

## CSCI262/862 Assignment 3

### Group Report

#### Group Members:

**Lindsay Plat,5553246**  
**Xinwei Zhang,4931233**  
**Yixiang Fan,5083898**

## Part 1: Initial Input

1. The way to store the vehicle types and statistics internally.

Read Vehicles.txt and store each vehicle's name in a string variable(sname) and other information in a struct (vehicleType). Then pair them together and insert to a map(vt). Vehicle's name and other information corresponding to the key and value of the map (vt).

Read Stats.txt and store each vehicle's name in a string variable(sname) and other information in a struct (vehicleStats). Then pair them together and insert to a map(vs). Vehicle's name and other information corresponding to the key and value of the map (vs).

2. Potential inconsistencies:

There are 4 types of Potential inconsistencies:

- a. The number of vehicle types being monitored is different from the actual number of vehicle types counted by program (vehiclesTypeCounter) in Vehicles.txt.
- b. The number of vehicle types being monitored is different from the actual number of vehicle types counted by program (vehiclesStateCounter) in Stats.txt.
- c. The number of vehicle types being monitored of Vehicles.txt is different from Stats.txt. (vehiclesTypeCounter and vehiclesStateCounter)
- d. The name of each vehicle in Stats.txt is not consistency with Vehicles.txt. (namelist)

## Part 2: Activity Engine

There are four nested for loops in total. The first for loop used to store each day's information in a multimap vday.

The second for loop used to store each vehicle type's information in vehicleSummary(vVS). The third for loop will generate one day's all the events of this vehicle type and store in vehicleDetail(vVD). Inside this for loop, there are two for loop side by side. The first one will generate vehicle's license randomly. After this, the program will generate events approximately consistent with below types.

There are five types of activity events, each event will be represented as an integer number and store in the flag variable.

1. Flag 0 represents vehicle arrival.
2. Flag 1 represents Vehicle departure via a side road.
3. Flag 2 represents Vehicle departure via the end of the road.
3. Flag 3 represents the vehicle parks on the road.
4. Flag 4 means the vehicle stops parking.
5. Flag 5 means the vehicle moves and possibly changes speed.

When the vehicle arrives, the program will calculate vehicle's speed from its speed mean and speed standard deviation by gaussrand function. Then generate its arrival time randomly and calculate corresponding hours and minutes, store all information into etemp (an instance of class entry). Finally store the arrival time and etemp into this day's data(vday).

The next for loop used to generate other four activities every minute during one day. If the remaining distance is less than the vehicle's speed per minute, the vehicle will departure via the end of the road in the next minute. The program will compare the average speed to the speed limit, and print appropriate alarm on the screen if it is overspeed. Finally, it will calculate and store vehicle's departure time and speed into this day's data(vday).

The probability of vehicle departure via a side road is generating randomly. If the Vehicle departure via a side road, the program will calculate and store its departure time and speed into this day's data.

If this type of vehicle is permitted to park on the road(parkflag=0), it can park on the road. By randomly, if it parks on the road, the number of parking vehicles will increase by one, this vehicle will be record in this days parking start vector (vvD.parkS) and insert all information in this day's data(vday).

The maximum time a vehicle can park on the road is 60 minutes. (PARKINGTIME 60). The vehicle will stop parking at a random time and will be record in this days parking end vector (vvD.parkE) then insert all information in this day's data(vday).

If the vehicle changes speed, the program will calculate and store time and speed into this day's data.

If the vehicle keep moving without any situation happened, calculate the reset distance and upload it to the distance variable.

Put this vehicle's events (vVD) into vehicle type's information(vVs). After generating all events of the total number of this vehicle type, store vVs into this day's detail (vDD). After all vehicle type of this day are generated, store the whole day's information (vday)into logs and days detail (vDD) into dayHead (used in analysis engine). Repeat the number of days' times to generate all the events for required days.

The activity engine is completed.

2.The log file called "logs.txt", first line is the day number, then record all the events in that day by time order.

From the second line, each line has the format as follows:

Activity time, Vehicle name, Vehicle license, Vehicle activity, Vehicle speed.

This format output the logs clearly and easy to read.

3.When Vehicle departure via the end of the road, the program will compare its average speed to the speed limit to determine whether the vehicle is overspeed or not. If the vehicle is overspeed the program will print an alarm on the screen.

## Part 3: Analysis Engine

1. The file containing the daily totals for the events is StatsEachDay.txt.
2. The vehicle speed may exceed the speed limit, so when reading the logs.txt the program will check speed of each event and record the overspeed vehicles in the AnalysisEngine.txt.
3. The program will check the number of parking vehicles and compare to the parking spaces. If all the spaces are full, the vehicle which would like to park on the road will be record as illegal parking and record in AnalysisEngine.txt.

## Part 4: Output Files

### 1. logs.txt :

Format :            Day number  
                     time, Vehicle Type, License, Event Code, Current Speed

Event Code :    0 - Arrival  
                     1 - Departure via side road  
                     2 - Departure via end road  
                     3 - Start parking  
                     4 - Stop parking  
                     5 - Changing speed

### 2. AnalysisEngine.txt

Format :            Day number  
                     Number of arrival  
                     Number of departure via side road  
                     Number of departure via end road  
                     Number of parking times  
                     Number of changing speed

### 3. StatsEachDay.txt

Format :            Day number  
                     Vehicle type: Number mean: Number standard deviation: Speed mean: Speed  
                     standard deviation:

Number mean and number standard deviation are based on all events before and including that day.

Speed mean and speed standard deviation are based on the events on that day.

### 4. Report.txt

Format :            Day number  
                     Volume threshold  
                     Total speed threshold  
                     Speed threshold of each vehicle type  
                     Anomaly number . Whether this day is OK or not

### 5. TotalStats.txt

Format :            Day number  
                     Vehicle type: Number mean: Number standard deviation: Speed mean: Speed

standard deviation:

These statistics are based on all events generated by activity engine.