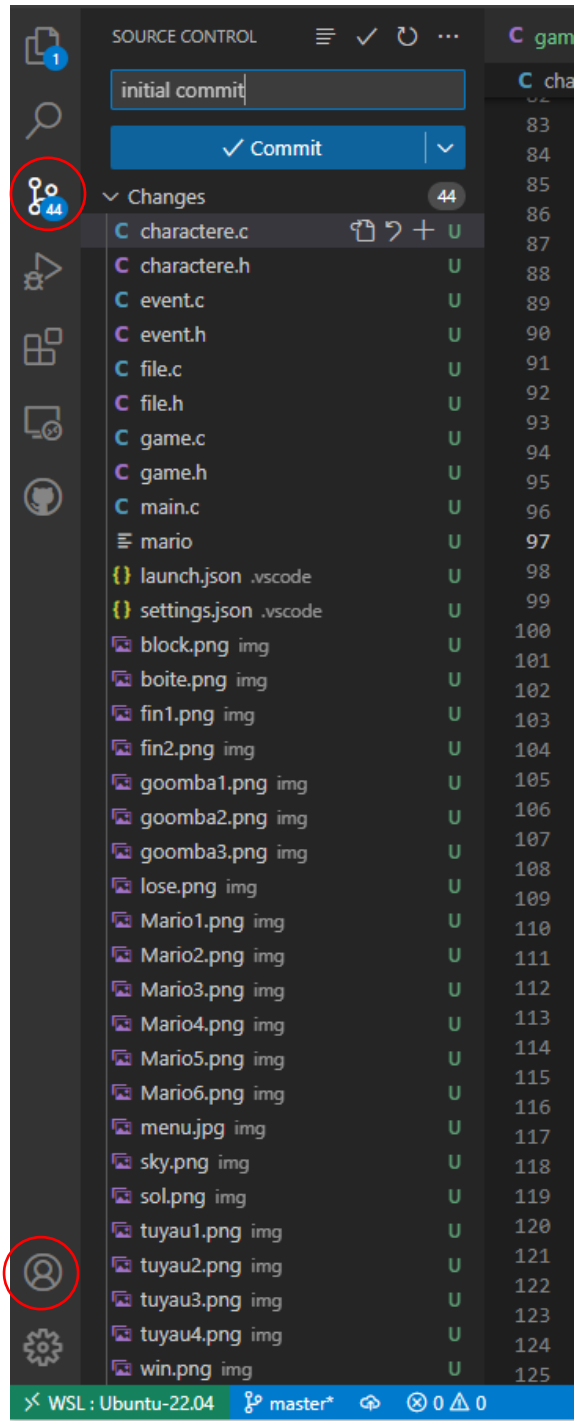
Visual studio code, github

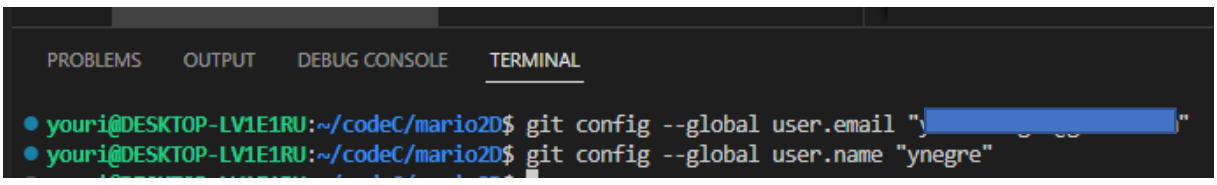
I/ Créer un dépôt et se lier dessus

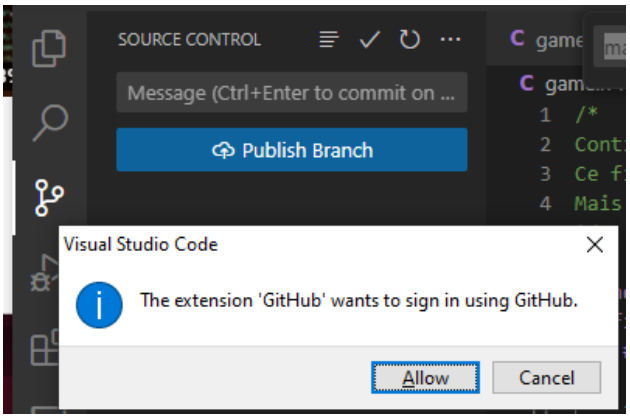


- Vous devez créer un compte Github et vous connecter dessus depuis votre navigateur internet.
- Vous devez cliquer sur l'icône 1 et vous connecter sur votre compte Github depuis VS code (sign in to sync setting, ….), suivez les boites de dialogue
- Vous devez dans un premier temps cliqué sur le bouton à gauche pour avoir accès au source control de votre projet.
- Vous devez ensuite cliquer sur initialiser un répertoire pour avoir accès à la fenêtre ci-contre.
- Vous pouvez alors réaliser votre premier commit en local, n'oublié pas de rajouter un message en haut.
- Précisez ensuite sur la fenêtre suivante que vous voulez indexer (staged) et commit l'ensemble de vos fichiers en même temps.
- Vous pouvez alors avoir une fenêtre vous proposant de configurer vos user.name et user.email. Acceptez et rentrez les lignes suivantes dans le terminal (voir les 2 images suivantes) :

