PROYECTO DE PROGRAMACIÓN DECLARATICA AVANZADA

UNIVERSIDAD COMPLUTEMSE DE MADRID

24 de Junio del 2014

Sergio Cabrerizo Galán

Laura Aragón Tebar

Autorización

Los abajo firmantes autorizan a que se publique el proyecto que han realizado para la asignatura de PROGRAMACIÓN DECLARATIVA AVANZADA de forma que podrá estar online y accesible para terceros, bajo las siguientes licencias: el software liberado bajo la licencia BSD (*) y la memoria, documentación, gráficos u otro contenido liberado bajo la licencia Creative Commons Atribución 4.0 Internacional (**). La autoría será respetada en todos los casos.

(*): http://directory.fsf.org/wiki/License:BSD_3Clause
(**): https://creativecommons.org/licenses/by/4.0/deed.es

Laura Aragón Tebar	Sergio Cabrerizo Galán
71227764F	50880445Y

Madrid, 24 de Junio de 2014

Índice

Autorización I
Capítulo 1: Introducción 1
1.1 Descripción general 1
1.2 Motivación 1
1.3 Estado del arte 1
Juego Morris de tres fichas 1
Juego Atrapados 2
Juego El gato y los ratones 2
Capitulo 2: Desarrollo 5
2.1 Tablero de cuadrantes 5
2.2 Tablero de direcciones 5
2.3 Funciones implementadas 6
2.4 Heurística 9
Heurística para el gato 9
Heurística para el jugador 9
Capitulo 3: Conclusiones y trabajos futuros 11
Bibliografía 13
Anexo I: Manual de usuario 15

Capítulo 1: Introducción

En este capítulo introductorio se muestra la motivación de este proyecto, así como el estado del arte relativo a las aplicaciones ya desarrolladas.

1.1 Descripción general

El juego de encerrar al gato es una versión del juego Chat Noir (http://www.gamedesign.jp/flash/chatnoir/chatnoir.html), cuyo objetivo consiste en encerrar al gato dentro del tablero, marcando las distintas casillas libres.

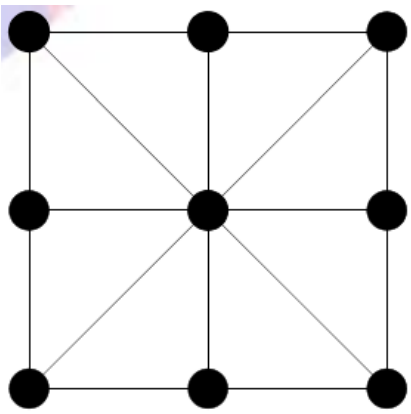
1.2 Motivación

Hay mucha variedad en los juegos de mesa, desde los clásicos como los jeroglíficos o puzzles a los más actuales como los sudokus.
Sin embargo, el proyecto desarrollado trata de un juego de mover piezas, semejante al de ajedrez, juegos de comer piezas o el cubo de Rubik.

El presente juego se encuentra encuadrado en la categoría de juegos de lógica, a través de los cuales se desarrolla el intelecto, se pueden observar los problemas desde una perspectiva nueva y original, y ayudan a afrontar los problemas de la vida.
Por ello, este es el centro de interés que nos ha llevado a desarrollar este proyecto.

1.3 Estado del arte

Juego Morris de tres fichas



Este juego también recibe el nombre de morris de los tres hombres o los nueve agujeros.

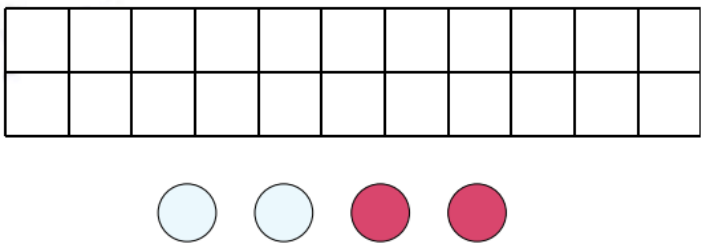
Objetivo

Cada jugador tiene como objetivo colocar sus fichas en una misma línea recta (horizontal, vertical o diagonal).

