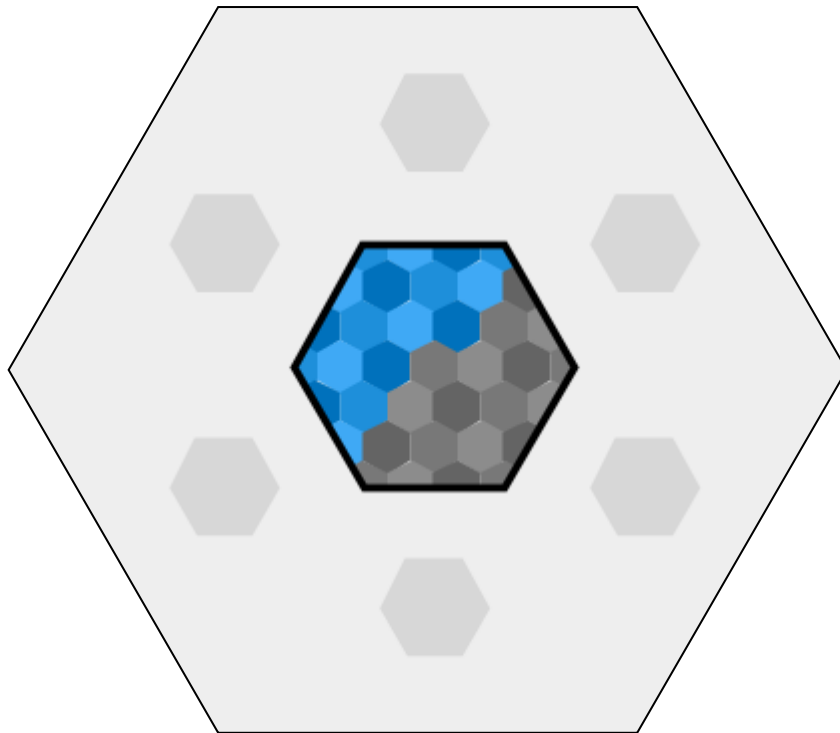


# Rapport d'avancement : Dorfromantik



**Marco ORFAO**  
**Maxence Raymond**  
**Gaston Chenet**

# Sommaire

## I. Introduction

1.	Sujet	03
2.	Informations complémentaires	03

## II. Jeu

1.	Fonctionnalités	04
2.	Structure du programme	11
3.	Algorithme des poches	20
4.	Améliorations ergonomiques	21
5.	Leaderboard et score	25

## III. Conclusions

1.	Maxence Raymond	26
2.	Gaston Chenet	26
3.	Maxence Raymond	27

# I. Rappel du sujet

## 1. Sujet

Le sujet est grandement inspiré d'un jeu déjà existant : "Dorfromantik". Le but est d'assembler des tuiles contenant une à deux couleurs parmi les couleurs suivantes : Bleu, Jaune, Vert clair, Vert foncé et Gris. Ces couleurs représentent des types prédéfinies de terrains comme la mer (Bleu), les champs (Jaune), le pré (Vert clair), les forêts (Vert foncé) et les montagnes (Gris) dans le but final de former un "paysage harmonieux".

Une partie se déroule de la façon suivante :

- Une tuile est automatiquement placé au centre du plateau et une pile de tuile est mis à la disposition du joueur.
- Un à un, le joueur doit placer la tuile se situant au sommet de la pile à côté d'une tuile déjà présente sur le plateau tout en essayant d'assembler le plus de tuiles de la même couleur.
- Le score de fin de partie se calcule en fonction du nombre de zone de couleurs assemblées par le joueur. Dans une zone, le nombres de tuiles la composant est élevée au carré. Le score final revient à additionner les nombre obtenus pour chaque zone.
- En fin de partie, le joueur a la possibilité d'inscrire son nom et son score sera sauvegardé.

## 2. Informations complémentaires

Le jeu est entièrement programmé en Java et une base de donnée est utilisée dans le but de sauvegarder les scores pour ainsi créer un tableau des scores pour que le joueur puisse comparer ses précédents scores.

## II. Jeu

### 1. Fonctionnalités

#### Général

Excepté dans le menu principal, le joueur a la possibilité à tout moment d'appuyer sur la croix pour fermer la fenêtre sur laquelle il se situe, une fenêtre modale "Quitter Dorfromantik" apparaît alors pour confirmer la fermeture du jeu (voir Figure 1.0). Dans cette fenêtre, le joueur a le choix entre s'il veut vraiment quitter le jeu (bouton "Oui") ou s'il change d'avis et veut rester sur le jeu (bouton "Non").

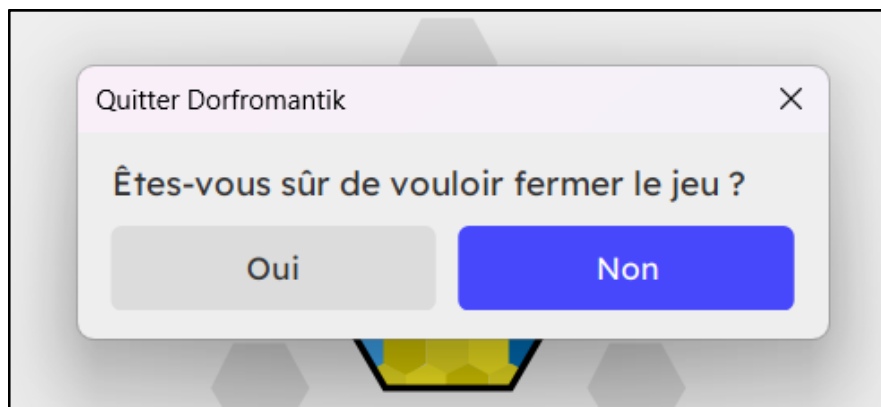


Figure 1.0 : Illustration de la fenêtre modale "Quitter Dorfromantik"

## Menu

Lorsque le joueur lance le programme, la première fenêtre qui lui est affichée est la fenêtre de menu principal. Dans ce menu, le joueur a la possibilité de :

- Voir un tableau des score vierge avec quatre boutons sous ce tableau permettant d'afficher les 14 meilleurs scores enregistré pour l'une des quatre séries de tuiles proposé par défaut au joueur. (voir Figure 2.0)



Figure 2.0 : Illustration du tableau des scores

- appuyer sur l'un des 3 boutons de contrôles suivant : Commencer la partie, Paramètres, Quitter. (voir Figure 2.1)
- Le bouton "Quitter" met fin au programme.
- Le bouton "Paramètres" ouvre une fenêtre destinée aux paramètres (voir Figure 2.2) où il est possible de régler le volume de la musique, des effets sonores ainsi que de couper tout les sons.
- Le bouton "Commencer la partie" (voir Figure 2.3) ouvre une fenêtre contenant quatre boutons permettant chacun de choisir l'une des pack de tuiles mis à disposition ou bien de donner au programme une seed personnalisée qui représente donc un pack de tuiles différent de ceux proposé par défaut au joueur.

