Universidad Tecnológica Nacional
AJEA – Actas de Jornadas y Eventos Académicos de UTN
Trabajo Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1c36
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Colección JIT en AJEA https://rtyc.utn.edu.ar/index.php/ajea/JITs

Jóvenes Investigadores Tecnológicos
Facultad Regional Rafaela 26 y 27 de octubre de 2023

Área: Sistemas de información e informática / Categoría: Estudiante de grado / Regional: Facultad Regional Santa Fe

Implementación de Agente Inteligente: Comparación de Estrategias de Búsqueda en Pokémon Unite.

Intelligent Agent Implementation: Comparison of Search Strategies in Pokémon Unite.

Valentín Fontana

Dpto. Ing. en Sistemas de Información, Facultad Regional Santa Fe, Universidad Tecnológica Nacional

valefontana15@gmail.com

Tomás Assenza

Dpto. Ing. en Sistemas de Información, Facultad Regional Santa Fe, Universidad Tecnológica Nacional

tassenza@frsf.utn.edu.ar

Benjamín Adam

Dpto. Ing. en Sistemas de Información, Facultad Regional Santa Fe, Universidad Tecnológica Nacional

benjaadam14@gmail.com

Resumen

En el contexto de un trabajo práctico de la materia "Inteligencia Artificial", se desarrolló un agente inteligente capaz de jugar una versión simplificada de "Pokémon Unite", donde el agente debe navegar un mapa, vencer a pokemones adversarios y llegar al "pokémon maestro" y vencerlo para ganar. Se utilizó el framework FAIA para implementar al agente y se creó una interfaz gráfica de usuario utilizando FXGL y JavaFX para visualizar el juego.

En el trabajo se describe, en detalle, la resolución incluyendo la representación del escenario del juego, la definición del estado del agente, el objetivo, cómo el agente obtiene las percepciones y las acciones que puede ejecutar. Por último, se evaluó el desempeño del agente utilizando distintas estrategias de búsqueda como BFS y Greedy en diferentes escenarios controlados, registrando la energía final del agente, luego de lograr el objetivo.

Palabras claves: Inteligencia Artificial, Agente, Algoritmos de búsqueda

Abstract

In the context of a practical assignment for the "Artificial Intelligence" course, an intelligent agent was developed capable of playing a simplified version of "Pokémon Unite", where the agent navigates a map to defeat opposing pokémons, reach the "master Pokémon," and defeat it to win. The FAIA framework was used to implement the agent, and a graphical user interface was created using FXGL and JavaFX to visualize the game.

This paper provides a detailed description of the solution, including the representation of the game scenario, the definition of the agent's state and goal, how the agent obtains its perceptions, and the actions that it can perform. Finally, the agent's performance was evaluated using different search strategies such as BFS and Greedy in various controlled scenarios, recording the agent's final energy after achieving the objective.

Keywords:

AJEA – Actas de Jornadas y Eventos Académicos de UTN
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Texto Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1636

Introducción

En el marco de la realización de un trabajo práctico integrador para la materia de 5° año “Inteligencia Artificial”, se desarrolló un agente inteligente que pudiera jugar un juego generado por la cátedra de forma autónoma sin intervención de un usuario.

El juego propuesto consta de una versión simplificada de “Pokémon Unite”, modelando a un agente que debe recorrer un mapa establecido, superando dificultades que se presentan a medida que avanza sobre el mismo.

El objetivo a resolver por el agente es, a partir de un nodo aleatorio de aparición, llegar al nodo de destino recorriendo el mapa y venciendo a los distintos pokemones adversarios que se presentan en su camino. Al llegar al nodo final se encuentra con el “pokémon maestro”, al que el agente debe vencer para ganar el juego. En su camino además, el agente puede juntar pokebolas que le permiten incrementar su energía para poder pelear contra sus adversarios. Además, según el nivel de energía que posea puede utilizar distintos poderes para derrotar a sus adversarios.

