

Rapport de soutenance 2

File Explorer

Lundi, 8 Avril 2024

Arthur Garraud

Tristan Faure

Iliane Formet

GroupAléa2

Sommaire

1	Introduction	3
2	Présentation du groupe	4
2.1	Arthur Garraud	4
2.2	Iliane Formet	4
2.3	Tristan Faure	5
3	Avancement du projet	6
3.1	Avancement à la dernière soutenance	6
3.2	Avancement prévu	6
3.3	Rappel de la répartition par membre	7
3.4	Commandes	8
3.5	Algorithmes	10
3.6	UI	12
3.7	Site internet	15
4	Conclusion	16
4.1	Conclusion	16
4.2	Pour la prochaine soutenance	16

1 Introduction

Il existe une multitude d'explorateurs de fichiers, notamment ceux du système d'exploitation de votre ordinateur comme F'inder pour MacOS ou File Explorer pour Windows (pour ne citer qu'eux).

Cependant, nous avons remarqué que l'algorithme de recherche de ceux-ci peut s'avérer être extrêmement lent et que le design pouvait être amélioré pour une meilleure expérience utilisateur.

Nous avons donc décidé de créer notre propre explorateur de fichiers, avec une interface plus agréable et de meilleurs algorithmes.

Cet explorateur de fichiers utilisera une interface graphique.

Il nous permettra d'apprendre à maîtriser Rust, à comprendre et implémenter la manipulation de fichiers et dossiers et de savoir concevoir une interface utilisateur simple et efficace.

Notre objectif est de réussir à créer un explorateur de fichiers qui sera non seulement plus esthétique (design simple sans superflu), mais surtout plus performant.

Cet explorateur sera réalisé en Rust et disponible uniquement sur Linux.

2 Présentation du groupe

2.1 Arthur Garraud

Passionné par l'informatique depuis des années, j'ai toujours été curieux et autodidacte dans de nombreux domaines. J'ai réalisé de nombreux projets qui m'ont permis d'apprendre à maîtriser des outils puissants, que ce soit des langages de programmation ou bien des applications de 3D ou de graphisme.

J'apprécie également le cinéma et j'ai l'habitude de me passionner pour un sujet précis pendant quelques semaines.

Maintenant j'ai décidé de finir ingénieur en informatique afin de pouvoir avoir la possibilité de travailler sur des projets et des missions très différentes de l'une à l'autre.

2.2 Iliane Formet

Depuis mon plus jeune âge, j'ai toujours été passionné par la technologie, la logique et les casse-tête. Ingénieur était donc ce à quoi je me destinais.

Lors d'un projet de labyrinthe pour robot en terminale, j'ai découvert les algorithmes et la programmation. Je me suis alors intéressé et ai commencé à toucher un peu à tout ce qui touche à ce domaine (jeux vidéo, sites internet, algorithmes de calcul). J'ai alors décidé de me tourner vers EPITA pour mes études. Je souhaite dans le futur utiliser mes compétences acquises en école à travers les cours, les conférences et les projets pour entreprendre.

Mes autres loisirs sont le sport, les jeux vidéo, la musique, le cinéma, la nourriture et les sorties entre amis.

Avec ce projet, j'espère perfectionner mes fondamentaux en Rust, mais aussi mieux comprendre les relations entre interface et algorithmes.

2.3 Tristan Faure

Plongé dans l'informatique depuis mes 10 ans par mon oncle. Toujours passionné par de nouveaux langages informatiques, j'ai longtemps cherché le domaine qui me passionne le plus en informatique.

J'ai premièrement commencé en autodidacte à apprendre le codage web, commençant par des sites avec HTML, CSS et JavaScript puis je me suis vite tourné vers la programmation de jeux vidéo en faisant de nombreux jeux sur navigateur, certains plutôt complet, comportait plusieurs mondes à la manière d'un Mario.

Je suis également passé par la programmation en python à l'école et seul afin de créer des bots discords et en LUA lorsque j'ai créé mon serveur GTA 5 RP en début d'année 2020. Toutes ces expériences et les longues heures de conférence et de tuto que j'ai regardé m'ont finalement orienté vers la data science et l'Intelligence Artificiel. J'ai donc décidé de rejoindre Epita pour atteindre mes objectifs. Je pense que ce choix m'aidera à progresser rapidement et efficacement.

