



UA

Wumpus

Daniel Asensi Roch
DNI: 48776120C

12 de noviembre de 2023

Índice

1. Introducción	2
1.1. Planteamiento del problema	2
1.2. Objetivos	2
2. Estado del Arte	3
2.1. Agentes BDI	3
2.2. El Mundo de Wumpus	3
3. Metodología	5
3.1. Análisis de Estrategias	5
3.1.1. Estrategia para Moverse	5
3.1.2. Estrategia para evitar muertes	6
3.1.3. Estrategias para conseguir el oro	6
4. Resultados	7
4.1. Partes positivas de la implementación	7
4.2. Partes negativas de la implementación	8
4.3. Potenciales mejoras	8
5. Código Fuente	10
5.1. Especie jugador inicialización	10
5.2. Predicados	10
5.3. Sensores de percepción	10
5.4. Planes	11
6. Evaluación del curso y autorreflexión	13
7. Conclusiones	14

1. Introducción

En el ámbito de la inteligencia artificial, la resolución de problemas complejos y dinámicos representa un desafío constante que impulsa el desarrollo de nuevas metodologías y enfoques. Una de estas metodologías es el uso de agentes BDI (Belief-Desire-Intention) en plataformas de simulación, como la plataforma GAMA. Esta práctica se centra en la implementación y evaluación de un agente BDI para resolver el desafío presentado por el mundo de Wumpus.

1.1. Planteamiento del problema

El mundo de Wumpus es un entorno clásico en la inteligencia artificial, que presenta un escenario de cueva cuadrículada donde un agente debe recolectar tesoros mientras evita trampas y enfrenta al Wumpus, una criatura peligrosa. Este entorno es ideal para explorar el comportamiento de los agentes inteligentes y su capacidad para tomar decisiones basadas en la percepción y el razonamiento.

Los agentes BDI son un enfoque de modelado y diseño de agentes inteligentes que se basa en tres componentes fundamentales: creencias, deseos e intenciones. Estos componentes permiten al agente formar una representación interna del mundo, establecer metas y tomar decisiones racionales para alcanzar sus objetivos. En el contexto del mundo de Wumpus, el agente BDI utiliza sus creencias para interpretar percepciones como brisas o resplandores, indicativos de trampas o tesoros cercanos, respectivamente. Los deseos del agente pueden incluir recolectar tesoros y evitar peligros, mientras que sus intenciones guiarán sus acciones concretas en el entorno.

1.2. Objetivos

Los objetivos de esta práctica son múltiples y se detallan a continuación:

- Familiarización con Agentes BDI: Adquirir un conocimiento sólido sobre el concepto de agentes BDI (Belief-Desire-Intention), comprendiendo cómo estas entidades teóricas pueden ser utilizadas para modelar y resolver problemas complejos en entornos dinámicos.
- Resolución del Mundo de Wumpus: Utilizar el entorno del mundo de Wumpus como un caso de estudio para explorar y demostrar la eficacia de los agentes BDI. Este objetivo incluye el desarrollo de estrategias para que el agente recolecte tesoros y evite peligros como trampas y el Wumpus.
- Toma de Decisiones Basada en Creencias, Deseos y Metas: Profundizar en el entendimiento de cómo un agente BDI toma decisiones racionales basadas en sus creencias (información sobre el entorno), deseos (objetivos a alcanzar) y metas (planes de acción concretos).
- Evaluación de Rendimiento y Eficacia: Evaluar críticamente el rendimiento y la eficacia del agente BDI implementado, analizando cómo se adapta y responde a diferentes configuraciones y desafíos presentados en el mundo de Wumpus.

2. Estado del Arte

La combinación de agentes BDI y el mundo de Wumpus representa un área de estudio fascinante en la inteligencia artificial. Los agentes BDI, con su estructura de creencias, deseos e intenciones, son particularmente adecuados para navegar y tomar decisiones en el entorno desafiante que presenta el mundo de Wumpus. Esta intersección ha sido explorada para estudiar cómo los agentes pueden formar representaciones internas del mundo, planificar bajo incertidumbre y adaptarse a entornos dinámicos.

