

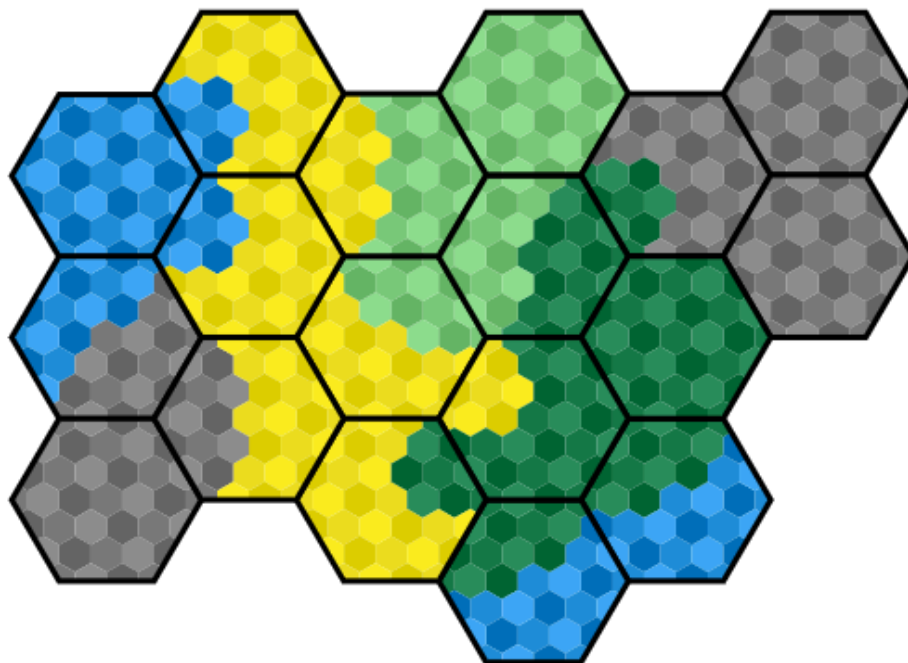
Rapport SAE31 2024 Paysages

SOMMAIRE :

Introduction.....	1
Description des fonctionnalités du programme ..	2-3-4
Présentation de la structure de chaque programme..	5
Analyse de l'algorithme qui identifie les poches.....	6-7
Explication pour chaque écran.....	8
Les scores.....	9
Conclusions personnelles.....	10

INTRODUCTION :

Le projet consiste à développer un jeu de placement de tuiles hexagonales inspiré du jeu Dorfromantik, où le joueur doit créer un paysage harmonieux en disposant des tuiles représentant différents types de terrains (mer, champ, pré, forêt, montagne). Le but est de maximiser le score en connectant les tuiles adjacentes ayant le même type de terrain, formant ainsi des "poches" de terrains connectés. Le joueur place une tuile à chaque tour, en choisissant la position et l'orientation de la tuile, avec pour contrainte que la nouvelle tuile doit être adjacente à une tuile déjà posée. La partie se termine après le placement de 50 tuiles, et le score est calculé en fonction de la taille des poches de terrains connectés. En plus du jeu local, le projet comprend un serveur de base de données qui stocke des séries de tuiles et les scores des joueurs.

