Agentes inteligentes

Intelligent agents.

Autores: Héctor Julián Cardona Montoya,

Karem Giovanna Parra Moreno y Alberto Andrés Ospina López

*Computación Blanda, Ingeniería de Sistemas y Computación, Universidad Tecnológica de Pereira.*

Correo-e: [julian.cardona@utp.edu.co](mailto:julian.cardona@utp.edu.co), [karem.parra@utp.edu.co](mailto:karem.parra@utp.edu.co),

[alberto.ospina@utp.edu.co](mailto:alberto.ospina@utp.edu.co).

***Resumen*— En el siguiente documento se explicará el funcionamiento de un agente inteligente mediante la implementación del juego snake con 3 diferentes tipos de algoritmos de búsqueda**

***Palabras clave—*. Neurona, red, algoritmo, señal, pesos, reconocer, clasificar, agente, inteligente, búsqueda, elección.**

***Abstract*— The following document will explain the operation of an intelligent agent by implementing the game snake with 3 different types of search algorithms**

***Key Word* —. Neuron, network, algorithm, signal, weights, recognize, classify, agent, intelligent, search, choice.**

1. INTRODUCCIÓN

La computación muchas veces se basa en el análisis de la naturaleza y sus fenómenos para aplicarlos en la resolución de problemas de diferentes maneras, un ejemplo de esto son los agentes inteligentes.

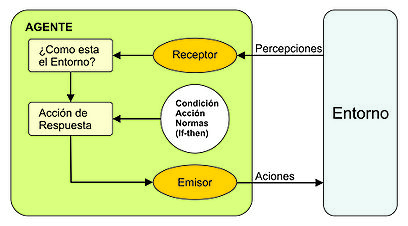
Los agentes inteligentes son una entidad software que, basándose en su propio conocimiento, realiza un conjunto de operaciones destinadas a satisfacer las necesidades de un usuario o de otro programa, bien por iniciativa propia o porque alguno de éstos se lo requiere.

Todas estas operaciones se pueden tomar como decisiones que tomará el agente referente a su ambiente, para poder lograr un objetivo dado o retornar un resultado esperado.

1. CONTENIDO
2. **Representación del problema**

En primer lugar tenemos la idea de darle autonomía a la serpiente del famoso y mundialmente jugado snake, así que primeramente tomaremos la serpiente como nuestro agente que dependiendo de lo que perciba del ambiente.

Hablamos que para poder percibir el ambiente la serpiente debe saber por donde se está moviendo para poder utilizar sus percepciones para tomar decisiones.  
Nuestro ambiente será todas aquellas cuadrículas o conjuntos de píxeles por donde nuestra serpiente podrá moverse, y nuestras percepciones serán si el siguiente movimiento será o no movimiento válido en las reglas que nosotros le proporcionamos a nuestro agente.



**Imagen 1**

Para nosotros es muy sencillo visualizar los movimiento válidos que tiene nuestra serpiente dependiendo de una posición, pero debemos recordar que principalmente sigue siendo un programa de unos y ceros y si no proporcionamos la reglas correctas nuestro agente se moverá de formas que no podremos entender y hasta hacer movimientos inválidos.

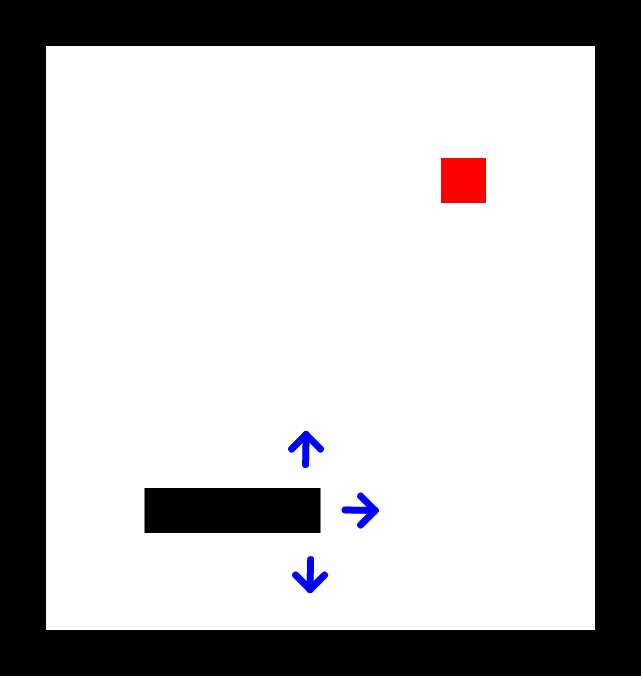
Tomamos nuestro conjunto de reglas más básico como:  
“Si en el próximo movimiento a realizar no excede el tamaño máximo de nuestra pantalla ni toca algún integrante de su propio cuerpo, el movimiento podrá realizarse de manera satisfactoria, en caso contrario el agente deberá explorar diferentes movimientos hasta encontrar la opción correcta.”

1. **Estados de nuestro agente inteligente**

En este agente inteligente que es nuestro snake que se basa en algoritmos de búsqueda, debemos especificar un conjunto de estados para que nuestro agente “sepa” dónde parar de buscar y comenzar otra nueva ruta de búsqueda.  
  