2.1. Agentes BDI

- **Orígenes y Desarrollo Teórico:** Los agentes BDI, basados en los conceptos de creencias (Beliefs), deseos (Desires) e intenciones (Intentions), tienen sus raíces en la teoría filosófica de la acción humana, especialmente en los trabajos de Bratman sobre la teoría de la acción intencional. Su desarrollo en la inteligencia artificial comenzó en la década de 1980, con investigadores como Michael Wooldridge y Nicholas Jennings contribuyendo significativamente a su formalización y aplicación práctica. [1]
- **Principios Clave:** Los agentes BDI se caracterizan por su capacidad para representar y razonar sobre el mundo (creencias), formular objetivos (deseos) y elaborar planes para alcanzar esos objetivos (intenciones). Esta estructura permite un modelado más cercano a la toma de decisiones humana, lo que los hace adecuados para entornos complejos y cambiantes. [2]
- **Aplicaciones y Avances Recientes:** Los agentes BDI han encontrado aplicaciones en diversas áreas, desde videojuegos y simulaciones hasta sistemas de soporte a la decisión en negocios y logística. Los avances recientes incluyen la integración de técnicas de aprendizaje automático para mejorar la adaptabilidad y eficiencia de estos agentes. [2]

2.2. El Mundo de Wumpus

Como se expone en el siguiente artículo sobre el Mundo de Wumpus [3]

- **Origen y Evolución:** El mundo de Wumpus fue introducido por Gregory Yob en 1975 como un juego de computadora simple. Se ha convertido en un entorno de prueba estándar en la inteligencia artificial para demostrar y evaluar la capacidad de razonamiento y toma de decisiones de los agentes inteligentes.
- **Características del Entorno:** El mundo de Wumpus es un laberinto cuadrícula donde un agente debe navegar, recolectar tesoros y evitar peligros como trampas y el Wumpus. Este entorno presenta desafíos únicos como la necesidad de inferir la ubicación de peligros basándose en percepciones indirectas (como el olor o una brisa).
- **Uso en Investigación y Educación:** Ha sido ampliamente utilizado en la educación en inteligencia artificial para enseñar conceptos como la lógica, la planificación y el

razonamiento bajo incertidumbre. Además, ha servido como un campo de pruebas para nuevas teorías y modelos en la IA.

3. Metodología

La metodología empleada en esta práctica se centra en el uso de un agente BDI para navegar y resolver desafíos en el mundo de Wumpus. El agente está programado para reaccionar a diferentes estímulos del entorno, como la presencia de oro, brisas, olores y trampas, y tomar decisiones basadas en sus creencias, deseos e intenciones.

3.1. Análisis de Estrategias

El agente utiliza varias estrategias para navegar por el entorno y alcanzar sus objetivos. Estas estrategias se basan en la percepción del entorno y en la toma de decisiones inteligente para evitar peligros y alcanzar el objetivo de recolectar oro.

3.1.1. Estrategia para Moverse

La estrategia de movimiento del agente en el mundo de Wumpus es crucial para explorar eficientemente el entorno y alcanzar los objetivos de evitar peligros y recolectar oro. Esta estrategia se basa en una serie de percepciones y decisiones que guían al agente a través del laberinto.

1. **Movimiento Aleatorio Inicial:** Al inicio, el agente añade un deseo de movimiento aleatorio (*move_random_desire*). Esto le permite comenzar a explorar el entorno de manera no dirigida, lo cual es esencial en las etapas iniciales cuando no se ha detectado oro ni peligros.
2. **Percepción y Reacción a Estímulos:** El agente está programado para percibir diferentes estímulos del entorno, como áreas de brillo (*glitterArea*), brisas (*breezeArea*) y olores (*odorArea*). Cada uno de estos estímulos afecta su estrategia de movimiento:
 - Al detectar brillo, el agente se enfoca en esa ubicación y cambia su estrategia de movimiento para acercarse al oro.
 - Al percibir brisas u olores, el agente activa un deseo de escape para evitar peligros, lo que altera temporalmente su patrón de movimiento.
3. **Estrategia de Patrullaje:** Cuando el agente no está en modo de escape o búsqueda de oro, sigue un plan de patrullaje (*patrolling*) que implica moverse a una de las celdas vecinas de manera aleatoria. Este movimiento se realiza a través de la selección aleatoria de un vecino y el desplazamiento hacia esa celda.
4. **Cambio de Estrategia según el Contexto:** El agente es capaz de cambiar su estrategia de movimiento en función de las percepciones y los deseos activos. Por ejemplo, al detectar un peligro, cambia a un modo de escape, y al detectar oro, cambia a una estrategia de búsqueda enfocada.

