

RAPPORT DE DÉVELOPPEMENT

Lacoste Antoine, Py Quentin, Weissend Eddy, Dichtel Jonathan, Valli Dylan



Fonctionnement de l'application :

Il est actuellement possible de lancer l'application, lors de son exécution 2 joueurs qui jouent aléatoirement au jeu.

Chaque joueur joue son tour aléatoirement en posant d'abord les cartes qu'il peut. Ensuite il attaque avec les cartes qui sont actives.

Chaque effet d'une carte peut-être soit activé en début de chaque tour, soit en fin de chaque tour soit à l'invocation de la carte, soit à la mort de la carte.

Les effets gérés sont :

- Soin d'une carte alliée ou de toutes les cartes alliées.

- Dégât direct sur une ou plusieurs cibles où le joueur choisit la(les) cible(s) ou dégâts directs sur toutes les cartes ennemis.

- Les buffs de points d'attaque ou de points de vie sur une carte alliée.

- Le joueur pioche une ou plusieurs cartes.

Ces effets peuvent être combinés avec les moments de l'activation de cet effet.

Deux méthodes `reOrganiserPlateau()` et `reOrganiserMain()` permettent de « caler » à gauche les cartes du plateau ou de la main du joueur à chaque tour.

Pour l'instant la création d'un deck n'est pas implémentée ainsi que la vue du cimetière.

Il y a l'affichage console qui permet de voir chaque action d'un joueur avec une description de l'action effectuée (carte posée, avec quelle carte il attaque telle carte, ...)

L'affichage graphique permet de voir l'évolution de la partie à la fin de chaque tour, chaque carte est représentée sur l'interface graphique. Chaque bouton visible correspond à une carte.

Les boutons du bas correspondent aux cartes de la main du joueur.

La 2^{ème} rangée de boutons correspond aux cartes posées du joueur.

La 3^{ème} rangée correspond aux cartes posées du joueur adverse.

La main du joueur adverse n'est donc pas représentée. Les informations relatives aux héros sont affichées dans la dernière rangée de bouton.

Dans la suite de ce rapport nous allons aborder le Bilan sous différente forme Tout d'abord la Bilan personnel puis le bilan technique

Bilan Personnel :

La conception et la programmation de ce projet nous a permis de nous rendre compte qu'une équipe n'était pas forcément homogène.

En effet notre Groupe est composé de 5 personnes est lors des différentes phases on a pu s'apercevoir que les gens sont plus à l'aise en conception, tout le monde a pu suivre la partie conception et a pu participer mais lorsqu'il est venu l'heure de programmer on a pu tout de suite se rendre compte que les lacunes de certains d'entre nous les ont découragé et que au final sur les 5 personne, Antoine Lacoste a beaucoup plus programmer (facilement >60% du code) ainsi qu'Eddy Weissend qui s'est plus impliqué dans le projet que les autres, quand à Jonathan Dichtel et Dylan Valli ils ont participé assez faiblement a la programmation mais on apporter leur connaissances pour le projet tandis que Quentin Py, du à son niveau très faible a complètement décrocher dès le début de la programmation .

Malgré tout, la cohésion du groupe lors des derniers jours pour le projet reste bonne, tout le monde à apporter ce qu'il a pu avec Quentin qui a repris la partie conception puisqu'il est plus à l'aise tandis que les autres ont terminé et donner les dernières détails dans le code.

Bilan Technique :

Comme Nous l'avons dit précédemment le bilan Technique reste bon sur la partie conception maigres un gros problème avec le logiciel BO UML puisque nous ne maitrisons pas ce logiciel au début et que vers la fin de la conception celui-ci nous a dit que le fichier était corrompu ce qui a entrainer des complications puisqu' il nous a été impossible de récupérer les cas d'utilisation, les diagrammes de séquences, et donc nous n'avions donc que les diagrammes de classe. Pour la partie programmation nous avons rencontré des problèmes lors de la synchronisation des compte sur Git Hub pour que tout le monde puissent programmer et lors des premier commit.

Maintenant en ce qui concerne les apports du projet au module COO/POO faisons un point personne par personne :

Eddy Weissend : Module COO : Pas grand-chose vu que les TP de ce module nous ont tout appris.

Module POO : J'ai pu apprendre les bases du code java et améliorer mon niveau qui été moyen au début et qui est finalement plus vers le bon je pense grâce aux TP POO.

Antoine Lacoste : Module COO : Pas grand-chose vu que les TP de ce module nous ont tout appris.

Module POO : maitriser les bases correctes, Héritage, Exception, Interface, Package, classe Abstraite, et SWING pour l'IHM

Jonathan Dichtel : Module COO : Maitriser toutes les sortes de diagramme nécessaires a la conception

Module POO : Héritage Exeption Swing et améliorer les bases

Dylan Valli : Module COO : Faire ce qu'on a vu en cours mais dans un cadre différent sans documentation ni aide

Module POO : Approfondir les Bases et les connaissances personnelles

Quentin Py : Module COO : Appliquer à une vraie situation la conception d'un projet

Module POO : Malgré le découragement les derniers jours m'ont permis de progresser un peu.