Un logiciel de gestion de versions décentralisé

—

Concepts de base

UNIVERSITÉ Grenoble Alpes

Position du problème

- La version 1 d'un logiciel fonctionne
  - Pour arriver à la version 2, une dizaine de bugs mineurs doivent être corrigés
  - Chaque bug est corrigé, mais à un moment dans la chaine de production, quelque chose s'est cassé…
  - Il faudrait revenir en arrière, ce serait plus simple de ne pas ajouter l'erreur que de la corriger…
- 2 codeurs (A et B)
    - A travaille sur le module 1
    - B travaille sur le module 2
    - → pas de problème…
  - …sauf que les deux modules dépendent d'une librairie commune développée pour le projet
    - A et B ont chacun besoin d'une fonction spécifique qu'ils ajoutent à la librairie
  - A et B mutualisent leur travail : rien ne marche
    - A se rend compte qu'il n'a pas ajouté sa modification de la librairie, espoir…
    - Il remplace le fichier lib.js par le sien… ça ne marche toujours pas
    - B se rend compte que la librairie doit aussi contenir ses modifications…
    - Galère…

UNIVERSITÉ Grenoble Alpes

Qu'est-ce que “git” ?

- Créé par Linus Torvalds
- Open source (http://git-scm.com/downloads)

```
loizbek@bajaja: ~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ ls
bin  browserquest.conf  client  launchServer.sh  LICENSE  node_modules  package.json  README.md  server  shared  tools
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git branch
* janus
  mapedit
  master
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git checkout mapedit
Basculement sur la branche 'mapedit'
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git branch
  janus
* mapedit
  master
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ ls
bin                client    epi1.html  histart1.html  index.html       LICENSE       package.json  quizz       server  TODO.md
browserquest.conf  dump.rdb  epi2.html  histart2.html  launchServer.sh  node_modules  print         README.md   shared  tools
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$
```

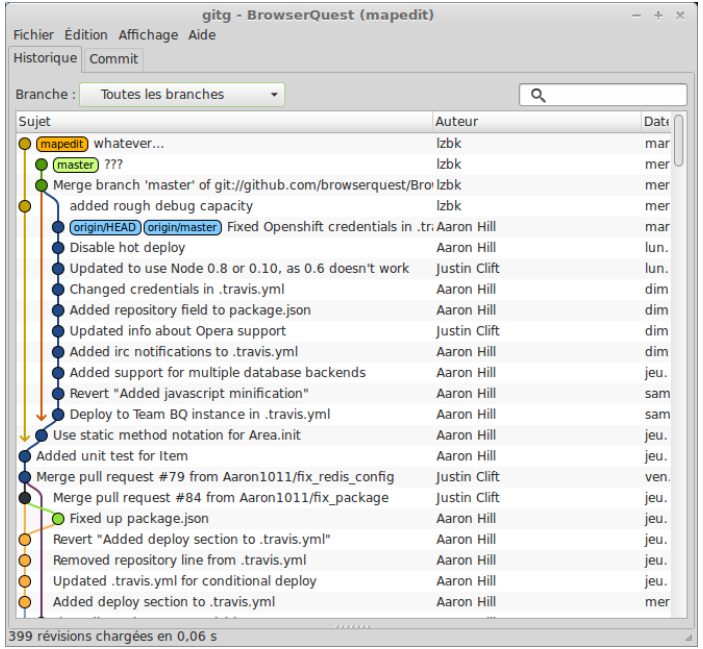
des
révisions...

UNIVERSITÉ Grenoble Alpes

Qu'est-ce que “git” ?

- Créé par Linus Torvalds
- Open source
  http://git-scm.com/downloads
- Plusieurs points de vue
  - En ligne de commande…
  - Avec une interface graphique…

```
loizbek@bajaja: ~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ ls
bin  browserquest.conf  client  launchServer.sh  LICENSE  node_modules  package.json  README.md  server  shared  tools
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git branch
* janus
  mapedit
  master
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git checkout mapedit
Basculement sur la branche 'mapedit'
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ git branch
  janus
* mapedit
  master
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$ ls
bin               client    epi1.html  histart1.html  index.html       LICENSE       package.json  quizz      server  TODO.md
browserquest.conf  dump.rdb  epi2.html  histart2.html  launchServer.sh  node_modules  print         README.md  shared  tools
loizbek@bajaja:~/Documents/travail/projets/JANUS/_testBQ/BrowserQuest$
```



UNIVERSITÉ Grenoble Alpes

Utilisation du terminal

Unix	DOS
Changer de dossier	
`~ $ cd chemin/du/dossier`	`> cd chemin\du\dossier`
Lister le contenu d'un dossier	
`~ $ ls`	`> dir`
Créer un dossier	
`~ $ mkdir nomDuDossier`	`> md nomDuDossier`
Copier un fichier	
`~ $ cp source destination`	`> copy source destination`
Supprimer un fichier	
`~ $ rm nomDuFichier`	`> del nomDuFichier`

Pour “remonter” d'un dossier “`..`”
Ex : `cd ..`
Utiliser la touche **tabulation** pour l'auto-complétion

Option “`-R`” → commande “récursive”
Ex : `cp -R src tgt`
Copie **src** et son contenu vers **tgt**

UNIVERSITÉ Grenoble Alpes

Quelques commandes unix en plus...

- `touch nomFichier`
  → nouvel accès au fichier (le crée s'il n'existe pas encore)
- `cat nomFichier`
  → affiche le contenu d'un fichier
- `nano nomFichier`
  → lance l'éditeur de texte (en ligne de commande) « nano »
  - Pour le quitter : ctrl+X
  - Il vous demandera si vous voulez sauvegarder les modifications

UNIVERSITÉ Grenoble Alpes

Exercice

Tout en ligne de commande...

- Créer un dossier `tmp`
- Aller dans le dossier `tmp`
- Créer un fichier `test1.txt`
- Modifier ce fichier (si possible avec nano)
- Afficher son contenu
- Le copier 2 fois vers `test2.txt` et `test1.md`
- Modifer `test2.txt`
- Créer un dossier `originaux` dans `tmp`
- Déplacer tous les fichiers `*.txt` dans `originaux`
- Copier `originaux` vers un dossier `nouveaux`
- Supprimer `tmp`

Des La commande(s) (sous-) “git”

- `git config` : Configurer l'outil git
- `git init` : Créer un dépôt git
- `git clone` : Créer un dépôt à partir d'un dépôt existant
- `git add` : Sélectionner des changements pour le prochain commit
- `git reset` : Désélectionner des changements pour le prochain commit
- `git commit` : Créer une nouvelle version
- `git checkout` : Changer de branche
- `git merge` : Fusionner deux branches
- ...

UNIVERSITÉ Grenoble Alpes

Cycle de vie (de base) d'un projet git

- Dépôt local
  - Initialisation
    - `git init`
  - Travail local
    - Modifications
    - Préparation du commit :
      - `git add fichier`
      - `git add .`
    - Commit
      - `git commit -m "<explication commit>"`

- Dépôt distant
  - Initialisation
    - `git clone <adresse>`
  - Travail
    - `git pull <dépot> <branche>`
    - Travail local
    - `git push <dépot> <branche>`