

Projet : Tetris

CP1 : Groupe de TD2

Réalisé par :

Kenmegne Kamdem Yvan Junior

Sous la supervision de :

D. Robillard, M. A. Liefoghe

SOMMAIRE

INTRODUCTION.....	3
I. ANALYSE.....	4
1. Description du jeu	4
2. Règles et mécanismes.....	4
II. CONCEPTION.....	6
2. Structures de données	6
3. La mécanique de jeu	7
4. La gestion des données de jeu	9
4. L'interface utilisateur	10
III. IMPLEMENTATION	11
1. Structures de données et prototypes de fonctions	11
2. La mécanique de jeu	11
3. La gestion des données	13
5. L'interface	14
IV. DIFFICULTES RENCONTREES	15
V. PERSPECTIVES.....	16
CONCLUSION	17
ANNEXE.....	18

INTRODUCTION

Ce rapport détaille le développement d'un jeu Tetris réalisé en Python. Le projet visait à créer une version fonctionnelle et interactive du jeu classique, en se concentrant sur l'implémentation des règles d'une Tetris et une interface utilisateur intuitive. Le jeu Tetris est un jeu de puzzle classique où les joueurs doivent assembler des blocs de différentes formes pour former des lignes complètes, qui disparaissent ensuite pour laisser place à de nouveaux blocs. Pour ce projet, notre objectif était de créer une version simple du jeu Tetris en utilisant le langage de programmation Python.

I. ANALYSE

1. Description du jeu

Le jeu Tetris est un jeu de puzzle emblématique créé par Alexey Pajitnov en 1984. Le but du jeu est de manipuler des blocs de différentes formes appelés "tétrminos" qui descendent du haut de l'écran de jeu. Les tétrminos sont composés de quatre carrés et peuvent être de différentes formes : carré, ligne, T, L, J, S et Z.

Le joueur peut déplacer et faire pivoter les tétrminos pour les faire tomber et les placer de manière à former des lignes horizontales complètes. Lorsqu'une ligne est complète, elle disparaît, libérant de l'espace pour de nouveaux tétrminos. Le joueur obtient des points lorsqu'une ou plusieurs lignes sont remplies de tétrminos. Le jeu se termine lorsque les tétrminos empilés atteignent le haut de l'écran.

Tetris est un jeu simple mais incroyablement addictif, et sa jouabilité intuitive en a fait l'un des jeux les plus populaires de tous les temps. Il a été décliné sur de nombreuses plateformes de jeu et a connu de nombreuses évolutions tout en conservant ses mécaniques de base. Son concept intemporel et sa difficulté progressive en font un jeu qui peut être apprécié par les joueurs de tous niveaux.

2. Règles et mécanismes

1. Objectif du jeu :

- L'objectif du jeu Tetris est de faire descendre les tétrminos du haut de l'écran et de les placer stratégiquement pour former des lignes horizontales complètes.
- Le joueur doit effacer autant de lignes que possible pour marquer des points et éviter que les tétrminos ne s'accumulent jusqu'en haut de l'écran.

2. Les tétrminos :

- Il y a sept types de tétrminos différents :
- Le tétrmino carré (constitué de 4 carrés identiques)
- Le tétrmino ligne (une ligne de 4 carrés)
- Le tétrmino T
- Le tétrmino L
- Le tétrmino J
- Le tétrmino S

- Le tétrimino Z
- Chaque tétrimino peut être déplacé horizontalement, tourné et laissé tomber vers le bas de l'écran.

3. Déplacement et rotation des tétriminos :

- Le joueur peut déplacer un tétrimino vers la gauche ou la droite en utilisant les touches directionnelles du clavier.
- Le joueur peut faire pivoter un tétrimino dans le sens des aiguilles d'une montre en appuyant sur la touche de rotation.
- Le joueur peut laisser tomber un tétrimino instantanément en appuyant sur la touche de descente rapide.

4. Lignes complètes et score :

- Lorsqu'une ligne horizontale est complètement remplie par des tétriminos, elle disparaît et toutes les lignes au-dessus descendent d'un cran.
- Le joueur marque des points pour chaque ligne complétée et éventuellement pour les "Tetris" (quatre lignes complétées simultanément).
- Plus le joueur fait disparaître de lignes simultanément, plus il obtient des points.

