

BUT2 Informatique
Groupe 3

Rapport

Relier les points

Clément JANNAIRE
Clémence DUCREUX

2024-2025

Sommaire

1.	Introduction	2
2.	Fonctionnalités	3
3.	Structure programme	9
4.	Diagramme de classe	13
5.	Algorithme	14
6.	Conclusion	17

Introduction



Le projet "Relier les points" est un jeu de réflexion où le joueur doit relier des points de couleur identique en traçant des chemins sur une grille. Chaque chemin ne peut pas croiser un autre, l'on doit couvrir l'ensemble de la grille pour compléter le niveau. Le jeu s'inspire librement des mécaniques de Flow Free et Dot Link.

Le développement a été réalisé avec Android Studio, en utilisant Java et les principes de la programmation orientée objet pour assurer une architecture modulaire et maintenable. L'interface utilisateur a été conçue en XML pour une expérience fluide et intuitive. Le projet suit une structure claire avec une séparation entre la logique métier et l'affichage. Le jeu a été testé sur tablettes et téléphones pour garantir une compatibilité optimale sur différents formats d'écran.

Fonctionnalités

Page d'accueil (Menu principal) :

Sélection de niveau

La sélection de niveau permet au joueur de choisir le puzzle qu'il souhaite résoudre. Une liste s'affiche avec tous les niveaux disponibles, et le joueur peut en sélectionner un pour commencer à jouer. Les niveaux valides sont affichés normalement, tandis que ceux comportant des erreurs sont grisés pour indiquer qu'ils ne sont pas jouables. Cela offre une navigation simple et claire pour accéder aux différents défis du jeu.



Validation des fichiers niveaux

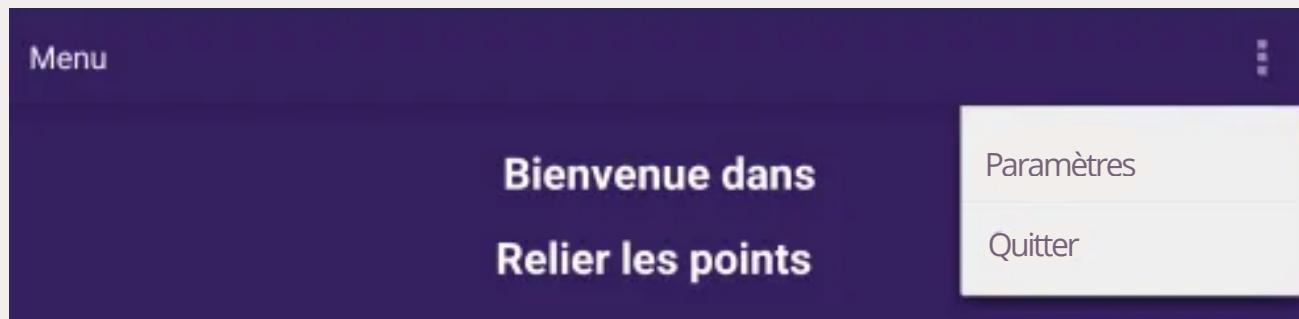
Un fichier de niveau valide est un fichier XML contenant une balise `<puzzle>` avec un attribut `size` entre 5 et 14 inclus, et un attribut `nom` facultatif mais jamais vide. Le fichier doit inclure au moins une balise `<paire>`, chacune contenant exactement deux balises `<point>`, avec des attributs `colonne` et `ligne` dont les valeurs sont des entiers ≥ 0 et $< \text{size}$. Chaque point doit avoir des coordonnées uniques dans tout le fichier. La syntaxe des balises et attributs (`puzzle`, `paire`, `point`, `size`, `nom`, `colonne`, `ligne`) doit être strictement respectée, sans attribut vide, avec uniquement des valeurs entières valides. Les commentaires XML sont autorisés. L'ordre des paires n'affecte pas la logique mais peut définir leur couleur en jeu.

Accès aux options

L'accès aux paramètres permet au joueur de personnaliser certains aspects du jeu selon ses préférences. Depuis le menu principal, un bouton dans la barre du haut ouvre une page dédiée où le joueur peut activer ou désactiver des fonctionnalités comme le mode achromate. Cela rend l'expérience plus confortable et adaptée aux besoins de chacun, tout en restant facile d'accès.

