# Gamified Virtual Reality Driving Simulator for Asserting Driving Behaviors


Ahmed Ali
Faculty of Media Engineering and Technology
German University in Cairo
Cairo, Egypt
ahmed.t.mansour@gmail.com

Ahmed Elnaggarz
Intelligent Interaction Lab
DHBW Stuttgart
Stuttgart, Germany
ahmed.elnaggar@dhbw-stuttgart.de

Dirk Reichardtz
Intelligent Interaction Lab
DHBW Stuttgart
Stuttgart, Germany
dirk.reichardt@dhbw-stuttgart.de

Slim Abdennadher
Faculty of Media Engineering and Technology
German University in Cairo
Cairo, Egypt
slim.abdennadher@guc.edu.eg



***Abstract*—Virtual Reality (VR) is a very promising field that allows the users to explore a 3D world and immerse in it. *Driving Simulators* are one of the most common VR applications that integrate the users within a learning experience. This project focuses on the most important driving behaviors that have to be adapted by any driver, such as switching lanes and giving priority to pedestrians in order to reduce the rate of car accidents. The main objective is to implement a driving simulator where the driver can exercise these behaviors in a form of 4 gamified scenarios with a scoring system and analyzing techniques. Microsoft sidewinder force feedback is used as an input to enhance the multisensory experience. VR tracking system is used as part of the system that analyzes the player's behavior to alter the scene and expose the driver to unexpected situations. The system includes different algorithms as well that measure the performance of each driver and track the improvements that occur to the targeted behaviors. The feedback of the players was gathered using a survey, while the improvement in their performance was measured using observation of each of their behaviors. The survey's results showed that 88% of the test subjects believe that this teaching experience may help correcting the addressed driving behaviors and 84% of them found it amusing. The statistical analysis of the change of the drivers' behaviors shows an average increase of 21% of the correct actions and an average decrease of 17% of the wrong actions.**



***Keywords—virtual reality; driving simulator; gamification; artificial intelligence; oculus rift***


## I. Introduction

Virtual Reality (VR) has been allowing individuals to recreate senses artificially for the past few decades and it has a potential for the future. Aside from that, VR may be able to change our current present. VR can easily model a whole world which can be used in teaching experiences that would immerse the user in its environment. When it comes to teaching experiences, e.g. teaching driving behaviors would be a very rich topic to discuss and explore [1].

Young people with little driving experience are often the main reason behind car accidents. These young drivers do not get real life exposure to real life situations while learning how to drive which results in driving mistakes that lead to fatal accidents. Hence, they need to be exposed to more driving time in a safe and reliable environment in order to live situations that are similar to real life, to see the potential consequences of driving mistakes, in order to change the behaviors that may lead to a crash and to gain more driving experience.

*Driving simulators* are widely used for many purposes such as assessment, rehabilitation, teaching driving skills, road safety researches and entertainment. They are favored over real driving situations when it comes to physical safety, analysis and certain teaching scenarios.

Driving simulators allow the users to control all the elements surrounding their learning experience or the factors that may affect the driving behavior. They also have the ability to reproduce the scenarios they need to stress on. On the other hand, real life driving experiences are random and uncontrollable, so having the fullest needed experience can only be guaranteed with driving simulators that generate random events [2].

This paper aims to test the capabilities of the *gamified virtual reality driving simulator* that was implemented earlier using Oculus Rift, Microsoft sidewinder force feedback and Unity. The simulator intends to teach the drivers certain driving behaviors like overpassing cars and giving priority to pedestrians by putting them in real life scenarios and tough situations that can't be faced in a real life driving school. The simulator uses *VR tracking system* and the scenarios' controller to expose the drivers to unexpected or unsafe scenarios without putting their lifes in danger. An analyzing system is integrated within the simulator to obtain accurate measurements for each driver and each behavior.

