Équipe : Le Conseil des Sept

Joël Collinet

Félix Erb

Célia Kocher

Vincent Moitry

Adam Rimelé

Carlo Spiga

Maxime Zimmer

**Projet Acrobatt 2014**

**La Clémence d'Auguste**

**Tuteur : Pierre Kraemer**

**COMPTE-RENDU – SEMAINE 1 – JEUDI 13 ET VENDREDI 14 FÉVRIER**

Rôles :

* **Vincent Moitry** : chef de projet, participe à la création du site de base.
* **Joël Collinet** : participe à la création du site de base.
* **Félix Erb** : analyse le code existant, participe à la création de l’application côté serveur.
* **Maxime Zimmer** : designer du site, participe à la création du site de base.
* **Carlo Spiga** : secrétaire, participe à la création de l’application côté serveur.
* **Adam Rimelé** : analyse le code existant, participe à la création de l’application côté serveur.
* **Célia Kocher** participe à la création de l’application côté serveur.

Au cours de la première semaine, le sujet est découvert. Un modèle à appliquer est rapidement adopté par l'ensemble de l'équipe. Côté serveur, on s'occupe de la génération de la partie en elle-même et des calculs (plateau, combats, …). Côté client, on réalise une application lourde par support (exécutable pour ordinateur, application pour Android), ainsi qu'une application web côté client (pour navigateur).

Une liste des informations à transférer est établie. Le serveur effectuant tous les calculs, il doit créer la partie, et envoyer le plateau de jeu avec sa taille aux clients (pour permettre un affichage optimal), les légions et équipes (nombre de légions, nombre d'équipes, position des légions, joueurs composant les équipes), la position du laurier, le temps restant avant la fin du tour courant, la durée totale de la partie.

Le nombre de joueurs est susceptible de varier. Il pourrait être intéressant de permettre à un joueur de rejoindre une partie en cours de réalisation, et permettre son affectation à une équipe/légion.

Il est éventuellement possible de gérer la distribution et l’utilisation des cartes procurant les effets spéciaux (si une carte est utilisée, elle est perdue, ne peut plus être utilisée ou récupérée, …).

Les liaisons entre client et serveur sont réalisées par la sérialisation d'objets en JSON, contenant toutes les informations nécessaires. La chaîne obtenue est transférée grâce à une socket. Le XML a également été évoqué, mais refusé. Sa structure arborescente est susceptible d'être moins adaptée au type d'informations que l'on souhaite transférer et plus complexe à manipuler lors de la lecture des données. Le CSV a également été mentionné, mais reste un format totalement inadapté au type d’informations transféré.

Si chacun doit pouvoir jouer sans inscription préalable, il pourrait être intéressant d’implémenter une section membre (avec mot de passe et informations personnelles). Les parties réalisées au préalable pourront ainsi être enregistrées, et on peut envisager un système de groupes/amis (un joueur rejoignant une partie peut proposer à ses amis de les y emmener automatiquement). Si un joueur n’est pas enregistré, on récupérera malgré tout son IP pour effectuer les traitements. Un joueur doit également pouvoir être spectateur d’une partie.

Il est prévu de coder un chat afin de tester simplement la bonne communication entre les différentes parties de la structure. Ce chat peut être conservé à l’avenir pour assurer une communication simple entre les joueurs d’une partie. Un chat par équipe pourrait également être utile pour faciliter l’élaboration d’une stratégie, mais le jeu serait alors moins fidèle à ce que l’on peut réaliser avec la version papier (où les joueurs élaborent ce type de stratégie plus difficilement).

Il faut également prévoir chaque changement d’état d’un joueur en cours de partie. S’il vient à subitement se déconnecter, il faudrait adapter la partie en conséquence (affecter ses légions à un autre joueur par exemple), un ping devrait suffire. Pour aller plus loin, on pourrait également prévoir les ralentissements de connexion. Il faut également prévoir le cas où le joueur n’a pas enregistré de coup avant la fin du temps imparti pour un tour.

Par tour, le joueur doit pouvoir définir le déplacement du laurier ou d’un légionnaire. Ce dernier doit pouvoir endosser une armure si sa case d’arrivée en contient une, mais il ne doit pas pouvoir se déplacer sur ce type de case s’il porte déjà une armure. Le joueur doit pouvoir annuler un déplacement avant la fin du temps imparti, s’il souhaite corriger sa stratégie par exemple.

Une charte de programmation sera également conçue. Au travers de cette charte, nous adopterons une syntaxe au sein du code (variables en minuscules, constantes en majuscules, … ). Il sera ainsi plus aisé de retrouver le rôle adopté par chaque point de code.

Une liste des objets et classes sera établie d’ici peu. Cette liste peut aider à concevoir d’éventuels diagrammes d’analyse simplifiés. On pourra ainsi visualiser plus aisément la structure que pourrait adopter l’application.

Tous les paramètres seront stockés dans un fichier de constantes, nous évitons ainsi de les “perdre” dans le projet. S’ils doivent être fréquemment changés, y accéder n’en sera que plus simple.

À notre disposition se trouve le code du jeu en lui-même. Nous n’excluons pas d’en récupérer certaines parties (les classes métier, l’affichage du plateau de jeu), mais nous pensons malgré tout repenser la structure de certaines fonctions pour qu’elle soit plus adaptée au protocole que nous souhaiterions adopter.