INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO
NOMBRE: Cerda García Gustavo
Materia: Inteligencia Artificial

Ejercicio de Laboratorio 6. Minimax y Poda Alfa-Beta

Ejercicios

1. Implementa tu propia versión del juego del gato (tic tac toe) empleando minimax:
    a. A partir de la implementación anterior, crea una versión que implemente la poda alfa beta.
2. Prueba tu juego con otros usuarios humanos.
3. Prueba tu juego con otro agente inteligente.
4. Diseña y de ser posible implementa una versión de tu juego con conectividad a red. *

* no es obligatorio

Código

Función `minimax (tablero, profundidad, es_maximizando)`

```python
#Nombre: minimax
#Descripción: Algoritmo Minimax para la computadora
#Entradas:
# - tablero: tablero actual del juego
# - profundidad: profundidad del árbol de búsqueda
# - es_maximizando: indica si la computadora está maximizando o minimizando
#Salidas:
# - mejor_valor: mejor valor obtenido por el algoritmo Minimax
def minimax(tablero, profundidad, es_maximizando):
    if verificarGanador("O"):
        return -10 + profundidad
    elif verificarGanador("X"):
        return 10 - profundidad
    elif esEmpate():
        return 0

    if es_maximizando:
        mejor_valor = float("-inf")
        for i in range(3):
            for j in range(3):
                if tablero[i][j] == " ":
                    tablero[i][j] = "X"
                    valor = minimax(tablero, profundidad + 1, False)
                    tablero[i][j] = " "
                    mejor_valor = max(mejor_valor, valor)
        return mejor_valor
    else:
        mejor_valor = float("inf")
        for i in range(3):
            for j in range(3):
                if tablero[i][j] == " ":
                    tablero[i][j] = "O"
                    valor = minimax(tablero, profundidad + 1, True)
                    tablero[i][j] = " "
                    mejor_valor = min(mejor_valor, valor)
        return mejor_valor
```

Este código implementa el algoritmo Minimax, una técnica de búsqueda en árboles utilizada para tomar decisiones en juegos de adversarios. Funciona considerando todas las posibles jugadas a partir del estado actual del juego y determinando la mejor jugada posible para el jugador maximizador (generalmente el jugador "X" en el juego del Tic Tac Toe) y la peor jugada posible para el jugador minimizador (generalmente el jugador "O" en el Tic Tac Toe).

- La función `minimax` evalúa todas las posibles secuencias de movimientos recursivamente.
- Si uno de los jugadores gana, asigna un valor de evaluación alto si es el jugador "X" y un valor bajo si es el jugador "O".
- Si el juego termina en empate, devuelve un valor neutro.
- Para cada posible movimiento, se simula el movimiento y se llama recursivamente a `minimax` con un nivel de profundidad incrementado.
- Dependiendo de si es el turno del jugador maximizador o minimizador, se maximiza o minimiza el valor de evaluación resultante.
- Finalmente, devuelve el mejor valor de evaluación encontrado para el jugador maximizador o el peor valor para el jugador minimizador.

Función `movimientoComputadora ()`

```python
#Nombre: movimientoComputadora
#Descripción: Realiza el movimiento de la computadora
#Salidas:
# - mejor_movimiento: mejor movimiento obtenido por el algoritmo Minimax
def movimientoComputadora():
    mejor_valor = float("-inf")
    mejor_movimiento = None

    for i in range(3):
        for j in range(3):
            if tablero[i][j] == " ":
                tablero[i][j] = "X"
                valor = minimax(tablero, 0, False)
                tablero[i][j] = " "
                if valor > mejor_valor:
                    mejor_valor = valor
                    mejor_movimiento = (i, j)

    return mejor_movimiento
```

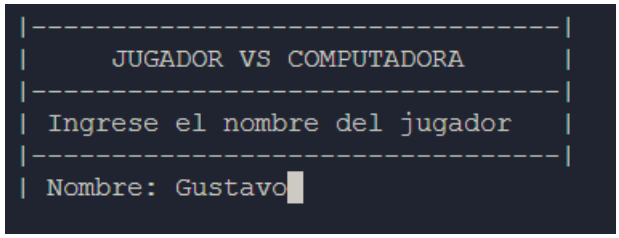
Este código implementa una función movimientoComputadora que utiliza el algoritmo Minimax para determinar el mejor movimiento para la computadora en el juego del Tic Tac Toe.

