AA2: Pathfinding

Miquel Postigo Llabrés
Axel Castells Monllau

Grup A 3r Grau

Algorismes

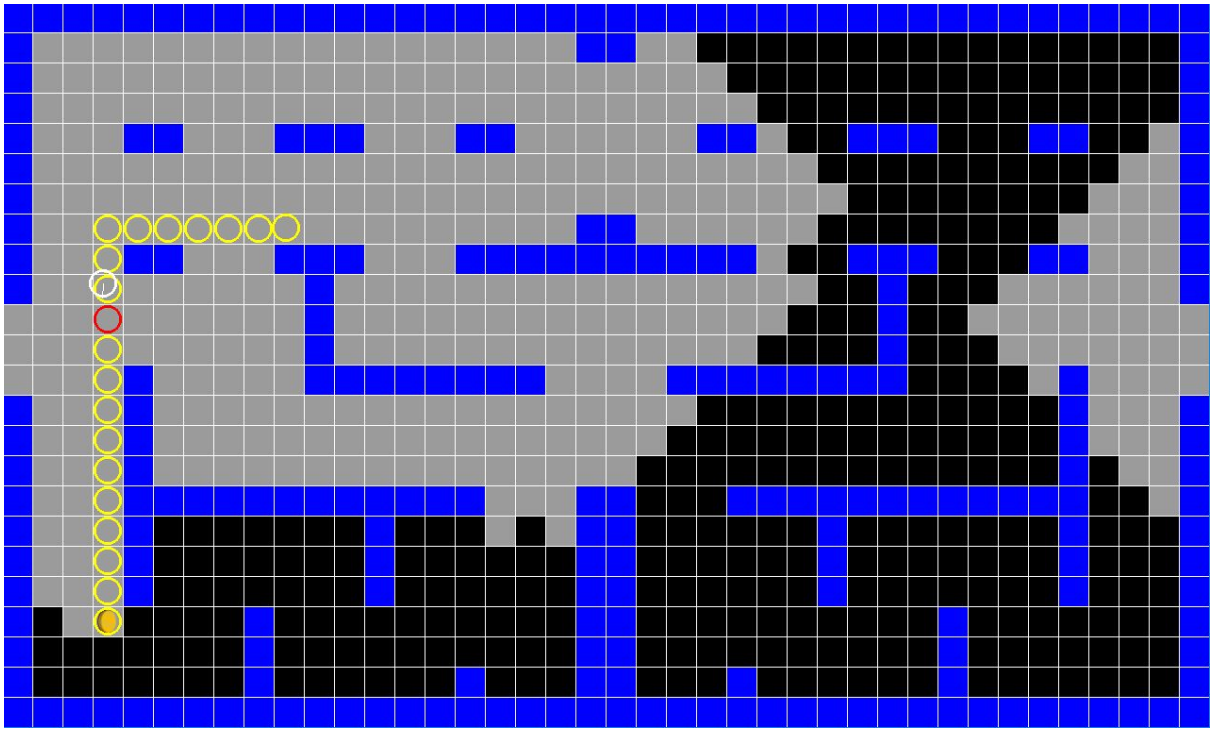
BFS

Mitjançant una cua (frontier), mantenim una referència al primer punt, es a dir, la posició actual de l'actor en el moment d'executar el BFS.
Per mantenir les referències dels nodes que ja hem visitat, utilitzem la següent estructura: ***map<pair<float, float>, Vector2D*>***.
D'aquesta manera, guardem com a clau del mapa la posició en format cel·la (on ***<float, float>*** es ***(node->x, node->y)***) del node actual dins l'iteració dels nodes.

En cada iteració del for, comprovem les condicions per executar un early exit, en cas de que el node actual sigui el node destí.

Un cop sortim del bucle for, recorrem a la inversa tots els punts visitats i els afegim a un vector de punts ***"pathPoints"***.

