ESGI - Module d'algorithmique et structures de données – 1ère année

Projet annuel

VorassicPark

1. Description du projet

Le projet consiste à écrire :

- un algorithme en pseudo code permettant d'arbitrer une partie de Vorassic Park entre deux joueurs, qui entrent leurs coups au clavier. Le but du jeu Vorassic Park est d'engranger plus de points que son adversaire en capturant les cases d'un espace commun.
- un programme en langage C correspondant à cet algorithme, ainsi qu'à d'éventuelles sophistications (voir avec M. Sananes pour les détails)

2. Informations pratiques

	Algorithmique	
Point projet/ présoutenance	Groupe 1 : 24/04	Groupe 2 : Avril (à préciser)
Date de rendu	Avant le 22/05 à minuit	
Soutenance	G1 : 26/05	G2 : 28/05

- Le partage de code (ainsi que sa mise en ligne sur internet - forums compris - est strictement interdit et sévèrement puni (0 en note de projet et convocation au conseil de discipline).
- L'entraide et les discussions entre binômes sont autorisées, voire encouragées, sous certaines conditions à la fois strictes et de bon sens : avoir déjà réfléchi à la structure de son algorithme/programme auparavant et l'écrire soi-même
- Livrables :
    - Vous devez envoyer par email un bref rapport qui décrit les fonctions que vous avez choisi de définir (paramètres, type de retour, et sémantique), les difficultés rencontrées, les limitations, et les bugs (s'il y en a).
- Les algorithmes sont à envoyer par email à fbaudoin@myges.com

3. Règles du jeu

3.1 Plateau de jeu

Le jeu se joue sur un plateau carré, de n x n cases. Chaque case contient un nombre entre 0 et 9.
On choisira la taille n de la grille - dans l'intervalle [5 ; 26] - au début du jeu. Chacune des colonnes est numérotée de 0 à n-1. Les cases contiennent des valeurs entre 0 et 9. Par défaut, ces valeurs sont fonction de la distance aux deux cases en coin haut gauche et bas droite, comme le montre l'exemple ci-dessous. Pour une taille de 5, la configuration de départ sera la suivante :

```
   | A | B | C | D | E |
-----------------------
 0 |  J | 1  | 2  | 3 | 4  |
-----------------------
```

1 | 1 | 2 | 3 | 4 | 3 |
---
2 | 2 | 3 | 4 | 3 | 2 |
---
3 | 3 | 4 | 3 | 2 | 1 |
---
4 | 4 | 3 | 2 | 1 | R |

Deux joueurs s'affrontent sur ce plateau, jaune et rouge. Au départ, jaune possède la case haut gauche tandis que rouge possède la case bas droite. C'est jaune qui commence. Chacun, à tour de rôle, doit jouer un coup. Il est interdit de passer son tour tant qu'il reste des coups à jouer. Lorsque plus aucun joueur ne peut jouer, la partie s'arrête.

3.2 Les coups - mode serpent

Un coup consiste à donner la case que l'on souhaite capturer. Attention ! Il y a une limitation aux coups jouables, on ne peut capturer une case que si :
1. elle est libre
2. elle est adjacente à la dernière case capturée
3. le mouvement se fait vers le haut, vers le bas, vers la droite ou vers la gauche. Les mouvements en diagonale sont interdits.

Dans notre exemple, jaune peut jouer soit B0, soit A1, et il est obligé de choisir l'une de ces deux options. Supposons qu'il joue A1, la grille devient :

| **A** | **B** | **C** | **D** | **E** |
---
0 | j | 1 | 2 | 3 | 4 |
---
1 | J | 2 | 3 | 4 | 3 |
---
2 | 2 | 3 | 4 | 3 | 2 |
---
3 | 3 | 4 | 3 | 2 | 1 |
---
4 | 4 | 3 | 2 | 1 | R |
---

et le score est maintenant de 1 à 0 en faveur de jaune. Au tour suivant, (après que rouge aura joué, supposons, D4), jaune devra capturer soit la case B1, soit la case A2. S'il joue B1 la grille devient :

| **A** | **B** | **C** | **D** | **E** |
---
0 | j | 1 | 2 | 3 | 4 |
---
1 | j | J | 3 | 4 | 3 |
---
2 | 2 | 3 | 4 | 3 | 2 |
---
3 | 3 | 4 | 3 | 2 | 1 |
---
4 | 4 | 3 | 2 | R | r |

Le score, à ce moment de la partie, est de 3 à 1 en faveur de jaune.

