

Puissance 4 : Ordinateur contre joueur

Le jeu



Puissance 4 est un jeu de stratégie combinatoire qui se joue exclusivement à deux joueurs.

Les règles de Puissance 4

Début de partie

Pour commencer une partie de puissance 4, on désigne le joueur qui commence.

Déroulement de la partie

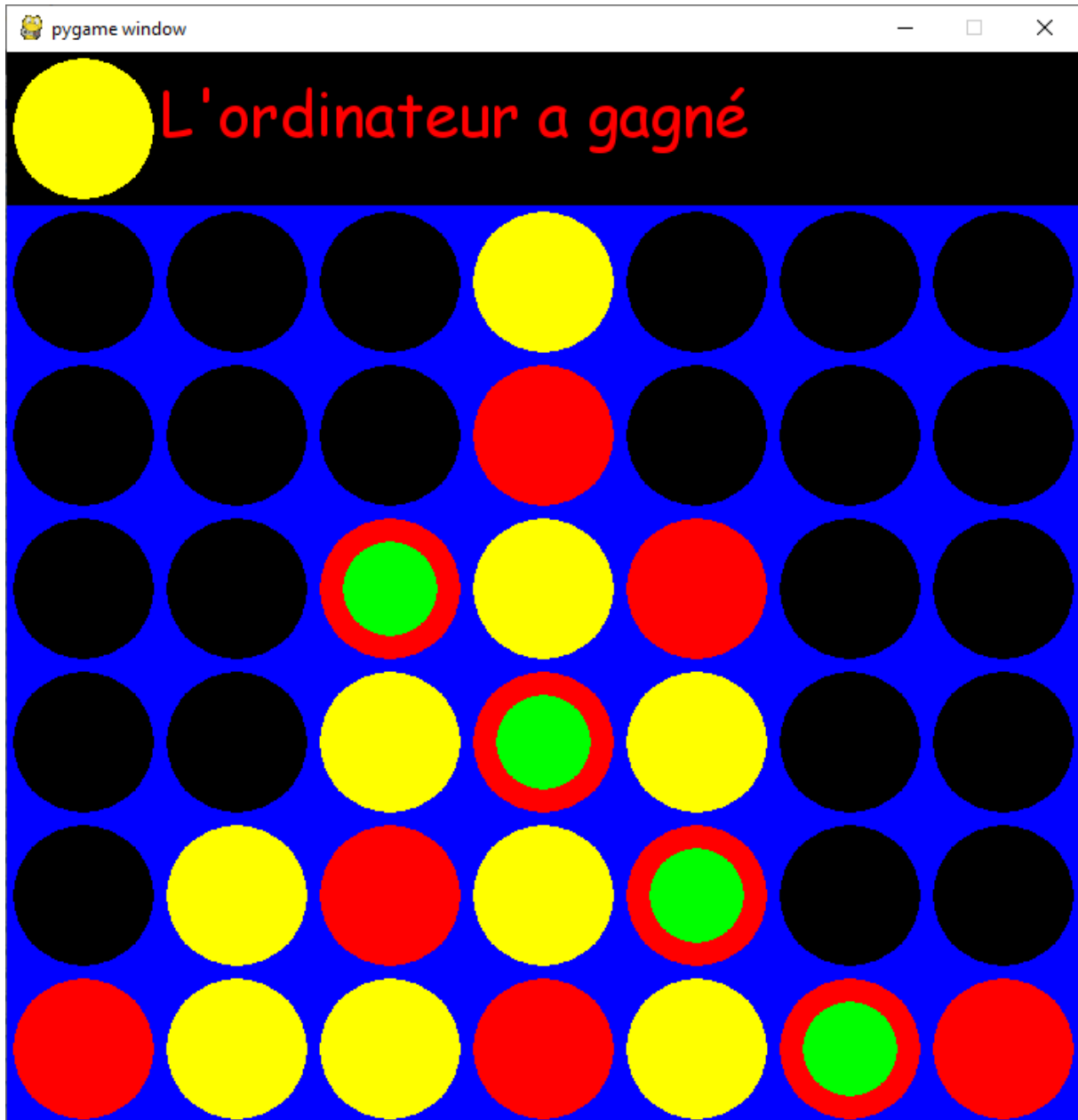
Il met un de ses jetons de couleur dans l'une des colonnes de son choix. Le jeton se retrouve en bas de la colonne. Le deuxième joueur insère à son tour son jeton, de l'autre couleur dans la colonne de son choix. Les joueurs répètent la manœuvre jusqu'à ce qu'un joueur aligne 4 pions.

