



# Tetris

REMISE 2

Mike Sarton & Olivier Cordier | Développement en C++ avec M. Wahid | 15 avril 2017

## Table des matières

<b>Introduction .....</b>	<b>2</b>
<b>Présentation des fonctionnalités du jeu .....</b>	<b>3</b>
<b>Les commandes du jeu .....</b>	<b>3</b>
<b>La partie graphique .....</b>	<b>3</b>
La partie de gauche .....	3
La partie de droite .....	3
<i>Le side board</i> .....	3
<i>Le choice board</i> .....	3
<b>La partie son .....</b>	<b>4</b>
<b>Les ajouts par rapport à l'énoncé.....</b>	<b>4</b>
La Shadow .....	4
La non immobilisation de la figure courante dès qu'elle touche un obstacle .....	4
La figure courante descend en permanence.....	4
<b>Les problèmes rencontrés / Les solutions implémentées .....</b>	<b>4</b>
<b>Les écarts .....</b>	<b>5</b>
<b>La compilation et les bugs .....</b>	<b>5</b>

# Introduction

Ceci est notre rapport de la remise 2 de notre projet Tetris. Celui-ci détaille tout ce qui a été fait pour cette remise ainsi que tout ce qui vous a été rendu. A ce stade, le jeu Tetris a été développé en interface graphique, en utilisant le Qt framework. Le jeu est fonctionnel, il a été testé sur plusieurs environnements différents, notamment Windows, OSX et Linux. Il permet de jouer une partie standard de Tetris, via l'option « Quick Start », c'est-à-dire une partie qui consiste à laisser l'utilisateur jouer la partie tant qu'il arrive à faire des lignes. Celle-ci s'arrêtera lorsque la prochaine figure ne peut plus rentrer dans le board, alors la partie est terminée.

Il est également possible de choisir plusieurs modes de jeu :

- Un score minimum à atteindre
- Un nombre de lignes à atteindre
- Un temps minimum que le joueur doit atteindre sans perdre la partie

Pour ces différents modes, il est laissé au joueur la possibilité de définir le score à atteindre, le nombre de lignes à atteindre ou le temps minimum.

Le jeu est constitué de niveaux, le joueur qui commence une partie a le niveau 0. Il n'y a pas réellement de niveau maximum, cependant la vitesse maximum est atteinte au niveau 14.

Suite à de nombreuses recherches sur le jeu Tetris, nous avons voulu respecter un maximum les principes de bases du jeu. Le board a une dimension de 10x20 et le nombre de figures possibles est à l'ordre de 7 (7 figures de 4 blocs chacunes). Ces figures pourront effectuer des rotations, aussi bien à gauche qu'à droite, tant que leurs positions dans le board le permettent. Nous plaçons la figure courante au centre du board avec le point de rotation à la 2<sup>e</sup> ligne en partant du haut, comme dans la plupart des jeux et notamment la version Tetris Gameboy de 1989.

La représentation de chacune des figures dans le board est basée sur la représentation de cette version de Tetris.

Nous avons passé beaucoup de temps à analyser les détails du jeu pour rendre celui-ci le plus jouable et le plus immersif possible.

Pour cela, beaucoup de détails du jeu original ont été implémentés. Comme par exemple, lors du fall de la figure, celle-ci ne disparait pas simplement pour réapparaître en bas du board mais une animation fait descendre case par case la figure jusqu'en bas, en vitesse accélérée.

## Présentation des fonctionnalités du jeu

Le jeu se joue en single player. Il est laissé la possibilité à l'utilisateur de :

- Déplacer la figure courante latéralement ainsi que vers le bas
- D'effectuer des rotations de la figure courante vers la gauche et vers la droite
- D'effectuer un drop de la figure, c'est-à-dire, celle-ci chutera à vitesse accélérée, l'utilisateur n'en aura plus le contrôle mais cela lui rapportera plus de points.

Le joueur peut, durant la partie, mettre le jeu en pause. Ensuite, il peut soit reprendre sa partie en cours, redémarrer la partie ou lancer une nouvelle partie dans un autre mode de jeu ou choisir le mode « quick start ».

## Les commandes du jeu

Le joueur déplace la figure courante avec les touches :

- Flèche bas
- Flèche droite
- Flèche gauche
- Touche W (rotation à gauche)
- Touche X (rotation à droite)
- Barre d'espace (drop)
- Touche P (pause)

## La partie graphique

La fenêtre de l'application se compose de deux parties distinctes :

### LA PARTIE DE GAUCHE

La partie de gauche contient le board, sur lequel le jeu se déroule.

