Université de Technologie de Belfort Montbéliard

LO43

Smallworld UTBM

Florent Jacquet
Antonin Waltz
Superviseur : Amine Ahmed Benyahia

Automne 2014

Table des matières

Table des figures

Chapitre 1

Présentation du projet

Durant ce semestre en LO43, nous avons choisi le projet *Smallworld* parmi tout ceux proposés.

Smallworld est un jeu de stratégie qui propose de gérer la destinée de peuples fantastiques, chacun disposant d'une particularité unique. En plus de ça, ceux-ci sont associés aléatoirement à un pouvoir spécial qui leur confére une capacité supplémentaire.

Le but est de conquérir des territoires afin d'obtenir le plus de points de victoire. Cependant au fur et à mesure de l'avancé du jeu, un peuple a tendance à s'affaiblir. Le joueur qui le contrôle peut alors le faire passer en déclin et choisir une nouvelle civilisation.

Nous avons adapter les peuples et les pouvoirs à l'environnement de l'école, en respectant les régles de base du jeu.

Chapitre 2

Organisation et répartition du travail

Notre groupe étant composé de 2 personnes, il était facile pour nous de communiquer sur l'avancé du projet, la façon d'aborder telle où telle contrainte technique où fonctionnelle ou tout simplement faire le point.

Dans un premier temps et après nous être approprié *Smallworld*, nous avons réaliser les spécifications UML ensemble afin de partir sur la même base de départ. Evidemment nous adapterons au fur et à mesure de l'implémentation en fonction de l'évolution des contraintes. Nous nous sommes également mis d'accord sur l'utilisation du pattern MVC, du polymorphisme, etc.

Concernant les outils, nous avons utilisé **Eclipse**, **Dia** pour l'UML et un dépôt **Git** pour la synchronisation des sources et le versionnage. Nous avons également profiter des fonctionnalités de **Git** pour effectuer un suivi efficace des fonctions à modifier où terminer.

La répartition du travail s'est globalement fait ainsi, sachant que chacun n'hésitait pas à intervenir dans les parties de l'autre si besoin était et qu'elle n'est pas définitive :

- Florent : Modèle, Contrôleur, Rapport
- Antonin : Vue, Rapport

Chapitre 3

Architecture du projet

3.1 Première approche (Cas d'utilisation)

Lorsque l'utilisateur lance le jeu, un premier menu s'offre à lui. Il peut choisir entre lancer une nouvelle partie, charger une partie déjà existante, afficher les règles où quitter le jeu.

Le seul cas qu'il est intéressant de détailler est celui d'une nouvelle partie. Après que le nombre de joueur ait été défini, les utilisateurs jouent chacun à tour de rôle. Ils ont alors le choix entre 4 actions :

- Acheter un peuple
- Passer en déclin
- Attaquer un terrain
- Se redéployer

L'ordre et les différentes interactions entre ces actions sont détaillés dans le diagramme de séquence.

3.2 Le modèle

C'est là le cœur du programme. Le modèle est une classe qui regroupe toutes les classes du jeu en lui même, c'est à dire la carte, les jetons, les joueurs où toute autre classe utile à la simulation du jeu par le programme. En revanche, il ne s'agit nullement de l'affichage, puisque cette tâche est laissée à la partie Interface Utilisateur, la Vue.

Cette partie est la seule pour laquelle nous avons réalisé un diagramme de classe, étant donné qu'elle représente une grosse partie du projet, et que beaucoup de classes entrent en jeu. Afin de mieux le structurer et de savoir toujours dans quelle direction partir, le diagramme UML se trouvant en Annexe 3 a été réalisé. Cela a permis d'avoiruimmédiatement un aperçu de l'implémentation globale, et de se mettre d'accord sur la conception du projet.

3.3 L'interface (la Vue)

Lors d'un tour de jeu, le joueur doit pouvoir effectuer les actions qu'il désire. En même temps il est nécessaire qu'il ait accès à toutes les informations utiles.

L'interface se divise en 4 parties distinctes :

- En haut, un récapitulatif des informations de base des joueurs adversaires : pseudo, peuple et pouvoir actif, icône d'avatar etc.
- À droite, la liste des combinaisons peuple-pouvoir avec le nombre de pièces d'or correspondantes qu'il est possible de choisir.
- Au centre, le plateau de jeu avec les territoires, le peuple par territoire, les attributs spéciaux etc.
- En bas, les informations relatives au joueur actif, ainsi que les boutons d'action

Elle est cependant susceptible d'évoluer lors de l'implémentation.

3.4 Le déroulement d'une phase de jeu (Séquence)

Cette partie est très importante, puisqu'il s'agit en somme des règles du jeu. Étant donné que le programme est évènementiel, il suffit de savoir quelle action peut être faite à quel moment, et cela se résume dans le diagramme de séquence que nous avons fait, qui se trouve en Annexe 2.