3.3 Les coups - mode pieuvre

Le principe est le même que pour le mode serpent, à une différence prêt : on peut aller dans une case dès qu'elle se trouve à côté d'une case déjà capturée (vs à la dernière case capturée pour le mode serpent). Donc, on ne peut capturer une case si:
1. elle est libre
2. elle est adjacente à une case déjà capturée.
3. le mouvement se fait vers le haut, vers le bas, vers la droite, vers la gauche ou en diagonale haut-droite, en diagonale haut-gauche, en diagonale bas- droite, en diagonale bas-gauche. Les mouvements en diagonale sont autorisés.

Dans l'exemple ci dessous, jaune peut jouer en B0, A1, A2, A3, B3, C3, C2, C1, C0

	A	**B**	**C**	**D**	**E**
0	J	1	2	3	4
1	1	J	3	4	3
2	2	J	4	3	R
3	3	4	3	2	R
4	4	3	2	1	R

3.4 Fin et gain de la partie

Le jeu s'arrête lorsque plus aucun joueur ne peut jouer. Ainsi, si rouge ne peut plus jouer mais que jaune peut encore jouer, la partie continue. Le vainqueur est celui qui a amassé le plus de points à la fin de la partie.

4. Spécification de votre programme

Le programme minimum doit:

a. demander la taille de la grille de jeu
b. générer et afficher la grille de départ : celle-ci sera remplie soit par défaut comme expliqué ci-avant, soit au hasard avec un générateur de nombres aléatoires.
c. demander le mode de jeu (mode serpent OU mode pieuvre)
d. demander à tour de rôle à chaque joueur le coup qu'il veut jouer
e. gérer le déroulement et la fin de partie.
f. afficher, après chaque coup l'état de la grille en mode texte dans la console, ainsi que les scores. Une représentation possible est celle indiquée dans ce document mais c'est loin d'être la seule.

5. Extensions possibles

Faire une ou plusieurs extensions vous permettra de glaner des points supplémentaires.
Aucun barème n'est officiellement fourni, le nombre de points dépendant à la fois :
- de la difficulté intrinsèque des améliorations apportées
- de leur degré de finition
- de la quantité d'améliorations apportées

Extension 1 : Sauvegarde/Lecture d'une partie dans un fichier. Vous pouvez décider de sauvegarder et de reconstituer une partie à partir d'un fichier texte comportant la grille ainsi que chaque coup.

Extensions 2 : Améliorations du jeu. Vous pouvez proposer des éléments nouveaux qui rende le jeu plus sophistiqué : Ajout de murs (des cases imprenables), des cases à valeur négatives, jeu multi joueur (4, un dans chaque coin)

Extension 3 : Affichage graphique. Vous pouvez décider d'afficher une vrai grille en mode graphique. Vous pouvez aussi réagir aux clics de la souris sur les zones de votre grille, au lieu de lire les coups sur la console.

Extension 4 : Intelligence artificielle. Vous pouvez écrire un programme qui permet à un joueur humain de jouer contre l'ordinateur. Vous pouvez par exemple vous documenter sur l'algorithme min-max. Il existe aussi des algorithmes très simples, facilement implémentables, et qui marchent relativement bien: ce sont les algorithmes dits gourmands (greedy en anglais).

Autres extensions : N'hésitez pas à proposer et programmer vos propres extensions du programme telles que: jeu en réseau, version portable, …

6. Critères d'évaluation

Volet algorithmique :
- à quel degré votre algorithme correspond-il aux spécifications du projet ?
- êtes-vous capable de modifier votre algorithme en réponse à un changement de spécification ?
- le découpage en fonctions rend il l'algorithme
    - factorisé (sans code dupliqué)
    - lisible et facilement maintenable