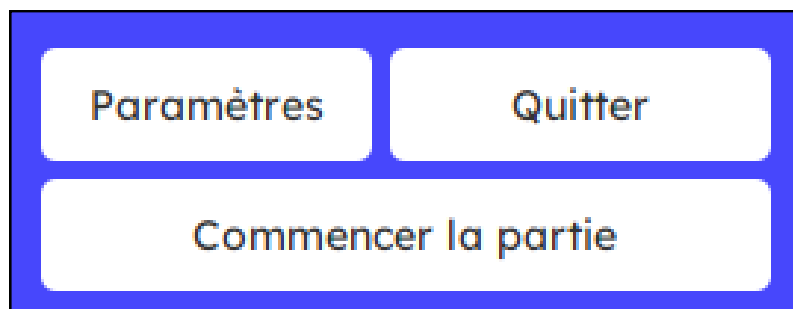


Figure 2.1 : Illustration des boutons de contrôles du menu

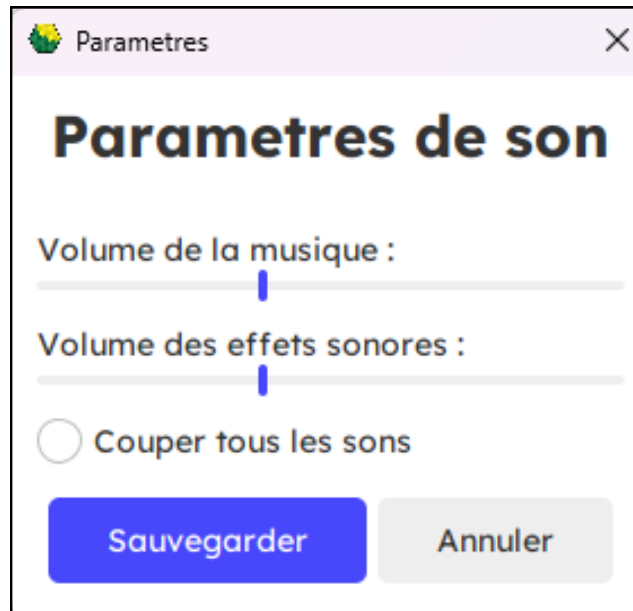


Figure 2.2 : Illustration de la fenêtre “Paramètre”

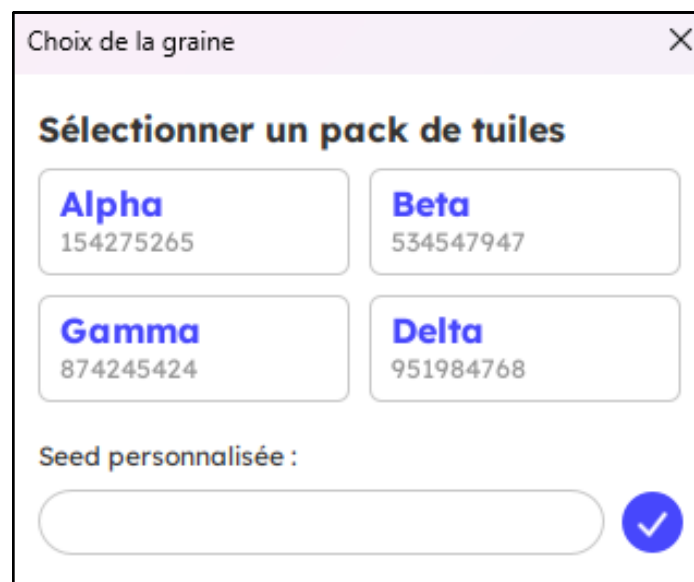


Figure 2.3 : Illustration de la fenêtre de choix du pack de tuiles

## Jeu

Une fois la partie lancée, le joueur a la possibilité de réaliser les actions suivantes :

- Placer une tuile dans l'un des emplacements prédéfinies autour d'une tuile déjà existante (voir figure 3.0)

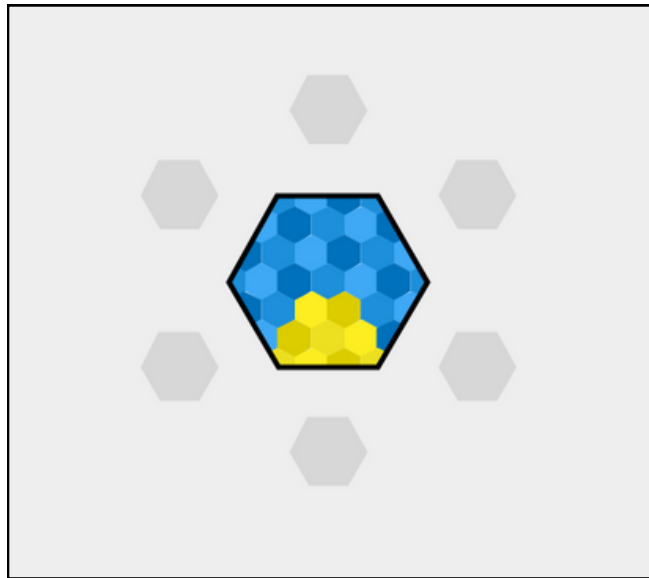


Figure 3.0 : Illustration des zones prédéfinies où une tuile est posable.

- Lorsque le joueur fait glisser le curseur de la souris au-dessus de l'une de ces zones prédéfinies, le joueur peut visualiser la tuile en taille réduite à l'emplacement qu'il pointe. (voir figure 3.1)

