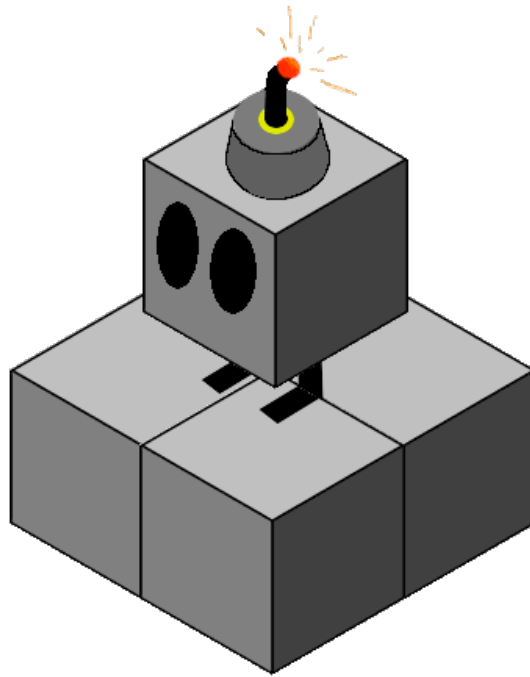

B65 – Projet synthèse

Document de conception



Titre : BomberBlock

Par **Simon Legros**
Maticule : **1379013**

Le 12 février 2018

Cégep du Vieux Montréal
225 rue Ontario, Montréal,
QC, H2X 1X6

Table des matières

Résumé du projet.....	3
Le projet en détails.....	4
Structure du projet.....	5
Côté Client.....	5
Côté Serveur.....	6
Pourquoi ce projet? Pourquoi Javascript?.....	6
Cas d'usage.....	7
Diagramme de classes.....	8
Conception des interfaces usagers (Mockup).....	9

Résumé du projet

Mon projet consiste en un jeu multijoueur sous forme d'application web, de style RPG/Survival avec une vue isométrique sur le monde. Les joueurs pourront se connecter à de nouvelles parties ou à des parties en cours et tenter de survivre contre les ennemis et ainsi remporter le plus de points possible. Tout cela est réalisable grâce à NodeJS, une plateforme logicielle libre orientée vers les applications réseau, permettant une communication synchronisé avec tous ses clients.

Le projet en détails

Dans le cadre du cours *B65 - Projet Synthèse* du DEC en informatique du Cégep du Vieux Montréal, j'ai la chance de pouvoir utiliser toutes les connaissances que j'ai acquises au cours des 3 dernières années, dans le développement d'un projet personnel où seul moi aura le pouvoir de le faire évoluer. Je crois que c'est un excellent moment pour montrer ce que je suis capable de faire et ainsi garnir mon portfolio d'un projet personnel complet et surtout complexe.

Étant moi-même consommateur de jeux vidéo, je comprend l'enthousiasme des gens pour ceux-ci et, bien sûr, pour la compétition. Dans un jeu vidéo, il y a un dynamisme et une forte interaction avec l'utilisateur qu'on ne retrouvera pas dans d'autres applications, tel un site d'achat en ligne. De plus, j'ai depuis longtemps un intérêt particulier pour la logique des jeux et leurs interfaces toujours plus modernes. J'ai donc décidé de développer un jeu pour mon projet synthèse de fin de DEC.