3 Avancement du projet

3.1 Avancement à la dernière soutenance

Voici le tableau de l'avancement des tâches au moment de la première soutenance :

Tâche	Avancement
UI	Prise en main de l'outil de création d'interface et première interface basique
Commandes bash	Implémentation des commandes bash de base
Algorithmes	Création d'un algorithme de recherche non optimisé

3.2 Avancement prévu

Voici le tableau de l'avancement des tâches prévu pour la seconde soutenance :

Tâche	Avancement
UI	Création de l'interface finale
Commandes bash	Connexion des commandes à l'interface
Algorithmes	Optimisation de l'algorithme de recherche et fonction de compression/decompression
Site Internet	Mise en place du site

3.3 Rappel de la répartition par membre

Resp = Responsable et Supp = Suppléant

Tâche / Membre	Arthur	Iliane	Tristan
Gestion du projet	-	Resp	-
UI	-	Supp	Resp
Commandes bash	Supp	Resp	-
Algorithmes	Resp	Supp	-
Site internet	Supp	-	Resp

3.4 Commandes

L'objectif pour cette soutenance était de finir l'implémentation des commandes bash restantes ainsi que la mise en relation des algorithmes avec l'interface.

Pour ce qui est des commandes bash, on a implémenter :

- File Copy : Copie d'un fichier
- Directory Copy : Copie d'un répertoire

Pour la copie de fichier, on a utilisé les modules `std::fs` et `std::io` comme pour les précédentes fonctions.

Pour ce qui est de la copie de répertoire, on a dû utiliser le module `fs_extra::dir::copy`. Le module `fs_extra` permet des opérations supplémentaires que celles offertes par `std::fs`.

Il a également fallu corriger certaines fonctions qui buggait.

Pour ce qui est de la mise en relation, la première chose implémentée est l'affichage du contenu du répertoire de travail actuel.

On a ensuite implémenté la navigation dans les répertoires. On a également ajouté la possibilité de revenir en arrière dans le chemin.

L'ajout de la création d'un nouveau dossier a été ajoutée, tout comme la création ou la suppression d'un fichier ou un dossier.

Les principaux problèmes pour cette partie du projet étaient comme lors de la première soutenance la gestion des potentielles erreurs ainsi que la gestion des paramètres.

En effet, étant donné que certaines boucles tournent en continu, on se retrouve avec énormément de valeurs empruntées après changement.

Pour ce faire, il a fallu définir chaque paramètre global en utilisant la librairie RefCell. Cette librairie permet d'avoir plusieurs emprunts de données mutables ou immutables à n'importe quel moment. Par exemple, le chemin du répertoire de travail actuel doit être accessible et modifiable à n'importe quel moment et dans n'importe quelle boucle, il a donc fallut le définir en utilisant RefCell.

De plus, à chaque utilisation, il faut cloner chaque paramètres pour éviter tout conflit.

Ce travail ainsi que la gestion de tous les bugs a constitué la plus grosse partie du travail.

Pour résumer, les fonctions qui avaient été implémentées lors de la première soutenance sont maintenant fonctionnelles via l'interface, de ce fait les objectifs sont accomplis.

Pour la soutenance finale, il faudra fixer la mise à jour de l'affichage du répertoire de travail après chaque action, implémenter les nouvelles commandes ainsi que rajouter de potentielles fonctionnalités.

3.5 Algorithmes

Depuis la dernière soutenance, on a grandement avancé sur la recherche parmi les noms des fichiers, en implémentant notamment un algorithme qui permet de récupérer bien plus rapidement que la dernière fois l'ensemble des noms de fichiers et leur chemin absolu à partir d'un dossier racine. De plus, ce nouvel algorithme, utilisant le package `libc` sous linux, permet d'éviter tout problème de permissions et donc d'indexer réellement l'ensemble des fichiers de l'ordinateur. Afin d'optimiser cet algorithme d'indexage l'on a fait en sorte de ne plus compter et suivre les liens symboliques qui grandissaient l'ensemble des noms de fichiers sans réel intérêt.

En utilisant ce nouvel ensemble plus fourni et plus qualitatif, l'on utilise toujours un algorithme de recherche binaire qui donne des résultats à une vitesse bien plus que satisfaisante, mais l'on a rajouté la possibilité de faire une recherche non sensible à la casse et également de trouver des fichiers à l'aide de leur préfix. Cela permet d'avoir un bien plus grand nombre de résultats, puisque avant il fallait obligatoirement rechercher le nom exact de ce que l'on recherchait.