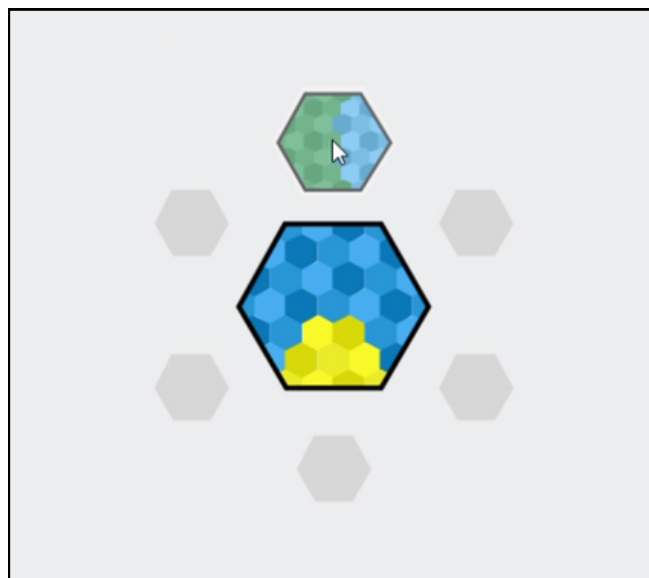


Figure 3.1 : Illustration de la visualisation miniature de la tuile posable



- Une fois que la tuile miniature apparaît, le joueur a la possibilité de donner à la tuile l'orientation qu'il souhaite grâce à la molette. S'il glisse la molette vers le haut, la tuile visualisée tourne dans le sens anti-horaire et à l'inverse, s'il glisse la molette vers le bas, dans le sens horaire.
- Après avoir choisi un emplacement ainsi que l'orientation de la tuile, le joueur a la possibilité de cliquer sur cette zone prédéfinie pour poser la tuile.
- Si le joueur maintient le clic gauche de la souris enfoncé, il a alors la possibilité de déplacer le plateau de jeu en déplaçant la souris.
- Le joueur a la possibilité de mettre en pause la partie s'il appuie sur la touche "ESC" ou bien "ECHAP" (pour un clavier français). Une fenêtre modale apparaît donc. (voir figure 3.2)

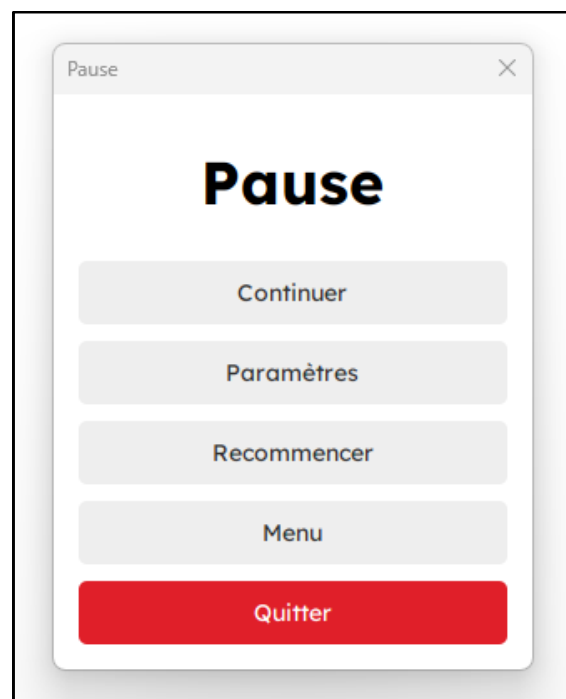


Figure 3.2 : illustration de la fenêtre modale "Pause"

- Dans cette fenêtre modale "Pause", 5 boutons sont mis à disposition. Le premier "Continuer" ferme la fenêtre modale pour permettre au joueur de continuer la partie. Le deuxième "Paramètres" ouvre la page de paramètre (voir Figure 2.2). Le troisième "Recommencer" recrée une instance du jeu avec la même seed. Le quatrième "Menu" ferme la fenêtre, met fin à la partie et affiche le menu principal de nouveau. Le cinquième "Quitter" met fin à la partie ainsi qu'au programme.

## Fin de partie

Une fois la partie terminée, une fenêtre modale "Fin de partie" s'ouvre (voir figure 4.0) et le joueur a la possibilité de :

- Enregistrer son score en inscrivant son pseudo dans le champ de texte ainsi qu'en validant en cliquant sur le bouton à sa droite, le score sera donc enregistré dans la base de données.
- Voir que lorsqu'il a enregistré son score, le message disant que le score n'a pas été enregistré change de couleur et de lui indique que le score a bien été enregistré.
- Appuyer sur le bouton "Rejouer" pour relancer une partie avec le même pack de tuiles.

- Appuyer sur le bouton "Menu" pour retourner au menu.
- Appuyer sur le bouton "Quitter" pour mettre fin au programme et ainsi quitter le jeu.
- Fermer la fenêtre de fin de partie en appuyant sur la croix, cela permet au joueur de pouvoir naviguer et d'observer le plateau de jeu de la partie.



Figure 4.0 : illustration de la fenêtre modale de fin de partie

## 2. Structure du programme

Pour ce qui est de la structure de notre programme, nous avons suivi le modèle MVC (Modèle Vue Contrôleur) appliqué uniquement au contenu des classes, pour l'organisation des classes,

nous les avons répartis dans plusieurs sous-packages dans le but de rassembler toutes les classes selon leurs utilités ainsi que leurs rôles. Le programme comptabilise un total de 7 sous packages se nommant : components, end, enums, game, gui, landing et utils.

Certaines classes comme "Main", "Options" ou bien "Package-info" ne font partie d'aucun de ces sous package, ils font donc tous parti du package principal "fr.kanassoulier.literomantik" (voir Figure 5.0)

