

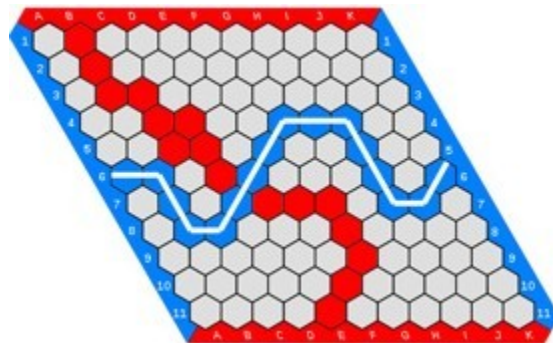
Projet S4 : Jeu de Hex

Groupe 27 (TDA3.2)

MMADI Anzilane

BEGG Rain-Alexandra

IFOUDINE Sara



I. Présentation du travail à faire

Ce projet s'agissait de l'implémentation du jeu de hex (un jeu de stratégie joué entre deux personnes) en C et en Java. A la fin de l'implémentation le résultat final devait présenter les fonctionnalités suivantes :

- initialiser le plateau
- choisir le joueur qui commence
- ne permettre que des coups légaux
- gérer les tours de chaque joueur
- détecter un vainqueur
- quitter le jeu avec ou sans sauvegarde
- et puis gérer une historique, la sauvegarde, la restauration, l'annulation du dernier coup, l'abandon de la partie en cours

en plus du code fourni en C et en Java, des requêtes de Base de Données étaient demandés, ainsi que des fichiers de test unitaires. Aussi, la gestion du projet (fiches préliminaires, fiches d'avancement, document de bilan) était considérée comme une grande composante du projet.

II. Objectifs

Au début du projet notre groupe s'est donnée la version 1 comme objectif, c'est à dire la possibilité de jouer le jeu humain contre humain. Cependant, on s'était mis d'accord pour tenter au maximum la version 2 si on dans le cas où on aura atteint nos objectifs.

En effet, on à réussi à implémenter l'intelligence artificielle en C, mais on a pas eu le temps de l'intégrer dans le jeu avec Java, donc on a créé un fichier qui permettra de tester son fonctionnement.

III. Répartition du travail

Au début du projet on s'est repartissait les tâches (en général) ainsi (voir annexe) :

- MMADI Anzilane – C
- BEGG Rain-Alexandra – Java
- IFOUDINE Sara – Base de Données

Même si on s'est mis d'accord pour se mettre en charge de ces parties là, il y a quand même

eu beaucoup d'entre-aide au niveau du codage et surtout au niveau de la réflexion des solutions aux problèmes rencontrés.

IV. Problèmes rencontrés et leurs solutions

Problème : Affichage du plateau

Vers la fin du projet on s'est rendu compte que plus on augmentait la taille du plateau, plus ça devenait difficile de ne pas se tromper lors de l'entrée des coordonnées.

Solution Par exemple pour une taille du plateau égale à 10, au lieu d'afficher le plateau ainsi:

```
W W W W W W W W W W W W/B
B . . . . . . . . . . B
B . . . . . . . . . . B
B . . . . . . . . . . B
B . . . . . . . . . . B
B . . . . . . . . . . B
B . . . . . . . . . . B
B . . . . . . . . . . B
B . . . . . . . . . . B
B . . . . . . . . . . B
B/W W W W W W W W W W W
```

on a décidé de l'afficher comme suit :

```
  0 1 2 3 4 5 6 7 8 9
W W W W W W W W W W W/B
0.B . . . . . . . . . . B
1..B . . . . . . . . . . B
2...B . . . . . . . . . . B
3....B . . . . . . . . . . B
4.....B . . . . . . . . . . B
5.....B . . . . . . . . . . B
6.....B . . . . . . . . . . B
7.....B . . . . . . . . . . B
8.....B . . . . . . . . . . B
9.....B . . . . . . . . . . B
      B/W W W W W W W W W W W
```

Problème : Méthode de recherche de groupes

Au début, on était parti pour parcourir tout le graphe et calculer les groupes à l'aide d'une pile, un liste pour chaque sommet et un tableau de hachage. Cependant, on s'est rendu compte que pour calculer les groupes après modification du graphe, on aurait été obligé de lui reparcourir à nouveau. Pour une taille de 3x3, ça n'aurait pas posé problème, mais pour les taille dépassant 5, cela aurait coûté cher.