Il a fallu implémenter cette fonction de recherche dans l'interface utilisateur et ce n'était pas une tâche facile due à l'asynchronisme des différentes tâches à effectuer. Cependant, en utilisant des références et en gérant correctement le système d'emprunts des variables, la fonction de recherche est désormais correctement implémentée dans l'interface.

Une fonction de compression/décompression de dossiers a également été implémentée et permet d'obtenir un fichier zippé d'un dossier et son contenu. Il est également possible d'extraire le contenu de ce fichier zippé. Ces 2 fonctions ont également été rajoutées dans l'UI.

D'ici la prochaine soutenance, il faudra ajouter des fonctionnalités supplémentaires, comme la possibilité de stocker des fichiers en ligne directement en passant par notre explorateur de fichiers, ou bien d'implémenter un moyen de compression pour les fichiers en s'adaptant en fonction du type de fichier (texte/vidéo/image).

3.6 UI

Dans la première soutenance, la section dédiée à l'interface utilisateur (UI) du projet d'explorateur de fichiers développé en Rust et utilisant GTK4 a mis en avant le développement de plusieurs composants clés pour améliorer l'expérience utilisateur :

1. Barre de recherche dynamique : Une barre de recherche élégante a été développée, qui apparaît lorsque l'utilisateur clique sur une icône de loupe située dans l'en-tête de l'application. Cette barre permet une saisie en temps réel et affiche instantanément les résultats correspondants, offrant une expérience de recherche réactive et efficace.

2. Affichage de l'heure et de la date : Pour fournir une référence temporelle précise aux utilisateurs, un affichage de l'heure et de la date a été intégré dans l'en-tête de l'application. Cette fonctionnalité s'actualise automatiquement pour refléter l'heure et la date actuelles, ajoutant ainsi une touche pratique et essentielle à l'application.

3. Column View Datagrid : Un élément visuel avancé, le Column View Datagrid, a été implémenté pour offrir une vue détaillée des données. Il se compose de deux colonnes contenant chacune 10 000 éléments, permettant une navigation rapide grâce à une barre de défilement intégrée. Ce Datagrid offre un aperçu de la capacité de l'application à présenter de grandes quantités de données de manière organisée et accessible.

Ces éléments de l’UI ont été conçus pour démontrer les possibilités offertes par GTK4 et Rust en termes de développement d’interfaces utilisateur robustes et modernes. Ils servent d’exemples préliminaires des fonctionnalités et de l’esthétique visée pour l’application finale, en mettant l’accent sur la simplicité, l’efficacité et l’engagement utilisateur.

Pour la deuxième soutenance, des avancées significatives ont été réalisées sur l’interface utilisateur (UI) du projet d’explorateur de fichiers, en plus des fonctionnalités précédemment développées :

1. Bouton d’ouverture des crédits et du lien du site internet : Un nouveau bouton a été créé, lequel, lorsqu’il est cliqué par l’utilisateur, ouvre une nouvelle fenêtre affichant les crédits de l’application ainsi que le lien vers le site internet du projet. Cette fenêtre représente un ajout important pour renforcer la transparence du projet et offrir un accès facile à des informations supplémentaires et ressources en ligne.