Reglas del juego

- Se echa a suertes el jugador que empieza a jugar la partida.
- Cada jugador coloca una de sus fichas de forma alternativa.
- Cuando están todas las fichas sobre el tablero, los jugadores mueven por turnos una ficha a una posición libre adyacente, a lo largo de una línea.
- Cada vez que le toca el turno, el jugador está obligado a mover una de sus fichas.
- Gana el jugador que consigue colocar sus tres fichas en línea recta (horizontal, vertical o diagonal).

Juego Atrapados



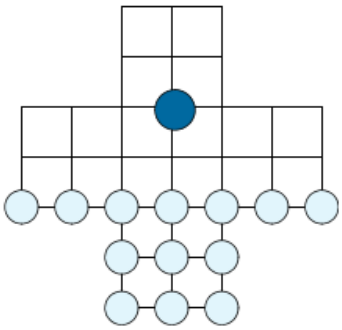
Objetivo

El objetivo del juego es impedir el movimiento del contrario.

Reglas del juego

- Se sortea el jugador que juega primero.
- La posición inicial es la siguiente:
- Cada jugador, por turno, mueve una de sus fichas en cada jugada.
- Cada jugador mueve una ficha de su color tantas casillas como quiera a lo largo de la fila donde está colocada.
- No está permitido saltar por encima de una ficha de otro color.
- Gana el jugador que consigue impedir el movimiento de todas las fichas del jugador contrario.

Juego El gato y los ratones



Objetivo

El objetivo del juego consiste para los ratones en bloquear al gato, y para el gato comerse a todos los ratones.

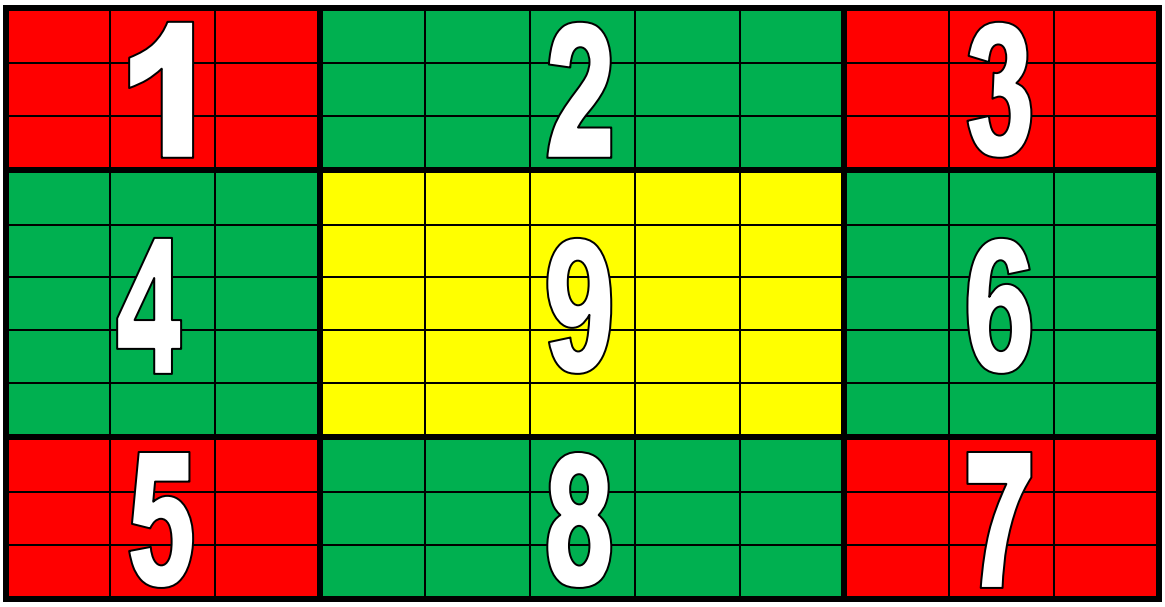
Reglas del juego

- Se echa a suertes el jugador que mueve la ficha del gato, que será el primero en mover y el que mueve las fichas de los ratones.
- Tanto las fichas de los ratones como la del gato se mueven a posiciones adyacentes siempre que estén vacías.
- El gato captura a los ratones saltando por encima de ellos a una casilla vacía. También podrá capturar más de un ratón en un mismo movimiento, encadenando varios saltos seguidos.
- Los ratones no pueden saltar sobre el gato ni tampoco capturarlo.
- El gato gana si captura diez ratones y los ratones ganan si acorralan al gato impidiendo que pueda moverse.

