

Introducción

El algoritmo A* (A Estrella) es una de las técnicas más utilizadas en la resolución de problemas de búsqueda y optimización de rutas. Este proyecto implementa una versión interactiva en un entorno web, que permite visualizar y planificar rutas en una cuadrícula.

Objetivo

Implementar el algoritmo A* de manera eficiente y visualmente comprensible, utilizando HTML, CSS y JavaScript. La aplicación permite a los usuarios:

- Definir dimensiones de la cuadrícula.
- Establecer posiciones de inicio, final, waypoints y obstáculos.
- Visualizar la cuadrícula generada y el camino óptimo calculado.

Tecnologías Utilizadas

- **HTML5:** Estructura y formulario para entradas del usuario.
- **CSS3:** Diseño visual de la cuadrícula y elementos de interacción.
- **JavaScript:** Lógica del algoritmo A*, generación dinámica de la cuadrícula y visualización del camino.

Estructura del Proyecto

Componentes

1. Formulario de entrada:

- Permite al usuario configurar las dimensiones de la cuadrícula, posiciones inicial y final, waypoints y obstáculos.
- Los datos ingresados se procesan para construir la cuadrícula y ejecutar el algoritmo A*.

2. Visualización de la cuadrícula:

- Se utiliza CSS Grid para generar una cuadrícula adaptable al tamaño especificado.
- Colores y estilos identifican elementos como:
 - Obstáculos: Rojo (#ff6f6f).
 - Camino calculado: Verde (#76e6a2).
 - Punto inicial: Azul (#6fa6ff).
 - Punto final: Naranja (#ffa74f).

- Waypoints: Amarillo (#ffeb3b).
- 3. **Algoritmo A*:**
 - Lógica principal implementada en JavaScript, que incluye:
 - Nodo con información de costos **g**, **h** y **f**.
 - Expansión de sucesores considerando movimientos horizontales, verticales y diagonales.
 - Cálculo de la heurística usando la distancia euclidiana.
 - Generación del camino final con backtracking.

Archivos

- **HTML:** Define la estructura de la página y los formularios de entrada.
- **CSS:** Diseña el formulario, cuadrícula y colores de los elementos.
- **JavaScript:** Contiene el algoritmo A* y la lógica para generar y renderizar la cuadrícula.

Funcionalidades

1. **Ingreso de Datos:**
 - El usuario define:
 - Tamaño de la cuadrícula (filas y columnas).
 - Posiciones inicial y final.
 - Obstáculos y waypoints opcionales.
2. **Cálculo del Camino:**
 - La función `calculatePath()` procesa los datos, ejecuta el algoritmo A* y genera el camino óptimo.
 - Si no existe un camino válido, se muestra un mensaje de error.
3. **Visualización Dinámica:**
 - La cuadrícula se genera en tiempo real con el tamaño especificado.
 - Los elementos interactivos (inicio, final, waypoints, etc.) se resaltan con colores identificativos.

Desafíos y Soluciones

1. **Desafío:** Manejar grandes cuadrículas.
 - **Solución:** Uso de CSS Grid y optimizaciones en la lógica de renderizado para mantener el rendimiento.
2. **Desafío:** Garantizar que el camino se encuentre incluso con obstáculos.

- **Solución:** Implementación de validaciones en el algoritmo y retroalimentación al usuario.

Posibilidades de Expansión

- **Replanificación en tiempo real:**
 - Actualizar el camino al introducir nuevos obstáculos.

Cómo probar la práctica

- Descomprimir el zip y abrir en el navegador el archivo **html**.
- Utilizando el editor de código *Visual Studio Code*, ir al apartado de extensiones y descargar la extensión **Live Server** (ver *Imagen 1*). Una vez descargada, botón derecho sobre el archivo **html** y seleccionar la opción *Open with Live Server* (ver *Imagen 2*).

