

Reporte Proyecto Final Inteligencia Artificial

Juan Manuel Sol Cabañas, Miguel Ángel Pech May y Ricardo Iván Aguirre Can

FACULTAD DE MATEMATICAS. UADY

INGENIERIA EN COMPUTACIÓN

DICIEMBRE 16, 2015

MAESTRO: M EN C. ALEJANDRO PASOS RUIZ

Abstract—En este documento se explica de manera detallada la realización del proyecto final de la materia de inteligencia artificial, que consiste en programar el juego “El mundo del wumpus” así como la I.A que lo jugará. Para realización del proyecto se requirió el uso un lenguaje de programación al cual escogimos JavaScript, así como algoritmos de búsqueda para poder programar la I.A que lo resolverá

INTRODUCCIÓN

La Inteligencia Artificial es un campo de la Computación actualmente en auge, debido a la necesidad de encontrar soluciones a problemas de la vida real de manera automatizada. Uno de los métodos utilizado para la docencia sobre Inteligencia Artificial es diseñar pequeños problemas o juegos que puedan resolverse mediante algoritmos de Búsqueda Heurística. Algunos pueden llegar a ser muy conocidos y extenderse por todo el mundo, y éste es el caso del problema de “El Mundo de Wumpus”.

TEORIA

Juego “El mundo del wumpus”

El juego es desarrollado en un tablero de dimensiones variables, generalmente cuadrangular. Encontraremos ciertos elementos que se distribuirán por todo el tablero, éstos serán:

- Cazador
- Wumpus
- Murciélagos
- Tesoros
- Pozos
- Existencia de Viento
- Existencia de Hedor
- Existencia de aleteo
- Existencia de Hedor y viento
- Existencia de Viento y aleteo
- Existencia de Hedor y aleteo
- Existencia de Hedor, Viento y aleteo



Fig 1. Imagen representativa de “El mundo del Wumpus”

El objetivo del juego es sencillo. El Aventurero debe de buscar por el tablero un tesoro y salir sin ser devorado por el Wumpus o caer en ningún pozo. Expresado en términos heurísticos, el Aventurero deberá encontrar el camino óptimo para realizar su cometido sin ser devorado o caer en algún pozo.

Algoritmo de Búsqueda

Busqueda en profundidad : Se comienza en el vértice inicial (vértice con índice 1) que se marca como vértice activo. Hasta que todos los vértices hayan sido visitados, en cada paso se avanza al vecino con el menor índice siempre que se pueda, pasando este a ser el vértice activo. Cuando todos los vecinos al vértice activo hayan sido visitados, se retrocede al vértice X desde el que se alcanzó el vértice activo y se prosigue siendo ahora X el vértice activo.

ALGORITMO BEP: Sea $G = (V, A)$ un grafo conexo, $V' = V$ un conjunto de vértice, A' un vector de arcos inicialmente vacío y P un vector auxiliar inicialmente vacío:

- 1) Se introduce el vértice inicial en P y se elimina del conjunto V' .
- 2) Mientras V' no sea vacío repetir los puntos 3 y 4. En otro caso parar.
- 3) Se toma el último elemento de P como vértice activo.
- 4) Si el vértice activo tiene algún vértice adyacente que se encuentre en V' :

Se toma el de menor índice. Se inserta en P como último elemento. Se elimina de V' . Se inserta en A' el arco que le une con el vértice activo. Si el vértice activo no tiene adyacentes se elimina de P .

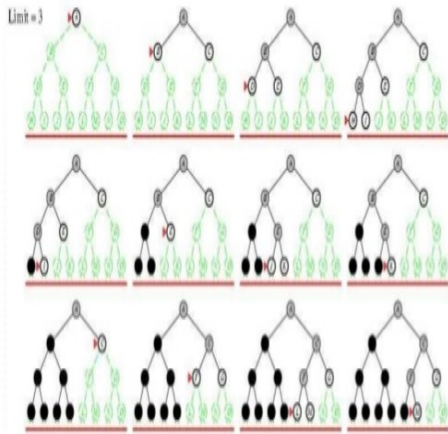


Fig 2. Visualización del algoritmo BEP

Algoritmo de aprendizaje

Aprendizaje por refuerzo : Una aproximación de IA, que se puede ver como el aprendizaje por interacción y aprendizaje basado en objetivos mientras se interactúa con un entorno externo, este aprendizaje se puede ver como aprendizaje por refuerzo que consiste en la recolección de recompensas mientras se avanza en el transcurso del juego.