Quitter le jeu

Quitter le jeu se fait directement depuis le menu principal grâce à un bouton prévu à cet effet. Lorsque le joueur le sélectionne, l'application se ferme complètement et toutes les activités en cours sont arrêtées. C'est un moyen simple et rapide de mettre fin à la session de jeu.



Menu des paramètres :

Affichage des règles

L'affichage des règles permet au joueur de comprendre rapidement le but du jeu et comment y jouer. Présentées de manière claire et concise, les règles expliquent notamment qu'il faut relier les points de même couleur sans croiser les lignes et remplir toute la grille. Cela aide les nouveaux joueurs à prendre en main le jeu facilement tout en rappelant les bases aux habitués.

RÈGLES DU JEU

Règles

- Reliez les points de même couleur.
- Tous les chemins doivent être utilisés.
- Vous ne pouvez pas traverser un autre chemin.
- Complétez la grille pour gagner !

Affichage des commandes

L'affichage des commandes indique au joueur comment interagir avec le jeu. Il explique les actions possibles, comme toucher un point pour commencer un tracé, glisser pour créer un chemin, ou appuyer sur le bouton "Réinitialiser" pour effacer un tracé en cours. Ces indications permettent de jouer intuitivement, sans confusion, et assurent une prise en main rapide et fluide.

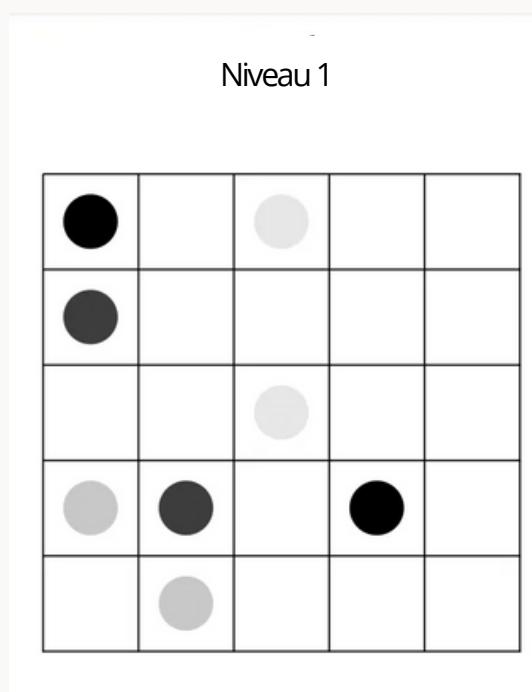
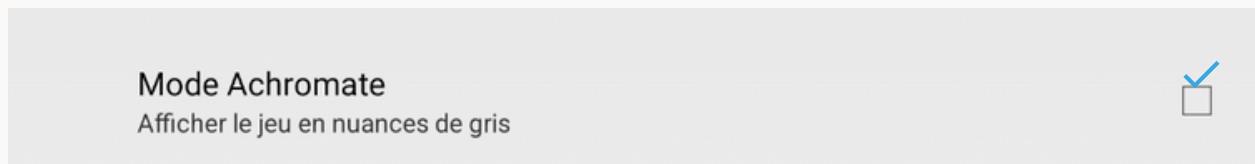
COMMANDES

Comment jouer

- Touchez un point pour commencer un chemin.
- Glissez votre doigt pour tracer un chemin.
- Relâchez pour terminer le tracé.
- Cliquez sur 'Réinitialiser' pour recommencer.

Activation du mode achromate

Le mode achromate est un paramètre accessible depuis le menu des paramètres qui, lorsqu'elle est activée, remplace les couleurs des points du jeu par des nuances de gris. Une fois activé, ce mode reste actif de façon globale pour tous les niveaux, et ce tant que l'utilisateur ne le désactive pas manuellement. Il peut être activé ou désactivé à tout moment via le menu des paramètres, sans limitation, et son effet s'applique immédiatement sur tous les éléments colorés du jeu.



