



IZANIC HUGO
GALLEGO JULIAN
AISSI JUDE-CHRIST

2024-2025

PROJET JAVA

HEXALANDIA ROMANTIK

Sommaire

Introduction

Fonctionnalités du programme

- Menu principal
- Options
 - Jouer
 - Tab
- Assemblage des tuiles
- Types de terrains et combinaisons possibles

structure du programme

- Diagramme de classes simplifié
- Structure du programme et organisation des packages
- Diagramme d'activité

Améliorations ergonomiques

- Explications des choix d'interface utilisateur
- Mouvement de la caméra

Évaluation des scores

- Tableau de score
- Comparaison des score

Conclusion

Introduction

Dans le cadre de ce projet, nous avons développé un jeu de stratégie inspiré du jeu Dorfromantik. L'idée principale est de permettre au joueur d'assembler des tuiles hexagonales représentant différents terrains en hexagones pour former un paysage cohérent.

Le développement a été réalisé en Java, avec la contrainte de n'utiliser que les outils et bibliothèques inclus dans l'API officielle. Les sources sont hébergées sur le serveur Gitea ([lien](#)) de notre département, où nous avons également suivi l'avancement du projet et les contributions de chaque membre de l'équipe via des commits .

Dans ce rapport, nous allons d'abord présenter les fonctionnalités principales du jeu et son fonctionnement. Nous décrirons ensuite l'architecture du programme à travers un diagramme de classes simplifié, avant de détailler l'algorithme qui permet de calculer les points du joueur en fonction des poches de terrains. Enfin, nous terminerons par une analyse de l'interface utilisateur et la manière dont les scores sont comparés entre les joueurs.

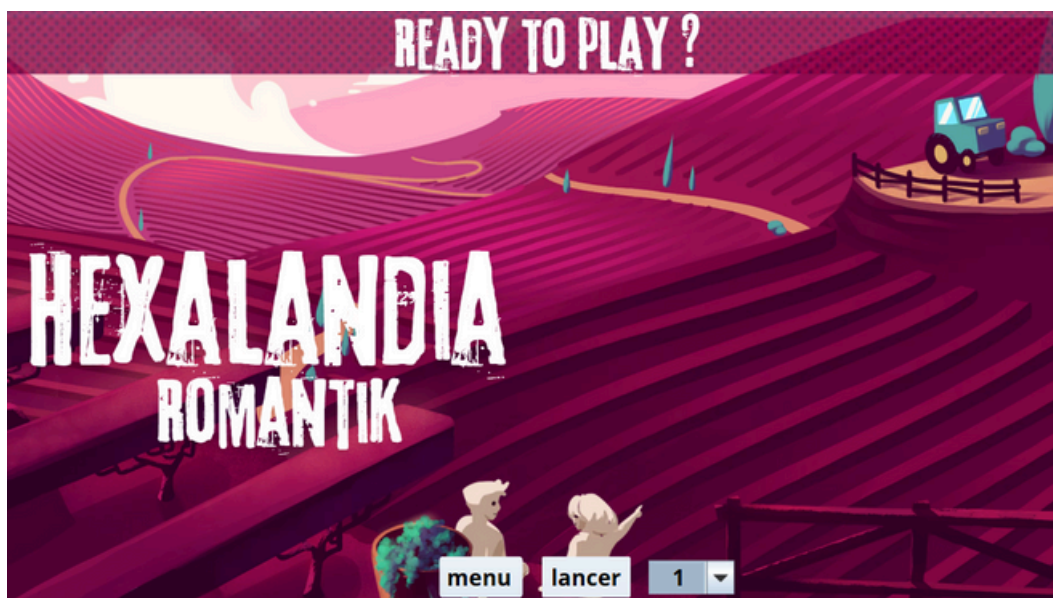
Fonctionnalités du programme

Menu Principal

Lors du démarrage du jeu, l'utilisateur accède à un menu.

