PeTraLi: A Simulation for The Observation of Drivers' Experiences Assisted Personal Traffic Lights

Çağlayan Sever[1], Kader Nikbay[1], and B. Tevfik Akgün[1]

[1]Computer Engineering Department, Okan University 34959 Tuzla, Istanbul, Turkey
caglayansever@gmail.com, kader.nikbay@okan.edu.tr,
tevfik.akgun@okan.edu.tr

Abstract

In this work, the measurement and improvement of drivers' attitudes are aimed by using data obtained from driving experiences in virtual traffic which is created by the 3D game platform and is enriched by assisting with the Personal Traffic Lights (PeTraLi). For this multi-user simulation which can also be seen as a serious game, vehicles for drivers, driverless vehicles, intersections equipted with PeTraLi system, roads and buildings are implemented. PeTraLi is a proposed system design in this work and includes a roadside equipment for each intersection and a small electronic device in the vehicle. In this paper, development of work, simulation systems and games, simulation systems and simulators of vehicle/driving/traffic are summarized, and PeTraLi system is introduced.

1. Introduction

In this work, the measurement and improvement of drivers' attitudes are aimed by using driving experiences in virtual traffic which is created by a simulation likes a 3D game that enriched by both autonomous and player controllable vehicles with a driver assistance for traffic lights. In this simulation, players drive their vehicles in a virtual traffic environment and try to accomplish their objectives. There are also autonomous vehicles taken place in this environment. These autonomous vehicles respect traffic rules, but may have different behaviors. Intersections of the simulation have no traffic lights. But there is a roadside equipment which monitors all the vehicles, and sends messages to a devices in the vehicles so that preventing accidents for own intersection. Main research subject is to observe how PeTraLi effects in traffic and how behaviors of drivers change. Expected or desired result is to decrease accidents may occur at junctions, intersections and to improve drivers' driving abilities.

Traffic and driving simulators, vehicle simulators, simulates real traffic environment effectively by using vehicle models, traffic lights, roads, buildings and other things. The main objective of this study is handling this subject like a serious game. As a result of this, we avoided special equipments and powerful computer hardware. We also avoided complex traffic management systems and complex autonomous vehicle models to meet minimum requirements. So in this work, sensors not included like we seen in complex vehicle simulation systems. We choose Unity3D as development platform because it has a very easy implementation system to build games on different platforms [1].

In Turkey, more than 70% of traffic accidents occur at intersections. Intersections, whose have no traffic lights with managing of passing is done by general traffic rules, increase accidents rate. In addition, buildings, plants or other blockades on one junction turn cause significant vision problems. NHTSA Technical Report (2007) of U.S. Department of Transportation stated as: "Fig. 1 shows the distribution of the critical reasons attributed to drivers in intersection-related crashes of the 756,570 intersection-related crashes with driver attributed critical reason, the most frequently assigned critical reason was inadequate surveillance (44.1%). In comparison, about 92 percent (1,289,283 crashes) of the nonintersection-related crashes had a critical reason attributed to drivers. The most frequent critical reason in the 1,289,283 non-intersection crashes with driver attributed critical reason was too fast for conditions/curve (22.8%) followed by performance error (15.9%), internal distraction (13.4%), and critical non-performance errors (10.8%)" [2].

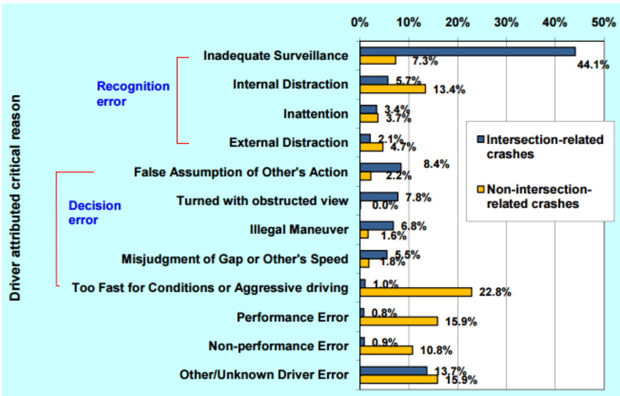

Fig. 1. Driver attributed critical reasons of intersection-related and non-intersection-related crashes.

We designed a conceptual driver assistance system named PeTraLi for intersections. There is a Roadside Equipment (or Roadside Unit, RSU) that monitors all the vehicles within its range. There is a small electronic device in each vehicle as a part of system. If there is an unwanted situation, the system sends messages to the related vehicles for warning drivers. Main goal of the simulation is to observe players driving experiences with using this driver assistance so that Personal Traffic Lights system can be verified.

