

PRACTICA 2

En la Practica 2, he implementado el código hasta el nivel 3.

- Para el nivel 2 he implementado Dijkstra con una cola de prioridad.

En el **struct nodo** he incluido un valor **entero coste** y he sobrecargado el **operador <** para que se ordenen en orden descendiente dependiendo del coste.

Para calcular el coste del nodo, la **función coste** se ejecuta en cada hijo y va sumando el coste de la casilla actual al que ya se tenía, activando el bikini y las zapatillas y devolviendo el valor de la casilla en la que nos situamos.

Como en este nivel se incluye la posibilidad de coger bikini y zapatillas, he implementado un nuevo **ComparaEstados2** en los que se incluyen estos dos nuevos elementos del **struct estado**.

Para aumentar la eficiencia del algoritmo, he creado una **función estaenCerrados**, que ve si ya hemos visto el nodo en cuestión, y si no es así, lo introduce en Abiertos. Para este caso, hemos sobrecargado el operador **==** para ver si dos estados son iguales.

- Para el nivel 3 he implementado Dijkstra con una cola de prioridad pero con algunas modificaciones.

En este caso he introducido 3 booleanos en el **struct estado** para ir activándolos cuando pasamos por cada objetivo.

En este nivel he implementado un nuevo **ComparaEstados3** en los que se incluyen estos 3 nuevos elementos del **struct estado**.

La **función objetivos_cumplidos** es la encargada de ir viendo si la casilla en cuestión es un objetivo y las cambia a true.

Para aumentar la eficiencia del algoritmo, he creado una **función estaenCerrados2**, que ve si ya hemos visto el nodo en cuestión, y si no es así, lo introduce en Abiertos.

Al final del algoritmo, vemos si el siguiente nodo en mirar esta en Cerrados, en tal caso, se salta al siguiente nodo.