

Anirudh Mantha
Modeling Challenge 1

When working on the modeling challenge the first thing that I did was analyze the three steps that were required to actually model the cubes movement. The first thing that I implemented was a make board method that would make the board represented as a string and take the one spot of the cube and mark it. Then to represent a string there were many ways to approach it such as a list, or a dictionary, rather I decided to use a list of tuples. Each list of three had a tuple of 2 components which modeled front-back, right-left, up-down. These tuples had a value of either 0 or 1, which determined its value of being either blue or white. The step that we were able to make was to integrate the cube and the board. After testing on a rubik's cube, drawing on a whiteboard, and collaborating with one another we were able to successfully make the cube move up left down and right. In order to get the possible children for this game, we did it very similarly to the sliding puzzles to get children. After checking the possibility of going up left down right, then we figured out that moving the cube in one direction does not affect 1 of the 3 pairs. For example, if we move up, then the right and left parts of the cube do not change values. However the up facing side will face backwards, and the down facing side will face forwards. This means that the tuples swap with one another. This swap allows the modeling to work successfully. After working with the "up" scenario, we moved onto implementing the same process for the down left and right cases. This took up a lot of class time because of how hard it was to visualize the different cases and imagine the cube in terms of a tuple. By the time we finished the modeling and were successful, class was over which meant that we had to work on it at home in an environment where my partner and I could not visualize properly. We decided that the best approach to this problem would be an A* algorithm that is slightly modified. Our approach I am confident was right, but we were only able to write the pseudo code. As there is an upcoming AP Physics test this monday, I was not able to invest enough time to solve the problem before the deadline therefore I am declaring failure. I was really interested in modeling it and was excited that I was successfully able to get the moves to work. In terms of the search, it is very similar to any other A* search. We just have to take into account the shifts in the orientation of the board (either 0 or 1) based on the color, but as long as we do that in other methods the A* should be straight forward. I will definitely try to solve this problem as soon as course work loosens up for other classes, but I was fascinated by this problem a lot.