



UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

DESARROLLO DE UN JUGADOR DE OMWESO

MEDIANTE TÉCNICAS DE INTELIGENCIA ARTIFICIAL

PROYECTO FIN DE CARRERA

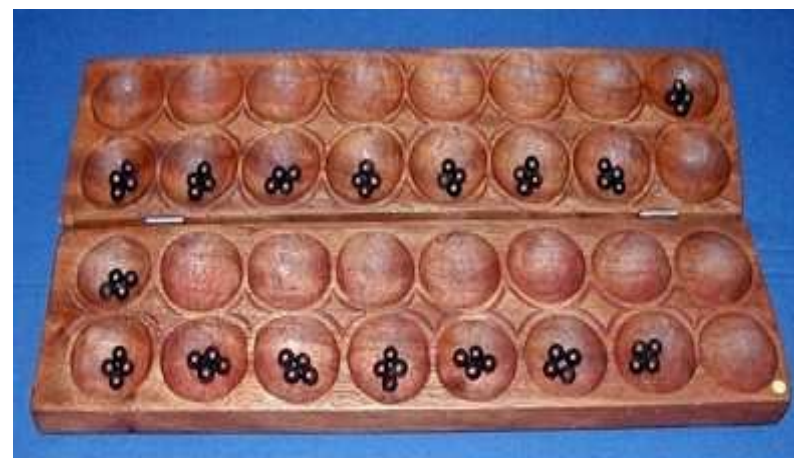
Autor: José David Torres Montiel

Tutor: Carlos Linares López

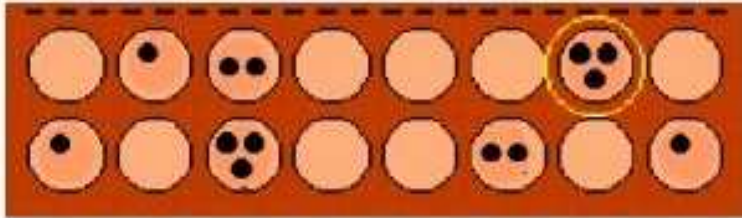
Tabla de contenidos

- Estado de la cuestión
- Objetivos
- Desarrollo
- Resultados
- Conclusiones
- Líneas futuras

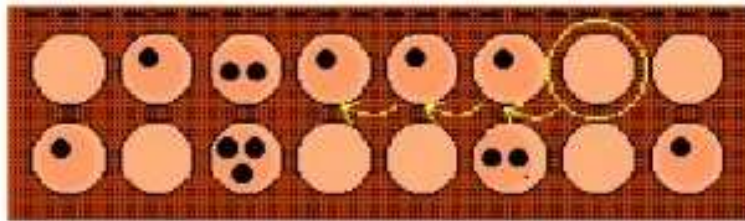
La familia Mancala. Tableros.



Movimiento de siembra



Se selecciona la casilla desde la que se iniciará el movimiento.

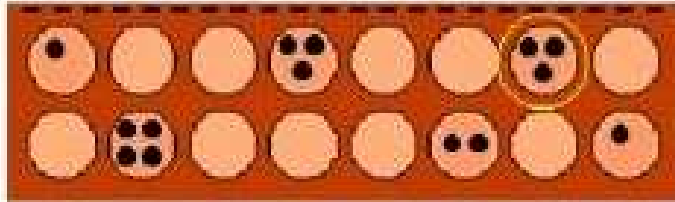


Se vacía la casilla y se “siembra” en las casillas siguientes.

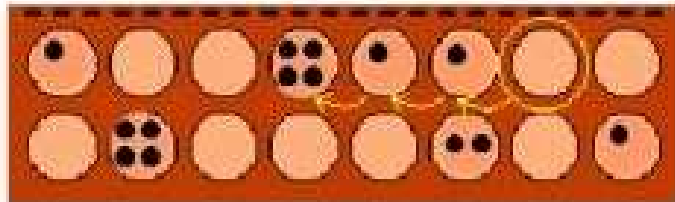
Se “siembra” soltando una semilla en cada una de las casillas siguientes, según un sentido contrario a las agujas de un reloj.

Imágenes © International Omweso Society (www.omweso.org)

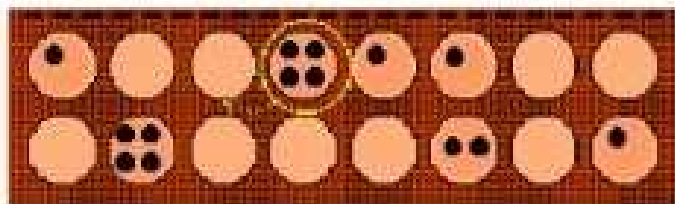
Movimiento encadenado



Si el movimiento de siembra acaba en una casilla que no está vacía ...