This driving simulator is meant to analyze the driver's behaviors and to be able to give a constructive feedback or to generate an unexpected event accordingly to assert the targeted behaviors. This represents a very good base for a potential future system that alters and generate different events according to the measured behaviors of the driver.

## II. Hardware and Software

### A. Oculus Rift Developer Kit version 2

Oculus Rift is one the most successful Head Mounted Display that totally immerses the user in a virtual 3D environment due to its high resolution display and large field of view. Oculus Rift allows the users to control their physical actions in the virtual world the same way they do in real life. A gyroscope, an accelerometer, and a magnetometer are included in Oculus Rift to keep track of the orientation of the player's head and apply the same adequate changes to the virtual world. Oculus Rift includes a position tracker that synchronize the position of the player in the virtual world with his position in the real world [3].

### B. Game Engine: Unity 3D

Unity 3D was the most adequate platform to build a driving simulator that uses Oculus Rift and is based on gamified concepts due to its capabilities, its wide community, the documentation offered by Unity developers, its asset store that includes free assets, standard assets offered by Unity itself and, last but not least, its powerful physics engine. It is worth mentioning that Unity highly supports driving games [4].

## III. System Architecture

### A. 3D Modeling

*1) City and Traffic System:* Unity's Terrain System was used to generate a large green area with trees randomly located. Streets are built using different squared shapes of tiles. Each cross road is either considered as a priority zone or a normal traffic zone monitored by a traffic system. A traffic lights' system is one genre of the traffic systems in the city. It controls from 4 to 8 traffic lights.

There are 3 types of lanes in the city: *single lane* which is a single lane road, *right lane* and *left lane* which are the two lanes of a two-lane road. Any lane that leads to a cross road has a traffic light or a stop sign at its end. Each type of lanes has its own dedicated type of traffic lights.

To spawn the cars, there are 6 spawning points dispersed around the city. The number of cars in the scene, spawning points and the paths are all controlled by the *Cars Traffic Manager*. On the other side, the pedestrians' spawning points are generated using an algorithm that sets one or two *Pedestrians Navigation Point* beside each lane. These nodes are used as spawning points and targets for the pedestrians.

*2) Cars:* The auto generated cars are low polygon models imported from Unity's asset store. The wheels' position and rotation is synchronized with those of the corresponding wheels colliders.


Fig 1. Screenshot taken during the gameplay for the driver approaching a cross road controlled by a traffic lights system

The drivable car's steering wheel is synchronized with the input of the *Microsoft sidewinder force feedback driving wheel*, so its rotation is similar to the one the player is exerting in real life.

Other features were added to the drivable car such as: three reflective mirrors, analog RPM, analog speedometer, digital speedometer, blinkers, transmissions' screen and a screen to show the instructions, the score and the time

### B. Physicss

Wheel collider is a special type of colliders designed specifically for grounded vehicles. It helps turning a 3D car model into a realistic moving car. Wheel colliders detect the collisions the same way normal colliders do.

The wheel colliders has two attributes that control the movement of the car the same way a car's motor engine does in real life. Those two attributes are the motorTorque and the brakeTorque. They are simply two torques applied on the wheel collider but in opposite directions with the intention to accelerate the car and stop or decelerate the car respectively.

The motorTorque is a given number that was chosen by the trial and error method to have the most realistic experience. However the brakeTorque is derived in the form of a math equation where the velocity, the distance required to stop, the radius of the wheel collider and the mass of the wheel collider are the most contributing factors [5]. The equation is derived from the *Second Law of Motion* and the acceleration's equation. It calculates the brakeTorque (T) required to be exerted on each of the four wheels, each of radius (r) and mass (m), to stop the car that has an initial velocity $v_i$ in a distance (d):

$$T = r * m * - v_i^2/d \quad (1)$$

The motorTorque and the brakeTorque are based on the input exerted by the player through the Microsoft sidewinder force feedback pedals.

