

## Introduction :

Groupe : Louis TANCHOU, Thomas RENAULT, Alexis HADJEM-BRANCHER

Lien vers les deux git (le premier ayant eu des bugs, nous avons dû le changer en cours de route) :

Git buggé : [https://grond.iut-fbleau.fr/hadjem-b/SAE31\\_2024.git](https://grond.iut-fbleau.fr/hadjem-b/SAE31_2024.git)

Git projet fini : [https://grond.iut-fbleau.fr/TanchouCorp/SAE31\\_2024\\_LTA.git](https://grond.iut-fbleau.fr/TanchouCorp/SAE31_2024_LTA.git)

## Sujet :

créer un jeu inspiré de Dorfromantik.

But du jeu : le joueur doit assembler des tuiles hexagonales pour former un paysage harmonieux. C'est à dire crée les biomes les plus gros possibles.

Qu'est-ce qu'un biome ?

Un biome est un environnement (prairie, montagne, lac, foret, champ) représenter chacun par une couleur.

Et donc vous l'aurez compris, le but est de rassembler les couleurs ensembles dans l'objectif d'avoir le plus de points possibles.

## Répartition du travail :

Au début du projet, comme nous ne savions pas très bien utiliser git, nous travaillons ensemble avec l'option "Code With Me" de notre IDE. C'est pour cette raison que, durant la première partie du projet, quasiment tous les commit viennent de Louis TANCHOU. Durant la seconde partie du projet, nous avons commencé à faire des branches pour pouvoir travailler chacun de notre côté sans créer de problème lors du push de notre travail, mais il peut toujours nous arriver de faire du "pair programming" ou du "mob programming" sur certaines fonctionnalités.

## Différentes fonctionnalité présentes :

Poser une tuile, retirer la dernière tuile posée, faire tourner la tuile; Déplacer la caméra, centrer la caméra; Voir le nombre de tuile restante sous la forme d'une pile, affichage de la composition de la première tuile, affichage de la composition de la deuxième et troisième tuile mais en ayant toujours la tuile première tuile au-dessus

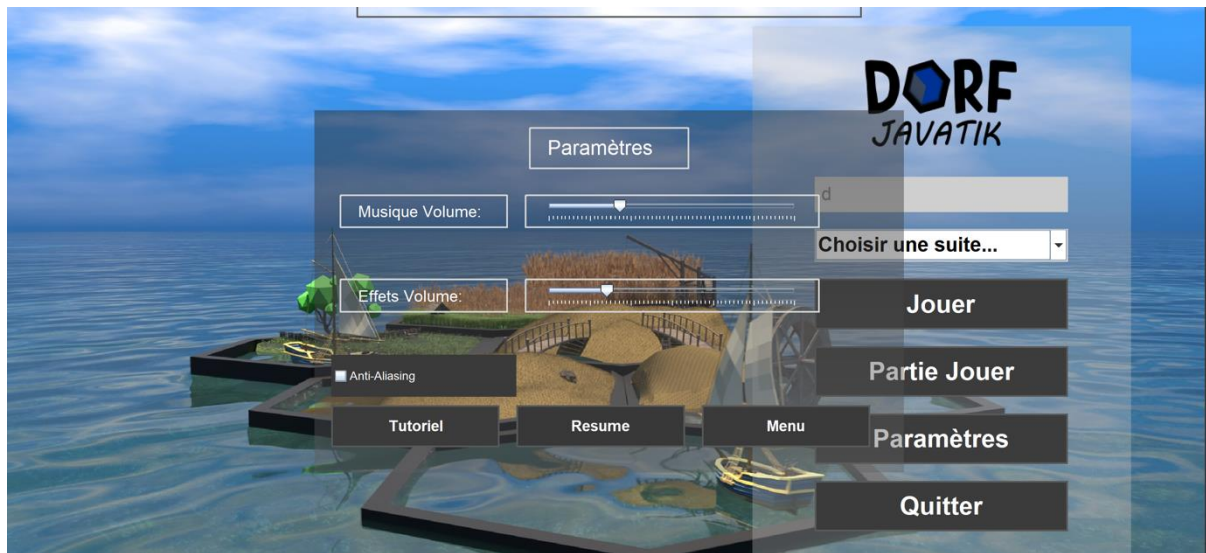
afin de ne pas révéler toute la composition; Comptage du score, possibilité de regarder différent score dans la bdd ,sauvegarde du score, du nom du joueur et de la seed dans la bdd; choix entre partie avec seed prédéfinie ou entièrement aléatoire; Paramétrage du son.