Capitulo 2: Desarrollo

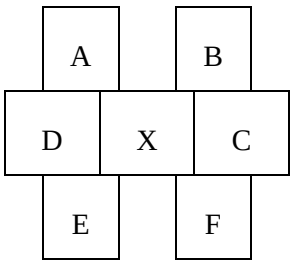
En este capítulo describiremos como se ha desarrollado el proyecto, analizando las distintas funciones usadas, la heurística, así como otros elementos tenidos en cuenta a la hora del desarrollo.

2.1 Tablero de cuadrantes



En esta imagen podemos observar la distribución, en cuadrantes, que hemos realizado del tablero para el cálculo de la heurística. La distribución ha sido realizada de forma que cada cuadrante tiene 5 bordes de tablero y un cuadrante central al cual no daremos mucha prioridad debido a que no es un cuadrante realmente crítico como lo son el resto.

2.2 Tablero de direcciones



En esta imagen reflejamos las como hemos distribuido las casillas de alrededor de una posición.

2.3 Funciones implementadas

Las funciones implementadas para el desarrollo del proyecto son las siguientes:

- *inicio*: En esta función se escribe el titulo del proyecto, el jugador elige el modo de juego y el rol, se dan unas breves instrucciones que como jugar, se crea el tablero de juego.
- *elegirModo(-Int)*: esta función nos devuelve el modo de juego que el jugador haya seleccionado.
- *escribirModos*: esta función escribe en pantalla los diferentes modos que el jugador puede seleccionar.
- *seleccionModo(-Int)*: esta función recoge el modo de juego que el jugador haya seleccionado y verifica que sea correcto.
- *uno_o_dos(+Int)*: comprueba que el modo introducido por el jugador sea 1 o 2.
- *explicarJuego(+Term)*: dado el modo de juego del jugador muestra por pantalla una breve explicación de cómo se juega.
- *mostrarTablero*: muestra por pantalla como está estructurado el tablero de juego.
- *crearTablero(+Int,-List)*: dado el tamaño del tablero crea una lista de listas, donde cada elemento de la lista es una fila del tablero.
- *crearFila(+Int,+List,-List)*: crea una lista, que corresponde con fila de nuestra tablero, e inserta tantos ceros como indique el tamaño del tablero.
- *ponerGato(+List,-List)*: coloca un dos (el gato) en la posición inicial de este (casilla 5)
- *colocarFichaFila(+Int,+Int,+Int,+List,-List)*: dado el jugador y la fila donde se quiere colocar la ficha, y nos devuelve el tablero con la nueva ficha colocada.
- *colocarFichaColumna(+Int,+Int,+List,-List)*: dado la columna donde se quiere colocar la ficha nos devuelve el tablero con la nueva ficha colocada.
- *mostrarTableroColores(+Int,-List)*: dado un tamaño y un tablero muestra por pantalla el tablero con colores (Verde, casillas libres, Rojo, casillas ocupadas por el jugador y Azul, casilla donde se encuentra el gato)
- *mostrarFilaColores(+List)*: dada una lista muestra una fila del tablero en colores.
- *juego(+Int,+Int,+Int,+List)*: dado el tamaño del tablero, el modo de juego, rol del jugador y el tablero, comienza el juego (Jugador vs Jugador, Maquina(Gato) vs Jugador o Maquina(Jugador) vs Gato).
- *jugador(+Int,+Int,+List,-List)*: recoge por pantalla la fila y la columna del jugador para posteriormente colocarla en el tablero.
- *colocarFicha(+Int,+Int,+Int,+List,-List)*: dado un jugador y la casilla donde se quiere colocar devuelve el tablero con el jugador colocado en la casilla indicada.

