



REALISÉ PAR:

IZANIC Hugo
GALLEGO Julian
AISSI Jude-Christ

2024-2025

RAPPORT DE PROJET

APPLICATION ANDROID - FLOW FREE



SOMMAIRE

Introduction (3)

Fonctionnalités de l'application

- Menu principal
 - Sélection et chargement des puzzles (4-7)
 - Interface utilisateur
 - Mode achromate
 - Persistence des données
-

Structure du programme (9)

Diagrammes de classes (9-10)

Algorithme de résolution et persistance des données (11)

Conclusion (12)

Conclusions personnelles (13)



INTRODUCTION

Le projet inspiré du jeu "Flow Free" ou "Dot Link" consiste à développer une application Android de type puzzle. Le concept du jeu est simple mais stimulant : il propose une grille carrée contenant des paires de points colorés que le joueur doit connecter par des chemins. Ces chemins doivent suivre des règles précises : ils ne peuvent être que horizontaux ou verticaux, ne doivent pas se croiser, et l'ensemble des chemins doit couvrir toutes les cases de la grille.

Notre objectif était de créer une application complète en Java pour Android, avec une interface utilisateur intuitive permettant au joueur de tracer des chemins à l'aide de gestes tactiles, de sélectionner différents puzzles stockés au format XML, et de bénéficier d'options d'accessibilité comme le mode achromate pour les personnes daltoniennes.

Le défi principal de ce projet résidait dans la mise en place d'une architecture logicielle robuste capable de gérer efficacement les interactions tactiles, de valider les chemins tracés par l'utilisateur, et de déterminer automatiquement si le puzzle est correctement résolu.



FONCTIONNALITÉS DE L'APPLICATION

Menu principal

Le menu principal constitue le point d'entrée de notre application. Il permet à l'utilisateur de:

- Ouvrir le menu de sélection de puzzle
- Accéder aux options de configuration
- Quitter l'application via un retour arrière

Le menu affiche la liste des puzzles disponibles, chacun identifié par son nom défini dans le fichier XML correspondant. Si ce nom n'est pas spécifié, c'est le nom du fichier (sans extension) qui est utilisé. Les puzzles dont le fichier XML contient des erreurs apparaissent grisés et ne peuvent pas être sélectionnés.

Interface de jeu

Pour optimiser l'expérience utilisateur lors de la sélection des puzzles, nous avons implémenté une ListView, solution Android standard offrant plusieurs avantages ergonomiques. Si une activité est mise en pause alors les boutons de celle ci sont desactive et reactiver quand l'activité reprend

Présentation des puzzles

- Chaque puzzle est représenté par un bouton personnalisé
- La taille du puzzle (par exemple "5x5") est clairement affichée sur chaque bouton
- Le nom du puzzle est visible à côté de son bouton



FONCTIONNALITÉS DE L'APPLICATION

Exemple d'un wireframe pour illustrer cette approche



Gestion des puzzles invalides

Nous avons mis en place un mécanisme visuel simple mais efficace pour différencier les puzzles valides et invalides :

- Les puzzles invalides sont grisés et non cliquables
- Un utilisateur comprend immédiatement quels puzzles peuvent être joués
- Pour implémenter cette fonctionnalité, on a simplement pas relui de listener aux boutons pour dont le puzzle était invalide.



Sélection et chargement des puzzles

L'application charge dynamiquement les puzzles depuis le répertoire `app/src/main/assets/puzzles`. Nous utilisons l'interface `XmlResourceParser` pour analyser les fichiers XML et extraire les informations nécessaires à la création des puzzles:

- Taille de la grille (entre données-données)
- Nom du puzzle
- Positions des paires de points

