

SAÉ 4.1 : MasterMind



Département informatique

Formation : BUT2 FI

Rapport de projet



Réalisé par :

Adam MEDDAHI

Kamil MARDACI

Youcef SAYAD

Responsable de projet :

Luc HERNANDEZ

Sommaire

Introduction	2
Fonctionnalités	3
Menu de sélection	3
Solo	3
Duo (hot seat)	4
Fin d'une partie	4
Pièce vide	5
Fonctionnalités annexes	6
Structure du programme	7
MenuActivity	7
SettingsActivity	8
ChoiceActivity	8
GameActivity	9
Algorithme	10
Conclusions	11

Introduction

Le Mastermind est un jeu de société populaire qui a été créé en 1970. Il oppose deux joueurs qui tentent de deviner la combinaison secrète de leur adversaire en un minimum de coups. Le jeu est composé d'un plateau, de pièces de combinaison et de pions d'évaluation. Le joueur défenseur choisit une combinaison secrète que l'attaquant doit deviner en formulant des propositions. Le défenseur évalue les propositions de l'attaquant en plaçant des pions de couleur pour indiquer les pièces bien ou mal placées.

Dans le cadre de ce projet, nous avons développé une application Android qui permet à deux joueurs de s'affronter sur un seul appareil, ou à un joueur de jouer contre l'ordinateur en mode automatique.

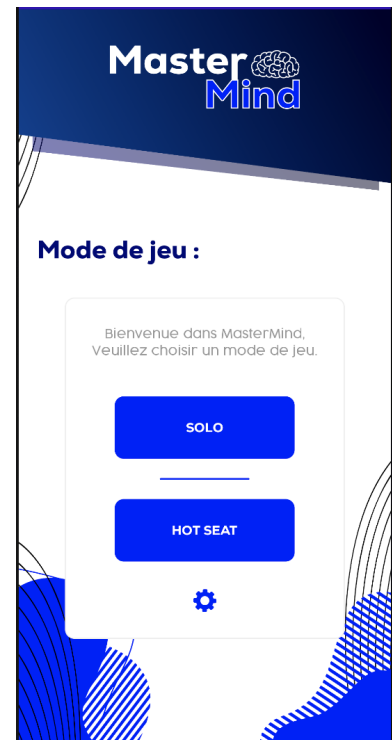
Ce rapport présente le travail que nous avons effectué pour développer l'application. Nous décrivons la structure de l'application, ainsi que l'algorithme que nous avons mis en place pour évaluer les propositions de l'attaquant. Nous présentons également notre méthode de développement, les outils que nous avons utilisés, ainsi que les difficultés que nous avons rencontrées et les solutions que nous avons trouvées.

Fonctionnalités

- Menu de sélection :

Notre application offre deux modes de jeu distincts : le mode **Solo** et le mode **Duo**.

Une roue crantée permet l'accès à la section des réglages.

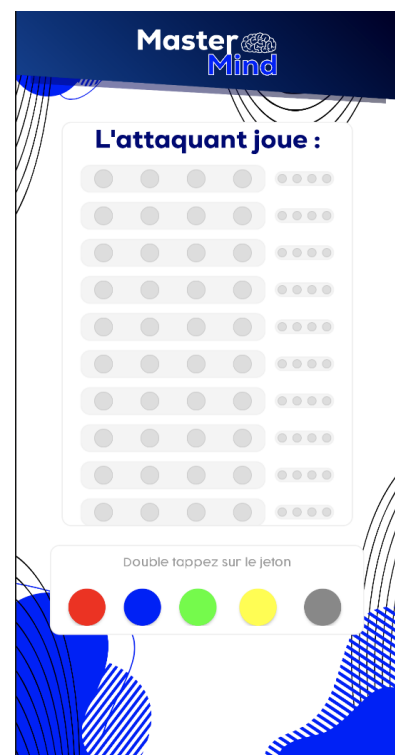


- Solo :

Le joueur joue contre l'ordinateur qui génère aléatoirement la combinaison secrète. Il prend le rôle d'attaquant et doit deviner cette combinaison à l'aide du plateau.