### C. Artificial Intelligence

*1) Auto Generated Cars:* The auto generated cars use the Waypoint Navigation method to move in the city. Waypoint Navigation method is frequently used in many car racing games to provide artificial intelligence for the cars in the game. This method is simply based on nodes cautiously placed in the game environment. The cars move between each point to complete a full path. There are 6 different paths in the environment. The path is assigned to the auto generated car according to its spawning point [6].

The steering angle of the cars' wheels is based on the location of the node the car is directed to. It is calculated in each frame by multiplying the max steering angle for the wheels by the ratio of the x value and the magnitude of the vector between the car's current position and the position of the node it is directed to [6].

The Waypoint Navigation was enhanced by adding 4 different states and the generated braking equation to allow the car to abide by all the traffic rules. The states are checked according to their priorities: avoidObstacle, Stop, isTurning and Move. The current state of the auto generated car is always the one with the highest priority that is turned on.

The auto generated car detects that there is an obstacle it has to avoid by generating three different rays from its middle, right front corner and left front corner. If any of the rays hits an obstacle, the **avoidObstacle** state is turned on. It is turned off again when none of the rays is hitting any object. **avoidObstacle** has the highest priority. Whenever it is on, the distance between the car and the obstacle is tracked as the *distance to collide*. If more than one ray is detecting an obstacle, the distance to collide will be equal to the minimum distance between each ray's origin and the obstacle it is hitting [7].

Each lane leading to a cross road has a stop area and is assigned to either a traffic light or a priority zone. The stop area attached to a traffic light turns the **Stop** state of the auto generated car on if the corresponding traffic light is red and either the distance between the car and the cross road is less than 3 meters or the car's velocity is more than 10 km/h. On the other hand, the stop area attached to a stop sign turns the state on if the distance between the car and cross road is less than half a meter and there exists a car that has a higher priority, which in this case is the car coming from the lane on the right side of the car, or the car's velocity is more than 15 km/h. The car remains stopped only if there are cars coming with a higher priority. When the car exits a stop area, its **Stop** is turned off.

Whenever the auto generated car is less than 50 meters away from the node it is directed to, it checks on the vector between that node and the node that follows. If a potential turn is found and the car's velocity is more than the maximum velocity allowed for the car while turning, the **isTurning** state is turned on. Otherwise, that state would be turned off.

If any of the above states is chosen as the current state of the car, a torque that depends on the distance that the car needs to stop in is applied to the brakeTorque attribute of the four wheel colliders of the car.

**Move** is the state that has the least priority. However, it is the default state of the car. When this state is on and the speed of the car is less than the maximum allowed speed, the moving forward torque is assigned to the front wheel colliders' motor torque of the car [8].

*2) Pedestrians:* The pedestrians moving in the city are based on the *Third Person Character* asset which is part of Unity's standard assets. The character is configured to move in the street and to adjust its animation with the current speed he is moving in. A controller was added to that pedestrian prefab to make him move in the walkable areas in the city and to abide by the traffic rules.

Unity's fundamental navigation system is used to make the pedestrians walk in the city. The Nav Mesh, which is the walkable area in the 3D world, is set as the sidewalks of the streets and the crossing areas in the traffic zones. A Nav Mesh Agent is any game object that navigates on the Nav Mesh given a target to reach. The agent's target is assigned randomly from the list of Pedestrians Navigation Points. When a pedestrian reaches its target, a new target is randomly assigned.

The controller of the pedestrians has two states: Move and trafficStop. Move is the default state of the pedestrian where it moves to its target in its desired velocity. trafficStop gets turned on when the pedestrian is in the range of a traffic light, willing to pass in its pedestrian crossing area and the traffic lights that may intersect with that crossing area are not red [9].

