File System Analyzer and Explorator (FSAE)



Auteur : Müller Pierrick et Spinelli Isaia

Prof : Bruegger Pascal

Assit : Zurbuchen Nicolas

Date : 03.01.2021

Salle : A4 – Lausanne

Classe : T-MobOp

Table des matières

[Introduction - 2 -](#_Toc58497034)

[But - 2 -](#_Toc58497035)

[Motivations - 2 -](#_Toc58497036)

[Analyse - 2 -](#_Toc58497037)

[Conception - 3 -](#_Toc58497038)

[Graphique - 3 -](#_Toc58497039)

[Mode explorateur - 3 -](#_Toc58497040)

[Mode analyseur - 3 -](#_Toc58497041)

[Classes - 4 -](#_Toc58497042)

[Implémentation - 5 -](#_Toc58497043)

[Problèmes rencontrées - 5 -](#_Toc58497044)

[Utilisation - 6 -](#_Toc58497045)

[Conclusion - 7 -](#_Toc58497046)

[Qui a fait quoi - 7 -](#_Toc58497047)

[Futur - 7 -](#_Toc58497048)

[Amélioration - 7 -](#_Toc58497049)

[Compétences acquises - 7 -](#_Toc58497050)

[Résultats obtenus - 7 -](#_Toc58497051)

[Sources - 7 -](#_Toc58497052)

[Annexes - 7 -](#_Toc58497053)

# Introduction

## But

Dans le cadre du cours T-MobOp (Systèmes d'exploitation mobiles et applications) nous avons dû proposer et réaliser un mini-projet personnel afin de se familiariser avec l’environnement de développement et les concepts d’Android.

## Motivations

Pour ce mini-projet, nous avons décidé de réaliser un explorateur de fichier mais par-dessus tout, un analyseur du système de fichier. Brièvement, l’explorateur de fichier permet de se déplacer dans toute l’arborescence du système et l’analyseur permet de fournir une bonne visualisions de la place mémoire prise par les différents répertoires.

Ce choix a été pris pour des raisons d’envie personnel. Certes, il existe déjà plusieurs explorateurs de fichier, cependant, c’est la partie analyse qui nous intéresse le plus. En effet, l’objectif principal est de permettre à tout le monde de prendre conscience, de manière agréable, évidente et compréhensible, la répartition et l’utilisation de la place mémoire.

# Analyse

Pour la partie explorateurs de fichier, voici les fonctionnalités qui devront être implémentées :

1. Déplacement dans la hiérarchie inférieure du système de fichier de répertoire en répertoire.
2. Enregistrement et affichage du chemin actuel depuis la racine.
3. Déplacement instantané à un répertoire dans la hiérarchie supérieure.
4. Distinction claire entre un répertoire et un fichier.
5. Affichage de la place mémoire de chaque item (répertoire ou fichier).
6. Affichage du nombre de fichier pour chaque répertoire.
7. Créer un nouveau dossier
8. Créer un nouveau fichier
9. Supprimer un dossier
10. Supprimer un fichier
11. Trier la liste des items (répertoire/fichier)

Il sera en tout temps possible de passer en mode « analyseur » afin de visuellement de manière ergonomique la place mémoire de chaque item ainsi que la répartition. De ce fait, plusieurs points devront être mis en place :

1. Possibilité de passer en mode analyseur facilement.
2. Visualisation aisée de la répartition de la place mémoire.
3. Affichage en pourcent ou MB de la place mémoire des items.
4. Possibilité de passer en mode explorateur facilement.
5. Possibilité de se déplacer dans un répertoire.

Voici toutes les fonctionnalités désirées dans notre application.

# Conception

## Graphique

Dans ce chapitre, nous allons voir comment nous avons pensé approximativement l’agencement des différents interfaces.

### Mode explorateur

3

2

4

1

1. Bar de menu avec les fonctionnalités à disposition.
2. Chemin actuel dans la hiérarchie du système de fichier.
3. La liste des items.
4. Bouton permettant de passer en mode analyseur.

### Mode analyseur

3

4

1

2

1. Bar de menu avec les fonctionnalités à disposition.
2. Chemin actuel dans la hiérarchie du système de fichier.
3. Bouton permettant de passer en mode explorateur.
4. Diagramme circulaire permettant la visualisation de la

répartition de la place mémoire.

## Classes

Dans ce chapitre, nous allons voir comment nous avons pensé approximativement l’organisation des différents objets/classes.

