

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	8
4	Analyse des problèmes	10
4.1	Organisation globale du logiciel	10
4.1.1	Gestion du projet	10
4.1.2	Structure du logiciel	10
4.2	Le diagramme de cas d'utilisation	11
4.3	Les diagrammes de séquence	11
4.3.1	Conquête	11
4.3.2	Déclin	11
4.3.3	Nouveau peuple	11
4.3.4	Gestion des points	11
4.4	Les diagrammes état-transistion	11
5	Le modèle	12
5.1	Le diagramme de classe du modèle	12
5.2	Implémentation	12
6	L'interface	13
6.1	Description et organisation	13
6.2	Solutions techniques	13

7	Prévisionnel des tâches	15
7.1	Fonctionnement général	15
7.2	Interface	15
8	Conclusion	16

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 : (ex Squelettes) Lors du redéploiement, rajoutez une unité par groupe de deux régions conquises ce tour-ci. Trois ans de plus à être étudiants, il y a de quoi devenir rachitique.

Etudiants Etranger : (ex Tritons) Attaquez une région adjacente à une infrastructure de transport (gare, arrêt de bus,...) coûte 1 unité de moins. Ils remplacent les tritons car eux aussi viennent de contrées lointaines et inexplorées.

Info : (ex Trolls) Placez une chambre avec un ordinateur sur chaque région occupée par des infos. La défense du territoire est augmenté de une unité, la chambre est détruite qu'à la disparition des infos de la région (elle reste en cas de déclin). Essayez de chasser un geek de devant son PC pour comprendre.

Décalé : (ex Zombies) Lors du passage en déclin, toutes les unités restent sur le plateau et vous jouez normalement avec eux avant votre autre peuple. Les décalés sont ceux que les profs souhaitent virés mais qui s'accrochent.

3.3 Les pouvoirs

Nous procédons de-même pour les pouvoirs :

Proche des entreprises : Tant que le peuple est pas en déclin, vous gagnez deux UVs supplémentaires à la fin de votre tour.

Fêtard : Toutes les régions nécessitent 1 unité de moins pour être envahies.

Drogué au café : A chaque fin de tour, choisissez deux régions et placez-y un thermos de café sur chaque, ces régions sont imprenables et immunisées.

Bilingues : Une fois par tour, placez un dictionnaire bilingue. Elle augmente la défense de la région de 1 et rapporte une UV supplémentaire à la fin de chaque tour (sauf si vous êtes en déclin). Elles disparaissent si vous quittez la région.

Grande Gueule : Vous pouvez jeter le dé de renfort avant de choisir une région à envahir (à chaque fois).

Qui sèche : Vous pouvez poursuivre votre expansion et passer en déclin juste après.

Connecté : Toutes les régions qui compte une prise RJ45 sont considérés comme adjacente pour vous. Ces régions nécessitent 1 unité de moins pour être envahie.

Et leur prof suiveur : Vous pouvez envahir une région avec une seule unité, une seule fois par tour. Placez alors le prof suiveur sur cette région, elle est imprenable et immunisée jusqu'à ce que le prof suiveur aille faire du soutien dans une autre région.

- Des amphis :** Prenez une UV supplémentaire pour chaque région avec un amphithéâtre à chaque fin de votre tour.
- Des salles infos :** Prenez une UV supplémentaire pour chaque région avec une salle infos à chaque fin de votre tour.
- Des lieux de vie :** Prenez une UV supplémentaire pour chaque région avec un lieu de vie à chaque fin de votre tour.
- Fayots :** Choisissez un adversaire que vous n'avez pas attaqué ce tour-ci, il ne pourra pas vous attaqué pendant son tour.
- Avec équivalences :** A la fin de votre premier tour, prenez 7 UV supplémentaires.
- En double filière :** Prenez une UV supplémentaire pour chaque région que vous occupez en fin de tour.
- Du club Welcome :** Vous pouvez envahir les infrastructures de transports.
- Ponctuel :** Les régions amphithéâtre et entreprise nécessitent 1 unité de moins pour être envahie.
- Du père 200 :** Toute région non vide conquise rapporte une UV supplémentaire à la fin du tour.
- Revenant d'Erasmus :** Vous pouvez conquérir n'importe quelle région, même une région non adajacente.

Chapitre 4

Analyse des problèmes

4.1 Organisation globale du logiciel

4.1.1 Gestion du projet

Tout d’abord, parmi les étudiants travaillant sur le projet, il y a un étudiant chinois. Pour faciliter la communication entre nous, nous avons donc décidé de travailler en anglais. C’est pourquoi, toutes les productions liées à ce projet sont en anglais, mis à part les rapports (ça lui permet aussi de travailler son français).