3.1.2. Estrategia para evitar muertes

1. **Detección de Peligros:** El agente está programado para detectar áreas de brisa y olor, indicativas de trampas y la presencia del Wumpus, respectivamente. Al percibir estos estímulos, el agente activa un deseo de escape (*want_escape*) para evitar estos peligros.
2. **Retroceso Inteligente:** En respuesta a la detección de peligros, el agente ejecuta un plan de escape (*escape*) que lo hace retroceder a la celda anterior, evitando así entrar en zonas peligrosas.
3. **Contabilización de Trampas Evitadas:** El agente lleva un registro de las trampas evitadas (*trampasEvitadas*), lo que permite evaluar la eficacia de la estrategia de evasión.

3.1.3. Estrategias para conseguir el oro

1. **Detección de Oro:** El agente percibe áreas con brillo (*glitterArea*), que indican la proximidad del oro. Esta percepción activa un deseo específico (*getGold*) para buscar el oro.
2. **Navegación Hacia el Oro:** Una vez activado el deseo de obtener oro, el agente ejecuta un plan (*exploreAroundGlitter*) para explorar las celdas cercanas a la ubicación detectada del brillo, aumentando así las posibilidades de encontrar el oro.
3. **Finalización de la Búsqueda:** Al encontrar el oro (*goldArea*), el agente elimina todas las intenciones y deseos actuales y pausa la simulación, indicando el éxito en la consecución del objetivo.

4. Resultados

La implementación del agente BDI en el entorno del mundo de Wumpus ha demostrado ser efectiva en lograr los objetivos principales: recolectar oro y evitar la muerte por peligros como trampas o el Wumpus. Sin embargo, el análisis de la estrategia y el comportamiento del agente revela que, aunque es exitoso, no opera de manera óptima. A continuación, se detallan los resultados clave y las áreas de mejora identificadas.

4.1. Partes positivas de la implementación

- **Éxito en la Evitación de Peligros:** El agente ha demostrado una capacidad notable para evitar peligros, como se evidencia en el registro de trampas evitadas (trampasEvitadas). Esto indica que la estrategia de percepción y reacción ante estímulos peligrosos es efectiva. **La media de los steps realizados para encontrar el oro en 20 simulaciones han sido 1300**
- **Recolección de Oro:** El agente ha sido capaz de localizar y recolectar el oro en el entorno del mundo de Wumpus. La estrategia de moverse hacia áreas de brillo y explorar alrededor de estas zonas ha resultado ser una táctica exitosa. Como podemos ver en la figura 2.

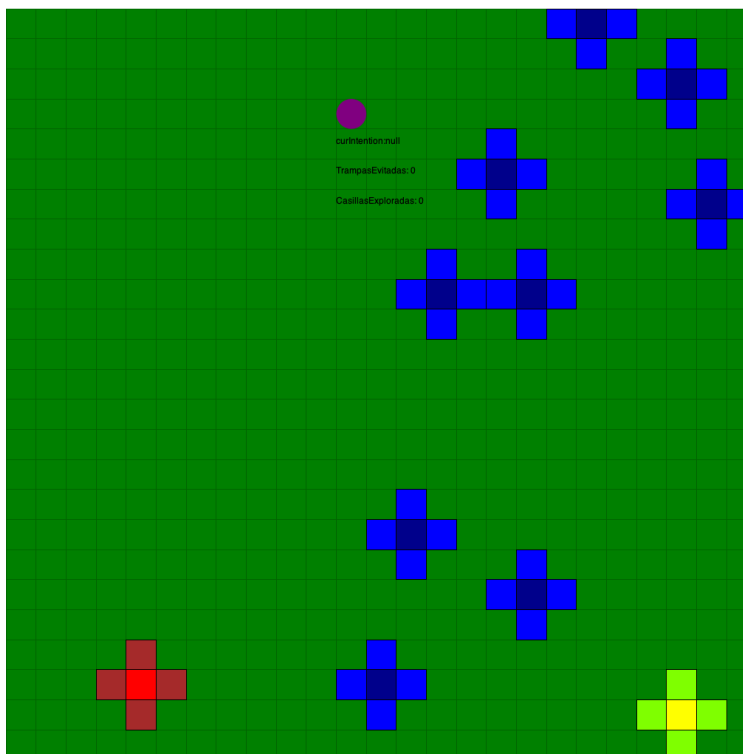


Figura 1: Inicio del mapa

Una vez ejecutada la simulación obtenemos que el agente ha llegado al oro.

