

SmallWorldUtbm

rapport de projet

Amani Younes - Michael Longo - Gabriel Notong - Pierre Rognon

Université de Technologies de Belfort-Montbéliard

automne 2012

Table des matières

1	Cahier des charges	4
1.1	Étude de l'existant et adaptation	4
1.2	Conception	4
1.3	Application	4
1.4	Contraintes et choix techniques	7

Introduction

Dans le cadre de l'Unité de Valeur LO43, il a été demandé aux étudiants de réaliser un projet de fin de semestre. Ce projet a pour but de mettre en application les différentes notions acquises en programmation orientée objet ainsi que de pratiquer un langage de programmation : le Java. L'objectif de ce projet est de créer un jeu de rôle à partir d'un jeu de société existant.

SmallWorld est un jeu de stratégie se rapportant à l'univers fantastique. A partir d'un peuple légendaire choisi, il s'agit d'effectuer des conquêtes sur un plateau de jeu comportant des territoires. L'occupation de ces territoires permet alors de remporter des points de victoire à chaque tour pour ensuite gagner la partie.

A l'occasion de ce projet, le jeu a été renommé SmallWorldUtbm et il a été demandé d'appliquer les grands principes du jeu d'origine tout en l'adaptant à la vie Utbohémienne. Cette adaptation a donc demandé une organisation en plusieurs étapes qui seront présentées dans une première partie. Sera ensuite abordée la mise en œuvre de ce projet comprenant des détails sur la conception et le développement de l'application. Enfin, les problèmes rencontrés et un bilan succinct seront dressés.

1 Cahier des charges

Le cahier des charges impliquait plusieurs parties. En effet, le sujet portant sur un jeu existant, une première étape consiste à découvrir les règles du jeu et les entités liées à celui-ci puis à les adapter au projet. Suite à cela, une seconde étape porte sur la réalisation d'une modélisation dans le langage UML. Le cahier des charges concernant l'application concerne la troisième partie du travail.

1.1 Étude de l'existant et adaptation

L'étude de l'existant doit permettre de réaliser plusieurs choses. Tout d'abord, elle doit permettre de comprendre les différentes règles du jeu. Elle doit ensuite aider à dissocier les différentes entités relatives à celui-ci. Cela doit avancer une première fois la réflexion sur la réalisation de l'application.

L'adaptation au sujet proposé est en lien direct avec l'existant. Elle doit en effet reprendre les grandes idées de l'existant tout en calquant les entités de l'UTBM. Cette adaptation doit être en réalité une sorte de première étape de conception puisqu'elle met déjà en jeu les différentes entités qui seront rassemblées lors de la conception proprement dite.

1.2 Conception

Pour la conception, le cahier des charges est en grande partie donné et la marge de manœuvre est faible. On doit donc réaliser :

- des diagrammes de cas d'utilisation ;
- des diagrammes de séquence ;
- un diagramme de classes.

Cette étape de conception s'appuie grandement sur l'adaptation faite lors de l'étude de l'existant. Les entités doivent en effet déjà avoir été isolées pour réaliser l'adaptation à notre contexte. L'étape de conception doit donc simplement transférer ces idées en respectant la syntaxe et les normes UML. Le diagramme de classes doit aussi permettre de développer plus facilement par la suite les classes Java en étant donc assez précis.

Suite à cela est réalisé le cahier des charges dit fonctionnel de l'application. Il permettra le développement de celle-ci.

1.3 Application

Le cahier des charges de l'application indique ce que devra faire celle-ci en développant son comportement. Un certain nombre de choses à faire sont donc déjà énoncées par le sujet ou par l'existant mais d'autres sont à indiquer. Ces besoins sont énumérés ci-dessous en quatre parties : tout d'abord ceux relatifs à un début de partie, ensuite d'autres en liens avec le cours de la partie, puis les besoins intervenant en fin de partie. Une dernière partie traitera des besoins liés à l'interface.

Début de partie

L'application doit répondre à un certain nombre de besoins en début de partie.

Tout d'abord, un certain nombre de règles générales ont été établies :

- l'application doit permettre d'effectuer des parties avec un nombre de joueur variables (de 2 à 5 joueurs) ;

- différents plateaux de jeu doivent être disponibles en fonction du nombre de joueurs dans une partie ;
- on doit pouvoir choisir le nom des joueurs ;
- l'application doit permettre au joueur de choisir un couple Peuple/Pouvoir ;

Pour cette application, un grand nombre de traitements se font de façon transparente par rapport au joueur. Ceux-ci sont traduits par les besoins suivants :

- l'application doit initialiser tous les peuples disponibles ainsi que les pouvoirs ;
- elle doit générer des couples aléatoires peuple/pouvoir afin que le joueur en choisisse une ;
- les territoires doivent être générés ainsi que les éléments potentiellement présents sur chacun d'eux ;
- l'application doit gérer l'argent liée à chaque couple peuple/pouvoir puisqu'un joueur peut payer pour prendre un couple.