*Les fonctionnalités sont décrites plus en détails dans les prochaines pages.*

Écran d'accueil :

Vous pouvez consulter les scores d'autre joueur, ou les vôtres en cliquant sur partie jouer

En appuyant sur le bouton paramètre ou sur la touche "échap", vous aurez accès aux paramètres :



En cliquant sur "Tutoriel", vous aurez accès à un tutoriel que voici :



En cliquant sur partie jouer, vous verrez apparaître une fenêtre avec toutes les parties jouées qui ont été mis en ligne par les utilisateurs, avec leurs pseudos, leurs scores et la série sur laquelle ils ont joué :

Joueur	Suite	Score
Louis	2601	1
th	1014	2
Louis	537	-2
hergz	219	1
th	214	2
thr	201	1
hergz	169	-2
hergz	131	-2
Louis	40	1
LT	5	2
Louis	0	1
Louis	0	-2
player	0	-2
player	0	1
player	0	1
Louis	0	-2
th	0	-2
th	0	-2

Et comme vous pouvez le voir, il est possible d'effectuer une recherche d'un pseudonyme d'utilisateur grâce à la fonction recherche pour voir ses différentes parties.

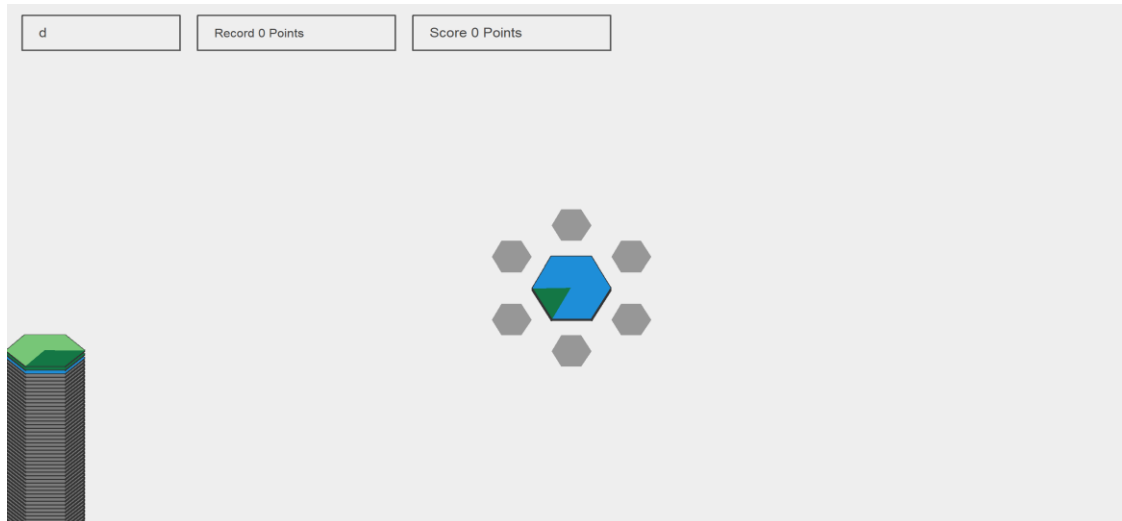
Pour lancer une partie, il faut choisir une série, qui est soit prédéfini, soit aléatoire.

Nous disposons d'un mode en ligne et d'un mode hors ligne. Le mode hors ligne apparait quand la connexion à la base de données est impossible.

Dans cette configuration, vous ne pourrez pas voir vos anciennes parties ainsi que celle des autres joueurs, vous ne pourrez pas accéder au tutoriel et lorsque que vous choisirez une série pour jouer, seule la série aléatoire sera disponible.

