Projet LO43 : SmallUTBM

Salomé Welche & Romain Dulieu & Haocheng Xu & Romain Thibaud

Automne 2014

Table des matières

Chapitre 1

Introduction

C'est dans le cadre de nos études à l'UTBM (Université de Technologie de Belfort-Montbéliard) que nous avons été ammenés à dévelloper une adaptation du jeu de plateau Smallworld©. L'objectif du cours était d'introduire la programmation orientée objet grâce aux deux languages qui en sont les fers de lances, à savoir le C++ et le Java. C'est en Java que ce logiciel doit être programmé.

Dans un premier temps, nous nous devons de bien assimiller les règles et les subtilités du jeu afin de pouvoir les traduire en language informatique. De plus, une bonne compréhension de ces dernières est nécessaire pour pouvoir les adapter au mieux à l'univers qu'est l'UTBM. En effet, l'objectif est d'offrir une expérience pensé autour de ce qu'est la vie à l'UTBM.

Une fois cette première phase terminée, il est nécessaire d'effectuer une analyse technique et profonde du logiciel. Il est important d'effectuer cette expertise en amont pour faciliter l'implementation future. Cette étape repose sur analyse UML mais aussi une réflexion autour de l'organisation du programme et des outils utilisés.

Enfin, nous entamons la phase de production. L'ensemble des tâches à effectuer est hiérarchisé afin de rendre une application la plus aboutie possible dans le temps impartit.

Chapitre 2

Le jeu : Smallworld

2.1 Description globale

Smallworld© est jeu de plateau se jouant de 2 à 5 joueurs. C'est un jeu de stratégie militaire au tour par tour à la manière d'un Risk©. L'originalité du jeu repose sur le fait que le plateau semble trop petit pour tout les joueurs et les unités trop peu nombreuses pour anéantir nos adversaire. Toute la subtilité repose alors sur la gestion d'un peuple de son avénement à lextinction de ce dernier en passant bien entendu par son âge d'or. L'objectif est donc de gérer un peuple et de décider quand un est-ce qu'il n'est plus rentable. On décide alors de le mettre en déclin et ensuite de choisir un nouveau peuple pour jouer avec.

Cette mécanique de jeu, qui demande au joueur d'être pragmatique, est amplifié par une faible présence d'aléatoire dans les phases de combatts. Au contraire de Risk©, où il y a lancé de dé à chaque combat, le rapport de force s'effectue sur le nombre d'unité en présence, pour prendre un territoire adverse, il faut attaquer avec plus d'unité que la défense. Il y a cependant deux execptions : lors de la dernière bataille d'un joueur il peut jeter un dé de renfort ou en ca de pouvoir spécial (qui sera détaillé plus tard).

Bien que la stratégie de conquête soit au centre du gameplay, l'objectif n'est pas d'exterminer tous nos compagnons de jeu. La fin de la partie est déterminée par un nombre de tour par joueur. Le vainqueur étant celui qui a accumulé le plus de points de victoire. Ces mêmes points de victoire servant de monnaie durant la partie, il faut alors savoir évaluer la rentabilité de chaque coup.

De plus, la rejouabilité du jeu est assuré par l'unicité de chaque partie. En effet, tous les peuples n'étant accessibles à tous moments, le joueur doit

s'adapter à chaque fois. A cela s'ajoute des pouvoirs qui sont aussi tirés aléatoirements et assigné à un peuple. Cette multipplicité de combinaisons peuple-pouvoir entraine par contre une difficulté d'équilibrage qui se ressent de temps à autre, certaines alliances parraissantes plus fortes que d'autres.

2.2 Règles du jeu

Chapitre 3

Adaptation des règles

3.1 Ambiance recherchée

L'objectif étant de seulement d'adapter l'univers heroic-fantasy du Smallworld© à l'univers de l'UTBM, il n'y a pas de recherche sur le système de jeu à proprement dit. Sans ces notions de gamme design, nous nous sommes intéresser à la cosmétique du jeu pour installer notre ambiance.

Nous avons voulez jouer sur les petites rivalités qu'il peut exister, ou du mois qu'il semble exister entre les différentes instances de l'UTBM.

3.2 Les peuples

3.3 Les pouvoirs

Chapitre 4

Analyse des problèmes

Chapitre 5

Le modèle

5.1 Le diagramme de classe du modèle

5.2 Implémentation

Chapitre 6

L'interface

6.1 Description et organisation

6.2 Solutions techniques

Chapitre 7

Prévisionnel des tâches

7.1 Fonctionnement général

7.2 Interface

Chapitre 8

Conclusion