



Et inviter ses collaborateurs

03 Les Merges (Aled)

(Non je rigole en fait c'est simple)

•



•

04 Le .Gitignore



•

•

ҮОЦРІ!

•

•

•



<u>une (seule)</u> personne doit créer le repo sur son compte github

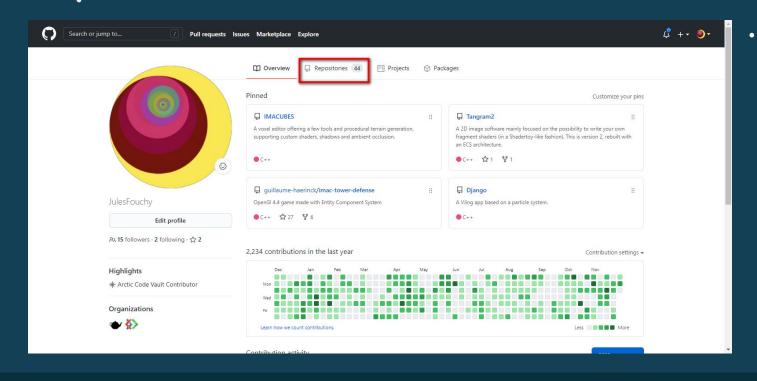
OPTION 1 : TU CRÉES LE REPO

Et invite ton pote

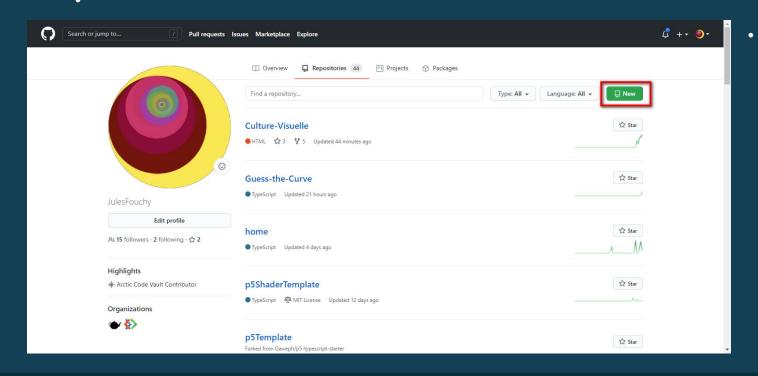


Et il t'invite

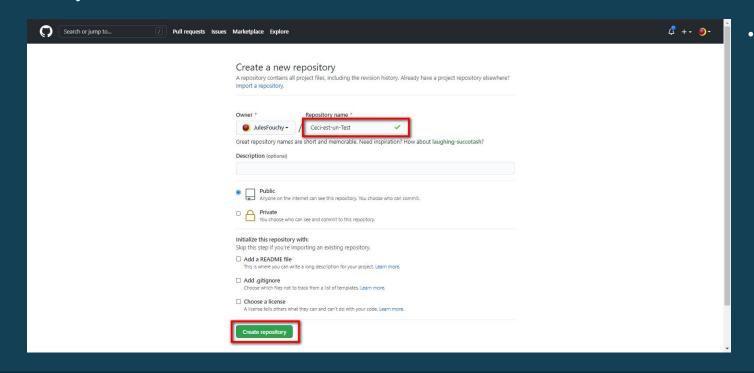
OPT**t**on 1 : Tu crées le repo



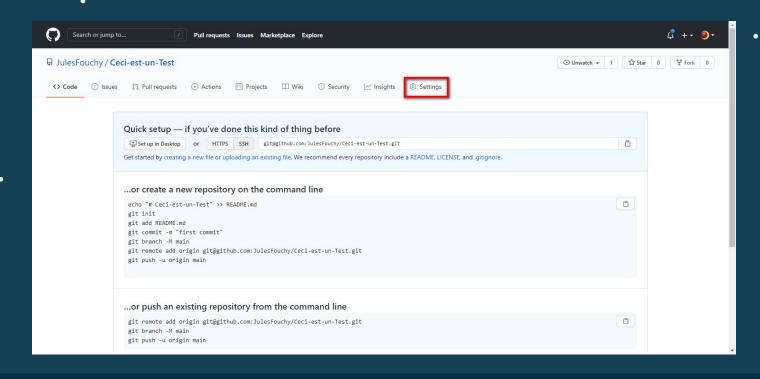
OPT**f**on 1 : Tu crées le repo



OPT**f**on 1 : Tu crées le repo

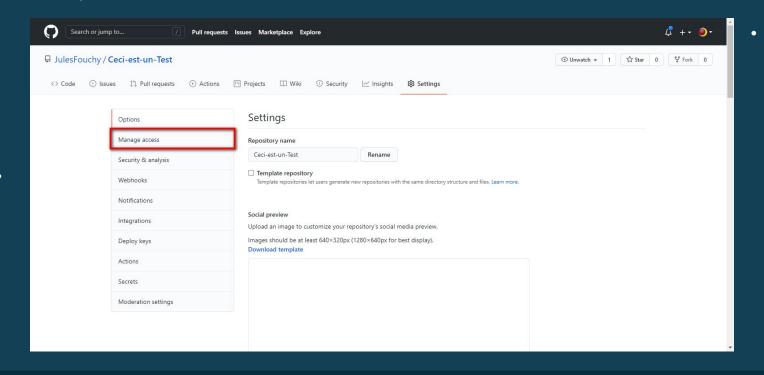


OPT**t**on 1 : Tu crées le repo

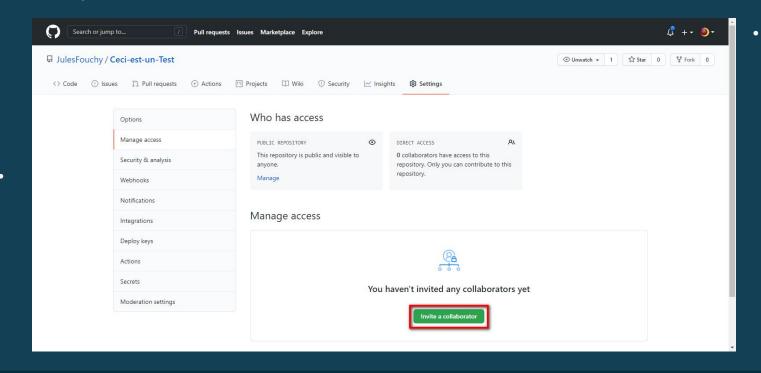


