Departamento de Computación
FCEFQyN, Universidad Nacional de Río Cuarto
Asignaturas: Diseño de Algoritmos - Algoritmos II
Primer Cuatrimestre de 2023

Trabajo Práctico No. 2

La resolución de este trabajo práctico debe ser completada a través de GitHub Classroom, antes de las **23:55 del lunes 5 de Junio de 2023**.
El código provisto como parte de la solución al problema planteado deberá estar documentado apropiadamente **usando Javadoc**. Aquellas soluciones que no requieran programación, como así también la documentación adicional de código que se desee proveer, debe entregarse en **archivos de texto convencionales**, en el mismo repositorio de entrega de la solución, con nombres que permitan identificar fácilmente su contenido.
Tanto la calidad de la solución, como el código y su documentación serán considerados en la calificación. Recuerde además que los trabajos prácticos **tienen defensa**, y que el trabajo en la resolución del problema debe realizarse de manera equitativa entre los miembros de cada grupo. Los grupos deben tener 3 integrantes.

1. *Trap the Cat*, es un entretenido juego de ingenio que se ha vuelto muy popular por su simplicidad. Fue lanzado en enero de 2019 por Reinout Stevens. El objetivo de este juego es evitar que un gato negro escape del campo de juego. Para ganar el juego se debe cercar al gato, marcando los lugares libres por los que puede avanzar hasta que el gato no pueda moverse. El gato gana si logra escapar por alguno de los bordes del tablero. Como referencia, puede analizar una versión del juego en: `https://trapthecat3.com/`

   La pantalla del juego contiene un tablero con 121 hexágonos y un gato negro. Cada hexágono tiene tres estados posibles:

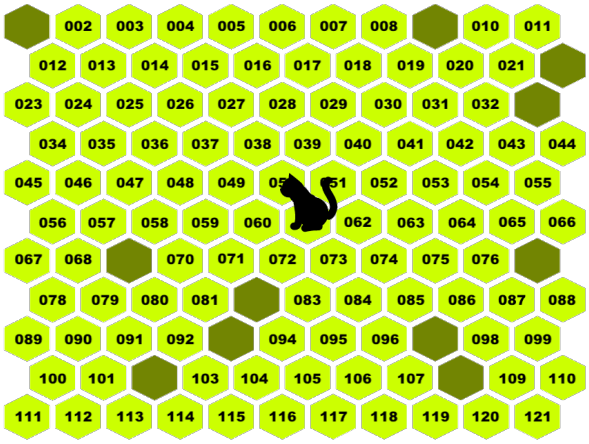
   - libre.
   - bloqueado o marcado.
   - ocupado por el gato.

   El gato solo puede avanzar de a **UN** paso a la vez, hacia cualquiera de las celdas **adyacentes** libres, y el usuario puede marcar(bloquear) cualquiera de los hexágonos libres del tablero.

   Cada hexágono tiene como máximo 6 adyacentes donde puede moverse el gato:

   El juego comienza con el *gato* jugando en el centro del tablero y algunos (no más de 15) hexágonos bloqueados, aleatoriamente. Por ejemplo, un tablero inicial posible es uno que contiene el gato en la posición central, los hexágonos $1, 9, 22, 33, 69, 77, 82, 93, 97, 102$ y 108 bloqueados y el resto de los hexágonos libres. Siempre comienza jugando quien quiere atrapar al gato, en su turno, el jugador, puede bloquear cualquier celda libre del tablero y luego pasa el turno al gato, que podrá moverse a cualquier celda adyacente libre si la hubiera.
   En la siguiente figura se puede visualizar el tablero inicial descripto en el párrafo anterior:



El juego termina cuando el gato no puede moverse hacia ninguna de las celdas adyacentes, en cuyo caso el ganador es el jugador o cuando el gato llega a cualquier hexágono ubicado en alguno de los bordes del tablero, en este caso gana el gato.

Se desea desarrollar una aplicación interactiva que permita jugar a *Trap the Cat*, con dos modos diferentes:

- Humano-Gato: en esta opción, uno de los jugadores es controlado por una persona y el gato controlado por la computadora.
- Computadora-Gato: en esta opción el juego se desarrolla automáticamente y la computadora deberá tomar las desiciones que mejor le convengan dependiendo del jugador que le toque representar (el jugador es representado por "max", mientras que el gato corresponde a "min")).

Deberán modelar e implementar una solución al problema planteado utilizando la ténica de búsqueda para problemas con adversarios vista en la materia. Para la parte algorítmica del motor de búsqueda se requiere que implementen *MiniMax con poda alfa-beta*. A tal fin se proveen clases e interfaces que deberán usar, y respetar como parte del ejercicio. La función de valoración de estados necesaria para la implementación de la técnica quedará a criterio de los desarrolladores.
No es necesario soportar una interfaz gráfica, es suficiente con una interfaz de consola. Tanto la calidad de diseño de la solución como la capacidad del programa para jugar a *Trap the Cat*, serán analizados como parte de la evaluación. Su solución debe incluir descripción de la representación elegida para el problema, e instrucciones de uso de la aplicación.
Para resolver este problema, deben emplearse las clases e interfaces provistas como parte del ejercicio.