Aurélien Laforge / Nathan Faganello

Rapport IN204

Projet TETRIS

I – Architecture du projet

L'architecture de notre projet TETRIS est assez simple. Nous avons séparé notre code en 4 fichiers .cpp (et .h associé) :

- Main.cpp : C'est dans ce fichier que nous avons placé notre boucle principale de jeu. Sa structure est détaillée dans le READ.me sur le GIT.
- Pieces.cpp/Pieces.h : on retrouve dans ce fichier la classe piece, représentative des pièces du jeu mais aussi le tableau int pieces[NbPieces][NbRota][4][4], représentant toutes les pièces et toutes les rotations possibles ainsi que les enums Couleur et Type, nous permettant d'y voir plus clair dans l'organisation des pièces.
- Board.cpp/board.h : ce fichier contient la classe Board, représentative du plateau de jeu, incluant toutes les fonctions permettant de déplacer les pièces, supprimer une ligne, …
- Game.cpp/game.h : on retrouve dans ce fichier les fonctions nécessaires au fonctionnement du jeu comme les fonctions de calcul du score et d'affichage graphique.

Nous avons choisi cette organisation car c'est celle qui nous est venu naturellement et qui nous paraissait la plus simple. Dans la pratique, elle s'est révélée être assez efficace et nous a permis d'y voir clair lors de l'implémentation du jeu et de nous répartir efficacement le travail.

II – les différentes classes utilisées

Pour l'implémentation du Tetris, nous avons eu besoin de créer 2 grandes classes principales : Board et Pieces.

- Board : Cette classe est la première qui nous est apparue indispensable. Elle implémente un plateau de jeu sous forme d'un tableau d'entiers. Chaque entier représente une couleur de pièce (y compris la couleur « VIDE »).

8	8	8
8	8	8
8	8	8
1	8	8
1	4	8
1	4	8
1	4	4

(Tableau de taille 7x3, avec un I en bas à gauche et un L au milieu)

Cette classe contient 3 champs :

(i) Int[][] Plateau : correspond au plateau de jeu, initialisé avec toutes les cases vides.

(ii) Piece Piece_courante : correspond à la pièce qui est actuellement en mouvement sur le plateau. Il s'agit de la pièce que l'on manipule.

(iii) Piece Piece_suivante : il s'agit de la pièce qui arrivera quand la pièce courante sera posée. Avoir cette pièce en mémoire nous permet de l'afficher au joueur en avance.

Ensuite, cette classe contient de nombreuses méthodes manipulant ces 3 champs. On y retrouve donc toutes les méthodes visant à déplacer/tourner les pièces, à créer des pièces, à supprimer des lignes.

- Pieces : après de longues réflexions sur comment représenter les différentes pièces du jeu, il nous a paru évident de devoir coder une classe Pieces. Tout d'abord, chaque pièce est représentée sous la forme d'une matrice 4x4 d'entiers afin de coller à la représentation du plateau. Cette matrice est constituée de 0 là où il n'y a pas de pièce, de 1 là où il y a une brique et d'un 2 pour la brique qui sert de centre de rotation. Ce 2 sert également ensuite de point de référence puisque c'est de cette brique qu'on se sert pour avoir la position de la pièce. Toutes les pièces et leurs rotations sont stockées dans un tableau de taille NbPieces x Nbrotations x 4 x 4 .

  La classe Piece est composée des champs : int type, int rota, int posX, int posY, Couleur couleur ainsi que de tous les getter/setter associés.

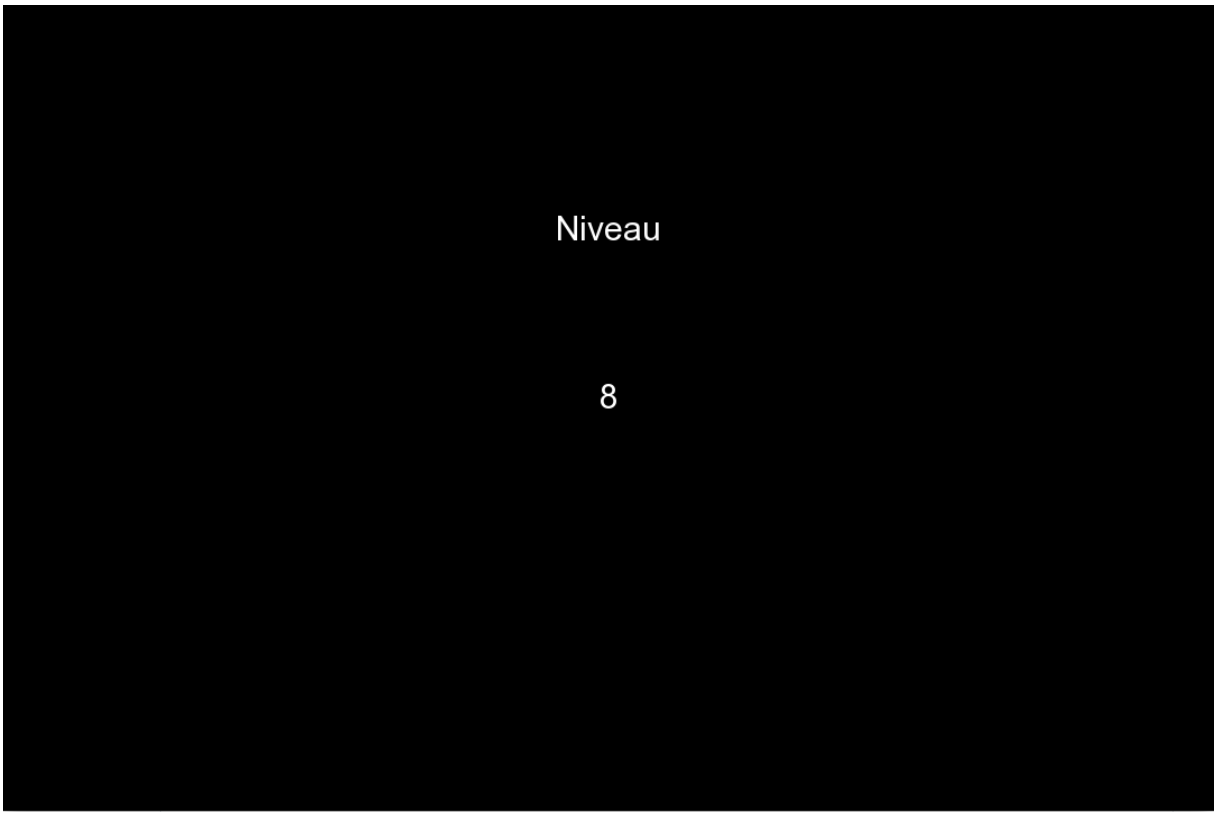
  Les champs type et rota permettent de retrouver la bonne représentation de la pièce dans le tableau de toutes les pièces/rotations.