- *comprobarMovimiento(+Int,+Int,+Int,+Int,+List)*: dado una casilla del tablero (fila y columna) y el tipo de jugador que el movimiento que se quiere realizar es correcto o no para dicho jugador.
- *movimientoGatoValido(+Int,+Int,+Int,+Int)*: comprueba que el movimiento que se desea realizar es válido para el rol de gato.
- estaLibreFila(+Int,+Int,+List) y estaLibreColumna(+Int,+Int,+List): nos indica si una casilla (fila y columna) tiene como elemento del tablero un cero lo cual indica que esta vacía
- *buscarGato(+Int,+List,-Int,-Int)*: dado el tablero busca la casilla donde se encuentra el gato (fila y columna).
- *estaGatoColumna(+Int,+Int,+List,-Int)*: esta función devuelve la columna donde se encuentra el gato.
- *estaGatoFila(+Int,+Int,+List,-Int)*: esta función nos devuelve la fila donde se encuentra el gato.
- *eliminarGato(+Int,+Int,+List,-List)*: dada la casilla donde se encuentra el gato y el tablero elimina al gato de dicha casilla y la pone a cero (vacía).
- *seguirjugando(+Int,+List)*: Comprueba que si el gato ha sido acorralado (gana el jugador) o el gato ha llegado a un borde del tablero (gana el gato) y lo muestra por pantalla.
- *comprobarAlrededorGato(+Int,+Int,+List)*: dado la casilla donde se encuentra el gato y el tablero comprueba si el gato no puede realizar más movimientos.
- *volverJugar*: esta función muestra por pantalla si se quiere volver a jugar.
- *seleccionRespuesta*: esta función recoge por pantalla la respuesta de si se quiere seguir jugando.
- *maquinajugador(+Int,+Int,+List,-List)*: esta función dado el modo de juego y el tablero si el modo es dos juega como el gato y si uno juega como jugador.
- *obtenerCuadrante(+Int,+Int,-Int)*: dada una casilla nos devuelve el cuadrante donde se encuentra casilla.
- *obtenerMejorCuadrante(+List,+Int,-Int)*: dado el tablero y cuadrante donde se encuentra el gato nos devuelve el mejor cuadrante para el movimiento del jugador según la heurística desarrollada.
- *obtenerHeuristicaCuadrante(+List,+Int,+Int,+Int,+Int,+Int,-Int)*: dado la casilla de inicio y final de un cuadrante y el cuadrante del gato nos devuelve la mejor heurística de todos los cuadrantes.
- *calcularHeuristica(+Int,+Int,+Int,+Int,+Int,-Int)*: esta función calcula la heurística basados en los parámetros elegidos.

- *calcularDiferenciaCuadrates(+Int,+Int,-Int)*: dados dos cuadrantes nos devuelve un porcentaje en función de su cercanía.
- *moverGato(+List,+Int,+Int,+Int,-List)*: dado el tablero y el mejor cuadrante nos devuelve el tablero con el gato en la mejor posición.
- *obtenerCasillaDestinoFinal(+List,+Int,+Int,-Int,-Int)*: dado el tablero y un cuadrante nos devuelve el barde más cercano a este.
- *obtenerCasillaMejorCuadrante(+List,+Int,-Int,-Int)*: dado el tablero y un cuadrante nos devuelve la mejor casilla dentro de dicho cuadrante.
- *obtenerMejorCasillaCuadrantePar(+List,+Int,+Int,+Int,+Int,-Int,-Int)*: esta función devuelve la mejor casilla dentro de un cuadrante par.
- *obtenerMejorCasillaCuadranteImpar(+List,+Int,+Int,+Int,+Int,+Int,+Int,-Int,-Int)*: esta función devuelve la mejor casilla dentro de un cuadrante impar.
- *obtenerCasillaDestino(+List,+Int,+Int,+Int,+Int,-Int,-Int)*: dado una casilla busca la casilla libre más próxima a dicha casilla.
- *obtenerDireccionMovimiento(+Int,+Int,+Int,+Int,-Int)*: dada una casilla y la casilla donde se encuentra el gato nos devuelve la dirección donde se debe desplazar.
- *obtenerMovimientoJugador(+List,+Int,+Int,+Int,-Int,-Int)*: dada la casilla del gato nos devuelve la casilla donde debe situarse el jugador.
- obtenerDireccionMovimiento2(+Int,+Int,+Int,+Int,-Int): dadas dos casilla nos devuelve la dirección.

