

Fiche descriptive de projet Hub

MTG MANAGER

Contexte et but du projet

Réaliser un outil pour aider à gérer le bon déroulement d'une partie du jeu "MAGIC: The Gathering" au format papier.

Cet outil permettra de guider les joueurs pendant la partie afin de pallier le manque d'expérience ou d'attention, pouvant être causé par des mécaniques ou des règles complexes.

Son utilisation est la suivante : le logiciel est lancé au début d'une partie. Chaque joueur peut envoyer son deck de jeu au format texte pour simplifier les interactions ultérieures. Pendant la partie, lorsqu'un joueur joue une carte, il peut entrer cette carte dans un champ de saisie pour l'ajouter à l'interface du logiciel et ainsi suivre les interactions de cette carte.

Le logiciel permet également d'indiquer les interactions des actions des autres joueurs sur les cartes du joueur concerné.

Résumé du projet

- Config / Parsing les cartes de chaque deck de jeu
- Créer une interface de zone de jeu (champ de bataille / cimetière / exil / zone de commandant / suspense)
- Système d'analyse de carte, via modèle entraîné
- Système du guide visuel (notification, surlignage, description)
- Automatisation des mécaniques basique du jeu (gestion des tours, gestion des ressources, application d'effet, gestion des point de vie, mode de jeu) [Hard part]
- Banque de données, pour effet et règles. -> Help menu
- traduction français / anglais [Bonus]

Porteur(s) du projet

Lucas Scheidecker - TEK2 - (à voir)

Nahim Kandari - TEK2 - (à voir)

Environnement technique / technologique

Exposer le contexte technique et technologique (matériel, langage, environnement d'exécution, ressources, ...) dans lequel le projet s'inscrit

Matériel : PC personnel

Langages : C++

Environnement d'exécution : Linux (migration si possible sur Windows)

Ressources : API Magic, SFML pour affichage, Chatbot API Key

Description du livrable

Détailler chaque élément (programmes, librairies, modules, assets, ...) du livrable et leur niveau de finition (déploiement, documentation, ...)

API MAGIC, pour récupérer les informations des cartes (visuel, description). Avec ces informations, on pourra les utiliser pour les filtrer et récupérer correctement ces interactions.

SFML, pour réaliser l'interface utilisateur. Celle-ci comporte un affichage avec toutes les cartes jouées, pouvant être sélectionnées pour un affichage zoom, une zone avec un champ de saisie, affichant ensuite les descriptions du mot ou de ces effets, une zone de notification pour indiquer chaque interaction devant être effectuée, des paramètres permettant de filtrer puis retirer ou déplacer des cartes de la zone de jeu. Affichage de chaque étape d'un tour, des points de vie ainsi que certaines informations liées à l'action du joueur comme le total de mana pouvant être produit par le joueur comme exemple.

ChatBot API key, pour générer des descriptions de cartes simples et guidées. Ceci sera affiché comme pop up dans l'interface.

Organisation et temporalité

1 – Tâche 1

Parsing des fichiers de config des decks de jeu

Temps estimé pour cette tâche: 10h

2 – Tâche 2

Récupération des informations d'une carte et interprétation des informations

Temps estimé pour cette tâche: 14h

3 – Tâche 3

Récupération et utilisation de l'API Magic, pour récupérer les images des cartes

Temps estimé pour cette tâche: 30h

4 – Tâche 4

Création de l'interface utilisateur.

Temps estimé pour cette tâche: 15h

5 – Tâche 5

Création du système du guide visuel, création de notification, surlignage d'élément ainsi que description optionnelle.

Temps estimé pour cette tâche: 30h

6 – Tâche 6

Ajout de la banque de données à l'interface, pour initialiser le menu d'aide.

Temps estimé pour cette tâche: 7h

7 – Tâche 7

Ajout d'un chatbot, pour expliquer les interactions du joueur à réaliser.

Temps estimé pour cette tâche: / (voir avec le temps restant)

Amélioration possible du projet en fonction du temps restant

Estimation du projet

Nous estimons à **30 XP** le projet, ce qui fait **15 XP** par personne.

- **Pourquoi ?**

Utilisation du C++, avec implémentation de design pattern (+ template + namespace)

Utilisation / Découverte d'une API Web

Algorithme de reconnaissance d'image

Implémentation de Chatbot