Compte rendu projet

Objects :

- **Hash** : divisée en deux fonctions, une fonction auxiliaire **chdc** qui prend en argument une liste du contenu de chaque Directory et retourne une chaine de caractères.

- **Store Object** : utilise **chdc** pour transformer le Directory en une chaine de caractères et la stocke dans un fichier dont le nom sera le hash du contenu (en hexadécimal)

- **Read_text_object: Digest_to_hex** pour transformer le hash en string

- **Store_work_directory**: 3 fonctions auxiliaires:

  -file_to_object : utilise le chemin d'un fichier ou d'un répertoire et retourne l'objet correspondant
  -obj2files : prend en argument un objet de type t et crée le fichier versionné le décrivant en utilisant **storeObj** (prend en paramètre un objet type t et crée le fichier lui-même).

- **Read_directory _object** : utilise **read_text_obj** pour lire le contenu d'un fichier le diviser en lignes avec **split_on_char** du module String afin de créer les fichiers décrits par les fichiers versionnés.
  La ligne 63 : afin de satisfaire l'obligation du pattern-matching qui doit être exhaustif on a étudié un cas sur lequel on ne tombera jamais.
  Nous avons aussi défini une **fonction lTOl** qui prend en argument un string et qui retourne un tuple à l'intérieur même de la fonction **read_directory_obj** parce qu'on l'a utilisé à l'intérieur et vice versa et la définir ailleurs crée un bug.

- **Clean_work_directory** : on a créé une fonction auxiliaire clean qui prend en argument le chemin d'un directory et qui ne retourne rien, elle efface récursivement les fichiers et répertoires après avoir effacé leur contenu grâce à la fonction rmdir du module Sys.

- **Restore_work_directory** : fonction auxiliaire récursive **rwdirectory** à laquelle on ajoute en paramètre le chemin des répertoires. Pour la création des répertoires on a utilisé mkdir du module Unix en ajoutant les permissions nécessaires.

  Bugs rencontrés :
  - **Hash** : difficultés dans la fonction chdc puisqu'à la dernière ligne on ne peut pas faire de retour à la ligne (corrigé)
  - **Store_work_directory** : on a toujours la création d'un fichier vide dont le hash commence par d41 (hash d'un contenu vide) (corrigé en effaçant **storeObj** à la ligne 77) Dans **file_to_object** la liste des fichiers du directory était inversée (corrigé avec **List.rev**)
  - **Read_directory** : pattern matching non exhaustif corrigé avec la ligne 63
  - **Clean_work_directory** : efface par erreur des fichiers de **.ogit** (corrigé avec un changement de chemin relatif)
  - **Restore_work_directory** : difficultés à savoir le **file_exists** du module Sys fonctionne aussi pour les répertoires.

-tout à l'air de fonctionner sauf le merge (non fait)-

Logs :

- **Date_fm** : fonction auxiliaire twoDig qui transforme les nombres à un chiffre en nombres à deux chiffres.

- **Set_head** : fonction auxiliaire l**_to_string** prend une liste et la transforme en paragraphe en remplaçant les sep par des retours à la ligne \n.
- **Get_head** : ouvre le fichier head et transforme son contenu en hash de type **Digest.t** (32 bits)

- **Store_commit** : une fonction récursive auxiliaire pour transformer la liste de hash du head en une chaine de caractères et à l'aide de la fonction **date_fm** on recrée le commit c en une chaine de caractères qu'on stocke dans un nouveau fichier dont le nom est le hash du contenu.

- **Read_commit** : fonction auxiliaire **s_to_date** qui prend en arg une chaine de caractères d'une date et la transforme en un objet de type date à l'aide du module Unix.
  Dans cette fonction on a eu beaucoup de bugs au niveau du pattern-matching qui était non-exhaustif, **nous avons essayé de le corriger.**

Commands :

- **Ogit_init** : problème de signature à chaque fois le fichier head étant effacé il n'est plus détecté

Ayant perdu du temps sur Objects et logs en essayant de corriger les erreurs commises, nous n'avons pas pu avancer dans Commands.