2.4 Heurística

Heurística para el gato

La heurística del gato consiste en comprobar en función de su posición a donde debe desplazarse, para ello si está en la fila o columna de al lado de un borde, primero mira si desde su posición es capaz de alcanzar un borde para finalizar la partida ganando. En caso de que no se encuentre en esas filas o columnas o bien no pueda alcanzar ningún borde libre, la heurística la calculamos obteniendo cual es el mejor cuadrante al que puede desplazarse teniendo en cuenta, las casillas del borde libres, las casillas ocupadas de cada cuadrante así como su distancia a ese cuadrante. Todo ello ha sido ponderado para imponer más peso a las condiciones que nos parecían mas decisivas a la hora de tomar la decisión.

$$Heurística = \left(\frac{ocupados}{tamaño} * 0.2\right) + \left(\frac{libres}{tamaño} * 0.5\right) + \left(\frac{1}{distancia} * 0.3\right)$$

Una vez obtenido el mejor cuadrante al cual puede desplazarse el gato, obtenemos de este una casilla de borde libre y calculamos la dirección que debe seguir para llegar a él.

Heurística para el jugador

La heurística del jugador consiste en a partir del movimiento del gato, teniendo en cuenta su posición y la posición a la que se mueve, calculamos cual ha sido la dirección de este movimiento y procedemos a tapar la siguiente casilla si siguiese ese movimiento

Capitulo 3: Conclusiones y trabajos futuros

Debido al poco conocimiento del juego, no hemos encontrado información acerca de este, tan solo webs con el juego para poder jugar a él. Por este motivo hemos tenido que desarrollar el juego desde cero sin tener ninguna referencia acerca de estrategias de cómo enfrentarnos al juego ni heurística alguna respecto a él. Respecto a las estrategias de juego se planteo una basada en un algoritmo minimax, esta fue descartada por el gran numero de nodos que se deberían explorar debido a que al ser un tablero de 11x11 a poco que poníamos un poco de profundidad en el árbol, el numero de nodos era demasiado grande y planteamos cambiarlo a realizar una heurística mediante una función que diseñamos por nuestra cuenta.

Como trabajos futuros se podría plantear la realización de un mayor estudio y mejora de la función de heurística para hacer al juego más complejo y que tuviese más dificultades a la hora de ganar.

Bibliografía

[1] Autor: *Universitat Jaume I de Castelló* [en línea]. [Consulta: Junio 2014].
http://repositori.uji.es/xmlui/bitstream/handle/10234/7093/LibroJuegos.pdf?sequence=1