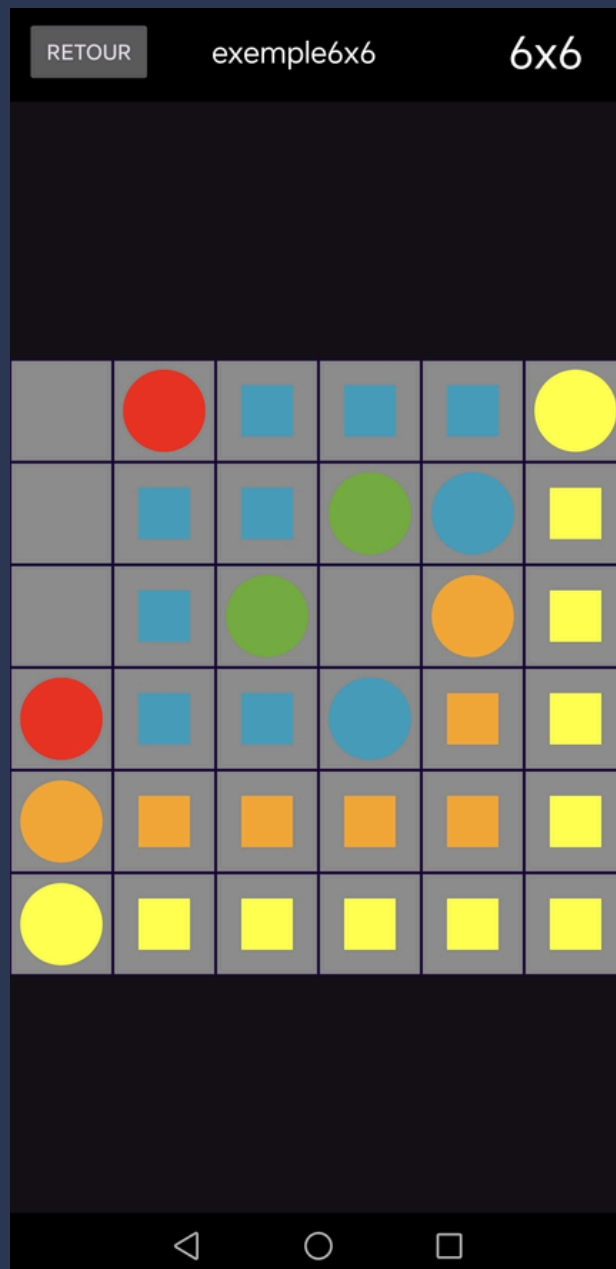


Interface de jeu

Le joueur peut:

- Tracer un chemin en posant son doigt sur un point et en le déplaçant de case en case jusqu'au point correspondant
- Remplacer un chemin existant en recommençant à partir d'une de ses extrémités
- Visualiser les chemins déjà tracés

Si le doigt du joueur sort de la grille, entre dans une case déjà occupée par un autre chemin, ou se lève avant d'atteindre le second point de la paire, le chemin est automatiquement annulé.



