Programación Lógica

Obligatorio 2 – 2013

Infectlog

Facultad de Ingeniería
Instituto de Computación
Grupo de Procesamiento de Lenguaje Natural

El objetivo de este obligatorio es implementar el juego Infection en Prolog.

Nota previa - IMPORTANTE

Se debe cumplir íntegramente el "Reglamento del Instituto de Computación ante Instancias de No Individualidad en los Laboratorios", disponible en:
`http://www.fing.edu.uy/inco/pm/uploads/Ense%f1anza/NoIndividualidad.pdf`

En particular está prohibido utilizar documentación de otros grupos o de otros años, de cualquier índole, o hacer público código a través de cualquier medio (news, correo, papeles sobre la mesa, etc.).

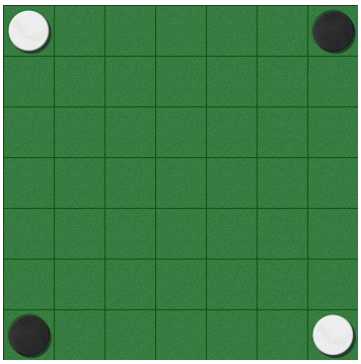
Juego Infectlog

El juego *Infectlog* está basado en el juego Infection [1] (también conocido como Ataxx), un juego de mesa para dos jugadores creado en 1988. Se juega con fichas blancas y negras sobre un tablero cuadrado, generalmente de 7x7, aunque existen variantes con distinto tamaño y forma de tablero, o diferente cantidad de jugadores.

A continuación se detallan las reglas de este juego.

Disposición del tablero

Inicialmente el tablero se dispone de la siguiente manera: hay dos fichas blancas y dos fichas negras. Las fichas blancas se ubican en los casilleros superior izquierdo e inferior derecho del tablero. Las fichas negras se ubican en los casilleros superior derecho e inferior izquierdo del tablero.

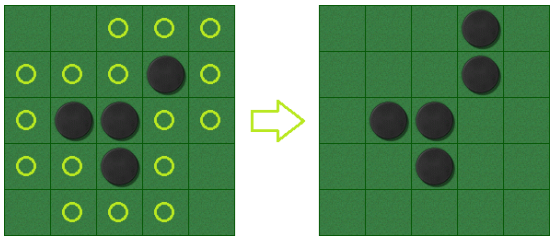


Tablero inicial para un juego de tamaño 7

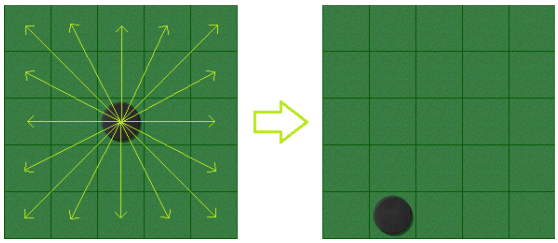
Dinámica del juego

Comienza moviendo el jugador que utiliza las fichas blancas y luego se procede por turnos. En cada turno, un jugador puede realizar uno de los dos movimientos siguientes:

- Colocar una pieza de su color adyacente a otra de su color (*clonar*).
- Mover una pieza de su color a una posición que esté a una distancia de dos casilleros de la posición original (*saltar*). La distancia se mide moviéndose dos casilleros en cualquier dirección, incluyendo diagonales (ver figura).

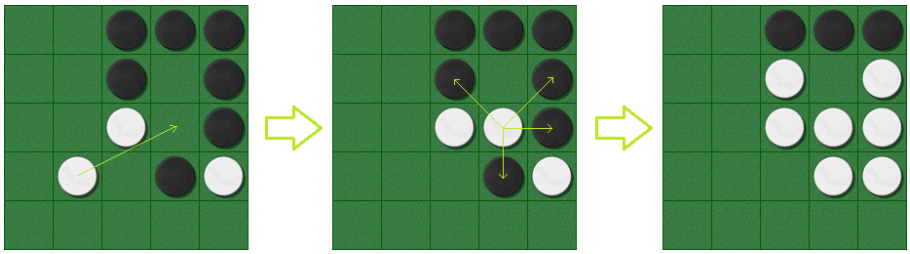


Posibles posiciones para clonar, y resultado del tablero luego de clonar



Posibles posiciones para saltar, y resultado del tablero luego de saltar

Luego de realizar cualquiera de los dos movimientos, se deben observar los ocho casilleros que se encuentran alrededor de la nueva ficha (la posición final en la que saltó, o la posición de la nueva ficha clonada). Todas las fichas que se encuentren en cualquiera de estos ocho casilleros pasan a tener el color del jugador que acaba de mover (a esta fase se le conoce como *infectar*).



En este ejemplo la ficha blanca salta, e inmediatamente infecta cuatro fichas negras que quedaron adyacentes, convirtiéndolas en blancas

En ese momento se termina el turno y el otro jugador debe realizar las mismas acciones utilizando las fichas de su color. En caso de que no tenga ningún movimiento disponible, debe saltear su turno y vuelve a jugar el otro jugador.

Fin del juego

El juego continúa hasta que ocurre alguna de las siguientes posibilidades:
- No hay más casilleros libres en el tablero.
- Uno de los dos jugadores se quedó sin fichas en el tablero.