Fonctionnalité en jeu :

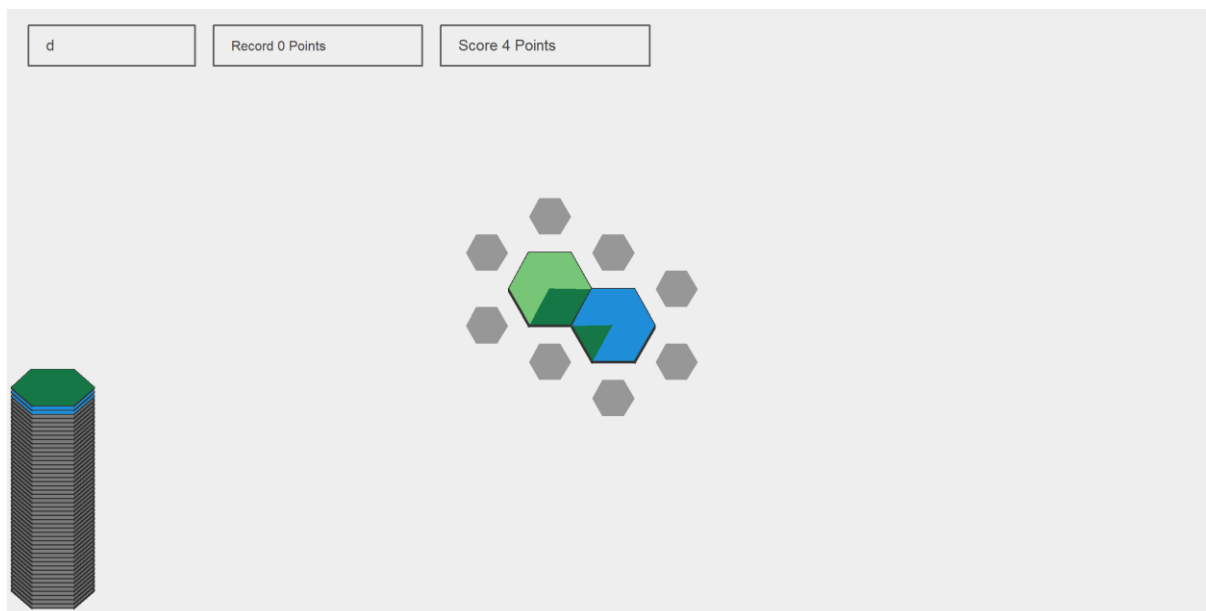
Nous voici dans une partie de DorJavatik et voici ci-dessous le plateau :



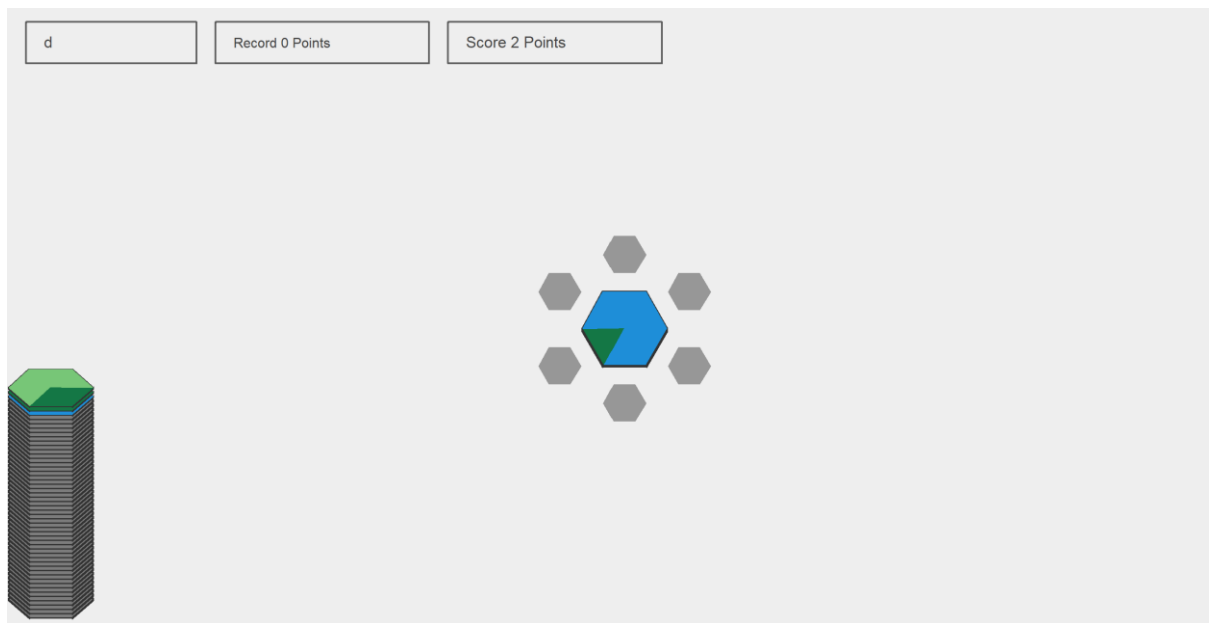
Nous avons notre tuile de départ au milieu, nous pouvons voir la prochaine tuile que nous devons jouer sur la gauche, c'est celle tout au-dessus de la pile.