Podemos identificar estos estados como estado actual:

El estado actual es la posición en donde se encuentra nuestro agente.  
Desde ese estado verificará dependiendo del ambiente y lo que percibe su siguiente movimiento válido.  
El estado final:  
Este estado es el que le indicará a nuestro agente que su búsqueda ha terminado y en ese momento nuestro agente conocerá su ruta y empezará a moverse.

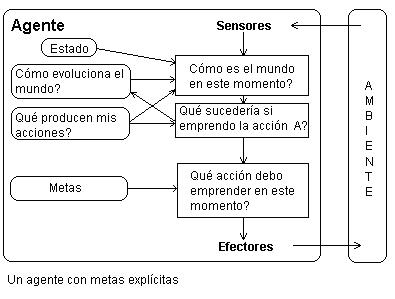
1. **Características de nuestro agente**



**Imagen 2**

Nuestro agente es un agente basado en objetivos, ya que su mayor objetivo es hallar su comida la cual hará crecer a nuestra serpiente y así ir aumentando su puntuación.

El agente debe saber las metas que se desean alcanzar, no es suficiente solo con saber el estado actual del entorno en el cual se encuentra. El agente deberá ser capaz de analizar la situación actual del entorno con las posibles acciones que se podrían ejecutar y de esta manera seleccionar la acción que más le convenga para alcanzar las metas de una manera más sencilla. Agentes basados en metas necesitan cierto tipo de información sobre sus metas. Estas metas van a detallar las situaciones a las que se desea llegar de este modo, el programa de agente puede combinar las metas con la información de los resultados (acciones) que emprenda y de esta manera poder elegir aquellas acciones que permitan alcanzar la meta.

**

**Imagen 3**

1. **Algoritmos de búsqueda**

Un [**algoritmo**](https://es.wikipedia.org/wiki/Algoritmo) **de búsqueda** es aquel que está diseñado para localizar un elemento con ciertas propiedades dentro de una [estructura de datos](https://es.wikipedia.org/wiki/Estructura_de_datos); por ejemplo, ubicar el registro correspondiente a cierta persona en una [base de datos](https://es.wikipedia.org/wiki/Base_de_datos), o el mejor movimiento en una partida de [ajedrez](https://es.wikipedia.org/wiki/Ajedrez).

La variante más simple del problema es la búsqueda de un número en un [vector](https://es.wikipedia.org/wiki/Vector_(programaci%C3%B3n)).

1. **Tipos de búsqueda**

Un problema típico de la [Inteligencia Artificial](https://es.wikipedia.org/wiki/Inteligencia_Artificial) consiste en buscar un estado concreto entre un conjunto determinado, al que se le llama espacio de estados. Imaginemos, por ejemplo, una habitación con baldosines en la que hay un libro. Un [robot](https://es.wikipedia.org/wiki/Robot) se desea desplazar por la habitación con el fin de llegar a dicho libro. ¿De qué manera lo hará? En este punto es donde entran en juego las estrategias y los **algoritmos de búsqueda**.

Cuando el sistema agente (en este caso, el robot) posee algún tipo de información del medio, se utilizan técnicas de búsquedas informadas; sin embargo, si carece de conocimiento alguno, se deberán emplear algoritmos de búsqueda no informadas. En nuestro ejemplo, y para este último caso, podemos imaginar un robot que no posea ningún tipo de visión artificial, que únicamente sea capaz de moverse en horizontal o vertical de un baldosín a otro y detectar si en el baldosín se halla el libro.

De esta forma, los algoritmos de búsqueda pueden ser:

Algoritmos [no informados](https://es.wikipedia.org/wiki/B%C3%BAsquedas_no_informadas) o ciegos: en general más ineficientes en tiempo y memoria que otros métodos.

Algoritmos informados

Algoritmos [heurísticos](https://es.wikipedia.org/w/index.php?title=Heur%C3%ADstica_(inform%C3%A1tica)&action=edit&redlink=1): destacan las Búsquedas Primero el Mejor ([Algoritmo voraz o Greedy](https://es.wikipedia.org/wiki/Algoritmo_voraz) y [Algoritmo de búsqueda A\*](https://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda_A*)) y de Mejora Iterativa ([Algoritmo Escalada Simple -Hill Climbing-](https://es.wikipedia.org/wiki/Algoritmo_hill_climbing) y Escalada por Máxima Pendiente)

Algoritmos de Búsqueda con adversario: destacan el [Minimax](https://es.wikipedia.org/wiki/Minimax) y el [Poda alfa-beta](https://es.wikipedia.org/wiki/Poda_alfa-beta).

1. CONCLUSIONES

Los agentes inteligentes nos ayudan a entender de manera muy concreta las bases de la Inteligencia Artificial, para de allí poder empezar a aprender y profundizar más los conocimientos.

REFERENCIAS

1. Tipos de Agentes inteligentes <https://www.mindmeister.com/es/283831289/tipos-de-agentes-inteligentes>
2. Definición de los agentes inteligentes en la inteligencia artificial. <https://es.wikiversity.org/wiki/Inteligencia_artificial/Agentes_inteligentes>
3. Inteligencia aritificial <https://es.wikipedia.org/wiki/Agente_inteligente_(inteligencia_artificial)>