

OpenCourseWare

Práctica 3: Reversi - Othello

Yago Sáez Achaerandio
Pedro Isasi Viñuela



1. Introducción

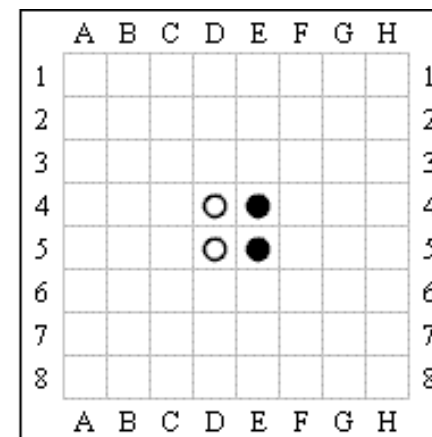
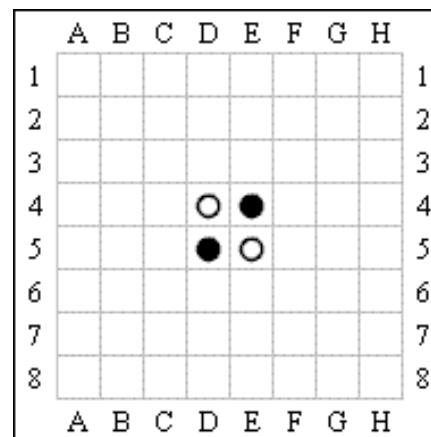
- El **othello** o **reversi** es un juego entre dos personas que comparten **64 fichas iguales** de caras distintas que se van colocando por turnos en un tablero dividido en **64 celdas**.
- Las caras de las fichas se distinguen por su color y cada jugador tiene asignado uno de esos colores ganando quien tenga más fichas sobre el tablero al finalizar la partida.
- El número máximo de posibles partidas es de aproximadamente 10^{54} .
- Por otra parte el número máximo de posiciones posibles se calcula aproximadamente en 10^{30} .

1. Introducción (II)

- Sus orígenes se sitúan en Inglaterra en el siglo XIX en donde los londinenses Lewis Waterman y John W. Mollett comercializaron en 1880 cada uno juegos con reglas parecidas.
- En 1971 el japonés Goro Hasegawa cambió dos reglas del juego y lo registró bajo el nombre de **Othello**® inspirado por la obra de Shakespeare del mismo nombre. ("[Otelu](#)" en español).
- Desde 1925 se juega un Campeonato Mundial de Othello
- El otelo crea interés entre los programadores debido a la sencillez de sus reglas: sólo un tipo de movimiento un único tipo de fichas etc. El primer torneo entre programas fue en 1979. En 1981 se celebró un torneo con jugadores humanos y programas que fue ganado por Hiroshi Inouie el entonces campeón mundial aunque perdió una partida contra el programa The Moor siendo ésta la primera vez que un programa ganaba a un campeón mundial en un juego de este tipo.
- Los programas han ido ganando en potencial terminando por ser imbatibles por ejemplo el programa Logistello en 1997 derrotó claramente al campeón mundial Murakami por 6 victorias a 0

1. Introducción (III): Reglas

- Las casillas se denotan numerando las columnas comenzando por la esquina superior izquierda del tablero con letras de la A a la H e igual con las filas pero con números del uno al ocho.
- En el otelo de inicio se colocan cuatro fichas tal como se ve en el diagrama de la izquierda: dos fichas blancas en D4 y E5 y dos negras en E4 y D5 (los diagramas de abajo muestran las dos posibles aperturas no siendo posible la segunda en el otelo).



1. Introducción (IV): Reglas

- Empezando por quien lleva las fichas negras los jugadores deben hacer un movimiento por turnos a menos que no puedan hacer ninguno pasando en ese caso el turno al jugador contrario. El movimiento consiste en colocar una ficha de forma que flanquee una o varias fichas del color contrario y dar la vuelta a esas fichas para que pasen a mostrar el propio color (diagonal u ortogonal)
- La partida finaliza cuando ningún jugador puede mover (normalmente cuando el tablero está lleno o casi de fichas) y gana quien en ese momento tenga sobre el tablero más fichas mostrando su color.

1. Introducción (V): Reglas

	A	B	C	D	E	F	G	H	
1		○	○	○	○	○	○		1
2			○	●	●	○			2
3	○	○	○	○	○	●	○		3
4	○	●	○	●	●	●	○		4
5	○	●	●	○	○	●	○		5
6	○	○	○	○	●	●	○		6
7			●	●	●	●			7
8			●	●	●	●			8
	A	B	C	D	E	F	G	H	

	A	B	C	D	E	F	G	H	
1		○	○	○	○	○	○		1
2			○	●	○	○			2
3	○	○	○	○	○	○	○		3
4	○	●	○	○	○	○	○		4
5	○	●	●	○	○	●	○		5
6	○	○	○	○	●	●	○		6
7			●	●	●	●			7
8			●	●	●	●			8
	A	B	C	D	E	F	G	H	

2. Othello

Vamos a jugar!!!

- El objetivo de la competición es proporcionar la mejor función de utilidad para el Othello. No se trata de la mejor arquitectura ni del mejor sistema de IA sino del mejor sistema de aprendizaje.
- El proceso de competición consiste en lanzar 2.000 partidas 1.000 como blancas y 1.000 como negras.
- Esta prohibido utilizar búsqueda en profundidad/árboles o utilizar bases de datos.

