

INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL
INTERDISCIPLINARIA DE INGENIERÍA
CAMPUS COAHUILA



“Proyecto: TicTacToe”

Alumna: María Elena Landeros Barrón

Curso de programación Python de básico a intermedio

05/08/2023

El juego del Tic Tac Toe, también conocido como "Gato" o "Tres en línea", es un juego de estrategia simple y popular que se juega en un tablero de 3x3 casillas. Cada jugador toma turnos para colocar su ficha (generalmente una X o una O) en una casilla vacía del tablero. El objetivo del juego es lograr tener tres fichas del mismo tipo en línea, ya sea horizontal, vertical o diagonal, antes que el oponente.

Aunque el juego del Gatito puede parecer trivial y sencillo, tiene varios beneficios y aplicaciones educativas y prácticas:

1. Aprendizaje de programación: Es una excelente manera para que principiantes en programación aprendan conceptos como el manejo de estructuras de datos (por ejemplo, listas o matrices), el uso de funciones, bucles y condicionales, y la implementación de lógica del juego.
2. Práctica en estrategia y lógica: Aunque el juego es simple, desarrollar una estrategia ganadora y anticipar las jugadas del oponente requiere habilidades de pensamiento lógico.
3. Introducción a la inteligencia artificial: El Tic Tac Toe es un juego popular para enseñar conceptos básicos de inteligencia artificial, como la implementación de algoritmos de búsqueda como el Minimax para desarrollar un jugador automático que juegue contra un ser humano.

A continuación, se explicará el código hecho en Python:

La línea de código "from collections import deque" importa la clase "deque" del módulo "collections" en Python.

"deque" es una abreviatura de "double-ended queue" (cola de doble extremo) y es una estructura de datos que permite agregar y eliminar elementos tanto desde el principio como desde el final de la cola de manera eficiente.

En el código, la clase "deque" se utiliza para implementar el mecanismo de cambio de turno en el juego del gato (Tic-Tac-Toe).

```
from collections import deque
```

```
turno = deque(["O", "X"])
tablero = [    #declaramos la variable tablero que es una multilista
              [" ", " ", " "],          #que tendrá cadenas de caracteres
              [" ", " ", " "],
              [" ", " ", " "],
            ]
```

La función `rotar_turno()` es una función que rota el turno de los jugadores en el juego del gato (Tic-Tac-Toe).

La variable `turno` es un objeto de tipo `deque` que almacena los dos símbolos de los jugadores: "O" y "X". Inicialmente, el primer jugador tiene el turno, por lo que el símbolo "O" se encuentra en la posición 0 del `deque`, y el símbolo "X" está en la posición 1.

Cuando la función `rotar_turno()` se llama, realiza una rotación en el `deque`, lo que implica que el primer elemento se mueve al final del `deque` y el segundo elemento se convierte en el primero. Esto cambia el turno del jugador.

Luego, la función devuelve el primer elemento del `deque` utilizando `turno[0]`, que corresponde al símbolo del jugador actual con el turno cambiado, y así permite alternar los turnos entre los jugadores durante el juego del gato.

```
def rotar_turno():  
    turno.rotate()  
    return turno[0]
```

```
def mostrar_tablero():  
    print("")  
    for fila in tablero:    #utilizamos un for que recorra la variable  
        #tablero, elemento por elemento  
        print(fila)        #cada elemento es una lista y la imprimirá por  
        #completo
```

La función `procesar_posicion(posicion)` es responsable de procesar la entrada del usuario, que es una cadena que representa una posición en el tablero del juego del gato (Tic-Tac-Toe).

La función toma la posición como argumento, que es una cadena con el formato "fila, columna". Por ejemplo, si el usuario ingresa "2,3", esto representa la posición de la fila 2 y la columna 3 en el tablero.

Dentro de la función, se utiliza el método `split(',')` para dividir la cadena en dos partes, separadas por la coma. Esto separa la fila y la columna en dos elementos individuales y los almacena en las variables `fila` y `columna`.

Luego, la función devuelve una lista que contiene dos elementos: `[int(fila)-1, int(columna)-1]`. Aquí, se convierte `fila` y `columna` en enteros mediante `int()` y se resta 1 a cada valor. Esto se hace para ajustar el valor a un índice de lista válido, ya que en Python, los índices comienzan en 0, mientras que las filas y columnas en el juego del gato comienzan en 1.

Por lo tanto, la función devuelve una lista con los índices válidos en la lista del tablero, que se puede usar para actualizar la posición correspondiente en el tablero cuando un jugador realiza un movimiento.

```
def procesar_posicion(posicion): #va a recibir lo que el usuario haya escrito
    fila, columna = posicion.split(',')
    return[int(fila)-1, int(columna)-1]
```

La función `posicion_correcta(posicion)` verifica si la posición proporcionada por el usuario en el juego del gato (Tic-Tac-Toe) es válida antes de actualizar el tablero con el movimiento realizado.

La función toma como argumento `posicion`, que es una lista de dos elementos: [fila, columna], representando la posición que el jugador ha elegido.

Dentro de la función, se realiza una doble verificación:

1. Se verifica que ambos elementos de la lista `posicion` están dentro del rango de 0 a 2, es decir, están dentro de las filas y columnas válidas del tablero 3x3. Esto se hace con la condición `0 <= posicion[0] <= 2 and 0 <= posicion[1] <= 2`.
2. Se verifica que la casilla correspondiente en el tablero, especificada por `tablero[posicion[0]][posicion[1]]`, esté vacía, es decir, que contenga un espacio en blanco. Si es así, la función devuelve `True`, lo que indica que la posición es válida y se puede realizar el movimiento.

