For my program, I had a struct called Robot that held its coordinates, box coordinates,door goal coordinates, its id, and whether its thread was live or not. I gave them all these properties because this is the information each robot thread needed to execute. The coordinates allowed me to compute the path to goal, the id allowed me to identify the robot when it wrote to the external file, and the islive property allowed me to tell OpenGL to stop rendering the box/robot when they were done.

When the program is initialized, each title was given a mutex lock so that I would be able to synchronize the robots. I had a function that generated random, untaken coordinates for each door,robot, and box. It used a map to store already taken coordinates, and would check against that before returning coordinates. The function also accounts for the box not being able to touch the borders.

When it came to moving the robots and boxes to the door, I first calculated the path the box must take in order to get to the goal. With this information I could figure out which directions the robot would have to push the box from to reach the goal.I stored the directions in vectors that had direction/step pairs, and then had functions that interpreted these instructions. For example, the function could read the pair {NORTH, 3} and then it would move the robot 3 steps north. I had the boxes and robots reserve spots whenever they moved with a mutex so that the program would be synchronized.

One of the limitations of the program is that if to many robots/boxes are provided or the grid is too small, it does not error handle it. Another limitation is that it does not handle deadlock, so sometimes the robots/boxes are frozen in place unable to move.

One of the difficulties I ran into was when I was reserving tiles on the grid for my box and robot. My program would run fine until the robot pushed the box once, and the robot and box would freeze. I realized that I was moving the box by one square, and then trying to move my robot to the grid square was previously on. The problem was that I didn't unlock the previous box tile before the robot tried to lock it, causing a deadlock. I solved this by unlocking the previous box tile before locking it with the robot.