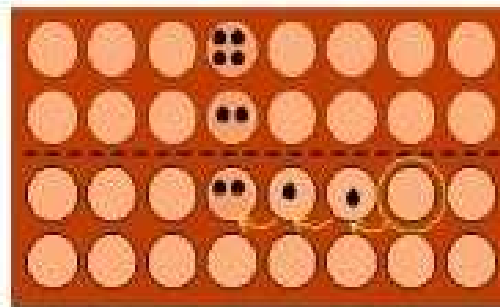
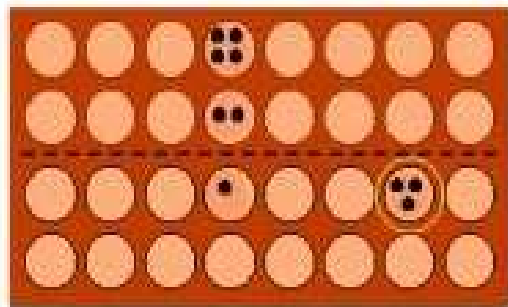
Se vuelve a coger las semillas de la casilla donde acaba la siembra ...



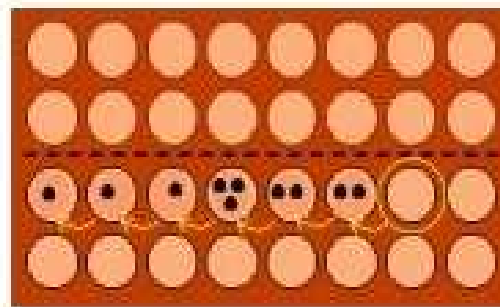
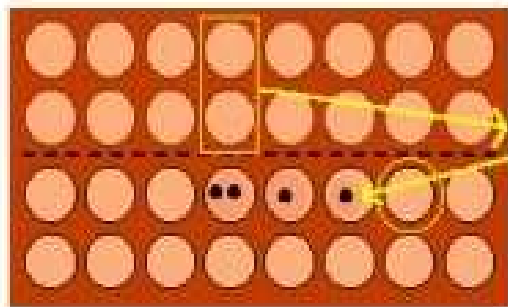
Y se inicia otro movimiento de siembra desde esa casilla.

Imágenes © International Omweso Society (www.omweso.org)

Captura. Reentrada de semillas.



Si una siembra acaba en una casilla no vacía.



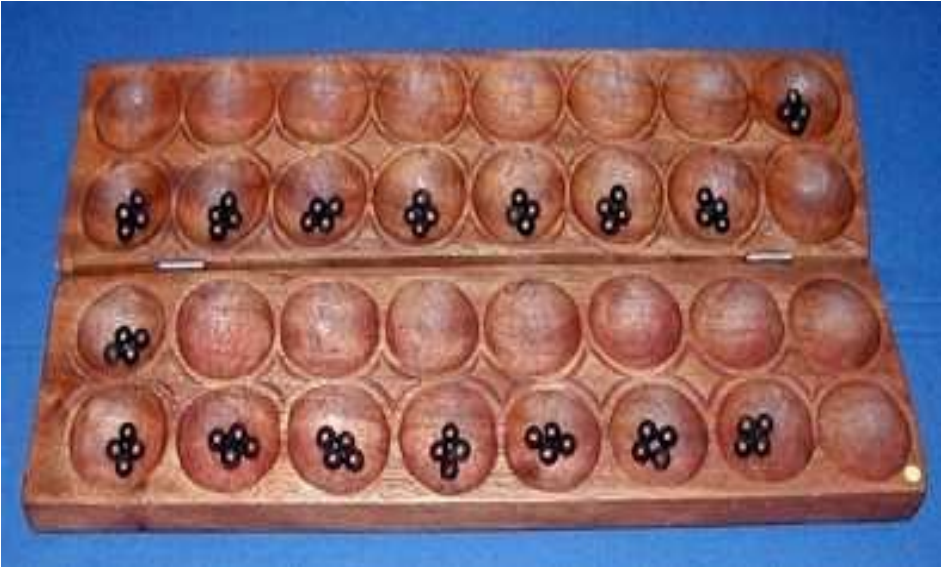
Si las dos casillas del contrario no están vacías enfrente de donde acaba.

Se realiza una captura vaciando las dos casillas del contrario implicadas, y se utiliza las semillas para iniciar una siembra desde la casilla que originó la captura. **La finalidad del juego es dejar sin movimientos al contrario.**

Efecto del movimiento de siembra.