*3) VR Tracking System:* Unexpected Events: Some unexpected events are scheduled to happen according to the behavior of the driver. One of the unexpected events is based on the VR Tracking System: it occurs if the player did not check his left side by looking into the car's left window while exiting the parking, as shown in figure 2. If he forgets to check his left side, a car appears suddenly on the left and hits him. Otherwise, he will move smoothly without any car being generated to collide with him

Fig. 2. Screen shot from Unity's gameplay where the player exits the parking without checking his left side and being hit by the auto generated car

## D. Analysis

Simulators have helped broaden the field of analysis within the virtual driving experiences because of their measurement capabilities. The analysis of the driver's performance leads to an accurate assessment for his driving behaviors which will certainly help asserting them. This paper focuses on the behaviors listed below which may lead to traffic accidents:

**Abiding by the speed limit**: The velocity of the player's car is tracked. If he passes the speed limit, an exceeding speed limit fault will be noted.

**Controlling the car**: It is affected when the player's car detects any collision. A collision fault is noted each time the player collides.

**Giving priority to pedestrians**: It is assessed each time a driver passes a walkable area. If while passing, a pedestrian walking along that same area is detected, a pedestrian's priority fault is noted.

**Use of blinkers**: This behavior is measured whenever the player is switching between two lanes or turning: it checks if the player has turned on the corresponding blinker or not. If a blinker is missed, a fault is noted. If a blinker is turned on correctly, a successful blinker is noted.

**Use of mirrors**: This behavior is measured whenever the player is switching between two lanes: it checks if the player has looked into the corresponding mirror or not. VR Tracking System is used to detect where the eye sight of the player is directed: It is based on issuing a ray from the center of the Oculus Rift in the forward direction; the first object that ray hits is what the player is looking at. If the player misses to look in the mirror when he should too, a fault is noted. If he looked in the corresponding mirror before switching between the lanes, a successful look in the mirror is noted.

**Overpassing**: It is analyzed when the player is moving between two lanes, in other words he is triggering the left and right lanes. If the player didn't use a blinker or look in the corresponding mirror while triggering the two lanes, he did an overpassing fault. Otherwise, he correctly overpassed or switched from lane to another.

**Passing Traffic Lights**: A collider is placed under each traffic light to detect the cars passing traffic lights. When the player exits that collider, the traffic light is checked: if it is red, a fault is noted otherwise a successful pass is noted.

**Giving priority to the cars in the priority zones**: It checks if the player has stopped beside the stop sign or not. When he stops, the lane leading to the east of the cross road is checked. If there is a car that is moving towards the cross road and the player didn't give it the priority to pass first, a giving priority fault will be noted. Otherwise, a correct given priority will be noted.

**Unfastened Seatbelt**: there is a dedicated button in the driving wheel controller to fasten the seatbelt. If it is not fastened and the car has started to move, points are deducted from the player's score. If the player exceeds 15 km/h, a sensor starts to beep to remind him of the seatbelt.

All the noted faults will result in a deduction of the points of his score

## IV. Scenarios

The Driving Simulator is implemented in a form of a game to add an entertaining factor to the experience and to keep the players motivated to improve their behaviors in order to achieve a higher score. The game constitutes of four scenarios: WarmUp Scenario, Follow Me Scenario, Follow Instruction Scenario and Timed Scenario.

The last three scenarios are implemented in the city with the complete traffic system, cars and pedestrians. The player earns and loses points according to his performance when any of the targeted driving behaviors are checked.

### *A. WarmUp Scenario*

The objective of this scenario is to introduce the player to the basics of the driving simulator and to get used to the virtual car and the virtual system. The player has to follow audio instructions in a 3 lanes road; he has to abide by the speed limit announced by the vocal instructor. He is requested also to switch lanes, face cars, pass them and stop in a certain distance. The player starts this scenario with a score of 1000 points and they get deducted whenever he makes a mistake.

### *B. Follow Me Scenario*

The objective of this scenario is to add some pressure on the player while driving in the city and yet he has to abide by the rules. The player is requested in this scenario to follow another car till they reach their final destination.