Retour à la page d'accueil

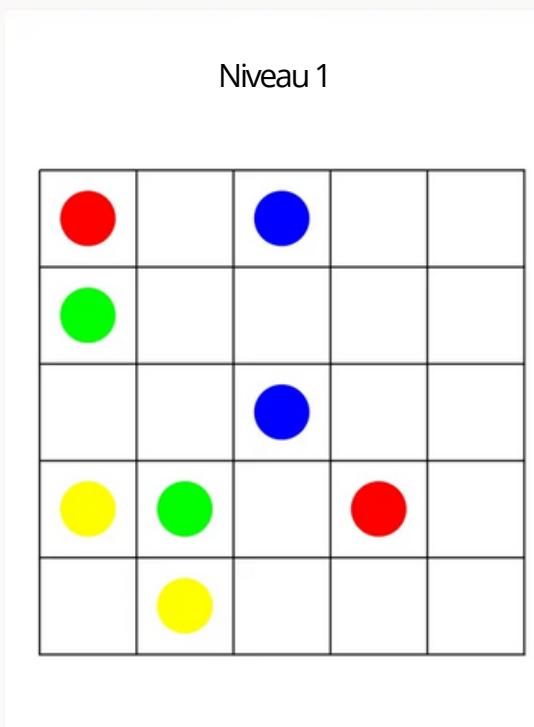
Un bouton (la flèche) permet de revenir au menu principal.



Interface en jeu :

Grille de jeu

La grille de jeu est l'espace principal où se déroule la partie. Elle se compose de cases disposées en lignes et colonnes, dans lesquelles sont placés les points colorés à relier. Le joueur doit tracer des chemins entre les points de même couleur sans les croiser, tout en remplissant entièrement la grille. C'est le cœur du jeu, où réflexion et logique sont mises à l'épreuve.



Bouton "Réinitialiser"

Supprime tous les traits présents sur la grille (remise à zéro de la progression) et est bloqué quand l'on a gagner la partie.

Réinitialiser

Suppression de traits individuels

La suppression de traits individuels permet au joueur d'effacer un chemin sans devoir recommencer toute la grille. Il suffit de retoucher le point de départ d'un trait déjà tracé pour le supprimer automatiquement. Cela offre plus de souplesse dans le jeu et permet de corriger facilement une erreur sans perturber le reste de la solution.

Détection de victoire

Lorsqu'un niveau est complété correctement, un message de victoire s'affiche. Cette à dire chaque paires de points est relié et il n'y a pas de case vide.

Blocage des actions après la victoire

Une fois la victoire détectée, il devient impossible de supprimer ou modifier les traits. L'unique action restante est de retourner au menu.

Structure programme

MainActivity.java :

C'est la page principale où le joueur joue réellement.

- **onCreate()** : initialise le niveau sélectionné et la grille.
- **onResume()** : recharge le niveau si on revient dessus.
- **onSaveInstanceState() / restoreGameState()** : sauvegarde/restaure le tracé quand on change d'écran.
- **onOptionsItemSelected()** : gère le bouton retour dans la barre du haut.

MenuActivity.java :

C'est le menu d'accueil du jeu.

- **onCreate()** : affiche la liste des niveaux disponibles.
- **getLevelsFromAssets()** : lit les niveaux depuis les fichiers XML.
- **getLevelNameFromFile()** : récupère le nom affiché pour chaque niveau.
- **isPuzzleValid()** : vérifie que le niveau est jouable.
- **onCreateOptionsMenu()** : ajoute le menu "Paramètres" et "Quitter".
- **onOptionsItemSelected()** : réagit quand le joueur clique sur un de ces boutons.

OptionsActivity.java :

C'est la page qui permet de modifier les paramètres du jeu.

- **onCreate()** : affiche les paramètres disponibles.
- **onCreateOptionsMenu()** : désactive le menu (inutile ici).
- **onOptionsItemSelected()** : permet de revenir en arrière.
- **onPreferenceChange()** : agit quand un paramètre change (ex : mode achromate).

GameView.java :

C'est la classe qui dessine la grille et gère les gestes du joueur.

- **onDraw()** : dessine la grille, les points et les tracés.
- **onTouchEvent()** : détecte les mouvements du doigt (toucher, tracer).
- **resetGame()** : efface tous les tracés.
- **setAchromateMode()** : active/désactive les couleurs en niveaux de gris.
- **isValidMove()** : empêche les mouvements en diagonale ou hors de la grille.
- **isCorrectEndPoint()** : vérifie qu'on arrive bien au bon point.
- **isAlreadyInCurrentPath() / isPathOccupied()** : évite de repasser sur une case déjà utilisée.