Ce menu lui offre trois options principales :

1. **Jeu(Jouer)** : L'utilisateur peut commencer une nouvelle partie en accédant à l'écran de jeu, où il pourra lancer une partie avec le numéro de série et ensuite placer des tuiles et former un paysage.
2. **tab(Tableau de score)** : Cette option permet de consulter les scores des parties précédentes, affichés de manière anonyme. Cela permet à l'utilisateur de comparer ses performances avec celles des autres joueurs ayant joué avec la même série de tuiles.
3. **Quitter** : Cette option permet tout simplement à l'utilisateur de quitter le jeu.



ASSEMBLAGE DES TUILES

Après avoir appuyé sur le bouton "Lancer", l'utilisateur accède à l'interface de jeu où il peut commencer à assembler des tuiles pour construire un paysage harmonieux. Le but de cette phase est d'assembler des tuiles hexagonales représentant différents types de terrains (mer, champ, forêt, montagne, etc.) afin de créer des zones cohérentes et maximiser son score.

À chaque tour, une nouvelle tuile est révélée au joueur. Le joueur doit alors décider où la placer et dans quelle orientation pour qu'elle s'imbrique parfaitement avec les tuiles déjà posées, en tenant compte des types de terrain adjacents. Pour offrir davantage de flexibilité stratégique, nous avons ajouté une fonctionnalité permettant au joueur de pivoter la tuile avant de la placer. Cela lui permet d'aligner les terrains de façon optimale et d'augmenter les chances de créer de grandes poches de terrain homogène, qui rapportent plus de points.

Le joueur peut se déplacer de manière indéfinie sur la tuile, c'est à dire de manière infinie puisqu'il n'y a pas de bords.

TYPES DE TERRAINS ET COMBINAISONS POSSIBLES

Comme nous l'avons mentionné précédemment, dans le jeu, le joueur peut placer des tuiles de différents types de terrains : forêt, mer, champ, pré, et montagne. Chaque tuile peut contenir un seul type de terrain ou une combinaison de deux terrains. Par exemple, une tuile peut avoir un côté forêt et un côté champ, ou encore être entièrement composée de mer.

