git-collaboratif-depots-distincts

Alice et Bob commencent à collaborer avec Git

paramétrages en vue de collaborer

Le but de Git est de collaborer. Faisons donc travailler ensemble Alice et Bob (et Gustave si besoin) dans un projet

Dépôt initial et invitations

Alice a créé un dépôt initial sur Gitlab qui s'appelle monprojet. Elle va inviter Bob sur son projet en tant que reporter (droit de lecture).

Bob fork le projet d'Alice

Bob va faire un fork du dépôt d'Alice sur Gitlab, puis cloner son propre fork pour obtenir une copie locale :

```
git clone https://gitlab.com/bob/monprojet.git
```

remotes

- Bob possède déjà un remote : Le sien, origin. Il peut le vérifier avec la commande `git remote -v`
- il ajoute celui d'Alice :

```
bob> git remote add alice https://gitlab.com/alice/monprojet.git
```

- puis vérifie ses 2 remotes en tapant `git remote -v`

Bob peut maintenant collaborer avec Alice.

Réalisation d'une fonctionnalité par Alice

- Alice prend une chose à réaliser (item de backlong) et implémente le code nécessaire
- Alice fait les tests et vérifie que ça marche
- `git commit -am \"Message de commit\"`

Alice pousse son main sur son remote

```
git push -u origin main
```

Bob collabore avec Alice

Bob fait d'abord comme Alice pour paramétrer son git et son dépôt local. Bob et Alice vont travailler tout d'abord sur leur branche main. Cela peut sembler plus simple mais en fait on va passer rapidement à un travail dans des branches spécifiques à chaque tâche et utiliser des Merge Request (MR ou Pull Requests, PR sur Github) :

Git avec les branches

car c'est plus clair et finalement pas plus compliqué ! mais n'allons pas trop vite ...

Bob configure ses remotes

Bob doit avoir 2 remotes :

- Le sien, origin qu'il crée au besoin en faisant:

```
git remote add origin https://gitlab.com/bob/monprojet.git
```

- celui d'Alice qu'il ajoute :

```
git remote add alice https://gitlab.com/alice/monprojet.git
```

- Bob tape `git remote -v` pour vérifier ses remotes
- Si il se trompe:

```
git remote remove alice
```

Bob récupère le main d'Alice

```
git fetch Alice main
```

Bob consulte la branche locale correspondant au main d'Alice

```
git branch -av
git checkout Alice/main
```

puis vérifie que le code d'Alice est correct

Bob revient dans son main

```
git checkout main
```

Bob merge le travail d'Alice et pousse les modifs dans son dépôt distant

```
git merge Alice/main
git push
```

Puis detruit si besoin la branche locale d'Alice :

```
git branch -d Alice/main
```

Bob invite Alice

Bob invite à son tour Alice sur son dépôt Gitlab en tant que reporter

Alice se met à jour

- Alice ajoute le remote de Bob :

```
git remote add bob https://gitlab.com/bob/monprojet.git
```

- Alice fetche le main de Bob pour se mettre à jour :

```
git fetch Bob main
```

- puis fusionne le code concerné (après l'avoir éventuellement contrôlé)

```
git merge Bob/main
```

2022 Gérard Rozsavolgyi roza [@] univ-orleans.fr