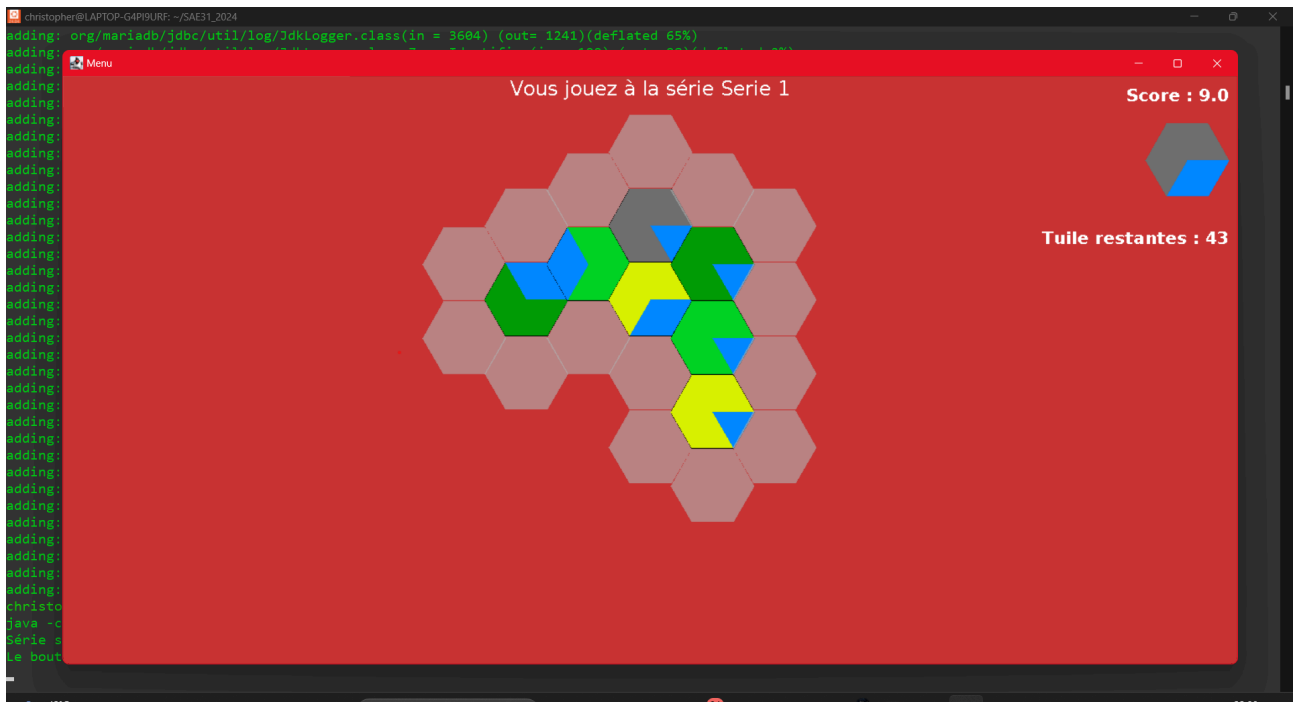


DESCRIPTION FONCTIONNALITÉS :



Le joueur a la possibilité de sélectionner une série parmi les trois disponibles grâce à un menu déroulant, lui permettant de personnaliser son expérience de jeu selon ses préférences. Une fois la série choisie, il peut lancer la partie en cliquant sur le bouton 'Jouer', qui l'amènera directement dans l'univers du jeu. S'il a besoin de se familiariser avec les règles ou de les revoir avant de commencer, un bouton 'Règles' est également disponible. Cela assure que même les nouveaux joueurs peuvent se lancer sans difficulté, en comprenant bien les bases du jeu et les particularités de chaque série

Le Jeu:



Le joueur a la possibilité de visualiser facilement les emplacements disponibles pour poser ses tuiles, qui sont indiqués en gris sur le terrain. Cette aide visuelle permet de repérer rapidement les endroits possibles pour un placement stratégique, favorisant une prise de décision rapide et intuitive. En un simple clic, le joueur peut déposer une tuile sur l'emplacement sélectionné. La prochaine tuile à placer est affichée en haut à droite de l'écran, accompagnée d'une option de rotation activable en appuyant sur la touche '**Espace**'. Cela permet au joueur d'ajuster l'orientation de la tuile pour un placement optimal. Cet emplacement en haut à droite affiche également le nombre de tuiles restantes, offrant un suivi en temps réel de la progression vers la fin de la partie.

En plus de cela, le joueur peut naviguer dans la fenêtre de jeu en utilisant les **touches directionnelles**, lui permettant de se déplacer aisément sur le terrain et de visualiser l'ensemble des zones disponibles pour poser ses tuiles. Cette combinaison d'options visuelles et de commandes simples rend l'expérience de jeu

fluide et immersive, en donnant au joueur un contrôle total sur la stratégie et la gestion de l'espace.

LES SCORES :

Pour permettre au joueur d'avoir une meilleure représentation de son niveau et de son score, nous avons mis en place un système de rang qui valorise ses performances tout au long du jeu. Ce système de rang attribue une position au joueur en fonction de son score, ce qui lui permet de se situer par rapport aux autres participants et de mesurer ses progrès au fil des parties.

De plus, le joueur a la possibilité de consulter le classement des scores des autres joueurs, qui est organisé de manière claire et intuitive, du plus élevé au plus bas. Cette fonctionnalité offre au joueur une vue d'ensemble sur les performances de ses concurrents et lui permet de visualiser l'écart de performance qui le sépare des autres compétiteurs. En ayant accès à cette information, le joueur peut identifier les stratégies à adopter pour améliorer son propre score, se fixer des objectifs plus ambitieux et se motiver à repousser ses limites lors des prochaines parties.

