Alex Liang, 301406554
John Mitton, 301415855

RushHour Solver Documentation | CMPT 225 | Igor Shinkar

Decisions

The rushhour class from Assignment 1 served a good base for this project already containing methods and ideas we would need to succeed. We decided to first try a BFS traversal of the different states (shown in Solver.bfsSearch) but eventually decided to use an A* algorithm instead because it is faster. We didn't end up using a heuristic in A* because the algorithm seemed to get solutions very quickly. Our A* uses PriorityQueue and HashMaps to determine the basic cost of each state from how many states stem from it, and finds the shortest path from start to solved. Each State saves its parent state so we have an easy path to follow.

Struggles

It was difficult to figure out how to make a faster algorithm such as A*. Once we had a faster algorithm for finding the path the next step was figuring out how to save each move and output it to a file. To do this we added a data field in our rushhour class to store what move was made to get to the current state from the previous state of the board, this occurred during our getStates method.

Changes along the way

After first trying BFS we realized that the algorithm was too slow and we would need to find something faster. Wanting to try A* we looked for inspiration and direction on google where we found a github rushhour project by tarek-eltalawi who used A* (https://github.com/tarek-eltalawi/Rush-Hour/blob/master/src/RushHour.java) after learning how that code worked we implemented a similar version into our own project. His project documentation pdf included in his github link helped us understand the tasks and goals needed to complete this implementation.

Partner Contributions

______Both of us worked together very well and brought up ideas together. We discussed everything about the project and worked through it from start to finish as a team. We talked and explained the code that we both did for our assigned parts. We also managed our time well and gave each other deadlines to finish our parts. In addition, we also hopped onto discord calls, and messaged each other frequently for updates on the project. John worked through getting the states of the board and the file output for this project while Alex worked on the A* algorithm and the hashing. Overall, it was a fun and enjoyable experience to be able to work as a team for a project.