- Inicializa `mejor_valor` con un valor muy bajo (negativo infinito) y `mejor_movimiento` como `None`.
- Itera sobre todas las posiciones del tablero.
- Para cada posición vacía, simula que la computadora coloca una ficha "X" en esa posición y llama al algoritmo `minimax` para evaluar el tablero resultante.
- Luego, se deshace del movimiento simulado.
- Si el valor evaluado es mayor que el mejor valor encontrado hasta ahora (`mejor_valor`), actualiza `mejor_valor` con el nuevo valor y guarda las coordenadas de esa posición como `mejor_movimiento`.
- Finalmente, devuelve las coordenadas del mejor movimiento encontrado.

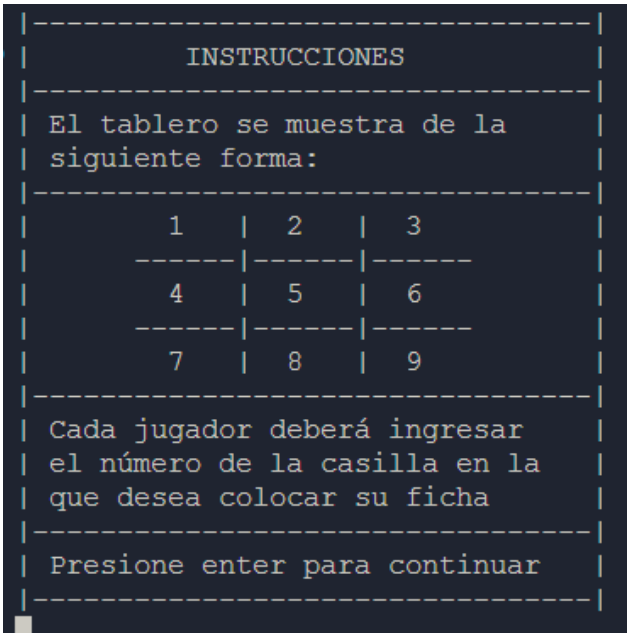
Pruebas

Humano vs Computadora

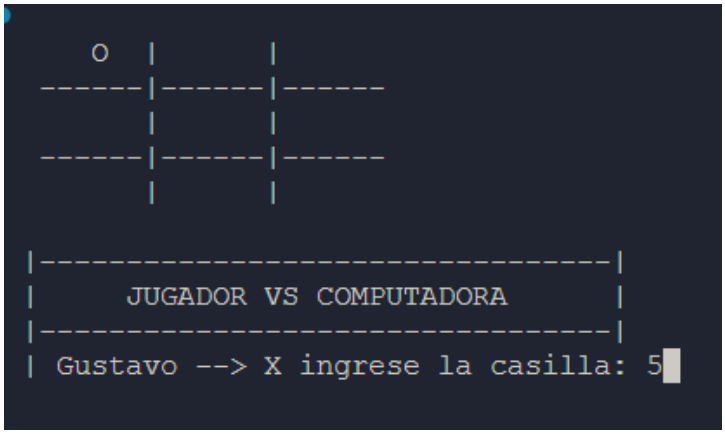
Primero ingresamos el nombre del jugador