Imágenes © International Omweso Society (www.omweso.org)

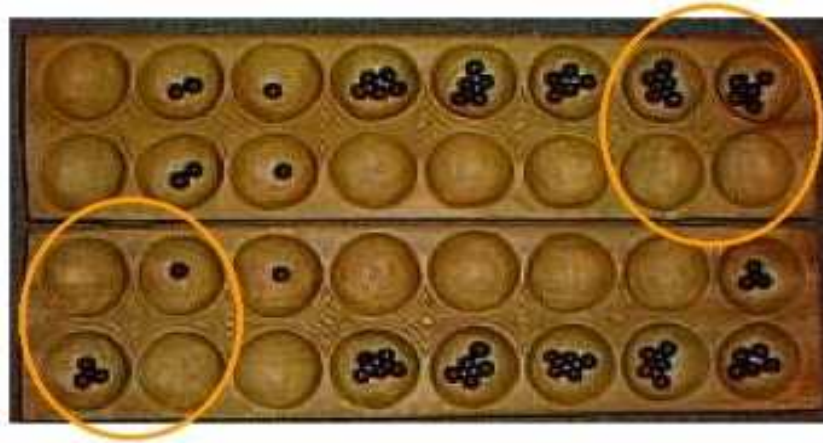
Un caso particular: Omweso



- Tablero 8x4.
- Dos filas cada jugador.
- 32 casillas no diferenciadas.
- 64 semillas no diferenciadas.

- Inicialmente cada jugador reparte 32 semillas en su lado del tablero.
- Movimientos de siembra encadenados.
- Captura con reentrada de semillas capturadas.
- **Captura inversa** desde las 4 casillas más a la izquierda.

Captura inversa



- Se permitirá iniciar un movimiento en el sentido de las agujas del reloj, (sentido contrario al habitual) siempre que:
 - El movimiento implique una captura.
 - El jugador inicie el movimiento desde las 4 casillas a la izquierda de su lado del tablero.

Imágenes © International Omweso Society (www.omweso.org)

Interesante para su estudio

- Juego de dos agentes, información perfecta y completa, de turnos secuenciales y de suma nula. Por qué es interesante?
 - No existen muchos estudios al respecto.
 - El efecto del movimiento de siembra, encadenamientos y reentrada capturas.
 - Movimientos encadenados infinitos.
 - No existen implementados juegos con IA.

Tabla de contenidos

- Estado de la cuestión
- **Objetivos**
- Desarrollo
- Resultados
- Conclusiones
- Líneas futuras

Objetivos

- Desarrollar una aplicación que permita jugar a Omweso.
- Desarrollar agentes que usen algoritmos de búsqueda.
- Posibilidad enfrentar varios agentes de IA.
- Creación de lotes de partidas y guardando logs.
- Estudiar diferentes funciones de evaluación.

Tabla de contenidos

- Estado de la cuestión
- Objetivos
- **Desarrollo**
- Resultados
- Conclusiones
- Líneas futuras

