

Assignment 3 - Sistemas Inteligentes

22 de Marzo, 2023

1 Enunciado

La tarea es programar Cuatela inteligente, un juego de mesa que se juega por turnos y entre dos jugadores.

	A	B	C	D
1	●			○
2		●	○	
3		○	●	
4	○			●

Figure 1: Tablero inicial de Cuatela

Algunas reglas:

- El tablero empieza con todas las piezas posicionadas como se muestra en la figura 1.
- Empiezan jugando siempre las negras.
- Cada ficha se pueden mover en 8 direcciones en horizontal, vertical o diagonal (N,S,E,O, NE, NO, SE,SO) cuando sea posible y no haya obstrucción. Siempre se mueve la cantidad máxima de lugares libres que tenga en esa dirección.
- Gana aquel jugador que acomode sus cuatro fichas en horizontal, vertical, formando un cuadrado o posicionando sus fichas en las cuatro esquinas como se muestra en la figura 2.

2 Instrucciones

- El humano tendrá que jugar el juego contra la computadora. Para esto, antes de empezar el juego, desplegar un menu que deje al humano elegir qué color de fichas quiere jugar.

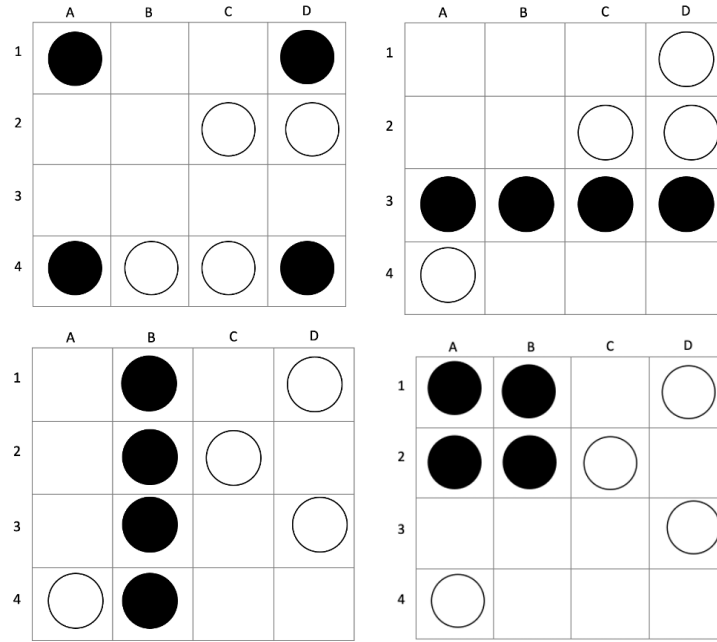


Figure 2: Tablero de Teeko

- Turno humano: el programa recibirá como entrada la posición de la ficha que desee mover y la dirección a la que desea mover. Imprimir un mensaje en caso de que el movimiento no sea válido. Ej: C2 SE
- Turno computador: el programa debe imprimir qué ficha escogió mover y hacia qué dirección.
- La computadora no debe tardar más de 10s en dar una respuesta para su siguiente movimiento.
- El tablero debe ser visible en todo momento, indicando también la cantidad de turnos jugados.
- Terminado el juego determinar el ganador, el tiempo promedio de respuesta de la computadora en responder (en ms).

3 Programa Inteligente

Para que el programa pueda tomar una decisión "inteligente" de dónde poner su ficha a continuación se debe realizar lo siguiente:

- Definir una heurística que determine el valor utilitario de un estado. Explica cuál es la lógica detrás de tu heurística.
- Definir una altura máxima para expandir el game tree que decidirá hasta donde se debe explorar dicho árbol. Experimenta con diferentes alturas y selecciona aquella cuyo tiempo de respuesta promedio sea $< 10s$.
- Implementar el algoritmo $MinMax + \alpha - \beta pruning$ y $MinMaxWithDepth(cut - off)$.

4 Experimentos

Realiza los experimentos, comenta y discute tus resultados:

- Prueba *MinMaxWithDepth(cut-off)* con 2 (o más) heurísticas diferentes, ¿cuál es mejor?
- Ejecuta el algoritmo *MinMax + $\alpha - \beta$ pruning* y reporta un gráfico #turno vs #podas. Cambia el orden de las acciones (para expandir por derecha) y reporta el mismo gráfico. ¿Es diferente? ¿Por qué?
- Ejecuta el juego una vez y reporta un gráfico #turno vs #estados expandidos para los algoritmos *MinMax + $\alpha - \beta$ pruning* y *MinMaxWithDepth(cut-off)*

5 Torneo de Máquinas Inteligentes

Se armará un torneo donde se enfrentarán el programa de un equipo con el programa del otro. El objetivo es seleccionar el programa más inteligente. El torneo se lo realizará el día lunes 3 de Abril del 2023 en donde se darán más detalles de su ejecución.

6 Sobre el Proyecto

El código será subido a un repositorio de GitHub. Se recomienda hacer commits graduales y no un solo commit con toda la práctica. Enviar solamente el link al repositorio en un archivo txt. El reporte del proyecto debe estar en el archivo README.md del proyecto. Tanto el reporte como el código deben estar en inglés.

El proyecto debe tener un README.md con las siguientes secciones:

- Descripción de la solución, donde se detalle las funciones heurísticas utilizadas.
- Experimentos realizados.
- Conclusiones.

Además, se tendrá una sección donde se indicará cómo ejecutar el proyecto (Installation Guide).

Fecha límite de entrega: Domingo 2/Abril/2023 23:59

Puntos en juego: 15

¡Éxitos!