

Git

Exercice 1 : Git

1. Installer Git :
 - Sur linux, taper dans un terminal `apt-get install git-all`;
 - Sur MacOS, télécharger git à l'adresse suivante <http://git-scm.com/download/mac>.
 - Sur windows, télécharger git pour windows à l'adresse suivante <http://git-scm.com/download/win>.
2. Créer un repertoire TD7 et initialiser un dépôt Git à l'intérieur.
3. Créer une classe `Animal` qui a un nom (attribut de type `String`) et une position dans le plan (deux attributs, un pour l'abscisse, un autre pour l'ordonnée). Faire un commit de vos modifications.
4. Créer deux classes, une classe `Chat` et une classe `Chien`. Ces classes héritent de la classe `Animal` et implémentent une méthode `seDeplacer(x,y)` qui déplace l'animal d'une translation de vecteur (x,y) . Faire un commit de vos changements.
5. Regarder l'historique des commits et regarder la différence entre votre version actuelle et le commit précédant.
6. Rajouter une classe `Lapin` et faire un commit qui écrase le commit précédant. Laisser volontairement une erreur dans votre code.
7. Donner un tag à votre version actuelle.
8. On va maintenant créer une fonctionnalité `FaireDuBruit` pour nos animaux. Créer une branche `bruit` et basculer dessus.
9. Commencer par corriger l'erreur précédemment laissée. Faire un commit.
10. Repasser sur la branche `master`. Faire un cherry-pick de votre correction puis repasser sur la branche `bruit`.
11. Certains animaux comme les chats et les chiens implémentent la méthode `faireDuBruit`, pas les lapins ou les tortues. Prenez cette information en compte pour implémenter cette fonctionnalité dans les classes `Chien` et `Chat`. Faire un commit.
12. Repasser sur la branche `master`. Remplacer les attributs `abscisse` et `ordonnée` par une classe `Point`. Faire un commit de vos changements.
13. Montrer le commit précédemment taggé avec `show`.
14. Effectuer un rebase de votre branche `bruit`.
15. Fusionnez votre branche `bruit` avec la branche `master` puis supprimer la branche `bruit`.
16. Créer un compte sur [GitHub.com](https://github.com)
17. La classe se répartit en binômes. Un membre de chaque binôme crée un projet privé et y invite son camarade comme collaborateur.

18. Le premier change le constructeur par défaut de `Point` avec les valeurs (10,10). Le second membre du binôme change le constructeur par défaut de `Point` avec les valeurs (20,20). Le premier push. Le second pull, corrige le conflit et push de nouveau.