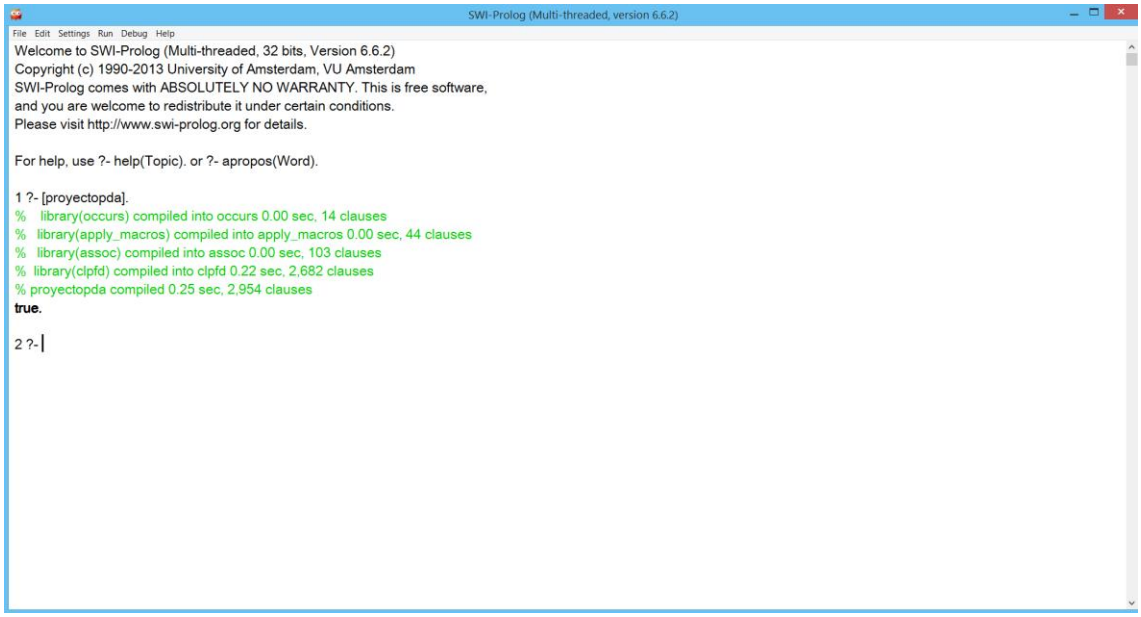
[2] Autor: swi-prolog 7.1.16 [en línea]. [Consulta: Junio 2014]
http://www.swi-prolog.org/

[3] Autor: GAMEDESIGN [en línea]. [Consulta: Junio 2014]
http://www.gamedesign.jp/flash/chatnoir/chatnoir.html

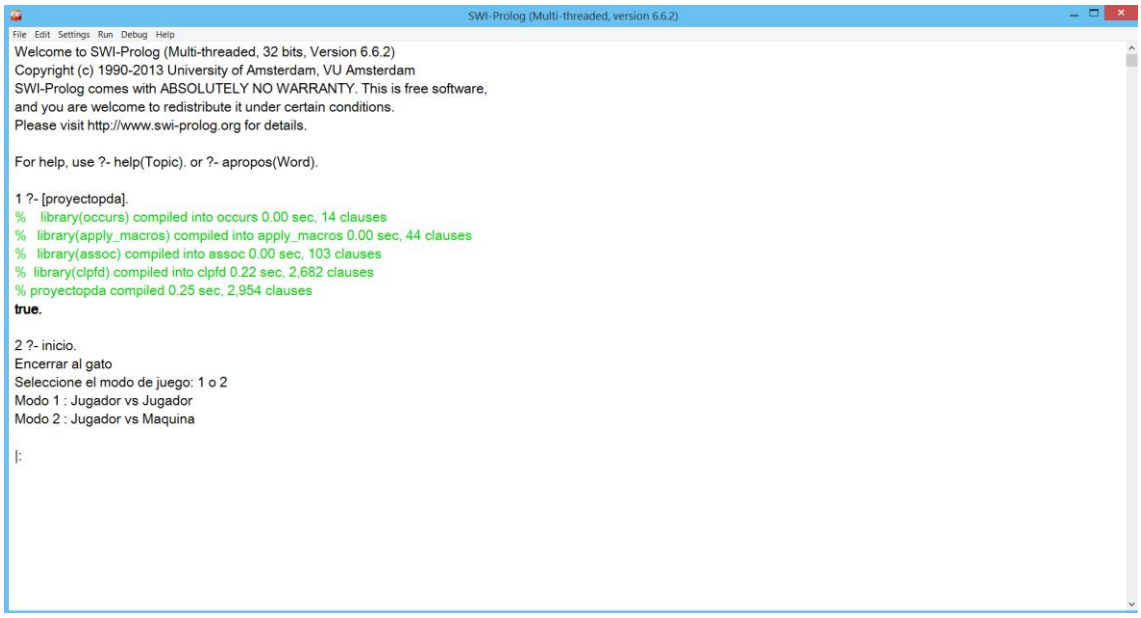
Anexo I: Manual de usuario

En este anexo presentamos un breve manual de la aplicación, en el podemos ver en algunas capturas de pantalla el tablero de juego en el cual, el color verde representa las casillas libres, el color rojo las fichas del jugador (que encierra al gato) y en color azul al gato.

