

Programmation orientée objet

TD6

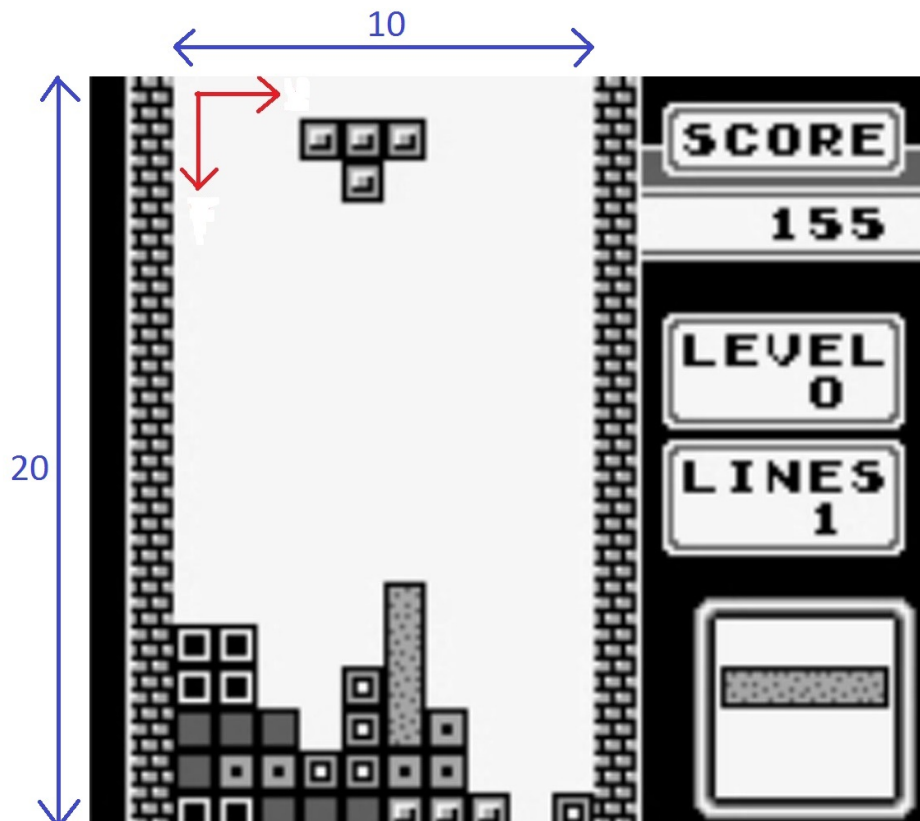
Romain Marie & Hervé Gaudin

Octobre 2019

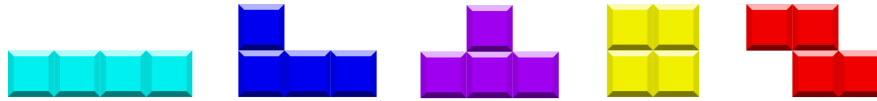


Dans ce TD, on va s'intéresser à la façon avec laquelle on pourrait (re)-programmer le célèbre jeu Tetris[®]. On va pour l'instant uniquement se concentrer sur le remplissage de la grille, c'est à dire la création et le déplacement des différentes pièces, ainsi que la suppression d'une ligne lorsque celle-ci est pleine. Nous laissons donc de côté l'aspect graphique !!

Dans ce jeu, des formes descendent dans une grille de dimensions 10×20 dont nous fixons arbitrairement l'origine dans le coin supérieur gauche (voir Figure 1).



Chaque forme est composée de 4 carrés pouvant être répartis de 5 façons différentes (+ 2 symétriques) :



A chaque instant, nous considérerons que le joueur peut effectuer 4 actions :

- Demander à la forme de se décaler vers la gauche
- Demander à la forme de se décaler vers la droite
- Demander à la forme de descendre d'une case
- Demander à la forme de tourner (dans le sens horaire ou anti-horaire d'un quart de tour, suivant un centre de rotation à choisir)

Bien sûr, les mouvements de la forme doivent être réalisables, c'est à dire qu'aucun de ses 4 carrés ne doit sortir des limites de l'espace de jeu, ni arriver à une position déjà occupée par un autre carré !

1 Modélisation du problème

1. Quelles classes vous paraissent nécessaires pour programmer ce jeu ?
2. Proposez les propriétés et méthodes que vous jugez nécessaires pour chacune d'entre elles.
3. Décrivez le contenu des algorithmes suivants :
 - (a) Descendre un carré
 - (b) Descendre une forme
 - (c) Tourner une forme en Carré
 - (d) Tourner une forme en S
 - (e) Initialiser une forme en haut de la grille (si possible !).
 - (f) Tester si une ligne est pleine
 - (g) Supprimer une ligne pleine
4. Décrivez le contenu de la méthode `main()`, qui simulera le jeu jusqu'à ce qu'une forme initialisée en haut de la grille soit en collision avec au moins un carré déjà en place.