N'Guyen Van Hoan Théo – Mourrier Thomas

Projet Kalah

Sommaire :

N'Guyen Van Hoan Théo – Mourrier Thomas

1 - Présentation

Le but de ce projet était de réaliser un jeu à deux joueurs, n'étant pas basé sur le hasard, et d'implémenter une IA pour celui-ci.

Ce projet a pu etre mené quasiment jusqu'au bout ; c'est à dire qu'il ne manque que l'IA que je n'ai pas su implémenter.

Ce programme permet à deux joueurs de jouer au jeu du Kalah, il gère la gestion des tours, des règles du kalah, il permet aussi de sauvegarder sa partie, de recommencer en un clic lorsque la partie est terminée, et fournit aussi des informations dans la console en temps réel.

Le programme a été développé avec Python 2.7, et utilise la bibliothèque PyGame pour fonctionner : celle-ci permet une gestion plus simple (adaptée à la création de jeux) des ouvertures de fenetres, de la création de « sprites » (éléments à l'écran) et leurs mouvements.

A son lancement, une fenetre s'ouvre : celle-ci est graphique et affiche le jeu en lui-meme. Une souris est nécéssaire pour jouer.
Le jeu est prévu pour etre appellé depuis un terminal, puisqu'il affiche des informations sur l'état du jeu dans celui-ci à chaque clic.

2 – Fichiers

Le programme est constitué de 4 fichiers :

KalahGui : Ce fichier gère quasiment tout le programme, il contient les classes définisssant le plateau, les « maisons », les graines,… Il contient aussi la logique de sauvegarde, les méthodes gérant la distribution des graines…

KalahKeyEvents : Ce fichier gère les clics de souris ; que ce soit sur les boutons ou bien sur les cases du plateau.

AI : Ce fichier devrait contenir l'IA, mais il ne s'agit que d'une ébauche. Il devait récupérer l'état du plateau en copie, et effectuer des opérations dessus, mais cette étape m'a posé problème. Il est quasiment vide, et jamais utilisé dans le programme

Kalah : Il s'agit de la boucle principale du jeu. On y trouve quelques déclarations de variables, et la boucle principale qui permet d'appeler les fonctions nécéssaires au tour de chaque joueur.

3 – Fonctionnement

Le programme fonctionne de la manière suivante : la boucle principale est appelée, elle créé les éléments du plateau puis regarde si une sauvegarde existe, elle est chargée si c'est le cas. Ensuite, une boucle infinie va dessiner les éléments créés, et gérer le entrées utilisateur.
Lors d'un clic, on vérifie que le coup qu'a essayé de faire le joueur est valide (la case est à lui, la case n'est pas vide, …) et on applique alors les méthodes gérant les cases et leurs graines.

Lorsque l'on sauvegarde, on sérialise les données et on produit un fichier contenant celle-ci.

Lorsque la partie est terminée, le bouton sauvegarde est remplacé par un bouton remettant le plateau à l'état initial.

4 – Comportements

Le programme a plusieurs comportements qui peuvent etre innantendus : il ne permet pas de recharger sa partie à n'importe quel moment, il va le faire automatiquement au lancement du jeu si il trouve un fichier de sauvegarde ; lorsque l'utilisateur tente de faire une action invalide, le jeu est « bloqu é » pendant une seconde.

Sous certaines conditions, le jeu peut crasher au démarrage après qu'une sauvegarde ait été faite : c eproblème est corrigé simplement en effaçant le fichier de sauvegarde.

5 - Conclusion

La création d'un jeu est un projet très étendu, qui permet de développer beaucoup de fonctionnalités : la sérialisation des données pour la gestion de sauvegardes, les entrées utilisateurs pour la gestion des clics, la gestion des graphismes,… Et bien qu'il n'ait pas pu etre mené complètement à bout, il nous a permit d'apprendre beaucoup de choses : notamment en Python.