MICRO-SHOGI

@claraantolingarcia
TBD
@GiraldTec

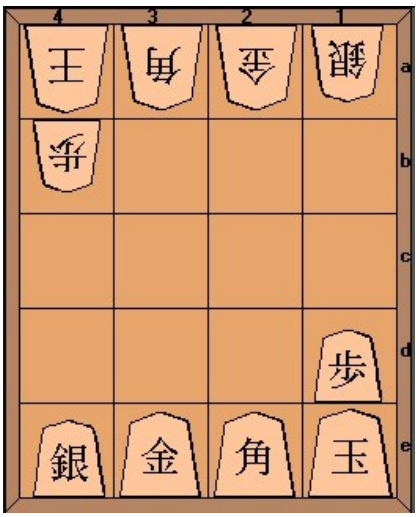
Introducción al juego

El **shogi** o **ajedrez japonés** es un juego de mesa para dos jugadores perteneciente a la misma familia que el ajedrez.

Se juega sobre un tablero de 9 filas por 9 columnas. Normalmente comienzan las negras, salvo que se decida lo contrario.

Cada jugador dispone de 20 piezas iguales a las del otro participante. Las piezas de los reyes sí que tienen un sutil diferencia de un trazo en su ideograma (el símbolo japonés escrito en la ficha), para definir cuáles son negras o blancas. Se diferencian las piezas de un jugador de las del otro por la dirección que señalan sobre el tablero, hacia el oponente.

En el caso del micro-shogi, se juega con un tablero simplificado de 4x5 y con una colección más reducida de piezas. Por otro lado, al capturar piezas del adversario, la pieza que captura promociona.



<u>Reglas y fin del juego</u>

Gana el jugador que consigue capturar el Rey del contrario. O el que hace que su rival se rinda.

El Rey contrario puede ser capturado sin necesidad de decir jaque u *ōte* 王手 en japonés. Del mismo modo, no hace falta decir jaque mate o *tsumi* 詰み en japonés.

Movimiento fichas

Únicamente están indicados los movimientos de las piezas que entran dentro del micro-shogi.

REY (‘K’)

ALFIL (‘B’)

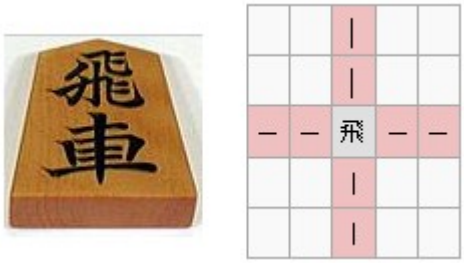
GENERAL DE ORO (‘G’) **/ PEÓN CORONADO** (‘+P’)

GENERAL DE PLATA (‘S’)

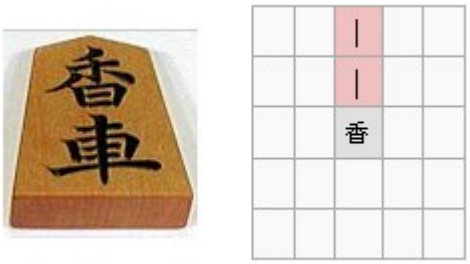
PEÓN (‘P’)

CABALLERO (‘N’)

CARRO (‘R’)



LANCERO (‘L’)



Promoción de las fichas

A diferencia del shogi estándar las fichas se promocionan cuando comen y es de carácter obligatorio. Así mismo, el mecanismo se diferencia en el hecho que una pieza promocionada, al comer, vuelve a su estado anterior. Esto no impide que pueda volver a promocionar.

La promoción funciona de esta manera

- El Rey no promociona.
- El General de plata promociona a Lancero y viceversa.
- El Alfil promociona a “Peón coronado” y viceversa.
- El General de oro promociona a torre y viceversa.
- El peón promociona a Caballo

Un caballero que alcance las dos últimas filas está atrapado. De la misma manera, un peón que alcance la línea final. Una lanza también se encuentra atrapada en la última fila, pero puede escapar si con ese movimiento ha comido y se transforma en un general de plata. Un General de plata que capture en un movimiento en la última fila, promociona a Lancero y se queda atrapado.

Las piezas atrapadas en las últimas filas pueden ser comidas.

