

SUPINFO Academic Dept.

**Algorithmique en Python**

Graded exercise

Durée : 4 heures

Version 1.0

Last update: 25/11/2015

Use: Students/Staff

Author: Laurent GODEFROY

SOMMAIRE

0 Préambule 3

1 Le jeu « Puissance 4 » 3

2 Programmation de ce jeu en Python 4

2.1 Implémentation du plateau 4

2.2 Affichage du plateau 4

2.3 Gestion d’un coup 5

2.3.1 Calcul de la hauteur d’une colonne 5

2.3.2 Une colonne est-elle pleine ? 5

2.3.3 Pose d’un pion par le joueur 5

2.3.4 Pose d’un pion par l’ordinateur 6

2.4 Vérification du fait qu’un coup soit gagnant ou non 6

2.4.1 Alignement horizontal 6

2.4.2 Alignement vertical 6

2.4.3 Alignement dans la première diagonale 6

2.4.4 Alignement dans la seconde diagonale 6

2.4.5 Coup gagnant ou non 6

2.5 Gestion d’une partie 7

2.6 Bonus : paramétrage de la taille du plateau et du nombre de pions à aligner 8

# Préambule

Votre rendu se fera sous la forme d’un document « .py » que vous nommerez en respectant la convention **ID-1ADS-votreCampus-TP** (exemple : 66280-1ADS-Tours-TP). Vous le joindrez à un mail dont l’objet sera ce même nom.

Vous enverrez ce mail à **votre formateur**, et vous mettrez **en copie** la boîte [1ADS@supinfo.com](mailto:1ADS@supinfo.com). Si vous ne mettez pas cette dernière adresse en copie aucune réclamation ultérieure ne sera recevable.

Le non respect de ces consignes entrainera automatiquement une non correction de votre examen et par conséquent une note de 0 (cf. règlement des études).

Vous êtes invités à commenter vos codes si vous le jugez nécessaire à une bonne compréhension par votre correcteur.

Pour rappel, seuls les quatre premiers supports de cours au format ppt sont autorisés. En aucun cas la correction des exercices des séances Labs n'est consultable. Pas de consultations de sites web.

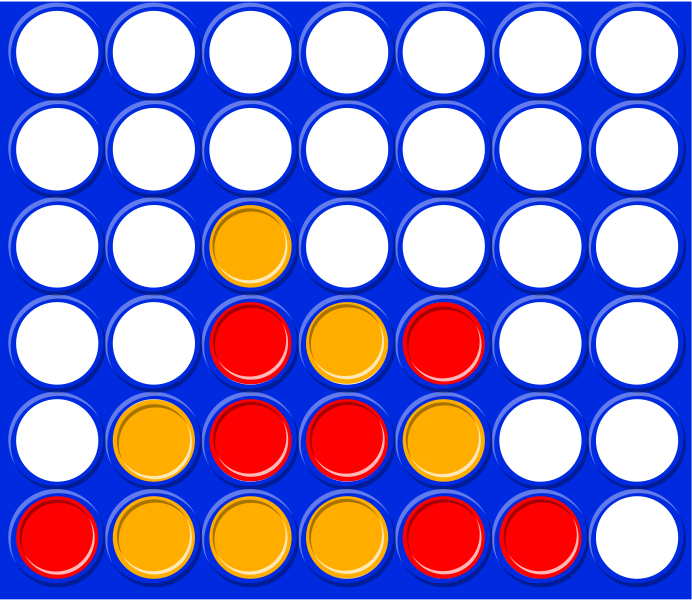
# Le jeu « Puissance 4 »

Le but de ce TP est d’implémenter en Python le célèbre jeu « Puissance 4 », commercialisé en 1974 par la société « Milton Bradley Company », plus connue sous le nom de « MB », et appartenant aujourd’hui à la firme « Hasbro ».

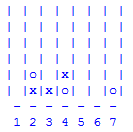
En fait, une fois terminé, ce programme vous permettra de jouer contre votre ordinateur, ce dernier jouant de façon aléatoire.

Rappelons brièvement les règles : il s’agit sur un plateau de jeu comportant 6 lignes et 7 colonnes, d’aligner au moins quatre de ses pions, et ce horizontalement, verticalement ou en diagonale. Les deux joueurs posent un pion chacun leur tour, et l’on considère que la grille est soumise aux lois de la gravité, c’est à dire que les pions « tombent » au fond des colonnes. Le premier joueur qui dans ces conditions arrive à aligner au moins quatre de ses pions est alors le gagnant du jeu.

Voici une « photo » d’une partie :



Le plateau que l’on affichera sera lui de conception plus modeste, mais tout à fait suffisante pour pouvoir jouer de façon satisfaisante :



# Programmation de ce jeu en Python

**Vous êtes invités à lire cette partie en entier avant de commencer à écrire vos codes.**

## Implémentation du plateau

Le plateau de jeu sera implémenté en Python par une liste à deux dimensions d’entiers (6 lignes et 7 colonnes) dont les valeurs seront :

* 0 pour une case vide.
* 1 pour un pion du joueur.
* -1 pour un pion de l’ordinateur.

