

Projets de Langage C et Programmation Shell (2006)

Modalités

- Il est possible (et conseillé) de travailler en binôme.
- Le travail est à rendre pour le vendredi 7 avril.
- Vous devez rendre un compte-rendu imprimé expliquant le problème et votre façon de le résoudre. Vous vous attacherez à expliquer non pas ce que fait votre programme (nous savons lire du code), mais pourquoi vous avez fait ces choix.
- Vous indiquerez également dans votre compte-rendu le temps que vous avez passé sur ce projet.
- Le partiel (sans document) comportera des questions sur votre projet. Ne copiez pas.
- Les sujets ne sont pas tous de la même difficulté, et nous en tiendrons compte lors de l'évaluation.

Sujet 1 : Le mastermind

L'objet du jeu est de deviner l'ordre de quatre pions colorés choisis au hasard (bleu, rouge, jaune ou vert) par l'ordinateur. L'ordinateur peut parfaitement choisir plusieurs fois la même couleur.

À chaque étape, on choisit une séquence de 4 pions et l'ordinateur indique combien de pions sont bien placés et combien de pions de la bonne couleur sont mal placés. Si au bout de 8 étapes, le joueur n'a pas trouvé la solution, l'ordinateur affiche le résultat. Vous *pouvez* également développer une IA pour ce jeu.

Sujet 2 : Encodeur/décodeur de code Morse

Le code Morse a longtemps été utilisé pour la transmission télégraphique de messages sur longue distance. Il consiste à remplacer chaque lettre de l'alphabet, chaque chiffre et certains éléments de ponctuation par un ensemble de points ('.' ou ti) et de traits ('-' ou ta). Le code peut par exemple être trouvé à l'adresse suivante : <http://www.babbage.demon.co.uk/morseabc.html>. Les caractères d'un mot sont séparés par un blanc et les mots par plusieurs blancs.

Le but du projet est de programmer un encodeur/décodeur de code morse qui puisse générer le code Morse correspondant à un fichier texte ASCII en le sauvegardant dans un autre fichier texte contenant des points, des traits et des espaces et à l'inverse traduire un fichier contenant du code Morse vers son équivalent ASCII.

Sujet 3 : Puissance 4

De hauteur 6 et de largeur 7, la grille du puissance 4 est soumise à la pesanteur. Les pions glissent vers la position la plus basse possible. Le jeu se joue à deux joueurs chacun responsable d'une couleur et l'objectif est d'aligner 4 pions de sa couleur. Le jeu doit s'arrêter et indiquer qui est le gagnant dès que cette condition est vérifiée.

Il faudra programmer le jeu en affichant les résultats dans la console (les couleurs étant remplacées par des formes du genre 'X' ou 'O'), d'abord de façon à ce que deux personnes puissent jouer l'une contre l'autre et ensuite *éventuellement* de façon à ce que l'on puisse jouer seul contre l'ordinateur (algorithme minimax).

Sujet 4 : Traitement d'images PPM

Une image au format PPM (Portable PixMap) est un fichier texte qui doit commencer par un en-tête comportant 4 indications obligatoires séparées par un ou plusieurs espaces ou bien un saut de ligne :

- un code (nombre magique) qui pourra être P3 ou P6 (en majuscules).
- la largeur de l'image (nombre de pixels par ligne nb_pixel).
- la hauteur de l'image (nombre de lignes nb_ligne).
- le niveau de couleur maximal (Ex : 255. Avec cette valeur, on pourra représenter 256 couleurs).

- d'éventuels commentaires commençant par le symbole #.

Suivent ensuite $3 \times \text{nb_ligne} \times \text{nb_pixel}$ valeurs entières (ASCII pour le format P3, binaires pour le format P6) comprises entre 0 et le niveau de couleur maximal. Chaque pixel est ainsi représenté par les valeurs RGB (red green blue) correspondant à sa couleur. Ces valeurs RGB (dans cet ordre) sont écrites dans le fichier pour tous les pixels d'une ligne, ligne par ligne, en partant du haut à gauche.

Le but du projet est de manipuler des images au format PPM (de type P3 dans un premier temps) en lecture/écriture en appliquant des traitements d'images simples comme la transformation en noir et blanc, les filtres de couleurs (n'afficher dans l'image-résultat que les pixels dont la couleur est proche d'une couleur donnée) ou le lissage spatial (la couleur d'un pixel devient la moyenne des huit pixels voisins).

Sujet 5 : Le jeu de la vie

Inventé en 1970 par John Conway, le jeu de la vie est un automate cellulaire aux propriétés remarquables. Il est constitué par une grille rectangulaire (théoriquement infinie) de cellules élémentaires pouvant avoir deux états : vivantes ou mortes. La règle d'évolution de chaque cellule en fonction de l'état des huit cellules voisines est très simple :

- une cellule morte possédant exactement trois voisines vivantes devient vivante (elle naît).
- une cellule vivante possédant deux ou trois voisines vivantes le reste, sinon elle meurt.

En dépit de la simplicité de ces règles, on peut observer sur la grille des comportements émergents singuliers, comme des structures périodiques, des glisseurs et des canons (voir http://fr.wikipedia.org/wiki/Jeu_de_la_vie pour une description de ces phénomènes).

Le but du projet est de programmer un jeu de la vie en mode texte (dans la console) pour des grilles de taille donnée et d'observer son évolution en fonction d'états initiaux différents qui devront être spécifiés dans un fichier texte. On pourra penser à pouvoir modifier les règles du jeu de la vie par un fichier de configuration (nombre de cellules nécessaires pour naître/mourir, etc.)

Sujet 6 : Les véhicules de Braitenberg

Ce sujet est plus difficile, mais plus amusant que les autres...

Un robot doit évoluer dans une pièce constituée d'obstacles (murs, meubles, autres robots, etc) grâce à huit capteurs (par exemple des sonars) lui procurant une information relative à la distance d'un obstacle dans une direction donnée (haut bas gauche droite et les quatre diagonales) afin d'éviter les chocs.

Le but est de trouver une règle de comportement du robot (dans quelle direction se déplacer étant données les informations des capteurs) qui soit la plus exploratoire possible (ne pas bouger du tout est un comportement possible, mais peu intéressant). On pourra discrétiser la pièce en la représentant comme une grille et considérer que le robot a la taille d'une case de cette grille. Le robot peut alors se déplacer sur n'importe laquelle des huit cases voisines de la sienne.

L'affichage de fera en mode texte dans la console et on devra pouvoir charger le plan d'une pièce à partir d'un fichier texte.

Voir <http://www.mindspring.com/~gerken/vehicles/> pour une introduction plus complète, et <http://www.lcc.gatech.edu/~mateas/courses/LCC6310-Fall2005/> pour un cours parlant de ces véhicules.