UNIVERSITE PAUL SABATIER

TOULOUSE III

M1 EEA

SME

TP Réalisation Systèmes - Git

Initiation à GIT

1 Prise en main Git

1.1 Configuration de votre git

Une des premières étapes à effectuer lors de l'utilisation de git sur une nouvelle machine est sa configuration. Cette étape va permettre de vous identifier durant les avancées de vos projet (savoir qui a commit, qui a push, etc ...).

À l'aide de votre terminal créez un dossier qui fera office de dépôt de travail. Placez vous dans ce dossier et initialisez ce dossier comme un dépôt Git.

Veuillez configurer votre git avant de commencer le cahier de TP sur votre terminal.

1.2 Premier commit

Dans ce dépôt, créez un fichier *main.cpp* et écrivez un simple programme en C++ qui permet d'afficher sur le terminal le message "Hello, World !". Compilez le et exécutez-le.

Avec la commande git appropriée ajoutez uniquement le fichier que vous venez de créer pour qu'il soit prêt à être soumis à la modification, commit. Vérifiez que vous l'avez ajouté avec la commande : **git status**.

Finissez par un commit de votre travail en mettant comme description du commit "Mon premier commit". Vérifiez que vous avez bien commit grâce à la commande : **git log**.

Les commits sont des étapes importantes que ça soit pour du travail individuel ou collaboratif, ne négligez pas le nom que vous donnerez à vos commits car ils permettront de localiser l'avancement de vos projets pour vous et vos collègues.

1.3 Gérer son dépôt

Si vous vérifiez l'état de votre dépôt, vous remarquerez qu'il y a des fichiers binaires non suivis (untracked) ou modifiés. L'outil Git permet d'indiquer quels sont les fichiers qu'on veut suivre et ceux qu'on ne veut pas. Cela permet d'avoir une meilleure traçabilité des fichiers modifiés.

Pour cela configurez votre dépôt en ajoutant votre ".gitignore" pour qu'il ignore tous les fichiers à part ceux importants au projet. Dans ce cas la indiquez à git d'ignorer les fichiers binaires produits par le compilateur ***gcc*** doivent être ignorés.

Ajoutez votre configuration et effectuez un commit.

2 Démineur !

Durant cette seconde partie du TP vous allez devoir mettre en pratique les différentes commandes principales vues durant le cours sur Git. Vous allez devoir récupérer un dépôt distant sur Github pour venir travailler sur le projet en binôme ou groupe avant de le remettre sur un autre dépôt distant de compte rendu de votre travail.

L'objectif sera la réalisation d'un jeu sur votre terminal, le démineur en C++, en manipulant simultanément l'outil Git.

2.1 Créez votre compte Github

Veuillez créer votre compte de travail sur le site Github !

Inscrivez-vous grâce au lien suivant : **https://github.com/join?source=login**

2.2 Clonez le dépôt distant du projet

Grâce à votre terminal placez vous sur le bureau de votre session. Ensuite effectuez un : **git clone** du projet situé à cette url : **https://github.com/Cours-Master-SME/Minesweeper_template.git**.

2.3 Un dépôt pour deux

Pour réaliser la suite du TP veuillez créer un seul dépôt distant sur votre compte Github parmi votre binôme en le nomant : **Repo_Nom1_Nom2**.

2.4 Ajoutez votre binôme à votre dépôt

Maintenant que votre dépôt distant est crée sur Github veuillez inviter votre binôme de travail à se joindre à votre projet de TP. Pour cela dirigez vous dans l'onglet "Settings", ensuite allez dans "Manage access" et ajoutez votre binôme avec le bouton "Invite teams or people".

2.5 Ajouter l'URL de votre dépôt distant

Dans cette partie on vous demandera de changer l'URL du dépôt du TP que vous avez clonnez pour lui attribuer votre nouveau dépôt crée sur votre compte Github. Pour cela utilisez les commandes : **git remote**.

```
git remote add origin https://github.com/YOUR_ACCOUNT/YOUR_REPO.git
git branch -M master
git push -u origin master
```

Maintenant compiler votre projet avec la commande ***"cmake ."***. Vous pouvez maintenant effectuez ***make***. Si tout s'est bien déroulé l'éxecutable du projet se trouvera dans le dossier ***bin***.

2.6 Choisissez votre branche

Il est temps de commencer à travailler sur le projet et de faire votre développement ! Répartissez vous maintenant le travail grâce à la création de deux branches.

Le binôme numéro 1 devra créer, sur son dépôt local, la branche **"Dev_Cells"** et le binôme numéro 2 devra créer à son tour la branche **"Dev_Game"**. Tapez dans votre terminal la commande git appropriée pour voir si votre branche de développement a bien été créée à côté de votre branche master.