Como trabajo previo se cuenta con un software generado por la cátedra llamado FAIA (Framework para un Agente que resuelve problemas de IA), el cual cuenta con distintos algoritmos de búsqueda,ya generados, tanto no informados (DFS, BFS), e informados (Greedy, A*) de manera que el foco se encuentra en la correcta implementación de los mismos [1].

Para mostrar de modo didáctico el comportamiento del agente dentro del mapa, se generó una GUI (Graphical User Interface - Interfaz Gráfica de usuario) haciendo uso de la herramienta FXGL, que es un potente motor de juegos que permite la generación de entidades con propiedades tales como: la velocidad o la aceleración, la visualización gráfica y la sencilla integración en distintos sistemas operativos por hacer uso del potente Framework multiplataforma JavaFX [2].

Existen en la literatura distintos trabajos referidos al modelado de agentes inteligentes. Korf [3] presenta el estudio de cómo los distintos algoritmos de búsqueda pueden ser aplicados a la resolución de distintas clases de problemas en la Inteligencia Artificial, estudiando también la eficiencia de los mismos. Abu Naser [4] ,por otro lado, presentó un trabajo sobre la generación de una herramienta de visualización de algoritmos de búsqueda, y cómo ésta ayudó a los estudiantes a mejorar su comprensión.

Tomando como punto de partida estas experiencias previas, se presenta en este trabajo la generación del agente que resuelve el juego propuesto mediante el framework FAIA, acompañado de una interfaz gráfica de usuario utilizando FXGL y la herramienta gráfica JavaFX.

Metodología

En esta sección, se describe la metodología utilizada para abordar el problema planteado en este trabajo.

Inicialmente, antes de implementar el agente en Java utilizando el framework FAIA, se definió de manera conceptual el escenario, las percepciones que obtiene el agente, los poderes que puede utilizar, el estado, el objetivo que se debe alcanzar y las distintas acciones que puede ejecutar el agente para alcanzarlo.

Escenario del juego

En base al mapa provisto por la cátedra (Figura 1), lo que se realizó fue representarlo conceptualmente en una estructura de grafo no dirigido, que permitía definir los distintos nodos y caminos entre ellos. (Figura 2)

AJEA – Actas de Jornadas y Eventos Académicos de UTN
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Texto Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1636

