

Informe

Práctica de Prolog



Universidad de La Laguna
Grado Ingeniería Informática 3º
Asignatura - Inteligencia Artificial
2020/2021

Alberto Cruz Luis
alu0101217734@ull.edu.es



Índice

- ❖ Introducción
- ❖ Características del Mundo de Wumpus
- ❖ Entorno de Simulación
- ❖ Funcionamiento del juego
- ❖ Detalles de la Implementación
- ❖ Conclusión
- ❖ Bibliografía



Introducción

El objetivo de esta práctica es la utilización del lenguaje de programación prolog. Este lenguaje se basa en la programación lógica y por lo tanto una nueva forma de programar.

En este caso implementaremos el juego del Mundo de Wumpus. Que es un juego basado en el conocimiento.

El mundo de wumpus es una cueva de la que está compuesto por habitaciones de las cuales podemos encontrarnos Oro, Pozos o el Wumpus.

El agente se encontrara en la casilla de salida y tendrá que ir recorriendo habitaciones hasta poder salir de la cueva con el mayor oro posible.

Características del Mundo de Wumpus

En el juego de wumpus tenemos una serie de características:

Usuario

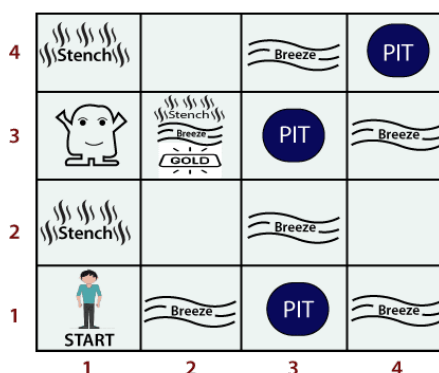
- Podrá elegir el tamaño del mundo. Por defecto 4 x 4.
- Mostrar dónde se encuentra en el mapa.

Agente

- Empezará en la casilla (1, 1).
- Podrá moverse: Arriba, Abajo, Derecha, Izquierda.
- Podrá realizar las siguientes acciones: Coger, Disparar, Escalar.
- Tiene 3 sensores para poder detectar lo que se encuentra en las habitaciones: Hedor(Wumpus), Brisa(Pozo), Brillo(Oro), Susto(Wumpus Muerto).

Recompensas

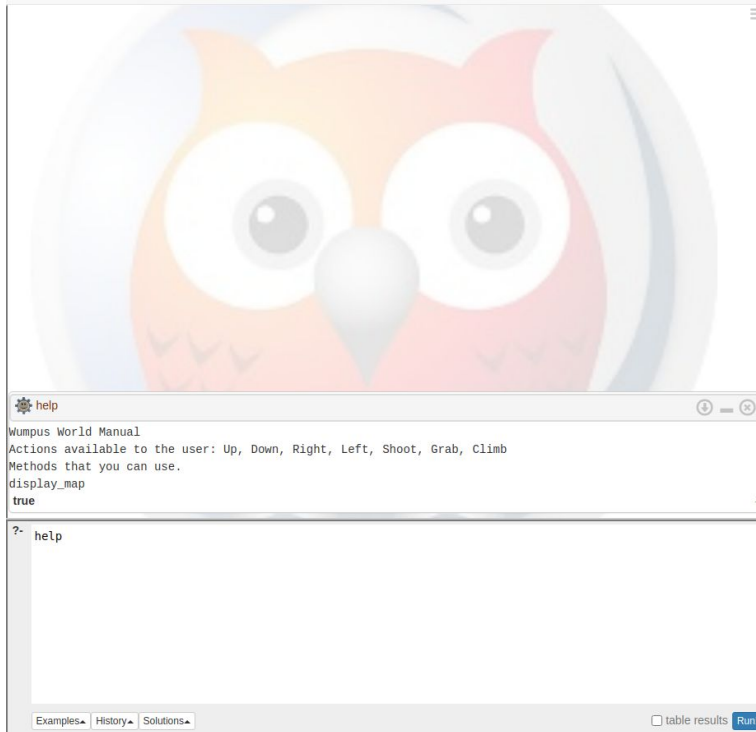
- +1000 puntos si el agente sale de la cueva con el oro.
- -1000 puntos si el agente es comido por el wumpus
- -1 punto por cada acción y -10 puntos por usar una flecha



A Typical Wumpus World

Entorno de Simulación

En este caso hemos usado una página web llamada [swi-prolog](#) en el que nos deja ejecutar prolog de manera online. En el que mediante la importación del código de nuestro programa podrás ejecutar aquí.



