

Projet LO43 : SmallUTBM

Salomé WELCHE & Romain DULIEU & Haocheng XU & Romain THIBAUD

Automne 2014

Table des matières

1	Introduction	3
2	Le jeu : Smallworld	4
2.1	Description globale	4
2.2	Règles du jeu	5
3	Adaptation des règles	6
3.1	Ambiance recherchée	6
3.2	Les peuples	6
3.3	Les pouvoirs	7
4	Analyse des problèmes	8
4.1	Organisation globale du logiciel	8
4.1.1	Gestion du projet	8
4.1.2	Structure du logiciel	8
4.2	Le diagramme de cas d'utilisation	8
4.3	Les diagrammes de séquence	8
4.3.1	Conquête	8
4.3.2	Déclin	8
4.3.3	Nouveau peuple	8
4.3.4	Gestion des points	8
4.4	Les diagrammes état-transistion	8
5	Le modèle	9
5.1	Le diagramme de classe du modèle	9
5.2	Implémentation	9
6	L'interface	10
6.1	Description et organisation	10
6.2	Solutions techniques	10

7	Prévisionnel des tâches	11
7.1	Fonctionnement général	11
7.2	Interface	11
8	Conclusion	12

Chapitre 1

Introduction

C'est dans le cadre de nos études à l'UTBM (Université de Technologie de Belfort-Montbéliard) que nous avons été amenés à développer une adaptation du jeu de plateau Smallworld[®]. L'objectif du cours était d'introduire la programmation orientée objet grâce aux deux langages 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 assimiler les règles et les subtilités du jeu afin de pouvoir les traduire en langage 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 pensée 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'implémentation 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 imparti.

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 avènement à l'extinction 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 combats. 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 exceptions : 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 aléatoirement et assigné à un peuple. Cette multiplicité de combinaisons peuple-pouvoir entraîne par contre une difficulté d'équilibrage qui se ressent de temps à autre, certaines alliances paraissent 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. Nous nous sommes appliqué à caricaturer ces rapports entre les différents corps de l'utbm, en s'appuyant parfois sur des stéréotypes infondés mais quand même sympathique. Notre volonté n'est pas de créer de clivages entre nos camarades. Nous avons d'ailleurs fait preuve d'autodérision par rapport aux instances auxquelles on appartient.

On a donc juste changé les noms des différents peuples, pouvoirs, ou encore élément de jeu. De temps à autres de petites modifications ont été apportées sur les mécaniques de jeux mais c'est essentiellement pour simplifier le passage sur informatique.

3.2 Les peuples

Nous allons ici exposer les différents peuples en les liants à l'anciens peuple, la description du pouvoir et en expliquant pourquoi nous les avons choisis.

EDIM : (ex Amazones). Les EDIMs commence avec 4 pions supplémentaires, et tant qu'ils ont plus de 9 pions, ils perdent 2 pions par tour.

Le clin d'oeil est ici fait à l'ancien peuple, car EDIM est le département avec le plus gros pourcentage de filles.

TC : (ex Mi-portions). Les TCs ne peuvent rentrer que par le côté droit du plateau (côté Sévenans), il installe alors leur premier appartement sur les deux premières régions conquises. Ces régions sont imprennables et immunisées. Quand on arrive en tronc commun, c'est souvent du lycée que l'on vient, et donc c'est la première expérience de vie indépendante pour la majorité, d'où le choix des plus petits êtres.

Administratif : (ex Nains) Tous les bureaux occupés par du personnel administratif (en déclin ou non) rapporte 1 UV supplémentaire en fin de tour. Les nains ayant un rapport particulier à l'argent, on les a mis en relations avec ceux qui gèrent le budget de l'UTBM.

Enseignants chercheurs : (ex Elfes) Lorsqu'un adversaire s'empare d'une région, vous récupérez tous les pions de la région et les réorganisez sur les régions qu'il occupe encore à la fin du tour. On a joué sur la rivalité elfes et orcs pour illustrer celle entre enseignant chercheur et vacataire.

Vacataires : (ex Orcs) Toute région conquise par des vacataires rapporte 1 UV supplémentaire à la fin du tour.

IMSI : (ex Géant) Attaquez une région adjacente à un atelier coûte 1 unité de moins. Les IMSI étant les "pousse cartons", il fallait un peuple grand et fort pour les représenter.

Associatif : (ex sorcier) A chaque tour, l'étudiant investit dans l'associatif peut remplacer un pion actif (seul) d'un autre peuple, adjacent à une région contrôlée par cet étudiant, par un pion associatif. L'associatif peut-être une raison d'échec à l'utbm, c'est pourquoi il est en relation avec ceux qui pervertissent les autres peuples.

GMC : (ex homme-rat) Pas de capacité spécifique, ils sont déjà l'élite (référence caricaturale aux discours de Mr Gomez).

Alternants : (ex Humains) Tout cas monde de l'entreprise occupé par vos alternants rapporte une UV supplémentaire en fin de tour. Cela montre la proximité entre alternant et entreprise.

EE : (ex Mages) Toute région comportant une source d'énergie rapporte une UV supplémentaire à la fin du tour. Le clin d'oeil semble évident (indice : ils étudient l'énergie).

Doctorants :

3.3 Les pouvoirs

Chapitre 4

Analyse des problèmes

4.1 Organisation globale du logiciel

4.1.1 Gestion du projet

4.1.2 Structure du logiciel

4.2 Le diagramme de cas d'utilisation

4.3 Les diagrammes de séquence

4.3.1 Conquête

4.3.2 Déclin

4.3.3 Nouveau peuple

4.3.4 Gestion des points

4.4 Les diagrammes état-transistion

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