

**École Polytechnique de Montréal**  
Département de génie informatique  
Hiver 2023  
**INF8405 – Informatique Mobile**

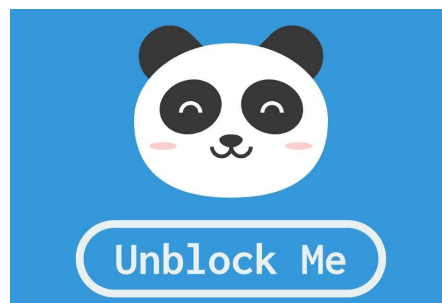
**POLYTECHNIQUE  
MONTREAL**

LE GÉNIE  
EN PREMIÈRE CLASSE



**TP1 :**

**Application de jeu Android - Unblock me!**



**Présenté par:**

- **1933097** Pritam Patel
- **1955204** Batikan Iscan
- **2169358** Imad Berrahil
- **1961278** Simon Tran

**Soumis à M : Mehdi Kadi**

**le:16 Février 2023**

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>Présentation</b>	<b>3</b>
<b>Difficultés rencontrées</b>	<b>6</b>
<b>Critiques et améliorations</b>	<b>7</b>
<b>Conclusion</b>	<b>8</b>
<b>Bibliographie</b>	<b>8</b>

## Introduction

Dans le cadre du cours d'informatique mobile, le but principal de ce premier TP est de s'initier au développement des services pour périphériques mobiles. Plus précisément, développer une application mobile dénommée **Unblock Me**. Cette application est destinée à la plateforme Android et implémentée en langage Kotlin avec l'IDE Android Studio. Pour que l'application soit fonctionnelle sur une grande partie des appareils, nous avons choisi la configuration par défaut sur Android 7 (API 24) afin de respecter les spécifications du laboratoire. Par la même occasion, nous avons pu mettre en pratique deux des 12 qualités de l'ingénieur définies par le BCAPG soient la conception et la communication.

## Présentation

L'application **Unblock Me** est un jeu de puzzle disponible sur le système d'exploitation Android. C'est un jeu simple mais stimulant dans lequel le joueur doit déplacer des blocs (Forme de bateaux) autour d'un plateau pour dégager un chemin pour que le bloc principal (Bateau gris) sorte.

La vue du jeu est une grille de 6x6 sur laquelle sont positionnées des blocs horizontaux 1\*N et des blocs aussi verticaux N\*1 avec N (2,3). Sur la troisième ligne à partir du haut se positionne le bloc principal (bateau gris) qui est coincé par ses pairs. Le but du joueur est de déplacer le bloc principal horizontalement jusqu'à le faire sortir de la grille de jeu avec le minimum de déplacements de blocs (Bateaux).

Pour la réalisation du projet, on procédera comme suit:

- Comprendre les règles et le fonctionnement du jeu : Il est important de comprendre les règles et le fonctionnement du jeu "Unblock Me" pour pouvoir le reproduire.
- Conception de l'interface utilisateur : Il est important de concevoir une interface utilisateur conviviale pour le jeu. Cela inclut la conception de la grille de jeu, des boutons de commande et de l'affichage du score.

- Développement du moteur de jeu : Le moteur de jeu gère les déplacements des blocs, vérifie si un mouvement est valide ou non et détecte quand le puzzle est résolu.
- Stockage des données de jeu : Il est important de stocker les données de jeu, telles que les scores et les états de jeu, pour pouvoir les charger à tout moment.
- Tests et débogage : Il est important que le jeu fonctionne comme prévu. En tant que tel, nous devons tester son comportement et nous assurer qu'il n'y ait pas de bugs.

Ce sont les étapes de base pour développer un jeu similaire à "Unblock Me" sur Android avec API 24.

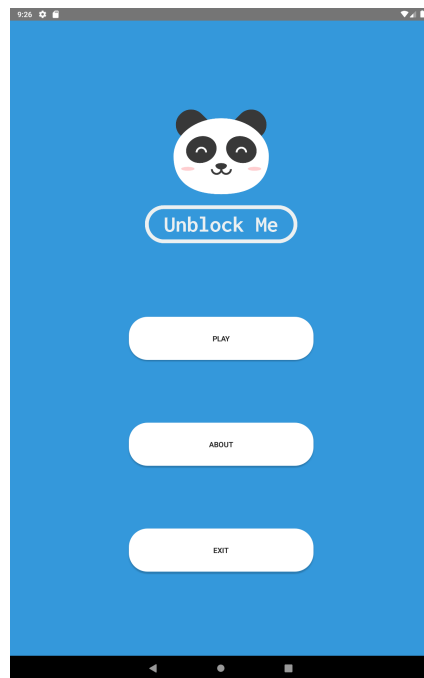
Puisque le but de ce travail pratique est de créer un jeu vidéo mobile, nous avons décidé d'utiliser la populaire librairie LibGDX. En effet, cette librairie permet de simplifier grandement le développement d'un jeu mobile, puisqu'elle permet de coder en Java ou Kotlin. Les interfaces Java/Kotlin de la librairie permettent de faire abstraction des éléments plus complexes liés à la gestion des graphiques. Cela permet ainsi de créer aisément un jeu qui fonctionne bien même sur des appareils mobiles peu puissants. De plus, la librairie permet de créer des versions du même jeu pour différentes plateformes (iOS, windows, mac...) sans devoir tout réécrire, ce qui pourrait devenir intéressant dans le futur. De plus, nous avons utilisé *Scene2d* qui fait partie de LibGDX afin de faciliter la gestion des objets graphiques en utilisant un *Stage* et des *Actors*.

