Obstacle Management

Avoiding obstacles

In our code, pixy returns the center of the obstacle it's seeing, and Arduino compares the obstacle's center to the robot's center. This comparison tells us whether the obstacle is on the left side of the robot or on the right side of the robot. Four cases arise in this way:

- Object is on the left side of the robot:
    - **Red Obstacle**: Steer less (turn at small angle)
    - **Green Obstacle**: Steer More (turn at greater angle)
- Object is on the right side of the robot:
    - **Red Obstacle**: Steer More (turn at greater angle)
    - **Green Obstacle**: Steer less (turn at small angle)

Avoiding wall hit:

While avoiding obstacle robot sometimes steer a lot and end up hitting the wall as it is not in "stayStraight" subroutine. To avoid this, we have utilized left and right sensors. If distance of any of these sensors is less than 22 (as our sensors give minimum value of 19,20) we'll steer in the opposite direction. In this way we will avoid hitting the wall.

Taking corner turn:

For corner turn, we are only using front sensor and Pixy2 camera. We say it a corner if pixy is not detecting an obstacle and front sensor gives distance between 70 and 50 cm as these are the max and min distance values for the corner turn, respectively.

After detecting a corner turn, we compare distance of left and right sensors and steer on the side with greater distance.

Handling Stop and Turning from red obstacles:

We are using encoder motor which gives us no of pulses that the motor takes as it covers distance. By hit and trial we found that our motor takes 28,000 pulse to complete 7 sections of the arena. We have utilized this information to stop the robot as well as to return on seeing red obstacle in the 2nd lap.

Both logics are explained as follows:

Stopping after 3 laps:

- Arduino keeps on monitoring pulses from the motor.

- As soon as pulse count reaches 28,000, we know that we are in the end or middle of section 7 and need to cross the 8th section, we start monitoring front sensor distance.
- If the front sensor sees a distance greater than 150 cm and less than 180 cm, we know that we have turned 90 degrees because there is no other way, where front distance can be greater than 150 cm. In this way our robot will always stop in the middle of the first half of section 1.
- In case there is an obstacle in front of the robot, and it cannot see a distance between 150 and 180 cm, robot will automatically stop before 34,000 pulses. In this scenario, robot will stop in the middle of second half of the first section.

Taking U turn on seeing red obstacle in the 2nd lap:

To take U turn on red obstacle in the 7th section of second lap, we complete one lap by completing 28,000 pulses and then looking for front distance as described in the obstacle avoidance section of the document, then we reset the pulse count to zero and start counting for next lap. Next actions are bulleted down:

- Keep on moving until pulse count reaches 28,000.
- Stop the car for 1 sec.
- Make the car straight if it has stopped at an angle.
- Look for obstacle color.
- If it's green, break the loop and keep going.
- Otherwise, avoid the obstacle using the "obstacleTurn" subroutine.
- Keep the car moving for 0.5 seconds to ensure that it has crossed the obstacle.
- After that Steer leftwards at a larger angle and keep going for 0.8 seconds, in this time obstacle can turn 180 degrees.
- Reset the pulse counter, up count the lap counter and keep going until it completes 28,000 pulses.