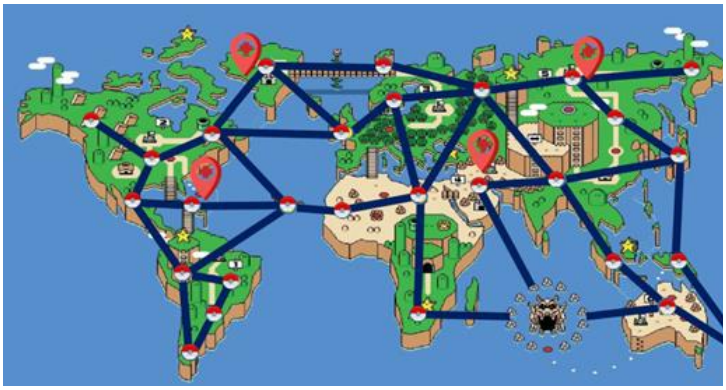

Figura 1. Mapa original provisto por la cátedra

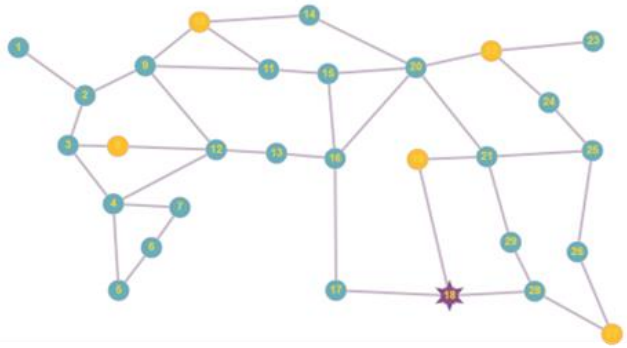

Figura 2. Grafo simplificado del mapa

El grafo representado en la figura 2 contiene a los nodos estándar en azul, estos son aquellos donde puede o no haber un pokémon enemigo (elegidos aleatoriamente al momento de la generación del mapa). En amarillo se encuentran los nodos que tienen pokebolas (siendo éstas a modo de ejemplo, pero luego en la implementación las mismas aparecen aleatoriamente), y el nodo donde se encuentra el pokémon maestro (al que el agente debe vencer para ganar el juego) en formato de estrella morada.

Estado del agente

Definimos al estado del agente como un arreglo compuesto por los siguientes elementos: **[ubicación, grafo, energía, *energiaInicial*, energiaGanada, nivel, habilidadesEspeciales, cantidadAdversarios]**

ubicación: Es un nodo, que representa el nodo del grafo en el que se encuentra el agente.

mapa: Es un grafo, que representa el mapa sobre el cual el agente se puede mover.

energía: Es un número real que representa los puntos de energía actual del agente.

energíaInicial: Número real que representa la energía con la que el agente empezó el juego.

nivel: Número entero entre 1 y 7, que representa el nivel del agente, y calculado según los puntos que haya ganado, según se definió en la tabla 1.

energíaGanada: Sumatoria de la energía ganada en cada pelea por el agente y de las pokebolas recolectadas.

habilidadesEspeciales: Es una matriz de 4 filas y dos columnas, donde cada fila representa una habilidad especial que el agente puede utilizar. La primera columna indica si la habilidad se puede usar actualmente (1 si se puede, 0 en caso contrario) y la segunda columna contiene un número entero que indica los ciclos que restan para volver a usar la habilidad. En la Tabla 1, se representa la matriz mencionada anteriormente.

AJEA – Actas de Jornadas y Eventos Académicos de UTN
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Texto Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1636

Habilitado para usar (Booleano)	Ciclos para usar ***Rayo Aurora*** (Integer)
Habilitado para usar (Booleano)	Ciclos para usar ***Rayo Meteórico*** (Integer)
Habilitado para usar (Booleano)	Ciclos para usar ***Rayo Solar*** (Integer)
Habilitado para usar (Booleano)	Ciclos para usar ***Satélite*** (Integer)

Tabla 1. Matriz de habilidades que puede utilizar el agente

cantidadAdversarios: Contiene un número natural que indica la cantidad de adversarios vivos en el ambiente. Cuando el agente vence un pokémon, éste se decrementa en una unidad.

Objetivo

El objetivo del agente es llegar al nodo donde se encuentra el maestro (nodo 18), con energía mayor a 0, y al llegar al nodo en el que se encuentra el maestro, vencerlo. Cuando esta condición se cumple, el agente gana el juego y finaliza la simulación.

Percepciones

El agente puede obtener información del ambiente para concretar el objetivo definido anteriormente. Para esto, el agente percibe información sobre el nodo donde se encuentra y además de los nodos adyacentes al que se encuentra. A continuación se describe la información que percibe el agente de cada uno de los nodos.

- tienepokémon (Variable booleana que indica si existe un pokémon en el nodo)
- tienePokebola (Variable booleana que indica si existe una pokebola en el nodo)
- pokémon (Objeto de tipo pokémon que contiene la energía del adversario e información sobre si es el maestro o no, en caso de que exista)
- pokebola (Objeto de tipo pokebola que contiene la energía de la pokebola, en caso de que exista).