### *C. Follow Instructions Scenario*

This scenario aims to assess all the player's behaviors, to make him experiment how to reverse park between two other cars and to expose him to unexpected scenarios based on the VR Tracking System, such as being hit by another car if he goes out of a parking without checking his left side. The player has to follow audio instructions in this scenario and these instructions are also replicated on the screen inside the car in the form of directional arrows.

### *D. Timed Scenario*

This scenario is a replica of the previous scenario but with a count downing timer. The goal of the scenario is to make the player repeat a scenario that he has already explored, without any pressure, under the time's pressure and then to compare how this pressure can affect his driving behaviors. In this scenario, only the audio instructions are available for the player, so he has to carefully listen and focus on the road at the same time.

## V. Results

During this research, 83 experimental sessions were conducted involving 13 test subjects. 11 of the test subjects have fully completed the test as required. Each of them performed 7 sessions by playing the warm-up scenario once, the follow me scenario twice, the follow instructions scenario twice, and the timed scenario twice.

The test subjects age varies between 20 and 23 years old. Candidates those were older than 25 were excluded from testing as the project's focus is on youth. 6 of the test subjects were males while 7 were females. Only 3 of them used a VR system before, while 10 of them had never used any VR system before. Two of the test subjects do not have a driving license while the rest of them do, five of the test subjects have 4 years of driving experience, two of them have 3 years of experience, only one test subject has 2 years of experience and another one has only one year of experience.


Fig. 3. One of the test subjects during the test, wearing Oculus Rift and driving the car using the steering wheel and the pedals.

The test subjects were asked to perform the warm-up scenario once to get used to the system, and the rest of the exercises twice to check if their driving behavior changed after playing the same scenario or not. The total amount of time that was taken to complete the 7 sessions for each test subject was approximately 45 minutes. During each scenario, the system recorded their driving behavior, and after completing the 7 sessions, they were asked to fill a survey to give their feedback about the system.

### *A. Driving behavior Analysis Results*

After the 83 sessions, the recorded data of the driving behavior were filtered and divided into two parts. The warm-up data were removed from the analysis as it was only made to make the players get used to the system. The rest of the data were categorized into 2 categories: the correct actions done by each test subject in each session and the mistakes done by each test subject in each session that are relevant to the driving behaviors in focus. Figure 4 shows the average number of correct actions that lead to a correct driving behavior that all players have made in each session, while figure 5 shows the average number of mistakes that lead to a bad driving behavior made by all the players in each session.

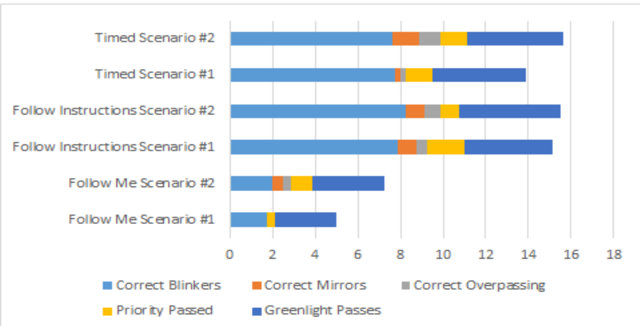


Fig. 4. The average number of correct behaviors in the same session

As shown in figure 4, each correct behavior is represented by a different color. Each session played by the test subject is represented by a block on the vertical axis. The number of times this behavior was done by the test subject is represented on the horizontal axis. We can clearly see that the average numbers of all the correct behaviors mentioned earlier which were done in the same scenario by all the players were increased when the players repeated the same scenario.

These results prove then that the system was able to reinforce the good driving behaviors for the test subjects. On the other hand, the comparison of the Timed Scenario results with the corresponding Follow Instructions Scenario results shows that the total numbers of correct behaviors in the Timed Scenario were less the total number of correct behaviors in the Follow Instructions Scenario despite that all the test subjects played the Follow Instructions Scenario first and the Timed Scenario is just its replica but with a time constraint. This concludes that the drivers care less about the correct driving behaviors when they are under pressure, in this case the pressure is the countdown timer.