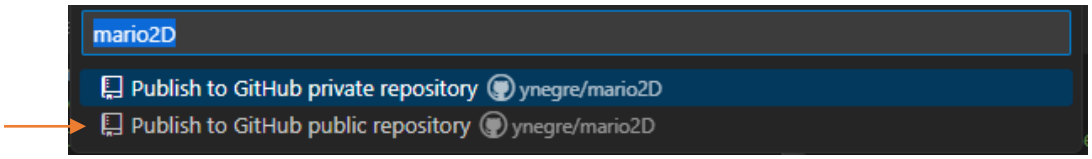
```
•	git config --global user.email "you@example.com"
•	git config --global user.name "Your Name"
```

PS : Le petit U à côté de chacun des fichiers signifie qu'ils n'ont jamais été encore commit et ne sont présent que dans mon workspace.

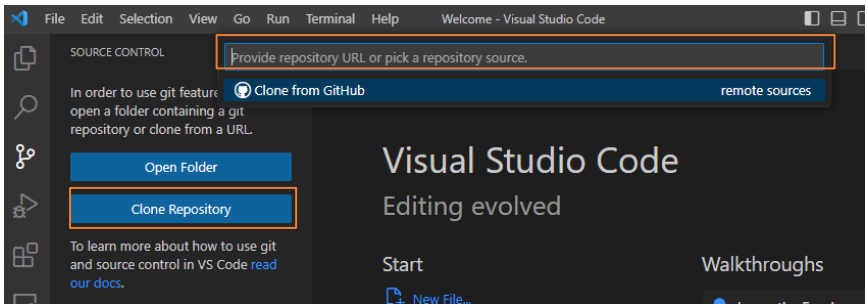


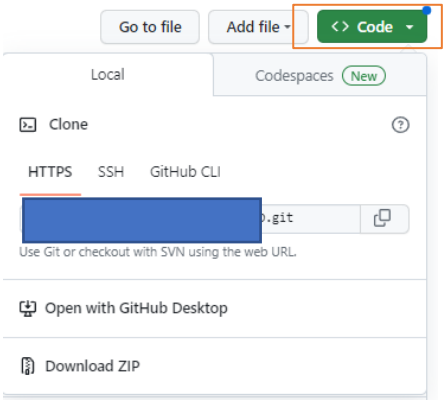


- Vous devez alors cliquer sur « publish branch » et cliquez sur « allow » dans la fenêtre suivante et autoriser les permissions entre vscode et Github. Le but ici est de stocker votre programme sur un repro de github.
- Il vous manque plus que à publiez le repro sur votre github (vous devez avoir un compte et y être connecté sur votre navigateur internet). Choisissez un repro public pour que votre binôme puis y avoir accès.



Félicitation vous venez de créer un repro sur github et de vous lier avec votre compte github. Vous n'avez besoin que d'un seul repro par binôme. Mais les 2 personnes du binôme doivent être lié au compte github. Pour se faire, la deuxième personne qui n'aura pas créé le dépôt devra juste cliquer sur « clone Repository » et mettre le lien du dépôt github (appuyer sur le bouton code et récupérer le lien https en .git). Vous devrez ensuite choisir un dossier local ou sera votre git et vous connecter à votre compte personnel github (ne pas oublier de configurer votre mail et username).

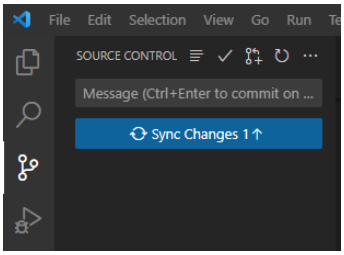




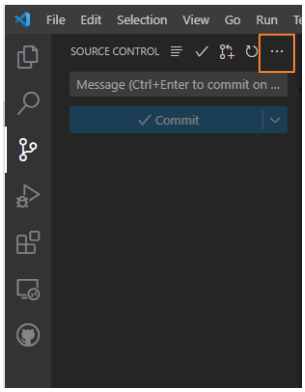
II/ Commit, pull et conflit

A chaque fois qu'une modification est effectuée et que vous souhaitez l'envoyer à votre binôme, il vous faudra effectuer les tâches suivantes :

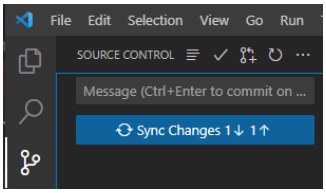
- Sauvegarder l'ensemble de vos changements
- Aller sur l'ongle (à gauche) source control
- Rédiger un message expliquant ce qu'il y a dans le commit
- Appuyer sur le bouton commit (vous voyez dans la liste en bas, l'ensemble des fichiers qui seront envoyé), puis sur le bouton « sync changes » situer au même endroit.