Ce diagramme s'étend sur la séquence de jeu d'un joueur, ce qui ne représente pas la séquence complète du programme, mais simplement la partie essentielle de celle-ci. On omet donc les étapes de fin de tour, ainsi que celles de l'initialisation et de la fin du jeu, qui ne sont pas ëssentiellesäu déroulement d'une partie.

Chapitre 4

Réalisation

4.1 Menu d'accueil

Au lancement du jeu, l'utilisateur arrive sur la page du menu d'accueil. Il peut alors ajouter où retirer des joueurs, lancer une nouvelle partie, parcourir les règles du jeu, charger une partie déjà existante où encore quitter. Un bouton retour est également disponible afin de pouvoir retourner facilement sur l'accueil suivant la page affichée.

4.2 Durant une partie

Lorsque l'on est en jeu, toutes les actions prévues dans les règles sont réalisables. On peut sélectionner un peuple, conquérir des territoires occupés où non, passer au joueur suivant et enfin passer en déclin. On dispose également des informations courantes, comme le nombre de jetons possédés ou le peuple actif. On peut également à tout moment sauvegarder sa partie, quitter le jeu et la recharger via le menu principal. Toute la fenêtre de jeu est également redimensionnable en permanence, les éléments affichés s'adaptant en conséquence.

4.3 Ce qui manque

Tout le jeu est fonctionnel, cependant nous n'avons pas implémenter les jetons spéciaux et les règles des pouvoirs. Ces derniers existent mais ne changent rien au déroulement de la partie. Nous n'affichons également pas les images correspondant à chaque combinaison peuple-pouvoir disponible. Il manque également la présence initiale de jetons peuple neutre afin d'empêcher les joueurs de conquérir un trop grand nombre de territoire durant les premiers tours.

4.4 Ce qui est différent du modèle UML

Durant l'implémentation, nous avons préféré changer l'héritage de spécialisation des cases du plateau, des peuples et des pouvoirs afin d'éviter une trop grande quantité de fichier et rendre l'arborescence du projet plus lisible. Désormais ce sont simplement des attributs type dans la sous-classe correspondante.

Chapitre 5

Bilan

5.1 Améliorations possibles

Nous aurions pu améliorer notre jeu de la façon suivante :
- Nous aurions pu ajouter une musique d'ambiance ainsi que des effets sonores lors d'un évènement.
- Il manque la grande carte pour pouvoir contenir un nombre de joueurs plus important.
- Il manque l'implémentation des règles des pouvoirs.
- Il manque l'implémentation des jetons spéciaux.
- Le jeu serait plus intéressant si nous avions ajouté davantage de peuples et de pouvoirs.
- Une page supplémentaire dans le menu afin de pouvoir configurer des options (graphiques, sons, etc.)

5.2 Conclusion

Le fait d'avoir à réaliser un jeu rendait le projet ludique et donnait envie de s'investir. Il nous a permis d'appliquer les concepts vus en cours, mais aussi d'avoir à aller chercher plus loin, que ce soit dans la documentation ou les possibilités offertes par le langage Java. Nous avons également dû utiliser des outils de versionnage, la maitrîse de ces outils ne pourra qu'être un plus dans notre carrière professionnelle.