Structure du programme

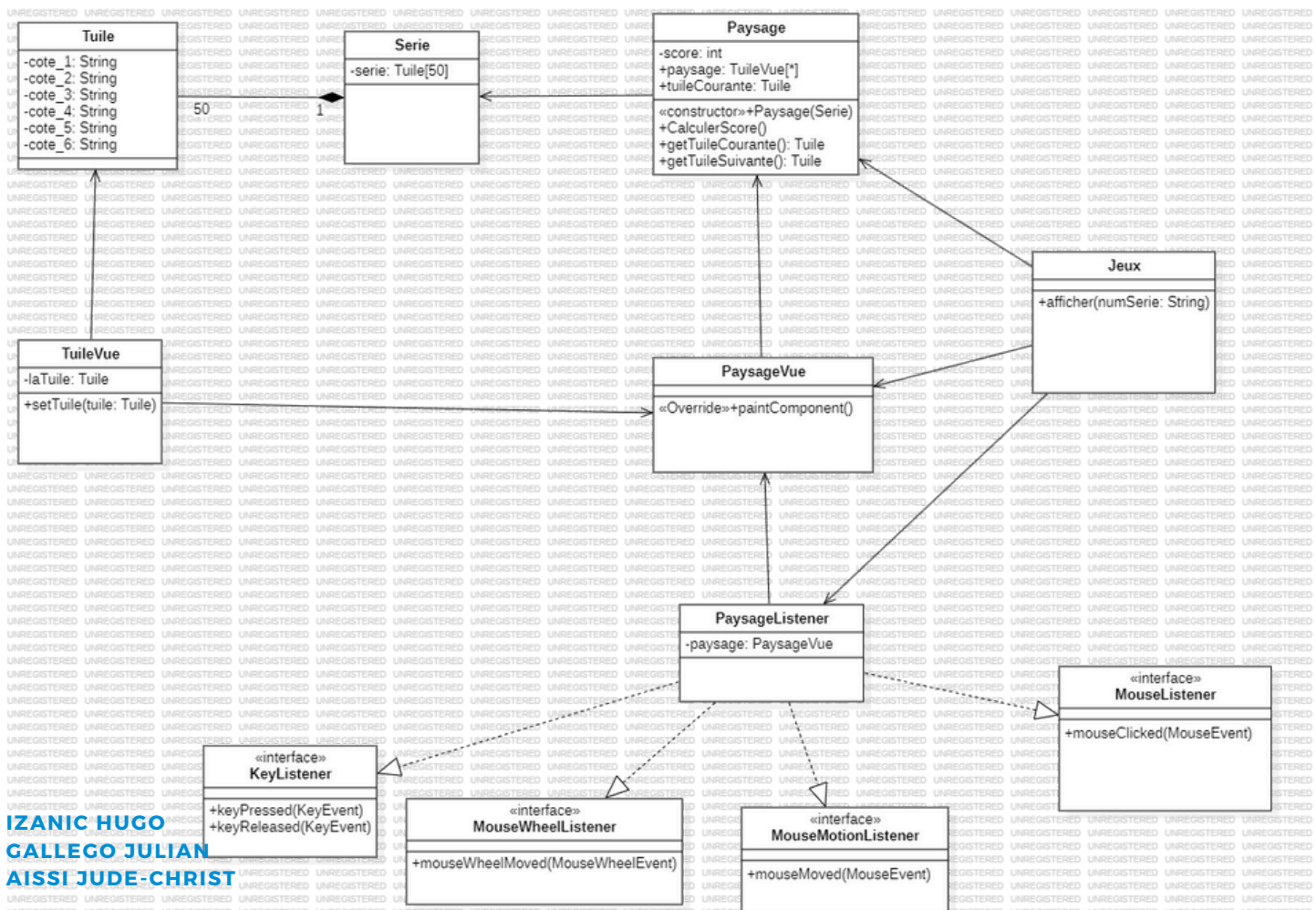
Dans notre base de données, nous avons créé deux tables principales pour organiser les informations de manière efficace :

- Series (num, objet) : Cette table stocke les différentes séries de tuiles disponibles pour les parties. Le champ objet permet de conserver les séries sous forme sérialisée, facilitant leur réutilisation dans le programme.
- Scores (numSerie, points) : Cette table enregistre les scores obtenus par les joueurs pour chaque série. Le champ numSerie établit une correspondance avec la table Series, afin de lier les scores à des séries spécifiques.