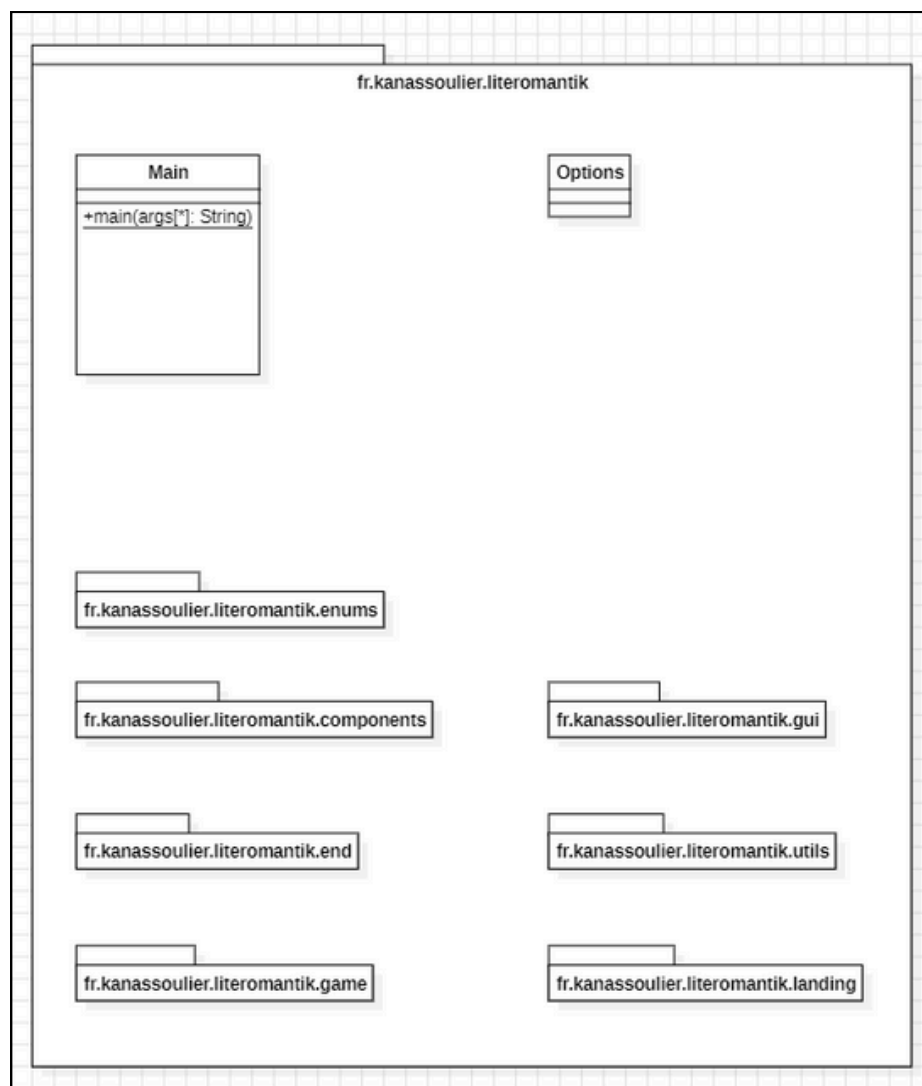


Figure 5.0 : Diagramme de classes représentant le package principal

## fr.kanassoulier.literomantik.components

Le sous package components ressource toutes les redéfinitions des différents composants utilisés tels qu'une version amélioré des JButton, JTextField, etc (voir Figure 5.1). Les classes présentes dans ce package sont les suivantes :

- KButton (vue)
- KButtonListener (contrôleur)
- KCheckBox (vue)
- KCheckBoxContent (vue)
- KCheckBoxContentListener (contrôleur)
- KSlider (vue)
- KSliderThrumb (vue)
- KSliderThrumbListener (contrôleur)
- KTextField (vue)
- KTextFieldInput (vue)
- KTextFieldSubmit (vue)
- KTextFieldSubmitListener (contrôleur)
- package-info (documentation)

Le 'K' précédant le nom de toutes les classes signifie "kanassoulier" qui est une signature de notre part (Kanassoulier est le rassemblement de "Kanaper", "Cassoulet" et "Goulier" qui sont nos pseudonymes ). Ce package ne contient que des vues et des contrôleurs.

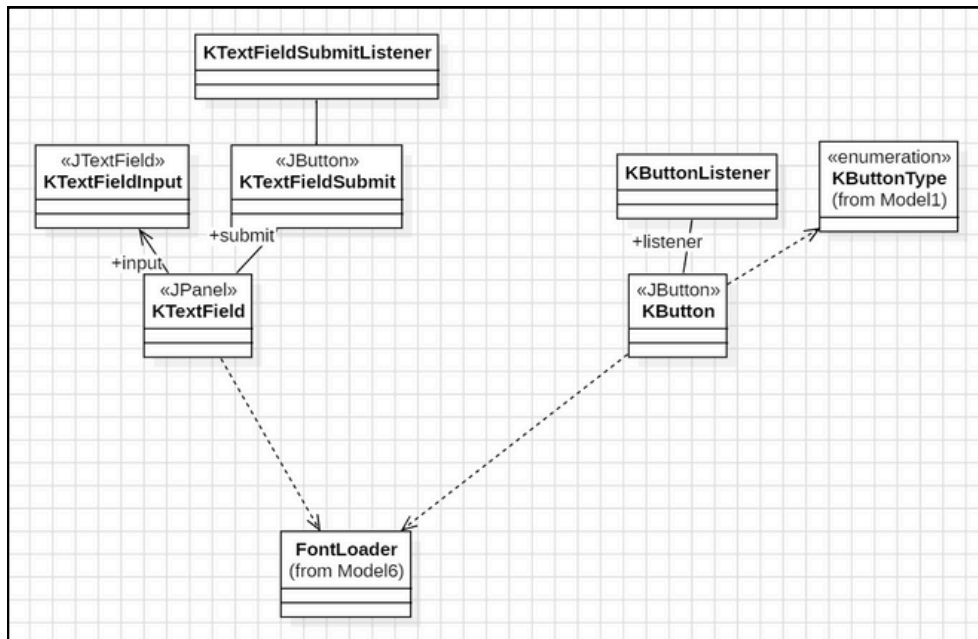


Figure 5.1 : Diagramme de classes représentant le sous-package `fr.kanassoulier.literomantik.components`

## `fr.kanassoulier.literomantik.end`

Le sous-package `end` recense toutes les classes qui sont liées à la fenêtre de fin de partie. (voir Figure 5.2) Les classes présentes dans ce package sont les suivantes :

- `EndGameInfo` (modèle)
- `EndGameTextFieldListener` (contrôleur)
- `EndMenu` (vue)
- `EndMenuButtonListener` (contrôleur)
- `EndMenuButtonRow` (vue)
- `EndMenuListener` (contrôleur)
- `package-info` (documentation)

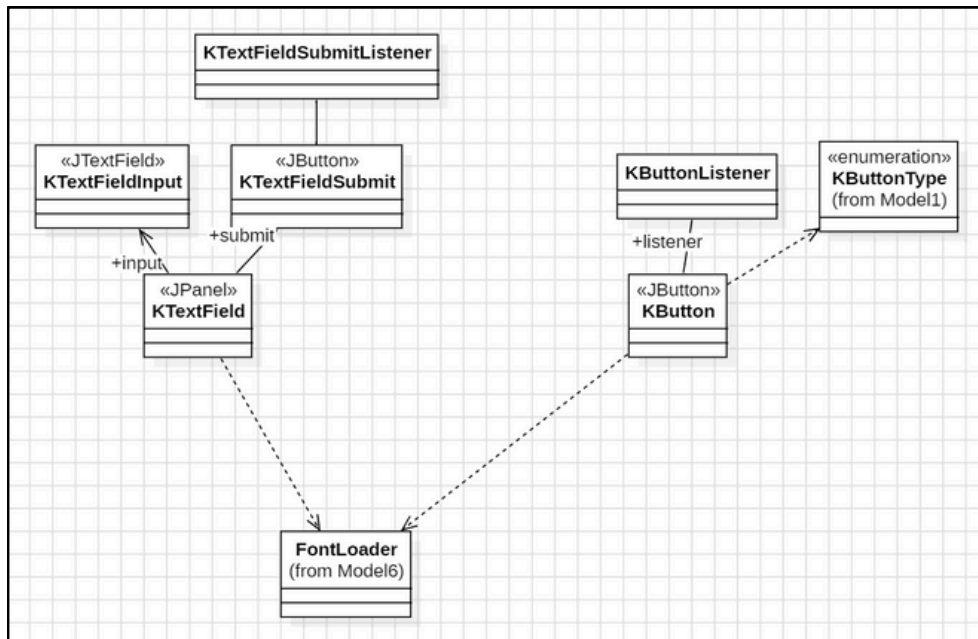


Figure 5.2 : Diagramme de classes représentant le sous package `fr.kanassoulier.literomantik.end`

