Rush Hour AI

Inteligência Artificial

Filipe Maia Antão - 103470

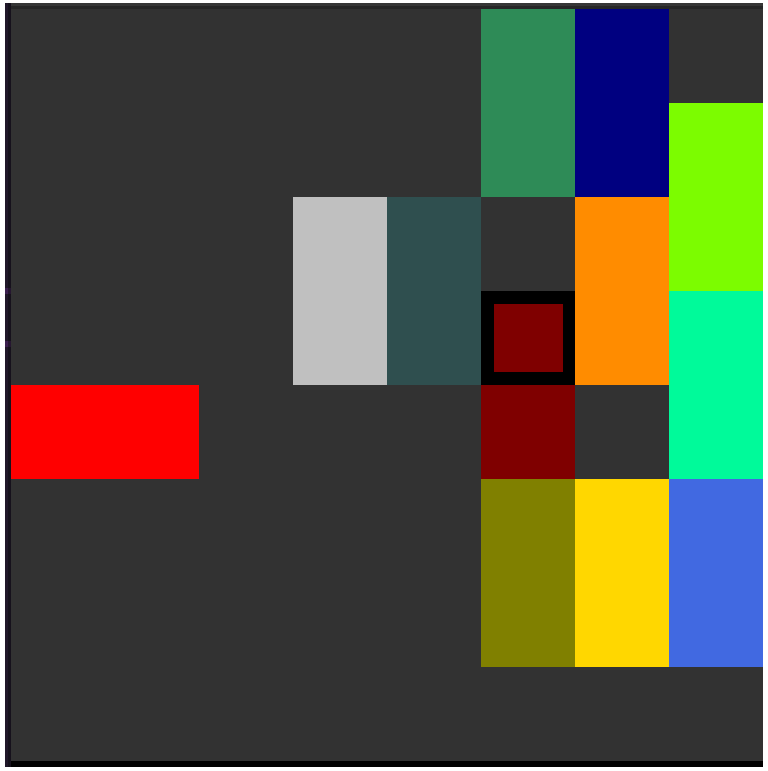
João Miguel Almeida Matos - 103341

Files

- *classes.py*: contains all base classes used to represent the information related to the game itself, like *Car*, and *Map2* (an extension of the Map found in *common.py*);
- *problem.py*: contains all classes used in the search for the level solution;
- *rush_hour_ai.py*: contains the class Bot, which, when provided with the game state, computes the corresponding solution and returns the inputs needed to achieve it;
- *student.py*: creates the Bot and calls its methods, sending the results to the server.

Search Algorithm

- In our first delivery, the search algorithm that we implemented was a BFS (Breadth-First-Algorithm).
- In the second one, we have a greedy algorithm, implementing a single heuristic. It assigns a value to each state/map, based on the number of cars blocking the exit and the minimum number of cars blocking the previous ones.
- In the presented example, the heuristic value is 3, because there are two cars blocking the exit, and one of them needs another car to move to be able to move itself.



Search Algorithm

- Regarding the Crazy Car mechanism, we check every grid to ensure no other car moved other than the one we did (when we did at all).
- If another car happened to move, it was by result of a crazy car – in this case, we compute a new solution using the new grid, since the last solution on computed based on an outdated grid, and therefore is outdated itself.

- We also ended up converting one of classes, namely *RushHourNode*, to tuples, since they make the bulk of our objects and tuples provide faster computing times.

Results

- The results obtained in the first delivery were very different than those from the second delivery, especially when increasing the complexity of the levels(8x8 grids)
- Overall, we get very good when it comes to the search for the solution in most levels (averaging values in 10^-1s and 10^0s). However, there's one level (level 25) that takes a lot longer to search for the solution(around 10^2s). For this reason, our agent can complete every level except for level 25. We suspect that in levels like this one, where there are not a lot of cars blocking the exit, but many cars blocking these few, our heuristic might not be the best.
- However, times were much better using our current algorithm – something very noticeable in bigger grids – so we decided to stick to it.

Final Notes

- We discussed this work with the following colleagues:
  - Raquel Paradinha (102491) and Tiago Carvalho (104142)
    - Crazy Car handling, Data Structures
  - Paulo Pinto (103234) and João Monteiro (102690)
    - Crazy Car handling, Algorithm Strategy

- We also used the following material as sources:
  - https://abbashommadi.github.io/AI-for-Rush-Hour-Game/
  - https://www.cs.huji.ac.il/w~ai/projects/2015/RushHour/files/report.pdf