Diagrama de casos de uso

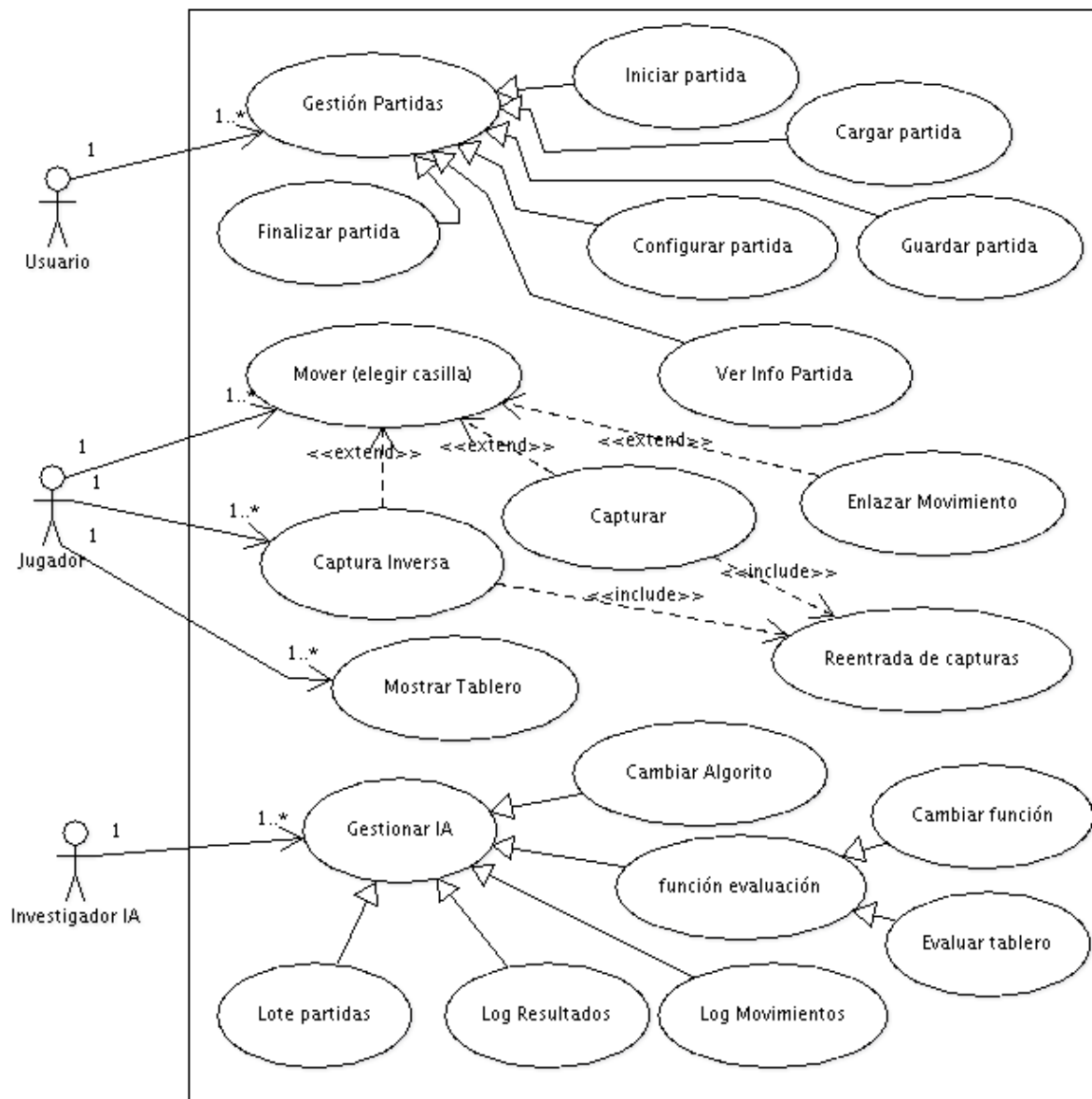
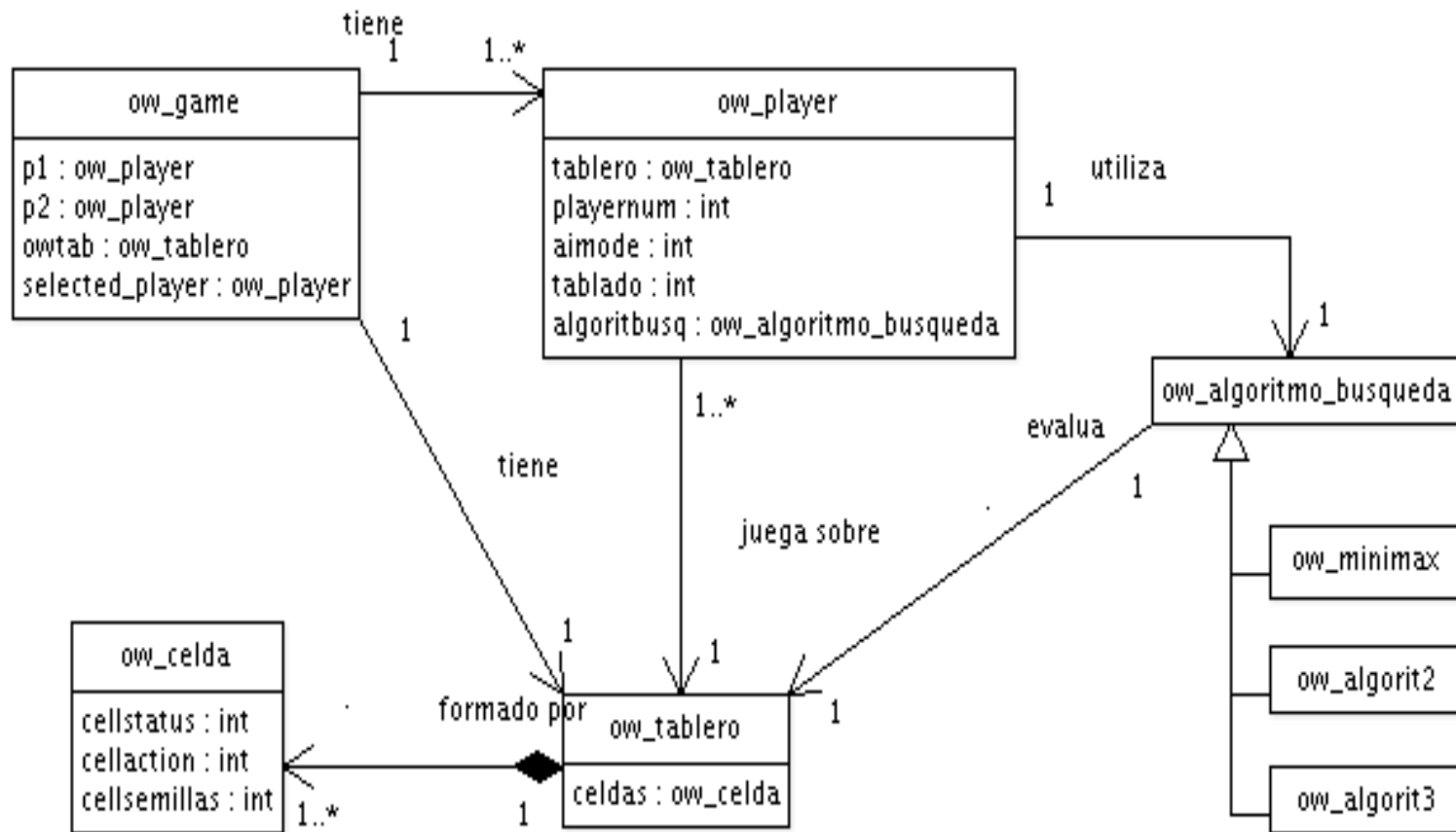