Autour de la tuile de départ, on peut voir des petits hexagones gris, ce sont les différents endroits où nous pouvons poser une tuile. En passant la souris dessus, nous pouvons voir une prévisualisation de ce que cela donnerait de poser la tuile à cet endroit.

Pour poser une tuile, il faut cliquer sur le bouton gauche de la souris tout en ayant la souris sur l'un des petits hexagones gris

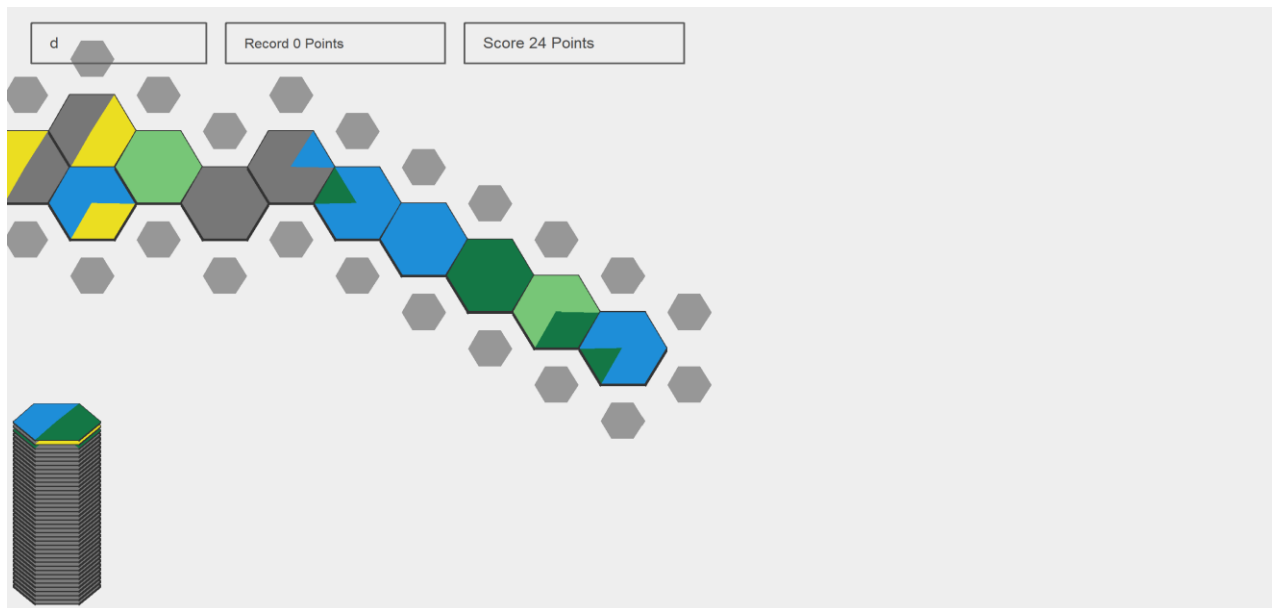


Et voilà notre tuile est posée. Mais malheureusement, nous nous sommes trompé d'endroit où nous voulions poser la tuile. Pas de problème, il est possible d'annuler son dernier coup et cliquant sur le bouton droit de la souris.

