

Javier Díez Bermúdez de Castro

100033025@alumnos.uc3m.es

ABSTRACTO

El presente artículo es un estudio sobre la implementación de un caso particular de juego no determinista: el parchís

1.INTRODUCCIÓN

El objetivo de esta práctica es programar en Java un jugador de parchís dotado de una cierta inteligencia artificial. Existen varias modalidades a la hora de practicar este juego, la modalidad elegida en este caso es la de un parchís con 4 colores y dos jugadores, de manera que cada jugador controla una pareja de colores.

Se trata de un problema de búsqueda con un adversario. Dado que el parchís es un juego no determinista, no existen reglas exactas que conduzcan de forma segura a un resultado óptimo. De hecho, si nos fijamos en los jugadores humanos podemos ver que cada uno aplica unas determinadas reglas en función de su estilo de juego. Las reglas elegidas son, por tanto, heurísticas. Lo cual no significa que sean incorrectas.

2.CUERPO DE LA MEMORIA

2.1 JUEGOS DETERMINISTAS vs JUEGOS NO DETERMINISTAS

Es posible separar en dos grupos bien diferenciados a los juegos. Por un lado tenemos a los juegos deterministas, en los cuales el azar no está presente, y por otro lado tenemos a los juegos no deterministas en los cuales el azar representa un factor clave en el resultado final.

A la hora de implementar un sistema capaz de jugar de manera inteligente a un juego determinado esa diferenciación es crucial ya que en un juego no determinista la función de utilidad depende del factor “suerte”, es decir debemos tener en cuenta la probabilidad de que ocurra un suceso u otro. De esta manera, a diferencia del caso de un juego determinista, la función de utilidad solo es capaz de arrojar resultados aproximados.

2.2 ESTRATEGIA MINIMAX

La estrategia mínimas consiste en elegir el mejor movimiento para uno mismo (MAX) suponiendo que el adversario (MIN) escogerá el mejor para sí mismo:

Los pasos que sigue el algoritmo son los siguientes:

1. Generar el árbol de juego, alternando movimientos de MAX y MIN y asignándoles los valores apropiados.
2. Calcular la función de utilidad de cada nodo final recorriendo recursivamente los nodos hasta el estado inicial
3. Elegir como jugada a realizar aquel primer movimiento que conduce al nodo final con mayor función de utilidad

2.3.DESCRIPCIÓN DEL JUEGO

Aunque el juego del parchís es bastante conocido, conviene presentarlo adecuadamente:

2.3.1.ORIGEN

Es éste un juego de los que se englobarían dentro de los llamados juegos de carreras. Es una variante casi idéntica del Ludo, introducido en Inglaterra en 1896. El Parchís, como el Ludo, son herederos directos del juego de origen indio denominado Parchisi. El Parchisi en su país de origen, la India, se juega sobre un tablero de paño en forma de cruz, extendido sobre el suelo o sobre una mesa. Existen en los palacios de Agra y de Allahabad grandes tableros de Parchisi con casillas de mármol rojo y blanco, en los que el emperador jugaba usando dieciséis

concubinas de su harén a modo de fichas vivientes.

2.3.2. REGLAS DEL JUEGO

El parchís se juega sobre un tablero en forma de cruz alrededor de la cual están dispuestas 68 casillas, 16 de las cuales (debidamente diferenciadas de las demás) constituyen los llamados seguros. Se emplean cuatro grupos de cuatro fichas con un color diferente para cada uno de los grupos.

Existen además cuatro espacios ajenos a lo que es el propio recorrido de la carrera y cada uno del color respectivo de cada grupo de fichas, o casas, en los que cada jugador almacena las fichas que en ese momento no se hallan incorporadas al juego.

Las fichas avanzan a lo largo del recorrido de 68 casillas según indique el dado de 6 caras que ha de arrojar cada jugador por turnos alternativos.

Al comienzo de la partida el tablero tiene este aspecto:

Al comienzo de la partida las fichas son colocadas en la casa del color correspondiente, salvo una ficha de cada color, situada en su correspondiente casilla de salida.

Los jugadores irán incorporando fichas en la zona de juego sólo cuando en su tirada les salga un cinco, y a razón de una sola ficha por tirada. Esta incorporación será obligatoria cada vez que a un jugador en su turno le salga un cinco y tenga alguna ficha en casa. En estos casos, y tras el consiguiente 5, el jugador sacará una de sus fichas de casa colocándola en la casilla/seguro de salida de su color.

Una vez que se tenga una ficha en el terreno de juego, el jugador que arroja el dado deberá avanzar su ficha tantas casillas como indique el dado, en el sentido de la numeración ascendente de las mismas (sentido contrario a las agujas del reloj). Si se tuviera más de una ficha en juego, podrá optarse por mover la que más interese al jugador.