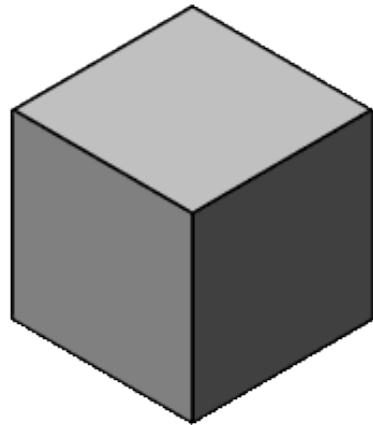
Le jeu, qui porte à ce jour le nom ***BomberBlock***, sera multijoueur en ligne, sous forme d'une application web et accessible directement sur mon site web. Pour l'instant, le type de partie et de combat est encore en négociation, mais voici les types qui y sont disputés :

- MMORPG (jeu de rôle massivement multijoueur en ligne) : Le joueur se connecte et fait évoluer son personnage dans un monde ouvert où il pourra accomplir des missions et gagner des récompenses, mais surtout survivre contre les multiples ennemis.
- *Survival* en équipe : Une équipe de (4) joueurs font alliance afin de survivre à une infinité de vagues de zombies affamés! Le but est de survivre le plus longtemps, mais aussi de s'entraider entre joueurs.

- Match à Mort en équipe : (2) équipes de (12) joueurs (maximum) s'affrontent pour une durée prédéterminée ou jusqu'à ce qu'une des deux équipes atteigne le score commun maximal, ex. 150 points. Les parties pourront être disputées dans quelques carte différentes.

Quelque soit le type de partie, les joueurs pourrons améliorer leur personnages et leur armements en échange de crédits gagnés en jouant. De plus, quelques bornes (magasins) d'armement seront placées dans chacune des cartes. Un joueur qui necessitera des soins ou un armement pourra donc s'en procurer à l'une de ces bornes.

Pour ce qu'il est de l'interface et de la vue du jeu, j'ai décidé de faire un 2D avec une vue isométrique, donnant ainsi une impression de 3D. C'est ce qu'on peut appeler de la fausse 3D.



Un cube isométrique

Structure du projet

Mon projet est principalement séparé en deux grandes parties : l'application du côté client qui sert d'interface pour le jeu, et le serveur multijoueur, côté serveur, qui s'occupe de connecter les joueurs à une partie et qui fait tout le calcul des déplacement et du dynamisme du jeu.

Côté Client

L'ensemble de l'application sera écrit en Javascript, avec l'aide de Phaser, un framework javascript orienté jeu web. Phaser permet un excellent support dans le développement d'un jeu en offrant le concept d'État (State) qu'un jeu doit avoir pour faire évoluer son utilisation. Ce framework est écrit en javascript et travail

sur un canvas, il est donc plus facile de l'utiliser avec d'autres parties de l'application.

Côté Serveur

L'ensemble du serveur sera aussi écrit en javascript car j'utiliserai NodeJS, une plateforme logicielle libre orientée vers les applications réseau. Il offre un synchronisme des requêtes et des réponses à l'ensemble de ses clients, permettant ainsi de faire du "Real-time multiplayer game". Le serveur aura donc la tâche d'écouter les événements émis par les clients, réagir en recalculant les positions, les orientations et les actions, et finalement retransmettre l'information à l'ensemble des clients.

Pourquoi ce projet? Pourquoi Javascript?

Il y a plusieurs raisons qui m'ont poussées à faire ce projet et à choisir Javascript plutôt que d'autres langages tout aussi performants ou même plus. Pour commencer, je voulais que mon projet soit facilement accessible, compatible et portable, surtout lors de sa présentation. Il était donc inévitable de penser à un projet Web et accessible rapidement de mon serveur chez Digital Ocean. Ensuite, je voulais avoir un jeu assez complexe et complet lors de la présentation. Je visait donc un langage facile à apprendre, mais aussi très performant pour résister à la demande de synchronisme de mon jeu. J'ai donc sans hésité pensé à Javascript. C'est un langage utilisé principalement pour le web, pour dynamiser des sites web, et pratiquement tous les jeux en ligne sont écrits avec ce langage. Javascript me permettra ainsi d'augmenter considérablement la vitesse de développement. Finalement, j'ai un énorme intérêt pour ce projet et je suis certain de prendre du temps pour le continuer après le DEC.

