

RAPPORT LIFAPOO

TETRIS

NAJI ADAM & ATAMNA MIRYAM

Décembre ————— 2023

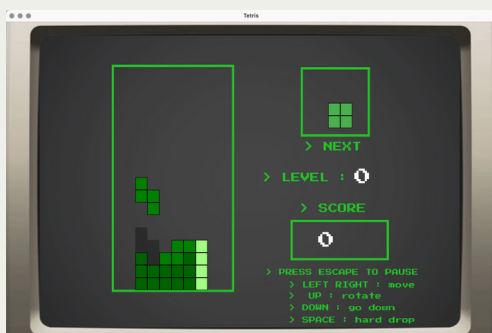
INTRODUCTION

Présentation du projet

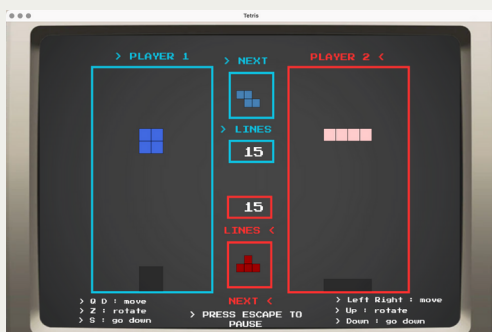
Nous avons récemment entrepris le développement d'une version revisitée du jeu Tetris. Les contraintes étant imposées de coder en Java de la manière la plus orientée objet possible en respectant l'architecture MVC strict, et en utilisant la bibliothèque Swing pour l'interface utilisateur graphique. L'objectif principal était de mettre en œuvre les concepts fondamentaux du Tetris, notamment la gestion des pièces et la logique du jeu, tout en ajoutant des fonctionnalités proposées par l'encadrant.

Le projet propose deux modes de jeu distincts : un mode solo offrant une expérience traditionnelle de Tetris et un mode multijoueur dynamique. En mode multijoueur, les joueurs ont la possibilité de s'affronter directement ou de défier une intelligence artificielle nommée Golmon spécialement conçue pour le Tetris. Cette IA peut être observée en action dans le mode solo ou constituer un adversaire "redoutable" en mode multijoueur. Cela ajoute une dimension stratégique au jeu, offrant aux utilisateurs la possibilité de tester leurs compétences contre une IA simple tout en créant une expérience multijoueur compétitive.

Aperçu du projet



< Partie solo



< Partie multijoueur



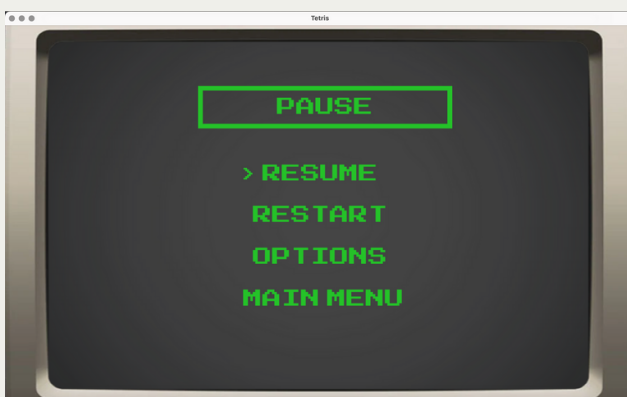
Menu principal

FONCTIONNALITÉS ET EXTENSIONS

Fonctionnalités

Dans notre Tetris, nous avons donc les fonctionnalités de base :

- Paramétrage du jeu : Nous avons implémenté un menu permettant aux joueurs de lancer différents types de parties, d'accéder au leaderboard, d'ajuster les options du son, et de quitter le jeu.
- Mouvement et rotation des pièces : Le jeu permet un mouvement fluide et une rotation des pièces. Cette fonctionnalité repose sur l'utilisation d'un clone de la pièce actuelle pour vérifier les collisions, garantissant ainsi des mouvements précis et évitant les conflits inattendus.
- Partie solo : Dans le mode solo, les joueurs peuvent profiter de sessions individuelles où l'objectif est de jouer jusqu'à ce que les pièces atteignent le sommet de la grille, en faisant le meilleur score possible.
- Pause : À tout moment pendant une partie, les joueurs peuvent mettre leur jeu en pause. Durant cette pause, ils ont la possibilité de régler les options du jeu, de continuer la partie, de la recommencer ou de quitter la partie.
- Score : Le score représente l'accumulation de points pendant la partie. À chaque ligne détruite, les joueurs gagnent des points. L'objectif du jeu est d'obtenir le meilleur score possible, ajoutant une dimension compétitive et incitant les joueurs à viser l'excellence.



Pause



Fin d'une partie solo

Extensions

Ensuite nous y avons ajouté les extensions suivantes

- Niveaux de difficultés : Selon l'avancé du score du joueur, les niveaux permet d'ajuster l'intensité du jeu en modifiant la vitesse de descente des pièces. À des niveaux de difficulté plus élevés, les pièces descendent plus rapidement, augmentant ainsi le défi pour le joueur. Cela offre une progression graduelle du défi.
- Sauvegarde des meilleurs scores : Selon le score du joueur, il a la possibilité d'apparaître dans le leaderboard, consultable à partir du menu qui contient les 5 meilleurs scores du jeu. Lorsqu'un joueur atteint un score remarquable, celui-ci est stocké dans un fichier, ce qui garantit une sauvegarde durable des performances individuelles, même en cas de fermeture du jeu. Cela permet aux joueurs de suivre leur progression au fil du temps et de rivaliser avec d'autres pour obtenir les meilleurs résultats, ajoutant ainsi un élément compétitif et incitatif à améliorer constamment ses résultats