Excepcionalmente, y sólo cuando ninguna de las cuatro fichas del color de la ficha a mover estén en su casa, el valor 6 del dado deberá traducirse en el tablero en un avance de 7 casillas.

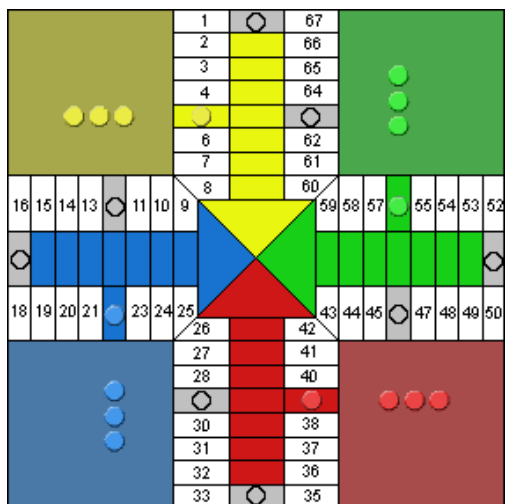
Siempre que en el turno se saque un 6, tras realizar el movimiento correspondiente se repetirá tirada hasta un máximo de dos repeticiones (es decir, tres tiradas).

Si un jugador opta en una tirada por colocar una ficha en una casilla ya ocupada por otra del mismo color, constituirá una barrera.

Las barreras hacen imposible el paso de cualquier ficha en su camino hacia la meta (incluso de las fichas propias de quien la constituye). Si una ficha no pudiera completar su avance por el número total de casillas que le correspondiera, el movimiento no se realizará: nunca se contará un número menor de casillas.

Quando un jugador tenga alguna barrera constituida por algunas de sus fichas y en su turno saca un 6 estará obligado a abrir barrera, o

Figura 1. Tablero de juego



Vence el jugador que antes consiga introducir cada una de sus fichas en su respectiva casilla final, siguiendo las pautas de movimientos que se describen a continuación.

sea, mover una de las fichas que forman la misma.

Cuando, tras contar el número correspondiente, una ficha caiga en una casilla ocupada por una ficha de otro color diferente, la primera deberá comer a la que allí estaba, volviendo la ficha comida a casa (lógicamente esperando un nuevo 5 para volver a entrar en juego). Cuando se come una ficha, el jugador que come deberá avanzar una cualquiera de sus fichas en número de 20 casillas.

Una excepción a la regla anterior es el caso de que la ficha que ya ocupaba primero la casilla, lo estuviera haciendo en un seguro, en cuyo caso podrán compartir esta casilla fichas de diferente color en un número máximo de dos por seguro.

Si una ficha debe acabar su movimiento en un seguro ocupado por 2 fichas, el movimiento no se realizará.

Las casillas de inicio de cada color operan como seguros normales con la particularidad de que si están ocupados por dos fichas y alguna de ellas, o ambas, es de color diferente al de la casilla, si al jugador que juega con el color de la casilla le quedase alguna ficha en casa, y en su turno le sale un 5, sacará ficha, a pesar de que no habría espacio en el seguro, comiendo a la última de las llegadas a esa casilla (contaría veinte con cualquiera de sus fichas posteriormente como en cualquier otra captura).

Al repetir tirada en una serie de dos 6 consecutivos, si en la correspondiente tercera tirada volviese a salir un tercer 6, la ficha con la que se hubiera realizado el último movimiento deberá volver a casa como si se la hubiera comido un contrincante, sin que nadie cuente 20 por esa captura, y el turno finaliza sin mover por el tercer 6.

Cuando una ficha ha dado una vuelta al tablero y llega a la zona baja del pasillo de

llegada de su color, empezará a ascender por las casillas que hay en él.

Para entrar en la casilla de meta el jugador correspondiente tendrá que hacerlo contando el número exacto de casillas que le correspondiera en su movimiento, desde aquella en la que está hasta la casilla central de llegada: si el trayecto fuera más corto que el número indicado por el dado, el movimiento no se llevaría a cabo.

Cuando una ficha alcanza la casilla final de llegada, el jugador deberá hacer avanzar cualquiera de sus piezas 10 posiciones, si le es posible con alguna de ellas.

En el pasillo de llegada de un color sólo podrán entrar fichas de ese color; por tanto, en él no son posibles las capturas.

Si en una serie de tiradas sucesivas de 6, con el segundo movimiento de 6, una ficha acaba en una casilla del pasillo de llegada de su color, y en la correspondiente tercera tirada sale otro 6, esa ficha no realizará ningún movimiento pero tampoco quedará capturada, como ocurriría si esto pasase en cualquier otro lugar del tablero.