## `fr.kanassoulier.literomantik.enums`

Le sous package `enums` ressource les énumérations qui seront utilisée dans le programme. (voir Figure 5.3) Les classes présentes dans ce package sont les suivantes :

- Biome (modèle)
- KButtonType (modèle)
- KeyBoardKey (modèle)
- SoundChannel (modèle)
- TileSide (modèle)
- package-info (documentation)

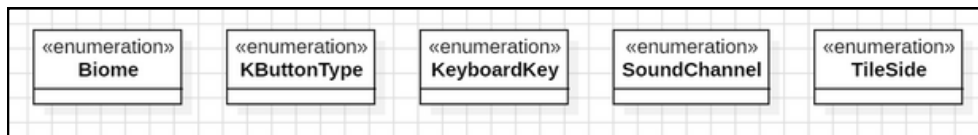


Figure 5.3 : Diagramme de classes représentant le sous package `fr.kanassoulier.literomantik.enums`

## `fr.kanassoulier.literomantik.game`

Le sous-package `game` recense toutes les classes qui sont liées au fenêtre de jeu (voir Figure 5.4). Les classes présentes dans ce package sont les suivantes :

- `Board` (contrôleur)
- `Cell` (contrôleur)
- `EmptyCell` (contrôleur)
- `Game` (vue)
- `GameInteractionListener` (contrôleur)
- `GameKeyListener` (contrôleur)
- `PlaceableArea` (vue)
- `PlaceableAreaListener` (contrôleur)
- `PlaceableTile` (contrôleur)
- `Tile` (vue)
- `package-info` (documentation)



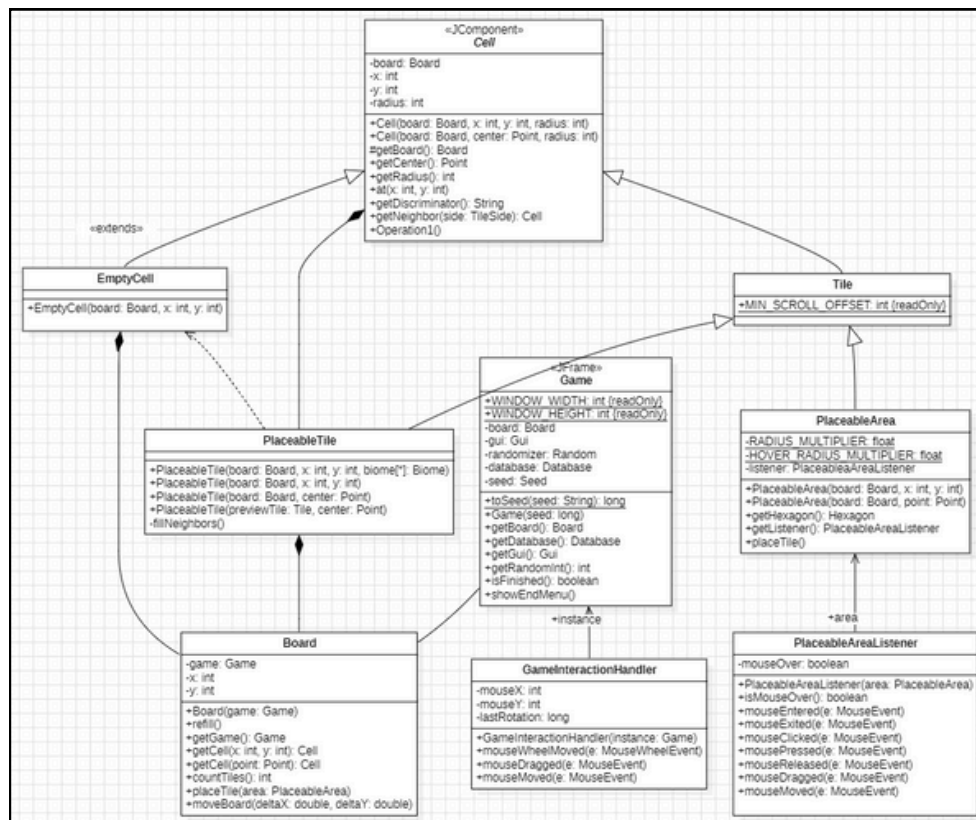


Figure 5.4 : Diagramme de classes représentant le sous package  
fr.kanassoulier.literomantik.game  
fr.kanassoulier.literomantik.gui

Le sous-package gui recense toutes les classes de l'interface graphique (voir Figure 5.5) Les classes présentes dans ce package sont les suivantes :

- CloseGameDialog (vue)
- CloseGameDialogButtonListener (contrôleur)
- CloseGameDialogListener (contrôleur)
- Gui (vue)
- KeyInfo (vue)
- PauseWindow (vue)
- PauseWindowButton (vue)
- PauseWindowButtonsListener (contrôleur)

- PauseWindowKeyListener (contrôleur)
- PreviewTile (vue)
- PreviewTileTimerListener (contrôleur)
- Scoreboard (vue)
- ScoreUpdate (vue)
- ScoreUpdateTimerListener (contrôleur)
- TileStack (vue)
- Settings (vue)
- SettingsButtonListerner (contrôleur)
- package-info (documentation)