Pour améliorer l'expérience utilisateur, un bouton 'Rejouer' est également disponible, permettant au joueur de retourner au menu principal pour commencer une nouvelle partie en un clic. En outre, un bouton 'Quitter' est présent pour permettre au joueur de quitter le jeu facilement à tout moment.



STRUCTURE DU PROGRAMME :

Pour ce projet nous avons décidé de diviser notre code en suivant le principe de MVP (modèle, vue, contrôleur). Voici les différents fichiers sources :

Pour les vues :

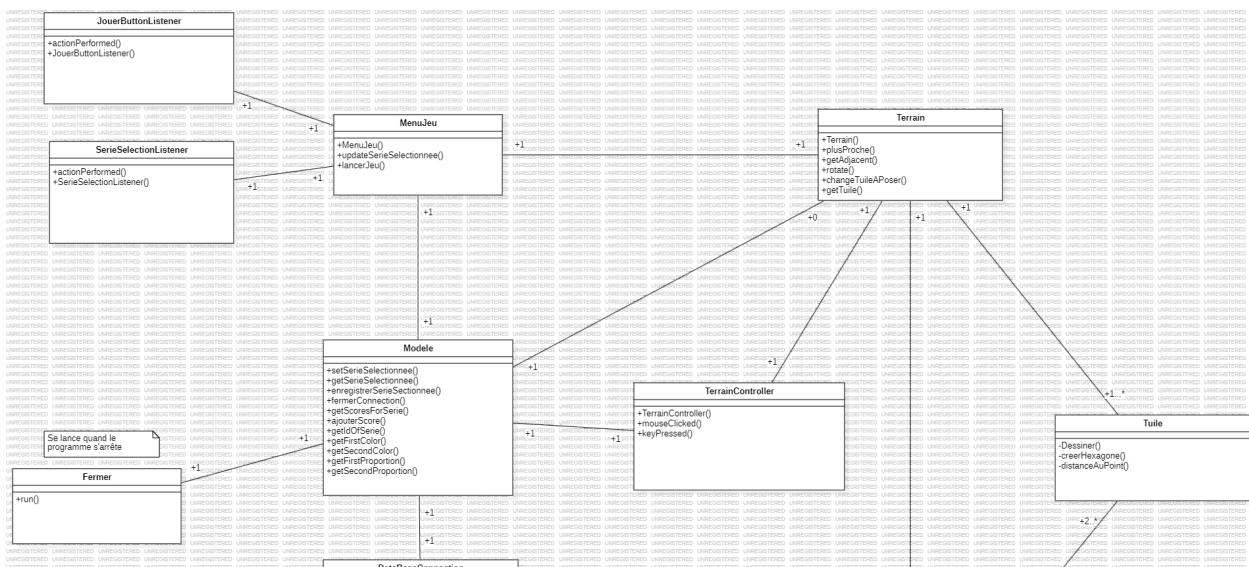
- Tuile, qui est une classe représentant les tuiles
- Menu jeu, qui permet de jongler entre les différents écrans
- Terrain, qui est utilisé pour faire apparaître les tuiles, et les gérer
- Fin qui affiche le menu de fin avec le score du joueur et permet de le comparer avec les autres joueurs

Pour les controllers :

- JouerButtonListener, qui gère le bouton «Jouer »
- SerieSelectionListener, qui gère le menu déroulant permettant de choisir sa série
- TerrainController, qui gère les événements en jeu, notamment les événements avec la souris et les flèches directionnelles
- Ferme, qui est une classe appelée quand la fenêtre se ferme afin de fermer la connection à la base de données

Pour les modèles :

- Models, qui gère les requêtes SQL
- ConnectionDataBase, qui permet de se connecter à la base de données



ALGORITHME DES POCHEs :

Pour l'algorithme des poches, nous avons une classe Poche.java. Celle-ci contient les informations suivantes : une liste des numéros des tuiles qui la composent, sa couleur et un numéro d'identification.

Les tuiles stockent dans un tableau de 2 entiers les identifiants des poches auxquelles elles appartiennent. Autrement dit, $[0, 0]$ équivaut à une tuile qui n'est dans aucune des poches. 2 entiers sont possibles car une tuile ne peut contenir que 2 couleurs maximum.

