

CAHIER DES CHARGES

I. PRESENTATION DU PROJET

Le jeu de Hex est un jeu de société combinatoire abstrait créé par John Nash en 1952. Il met en confrontation deux joueurs sur un tablier en forme de losange dont les cases sont hexagonales. On peut trouver des similitudes dans d'autres jeux de sociétés comme le Go, par exemple. C'est un jeu de stratégie qui a inspiré beaucoup de domaines de recherche, non seulement dans la théorie des jeux, mais encore dans des branches des mathématiques comme la topologie ou la géométrie algébrique.

Le tablier de jeu peut être de taille variée, mais il est traditionnellement de taille 11 x 11, mais on peut trouver 13 x 13 ou bien 19 x 19. Les recherches ont montré que lorsque la taille du tablier est strictement inférieure à 9, le premier joueur a plus de chance de gagné.

II. DESCRIPTION DE LA DEMANDE

RESULTATS ATTENDUS :

- but de jeu : jouer à deux, pouvoir gagner
- joueurs différenciés par deux couleurs
- pour gagner : lier les deux bords avec une chaîne de pions de même couleur
- but du projet : apprendre à travailler en groupe, gérer un projet, mettre en commun les différentes UE dans un même projet

FONCTIONNALITES :

- 1 initialiser le plateau pour le jeu (N x N)
- 2 gérer le jeu humain contre humain
- 3 choix du joueur qui commence
- 4 Gérer les tours de jeu de chaque joueur
- 5 Vérifier la légalité des coups des joueurs
- 6 Détection d'un vainqueur éventuel
- 7 Gérer une historique, la sauvegarde, la restauration, l'annulation du dernier coup, l'abandon de la partie en cours, lancer une nouvelle partie en cours de partie
- 8 Quitter le jeu avec ou sans sauvegarde

III. CONTRAINTES

CONTRAINTES DE DEVELOPPEMENT

- interface et fonctionnalités du jeu en java
- calcul du jeu en c
- utilisation d'une Base de données pour infos sur les joueurs et les parties
- utilisation des structures de données pour gérer les calculs

CONTRAINTES EN TEMPS

- temps : à rendre avant le 22 mai

IV. DEROULEMENT DU PROJET

LES EXIGENCES DU PROJET

EXIGENCE 1 :

- Le fournisseur devra concevoir un cahier des charges et le rendre le 13 mars 2017

EXIGENCE 2 :

- Le fournisseur devra faire un dossier de conception et le rendre le 27 mars 2017

EXIGENCE 3 :

- Rendre 3 fiches d'avancements et les rendre aux dates suivantes :
 - fiche 1 : 17 avril 2017
 - fiche 2 : 8 mai 2017
 - fiche 3 : 15 mai 2017

EXIGENCE 4 :

- Remettre le projet final (code + bilan) le 22 mai 2017

PROGRAMMATION DU JEU

EXIGENCES 1 : V1 Humain contre humain

EXIGENCES 1.1 : INTERFACE (EN JAVA) :

- Initialisation de la taille du plateau (N x N)
- Affichage de l'ensemble des fonctionnalités sous forme de menu
(Voir partie fonctionnalités).
- Affichage des informations

-joueurs, numéro du tour de jeu, dernier coup joué, nombre de coups...

EXIGENCES 1.1.1 : Gérer les actions de l'utilisateur

➔ Livrable : action effectuée et sauvegardée

EXIGENCES 1.1.2 Lancer les fonctionnalités (voir partie fonctionnalités)

EXIGENCES 1.1.3 Mise à jour de « l'interface »

-menus, plateau de jeu avec les nouvelles cases jouées

➔ Livrable : Interface du jeu fini

EXIGENCES 1.2 : CONTENU TECHNIQUE DU JEU (en C)

EXIGENCES 1.2.1 : Création d'une SD dynamique à l'aide d'un graphe

➔ Livrable : SD (structure de données) dynamique pour gérer le plateau du jeu

EXIGENCES 1.2.2 : Calculer les groupes et la détection d'un groupe gagnant

EXIGENCES 1.2.3 : Gérer dynamiquement la liste des groupes de chaque joueur

➔ Livrable : Liste contenant les différents groupes de chaque joueur

EXIGENCES 1.2.4 : programmer une SD pour le graphe réduit, s'en servir pour les calculs du jeu

➔ Livrable : Permet de trouver le gagnant de la partie

➔ Livrable : Bibliothèque de fonctions C permettant des calculs en Java

EXIGENCES 1.3 : BASES DE DONNEES

-retrouver les parties effectuées par un joueur à une date donnée et aussi toutes les parties effectuées par un joueur

-retrouver les parties gagnées par un joueur

-retrouver les parties abandonnées auxquelles à participé un joueur

-retrouver les joueurs rencontrés par un joueur donné

-retrouver les parties débutées par un coup donné

➔ Livrable : Une bibliothèque de parties qui sauvegardera les informations concernant les joueurs et les parties

➔ Livrable V1 : Jeu de Hex humain contre humain fonctionnel

EXIGENCES 2 : -V2 Humain contre ordinateur

EXIGENCES 2.1 : V1 ➔ Livrable : V1 fini

EXIGENCES 2.2 : CONTENU TECHNIQUE DU JEU (en C)

EXIGENCES 2.2.1 : Construire une SD dynamique pour représenter un arbre de jeu gérant les coefficients minimax

EXIGENCES 2.2.2 : Programmer le minimax exhaustif sur un plateau de taille N (implique : gestion des groupes, groupe gagnant)

EXIGENCES 2.2.3 : le tester pour l'arbre de jeu d'un plateau 2 × 2

EXIGENCES 2.2.4 : Parcourir l'arbre pour trouver une stratégie gagnante

EXIGENCES 2.2.5 : Essayer l'algo sur des N plus grands et commenter dans le document de bilan chiffres à l'appui

➔ Livrable : Bibliothèque de fonctions C permettant des calculs en Java

➔ Livrable V2 : Jeu de Hex humain contre ordinateur et fonctionnel

EXIGENCES 3 : V3 Humain contre ordinateur (avec Intelligence Artificielle)

EXIGENCES 3.1 V2 ➔ Livrable : V2 fini

EXIGENCES 3.2 : CONTENU TECHNIQUE DU JEU (en Java ou C)

EXIGENCES 3.2.1 : Détecter les ponts

EXIGENCES 3.2.2 : Gérer les groupes de groupes séparés par des ponts

EXIGENCES 3.2.3 : Mettre à jour ces SD à chaque tour de jeu pour chaque joueur

EXIGENCES 3.2.4 : programmer un comportement de l'ordinateur cherchant à connecter ses groupes et couper ceux de l'adversaire

➔ Livrable : Bibliothèque de fonctions C permettant des calculs en Java

➔ Livrable V3 : Jeu de Hex humain contre ordinateur et fonctionnel (avec Intelligence Artificielle)

