**A\* Algorithm**

**Node**: anything in a graph that can be visited, for instance in a triangle we can consider the nodes as the three vertexes(points) of the triangle.

**Edges**: Connect nodes to each other, The edges in the triangle would be the lines of the triangle connecting each vertex (node).

**Weighted Edge**: Takes a specific amount of time or is a specific length which determines how long it would take to get to the node.

**Unweighted Edge**: All edges have the same value and non are longer or shorter than any other.

**Open Set:** Open Sets keep track of the next nodes we want to consider (it is a queue)

**F(n) = G(n) + H(n)**

**F(n):** addition of the two below, estimate of how long it may take to get to the end node and decides which node to look at first. (this is how we prioritise)

**G(n):** Current shortest distance to get from the start node to the current node.

**H(n):** A Function to give us an estimate of the distance from node (n) to the end node

**How does it work**

**The Goal**: Find the shortest path from point A to point B In a graph using the edges.

Diagram

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nodes | F | G | H | Last |
| A | 0 | 0 | 0 |  |
| B | Infinity | Infinity | Infinity |  |
| C | Infinity | Infinity | Infinity |  |
| D | Infinity | Infinity | Infinity |  |
|  |  |  |  |  |

All scores (F, G, H) start at infinity as we haven’t compared or looked at them yet.

Open Set: (0, A)

When starting we always start by placing our first node into the open set together with the distance to that node (our F score). In this example it would be (0, A), it is 0 because it is our starting node.

Open Set:

We now can pull out (0, A) from the open set to start having a look at it. The first step when looking at a node is to look at who its neighbours are, in this example the only neighbour to A is C and there are only two ways (edges) to get to node C.

We start by picking either edge (no way to pick a specific edge this is random). We now compare the distance of the edge that was picked (in this case we started with edge 1) with the distance (G score) of node C to find which is shorter, and of course compared with infinity the distance of 1 is shorter so after that we can update the G score to 1 for node C.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nodes | F | G | H | Last |
| A | 0 | 0 | 0 |  |
| B | Infinity | Infinity | Infinity |  |
| C | 2 | 1 | 1 | A |
| D | Infinity | Infinity | Infinity |  |
|  | Infinity | Infinity | Infinity |  |

Now that we found a route to node C we can take a guess on how long it will take to get to the end (node D) from C. This is a total guess so we will put it as a 1. This now means that the F score will equal 2.

Open Set: (2, C)

Now we add Node C into the open set with its F score of 2, now we look at the open set to pick out the next node we will look at with the shortest F score (this example there is only one node in the set so it will automatically be that one)

Open Set:

So, from node C we can look at Node B. The current distance from node C to node B would be from A to C (1), then C to B (3) which means a total of (4), so our current G score would be 4. So, we do the same as last time and compare the current G score of node B with our distance that was found (4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nodes | F | G | H | Last |
| A | 0 | 0 | 0 |  |
| B | 6 | 4 | 2 | C |
| C | 2 | 1 | 1 | A |
| D | 3 | 3 | 0 | C |
|  | Infinity | Infinity | Infinity |  |

In this example since B is infinity the current distance of 4 is shorter so the table is updated. Now we can complete the H score and guess how long it may take (2). Now we have looked at node B we can also add it into the open set with its current F score.

Open Set: (6, B)

So now the next node we can travel from C would be D (the end node). First, we have to calculate the G score which would be 3 in this example. And update the table, now we update the H score guessing distance from Node D to Node D which would obviously be 0. And F score.

Now add it to the Open Set

Open Set: (6, B), (3, D)

From the open set we pick the node with the shortest F score (in this example it would be (3, D)) and as soon as we take the end node out of the open set, we can immediately finish the algorithm.

Which means the shortest path is (A, C, D)