El juego finaliza cuando un jugador logra introducir cada una de sus fichas en su respectiva casilla final.

2.4.CONJUNTO DE REGLAS CONSIDERADAS

Tal y como se ha comentado en la introducción de la memoria, estas reglas son heurísticas y no pretenden ser las mas correctas. El proceso de elección es el siguiente:

Se parte de una disposición del tablero, un valor del dado (o de 20 o 10 si hemos comido o alguna pieza ha llegado al final) con el que debemos mover una pieza, y un conjunto de piezas que pueden realizar el movimiento.

Para cada pieza, y en función de las reglas, se evalúa su *rating* (el cual constituye nuestra función de utilidad) que es un número que nos mide lo bueno o malo que resulta mover dicha

pieza. Una vez calculado el *rating* de cada una de las piezas, elegimos aquella que posea el mayor *rating*.

Se va a optar por una estrategia Minimax, ya que aunque no podemos conocer con exactitud los movimientos del jugador contrario (dependen del dado) si vamos a suponer por ejemplo que, si puede, nos comerá.

Las reglas son las siguientes:

1- Si con el movimiento alcanzamos un seguro, sumamos 10 puntos al *rating*.

2- Si la pieza a evaluar se encuentra en un seguro que no sea una casilla de inicio de un color distinto al suyo (puesto que en ese caso podría ser capturada), se restan 5 puntos a su *rating*. No conviene mover las piezas que se encuentran seguras.

2.a) Calculamos el número de piezas enemigas que se encuentra detrás de ella en un rango en el que pueden capturarla. En este caso, un rango de 7 casillas que es el máximo movimiento que podemos conseguir con el dado. Se restan 9 puntos por cada una de las piezas enemigas encontradas.

2.b) Si además esta compartiendo el seguro con alguna pieza enemiga, se restan 6 puntos más al *rating*. La idea es limitar en la medida de lo posible los movimientos del adversario. Aunque nos desplazásemos a otro seguro, si avanzamos esa pieza también estaríamos posibilitando el avance de la pieza adversaria.

3- Si con el movimiento no alcanzamos un seguro, comparamos la posición final que obtendría la pieza con la que tiene actualmente de la siguiente manera, Calculamos para cada una de las dos posiciones (inicial y final) el número de piezas enemigas que tenemos por delante dentro de un rango de captura (7 casillas) y que no estén en un seguro, y el número de piezas enemigas que tenemos detrás y nos pueden capturar. Una vez calculados

calculamos el incremento del *rating* mediante la siguiente fórmula:

$$\Delta \text{rating} = 3 \cdot (\text{incremento} _ \text{piezas} _ \text{atacadas}) + 9 \cdot (\text{reduccion} _ \text{piezas} _ \text{atacantes})$$

Hay que observar que tanto el incremento de piezas atacadas como la reducción de piezas atacantes pueden ser negativos. Este sistema pretende aumentar la probabilidad de realizar una captura y de minimizar la probabilidad de que nos capturen. No podemos asegurar que ocurra ninguna de las dos cosas.

4- Penalización en función del camino recorrido. Puesto que no es bueno dejar piezas rezagadas, se aplica una penalización en función del recorrido:

$$\Delta \text{rating} = 1 - \text{recorrido} / 10$$

5- Comer pieza enemiga: se suman 30 puntos al *rating*.

6- Comer pieza compañera: se restan 15 puntos al *rating*. Aunque no es bueno comer a una pieza compañera, es posible que a pesar de todo sea la mejor opción.

7- Si el movimiento nos obliga a sacar una pieza de casa (un 5), debemos seleccionar, de entre los dos colores posibles, aquel que resulte óptimo:

7.a) Si al sacar la pieza comemos una enemiga, sumamos 30 puntos al *rating*.

7.b) Si al sacar la pieza comemos una compañera restamos 20 puntos al *rating*.

7.c) De los dos posibles colores calculamos aquel que tiene un mayor número de piezas fuera. Restamos 2 puntos al *rating* de las piezas de ese color que estén en casa y le sumamos 2 puntos a las piezas del otro color que estén en casa. Esta regla pretende equilibrar el orden en que sacamos las piezas para no vernos en la posición (sobre todo al inicio) de que tenemos casi todas las piezas en la misma posición del tablero, lo cual restringe nuestra capacidad de maniobra.

