

PDF Explication commande GIT :

Liaazid Gracia Mikail
Silue Emmanuel

Pour commencer, nous avons créé notre repo sur github. Ensuite nous voulons cloner ce même repo sur nos pc, pour ce faire, nous utilisons la commande « **git clone (clé https/ssh)** ».
Quand cela a été fait, nous avons mis en place la structure du projet.

— src/	# Code source du jeu
— tests/	# Dossier pour les tests
— docs/	# Dossier pour la documentation
— README.md	# Description du projet

Pour la convention de nos nommage de branches, nous utiliserons le « **Gitflow** ».
Pour la convention des messages de commit, nous utiliserons « **Conventional commit** ».
Pour le « **Workflow** », nous utiliserons les « **Feature branch** ».

Pour continuer, nous avons réalisé les premières fonctions d'initialisation du jeu, et pour être efficace, nous avons décidé de travailler sur des branches différentes. Pour ce faire nous avons tapé la commande suivante : « **git branche ma-branche** ».
Et nous avons basculé dessus : « **git checkout -b ma-branche** »

```
8158 git checkout -b chore/initialisation
```

Ensuite quand le travail a été fait, nous voulons repasser sur la branche main pour pouvoir push.
Pour ce faire nous avons tapé les commandes suivantes :

« **git add.** »

« **git commit -m [message]** »

« **git switch main** »

La commande switch main permet de se balader de branche en branche.

```
→ projet-puissance4 git:(chore/initialisation) X git add .
→ projet-puissance4 git:(chore/initialisation) X git commit -m "WIP : main and unc init"
[chore/initialisation 77654e8] WIP : main and unc init
2 files changed, 44 insertions(+), 1 deletion(-)
→ projet-puissance4 git:(chore/initialisation) git switch main
Basculement sur la branche 'main'
Votre branche est à jour avec 'origin/main'.
```

Après cela, nous voulons mettre en place notre jeu, pour ce faire nous allons initialiser une nouvelle branche que nous appelons « jeux ».

```
6be7d8a..f521b58 main -> main
● → projet-puissance4 git:(main) git checkout -b jeux
Basculement sur la nouvelle branche 'jeux'
○ → projet-puissance4 git:(jeux) X
```

Ensuite, quand notre fonction a été mise en place, nous voulons envoyer la branche sur le dépôt distant. Pour ce faire nous avons fait comme ça :

```
create mode 100644 src/jeux.go
→ projet-puissance4 git:(jeux) git push origin jeux
Énumération des objets: 10, fait.
Décompte des objets: 100% (10/10), fait.
Compression par delta en utilisant jusqu'à 18 fils d'exécution
Compression des objets: 100% (5/5), fait.
Écriture des objets: 100% (6/6), 1022 octets | 1022.00 Kio/s, fait.
Total 6 (delta 0), réutilisés 0 (delta 0), réutilisés du paquet 0 (depuis 0)
remote:
remote: Create a pull request for 'jeux' on GitHub by visiting:
remote:   https://github.com/MikailLG/projet-puissance4/pull/new/jeux
remote:
To github.com:MikailLG/projet-puissance4.git
 * [new branch]   jeux -> jeux
```

Ce projet nous a permis de mettre en œuvre toutes les commandes gits ainsi que les normes à adaptée. En effet la normes des commits nous a fait prendre conscience qu'il était important d'avoir de bon nom de commit ainsi qu'une norme pour que le projet sois mené à bien. Les branches nous ont permis de pouvoir travaille chacun de notre coté sans avoir de problème au niveau de l'ajout de fonctionnalité. Nous avons trouvé ça très utile pour nos futures projets