Tommaso Chinello

Daniel Felipe Bustamante Pérez

Practica de laboratorio 1B

Búsqueda A*

- En búsqueda A* usamos una cola prioritaria (PrioityQueue) ya que el algoritmo de Búsqueda A* utiliza el costo acumulado y el coste del heurístico, organizado así que camino tiene prioridad, ya que la cola prioritaria nos permite guardar por prioridad esta es la estructura de datos ideal.

Buscando todas las esquinas (All the corners)

- En la búsqueda de todas las esquinas usamos una tupla, la cual contiene las esquinas del mapa, para guardar las esquinas ya visitadas se hace uso de una lista de booleanos, se usa una lista ya que esta se puede modificar sin problemas

Problema de las cuatro esquinas: Heurístico

- En la búsqueda de las 4 esquinas usamos una tupla, donde almacenamos la posición actual y una lista de las esquinas, como en el punto anterior, usamos la distancia de manhattan que esta implementada en útil.py. Para tener una heurística admisible, consistente y eficiente, hemos utilizado la distancia mínima entre la posición actual y las esquinas no visitadas.

Comiendo todos los puntos (Eating All The Dots)

- En comiendo todos los puntos usamos una tupla para almacenar la posición de Pacman y foodgrid la cual almacena o simboliza la presencia de comida, para el heurístico hacemos uso de la distancia de manhattan. En este caso, hemos utilizado la distancia máxima entre la posición actual y las esquinas no visitadas.

Búsqueda subóptima (Suboptimal Search)

- En búsqueda subóptima buscamos el punto de comida más cercano en cada paso, usamos la posición actual y la posición de la comida en el laberinto, usamos Uniform Cost Search para encontrar la ruta hacia el punto más cercano.