

ANALISTA
PROGRAMADOR



Evaluación Sumativa III

Juego Gato

| | |
|-------------|--------------------------------|
| NOMBRE: | GERSON CORDERO HERRERA |
| CARRERA: | ANALISTA PROGRAMADOR (217) |
| ASIGNATURA: | INTRODUCCION A LA PROGRAMACION |
| PROFESOR: | JAVIER MILES AVELLO |
| FECHA: | 21-05-2023 |

Contenido

| | |
|--------------------------------------|--------------------------------------|
| Evaluación Sumativa III | 1 |
| Juego Gato | 1 |
| 1 Introducción | 3 |
| 2 Manual de Usuario | ¡Error! Marcador no definido. |
| 3 Conclusión | 8 |
| 4 Código..... | 4 |
| 4.1 Versión Online | 4 |
| 4.2 Código completo | 4 |

1 Introducción

El juego del gato, también conocido como tic-tac-toe o tres en raya, es un juego clásico que consiste en colocar tres símbolos iguales en una cuadrícula de 3x3, de forma horizontal, vertical o diagonal. Es un juego sencillo pero divertido, que puede ser jugado por dos personas o por una persona contra la computadora. En este trabajo, presentaremos el código utilizado para programar un juego simple del gato en el lenguaje de programación Python. Explicaremos las funciones y variables que se usan para crear la interfaz gráfica, el tablero, la lógica del juego y el detector de ganador. También mostraremos algunos ejemplos de ejecución y comentaremos las posibles mejoras y extensiones que se podrían hacer al código.

2 Código

2.1 Versión Online

En el siguiente link, hay una versión Online del código utilizado en este trabajo.

[Inacap---Programacion/Juego_Gato at main · GersOn23/Inacap---Programacion · GitHub](https://github.com/GersOn23/Inacap---Programacion/Juego_Gato)

Código completo

Este es el código completo escrito para este trabajo:

```
#mostrar matriz en formato
'''
1 2 3
4 5 6
7 8 9
'''

def Homepage ():
    print("\t-----")
    print("\t          MARCADOR          ")
    print("\t          GATASTROFE 4       ")
    print("\t          INSERT COIN        ")
    print("\t-----")

    print("          pulse cualquier tecla para iniciar")

    print("\t-----\n")

def ganador(pos_jug, act_jug):

    comb_ganar = [["0,0", "0,1", "0,2"], ["1,0", "1,1", "1,2"], ["2,0", "2,1", "2,2"], ["0,0", "1,0", "2,0"], ["0,1", "1,1", "2,1"], ["0,2", "1,2", "2,2"], ["0,0", "1,1", "2,2"], ["0,2", "1,1", "2,0"]]

    #Loop para ver si alguna combinacion ganadora ya salió
    for x in comb_ganar:
        if all(y in pos_jug[act_jug] for y in x):

            #verdadero si ya se cumplió
            return True

    #Falso si no se ha cumplido
    return False

def empate(pos_jug):
    if len(pos_jug['X']) + len(pos_jug['O']) == 9:
        return True
```

```
        return False          # AQUI VOY HAY QUE REVISAR QUE ESTO FUNCIONE Y MUESTRE EL
                                MARCADOR

def inicio():
    import os

    mimatrix=["#  ","# "," #"],["#  ","# "," #"],["#  ","# "," #"]

    print("Jugador 1")
    namejugador1 = input("Ingrese el nombre: ")
    jugador1=input('Jugador 1, Elija X or O:  ')

    print("\n")

    if(jugador1=='X'):
        jugador2='O'
    else:
        jugador2='X'

    print("Jugador 2")
    namejugador2 = input("Ingrese el nombre: ")
    print('Jugador 2, utiliza: ',jugador2 )

    jugar(jugador1, jugador2, namejugador1, namejugador2, mimatrix)

def reinicio(jugador1, jugador2, namejugador1, namejugador2):
    import os

    mimatrix=["#  ","# "," #"],["#  ","# "," #"],["#  ","# "," #"]

    jugar(jugador1, jugador2, namejugador1, namejugador2, mimatrix)

def jugar(jugador1, jugador2, namejugador1, namejugador2,mimatrix):

    import os
    opcion='99'
    Jugada=1
    print("\n")
    pos_jug = {'X':[], 'O':[]}
    act_jug=jugador1
    posicion={}
    Tablero=[]
    posicion['celda']=10
    posicion['ocupadopor']=''
    Tablero.append(posicion)
    numero=10
```

```
contador_victorias=[]

while opcion != '0':
    os.system('cls')
    for i in range(0,len(mimatrix)):
        print("\n")
        for j in range(0,len(mimatrix)):
            print(mimatrix[i][j],end="\t")

    if(Jugada==1):
        jugadoractual=namejugador1
    else:
        jugadoractual=namejugador2

    if ganador(pos_jug, jugador1):
        print("\n\n¡El jugador ", jugadoractual, " ha ganado!")

        reincio=input("presione 3 para reiniciar: ")
        if reincio=="3":
            pos_jug.clear
            celda.clear
            Tablero.clear
            print("fin")
            reinicio(jugador1, jugador2, namejugador1, namejugador2)

    if empate(pos_jug):
        print("Empate")
        print("\n")
        reincio=input("presione 3 para reiniciar: ")
        if reincio=="3":
            pos_jug.clear
            celda.clear
            reinicio(jugador1, jugador2, namejugador1, namejugador2)
    numero = int(input(f"\n\nJUGADOR ACTUAL: {jugadoractual}
({act_jug})\n\n¿Que espacio utilizara?: "))

    if numero < 1 or numero > 9:
        print("Ingrese un numero mayor a 1 y menor que 9")
        continue

    fila = (numero - 1) // 3
    columna = (numero - 1) % 3
    opcion = str(fila) + ',' + str(columna)

    print(opcion)
    if(opcion !=0):
        celda=opcion.split(',')

```

```
        for posicion in Tablero:
            if posicion['celda']==numero:
                print("¡No puede colocar aquí, está ocupado! Vuelva a
intentarlo")
                continue
            else:

                if(Jugada==1):
                    mimatrix[int(celda[0])][int(celda[1])]=jugador1
                    pos_jug[jugador1].append(opcion)

                else:
                    mimatrix[int(celda[0])][int(celda[1])]=jugador2
                    pos_jug[jugador2].append(opcion)

        if(Jugada==1):
            Jugada=2
            act_jug=jugador2
        else:
            Jugada=1
            act_jug=jugador1

Homepage()
inicio()
```

3 Conclusión

La programación del juego del gato en Python es una tarea sencilla y divertida. En este trabajo se presenta el código utilizado para programar un juego simple del gato en el lenguaje de programación Python. Se explican las funciones y variables que se usan para crear la interfaz gráfica, el tablero, la lógica del juego y el detector de ganador. También se muestran algunos ejemplos de ejecución y se comentan las posibles mejoras y extensiones que se podrían hacer al código.