GameLogic.java :

C'est le cœur du jeu, où les règles sont appliquées.

- **loadLevelFromAssets()** : charge les points et la grille depuis un fichier.
- **savePath() / removePath()** : sauvegarde ou efface un tracé.
- **isPathExists() / isPathStartingAt()** : vérifie l'existence d'un chemin.
- **getPointAt()** : récupère un point à une position donnée.
- **checkWin()** : vérifie si tous les points sont reliés correctement.
- **resetGrid()** : efface tous les chemins.

FlowPoint.java :

Ce fichier représente un point coloré à relier.

- Contient les coordonnées et la couleur du point.
- Méthodes : **getX()**, **getY()**, **getColorIndex()**.

assets/puzzles/xxx.xml :

- Chaque fichier contient un niveau.
- Un nom et une taille de grille.
- Des paires de points à relier.

activity_main.xml :

- Affiche la grille de jeu, le titre et le bouton "Réinitialiser".

activity_main_menu.xml :

- Affiche le menu principal, le titre du jeu et la liste des niveaux.

activity_options.xml :

- Affiche les paramètres du joueur.

preferences.xml :

- Définit les paramètres disponibles dans la page des paramètres (ex : activer le mode achromate).

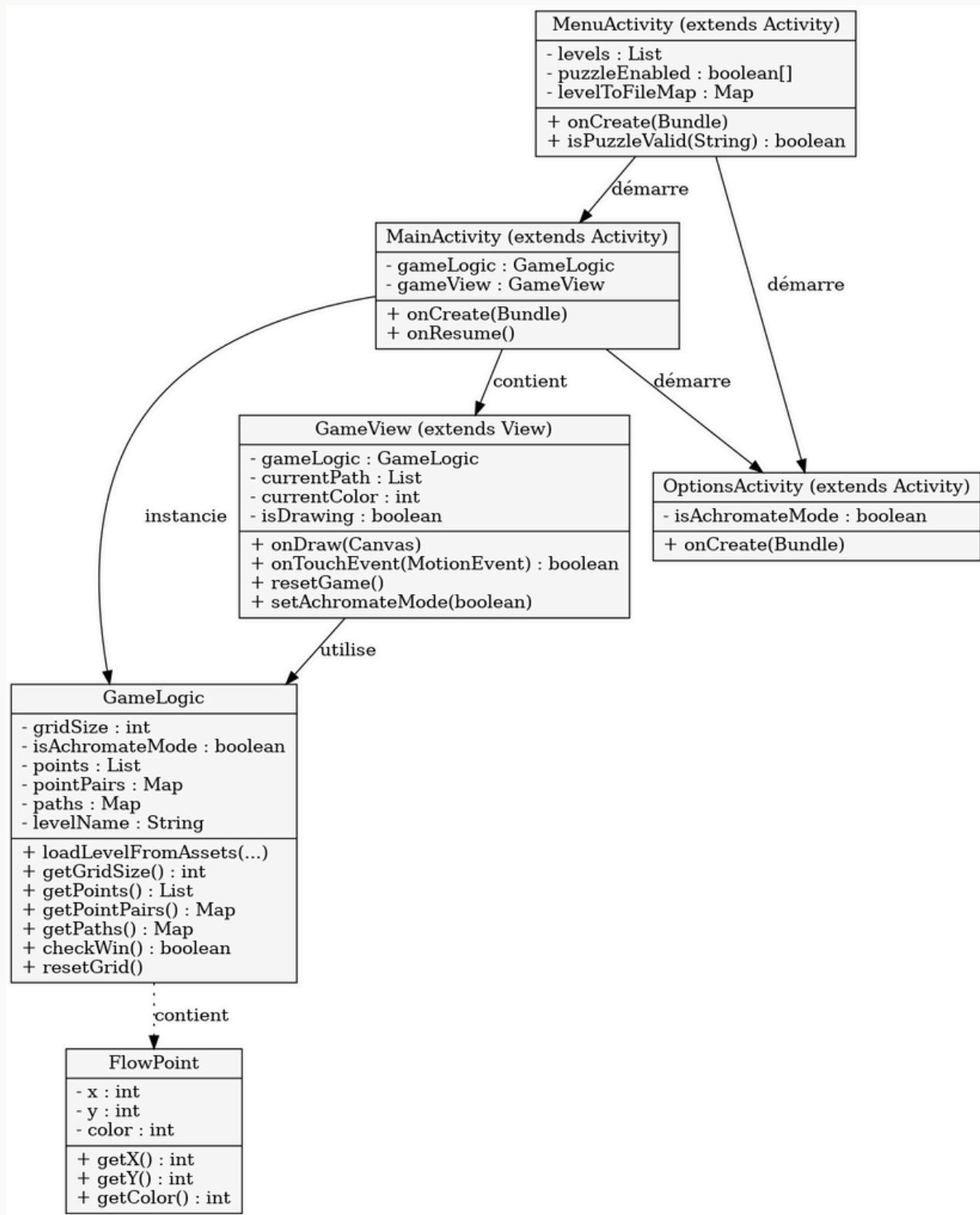
AndroidManifest.xml :