Cours de partie

Du côté de l'utilisateur, on a les besoins suivants :

- Pour la phase de conquête de territoires :
 - il doit pouvoir tant qu'il n'a encore rien pris durant le tour cliquer sur un bouton de passage en déclin du peuple. Le joueur finit alors son tour directement ;
 - si le joueur a passé son peuple en déclin au tour précédent, l'application doit lui proposer de choisir un nouveau peuple lorsqu'il joue à nouveau ;
 - il doit pouvoir cliquer sur les territoires qu'il peut prendre (géographiquement) ;
 - l'application dans dans ces deux cas demander confirmation au joueur ;
 - l'application doit proposer lorsque le joueur n'a pas assez de pions pour attaquer un territoire de lancer un dé : si il accepte, on doit alors passer à la phase suivante.
 - le joueur doit pouvoir cliquer sur un bouton de fin de tour pour passer à la phase suivante.
- Pour la phase de redéploiement :
 - le joueur doit pouvoir cliquer sur tous ses territoires ;
 - l'application doit alors proposer le nombre de pions à placer sur celui-ci ;
 - le joueur doit pouvoir cliquer sur un bouton de fin de redéploiement lorsqu'il a fini ;
 - l'application ne doit pas proposer de bouton de fin tant que le joueur a des pions dans les mains.
- Pour la phase de redéploiement des autres joueurs, les besoins sont les même que pour celle du joueur qui vient d'effectuer son tour.

En arrière-plan, l'application doit :

- ne permettre au joueur que de cliquer sur les territoires auxquels il peut accéder ;
- calculer à chaque changement le nombre d'unités en main ;
- effectuer les changements d'occupant pour les territoires abandonnés ou conquis ;
- vérifier dans le cas où des joueurs ont des peuples en déclin que ceux-ci ont toujours des territoires. S'ils n'en n'ont plus, l'application doit supprimer le peuple en déclin du joueur et le remettre dans la liste des peuples libres ;
- en fin de tour, calculer les points de victoire gagnés par le joueur. Ceux-ci sont calculés à partir des territoires occupés par son peuple mais aussi par ceux occupés par son peuple en déclin et en fonction de ses éventuels bonus.

Fin de partie

Pour l'utilisateur, l'application doit simplement indiquer quel est le gagnant de la partie. Le joueur doit pouvoir ensuite soit rejouer une partie, soit quitter le jeu.

En arrière-plan, pour indiquer quel est le gagnant, l'application doit calculer quel joueur a le plus de points de victoire accumulés.

Apparence et mise en page générale

La réalisation de cette application doit intégrer une interface graphique. Un certain nombre de règles sont donc à énoncer :

- l'interface doit fournir les informations suivantes au joueur lors de son tour dans un cadre d'informations sur le joueur dans un coin de la fenêtre de jeu. Les informations suivantes seront mises en évidence dans un liseret de la couleur décernée au joueur :
 - le nom du joueur ;
 - son peuple actif ainsi que le pouvoir associé ;D'autres informations complétant les premières seront indiquées dans le cadre hors du liseret :
 - le nombre de points de victoire en cours ;
 - le nombre d'unités totales qu'il possède ;
 - le nombre de territoires occupés à l'instant où il joue ;
 - le nombre d'unités qu'il a en main à l'instant où il joue ;
 - l'éventuel peuple en déclin qu'il possède.
- elle doit aussi comprendre des informations sur les territoires mises à jour en fonction du territoire pointé par la souris. Ces informations, contenues dans un cadre d'information du territoire dans un coin de la fenêtre, sont :
 - l'occupant du territoire ;
 - le nombre d'unités présentes sur celui-ci ;
 - le coût de l'attaque pour le joueur en train de jouer ;
 - les éventuels éléments que contient le territoire.Le titre du cadre doit avoir pour fond la couleur décernée au joueur.
- les territoires doivent de plus indiquer sur leur espace le nombre d'unités contenues sur celui-ci. Ce chiffre doit être accompagné d'une couleur en arrière-plan correspondant à la couleur décernée au joueur. Si le peuple occupant est en déclin, la couleur sera adoucie ;
- un dernier cadre dont le titre a aussi pour fond la couleur du joueur doit contenir les boutons suivants :
 - fin du tour ;
 - passage en déclin du peuple ;
 - fin du redéploiement.Ces boutons doivent être disponibles en fonction de l'étape de jeu du joueur.

- un cadre temporaire doit apparaître pour expliquer au joueur les différents choix impossibles lorsque celui-ci les tente. Ce cadre doit être bien visible et aura une couleur de police rouge. Il doit servir par exemple lorsque l'on clique sur un territoire impossible à attaquer à l'indiquer au joueur.

Des informations entre les étapes et les tours doivent aussi apparaître pour situer le joueur. Ces informations sont :

- une page temporaire indiquant l'étape de conquête du joueur en cours avant que celui-ci attaque des régions ;
- une page temporaire affichant le redéploiement du joueur en cours ;
- une page temporaire affichant le redéploiement des éventuels joueurs qui ont des unités en main suite à des pertes de territoires ;
- enfin en fin de tour, une page temporaire indiquant le nombre de points de victoire durement acquis lors du tour du joueur.