Diagrama de clases



Evaluación diferencia de material

Se entiende por diferencia de material, la diferencia entre el número de semillas que tiene un jugador y otro en un momento del juego.

$$F(x) = \frac{(\zeta_1 - \zeta_2)}{(1 + \zeta_1 + \zeta_2)}$$

Siendo $F(x)$ la función que evalúa la diferencia de material, y siendo ζ_1 y ζ_2 , el número de fichas que poseen el jugador 1 y el jugador 2.

La diferencia de material promueve las capturas.

Evaluación de movilidad

- Se entiende por movilidad, la cantidad de casillas desde la que se puede iniciar un movimiento el jugador durante su turno.

$$f_i = \frac{(\zeta_1 - \zeta_2)}{(1 + \zeta_1 + \zeta_2)}$$

- Donde ζ_1 y ζ_2 son las variables que miden la cantidad de jugadas disponibles que tienen el jugador 1 y el jugador 2.

Evaluación de la dispersión

- Aumentando la dispersión de nuestro material aumentará nuestra movilidad, y además reduciremos la cantidad de material capturado en una sola captura.

$$\sigma_i = \sqrt{\frac{1}{n}(c_i - \mu)^2}$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (c_i - \mu)^2}$$

- Definiremos el cálculo de la dispersión de las semillas en el tablero como:

$$f_i = \frac{(\zeta_1 - \zeta_2)}{(1 + \zeta_1 + \zeta_2)}$$

$$\zeta_{k=12} = \sigma_{max} - \sigma_{k=12}$$

$$\sigma_{max} = \sigma_{(0,0,\dots,0,8n)} = \sqrt{\frac{1}{n}[(n-1)(0-\mu)^2 + (1)(8n-\mu)^2]}$$

$$\sigma_{max} = \sigma_{(0,0,\dots,0,8n)} = 8\sqrt{(n-1)}$$

Evaluación del alcance

- Se entiende que una casilla está en alcance cuando iniciando un movimiento desde ella, el movimiento acabaría en una casilla de la fila interior, la fila desde la que se realizan las capturas.

$$f_i = \frac{(\zeta_1 - \zeta_2)}{(1 + \zeta_1 + \zeta_2)}$$

- Donde ζ_1 y ζ_2 son las variables que miden la cantidad de casillas en alcance que tienen el jugador 1 y el jugador 2.