Notre programme est conçu pour faciliter la gestion des étapes de compilation, de documentation et d'exécution grâce à des commandes Makefile dédiées :

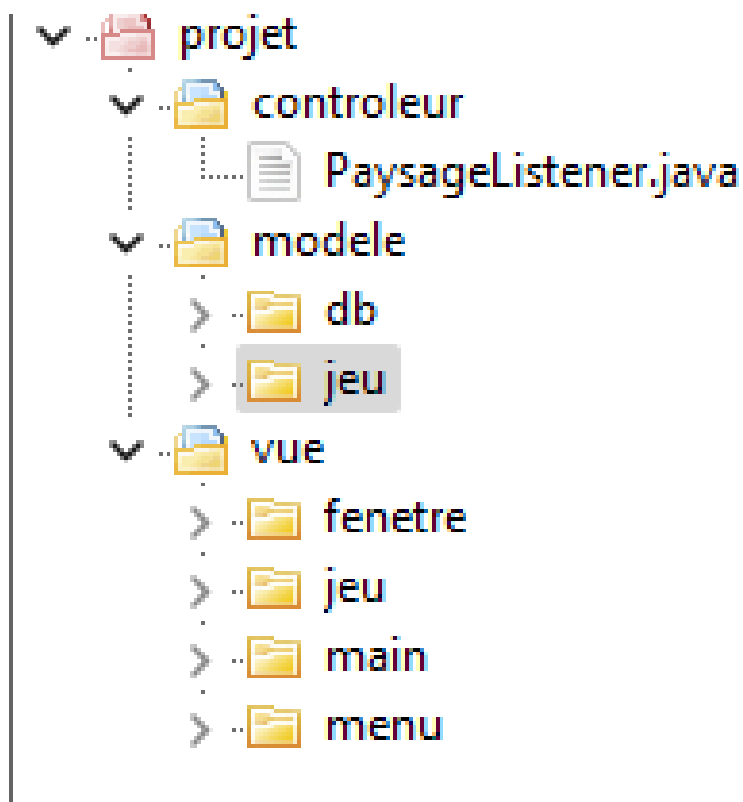
- make javadoc : Génère automatiquement la documentation du code sous forme de fichiers HTML, en utilisant Javadoc. Cela permet d'avoir une référence complète des classes et méthodes disponibles.
- make jar : Crée un fichier .jar exécutable du projet, rendant le programme facilement portable et simple à exécuter.
- make : Compile le code source en exécutables Java.
- make run : Lance le programme pour une exécution immédiate après la compilation.