Au début d’une partie cette liste ne comportera bien sûr que des 0.

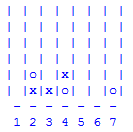
## Affichage du plateau

Ecrire une procédure prenant en paramètre un plateau et qui en réalise l’affichage selon la convention suivante :

* Un O sera représenté par un ‘ ‘.
* Un 1 sera représenté par un ‘x’.
* Un -1 sera représenté par un ‘o’.

On fera en sort de délimiter les colonnes avec des “|“, et d’inscrire en dessous de chaque colonne son numéro.

Le plateau s’affichera donc comme cela (au bout de quelques tours de jeu) :



La liste correspondante à ce plateau étant [[0,0,0,0,0,0,0], [0,0,0,0,0,0,0], [0,0,0,0,0,0,0], [0,0,0,0,0,0,0], [0,-1,0,1,0,0,0], [0,1,1,-1,0,0,-1]].

## Gestion d’un coup

### Calcul de la hauteur d’une colonne

Ecrire une fonction prenant en paramètre un plateau et un numéro de colonne et qui retourne le nombre de pions de la colonne en question.

### Une colonne est-elle pleine ?

Ecrire une fonction prenant en paramètre un plateau et un numéro de colonne et qui retourne un booléen indiquant si la colonne en question est pleine ou non.

### Pose d’un pion par le joueur

Ecrire une fonction prenant en paramètre un plateau, qui fait saisir au joueur un numéro de colonne valide (nombre entre 1 et 7 correspondant à une colonne non pleine), et qui retourne ce numéro.

### Pose d’un pion par l’ordinateur

Ecrire une fonction prenant en paramètre un plateau, qui tire au hasard un numéro de colonne valide (nombre entre 1 et 7 correspondant à une colonne non pleine), et qui retourne ce numéro.

Indication : la fonction « **randint(p,q)**» du module « **random** » retourne un nombre aléatoire entier supérieur ou égal à **p** et inférieur ou égal à **q**.

## Vérification du fait qu’un coup soit gagnant ou non

### Alignement horizontal

Ecrire une fonction prenant en paramètre un plateau, un numéro de colonne correspondant au dernier coup joué, et un entier indiquant qui vient de jouer (cet entier vaudra 1 si c’est le joueur et -1 si c’est l’ordinateur).

Cette fonction retournera un booléen indiquant si le dernier coup joué réalise un alignement horizontal d’au moins quatre pions appartenant à celui venant de jouer.

### Alignement vertical

Sur le même modèle, écrire une fonction testant cette fois-ci un alignement vertical.

### Alignement dans la première diagonale

Sur le même modèle, écrire une fonction testant cette fois-ci un alignement dans la première diagonale (en bas à gauche vers en haut à droite).

### Alignement dans la seconde diagonale

Sur le même modèle, écrire une fonction testant cette fois-ci un alignement dans la seconde diagonale (en haut à gauche vers en bas à droite).

### Coup gagnant ou non

Ecrire une fonction prenant en paramètre un plateau, un numéro de colonne correspondant au dernier coup joué, et un entier indiquant qui vient de jouer (cet entier vaudra 1 si c’est le joueur et -1 si c’est l’ordinateur).

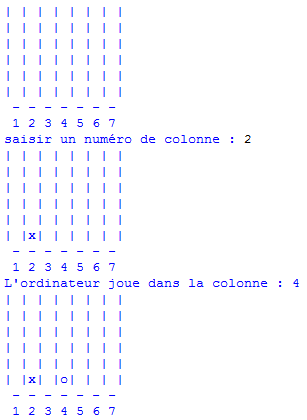
Cette fonction retournera un booléen indiquant si le dernier coup joué réalise un alignement d’au moins quatre pions appartenant à celui venant de jouer dans l’une des quatre directions.

## Gestion d’une partie

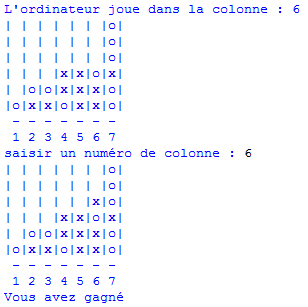
Ecrire une procédure principale utilisant les fonctions et procédures précédentes qui permet à l’utilisateur de jouer contre l’ordinateur.

On supposera que le joueur commence la partie. On fera jouer alternativement celui-ci avec l’ordinateur jusqu’à que l’un des deux réalise un coup gagnant ou que le plateau soit rempli. On indiquera pour finir le résultat : « match nul », « victoire du joueur » ou « victoire de l’ordinateur ».

Voici un exemple de début de partie :



Et voici un exemple de fin de partie :



## Bonus : paramétrage de la taille du plateau et du nombre de pions à aligner

Modifier le code précédent pour que l’on puisse jouer avec des plateaux de taille quelconque (et non plus nécessairement 6 lignes et 7 colonnes), et un nombre de pions à aligner quelconque (et non plus nécessairement 4).