

Software Assignment Documentation

Very High level functionality

MOD003212

Contents

App User Guide	3
Design.....	3
Pseudocode	3
Class Diagram	4
Activity Diagram	5
Testing	6
Screenshots	7
References.....	10

App User Guide

This program is a Petrol Station simulator for Petrol Somewhat Unlimited Ltd. As soon as you execute the program, an automated process will start to direct each spawning vehicle to a free gas pump providing the fuel the vehicle needs and saving the transaction. There are 3 types of vehicles: HGV that only run on diesel; Vans run on both Diesel and LPG and finally Cars on all three types of fuel (Unleaded). If no pump is available vehicles will start forming a queue. If the service delays then many drivers will turn around and fuel at 'Petrol Nearly Unlimited Ltd' which is only 5 miles down the road. To view the transactions simply press the List 'Transactions' button and then the 'Refresh' button to update the list box data. When you wish to exit the program simply close the window.

Design

Pseudocode

do

 Spawn a Vehicle

```
    if it is a car then
        fuelType = Rand(Unleaded, Diesel, LPG)
        fuelTank = 40
        currentFuel <= Rand(1/4 * FuelTank)
```

```
    if it is a Van then
        fuelType = Rand(Diesel, LPG)
        fuelTank = 80
        currentFuel <= Rand(1/4 * FuelTank)
```

```
    if it is a HGV then
        fuelType = Diesel
        fuelTank = 150
        currentFuel <= Rand(1/4 * FuelTank)
```

 Select an available pump

```
        if a pump is free then
            change the pump status to busy
            create a transaction item
```

```
        else all pumps are busy
            increment a value to the queue of waiting vehicles
                if queue of waiting vehicles is > 5
                    then stop spawning vehicles
```

```
    A vehicles just left
    Change pump status to free
    Add transaction item to the counter
    decrement a value to the queue of waiting vehicles
        if queue of waiting vehicles = 5
            then start spawning vehicles again
```

while(Program is running)

This Pseudo Code is a way of simply describing a set of instruction that my program follows to solve the final problem of representing a working petrol station. (BBC, 2017)

