

# Prototypage ST2PTT Gestion de version

M2 2021-2022

remi.griot@efrei.fr



# Gestion de version

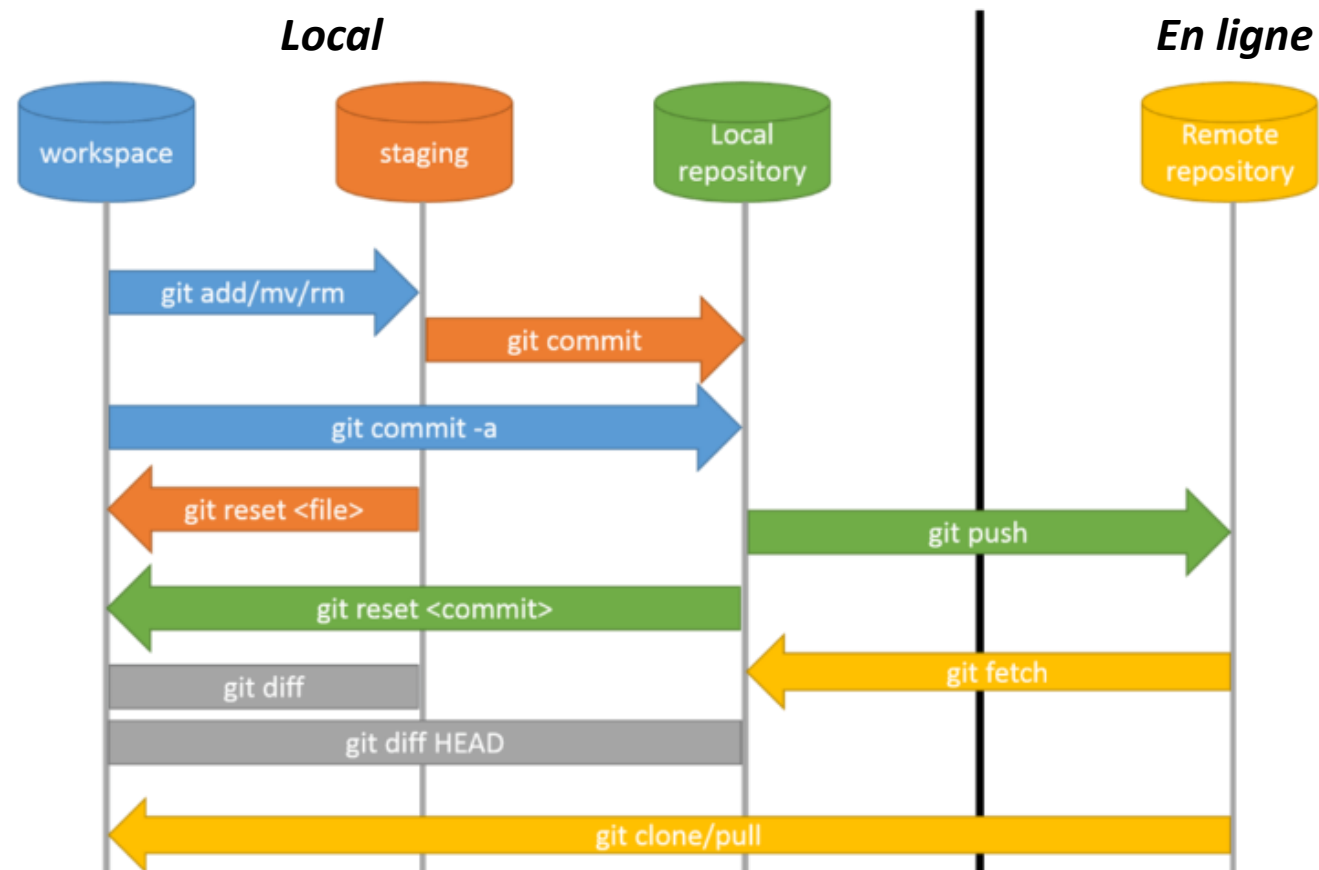
## *Intérêt de la gestion de version*

- Travail collaboratif
- Travail individuel sur plusieurs machines
- Sauvegarde de versions



# Gestion de version

Echange entre le dossier de travail (workspace)/l'index (staging)/le dépôt local/le dépôt en ligne

