



INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE COMPUTO

“Ejercicio Práctico 3: Algoritmo minimax y poda alfa beta”

Nombre: Lizandro Hernandez Ramirez

Grupo: 6CV2

Materia: Inteligencia Artificial

Profesor: Andrés García Floriano

Introducción

En esta práctica, el objetivo era desarrollar un juego de "gato" en una versión de 4x4 casillas. A diferencia del juego 3x3, aquí se necesita alinear cuatro fichas para ganar. El reto principal no solo fue crear el juego, sino también programar al oponente, capaz de tomar decisiones lógicas para intentar ganar o empatar la partida. El proyecto se implementó en dos lenguajes, Python y C, ambos con una interfaz interactiva.

Desarrollo

Para que el juego funcionara, lo primero fue crear un tablero de 4x4 y una función que revisara constantemente, después de cada turno, si algún jugador había formado una línea de cuatro fichas, ya sea en horizontal, vertical o en diagonal. Si el tablero se llenaba sin un ganador, el juego terminaba en empate.

El principal objetivo del proyecto fue la creación de la inteligencia del oponente. Para lograr que la computadora "pensara", se utilizó el algoritmo Minimax. Este algoritmo funciona de la siguiente manera: mira hacia el futuro, simula todos los movimientos posibles que podría hacer, y para cada uno de esos movimientos, simula todas las respuestas que el jugador humano podría dar, continúa así durante varios turnos, creando un árbol de posibilidades. El objetivo de la computadora es elegir el movimiento que la lleve por el mejor camino posible, asumiendo que el jugador humano siempre escogerá la jugada que sea peor para ella.

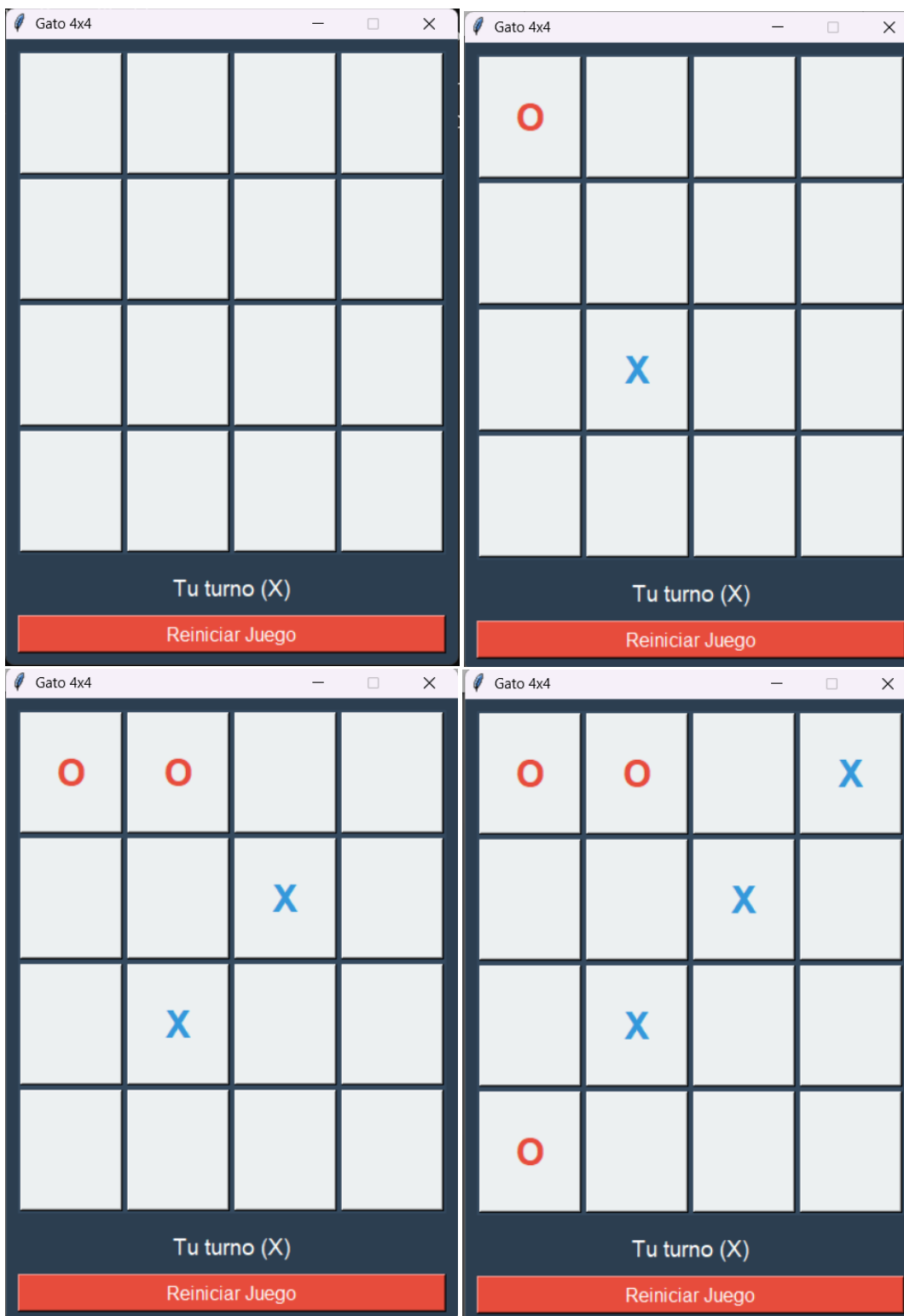
Sin embargo, en un tablero de 4x4, la cantidad de jugadas futuras aumenta en comparación de las de un tablero 3x3, y el algoritmo Minimax se haría lento ya que la computadora se quedaría "pensando" por mucho tiempo. Para solucionar esto, se implementó la poda Alfa-Beta. Esta técnica es una mejora para Minimax cuya función es "podar" o cortar ramas enteras del árbol de posibilidades que sabe que no necesita explorar. Si la computadora está analizando una jugada y se da cuenta de que esta llevará a un resultado peor que uno que ya encontró antes, entonces deja de analizar ese camino y pasa al siguiente. Esto hace que la toma de decisiones sea más rápida y eficiente sin sacrificar la calidad de la jugada.

Para la interfaz, en la versión de Python se usó la librería tkinter, que permite crear una ventana gráfica con botones en los que se puede hacer clic para colocar la ficha. En el programa de C, se optó por la librería ncurses, que crea una interfaz en la terminal. Aunque no usa un mouse, permite seleccionar las casillas con las flechas del teclado y usar colores..

Conclusión

El resultado final fue un juego de gato 4x4 funcional con un oponente de computadora que representa un desafío. El uso del algoritmo Minimax con la poda Alfa-Beta demostró ser una buena opción para darle a la máquina la capacidad de analizar el juego y tomar las mejores decisiones. Esta práctica sirvió para comprender cómo se pueden diseñar algoritmos que simulan el razonamiento lógico y cómo resolver problemas prácticos de programación, como la optimización del rendimiento y la creación de una interfaz para el usuario.

Demostración (python)



Gato 4x4

O	O	O	X
X		X	
	X		
O			

Tu turno (X)

Reiniciar Juego

Gato 4x4

O	O	O	X
X	O	X	X
	X		
O			

Tu turno (X)

Reiniciar Juego

Gato 4x4

O	O	O	X
X	O	X	X
X	X	O	
O			

Tu turno (X)

Reiniciar Juego

Gato 4x4

O	O	O	X
X	O	X	X
X	X	O	O
O			X

Tu turno (X)

Reiniciar Juego