Cas d'usage

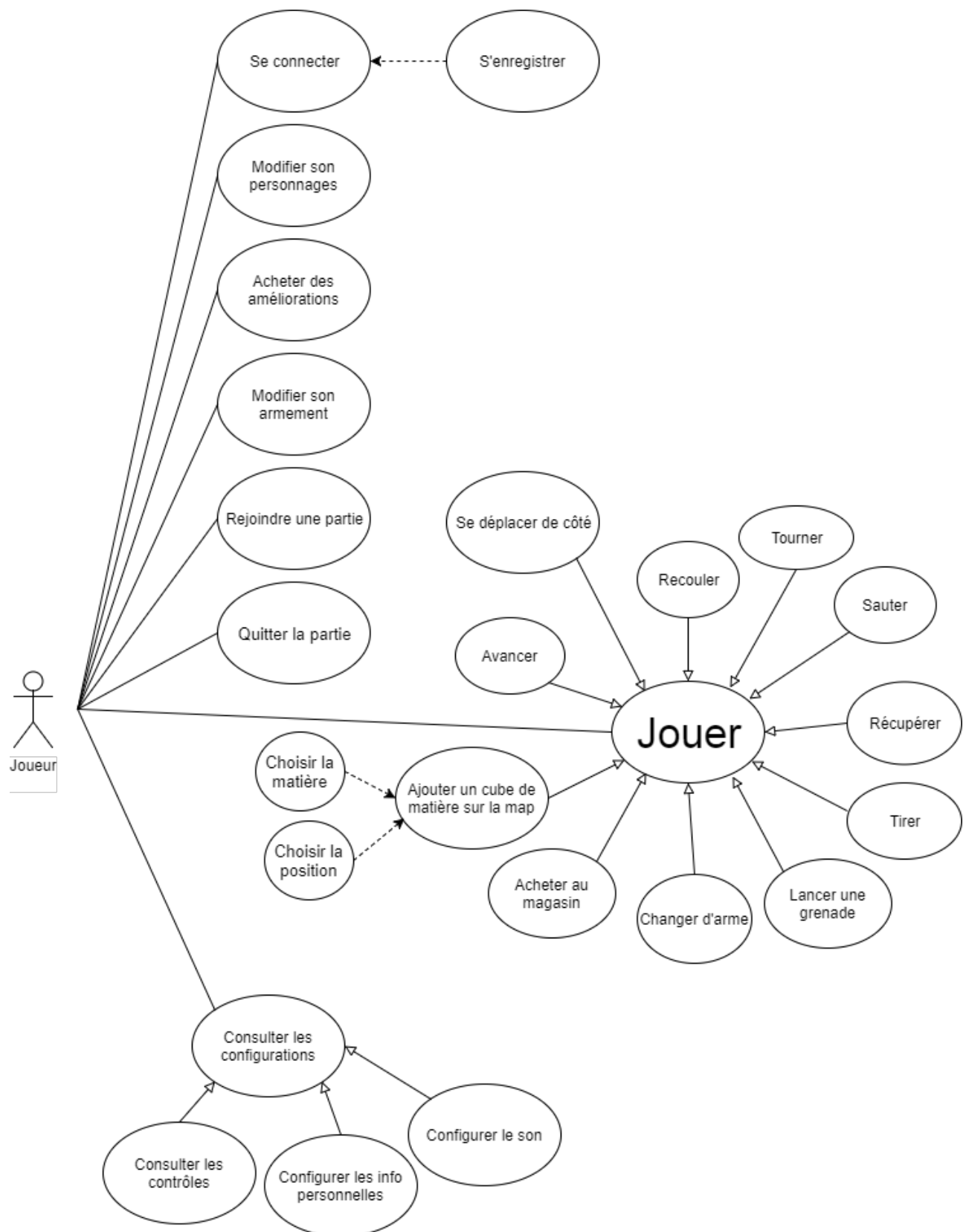
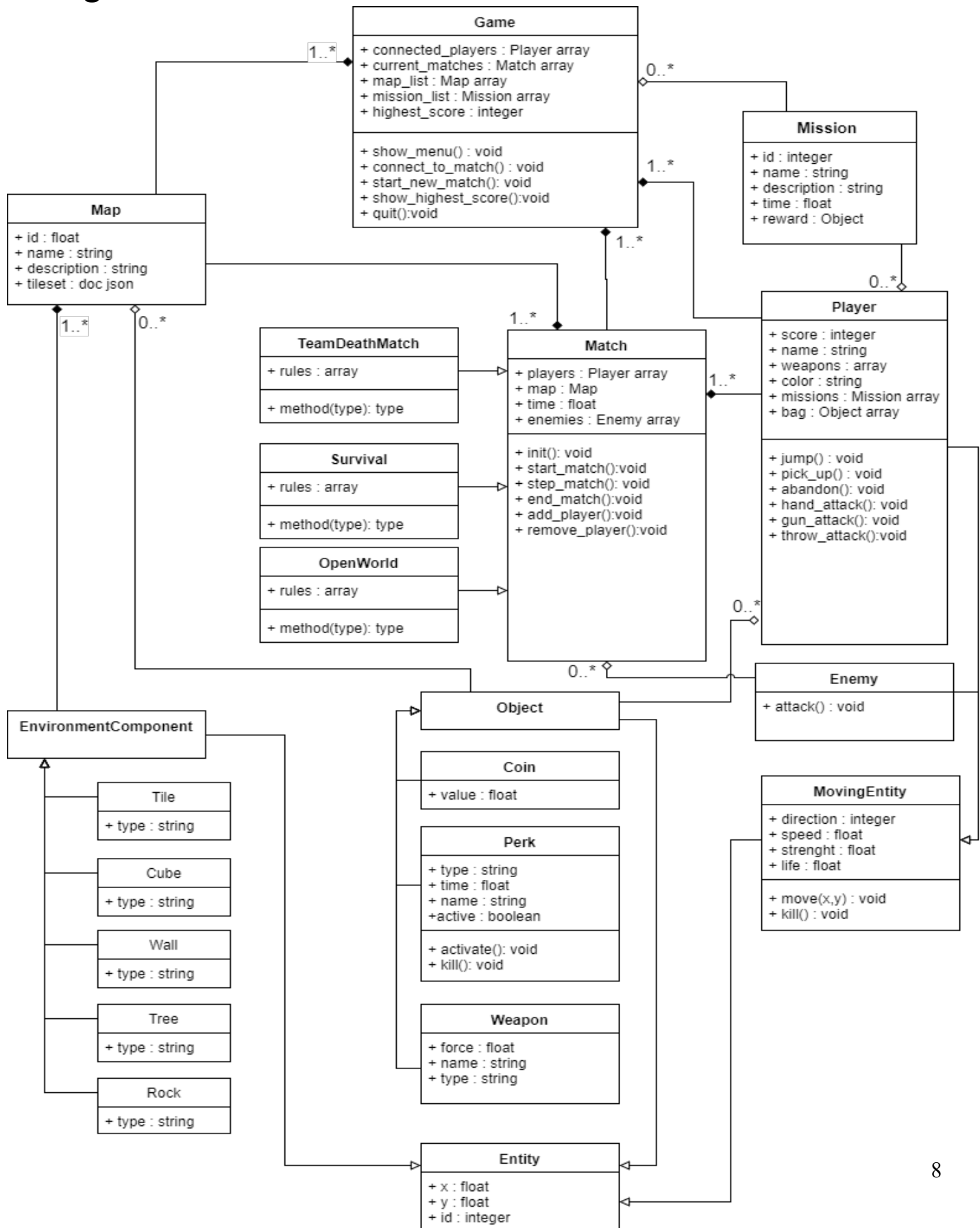


Diagramme de classes



Conception des interfaces usagers (Mockup)