Modos de juego

Hemos incluido varios modos de juegos antes de iniciar la partida, para tener la posibilidad de jugar contra una persona, de forma que no necesitamos ninguna heurística ni árboles Min&Max ya que el jugador pone la ficha que quiera el la casilla que él elija siempre y cuando ese movimiento sea válido, de no ser así no se efectuará ningún movimiento y se volverá a pedir un nuevo movimiento.

a -> Jugador vs Jugador
b -> Jugador vs PC
c -> PC vs PC

Instrucciones de juego:

➔ Se carga el fichero **microshogi** y se ejecuta **?- comienzaPartida.** Y se elige un modo de juego:

```
1 ?- comienzaPartida.
haz un movimiento (ej: "(3,3),(3,2)") o escribe "salir" o "exit" para salir
¿Qué modalidad quieres?
        a: Jugador1 vs Jugador2
        b: Jugador vs PC
        c: PC1 vs PC2
        eoc -> Salir
|:
```

➔ Luego decidiremos si queremos empezar los primeros o los segundos:

```
Quieres empezar el primero o el segundo?
        1: Primero
        2: Segundo
|:
```

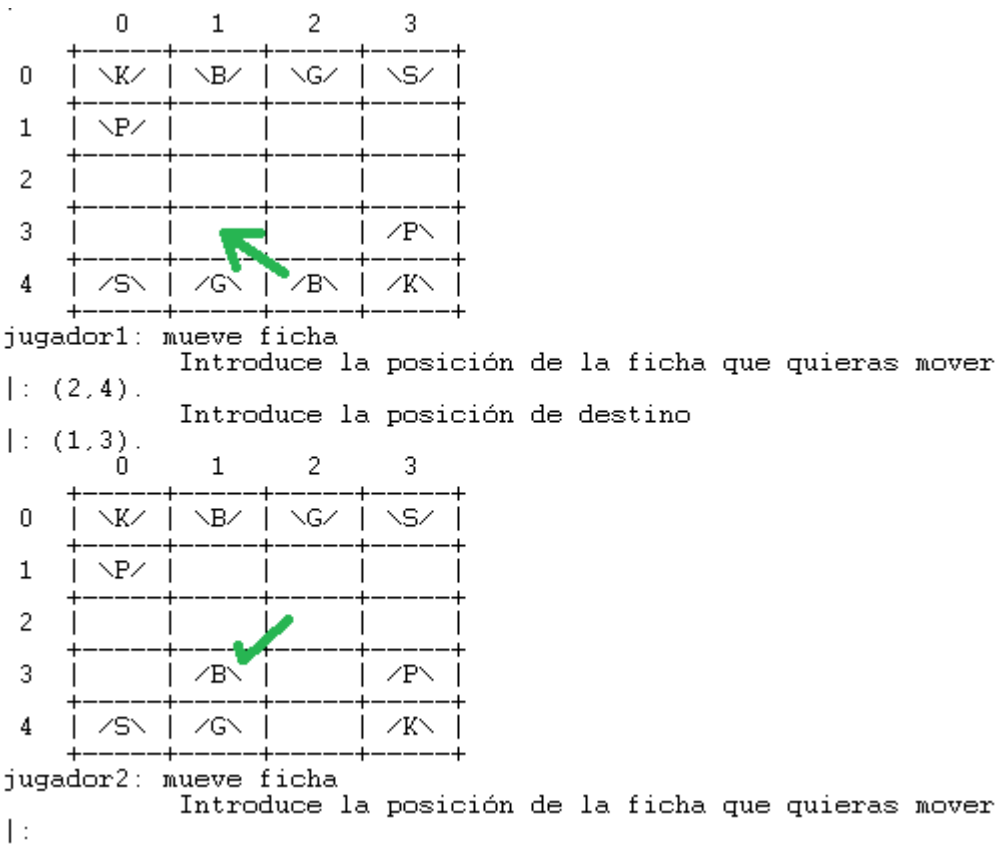
➔ En caso de jugar contra el PC, modos ‘b’ y ‘c’ tendremos que decidir el nivel del árbol al que queremos llegar, cuanto más alto sea el número del nivel, más preciso y mejor será el movimiento pero más tiempo tardará en calcularse:

```
¿A qué nivel de árbol quieres jugar?
|:
```

