

Universidad Mariano Gálvez De Guatemala
Facultad de Ingeniería en Sistemas de Información
Inteligencia Artificial
Campus Huehuetenango
Ingeniero: Jorge García



“Juego Wumpus”

Diego Alejandro Samayoa Gómez Carné: 4490-19-4714

Juan Amilcar Larias Méndez Carné: 4490-19-9119

Huehuetenango 09 de mayo, 2023







Juego wumpus

El juego se desarrolla en un tablero de 10 x 10 dimensiones.

El cazador debe de buscar por el tablero el oro sin ser comido por el wumpus o caer en ningún pozo. En términos heurísticos, el cazador necesita encontrar el camino más óptimo para realizar el recorrido y ganar.

Elementos de tablero

1. Casillas

Nombre	Descripción	Ilustración
Casilla vacía	Espacio sin valor, se pueden recorrer sin riesgos o premios.	
Casilla del wumpus	Espacio donde se genera el monstruo wumpus.	
Casilla del pozo	Espacio donde se genera el pozo con un agujero.	
Casilla del oro	Espacio donde se genera el premio del juego.	
Casilla de hedor	Espacio colocado de forma adyacente al wumpus.	
Casilla de brisa	Espacio colocado de forma adyacente a los pozos.	

2. Cazador

Es el agente que se moverá por el tablero. Únicamente puede ganar si consigue el oro. Según el algoritmo de búsqueda decida el camino a seguir, este se moverá según las coordenadas.

El cazador puede hacer las siguientes acciones:

- Moverse a la izquierda, derecha, arriba y abajo, pero nunca en diagonal.
- Morir si cae en la casilla del wumpus.
- Morir si cae en la casilla del pozo.
- Ganar si cae en la casilla del oro.
- Cambiar sus movimientos si detecta algún obstáculo.

3. Wumpus

Es el monstruo que se genera aleatoriamente en el tablero. Únicamente existirá uno en el mapa y tendrá una posición estática. Alrededor de su posición se generará un hedor la cual creará casilla de hedor adyacentes para advertir al cazador.

Si el cazador cae en su casilla será comido y perderá.

4. Pozos

Es el hoyo que se genera aleatoriamente en el tablero. Existirán dos en el mapa y tendrán una posición estática. Alrededor de sus posiciones se generará una brisa la cual creará casilla de brisa adyacente para advertir al cazador.

Si el cazador cae en su casilla perderá.

5. Oro

Encontrar el oro es el principal objetivo que tiene que conseguir el cazador. Únicamente existirá una casilla de este tipo y su posición será estática. En sus posiciones adyacentes no se generará ningún objeto para alertar al cazador.

Si el cazador cae en su casilla este ganará.

Método de búsqueda

Para realizar la búsqueda del camino más corto en el tablero se usó el método A*.

1. Se crea una función heurística que utiliza la distancia de Manhattan para estimar la distancia entre dos puntos del tablero. La distancia es la suma de las coordenadas horizontales y verticales.
2. La función astar implementa el algoritmo de búsqueda A*. Hace uso de un punto de inicio, punto final y el arreglo del tablero.
3. Se registran los nodos visitados.
4. Se realiza un bucle que se ejecuta mientras haya nodos en la lista de nodos abiertos.
5. En cada iteración se selecciona el nodo con menor valor de la lista de nodos abiertos.
6. Si el nodo seleccionado es el objetivo, se reconstruya el camino desde el punto de inicio usando el registro de los nodos visitados.
7. El nodo seleccionado se marca como cerrado.
8. Para cada vecino, se calcula el costo de llegar a ese vecino desde el nodo actual y se realiza lo siguiente:
 - a. Si el vecino ya está en la lista de nodos abiertos, se compara el nuevo costo con el costo existente. Si el nuevo costo es mayor, se ignora este vecino. Si el nuevo costo es menor, se actualizan los valores y se registra el nuevo camino.
 - b. Si el vecino no está en la lista de nodos abiertos, se agrega a la lista con el costo actualizado.
 - c. Si el vecino está en la lista de valores excluidos, se busca un nodo vecino el cual sea válido.
9. Se inicializan las estructuras de datos necesarias: open_set es una lista de nodos abiertos, closed_set es un conjunto de nodos cerrados y came_from es un diccionario que registra los nodos padres de cada nodo visitado.
10. Se repite el ciclo hasta encontrar el objetivo o agotar los nodos abiertos.
 - a. Si no se encuentra un camino adecuado, se devuelve none.
11. Se imprimen las coordenadas utilizadas para marcar el camino.

