

## TP 3 (2025-11-07) — Git (noté)

Rendu papier à la fin de la séance

Fichiers/liens par mail

Sujet du mail: "TOI/TP3 - <Nom complet>"

Décrivez votre réflexion — rédigez pour être lisible par un humain

---

### Déroulement et travail attendu

Dans ce TP vous allez effectuer des manipulations sur un dépôt Git existant. Il vous sera demandé les points suivants :

1. Pour chaque question, expliquez les étapes et commandes. Réfléchissez à vos choix et justifiez les autant que possible.
2. Remarquez que les commits contiennent votre nom et e-mail
3. Aucun fichier généré dans vos commits ou rendus
4. Rendez votre répo (ou seulement certains commits) sous la forme qui vous semble la plus adaptée (Git pack, git-format-patch(1), Forge publique, Forge privée, ...)

### Sujet du TP : réparer le dépôt de Kevin

Kevin a un projet personnel sympa qui consiste à faire le rendu de l'ensemble de Mandelbrot.

Cependant, il a mal récupéré les sources et il semble qu'elles soient cassées. De plus, il avait codé la couleur dans une des branches et n'a pas réussi à la faire marcher.

Dans ce TP nous allons suivre les étapes pour réparer son code et retrouver une fractale de Mandelbrot fonctionnelle et colorée !

Pour récupérer le code :

```
git clone https://github.com/trosh/toi2025.git répo/à/créer
git clone chemin/vers/kevin.pack répo/à/créer
```

### Exercice 1 : structure et compréhension du code

Pas de panique ! Il est normal qu'à cette étape le code ne compile pas.

1. Avant tout configurez votre Git pour avoir votre nom dans les commits
2. Sur la branche master, le code qui vous a été donné est séparé en de multiples fichiers. Décrire le rôle de chacun d'eux et de quelle manière ils seront référencés dans les binaires finaux, un schéma est suffisant.

3. Réalisez une brève documentation du header `ppm.h`. Commit de documentation dans le code accepté ou bien sur vos notes. Notez que `./test` compile et crée un fichier `test.ppm`. Cela peut vous aider à démêler les appels ...

## **Exercice 2 : réparer le code et se l'approprier**

1. Corrigez le code de `mandel.c` et le `Makefile` pour qu'il compile et faites un commit (directement sur master) de cette correction.
2. Pourriez-vous rapidement expliquer les étapes du code en citant les blocs de lignes (il ne vous est pas demandé de mathématiques) si vous n'êtes pas à l'aise parlez des boucles, des entrées et des sorties.
  1. Par exemple lignes 29–30 : création d'un image PPM.
  2. ...

## **Exercice 3 : un peu de couleur**

Kevin a une version du code en couleur dans sa branche `color_support`, cependant, vous avez modifié la master.

1. Comment pourriez-vous ramener cette branche au-dessus de la master ?
2. Faites cette opération, est-il nécessaire de faire une étape manuelle, si oui laquelle ? Détaillez votre gestion de cette opération.
3. Ramenez la couleur au-dessus sur la master !

## **Exercice 4 : sauver cela ailleurs**

Vous avez donc cloné le code de Kevin et maintenant vous voulez mettre cela dans un autre dépôt à vous, afin de le rendre au professeur qui va attentivement décortiquer les commits !

1. Quel type de dépôt choisissez-vous ? Expliquer le processus de création (si nécessaire)
2. Quelles commandes utilisez-vous pour envoyer le travail ? (toutes les branches si possible)

## **Exercice 5 : faites vous plaisir**

La base de Kevin vous a peut être aidé à avoir un Mandelbrot fonctionnel ; avez-vous des idées pour aller plus loin ? (Zoom, parallélisme, couleurs, ...) Surprenez le professeur qui sera ravi de vos exploits !