Chapitre 6

Annexes

6.1 Diagramme des cas d'utilisation

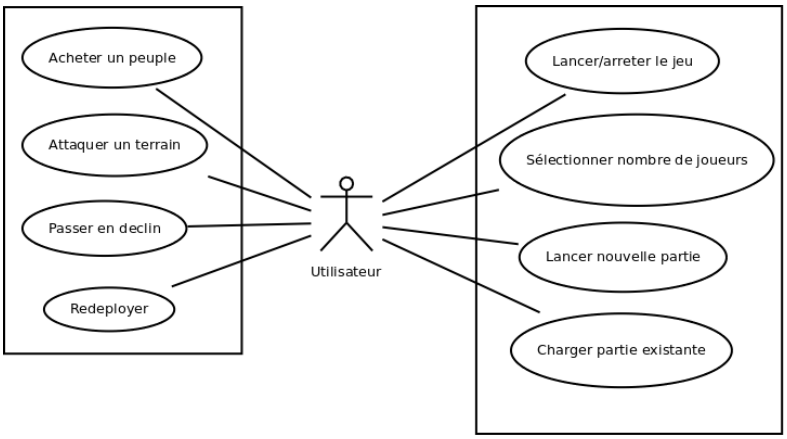

Figure 6.1 – Diagramme des cas d'utilisation du *Smallworld*

6.2 Diagramme de séquence

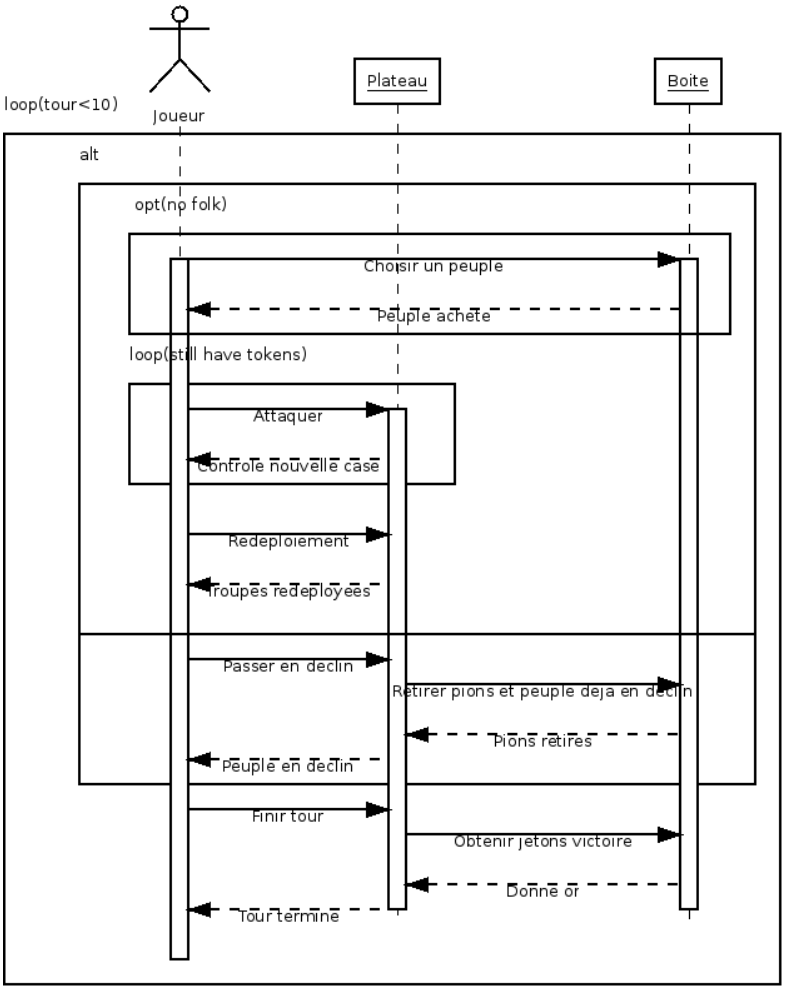