Class Diagram

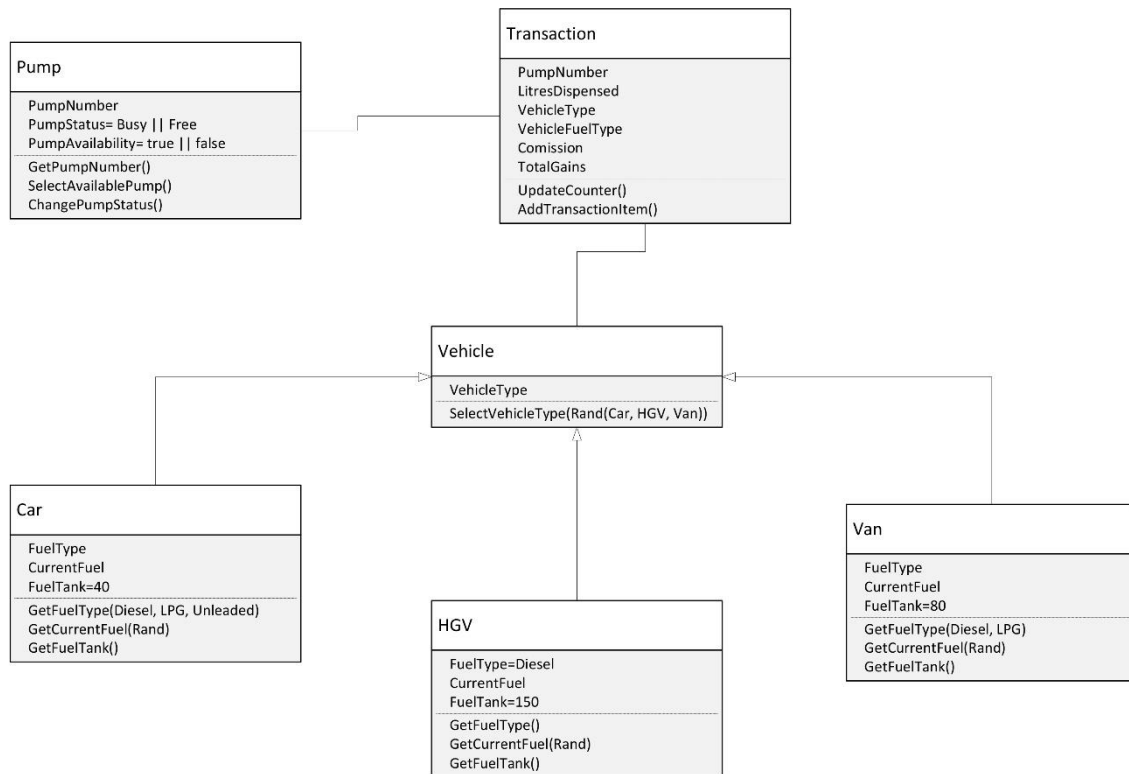


Figure 1 Class Diagram

In my project I decided to create a class, named vehicle, which holds all the vehicle data and it will be responsible for spawning a vehicle with random characteristics every time.

The pump class has one main function responsible for automating the process of selecting an available pump, returning the selected pump number.