Voici le diagramme de classe **simplifié** du jeu:



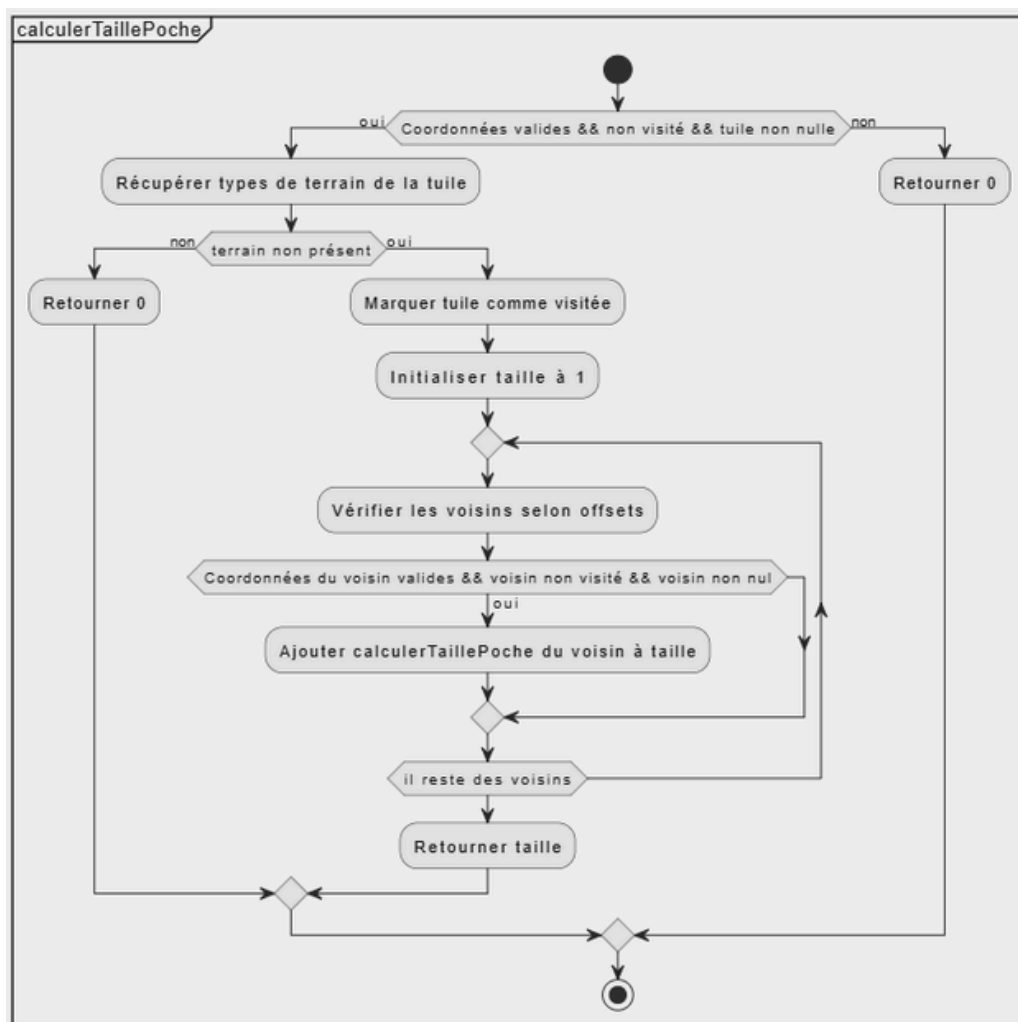
Modèle MVC

Nous avons structuré notre projet selon le modèle MVC (Modèle-Vue-Contrôleur), ce qui nous a permis de gérer les différentes parties du programme de manière claire et efficace. Chaque package a été créé avec un rôle bien défini : le Modèle contient les classes qui définissent les données du jeu (comme les tuiles et les types de terrain), la Vue prend en charge l'affichage et l'interface utilisateur, et le Contrôleur s'occupe de la gestion des interactions entre les données et l'interface, en traitant les actions du joueur. Cette organisation par packages, fondée sur le MVC, a permis de rendre le code plus lisible et modulaire, facilitant ainsi les développements futurs et les modifications éventuelles.



Analyse de l'algorithme de identification des poches

Nous identifions les poches a l'aide d'une méthode récursive. Nous avons élaboré un diagramme d'activité afin de modéliser notre idée. A chaque fois qu'une tuile est posée les nouvelles tuiles sont identifier.