MAINTENANT QUE LE REPO EST CRÉÉ, PLUS QU'À INVITER DESSUS TOUS LES MEMBRES DU PROJET

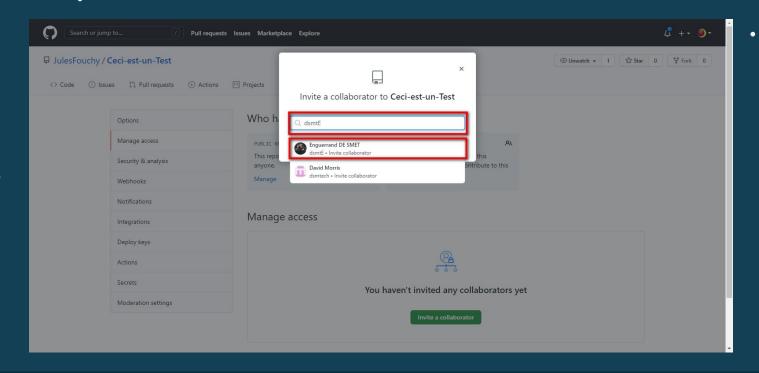
OPT**f**on 1 : Tu crées le repo



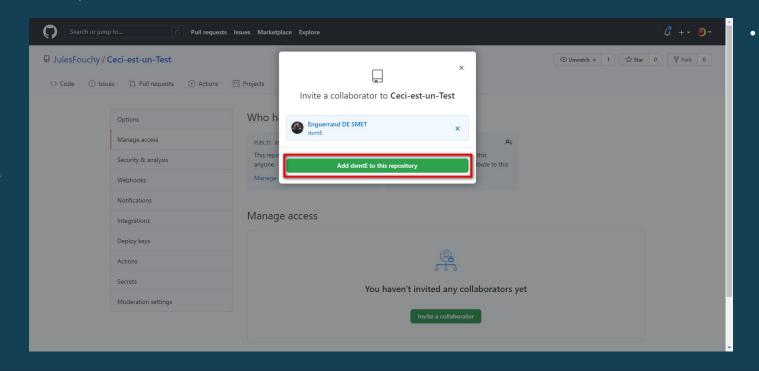
OPT**t**on 1 : Tu crées le repo



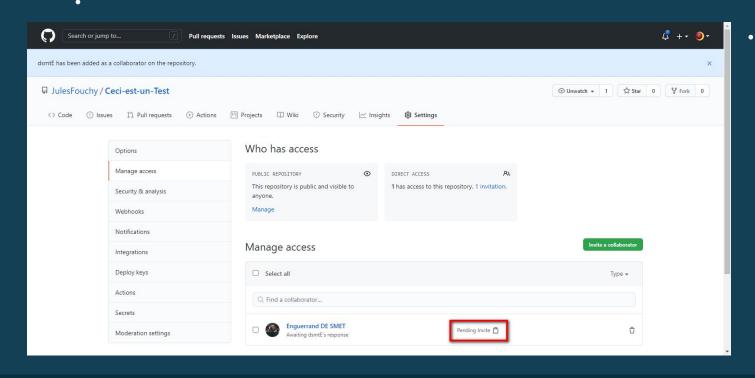
OPT**t**on 1 : Tu crées le repo



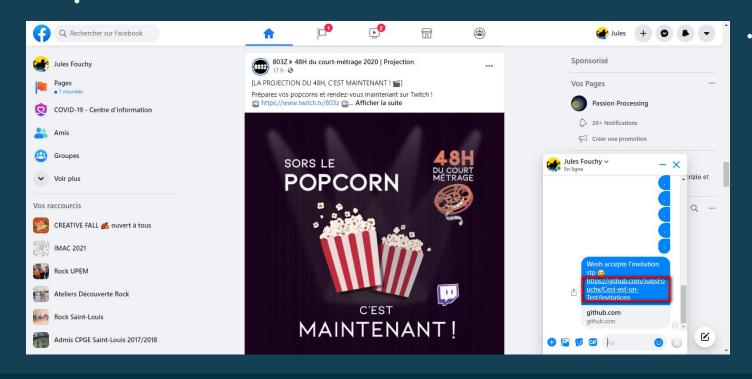
OPTTON 1 : TU CRÉES LE REPO



OPT**t**on 1 : Tu crées le repo



OPTION 2 : TU ATTENDS QUE TON POTE CRÉE LE REPO



IL VA T'ENVOYER LE LIEN D'INVITATION DE LA SLIDE PRÉCÉDENTE, TU N'AS QU'À CLIQUER ET Accepter

GitHub







@JulesFouchy has invited you to collaborate on the

JulesFouchy/Ceci-est-un-Test repository