Activity Diagram

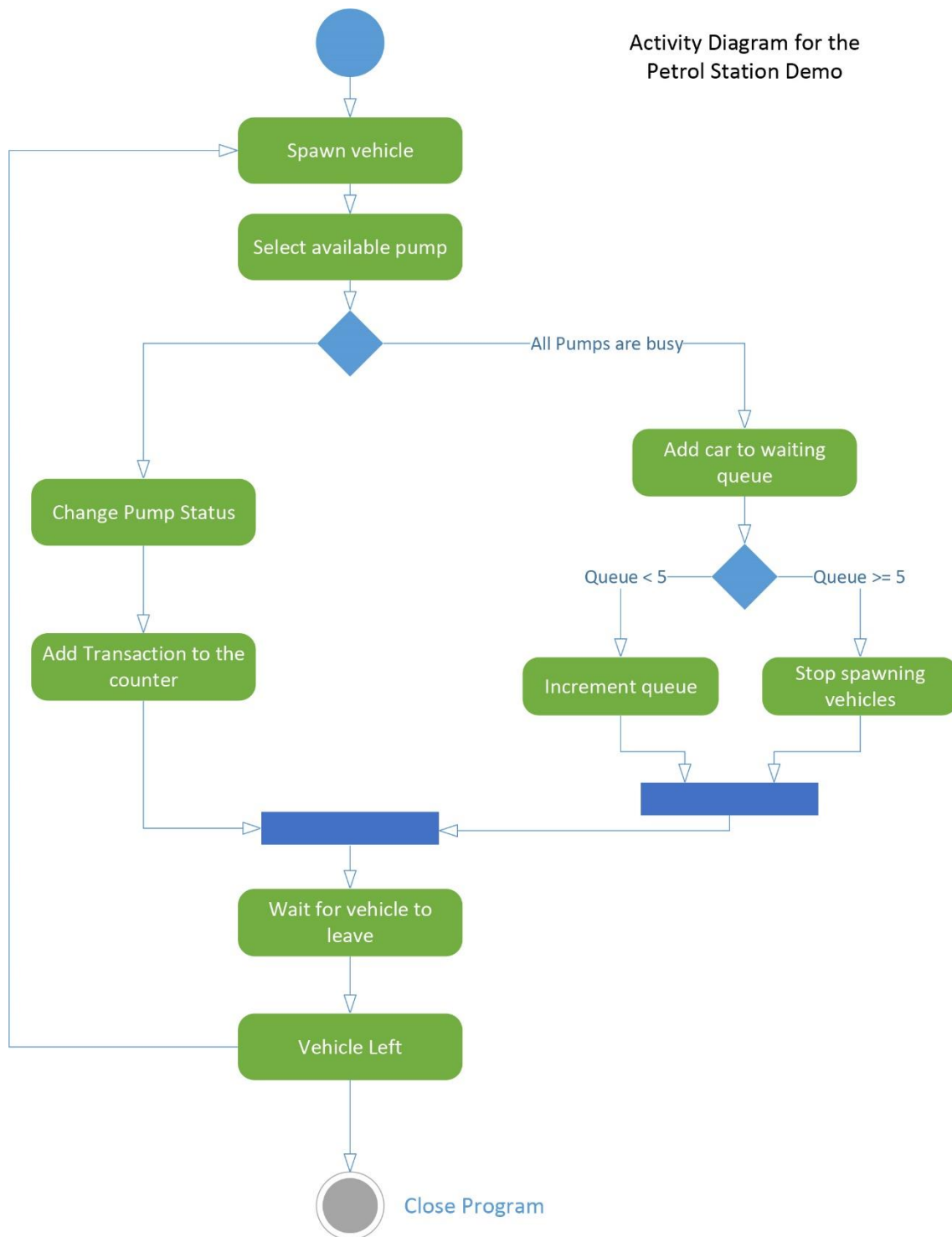


Figure 2 Activity Diagram

This activity diagram simply describes the infinite loop that the program goes through when it's run. It shows some of the decision points present throughout my code and way of thinking.

Testing

Test No.	Purpose of test	Expected Result	Pass/Fail	Date
1	Make sure all types of vehicles are being spawned.	All vehicle types are spawned correctly.	Pass	5/5/17
2	Make sure that obstructed pumps don't get selected.	Automated process should check and never outputs a busy pump.	Pass	5/5/17
3	Make sure all fuel types are being selected correctly.	All fuel type is dispensed correctly.	Pass	5/5/17
4	Transactions display correctly.	No items/transactions repeated in the list box	Pass	5/5/17
5	Make sure the picture boxes are correctly displayed.	Picture box display according to vehicle type.	Pass	5/5/17
6	Timer intervals get set according to the fuel dispensed.	Different timer intervals.	Pass	5/5/17
7	Make sure the display is updated correctly.	All the labels are up to date.	Pass	5/5/17
8	Make sure the queue never goes beyond 5.	The queue only reaches 5 and then stops spawning vehicles.	Pass	5/5/17

Screenshots

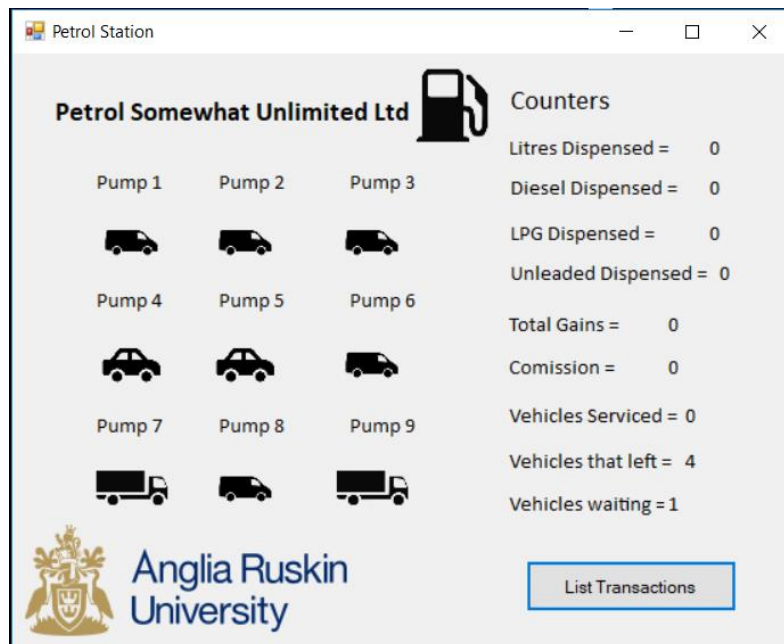


Figure 3 Evidence of all the vehicle types being spawned.