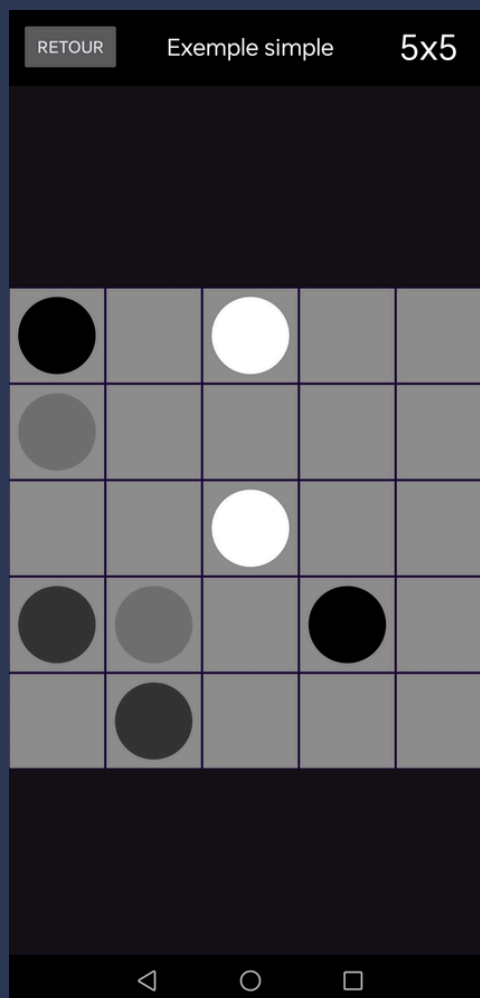
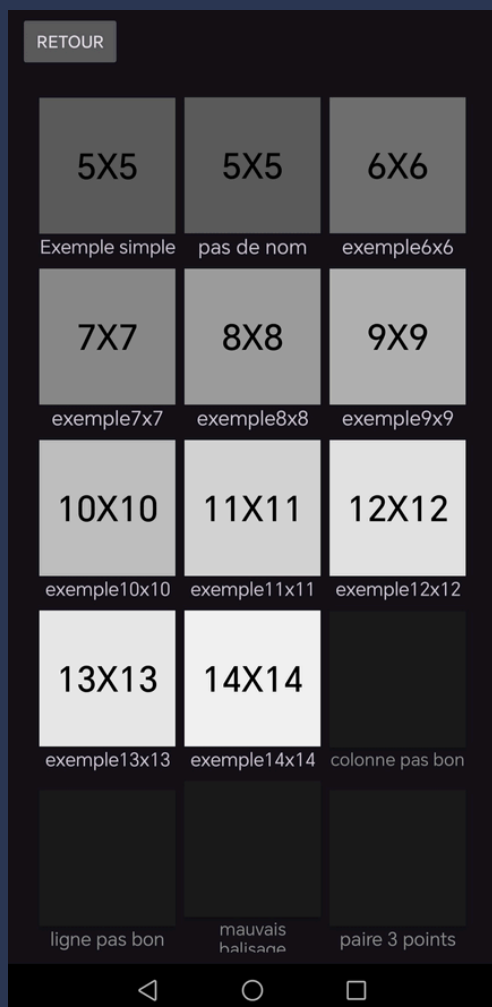


Détection de résolution

L'application détecte automatiquement lorsque le puzzle est résolu, c'est-à-dire quand toutes les paires de points sont reliées par des chemins valides et que toutes les cases de la grille sont occupées. Un message de félicitation est alors affiché, et le joueur ne peut plus modifier les chemins.

Mode achromate

Pour améliorer l'accessibilité de notre application, nous avons implémenté un mode achromate. Dans ce mode, les couleurs des points et des chemins sont remplacées par différentes nuances de gris, tout en gardant une distinction claire des points et chemins. Cette option est accessible depuis le menu principal via l'activité de configuration.





STRUCTURE DU PROGRAMME

Un puzzle est une grille composer d'un tableau double dimensions de type Case. La valeur de chaque case est soit de type Chemin ou de type Point. C'est seulement une fois que le tracer entre deux point est correct que le chemin effectuer par l'utilisateur est mit a jour dans la grille du puzzle. Toutes les cases parcourut on une valeur de type Chemin.