The previous work of this study carried out as a game named OKANOM for improving abilities of driver for intersection passing scenarios [3]. Recommendations of the players were based on there must be a driver assistance. So we designed a Personal Traffic Lights concept, changed the project completely and rewrote scripts that suit new project. For this case, we changed also vehicle model and implementation of city. Development is still ongoing. While paper writing, vehicle model, autonomous vehicle control, player control assisted by a PeTraLi signals, environment rendering and some basic accident preventing scenarios are implemented. PeTraLi project is supported by OKAN University Computer Research and Application Center, and, Transportation Technologies and Intelligent Automotive Systems Application and Research Center.

2. Simulation Software and Game Engines

In this section, information about 3D simulation software and game engines is given and Unity3D game engine is introduced. Information about driving and traffic simulation systems / simulators is also given in the next section.

Though there are separately differences in main objectives, a 3D simulation software and a 3D game engines have common things especially at the real-time graphic rendering. The definition of real-time in here is producing user-perceivable frames within the time. Displaying 3D objects on the screen according to user and computer control is managing by libraries (ie. OpenGL, DirectX) that manages graphics subsystem which accessed by low-level programming (C language). Several companies created simulation the software/game engines to produce and manipulate environments and objects within a complex graphical software. And these softwares not only produce graphics also support multimedia components (sound, effect, video etc.).

Unlike games, a 3D simulation system needs to do more precise calculations. Therefore, to realize these precise calculations there must be more computing power. The main objective of video games is to live player in a dream which is desired to create. To provide this purpose in real-time, some rendering tricks are used as visual trickery. Given the development process, high cost of required hardware and multitude of special software to be produced led to the earlier examples of simulation systems. Generally the demands for these products are very low for companies which product them in low values and provide all information and software accumulation on the subject. Expansion of simulation systems were not possible at first time, because of there are the high hardware system costs of libraries and software packages to work on. The size of the demanding video games was increased, so it reduced the cost of graphics hardware prices and revealed high performance game engines. Today game engines are important parts of simulation systems.

As a result of the research, it was decided to use the Unity3D as a game engine. The main reasons of choosing Unity3D are can be programmed with scripting language, working on multiple operating systems and producing executable software in different platforms is easy. Free/student version of Unity3D is enough to develop this kind of simulation at developing period. Basic properties of Unity3D are supporting fast development, having interfaces those give developer to make changes easily and having an asset store which have enormous resources to build 3D games and simulations [1]. Unity3D has a multiplayer support with networking system. It has own web player that requires nothing to play games on browsers. Have three programming language support (C#, Javascript and Boo Script) and each one has very rich libraries. Also Unity has own community to share developers experience.

3. Driving and Traffic Simulators

Most important parts of a driving simulator are 3D graphical software, simulation, modeling, calculation software and equipments. This section is about some driving and traffic simulations. These simulations are chosen because of they have similarities to our project topic. Most simulation products are not completely enough. For example topics like vehicle dynamics, modeling and rendering need to be done and must add to simulation project. As we express in introduction section there is difference between producing environment and running environment. And this causes hardware differences between computer systems.

3.1. Sumo

SUMO is an open source, highly portable, microscopic and continuous road traffic simulation package designed to handle large road networks. It is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Center. SUMO is open source, licensed under the GPL [4].

3.2. Carnetsoft

Carnetsoft was developed in Holland. Carnetsoft provides low-cost professional driver training and research simulators for driver education, driving schools and research institutes and universities [5].

3.3. City Car Driving

This simulator provides driver education in Russia and designed to help users experience car driving in a big city, the countryside and in different conditions or go just for a joy ride [6].

3.4. Traffic Talent

This simulator is designed like an online vehicle driving game. Players must complete goals in given time [7].

3.5. Ohio State University Driving Simulator

All setups feature RTI's SimCreator™ and SimVista™ software systems for controlling the system and creating driving scenarios, including a Motion Drive algorithm for motion base control and multi body dynamics editing software. A fully-instrumented vehicle cab mounted on a 6 degree-of-freedom MOOG 2000e motion base, with a 260° front-projection cylindrical edge-blended screen and three high-resolution projectors [8] [9].

4. Rendering

In this section, three important parts of the simulation are explained. These are the vehicle model, the city model and user interface. The vehicle model part has components like rigid body of vehicle, dynamics of vehicle model and some other important parameters. Under the settlement part, the creation of roads, junctions and intersections and autonomous vehicles routes are explained. The last part of rendering is the user interface. In this part, control and routing of players vehicles are introduced. PeTraLi system will be explained in next sections.