You can accept or decline this invitation. You can also head over to https://github.com/ JulesFouchy/Ceci-est-un-Test to check out the repository or visit @JulesFouchy to learn a bit more about them.

This invitation will expire in 7 days.

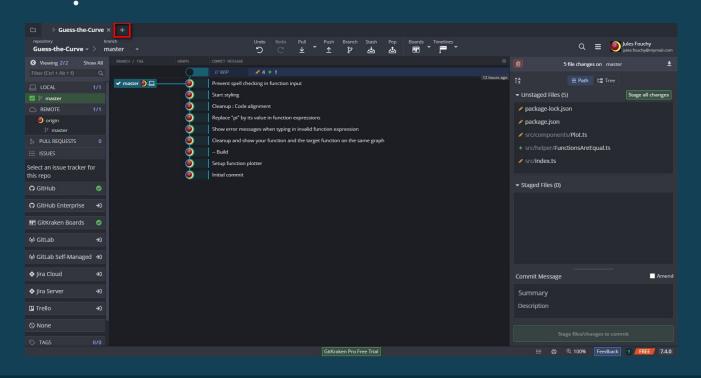
View invitation

MAINTENANT QUE TOUT LE MONDE EST SUR LE REPO, PLUS QU'À LE CLONER SUR Votre ordi pour pouvoir commencer à travailler!

C'EST LE MOMENT OÙ ON LANCE GITKRAKEN



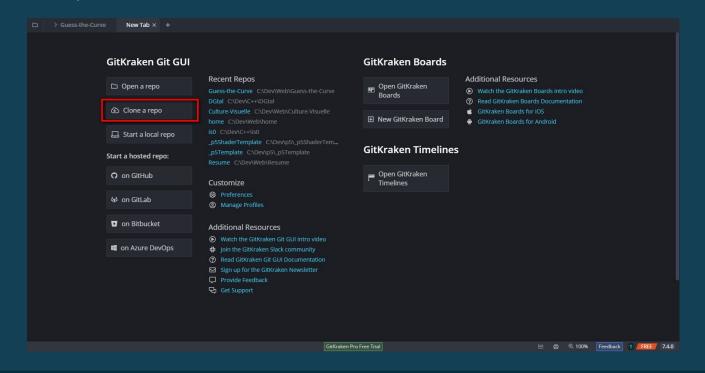
CLON**è**r le r<u>epo</u>'



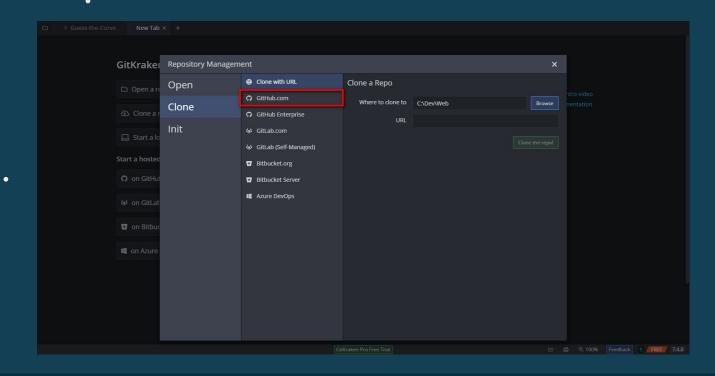
•

(NB : SI VOUS N'AVEZ PAS ENCORE DE REPO OUVERT, PAS BESOIN DE CETTE ÉTAPE QUI OUVRE UN Nouvel onglet)