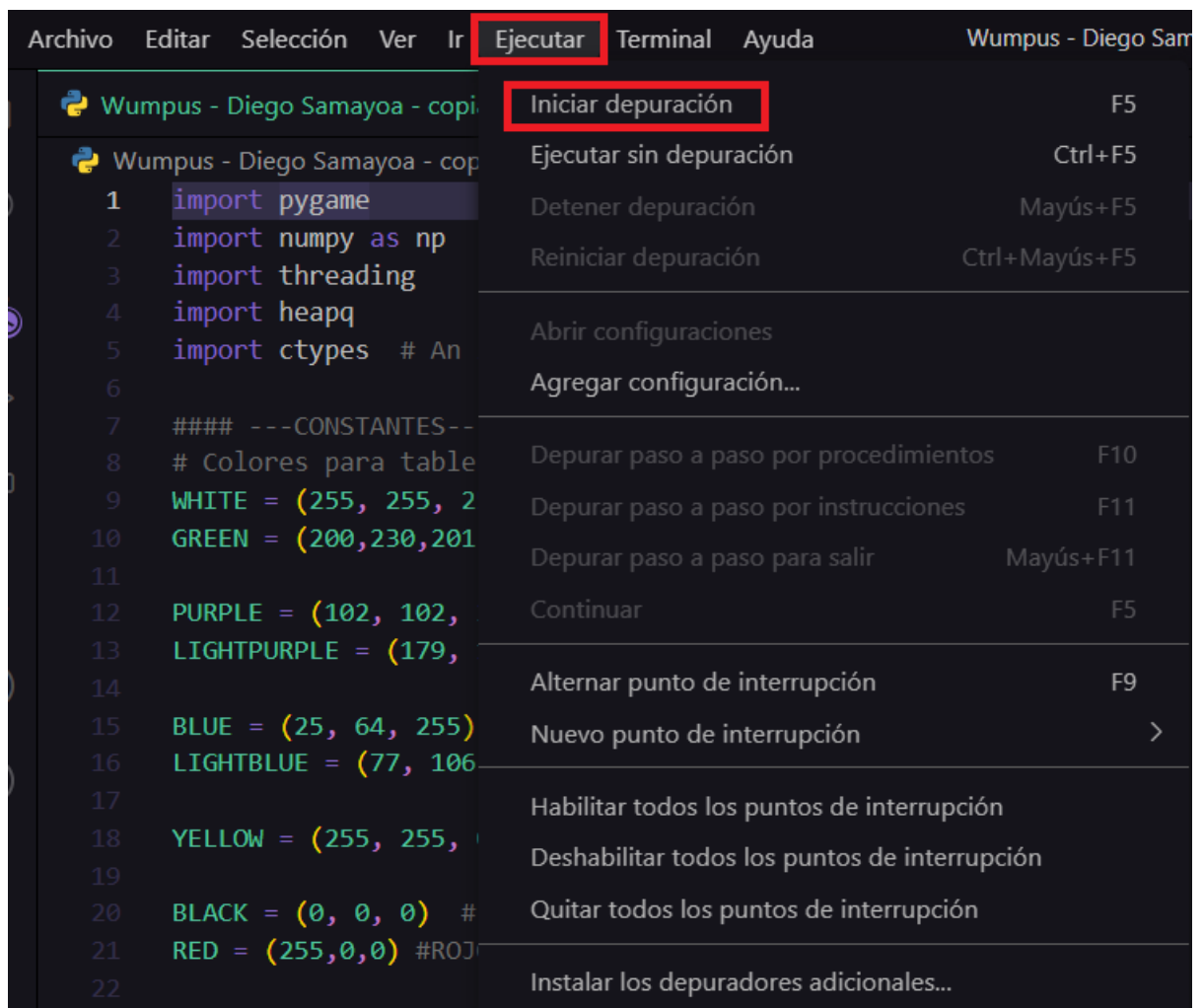
Software Requerido

- Python 3
- Pygame
- Numpy
- Threading
- Heapq
- Ctypes

Pasos para ver el funcionamiento del juego.

1. Ejecutar el código Main.

Desde la pestaña <Ejecutar> y luego en <Iniciar depuración>



2. Nos generará una vista del juego.



Cada elemento, exceptuando el jugador, se generará en una