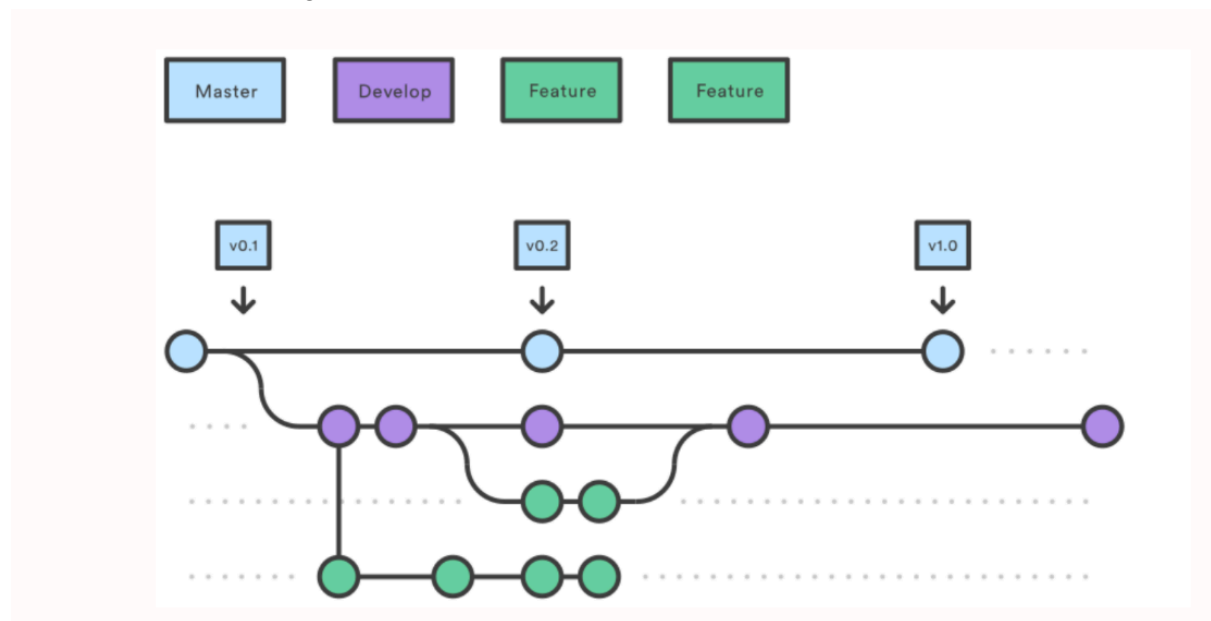


TP Puissance 4

HUART Geoffroy et BERARD Samuel

Convention de nommage des branches



Conventions de nommages : Conventionnels commits

WorkFlow : Feature Branch Workflow

On a choisii ces conventions car nous avons un projet de petite taille et nous utilisons les outils de base

Listes des Branches utilisées :

- Main : Branche comportant le code principale et fonctionnel
- Develop : Branche de raliement de toutes les autre branches de développement comportant le code en phase de tests
- Feat/etat : Branche de développement des fonctions suivant l'état du jeu à chaque tour
- Fix/grille : Branche de réparation du code de la grille
- Feat/Tourjoueur1 : Branche de développement des fonction mettant en place les tours des joueurs
- Feat/codesgrilles : Branche de développement d'une nouvelle grille
- Feat/grilles : Branche de développement de la grille
- Feat/variables : Branche de développement des variables
- Feat/player : Branche de développement des joueurs (définition des joueurs, leur symboles et leur noms)

```

PS C:\Users\ghuar\Documents\Ynov 2526\TPpuissance4> git push origin 'Feat/grille'
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 503 bytes | 167.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'Feat/grille' on GitHub by visiting:
remote:   https://github.com/ghuart9207-cmd/TPPUISSANCE4/pull/new/Feat/grille
remote:
To https://github.com/ghuart9207-cmd/TPPUISSANCE4.git
 * [new branch]      Feat/grille -> Feat/grille

```

Git push origin <branche>: Sert à déposer un commit sur le dépôt distant pour ensuite pouvoir le récupérer de n'importe quelle machine.

Nous avons exécuté cette commande afin d'envoyer nos modifications locales sur le dépôt distant et les récupérer sur nos ordinateurs personnels.

```

PS C:\Users\ghuar\Documents\Ynov 2526\TPpuissance4> git pull origin 'develop'
From https://github.com/ghuart9207-cmd/TPPUISSANCE4
 * branch      develop      -> FETCH_HEAD
Updating 5b83ac8..3cd288c
Fast-forward
 go.mod       | 3 +++
 go.sum       | 0
 main.go      | 17 ++++++
 player/player.go | 13 ++++++
 4 files changed, 33 insertions(+)
 create mode 100644 go.mod
 create mode 100644 go.sum
 create mode 100644 main.go
 create mode 100644 player/player.go

```

Git pull origin <branche>: Sert à récupérer un commit sur le dépôt distant pour ensuite pouvoir le réutiliser et travailler dessus depuis une autre machine.

Nous avons exécuté cette commande afin de récupérer le code présent sur le dépôt en local pour y apporter des modifications.

```

PS C:\Users\ghuar\Documents\Ynov 2526\TPpuissance4> git switch 'develop'
Switched to branch 'develop'
Your branch is behind 'origin/develop' by 3 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

```

Git switch <branche> : Sert à changer de branche pour travailler depuis celle-ci et non une autre.