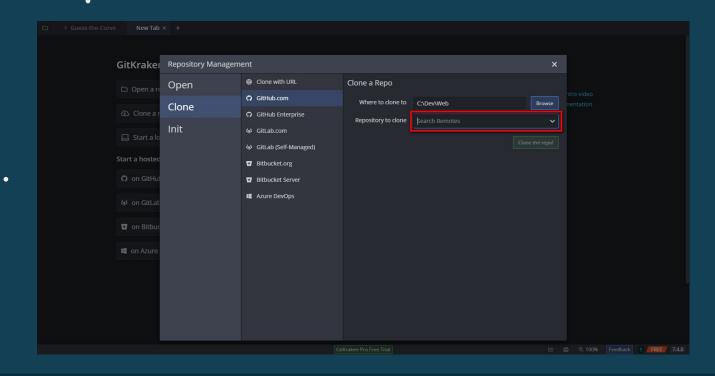
•



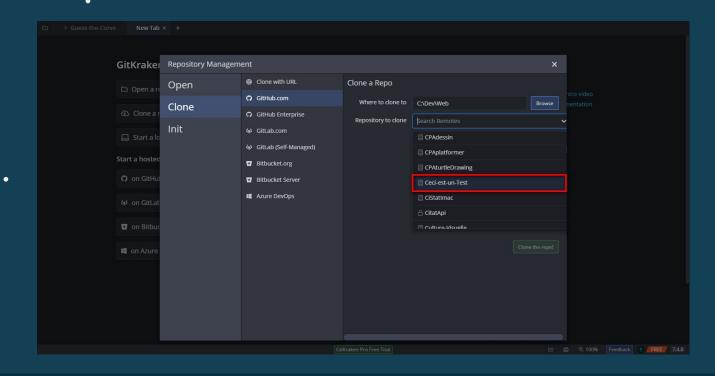
•



•

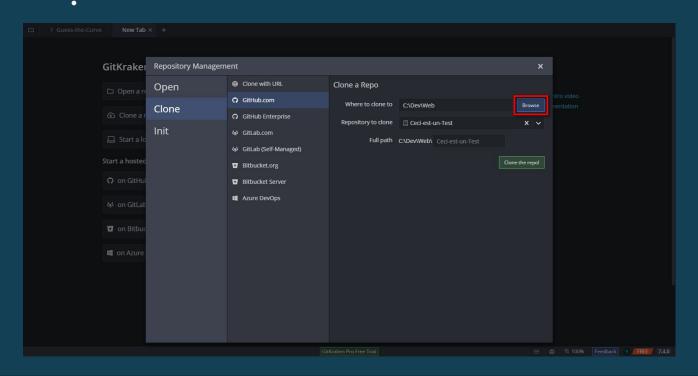


•



•

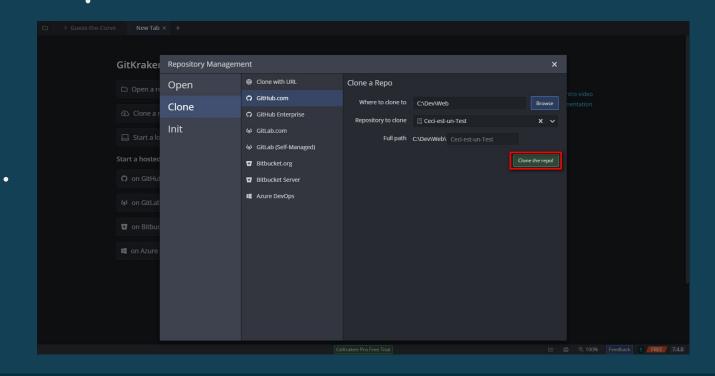
•



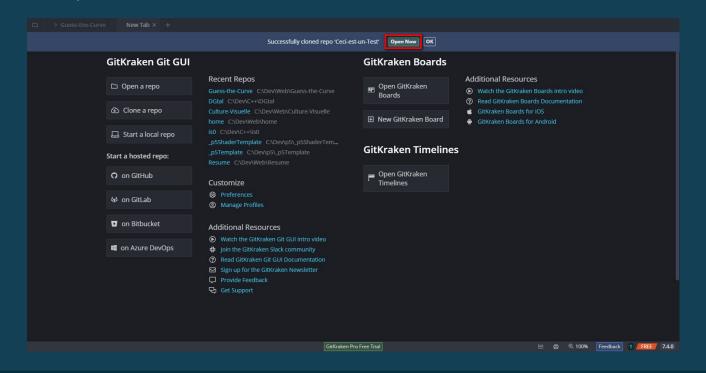
•

CHOISISSEZ DANS QUEL DOSSIER ENREGISTRER LE REPO

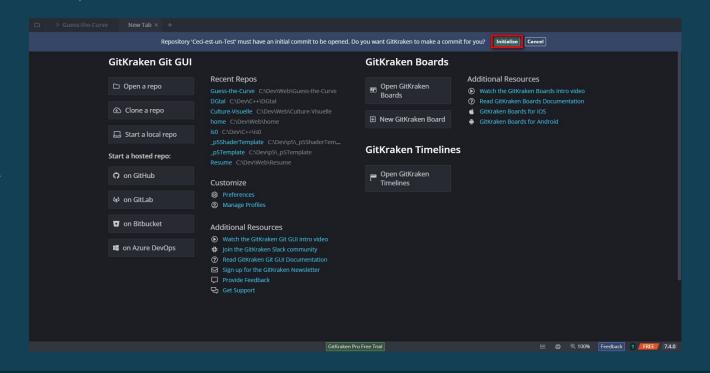
•



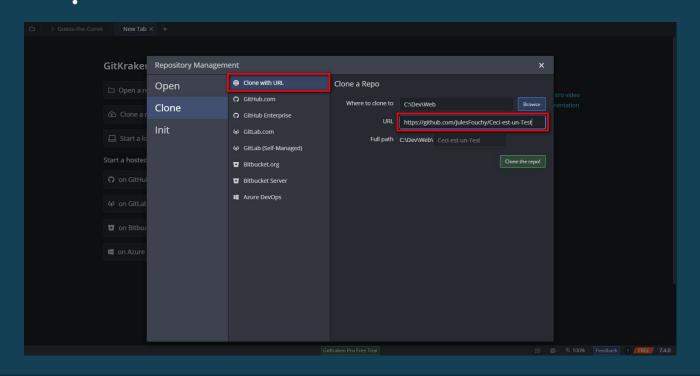
•



•



•

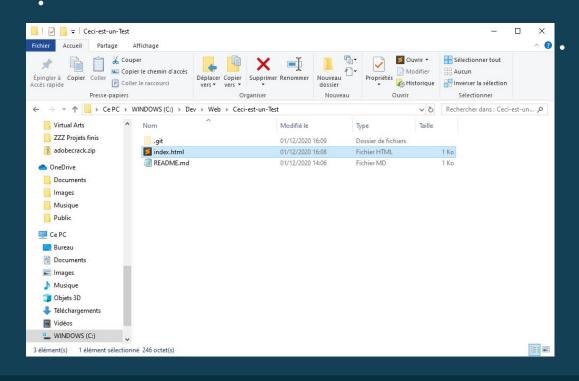


•

NB : ON PEUT AUSSI CLONER LE REPO AVEC L'URL, OU DE 1000 AUTRES MANIÈRES