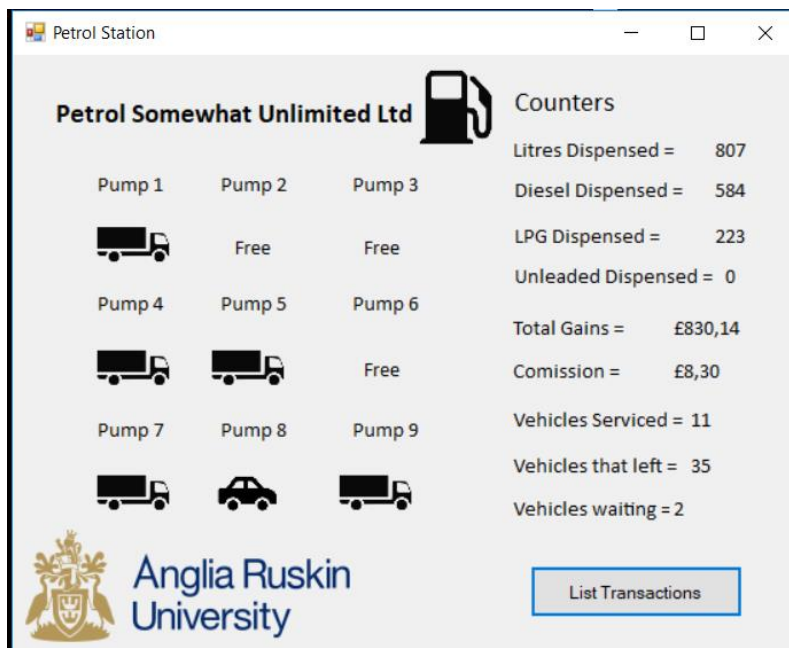


Figure 4 Evidence of the obstructed Pumps not being selected.

TransactionForm

Pump Number	Vehicle	Litres Dispensed	Fuel Type
3	Van	71	LPG
2	Van	69	Diesel
1	Van	68	Diesel
6	Van	72	LPG
5	Car	40	LPG
4	Car	40	LPG
9	HGV	121	Diesel
8	Van	68	Diesel
7	HGV	122	Diesel
5	HGV	116	Diesel
4	HGV	114	Diesel
3	Van	68	Diesel
2	Van	68	Diesel

Refresh

Figure 5 The list box is displaying the correct transactions and the requested data.

Counters

Litres Dispensed =	327
Diesel Dispensed =	179
LPG Dispensed =	109
Unleaded Dispensed =	39
Total Gains =	£324,04
Comission =	£3,24
Vehicles Serviced =	6
Vehicles that left =	26
Vehicles waiting =	1

List Transactions

Figure 6 All three types of fuel are being correctly dispensed.

