



Tetris

REMISE₃

Version graphique et réseau

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

Table des matières

Introduction.....	2
Présentation des fonctionnalités du jeu.....	2
Les commandes du jeu (identiques à la version 2.0)	2
La partie graphique	3
La partie son.....	3
Les ajouts par rapport à l'énoncé	3
Les problèmes rencontrés.....	3
Les écarts.....	4
La compilation et les bugs	4
Bilan des bugs	4
Conclusion générale.....	5

Introduction

Ceci est le rapport de la remise 3 de notre projet Tetris, en réseau. Le jeu Tetris est développé en interface graphique et permet dorénavant, non seulement le jeu en single player, mais également le jeu en réseau. Il est possible de jouer à deux joueurs en utilisant un serveur qui permettra l'échange de données entre les deux clients.

Le jeu est fonctionnel et a été testé sur les différents environnements suivants : Windows, OSX et Linux. Il reprend toutes les options qui existaient déjà dans la version précédente à savoir, en interface graphique et mode solo. Ainsi que toutes ces mêmes fonctionnalités en partie multijoueur (2 joueurs uniquement).

Présentation des fonctionnalités du jeu

Il est dorénavant possible de jouer une partie en mode multiplayer, à savoir 2 joueurs. 2 joueurs peuvent s'affronter dans différents modes de jeu :

- Quick Game (partie rapide)

La fin de partie arrive uniquement dans le cas où un des joueurs ne peut plus placer la figure courante dans son board. Il est donc « game over ».

- New Game (atteindre un score, atteindre un nombre de lignes et survivre un temps défini)

En mode multijoueur, lorsqu'un joueur effectue un nombre K de lignes (4 maximum), il envoie K-1 lignes qui apparaissent à l'écran de son adversaire, permettant de pousser l'adversaire à la faute et ainsi provoquer sa défaite.

Les commandes du jeu (identiques à la version 2.0)

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)

Des commandes secrètes existent mais étant secrètes, ne seront pas expliquées ici...

La partie graphique

La partie graphique est relativement la même que dans la version précédente, cependant, l'ergonomie des boutons d'option a été revue pour faciliter la prise en main du jeu.

Il y a également un nouveau bouton permettant d'activer le serveur pour jouer en multijoueur.

(Ceci sera développé un peu plus bas).

La partie son

Les sons et musiques de la version précédentes sont conservés à l'identique.

Les ajouts par rapport à l'énoncé

Nous avons développé le jeu en réseau via 2 applications distinctes, à savoir le client et le serveur. L'interface client était en interface graphique tandis que le serveur, lui, ne fonctionnait qu'en mode console. Le projet était fonctionnel mais nous trouvions que la prise en main de l'application n'étant pas si évidente du fait de devoir lancer manuellement et indépendamment du jeu le serveur. Nous avons donc revu complètement l'infrastructure client/serveur pour développer une nouvelle version qui ne comporte plus qu'une seule application qui peut jouer le rôle du client et du serveur.

Un easter egg a été dissimulé par les développeurs dans l'application.

Les problèmes rencontrés

Il y a eu tellement de problèmes rencontrés qu'il serait difficile de tous les expliciter ici, en tout cas, peu d'entre eux n'ont pu être résolus.

Les principaux ont été :

1. La sérialisation de nos objets
2. Les différentes gestions de fins lors de fermeture inopinées d'un client ou du serveur
3. Etant donné le peu de connaissances théoriques sur les flux, les threads et la notion de client/serveur, gérer les discussions par le serveur a nécessité beaucoup de recherche et de temps

Les écarts

Nous pensons avoir respecté scrupuleusement l'énoncé du projet.

La compilation et les bugs

Le projet se compile sans erreur, cependant, de nombreux warnings subsistent. Ayant développé sur différents environnements, aussi bien différentes versions de l'IDE Qt que de versions de système d'exploitation, nous avons constaté que la quantité de warning varie fortement en fonction de ces différents paramètres. Dès lors, nous n'avons traité que ceux qui semblaient essentielles au bon déroulement de l'application et avons laissé les warnings qui n'affectaient en rien l'exécution de celle-ci.

Les warnings principaux concernent la non-portabilité des objets Qt, des switchs qui ne traitent pas le cas par défaut.

Nous avons conscience d'un bug qui n'arrive pas systématiquement et qui concerne l'envoi de messages entre clients. Lors d'une gestion de fin de partie multijoueur en jouant avec un certain gamemode, il arrive que lorsqu'un joueur gagne, que son adversaire ne reçoive pas le message indiquant la fin de partie. Mais un message de restart, sa partie recommence donc, hors qu'elle aurait dû s'arrêter. Il suffit dès lors que le joueur quitte cette partie s'il veut recommencer un autre type de partie.

Etant donné que le serveur est exécuté par un des clients, nous avons mis en place un système qui, dès lors que le client « héberge » la partie, affiche son adresse IP locale ainsi que le port pour qu'il informe son adversaire des paramètres à fournir pour pouvoir s'y connecter. Il arrive que l'adresse IP locale fournie n'apparaisse pas, qu'un fort ralentissement se fasse ressentir ou que l'adresse IP de la gateway apparaisse au lieu de l'adresse de la machine locale. Ceci n'est pas fréquent mais arrive si on s'amuse trop à couper/restart trop souvent le serveur.

Lorsque le serveur vient d'être activé et que le client veut s'y connecter trop rapidement, il se peut que le message « server unreachable » apparaisse. Ceci n'est pas vraiment un bug mais est plutôt logique étant donné que le serveur n'est pas encore en place pour recevoir les messages du client. Il suffit simplement de retenter une connexion.

Bilan des bugs

A priori, l'application ne devrait pas planter. Les quelques bugs résiduelles pourraient être résolus si nous y passions le temps. Mais notre quota d'heures passées sur ce projet a été très largement dépassé. (Sans nous déplaire, car nous avons beaucoup appris)

Il est également possible qu'un bug ait pu nous échapper. L'erreur est humaine.

Conclusion générale

Notre application a énormément évolué depuis la remise 1. Nous avons fait beaucoup d'erreurs mais nous avons également beaucoup appris et sommes très satisfaits du résultat obtenu à ce jour. Nous avons développé ce jeu Tetris en gardant comme ligne de conduite principale de développer un jeu fluide et agréable à jouer. Et malgré quelques défauts de l'application (après tout nous sommes en phase d'apprentissage), le résultat est un jeu fonctionnel avec lequel nous avons beaucoup de plaisir à jouer aussi bien en mode solo qu'à deux joueurs.

Vous nous aviez demandé d'être bref dans ce rapport, nous avons fait ce que nous avons pu. Il y aurait encore beaucoup à dire, nous répondrons à toutes vos questions le jour de la défense orale.

Merci de l'attention que vous porterez à notre projet.

Mike Sarton & Olivier Cordier.