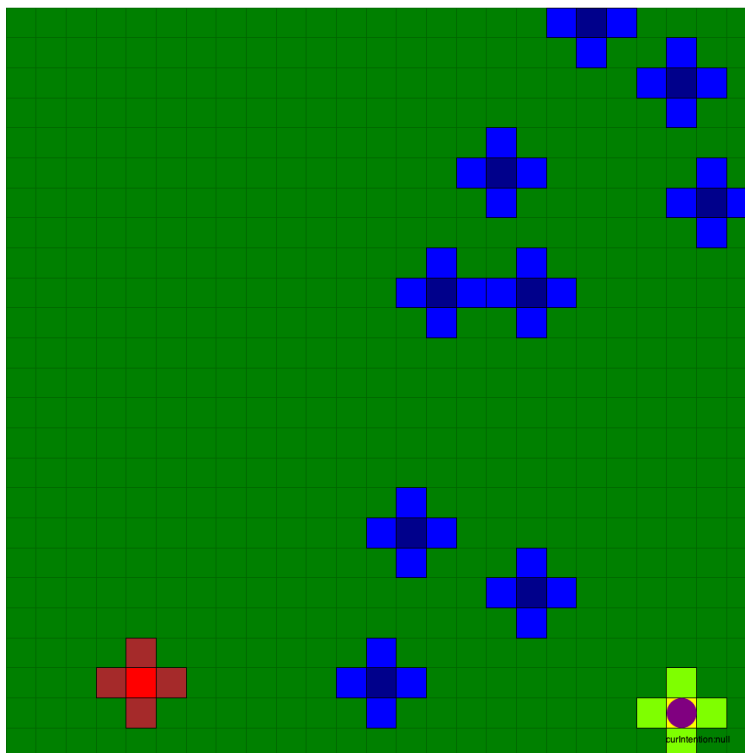


Figura 2: Mapa solucionado

Por otro lado, también hemos obtenido la cantidad de movimientos realizados por el agente, los cuales aparecen marcados en rojo y en azul la cantidad de trampas evitadas [3](#)

Además, para mayor control del agente se le ha añadido a su aspecto un texto que nos da información de sus intenciones, como se puede ver en la siguiente figura [4](#).

4.2. Partes negativas de la implementación

- **Falta de Optimización en el Movimiento:** A pesar de lograr sus objetivos, el agente no muestra un patrón de movimiento optimizado. La estrategia de movimiento aleatorio, aunque efectiva para la exploración inicial, puede llevar a una exploración redundante o ineficiente de las celdas.
- **Ineficiencia en la Búsqueda de Oro:** Mientras que el agente puede encontrar y recolectar el oro, la estrategia no está optimizada para hacerlo de la manera más rápida o eficiente. El agente a menudo explora celdas innecesarias o toma rutas indirectas hacia el oro.

4.3. Potenciales mejoras

- **Potencial para Mejoras:** Los resultados indican que hay un margen significativo para mejorar la eficiencia del agente. Esto podría incluir la implementación de algoritmos