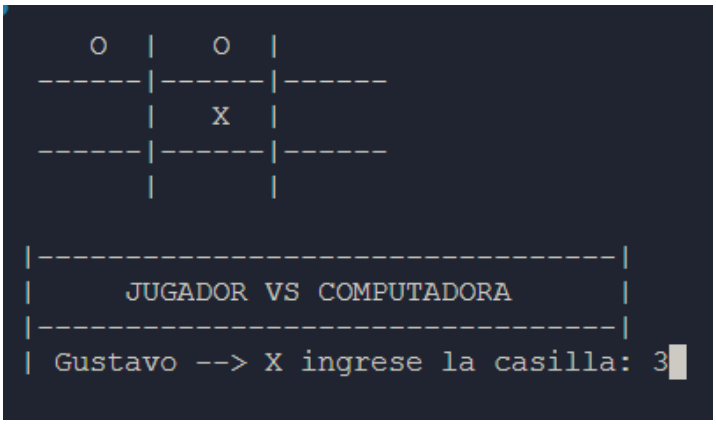
Se muestran las instrucciones



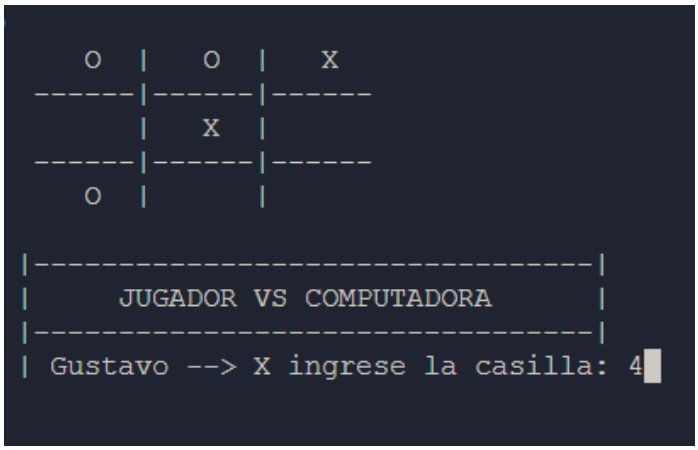
La computadora comenzó, entonces nos toca elegir una casilla que será la 5



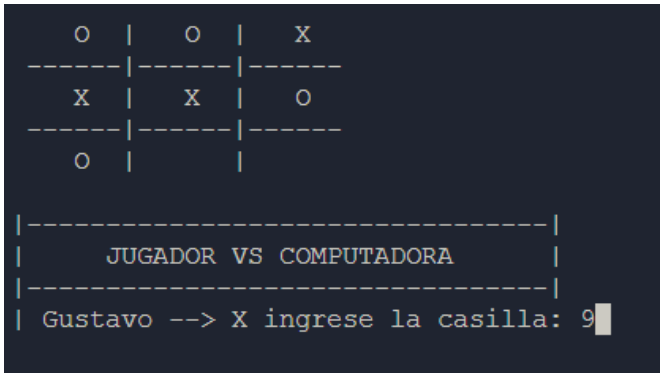
En el turno de la computadora, intenta ganar ya que no hay suficientes casillas del jugador X para bloquear la jugada, entonces la bloqueamos nosotros

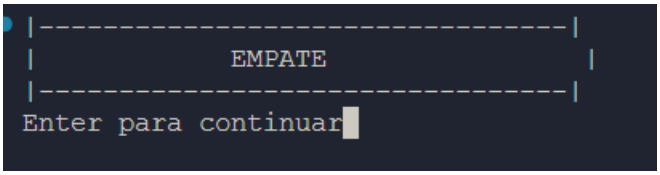


Como ahora si pudiéramos ganar la computadora nos bloquea la jugada y ahora nos vuelve a tocar bloquear su jugada, pero esta nos intentara bloquear eligiendo la casilla 6.



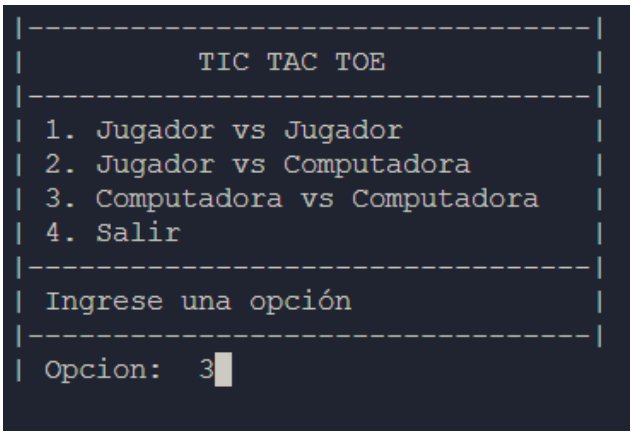
Este gato ya se ahogó, será un empate.



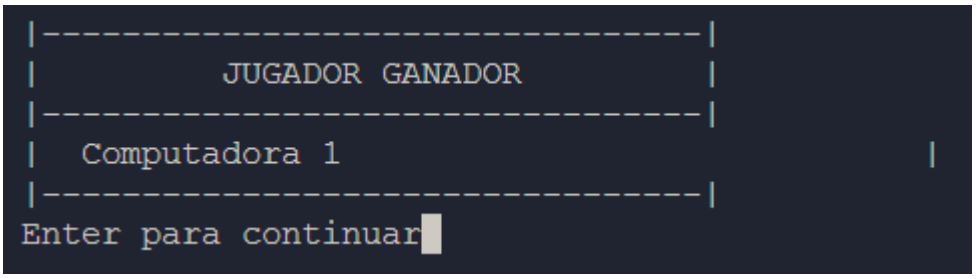


Computadora vs Computadora

Seleccionar el modo



Gano la computadora 1, aquí se llena por ticks el tablero



También hay empates