5. Fin de partie :

- La partie se termine si les tétriminos empilés atteignent le haut de l'écran et ne peuvent plus descendre.
- Le joueur peut alors consulter son score pour évaluer sa performance.

II. CHOIX DES TECHNOLOGIQUES

- 1- Langage : Python a été choisi pour sa simplicité, sa lisibilité et ses bibliothèques graphiques.
- 2- Bibliothèque graphique : Pygame a été utilisé pour gérer l'affichage graphique, les événements utilisateur et la gestion du temps.
- 3- Bibliothèque : Numpy qui a été utilisé pour faire le tableau de jeu ou circule les tétraminoes
- 4- D'autres appel de fonction tels que : « random »

III. CONCEPTION

Le code source est organisé en plusieurs modules pour assurer une meilleure modularité et maintenabilité :

* 'jeuxtetris.py' : Contient le programme principal, gérant la boucle de jeu principale et l'interaction entre les modules.

* 'Tetramino.py' : Définit la classe 'Tetramino', représentant un tetramino individuel de Tetris (Tetromino).

* 'TableauDeJeu.py' : Définit la classe 'Grille', représentant la grille de jeu et contenant les méthodes pour la manipulation des blocs et la détection des lignes complètes.

* 'interface.py' : Qui nous été fourni pour gérer l'affichage du jeu à l'aide de Pygame, incluant le rendu des blocs, du score et des autres éléments visuels.

* 'Point.py' : Qui permet de représenter les blocs du tetraminos .

2. Structures de données

1. Grille de jeu :

- La grille de jeu est généralement représentée comme une matrice de dimensions 10x20 dans la version classique de Tetris.
- Chaque case de la grille peut être vide ou occupée par un tétrimino en cours de chute.

2. Tétriminos :

- Les tétriminos sont des formes composées de quatre cases.
- Chaque tétrimino est stocké sous forme de tableau 2D ou de liste de coordonnées relatives à sa position dans la grille.
- Lorsqu'un tétrimino est en mouvement, ses coordonnées sont mises à jour pour refléter sa nouvelle position.

3. Lignes complètes :

- Les lignes complètes sont détectées en parcourant chaque ligne de la grille pour vérifier si toutes les cases sont remplies.
- Lorsqu'une ligne est complète, toutes les lignes au-dessus d'elle sont décalées vers le bas pour remplir le vide.
- Cette opération impacte la grille en modifiant la disposition des cases et en libérant de l'espace pour de nouveaux tétriminos.

4. Score et statistiques :

- Le score, le niveau et les statistiques du joueur sont généralement stockés sous forme de variables.
- Le score est incrémenté chaque fois qu'une ligne est complétée ou qu'un tétrimino est placé.
- Les statistiques, telles que le nombre de lignes complétées ou le temps de jeu, peuvent être affichées à l'écran pour informer le joueur de sa progression.

3. La mécanique de jeu

La mécanique du jeu constitue le cœur du projet de Tetris, dictant le comportement du jeu en réponse aux actions du joueur. Mon approche a consisté à diviser cette mécanique en plusieurs composants, chacun responsable d'aspects spécifiques du jeu.

1. Chute des Tétriminos :

- Au début de chaque niveau, un tétrimino est généré en haut de la grille et commence à tomber vers le bas.
- Le joueur peut déplacer latéralement le tétrimino et le faire tourner pour le positionner de manière optimale avant qu'il ne touche le bas.

2. Mouvement et Rotation des Tétriminos :

- Les tétriminos peuvent être déplacés vers la gauche ou la droite par le joueur pour les positionner dans la meilleure colonne.
- Il est également possible de faire pivoter les tétriminos dans le sens horaire pour les ajuster dans des espaces restreints.

3. Marquage des Lignes Complètes :

- Lorsqu'une ligne horizontale est entièrement remplie par des tétriminos, elle est "marquée" comme complète.
- Les lignes complètes disparaissent de la grille, permettant aux tétriminos au-dessus de descendre et libérant de l'espace de jeu.

4. Fin de Partie :

- La partie se termine si un tétrimino atteint le sommet de la grille et ne peut plus descendre.
- Le score du joueur est calculé en fonction du nombre de lignes complètes effacées.

