COSC 3P91 – Assignment 1 – 5987722

ABHIJEET PRAJAPATI, Brock University, Canada

ACM Reference Format:
Abhijeet Prajapati. 2022. COSC 3P91 – Assignment 1 – 5987722. *ACM Comput. Surv.* 0, 0, Article 1 (February 2022), 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

This assignment is part of a much bigger assignment. In this paper, I'll talk about the choices I made when implementing the different systems. Each section will talk about a different package.

2 MAPS PACKAGE

The map is supposed to be represented as a graph data structure, but the basic graphs' edges don't typically hold data such as vehicles that exist on the edges solely, outside of the vertex. Graphs can have weights but those modifiers are still stored on the vertex itself, where as in this case, edges are required to hold data as well. So, I made my own graph structure, with vertexes being intersections and roads serving as the edges. Intersections control the flow of traffic with traffic lights in the real world and to accomplish that, I made a Traffic light class that contains an enumeration that just cycles through the 3 different colors a traffic light can be, with a method that will control the actual cycle of the colors in the correct order. There should be a total of 2 different traffic lights, there are 4 in the real world for the 4 different directions people see, but in our code, we can just use 2, with 1 being able to control 2 sides. When a vehicle reaches an intersection from a road, the user will be presented a choice between 0 to the total number of roads attached to that intersection (this allows u-turns) to decide which road they want to go down. The road will contain a lane class that will help it keep track of each vehicle on that road, with a specific number of lanes on each road. The roads will have a top intersection variable and a bottom intersection variable that will also serve as points like an edge. In the real world, traffic flows both ways with there being 2 different roads facilitating that, and just like that, there should be 2 roads connected to an intersection from the same area that have vehicles going one way only which will be set when the user makes the map and sets the top and bottom intersections of a road. This may also make it impossible for vehicles to ever go on the wrong side of the road. The Maps class will also hold a list of every Intersection, Road and Vehicle on the map to be easily referenceable.

3 VEHICLE PACKAGE

The vehicle package contains all the objects and components that will comprise a vehicle object. There will be an interface class in this package called vehicle that will be implemented by a Car, Bus, and Truck class.

Author's address: Abhijeet Prajapati, Brock University, 1812 Sir Isaac Brock Way, St. Catharines, ON, L2S 3A1, Canada.

2022. 0360-0300/2022/2-ART1 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3.1 Vehicle Class

This is the interface, it will contain the methods that pertain to increasing and decreasing the speed of a vehicle, changing lanes, getting information about its surroundings, controlling it's behaviour at an intersection and when it gets damage. All these methods will be implemented in any class that inherits this, so far it is the Bus, Car and Truck classes. Each vehicle that inherits this will have a variable that will hold a unique ID number that will help to easily identify that vehicle which will server additional functionality such as preventing a copy of the same vehicle from being added, and making it easier to find/identify a specific vehicle.

3.2 Bus Class

This class implements the Vehicle interface and also has 3 new methods that will add passengers from the User Package since buses can hold many more people than cars and trucks can. So these passengers will be held in either an Array with a pre-set capacity, just like real busses can only hold so many people. Buses are more durable but also more slower than most other vehicles, they have to be tougher since they are carrying more people than other vehicles and that added bulk makes their max speed much lower so the appropriate variables reflect that, but in the constructor, the options to change that will be presented in case a better bus was invented.

3.3 Car Class

The car class is another implementation of the Vehicle interface, it adds behaviour for the methods in a way that would be appropriate for a car. It's top speed and health are also different than the Bus.

3.4 Truck Class

Like the other 2 implementations of the vehicle class, it adds specific behaviour that lines up with what a truck is capable of. It also has an additional method that can attach a trailer to the truck.

4 USER PACKAGE

The user package contains a interface called User. The User interface (pun intended) contains some required methods that will be implemented by a Player class and an AI class. This will have the basic methods such as getting user information, getting and setting vehicle information, and also requesting information about it's surroundings.

4.1 User Class

This class is the one that will be used to add functionality for any player that is added to the game. Currently it only contains a methods that is used to get input from the player to decide what the vehicle will do.

4.2 AI Class

The AI class will be used to control the behaviour for any vehicle on the map that is not controlled by a player. It will contain a method that will retrieve the board state then an additional method that will decide on how to move it's own vehicle based on that board state.

5 GUI PACKAGE

The GUI package contains methods that will present the player with the game visually. It will contain methods that update the map shown after every move is made, along with drawing the different Intersections, Roads and Vehicles.

6 MAIN GAME LOGIC PACKAGE

This package will contain a class that serves as the main driver of the program. It will be responsible for creating new Intersections, Roads, Vehicles and adding new players.