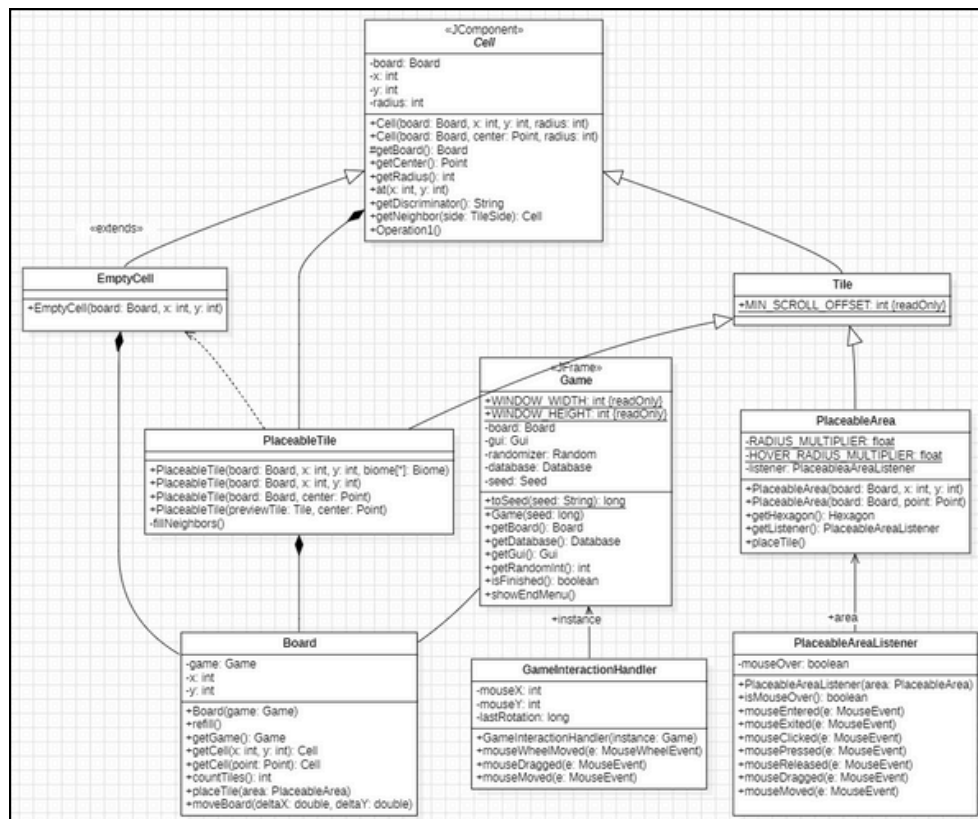


Figure 5.5 : Diagramme de classes représentant le sous package fr.kanassoulier.literomantik.gui

## fr.kanassoulier.literomantik.landing

Le sous package landing recense toutes les classes du menu principal (voir Figure 5.6). Les classes présentes dans ce package sont les suivantes :

- LandingMenu
- LandingMenuControlButton
- LandingMenuControlButtonContainer
- LandingMenuControlButtonListener
- LandingMenuLeaderboard
- LandingMenuSidebar
- SeedSelector
- SeedSelectorButton
- SeedSelectorButtonListener
- SeedSelectorSubmitListener

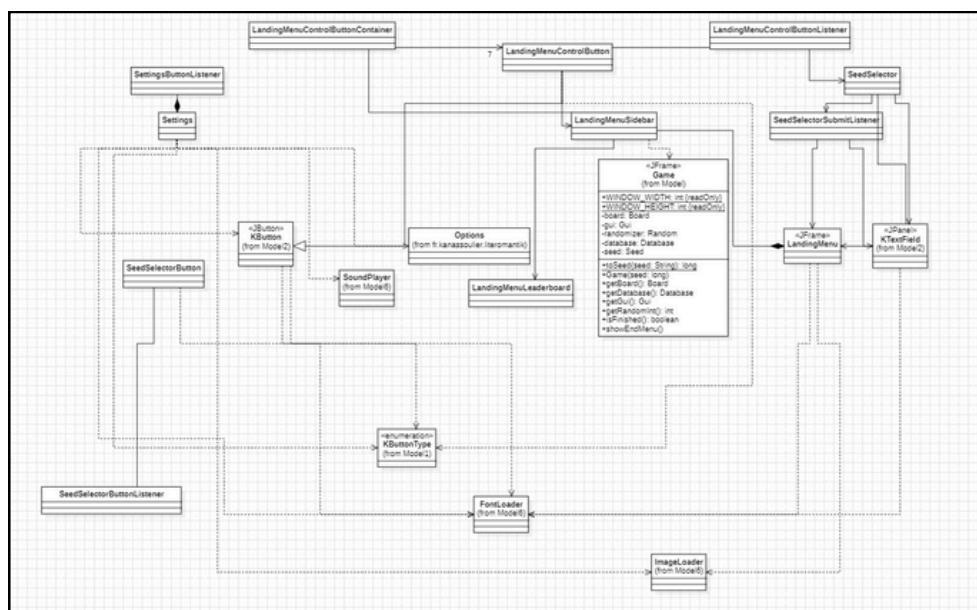


Figure 5.6 : Diagramme de classes représentant le sous package fr.kanassoulier.literomantik.landing

## fr.kanassoulier.literomantik.utils

Le sous package landing recense toutes les classes utilitaires (voir Figure 5.7). Les classes présentes dans ce package sont les suivantes :

- Database
- Environment
- FontLoader
- Hexagon
- ImageLoader
- ScoreLogic
- Seed
- SoundPlayer
- package-info

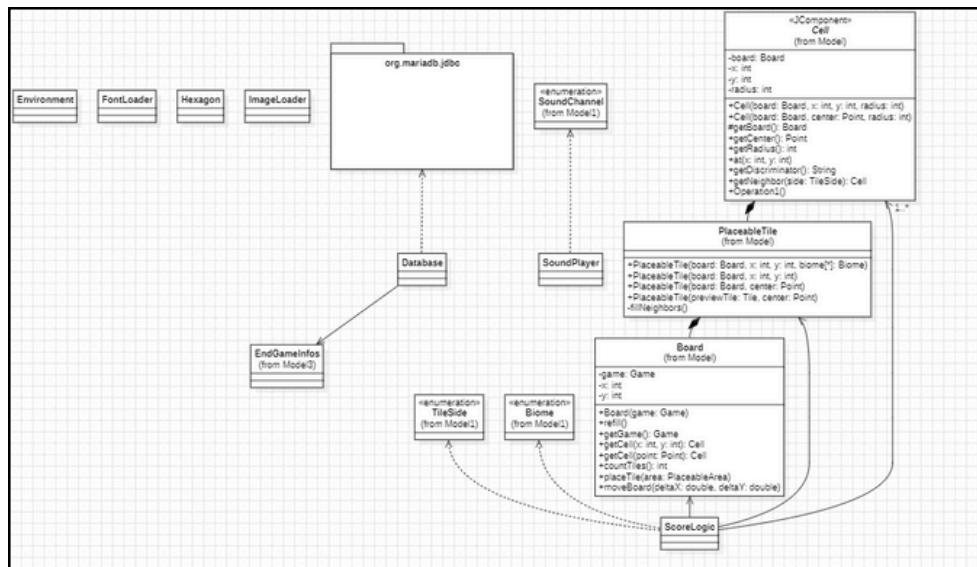


Figure 5.7 : Diagramme de classes représentant le sous package `fr.kanassoulier.literomantik.utils`

### 3. algorithme des poches

L'algorithme des poches commence par choisir un biome puis va se placer sur la première tuile posée. Il va ensuite regarder si la tuile contient le biome qu'il cherche, s'il ne le trouve pas, il marque la tuile comme déjà visité et passe à la deuxième tuile posée. Si il trouve le biome, il va regarder ses voisins directs et ainsi de suite jusqu'à ce que l'entièreté des tuiles soient visitées. Il recommence ensuite pour le biome suivant et ainsi de suite. Voici un diagramme d'activité pour l'illustrer :