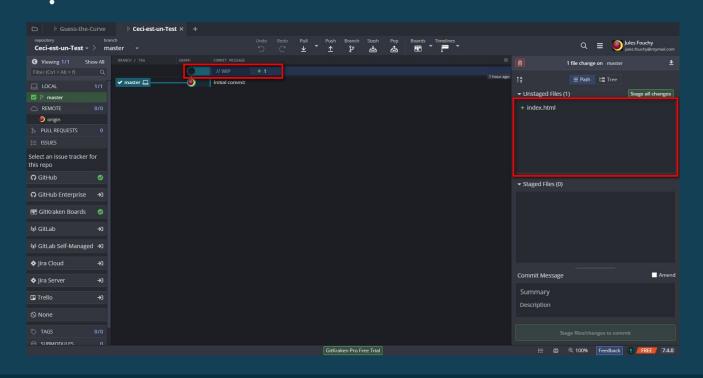


1 - ÉCRIRE DU CODE



DANS VOTRE DOSSIER DU REPO FRAÎCHEMENT CLONÉ, VOUS POUVEZ COMMENCER À AJOUTER TOUT Le code que vous voulez

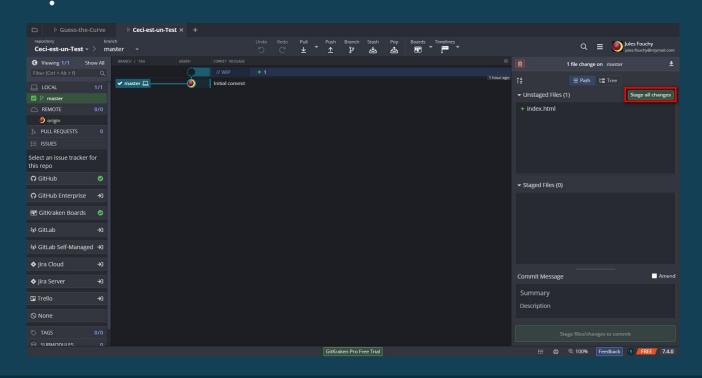
2 - VOIR LES CHANGEMENTS DANS GITKRAKEN



•

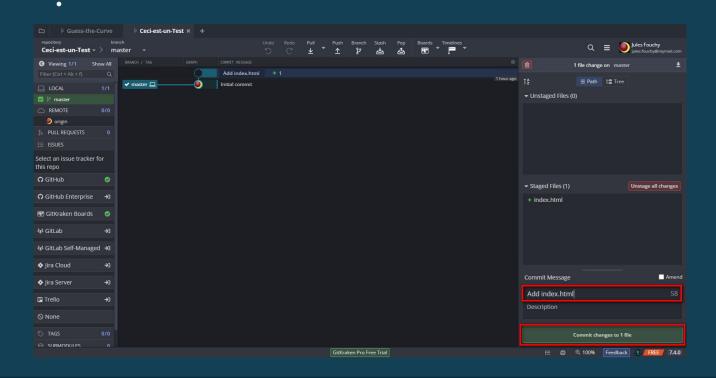
DÈS QUE VOUS MODIFIEZ, CRÉEZ OU SUPPRIMER UN FICHIER, TOUS LES CHANGEMENTS Apparaissent ICI. Vous pouvez les visualiser en cliquant sur chaque fichier.