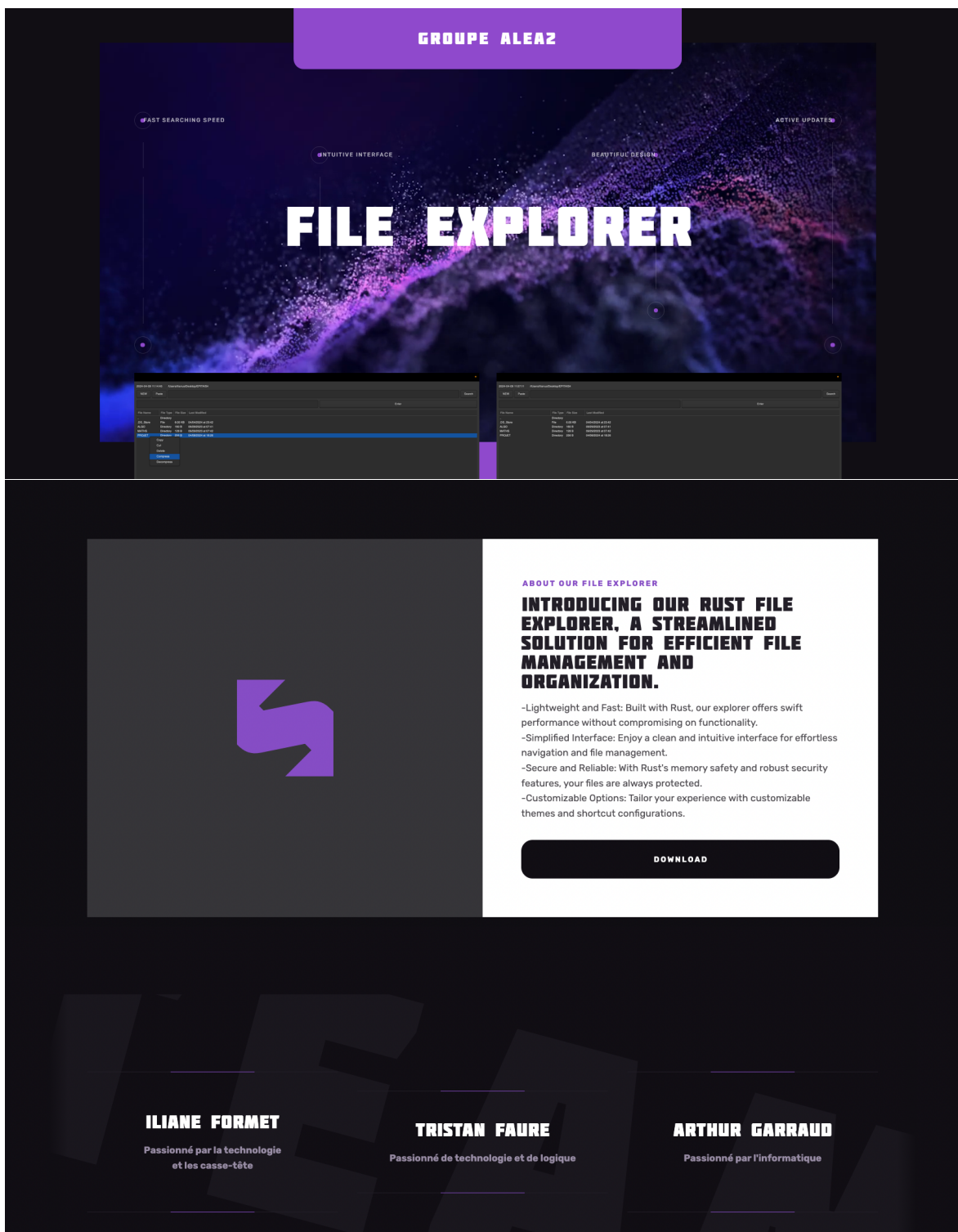
2. Bouton ouvrant une fenêtre avec un menu déroulant et des zones de texte : Un autre bouton a été intégré, permettant d’ouvrir une fenêtre contenant un menu déroulant ainsi que des zones de texte. Cette fonctionnalité enrichit l’interaction de l’utilisateur avec l’application, en lui permettant de réaliser des sélections précises et de saisir des informations de manière structurée.

Pour la troisième soutenance, l'objectif sera de connecter ces éléments nouvellement développés avec le reste de l'interface utilisateur pour obtenir une interface finale esthétique et fonctionnelle. Ce processus inclura l'intégration des divers composants de l'UI de manière cohérente, assurant une expérience utilisateur fluide et intuitive. De plus, il est prévu d'ajouter de nombreuses options supplémentaires utiles à l'utilisateur, visant à accroître la praticité et l'efficacité de l'explorateur de fichiers. Ces améliorations sont cruciales pour réaliser une application qui non seulement répond aux besoins fonctionnels de ses utilisateurs mais qui est également agréable à utiliser au quotidien.

3.7 Site internet

Le site internet a été mis en place.

Il sera mis à jour d'ici la prochaine soutenance.



4 Conclusion

4.1 Conclusion

Nous avons donc bien respecter les objectifs que nous nous étions fixés :

- L'interface est quasiment finie
- Les commandes sont fonctionnelles et les nouvelles implémentées
- Nous avons un algorithme de recherche efficace ainsi qu'un algorithme de compression/decompression
- Le site a été mis en place

4.2 Pour la prochaine soutenance

Voici le tableau de l'avancement des tâches prévu pour la prochaine soutenance :

Tâche	Avancement
UI	Fin de la création de l'interface finale
Commandes bash	Mise en place des algorithmes sur l'interface finale et implémentations de nouvelles fonctionnalités
Algorithmes	Amélioration des algorithmes et ajouts d'autres fonctionnalités
Site Internet	Mise à jour et amélioration du site