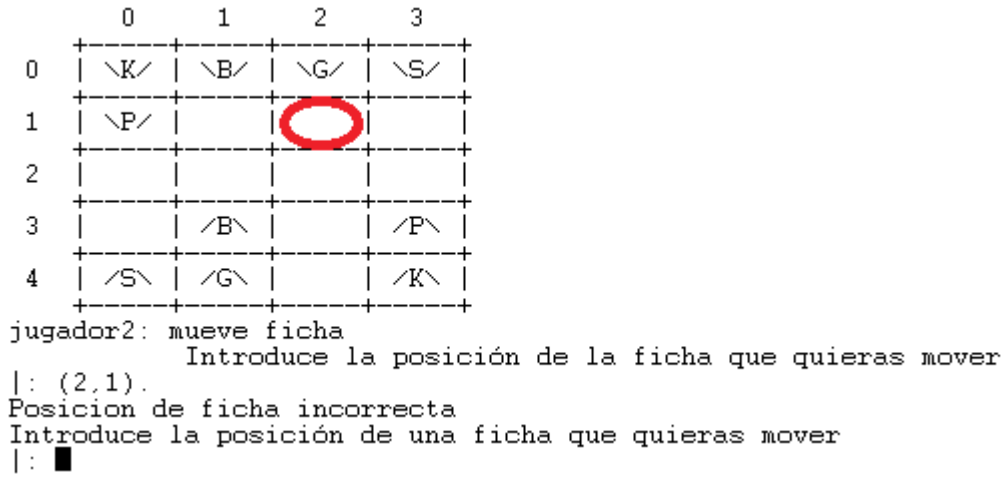
➔ Ya tenemos todo listo para empezar a jugar. (ej con el modo a)

```
      0     1     2     3
   +-----+-----+-----+-----+
 0 | \K/ | \B/ | \G/ | \S/ |
   +-----+-----+-----+-----+
 1 | \P/ |     |     |     |
   +-----+-----+-----+-----+
 2 |     |     |     |     |
   +-----+-----+-----+-----+
 3 |     |     |     | /P\ |
   +-----+-----+-----+-----+
 4 | /S\ | /G\ | /B\ | /K\ |
   +-----+-----+-----+-----+
jugador1: mueve ficha
          Introduce la posición de la ficha que quieras mover
|:
```

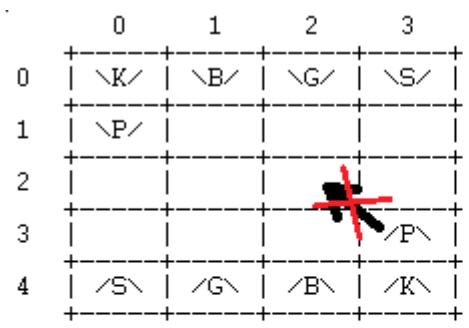
➔ El jugador en este momento deberá elegir un movimiento para una de sus fichas:



➔ No puede elegir una casilla ocupada por una ficha del contrincante o simplemente vacía ya que si no no se efectuará el movimiento, y se volverá a pedir una ficha para mover:

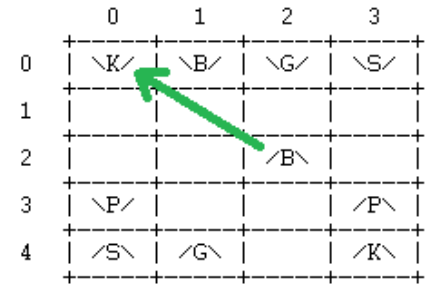


➔ Tenemos un control de errores que nos permite evaluar la ficha seleccionada y la casilla destino de forma que comprobaremos, una vez que sabemos que la ficha es suya, si el movimiento que queremos realizar está dentro de las posibilidades de la ficha, cada ficha tiene un movimiento distinto, por lo tanto tendremos una comprobación distinta para cada una de las fichas de cada jugador.