Figure 6.2 – Diagramme de séquence d'un tour de jeu

6.3 Diagramme de classes

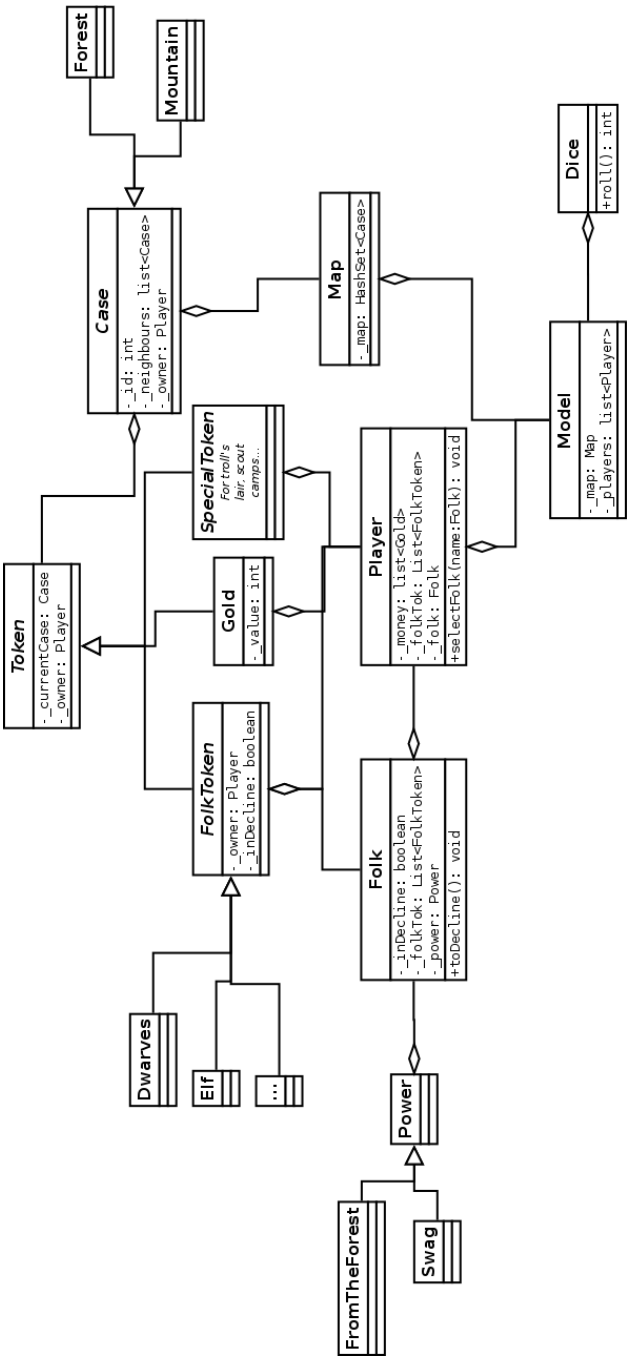

Figure 6.3 – Diagramme de classe du modèle

6.4 Interface

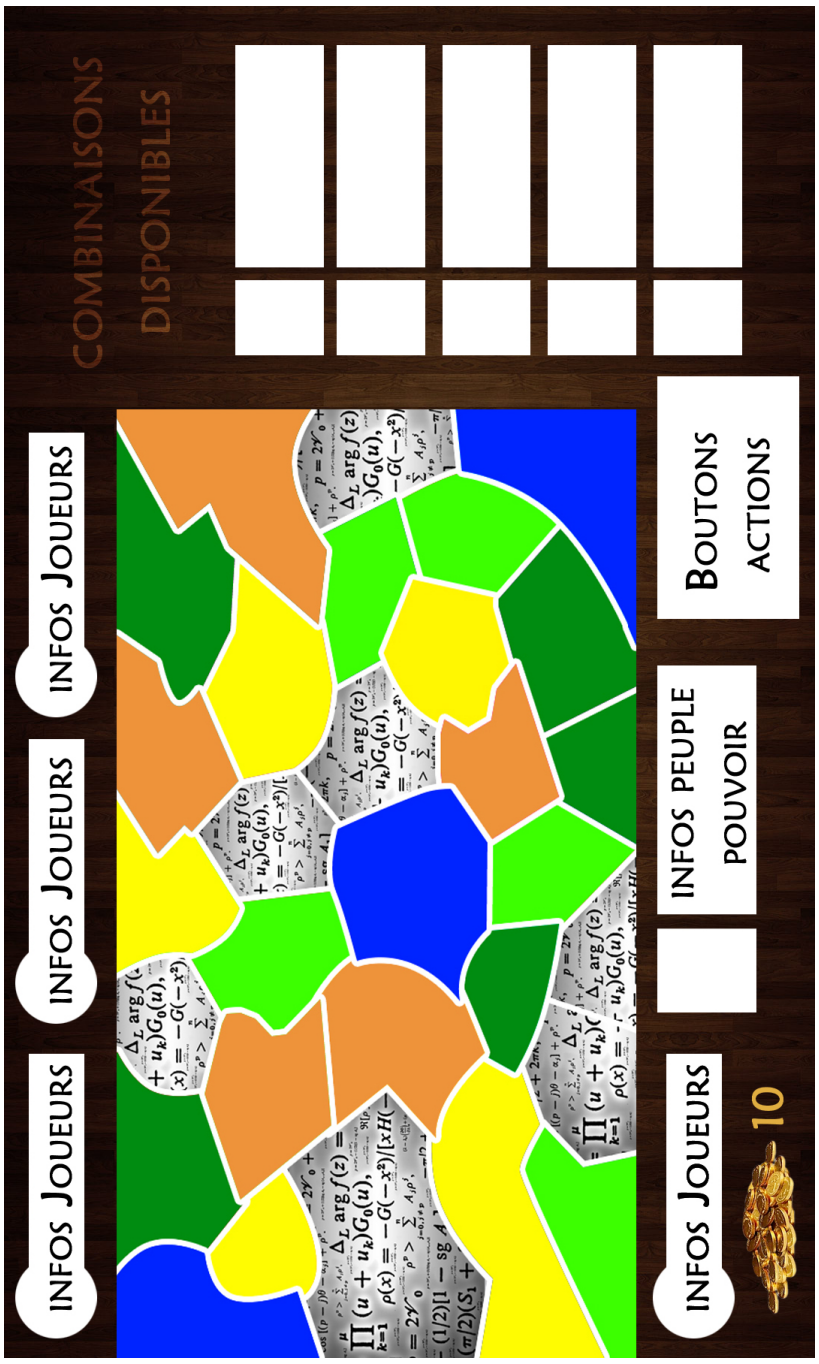

Figure 6.4 – Maquette de l'interface lors d'un tour de jeu