To Git or not to Git?

Git c'est l'outil qui va faciliter la gestion de ton projet, des fichiers de codes et autres, également du travail à plusieurs en parallèle. Cela permet d'avoir un suivi chronologique des modifications apportées au projet (ou plutôt des différentes versions), aux codes sources, à la doc et tout ce qui compose le projet.

Grâce à git tu as donc un historique et une traçabilité de l'évolution de ton projet, qui te permet d'une part de voir où en est le projet, et quelles modifications ont été apportées, par qui et à quelle date. Pratique pour te rafraichir la mémoire lorsque cela fait plusieurs mois que tu n'as pas travaillé sur le projet en question, et également pratique pour faire marche arrière si tu te rends compte que la nouvelle version n'est finalement pas adaptée, ou provoque des bogues.

Commit

Un « commit » c'est la validation des modifications apportées à un ou plusieurs fichiers, avec un commentaire récapitulatif des modifications apportées à ceux-ci. Cela te permet d'enregistrer les différentes versions de ton projet sous forme d'historique et de pouvoir à tout moment revenir dans le temps à un instant t. Egalement, cela enregistre l'auteur, son mail, et la date du commit.

Git log

La commande git log est tout simplement l'affichage de ton historique de versions avec le commentaire associé, l'auteur de la soumission, son email, la date et l'heure de soumission. C'est donc la liste de tous les « commit » effectués sur le projet, pour visualiser le fil des évènements chronologiques. De plus, chaque commit possède son identifiant unique, qui est également affiché, et qui te permet de revenir à cet instant, sans pour autant perdre les commit plus récents. Tu peux naviguer entre les différentes versions aisément.

Branche

Une branche est le parcours que suit un projet. Par convention la branche principale (le tronc de l'arbre) s'appelle « master ». Cette branche principale peut donc avoir des branches secondaires, des « ramifications », dans lesquelles le projet peut suivre des modifications différentes, ou des tests, sans toucher à la branche principale (qui serait donc le « vrai » projet). Une branche peut aussi correspondre à un « morceau » de projet, codé par une personne pour ajouter des fonctionnalités par exemple, puis un autre morceau codé par une autre personne dans une autre branche.

Ces branches peuvent ensuite être fusionnées entre elles, et/ou à la branche « master » pour faire évoluer le projet dans une direction suite à toutes ces modifications, ou pour l'ajout de nouvelles fonctionnalités codées séparément par d'autres personnes par exemple.

En *Figure 1* Exemple d'un git log est représentée une capture d'écran d'un git log sur la branche « master » d'un projet de cours de programmation JAVA.

Il faut savoir qu'il est ensuite possible de centraliser son travail local sur une plateforme/serveur externe ou sur le cloud, afin d'y avoir accès à tout moment via internet, et avec éventuellement une interface plus informative et intuitive à la consultation. Par exemple, une très connue et utilisée est « GitHub ».

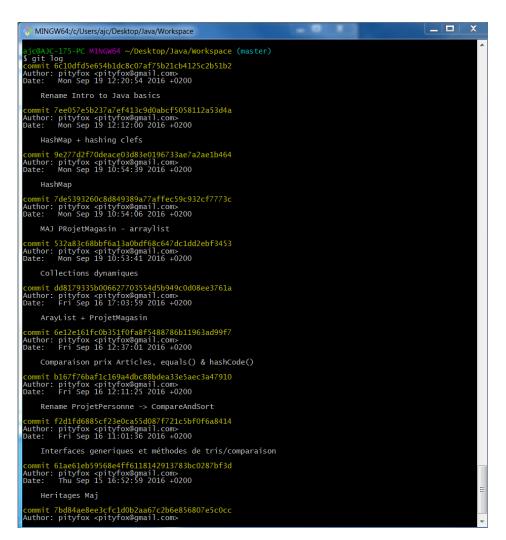


Figure 1 Exemple d'un git log