Terrain.java se souvient donc, en plus de la liste des tuiles, de la liste des poches.

Pour déterminer si une poche s'est créée, on récupère la tuile qu'on vient de poser. Avec cette tuile, on récupère 2 éléments : ses tuiles adjacentes et son tableau de 6 couleurs qui correspondent à ses 6 côtés.

On crée une variable de type Tuile pour chaque tuile adjacente afin de s'y retrouver dans la manipulation du code.

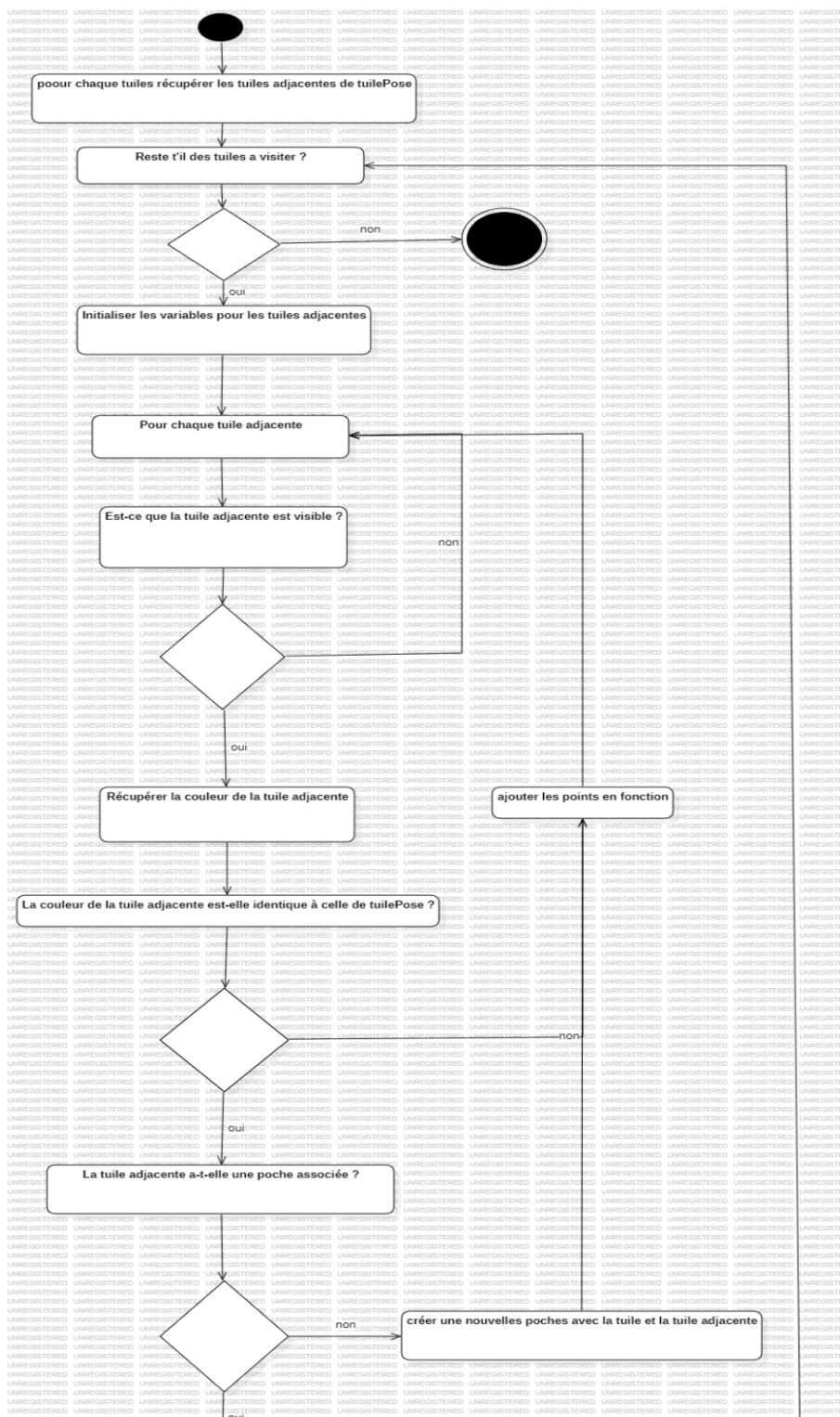
Pour chaque variable de type Tuile, autrement dit, pour chaque tuile adjacente on procède de la manière suivante :