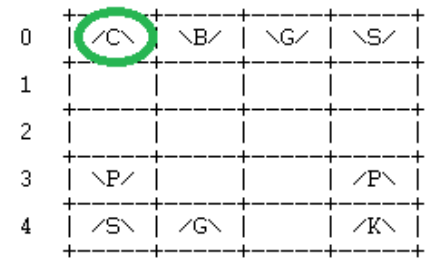



```
jugador: mueve ficha
          Introduce la posición de la ficha que quieras mover
|: (3,3).
          Introduce la posición de destino
|: (2,2).
Posicion de ficha incorrecta
Introduce la posición de una ficha que quieras mover
|:
```

➔ Gana el que consiga comer el Rey contrincante:




```
jugador1: mueve ficha
          Introduce la posición de la ficha que quieras mover
|: (2,2).
          Introduce la posición de destino
|: (0,0).
```




```
jugador2: mueve ficha
          Introduce la posición de la ficha que quieras mover
******* Game Over *******
¡¡¡El jugador jugador1 ha ganado la partida!!!
**************************
haz un movimiento (ej: "(3,3),(3,2)") o escribe "salir" o "exit" para salir
```

➔ Si queremos salir del juego, basta con darle a cualquier letra que no sea ‘a’, ‘b’, ó ‘c’ en el modo de juego

```
¿Qué modalidad quieres?
        a: Jugador1 vs Jugador2
        b: Jugador vs PC
        c: PC1 vs PC2
        eoc -> Salir
|: exit.
Hasta pronto!
true.
```

Motivación y herramientas usadas

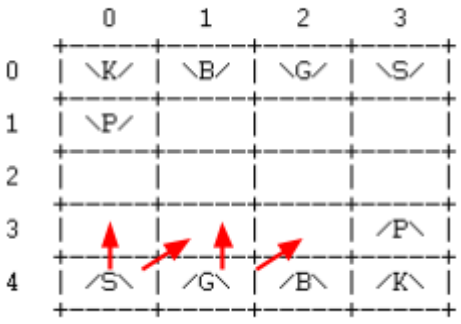
Barajamos otros tipos de juegos para resolver, pero el que más nos ha motivado para nuestra práctica ha sido el juego de micro shogi, ya que es muy parecido al ajedrez, un juego de mesa ya conocido pero con unas cuantas modificaciones, como la promoción de las fichas al comer, los movimientos tan peculiares de éstas, etc. que hace de micro shogi un juego con ganas de implementar. Cabían tres posibilidades por hacer: Shogi con un tablero 9 x 9, Mini Shogi con un tablero 5 x 5 y Micro Shogi con un tablero 5 x 4. Al final, viendo la complejidad del árbol de búsqueda que tiene cada uno, nos decidimos por el Micro Shogi para poder asegurar mejor eficacia.

Aunque nuestra idea original era realizar el proyecto usando *frolog*, por recomendación del profesor, nos decantamos por utilizar únicamente *Prolog*, puesto que no íbamos a provechar más que unas pocas de las características que nos aporta éste y requeriría un cierto tiempo de adaptación durante el cual nuestro proyecto estaría detenido.

Diseño del jugador máquina

Nuestra versión de Micro-Shogi tiene la posibilidad de establecer uno o los dos jugadores como virtuales. En el desarrollo del juego reciben los apelativos de PC, PC1 y PC2. El método de selección de movimientos de estos jugadores virtuales se basa en la selección de un movimiento inicial, el uso del algoritmo Min&Max con poda Alfa-Beta, y en el cálculo de una función heurística específica diseñada por nosotros.

Para mejorar el rendimiento y la rapidez del juego hemos decidido que la IA realice aperturas de libro con los movimientos más comunes y que mejoran las puntuación inicial como primera jugada, estas “aperturas de libro” las hemos escogido observando partidas de otros jugadores humanos. Por ello, la primera jugada realizada por el jugador de IA se escoge entre una de las 4 opciones prefijadas al azar.



Nuestro algoritmo Min&Max está diseñado para moverse por el espacio de configuraciones de tablero. Es importante que el usuario introduzca una profundidad de búsqueda a la cual el Min&Max, al recorrer el árbol de movimientos, se detenga y asigne a las configuraciones “hoja” un valor específico acorde a su viabilidad como victoriosa.