Diagramme de classes - Activité principale

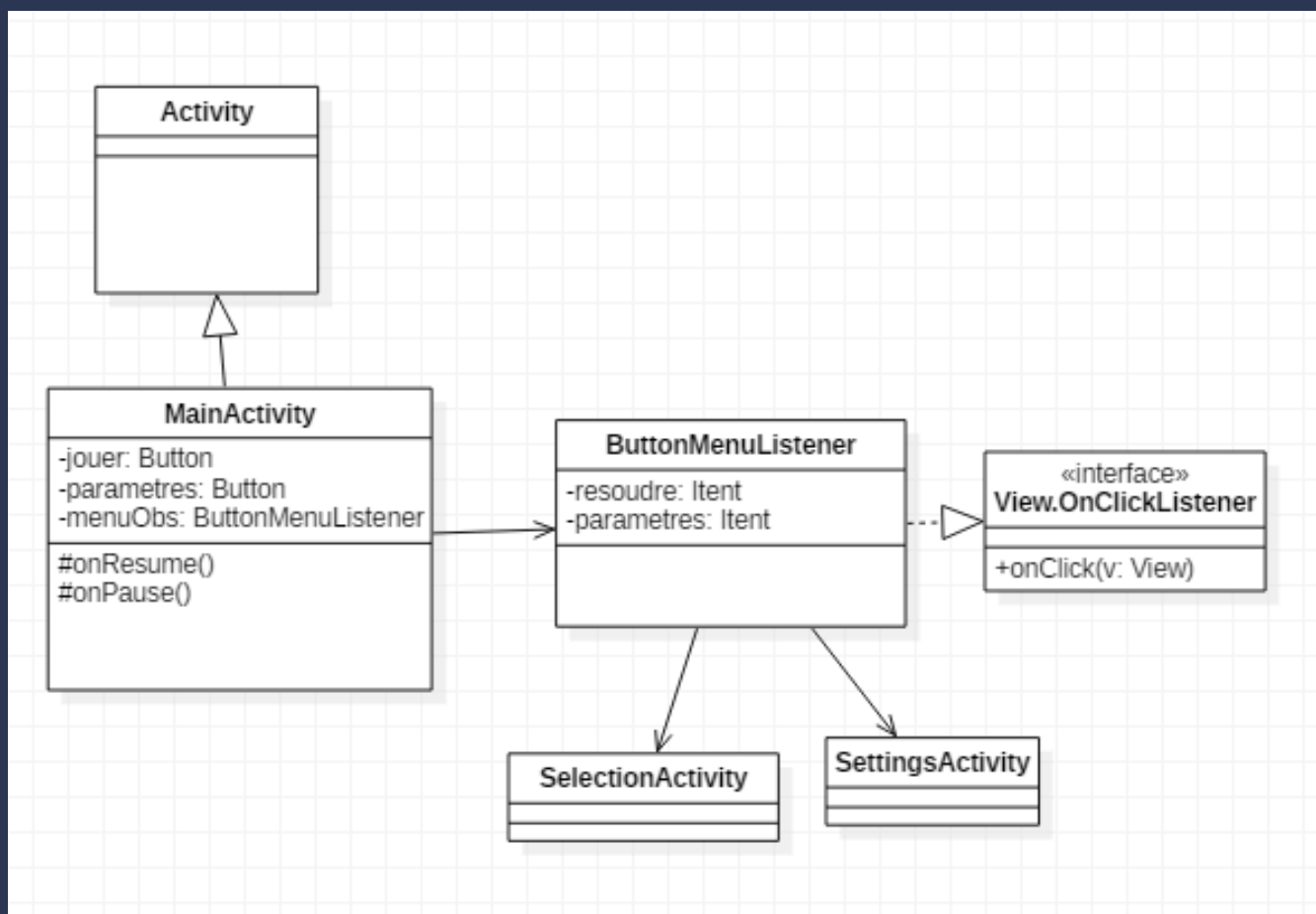




Diagramme de classes - Activité de preference

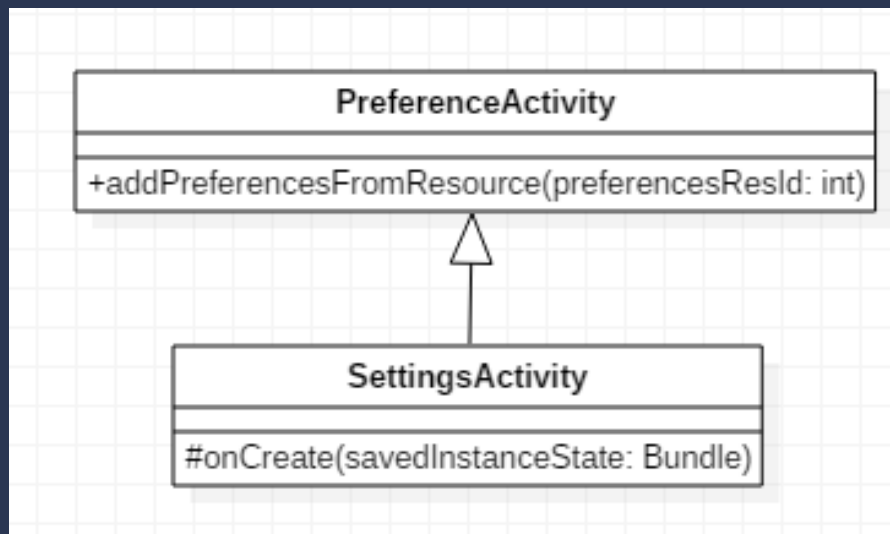


Diagramme de classes - Activité de Selection de