**Assignment 3**

CS-XXXX.YYY

**Chart

Description automatically generatedScenario**

Figure 1

* **Goal**: Move the Ego vehicle through the parking lot and into the empty space.
* **Constraints:**
  + Use only the sensor data which is given through the RTOS.
  + Use only the actuators that are given by the RTOS.
  + Stay in the parking lot
  + Don’t hit the parked cars
  + Don’t hit pedestrians

**Sensors**

* A 5 pixel by 5 pixel color camera
* 16 lidar sensors placed in a circular ring around the Ego vehicle
* A compass which tells you how different your current orientation is from North.
* A speedometer

**Actuators**

* Acceleration Subsystem
* Brakes
* Steering Controller

**Problem 1: Parking Search (20 points)**

The Ego vehicle will need to search the parking lot for an empty parking space. The location of the empty space is randomized, and so is the starting position and orientation of the Ego vehicle.

For now, focus on creating the tasks necessary solely for avoiding parked cars while searching – you might find it helpful to simply ignore the pedestrian-avoidance constraint for now. You are allowed to reuse your old code from Project 2.

You are free to employ whatever search strategy you wish. A simple approach for solving these kinds of problems is known as the “wall follower strategy”, although some people call it “the right hand rule”. The idea behind the strategy is simple:

* Place your hand on the right wall of a maze
* Move forward along the wall of the maze, and never remove your right hand from the wall.
* Continue moving until you reach the goal position.

This strategy is described in more detail in this Wikipedia article, which also includes other strategies for solving mazes:

* <https://en.wikipedia.org/wiki/Maze-solving_algorithm>

**Problem 2: Obstacle Avoidance (35 points)**

There will be numerous pedestrians which are walking around the parking lot. Also, some vehicles will randomly reverse out of their parking space before driving forward back into them.

Pedestrians are randomly placed, and they will follow random paths. If a pedestrian or another car is hit by your Ego vehicle, they will likely sue both you and the company. You will be fired, and you might even potentially face jail if a judge decides that you engaged in gross negligence while developing your control tasks.

Thus, it is very important not to hit any of these obstacles. If a pedestrian is walking towards to your vehicle, they will turn away once they get too close. The other vehicles will only reverse out of their parking space before immediately moving back into it, and they will not drive around the parking lot. You may employ any avoidance strategy you wish, so long as you avoid hitting an obstacle.

**Problem 3: Parking Procedure (35 points)**

Once the empty parking space has been found, the Ego vehicle will begin to maneuver itself into the empty space. While doing so, the Ego vehicle was still avoid hitting pedestrians or the other vehicles on either side of the parking space.

You are free to use whatever parking strategy you wish.

**Problem 4: Documentation (10 points)**

You must document and describe your implementation in a final report. This documentation must include:

* Information about the tasks which you defined to control the Ego vehicle
  + Task name
  + Task description
  + Dependencies on other tasks and/or sensor data
  + Is it periodic, aperiodic, or sporadic?
  + Period
  + Deadline
* Functional and non-functional requirements of the system
* A graph showing the task schedule for at least two cycles of the simulation.
  + This can be generated by examining the debug options located in RTOS panel on the Ego vehicle object in the Unity scene.
  + Describe the graph. Point out which tasks are being executed, and explain at what point in the simulation the task is occurring at.

Please note that while bad documentation may lose you at most 10 points, a complete lack of documentation will result in an automatic -50 points penalty.