Rush Hour

Using artificial intelligence to
solve rush hour puzzles

Leonardo Flórido - 103360

IA
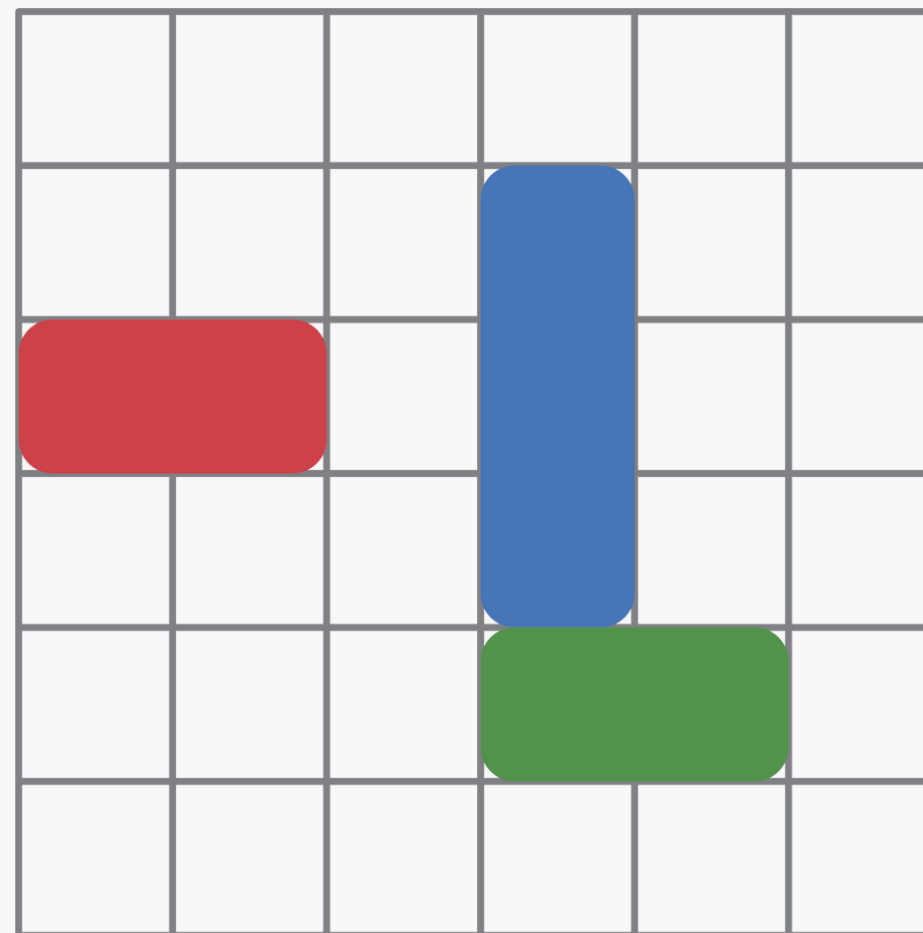
Approach

A* Algorithm

The A* algorithm was adapted from the version found on Wikipedia. It operates by considering both the cost and heuristic values, determining which nodes to compute in order to generate the final path. The resulting nodes will be reconstructed into the desired path.
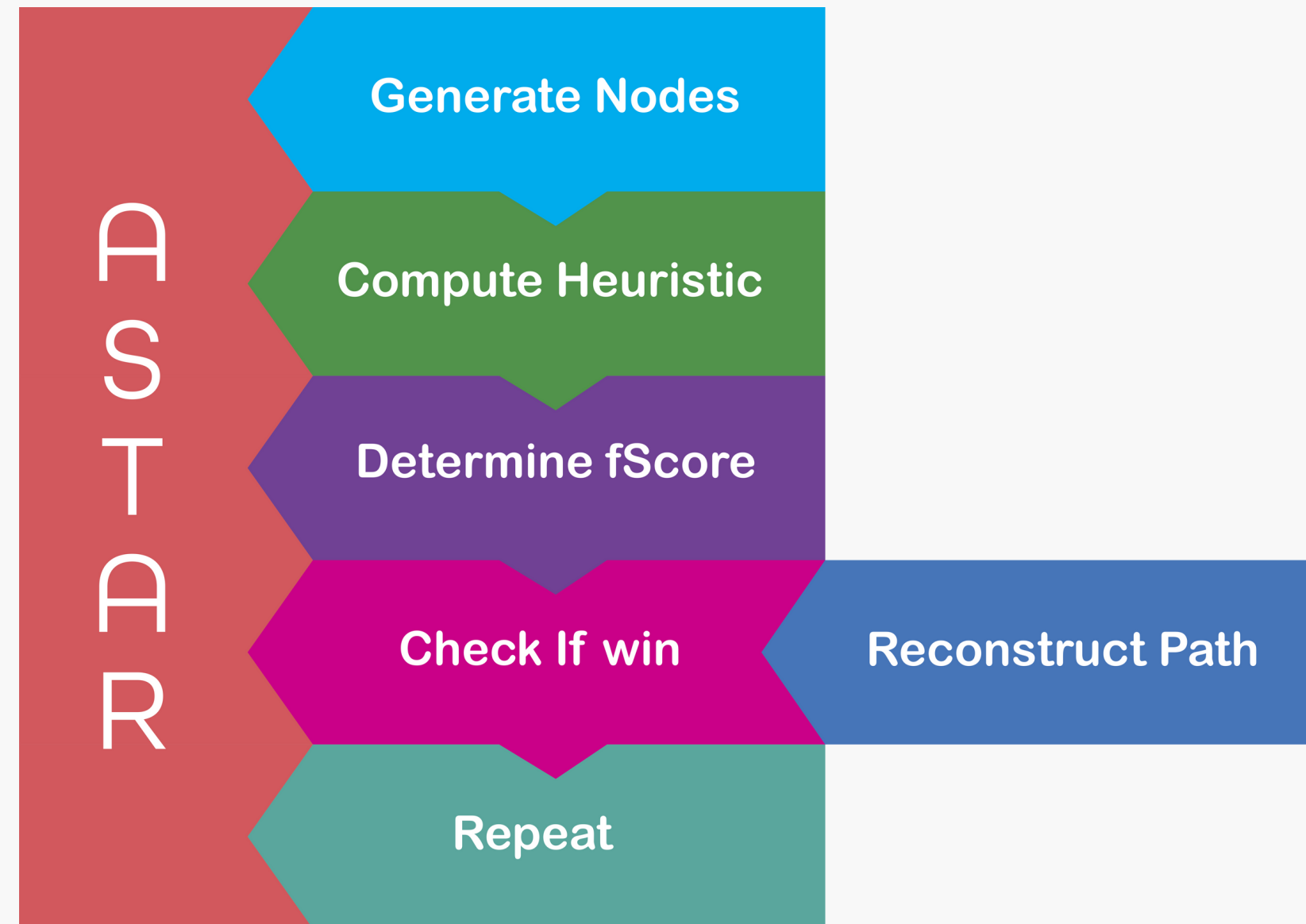
Crazy Car

Upon determination of the path, the information will be transmitted to our agent. In the event that the crazy car alters our state list, a new solution will be computed. If no such interference occurs, it will be disregarded.

Heuristic

The heuristic calculation involves determining the sum of the number of cars that are in front of car A and the number of blocks or squares from car A to the farthest right column.