<u>3 - Choisir les changements à inclure dans le commit °</u>



•

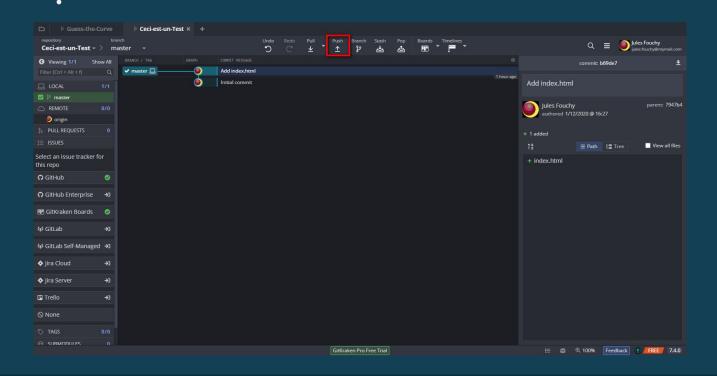
LA PLUPART DU TEMPS VOOS ALLEZ TOUT INCLURE, MAIS VOUS AVEZ AUSSI LA POSSIBILITÉ DE Choisir fichier par fichier, voire changement par changement à l'intérieur d'un fichier *



•

DONNEZ UN TITRE À VOTRE COMMIT, ET C'EST PARTI!

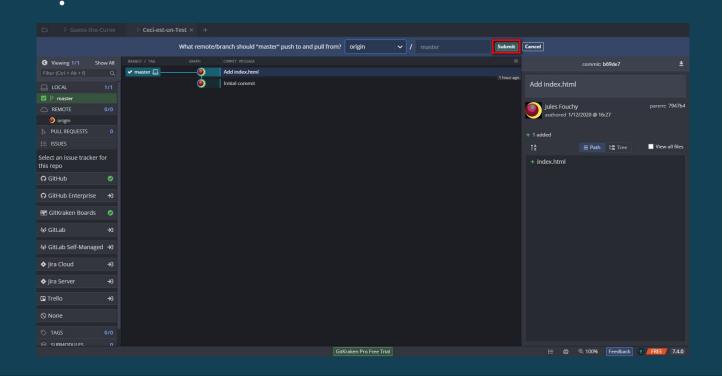
5 - PUSH



•

POUR L'INSTANT LE COMMIT EST SAUVEGARDÉ DANS L'HISTORIQUE DE VOTRE REPO LOCAL, MAIS Pour qu'il soit mis en ligne sur le repo github, il paut push

5 - PUSH



•

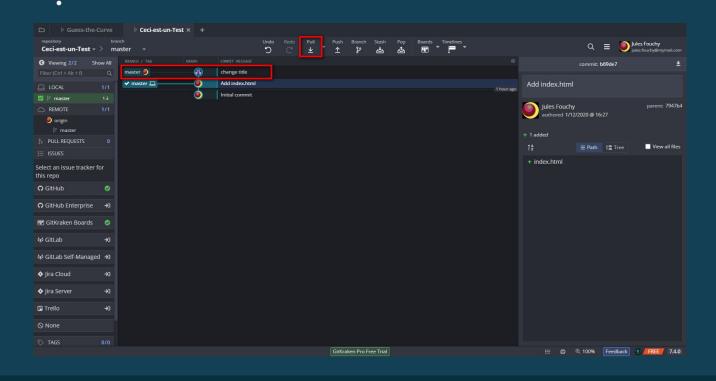
IL SE POURRAIT QU'IL VOUS POSE LA QUESTION LORS DE VOTRE PREMIER COMMIT. Faites juste submit



FÉLICITATIONS! VOS CHANGEMENTS SONT EN LIGNE; LES AUTRES MEMBRES PEUVENT DÉSORMAIS RÉCUPÉRER VOTRE CODE ET VOUS POUVEZ COMMENCER À COLLABORER

6 - PULL

•



•

LORSQUE LES AUTRES ONT COMMIT ET PUSH DES CHANGEMENTS, VOUS LES VERREZ APPARAÎTRE ICI. Pour les récupérer, il vous suffit de faire un pulc



•

•

•

•

•

•

•

•

SI VOUS ÊTES PLUSIEURS À TRAVAILLER SUR UN MÊME FICHIER CHACUN DE Votre côté, au moment où vous allez faire vos commits, il va.falloir Fusionner (merge) vos différents changements.



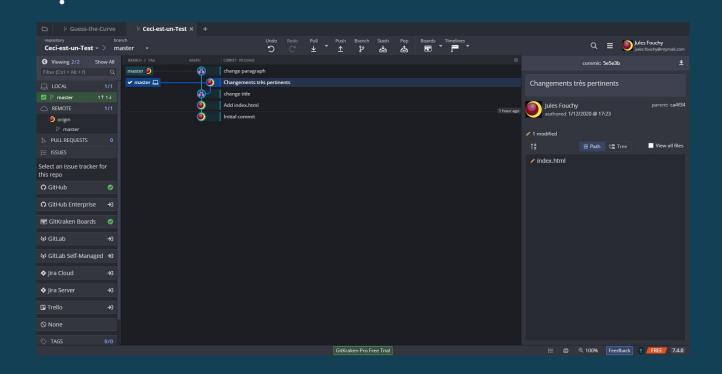
En général, c'est assez simple à résoudre (voire ça se fait automatiquement).