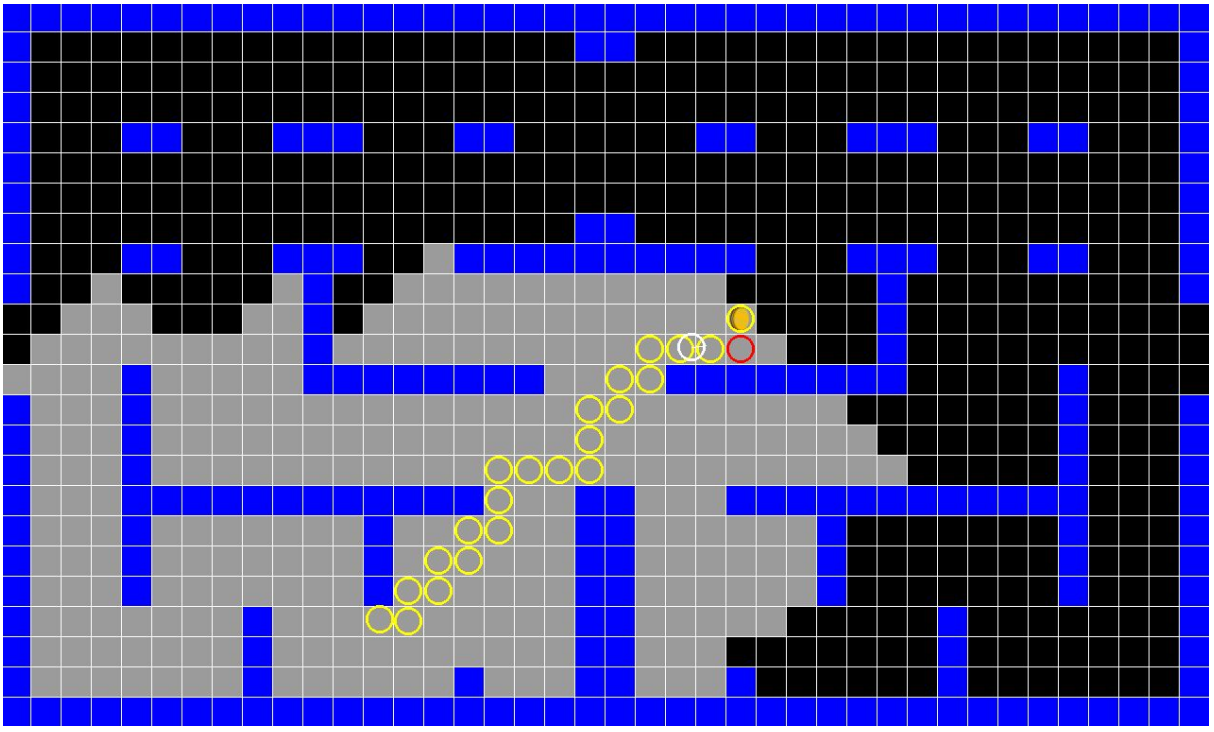
Finalment, recorrem ***pathPoints*** i afegim cada Vector2D* en ordre invertit a un nou vector de Vector2D* que serà el nostre path.



Dijkstra

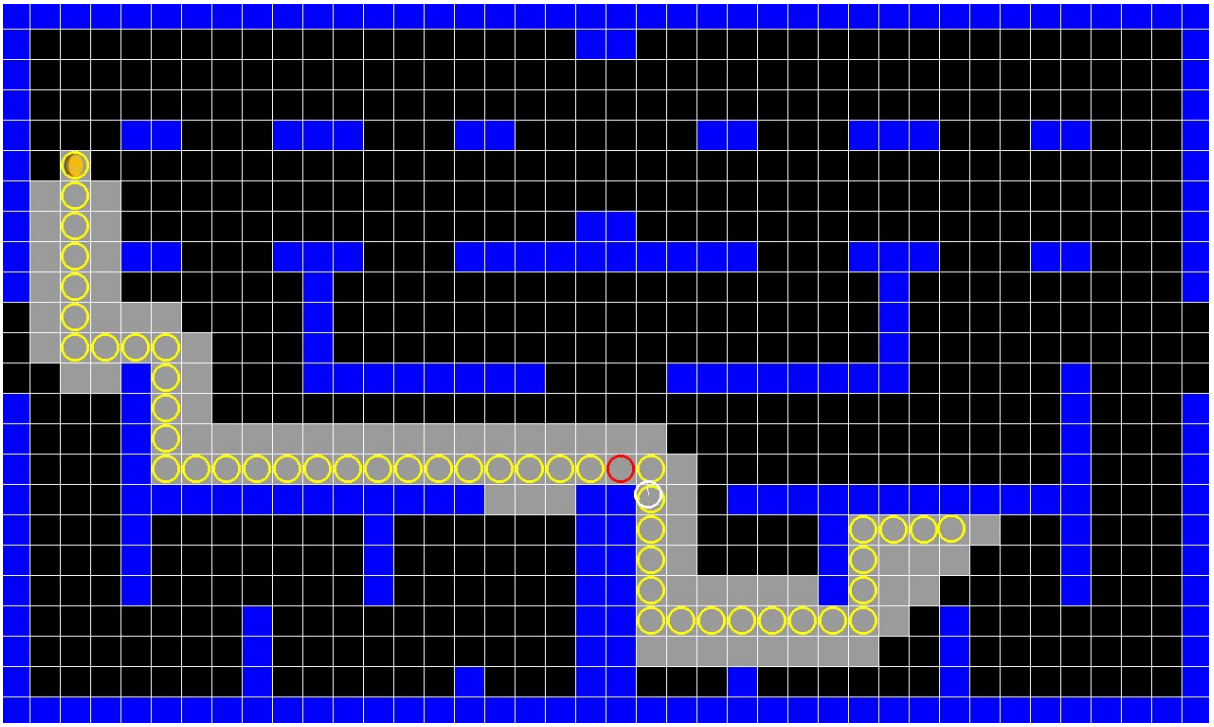
En el cas de Dijkstra, hem optat per utilitzar una cua de prioritat per facilitar-nos la feina.
Mitjançant ***"costSoFar"*** emmagatzemem tots els costs dels nodes a mesura que s'exploren dins del for.
En cada iteració, usem ***"newCost"*** per poder acumular el cost del node anterior amb l'actual, i afegir a **"*costSoFar"*** la posició del nou node amb el cost acumulat.
Un cop fet això, comprovem que el cost de cada node explorat a cada iteració sigui menor que el ***"newCost"*** abans d'afegir-lo al vector de nodes visitats.