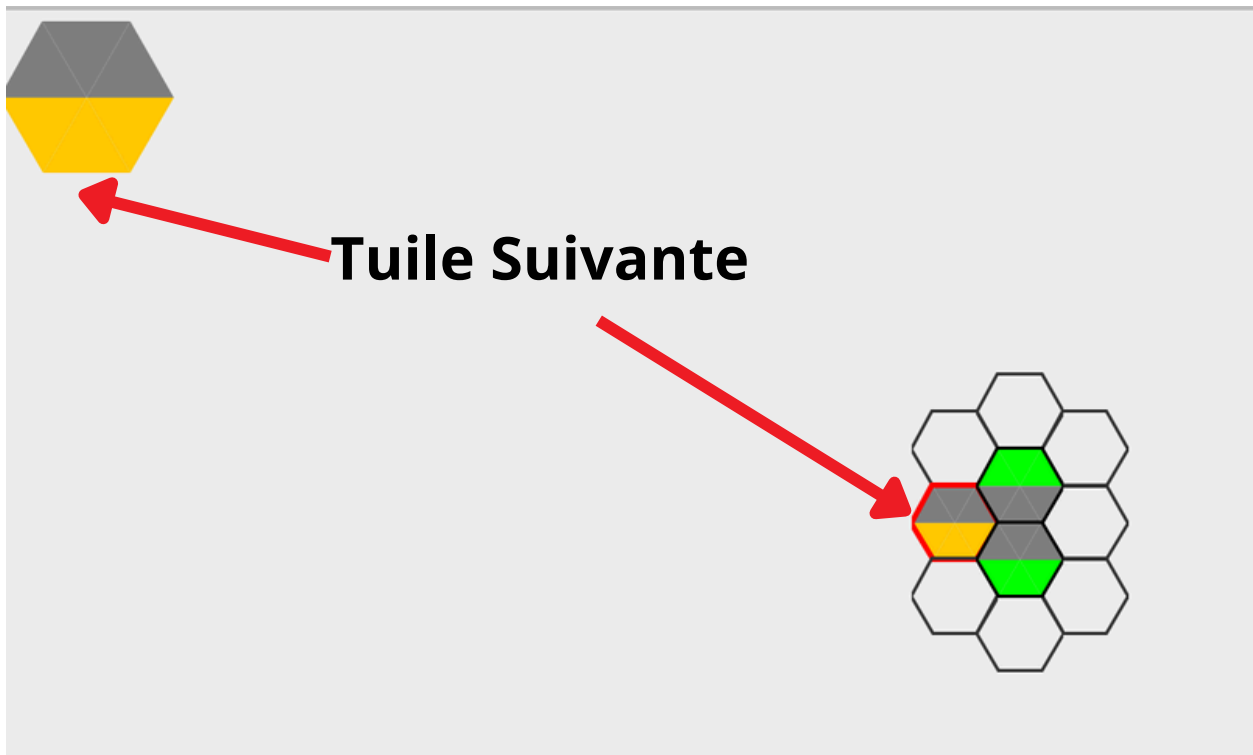


Améliorations ergonomiques

EXPLICATIONS DES CHOIX D'INTERFACE UTILISATEUR

L'interface de jeu a été pensée pour offrir au joueur une expérience visuelle claire et agréable, tout en mettant en avant les éléments essentiels pour qu'il puisse se concentrer sur sa stratégie de placement. Voici les principaux choix qui ont guidé la conception de l'interface en cours de partie :

Affichage du prochain bloc : Pour permettre au joueur d'anticiper ses actions, l'interface affiche toujours la prochaine tuile à placer qui est entourée en rouge. Cette fonctionnalité aide à planifier les combinaisons de terrains et améliore la fluidité de jeu en donnant au joueur une vision d'ensemble sur ses prochaines options.



Choix des couleurs : Nous avons choisi des couleurs équilibrées, ni trop claires ni trop foncées, afin que chaque type de terrain soit facilement reconnaissable sans agresser les yeux. Les tons utilisés aident à distinguer chaque terrain (bleu pour la mer, vert pour la forêt, gris pour la montagne, orange pour le champ etc.) de manière intuitive, tout en donnant un aspect harmonieux au paysage au fur et à mesure qu'il se construit.

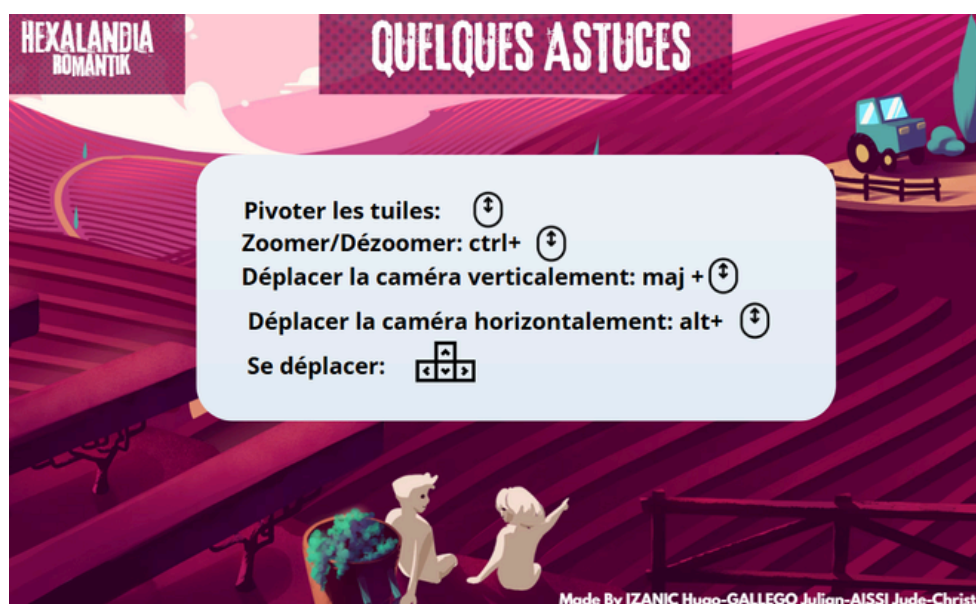
MOUVEMENT DE LA CAMÉRA

Pour améliorer l'expérience de jeu, nous avons intégré plusieurs combinaisons de touches qui permettent à l'utilisateur de contrôler la caméra :

- Ctrl + Molette : permet de zoomer sur le paysage, utile pour voir les détails des tuiles.
- Maj + Molette : permet de se déplacer verticalement dans le paysage.
- Flèches du clavier : permettent de déplacer la vue horizontalement.