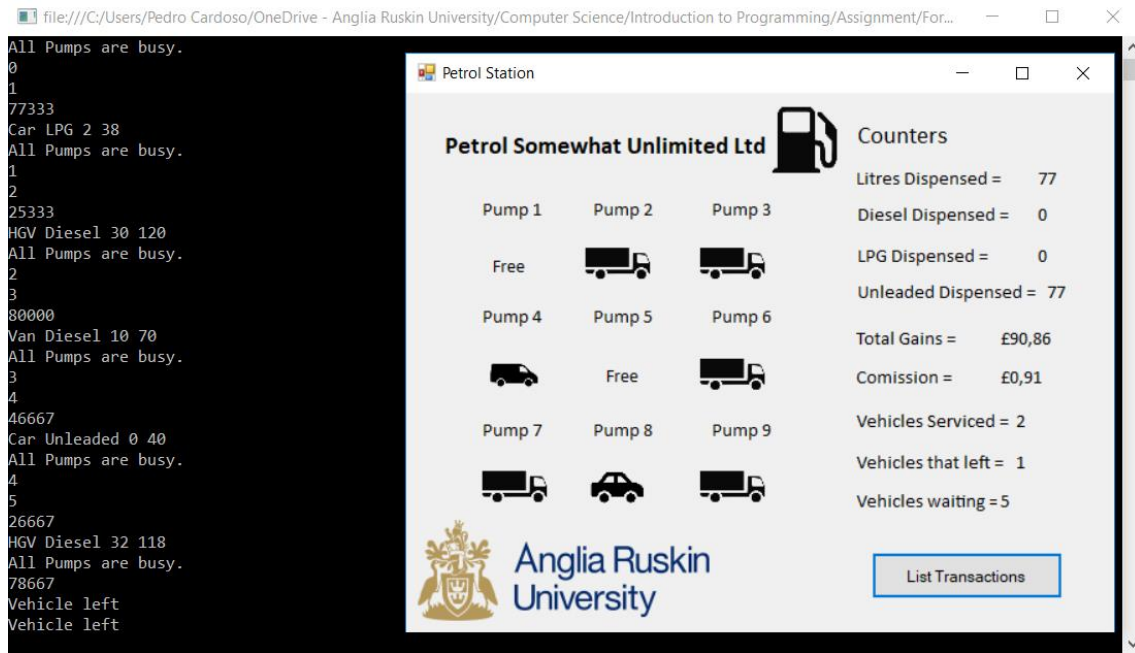


Figure 7 Evidence of the vehicle wait queue reaching five and stopping there.

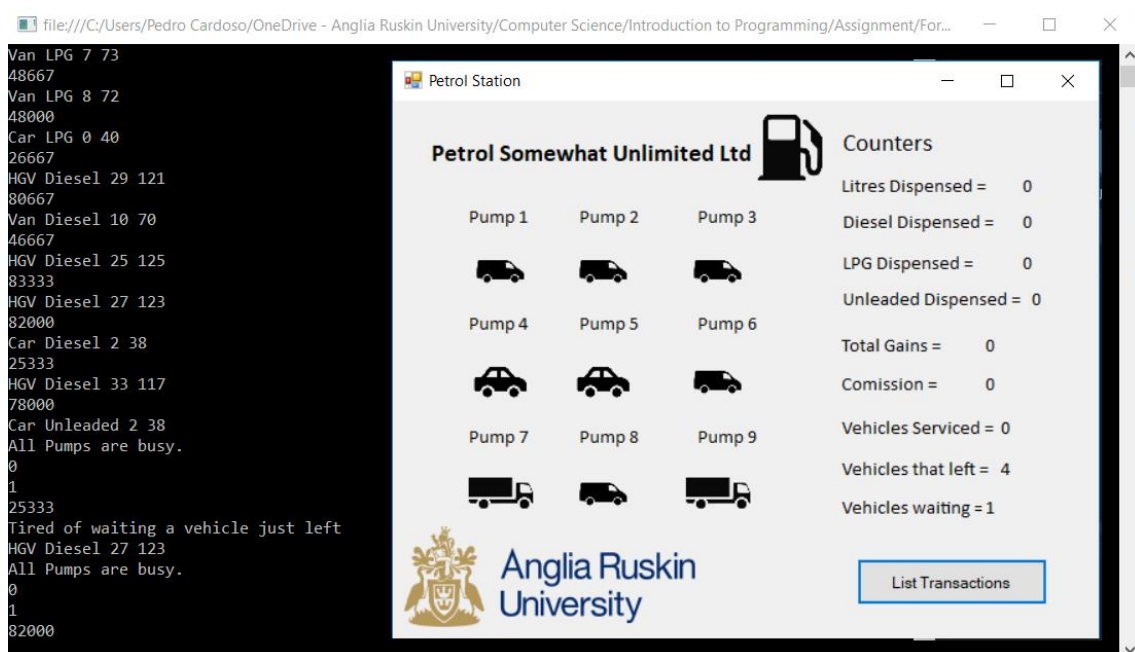


Figure 8 Most of my debugging and testing was made using the console output mode. This allowed me to keep track of my code.

References

BBC, 2017. *Designing an algorithm*. [Online]

Available at: <http://www.bbc.co.uk/education/guides/z3bq7ty/revision/2>

[Accessed 5 May 2017].

Camilleri, D., 2017. *Noun Project*. [Online]

Available at: <https://d30y9cdsu7xlg0.cloudfront.net/png/54821-200.png>

[Accessed 31 March 2017].

GETECH, 2014. *Bus, Coach & HGV*. [Online]

Available at:

http://getech.org.uk/Images/uploads/icon_range_hgvApr_Wed_9_2014_1649.png

[Accessed 31 March 2017].

iconsDB, 2017. *Black car icon*. [Online]

Available at: <http://www.iconsdb.com/icons/preview/black/car-xxl.png>

[Accessed 31 March 2017].