Compte Rendu Détaillé – Application Android TP3

Introduction

Ce projet consiste à développer une application Android native, s'appuyant sur les concepts fondamentaux abordés dans la `Séance 3: Thème- Style et Navigation`. L'application met en œuvre la navigation entre activités, la gestion des ressources, l'internationalisation, la validation des entrées utilisateur et le respect du cycle de vie des activités.

Objectifs pédagogiques

- Comprendre la structure d'un projet Android (manifest, ressources, activités)
- Manipuler les composants de base : `Activity`, `Intent`, `Bundle`
- Utiliser les ressources (`strings.xml`, `styles.xml`, layouts, drawables)
- Appliquer l'internationalisation (i18n)
- Gérer le cycle de vie d'une activité
- Concevoir une interface utilisateur ergonomique

Fonctionnalités de l'application

1. **Interface utilisateur**

- **Champs de saisie** : Deux `EditText` pour entrer les nombres à additionner.
- **Boutons** :
  - `SUM` pour calculer la somme.
  - `RESET` pour effacer les champs et le résultat.
- **Affichage du résultat** : Un `TextView` affiche dynamiquement la somme.
- **Design** : Utilisation de styles personnalisés et de formes pour les boutons (via `drawable/buttonshape.xml`).

PROF

2. **Gestion des ressources**

- **Chaînes de caractères** externalisées dans `strings.xml` pour faciliter la traduction et la maintenance.
- **Styles** définis dans `styles.xml` pour une cohérence visuelle.
- **Layouts** structurés en XML pour séparer la logique de l'interface.

3. **Internationalisation**

- L'application adapte automatiquement son interface à la langue du système (français/anglais) grâce à plusieurs fichiers `strings.xml` (`values/`, `values-en/`).

4. **Cycle de vie des activités**

- Implémentation de logs dans les méthodes `onCreate()`, `onPause()`, `onStop()` pour observer le comportement de l'application lors des changements d'état.
- Sauvegarde/restauration des données utilisateur si nécessaire.

5. **Validation et robustesse**

- Vérification des entrées utilisateur pour éviter les erreurs de calcul.
- Affichage de messages d'erreur en cas de saisie invalide.

Architecture technique

- **MainActivity** : Accueil, saisie du nom, navigation vers l'écran de calcul.
- **DetailActivity** : Réalisation du calcul, affichage du résultat, gestion du cycle de vie.
- **Ressources** :
  - `strings.xml` (FR/EN)
  - `styles.xml`
  - `drawable/buttonshape.xml`
  - `layout/activity_main.xml`, `layout/activity_detail.xml`
- **Manifest** : Déclaration des activités et configuration de l'application.

Bonnes pratiques issues du cours

- **Séparation des responsabilités** : Interface en XML, logique en Java/Kotlin.
- **Utilisation des ressources** : Centralisation des textes, styles et images.
- **Respect du cycle de vie** : Utilisation des callbacks pour optimiser la gestion mémoire et l'expérience utilisateur.
- **Internationalisation** : Préparation de l'application pour plusieurs langues.
- **Validation** : Toujours vérifier les entrées utilisateur.

Conclusion

PROF

Ce TP met en pratique les notions essentielles du développement Android vues en cours :

- Création d'interfaces réactives et multilingues
- Gestion efficace des ressources
- Navigation et transmission de données entre activités
- Respect du cycle de vie et robustesse de l'application

L'application obtenue est simple, mais elle constitue une base solide pour des développements Android plus avancés.

Réalisé par : Hamza El Ghazouani