BOURGEOIS Charles SIO2
StJoSup, Le Havre

Compte-rendu
Git

2023

Sommaire

1) Qu'est-ce que Git ?

1.1) Définition

Git est un logiciel de gestion de versions décentralisé.

Il est gratuit et a été créé en 2005 par Linus Thorvald.

Il permet aux développeurs de travailler simultanément sur un même projet, de suivre l'historique des modifications sur des branches et de gérer ces dernières de façon efficace et intuitivement.

Git offre également des fonctionnalités utiles, comme la fusion de branches et la gestion des conflits. En effet, gérer les conflits dans un travail collaboratif est extrêmement important, car Git, si des modifications ont lieu sur le même fichier, ne pourra pas déterminer automatiquement la bonne version du fichier et cela reviendra au développeur de le faire manuellement.

Il est important de gérer les conflits dans un projet Git car cela garantit que toutes les modifications sont prises en compte de manière appropriée et que le code fonctionne correctement. Si les conflits ne sont pas résolus, cela peut entraîner des erreurs dans le code, des bogues et des incompatibilités qui peuvent causer des problèmes dans le projet.

Git est donc un outil essentiel du travail dit « *collaboratif* ».

1) Qu'est-ce que Git ?

1.2) Le versioning

Le versioning est une méthode de gestion des versions sur un même projet.

Cela consiste à suivre les modifications apportées à un ensemble de fichiers (ou à un projet) au fil du temps, en créant des versions différentes à chaque étape du développement. On l'utilise pour le développement de logiciels, sites web et dans d'autres projets collaboratifs.

C'est un suivi de l'historique des modifications apportées au code source, cela permet également de revenir à la version précédente en cas de bug sur une nouvelle version.

Il existe plusieurs types de systèmes de versioning, tels que le versioning centralisé, dans lequel un serveur central contient toutes les versions du code source, et le versioning décentralisé, dans lequel chaque développeur possède une copie locale du code source.

1) Qu'est-ce que Git ?

1.3) GitLab

GitLab est ce que l'on appelle un logiciel de forge basé sur Git et proposant l'entièreté de ses fonctionnalités.

Cette plateforme offre des fonctionnalités complètes de gestion de version, de suivi des problèmes, de pipelines d'intégration continue, de tests automatisés, de déploiement et de collaboration en temps réel.

Durant les TP concernant Git, toutes mes manipulations se sont fait grâce à GitLab.

GitLab est un que logiciel open source que l'on peut installer sur son propre serveur ou en tant que service cloud qui peut être utilisé en ligne. GitLab permet également d'effectuer des documentations libres, sur le code que l'on intègre au projet.

Il reprend tout ce que l'on a évoqué à propos de Git.

2) Equivalent à Git : TortoiseSVN

TortoiseSVN est un logiciel de gestion qui permet également de gérer les versions d'un fichier et de dossier sur un serveur distant.

Il s'ajoute, comme une extension, à l'explorateur de fichiers Windows, ce qui facilite son utilisation et ses fonctionnalités directement disponible depuis le système d'exploitation Windows.

TortoiseSVN permet la vérification des fichiers, la navigation entre différentes versions du fichier, création de branches et fusion des fichiers.

C'est donc également un logiciel de gestion de projet en équipe collaborative.

3) Mise en place – Travail collaboratif

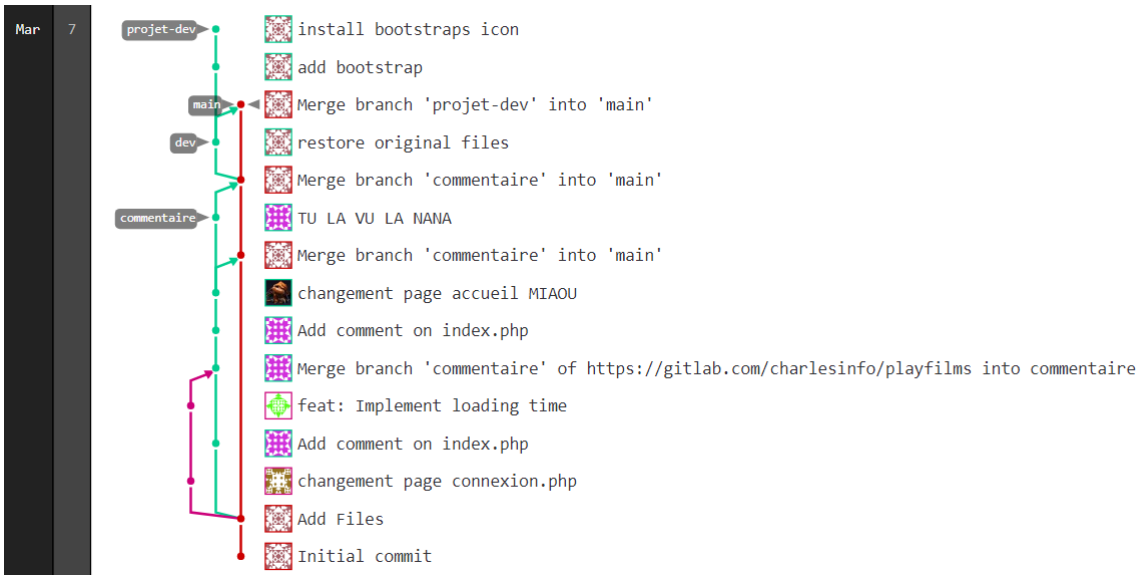
Durant ce TP, 4 étudiants et moi avons décider de mettre en place GIT dans un de mes projets afin de tester les fonctionnalités proposés et de mettre en place le travail collaboratif.