Nous avons exécuté cette commande afin de changer de branche ce qui nous permet de modifier et ajouter du code sans impacter la branche principale qui elle, est fonctionnelle.

```
samba@LAPTOP-SAM MINGW64 ~/Desktop/TPPUISSANCE4 (develop)
$ git switch -c 'feat/variables'
Switched to a new branch 'feat/variables'
```

Git switch -c <branche> : Sert à créer une nouvelle branche et à changer immédiatement de branche pour travailler sur celle-ci.

Nous avons exécuté cette commande afin de créer une nouvelle branche sur laquelle on pourra travailler sur une nouvelle fonctionnalité en particulier pour la fusionner avec develop une fois terminé.

```
samba@LAPTOP-SAM MINGW64 ~/Desktop/TPPUISSANCE4 (feat/tour-j2)
$ git stash -m "modifs tour j2"
Saved working directory and index state On feat/tour-j2: modifs tour j2
```

Git stash -m "mess" : Sert à mettre de côté les modification en staging area afin de pouvoir se déplacer sur une autre branche sans que les modification locales de la branche où l'on provient s'y applique.

Nous avons exécuté cette commande afin de pouvoir se déplacer sur la branche de l'autre pour l'aider sans impacter avec nos modifications locales de la branche où l'on était à l'origine.

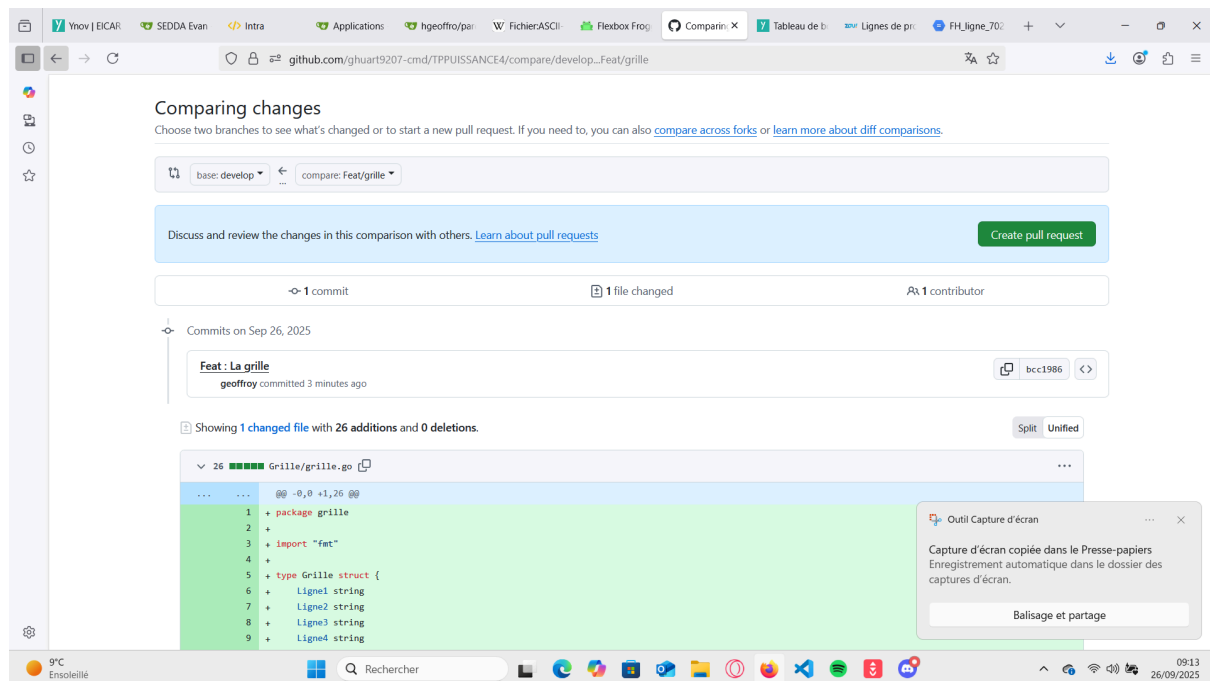
```
samba@LAPTOP-SAM MINGW64 ~/Desktop/TPPUISSANCE4 (develop)
$ git stash clear
```

Git stash clear : Sert à supprimer les stash précédemment faits avec git stash -m "mess" Nous avons exécuté cette commande afin de nettoyer le stock de stash qui ne sont plus utiles histoire de garder quelque chose de propre et organisé car cela évite aussi toute fausse manipulation handicapante.

```
samba@LAPTOP-SAM MINGW64 ~/Desktop/TPPUISSANCE4 (develop)
$ git branch -D feat/tour-j2
Deleted branch feat/tour-j2 (was adf4055).
```

Git branch -d <branche> : Sert à supprimer une branche qui ne sers plus ou qui à été totalement ratée.

Nous avons exécuté cette commande afin de supprimer une branche qui n'avait plus lieu d'être puisqu'en effet nous avons essayé de nous départager une tâche en deux ce qui n'a pas été possible finalement donc nous avons re fusionner la tâche et supprimé la branche inutile.



Nous avons fait un pull request via l'interface de GitHub puisque c'est l'hébergeur de notre dépôt distant cela nous a permis de fusionner les branches après avoir travaillé dessus par exemple nous avons dernièrement fusionné la branche develop et main afin d'avoir la version final de notre code sur la branche main.

Grâce à ce TP nous avons développé de nouveau notre maîtrise de git, ses commandes et ces fonctionnalités pour le travail de groupe. Voulant nous répartir les tâches au mieux nous avons tenté de diviser certaines tâches nous handicapant finalement plus qu'autre chose nous avons donc perdu du temps sur cela puisqu'il a fallu faire machine arrière sur pas mal de changement.