A modo de ejemplo, en la Figura 3 se muestra una salida por consola de la percepción que obtiene el agente al estar en un nodo determinado.

```
--------------------------------------------------------------------------------
Percepcion Nodo 17: PercepcionNodo{, tienePokemon=false, tienePokebola=false}
Percepcion Nodo 18: PercepcionNodo{, tienePokemon=true, tienePokebola=false}
Percepcion Nodo 19: PercepcionNodo{, tienePokemon=false, tienePokebola=true}
Percepcion Nodo 28: PercepcionNodo{, tienePokemon=true, tienePokebola=false}
--------------------------------------------------------------------------------
```

Figura 3. Salida por consola de una percepción

En la figura 3, el agente se encuentra en el nodo 18 (Nodo representado en verde por ser el actual), donde se obtiene la percepción de que existe un pokémon, como así también de que existe un pokémon en un nodo vecino (Nodo 28), y que existe una pokebola en el Nodo 19. Esta información que obtiene el agente sobre el ambiente a partir de una percepción, luego la utiliza para actualizar su estado interno y a partir de esta información, poder determinar cuál es la acción más conveniente a ejecutar para lograr su objetivo, mediante la aplicación de algoritmos de búsqueda.

Acciones

A continuación, se definen las posibles acciones que puede tomar el agente:

- Ir a Nodo N: Permite al agente desplazarse a un nodo adyacente al nodo que se encuentra actualmente.
- Elegir usar Rayo Aurora:Permite al agente aumentar su energía en un 20% en caso de que esté disponible.
- Elegir usar Rayo Meteórico: Permite al agente aumentar su energía en un 30% en caso de que esté disponible.

AJEA – Actas de Jornadas y Eventos Académicos de UTN
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Texto Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1636

- Elegir usar Rayo Solar: Permite al agente aumentar su energía en un 50% en caso de que esté disponible.
- Juntar Pokebola: Permite al agente aumentar su energía al encontrar una pokebola en el mapa.
- Huir: Permite al agente escapar de un pokémon que se encuentra en el nodo actual, es decir desplazarse sin tener que pelear. Cabe destacar que al decidir usar esta acción, la energía del pokémon disminuye en ¼ de la energía de su oponente.
- Elegir usar Satélite: Permite al agente obtener la ubicación actual de todos los adversarios en el mapa.

Estrategias de búsqueda

Para elegir la mejor acción a ejecutar en un momento, con el fin de alcanzar el objetivo, el agente puede encontrarla mediante la utilización de distintos algoritmos. Los utilizados se explican a continuación:

Búsqueda Primero en Anchura (BFS): Esta estrategia explora el espacio de búsqueda por niveles, comenzando desde el nodo inicial y expandiendo gradualmente los nodos vecinos antes de adentrarse en niveles más profundos. Se utilizó para determinar una solución que minimice la cantidad de acciones requeridas para llegar al objetivo.

Búsqueda Voraz (Greedy Search): La estrategia de búsqueda voraz utiliza una heurística para evaluar los nodos y seleccionar el próximo nodo a explorar. En este caso, se empleó el algoritmo de Dijkstra como heurística para estimar la distancia desde el nodo actual al nodo objetivo. Esta estrategia busca soluciones localmente óptimas, tomando decisiones basadas en información disponible en el momento. Es decir, este método permite elegir la acción que minimiza la cantidad de nodos a recorrer según la información que posee en un nodo determinado.

Búsqueda de Costo Uniforme (UCS): Esta estrategia asigna un costo a cada acción y busca la secuencia de acciones de menor costo. Para implementar la función de costo, se utilizó una función monótona creciente que considera la sumatoria de la energía utilizada en las acciones para lograr el objetivo.

Definición del Escenario

Para llevar a cabo la evaluación de las estrategias de búsqueda, se definió un escenario controlado en el entorno pokémon. Los parámetros del escenario incluyen el nodo inicial del agente, la energía inicial del agente, los nodos donde se ubican los pokemones, las posiciones de las pokebolas, la energía de los pokemones adversarios y la energía del pokémon maestro. Estos parámetros se mantuvieron constantes en todas las ejecuciones de las estrategias para garantizar la comparabilidad de los resultados.