puzzle : SelectionActivity

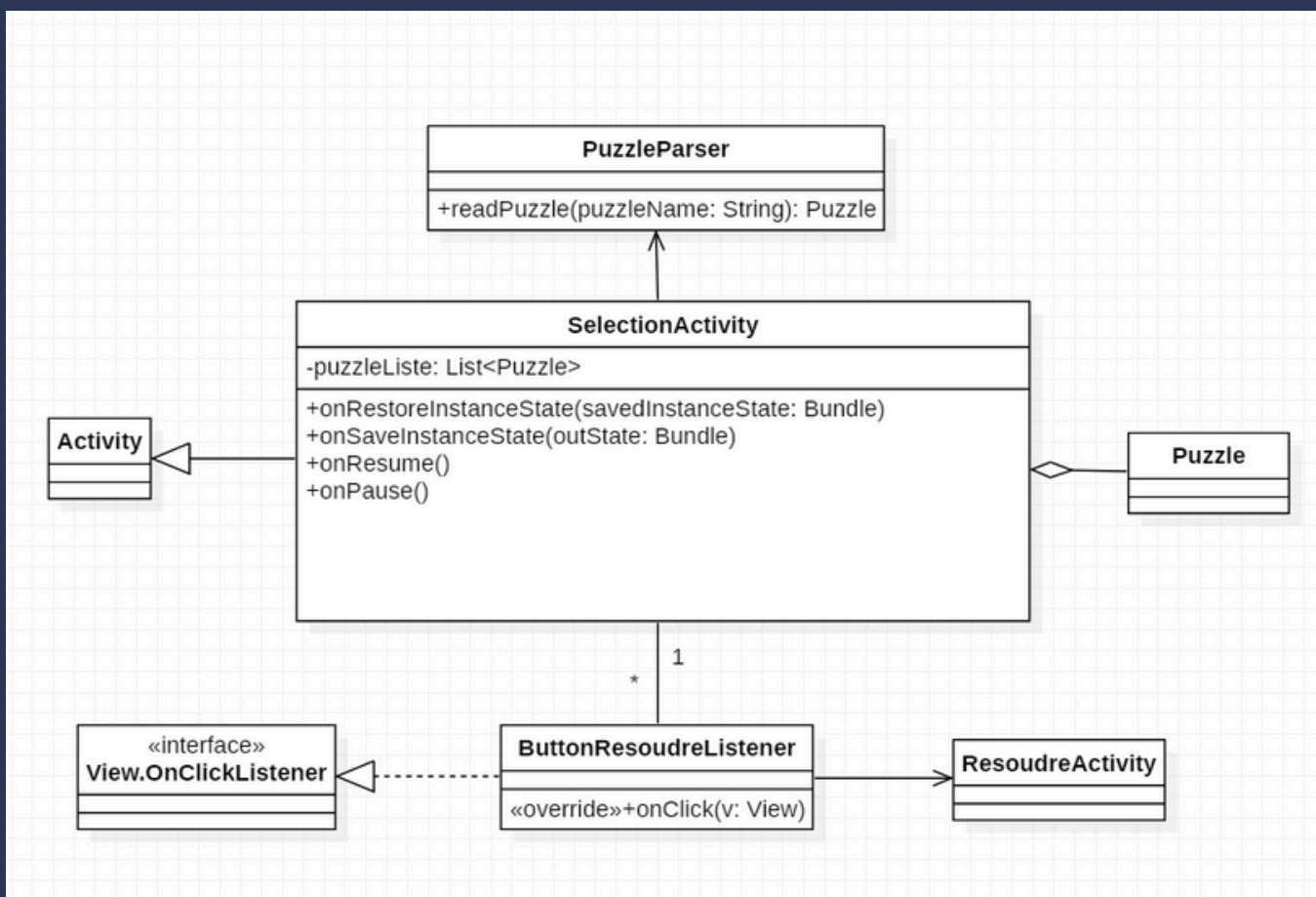
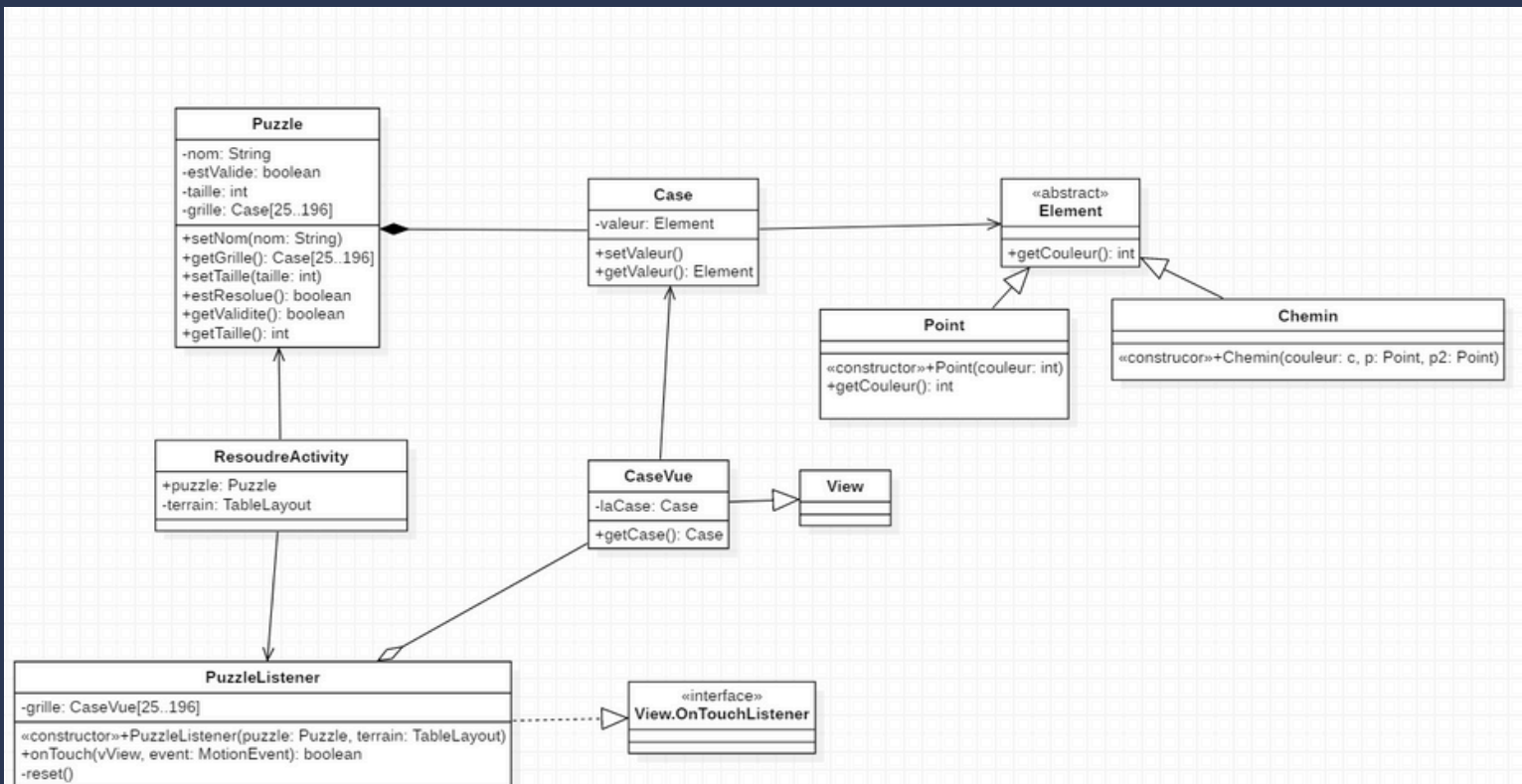