Ces contrôles simplifient la navigation et permettent au joueur de se concentrer pleinement sur la création du paysage, rendant ainsi la partie plus fluide et agréable.

Pour d'autant plus améliorer l'expérience de jeu et aider le joueur à se familiariser avec les commandes, nous avons ajouté une page de conseils accessible avant le début de chaque partie. Cette page récapitule les combinaisons de touches essentielles, comme le pivotement des tuiles ou le déplacement de la caméra. En donnant ces informations dès le départ, nous facilitons la prise en main du jeu et permettons au joueur de se concentrer pleinement sur la stratégie, sans se perdre dans les contrôles.

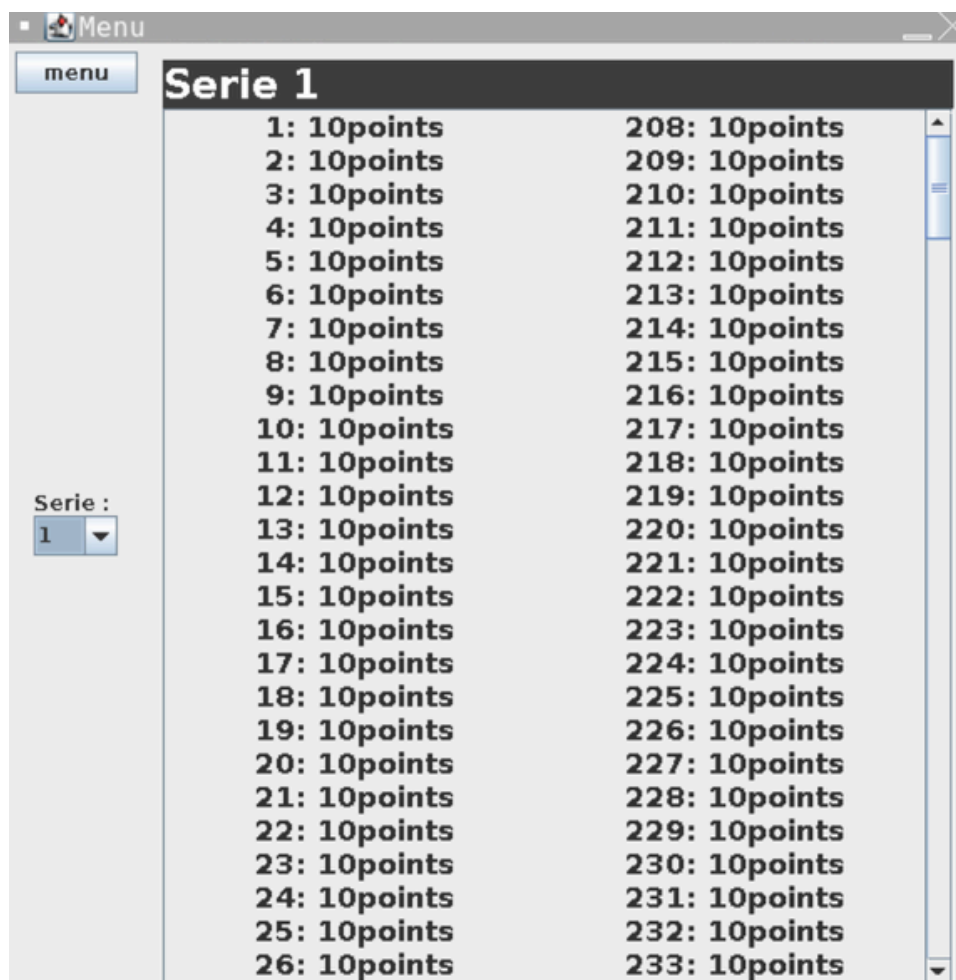


Evaluation des scores

TABEAU DE SCORE

Tout d'abord le joueur a acces ,note tableau des scores est similaire a ceux qu'on retrouve dans les jeux arcade sauf que le notre est anonyme comme demander.le tableau de score affiche un classement détaillé des champs en fonction de leurs performances au sein d'une série donnée. Chaque ligne du tableau représente un champ identifié par un numéro (1, 2, 3, etc.) et indique le score en points obtenu par ce champ. Ces scores permettent de voir, pour chaque série de tuiles, les résultats des champs en fonction de leur taille et de leur composition.

Pour faciliter la navigation entre différentes séries, un menu déroulant permet à l'utilisateur de sélectionner la série de tuiles de son choix.



The screenshot shows a window titled 'Menu' with a 'menu' button and a dropdown menu set to 'Serie 1'. The main area displays a table of scores for 33 fields, arranged in two columns. Each entry consists of a field number followed by '10points'.

Serie 1	
1: 10points	208: 10points
2: 10points	209: 10points
3: 10points	210: 10points
4: 10points	211: 10points
5: 10points	212: 10points
6: 10points	213: 10points
7: 10points	214: 10points
8: 10points	215: 10points
9: 10points	216: 10points
10: 10points	217: 10points
11: 10points	218: 10points
12: 10points	219: 10points
13: 10points	220: 10points
14: 10points	221: 10points
15: 10points	222: 10points
16: 10points	223: 10points
17: 10points	224: 10points
18: 10points	225: 10points
19: 10points	226: 10points