El usuario deberá abrir el compilador de Prolog e introducir la secuencia “[proyectopda].” para cargar el programa en el compilador.



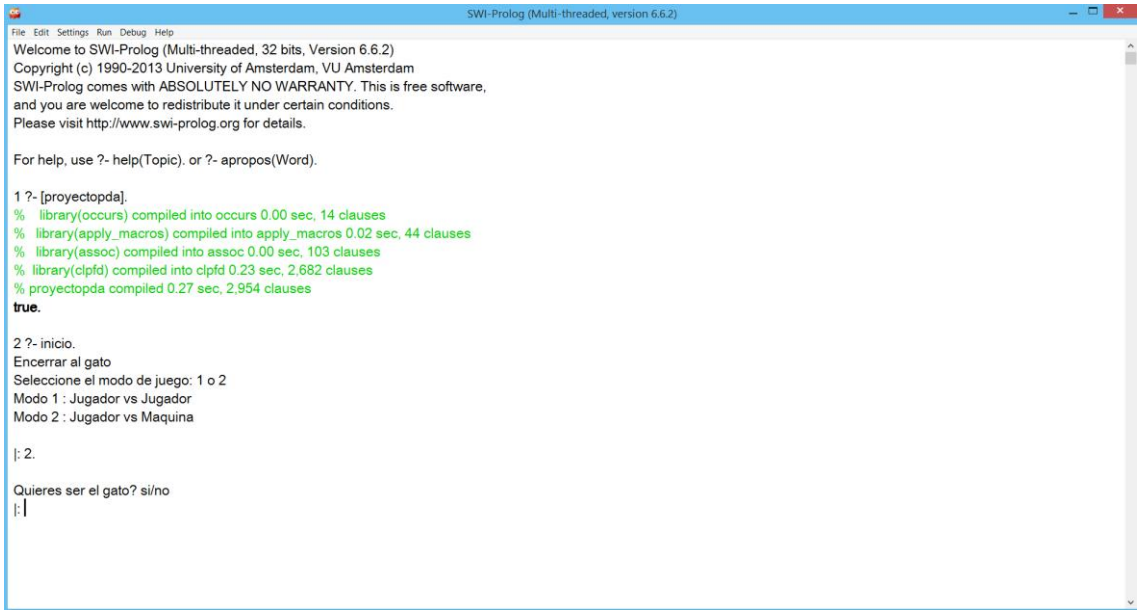
El jugador deberá introducir la instrucción “inicio.” para iniciar el juego.



El jugador deberá elegir uno de los modos de juego, “1.” para jugar contra otro jugador y “2.” para jugar contra la máquina.

Selecciona el modo de juego 1.

Selecciona el modo de juego 2.



El jugador en caso de elegir el modo de juego contra la máquina deberá seleccionar si desea jugar como gato “si.” o bien prefiere intentar encerrarlo “no.”

Seleccionamos jugar como gato.

```
SWI-Prolog (Multi-threaded, version 6.6.2)
File  Edit  Settings  Run  Debug  Help
Quieres ser el gato? si/no
|: si.

Eres el gato, tu objetivo consiste en llegar a uno de los bordes del tablero

 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
  01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
  01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
  01 | 02 | 03 | 04 | 05 | GT | 07 | 08 | 09 | 10 | 11
 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
  01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
  01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11

Comienza moviendo el jugador
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|G|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
Turno Jugador.
O|O|O|O|O|O|O|O|O|O|O
```