**3**

**2**

**1**

Activity

Activité principale

Fragment

Vue en mode analyseur

Fragment

Vue en mode explorateur

Broadcast

(Update)

Cliquable

Cliquable

RecyclerView

Affichage du chemin actuel

Cliquable

Callbacks

Initialisation : String (path)

Update

Initialisation : String (path)

Comme on peut le voir ci-dessus, l’activité principale managera 3 principaux objets différents :

1. RecyclerView : Permet de mémoriser et d’afficher en tout temps le chemin actuel dans la hiérarchie du système de fichier de l’utilisateur.
2. Fragment : Permet d’afficher et mémoriser tous les items du répertoire courant (mode explorateur).
3. Fragment : Permet d’afficher la répartition de tous les items du répertoire courant (mode analyseur).

Ces trois entités seront bien évidemment cliquables pour permettre ou faciliter la navigation dans l’arborescence du système de fichier.

De plus, il sera possible d’interagir avec ces entités :

1. RecyclerView : Pour chaque déplacement dans l’arborescence, le chemin courant devra être mis à jour.
2. Fragment : Il sera nécessaire lors de l’instanciation d’un fragment du mode explorateur, de lui fournir le chemin courant. De plus, il sera possible de mettre à jour ce fragment dans le cas où un dossier/fichier serait supprimé ou ajouté.
3. Fragment : Il sera nécessaire lors de l’instanciation d’un fragment du mode analyseur, de lui fournir le chemin courant. De plus, dans le cas où un déplacement serait effectué, il sera nécessaire d’en informer l’activité principale.

# Implémentation

Dans ce chapitre nous n’allons pas décrire le code implémenté car cela risquerait d’être trop fastidieux. Néanmoins, nous allons lister les différentes étapes d’implémentation réalisées après la conception ainsi que quelques points ne concernant pas directement l’implémentation de code. Incontestablement, tous le code du projet est disponible en annexe est commenté.

Pour commencer, voici la liste des grandes étapes réalisées :

* Recherche pour le mode explorateur de fichier.
* Recherche pour le mode analyseur de fichier.
* Création et implémentation d’un projet pour le mode explorateur de fichier.
* Création et implémentation d’un projet pour le mode graphique (analyse).
* Fusion des deux projets. (Partie la plus complexe)
* Amélioration visuelles de l’application.
* Ajout de différentes fonctionnalités.
* Refactor du projet.
* Recherche et correction de bugs.
* Ajout d’une fonctionnalité non prévue
* Refactor du projet.
* Recherche et correction de bugs.

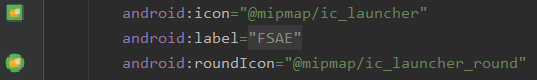
Bien entendu, la partie recherche et débogage est une tâche constante lors de la réalisation d’un projet.

Voici quelques modifications apportées au fichier « AndroidManifest.xml » :

1. La point le plus important, les permissions de l’application :

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />  
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

1. La modification du nom de l’application ainsi que son icone personnalisé :



Dans le fichier « build.gradle » il a été nécessaire d’ajouter la dépendance pour la partie graphique du projet :

// https://github.com/PhilJay/MPAndroidChart  
implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0'

Finalement, beaucoup d’icônes ont été crées

## Technologies utilisées

## Fonctionnalité supplémentaire

### Analyse

### Conception

### Implémentation

### Profiling

# Problèmes rencontrées

<application  
 android:requestLegacyExternalStorage="true"

Premier lancement de l’application

Visualisation des items vide dans le Pie chart

# Utilisation

Une vidéo démonstrative est disponible en annexe dans le dossier fourni.

# Conclusion

## Qui a fait quoi

## Améliorations

Refactor

Visualisation ergonomique des images classées

Priorisé la navigation dans le stockage externe (Carte SD).

## Compétences acquises

Kotlin

Android Studio

1. Plus précisément, toutes les technologie utilisées
2. L’outil de profiling d’Android Studio
3. La création est gestion de device virtualisé

Conception, implémentation, recherche et débogages d’un projet Android.

## Résultats obtenus

# Sources

Recherche de fichiers par extensions : <https://stackoverflow.com/questions/11015833/getting-list-of-all-files-of-a-specific-type>

Librairie graphique pour Android : <https://github.com/PhilJay/MPAndroidChart>

Tutoriel explorateur de fichier : <http://thetechnocafe.com/build-a-file-explorer-in-kotlin-part-1-introduction-and-set-up/>

# Annexes

1. Vidéo démonstrative
2. Projet complet Android Studio (.zip)
3. Application (.apk)

Date : 03.01.21

Nom de l’étudiant : Müller Pierrick et Spinelli Isaia