En cualquiera de los dos casos se declara el fin del juego y el jugador que tenga más fichas de su color en el tablero es declarado como ganador. Si se está jugando con un tablero de tamaño par, es posible que ambos jugadores terminen con la misma cantidad de fichas, en ese caso se declara un empate.

Requerimientos a implementar

Se debe implementar en Prolog el juego de *Infectlog* para dos jugadores, el cual debe seguir la mecánica descrita y cumplir con las siguientes características:

Al inicio del juego se le debe preguntar al usuario:

- Qué tipo de jugador controlará las fichas blancas y qué tipo de jugador controlará las fichas negras. Los tipos de jugadores pueden ser: *humano* o *máquina*.
- Cuál es el tamaño del tablero. El tamaño mínimo del tablero es 5x5, el tamaño máximo es 9x9.

Manejo de la máquina

Para los jugadores de tipo *máquina* se debe implementar una inteligencia artificial con una estrategia *buena* y *eficiente*. Se sugiere utilizar el algoritmo de *minimax* [2] para implementar la estrategia de la máquina. La cantidad de niveles a utilizar en el algoritmo debe tomarse como un parámetro del sistema (ver el predicado **`minimax_depth/1`** en **`infectlog.pl`**).

Una vez que se eligió el movimiento de la máquina, se debe mostrar un mensaje detallando dicho movimiento en la barra de estado, luego se debe efectuar el movimiento y su correspondiente fase de *infectar*. Los mensajes detallando el movimiento deben ser como los siguientes:

- *Jugador negro clona en 4,3*
- *Jugador blanco mueve de 4,2 a 4,4*

Manejo del humano

Los jugadores de tipo *humano* deben esperar comandos del usuario. Cuando le toca el turno a un humano, las posibilidades son las siguientes:

- Si hace click en un casillero vacío adyacente a una ficha de su color, se realiza un movimiento de tipo *clonar*.
- Si hace click en un casillero con una ficha de su color, se debe recordar la ficha que acaba de clickear. Luego se debe esperar un segundo click, con las siguientes opciones:
  - Si hace click en un casillero que está a distancia 2 de la ficha original, se realiza un movimiento de tipo *saltar*.
  - Si hace click en un casillero vacío adyacente a la ficha original, se realiza un movimiento de tipo *clonar*.
  - Si hace click en otra ficha de su color, se selecciona esta nueva ficha y pasa de nuevo a esperar un segundo click.
  - Si hace click en cualquier otro lado, se cancela el movimiento de tipo saltar y espera un nuevo click (que puede ser clonar o saltar).
- En cualquier otro caso, se muestra en la barra de estado un mensaje de error, y se vuelve a esperar el siguiente click.

Luego de realizar el movimiento, se reliza la fase de *infectar* correspondiente.

Solo se debe permitir que el humano realice movimientos válidos según las reglas del juego. Una vez que se realiza un movimiento permitido, se debe mostrar en la barra de estado un mensaje detallando el movimiento, en el mismo formato que para la máquina.

Fin del juego

Cuando se cumple alguna de las condiciones de fin de juego (detalladas en la sección anterior) se debe desplegar un mensaje modal con la siguiente información:

- Jugador que ganó (o empate).
- Cantidad de fichas del jugador blanco.
- Cantidad de fichas del jugador negro.

El diálogo debe preguntarle al usuario si desea jugar de nuevo. En caso afirmativo se debe reiniciar el juego preguntándole de nuevo los parámetros para jugar. En caso negativo se debe finalizar la ejecución.

Insumos

Se proveen los siguientes módulos Prolog:
- `graficos.pl`: Contiene todos los predicados para el manejo gráfico y de interacción con usuario. Se recomienda leer la especificación de los predicados exportados por este módulo para entender el funcionamiento de las funcionalidades gráficas.
- `infectlog.pl`: Módulo principal de la solución. Se provee una implementación básica que muestra los principios de interacción con `graficos.pl`.

Observaciones

La implementación debe realizarse de manera que pueda ser ejecutada en la plataforma SWI-Prolog.

Forma de entrega

La entrega se realizará a través del espacio eva del curso, en una página destinada a tal fin que se habilitará cerca de la fecha de entrega. Se debe entregar un solo archivo '`grupo##.zip`', donde ## es el número del grupo que realiza la entrega, conteniendo todos los módulos de la solución y el informe de la misma, ítems indicados en el apartado **Entregable**.

Fecha de entrega

Los trabajos deberán ser entregados siguiendo el procedimiento descrito anteriormente antes del lunes 10/06/2013, sin excepciones. No se aceptará ningún trabajo pasada la citada fecha.

Entregable

El archivo a entregar debe contener:
1. Implementación del juego Infectlog, con todos los módulos necesarios para la ejecución del juego. Incluir también el módulo `graficos.pl` y los archivos de imágenes.
2. Informe breve (5 páginas) en formato pdf detallando la estructura de los módulos, los predicados principales, las decisiones de diseño tomadas, heurísticas utilizadas, consideraciones de eficiencia y problemas encontrados.

Referencias

[1] http://en.wikipedia.org/wiki/Ataxx

[2] http://en.wikipedia.org/wiki/Minimax