Les tentatives précédentes et leur résultat sont également affichés pour permettre à l'attaquant de s'adapter.

La partie se termine lorsque le joueur trouve la combinaison secrète ou lorsqu'il a épuisé ses dix tentatives.

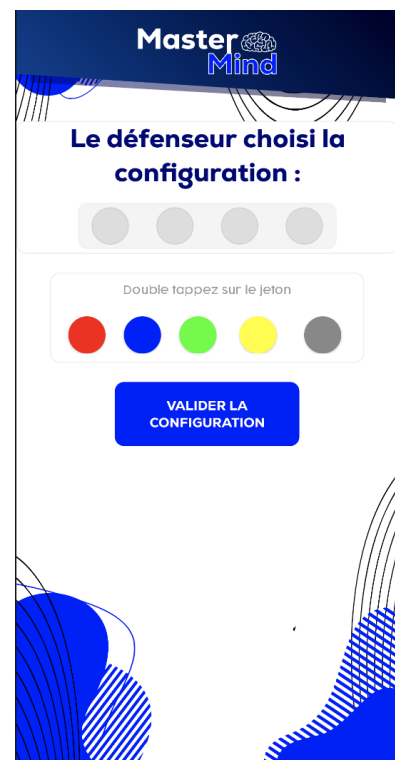


- Duo (hot seat) :

Deux joueurs peuvent jouer en hot seat en partageant le même appareil.

Le joueur défenseur choisit d'abord la combinaison secrète, à l'aide d'une grille de 4 cases qu'il doit remplir en appuyant sur les pions disponibles.

L'application affiche ensuite un plateau de jeu contenant les pions d'évaluation et les différentes combinaisons proposées, ce qui permet à l'attaquant de jouer.

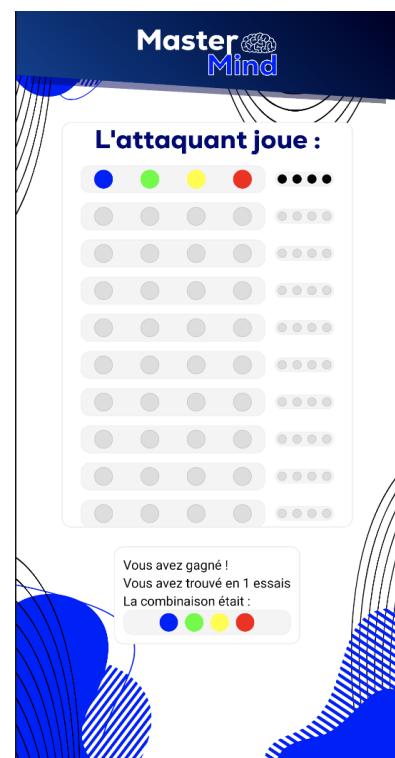


- Fin d'une partie :

Lorsqu'une partie se termine, un récapitulatif apparaît indiquant la victoire ou la défaite, avec le nombre d'essais et révèle la combinaison secrète.

L'historique des coups et les évaluations restent disponibles.

Une tape n'importe où ramène au menu principal.

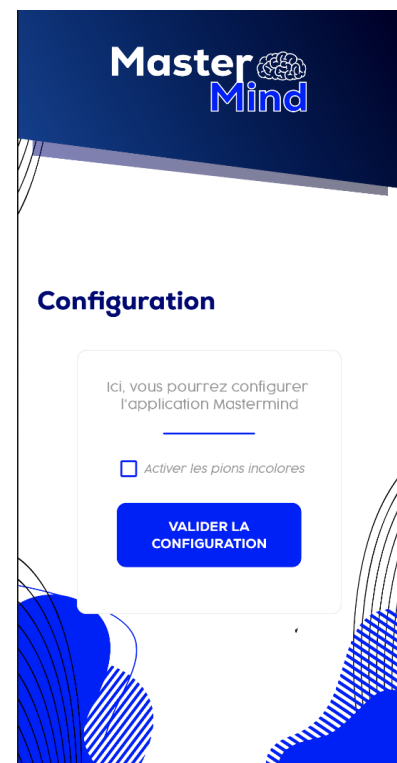


- Pièce vide :

Cette règle optionnelle consiste à autoriser la combinaison secrète à contenir des pièces vides, c'est-à-dire des positions où aucune couleur n'est sélectionnée.