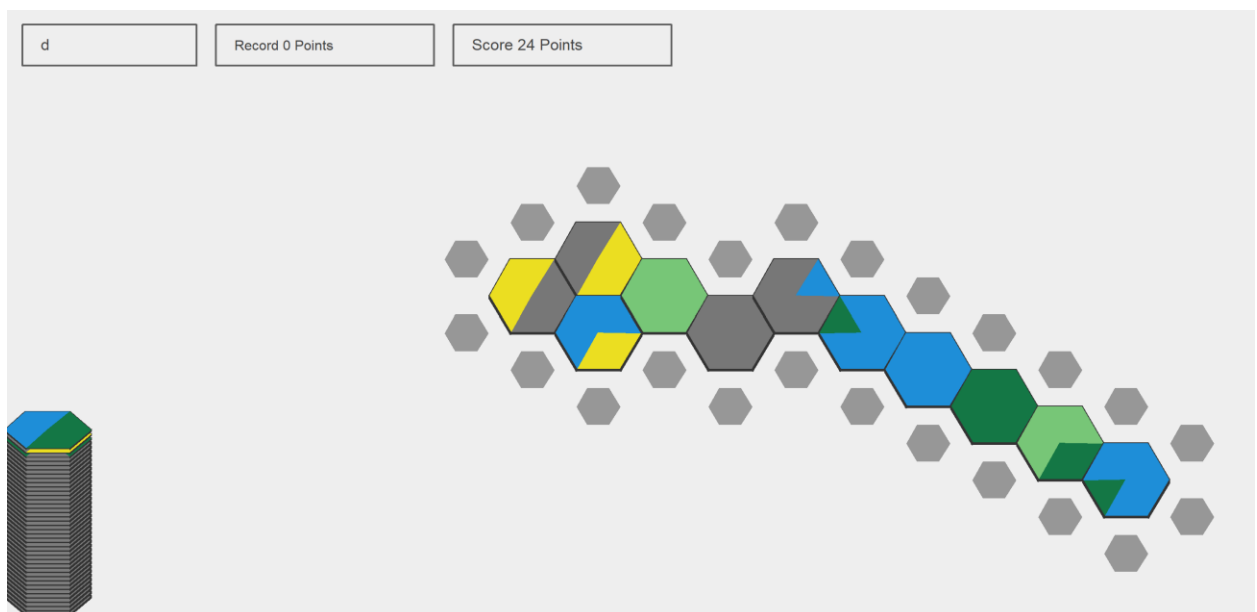


Si vous voulez tourner la tuile dans un sens qui vous avantage, vous pouvez la faire tourner en utilisant la molette de votre souris

Vous avez trop joué d'un côté et vous ne pouvez plus jouer de ce côté ?

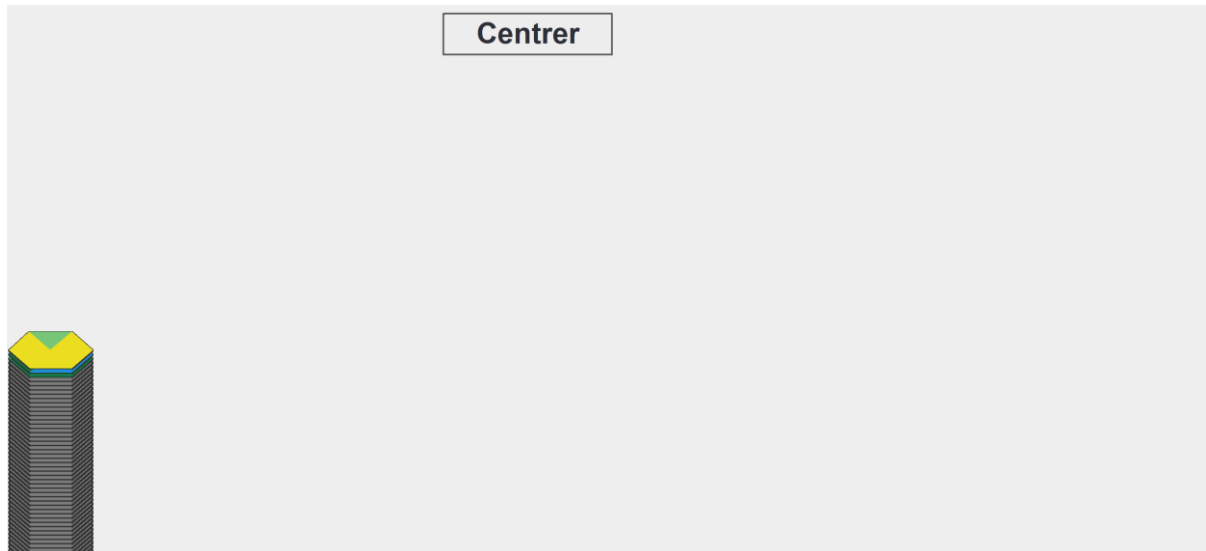


En restant appuyer le bouton gauche de la souris, il est possible de se déplacer.



