

- Client **git** installé (en natif ou avec **vscode**)
- Utilisation du dépôt pour la gestion des sources (opération clone, commit,tag, push ...)
- Formation/assistance des développeurs
- Tous les fichiers dignes d'intérêt (scripts, fichiers de configuration, ...) devront être stockés sur le dépôt **git**
- **Pour cette mission nous avons installés sur les machines des développeurs (étudiants SLAM) le logiciel Git Bash.**

Définition :

Git Bash est un logiciel qui fournit une interface en ligne de commande (CLI) permettant d'utiliser Git, un système de contrôle de version populaire, sur le système d'exploitation Windows.

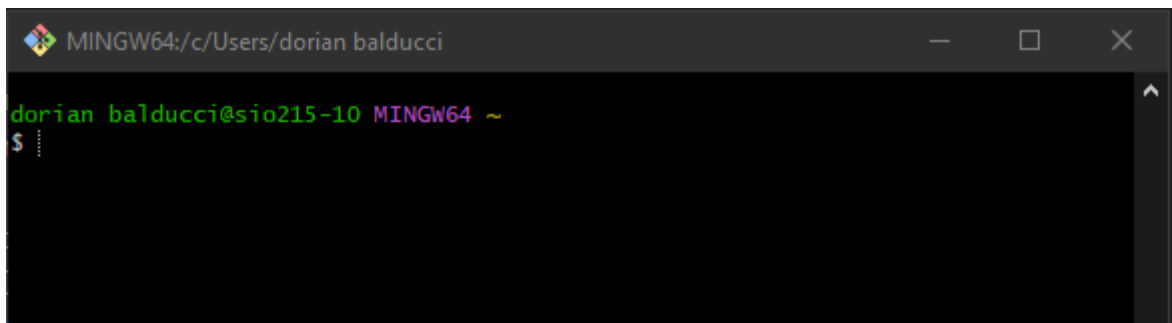


Il offre une expérience similaire à celle de la ligne de commande Git sous Linux ou macOS, permettant aux utilisateurs de gérer leurs dépôts Git, de créer des branches, de fusionner des modifications et d'effectuer d'autres opérations Git.

Git Bash inclut également un ensemble d'outils Unix couramment utilisés, ce qui le rend compatible avec de nombreux scripts et commandes Bash existants.

Lien téléchargement : <https://git-scm.com/download/win> ou <https://gitforwindows.org/>

Voici l'interface de Git Bash :



Comme sur Linux, voici les commandes à utiliser (dans l'ordre) pour envoyer un fichier sur un dépôt Git avec Git Bash sur Windows :

1. **git clone http://gitea.lyc-lecastel.fr/nom.session.utilisateur/nom.depot** : Cette commande permet de cloner le dépôt Git distant à partir de l'URL fournie (lien à récupérer sur le site Gitea du lycée Le Castel) :
2. **mkdir repertoire_exemple_depot** : Cette commande crée un répertoire local pour le dépôt cloné, permettant de le gérer localement. (Se déplacer dedans avec : **cd /nom.depot/repertoire_exemple_depot** puis copier les fichiers à envoyer avec **cp /chemin/du/fichier/a/envoyer/ . /!** bien mettre un « un espace et un point » à la fin du chemin.)
3. **git status** : Cette commande affiche l'état actuel du dépôt, montrant les fichiers modifiés ou non suivis.
4. **git add *** : Cette commande ajoute tous les fichiers modifiés ou non suivis au suivi de version, les préparant pour la prochaine étape.

5. **git status** : Après avoir ajouté les fichiers, cette commande affiche à nouveau l'état du dépôt pour refléter les modifications.
6. **git config --global user.email "user@exemple.com"** : Cette commande configure l'adresse e-mail de l'utilisateur pour les commits ultérieurs.
7. **git commit -am "Écrire un commentaire"** : Cette commande crée un commit avec un message de commentaire spécifié, en enregistrant les modifications dans l'historique du dépôt.
8. **git tag v0.0.1** : Cette commande crée une étiquette (tag) pour marquer une version spécifique du dépôt.
9. **git status** : Après le commit et l'étiquetage, cette commande affiche à nouveau l'état du dépôt.
10. **git push** : Cette commande pousse les modifications locales vers le dépôt distant.
11. **git push --tag** : Cette commande pousse également les étiquettes (tags) créées vers le dépôt distant.