Main Track Documentation

Name: Anam, Tatiana

Approach #1:

Core Routine:

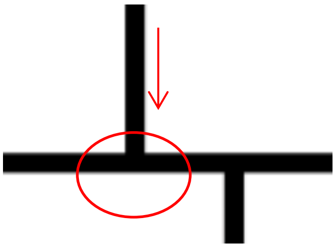
Our first approach was a dynamic approach that utilizes a combination of the eight sensors and timers. The basic line following algorithm includes going straight when sensors 4 and 5 detect black. If sensor 4 detects white and sensor 5 detects black or vice versa, this means that the robot is slightly off to the left (or right). We correct this by slightly decreasing the speed for one of the wheels. This is also the algorithm used on the curved sections of the track as well as the zig-zag portion.

We used a right-turn biased algorithm, so the robot prioritizes right turns and uses this to pass the squares in the beginning of the track. If the robot is on the track, sensor 4 and sensor 5 must always be detecting black. Hence, we use a nested if inside the sensors 4 and 5 condition to check our turning conditions. If sensor 3 detects black and sensor 6 detects white, the robot turns 90 degrees to the right.

In order to stop the robot after two laps, we use the condition where all sensors detect black except 1 and 8 as well as a lap counter. When the lap counter is 1, the robot moves a few centimeters to pass over the start line. When the lap counter is 2, the robot stops.

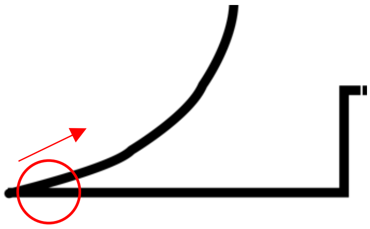
We have two special cases that arise which we handle in the following way:

Special case 1:



When the robot approaches this turn from above, sensors 6 and 3 both detect black. We use this condition to force the robot to turn right and avoid confusion.

Special case 2:

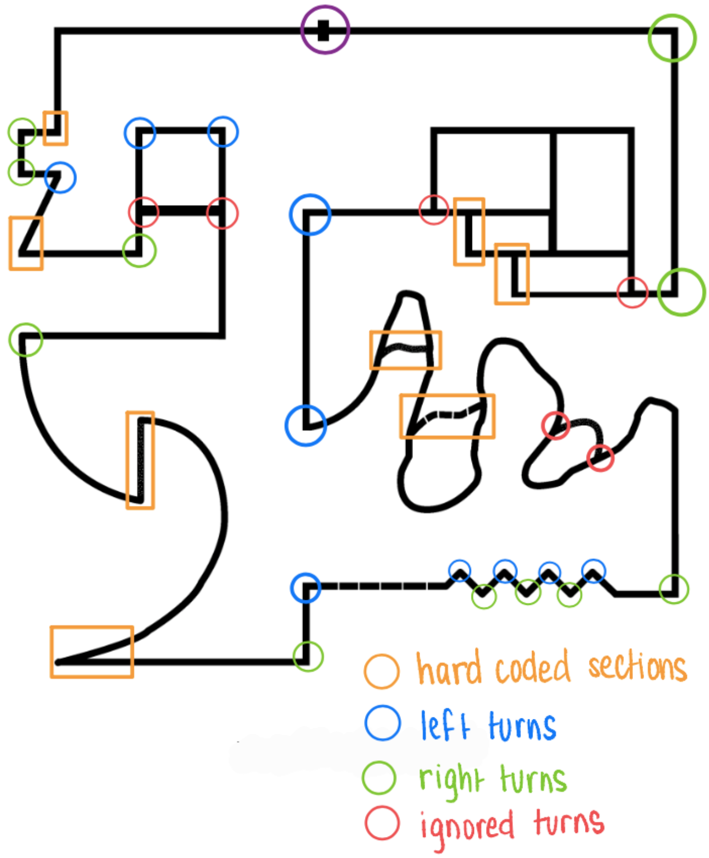


We use a special condition to pass this narrow turn. When sensors 1,2,4,5 detect black and sensors 3,6,7,8 detect white, we rotate the robot 150 degrees right and then make it continue a few centimeters in that direction.

Noise Handling: Our right-turn biased algorithm ignores left turns which include most of the noise portions of the track. The portions that cannot be avoided utilize the basic line following algorithm.

Approach #2:

Core Routine:



Our second approach utilizes left turn and right turn counters as well as hard coding in certain sections of the track. Whenever the robot detects the sensor condition for a left or a right turn, the corresponding turn counter is incremented. We use these counters to decide if a turn should be made, ignored or if hard code needs to be implemented. We also integrated the basic line following algorithm from approach #1. The picture on the left shows the turns that were implemented and ignored as well as the hard coded sections. To stop the robot after two laps we used the same sensor condition from approach #1 but we added code to reset all the turn counters after the first lap is over.

Noise Handling: Since our algorithm uses counters, we can choose which turns to ignore and which turns to hard code to overcome the noise sections.