

# Interfaces Graphiques

Projet Droshed

Leroux Gwenaël  
Dergal Nacer



# Sommaire

I/ Architecture du projet

II/ Difficultés rencontrées et moyens mis en œuvre

III/ Répartition et organisation du travail

## I/ Architecture du projet

### **Le lancement :**

Lors du lancement de l'application, MainActivity appelle LoginActivity pour identifier l'utilisateur et récupérer la liste des modèles dans le même temps. LoginActivity lance une AsyncTask (DownloadModels) pour communiquer avec le serveur.

Une fois l'identification effectuée, MainActivity affiche via un Adapter la liste des modèles à l'utilisateur.

Lors du clic de l'utilisateur sur un modèle, MainActivity lance une AsyncTask (DownloadModel) qui va se charger de télécharger le fichier XML définissant le modèle. Celui-ci récupère ce fichier sur le serveur, le parse et l'ajoute en base de données et lance à son tour une AsyncTask (DownloadData) qui va parcourir l'ensemble des versions disponibles sur le serveur à partir de celle présente en local (si c'est le cas). Pour finir DownloadData va lancer l'activité SheetActivity dans son onPostExecute().

### **L'édition de feuille :**

Au lancement de SheetActivity, on va récupérer l'ensemble des informations en BDD concernant la feuille et créer avec un objet Sheet contenant des Cells (informations liées à une cellule) et des TypesCells (défini le type attendu dans les cellules correspondant à son numéro de colonne). L'objet Sheet fait les modifications en mémoire et en BDD via la classe utilitaire SheetDAO (en fait cette classe est utilisée pour chaque action faite sur la BDD).

Ensuite SheetActivity affiche une SheetView. Cette vue affiche les cellules de l'objet Sheet. Cette vue est transposable (colonnes et lignes inversées), zoomable, dézoomable et scrollable (verticalement, horizontalement et en diagonale). La transposition se fait par sélection de 'Invert Sheet' dans le menu. Le zoom se fait par pincement. Le scroll se fait en glissant son doigt sur l'écran. Lorsque l'utilisateur appuie longtemps sur une cellule, SheetView lance l'activité AskTextCell, il peut alors modifier son intitulé à condition qu'il soit éditable et qu'il corresponde au type attendu. Pour rajouter une ligne, l'utilisateur n'a qu'à sélectionner l'option dans le menu.

### **Le service :**

Au lancement de SheetActivity, celle-ci lance un service PutService qui va toutes les 10 secondes regarder les changements effectués depuis le dernier PUT sur le serveur ou le dernier GET. Il forme un XML avec ses modifications et tente de l'envoyer au serveur si la connexion le permet. Si c'est le cas, le service appelle l'AsyncTask (PutData) qui va envoyer les données aux serveurs et augmenter en local le numéro de version de la feuille.

## II/ Difficultés rencontrées et moyens mis en œuvre

### **AsyncTask :**

Au début nous avons du mal avec utiliser correctement les asyncTasks notamment avec le `onPostExecute()`. Nous n'arrivions pas à continuer le traitement normalement effectué dans `MainActivity` dans cette fonction. Pour résoudre le problème chacune de nos `asyncTask` dispose d'une classe interne `Entry` lui fournissant tout ce qu'il faut pour travailler.

### **Gesture :**

Nous avons eu un peu de mal avec les Gestures mais en effectuant des recherches cela a vite été résolu.

### III/ Répartition et organisation du travail

Pour la répartition du travail, nous avons décidé que Gwenaël s'occuperait de la partie en arrière (service, BDD ...) et Nacer de la partie graphique (SheetView ...).

Néanmoins nous avons pu travailler un peu sur la partie de l'autre pour l'enrichissement personnel. Cela a été une bonne expérience et une excellente répartition selon nous.