

*Instituto Tecnológico de Costa Rica*

*Unidad de Computación*

Tarea Programada lll

*Danny Rojas Arguedas*

*Sede San Carlos*

*17/05/2019*

# Introducción

En el siguiente trabajo se analiza la creación de un juego de culebra (Snake) de estilo laberinto, en el cual dicha culebra tendrá que comer algunas manzanas puestas aleatoriamente en el campo de juego, esto la hará crecer, y una vez comidas todas las manzanas esta acabará su juego. La culebra tendrá que desplazarse desde su lugar actual hasta encontrarse con cada manzana, la ruta que esta tomará será calculada en el lenguaje de programación Prolog, en el cual se hará llamada y este devolverá la ruta más corta para llegar a dicha manzana.

La parte gráfica de desarrolla en el lenguaje de programación Java, el cual se encargará de realizar el desplazo de la culebra en la ruta indicada por Prolog. El usuario tendrá la posibilidad de escoger entre tres mapas, los cuales fueron creados por defecto por el programa al iniciar la sesión.

# Análisis del problema

El problema por resolver es la confección de un juego de culebra, en este caso este contará con obstáculos tipo paredes que le impedirán desplazarse por todo el mapa de manera libre. El juego deberá constar de al menos tres mapas con obstáculos ya creados, estos para ser escogidos por el usuario para que de inicio el juego de la culebra, también se le deberá solicitar al usuario la cantidad de manzanas que este desea colocar en el mapa, estas posteriormente serán comidas por la culebra para poder finalizar el juego.

El mapa debe ser creado inteligentemente, al usuario seleccionar un mapa, este deberá crear las conexiones necesarias entre los nodos posibles para que la culebra pueda realizar movimiento entre el mapa. Cuando se cree el mapa este deberá construir las conexiones del mismo en Prolog, por lo que deberá llamar una regla en Prolog que realice dicha acción.

Cuando se inicie el juego, la culebra deberá guiarse o dirigirse hacia las diferentes manzanas que estarán en el mapa, esto buscando mediante Prolog la ruta más corta para llegar a cada X manzana que haya elegido.

# Solución del problema

La interfaz gráfico del juego fue desarrollada en el lenguaje de programación Java, el cual mediante los diferentes componentes que contiene facilita dicha acción. El juego contiene un apartado que pedirá al usuario la escogencia entre tres mapas ya creados, en el momento que el usuario seleccionó alguno de los tres mapas este creará sus respectivas conexiones lógicas en Prolog, cada unos de los mapas son generados mediante una matriz 6x6 de botones. Estos botones son instancias de la clase llamada “Cuadro” la cuál tiene los atributos de un “JButton”, esta clase tiene una serie de atributos para identificar cada uno de los botones, el atributo tipo para distinguir entre un botón “Manzana”, “Bloque” o camino libre, también un “x” y un “y” para saber su posición en la matriz de botones y el nombre para poder hacer las conexiones mediante un nombre, el cual es generado con el formato de las coordenadas en las que se sitúa en la matriz de botones.

Para el funcionamiento del juego, al iniciar el programa en la parte de la ventana del juego, cuando el usuario ya ha escogido el mapa y se generaron las conexiones en Prolog con la función escrita “conexiones()” que llama a Prolog, este llama la función “colocarManzanas()”, la cual colocará una manzana en un lugar random del mapa, esta se descontará del contador que lleva la cantidad de manzanas que eligió el usuario, también se llamará dicha función cada vez que la culebra coma una manzana. Seguidamente, el juego espera que el usuario coloque donde estará la serpiente, para seguidamente dar inicio al juego. Una vez colocada la serpiente y que el usuario selecciona el botón “iniciar”, se creará un hilo que llevará el movimiento y funcionamiento de la culebra, el cual estará en funcionamiento hasta que se acaben todas las manzanas que el usuario indicó al inicio, esta cada vez que inicie el ciclo llamará a la función “ruta()”, la cual se encarga de llenar una lista llamada “soluciones”, con todos los nodos de la lista que devolvió Prolog al calcular la ruta más corta para x objetivo, dicha lista será recorrida en el hilo para hacer que avance la culebra por el camino indicado.

Para llevar el crecimiento y pintado de la culebra en el mapa, se creó la función “pintarCuerpoCulenra()”, la cual funciona con una lista que se registró al inicio para guardar cada una de las posiciones por donde ha caminado la culebra, y con una variable que lleva la cantidad de manzanas que se ha comido la culebra, entonces, para realizar el proceso se recorre dicha lista desde su primer elemento hasta la cantidad de manzanas comidas y se pintará en el mapa, de esta forma se obtendrá la culebra del tamaño indicado. Para recorrer esta lista y que funcionara de dicha forma, se tuvo que someter a un “Collections.reverse” el cual le hace un reverse a la lista, para así contar solo con las últimas posiciones donde estuvo la culebra.

