

## A\*

The A\* (A-Star) search algorithm is a path-finding algorithm with many uses, including Artificial Intelligence for games. The search builds the route from tile to tile until it reaches the goal.

To help choose the best possible tile the algorithm will use an equation that takes into account the distance from the tile to the goal and the cost of moving to that tile.

Let's work through an example, so you can see how this comes together.

Movement Cost for Terrain:

Non-walkable:

N/A =

Water (~)

Walkable:

1 = Flatlands (. or @ or X)

2 = Forest (\*)

3 = Mountain (^)

Test Map:

@\*^^^

~~\*~.

\*\*...

^..\*~

~~\*~X

@ = User start

X = The goal tile

Step 1: Search the surrounding tiles for walkable choices.

The only possible tile for the first step is to the right: a forest (\*) tile.

Step 2: Go through that list finding the cost of movement for each of those choice tiles. The cost of movement is the path cost so far, plus the cost to move to the tile being considered.

Our cost so far is 0, since we just started. The cost of a forest is 2 so the total cost of this move is 2 (0 + 2).

Step 3: Determine the distance from the choice tiles to the goal and add that to the cost from Step 2 to find the score for each tile.

You can calculate the distance from the tile to the goal using Manhattan distance formula  $|x1 - x2| + |y1 - y2|$ .

For our example that is:

$$|1 - 4| + |0 - 4| = |-3| + |-4| = 7$$

Knowing that we can produce the final score for this move:

$$\text{score} = \text{cost} (2) + \text{distance to goal} (7) = 9$$

Step 4: Choose the best tile to move to by comparing the score of the surrounding tiles and choosing the lowest.

Step 5: Repeat Steps 1-4 until you reach the goal tile. Be careful to prevent checking tiles twice and/or circling back.

Test Map Solution:

```
##^^^    # = Best path
```

```
~~#~.
```

```
* * .#.
```

```
^ . .#~
```

```
~~*~#
```

When you have a working solution, try it out on this move involved map:

large\_map.txt

Please also supply your large\_map.txt with the "#" path your logic walked as part of the solution.

---