In [45]:

```python
from easyAI import TwoPlayersGame
from easyAI.Player import Human_Player
from neo4j import GraphDatabase
from tkinter import messagebox, ttk
from tkinter import *
import tkinter
import tkinter as tk
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('ggplot')
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
```

In [46]:

```
#CLASE PAR CREAR NODO CENTRAR-PARQUE CENTRAL
class CLASE_NEO4J(object):
    def __init__(self):
        self._driver = GraphDatabase.driver("bolt:neo4j://localhost:7687", auth=("neo4j
    def close(self):
        self._driver.close()
    def costoA(self,origen,destino):
        with self._driver.session() as session:
            greeting = session.write_transaction(self._EJECUTAR_A,origen,destino)
            print(greeting)

    def KNN(self,buscar):
        with self._driver.session() as session:
            greeting = session.write_transaction(self._EJECUTAR_KNN,buscar)
            print(greeting)
    #METODO USAR EL ALGORITMO DE RECOMENDACION POR SIMILITUD KNN en base a la cantidad

    @staticmethod
    def _EJECUTAR_KNN(tx,buscar):
        #SE GENERA EL GRAFO CON LOS NODOS PARA EL ALGORITMO DE KNN
        result1 = tx.run("CALL gds.graph.create('graficoknn12',{ "+buscar+": {label: ''

        #SE EJECUTA EL ALGORITMO KNN
        result = tx.run("CALL gds.beta.knn.stream('graficoknn12', {    topK: 1,    node
        lista =[]
        for io in result:
            print(io)
            var = str(io.get("Lugares1"))+"   :   "+str(io.get("Lugares2"))+"   :   "+s
            lista.append(var)
        combo["values"]=lista

    @staticmethod
    def _EJECUTAR_A(tx,origen,destino):
        #SE EJECUTA EL ALGORITMO A *
#           result3 = tx.run("MATCH (start:Lugares {nombre:'HOTEL_MONTECARLO'}),
        result3 = tx.run("MATCH (start:Lugares {nombre:'"+origen+"'}), (end:Lugares {no
        lista =[]
        for io in result3:
            print(io)
            var = str(io.get("lugares"))+"   :   "+str(io.get("cost"))
            lista.append(var)
        comboco["values"]=lista

# CALL gds.graph.drop('graficoknn12')
# MATCH (n) OPTIONAL MATCH (n)-[r]-() DELETE n,r
#SE INICIALIZA LA CLASE DE LOS METODOS DE NEO4J
grafo=CLASE_NEO4J()
```

In [ ]:

```
#<<<<<<<<<<<------------------EASYIA------------->>>>>>>>>>>

def lanzardd():
        print("SE EJECUTA EL LANZAR")
class TicTacToe(TwoPlayersGame):
    """ The board positions are numbered as follows:
            7 8 9
            4 5 6
            1 2 3
    """

    def __init__(self, players):
        self.players = players
        self.board = [0 for i in range(9)]
        self.nplayer = 0 # player 1 starts.


    def possible_moves(self):
        return [i+1 for i,e in enumerate(self.board) if e==0]

    def make_move(self, move):
        self.board[int(move)-1] = self.nplayer

    def unmake_move(self, move): # optional method (speeds up the AI)
        self.board[int(move)-1] = 0

    def lose(self):
        """ Has the opponent "three in line ?" """
        return any( [all([(self.board[c-1]== self.nopponent)
                      for c in line])
                      for line in [[1,2,3],[4,5,6],[7,8,9], # horiz.
                                   [1,4,7],[2,5,8],[3,6,9], # vertical
                                   [1,5,9],[3,5,7]]]) # diagonal

    def is_over(self):
        return (self.possible_moves() == []) or self.lose()

    def show(self):
        lanzardd()
        print ('\n'+'\n'.join([
                        ' '.join([['.','O','X'][self.board[3*j+i]]
                        for i in range(3)])
                 for j in range(3)]) )
        print(self.scoring())

    def scoring(self):
        return -100 if self.lose() else 0


if __name__ == "__main__":

    from easyAI import AI_Player, Negamax
    ai_algo = Negamax(6)
    TicTacToe( [Human_Player(),AI_Player(ai_algo)]).play()
```

```
SE EJECUTA EL LANZAR

. . .
. . .
. . .
0

Move #1: player 2 plays 1 :
SE EJECUTA EL LANZAR

X . .
. . .
. . .
0
```

