Projet HEX par Robert Gauvain et Fuzier Antoine

FICHE BILAN

Notre objectif: réaliser la V3.

I. Organisation du travail

- Répartition du travail : Répartition équitable, avec pour chacun un minimum de C et de java à faire.
- Communication dans le groupe : Pour mettre à jour et partager les différentes taches effectuées, nous utilisons Github, ce qui nous permet de voir qui à modifié quoi et quand. Nous pouvons ainsi facilement utilisé le travail de l'autre pour la suite des opérations lorsqu'il y a des dépendances entres les fichiers.

II. Réalisation de la V1

- Créer le graphe : Le bon fonctionnement du projet commence par l'implémentation d'un graphe dynamique fonctionnelle en langage C qui permet de gérer le jeu, d'insérer un pion au bon endroit, de détecter les coups possible et si il y a un gagnant ou non.
- Appeler les fonctions du graphe : Pour créer l'interface nous appelons les fonctions principales du graphe pour pouvoir les utiliser en Java. Ainsi, nous mettons en place notre bibliothèque qui convertit le C en Java.
- Créer l'interface : Interface console en Java qui permet à deux joueurs de s'affronter, de sauvegarder et charger une partie et d'annuler le dernier coup joué. On utilise les fonctions C pour récupérer le graphe, c'est à dire le plateau de jeu, et nous l'affichons. Nous proposons toutes ces fonctionnalités à l'utilisateur en affichant un menu et en lui proposant d'y accéder régulièrement. Nous gérons dans cette interface toute erreur éventuelle que pourrait saisir l'utilisateur, ou la mauvaise gestion qu'il pourrait en faire, comme charger une partie qui n'existe pas ou annuler un coup sans n'avoir rien joué, etc...

III. Réalisation de la V2

- Création d'une intelligence artificielle : Nous implémentons un joueur non humain qui joue des coups. L'utilisateur à le choix entre jouer ou non contre cet IA. Nous utilisons l'arbre minimax pour permettre à l'ordinateur de placer un pion de manière réfléchi.

IV. Réalisation de la V3

- Création d'une super IA : Cette fois nous proposons à l'utilisateur des niveaux de difficultés, de facile à impossible. Pour cela nous utilisons le graphe réduit qui permet de détecter les ponts entre groupe de pions, donc l'ia fait en sorte soit de bloquer l'avancé de l'adversaire ou de faire progresser la sienne de manière optimal

V. Les difficultés

- V1 : Pas de problème particulier pour faire le graphe. Nous avons eu quelques difficultés pour appeler les fonctions C que nous avons résolu en effectuant des recherches. Par contre pour l'interface en java, il y a fallu réfléchir à un moyen efficace de mettre en

place un système de sauvegarde et de chargement. Nous avons décidé d'utiliser deux fichiers, un pour une sauvegarde normal et un autre pour une sauvegarde temporaire. Le fichier temporaire enregistre la partie à tout moment et à chaque modification, si l'utilisateur veut sauvegarder la partie, on la copie dans le fichier de sauvegarde normal. Ainsi, nous pouvons jouer et conserver notre sauvegarde tout en faisant d'autre partie.

- V2 : A cause de la complexité importante de l'arbre minimax, surtout avec une grande taille le temps de chargement devient très long, il nous a fallu trouver une manière de réduire celle-ci. Pour cela on a diminué le nombre de fils par nœud (ceux ne renvoyant pas la valeur 1). Comme résultat, pour un graphe de taille 2x2, le temps d'exécution est de 0,3ms pour un processeur QuadCore de 1,7GHz.
  Du côté de l'interface java, il a fallu intégrer l'IA dans les choix de jeu et ce fut compliqué de l'intégrer aux fonctions de sauvegarde, de chargement et d'annulation de coup. Pour régler ce problème on rajoute toutes les informations sur l'IA dans le fichier de sauvegarde temporaire, comme son pion, si il a commencé le jeu, son niveau de difficulté etc. on peut donc tout récupérer facilement et gérer les éventements.
- V3 : Pas de problèmes particulier après avoir implémenté le graphe réduit que nous avons utilisé pour gérer le niveau de l'IA.

VI. Ce que nous avons appris

- Nous avons appris à gérer des fichiers en java, à appeler des fonctions C et à organiser un projet en groupe.

Conclusion : Ce projet nous a appris à nous organiser pour respecter des échéances et répartir efficacement la masse de travail. Finalement nous avons réussi à atteindre notre but. Nous avons seulement échoué à faire une interface graphique complète et avons donc gardé celle de base.