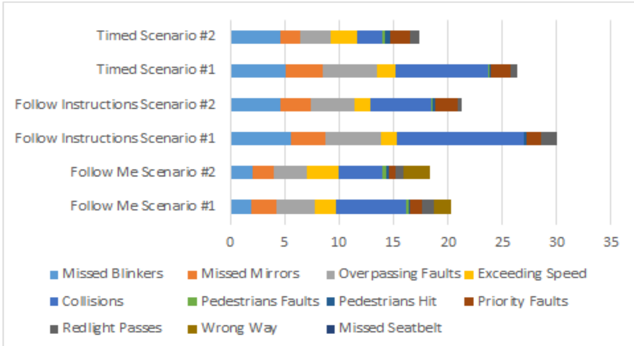


Fig. 5. The average number of incorrect behaviors in the same session

Moreover, by analyzing the results from Figure 5 which contains a graph similar to the graph of Figure 4 with the same structure but with different behaviors, we can conclude that the average number of mistakes that led to bad driving behaviors that was recorded were decreased after repeating the same scenario for another time by the same test subject.

The change of correct actions and wrong actions for each user were analyzed to determine the average increase or decrease of the users that used our system. We found that the average change in the correct actions that leads to bad behaviors were increased by 21% , while the average change in the wrong actions were decreased by 17%.

### *B. Survey Results*

The test subjects were asked to fill out a survey in order to give their feedback on the system. The data collected from the survey is used in a combination with the data gathered during the gameplay to generate conclusions regarding the effectiveness of the driving simulator and how can it be improved. The test subjects were asked to rate the system's performance on a scale from 1 to 5, where 0 is bad and 5 is excellent.

As shown in figure 6, the average of rating given to the quality of the graphics by all the participants was 3.5, which is a reasonable rating due to the use of low polygon models. Quality of AI of the traffic system, cars and pedestrians were rated 3.8, 4 and 3.5 respectively. The pedestrians AI had the least rating because of two main issues: The first was pedestrians blocking each others' ways and then getting stuck for a while. The other issue was pedestrians getting stuck sometimes between the sidewalk and the street while crossing a road. Those two issues were due to initial problems with the streets 3D models. All the test subjects agreed that they were fairly assessed and that the game was entertaining.

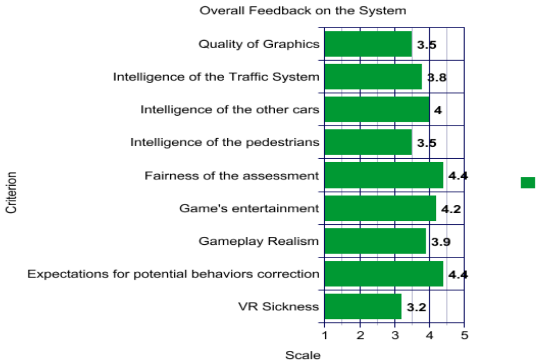


Fig. 6. Test Subjects' feedback on the whole system

Moreover, they felt the gameplay was real and they highly believe that this simulator is capable of asserting the driving behaviors in the scope of the project. The average amount of pain was 3.2 in the participants trials. The experience has been painful to more than half of the test subjects. Two of them couldn't continue testing as they experienced a simulator sickness. However, the headache that most of the participants experienced wasn't part of a VR Sickness. It was due to the use of Oculus Rift B lenses and not A lenses that are made for people with a correct sight. Unfortunately, the A lenses had a scratch that affected the gameplay. We asked the test subjects to stop playing whenever they start experiencing any pain. 4 test subjects achieved the target without taking any rest. Most of the other test subjects took a few minutes rest between each one or two scenarios.

