Auteur : Spinelli Isaia et

Simonet Yoann

Prof : [Dutoit Fabien](https://mse.hes-so.ch/consultation/horaires/?annee=2019&trimestre=1&id=8329&type=1)

Assist. : Christophe Greppin

Date : 20.09.2019

Classe : SYM-A

Laboratoire 1

Systèmes mobiles

Table des matières

[Introduction - 2 -](#_Toc19862548)

[Réponse aux questions - 2 -](#_Toc19862552)

[Conclusion - 3 -](#_Toc19862553)

[Difficultés rencontrées - 3 -](#_Toc19862554)

[Compétences acquises - 3 -](#_Toc19862555)

[Résultats obtenus - 3 -](#_Toc19862556)

## Introduction

Ce laboratoire est constitué de plusieurs manipulations destinées à implémenter une application élémentaire sur un émulateur et/ou sur un smartphone Android afin de nous familiariser avec l’environnement de développement Android.

## Réponse aux questions

### 1.Comment organiser les textes pour obtenir une application multi-langues (français, allemand, italien, langue par défaut: anglais)?Que se passe-t-il si une traduction est manquante dans la langue par défaut ou dans une langue supplémentaire?

Il ne faut pas écrire les strings en dure dans le code. Il est préférable d’utiliser des références sur un fichier xml.

Si la basile XML existe mais qu’elle est vide, ça n’affiche rien. Si elle n’existe pas, le programme ne build pas.

### 2.Dans l’exemple fourni, sur le dialogue pop-up, nous affichons l’icône android.R.drawable.ic\_dialog\_alert, disponible dans le SDK Android mais qui n’est pas très bien adapté visuellement à notre utilisation. Nous souhaitons la remplacer avec notre propre icône, veuillez indiquer comment procéder. Dans quel(s) dossier(s) devons-nous ajouter cette image ? Décrivez brièvement la logique derrière la gestion des ressources de type «image» sur Android.Info: Googlemet à disposition des icônes open source dans le style «Material Design» utilisé actuellement sur Android:https://material.io/resources/icons/

On doit aller changer l’image qui se trouve dans res/mipmap-mdpi/.

Les image que l’on met dans le dossier res sont directement charger dans l’exécutable de l’application, pour pouvoir y accéder il suffit de mettre leur chemin da res exemple : *R.mipmap.error\_log*

### 3.Lorsque le login est réussi, vous êtes censé chaîner une autre Activity en utilisant un Intent. Si je presse le bouton "Back" de l'interface Android, que puis-je constater ? Comment faire pour que l'application se comporte de manière plus logique ? Veuillez discuter de la logique derrière les activités Android.

Le mot de passe ainsi que l’email sont toujours dans les champs de saisie.

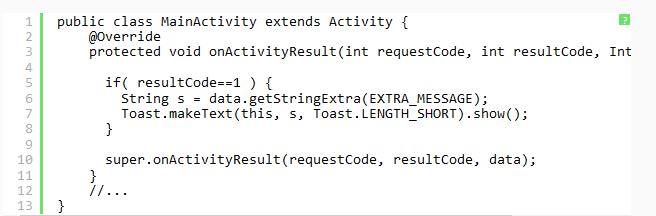
Il faut aller clear les champs dans avec une fonction onRestart().

On remarque que comme l’activité n’est pas détruite elle conserve son état. (voir question 8).

### 4.On pourrait imaginer une situation où cette seconde Activity fournit un résultat (par exemple l’IMEI ou une autre chaîne de caractères) que nous voudrions récupérer dans l'Activity de départ. Comment procéder ?

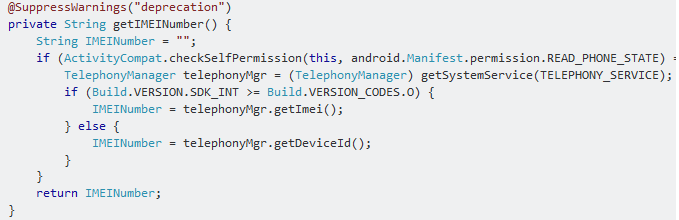
Dans la seconde Activity Il faut aller set le résultat avec la fonction setResult ex :Activity2.this.setResult();

Et il faut mettre en place un callback dans la première activité ex :



### 5.Vous noterez que la méthode getDeviceId()du TelephonyManager, permettant d’obtenir l’IMEI du téléphone, est dépréciée depuis la version 26 de l’API. Veuillez discuter de ce que cela implique lors du développement et de présenter une façon d’en tenir compte avec un exemple de code.

Il ne faudra en tenir compte pour la rétrocompatibilité sur les anciennes versions :



### 6.Dans l’activité de login, en plaçant le téléphone (ou l’émulateur) en mode paysage(landscape), nous constatons que les 2 champs de saisie ainsi que le bouton s’étendent sur toute la largeur de l’écran. Veuillez réaliser un layout spécifique au mode paysage qui permet un affichage mieux adapté et indiquer comment faire pour qu’il soit utilisé automatiquement à l’exécution.

