Vehicle-Handover-Library.md

## Description

The **VehicleHandoverLibrary** offers functionality to transfer vehicles between different Traffic Simulations. For every team there is a designated **Amazon SimpleQueueService** in the background. Users of the library may push vehicles to their group's queue. Furthermore, users are able to subscribe to other queues for incoming vehicles.

## Changelog

| Assembly Version | Changes | From |
|---|---|---|
| 1.0.0.0 | Initial Version | dschoeninger.itsb-m2016@fh-salzburg.ac.at |

## Prerequisites

- Download the **VehicleHandoverLibrary.dll** and add it as a reference to your project. (https://www.dropbox.com/sh/mk5hqay2orn6vex/AADKabqOg_4dlwaI8GEKw3Pta?dl=0)
- Add the following **NuGet** packages to your project
    - Newtonsoft.Json (version: 10.0.2)
    - AWSSDK.SQS (version: 3.3.2.2)
- Your **packages.config** should now contain the following dependencies:

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="AWSSDK.Core" version="3.3.13.4" targetFramework="net452" />
  <package id="AWSSDK.SQS" version="3.3.2.2" targetFramework="net452" />
  <package id="Newtonsoft.Json" version="10.0.2" targetFramework="net452" />
</packages>
```

- The example code below shows how to use the library.

## Example Code

```
using VehicleHandoverLibrary;

class Program
{
        static void Main(string[] args)
        {
                // Create Receiver & Subscribe to GROUP2's message queue
                var vehicleReceiver = new VehicleReceiver(Groups.GROUP02);
                vehicleReceiver.ReceiveEventHandler += VehicleReceiver_ReceiveEventHandler;

                // Create sender that pushes to GROUP02's message queue
                var vehicleSender = new VehicleSender(Groups.GROUP02);

                // Define vehicle
                var vehicle = new Vehicle();
                vehicle.Length = 5;
                vehicle.Width = 2.3;
                vehicle.MaxAcceleration = 9.81;
                vehicle.MaxDeceleration = 12.3;
                vehicle.MaxVelocity = 300;
                vehicle.Type = VehicleType.CAR;

                // Push vehicle
                vehicleSender.PushVehicle(vehicle);
                Console.ReadLine();
        }

        private static void VehicleReceiver_ReceiveEventHandler(object sender, VehicleEventArgs e)
        {
                Console.WriteLine("Received " + e.Vehicle.ToString());
        }

}
```

## Datamodel

The library only consists of a **Vehicle** class, a **VehicleType** enumeration and a **Groups** enumeration.

### Vehicle class

- Namespace: VehicleHandoverLibrary
- Properties:

```
// Maximum acceleration in m/s^2
public double MaxAcceleration { get; set; }

// Maximum deceleration in m/s^2
public double MaxDeceleration { get; set; }

// Maximum velocity in m/s
public double MaxVelocity { get; set; }

// With of the vehicle in m
public double Width { get; set; }

// Length of the vehicle in m
public double Length { get; set; }
```

### VehicleType enum

- Namespace: VehicleHandoverLibrary
- Enum:

```
public enum VehicleType { CAR, TRUCK, BIKE };
```

### Group enum

- Namespace: VehicleHandoverLibrary
- Enum:

```
public enum Groups { GROUP01, GROUP02, GROUP03 }
```