

Proyecto #1: Implementación de laberinto

Descripción General:

Para este primer proyecto, los estudiantes deberán implementar una solución que utilice el enfoque de los algoritmos *backtracking*. Se requiere el desarrollo del juego de laberinto clásico donde se debe resolver el encontrar la salida a partir de un punto de partida a través de diferentes caminos y bloqueos. El programa debe ser capaz de generar laberintos aleatorios, resolverlos desde un punto de partida definido o desde cualquier punto seleccionado por el usuario, y permitir guardar y cargar soluciones previamente generadas. El programa debe ser desarrollado utilizando Python y Flask para la interfaz gráfica.

Definición de backtracking: Estrategia de resolución de problemas que explora todas las posibles soluciones y retrocede cuando una solución parcial no es válida.

Requisitos:

1. Generación del Laberinto:

- a. El programa debe ser capaz de generar laberintos de diferentes tamaños, en una matriz de celdas (por ejemplo, 10x10, 20x20, etc.).
- b. Los laberintos deben tener un camino válido desde un punto de inicio hasta un punto de salida.
- c. La generación del laberinto debe ser dinámico y no estático.
- d. Debe de ser posible guardar el laberinto generado en un archivo para poder cargarlo en cualquier otro momento. Utilice la forma de almacenamiento que desee.

2. Resolución del Laberinto:

- a. El programa debe incluir una función que resuelva el laberinto utilizando backtracking desde un punto de partida definido, encontrando un camino hasta la salida.

- b. Debe ser capaz de resolver el laberinto desde cualquier punto seleccionado por el usuario dentro del laberinto.
- c. En el proceso de solución del laberinto el programa debe mostrar de forma gráfica el camino que está tomando el algoritmo para encontrar la solución, indicando las celdas que visita.

3. Guardar y Cargar Soluciones:

- a. Debe incluir la opción de guardar la solución del laberinto en un archivo para su posterior uso.
- b. Al cargar una solución previamente guardada, el usuario podrá seleccionar un punto de partida diferente para ver cómo el algoritmo resuelve el laberinto desde esa nueva ubicación.

4. Documentación Interna del Código:

- a. Todo el código debe incluir comentarios claros que expliquen el funcionamiento de cada sección.
- b. La estructura del código debe seguir buenas prácticas de programación, con un enfoque modular y reutilizable

5. Visualización del Proceso de Solución:

- a. La solución debe visualizarse paso a paso, mostrando las celdas que el algoritmo visita y cómo va construyendo el camino hacia la salida.
- b. Las celdas por las que pasa el algoritmo deben ser marcadas con un color diferente, y la solución final debe resaltarse claramente.

6. Enfoque optimizado versus fuerza bruta:

- a. El programa debe implementar dos enfoques para resolver el laberinto:
 - i. **Fuerza bruta:** Probar todas las posibles rutas sin optimizaciones.
 - ii. **Optimización:** Utilizar un enfoque optimizado, como priorizar direcciones o memorizar caminos que no llevan a la salida (evitar bucles innecesarios).
- b. El programa debe permitir que se seleccione el enfoque que se quiere utilizar.

7. Opcionales (Puntos Extra):

- a. Los estudiantes que entreguen una bitácora detallada de su trabajo diario, con los avances realizados, los problemas encontrados y las soluciones

implementadas, recibirán puntos adicionales. Esta bitácora debe incluir las horas trabajadas cada día y el progreso alcanzado.

- b. Los estudiantes que entreguen la visualización del laberinto con animación y una figura de su gusto mientras se resuelve el mismo recibirán puntos adicionales.

Entrega:

La entrega del proyecto debe incluir:

- Código fuente completo.
- Documentación interna y externa.
- Comparación detallada entre la solución por fuerza bruta y la optimizada.
- Instrucciones para ejecutar el programa y cargar/guardar laberintos y soluciones.

Formato de Entrega:

El proyecto debe de entregarse por medio de la documentación externa con el nombre “Nombre1_Nombre2-Proyecto1”, que contenga las instrucciones para ejecutar el proyecto y el enlace del repositorio de Github para descargar y ejecutarlo.

Fecha de Entrega:

El proyecto debe entregarse el día 04 de octubre del 2024 con hora límite de las 11:55 p.m., cualquier proyecto entregado después de la hora establecida será penalizada con 2^n por ciento, donde n representa n la cantidad de días de retraso por la entrega.