Fin de la partie

Le premier joueur à aligner 4 pions, que ce soit horizontalement, verticalement ou en diagonale, remporte la partie.

Nous partons de la version graphique du jeu et nous y ajoutons une partie Intelligence artificielle pour trouver la meilleure colonne à jouer lorsque c'est au tour de l'ordinateur.

Affichage du plateau de jeu



Lorsqu'un joueur gagne, de façon à repérer facilement les jetons alignés, nous marquons les jetons gagnants avec une pastille verte.

Cela entraîne donc une modification des fonctions `dessiner_plateau` et `coup_gagnant`.

Si un coup est gagnant, la fonction modifie le plateau en y inscrivant la valeur 3 dans les cases gagnantes si c'est l'ordinateur qui gagne ou 4 si c'est le joueur.

La fonction `coup_gagnant` est aussi utilisée par l'algorithme `Min_Max` de recherche de la meilleure colonne pour le jeu de l'ordinateur, mais dans ce cas il ne faut pas modifier le plateau en cas de victoire. La fonction `coup_gagnant` aura donc un paramètre supplémentaire indiquant s'il faut modifier ou non le plateau.

Nous utilisons les constantes suivantes :

#les constantes

`LARGEUR_CASE = 100`

`BLEU = (0,0,255)`

`ROUGE = (255,0,0)`

`JAUNE = (255,255,0)`

`NOIR = (0,0,0)`

`BLANC = (255,255,255)`

`VERT = (0,255,0)`

`ROUGE_CLAIR = (254,180,180)`

`VERT_CLAIR = (33,255,33)`

`BLEU_CLAIR=(0,176,240)`

`VIOLET = (153,0,204)`

`VIOLET_CLAIR = (232,167,255)`

`RAYON= int(LARGEUR_CASE/2)-4`

`NB_LIGNE = 6`

`NB_COLONNE=7`

`ORDINATEUR = 1`

`HUMAIN = 2`

`CASE_VIDE = 0`

`GAGNANT_ORDI = 3`

`GAGNANT_HUMAIN = 4`

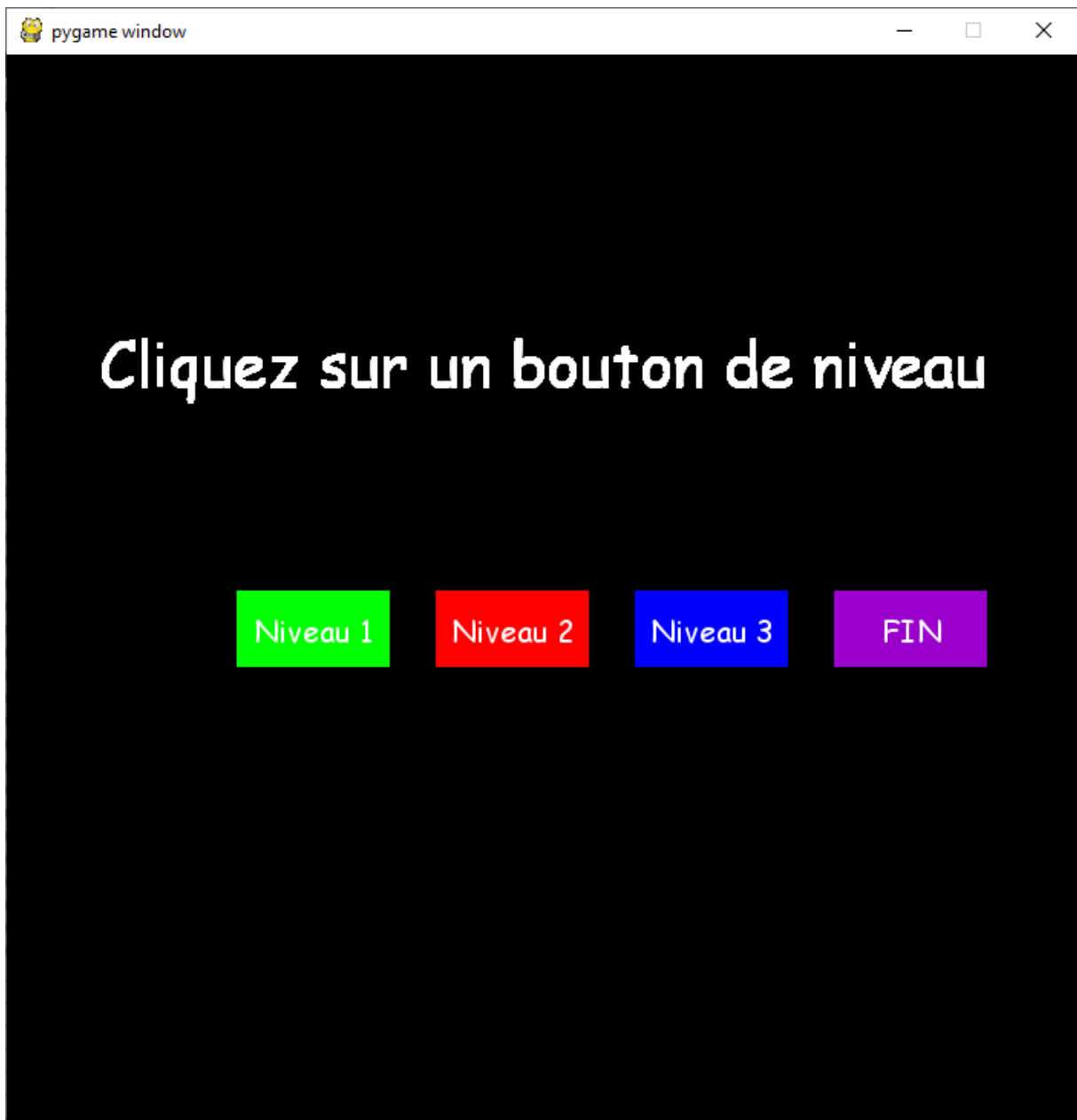
`COULEUR_ORDI = ROUGE`

`COULEUR_HUMAIN = JAUNE`

COULEUR_GAGNANT = VERT

Au lancement du jeu nous affichons la même fenêtre d'instruction que pour le jeu de tetris.

Ici les niveaux de jeu définissent la profondeur de recherche de l'algorithme Min_Max.



Les fonctions remplacées :

Les fonctions trouver_Ligne_Vide et lacher_jeton sont remplacées par une seule fonction Applique_Coup qui joue la colonne indiquée en paramètre. Elle fait appel à une fonction Cases_libres_colonnes qui lui indique le nombre de cases libres dans la colonne choisie.

