



TECNOLÓGICO
NACIONAL DE MÉXICO



INTELIGENCIA ARTIFICIAL

Profesor:

Alcaraz Chavez Jesus Eduardo

Nombre:

Kevin Alejandro Cuevas Crisantos

Numero de Control:

21121503

Proyecto 1

Reporte Técnico

Proyecto 1: Visualización del Algoritmo A* para Búsqueda de Rutas Óptimas**

1. Introducción

El presente proyecto implementa y visualiza el algoritmo de búsqueda informada A* (A estrella), el cual es ampliamente utilizado en áreas como inteligencia artificial, robótica, videojuegos y sistemas de navegación, debido a su capacidad para encontrar rutas óptimas de manera eficiente.

El sistema fue desarrollado en **Python**, utilizando la biblioteca **Pygame**, con el objetivo de ofrecer una representación gráfica interactiva del funcionamiento del algoritmo, permitiendo al usuario definir un punto de inicio, un punto final y obstáculos dentro de una cuadrícula, para posteriormente observar el proceso de búsqueda de la ruta más corta.

2. Objetivo del Proyecto

Desarrollar una aplicación interactiva que permita:

- Comprender visualmente el funcionamiento del algoritmo A*.
- Simular la búsqueda de la ruta más corta entre dos puntos.
- Analizar el impacto de los obstáculos y los costos de movimiento.
- Facilitar el aprendizaje del algoritmo mediante una representación gráfica clara.

3. Descripción General del Algoritmo A*

El algoritmo A* es un algoritmo de búsqueda informada que combina:

- **Costo acumulado (g):** distancia real desde el nodo inicial.
- **Heurística (h):** estimación del costo desde el nodo actual hasta el nodo objetivo.
- **Función de evaluación (f):**

$$f(n) = g(n) + h(n)$$

Este enfoque permite priorizar los nodos que parecen más prometedores, reduciendo el espacio de búsqueda y garantizando la obtención de la ruta óptima siempre que la heurística sea admisible.

4. Estructura del Sistema

El proyecto está compuesto por los siguientes componentes principales:

4.1 Representación del Entorno

El entorno se modela como una cuadrícula de **10 × 10**, donde cada celda corresponde a un nodo. Cada nodo contiene información relevante para el algoritmo, incluyendo:

- Posición en la cuadrícula.
- Estado (inicio, fin, obstáculo o camino).
- Costos g, h y f.
- Referencia al nodo padre.

4.2 Clase Nodo

La clase Nodo representa cada celda de la cuadrícula y contiene:

- **Atributos principales:**
 - fila, col: posición.
 - g, h, f: valores de costo.
 - padre: referencia para reconstruir el camino.
 - numero: identificador único del nodo.
- **Métodos funcionales:**
 - Identificación de estado (inicio, fin, pared).
 - Actualización visual del nodo.
 - Dibujo del nodo en pantalla.

Esta estructura permite encapsular tanto la lógica del algoritmo como su representación gráfica.

5. Heurística Utilizada

Se emplea la **distancia Manhattan**, definida como:

$$h(n) = |x_1 - x_2| + |y_1 - y_2|$$

Esta heurística es adecuada para entornos en forma de cuadrícula y garantiza la optimalidad del algoritmo al no sobreestimar el costo real hacia el nodo objetivo.

6. Costos de Movimiento

El sistema contempla movimientos en ocho direcciones:

- **Horizontal y vertical:** costo = 10.
- **Diagonal:** costo = 14.

Esta diferenciación permite simular distancias reales aproximadas y aporta mayor realismo al cálculo del camino óptimo.

7. Funcionamiento del Algoritmo

El algoritmo A* sigue los siguientes pasos:

1. Inicializar el nodo de inicio con costo $g = 0$.
2. Insertar el nodo inicial en la lista abierta (cola de prioridad).
3. Seleccionar el nodo con menor valor f .
4. Evaluar sus vecinos válidos.
5. Actualizar costos y referencias si se encuentra un camino más corto.
6. Repetir el proceso hasta alcanzar el nodo final o agotar las opciones.
7. Reconstruir el camino óptimo utilizando los nodos padre.

La cola de prioridad se gestiona mediante la estructura `heapq`, garantizando eficiencia en la selección del siguiente nodo a explorar.

8. Interfaz Gráfica e Interacción

La aplicación permite al usuario interactuar de la siguiente manera:

- **Clic izquierdo:**
 - Primer clic: seleccionar nodo inicial.
 - Segundo clic: seleccionar nodo final.
 - Clics posteriores: colocar obstáculos.
- **Botón “Buscar”:** inicia la ejecución del algoritmo A*.
- **Botón “Salir”:** cierra la aplicación.

El camino encontrado se muestra visualmente en color azul, permitiendo identificar claramente la ruta óptima.

9. Validación y Resultados

Durante las pruebas realizadas:

- El algoritmo encontró correctamente la ruta más corta cuando existía.
- Se verificó el correcto manejo de obstáculos.
- El sistema detecta escenarios donde no existe un camino posible.
- La visualización permitió comprender claramente cada etapa del proceso.

10. Aplicaciones del Proyecto

Este proyecto puede aplicarse en:

- Sistemas de navegación y mapas.
- Videojuegos con inteligencia artificial.
- Simulación de robots móviles.
- Enseñanza de algoritmos de búsqueda informada.

11. Conclusiones

El desarrollo de este proyecto permitió comprender de manera práctica y visual el funcionamiento del algoritmo A*, evidenciando su eficiencia y aplicabilidad en problemas de búsqueda de rutas. La combinación de una estructura de datos adecuada, una heurística admisible y una interfaz gráfica interactiva facilitó el aprendizaje del algoritmo y demostró su importancia dentro del área de inteligencia artificial.

Este proyecto refuerza la comprensión de conceptos fundamentales como heurísticas, grafos, costos y optimización, sentando una base sólida para el desarrollo de sistemas inteligentes más complejos.