Combinación lineal

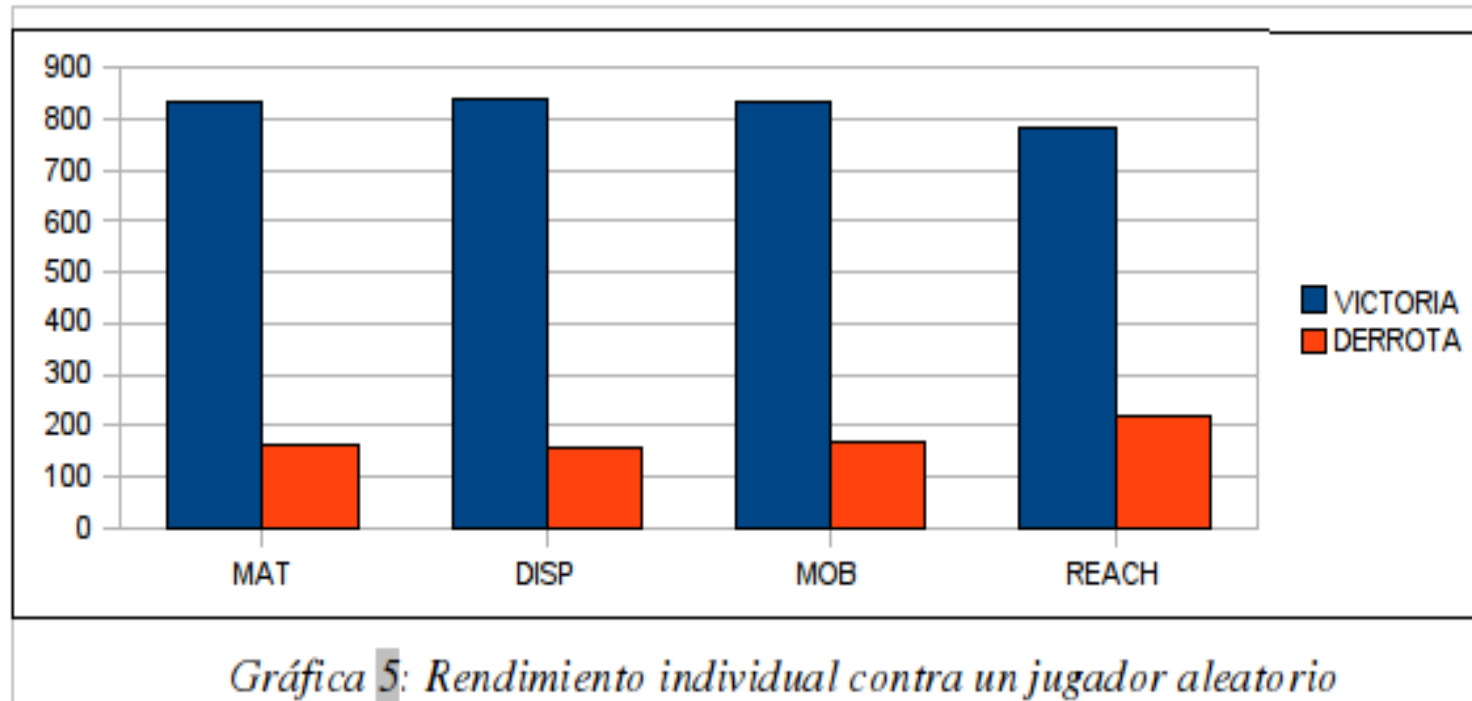
$$F(x) = \sum_{i=1}^k \omega_i f_i = \omega_1 f_1 + \omega_2 f_2 + \dots + \omega_k f_k$$

- Deberemos decidir que peso dar a cada característica en la combinación lineal, mediante experimentación.

Tabla de contenidos

- Estado de la cuestión
- Objetivos
- Desarrollo
- **Resultados**
- Conclusiones
- Líneas futuras

Rendimiento individual



- La gráfica representa el número de victorias de cada característica individual enfrentándola contra un jugador aleatorio.

Cálculo pesos comb. lineal (1)

	<i>MAT</i>	<i>DISP</i>	<i>MOB</i>	<i>REACH</i>	<i>SCORE</i>
<i>MAT</i>	-	440	493	672	1605/3000
<i>DISP</i>	559	-	513	640	1712/3000
<i>MOB</i>	507	486	-	658	1651/3000
<i>REACH</i>	328	360	341	-	1029/3000

Tabla Resultados 6: Resultado medio de enfrentar las características

$$\omega_{mat} = \frac{C_{mat, disp}}{(\min(C_{i, disp}))} = \frac{440}{360} = 1,22$$

$$\omega_{mat} = \frac{C_{mat, mob}}{(\min(C_{i, mob}))} = \frac{493}{341} = 1,44$$

$$\omega_{mat} \in [1,22; 1,44]$$

$$\omega_{mat} = \frac{1}{2}(1,22 + 1,44) = 1,33$$

Calculamos de la misma manera el resto de los pesos:

$$\omega_{mat} = 1,33; \omega_{disp} = 1,60; \omega_{mob} = 1,44; \omega_{reach} = 1$$

$$F_1 = \sum_{i=1}^k \omega_i f_i = (1,33 f_{mat}) + (1,60 f_{disp}) + (1,44 f_{mob}) + (1,00 f_{reach})$$

Cálculo pesos comb. lineal (2)

Eliminamos la característica con peor rendimiento individual, el alcance.