On créé un nouveaux layout avec le même nom que celui du portrais et ensuit on le place dans le dossier layout-land des ressources.

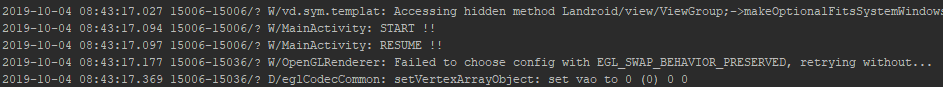
Et tout se fait automatiquement.

### 7.Le layout de l’interface utilisateur de l’activité de login qui vous a été fourni a été réalisé avec un LinearLayout à la racine. Nous vous demandons de réaliser un layout équivalent utilisant cette fois-ci un RelativeLayout.

Nous avons refait le layout comme demander voir authent.xml

### 8.Implémentezdans votre code les méthodes onCreate(), onStart(), onResume(), onPause(), onStop(), etc... qui marquent le cycle de vie d'une application Android, et tracez leur exécutiondans le logcat. Décrivez brièvement à quelles occasions ces méthodes sont invoquées. Vous expliquerez aussi l’enchainement de ces appels lorsque l’on passe d’une activité à l’autre. Comment pouvez-vous factoriser votre code pour éviter de devoir réimplémenter ces méthodes dans chacune de vos activités ?

Pour commencer les méthode onCreate() -> onStart() -> onResume() sont appelées.



Ensuite si on passe a une autre activité on passe en onPause() - > onStop().

Si on revient dans mainActivity avec la flèche on retombe sur onDestroy->onCreat() ->onStart() ->onResume()



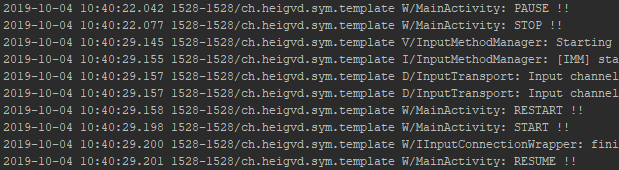


Du coup les état de l’ activité sont perdus.

Par contre si on revient avec le bouton android on passe par onRestart() -> onStart() ->onResume()

du coup les états de l’ activité sont conserver.

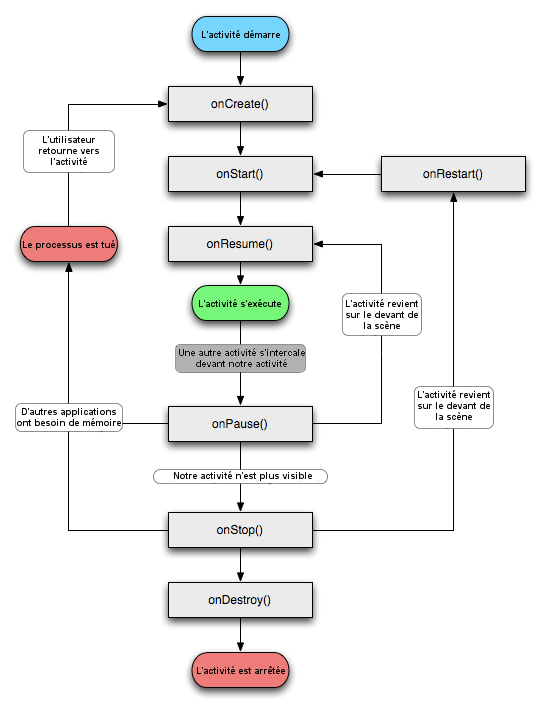
Si on verrouille le téléphone l’ activité se met en pause puis restarte quand on le déverrouille.



Finalement quand on quitte l’application on passe par onPause() -> onStop() -> onDestroy()



Si on veut avoir une vision d’ensemble on peut se référer à ce schéma :



Pour éviter de devoir réimplémenter ses méthodes dans chaque activité on pourrait créer une super classe Trace qui hériterai de AppCompatActivity.Ensuite on fera que toute nos activités héritent de Trace.

### 9.Question Bonus facultative - S’il vous reste du temps, nous vous conseillons de le consacrer à mettre en place la résolution des permissions au runtime.

Afin d’avoir les permissions au runtime, nous avons décidé de demander au démarrage (OnStart) de l’application les permissions. S’il refuse, l’application se ferme automatiquement. Nous avons fait que s’il accepte les permissions, nous ne les redemanderons jamais. Cependant, s’il refuse, nous demanderons à chaque ouverture de l’application la permission qu’il a refusée.

## Conclusion

### Difficultés rencontrées

Lire la carte SD sur notre téléphone car la fonction ne renvoie pas le bon chemin.

Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_DOWNLOADS*);

La résolution les permissions au runtime, il a fallu utiliser un callback.

### Compétences acquises

Mise en place d’une application simple sur Android !

### Résultats obtenus

Nous pensons avoir réussi complétement le laboratoire avec la question bonus.

Date : 06.10.19

Nom de l’étudiant : Spinelli Isaia et Simonet Yoann