L'intelligence artificielle

Pour trouver la colonne à jouer par l'ordinateur, j'ai traduit en python l'algorithme Min_Max écrit en Pascal par Fabien Torre.

Pour une présentation de l'algorithme Min_Max voir la partie 15.4 du module 15 de Didier Muller : <https://www.apprendre-en-ligne.net/pj/echechs/chapitre15.pdf>

Dans notre cas :

Pour « évaluer » une colonne, c'est-à-dire savoir si elle est bonne ou mauvaise, il faut lui attribuer une valeur. Chaque colonne est analysée selon plusieurs critères :

- Elle mène à la victoire
- Elle fait gagner l'adversaire
- Elle maximise les chances de gagner

Une note est calculée en fonction des jetons présents sur le plateau.

Lorsque c'est à l'ordinateur de jouer. Il génère la liste de tous les coups possibles, c'est-à-dire toutes les colonnes jouables.

- Il joue une colonne.
- Il attribue une note à cette colonne.
- Il se met à la place de l'adversaire et il fait exactement la même chose en attribuant une note à l'adversaire, et ainsi de suite, jusqu'à une certaine profondeur

Le programme va donc élaborer un arbre de recherche.

Il descend dans l'arbre jusqu'à la profondeur indiquée en donnant la valeur 10000 si le coup est gagnant, - 10000 si c'est l'adversaire qui gagne.

Il « remonte » ensuite ces valeurs vers le haut de l'arbre.

Lorsqu'on arrive en haut de l'arbre (profondeur = 0) le programme évalue l'intérêt globale de la colonne en fonction de l'intérêt qu'a cette colonne pour l'ordinateur et pour le joueur.

Déroulement de la partie

Au niveau 1, la profondeur de l'algorithme Min_Max est fixée à 1.

Comme dans ce cas l'ordinateur ne joue pas très bien, on le fait jouer en premier, dans la colonne 3, qui est la meilleure case pour démarrer.

Pour les autres niveaux, la profondeur est la même que le numéro des niveaux et le programme tire au sort celui qui commence.

Il arrive parfois que le programme s'arrête sans que j'aie pu en trouver la cause. Peut être ai-je cliqué trop vite lorsque c'est mon tour de jouer ?

En tout état de cause, ne pas cliquer trop vite après le jeu de l'ordinateur, attendre que le jeton du joueur se déplace bien, avant de cliquer dans une colonne.