	<i>MAT</i>	<i>DISP</i>	<i>MOB</i>
<i>MAT</i>	-	440	493
<i>DISP</i>	559	-	513
<i>MOB</i>	507	486	-

Tabla Resultados 7: Resultados tras eliminar la característica del alcance

Y calculamos los pesos de nuevo sin la intervención del alcance:

$$\omega_{mat} \in [1,00; 1,00]$$

$$\omega_{mat} = 1$$

$$\omega_{disp} \in [1,05; 1,10]$$

$$\omega_{disp} = \frac{1}{2}(1,05 + 1,10) = 1,07$$

$$\omega_{mob} \in [1,00; 1,10]$$

$$\omega_{mob} = \frac{1}{2}(1,00 + 1,10) = 1,05$$

Con lo que tendríamos nuestra nueva función de evaluación:

$$F_2 = \sum_{i=1}^k \omega_i f_i = (1,00 f_{mat}) + (1,07 f_{disp}) + (1,05 f_{mob})$$

Cálculo pesos comb. lineal (3)

Enfrentaremos las dos funciones de evaluación que hemos creado:

$$F_1 = \sum_{i=1}^k \omega_i f_i = (1,33 f_{mat}) + (1,60 f_{disp}) + (1,44 f_{mob}) + (1,00 f_{reach})$$

$$F_2 = \sum_{i=1}^k \omega_i f_i = (1,00 f_{mat}) + (1,07 f_{disp}) + (1,05 f_{mob})$$

Obteniendo los siguientes resultados:

	<i>F1</i>	<i>F2</i>	<i>RAND</i>	<i>MAT</i>	<i>DISP</i>	<i>MOB</i>	<i>REACH</i>
<i>F1</i>	-	503	895	490	443	514	681
<i>F2</i>	491	-	892	499	525	523	690

TablaResultados 8: Resultados comparativos de F_1 y F_2

- Los enfrentamientos F1/F2 resultados próximos al 50% de victorias.
- F2 tiene resultados superiores contra características individuales.
- F2 supone un ahorro de recursos al eliminar el cálculo del alcance.
- **Se recomienda la utilización de F2.**