posición aleatoria dentro del tablero, por lo que cada vez que ejecutemos el código, la vista será distinta y tendremos un nuevo juego diferente a los anteriores.



3. En cada nuevo juego contara con:

3.1 Un jugador

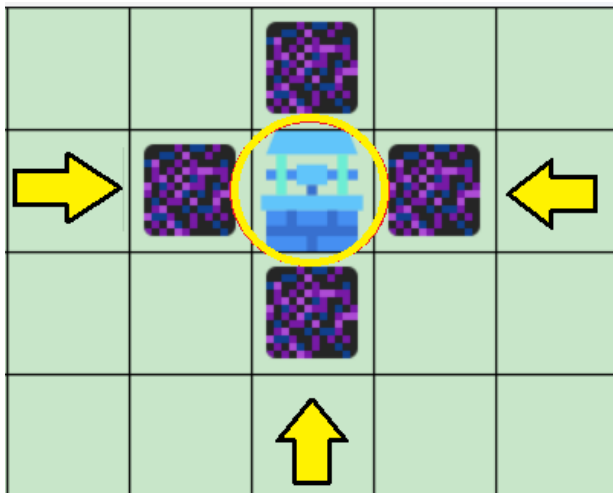
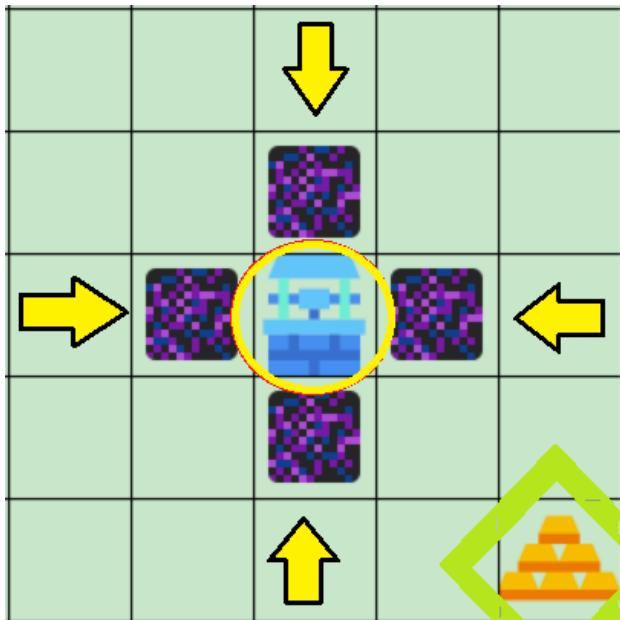
Que empieza por defecto en la primera posición.



