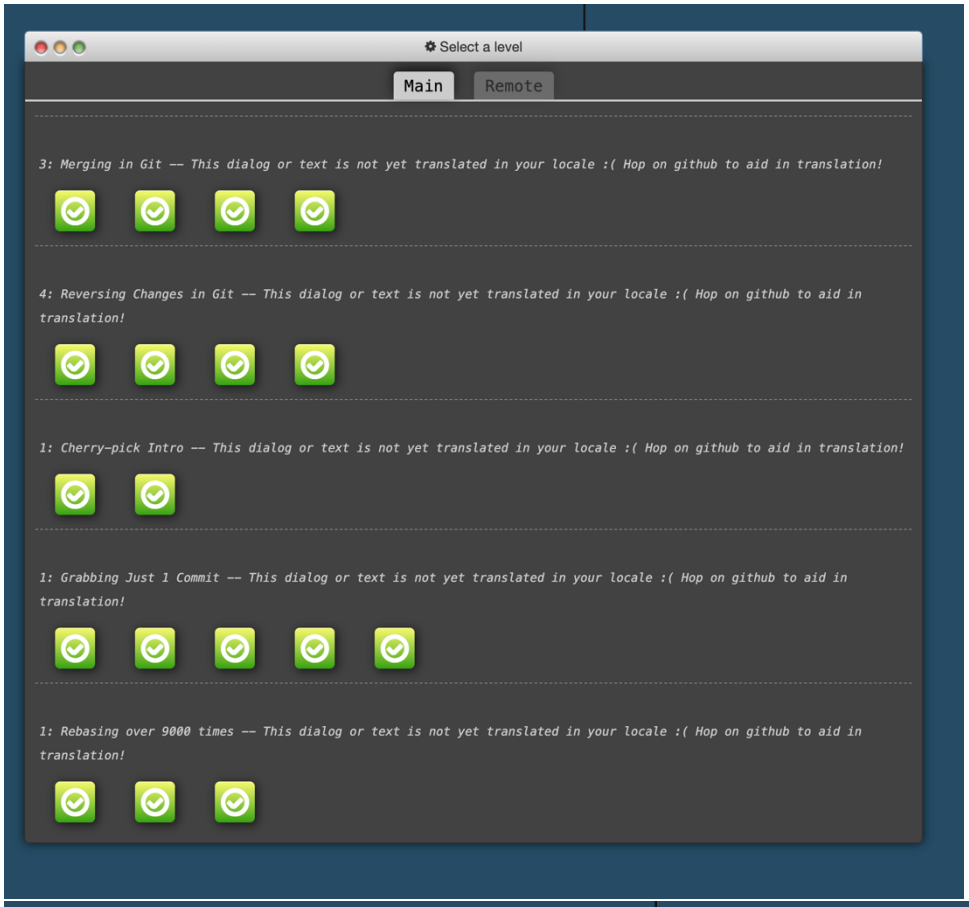
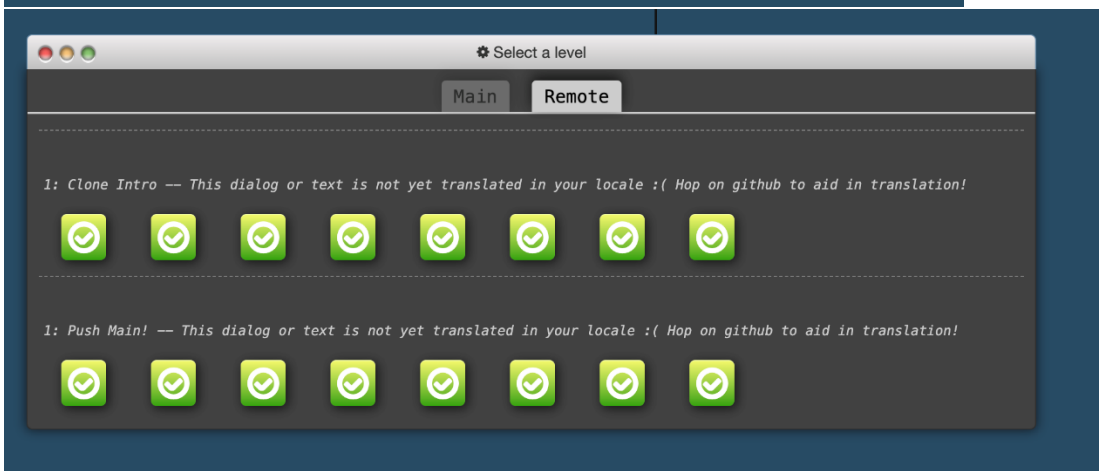
SCHERRER Guillaume

Rapport commandes Git

Screenshots LearnGitBranching:





Définition rapide des commandes Git suivantes:

-„**Git config**“: permet de configurer notre profil lors de la création d'un Repo d'un projet
On peut ainsi définir son pseudo, son email, son éditeur de texte par défaut etc…
En utilisant l'option -global, les réglages sont valables pour tous les projets sur la machine.
Exemple : $ git config user.name « Guillaume Scherrer”
$ git config - - list affiche les réglages courants

-„**Git init**“: créé un nouveau Repo git d'un projet.
Un sous-répertoire .git est crée qui contient les fichiers de description du dépôt

-„**Git status**“: affiche l'état des fichiers du Repo. Il informe sur ce qui s'est passé concernant git add et git commit. Cela permet de voir l'état des changements en vue d'un commit ou si un changement a besoin d'un commit ou alors si il n'y a rien à faire.

-„**Git add**“: ajoute tous les fichiers qui devront être pris en compte pour le prochain commit (commit=on fige l'état du dépôt).
Exemple : $ git add exemple.c

-„**Git push**“: transfère les fichiers (qui ont eu un commit) du Repo local vers le Repo sur le serveur distant github.

-„**Git merge**“: permet de fusionner 2 branches ensemble. En général l'on se sert de cette commande pour intégrer des changements effectués sur une branche sur la branche principale. S'il y a des conflits, un merge automatique n'est pas possible
Exemple : $ git merge ma_branche va fusionner ma_branche avec la branche courante.

-„**Git diff**“: affiche les différences faites entre 2 commits , 2 branches, 2 fichiers, 2 versions de fichiers etc…
Exemple : $ git diff branche1 branche2

-„**Git blame**“: permet de voir l'historique des modifications ligne par ligne d'un fichier (à quelle date et par qui).
Exemple : $ git blame mon_fichier.txt