



LES ETAPES POUR DEVENIR JOUEUR PRO AU "JEU DE NIM"

A BASIC GUIDE FOR BEGINNERS



SOMMAIRE



PRESENTATION



DIFFERENTS MODES



PROGRAMMATION INITIALE



STRATEGIE GAGNANTE



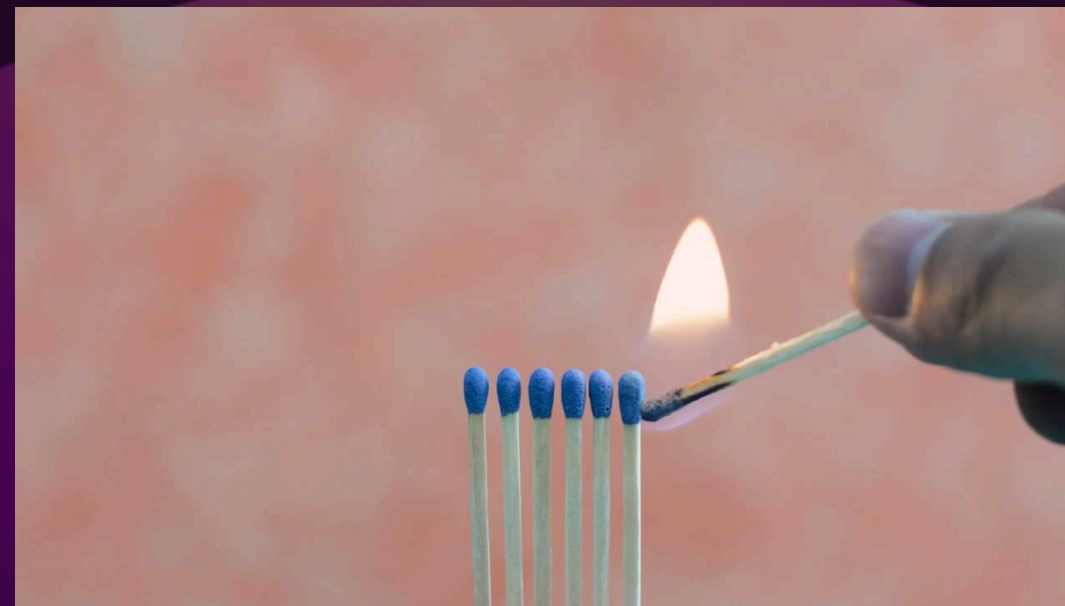
MINIMAX



CONCLUSION



JEU DE NIM




JEU DE STRATEGIE




PROGRAMMATION INITIALE

- Différentes fonctions pour chaque action du programme
- input pour le choix du joueur et le nombre d'alumette
- Choix aléatoire (entre 1,2 et 3) pour le choix de l'ordinateur
- Menu pour gérer les 3 modes différents

```
def ordi_aleatoire():  
    import random  
  
    batonnets = [1] * 13  
  
    def afficher_batonnets():  
        print("Il reste :", len(batonnets))  
  
    def jeu_termine():  
        return sum(batonnets) == 0  
  
    def choix_joueur():  
        choix = 0  
        while choix < 1 or choix > 3 or choix > sum(batonnets):  
            choix = int(input("Combien de batonnets voulez-vous retirer ? (1, 2 ou 3) "))  
        return choix  
  
    def choix_ordi():  
        choix = random.randint(1, 3)  
        while choix > sum(batonnets):  
            choix = random.randint(1, 3)  
        return choix  
  
    def retirer_batonnets(choix):  
        for i in range(choix):  
            batonnets.pop()
```



```
def jouer_joueur():  
    print("~~~~~")  
    print("C'est votre tour.")  
    afficher_batonnets()  
    choix = choix_joueur()  
    retirer_batonnets(choix)  
    print("Vous avez retiré", choix, "batonnets.")  
    print("~~~~~")  
    print("\n")  
  
def jouer_ordi():  
    print("~~~~~")  
    print("C'est le tour de l'ordinateur.")  
    afficher_batonnets()  
    choix = choix_ordi()  
    retirer_batonnets(choix)  
    print("L'ordinateur a retiré", choix, "batonnets.")  
    print("~~~~~")  
    print("\n")
```





STRATEGIE GAGNANTE

- Stratégie gagnante pour l'ordinateur

```
def choix_ordi(choix_joueur):  
    if choix_joueur == 1:  
        return 3  
    elif choix_joueur == 2:  
        return 2  
    elif choix_joueur == 3:  
        return 1
```



MINIMAX

```
def minimax1():
    def generer_arbre(etat):
        if etat == 0:
            return -1
        valeurs = []
        for coup in range(1, min(4, etat + 1)):
            valeur = generer_arbre(etat - coup) * -1
            valeurs.append(valeur)
        return max(valeurs)

    def jouer_partie():
        etat = int(input("Avec combien d'allumettes voulez-
vous commencer le jeu ? \n"))

    print("\n~~~~~")
    print("État initial du jeu:", etat)
```

```
while True:

    print("~~~~~")
    coup_joueur = int(input("Combien d'allumettes
retirez-vous (1-3) ? "))
    etat -= coup_joueur
    print("État actuel du jeu:", etat)
    if etat == 0:
        print("L'IA a gagné !")
        break

    print("~~~~~")
    print("L'IA choisit son mouvement...")
    meilleur_coup = max([coup for coup in range(1,
min(4, etat + 1))],
key=lambda coup: generer_arbre(etat - coup))
    etat -= meilleur_coup
    print("L'IA retire", meilleur_coup,
"allumettes.")
    print("État actuel du jeu:", etat)
    if etat == 0:
        print("Vous avez gagné !")
        break

    jouer_partie()
```



THANK YOU

WE HOPE YOU LEARNED SOMETHING NEW!