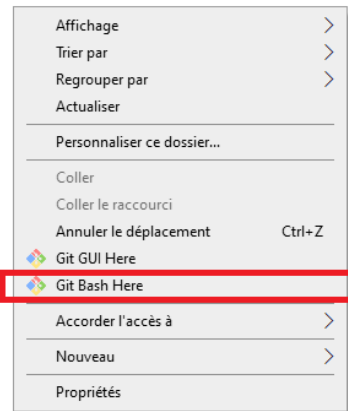


# Gestion de version

## Installation et utilisation

- Création d'un compte sur GitHub
  - GitHub: Hébergement de dépôt Git
- Installation de Git
  - Git: logiciel de gestion de version (gestionnaire de version)
- Utilisation en ligne de commande

01_doc	24/01/2022 13:34	Di
02_src	10/01/2022 16:18	Di
03_md1	20/01/2022 10:07	Di
04_bin	10/01/2022 16:18	Di
.gitignore	20/01/2022 10:07	Di
LICENSE	10/01/2022 16:14	Fi
README	17/01/2022 15:14	Fi



A screenshot of a MINGW32 terminal window. The title bar reads 'MINGW32: c:/git/2022\_ST2PTT/2022\_ST2PTT'. The prompt is 'Remi@DESKTOP-HQH070V MINGW32 /c/git/2022\_ST2PTT/2022\_ST2PTT (main)'. The command entered is '\$ git commit -m "First commit"'. The terminal output is currently empty.



# Gestion de version

## *Configuration*

# Clone d'un dépôt (en ligne -> local)

```
git clone https://github.com/InnovationLab-EFREIParis/Stage_1DOF_DroneBench.git
```

# Initialisation du dépôt (création des fichiers nécessaires à git dans le dossier .git)

```
git init
```

Cette commande crée une suite de fichier dans le dossier caché .git

Notamment le fichier `.gitignore` permettant de lister les fichiers ignorés par le gestionnaire de version



# Gestion de version

## Commande agissant en Local

# ajout du fichier monfichier.txt à l'index

`git add monfichier.txt`

# ajout de tous les fichiers encore non présents à l'index

`git add .`

# retire le fichier monfichier.txt de l'index

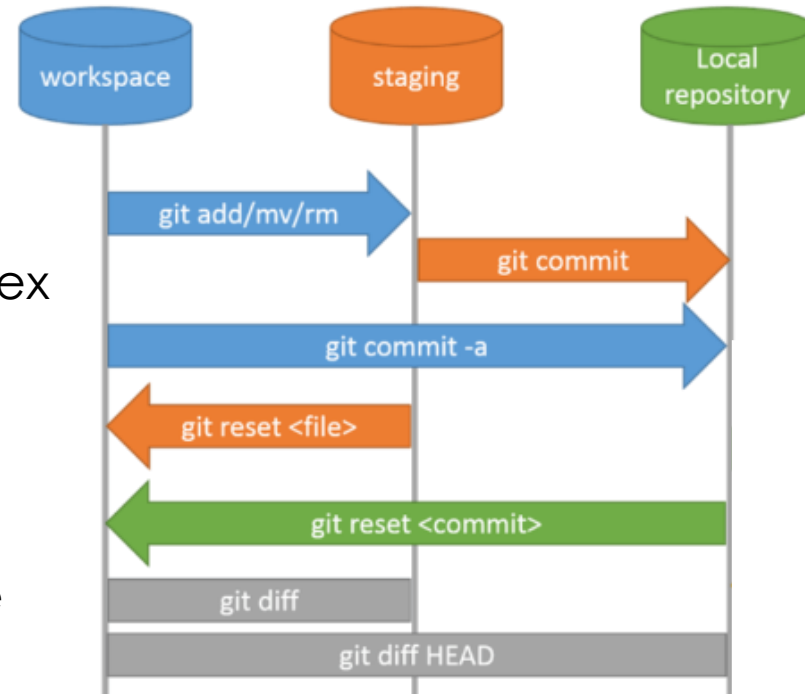
`git rm`

# Sauvegarde en local de ma version, accompagné d'un message "Version 1"

`git commit -m "Version 1"`

# Statut de la cohérence entre l'index, le dépôt local, et le dossier de travail

`git status`



# Gestion de version

## *Local vers distant*

# Transfert les commit du dépôt local vers le dépôt maître (en ligne)

git push

## *Distant vers local*

# télécharge en local le dépôt distant

git fetch

# liste les différences entre origin (local) et main(distant)

git diff origin/main

# du dépôt local vers l'espace de travail

git merge

# met à jour l'espace de travail

git pull # (=git fetch; git merge)

