

Rapport de développement Jalon 3.2

Groupe 1.2

Mardi 4 Juin

0.1 Rapport de développement

Notre jeu est jouable en local, et suit le modèle MVC nous permettant d'ajouter prochainement une implémentation d'un mode en réseau. En dehors de l'ajout du réseau, nous sommes en train de finir le développement de l'IHM en version Swing, néanmoins nous avons implémenter toutes les autres fonctionnalités prévues.

Pour appliquer le modèle MVC, nous utilisons deux interfaces, GameController et GameView. GameController permet au différentes interfaces de communiquer à la Game ou dans prochainement au réseau. Quand à elle, l'interface GameView permet de communiquer aux différentes IHM, les données demandées ainsi que celles à afficher. Celles-ci sont séparés dans les IHM, afin de limiter la logique de l'application au GameController. Pour ce qui est des IHM, notre version console est finie, mais ce n'est pas encore le cas de celle en Swing.

Celle-ci implémente GameView et redéfinit ses méthodes afin de personnaliser les demandes et l'affichage à l'affichage. Le jeu est pensé pour pouvoir être jouable avec un nombre définissable de joueurs, de pions par joueurs. La taille du plateau de jeu est également définissable. Que ce soit en local, ou en réseau sur plusieurs machines, notre jeu peut supporter plusieurs joueurs et les gérer tous en même temps. Il n'existera dans le cas du réseau qu'une seule partie qui sera sur le serveur, et chaque joueur recevra des ordre d'affichage ou de demande sur son IHM depuis un GameController.

Nous avons initialement pensé que notre jeu devait simplement supporté le 1 contre 1, mais au final nous avons décidé de le rendre plus malléable en permettant à l'utilisateur de modifier cette donnée. Nous pouvons créer un sort en choisissant son élément et sa forme, une fois la partie lancée nous avons le choix entre lancer un sort ou se déplacer, pour ces méthodes on vérifie si la destination est valide, on gère aussi les points d'actions et de mouvement, les effets s'appliquent correctement ainsi que le cooldown¹.

Nous hésitions à ajouter des améliorations ou *buff*², mais finalement nous avons décidé ne pas implémenter cette option afin de nous concentrer sur des éléments plus importants de notre projet.

1. Le nombre de tour de rechargement entre 2 lancement de ce sort
2. en tant qu'effets positifs sur certains sorts

0.2 Rapport de test

Nous avons réalisé les tests pour les classes *Pawn*, *Board* ainsi que certaines méthodes de *Game*. Ces tests nous ont permis de détecter les erreurs³ et donc de vérifier que notre application soit conforme au cahier des charges. Pour la classe **Game** nous ne testons que la méthode `EndGame()`, afin de vérifier que celle-ci fonctionne correctement sans tester chaque cas possible. Pour les deux autres classes (**Pawn**, **Board**, nous testons intégralement leurs méthodes publiques. La phase de test a été très utile au fur et à mesure du développement car nous avons fréquemment changer la manière donc nous voyons et nous simplifions notre application.

0.3 Documentation Utilisateur

Lors de l'exécution du programme le joueur se trouve sur le menu principal, il peut alors choisir entre :

- Héberger une partie
- Rejoindre une partie
- Créer une partie locale
- Créer des pages de sorts

Héberger une partie redirige le joueur dans une salle d'attente ou il pourra sélectionner les pages de sorts qu'il va attribuer à ses pions. La partie est lancée lorsqu'au moins un autre joueur rejoint le salon et qu'il est prêt.

Rejoindre une partie permet au joueur de rejoindre une partie existante sur un ordinateur distant en saisissant son adress IP, le joueur va ensuite être redirigé dans une salle d'attente ou il pourra sélectionner les pages de sorts qu'il veut attribuer à ses pions, la partie va ensuite être lancée.

Créer une partie locale permet à au moins deux joueurs de de jouer sur un même écran et un même ordinateur.

Créer des pages de sorts permet au joueurs de créer des pages de sorts, qu'ils vont, par la suite, attribuer à leur pions. Le joueur va donc être redirigé sur un menu ou il pourra créer chacun des trois sorts qui constituent une page de sorts, pour valider la création d'une page de sorts le joueurs doit avoir créé les trois sorts qui la composent en sélectionnant pour chacun d'eux

3. Mauvaises intialisations, méthodes eronnées...

leur élément et leur forme, il faudra également saisir le nom de la page afin de valider la création de celle-ci.

Au cours d'une partie un joueur peut *sélectionner* un pion, il peut ensuite choisir entre déplacer le pion ou lui faire lancer un sort. Pour effectuer le déplacement d'un pion le joueur doit saisir sa destination ou le sort que le pion doit lancer. Si le déplacement n'est pas valide (qu'il est hors de portée ou que la destination est mal saisie) le système va réitérer la demande de déplacement. Plusieurs informations sont visibles au cours d'une partie, le joueur peut trouver une description détaillée (*Points de vie, Points d'action, Points de mouvement et page de sorts*) de son pion courant (*le pion qu'il a sélectionné*), il peut également trouver une brève description concernant les autres pions qui constituent son équipe mais également une description des pions des équipes adverses.

Notions utilisées

Un joueur possède un nombre de pions qu'il définit en début de partie.

Un pion possède des points de vie, des points d'action, des points de mouvement ainsi qu'une page de sorts

Les points de vie représentent l'état de santé d'un pion, lorsqu'ils tombent à 0 le pion est hors jeu.

Les points d'action permettent au joueur de lancer des sorts, ils représentent le coût d'un sort, ils se rechargent au cours d'une partie.

Les points de mouvement permettent au pion de se déplacer, le nombre qu'ils indiquent est en réalité le nombre de cases qu'un pion peut parcourir.

Les pages de sorts constituent un atout majeur, chacune est constituée de trois sorts, il faut les concevoir de manière stratégique.