4.1. Vehicle Model

An open source vehicle model is selected in this project [10]. This vehicle model which meets requirements of simulation is also improved. In Unity3D, an object model has three different components. These components are rigid body, transform and collider of the object. To create a complex game objects, smaller pieces of model are combined. In this method, all the components of the game object are created separately and then these components are associated with each other. As an example of tire a sub-component of a vehicle, first the mechanical parts of tire are rendered then the step of coating the outer component. Combining this transform model of tire and rigid body of model introduced above generates game object. Programming of this object collision will complete the process. After creating game object, rigid body model of object and their features are made (Fig.2).

One of the sensors used in autonomous vehicles is the LIDAR (Laser Imaging Detection and Ranging), which is used to determine presence of other objects that exist within defined regions and their distance from the vehicle [11][12]. At this study, raycasts used to simulate LIDAR sensor (Fig. 3). There are only a few raycasts at front and side parts of autonomous vehicle. Thus, within the specified driverless vehicles are available to detect objects. Driverless vehicles with the help of the scenarios are developed. If there is an obstacle in front of the vehicle, autonomous vehicle slows and maneuvers around the obstacle. If the obstacle completely blocks a vehicle in its route, autonomous vehicle waits until obstacle disappears.

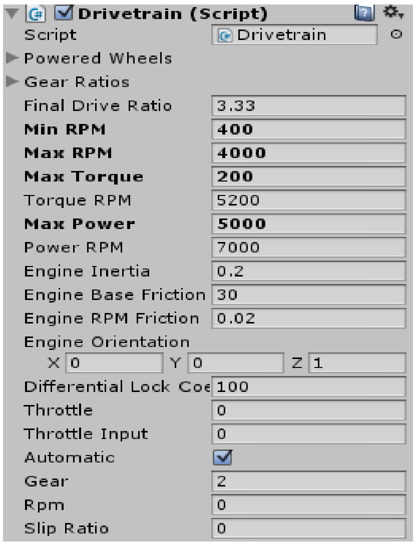

Fig. 2. Object Properties

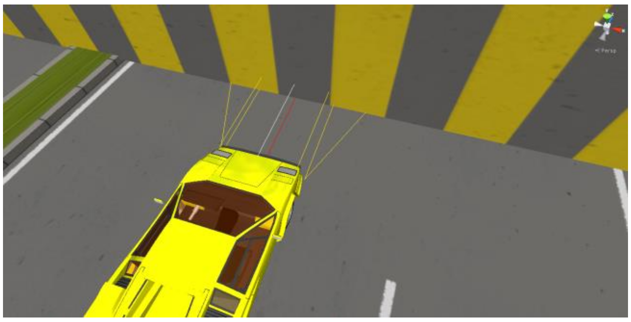

Fig. 3. Raycasts

4.2. City Model

A 3D city and road models are developed by using Esri CityEngine [13]. The plane, roads and other city objects are designed as the first step and then they are converted to the Unity3D environment by using for Esri CityEngine SDK (Fig. 4). The biggest problem is that recognition of road lines by an autonomous vehicle. Autonomous vehicle routes its way with using waypoints. We wrote a script for determining waypoints for a selected road segment (Fig.5). Although the waypoints are on center of the roads, autonomous vehicle may ignore some waypoint positions when turning or passing an obstacle. Assigning a mission to the autonomous vehicle is carried out by determining start and end point. A related program of simulator finds one of shortest paths with using Dijkstra's algorithm, and converts this path to a waypoint list. Same method is used for players' missions. In this case, player is directed with navigation signals.

Fig. 4. A City Object Created by Using Esri CityEngine

4.3. User Interface

PeTraLi offers a simple interface to the player. Game controls are kept simple because as intended for use in different platforms. Also, considering the mobile environment, there is usage of a single display. Therefore, instead of the screen filled with indicators, screen displays vehicles and roads as possible as. The situation of real-life driver looking to the road has been created. To keep players in their routes of objectives, the simulation gives an arrow indicator for navigation and two circles for traffic lights to the player (Fig.5).

Fig. 5. Waypoints

5. The PeTraLi System

Main purpose of the system is to inform driver for safety passing through an intersection which has no traffic lights. For example, if there is an approaching vehicle with high speed on your left side, the system will send a stop/warning signal to you. The PeTraLi system can be also use in intersections with traffic lights. But in this case, warning driver via lights by the device is obsolete so that driver has to concentrate original traffic lights. Proposed Personal Traffic Lights System consists of a roadside equipment that controls all the system, some sensors which are installed at each directions of the intersection and center of intersection, and a small electronic device in the each vehicle. PeTraLi is designed as a standalone system, and is not connected online to a traffic management system or is not being a part of ITS. There is no recording any information or vehicle plates for respecting driver's privacy, unless there is an accident. The electronic device in the vehicle is also a standalone unit and is used for carrying messages from the roadside equipment to driver with converting them into forms of light, sound and mechanical force provided by a smart seatbelt. So, there is no need any connection between device and the ECU of vehicle. The system monitors all vehicles regardless they have electronic devices or not. As a result, only vehicles with electronic devices get these messages. System has got lidars, cameras and radars for sensing vehicles within its zones (Fig. 6). System locates an incoming vehicle, recognizes from its plate, gives an identification, and monitors it until leaving zones.

