Université de Bordeaux - Collège S & T — Licence S&TS Mention Informatique S5
2015/2016

UE INF5011 — Programmation 3

Projet

À remettre le vendredi 11 Décembre 2015

Jeu de Sudoku

On peut trouver une description du jeu de Sudoku à l'url suivante :
`https://fr.wikipedia.org/wiki/Sudoku`

1 *Sudoku* interactif

Dans cette première partie, il est demandé de réaliser une interface en mode texte pour le jeu de *Sudoku*. On demandera au joueur, la référence de la case où il désire jouer (entrée sous forme d'une lettre désignant la colonne et d'un chiffre désignant la ligne) ainsi que le chiffre à placer dans la case.
Une référence de case ou un chiffre invalide devra être signalée au joueur de manière appropriée. Une case est référencée par un couple de coordonnées L, i où L est une lettre entre A à I représentant la colonne et i un chiffre représentant la ligne.
Un exemple de sortie possible est présenté Figure 1.

Modalités

Le point d'entrée du programme sera une fonction `sudoku (grid)` prenant en paramètre une grille sous forme de tableau à deux dimmensions et se terminant avec la partie.
Vous rendrez pour cette section

- un fichier `.lisp`,
- **ou** un ensemble de fichiers `.lisp` dont un sert à charger tous les autres,
- **ou** un fichier`.asd` (Makefile) et un ensemble de fichiers `.lisp`.

2 Stratégies non interactives

On souhaite des stratégies jouables par l'ordinateur.

2.1 Stratégie aléatoire

Pour mettre en place la possibilité d'avoir des stratégies non interactives, commencer par implémenter une stratégie aléatoire : le programme joue un coup aléatoire parmi les coups possibles.

2.2 Intelligence artificielle

On souhaite dans un second temps de réaliser une **intelligence artificielle** pour le jeu c'est-à-dire des stratégies plus *intelligentes* que la stratégie aléatoire. On pourra développer une ou plusieurs stratégies et tester leurs performances.

2.3 Tournoi

Une petite partie de l'évaluation du projet se fera en faisant jouer les stratégies sur un ensemble de grilles proposées par les enseignants enseignants. Les IA devront répondre en moins de un dixième de seconde. [1]
Le programme `tournoi` vérifiera que les coups proposés par une stratégie sont des coups valables et détectera la fin de jeu.
Une IA ne respectant pas les règles du jeu, ou ne répondant pas dans le temps imparti sera automatiquement disqualifiée.

Modalités

Pour cette section, on demande **un seul** fichier `.lisp` (qui pourra être obtenu par concaténation d'un ensemble de fichiers `.lisp` convenablement commentés).
Le programme devra fournir deux points d'entrée :

1. une fonction nommée `init-standalone (grid)` dont le paramètre `grid` est un tableau à deux dimensions correspondant à la grille à résoudre et grâce à laquelle, on initialisera les structures de données nécessaires au fonctionnement de la stratégie proposée.
2. une fonction nommée `main-standalone ()` retournant trois valeurs : le numéro de ligne, le numéro de colonne et le chiffre joué, `NIL` si jouer est impossible).

1. On considérera les machines utilisées en TD comme référence de performance.

3 Modalités générales

1. Le travail doit être réalisé en **trinômes** constitué au sein d'un même groupe de TD. La composition des trinômes doit être transmise aux enseignants de TD au plus tard le lundi **9 novembre 2015**.
2. L'ensemble des fichiers associés au projet doit être géré avec un outil de gestion de révision (`svn`, `git`, ...).
3. Chaque enseignant de TD proposera sa solution pour la remise du projet (archive, mail, soutenance,...).
4. La plus grande partie de l'évaluation portera sur la qualité et l'organisation du code.

4 Interface Web (optionnel)

Réaliser une interface Web en s'aidant de l'ébauche d'interface Web fournie basée sur la bibliothèque `hunchentoot`.

```
SUDOKU> *grid*
#2A((1 0 0 0 0 4 0 0 5)
    (0 0 0 9 5 0 0 8 0)
    (0 0 0 0 0 3 0 9 0)
    (0 0 5 0 0 2 0 0 4)
    (0 0 1 0 6 0 7 0 0)
    (7 0 0 3 0 0 2 0 0)
    (0 6 0 5 0 0 0 0 0)
    (0 8 0 0 1 6 0 0 0)
    (5 0 0 2 0 0 0 0 7))
SUDOKU> (sudoku *grid*)
   | A  B  C | D  E  F | G  H  I |
*********************************
 1 | 1       |       4 |       5 |
 2 |         | 9  5    |    8    |
 3 |         |       3 |    9    |
*********************************
 4 |       5 |       2 |       4 |
 5 |       1 |    6    | 7       |
 6 | 7       | 3       | 2       |
*********************************
 7 |    6    | 5       |         |
 8 |    8    |    1  6 |         |
 9 | 5       | 2       |       7 |
*********************************
56
C L? B 1
Value? 2
   | A  B  C | D  E  F | G  H  I |
*********************************
 1 | 1  2    |       4 |       5 |
 2 |         | 9  5    |    8    |
 3 |         |       3 |    9    |
*********************************
 4 |       5 |       2 |       4 |
 5 |       1 |    6    | 7       |
 6 | 7       | 3       | 2       |
*********************************
 7 |    6    | 5       |         |
 8 |    8    |    1  6 |         |
 9 | 5       | 2       |       7 |
*********************************
55
C L? C 1
Value? 9
   | A  B  C | D  E  F | G  H  I |
*********************************
 1 | 1  2  9 |       4 |       5 |
 2 |         | 9  5    |    8    |
 3 |         |       3 |    9    |
*********************************
 4 |       5 |       2 |       4 |
 5 |       1 |    6    | 7       |
 6 | 7       | 3       | 2       |
```