Les combinaisons candidates peuvent laisser vides certaines positions, ce qui équivaut à ajouter une nouvelle couleur et augmente la difficulté pour l'attaquant.

Elle est activable à partir du menu dans une section de réglages.



- Fonctionnalités annexes :

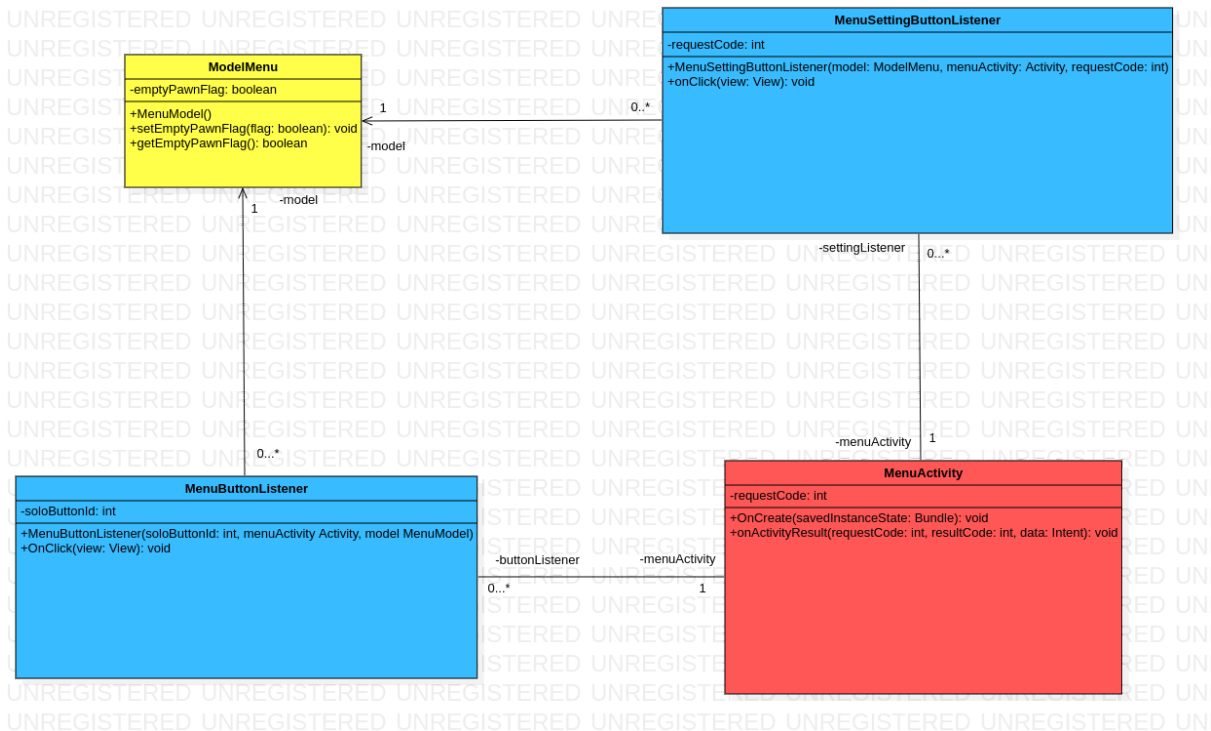
En cas de retour en arrière, toutes les activités mènent au menu. Dans le cas du menu, un retour arrière termine l'application.

Pour des raisons d'ergonomie, le choix de la couleur d'une pièce se fait en un seul geste dans les deux activités concernées.