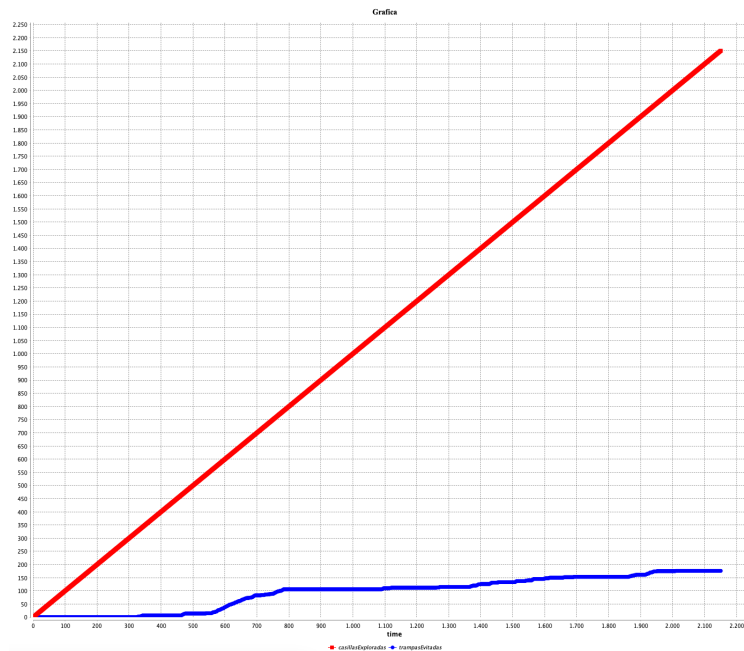


Figura 3: Gráfica de ciclos, movimientos y trampas evitadas

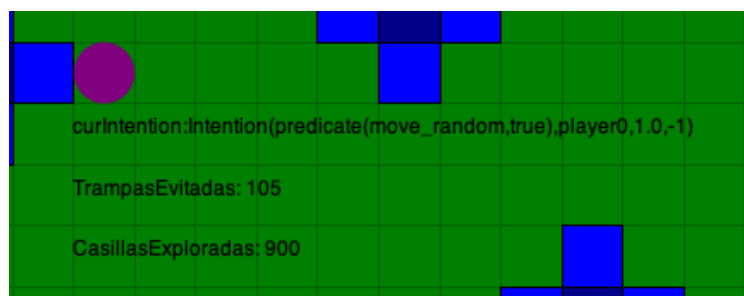


Figura 4: Información en aspecto del agente

de búsqueda más sofisticados, como A* o búsqueda en profundidad, para optimizar la ruta hacia el oro y minimizar la exploración innecesaria. [4]

- **Balance entre Seguridad y Eficiencia:** Un aspecto crucial en la mejora del agente será encontrar un equilibrio óptimo entre evitar peligros y buscar rutas eficientes. Esto requerirá un ajuste más fino de las percepciones y las decisiones del agente.

5. Código Fuente

Al tener la plantilla generada por el profesor, todos los alumnos, adjuntaré el código implementado por mí, en la especie *Player*

5.1. Especie jugador inicialización

```
1  /**Agente del jugador */
2  species player skills: [moving] control: simple_bdi {
3
4
5  // Posicion de inicio del jugador
6  gworld maCellule <- one_of(gworld);
7  // Posicion anterior del jugador
8  gworld pastCellule <- maCellule;
9
10
11  init {
12    location<-maCellule.location;
13    do add_desire(move_random_desire);
14
15  }
16
```

5.2. Predicados

```
1  predicate move_random_desire <- new_predicate("move_random");
2  predicate want_escape <- new_predicate("want_escape", true);
3  predicate getGold <- new_predicate("getGold", true);
4
5  int trampasEvitadas <- 0;
6  int casillasExplorada <- 0;
7
```

5.3. Sensores de percepción

```
1  /**Detecta una zona de brillo y para al agente encima */
2  perceive target:glitterArea in: 1{
3
4    focus id:"glitterLocation" var:location strength:10.0;
5
6    ask myself{
7      do remove_intention(want_escape, true);
8      do remove_intention(move_random_desire, true);
9    }
10
11  }
12
```

```

13
14 //Detecta el tesoro
15 perceive target:goldArea in: 1{
16
17     ask myself{
18         do remove_belief( get_predicate(get_belief_with_name("glitterLocation"))
19     );
19     do remove_intention(getGold,true);
20     do remove_intention(want_escape, true);
21     do remove_intention(move_random_desire, true);
22     ask world{
23         do pause;
24     }
25
26 }
27
28 }
29
30 /**Detecta una zona de brisa y hace retroceder al agente una casillas */
31 perceive target:breezeArea in: 1{
32
33     ask myself{
34         do remove_intention(move_random_desire, true);
35         do add_desire(want_escape);
36     }
37 }
38
39 /**Detecta una zona de edor y hace retroceder al agente una casillas */
40 perceive target:odorArea in: 1{
41
42     ask myself{
43         do remove_intention(move_random_desire, true);
44         do add_desire(want_escape);
45     }
46 }

```

5.4. Planes

```

1
2 //Creencia de busqueda de oro
3 rule belief: new_predicate(glitterLocation) new_desire: getGold;
4
5
6 //Exploracion celulas cercanas al oro
7 plan exploreAroundGlitter intention: getGold {
8
9     map glitter <- get_predicate(get_belief_with_name("glitterLocation")).
10    values("location_value");
11    gworld celdaGlitter <- gworld(glitter['location_value']);
12    gworld celdaActual <- gworld({location.x, location.y});