In [ ]:

```
import tkinter as tk
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('ggplot')
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg


manejo=0.0
edad=0
resultados =([])
resultadosm =([])

hotel_palabras = ["dormir","vacaciones","viajar"]
restaurante_palabras = ["comer","beber"]
def listar_pelis():
    st = str(combo.get()).split(" : ")
    #PERSONA
    print("VALOR DE PERSONAS>> ",str(st[0]))
    per = str(st[0])
    grafo.VISTAS(str(per).replace(" ",""))
    tkinter.messagebox.showinfo(title="SE PRESIONA", message="Se Listan las peliculas
def costo():
    ori=str(orig.get())
    des=str(dest.get())
    grafo.costoA(ori,des)
    tkinter.messagebox.showinfo(title="ALGORITMO COSTO", message="Algoritmo COSTO,reco

def  validar():
    palabra = pala.get()
    print("valor de a es > ",palabra)
    if palabra !="":
        cont =0
        for hp in hotel_palabras:
            if str(hp).replace(" ","")==str(palabra).replace(" ",""):
                #SE EJECUTA LA RECOMENDACION CON NE4OJ
                print("SE EJECUTA EL KNN PARA HOTELES",palabra,hp)
                palabr = "Hoteles"
                grafo.KNN(palabr)
                tkinter.messagebox.showinfo(title="SE EJECUTARA KNN SIMULITUD", messag

                break
            elif str(restaurante_palabras[cont]).replace(" ","")== str(palabra).replac
                #SE EJECUTA LA RECOMENDACION CON NE4OJ
                print("SE EJECUTA EL KNN PARA RESTAURANTES",palabra)
                palabra = "Restaurantes"
                grafo.KNN(palabra)
                tkinter.messagebox.showinfo(title="SE EJECUTARA KNN SIMULITUD", messag
                break
            cont =cont +1

def ver_peli():
    st = str(combo.get()).split(" : ")
    #PERSONA
    per = str(st[0])
    st2 = str(peli.get())
    #PELICULA
    pel = st2
    grafo.VER_PELICULA(str(per).replace(" ",""),str(pel).replace(" ",""))
    tkinter.messagebox.showinfo(title="SE PRESIONA", message="Esta viendo la pelicula
 #--- Raiz ---
root = tk.Tk()
root.geometry('940x450')
root.title("Inteligencia Artificial <--> Sistema Recomendador Lugares")
#------------

#-- Frames ---
left_frame = tk.Frame(root)
left_frame.place(relx=0.03, rely=0.05, relwidth=0.25, relheight=0.9)

right_frame = tk.Frame(root, bg='#C0C0C0', bd=1.5)
right_frame.place(relx=0.3, rely=0.05, relwidth=0.65, relheight=0.9)
#---------------
#LABEL--------------

label = Label(left_frame, text="INGRESE PALABRA", relief=RAISED )
label.place(relx=0.05, rely=0.35,relheight=0.03, relwidth=1)

lbl1 = Label(left_frame, text="Algoritmo de Costo --> Ingrese 'Origen' y Destino", rel
lbl1.place(relx=0.05, rely=0.75,relheight=0.03, relwidth=1)
#-------------
#--- Botones ---

B1 =tk.Button(left_frame,text="BUSCAR",background = "orange",command = validar)
B1.place(relx=0.05, rely=0.45,relheight=0.06, relwidth=1)

B1 =tk.Button(left_frame,text="APLICAR ALGORITMO COSTO",foreground = "white",backgroun
B1.place(relx=0.05, rely=0.90,relheight=0.06, relwidth=1)

#----COMBOBOX---------
combo = ttk.Combobox(left_frame, state="readonly")
combo.place(relx=0.05, rely=0.60,relheight=0.03, relwidth=1)

comboco = ttk.Combobox(left_frame, state="readonly")
comboco.place(relx=0.05, rely=0.95,relheight=0.03, relwidth=1)


#-------ENTRY----TEXT
pala = Entry(left_frame, bd =5)
pala.place(relx=0.05, rely=0.38,relheight=0.03, relwidth=1)

orig = Entry(left_frame, bd =5)
orig.place(relx=0.05, rely=0.80,relheight=0.03, relwidth=1)

dest = Entry(left_frame, bd =5)
dest.place(relx=0.05, rely=0.85,relheight=0.03, relwidth=1)