# Análisis de resultados

|  |  |  |
| --- | --- | --- |
| ***Tarea/Requerimiento*** | ***Estado*** | ***Observaciones*** |
| *Creación de al menos tres mapas de juego* | *100%* |  |
| *Colocación de la culebra en el mapa de juego – Poder elegir donde ponerla* | *100%* |  |
| *Colocación de las manzanas en el mapa de juego* | *100%* |  |
| *Colocación de obstáculos en el mapa de juego* | *100%* |  |
| *Creación de los caminos o conexiones del mapa en Prolog* | *100%* |  |
| *Búsqueda de la ruta más corta en Prolog* | *100%* |  |
| *Movimiento de la culebra hasta cada manzana (vertical y horizontalmente)* | *100%* |  |
| *Recalculo de Ruta cada vez que se come una manzana la culebra* | *100%* |  |
| *Poder elegir la cantidad de manzanas a utilizar en el mapa* | *100%* |  |
| *La manzana desaparece cuando la serpiente la come* | *100%* |  |
| *Crecimiento de la culebra al comer manzanas* | *100%* | *La culebra tiene la posibilidad de pasar sobre su propio cuerpo, por eso pareciera que decrece, pero en realidad solo está avanzando.* |

# Manual de Usuario

 Al iniciar la aplicación, esta mostrará una ventana en la cual se podrá elegir entre iniciar el juego o salir de inmediato. Se muestre en la Figura 1.

Figura 1

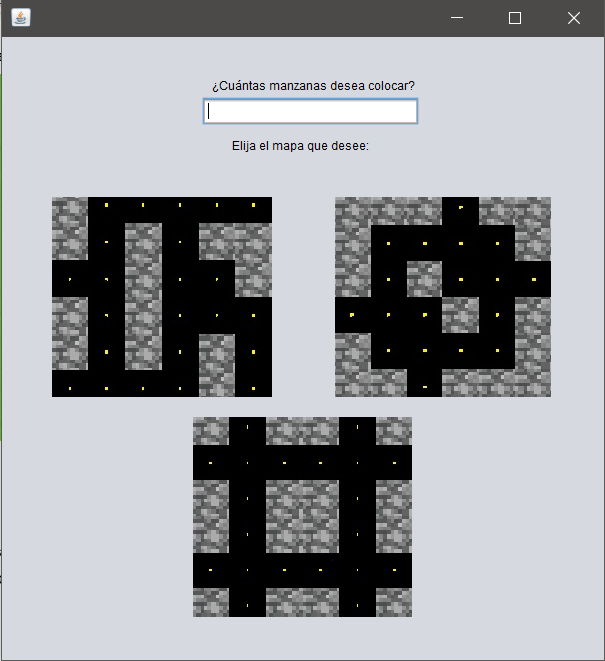
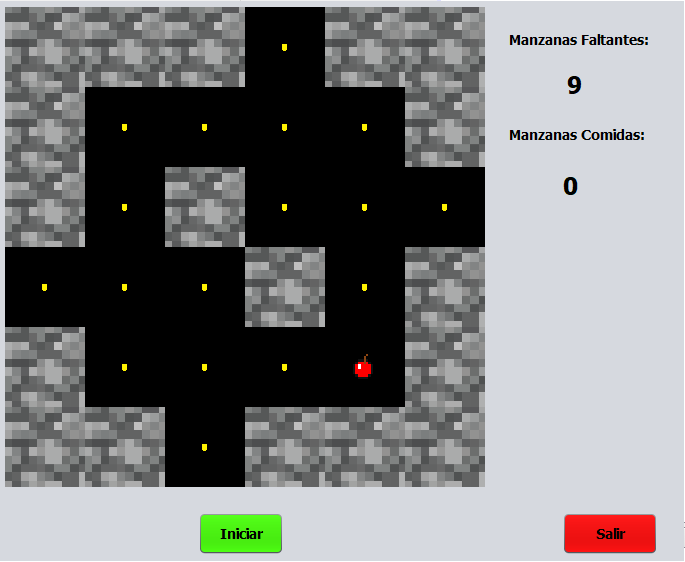
 Seguidamente, si el usuario elige “START” para iniciar el juego, a este se le mostrará una ventana que le pedirá dos parámetros para ingresar o iniciar la partida, el número de manzanas que aparecerán durante el juego (las manzanas que el Snake consumirá para crecer) y el mapa deseado para esa partida. Se muestra dicha vista en la Figura 2.

Figura 2

Para finalizar con la interacción con el usuario, este después de haber seleccionado el mapa, deberá colocar la culebra en algún punto o espacio disponible (cuadros negros Figura 3). Una vez colocada la culebra en el campo de juego, deberá proceder a dar “Iniciar” al juego, el cuál dará inicio a la partida del Snake. Se muestra la vista en la Figura 3, 4 y 5.