Nous avons décidé d'utiliser le patron de conception *Model-View-Presenter* (MVP) pour gérer la logique du jeu, puisque cela permettait d'aisément séparer la logique de présentation de la logique du jeu et des données.

Au démarrage de l'application, on se retrouve bien évidemment sur le menu d'accueil qui comporte 3 boutons : **Play**, **About**, **Exit**, comme le montre la figure 1 ci-dessous afin de changer entre les différentes pages.

- En appuyant sur le bouton "Play", l'utilisateur accède à la page de jeu où le premier niveau de puzzle est chargé et prêt à être joué.

- En appuyant sur le bouton "About", l'utilisateur accède à un menu où les noms des membres de notre équipe sont affichés.
- En appuyant sur le bouton "Exit", l'utilisateur peut fermer l'application.



**Figure 1:** vue principal du jeu Unblock Me

L'interface du jeu contient l'attraction principale : la matrice de jeu contenant les blocs que l'utilisateur déplace afin de libérer le bateau gris. De plus, cette interface comporte un compteur de coups en haut à droite, le numéro de niveau et des boutons pour jouer aux énigmes précédentes et suivantes (si disponibles) en haut à gauche ainsi que des boutons supplémentaires pour revenir au menu principal, pour annuler le dernier coup et pour réinitialiser le puzzle à son état d'origine. Une fois que le bateau gris entre en contact avec le côté droit de l'écran, l'écran de victoire s'affiche.



**Figure 2 :** Vue de jeu Unblock Me

La classe principale contient un *listener* d'événements tactiles pour détecter les différentes intersections avec les éléments. Dans notre projet, un historique des actions est conservé pour retourner en arrière et une liste de chaque bloc est conservée.

L'initialisation de chaque level se fait par désérialisation d'un fichier contenant les informations du niveau donc, une fois que les objets sont reçus, ils peuvent être désérialisés et utilisés comme des objets. Ainsi, on peut créer de nouveaux niveaux facilement à travers ces données.

## Difficultés rencontrées

La première étape pour réaliser ce jeu était de comprendre les règles du jeu et de définir les fonctionnalités nécessaires à sa mise en œuvre. Pour cela, il a fallu créer les éléments graphiques et les mettre en place sur l'interface utilisateur. Enfin, il a fallu développer la logique du jeu, la gestion des niveaux et des scores, et la gestion des entrées utilisateurs.

Pour un développeur expérimenté, la réalisation de ce jeu peut être relativement simple, car LibGDX fournit une gamme d'outils et de fonctionnalités pour la création de jeux. Cependant, pour un développeur novice ou qui n'a pas d'expérience en développement de jeux, cela peut être plus difficile et nécessiter une formation et une pratique supplémentaires. Afin de procéder à la réalisation du jeu, nous avons décidé d'utiliser **Kotlin** à la place de **Java**, car il est très concis, moderne et parce que la documentation d'Android le recommande. Pour le développement de notre application Android, on aura besoin de moins de codes pour accomplir ce que des langages comme Java peuvent faire. Cela permet d'économiser beaucoup de temps et d'énergie utilisés dans le processus, puisqu'il prend également moins de ressources. Dans la réalisation du projet, on a dû faire face à certaines difficultés.

Parmi ces difficultés :

- L'intégration de **Libgdx** avec Android, puisque la spécification de l'architecture a demandé un certain temps de compréhension et d'implémentation. Nous avons dû nous familiariser avec la librairie.
- La prise en charge des collisions et l'adaptation des résolutions a rendu la tâche un peu complexe
- Nous avons aussi rencontré des difficultés à restreindre le déplacement des blocs pour empêcher la superposition des blocs et empêcher les blocs de sortir de la grille
- L'utilisation de Kotlin. Il a fallu surmonter une certaine courbe d'apprentissage, puisqu'aucun membre de notre équipe n'était familier avec le langage.
- Puisque nous n'avons pas d'expérience en développement de jeu vidéo, il a été difficile de décider de quels patrons de conceptions utiliser. Au final, nous avons choisi le MVP: Model-View-Presenter

## Critiques et améliorations

Nous avons particulièrement apprécié les détails se retrouvant dans les exigences fonctionnelles de l'application. En effet, en cas de doute, nous n'avions qu'à nous référer au chargé du TP. Pour nous éclaircir le tout afin de pouvoir réaliser une interface utilisateur conviviale. Nous avons donc dû apprendre beaucoup de choses.

Par conséquent, ce laboratoire n'a pas été trop difficile à entreprendre, excepté les quelques difficultés que nous avons pu surpasser.

Cependant, nous aurions aimé avoir plus de liberté dans l'implémentation de certaines fonctionnalités. En effet, les consignes du laboratoire étaient très précises et imposaient des spécifications exactes. Beaucoup de temps a été consacré à remplir certains critères qui ne semblaient pas particulièrement pertinents. Nous aurions aimé avoir une plus grande liberté pour ajouter notre touche personnelle.

## Conclusion

Ce rapport a été très utile pour approfondir nos connaissances en développement mobile et de nous familiariser avec l'IDE Android studio. Nous avons aussi appris à utiliser la bibliothèque LibGDX pour le développement de jeux. De plus, il nous a permis d'apprendre le langage Kotlin et certaines bonnes pratiques pour le développement mobile.

## Bibliographie

*Documentation :* *android developers*. Android Developers. (n.d.). Retrieved February 16, 2023, from <https://developer.android.com/docs>

*Source & documentation.* libGDX. (n.d.). Retrieved February 16, 2023, from <https://libgdx.com/dev/>

*Kotlin docs: Kotlin.* Kotlin Help. (n.d.). Retrieved February 16, 2023, from <https://kotlinlang.org/docs/home.html>