2.7 Fonctionnalité du jeu

Fonctionnalité de la cellule

L'objet ***cellule*** correspondra à une case du démineur. La cellule a plusieurs états. Elle peut être découverte, contenir une mine ou contenir un chiffre. Le chiffre indique le nombre de mines qui se trouve aux alentours de la cellule. Se placer sur la branche **Dev_Cells** pour ajouter la fonctionnalité des cellules. À vous de décider quand faire des commits et quels noms auront ces commits.

- Complétez la méthode ***addMine*** qui permet d'affecter une mine à la cellule. Cette méthode permet d'affecter une mine à la cellule lors de l'initialisation du jeu.
- Modifier la méthode ***getNeighbours*** qui permet d'indiquer le nombre de mines aux alentours de la cellule. Cette méthode permet de récuperer le nombre de mines aux alentours de la cellule pour l'afficher.
- Modifier la méthode ***isDiscovered*** qui permet de savoir si la cellule a été découverte. Cette méthode est utilisée lors de l'affichage mais aussi lorsque l'on choisit de révéler une case.
- Modifier la méthode ***isAMine*** qui permet d'indiquer si la cellule contient une mine. Cette méthode sera utilisée comme condition d'arrêt lorsqu'on découvre les cases par récursivité mais aussi pour afficher les mines lorsque l'on a perdu.
- Modifier la méthode ***hasNeighbours*** qui permet de savoir si la cellule possède des voisins mines. Cette méthode est utilisée aussi comme condition d'arrêt lorsqu'on découvre les cases par récursivité.

Fonctionnalité de la grille

L'objet ***Game*** correspond à la plateforme du jeu qui va contenir un tableau à 2 dimensions des cellules du jeu, qui va être la grille du jeu. Cet objet devra gérer l'affichage de la grille et la découverte des cellules dans un premier temps. Se placer sur la branche **Dev_Game** pour ajouter la fonctionnalité du jeu. À vous de décider quand faire des commits et quels noms auront ces commits.

- Modifier la méthode ***getX*** qui permet de retourner le nombre de lignes de la grille. Cette méthode est utilisée pour borner le déplacement dans la grille au niveau des lignes.
- Modifier la méthode ***getY*** qui permet de retourner le nombre de colonnes de la grille. Cette méthode est utilisée pour borner le deplacement dans la grille au niveau des colonnes.

2.8 Intégration des deux fonctionnalités

Apres avoir completé les deux fonctionnalités précédentes, intégrer ces fonctionnalités dans la branche **master** et tester le fonctionnement.

2.9 Fonctionnalité drapeau

Dans le jeu démineur, on a la possibilité d'ajouter des drapeaux sur des cases si on pense qu'une mine s'y trouve. Créer une nouvelle branche **Dev_Flags** et se deplacer dessus.

- Modifiez la méthode ***isFlagged*** pour indiquer si la cellule contient un drapeau. On utilise cette méthode, lors de l'affichage, pour marquer les cases qui ont un drapeau par un F.
- Modifier la méthode ***flag*** qui permet de mettre ou d'enlever un drapeau sur la cellule. Cette méthode est utilisée lorsqu'on veut placer ou enlever le drapeau sur la case.

Intégrez cette fonctionnalité à votre branche **master** en utilisant le rebase et vérifiez le fonctionnement du jeu.

2.10 Fonctionnalité de fin de jeu

Si le joueur tombe sur une mine, le jeu doit savoir que le joueur a perdu et afficher un message en indiquant qu'il a perdu. Si le joueur découvre toutes les cases sans tomber sur une mine, le jeu doit indiquer si le joueur a gagné et afficher un message indiquant qu'il a gagné.

Sur une nouvelle branche développez les deux fonctions suivantes :

- L'événenement qui permet de savoir si le joueur a découvert toutes les cases sans tomber sur une mine.
- L'événememement qui permet de savoir si le joueur est tombé sur une mine.

Indice : La seule méthode à compléter/modifier est ***Game : :discover***.

Fusionnez vos fonctionnalités en les récuperants du dépot distant. On doit ici retrouvez deux commits.

2.11 Ajouter en dernier les enseignants à votre dépôt

Pour cette dernière étape il vous est demandé d'ajouter les deux enseignants à vos dépôts distants pour qu'ils puissent récupérer votre travail. Vous trouverez les enseignants sous les pseudonymes suivants :

- Pedro Carvalho Mendes : PedroCarvalho64
- Nicolas Otal : NicolasO-git