Structure du programme

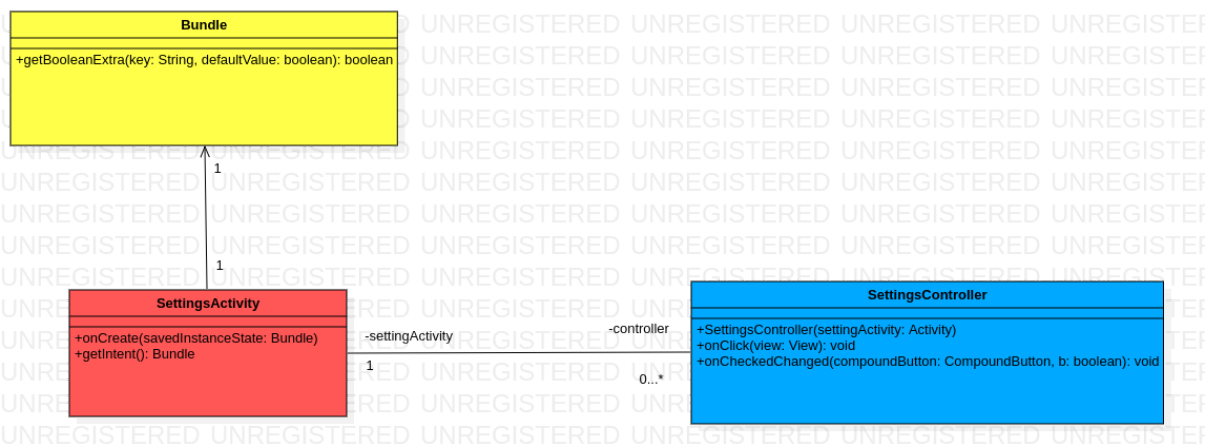
Tous les diagrammes présentés ci-dessous sont disponibles dans le dépôt du projet.

MenuActivity :



SettingsActivity :

Ici, nous avons considéré comme plus approprié d'utiliser l'attribut Bundle de l'activité (récupérable via getIntent()) pour représenter le modèle. Ce choix a été motivé par l'économie de ressources (évite de recréer un objet Bundle) et par l'imbrication d'élément du modèle et de la vue nativement présente en android (l'activité qui, en même temps, gère le **cycle de vie de la vue** et **contient le bundle** récupérable via getIntent()). On évite aussi d'ajouter la "lourdeur" du MVC pour une aussi petite application.