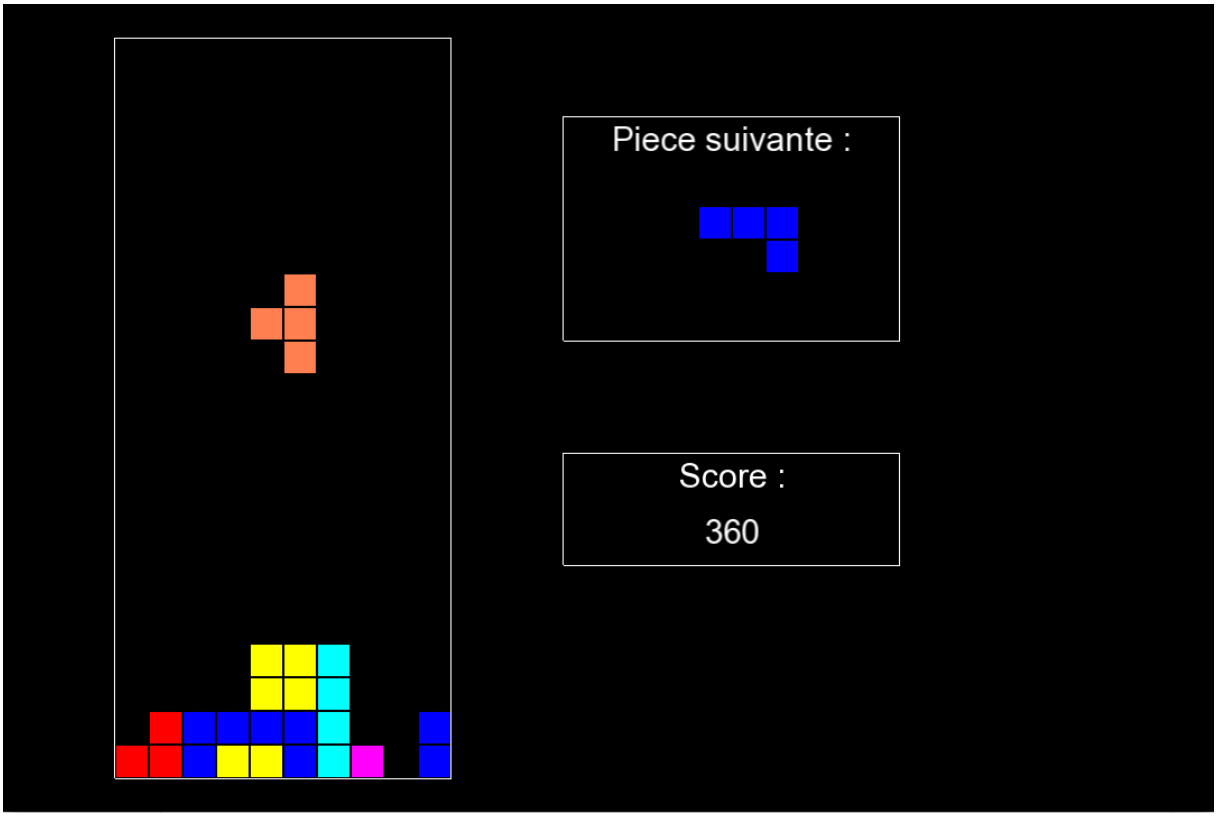
III – Le jeu

Notre jeu Tetris peut se jouer en solo ou en multijoueur, et selon différents niveaux. Il est laissé le choix au joueur au démarrage du jeu de choisir les options qui lui conviennent. Voici quelques images illustratives :


Bienvenue dans notre Tetris
Pour parcourir les differents menus, utilisez les fleches directionnelles.
Appuyez sur Entree pour valider votre choix.
Appuyez sur Echap pour revenir en arriere ou mettre le jeu en pause.
Appuyez sur Entree pour continuer


Choix du mode
Solo
2 joueurs





(i) Solo :

Le jeu solo fonctionne très bien. Toutes les fonctions du Tetris sont implémentées : chute accélérée des pièces, rotation, déplacement. Il manque peut-être seulement le

kick des pièces lorsque l'on souhaite les tourner alors qu'elles sont sur le bord. Le jeu s'arrête lorsque la pièce suivante ne peut pas apparaître sur le plateau.

(ii) Multijoueur :

Malheureusement, nous avons codé toute une partie multijoueur que nous n'arrivons pas à utiliser. En effet, nos ordinateurs n'arrivent pas à se connecter l'un à l'autre au début du jeu. Nous n'avons donc pas pu tester le code que nous avons réalisé.