

# Projet : Puissance 4 Contre la Machine

## Contre-rendu 3

Lors de cette séance, en modifiant quelques paramètres, j'ai réussi à faire une version qui se rapproche presque du résultat voulu, à la seule différence que le score qu'il renvoie diffère systématiquement de 1 ou 2 points, ce qui va changer le coup joué par l'ordinateur légèrement. J'ai gardé cette version au cas-où, mais je vais tout de même essayer de l'améliorer. Voici la version en question :

```
histo = []
columns = [0,0,0,0,0,0,0]

def canPlay(column):
    if columns[column] > 6:
        return False
    return True

def play(column):
    if canPlay(column):
        columns[column] += 1
        histo.append([columns[column],column])

def playlist(plays):
    for i in range(len(plays)):
        columns[int(plays[i]) - 1] += 1
        histo.append([columns[int(plays[i]) - 1], int(plays[i]) - 1])

def isWinningMove(column):
    if canPlay(column):
        histoTempo = histo
        histoTempo.append([columns[column],column])
        columnsTempo = columns
        columnsTempo[column - 1] += 1

        histoTempoOrdi = []
        for i in range(len(histoTempo) // 2):
            histoTempoOrdi.append(histoTempo[2*i])
        pos = [columnsTempo[column - 1],column]
        if checkLine(pos, histoTempoOrdi) or checkColumn(pos, histoTempoOrdi) or checkDiago(pos, histoTempoOrdi):
            return True
        return False

def nbMoves(histo):
    return len(histo)

def checkLine(pos, histo): #Prend en paramètre la position du jeton testé et l'historique des coups du joueur concerné uniquement
    line = pos[0]
    column = pos[1]
    if [line, column + 1] in histo and [line, column + 2] in histo and [line, column + 3] in histo:
        return True
    elif [line, column - 1] in histo and [line, column - 2] in histo and [line, column - 3] in histo:
        return True
    return False

def checkColumn(pos, histo): #Prend en paramètre la position du jeton testé et l'historique des coups du joueur concerné uniquement
    line = pos[0]
    column = pos[1]
    if [line - 1, column] in histo and [line - 2, column] in histo and [line - 3, column] in histo:
        return True
    return False

def checkDiago(pos, histo): #Prend en paramètre la position du jeton testé et l'historique des coups du joueur concerné uniquement
    line = pos[0]
    column = pos[1]
    if [line + 1, column - 1] in histo and [line + 2, column - 2] in histo and [line + 3, column - 3] in histo:
        return True
    elif [line - 1, column - 1] in histo and [line - 2, column - 2] in histo and [line - 3, column - 3] in histo:
        return True
    return False

alpha = -42
beta = 42
```

```

def negaMax(histo, columns, alpha, beta):
    #Renvoie le score de chaque plateau possible
    print(histo)
    histoTempo = []
    columnsTempo = []
    for i in range(len(histo)):
        histoTempo.append(histo[i])
    for i in range(len(columns)):
        columnsTempo.append(columns[i])
    if nbMoves(histo) >= 42:
        #Vérifie si le plateau est un ex aequo
        return 0

    for i in range(0,7):
        #Vérifie si l'ordi peut gagner au prochain tour
        if canPlay(i) and isWinningMove(i):
            return (48 - nbMoves(histo))//2
        #Initialise le pire score possible pour calculer le meilleur
    maxi = (48 - nbMoves(histo))//2
    score = (48 - nbMoves(histo))//2
    if beta > maxi:
        beta = maxi
        if alpha >= beta:
            return beta

    for i in range(0,7):
        score = -50
        if canPlay(i):
            columnsTempo[i] += 1
            #On simule qu'on joue dans la colonne i
            histoTempo.append([columns[i],i])
            score = max(score, -negaMax(histoTempo, columnsTempo, -beta, -alpha))
            #A chaque fois qu'on change de joueur, le score change de signe
            if score >= beta:
                return score
            alpha = max(alpha, score)

    return score

```

En modifiant encore quelques paramètres dans cette version, je me suis cependant rendu compte d'un gros problème : avec la manière dont Python est faite, lorsque j'affecte une nouvelle valeur à `histoTempo` ou `ColumnsTempo`, cela va aussi modifier les valeurs de `histo` et `columns` ce qui pourrait potentiellement empêcher la récursivité de s'effectuer correctement et de tester tous les cas possibles.

J'ai jusqu'à la 4<sup>e</sup> séance pour finaliser l'algorithme : je vais donc essayer chez moi de le faire fonctionner au moins en Python puis de le recréer dans l'IDE Arduino.