El programa se podrá ejecutar mediante consultas al conocimiento.

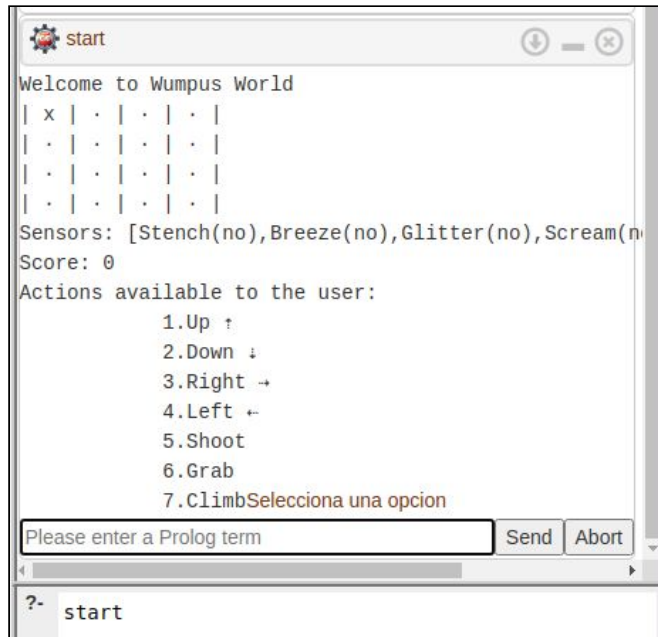
Para ello hemos creado una serie de predicados que el usuario puede consultar mediante un manual de ayuda que se accionará mediante el predicado help.

Funcionamiento del juego

Abrimos el entorno de [simulación](#).

Creamos un nuevo Programa y copiamos el [código](#)

Ejecutamos el predicado start.



A continuación iremos ejecutando las diferentes acciones que nos proporciona el juego hasta conseguir hacerse con todo el oro y salir de la cueva con el.

Detalles de la Implementación

Para la realización de este juego hemos seguido un paradigma de programación basado en el conocimiento, algo diferente de lo que estamos acostumbrados a usar.

En primer lugar hemos declarado unos predicados como el start o help que engloban a su vez otros predicados que son los que llevarán la lógica del juego.

Al principio del código hemos tenido que definir algunos **predicados** como **dinámicos**. La cuestión es que nuestro juego necesita recordar cuál fue la acción que realizó anteriormente. Ya sea para poder realizar movimientos y que el jugador vaya avanzando de casillas como saber si ya ha recogido oro o tan igual haya matado a un wumpus.

Por ello utilizaremos lo que se conoce como base de datos o estados en prolog.

Las acciones que se podrán realizar son las siguientes:

assert: Para insertar conocimiento a un predicado.

retract: Para eliminar conocimiento de un predicado.

Si quieres conocer más acciones de los predicados dinámicos -> [Predicate Dinamic](#)

La utilización de **listas** en prolog ha sido necesario para cosas como la creación del mapa en donde se encuentra el jugador.

Para ello hemos usado predicados como el **create_matrix** o el **write_matrix**.

Otro recurso que hemos tenido que utilizar ha sido los predicados **write**, **format** para poder escribir en la pantalla y el predicado **read** para poder leer lo que escribe el usuario.

Conclusión

Tras la realización de esta práctica he podido aprender una nueva manera de programar. De la que al principio me costó entender como hacer cosas que en otros lenguajes como c++, python, javascript... Hubiese sido realmente sencillo. Cosas como la creación de una mapa mediante una matriz. Algo que muchas horas me llevo hacer. O cosas como poder guardar la acción que realice antes que gracias a los predicados dinámicos pude resolver. Algo que en otros lenguajes ni te lo pasarías por la cabeza.



Bibliografía

<https://www.javatpoint.com/the-wumpus-world-in-artificial-intelligence>

<https://swish.swi-prolog.org/>

[https://www.swi-prolog.org/pldoc/doc_for?object=\(dynamic\)/1](https://www.swi-prolog.org/pldoc/doc_for?object=(dynamic)/1)

[https://programacion.net/articulo/curso avanzado de prolog 166/3](https://programacion.net/articulo/curso_avanzado_de_prolog_166/3)