En caso de que se generará una trampa en la primera posición, el jugador se generará en la posición $x=2$, $y=2$

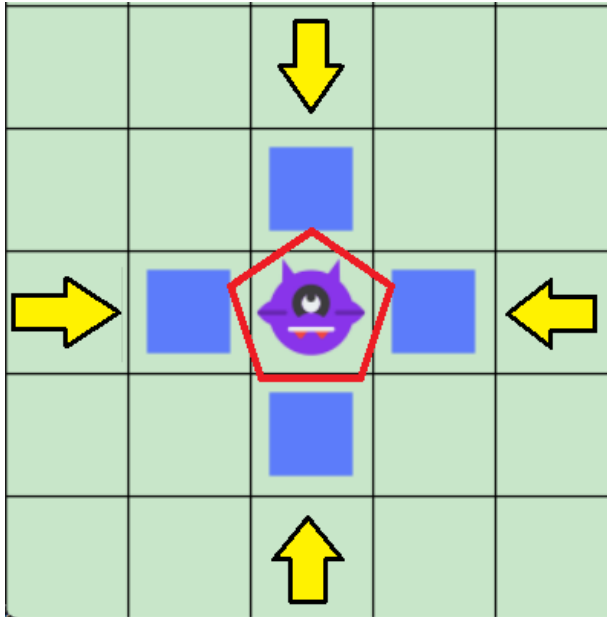
Se generarán trampas (Círculos amarillos)

En donde habrá advertencias en sus cuatro ejes (Flechas Amarillas).



Se generará el Wumpus (Pentágono rojo)

De igual forma se generará advertencias en sus cuatro ejes, indicando que dentro de ese rango se encuentra el Wumpus.