J'ai donc créé un projet GIT, auquel j'ai ajouté, grâce à la fonction Clone de GIT, mes fichiers.

Une fois mes fichiers fait j'ai envoyé un lien d'invitations à 4 autres étudiants qui ont pus se connecter sur mon projet GIT et modifier les fichiers contenu dans ce projet.

Chacun a effectué sa modification et nous avons ensuite effectué ce que l'on a fait un commit, qui consiste à écrire une légende de la modification qui a été faite. Une fois tous les commit fait, nous avons fait un push et une Merge Request, qui consiste à vérifier le contenu des modifications (une sorte de review) puis de mettre toutes les modifications sur une branche.

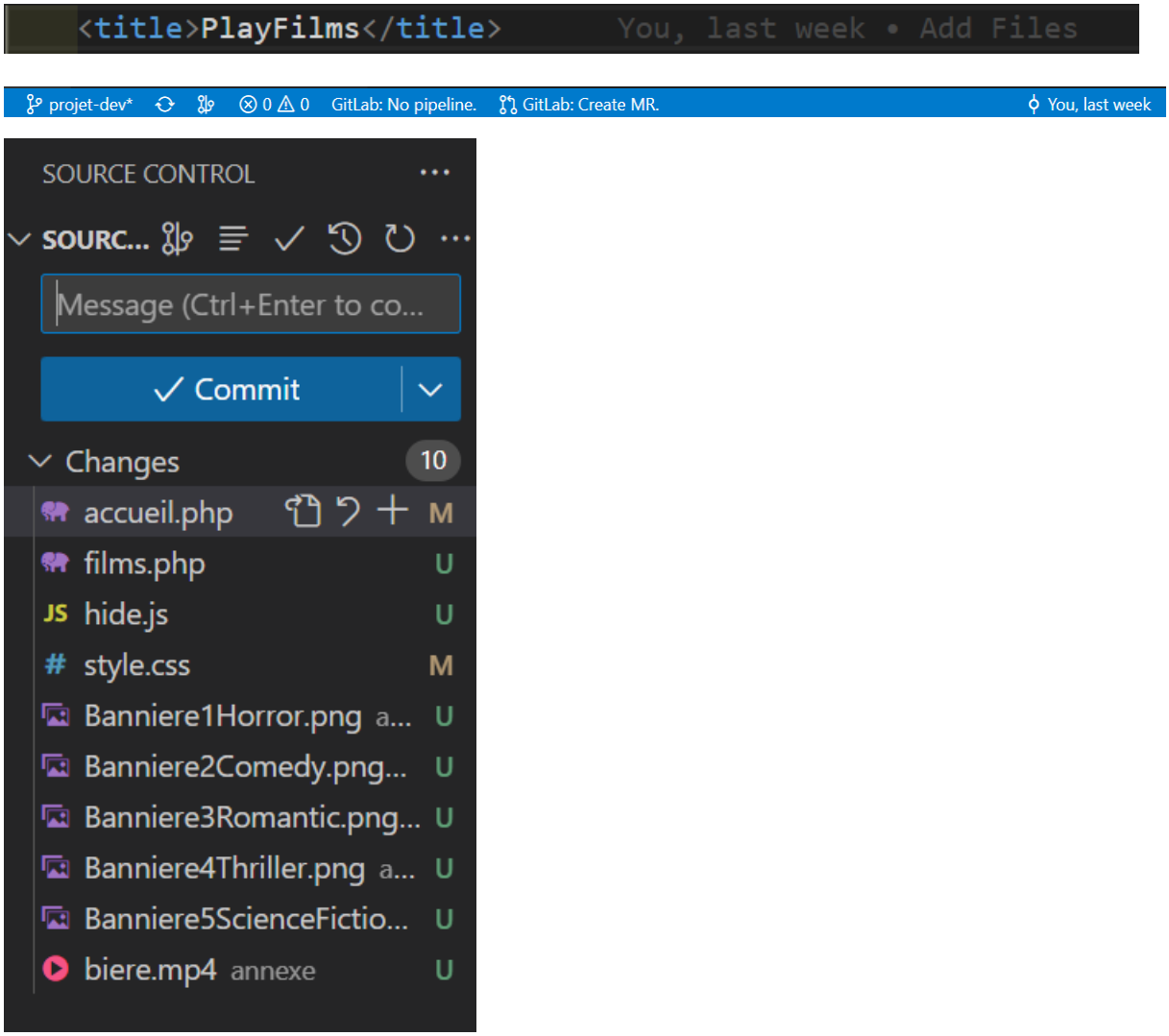
Voici la représentation graphique :



La merge request permet d'assembler le tout sur notre branche Main qui contient le projet dans son ensemble.

Une option supplémentaire était d'ajouter l'extension GIT directement depuis l'IDE Visual Studio Code.

En effet, grâce à Visual Studio Code, nous pouvons manipuler directement chaque fonctionnalité de GIT, le déplacement de branches en branches, visualiser les modifications faites et l'auteur de cette modification.



Ici, nous pouvons voir tous les fichiers ayant eu une modification avant le commit and push.

4) Conclusion

GIT est un logiciel essentiel en entreprise, en effet c'est sur ce logiciel que tout projet est fondé. Il permet de travailler en groupe, de manipuler différentes versions d'un grand projet, de pouvoir sauvegarder et maintenir un projet à jour.

Dans le contexte du BTS, il peut être très utile pour des projets de groupe nécessitant une « review » d'un professeur.

Il est donc nécessaire de bien approprier GIT et de connaître le terme de versioning.