There is only one-way I2V communication from PeTraLi roadside equipment to vehicles so that keeping the system simple and cheap as much as. There is a dedicated short-range communication which carries three types of encrypted messages from system to the devices installed in the vehicles. Each devices is loaded an unique identification related own vehicle's plate. The system broadcasts all the messages, whereas each device accepts only own labelled messages and general messages, and ignores all the others. The message types are: broadcasting general messages like emergency (example: all vehicle stop because there is an accident) and entering zone indication messages (heartbeat) to all vehicles, secondly messaging for recognizing and informing a particular vehicle, and lastly sending light information to a particular vehicle.

There are three warning methods to driver: yellow and red lights on the electronic device, sound tones generated by same device and a mechanical force as optional. There is a smart seatbelt for this purpose which inputs from the electronic device. This smart seatbelt vibrates its belt to warn driver about stop signal when the system senses any dangerous situations for this vehicle. There is no other sensing mechanism for observing driver's condition or mood, so vibrating occurs for each similar case.

Fig. 6. Zones of PeTraLi for a four-way intersection

6. Operation Steps of the System

The system sends message packages with labeled as broadcasting or identified which is obtained from vehicle's plate. The "entering zone" broadcast message is continuously send as a heartbeat. So, while a vehicle is entering such a zone, driver will be informed by an "entering zone" signal. Later on, the system recognizes vehicle's plate then sends "recognized" message. In this case, driver is also informed that he/she may get other types of signals. If there is no recognition signal, driver knows that there is a passing through without assistance. Yellow light signal indicates a controlled passing without any problem (Fig. 7), whereas red light signal indicates there is a problem so driver must be careful or stop directly. The system analyses trajectories of vehicles within zones and predicts if there are possible dangerous cases for preventing accidents. If there is a dangerous case, system will send messages to the related vehicles. In figure 8, black lines are shown for clarity, there are not included in simulation. When driver leaves the zones (means there is no heartbeat), all the signals will be off state again.

Fig. 7. A safety case

Fig. 8. A dangerous case

7. Conclusion

In this work, a driver assistance system was examined. To observe drivers' experiences for this design goal, a simulation for single user was implemented with using a chosen game engine. Simulation goals have been met and implemented. At the simulation side of this project, studies will be expanded for multi-user and especially in the complex scenarios. The study is aimed to be tested on the experimental groups and to examine results of testing.

8. References

[1] http://unity3d.com/ (Visited Date August 2015)

[2] National Highway Traffic Safety Administration, U.S. Department of Transportation, "Crash Factors in Intersection-Related Crashes: An On-Scene Perspective (NHTSA Technical Report DOT HS 811 366)", 2007.

[3] Şen, K. A., Nikbay K., Akgün, T, "Sürücüsüz araç trafiği ile sürücü davranışı iyileştirmeye yönelik bir oyun: OKANOM", ELECO 2014, Bursa, 2014. (In Turkish)

[4] http://sumo-sim.org/ (Visited Date August 2015)

[5] http://www.carnetsoft.com/ (Visited Date August 2015)

[6] http://citycardriving.com/ (Visited Date August 2015)

[7] http://www.unity3dgames.eu/traffic-talent.html (Visited Date August 2015)

[8] http://drivesim.osu.edu/about/ (Visited Date August 2015)

[9] http://citr.osu.edu/CrIS/?page_id=362 (Visited Date August 2015)

[10] https://www.assetstore.unity3d.com/en/#!/content/10 (Visited Date August 2015)

[11] http://www.sick.com/ (Visited Date August 2015)

[12] Akgün, T., Koç, Z., Güner, Ş., Öztürk, B., Özkan, B., Üstün, Ö., Tuncay, N., Özgüner, Ü., "A Study on Autonomous Vehicle Development Process at Okan University", 2012 IEEE International Conference on Vehicular Electronics and Safety, 2012, pp.369-374

[13] Özgüner, Ü. Acarman, T., Redmill, K., Autonomous Ground Vehicles, Artech House, USA, 2011