An A\* Drama

Strange are the confluence of events that lead to the creation of things. This month’s blog is a weird amalgam of a computer science algorithm, some PowerPoint diagrams, a dash of video editing, and finally a pinch of royalty free music. And out popped Blog Wyrm’s first video.

I suppose this blog post started first with some analysis I had decided to do on the A\* algorithm and then it just grew from there. I had employed the A\* algorithm on a scheduling problem some years back. This application was advocated by one of the developers on my team but I didn’t have the time to properly learn it during development and deployment. I ended up using it as a black box with a silent vow to get around to learning it one of these days.

Years passed, and my son was learning it as part of a course he was taking and I thought now was the time. So, I purchased *AI for Game Developers*, by David M. Bourg and Glenn Seemann and dug into Chapter 7 entitled *A\* Pathfinding*. If you happen to read their book, you’ll find that the approach is highly influenced by and patterned after their material. To my credit, I slogged through a complete example and then animated in a nice little video. If you are interested in that, it can be found at the bottom of this post.

The premise of the A\* algorithm is that the starting point and ending point for some sort of journey are known but that the specific path from A-to-B, as it were, is not. The personal picture I like to entertain is embarking on a car trip from New York to Los Angeles. I know where I am starting and I know where I am to finish, but should I drive through Pennsylvania or Maryland or further south? Does Kentucky figure into my route? What about Idaho? And so on. The hope is to find the shortest path that connects the start with the end with only a general notion of how far apart the two are.

For simplicity, let’s consider a much simpler problem of a snake trying to find the man who poked him while he dozed on a warm rock in the summer sun.

Our slithering protagonist abstracts the terrain and lays down a grid, with himself starting at the grid point (4,2). He sees that the man is at the grid point (11,4) so he begins to formulate his revenge. However, being impatient to return to his sun-bathed nap our hero wants to dispatch the man as soon as possible. He needs the shortest path. Being mathematically minded and knowledgeable about algorithms, he decides on using the A\* algorithm.

The A\* algorithm consists of the following steps:

1. Assume that the starting spot is open
2. Take a look around at the nearest neighbor spots
3. If they can be occupied, open them by
   1. Putting them on the open list for subsequent visit
   2. Score them with a perceived cost to move onto them
   3. Score them with an estimated cost representing how far they are from the destination square
   4. Total the two scores to give a total cost for the spot
   5. Assign a pointer from the open spot to the parent spot
4. If they can’t be occupied (e.g. the are boundaries or forbidden) ignore them
5. Close the starting spot
6. Pick the lowest cost open spot on the list (picking arbitrarily if there are ties)
7. Open and score any unopened spot
8. Close the spot and repeat

If one of the open spots is the destination spot, don’t bother scoring it (i.e. give it a score of 0) and back-track from it to the start, following all the pointers. The resulting path can then be traversed in the forward fashion.

For the simple example examined here, the perceived cost will be taken to be 1 for all possible spots, meaning the speed with which any spot can be crossed is the same. In general, spots own a terrain cost that may make it more or less expensive than neighboring spots (the difference between a highway and a surface street each with dramatically different speed limits).

The estimated cost is evaluated using a heuristic that counts the spots between the open spot and the final spot with no regard to barriers or terrain.

Returning to our intrepid reptile, his spot at (4,2) is now shaded blue, meaning it is in the process of being closed. He scores the spots at (3,2), (3,1), (4,1) and (5,1) and finds that the total costs are 9, 9, 8, and 7 respectively (the heuristic cost for (4,1) is 8 because it takes eight moves, horizontally, vertically, or diagonally to get to (11,4)).

Of the four spaces he has now opened, he selects (5,1) to next consider as its cost of 7 is the lowest of the bunch. Spot (4,2) is now closed (shaded gray) and examining spot (5,1) then opens spots at (6,1) and (6,2).

Each of these has a score of 6 so he arbitrarily picks (6,1) to examine, opening the additional spots at (7,1) and (7,2). Again, he picks arbitrarily between these two spots, since both own a score of 5 and are the cheapest members of the open list.

His full adventure can be found in the video below, including a frame-by-frame look at just how he tries to put his revenge in motion. Spoiler: the man gets what’s coming to him in the end.