Malheureusement, vous vous êtes perdu à trop vous déplacer ?

Quand vous êtes trop loin du centre, un bouton “Centrer” apparaît en haut de votre écran, il suffit d’appuyer dessus pour retourner au centre du plateau.



## Présentation structure du programme :

Sur le git, il y a deux diagrammes de classes, le premier à uniquement ce qui permet au jeu de fonctionner pour avoir une version plus petite car cela devient très rapidement confus. Et le deuxième, a presque toute les classes présentes dans le projet, nous avons estimé que mettre la mise en forme et les pop-ups aurait rendu le diagramme illisible.

Lien vers le git : [https://grond.iut-fbleau.fr/TanchouCorp/SAE31\\_2024\\_LTA.git](https://grond.iut-fbleau.fr/TanchouCorp/SAE31_2024_LTA.git)

## Analyse de l’algorithme qui gère les poches :

Une poche contient une liste de tuile et la couleur accordé à la poche

Chaque tuile connaît ses deux biomes et les deux poches dans lesquelles elle est, sont contenues dans un tableau à deux places, avec le premier emplacement du tableau qui est la poche qui gère le premier biome. Si les deux biomes sont les mêmes, alors les deux poches seront les mêmes, mais la tuile ne sera ajoutée qu’une seul fois à la poche

Lorsqu’une tuile est posée, nous créons deux listes, une qui va contenir les poches des voisins qui ont une correspondance de biome avec le premier biome de notre tuile posé et une autre pour le deuxième biome.

On va rechercher ses voisins qu'on va garder dans un tableau, leur emplacement dans le tableau est important car cela représente aussi leur position par rapport à la tuile.

Ensuite grâce à ce tableau, nous allons vérifier s'il y a une correspondance de biomes entre la tuile et ses voisins dans la direction voulu. On récupère un tableau de boolean avec true à l'emplacement du voisin si le biome correspond

Ensuite, nous recherchons dans ce tableau, les emplacements où nous avons une correspondance. S'il y a correspondance, nous récupérons la poche du voisin du biome qui correspond au biome de notre tuile à cet emplacement.

Nous l'ajoutons à l'une des deux listes de poches que nous avons créées préalablement. Si les deux biomes sont les mêmes, la poche sera ajoutée aux deux.

Ensuite nous vérifions la taille des deux listes, en commençant par la première.

Si la taille de la liste est égale à 0, alors nous créons une nouvelle poche.

Si elle est égale à 1, la tuile va s'ajouter à la poche du voisin.

Si elle est supérieure à 1, nous recréons une nouvelle poche, nous parcourons les poches des voisins et on transplante ses tuiles dans cette nouvelle poche. Nous actualisons les tuiles pour qu'elles connaissent leur nouvelle poche.

Maintenant que la première est faite, on fait la même chose avec la deuxième

Dans le cas où les deux biomes sont le même, on le fait qu'avec la première liste, mais tout en actualisant la seconde poche de notre tuile. Les deux seront les mêmes, mais la tuile ne sera ajoutée qu'une seul fois à la poche.

## Diagramme d'activité :

Présent sur le git :

[https://grond.iut-fbleau.fr/TanchouCorp/SAE31\\_2024\\_LTA.git](https://grond.iut-fbleau.fr/TanchouCorp/SAE31_2024_LTA.git)



## **Explication changement ergonomique :**

### **Écran d'accueil :**

Au tout début, nous n'avions qu'un écran simpliste qui nous permettait juste de lancer une partie. Mais nous nous sommes vite rendu compte que c'était trop simpliste, alors nous avons décidé d'utiliser JavaFX pour le refaire.

Nous avons une vidéo en fond qui représentait une reproduction 3D du jeu Dorfromantik, nous avons mis le choix du nom du joueur, le choix de la série et le lancement de la partie sur la partie droite.

Mais après une discussion avec vous, nous avons appris que l'on n'avait pas le droit d'utiliser JavaFX alors nous avons recréer la même chose sans. La vidéo qui tournait en fond est devenu une image et nous avons perdu le flou pour voir l'image à travers sur la partie de droite.

