

Projet Python - Card Game



Egor Berezowsky
Quentin Ysambert
Damien Bidaud
Thibault Dassonvillé

Sommaire :

Profondeur de jeu	3
Ajout de propriétés	
Ajout d'actions serviteur	
Ajout de cartes sort	
Création/Gestion de decks	
Graphique	4
Jouabilité	5
Intelligence artificielle	
Partie en réseau local	

I - Profondeur de jeu

Pour notre projet de python, il nous été demandé plusieurs fonctionnalités et notamment de donné plus de profondeur au jeu. Cela passe par plusieurs ajouts comme des actions et propriétés pour les serviteurs, des cartes de sort ou encore la possibilité de modifier et gérer ses decks

1 - Ajout de propriétés

Plusieurs mécanismes ont été rajoutées aux cartes de serviteurs. Par exemple, certains serviteurs ont une compétence appelée provocation. La provocation fonctionne d'une manière particulière, si on attaque un joueur qui a invoqué une carte possédant la provocation, ce n'est pas le joueur ou les autres cartes qui prennent les dégâts mais le serviteur avec la provocation qui va intercepter l'attaque du joueur adverse.

La compétence Bouclier a aussi été implémentée. Une carte qui a un bouclier possède plus de vie que les autres, elle est donc plus difficile à tuer. Son bouclier est toujours égale à sa vie de base. Par exemple, un serviteur qui a 3 points de vie mais qui a "bouclier" devra subir 6 points de dégâts avant de mourir.

La compétence camouflage est une compétence intéressante. Un serviteur qui possède camouflage ne peut pas être attaqué ou ciblé par un sort avant d'avoir lui-même attaqué. Si on tente quand même de l'attaqué, le jeu génère une exception qui est ensuite pour affiché un message d'erreur : "Cette cible ne peut pas être attaqué".

Une autre fonctionnalité importante que nous avons mis en place est le système de faiblesse. En effet, chacun des serviteurs possède un attribut particulier. Ils existent trois de ces attributs : "magic", "distance" et "physical".

Nous avons ensuite mis en place un système de faiblesse entre ces trois éléments. En effet, les serviteurs "magic" font deux fois plus de dégâts aux cibles "physical", les "physical" deux fois plus de dégâts aux "distance" et les "distance" deux fois plus de dégâts aux "magic". Ce triangle de faiblesse est donc très important à prendre en compte quand on joue.

2 - Ajouts d'actions aux serviteurs

Pour rajouter des mécanismes et de la profondeur au jeu, nous avons rajouté des actions pour chaque serviteur. En effet, nous ne voulions pas que les serviteurs puissent juste attaquer mais qu'ils puissent avoir d'autres fonctionnalités.

3 - Ajouts de cartes de sorts

Pour enrichir le gameplay, en plus des cartes de serviteurs, nous avons mis en place des cartes de sorts. Ce sont des cartes qui ont un effet immédiat et qui disparaissent juste après avoir été jouée. Certaines infligent des dégâts, d'autres rendent des points de vie.

Pour les sorts qui infligent des dégâts, elle sont aussi soumis au principe de type de dommage "magic", "physical" et "distance". Elle ne peuvent pas cibler une cible "camouflée" et sont aussi interceptée par les serviteurs ayant la "provocation".

4 - Gestion et création de deck

Cette fonctionnalité est disponible mais ne possède pas d'interface qui lui est propre. En effet, pour créer et modifier des decks, il faut rajouter manuellement les cartes dans un fichier spécifique "/src/res/cards/deck.json". Il faut indiquer toutes les informations des chaque cartes (Id, nom, attaque, vie, ...). Le programme va ensuite, à partir de ce fichier, créer les decks qui seront jouables en jeu.

II - Axe Graphique

Pour la partie graphique, plusieurs fonctionnalités ont été mises en place. Au niveau de l'interface, nous affichons un menu principal permettant de choisir son mode de jeu comme une partie solo, partie multijoueur ou les options. Il apparaît donc un menu avec plusieurs boutons cliquables et un sous-menu pour les options du jeu. En cliquant sur "Partie", le programme affiche le plateau de jeu.

Le plateau de jeu se présente de la façon suivante : on peut voir son deck et sa main, la main est constituée de plusieurs cartes. Chaque carte est affichée avec ses points de vie, son attaque, son coût en mana et sa capacité. Lorsque l'on passe la souris sur une des cartes, elle apparaît en plus grande au centre de l'écran afin de mieux la voir. Lorsque l'on clique sur une carte et que la quantité de mana est suffisante, une animation déplace la carte vers un coin de l'écran. De plus, au début de chaque tour, une animation déplace une carte du deck vers la main du joueur.

Le plateau de jeu présente aussi les noms, les points de vie et de mana de chaque joueur. Les points sont actualisés après une attaque ou un soin. Un bouton dans un coin de l'écran permet aussi de passer son tour si on le souhaite.

III - Axe jouabilité

1 - Intelligence artificielle

Nous avons développé une artilange artificielle qui permet à un joueur de jouer une partie tout seul. Pour cela nous avons fait en sorte qu'en début de partie, l'IA puisse sélectionner un sort pour l'utiliser soit contre une carte sur le terrain ou contre le joueur adverse. Il peut aussi invoquer une créature pour la placer sur le terrain et ensuite utiliser les cartes placées sur son terrain pour les jouer contre les cartes adverses ou attaquer les points de vie de l'adversaire.

2 - Partie en réseau local

Les principales actions du jeu sont réalisées sur le serveur et quand le moment où le deuxième joueur doit jouer arrive un objet contenant l'état courant du jeu est envoyé au client qui peut jouer. Ensuite, après qu'il est terminé de jouer, un nouvel objet est renvoyé au serveur pour pouvoir continuer la partie et ainsi de suite jusqu'à la fin de la partie.