Procedimiento de Evaluación

Se llevaron a cabo varias ejecuciones de cada estrategia de búsqueda utilizando el escenario definido. Cada ejecución consistió en simular el proceso de toma de decisiones del agente, siguiendo las acciones determinadas por la estrategia de búsqueda correspondiente. Se registraron los resultados de cada ejecución, incluyendo la energía final del agente, las acciones tomadas y si se cumplió el objetivo de vencer al pokémon maestro.

Posteriormente, se realizaron pruebas de rendimiento en diferentes condiciones, como escenarios aleatorios con nodos y energías variables, con el fin de evaluar la robustez y adaptabilidad de las estrategias de búsqueda en diferentes situaciones.

Resultados

Los resultados obtenidos de las ejecuciones se analizaron para evaluar el rendimiento de cada estrategia. Para evaluar el comportamiento de los algoritmos no informados se utilizó BFS, y para evaluar el comportamiento de los algoritmos informados

AJEA – Actas de Jornadas y Eventos Académicos de UTN
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Texto Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1636

se utilizó Greedy. En la Tabla 2 se presentan una serie de pruebas realizadas, donde se observa la energía final con la que terminó el agente en las distintas iteraciones realizadas para cada algoritmo.

BFS	**Greedy**
9.48	13.96
9.33	16.05
16.78	12.27
11.41	10.47
10.17	18.20
14.19	11.61
9.79	11.94
16.31	10.79
Perdió	17.14
E Promedio = 11.82	**E Promedio = 13.19**

Tabla 2. Ejecución y comparación de escenarios aleatorios

Conclusión

La búsqueda no informada en anchura (BFS) resultó ser una estrategia efectiva para encontrar el camino a la solución con menor cantidad de acciones ejecutadas. Sin embargo, la desventaja de esta estrategia radica en que no considera información sobre cuán lejos está, a partir de su estado actual al estado objetivo, (no tiene una función heurística que le permita estimar este dato). Por lo que no es capaz de tomar decisiones más eficientes. Además, en algunos casos, puede requerir un tiempo de ejecución mayor debido a la exploración de nodos innecesarios, y además mayor consumo de memoria, lo que en algunas ejecuciones provocó que no se pueda llegar a una solución ya que el consumo de memoria era excesivo. Por otro lado, la búsqueda informada (Greedy), permite una exploración más inteligente de los nodos, tomando en cuenta información sobre el estado actual y calculando una secuencia de acciones, utilizando una función heurística para llegar objetivo, lo que permite tomar decisiones más eficientes. La búsqueda informada con el algoritmo de Dijkstra permitió explorar menos nodos y llegar al objetivo con una energía mayor a la del algoritmo no informado.

Referencias bibliográficas

[1] Roa, J., Pividori, M., Gutiérrez, M., Stegmayer, G. How to Develop Intelligent Agents with FAIA. In: Francisco V. Cipolla Ficarra, AINCI and ALAIPO, eds.: Quality and Communicability for Interactive Hypermedia Systems: Concepts and Practice for Design, IGI Global, USA, 2010. DOI:10.4018/978-1-61350-456-7.ch209. p. 120-140.

[2] Baimagambetov, A. (2022). *Learn JavaFX Game and app development: With FXGL 17*. Brighton: Apres. DOI: 10.4018/978-1-61350-456-7.ch209

AJEA – Actas de Jornadas y Eventos Académicos de UTN
Libro de actas de resúmenes DOI: https://doi.org/10.33414/ajea.1640.2024
Texto Completo Publicado en AJEA – Full Text Published in AJEA
https://rtyc.utn.edu.ar/index.php/ajea/article/view/1636

[3] Korf, R.E. (1998) “Artificial Intelligence Search Algorithms”. University of California. Disponible en <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e631be2638577c34fbfce510fe2a17a48d7a11e4>.

[4] Abu Naser, S.S. (2008). “Developing Visualization Tool for Teaching AI Searching Algorithms”, *Information Technology Journal, 7.* ISSN: 1812-5638