Solution

On a décidé de redéfinir la structure des sommets en ajoutant un booléen disant si un sommet donné appartient à un groupe ou non, et aussi on ajoutant un int représentant la position du premier sommet ajouté à cette liste. Donc, les listes s'identifient à partir de cette position sauvegardée, ainsi on peut agrandir les listes au fur et à mesure et vérifier si une liste contient deux bords parallèle du plateau (et est donc un groupe gagnant).

Problème : Mise en commun entre C et Java

Solution

En s'exerçant sur des exemples (exemples en ligne, encapsulation en encapsulation, création des objets à partir des méthodes natives...) on a réussi à faire le lien entre les deux langages.

Problème : La portabilité du code

On essayé pendant très longtemps de rendre le jeu portable pour tout les systèmes car les fichiers de librairie change en fonction.

Solution

On a pas trouvé de solution, donc on s'est mis d'accord pour le rendre jouable uniquement sur le système Linux. Après on mettra dans le README les démarches à faire pour le rendre jouable sur les systèmes différents de Linux.

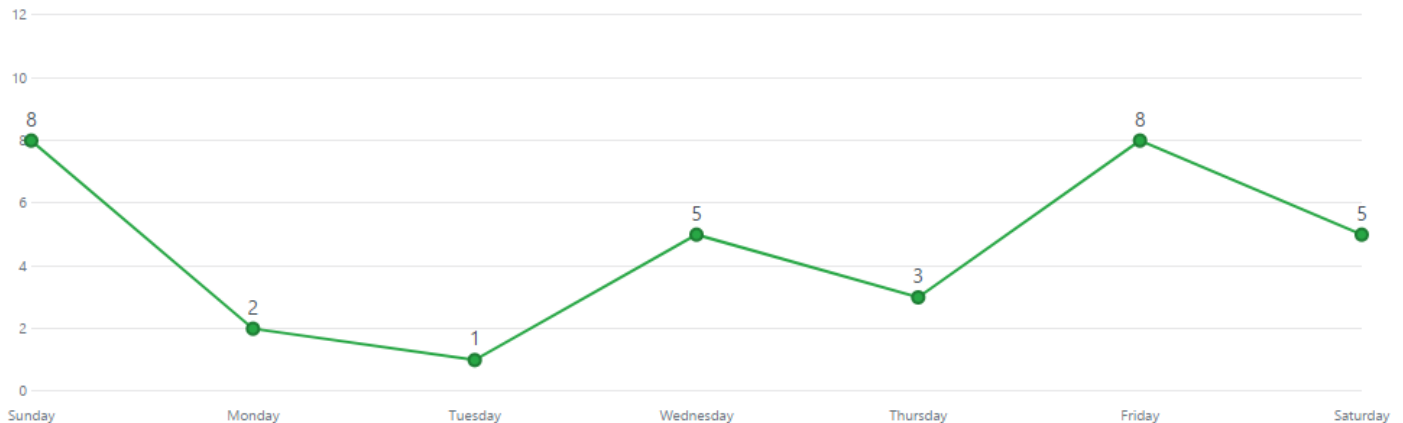
Problème. Incohérences du code entre co-équipiers

Au début on s'envoyait nos modifications par mail, mais cela nous amener à des incohérences du code entre-nous.

Solution

On s'est mis d'accord pour utiliser une éditeur de version (github) pour gérer nos modifications à nous et pouvoir les comparer avant de se décider sur une modification final.

Voici un graphe représentant le nombre des commits fait la semaine du 14 mai :



Voici le lien du projet : <https://github.com/RainAlexandra/ProjetS4BEGGIFOUDINEMMADI>

V. Conclusion

Au final, on a atteint tout nos objectifs. Par contre on à eu vraiment du mal à respecter les échéances qu'on s'est donné au début du projet. Par conséquent, même si l'intelligence artificielle à était implémenté, on a pas pu aller jusqu'au bout avec.

*L'image sur la page de garde a été prise depuis la page Wikipédia du jeu de hex.