Pour que votre binôme puisse récupérer les fichiers envoyés sur le github, il doit faire une demande de pull, pour cela il faut cliquer sur les 3 petits points puis sur pull. Les changements seront tout de suite intégrés dans votre code.



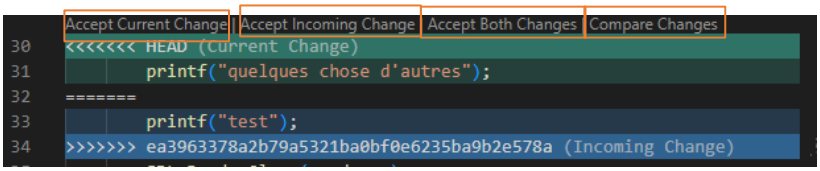
Parfois il peut y avoir des conflits entre le code que vous voulez envoyer sur le dépôt et celui qui s'y trouve (changer par votre binôme). C'est pour cela que vous devez à chaque fois que vous faîtes un commit en local (avant de sync avec le Github) vous devez faire un pull. Vous verrez peut être la fenêtre suivante (avec un synch changes arrivant et partant) symbolisant un problème de conflit.



Vous devez alors régler le conflit à la main, en utilisant une des 2 méthodes suivante :

- Régler le conflit in line
- Régler le conflit via le merge editor.

La première option doit vous amener vers quelques choses de similaire à ceci :



Ce qui est en vert représente ce qui est en local, tandis que ce qui est en bleu représente les changements qui arrive depuis le Github. Il s'agit de 2 lignes avec un printf, qui sont en réalité sur la même ligne dans le code, ici on me demande soit :

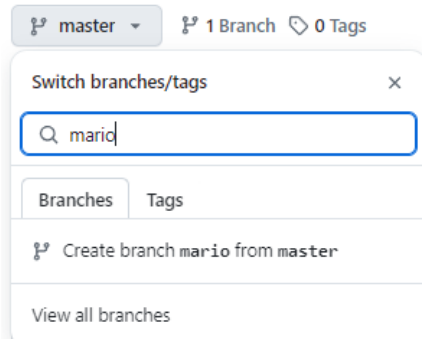
- De garder les changements présents en local (accept current change)
- De garder les changements arrivant (accept incoming change)
- De garder les 2 (accept both changes)
- De comparer les 2 changements et de faire des changements plus minutieux

Il vous suffit alors de cliquer sur une des 4 options au-dessus en fonction de ce que vous souhaitez faire.

Vous pouvez aussi ouvrir le merge editor, normalement vous devez avoir un petit onglet bleue ouvert en bas à droite de votre fenêtre (Resolve in merge editor). En cliquant dessus vous pouvez voir 3 fenêtres s'ouvrir, celle représentant votre code, celle représentant le code reçu et finalement celle représentant le code de base sans ces changements. Vous pouvez alors assez facilement comparer et effectuer les modifications adéquates de façon similaire à la méthode in line ou à la main.

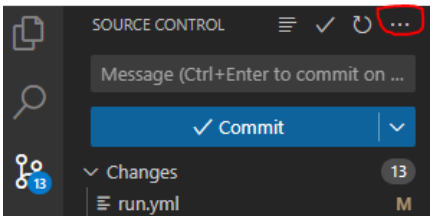
Bien entendu il faudra réaliser cette opération pour chaque conflit que vous aurez. Une fois que tous les conflits sont résolus, vous pouvez envoyer l'ensemble sur le github.

III/ Branche



Vous pouvez créer plusieurs branches pour que chaque personne code sur une branche séparé. Pour se faire vous devez aller sur github, cliquez sur master et rentrez le nouveau nom de votre branche et cliquez sur « create branch NAME from master ».

Il vous suffit alors d'ouvrir votre vscode dans l'onglet propre à Github, vous cliquez ensuite sur les 3 points -> Pull, Push -> Fetch.



Il vous suffit alors de cliquez sur « master » en bas à gauche de VS code pour qu'une nouvelle fenêtre de dialogue s'ouvre et que vous puissez choisir la nouvelle branche que vous avez juste créé. Félicitation, maintenant tout vos commit seront sur la nouvelle branche. Quand la fonctionnalité sera fini, vous n'aurez plus que à fusionner sur la branche master.