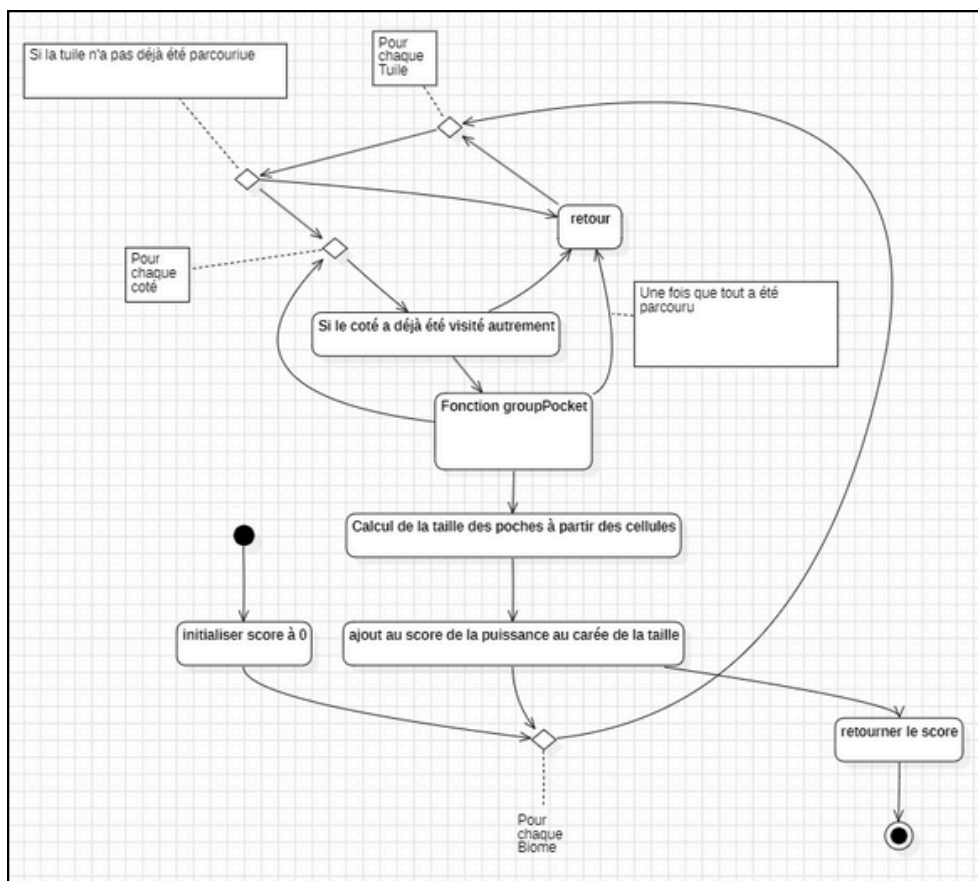


Figure 6.0 : Diagramme d'activité représentant l'algorithme des poches

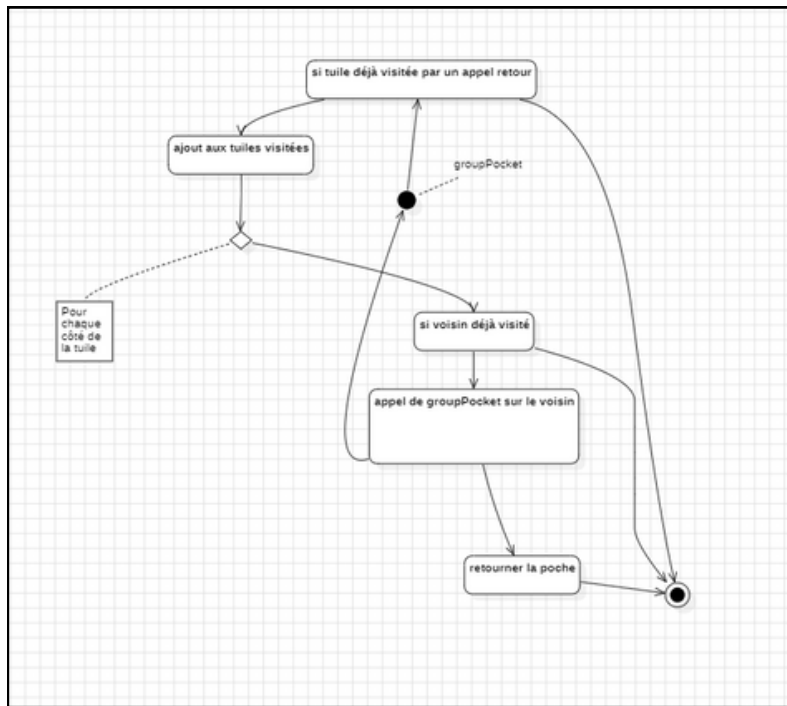


Figure 6.1 : Suite du diagramme d'activité représentant l'algorithme des poches

### 3. Améliorations ergonomique

#### Menu

Pour l'écran de menu (voir Figure 7.0), on a fait le choix de mettre le moins de texte possible et d'opter pour une interface minimaliste et intuitive dans le but de ne pas surcharger l'utilisateur d'indices visuels. Les boutons ont des bords lisses et des coins arrondis ainsi qu'une typographie de bonne taille et lisible pour que les malvoyants n'aient pas de mal à lire le nom de chaque bouton. Ensuite, chaque bouton possède une couleur indicatrice de son importance et un indice visuel signifiant que l'utilisateur survole le bouton avec sa souris.

