

Jeu de Tetris et génération de polyominos

Enrico Formenti
enrico.formenti@unice.fr

February 6, 2016

Nombre d'étudiants souhaités: 2

1 Description du sujet

Le but du stage est de programmer une variante simplifiée du classique jeu de Tetris. Le jeu doit être programmé en C ou C++ en utilisant une bibliothèque graphique standard (par exemple SDL). Le programme doit permettre de sauvegarder une partie en cours et de la reprendre plus tard.

Les étudiants devront en particulier intégrer un algorithme très récent de génération exhaustive de polyomino préfixes et d'en permettre la visualisation sur la scène de jeu.

Une attention particulière devra être accordée à la documentation du code, à la réalisation du manuel utilisateur et celui du programmeur. Pour cela, l'utilisation de l'outil Doxygen est à privilégier.

1.1 Le jeu de Tetris classique

Le jeu de Tetris classique prévoit l'utilisation d'un certain nombre de polyominos, dits *pièces*, sur une scène de jeu. La figure 1 illustre l'ensemble de ces pièces à une rotation près.

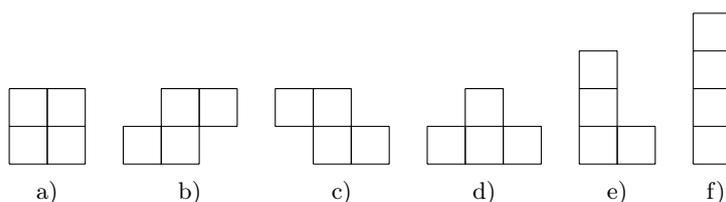


Figure 1: Pièces du jeu de Tetris classique (à une rotation de 90° près). Par la suite on colorera les pièces pour des raisons esthétiques mais la couleur n'a aucune importance au niveau du jeu.

La scène de jeu est un rectangle de dimension $w \times h$ ouvert dans la partie haute (voir figure 2). Une pièce est proposée au joueur tout en haut de la scène et tombe vers le bas au cours du temps. La pièce s'empile sur les autres pièces déjà en place sur la scène ou quand elle arrive sur le fond. L'utilisateur a la possibilité de bouger la pièce vers la droite ou vers la gauche ou de la faire tourner d'un multiple de 90° (voir la pièce bleu de figure 2 par exemple).

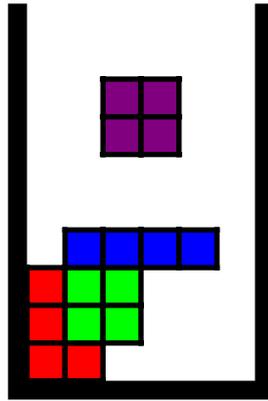


Figure 2: une partie de jeu de Tetris classique.

1.1.1 La règle d'effacement

Si pendant la partie les pièces s'accumulent de sorte qu'une ligne horizontale du terrain de jeu soit complètement remplie alors on applique la *règle d'effacement* qui consiste à effacer complètement toute la ligne. Tout carré se trouvant au dessus de cette ligne tombe verticalement jusqu'à qu'il trouve une autre pièce ou le fond de la scène. Si plusieurs lignes sont remplies au même temps par une même pièce alors toutes ces lignes sont effacées en même temps. La figure 3 illustre l'application de cette règle. Remarquez que dans ce cas deux lignes s'effacent en même temps et que de la pièce rouge toute à droite subsiste un seul carré qui tombe tout au fond de la scène car il ne rencontre aucun obstacle. Par contre le carré plus à gauche de la partie supérieure de la pièce de type *a*) violette s'est arrêtée à la ligne 4 car elle a rencontré un obstacle (pièce de type *a*) verte).

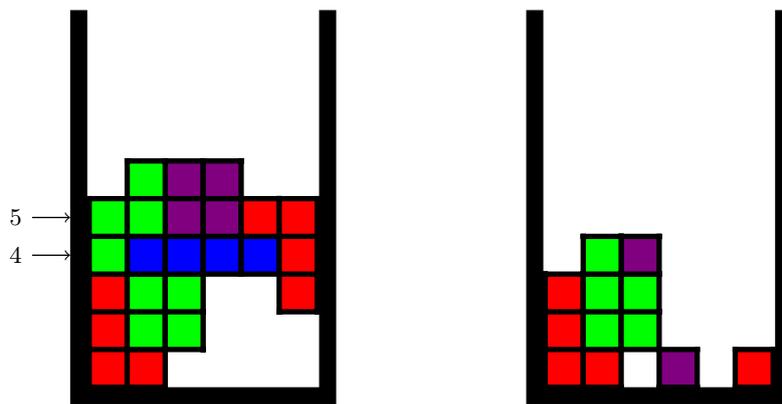


Figure 3: une partie de jeu de Tetris classique. Les lignes 4 et 5 sont *remplies (gauche)* et donc on applique la règle d'effacement (*droite*).

