Path Finding Algorithm for Martian Cities

BY AKARSH KUMAR

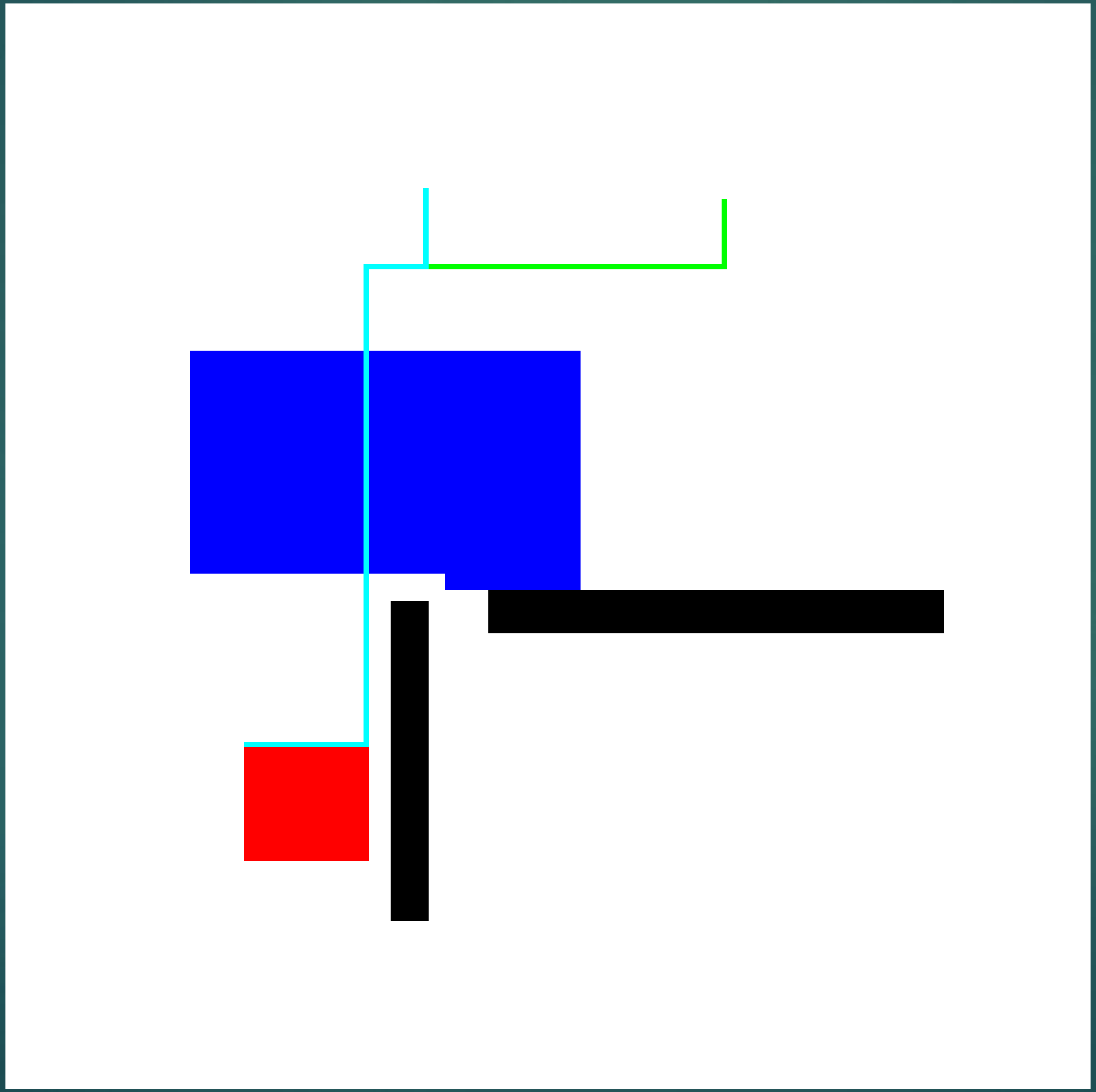
Path Finding

- Path finding is the act finding the shortest path from one point to another, given a graph of nodes to traverse through.
- Dijkstra's algorithm is used to find the shortest path from one point to another.
- A* is an optimization to Dijkstra's that is helpful if there is one goal.
- Both path finding algorithms start from the starting point and expand in a circle/oval shape outward calculating the shortest paths to each node it traverses.

Warm Up Problem

- The shortest path was found by using the A* algorithm.
- This was ideal because there was one ending point only.
- Heuristic was made by calculating the Manhattan distance to the closest end location.
- Rules: Green is start, red is goal, white pixels cost 1 to move on, blue pixels cost 2 to move on, black pixels cannot be moved on.
- The solution is the shortest path highlighted in cyan (next slide).

Warmup Shortest Path Solution



Shortest Path to Martian Cities

- To find the shortest path from my Martian city to others, I used Dijkstra's algorithm because A* is not helpful for multiple goal cities.
- My implementation of the frontier used PriorityQueue in order to get the lowest cost Coordinate on frontier fast.
- The

The Expanding Frontier