Pour faciliter le travail en groupe nous avons aussi mis en place un gestionnaire de version, à savoir Git. Hébergé sur GitHub[®], cela permet de travailler en simultané sur le projet sans se marcher dessus, et d’avoir des sauvegardes des différentes phases de notre projet en cas de bug.

D’un point de vue logiciel, nous avons utilisé ArgoUML comme AGL (pour la création des diagrammes UML) et Eclipse comme IDE pour la programmation en java. Étant donné que nous avons effectué les diagrammes UML en amont, nous avons pu générer la structure du code via ArgoUML, et nous n’avons plus qu’à remplir les fonctions.

Concernant la répartition des tâches, Romain T. s’occupe de la partie interface pendant que Salomé, Haocheng et Romain D. s’occupe de la conception du programme. Ils se sont occupés de construire les diagrammes UML auparavant. Ce rapport étant, lui, un produit collégial.

4.1.2 Structure du logiciel

Nous avons adopté pour ce projet un patron modèle-vue-contrôleur. La force de ce paradigme est d’être très modulable. En effet, il fonctionne en sé-

parant l'interface du corps du programme et les fait dialoguer via un contrôleur. L'avantage est que si on change le type de la base de données par exemple, cela n'auras que très peu d'impact sur l'interface.

De plus, il est plus simple pour nous de travailler dessus, le code étant plus aérer et bien segmenté, nous pouvons plus facilement nous répartir le travail, et donc être plus efficaces.

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

Nous avons décidé de partitionné notre interface en 4 panneaux :

- Un pour le plateau de jeu
- Un pour les actions du joueur
- Un pour la pioche
- Un pour rappeler l'ordre des actions

Le plateau sera une image de fond avec des éléments clicables dessus où apparaitront les peuples. Les actions joueurs et la pioches ne seront que des éléments clicables, alors que le panneau d'information ne sera qu'une succession d'éléments textuels à caractères purement informatif.

6.2 Solutions techniques

Nous avons commencer une implementation de l'interface en utilisant la bibliothèque graphique Swing. Mais cette bibliothèque ne nous convient pas assez, elle est lourde à mettre en place pour avoir des éléments dynamique, et elle n'est maintenu que pour des correctifs de bug, et donc ne propose plus d'avancées technologiques.

Pour ces raisons, nous sommes entrain d'étudier une migration vers JavaFX. De plus la communauté autour de cette bibliothèque semble très active sur le net, et donc offre un soutien et des tutoriels. Elle offre aussi la possibilité de travaillé avec des formats d'image en plus que ceux disponible sur swing (es : svg) permettant de travaillé avec des logiciels orienté vers l'image tel que Photoshop[®], Illustrator[®], ou encore Gimp[®].

Elle a aussi l'avantage d'être très portable (Linux, Mac, Windows, ou encore Android). Cela permet d'éventuellement penser à un portage multi-plateforme.

Chapitre 7

Prévisionnel des tâches

7.1 Fonctionnement général

7.2 Interface

Il y a plusieurs tâches à effectuer :

Une création et initialisation des panneaux

Création des éléments clicables

Rafraichissement de l'interface

Gestion des cliquer-glisser

Habillage de l'interface

Création des cartes pour 2,4, et 5 joueurs

Optimisation de la détection des régions

Création d'animations

Chapitre 8

Conclusion

Ce projet est plutôt plaisant pour nous, vu notre inclination pour les jeux de plateau tel Smallworld[®] (même si Seasons est mieux quand même). Il ne fut pas trop dur de nous impliquer dans l'analyse du jeu et son adaptation, ce fut même agréable et marrant.

Au delà de l'aspect ludique du projet, nous avons pu découvrir ce qu'est un travail en équipe, car nous sommes quatre et que c'est la première fois que nous sommes autant dans un groupe. De plus, notre groupe est multiculturel. Nous avons donc du trouver des solutions afin d'être les plus efficaces possibles.

Nous avons aussi mis en place des technologies que nous ne connaissions pas encore ou pas bien, ce qui est toujours enrichissant. Cependant, le temps d'adaptation nous a par contre parfois handicapé, et souvent ralenti.

L'objectif était de finir ce rapport à 19h le vendredi 19 décembre et cela tombe bien, nous avons notre soirée jeu de plateau en suivant.