The project is about having Cozmo traverse through a maze. The difficulty of the maze differ in two levels – easy and hard. For the beginning, which aims for a C in the project will make an easy maze and Cozmo will be given specific instructions on how to go through the maze that was made by the user. For the B part, the user will make an algorithm that would produce a hard maze. For the A part, Cozmo tries to traverse through a hard maze, Cozmo will only be given a maze that is hard without any instructions. Cozmo will find its own path to the exit.

From the start, it was first time for me to ever own a robot. With this being said, I was only familiar to handling softwares, not hardwares. This made me look so much forward for the class and when the professor told us that we were required to buy a cozmo, I was excited. After getting cozmo, I thought about what to do for the project. I first thought that it would be nice for cozmo to guess what pose I was doing, so I could play games with cozmo. When I told this idea, professor said it would be machine learning project, not a cozmo project, so it was denied. Instead, she suggested me to make a maze, and have cozmo traverse it, and it sounded like a good idea, so I was glad to adopt the idea.

First, when I started on the project, I thought about how I should make a maze. Of course, I would make it with python list, but how would I traverse through multiple lists at the same time? This was a maze, so I needed to handle multiple lists, since it needs think about whether or not the an intersection is a way to the exit or not. I figured out that there is a way to traverse through lists if I made it into numpy array. At first, I tried to solve it manually by iterating through the list with loop, but it did not work efficiently for the reason that I had to store every space of the intersection and I had to return to that intersection once I made a decision. Another method I tried was to just give a specific instruction to cozmo to go through the maze. However, I realized this is not the right way to do it, because I felt like I was cheating. After, I searched online to find an effective way to solve the maze and stumbled upon an amazing github., which solves a maze with dijkstra's algorithm. It would take in a numpy array and calculate it with dijkstra's algorithm. After calculating the path, it would return a list. containing instructions on which way to go. When I receive the instructions, I instructed cozmo to go to that direction assuming that each instruction is 42mm far.

However, after solving the maze, I figured out that I actually had to construct a maze in real life to show that comzo was solving the maze. I first went to Blink to get a nice cardboard, but saw that the cardboard cost over $60, so I had to look other way. I got a box that was previously delivered from amazon to my house. It was almost a right size to demonstrate how I was solving the maze. I made walls out of other boxes and made sections like a real maze. When I actually had cozmo solve it, there were few problems. First, the maze was not exactly the same as the instruction meaning that the dijkstra's algorithm would solve the maze without considering the walls, but the wall had important role since the distance on how much cozmo should travel will vary depending on how many walls there are in one instruction. I had to make adjustments to each instruction while iterating through the instruction. After making adjustments to each iteration. somehow cozmo was not moving exactly as instructed to. It had something to with the surface that the cozmo was travelling through. It would constantly change its distance even if the distance was right before. I made adjustments in median so it would be almost as good to travel through

without bumping into the walls. After those small adjustments, I was finally able to make cozmo solve the maze with given instructions.

After, I tried to make cozmo traverse through the maze without any given instructions. I installed opencv, which will calculate a largest contour from the given input image given by cozmo. The real challenge here was deciding on what line is the perfect for cozmo to handle. I first tried to make cozmo follow a black line, but it did not work well because cozmo itself would make black shades, which cozmo thought it was a black line. Also, cozmo's sight was very small that the image was not wide enough for it to see and make decision or where to go. Also, the directions on which way cozmo want to go would vary depending on what it detects. I had to restrict certain parameters for cozmo, if the values were too small. Additionally, cozmo had to be exactly in the center to accurately follow the line. Overall, it was able to follow a line initially, but as it was going through the lines, based on the wrong decisions, it would divert to wrong ways.

For installing packages needed for running this project is by running command "pip install -r -requirements.txt." This will include all the packages needed and the command will automatically install all the requirements in one sequence. This project was tested on both MacOS and Windows. When starting the project, put Cozmo at the entrance, then go to the directory of this project. Disable comment the cozmo.run_program(solve_simple_maze) under drive_cozmo.py. When solving hard maze, place cozmo in cent of a white lane with black background and disable comment the cozmo.run_program(solve_simple_maze) under drive_cozmo.py.

The address of the Github is https://github.com/jaykangheo/maze_solve.git

This project would not have been possible without the help I got from Zudoxingdong (https://github.com/zuoxingdong/mazelab.git), who thankfully uploaded thoroughly on how to make different versions of a maze. Also, with help from Constantin (https://medium.com/@const.toporov/line-following-robot-with-opencv-and-contour-based-approach-417b90f2c298) was helpful when I was making Cozmo follow a line through the maze. Lastly, I am very thankful to Denise who inspired me to take on this project and develop it. I learned a lot, which would not have been possible if it were not her suggestion.