

UNIVERSIDAD DEL VALLE DE GUATEMALA

Inteligencia artificial

Sección 30

Ing. Javier Fong



Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

PROYECTO 1

Jose Pablo Santisteban Vargas, 21153

GUATEMALA, 4 de marzo de 2024

DISCUSIÓN

La implementación de los algoritmos de búsqueda en el laberinto ha sido una experiencia enriquecedora en mi aprendizaje como estudiante universitario de inteligencia artificial. A lo largo de este proceso, pude comprender en profundidad cómo funcionan algoritmos como BFS, DFS, A* y Greedy First Search, y cómo se aplican en la resolución de problemas prácticos como encontrar caminos en un laberinto.

Durante la adaptación de los algoritmos de búsqueda al laberinto, encontré algunas dificultades significativas. Una de las principales dificultades fue integrar eficientemente la representación del laberinto en la lógica de los algoritmos. La necesidad de definir cómo se representaría el laberinto en términos de estructuras de datos y cómo se calcularán las vecindades de cada celda para la exploración fue un desafío inicial. Además, adaptar los algoritmos para que manejen correctamente los muros y las celdas libres del laberinto añadió complejidad al proceso de implementación.

Otro desafío importante fue la implementación de las funciones de heurística. Calcular una heurística efectiva que guíe adecuadamente la búsqueda hacia la dirección correcta sin sobrecargar el proceso fue un desafío en sí mismo. Además, asegurarse de que estas heurísticas se integran correctamente en los algoritmos de búsqueda, como A* y Greedy First Search, y que influyeran en la dirección de la exploración de manera efectiva, requirió un análisis cuidadoso y pruebas iterativas.

¿En qué casos es mejor cada algoritmo?

- BFS (Breadth-First Search): Es ideal cuando se busca encontrar el camino más corto desde el inicio hasta el destino en laberintos donde todas las celdas tienen el mismo costo. Es especialmente útil cuando se necesita encontrar la solución más rápidamente sin preocuparse por el costo del camino.
- DFS (Depth-First Search): Es útil cuando se busca explorar exhaustivamente un laberinto y no necesariamente encontrar el camino más corto. Puede ser útil en laberintos con caminos muy largos o cuando solo se necesita encontrar una solución, no necesariamente la mejor.

- ***A (A-Star)**:** Es excelente cuando se necesita encontrar el camino más corto en laberintos donde las celdas tienen diferentes costos y se dispone de una buena heurística que puede guiar la búsqueda hacia la dirección correcta.
- **Greedy First Search:** Similar a A*, pero con una heurística más simplificada. Es útil cuando se necesita encontrar una solución rápidamente y la heurística proporciona una buena estimación del costo restante.
- **Depth Limited Search:** Es útil cuando se desea limitar la profundidad de la búsqueda para evitar caer en ciclos infinitos en laberintos donde DFS podría no terminar.

¿Alguno es siempre mejor que otro?

No, ningún algoritmo es siempre mejor que otro en todos los casos. La elección del algoritmo adecuado depende de varios factores, como la estructura del laberinto, los costos asociados con cada celda, la heurística disponible y los requisitos específicos del problema, como la velocidad de ejecución y la optimización del camino encontrado. Cada algoritmo tiene sus propias ventajas y limitaciones, y es importante seleccionar el que mejor se adapte a las necesidades y restricciones del problema en cuestión.

CONCLUSIÓN

En conclusión, la implementación de algoritmos de búsqueda en un laberinto ha sido una experiencia educativa valiosa que ha fortalecido mis habilidades en inteligencia artificial y programación. A través de este proyecto, he podido comprender en profundidad cómo funcionan algoritmos como BFS, DFS, A* y Greedy First Search, así como también la importancia de las heurísticas en la optimización de la búsqueda.

Adaptar estos algoritmos al contexto del laberinto y desarrollar funciones de heurística efectivas presentó desafíos interesantes que me permitieron aplicar mis conocimientos teóricos en situaciones prácticas. Además, explorar las ventajas y limitaciones de cada algoritmo en diferentes escenarios me ha proporcionado una comprensión más completa de sus aplicaciones y me ha preparado para enfrentar problemas similares en el futuro.

Link de repositorio

https://github.com/Jskempo/LAB1_IA/tree/PROYECTO1