Tetris : User guide

Fonctionnement du jeu

Le jeu se démarre dans la classe « Programm_entry ». L'utilisateur voit le menu apparaitre, il a alors 3 choix. Pour sélectionner une des options du menu, l'utilisateur doit utiliser les flèches du clavier et la touche entrée. Il peut commencer à jouer en sélectionnant « play ». Le niveau par default sera 1. S'il veut changer de niveau, l'utilisateur peut aller dans setting sélectionner son niveau à l'aide des flèches droite et gauche, et retourner dans le menu. Le niveau 2 sera plus rapide que le niveau 1.

Une fois que le joueur a cliqué sur play, le jeu Tetris se lance suivant les règles du Tetris. Le but du jeu étant de remplir le plus de lignes possibles avec les blocs qui tombent. Le joueur a perdu lorsque les blocs atteignent le haut du plateau de jeu. A chaque fois qu'un bloc se pose le joueur gagne 50 points et lorsqu'il arrive à remplir une ligne il gagne 500 points de plus. Le score dépend aussi du niveau. Si le joueur est au niveau 2, il gagnera 2 fois plus de points qu'au niveau 1 pour les mêmes actions.

Explication des classes

Le jeu se compose de 7 classes :

Programm_entry : C'est la classe main de ce programme, c'est dans cette classe qu'on va lancer le jeu. Cette classe est composé de deux méthodes. La première méthode « Main » lance le jeu en appelant la classe Menu (« Interface ») ainsi que la deuxième méthode de cette classe « Generate_console » qui permet de générer une console, son titre et de définir ses dimensions.

Interface : C'est la classe qui permet la création du menu. Cette classe est composée de diverses méthodes qui permettent de gérer l'affichage du menu et aussi le choix de l'utilisateur.

Options : C'est la classe qui gère le choix du niveau de l'utilisateur ainsi que la vitesse de jeu en fonction du niveau.

Gameboard : C'est la classe qui va permettre de dessiner le plateau de jeu ainsi que d'afficher le score et le compteur de nombre de lignes pleines. Les compteurs de score et de lignes sont calculés dans la classe Game.

Shape : C'est la classe qui va permettre de construire les blocs ainsi que de gérer leurs angles de rotations. Nous avons choisis de représenter 5 tétriminos (I, O ,Z, T, L). Pour dessiner ces formes nous avons créé des tableaux de dimensions différentes selon la forme des tétriminos, et qui vont être rempli de 1 et de 0 selon leurs rotations. Avec ses tableaux nous pouvons afficher des carrés de couleur aux endroits où se trouvent les 1 dans les tableaux. Quand l'utilisateur appuie sur une des flèches droite ou gauche du clavier, la forme forme précédente est effacée pour laisser place à la nouvelle rotation. Nous avons également défini un array virtuel en mémoire qui représente le plateau de jeu graphique. A

chaque fois qu'un bloc est posé, cet array est rempli de 1 aux endroits où se trouve le bloc. C'est avec ce tableau qu'on va pouvoir détecter et supprimer les lignes pleines. Cette classe nous permet aussi de gérer les collisions des blocs entre eux. Au fur et à mesure que les blocs descendent, une méthode check_collision va parcourir toutes les cases voisines dans l'array pour savoir si on entre en collision avec un bloc déjà posé. Si un bloc rencontre un autre bloc par le bas alors il s'arrête. Au moment où l'utilisateur appuie sur les flèches du haut, de gauche ou de droite, la méthode va parcourir les cases voisines en anticipant la position de la forme pour déterminer si l'action est possible. Par exemple s'il y a déjà une forme à gauche ou à droite, l'utilisateur ne pourra pas déplacer la forme qui descend. Idem s'il veut faire tourner une forme mais qu'il n'y a pas la place. Ces vérifications sont également faites en bordures du plateau.

Game : C'est la classe qui permet de gérer le déroulement du jeu. Dans cette classe on appelle les classes Shape et Gameboard. Dans cette classe on gère la suppression d'une ligne pleine, la fin du jeu. On met en lien les différentes classes pour générer graphiquement les éléments dont on a besoin et pour gérer l'utilisation des flèches du clavier.

Utilities_fonctions : C'est la classe qui gère les fonctions annexes au jeu. Par exemple non peut retourner l'indice d'une valeur dans un tableau et également l'indice suivant.