▶ **HX4 & HX1 – Option Informatique**
Année 2011, Huitième TP Caml
Louis Jachiet (`http://www.eleves.ens.fr/home/jachiet/`)

Puissance 4 :: IA

1 Interface

On va représenter les grilles du puissance 4 sous la forme d'une matrice 7×6. Une case contenant un zéro est vide, une qui contient un contient un pion jaune, une qui contient deux, un rouge.

```
let dessine =
  open_graph "␣700x600" ;
  let corres = [|white;yellow;red|] in
  let decal x = x*100+50 in
  fun t ->
    for x = 0 to 6 do
      for y = 0 to 5 do
        let nx,ny = decal x,decal y in
        set_color corres.(t.(y).(x)) ;
        fill_circle nx ny 47 ;
        set_color black ;
        moveto (nx-50)(ny-50) ;
        lineto (nx+50)(ny-50) ;
        lineto (nx+50)(ny+50) ;
        lineto (nx-50)(ny+50) ;
        lineto (nx-50)(ny-50)
      done
    done
      ;;
```

▶ **Question 1** *Écrire la fonction* hauteur *qui prend un entier c et une grille et renvoie le nombre de pions dans la colonne c.*

▶ **Question 2** *Écrire la fonction* joue *qui prend une grille, c et j et met un pion de couleur j dans la colonne c au bon endroit*

▶ **Question 3** *Écrire la fonction* annule *qui prend une grille et un indice d'une colonne et enlève le dernier pion mis dans cette colonne*

▶ **Question 4** *Écrire la fonction* nul *qui prend une grille et renvoie un booléen selon si toutes les cases sont pleines ou pas*

▶ **Question 5** *Écrire la fonction* lit_long *qui étant donné ax, ay, coul, x, y parcourt les cases* $c_k = (x + k \times ax, y + k \times ay)$, *k=1,2,... jusqu'à ce que la case* c_k *soit en dehors de la grille ou d'une couleur différente de k. Elle renvoie alors le k correspondant.*

▶ **Question 6** *Écrire la fonction* lit_dir *qui prend une grille,* (x, y) *et* (ax, ay) *et renvoie un booléen indiquant s'il existe des cases alignés dans la même direction, de même couleur et dont l'une est* (x, y).

▶ **Question 7 *** *Écrire une fonction* gagne *qui prend une grille, une colonne où l'on vient de jouer et teste si la partie vient d'être gagnée. On peut faire 2 sous-fonctions : une qui compte le nombre de jetons consécutifs dans un sens donné à partir d'une coordonnée donnée, une qui regarde dans les deux sens et dit si la direction est gagnante. Enfin, on regarde si, dans la liste* $[(1,0);(1,1);(0,1);(-1,1)]$, *une direction est gagnante.*

2 Joueurs

Une stratégie en caml sera une fonction *(int vect vect * int -> int)*. Elle prend une grille et une couleur et renvoie l'indice de la colonne où elle veut jouer.

▶ **Question 8** *Écrire une fonction* partie *qui prend une grille, deux stratégies* f_1, f_2 *et deux couleurs de joueurs* c_1, c_2 *et simule l'exécution de la partie en affichant tout au fur et à mesure.*

2.1 IA_{random}

▶ **Question 9** *Écrire une IA qui joue au hasard. (utiliser random__int.)*

2.2 IA humaine

On veut pouvoir jouer à la souris. Pour récupérer la colonne où le joueur a fait un click :

```
let st = (wait_next_event [Button_down]).mouse_x/100 in
```

▶ **Question 10** *Écrire une fonction* humain *qui fait jouer un humain.*

3 IA intelligente

▶ **Question 11** *Programmer les différentes IAs ci-dessous.*

3.1 IA_1

IA_1 regarde si elle peut gagner en jouant un coup et sinon elle laisse IA_{random} jouer.

3.2 sIA_n

sIA_n utilise la stratégie récursive suivante, qui associe à un couple (grille,joueur) un des trois états suivants : je suis sûr de pouvoir gagner (G), je suis ni sûr de gagner, ni de perdre(I), je suis sûr de perdre si l'autre joue bien (P).
Sa stratégie repose sur l'idée suivante :
Je suis G si et seulement si je peux gagner tout de suite ou si en jouant dans une bonne colonne j'arrive dans un état P.
Je suis P si et seulement si quelque soit la colonne où je joue j'arrive dans un état G.
Sinon je suis I.
On pourrait n'avoir aucun nœud I et notre algorithme serait parfait, mais le nombre de branches à explorer serait extrêmement important. On choisit donc de dire que les nœuds qui ont k cases remplies de plus que notre nœud de départ ont un état donné par ce que dirait IA_1.

3.3 $\alpha - \beta$

Cette fois-ci on n'a plus seulement l'opposition Gagnant/Perdant/Indéfini mais à chaque nœud on associe une valeur $f(nud)$. On travaille par récurrence. La valeur d'un nœud à profondeur k est définie comme suit : Si x est en dehors de $]\alpha;\beta[$, ou si la profondeur du nœud est n on renvoie x. Sinon, on renvoie moins le min de ce que peut faire l'opposant sur chacun des coups que l'on peut jouer.

▶ **Question 12 *** *Proposer des fonctions f, les pseudo-coder. Coder l'algorithme.*
■