IA

Rush Hour

The Process

A S T A R

Generate Nodes

Compute Heuristic

Determine fScore

**Check If win**   **Reconstruct Path**

Repeat

Summary

At each iteration, nodes will be generated starting from the node with the lowest calculated fscore. If one of the newly generated nodes corresponds to the winning state, the generation of nodes will be discontinued and the path will be reconstructed. If this is not the case, the process will be repeated.
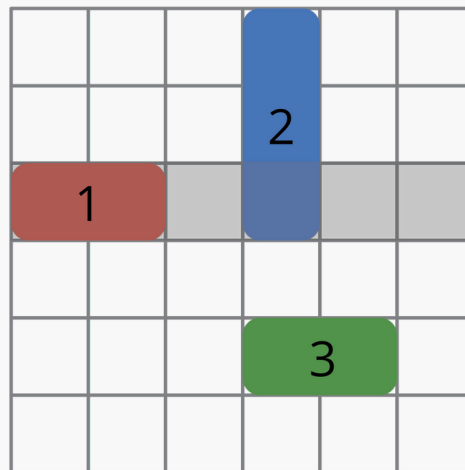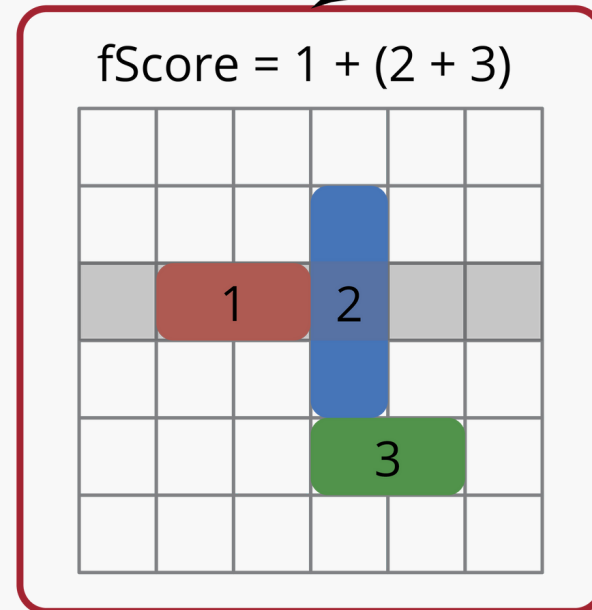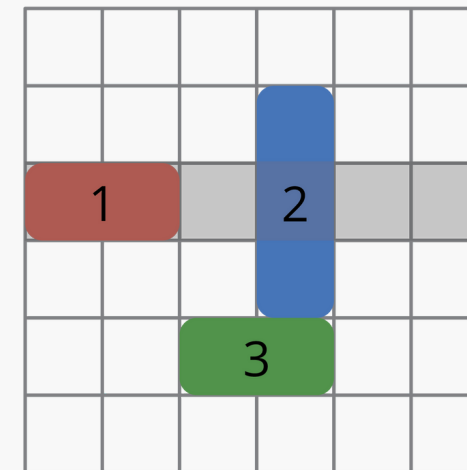
IA

gscore 0
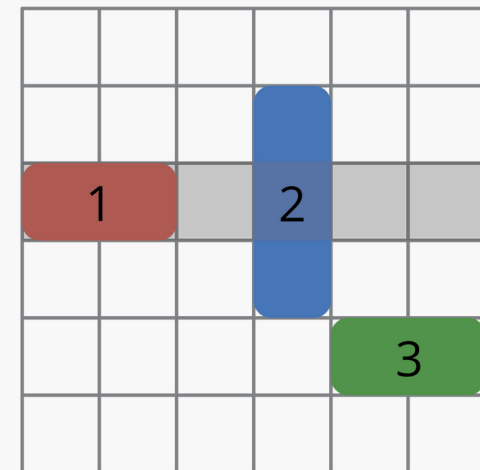
Initial state
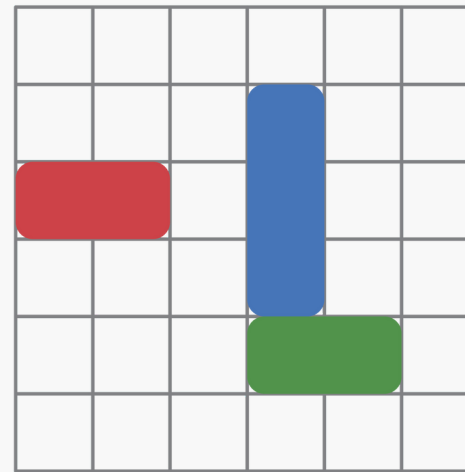
gscore 1

fScore = 1 + (2 + 4)

fScore = 1 + (2 + 3)

fScore = 1 + (2 + 4)

fScore = 1 + (2 + 4)

Test Win

If not win, open node and repeat process

fscore = gscore + heuristic

Always chooses the min fscore

IA

Benchmark

Opinion

An issue with the previous code, wherein in instances where the A* algorithm is unable to find a solution, the program would encounter an error. This occurrence was observed on a few occasions. To address this, if the A* algorithm fails to find a solution, the program will retrieve status information from the server and continuously recalculate until a solution is found. Once found, the program will commence movement of the pieces.

Efforts were made to develop an efficient algorithm. The initial approach involved using the class Map provided in the source code was deemed insufficiently efficient. A new code was created, resulting in a four-fold increase in the speed of computing the path to the solution. The solution is considered to be suitable for the problem.

References

https://en.wikipedia.org/wiki/A*_search_algorithm

IA