LeaderBoard >

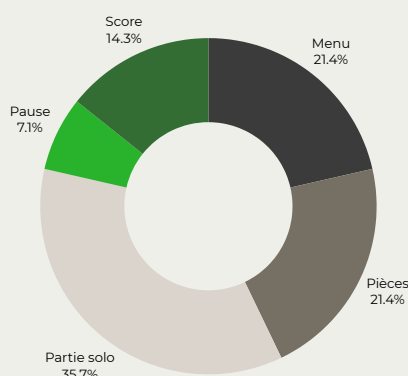


- Mode multijoueur : Si le joueur souhaite, il peut jouer simultanément dans le même jeu, avec un autre joueur, chacun ayant pour objectif de supprimer 15 lignes pour remporter la partie. Chaque joueur a son propre espace de jeu, et si l'un des deux joueur perd avant d'avoir pu supprimer les 15 lignes, c'est l'autre joueur qui remporte la partie. L'objectif compétitif est de survivre en supprimant le plus rapidement des lignes. Cette compétition ajoute une dimension stratégique au jeu. Ce mode permet d'affronter un autre joueur ou une intelligence artificielle nommé Golmon. Les joueurs peuvent ainsi affronter une IA ce qui étend la diversité des options de jeu, offrant aux joueurs la flexibilité de choisir entre des adversaires humains ou une IA compétitive lorsqu'ils explorent le mode multijoueur de Tetris.

- Rendu graphique et sonore : Notre rendu graphique comprend quatre vues distinctes : vueMenu, vueSolo, vueMulti, et VC chacune visant à un but spécifique dans le jeu. Pour coordonner ces vues, nous avons utilisé un JFrame (VC) en tant que vue principale, assurant la gestion globale de l'interface utilisateur. La vueMenu représente l'écran d'accueil, permettant de naviguer entre les différents modes de jeu et options. La vueSolo est dédiée au mode de jeu solo, la vueMulti à celui du multijoueur, et VC agit comme le contrôleur principal, coordonnant les interactions entre ces vues et gérant la logique globale du jeu. De plus, nous avons intégré une expérience sonore complète dans notre Tetris en utilisant trois musiques distinctes, visant à offrir une ambiance sonore adaptée à chaque aspect spécifique du jeu. Le joueur a la possibilité d'ajuster le volume en augmentant, diminuant ou coupant le son selon ses préférences, à partir du menu options.
- Implémentation d'une IA : Notre intelligence artificielle (IA), que nous avons nommée "Golmon", constitue une composante stratégique de notre Tetris, offrant la possibilité de le voir performer une partie solo et d'autre part la possibilité de défier le joueur en mode multijoueur. Le fonctionnement de l'IA repose sur un ensemble de méthodes clés qui guident ses décisions de jeu de manière réfléchie. La méthode golmonPlay() orchestre le mouvement de Golmon en choisissant la meilleure position possible parmi celles générées. La sélection de cette position optimale est déléguée à la méthode choisirMeilleurePosition(), qui évalue les différentes positions possibles en fonction de la hauteur, calculé dans la méthode calculHauteurGrille(Grille grille), résultante de la grille après placement. Le processus de sélection se base sur des critères stratégiques. La méthode genererPositionsPossibles() explore les positions potentielles de la pièce actuelle en fonction de sa forme, et Golmon choisit celle qui minimise la hauteur cumulative de la grille après le placement. Cette approche intelligente permet à Golmon de prendre des décisions tactiques, favorisant des hauteurs de grille optimales. Ainsi, que ce soit en mode solo, où les joueurs peuvent observer Golmon en action, ou en mode multijoueur, où ils peuvent affronter cette IA "redoutable", Golmon offre une expérience de jeu stimulante et compétitive.

Répartition du temps

Nous avons réalisé notre projet Tetris dans un délai de quatre semaines, consacrant les deux premières semaines au développement des fonctionnalités essentielles du jeu, puis les deux semaines suivantes à l'implémentation des extensions.



< Semaine 1 et 2

Semaine 3 et 4 >

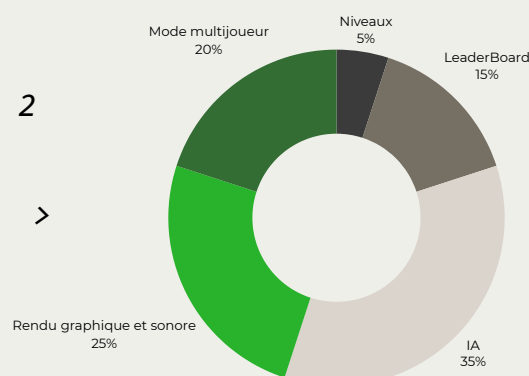


DIAGRAMME DES CLASSES

Cf Annexe

CONCLUSION

En conclusion, ce projet a été une opportunité enrichissante pour développer nos compétences en programmation Java et en interface graphique Swing. Plus particulièrement l'implémentation d'une intelligence artificielle pour les modes solo et multijoueur a représenté un défi stimulant, nous offrant une expérience pratique dans la conception et l'intégration d'algorithmes de prise de décision. De plus, ce projet nous a permis d'apprendre et d'appliquer les principes de la programmation orientée objet, qui nous sera forcément utile dans la suite de nos études.

