

**POLYTECHNIQUE  
MONTRÉAL**

TECHNOLOGICAL  
UNIVERSITY



INF8405 - Informatique Mobile

Hiver 2020

Polytechnique Montréal

TP #1

Application de jeu Android - Tic-tac-toe!

Soumis à Bilal Itani le 11 février 2020

Sami Tamer Arar 1828776

Aurélien Bouis 1787523

Julien Legault 1847125

# Table des Matières

<b>Introduction</b>	<b>3</b>
<b>Présentation des travaux</b>	<b>3</b>
<b>Difficultés rencontrées</b>	<b>4</b>
<b>Critiques et améliorations</b>	<b>5</b>
<b>Conclusion</b>	<b>5</b>

# Introduction

Le but de ce laboratoire était de développer une application mobile permettant à deux utilisateurs de jouer des parties de tic-tac-toe sur des grilles de différents formats tout en maintenant le pointage des joueurs. L'objectif académique de ce laboratoire était d'initier les membres de l'équipe au développement android. Au niveau des compétences en ingénierie, Ce travail avait pour but de nous faire développer notre compétence en communication et en conception. Dans un premier temps, nous ferons la présentation des travaux. Dans un deuxième temps, nous parlerons de nos difficultés rencontrés. Finalement, nous donnerons notre avis sur le travail et terminerons avec une conclusion.

## Présentation des travaux

Tout d'abord, le projet est séparé en 3 activités. La première est *SplashScreenActivity*. Elle contient une animation jouée au lancement de l'application. Un logo sous forme de texte est affiché au démarrage de l'application. En effet, cette animation a pu être réalisée en utilisant la librairie *AndroidViewAnimations*. Elle est disponible sur github : <https://github.com/daimajia/AndroidViewAnimations>

Ainsi, cette librairie permet d'animer un élément d'affichage grâce à une variété d'animation possibles. Aussi, elle permet de spécifier la durée de ces animations. Au démarrage de l'activité *SplashScreenActivity*, un compte à rebours de 4 secondes est démarré. Après ces 4 secondes, on passe à l'activité suivante.

La deuxième activité est *MainActivity*. Cette activité représente le menu principal du jeu. On y retrouve 3 boutons. Premièrement, il y a un bouton « Exit » qui termine le processus de l'application. Deuxièmement, il y a un bouton « About » qui affiche une fenêtre de type *AlertDialog*. On peut voir, à l'intérieur de cette fenêtre, les auteurs de l'application. Troisièmement, le bouton « Play » permet d'afficher une fenêtre *AlertDialog*, mais avec la possibilité de sélectionner la taille de grille du jeu de l'activité suivante. Lorsque l'utilisateur confirme son choix, l'activité suivante est lancée par l'utilisation d'un Intent en passant en paramètre avec la méthode *putExtra* une valeur qui détermine la taille de la grille.

La troisième activité est *GameActivity*. Cette activité utilise la valeur rentrée en paramètre lors du changement de l'activité pour générer une grille de la taille sélectionnée. Cette activité contient une grille 5x5 composés de boutons pour permettre à l'utilisateur d'interagir avec la grille. Lorsque l'utilisateur démarre un jeu de taille inférieur à 5x5, c'est-à-dire 4x4 ou 3x3, des boutons de la vue sont alors « cachés » et désactivés. Chaque fois qu'un utilisateur entre X ou O dans la case cliquée, un algorithme de vérification de la grille qui vérifie s'il y a un vainqueur est roulé. Si la grille est pleine, et qu'il n'y a pas de vainqueur, c'est une partie nulle.

L'algorithme de vérification du gagnant est effectué en 3 étapes : parcours des colonnes, parcours des rangées, parcours des diagonales. Il retourne *true* s'il y a une colonne, rangé ou diagonale avec tous les éléments identiques.

Pour entreposer le score du joueur 1 et du joueur 2, nous avons utilisé la méthode *getSharedPreferences()* qui nous permet de créer une collection privée dans la mémoire de l'application. À l'intérieur de cette collection, nous pouvons appeler *edit()* sur la collection pour ensuite entreposer le score du joueur 1 et du joueur 2 avec *putInt()* et de les associer à une clé. Pour récupérer les données, on utilise *getInt()* en passant la clé des valeurs en paramètre.

On a aussi géré lorsque l'utilisateur quitte la partie en pressant sur le bouton « Home » de l'appareil en faisant un *override* sur la méthode de l'activité *onUserLeaveHint()* pour recréer l'activité pour effacer toute trace de progression de la partie en cours lorsque l'utilisateur revient sur l'application.

Par ailleurs, pour faire jouer du son, nous avons utilisé la classe *MediaPlayer* et ses méthodes.

## Difficultés rencontrées

Lors du développement du jeu, nous avons rencontré certaines embûches plus ou moins difficiles à surpasser. La première est dûe à la découverte des nouvelles technologies liées au développement sur Android (Android Studio, Kotlin notamment). Ces dernières ont imposé un temps d'apprentissage et de familiarisation non négligeable.

Par ailleurs, nous avons eu des difficultés à faire en sorte d'adapter les boutons à l'échelle lorsque les cases des grilles 4\*4 et 5\*5 apparaissent. En effet, quelques propriétés graphiques permettant cette adaptation ont dû être trouvées puis ajoutées.

## Critiques et améliorations

Nous avons apprécié la formulation de l'énoncé qui est clair et concis (à la fois très spécifique et laissant tout de même un certain degré de liberté notamment par rapport à la conception). En revanche, l'aspect chronophage de la découverte du développement sur Android nous est apparu assez contraignant. Ainsi, il serait peut-être envisageable de prévoir une petite initiation aux concepts de base de la programmation Android lors des futures sessions.

## Conclusion

En définitive, ce laboratoire nous a permis (pour 2 d'entre nous) de découvrir le développement mobile sur Android, chose que nous n'avions jamais faite auparavant. Nous en sommes d'ailleurs assez satisfaits, car c'est un volet important de la programmation de nos jours. Enfin, le jeu du Tic Tac Toe nous est apparu comme un choix pertinent pour une première prise de contact avec les technologies de développement sur mobile. Par ailleurs, il serait intéressant d'introduire une fonctionnalité multijoueur à notre application qui supportera plusieurs appareils en même temps.