1.1.2 Fin d'une partie

La partie termine si le joueur empile une pièce à une position qui dépasse la hauteur h de la scène.

1.2 La variante à implémenter

Nous voulons implémenter une version simplifiée du Tetris classique dans laquelle seulement les pièces de type f) (voir figure 1) sont autorisées et la règle d'effacement est interdite. Par contre, afin de rendre la partie plus intéressante, nous allons imposer deux contraintes: *empilement* et *surface*. De plus, nous autorisons, dans une certaine mesure, les dépassements de marge (voir plus loin).

1.2.1 La contrainte d'empilement

Cette contrainte impose que chaque pièce doit forcément s'empiler sur une pièce qui est déjà sur la scène à l'exception, bien sûr, de la toute première pièce. Si la contrainte n'est pas respectée alors la pièce est rejouée jusqu'à que la contrainte est respectée par le joueur.

1.2.2 La contrainte de surface

Au début du jeu on fixe la surface totale S des pièces qui seront tirées. La surface d'une pièce est le nombre de carrés unité qui la forment. Par exemple, les pièces en figure 1) ont toutes surface 4 unités. Notons $s(p)$ la surface de la pièce p et supposons que jusqu'à l'étape t du jeu on ait tiré les pièces p_1, p_2, \dots, p_t , alors à l'étape successive nous aurons le droit de tirer une pièce de surface au plus $S - \sum_{i=1}^t s(p_i)$.

1.2.3 La règle de dépassement de marge

Quand une pièce est déposée, il est autorisé de dépasser les marges droit, gauche ou haut de la scène pourvu que la contrainte d'empilement soit respectée. A tout fin pratique, cela signifie que la scène est agrandie de la longueur du dépassement. La figure 4 montre un exemple d'application de cette règle par rapport au marge droit.

1.2.4 Fin d'une partie

La partie termine quand la contrainte de surface ne peut plus être satisfaite, c'est-à-dire quand la surface disponible à tirer est 0.

2 Lieu du stage

Les étudiants peuvent tranquillement développer le programme chez eux. Par contre, il est demandé d'avoir des entretiens réguliers (par exemple une fois par semaine 1h) avec l'encadrant pour reporter les progrès, discuter des points de difficulté, des choix d'implémentation, *etc.* Les entretiens auront lieu dans le bureau de l'encadrant au laboratoire l'I3S.

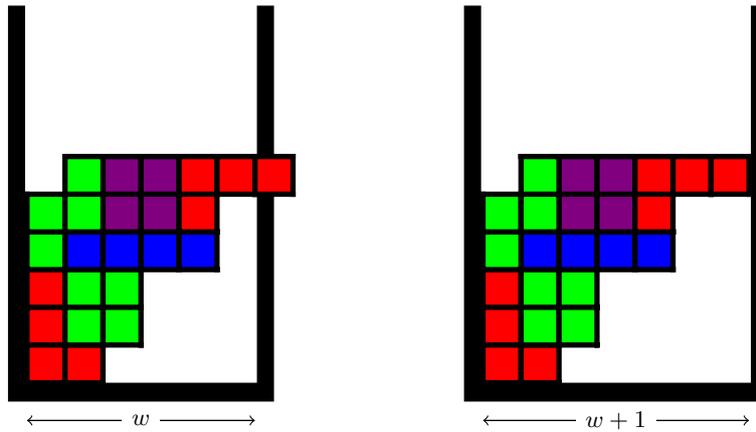


Figure 4: dépassement de marge pendant le jeu (*gauche*) et nouvelle scène de jeu ajustée (*droite*).

Prérequis

Aucun prérequis particulier sinon un bon cours de programmation C ou C++ et un bon cours d'algorithmique et structure des données.