Finalment, fem el mateix que amb BFS per reordenar i invertir el vector per a que serveixi a l'agent com a path.



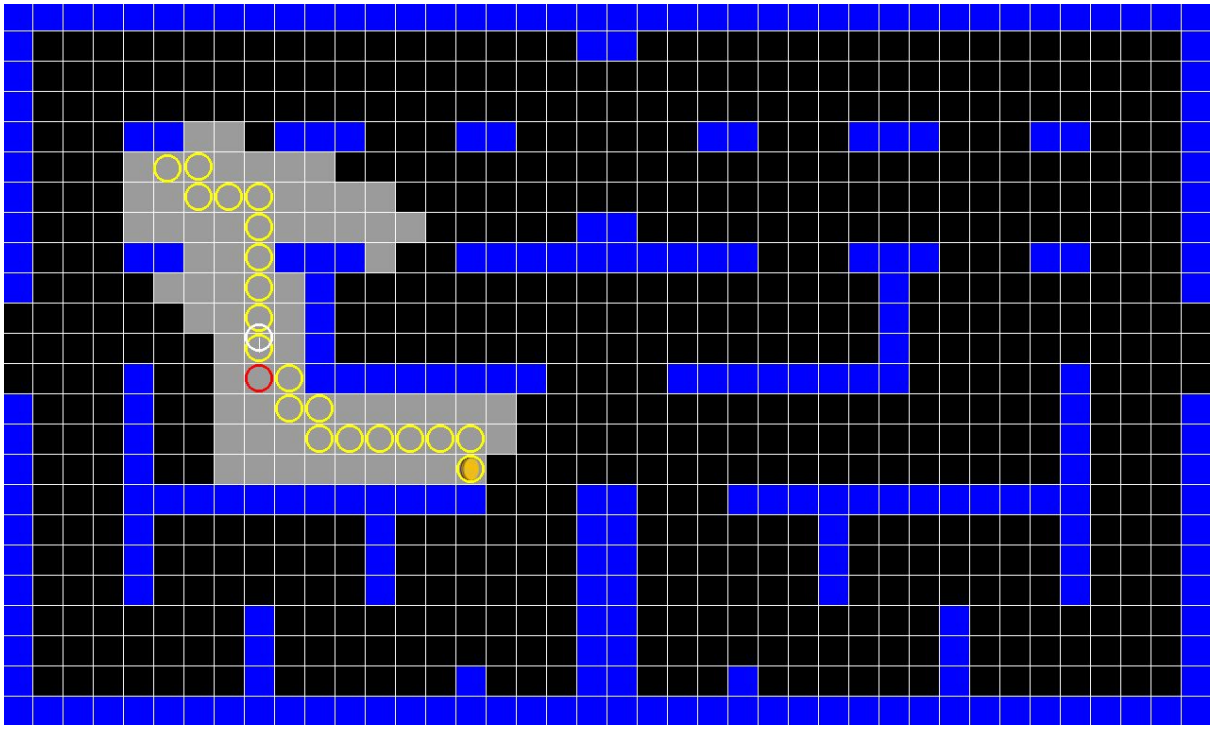
Greedy

Utilitzant una heurística Manhattan, calculem a cada iteració la distància entre el node que estem explorant ara i el node destí, i els afegim al "***frontier"***, que es una cua de prioritat. Gràcies a una sobrecàrrega de l'operador <, podem comparar ***pair<Vector2D*, float>***, on el ***float*** es el resultat de l'heurística i així, la ***"frontera"*** s'ordena internament.
D'aquesta manera, obtenim una cua amb tots els nodes ordenats de menor a major cost.