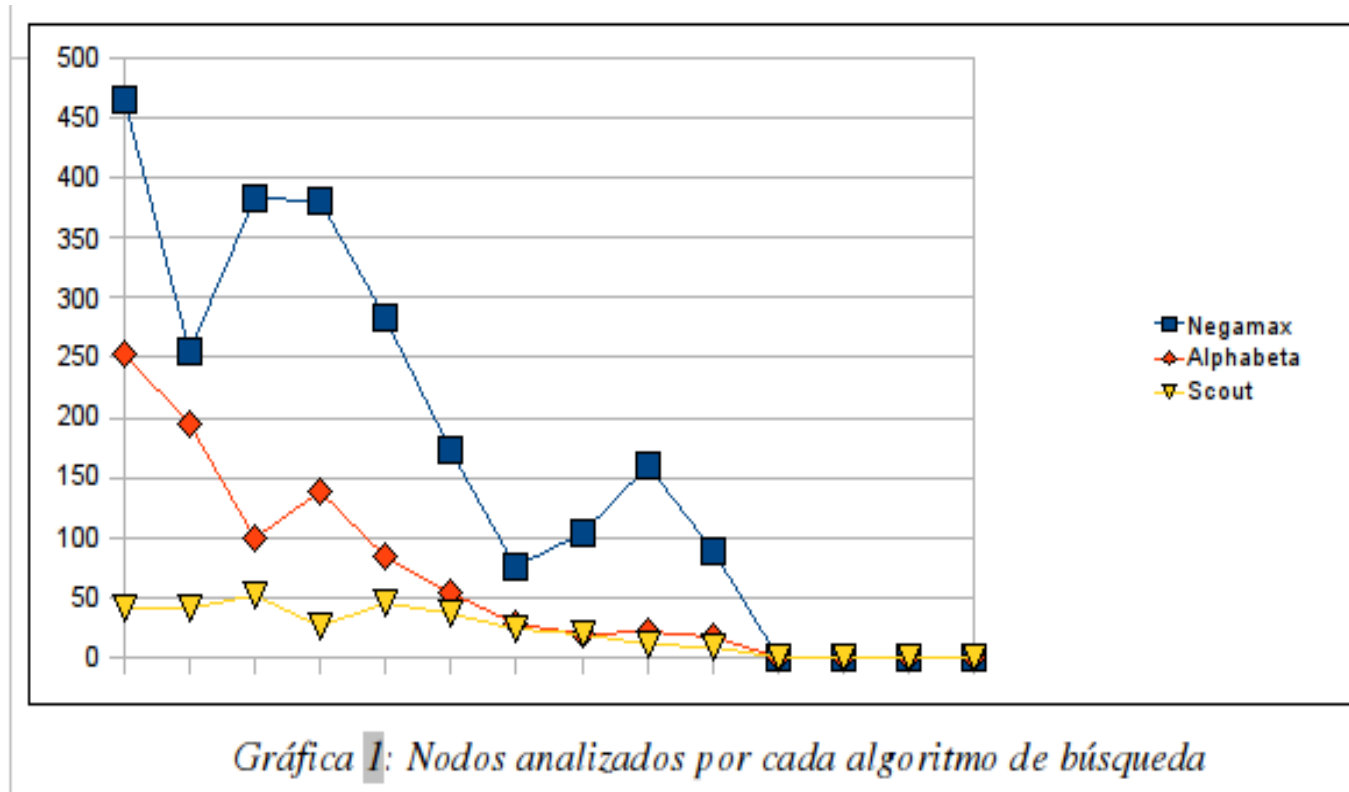
Rendimiento de los algoritmos

	Negamax			Alfabeta			Scout	
	nodes	time		nodes	time		nodes	time
1	466	0.0162	1	254	0.0064	1	42	0.0136
2	255	0.0093	2	195	0.0067	2	41	0.0425
3	383	0.0265	3	99	0.0030	3	51	0.0291
4	381	0.0329	4	138	0.0043	4	26	0.0345
5	284	0.0282	5	85	0.0028	5	46	0.0097
6	174	0.0223	6	54	0.0021	6	36	0.0358
7	76	0.0029	7	29	0.0013	7	23	0.0076
8	104	0.0058	8	20	0.0008	8	19	0.0200
9	160	0.0057	9	22	0.0008	9	11	0.0040

Tabla Resultados 1: Nodos analizados y tiempo consumido.

- La tabla representa los nodos analizados y tiempo consumido durante los 9 primeros turnos de tres partidas en las que se utilizó Negamax, Alfabeta y Scout, respectivamente. Profundidad 3.

Rendimiento de los algoritmos (2)



- La gráfica representa los nodos analizados durante los 9 primeros turnos de tres partidas en las que se utilizó Negamax, Alphabeta y Scout. Profundidad 3.

Resultados contra humanos

	F_2 PROF3	F_2 PROF4
HUMANO1	5/10	1/10
HUMANO2	4/10	2/10

TablaResultados 9: Resultados de F_2 contra jugadores humanos

- La gráfica representa el número de victorias de jugadores humanos contra un agente utilizando Alfabetá.

A profundidades superiores a 6, los jugadores humanos perdieron siempre.

Tabla de contenidos

- Estado de la cuestión
- Objetivos
- Desarrollo
- Resultados
- **Conclusiones**
- Líneas futuras

Conclusiones

- Repaso de los objetivos
 - El tablero
 - El interfaz
 - Los agentes de IA
 - Lote de partidas y log de resultados
- Implementación del sistema
 - Portable y ampliable
 - Eficiente y rápido

Conclusiones (2)

- **Conclusiones de los resultados**
 - Rendimiento de los algoritmos de búsqueda.
 - Las funciones de evaluación.
 - Movimiento de siembra, encadenamiento y reentrada de capturas.

Tabla de contenidos

- Estado de la cuestión
- Objetivos
- Desarrollo
- Resultados
- Conclusiones
- **Líneas futuras**

Líneas futuras

- Mejoras a la implementación.
 - Juego online
 - Implementación *multithread*
- Mejoras a la Inteligencia Artificial
 - Algoritmos de segunda generación
 - Funciones de evaluación no lineales
 - Funciones dependientes del turno
 - Bases de datos de movimientos

DEMO

APPLICATION MAIN MENU

```
newgame [new]   create a new game.
loadgame [load] load an existing game.
savegame [save] save the current game.
setup    [setup] edit the cell contents of the current board.
endgame  [end]   end the current game.
play     [p]     continue playing the current game.
manageai [ai]    manage artificial intelligence.
version  [ver]   show program version.
help     [h]     show this help screen.
quit     [q]     exit from omweso.
```

omweso:/> █

omweso[S1]:/player1> b

```
[N]  H G F E D C B A [N]
0-----0
1 | 04 00 00 00 00 00 00 00 | 1
0 | 00 04 04 04 04 04 04 04 | 0
0-----0
0 | 04 04 04 04 04 04 04 00 | 0
1 | 00 00 00 00 00 00 00 04 | 1
0-----0
[S]  a b c d e f g h [S]
```

omweso[S1]:/player1> █