#----------------------COMPONENTE DEL JUEGO EL DERECHA FRAME----------

#------------------LABELS
lbltit = Label(right_frame, text="JUEGO DE TIC-TAC-TOUR-19", relief=RAISED )
lbltit.place(relx=0, rely=0,relheight=0.03, relwidth=1)

lblusu = Label(right_frame, text="Usuario",foreground = "red", relief=RAISED )
lblusu.place(relx=0, rely=0.03,relheight=0.03, relwidth=0.5)

lbluia= Label(right_frame, text="IA usuario",foreground = "blue", relief=RAISED )
lbluia.place(relx=0, rely=0.06,relheight=0.03, relwidth=0.5)


punt= Label(right_frame, text="PUNTAJE USUARIO", foreground = "green",relief=RAISED )
```

```
punt.place(relx=0, rely=0.09,relheight=0.03, relwidth=0.5)

#--- Botones ---

lanza =tk.Button(right_frame,text="LANZAR DADO",foreground = "white",background = "blu
lanza.place(relx=0.5, rely=0.12,relheight=0.06, relwidth=0.5)


#-----------------ENTRYS TEXTO
usun = Entry(right_frame, bd =5)
usun.place(relx=0.5, rely=0.03,relheight=0.03, relwidth=0.5)

usuia = Entry(right_frame, bd =5)
usuia.place(relx=0.5, rely=0.06,relheight=0.03, relwidth=0.5)

puntt = Entry(right_frame, bd =5)
puntt.place(relx=0.5, rely=0.09,relheight=0.03, relwidth=0.5)

#-----------FRAME-JUEGO---->

juego = tk.Frame(right_frame, bg='brown', bd=5)
juego.place(relx=0, rely=0.2, relheight=1,relwidth=1)

#---TABLA DE LOS NODOS SEGUN SU NUMERO
#[0][i]
p0 = Entry(juego, bd =5)
p0.place(relx=0, rely=0.03,relheight=0.03, relwidth=0.2)
p01 = Entry(juego, bd =5)
p01.place(relx=0.2, rely=0.03,relheight=0.03, relwidth=0.2)
p02 = Entry(juego, bd =5)
p02.place(relx=0.4, rely=0.03,relheight=0.03, relwidth=0.2)
p03 = Entry(juego, bd =5)
p03.place(relx=0.6, rely=0.03,relheight=0.03, relwidth=0.2)
p04 = Entry(juego, bd =5)
p04.place(relx=0.8, rely=0.03,relheight=0.03, relwidth=0.2)
#[1][i]
p1 = Entry(juego, bd =5)
p1.place(relx=0, rely=0.06,relheight=0.03, relwidth=0.2)
p11 = Entry(juego, bd =5)
p11.place(relx=0.2, rely=0.06,relheight=0.03, relwidth=0.2)
p12 = Entry(juego, bd =5)
p12.place(relx=0.4, rely=0.06,relheight=0.03, relwidth=0.2)
p13 = Entry(juego, bd =5)
p13.place(relx=0.6, rely=0.06,relheight=0.03, relwidth=0.2)
p14 = Entry(juego, bd =5)
p14.place(relx=0.8, rely=0.06,relheight=0.03, relwidth=0.2)
#[2][i]
p2 = Entry(juego, bd =5)
p2.place(relx=0, rely=0.09,relheight=0.03, relwidth=0.2)
p21 = Entry(juego, bd =5)
p21.place(relx=0.2, rely=0.09,relheight=0.03, relwidth=0.2)
p22 = Entry(juego, bd =5)
p22.place(relx=0.4, rely=0.09,relheight=0.03, relwidth=0.2)
p23 = Entry(juego, bd =5)
p23.place(relx=0.6, rely=0.09,relheight=0.03, relwidth=0.2)
p24 = Entry(juego, bd =5)
p24.place(relx=0.8, rely=0.09,relheight=0.03, relwidth=0.2)
#[3][i]
p2 = Entry(juego, bd =5)
p2.place(relx=0, rely=0.12,relheight=0.03, relwidth=0.2)
p21 = Entry(juego, bd =5)
```

```
p21.place(relx=0.2, rely=0.12,relheight=0.03, relwidth=0.2)
p22 = Entry(juego, bd =5)
p22.place(relx=0.4, rely=0.12,relheight=0.03, relwidth=0.2)
p23 = Entry(juego, bd =5)
p23.place(relx=0.6, rely=0.12,relheight=0.03, relwidth=0.2)
p24 = Entry(juego, bd =5)
p24.place(relx=0.8, rely=0.12,relheight=0.03, relwidth=0.2)
#[4][i]
p2 = Entry(juego, bd =5)
p2.place(relx=0, rely=0.15,relheight=0.03, relwidth=0.2)
p21 = Entry(juego, bd =5)
p21.place(relx=0.2, rely=0.15,relheight=0.03, relwidth=0.2)
p22 = Entry(juego, bd =5)
p22.place(relx=0.4, rely=0.15,relheight=0.03, relwidth=0.2)
p23 = Entry(juego, bd =5)
p23.place(relx=0.6, rely=0.15,relheight=0.03, relwidth=0.2)
p24 = Entry(juego, bd =5)
p24.place(relx=0.8, rely=0.15,relheight=0.03, relwidth=0.2)
#-----------------
root.mainloop()
```

In [ ]:

```

```

In [ ]:

```

```