(Mais ceci dit, si vous pouvez vous organiser pour travailler sur des parties du code séparées et ne pas vous marcher sur les pieds, c'est mieux).



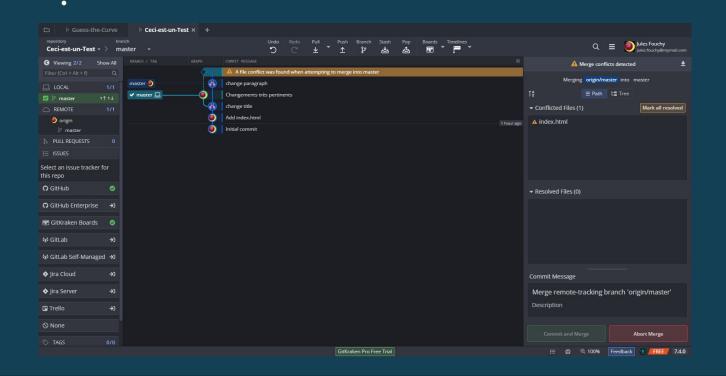
•

•



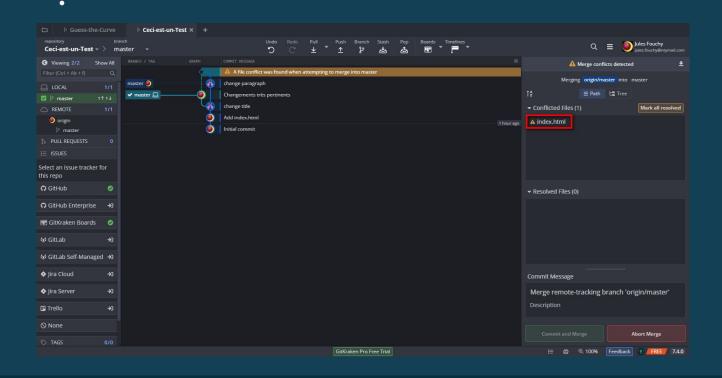
•

MERGE EN APPROCHE!



•

VOILÀ CE QUI SE PASSE UNE FOIS QUE JE PULL

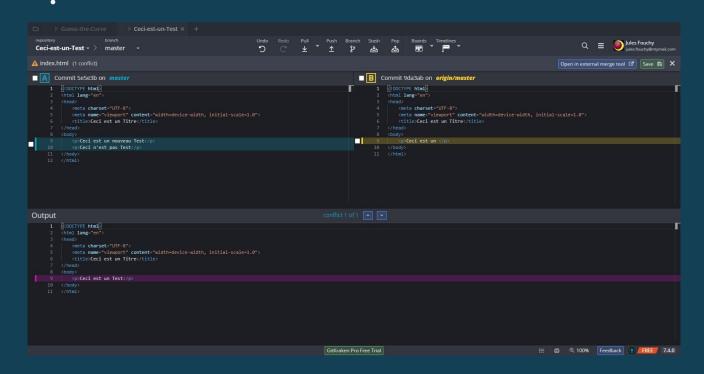


•

POUR CHAQUE FICHIER EN CONFLIT, VOUS POUVEZ VOIR LES DIFFÉRENCES ET DÉCIDER QUOI GARDER

MERGE - LA FENÊTRE DE DIFFS

•

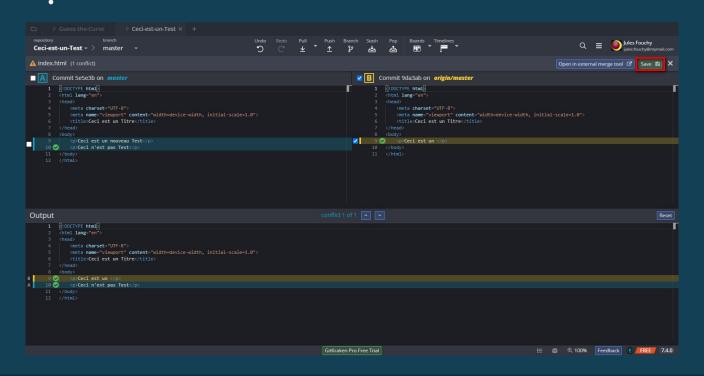


•

À GAUCHE : VOS CHANGEMENTS. À DROITE : LES AUTRES CHANGEMENTS En bas : le fichier fusionné que vous êtes en train de créer

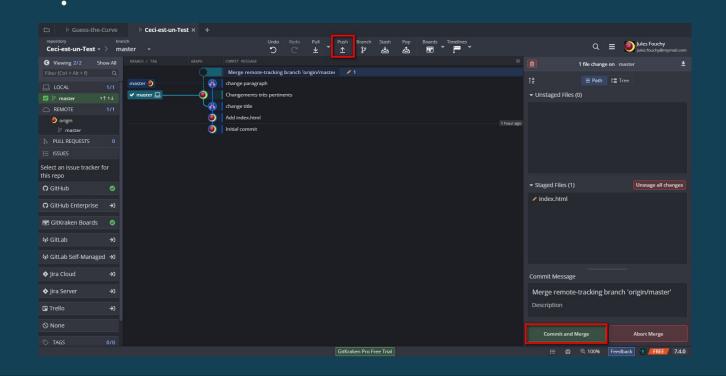
MERGE - LA FENÊTRE DE DIFFS

•



•

VOUS POUVEZ SÉLECTIONNER, LIGNE PAR LIGNE OU BLOC PAR BLOC, DEPUIS QUEL FICHIER PRENDRE Les Changements. Une fois le merge fini, plus qu'à sauvegarder.