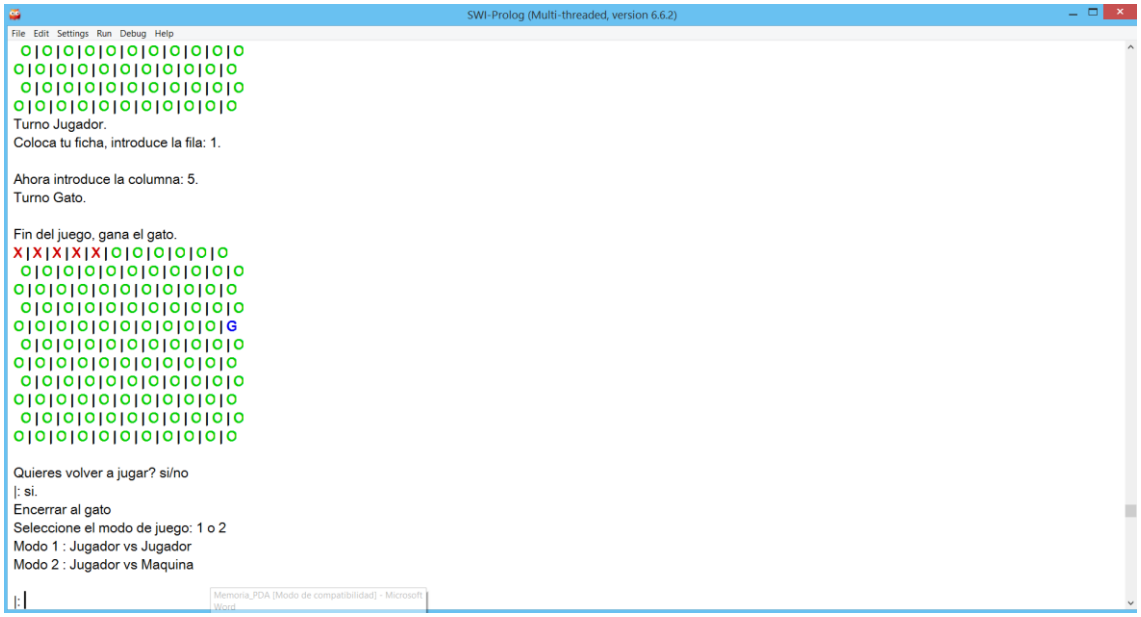
Seleccionamos no jugar como gato.

```
SWI-Prolog (Multi-threaded, version 6.6.2)
File  Edit  Settings  Run  Debug  Help
Quieres ser el gato? si/no
|: no.

Eres el jugador, tu objetivo consiste en atrapar al gato y evitando ademas que llegue a algun borde del tablero

 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
  01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
  01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
  01 | 02 | 03 | 04 | 05 | GT | 07 | 08 | 09 | 10 | 11
 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
  01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
  01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11
 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11

Comienza moviendo el jugador
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|G|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
Turno Jugador.
Coloca tu ficha, introduce la fila:
```

El jugador procederá a introducir la fila y columna donde desea colocar una ficha o bien mover al gato. Para ello se le solicitara primeramente la fila y posteriormente la columna.

```
SWI-Prolog (Multi-threaded, version 6.6.2)
File  Edit  Settings  Run  Debug  Help
Comienza moviendo el jugador
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|G|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
Turno Jugador.
Coloca tu ficha, introduce la fila: 1.

Ahora introduce la columna: 1.
Turno Gato.

X|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|G|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
Turno Jugador.
Coloca tu ficha, introduce la fila:
```

Una vez terminada la partida el jugador deberá seleccionar si desea volver a jugar. En caso de querer volver a jugar “si.” se volverá al inicio del juego y en caso de no querer jugar una nueva partida “no.” se cerrará la aplicación.

```
SWI-Prolog (Multi-threaded, version 6.6.2)
File  Edit  Settings  Run  Debug  Help
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|G|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
Turno Jugador.
Coloca tu ficha, introduce la fila: 1.

Ahora introduce la columna: 5.
Turno Gato.

Fin del juego, gana el gato.
X|X|X|X|X|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|G
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O
 O|O|O|O|O|O|O|O|O|O|O
O|O|O|O|O|O|O|O|O|O|O

Quieres volver a jugar? si/no
|:
```

Seleccionamos volver a jugar, y comienza de nuevo todo.