Nous avons mis un message d'erreur ou de succès pour la connexion à la base de données qui prenait trop de place alors nous l'avons déplacé en haut à droite

### **Écran de jeu :**

L'écran de jeu actuelle ressemble beaucoup à celui que l'on a créé au début, avec les hexagones gris pour connaître les endroits où l'on peut poser une tuile. Les plus gros changements ont été décoratif, comme avec la pile de tuile en bas à gauche ou en essayant de créer une pseudo 3D sur les tuiles posés en créant des ombres.

Le vrai ajout ergonomique est surtout le bouton centrer qui apparait quand on s'éloigne trop du centre, car en se baladant, il est vite possible de se perdre.

Et par la suite, pour vraiment ne pas se perdre, nous avons ajouté le fait que l'on bloque le déplacement dans une direction où nous ne voyons plus de tuile.

Nous avons aussi déplacé les messages concernant l'accès à la base de données pour ne pas gêner la visualisation du score.

### **Écran partie jouer :**

Lui aussi fait avec JavaFX au début, a dû être changé par la suite, mais à part le changement qualitatif visuel, cette page est restée ergonomiquement la même.

Et encore pareil pour le pop-up de connexion à la base de données.

## Évaluation des scores :

Les scores étant relié à une série, les scores de la même série peuvent être comparés entre eux, mais il est assez difficile de comparer deux scores qui ne viennent pas de la même série car cela dépend fortement de la génération de la série, une série peut être plus “simple” qu’une autre, par exemple la même couleur apparaît beaucoup de fois d’affilé.

## Conclusion Personnelle :

Louis TANCHOU :

Ce projet m’a beaucoup apporté, étant donné que j’ai eu plein d’idée de DLC en plus pour le jeu j’ai creusé tout seul de mon côté ce qui m’a appris plein d’autres choses, le jeu en lui-même m’a beaucoup appris surtout sur la gestion des vues avec swing et la façon de peindre et de gérer un plateau de jeux (moteur 2D simpliste), la consigne nous empêchant de déclarer des lambdas m’a aussi été utile pour apprendre à mieux organiser, factoriser et séparer mon code en plusieurs parties, j’ai également appris à gérer un repository j’ai découvert une nouvelle façon de coder et ce projet m’a ouvert la voie à plein d’autres. Le seul point négatif c’est le deadline trop court pour mes idées.

Thomas RENAULT :

J’ai beaucoup apprécié ce projet, lors de son démarrage je m’attendais à quelque chose de simple car cela me faisait penser à un des projets de l’an dernier, mais ce ne fut pas le cas, étant donné que la forme des tuiles est différente, de plus la contrainte d’avoir la possibilité de placer 50 tuiles dans n’importe quelle direction, et fait que la tuile doit être adjacente à une autre tuile, ces différences avec mon projet de l’an dernier m’ont permis d’apprendre de nouvelles choses mais aussi d’appliquer des choses auxquelles je n’avais pensé qu’à la fin du projet de l’an dernier, telle que faire tourner la tuile, le hashmap pour ranger les tuiles, les menus, les déplacements avec le dragclick. Mais ce que j’ai le plus apprécié, c’est le fait de travailler en MVC, ce qui permet de mieux comprendre l’objectif des classes, surtout lorsque l’on travaille en équipe. J’ai aussi appris à utiliser les branches sur Git lors de ce projet.

Alexis HADJEM-BRANCHER :

Ayant déjà fait un projet inspiré du même jeu l'année dernière, je pensais que cela serait assez simple, mais je me suis trompé. Tout d'abord, coder avec la méthode MVC était nouveau pour moi, j'avais du mal à m'y faire, et au fur et à mesure que je créais d'autres classes, je me rendais compte que sur la première classe j'en faisais trop. Plus je progressais dans le projet, plus je m'améliorais et plus je me rendais compte de mes erreurs. J'avais aussi sous-estimé la difficulté de gérer les points, qui était plus compliqué à faire que ce que je pensais. Pour ce projet, nous sommes rentrés beaucoup plus en profondeur dans la création d'un jeu et j'ai trouvé ça satisfaisant de mener à bien un tel projet.