20: 10points	227: 10points
21: 10points	228: 10points
22: 10points	229: 10points
23: 10points	230: 10points
24: 10points	231: 10points
25: 10points	232: 10points
26: 10points	233: 10points

COMPARAISON DES SCORES

Pour permettre aux joueurs d'évaluer leurs performances, notre programme inclut une fonctionnalité de comparaison de score. À la fin de chaque partie, le score obtenu par le joueur est comparé aux 5 scores inférieur et supérieur enregistrés pour la même série de tuiles. Cela permet au joueur de voir où il se situe par rapport aux autres, offrant un aspect compétitif et motivant pour améliorer son propre score. Les scores sont anonymisés pour respecter la confidentialité, tout en offrant une vue d'ensemble de la qualité du score dans le contexte des autres partie

Recapitulatif Serie 1		
Point:	131	
Position:	16/80	
11	238 points	
12	232 points	
13	216 points	
14	194 points	
15	166 points	
16	131 points	
17	116 points	
18	103 points	
19	98 points	
20	72 points	
21	66 points	

Conclusion

Izanic Hugo

Au cours de ce projet, nous avons appliqué **tous** l'ensemble des notions abordées dans le module de Développement 3.1. Les questions d'ergonomie se sont révélées être un défi bien plus important que dans la SAÉ précédente, exigeant une réflexion approfondie pour améliorer l'expérience utilisateur. Ce projet m'a également permis de renforcer mes compétences en Java et de gagner en confiance dans la maîtrise du langage.

Gallego Julian

Ce nouveau projet m'a permis de renforcer mes capacités en appliquant toutes mes connaissances en pratique grâce à ce projet. J'ai eu beaucoup de mal coder le jeu en lui-même notamment pour le score. Cependant j'ai eu plus de facilité à faire les différents menus ainsi que l'affichage du jeu.

AISSI Jude-Christ

Ce projet m'a permis de grandement développer mes compétences en conception d'interfaces en Java. J'ai appris à mieux comprendre les principes de design et l'importance de l'ergonomie. De plus, j'ai compris que la remise en question est essentielle pour identifier et corriger mes erreurs. Cette expérience m'a non seulement amélioré en tant que développeur, mais aussi renforcé ma capacité à analyser et à optimiser mes processus de travail.