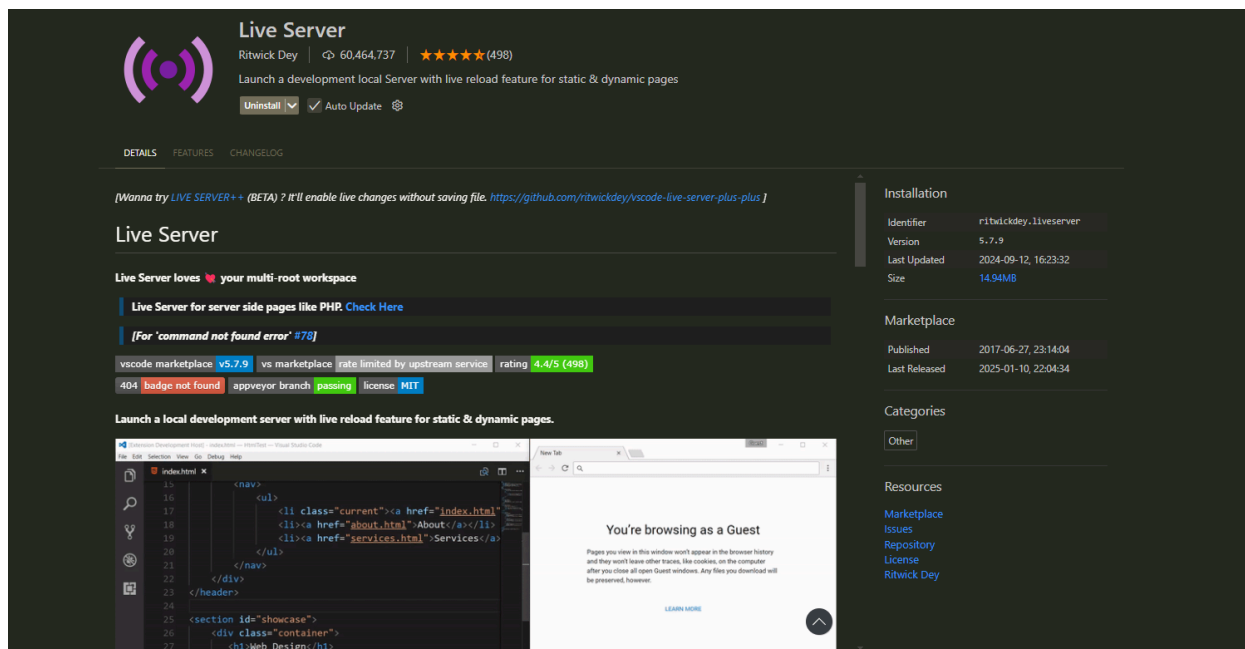


Imagen 1

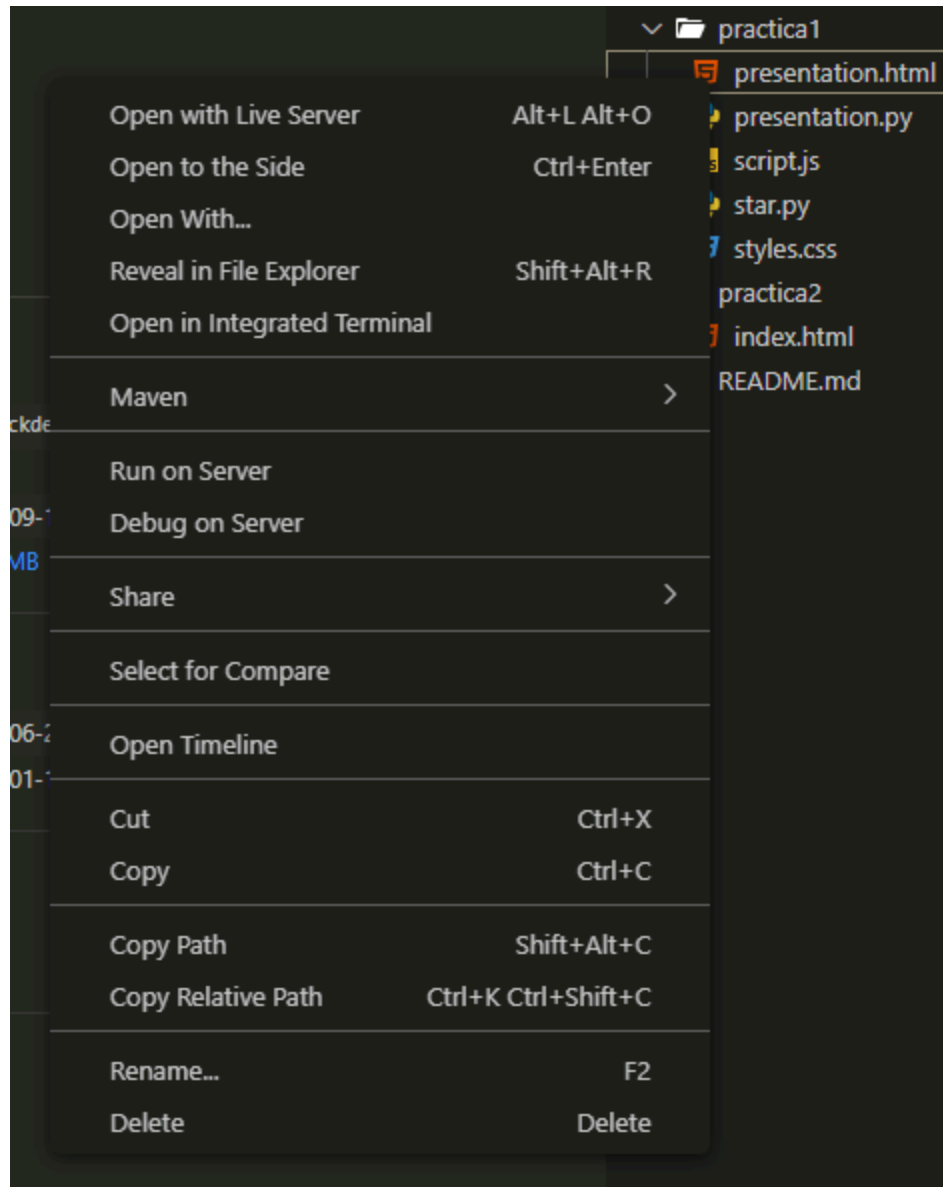


Imagen 2

Conclusión

El proyecto proporciona una implementación intuitiva y eficiente del algoritmo A*, facilitando la visualización y comprensión del funcionamiento del mismo. Las posibilidades de personalización y expansión hacen que esta herramienta sea adecuada tanto para educación como para aplicaciones prácticas.

José Waldo Villacres Zumaeta

Daniel Fernández Ortiz.