4. Pasos para encontrar el camino más óptimo.

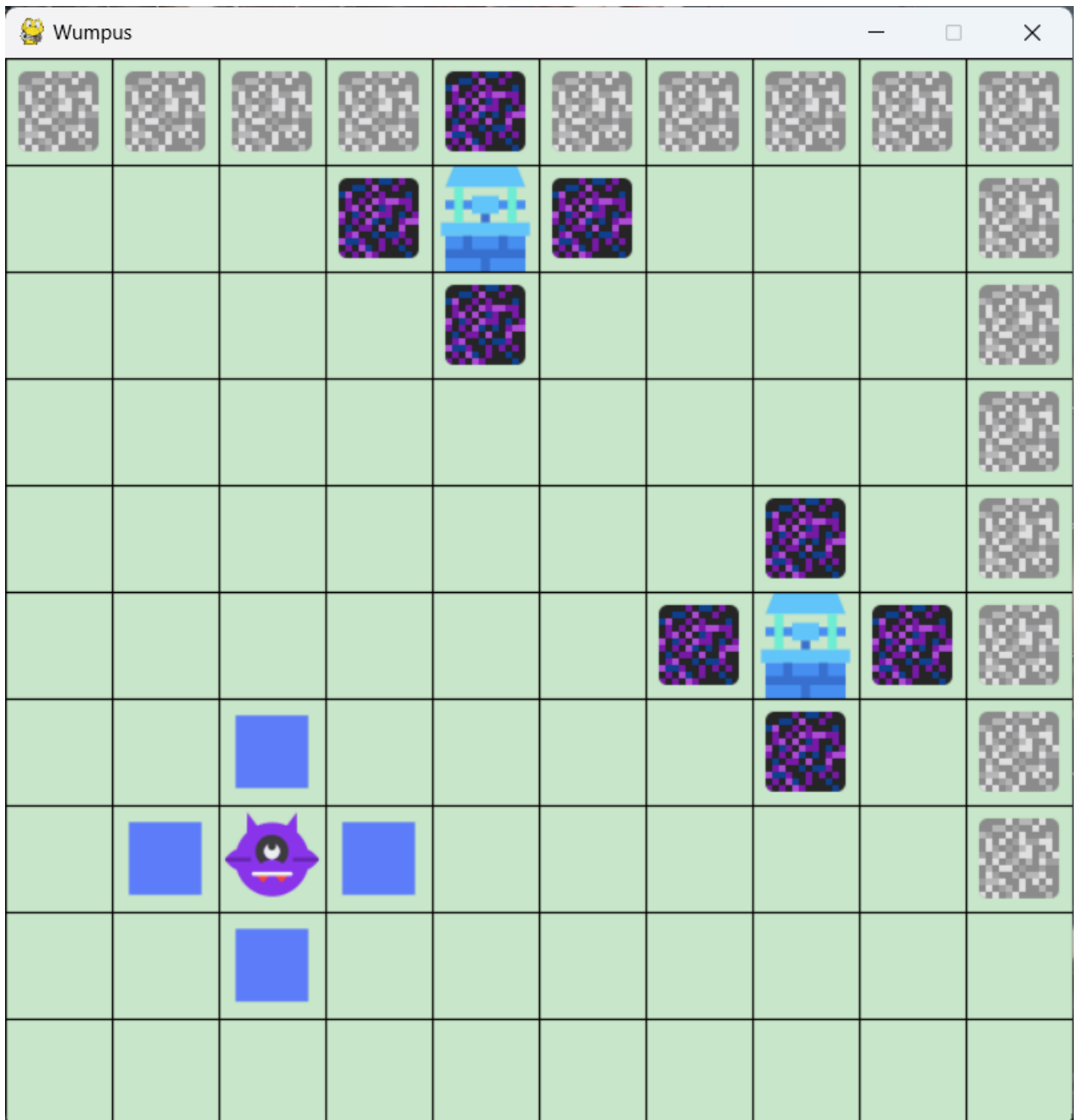
4.1 Barra espaciadora

Podremos darle un recorrido por todo el tablero, haciendo así un reconocimiento del mapa.

Space

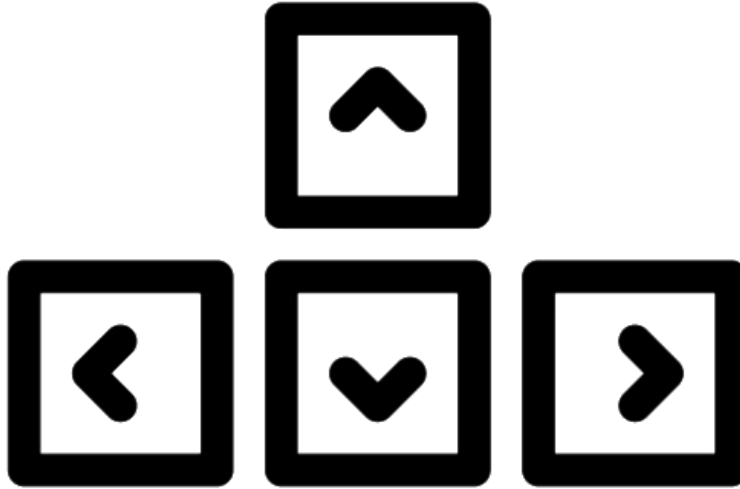
4.2 Tecla S

Nos generará de manera visual cual es el camino más óptimo hacia el oro. Llegando así al objetivo del juego.



4.3 Teclas Adicionales.

Con las teclas habituales de movimiento, podremos mover de manera manual el jugador, haciendo nuestro propio camino hacia el oro.



4.4 Teclas P.

Se mostrará un cuadro de texto con las instrucciones del juego.

Bienvenido, toma las instrucciones

×

En primera casilla se encuentra el cazador, para ganar tienes que conseguir el oro, pero ten cuidado con los pozos y el wumpus ya que si te topas con ellos perderás el juego.

CONTROLES:

Flecha derecha: mueve al cazador una posición a la derecha
Flecha abajo: mueve al cazador una posición hacia abajo
Flecha izquierda: mueve al cazador una posición a la izquierda
Flecha arriba: mueve al cazador una posición hacia arriba

Tecla ESPACIO: recorre automáticamente el tablero
Tecla S: encuentra automáticamente el camino más corto hacia el objetivo
Teclado P: muestra de nuevo este mensaje

Aceptar

Enlace a repositorio

<https://github.com/DiegoSamayoaGomez/Wumpus-IA.git>