•

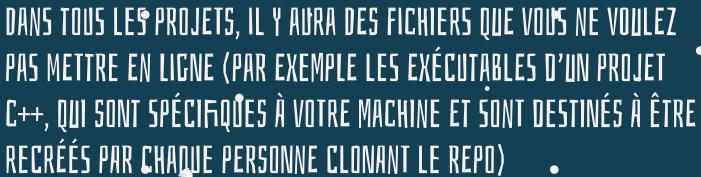
UNE FOIS LE MERGE FINI, PLUS QU'À "COMMIT AND MERGE" PUIS PUSH • /!\ règle d'or : avant de commit et push un merge, testez le ! assurez vous que le code marche comme prévu ! •



•

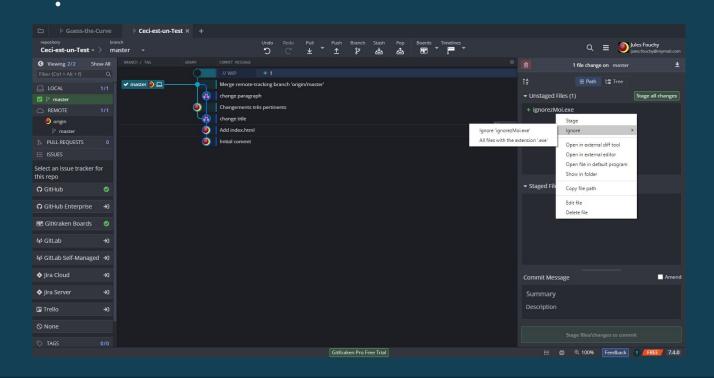
•

•





.GITIGNORE



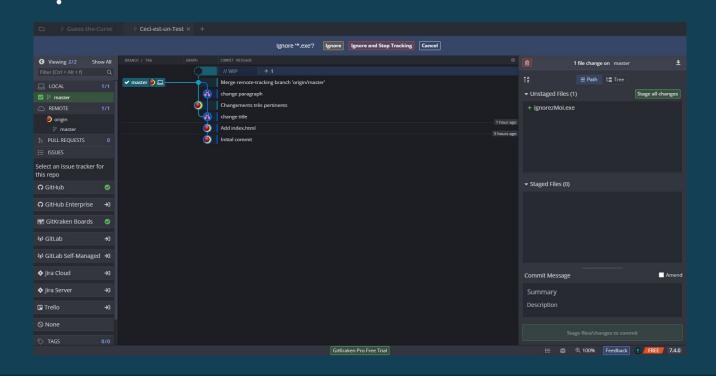
•

CITKRAKEN PEUT S'OCCUPER DU .GITIGNORE POUR VOUS AVEC UN CLIC DROIT SOR LE FICHIER EN QUESTION

.GITIGNORE

•

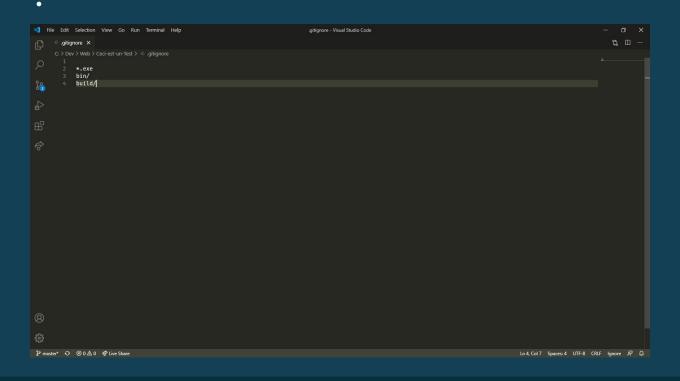
•



•

IGNORE : LE FICHIER EST AJOUTÉ À VOTRE GITIGNORE Stop tracking : le fichier est supprimé du repo github si vous l'aviez précédemment commit par erreur

.GITIGNORE



VOUS POUVEZ AUSSI ÉDITER LE FICHIER À LA MAIN DANS N'IMPORTE QUEL ÉDITEUR DE TEXTE (Car gitkraken ne propose pas toujours l'option d'Ignore que vous voulez)



•

RÈGL**è**s d'or

- Dès qu'on a fini quelque chose, avant de passer à la suite, on commit ce qu'on vient de faire! Plus vous faites de petits commits réguliers, mieux c'est!
- On ne commit que quand on a testé et vérifié que ça marche
- On relit toujours ses changements avant de commit!

ET VOILÀ! VOUS AVEZ.VU L'ESSENTIEL DONT Vous aurez besoin pour tous vos projets de prog





THANKS!



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**