- Indique les pages du jeu, les thèmes, et les infos globales de l'application.

styles.xml :

- Définit les couleurs, les styles de menu, les thèmes (comme le violet).

Diagramme de classe



Algorithme

Algorithme vérification de la victoire

La méthode checkWin() est chargée de déterminer si le joueur a résolu le puzzle en respectant l'ensemble des règles. Elle s'appuie sur deux conditions fondamentales :

- Chaque paire de points de même couleur doit être correctement reliée par un chemin continu.
- Toutes les cases de la grille doivent être entièrement utilisées par ces chemins.

Initialisation de la grille de suivi

La méthode commence par récupérer la taille de la grille via getGridSize(), puis crée une matrice booléenne `usedCells` représentant les cases de la grille. Cette matrice servira à vérifier qu'aucune case n'est laissée vide à la fin.

Vérification des chemins

Le programme parcourt chaque entrée de paths, une structure de type Map<Integer, List<int[]>, où chaque clé représente une couleur (identifiée ici par un entier), et chaque valeur est une liste de coordonnées correspondant au chemin tracé par le joueur pour cette couleur.

Pour chaque chemin :

- Il vérifie que le chemin contient au moins deux points. Si ce n'est pas le cas, la méthode retourne immédiatement false.
- Il récupère les coordonnées du premier et du dernier point du chemin, censés correspondre aux deux points de départ et d'arrivée de la paire de couleur.
- Il vérifie que ces deux points sont bien des FlowPoints valides, et surtout qu'ils ont la même couleur.
- Si la couleur ne correspond pas, ou si un des points est invalide, la méthode retourne false.

Ensuite, pour chaque case traversée par ce chemin, la méthode marque la case correspondante comme utilisée dans la matrice `usedCells`.

Vérification de l'utilisation complète de la grille

Une fois tous les chemins parcourus, la méthode inspecte chaque case de la grille. Si au moins une case n'a pas été utilisée, cela signifie qu'un chemin ne couvre pas toute la grille (ce qui est requis pour gagner), et donc la méthode retourne false.

Victoire

Si toutes les vérifications sont passées avec succès (tous les chemins relient bien les bons points, et aucune case n'est vide), la méthode retourne true, indiquant que le joueur a résolu le niveau.

Conclusion

Clément :

Réaliser ce jeu a représenté un vrai défi, car il a fallu penser à la fois à l'ergonomie, à la robustesse du code et à l'expérience utilisateur. Chaque fonctionnalité, comme la gestion des couleurs ou la validation des niveaux, a exigé réflexion et rigueur. J'ai également pris conscience de l'importance de tester fréquemment pour éviter des erreurs subtiles.

Clémence :

Ce projet m'a permis de consolider mes compétences en développement Android natif, notamment dans la gestion de l'interface utilisateur, la lecture de fichiers XML, et l'implémentation d'une logique de jeu structurée. J'ai pu mieux comprendre l'intérêt de la modularité du code, et apprendre à corriger des bugs complexes liés à l'interaction tactile. Cette expérience m'a rendu plus autonome et confiant dans mes capacités à gérer un projet mobile complet de bout en bout. J'ai beaucoup apprécié ce projet.