Si alguna de las dos condiciones no se cumple, la función devuelve `False`, lo que significa que la posición no es válida y el jugador debe elegir otra posición para su movimiento.

```
#verificamos si cada uno de los rangos están en el rango de 0 a 2
def posicion_correcta(posicion):
    if 0 <= posicion[0] <= 2 and 0 <= posicion[1] <= 2:
        if tablero[posicion[0]][posicion[1]] == " ":
            return True
    return False
```

Función para actualizar el tablero del juego.

```
def actualizar_tablero(posicion, jugador):
    tablero[posicion[0]][posicion[1]] = jugador
```

La función `ganador(j)` es responsable de verificar si un jugador ha ganado el juego del gato (Tic-Tac-Toe) después de realizar un movimiento.

La función toma un argumento `j`, que representa el símbolo del jugador que se está comprobando si ha ganado ("O" o "X").

Dentro de la función, se verifican todas las posibles combinaciones de ganar el juego:

1. Se comprueba si alguna de las filas contiene tres símbolos iguales (j) con la condición `tablero[0] == [j,j,j] or tablero[1] == [j,j,j] or tablero[2] == [j,j,j]`.

Si alguna de estas condiciones se cumple, la función devuelve `True`, lo que indica que el jugador `j` ha ganado el juego. Si ninguna de las condiciones se cumple, la función devuelve `False`, lo que significa que el jugador no ha ganado y el juego continúa.

```
def ganador(j):
    #Se comprueba si alguna de las filas contiene tres símbolos iguales (j)
    con la condición
    if tablero[0] == [j,j,j] or tablero[1] == [j,j,j] or tablero[2] ==
[j,j,j]:
        return True
    elif tablero[0][0] == j and tablero[1][0] ==j and tablero[2][0] == j:
        return True
    elif tablero[0][1] == j and tablero[1][1] ==j and tablero[2][1] == j:
        return True
    elif tablero[0][2] == j and tablero[1][2] ==j and tablero[2][2] == j:
        return True
    elif tablero[0][0] == j and tablero[1][1] ==j and tablero[2][2] ==
j: #revisaremos diagonales
        return True
    elif tablero[0][2] == j and tablero[1][1] ==j and tablero[2][0] ==
j: #revisaremos diagonales
        return True
    return False
```

Este fragmento de código asegura que los movimientos del jugador sean válidos, actualiza el tablero en consecuencia, y verifica si hay un ganador después de cada movimiento. Si un jugador gana, el juego termina y se muestra un mensaje de felicitación al ganador. Si no hay un ganador, el juego continúa hasta que se ingrese "salir" o hasta que se llene todo el tablero y termine en empate.

```
def juego():
    mostrar_tablero()
    jugador = rotar_turno()
    while True:
        posicion = input("Juega {}. Elige una posición fila, columna de 1 a
3: ".format(jugador))
        if posicion == "salir":
```

```
        break

    try:
        posicion_1 = procesar_posicion(posicion)
    except:
        print("Error, posición {} no es válida".format(posicion))
        continue
    if posicion_correcta(posicion_1):
        print("correcta")
        actualizar_tablero(posicion_1, jugador) #actualizamos tablero
        indicando la posicion que se puso y el jugador que lo hizo
        mostrar_tablero()

        if ganador(jugador): #funcion para el ganador
            print("Jugador de {} es el ganador!!".format(jugador))
            break
        jugador = rotar_turno()

    else:
        print("Posición {} es incorrecta".format(posicion))

juego()
```

```

PS C:\IPN\CURSO_PYTHON> python -u "c:\IPN\CURSO_PYTHON\Gato.py"

[' ', ' ', ' ']
[' ', ' ', ' ']
[' ', ' ', ' ']
Juega X. Elige una posición fila, columna de 1 a 3: 1,1
correcta

['X', ' ', ' ']
[' ', ' ', ' ']
[' ', ' ', ' ']
Juega O. Elige una posición fila, columna de 1 a 3: 2,3
correcta

['X', ' ', ' ']
[' ', ' ', 'O']
[' ', ' ', ' ']
Juega X. Elige una posición fila, columna de 1 a 3: 1,2
correcta

['X', 'X', ' ']
[' ', ' ', 'O']
[' ', ' ', ' ']
Juega O. Elige una posición fila, columna de 1 a 3: 2,1
correcta

['X', 'X', ' ']
['O', ' ', 'O']
[' ', ' ', ' ']
Juega X. Elige una posición fila, columna de 1 a 3: 1,3
correcta

['X', 'X', 'X']
['O', ' ', 'O']
[' ', ' ', ' ']
Jugador de X es el ganador!!
PS C:\IPN\CURSO_PYTHON> █

```

Se observa en esta imagen el modo en el que puedes elegir la fila, columna y cuando hay un ganador.

En resumen, el juego del Tic Tac Toe no solo es una forma de entretenimiento, sino que también tiene valor educativo y práctico en el campo de la programación y la estrategia. Además, puede ser una excelente manera de compartir tiempo con amigos, familiares o compañeros de clase.

Liga de GitHub: [MariaLanderos/TicTacToe \(github.com\)](https://github.com/MariaLanderos/TicTacToe)