7.d) Analizamos la posición de la casilla inicio mirando el número de piezas enemigas

que tenemos tanto delante como detrás dentro de un rango de captura

$$\Delta rating = 3 \cdot n_{\text{piezas_capturables}} - 9 \cdot n_{\text{piezas_enemigas_atras}}$$

2.5. ESTRUCTURA DEL PROGRAMA

Los ficheros de que se compone este programa son las siguientes:

Tablero.java

Extiende de JApplet. Es un applet que presenta por pantalla el contenido del tablero. Sus atributos mas destacados son un objeto casillero y un objeto jugador

Pieza.java

Esta clase sirve para representar las piezas del tablero. Los atributos principales de una pieza son su color, su recorrido y la casilla en la que se encuentra.

Casilla.java

Representa una de las casillas del tablero. Contiene un vector de piezas que, salvo excepciones, contendrá un máximo de dos piezas. Una de las principales características de las casillas es que pueden ser o no, seguras. Es decir que puedan o no realizarse capturas.

Casillero.java

Contiene la totalidad de las casillas del tablero así como los principales métodos para la comprobación de jugadas y su realización. Las casillas se organizan en un vector de casillas que posee la siguiente estructura: Las cuatro primeras casillas corresponden a la casa inicial de cada color siguiendo el orden *amarillo, azul, rojo y verde*. Las siguientes 68 casillas corresponden al recorrido del tablero, común a todas las piezas. Por último tenemos cuatro grupos consecutivos de 8 casillas que

corresponden al pasillo final de cada color siguiendo el mismo orden usado para la casa inicial:

Tabla 1. Organización en memoria

0	1	2	3	[4..71]	[72..79]	[80..87]	[88..95]	[96...103]
Casa Amarilla	Casa Azul	Casa Roja	Casa Verde	Tablero	Pasillo Amarillo	Pasillo Azul	Pasillo Rojo	Pasillo Verde

Jugador.java

Representa al jugador virtual. Con tiene los métodos necesarios para hallar las piezas que pueden realizar un determinado movimiento y aplicarlas las reglas anteriormente descritas con el fin de elegir la pieza a mover.

3.CONCLUSIONES

Al no ser el parchís un juego no determinista, las reglas a aplicar pueden ser un motivo de debate. No es un juego como el ajedrez o las damas en los cuales, sobre todo en este último, existen reglas bien definidas para valorar una determinada posición y elegir en consecuencia la jugada óptima. Como ha podido verse, algunas de las reglas se basan en el aumento o disminución de la probabilidad de que ocurra algún suceso. Esta es una de las principales características de los juegos no deterministas.

Cabe destacar, que en el caso del parchís es posible crear árboles de decisión ya que una jugada puede dar lugar a otra. Esto ocurre tanto cuando comemos una pieza como cuando una pieza llega al final de su recorrido o en el caso de sacar un 6.

Lamentablemente no me ha sido posible incluir esta recurrencia en la toma de decisión debido a problemas de tiempo. De esta forma el “árbol de decisión” del programa solo puede llegar a

constar de un máximo de 8 hojas. Aunque esa mejora aumentaría las prestaciones del jugador, en el transcurso de una partida normal la proporción de jugadas cuya decisión diferiría de la actual sería baja puesto que el conjunto de casos en los cuales necesitaríamos recurrencia no es excesivamente alto.

Las principales dificultades de este trabajo fueron, por un lado la creación de una interfaz gráfica intuitiva y fácil de manejar e interpretar y, por otro lado, los algoritmos no solo de juego razonado si no los de control de la corrección de movimientos y control de el cumplimiento de las reglas de juego en general.

Las posibles ampliaciones del trabajo podrían ser la elección de los colores con los cuales el jugador desea jugar, aumentar la profundidad en el razonamiento de la máquina así como la posibilidad de elegir el grado de dificultad del programa.

Sería posible también dar al usuario la posibilidad de elegir la “personalidad” del rival contra el cual quiere jugar. Como se ha mencionado, las reglas de evaluación son heurísticas y por tanto no todo el mundo aplica las mismas reglas, o no le da tanta importancia a unos términos estratégicos como a otros. Hay personas más agresivas en su juego y otras mas conservadoras, creo que sería sencillo modelar este comportamiento, a elección del usuario, si mas que cambiar ligeramente la ponderación que se aplica a determinadas reglas de evaluación.

4.REFERENCIAS.

[1]Apuntes de la asignatura:

<http://www.it.uc3m.es/jvillena/irc/descarga.htm?url=material/02.pdf>

[2]Apuntes de la asignatura “Laboratorio de programación”:

<http://www.it.uc3m.es/tlp/materiales/interfacesGraficasII.pdf>

[3]API de java:

<http://java.sun.com/j2se/1.4.2/docs/api/>

[4]Addison-Wesley-*Data Structures and Algorithm Analysis in Java*