1.4 Contraintes et choix techniques

Le cahier des charges comprend différentes contraintes qui ont été imposées par le propos du sujet. D'autres contraintes ont été rajoutées par l'équipe. Ces différentes contraintes sont de différents types.

De type temporel :

- le rapport du projet doit être rendu pour le 4 janvier et le projet doit donc être terminé pour cette même date ;
- les emplois du temps des différents membres du projet étant différents, des créneaux horaires en commun doivent être trouvés pour les réunions de projet.

De type technique :

- le langage de programmation est imposé : le Java ;
- de plus, la bibliothèque graphique présentée en cours est Swing. C'est donc cette dernière qui doit être utilisée ;
- afin d'éviter les problèmes de compatibilité, un environnement de développement est imposé : Eclipse. Il a l'avantage d'être largement utilisé et d'être disponible sur les systèmes d'exploitations les plus utilisés (Linux, Windows, MacOS) ;
- afin de pouvoir travailler chacun chez soi entre les différentes réunions, un gestionnaire de version est mis en place : Git. Ce dernier a été choisi car lié au site Github.com, il a l'avantage d'être simple à utiliser et de proposer des informations pratiques sur le développement. Le choix se justifie aussi du fait qu'un plugin liant Git à Eclipse est disponible. Ce dernier permet d'utiliser les deux en lien étroit et facilement ;
- pour regrouper les éléments et les territoires du jeu, une base de données doit être utilisée : il a été choisi SQLite pour représenter cette base de données. SQLite présente l'intérêt de ne pas nécessiter de serveur et d'avoir une interface graphique pour créer la base de données. C'est aussi un système léger puisque ne générant qu'un fichier pour une base de données, il est portable et très répandu.

SmallWorldUtbm est l'adaptation du jeu de société SmallWorld au contexte de l'Université de Technologies de Belfort-Montbéliard. Les règles du jeu doivent donc rester proches de ce jeu sans pour autant en faire la copie conforme. Le but reste donc le même c'est-à-dire diriger la destinée de peuples dans un monde où chacun doit donc lutter pour sa survie. De peuples de substitution : Étudiants TC, Étudiants branche, Professeur de connaissances scientifiques, Professeur d'humanités, Chercheurs, le CRI, les Thésards les Rats etc.. Sont un exemple ceux que les différents joueurs devront choisir pendant les différentes parties qu'ils disputeront. De pouvoirs associés aux peuples qui les accompagneront dans leurs différentes conquêtes de nouveaux territoires : Etudiants TC : nombreux, pas de pouvoirs ; Etudiants branche : quand ils perdent un territoire, on ne défais pas de pion ; CRI : si sur un territoire qui est une salle d'ordinateurs, a un bonus d'attaque +1 pion par territoire salle ordinateur ; Professeurs scientifique : sur chaque lancement de dé, on ajoute 2 au résultat ; Professeurs d'humanités : si sur un

territoire de partiel, ce territoire devient inattaquable ; Thésard : il prend un pion de plus s'il a conquis deux régions non vides ou plus ; Rats : ils annulent le bonus de défense de la nourriture et gagnent un pion de peuple par tour. D'éléments apportant des bonus de défenses une fois posés sur un territoire : Machine à café : bonus de défense +1 pion ; Photocopieuse : +1 pion de victoire pour le possesseur du territoire ; Nourriture : bonus de défense +1 pion, sauf contre les rats ; Salle de partiel : région inattaquable par un étudiant ; Espace plein air ; Salle informatisée. De pouvoirs spéciaux qui permettrons aux peuples qui les ont d'avoir une petite avance sur les autres : Geek : bonus de défense +1 pion si sur une salle informatisée, bonus d'attaque sur une région salle informatisée +2 pions ; Intello : bonus de défense +1 pion sur une salle de partiel, bonus d'attaque sur les salles de partiel +2 pions ; Fêtard : à chaque tour, 2 pions de peuple de plus à l'attaque ; Paresseux : bonus de défense partout +1 pion ; Avare : +1 pion de victoire par région conquise dans le tour ; Faux-cul : peut toujours attaquer par les bords de la carte en plus des régions adjacents ; Opportuniste : a le droit à un lancer de dé pour chaque conquête ; Joueur : +2 pions sur le dé à chaque lancer ; Voyageur : peut attaquer partout sur la carte ; Nerveux : si sur une machine à café, bonus de défense +2 ; Fumeur : sur un espace plein air, +2 points de victoire à chaque fin de tour ; Associatifs : +1 pion de peuple à chaque tour ; Bagarreux : +1 pion de peuple à l'attaque ; Gloutons : pose de la nourriture à chaque endroit où ils vont. Le jeu devra être multi-joueurs : au minimum deux joueurs. L'association de peuples et pouvoirs ainsi que les pouvoirs spéciaux se fera de façon aléatoire dès le lancement du jeu.