1.

La heurística implementada para conseguir GFS y A* es la distancia de Manhatan ya que solo nos movemos en horizontal y vertical y no en diagonal por lo que usamos esta heurística simple que nos permite calcular la distancia hasta el goal

2.

	Mínimo	Máximo	Media
BFS	42	606	251,6
Dijkstra	42	704	261,25
Greedy	17	89	55,15
A*	26	646	149,45

3.

En el ejercicio 2 hemos implementados 2 enemigos a partir de la clase Agent pero con la diferencia de que estos ya tienen unas rutas establecidas que siguen, lo que hacen estos es cambiar el peso que tiene el nodo donde se encuentra poniendo un peso exagerado de 20, luego obtenemos los vecinos de este nodo y le pones un peso menor siendo este de 15 y para acabar, repetimos buscando los nodos vecinos de los nodos vecinos del nodo del enemigo y le ponemos un peso de 10, este método junto al pathfinding usado en este caso que es dijkstra ya nos sirve para evitar a los enemigos y tomar turas evadiendose.

4.

Travelling salesman problem responde al problema de dar una lista de nodos/ciudades/localizaciones y las distancias que hay entre ellas, cuál es la ruta más corta posible que visita cada nodo exactamente una vez y finaliza en el nodo original.

El problema presenta N! rutas posibles aunque no nos interesa calcular el nodo principal quedando con la fórmula (N-1) pero como no importa la dirección en que se desplace el viajante, el número de rutas a examinar se reduce nuevamente en un factor 2 quedando (N-1)!/2.

El problema que reside en este método es la cantidad de recursos que utiliza ya que cada vez se añaden más nodos a explorar más recursos gastamos y más cálculos

tenemos que hacer, el tema es el gran aumento de cálculos que se encuentran al introducir poca cantidad de nuevos nodos dando valores como:

5 nodos = 12 rutas diferentes
10 nodos = 181440 rutas diferentes
30 nodos = 4*10^30 rutas diferentes

Lo que vemos es que a la mínima que metemos más nodos la cantidad de rutas se dispara, sumado a que en el mundo de videojuego encontraríamos una gran cantidad de nodos por los cuales movernos, sumado a que nos interesaría usar un algoritmo de busca de pathfinding mejor llegaría a tener un grandísimo costo el cual no sería posible de aguantarlo

El algoritmo que hemos aplicado para dada una cantidad (en nuestro caso) de 5 monedas recorra todas haciendo el camino más corta entre monedas es aplicando la heurística de Manhatan, en este caso antes de pasarle el goal al A* para dirigirse a este lo que hacemos es recorrer todo el vector que contiene las posiciones de las monedas y comparamos sus distancias desde el start, retornando la moneda más corta desde start, y este proceso lo repetimos cada vez que conseguimos una moneda, sumado a que el pathfinding utilizado es A* conseguimos encontrar el camino más corto a todas las monedas del mapa