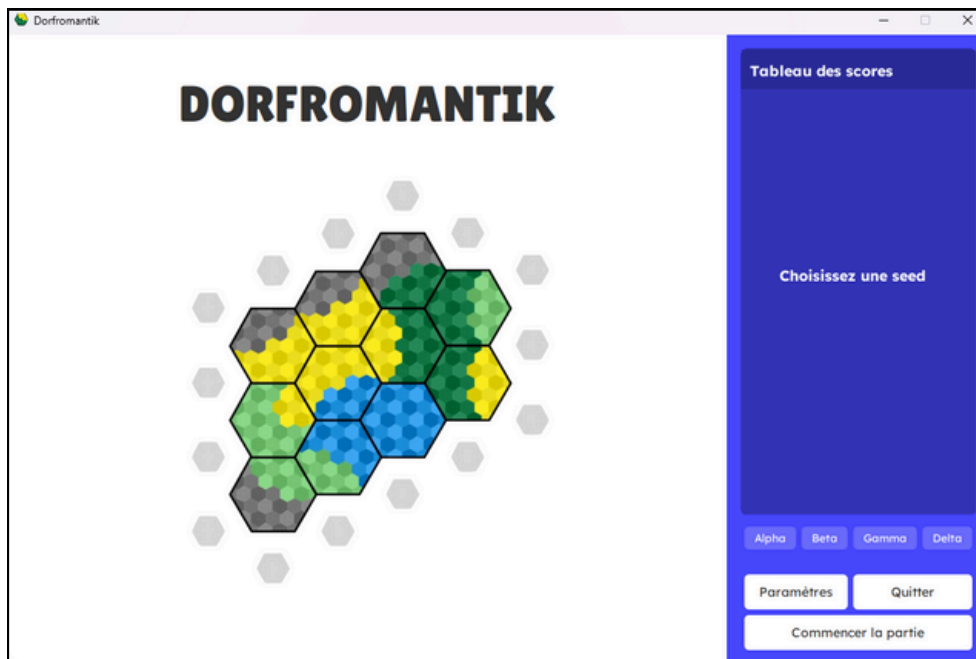


Figure 7.0 : Illustration de la fenêtre de menu

## Paramètres

Pour la page de paramètres, nous avons également opté pour une interface minimaliste. Nous avons également décidé de mettre en valeur le bouton le plus important en lui donnant une couleur qui se démarque du reste du menu pour bien le rendre visible. (voir Figure 2.2)

## Choix de la graine

Pour la page du choix de la graine, nous avons fait le choix de mettre en valeur le nom attribué aux seed par défaut pour mettre cette information en évidence. (voir Figure 2.3)

## Jeu

Pour la page de jeu, les commandes sont affichés en haut à droite pour s'assurer que l'utilisateur n'ait pas à avoir de mauvaise surprise en appuyant sur une touche par hasard. Ensuite, il lui a permis de voir où est-ce qu'il peut poser une tuile et il peut aussi visualiser l'orientation de la tuile par rapport à l'endroit que l'utilisateur pointe pour correctement visualiser l'endroit et l'orientation de la tuile qu'il souhaite poser comparé aux tuiles voisines. Enfin, en plus d'indiquer le nombre de tuiles restantes, un indice visuel sous forme d'une pile de tuile informe l'utilisateur sur son stock de tuiles. (voir Figure 7.1)

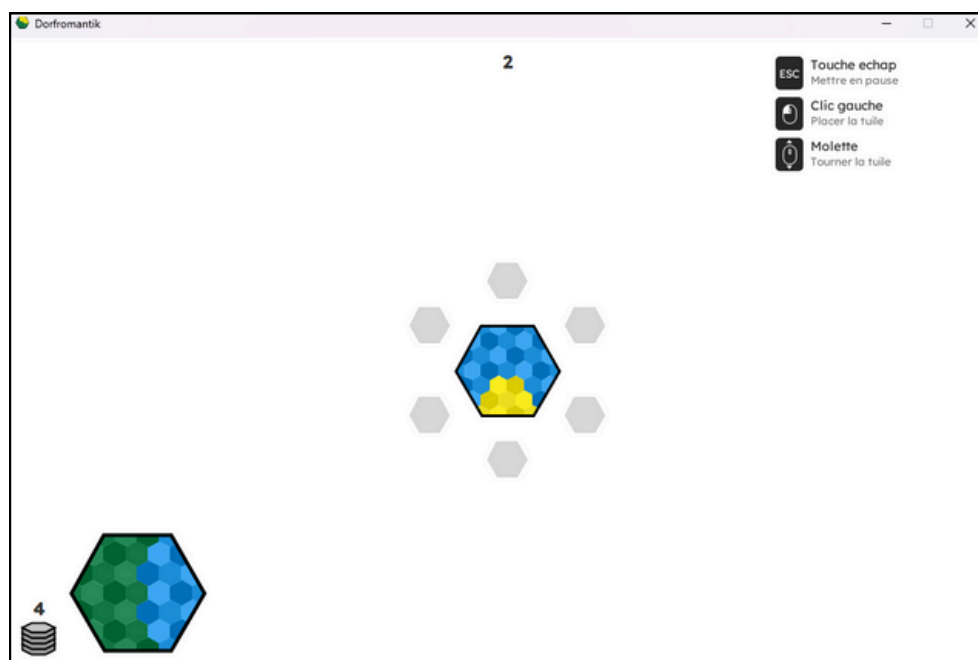


Figure 7.0 : Illustration de la fenêtre de jeu



## Pause

Pour la page de pause, nous avons fait le choix de mettre en évidence le bouton pour quitter pour éviter que l'utilisateur appuie dessus par erreur.

## Fin de partie

Pour la page de fin de partie, le bouton pour rejouer est aussi mis en évidence pour inciter l'utilisateur à rejouer une partie.

## 4. **Leaderboard et scores**

À la fin de la partie, le programme récupère dans la base de donnée le score le plus élevé pour la seed de la partie venant de se terminer et, selon si le score du joueur est supérieur ou inférieur au meilleur score, deux messages peuvent s'afficher. Tout d'abord, si le score est inférieur au meilleur score, le meilleur score enregistré pour cette seed sera affiché en dessous du score du joueur. Enfin, si le score est supérieur au meilleur score enregistré, le message annonçant le meilleur score enregistré change et devient "nouveau record", lui permettant de savoir qu'il a réalisé le meilleur score pour cette seed.

## III. Conclusion

### 1. Maxence Raymond

Cette SAE m'aura permis de développer mes compétences en java et notamment dans la partie logique et implémentation. Elle m'aura permis d'apprendre le fonctionnement des .jar, de gérer les appels ressources systèmes de manière efficace, de gérer efficacement par soi même l'affichage et d'utiliser une base de données de manière efficiente, sans compter le développement de mes compétences en UML et dans l'utilisation de make. Je peux donc dire que cette SAE aura été enrichissante.

### 2. Gaston Chenet

Ma participation à ce projet m'a permis de renforcer mes compétences en travail d'équipe sur un projet simultané, notamment en optimisant la répartition stratégique des tâches pour éviter les conflits entre mes programmes et ceux de mes coéquipiers, ainsi qu'en rédigeant une documentation Java claire et précise. J'ai également approfondi mes compétences dans la création de rapports structurés, en suivant un plan détaillé et en intégrant l'UML. De plus, le travail réalisé représente une preuve tangible de mes compétences, que je pourrai mettre en valeur sur mon CV et dans mon portfolio.

## 3. Marco Orfao

Personnellement, j'ai trouvé cette SAE très enrichissante. J'ai pu grandement élargir mes champs de compétences et apprendre à mieux diviser les travaux dans une équipes dans le but de la rendre plus efficace. J'ai aussi pu apprendre beaucoup d'aspect du java ainsi que beaucoup de norme dans le monde de la programmation grâce à des coéquipiers talentueux qui ont pu rattraper mes erreurs et m'apprendre à ne plus les reproduire. J'ai aussi pu expérimenter tout le début de la création de cette SAE en peer programming avec mes coéquipiers, cela a pu me faire découvrir d'autres façons de coder plus efficacement.