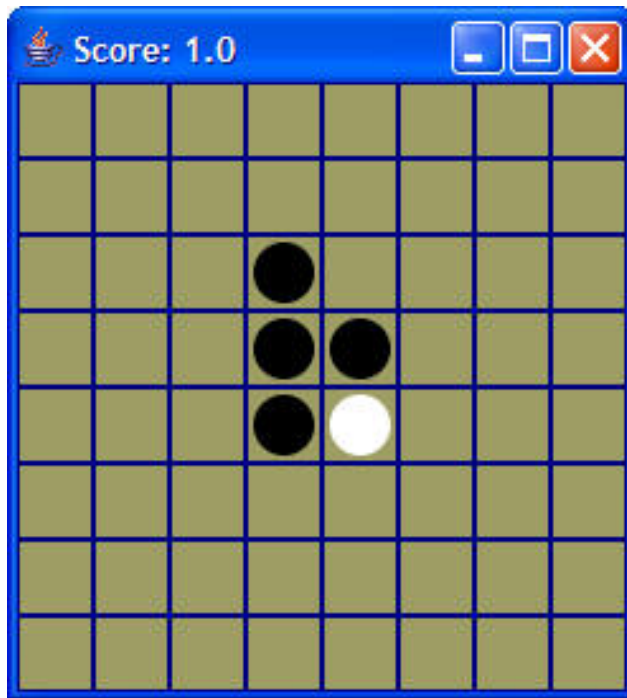
2. Othello



0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

negra: 1
vacía: 0
blanca: -1

2. Othello



negra: 1

vacía: 0

blanca: -1

Vector de salida:

```
{0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1  
0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 -1 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0}
```

2. Othello

negra: 1
vacía: 0
blanca: -1

Posible vector de soluciones: (separado por espacios)

**0.12 0.54 -0.33 -0.67 0.21 0.46 -0.74 0.1 -0.2 0.14 -0.7
-0.5 0.79 -0.7 0.21 0.25 -0.7 0.4 0.2 1 0.02 0.01 -0.07
0.4 -0.6 0.82 0.1 1 1 0.2 -0.25 0.3 0.1 0.62 0.24 1 -1
0.02 0.4 0.16 -0.68 0.5 -0.4 0.2 0.1 0.66 0.32 -0.6 0.9
0.1 0.36 0.2 0.46 0.41 -0.21 0.12 -0.4 0.55 0.18 0.23
-0.92 0.17 -0.2 0.1**

2. Othello

negra: 1

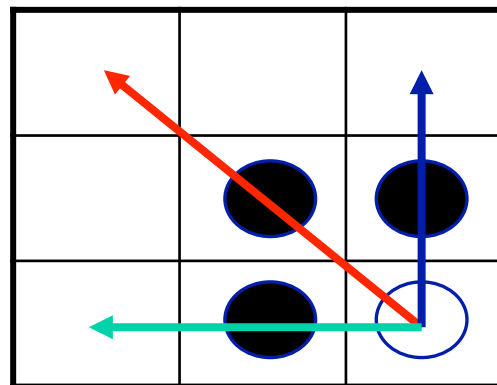
vacía: 0

blanca: -1

Ejemplo 1. juegan blancas en un supuesto tablero de 3x3:

Estrategia 1: ahora esta jugando
con blancas

-0,62	-0,03	-0,54
0,76	-0,91	0,23
0,4	0,2	0,4



Estrategia 2: ahora esta jugando
con negras

-0,12	0,13	-0,54
0,36	0,12	0,3
0,14	-0,2	1

2. Othello

negra: 1

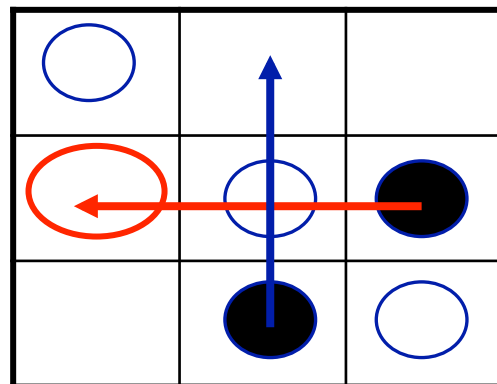
vacía: 0

blanca: -1

Ejemplo 1. juegan negras en un supuesto tablero de 3x3:

Estrategia 1: ahora esta jugando
con blancas

-0,62	-0,03	-0,54
0,76	-0,91	0,23
0,4	0,2	0,4



Estrategia 2: ahora esta jugando
con negras

-0,12	0,13	-0,54
0,36	0,12	0,3
0,14	-0,2	1

Práctica 3: Búsqueda de estrategias en Othello

La práctica consiste en buscar la estrategia más robusta posible para el juego del othello.

Para ello deberán generarse mediante técnicas de computación evolutiva arrays de números reales (-1.0 a 1.0) que se pasarán a una función de evaluación que las enfrentará devolviendo si han ganado blancas, negras o empate. Función de ejemplo:

Resultado = Evaluación (ArrayBlancas, ArrayNegras)

Resultado: puede ser: $-x$ (han ganado las blancas por x piezas)
0 (empate)
 $+x$ (han ganado las negras por x piezas)

- Las mejores estrategias lucharán entre sí (serán las más robustas ante múltiples situaciones).
- Las soluciones deberán de probarse contra una estrategia que daremos los profesores 1.000 partidas con negras y 1.000 partidas con blancas (la misma solución) para comprobar cuantas partidas ganan empatan o pierden. Eso dictaminará cuan buena es la solución.
- Se debe entregar una memoria con la descripción de la técnica utilizada, las pruebas y los resultados, el código en un disquete adjunto a la memoria y finalmente el vector resultado.