Diagramme de classes - Activité de résolution de puzzle : ResoudreActivity



Gestion des événements tactiles

La gestion des événements tactiles est un aspect crucial de notre application. Nous avons implémenté une logique robuste qui:

1. Détecte le début d'un tracé lorsque l'utilisateur touche un point
 2. Suit le mouvement du doigt et valide les déplacements case par case
 3. Vérifie les contraintes en temps réel (sortie de grille, collision avec un autre chemin)
 4. Finalise le chemin lorsque le point de destination est atteint
- Cette implémentation permet une expérience utilisateur fluide et intuitive, essentielle pour ce type de jeu.



Systeme de persistance

Pour garantir qu'une partie en cours n'est pas perdue lorsque l'application est mise en pause, nous utilisons les méthodes `onSaveInstanceState()` et `onRestoreInstanceState()` pour sauvegarder et restaurer l'état du jeu.

Algorithme de résolution

A chaque fois qu'on créer un chemin valide le programme vérifie si aucune cases de la grille est vide si oui alors le Puzzle est résolut

Pour optimiser les performances, nous maintenons une structure de données qui garde trace des cellules occupées, évitant ainsi de parcourir toute la grille à chaque vérification.



Conclusion

Le développement de cette application Android nous a permis de mettre en pratique de nombreuses compétences techniques et de conception logicielle.

Parmi les défis relevés, la gestion des interactions tactiles a été particulièrement complexe, notamment pour valider les chemins en temps réel et gérer les cas d'erreur de manière élégante. La conception d'une architecture logicielle modulaire et extensible a également nécessité une réflexion approfondie.

Si nous devons poursuivre ce projet, plusieurs améliorations pourraient être envisagées:

- Ajout d'un système de niveaux de difficulté
- Implémentation d'un générateur de puzzles aléatoires
- Création d'un mode "indice" pour aider les joueurs bloqués
- Amélioration des performances pour les puzzles de grande taille



Conclusions personnelles

Izanic Hugo

Ce projet m'a permis d'approfondir mes connaissances en développement Android et en manipulation des fichiers XML. J'ai particulièrement apprécié la mise en œuvre de la gestion des événements tactiles et la logique derrière la détection de la résolution du puzzle. L'utilisation de Java et des classes vues en cours a été un défi stimulant, m'obligeant à structurer proprement mon code et à optimiser la gestion des chemins. Cette expérience m'a conforté dans mon intérêt pour le développement mobile et m'a apporté une meilleure maîtrise des bonnes pratiques en programmation orientée objet.

Gallego Julian

Ce projet m'a permis de développer ma capacité à m'adapter aux imprévus et à trouver des solutions face aux difficultés techniques. La gestion des erreurs dans les fichiers XML et l'optimisation de l'interface ont nécessité une réflexion approfondie et une approche méthodique. J'ai également appris à mieux organiser mon travail pour respecter les délais imposés, ce qui est une compétence essentielle dans le développement logiciel. Cette expérience m'a donné plus de confiance en ma capacité à gérer des projets informatiques complexes.

AISSI Jude-Christ

L'un des aspects les plus enrichissants de ce projet a été la réflexion sur l'expérience utilisateur. Assurer une interface fluide et intuitive tout en garantissant une bonne jouabilité a été un véritable défi. Si nous avions plus de temps, nous pourrions encore améliorer l'ergonomie en ajoutant des animations ou des effets visuels plus dynamiques. Ce projet m'a fait prendre conscience de l'importance du design et de l'accessibilité dans une application, et cela m'encourage à approfondir ces aspects à l'avenir.