### LA PARTIE DE DROITE

La partie de droite est composée de « deux parties » :

#### Le side board

Celui-ci affiche en temps réel, pendant la partie, le score du joueur, le niveau, le nombre de lignes effectuées, la prochaine figure qui arrivera sur le board, le temps de jeu écoulé (celui-ci s'arrêtant pendant les pauses) ainsi que l'objectif à atteindre lorsqu'on est dans un mode de jeu particulier.

#### Le choice board

Celui-ci comprend les différents boutons permettant de redémarrer le jeu, lancer une nouvelle partie ainsi que les options relatives aux modes de jeu particuliers.

## La partie son

Une des musiques du jeu Tetris original accompagne le joueur pour rendre l'expérience de jeu encore plus captivante. Chaque type de mouvement déclenche également des sons personnalisés.

Les mouvements de déplacement, les lignes effectuées, le début et la fin de partie, et également l'action qui consiste à effectuer un Tetris, c'est-à-dire, le fait d'éliminer 4 lignes simultanément (maximum de lignes possibles à faire en un seul coup). Et d'autres...

## Les ajouts par rapport à l'énoncé

### LA SHADOW

En temps réel, durant le jeu, l'utilisateur peut voir la « Shadow », qui consiste en une figure « transparente » qui représente la figure courante à la position final d'un drop à l'instant T.

### LA NON IMMOBILISATION DE LA FIGURE COURANTE DES QU'ELLE TOUCHE UN OBSTACLE

Dans certains Tetris, lorsque la figure courante touche, soit le sol, soit une figure par le bas, celle-ci s'immobilise définitivement avant de voir apparaître la prochaine figure. Toujours pour respecter un jeu Tetris classique, nous avons fait en sorte que le joueur puisse continuer à déplacer la figure courante un certain temps au moment où la figure courante se retrouve bloquée dans son déplacement sur l'axe vertical. Cela permet au joueur d'ajuster le déplacement de sa figure courante et notamment, de combler des trous qui seraient inaccessibles sans cela.

### LA FIGURE COURANTE DESCEND EN PERMANENCE

En effet, nous avons fait en sorte que la figure courante ne puisse être stabilisé indéfiniment sur l'axe horizontale, soit par des mouvements gauche-droite répétés rapidement par l'utilisateur, soit par des rotations successives, comme rencontrés sur beaucoup de jeu Tetris amateur.

## Les problèmes rencontrés / Les solutions implémentées

Un des plus gros soucis rencontrés lors de la programmation de ce jeu, fût l'utilisation du temps pour gérer les déplacements « automatiques » de la figure, qui au départ a une vitesse initiale de chute, celle-ci augmentant avec les niveaux ainsi que la vitesse qui doit être systématiquement modifié, lors des drops.

Nous nous sommes rendus compte que l'utilisation de threads aurait pu nous être très utiles pour régler certains de ces problèmes, mais n'ayant pas encore vu cette matière, nous avons réglé les différentes difficultés rencontrées, notamment par l'utilisation de plusieurs timers.

## Les écarts

Nous avons fait un maximum pour que notre programme soit complet mais surtout aussi fonctionnel que possible. Il ne nous a, dès lors, pas été possible d'implémenter autant de fonctionnalités que nous l'aurions voulu, privilégiant la robustesse et la fiabilité de notre application.

D'ailleurs, nous avons actuellement une version légèrement plus évolué que celle que nous vous remettons, mais qui n'a pas suffisamment été testée que pour vous la remettre à ce stade du projet.

Il nous était notamment demandé de laisser la possibilité à l'utilisateur de redimensionner la taille du board. Cela a été prévu dans le modèle conceptuel et ne pose en soit aucun problème, le fait de remplacer dans le code les valeurs de la taille du board (hauteur et largeur) n'affectera pas le jeu, néanmoins, nous n'avons pas optimisé l'affichage pour que le rendu soit optimal.

Il en est de même pour les figures personnalisées que nous n'avons pas encore implémentées, faute de temps.

## La compilation et les bugs

A ce stade-ci du projet, le programme compile sans erreur ni warning sur les différents environnements où nous l'avons testé.

Nous ne pouvons, bien sûr, pas garantir que notre application est sans faille. Ce que nous pouvons dire, c'est que nous avons passé beaucoup de temps à tester le jeu et à corriger, au fur et à mesure, certains bugs mineurs, et que jusque-là, nous n'avons rencontré aucune erreur d'exécution dans la version que l'on vous remet.

Si vous deviez rencontrer, lors de l'utilisation du programme, des comportements qui vous sembleraient anormaux, il est fort probable que cela soit simplement un choix des développeurs et un peu moins improbablement possible qu'en fait, nous ayons, par inadvertance, programmé une quelque fonction provoquant des résultats aléatoirement inattendus... ;)