```

```
13
14   if (celdaGlitter = celdaActual){
15     list<gworld> vecinos <- [];
16     ask celdaGlitter{
17       vecinos <- neighbors;
18     }
19
20     gworld vecino <- one_of(vecinos);
21     casillasExplorada <- casillasExplorada + 1;
22     do goto target: vecino on: gworld speed: 24.0;
23   }
24   else{
25     casillasExplorada <- casillasExplorada + 1;
26     do goto target: celdaGlitter on: gworld speed: 24.0;
27   }
28
29
30 }
31
32
33
34 /**Hace escapar al agente retrocediendo una casilla */
35 plan escape intention:want_escape {
36   do goto target: pastCellule on: gworld speed: 24.0;
37   trampasEvitadas <- trampasEvitadas + 1;
38   casillasExplorada <- casillasExplorada + 1;
39   do remove_intention(want_escape, false);
40   do add_desire(move_random_desire);
41 }
42
43
44
45 /**Plan de patrulla para moverse por el mapa libremente */
46 plan patrolling intention:move_random_desire {
47
48   gworld playerPosition <-gworld({location.x, location.y});
49   pastCellule <- playerPosition;
50
51   list<gworld> vecinos <- [];
52   ask playerPosition {
53     vecinos <- neighbors;
54   }
55
56   /**Me guardo mi posicion anterior */
57   gworld vecino <- one_of(vecinos);
58   casillasExplorada <- casillasExplorada + 1;
59   do goto target: vecino on: gworld speed: 24.0;
60 }
```

6. Evaluación del curso y autorreflexión

¿Qué te gustó?

Lo que más me gustó de trabajar en este proyecto fue la oportunidad de aplicar conceptos teóricos de inteligencia artificial en un entorno práctico y desafiante. La implementación de un agente BDI en el mundo de Wumpus me permitió explorar de manera concreta cómo las creencias, deseos e intenciones pueden influir en la toma de decisiones de un agente. Además, me fascinó ver cómo un conjunto de reglas y comportamientos relativamente simples pueden dar lugar a un comportamiento emergente complejo y adaptativo en el agente.

¿Qué desafíos encontraste?

Uno de los principales desafíos que enfrenté durante este proyecto fue la adaptación a la plataforma GAMA y su lenguaje de programación específico, GAML. Aprender GAML, con su sintaxis y estructura únicas, fue inicialmente una tarea ardua, especialmente al modelar comportamientos complejos de agentes como los BDI. Además, la depuración y optimización de los modelos en GAMA requerían un análisis detallado y experimentación iterativa, lo que a veces se veía complicado por la limitada documentación y recursos de aprendizaje. Este proceso no solo implicaba comprender teóricamente los conceptos avanzados de inteligencia artificial, sino también aplicarlos efectivamente en un entorno de simulación específico, lo que representó un reto significativo pero a la vez enriquecedor en mi desarrollo profesional.

¿Cómo crees que podrías aplicar lo que aprendiste en el futuro?

Los conocimientos y habilidades adquiridos en este proyecto son altamente transferibles a otros ámbitos de la inteligencia artificial y la robótica. Por ejemplo, la comprensión de cómo diseñar agentes que pueden tomar decisiones basadas en información incompleta o incierta es crucial en el desarrollo de sistemas autónomos, como vehículos autónomos o robots de exploración. Además, la experiencia en la implementación de algoritmos de búsqueda y navegación puede ser aplicada en el desarrollo de software para la planificación de rutas en logística o en aplicaciones de mapeo. Finalmente, este proyecto ha reforzado mi capacidad para abordar problemas complejos y desarrollar soluciones creativas, habilidades que son valiosas en cualquier campo tecnológico o de investigación.

7. Conclusiones

Referencias

- [1] S. Bussmann, N. R. Jennings, M. Wooldridge, S. Bussmann, N. R. Jennings, and M. Wooldridge, “Agent-based production control,” *Multiagent Systems for Manufacturing Control: A Design Methodology*, pp. 9–52, 2004.
- [2] Contributors to Wikimedia projects, “Belief–desire–intention software model - Wikipedia,” Oct. 2023, [Online; accessed 12. Nov. 2023]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Belief-desire-intention_software_model&oldid=1178367320
- [3] J. Perry and H. E. Sevil, “A multi-agent adaptation of the rule-based wumpus world game,” *International Journal of Artificial Intelligence and Soft Computing*, vol. 7, no. 4, pp. 299–312, 2022.
- [4] V. T. T. Mariano, F. d. J. N. Cárdenas, E. A. Hernández *et al.*, “Análisis de algoritmos de búsqueda en espacio de estados.” *Ciencia Huasteca Boletín Científico de la Escuela Superior de Huejutla*, vol. 3, no. 5, 2015.