4. Méthodologie

Pour atteindre ces objectifs, j'ai divisé le travail en plusieurs étapes :

1. Compréhension des règles du jeu Tetris
2. Conception de l'architecture du jeu en utilisant des diagrammes UML
3. Programmation des différentes fonctionnalités du jeu en Python
4. Test et débogage pour garantir son bon fonctionnement

IV. IMPLEMENTATION

1. Classe : Point

J'ai écrit classe point ayant les méthodes tels que :

- La méthode « **déplacement** » qui permet de faire déplacer les points du tetramino.
- La méthode « **add** » qui permet d'ajouter deux points.
- La méthode « **multiplication** » qui m'a permis de multiplier un point par une matrice 2×2 qui est en réalité une liste de deux points.
- La méthode « **rotation** » qui multiplie un point par la matrice de rotation des angles (0,90,180,270) degré.

2. La classe : Tetramino

Dans cette classe j'ai écrit les méthodes tels que :

- Le constructeur « **__init__** » ou j'ai initialisé la position et le degré de rotation du tetramino.
- La méthode « **tourne** » qui tourne le tetramino dans le sens des aiguilles d'une montre en ajoutant 90 à chaque fois au degré de rotation.
- La méthode « **image** » je retourne une liste qui a chaque point dans la forme du tetramino affecte un déplacement, soit pour roter le tetramino, soit pour le déplacer.
- La méthode « **translater** » ajoute un vecteur déplacement à la position du tétramino.
- La méthode « **cloner** » qui retourne une copie du tetramino.

3. La classe : TableauDeJeu

Dans cette classe j'ai écrit les méthodes tels que :

- Le constructeur « **__init__** » ou j'ai initialisé un tableau Numpy rempli de zéro, le score et le nombre de ligne à supprimer.
- La méthode « **generer_tetramino** » qui prend en paramètre l'écran qui est une instance de la classe l'interface. Je définis la liste des tetramino possible et dans cette liste on choisit aléatoirement un tetramino dans cette liste, puis on la retourne avec une couleur spécifique pour chaque tetramino.
- La methode « **dessiner** » premièrement cette méthode permet d'effacer le tetramino a sa position précédente, deuxièmes elle dessine les tetraminos qui sont déjà tombe et troisièmement elle dessine le tetramino à sa nouvelle position
- La méthode « **calculer_score** » qui prend en paramètre le nombre de ligne complète. Elle vérifie le nombre de lignes complétés et ajoute au score 40 si c'est une seule ligne, 200 si c'est deux lignes, 300 si c'est trois lignes et 400 si c'est quatre lignes complétées, puis retourne le score.
- La méthode « **verifier_lignes_complete** » qui initialise le nombre de lignes complet à zéro et parcourt le tableau Numpy pour identifier les lignes qui n'ont pas de zéro sur toutes leurs colonnes, puis compte ces lignes au fur et à mesure et retourne le résultat.
- La méthode « **verifier_collision** » qui vérifie à chaque fois que les points du tetramino ne dépassent pas la hauteur et la largeur du tableau de jeu.
- La méthode « **est_partie_finie** » qui parcourt la première ligne du tableau Numpy pour vérifier s'il y a un tetramino.

5. La boucle principale du jeu : « jeuxetris.py »

Dans cette partie j'ai fait appel à toutes mes classes et leurs méthodes pour réaliser le jeu le jeux Tetris proprement dit.

V. Discussions

Le projet a permis d'approfondir les connaissances en Python en général et l'utilisation des bibliothèques tel que Numpy et Pygame, bien qu'il nous a été demandé de ne pas l'utiliser mais il est utilisé dans l'interface graphique, notamment en ce qui concerne la programmation orientée objet, la gestion des événements et l'affichage graphique.

5.1 Difficultés rencontrées

La principale difficulté rencontrée a été la gestion des collisions entre les tetraminos, la gestion de la rotation des tetraminos et la grille.

5.2 Améliorations possibles

Le jeu pourrait être amélioré en ajoutant des fonctionnalités supplémentaires, telles que :

- * Différents niveaux de difficulté.
- * Possibilité de mettre le jeu en pause.
- * Effets sonores.
- * Un menu de démarrage.

CONCLUSION

En conclusion, la réalisation de ce jeu Tetris en Python a été enrichissante sur le plan technique et pédagogique. Ce projet nous a permis de mettre en pratique nos connaissances en programmation et de développer de nouvelles compétences. Nous sommes fiers du résultat final et espérons que les joueurs apprécieront notre jeu.

ANNEXE

Déroulement d'une partie

