt.



Question 2.12. *Vous pouvez retourner à la version plus récente de votre dépôt avec* git checkout master*. Vérifiez que cela est bien le cas. Que dit maintenant* git status *?*

J'ai utilisé la commande git checkout master, celle-ci ma permit de revenir à la version la plus récente de mes dépôts et ma rappeler le nom que j'avais mis.



J'ai en suite fait un git status pour confirmer que je sois bien dans la dernière version.