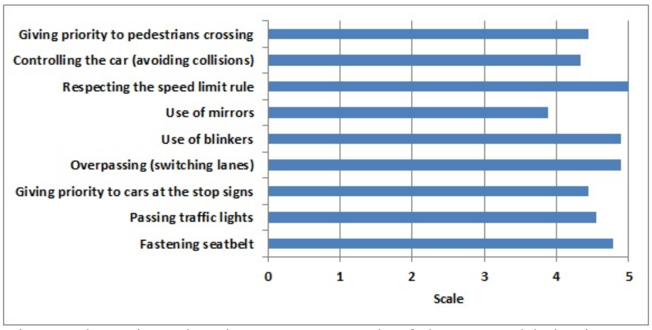


Fig. 7. The ratings the player gave to each of the targeted behaviors as implementation and assessment

Finally, according to the results obtained which are shown in figure 7, the test subjects haven't faced any difficulty using the driving wheel to perform the various driving behaviors. However, they could not use the mirrors easily because in order to look into the mirrors the test subjects must move their heads to the right or to the left or even bend their whole bodies, in order for Oculus to change the camera position.

## VI. Conclusion And Future Work

The advancement in the VR systems as well as the gaming engines allow the researchers to develop a safe environment for simulating real life scenarios and a rich learning platform. Using these technologies, a VR driving simulator was built to enhance the driving skills for drivers. The system used Oculus Rift, Unity, and Microsoft wheel to simulate real life scenarios for drivers. Various algorithms were designed to control the flow of these scenarios, as well as to analyze the users behavior during each scenario. Both the data collected through each session as well the survey that was filled by the test subjects proved that we could change the driver's behavior through a safe and entertaining environment.

Several features could be added to improve the functionally of the system. Additional levels can be added to the game targeting drivers with different experiences. A group of levels with special set of instructions can be dedicated for the people who are still learning how to drive. Those levels can also include special modules for how to drive on a Highway, in a rainy weather, during a rush hour or at night and how to park a car. Extra unexpected events would be a great plus to the experience such as the feature of having a ringing mobile phone, cars breaking the traffic rules and events based on the driver's eyes' orientation. In addition to unexpected events that are only generated according to the measured behaviors of the player. Tracking the drivers behaviors in real life after trying the system would add more credibility to the results. Further measurements can be done during the testing such as the number of excessive braking, acceleration and cornering events with the corresponding intensity.


## Acknowledgment

We would to thank all the test subjects who volunteered to test the system.


## References


[1] P. v. L. J.C.F. de Winter and R. Happee, *Advantages and Disadvantages of Driving Simulators: A Discussion*. Department of BioMechanical Engineering, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, the Netherlands, 2012.

[2] J. De Winter, P. Van Leuween, and P. Happee, "Advantages and disadvan- tages of driving simulators: A discussion," in *Proceedings of Measuring Behavior*. Citeseer, 2012, pp. 47–50.

[3] K. D. A. K. M. Parth Rajesh Desai, Pooja Nikhil Desai, "A review paper on oculus rift-a virtual reality headset," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 13:4, pp. 175–179, 2014.

[4] M. S. Dominik Kirner, *Creation of a 3D first person adventure game with the oculus rift powered by unity*. University of Applied Sciences, Hochschule Offenburg, February 2012.

[5] P. Srisuchat, *Development of a car physics engine for games*. Department of Creative Technology, Faculty of Science and Technology, Bournemouth University, England, 2012.

[6] Y.-B. L. Jung-Ying Wang, *Game AI: Simulating Car Racing Game by Applying Pathfinding Algorithms*. International Journal of Machine Learning and Computing, Vol 2, No. 1, February 2012.

[7] J. W. F. F. Y. Norman Badler, *Vehicle Traffic Simulation*. Department of Computer and Information Science, School of Engineering and Applied Science, University if Pennsylvania, United States, 2011.

[8] C. P. Aung Sithu Kyaw, Thet Naing Swe, *Unity 4.x Game AI Programming*. PACKT Publishing, Birmingham, Mumbai, 2013.

[9] K. V. Kristinsson, *Social Navigation in Unity3D*. School of Computer Science, Reykjavik University, Iceland, 2015.