

Búsqueda adversaria para aprender a jugar a Otelo

Eloy Sancho Cebrero
Universidad de Sevilla
Sevilla, España
elosanceb@alum.us.es

Iván Fernández Limárquez
Universidad de Sevilla
Sevilla, España
ivaferlim@alum.us.es

Abstract—Este es el resumen del trabajo. Aquí puedes describir brevemente el objetivo, el método utilizado y los resultados más importantes. Suele ocupar unas 5–6 líneas.

Index Terms—Monte Carlo Tree Search, MCTS, Upper Confidence Bound for Trees, Othello, Otelo, Reversi, Python, Inteligencia Artificial

V. CONCLUSIONES

Resume lo aprendido, dificultades enfrentadas, mejoras posibles.

I. INTRODUCCIÓN

Describe aquí el contexto del proyecto, motivación y objetivos.

II. PRELIMINARES

Preliminares

III. IMPLEMENTACIÓN

Para la implementación del proyecto se han realizado los siguientes pasos: se ha creado una versión inicial de Otelo utilizando Pygame, se ha creado un agente básico utilizando el algoritmo de Monte Carlo Tree Search, se han generado datos a partir de ese agente básico, se ha creado y, posteriormente, entrenado una red neuronal a partir de los datos generados, se ha utilizado la red neuronal como sustituta de una de las funciones de las que hace uso la implementación inicial del algoritmo MCTS y, finalmente, se ha adaptado la versión inicial de Otelo (en la que no se podía jugar contra ningún adversario) para que el agente desarrollado pudiese jugar contra el usuario.

A continuación, desarrollaremos la implementación de cada uno de los pasos mencionados.

A. *Otelo en pygame*

El primer paso a seguir en la implementación del proyecto fue la creación del juego Otelo que nos serviría como base para crear tanto el agente MCTS como el agente final que haría uso de la red neuronal.

B. *Agente MCTS con UCT*

C. *Generación de datos*

D. *Diseño de la red neuronal*

E. *Entrenamiento de la red neuronal*

F. *Agente adversario*

IV. PRUEBAS Y EXPERIMENTACIÓN

Explica qué datos has obtenido, cómo se comporta el agente, etc. Puedes incluir tablas, gráficos, o descripciones.