A*

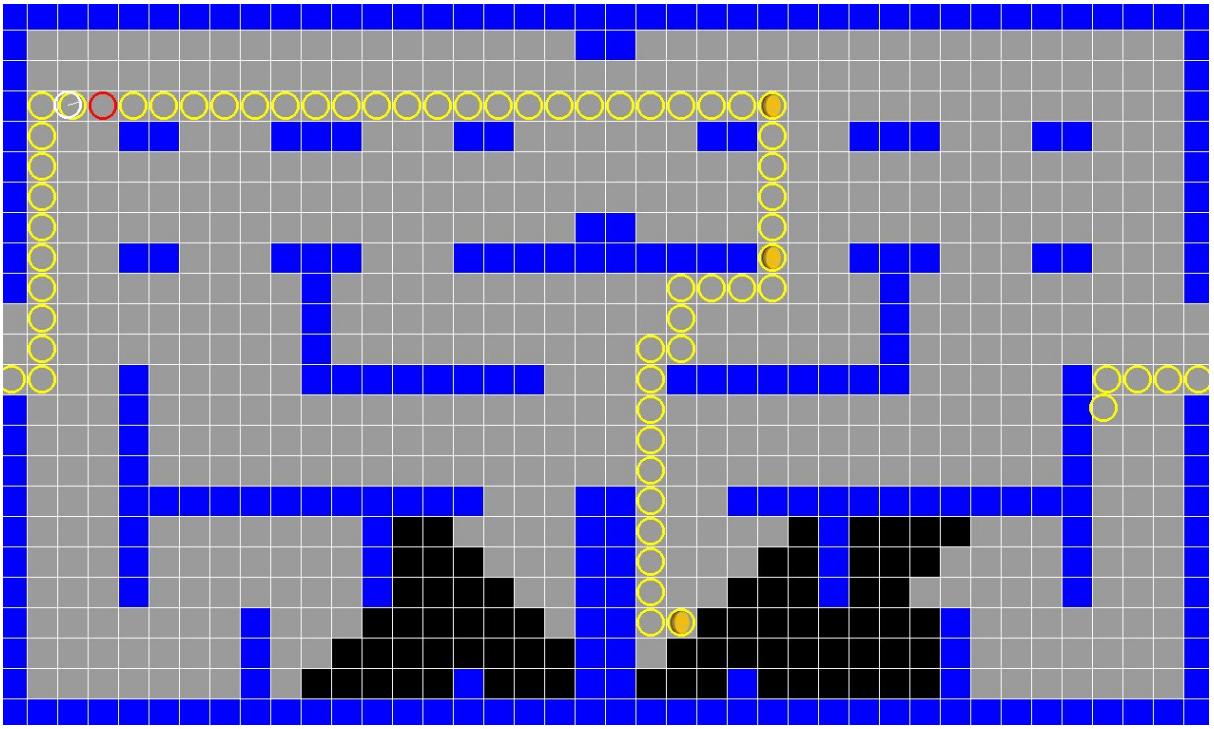
Utilitzant el mateix patró que Dijkstra i Greedy, calculem el ***"newCost"*** amb la heurística manhattan i el cost acumulat de ***"costSoFar"***, d'aquesta manera, omplim la ***"frontera"***, que també es una cua de prioritat, tenint en compte el cost acumulat dels nodes i la heurística.



A* Multitarget

Utilitzant com a paràmetre d'entrada un vector de Vector2D*, recorrem un for per reordenar mitjançant l'heurística manhattan i amb l'ajuda d'un map, el vector de Vector2D* que son les posicions per les que ha de passar el path.

Un cop tot ordenat, cridem N vegades A*, on N es el nombre de posicions que tenim al nostre vector.



Heurístiques

Manhattan

Hem utilitzat una heurística manhattan en aquells algorismes que hem necessitat determinar la distància entre dos nodes per establir prioritats.

L'hem calculat mitjançant la següent fórmula: ***(abs(a.x - b.x) + abs(a.y - b.y))***.

Calculant el valor absolut ens assegurem d'obtenir una distancia sense signe, que distorsionaria els resultats dels càlculs.

Costs dels nodes

Hem atribuit tres tipus diferents de cost per tal de veure el funcionament dels algorismes Dijkstra i A*. Aquests valors són de 2 per al color vermell més fosc, 5 per al vermell intermig i el vermell més clar representa un pes de 500.