- Si la tuile adjacente est visible, on récupère la couleur adjacente à la tuile qu'on vient de poser. Sinon, on n'en fait rien.
- On vérifie si la couleur de la tuile adjacente correspond bien à la couleur de la tuile qu'on vient de poser. Si c'est le cas, alors on récupère les identifiants des poches stockés dans la tuile adjacente.
- Si le tableau des identifiants est vide ($[0, 0]$), alors cela veut dire que la tuile n'est dans aucune poche et donc qu'on peut créer une nouvelle poche en lui donnant les tuiles de la poche, sa couleur et un entier d'identification. Sinon, on boucle sur les poches stockées par le terrain.
- On récupère la poche qui a pour identification la première valeur de mon tableau d'ID et on vérifie sa couleur, s'il s'agit de la couleur de la tuile qu'on vient de poser, alors on ajoute la tuile qu'on vient de poser

au tableau de la poche. Sinon, on vérifie la deuxième valeur d'identification.

- Si après avoir vérifier le tableau d'identification, on ne trouve pas de poche avec la bonne couleur (normalement cela se produit si une tuile est dans qu'une seule poche et donc que le tableau est de type : $[n, 0]$ où $n > 0$), alors on crée une nouvelle poche.

Le calcul du score est maintenant simple vu qu'on a besoin de simplement boucler sur la liste des poches stockées par le terrain, d'y récupérer la taille et d'ajouter la taille exposant 2 au score. On boucle sur toutes les tuiles et on ajoute $1^{\wedge}2$ pour toutes les tuiles dont le tableau d'identification est de type $[0, 0]$.



EXPLICATION ÉCRAN

Menu Accueil : Sur l'écran d'accueil, nous avons optimisé l'ergonomie en divisant la fenêtre en deux sections pour rendre l'interface claire et intuitive. Dans le tiers gauche, un menu rouge foncé attire l'attention et propose un accès immédiat aux options de jeu. Ce menu est marqué par un titre 'Nouvelle Partie', stylisé comme un tagueur pour donner une touche moderne et dynamique et en dessous le bouton Jouer en noir de fond et orange pour le style . En dessous, un bouton noir avec une écriture orange 'Series' permet de choisir une série de tuiles pour commencer à jouer. Ce choix est facilité par un bouton scrollpane, permettant à l'utilisateur de faire défiler les series sans encombrer l'espace.

Les deux tiers restants de l'écran affichent une image de fond inspirée du jeu **Dorfromantik**, créant une ambiance immersive. Un rectangle placé au centre de cette section présente les règles du jeu de façon accessible et lisible, permettant aux nouveaux utilisateurs de se familiariser rapidement avec les règles avant de commencer. Cette mise en page assure une distinction visuelle claire entre les options de navigation et le contenu informatif, améliorant ainsi l'expérience utilisateur.

Jeu : Sur l'écran de jeu, nous avons maximisé l'espace pour que le terrain occupe toute la fenêtre, offrant une immersion totale dans le jeu. En haut à droite, un espace compact et bien organisé permet d'afficher des informations essentielles : la tuile suivante, le score actuel, et le nombre de tuiles restantes. Afin de garder une continuité visuelle avec le menu principal, le fond de l'écran reste en rouge foncé, ce qui renforce l'identité visuelle du jeu et évite le gris par défaut de Java. Cette disposition ergonomique permet aux joueurs de suivre facilement leur progression sans quitter des yeux le terrain, offrant ainsi une expérience à la fois cohérente et immersive.

Fin du jeu :

Sur l'écran de fin, le fond reste en rouge foncé pour conserver la cohérence visuelle du jeu. Au centre, le score final du joueur est mis en évidence avec l'intitulé 'Score : '. Juste en dessous, le rang du joueur est affiché de manière claire et valorisante, par exemple : 'Vous êtes au 4^e rang dans la série'. Cette indication de rang permet au joueur de se situer

facilement par rapport aux autres participants, en rendant sa performance plus significative.

En-dessous, une section 'Score des autres joueurs' affiche les positions restantes en blanc, rappelant la couleur de l'écriture tout au long du jeu pour une continuité visuelle et une lecture facile. Initialement, nous avions prévu un bouton pour accéder à un tableau détaillé des scores, comprenant le classement des joueurs, les séries, et les meilleurs scores. Par manque de temps, nous avons implémenté un scrollpane affichant directement les scores des autres joueurs dans l'ordre décroissant, du rang le plus élevé (meilleur score) vers les rangs inférieurs. Cela permet au joueur de consulter rapidement son classement parmi les autres et d'apprécier sa position en un coup d'œil.