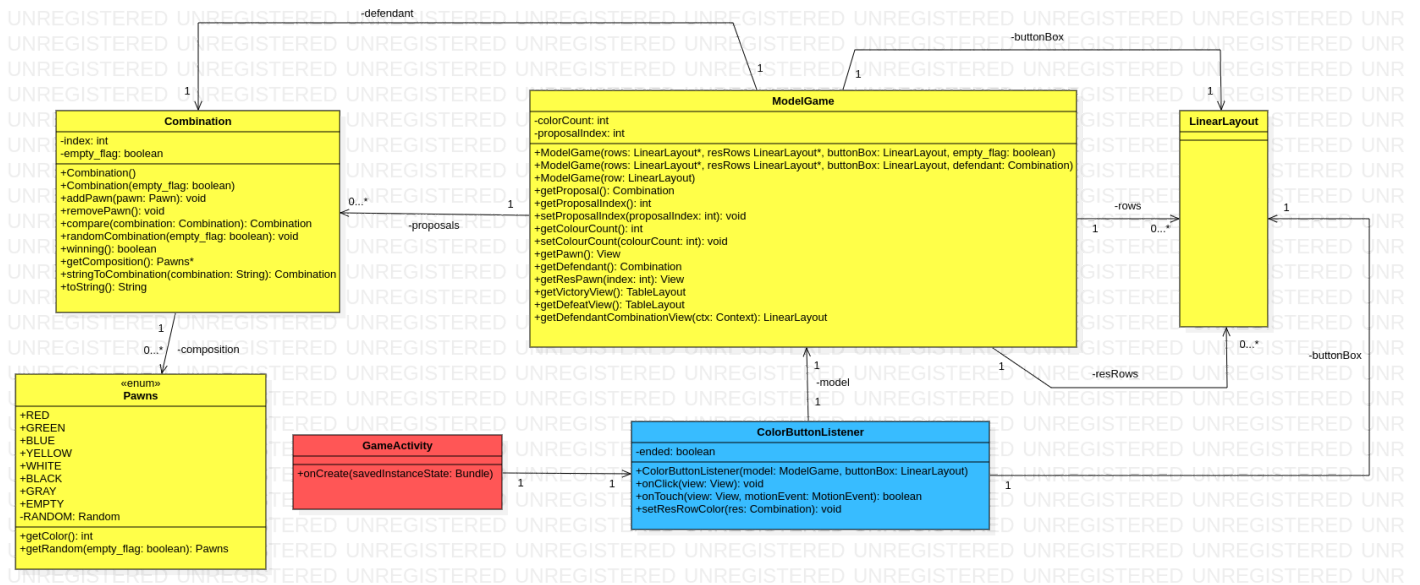


ChoiceActivity :

Ici, nous avons eu la même réflexion que pour SettingsActivity.



GameActivity :



Algorithme

L'algorithme utilisé pour le Mastermind est relativement simple, mais efficace. Il est conçu pour évaluer la combinaison candidate proposée par l'attaquant et déterminer combien de pièces sont bien placées et combien sont mal placées.

L'algorithme compare chaque pièce de la combinaison candidate avec la pièce correspondante de la combinaison secrète. Si les deux pièces sont identiques, cela signifie que la pièce est bien placée. Si elles sont différentes, l'algorithme vérifie si cette pièce se trouve ailleurs dans la combinaison secrète. Si c'est le cas, cela signifie que la pièce est mal placée.

Une fois que toutes les pièces de la combinaison candidate ont été comparées avec les pièces de la combinaison secrète, l'algorithme retourne le nombre de pièces bien placées et le nombre de pièces mal placées.

Le nombre de pièces bien placées et mal placées est utilisé pour évaluer la proposition de l'attaquant. Ces informations sont ensuite affichées à l'attaquant pour l'aider à déterminer sa prochaine proposition.

Il est important de noter que cet algorithme ne garantit pas que l'attaquant trouvera la combinaison secrète en un nombre limité de coups, mais il fournit une méthode efficace pour évaluer chaque proposition et aider l'attaquant à progresser vers la combinaison secrète.

Conclusions

Adam :

Je suis très satisfait du travail accompli dans le cadre de ce projet. En tant que membre du groupe, j'ai pu contribuer à la conception de l'application Mastermind pour Android. J'ai appris beaucoup de choses au cours de ces dernières semaines, notamment sur la programmation en Java et sur la méthodologie de développement d'une application Android. J'ai également apprécié la collaboration avec mes deux collègues qui ont été très impliqués dans le projet. Je suis fier du résultat obtenu et j'espère que notre application sera appréciée.

Kamil :

Je suis très heureux d'avoir participé à ce projet en équipe. La conception d'une application Android n'était pas facile, mais nous avons réussi à relever le défi ensemble. J'ai particulièrement apprécié de travailler sur la partie algorithmique du jeu, qui était pour moi un véritable défi intellectuel. Je suis satisfait de notre application et j'ai pris du plaisir à la réaliser. Je remercie mes deux collègues pour leur collaboration et leur soutien tout au long du projet.

Youcef :

Je suis fier de notre travail en équipe pour le développement de l'application Mastermind pour Android. Cette expérience a été très enrichissante pour moi, car elle m'a permis d'apprendre beaucoup de choses sur la programmation en Java et sur le développement d'applications mobiles. J'ai particulièrement apprécié la conception de l'interface utilisateur, qui était pour moi un véritable défi créatif. Je tiens également à remercier mes deux collègues pour leur collaboration tout au long du projet.