I. DESARROLLO DEL PROYECTO

A. Objetivo del proyecto

Tener una I.A que se encargue de jugar el juego de “El Mundo del Wumpus” a través de una interacción “Agente” y “Entorno”. En la que el agente reaccionará a los cambios en el entorno con forme vaya avanzando en el laberinto.

B. ¿Que procede?

Implementar un algoritmo de aprendizaje por estados, tales estados se dividirán en

- Muerto
- Vivo

Mientras el agente esté vivo puede percibir percepciones del entorno que consisten en oír el aleteo de los murciélagos, oler el hedor del wumpus y sentir la brisa del pozo, también puede llegar al estado “ganar” o estado “ganador”. Al estado ganar, solo se llega a él si se encuentra el oro. Al estado “Muerto”, se llegue de más maneras, una es entrar a la cueva y caerse en el pozo, encontrarse al wumpus sin tener el arco y la flecha y la ultima es morir al entrar en la cueva de los murciélagos vampiro.

C. Diseño

Para programar el juego del “Mundo del Wumpus” dividimos la programación en 5 clases, sin contar a la clase “main” :

- Laberinto
- Cuevas
- Cazador
- Juego
- Search

La primera clase es la clase “Cueva”, en ella se encuentra la declaración de los objetos que estarán dentro de las cuevas del juego que son Pozos, Flecha, Wumpus y Murciélagos. Así como los indicios para saber que hay en cada cueva, que son; Peste, Aleteo y Brisa. La Peste será para indicar que en la cueva está el wumpus, el aleteo para los murciélagos y la brisa para el pozo respectivamente.

La segunda clase es la clase “Laberinto”, en esta solicitamos dos números “m” y “n”, que servirán para declarar el número de filas y de columnas que tendrá nuestro laberinto, después de crear el laberinto le podremos una cantidad de cuevas para ello llamaremos a la clase “Cueva” y la usaremos para colocar cuevas en nuestro laberinto.

La tercera clase es la clase “Cazador” que se encarga de colocar al cazador en el laberinto, para ello buscará un espacio libre de cuevas y sensaciones, también se encargará de ubicar al cazador en todo momento, además de saber si el cazador tiene la flecha.

La quinta clase es la clase “Search” es el algoritmo de búsqueda por profundidad, en esta clase se solicita el problema y se define el método que se usará para resolver el problema, que en nuestro caso es el algoritmo de búsqueda por profundidad.

La sexta clases es la clase “Juego” en ella se define el progreso del juego, es decir los movimientos del jugador, así como los estados en los que se encuentra el cazador, los cuales son vivo o muerto.

II. RESULTADOS Y CONCLUSIONES

RESULTADOS

Con la programación del algoritmos de búsqueda por profundidad y el algoritmo de aprendizaje por refuerzo, podemos tener una I.A que busque le oro en la cueva y gane. Ahora con nuestro conjunto de condiciones “IF” podemos saber que casillas son las candidatas a contener el wumpus, el pozo o los murciélagos. Si la I.A evita estas casillas se puede considerar como un éxito, ya que el objetivo se cumple. Ya que con tener la interacción del entorno y el agente se cumple con los objetivos del proyecto.

CONCLUSIÓN

Como pudimos ver la inteligencia artificial está compuesta de varios elementos, uno es el algoritmo que se usa para resolver el problema y otro es el algoritmo de aprendizaje. Otro punto en cuestión es la definición del agente y entorno, en el caso del juego del wumpus nuestro agente es la clase “Juego” y el entorno es la clase “Laberinto”, en la clase “main” se encuentra la interacción de ambos objetos, ya que esta clase llama a las demás y se hace la interacción en las casillas del laberinto y así podemos saber en qué estado esta nuestro cazador.

Al concluir con la implementación de todo el programa, tanto de la inteligencia artificial como del juego, e intentar unir estas partes, se detectó que al crear el laberinto, por alguna razón éste se llenaba con el primer elemento señalado en la función “main”, por ejemplo, haciendo la prueba al ingresar únicamente un Wumpus en una matriz

de 2x2, mediante la función creadora del laberinto que es: `laberinto2.generarmundo(m,n,1,0,0)`, con m y n iguales a 2. Al final el laberinto quedó lleno de Wumpus.

Es importante mencionar que todo el juego del Wumpus se basó en el código de uno de nuestros integrantes del equipo, es un antiguo proyecto de la clase de programación. Durante el transcurso de la realización del proyecto transcribimos el juego en código c++ al lenguaje Javascript y le agregamos la implementación de la I.A.

REFERENCES

- [1] Inteligencia Artificial, Un enfoque moderno. Stuart Russell, Peter Norvig