

PRÁCTICA 1: ALGORITMO A*

- Lenguaje de programación utilizado: He decido programar el algoritmo en el lenguaje C++ ya que este me brinda la libertad, robustez, flexibilidad y potencia que consideraba necesaria para programar el algoritmo.
- Procedimiento de implementación:
 1. El programa ha sido creado subdividiéndolo en varios módulos que unidos constituirán la aplicación. Estos módulos son:
 - main.cpp: Contiene el punto de arranque de la aplicación, su labor se limita a pintar por pantalla el menú con el que interactuará el usuario para que este decida como quiere utilizar el algoritmo, pudiendo elegir entre crear el mismo una rejilla para resolver o cargarla desde un fichero de texto plano. También podrá decidir entre ejecutar el algoritmo por pasos o mostrar directamente la solución.
 - rejilla.h: Este módulo contiene la codificación del objeto rejilla, el cual podrá cargar una rejilla desde un fichero o diseñarla por teclado. Posee comprobaciones para garantizar que la rejilla esté bien formada y también proporciona los métodos accesores y modificadores necesarios para que el algoritmo pueda trabajar con esta rejilla. También implementa un operador de salida por pantalla para dibujar la rejilla.
 - nodo.h: En su interior está definido el objeto Nodo, el cual será usado por el algoritmo para formar la red de nodos sobre la que trabajamos para realizar las expansiones hacia otras casillas. Este objeto contiene los métodos accesores y modificadores para la consulta o modificación de sus atributos. Los atributos de un nodo son los siguientes:
 1. Coste total.
 2. Penalización.
 3. Un vector de nodos hijos.
 4. Un vector de nodos padres.
 5. Posición(x,y) → Que actúa como identificador unívoco del nodo.
 6. El coste desde el nodo salida hasta el actual.
 7. El coste estimado de ir desde el nodo actual al nodo meta.
 8. El coste de desplazarse desde el nodo padre hasta el nodo actual.
 - aEstrella.h: Es el módulo que contiene la implementación del algoritmo.

- comparadores.h: Contiene los objetos comparadores necesarios para que las estructuras de datos actúen (ordenen los datos) en base al comportamiento requerido por el algoritmo.
 - tipos.h: Contiene los tipos de datos que han sido definidos por mí como programador y que no se encuentran en el estándar de C++.
 - constantes.h: Contiene todas las constantes que van a ser utilizadas por el algoritmo.
- 2. Estructuras de datos que he considerado necesarias para la implementación del algoritmo:
 - Vectores.
 - Colas de prioridad.
- Parte básica realizada:
 1. Algoritmo A* básico para ir desde un nodo origen a un nodo destino.
- Ampliaciones realizadas:
 1. Puntos de paso: El programa puede trabajar con puntos de paso, de forma que cada vez que resuelva el trayecto entre dos puntos de paso mostrará el coste de ese trayecto. Cuando se ha alcanzado el último punto de paso es cuando se traza la ruta a la/s meta/s de la rejilla.
 2. Múltiples metas: Cuando el programa trabaja con múltiples metas tratará de alcanzar aquella que le es más conveniente para finalizar el recorrido.
 3. Casillas prohibidas: Son aquellas casillas que no se pueden utilizar como parte de un recorrido.
 4. Celdas con penalización: Estas casillas son las que suponen un coste adicional en caso de trazar un camino a través de ellas.
 5. Detección de rejilla irresoluble: Si el programa no puede alcanzar la meta o algún punto de paso, se mostrará el mensaje de “EL PROBLEMA ES INSATISFACTIBLE” para indicar que no se puede alcanzar algún objetivo. Sin embargo mientras el programa pueda realizar trayectos entre puntos intermedios mostrará el coste que le supone llegar hasta allí y ya cuando surja un objetivo inalcanzable será cuando se informe al usuario de ello.