

Alfredo González Gaviña

Presenta:

Tarea 2

[8 Reinas]

Materia:

[IA]

# Introducción

El juego consiste en llenar cada fila y cada columna del tablero de ajedrez con una reina, pero ninguna debe atacarse entre sí. Se diría que es algo sencillo si se piensa en llenar el tablero con reinas siguiendo un patrón de movimiento de caballo (L) o bien llenar primero las casillas impares y después las pares. El 1er método es capaz solamente de llegar a la 4ta reina al intentar colocar la 5ta reina el tablero ya está completamente ocupado, con el 2do caso se consigue llegar hasta la 7ma reina sin embargo al intentar colocar la 8va reina el tablero se ha ocupado totalmente. Se demuestra de esta forma la complejidad que puede alcanzar este aparente simple juego.

# Metodología

Para solucionar este problema se utilizó el método de backtrackin o como se dijo en clases pasadas “barriendo el cuarto”, de forma sintetizada y aplicada al problema este método consiste en lo siguiente

Si se puedo colocar una reina en la posición indicada(su valor sea igual a “\_” ) se devuelve 1 y se suma al total de reinas para aumentar la cantidad de reinas, se verifica que la casilla que se intenta ocupar este libre una vez ocupada las casillas adecuadas son atacadas, adquieren valor igual a “#”

Si no se verifica que la casilla siguiente este libre, en caso de no estarlo se consigue la siguiente casilla y se hace lo mismo hasta terminar toda la fila, si toda la fila está ocupada se regresa -1 y se suma al total de reinas para decrecer la cantidad, el programa sube hasta la fila anterior y el proceso se repite hasta regresar -1 o 1

….n-1

….n-1

….n-1

#

#

#

Si la fila está llena se regresa -1 y se sube a la columna anterior, se busca a la reina y se intenta mover de casilla

R

Si se puede colocar se ataca y se regresa 1, se pasa a la fila siguiente

-1

+1

Figura 1.1 diagrama sintetizado del algoritmo de solución

Para esto se empleó C++ dada su rapidez, claridad y el dominio adquirido en este lenguaje de programación, así pues se crearon 2 clases la clase jugador y la clase casilla la primera abstrae al jugador quien puede ejecutar operaciones sobre el tablero, esta clase cuenta con 12 métodos y un constructor por defecto, carece de propiedades dado que por el momento eran innecesarias. La clase casilla abstrae una casilla del tablero cuenta con 2 propiedades y un constructor por defecto.

# Resultados

El programa resuelve de forma satisfactoria el problema planteado en un tiempo muy corto (cuestión de segundo) el estado de solución puede variar dependiendo de la posición inicial indicada por el usuario, es decir la configuración del tablero puede llegar a variar dependiendo de qué casillas se le den como primer movimiento.

A continuación se muestran los resultados:

|  |  |
| --- | --- |
| Estado de inicio  (coordenada) | Estado de salida |
| (3,4) |  |
| (6,7) |  |

# Análisis

El problema ciertamente es más complejo de lo que aparenta y fueron necesarios varios análisis para dar con la solución correcta a continuación se da detalle de estos análisis

En un principio se intentó utilizar métodos recursivos para solucionar el problema, sin embargo la recursividad es una herramienta muy compleja y muy difícil de depurar dado que se debe seguir una buena parte de la pila de llamadas del programa por esta razón la recurvada se descartó y se implementó una solución iterativa siendo está más simple e incluso más rápida que la ya mencionada.

Si se aprecia con claridad, la solución del problema no supone una clase casilla, la inclusión de esta clase resulta innecesaria sin embargo se optó por conservarla para no perder tiempo en el desarrollo del programa

La clase casilla es en realidad una estructura de datos, se optó por usar una estructura para no perder tiempo en declarar métodos amigos (operador friendo)

Algunos métodos de la clase casilla fueron suprimidos tal como el medo “Indexa que fue eliminado por ser inútil.

La clase jugador es indispensable, es la encargada de empaquetar todos los métodos que puede realizar un jugador, mover una pieza quitarla y ver el tablero, resulta realmente útil

# Conclusión

La inteligencia Artificial es una muy amplia al contrario de lo que la mayoría de personas se imagina no se trata de crear maquinas brillantes ni sistemas capaces de exterminarnos, como Hollywood nos ha hecho creer la inteligencia artificial puede llegar a ser (por el momento) un algoritmo que sea capaz de resolver un problema lógico como el juego de las 8 reinas, que hay que decirlo ni siquiera algunos humanos son capaces de resolver a mano, la inteligencia artificial no depende de un único lenguaje de programación, en este caso se usó c++, pero se puede traducir a cualquier otro lenguaje de programación, en este “Arte” como es nombrado por algunos autores lo verdaderamente importante es entender que se debe de hacer y cómo se debe de hacer .