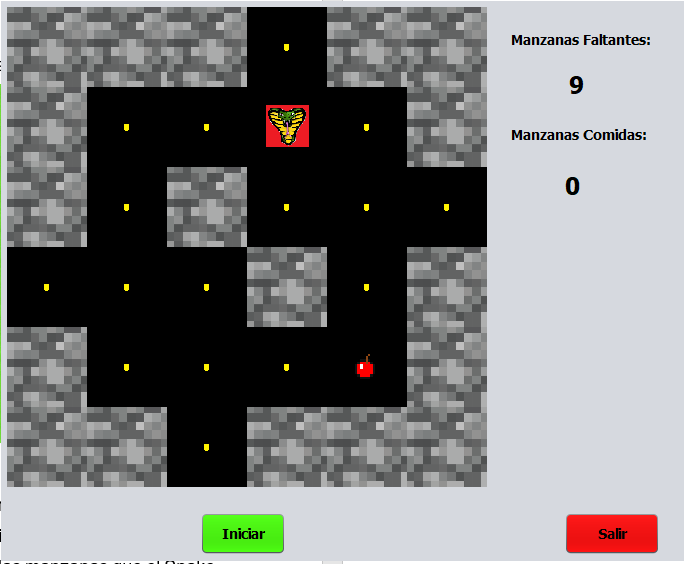
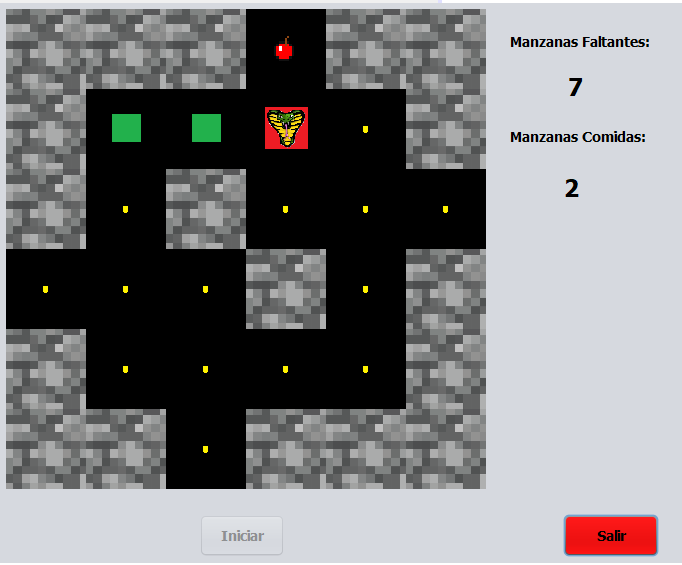
Figura 3 (antes de colocar el Snake)

Figura 4 (Snake colocado)

Figura 5 (Snake en movimiento comiendo manzanas)

También, dicha vista lleva un conteo de las manzanas que faltan por comer y las manzanas que el Snake ha comido ya.

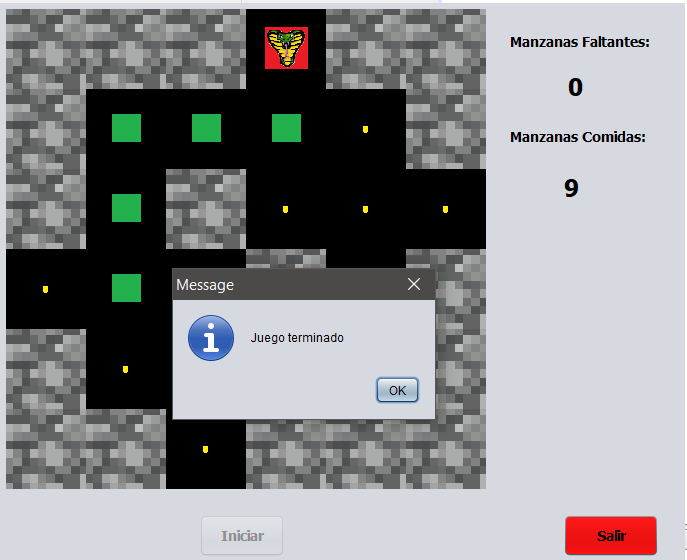
 Una vez comidas todas las manzanas, la aplicación le muestra al usuario una alerta de que el juego ya ha sido terminado, y este procederá a seleccionar el botón de “Salir” para terminar. Se muestra en la Figura 6.

Figura 6

# Conclusiones

El uso de la programación lógica puede resultar muy efectiva en algunos casos de uso, esto debido a que los resultados suelen ser bastante rápidos y eficaces, gracias a que este hace uso del bracktraking para obtener los dichos resultados.

Prolog como lenguaje lógico suele ser usado en el campo de la inteligencia artificial, lo cual es una muy buena opción para su desarrollo.

# Referencias

Conectar prolog con Java paso a paso - YouTube. (n.d.). Retrieved May 13, 2019, from <https://www.youtube.com/watch?v=pJ42DK7FMOo&t=>

Java - JPL/SWI Prolog configuration failure - Stack Overflow. (n.d.). Retrieved May 13, 2019, from <https://stackoverflow.com/questions/12283471/jpl-swi-prolog-configuration-failure>

Tutorial Prolog – YouTube. (n.d.) Retrieved May 12, 2019, from <https://www.youtube.com/watch?v=7lwW78BljzI&list=PLHNkID2PAnJmoXlM0MtgMmNVpkarf2Cd4>