

# Projet LO43 : SmallUTBM

Salomé WELCHE & Romain DULIEU & Haocheng XU & Romain THIBAUD

Automne 2014

# Table des matières

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

<b>7</b>	<b>Prévisionnel des tâches</b>	<b>9</b>
7.1	Fonctionnement général . . . . .	9
7.2	Interface . . . . .	9
<b>8</b>	<b>Conclusion</b>	<b>10</b>

# 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<sup>®</sup>. 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<sup>®</sup> 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<sup>®</sup>. 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<sup>®</sup>, 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.

## 2.2 Règles du jeu

## Chapitre 3

# Adaptation des règles

3.1 Ambiance recherchée

3.2 Les peuples

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