Esta aproximación mediante el algoritmo Min&Max choca con la mayoría de estudios que vimos en nuestra investigación sobre las IA’s del Shogi, los cuales hacían un uso extendido del algoritmo de Montecarlo, mucho más adecuado para juegos cuyo fator de ramificación es tan sumamente elevado. De todas formas, dado el reducido número de piezas del Micro-Shogi, su pequeño tablero y algunas mecánicas propias del Shogi que decidimos obviar para nuestra versión, nuestro espacio de estados se reduce, evitando así posibles problemas de rendimiento.

En cuanto a lo referente a la función heurística seguimos este proceso mental:

- No hay Rey propio => -99999
- No hay Rey adversario => +99999

En caso de faltar ambos reyes, existe un corte en la primera regla, que hace que sólo cuente para el jugador la ausencia de su propio Rey.

En cualquier otro caso, se hace un cálculo de los puntos propios y del contrincante, y se calcula la diferencia de pesos de las fichas que hay en juego de ambos jugadores.

El cálculo de puntos del jugador que mira en el árbol de búsqueda se realiza de la manera:

- Penalización por fichas bloqueadas: Referente a el peón, el caballo, o la lanza
  => - 50 * Peso de la ficha

- Bonificación por fichas adelantadas: Consideramos el “campo” de cada jugador como las dos filas más cercanas al borde del tablero.
  => + 100 * Peso de la ficha alejada del “campo”.

- Penalización por avance del Rey: Dado que el Rey es la ficha más importante, que avance mucho implica desprotegerlo. Se penaliza de la forma:
  => - 0 si se mantiene en su fila,
  => - 50 si ha avanzado una fila,
  => - 50000 si ha avanzado más de una.

El cálculo de puntos del adversario se hace igual, obviando la penalización del adelantamiento del Rey, y dividiendo el resultado por 2.

Finalmente se resta la puntuación contraria a la propia y se suma la diferencia de puntos por fichas en juego para obtener un valor propio, que cuanto más alto sea mayor es la viabilidad de la configuración que tenga asociada como victoriosa.

Reparto de trabajo

Hemos procurado repartir el trabajo de la manera más equitativa posible, con el objetio de que todos pudieramos aprender algo nuevo y aplicar nuestros conocimientos en el tema de más de una manera. Inicialmente, nos repartimos el trabajo de esta manera:

- **@GiraldTec**: Investigación sobre la heurística e implementación y generación de movimientos.
- **@claraantolingarcia**: Diseño de la interfaz del tablero e implementación de la heurística.
- **TBD**: Implementación del algoritmo Min&Max y generación de movimientos.

Según hemos ido terminando nuestras respectivas partes, nos hemos ido dedicando a completar las partes más atrasadas, y una vez terminado. Al diseño de la heurística, que es el apartado más complejo del proyecto.

Conclusiones

Tuvimos problemas en la implementación del algoritmo Min&Max a la hora de introducir las podas en la toma de decision. Hemos conseguido adaptar la poda Alfa en dicho proceso, pero dadas las podas excesivas podas que obteníamos con la poda Beta, concluímos qeu habría un error en el sistema y decidimos obviarla. Aún así, en el caso de que el usuario juegue contra la máquina, las jugadas son aceptables y no entorpecen demasiado la jugabilidad.

Llevar a cabo la práctica ha sido, en conclusión, un buen ejercicio de progamación declarativa. No hemos usado librerías extra de listas o de dominios finitos. Pero en el desarrollo de la práctica aplicamos todos los conocimientos adquiridos en la materia. Con la práctica, también hemos aprendido sobre el Shogi y sus variantes, hemos conocido más a fondo los algoritmos que se suelen usar para programar Inteligencias Artificiales en los juegos, y aunque no llegamos a usar Frolog, Prolog se ha mostrado una herramienta potente, la cual nos atrevemos a afirmar controlamos mejor que al principio del curso.

Bibliografía

- Shogi hecho en java
  http://sourceforge.net/projects/yagura-shogi/
- Micro-Shogi y Min&Max
  https://github.com/PaulVaroutsos/Scripts-Algorithms-and-Other-Projects
- Algoritmo PN en el Shogi
  http://www.sciencedirect.com/science/article/pii/S0004370201000844
- Nuestro repositorio
  https://code.google.com/p/micro-shogi-prolog-prompt/