Program 1 Design

Alexander DuPree

October 5, 2018

1 Task and Purpose

Program number one serves as an introductory program for the construction, creation, and application of object oriented design patterns. As such, we were tasked with creating a object oriented program that will simulate the movement of a car along a one way street. This program will contain certain characteristics and support the following features:

1. Street
    - Speed limit
    - Number of lanes
    - Lane reservation i.e. bus lanes, bike lanes etc.
    - Vechicle list and position along the roadway
2. Vehicles
    - Current Velocity
    - Braking Distance
    - Location communication to nearby vehicles
3. Stop Lights and Crosswalks
    - Location information
    - Signal status i.e. green, red, orange.

2 Design Considerations

Object oriented programming requires a modular design with an emphasis on encapsulation, inheritence, and polymorphism. Accomplishing this design will require multiple classes each with a single responsibility as their focus. This will help encapsulate and protect data related to a specific task while keeping the program organized and scalable. This assignment aim's to apply those design principles and as such, I will be keeping each class as small as possible, and create only as much production code required to accomplish a specific task while avoiding the excessive use of "getters/setters".

3 Classes, Relationships, and Responsibilites

The following is a basic outline of each class to be created for this program and their responsibility and relationship to other classes.

3.1 Position

Responsibilities: The primary responsibility of the Position class is to manage location data in our vehicle simulation. Positional data will be stored as an ordered pair of integers (X, Y); with the origin (0,0) representing the location at the beginning of the street in the leftmost lane. Coordinates along the x direction represents each lane of travel along the street, and Y value coordinates represent the distance displaced from the beginning of the street.

Relationships: The Position class will act as the base class in the hierarchical structure to most other objects in our system such as the vehicle and street signal classes. Because every simulated object travelling down our street in the program requires positional information it seemed logical to design each class as a derived position object with specialized methods.

Functions: The Position base class will have public methods for reading the location data as the X, Y coordinates of the object will be used as the indices in the Traffic Matrix class. Furthermore, the base class will have a display method that will be overloaded in the derived classes so that each object can display to the console their specific symbolic representation.

3.2 Vehicle

Relationships: The vehicle class is derived from the position base class and in turn, acts as the base class for future derivations that may be added to this projecct like bikes, buses, and other vehicular objects that travel along the roadway.

Responsibilities: Vehicle objects maintain the responsibility of communicating their positional data however, they also manage their location with the added element of velocity and braking distance. Each vehicle will be able to speed up or slow down to a specific velocity, and calculate their maximum braking distance, all while updating their positional data.

Functions: The vehicle class will have a function to calculate braking distance as well as methods to modify its current velocity. The vehicle class will also specialize the base class display method to print out to the console its symbolic representation.

3.3 Street Signal

Relationships: The street signal class is derived from the position base class as it will have a fixed position on the roadway.

Responsibilities: The Street Signal primary responsibility is to communicate to all position objects attempting to enter the intersection whether or not they can pass. Because the Street Signal class is derived from the Position base class the Street Signal will be able to broadcast its location to incoming vehicle objects, allowing them to adjust velocity and brake properly.

Functions: The Street Signal class will have public methods for checking the signal status. i.e. green, red, orange. The Signal will change to green when a certain number of vehicles have reached its threshold and switch back to red as vehicles leave the intersection.

3.4 Traffic Manager

Relationships: The Traffic Manager is not part of a hierarchical structure but will possess containment relationships to adequetely manage the traffic within the system. The Traffic Manager will possess an array of linear linked lists to store the position objects in the system. Each index in the array will represent a lane on the street with each LLL containing all objects on that lane. The Traffic Manager will also contain a Traffic Matrix object. The Traffic Matrix roles and responsibilities are outlined in further detail below.

Responsibilities: The responsibility of the Traffic Manager is to instantiate and manage all position objects in the system. Each position object will be stored onto the array of LLL and a pointer to the object will be pushed into the correct position within the Traffic Matrix. After each

iteration of user input the Traffic Manager will update each objects position within the system and relay those updates to the composed traffic matrix object.

Functions: The Traffic Manager will have methods to instantiate certain objects at specific locations as well as an update method that will update the position data of each object based on the position objects calculations. Furthermore, the Traffic Manager will have a display method that will display to the user a graphical representation of the street and the vehicles within it.

3.5 Traffic Matrix

Relationships: The Traffic Matrix is a standalone class that will be composed into the Traffic Manager object. The Traffic Manager acts as the client of the Traffic Matrix, relying on the matrix for collision notification and random access for Vehicles located on the roadway.

Responsibilities: The Traffic Matrix primary responsibility is to handle the movement of vehicles along the roadway. Because the dimensions of the road are fixed and can only contain a certain number of vehicles the Traffic Matrix will utilize a dynamic 2D array of Position pointers to represent position and empty space along the roadway. The positional data of each Position object will be used as the indice for the 2D array. Indices with Null pointers will represent an empty space for other position objects to occupy and occupied indices will report a collision to the manager if an object is attempting to move there.

Functions: The Traffic Matrix will have methods to update the position of each object in the system as well signal lane changes to the manager class. Traffic Matrix will also have a display method that will assist in detailing the graphical representation of the street to the console.