Scores :

Dès qu'un joueur termine une partie, il obtient un score qui lui est attribué en fonction de la série à laquelle il a joué. Ce score est ensuite envoyé à notre base de données, où il est enregistré avec les autres performances des joueurs. La base de données est conçue pour stocker les scores de manière structurée, permettant ainsi un tri efficace. Tous les scores sont classés par ordre décroissant, ce qui signifie que le meilleur score de la série est affiché en haut de la liste.

Une fois que le score du joueur est ajouté à la base de données, le système procède à une comparaison avec le meilleur score enregistré pour cette série. Cette comparaison est cruciale car elle permet de déterminer si le joueur a réussi à dépasser le score le plus élevé de cette série. Si le score du joueur est supérieur à celui du meilleur score précédent, cela signifie qu'il a établi un nouveau record, ce qui est une réalisation significative et valorisante.

Parallèlement à cette comparaison, le système détermine également le rang du joueur en fonction de son score. Ce rang indique la position du joueur par rapport à ses adversaires, offrant ainsi une perspective sur sa performance relative. Par exemple, si un joueur obtient un score qui le place au 3ème rang, cela signifie qu'il a surpassé deux autres joueurs mais est toujours derrière le joueur ayant le score le plus élevé. Ce classement est affiché dans la base de données, permettant au joueur de visualiser rapidement sa position parmi les compétiteurs.

Cette approche de classement et de comparaison ne sert pas seulement à positionner les joueurs, mais elle stimule également la motivation et la compétition. Les joueurs sont incités à améliorer leurs performances pour non seulement atteindre de nouveaux scores élevés, mais aussi pour grimper dans le classement et rivaliser avec d'autres joueurs.

CONCLUSIONS PERSONNELLES :

Wilfried :

Personnellement, j'ai beaucoup apprécié travailler sur ce projet (même s'il était plus dur que les précédents je trouve). L'une des principales raisons est que ce projet m'a permis de créer quelque chose de concret et d'aller plus loin que les exercices pratiques habituels. Le fait de travailler en équipe de trois, contrairement aux projets précédents, m'a permis de développer mes compétences en collaboration, car certaines parties du code écrites par mes coéquipiers nécessitent une compréhension approfondie de ma part. De plus, cette expérience m'a incité à être plus rigoureux dans ma manière de coder, afin de produire un code compréhensible non seulement pour moi-même, mais aussi pour mes deux camarades.

Christopher :

Pour ma part, j'ai eu du mal à comprendre le jeu au départ. Une fois avoir bien compris l'ensemble du projet, je l'ai trouvé compliqué mais à chaque étape du jeu, on y arrivait quand même. Cependant, on a malheureusement pas réussi à atteindre nos objectifs, notamment pour le score. Mais tout de même, j'ai aimé travailler sur ce projet, bien que le résultat ne soit pas celui escompté, car cela m'a appris beaucoup de choses et m'a permis d'assimiler quelques notions de programmation en Java.

Abed :

Points Positifs :

Ce projet a été très enrichissant à coder car il différait de mes expériences précédentes, notamment avec le *Snake* en C et le *Sudoku* en Java. Ici, l'objectif n'était plus seulement de programmer une logique de jeu précise, mais de créer un paysage avec des tuiles

hexagonales. Cette approche est particulièrement satisfaisante, car elle offre une grande liberté au joueur dans l'agencement des tuiles, ce qui permet des possibilités variées et encourage la créativité. L'idée de base de construire un paysage apporte un aspect relaxant au jeu, contrairement à des jeux plus orientés vers l'analyse, comme le *Sudoku*.

Points Négatifs :

Cependant, j'ai rencontré des difficultés au cours du développement, notamment en raison d'une compréhension incomplète du sujet au départ. Cette confusion initiale, en particulier au niveau de la gestion des "poches" (probablement les espaces vides ou connexions manquantes dans le placement des tuiles), a compliqué la progression et l'implémentation du jeu. Le manque de temps a également été un facteur limitant, car je n'ai pas pu intégrer toutes les fonctionnalités souhaitées. Cela a été frustrant, car j'aurais voulu approfondir davantage certains aspects du jeu et lui donner plus de profondeur.