

Datos generales

El driver de usuario tiene un usuario ya iniciado(admin-contraseña:1234) para poder comprobar el funcionamiento del driver. Aqui se pueden jugar las diferentes partidas, añadir problemas a la base de datos y ver y editar estos problemas. Se adjunta un archivo problemasSistema.txt con varios FEN con los que hemos probado el algoritmo. El mismo driver tiene la opcion de cargarlos en tiempo de ejecucion siguiendo sus instrucciones.

El driver de problema solamente comprueba la gestion de problemas, con la comprobacion de si el problema (FEN) tiene solución.

Algoritmo

El algoritmo utilizado para ver si los problemas tiene solucion, se basa en el algoritmo "minimax" recursivo, donde se busca recursivamente una opcion de mate de entre todos los movimientos posibles, en los n passos. Este algoritmo es de backtracking y en su version simple no se recorta el arbol de recursividad. En esta version la unica euristica que utilizamos es binaria, sobre si existe jaque mate en ese estado del algoritmo.

Para realizar la accion de mover de la IA, se utiliza el mismo algoritmo.

Estructuras de datos y demás

En los drivers hemos utilizado el metodo de class.getMethod(name) para no poner el tipico switch-case, mejorando un poco la lectura del problema. Hemos seguido los patrone de diseño singleton y factory(solo la idea) para mejorar el diseño del sistema. Las estructuras de datos que hemos utilizado han sido la ArrayList, TreeSet y TreeNode(que encontramos por internet). Esta ultima nos facilita el algoritmo