

## The development of a hardware- and software-based simulation platform for the training of driver candidates

Abdullah ÇAVUŞOĞLU<sup>1,\*</sup>, İsmail KURNAZ<sup>2</sup>

<sup>1</sup>Department of Computer Engineering, Faculty of Engineering and Natural Sciences,  
Yıldırım Beyazıt University, 06110 Ankara, Turkey

<sup>2</sup>Department of Computer Systems Education, Faculty of Technical Education, Karabük University,  
78050 Karabük, Turkey

Received: 02.08.2011 • Accepted: 25.09.2011 • Published Online: 27.12.2012 • Printed: 28.01.2013

**Abstract:** In this study, a traffic simulation system (TSS) to provide novice driver candidates with opportunities for training and testing has been developed. Autonomous vehicles that follow the flow of traffic in a microsimulation environment were prepared using a hierarchical concurrent state machine. Autonomous vehicles are skilled and have behavioral aspects such as tracking the lanes, following the vehicles, complying with traffic rules, decelerating, accelerating, and intersection crossings. These driver behaviors are parameterized to allow the autonomous agents to act while disregarding the rules (i.e. aggressive behaviors) or behaving normally. The TSS is operated in 3 different modes, namely the orientation, training, and testing modes. During the orientation phase, the candidate driver's target is to get used to the system and the vehicle, along with traffic rules in city conditions. During training, the driver is warned orally (when necessary) and given written handouts about the mistakes that occurred during the training session. The rules to be obeyed in traffic are kept in an XML file in the system and the data are used to follow the driver's behaviors. During the testing phase, an evaluation mechanism is employed. It observes whether the drivers obey or disregard the rules during the testing time interval. At the end of the test drive, a report showing the driver's mistakes is given. The reports and test results can be recorded in the database with the drivers' names for further analysis and evaluation processes. Driver candidates perform the driving using a steering wheel and pedals, along with a screen platform consisting of 3 monitors. With the TSS user interface, parameterized items such as vehicle selection, the crowdedness of the traffic, the timing of the traffic lights, and the determination of air conditions are possible. The system provides a cheap, affordable, and nonrisky platform for the trainees. Test results show that the system improves the drivers' skills and builds trust in them.

**Key words:** Traffic simulation system, agent modeling, scene graphs

### 1. Introduction

One of the problems of overpopulated, developed cities is the traffic problem. Among the constituents that form the traffic system, the drivers are the most significant elements. Training activities to increase driving abilities may contribute to the optimization of traffic security and solve traffic problems. Secure driving skills for all drivers in traffic are acquired at the end of the training process. One of the elements of secure driving skills is the ability to learn the traffic rules. When training the drivers, or in similar activities to be carried out in real conditions thought to be expensive or risky, providing a realistic simulation platform for driver training

\*Correspondence: [abdullah.cavusoglu@gmail.com](mailto:abdullah.cavusoglu@gmail.com)

may be the answer. Traffic simulation activities involving vehicles, pedestrians, and traffic signs have been a field of application and research studies since the 1950s [1,2] and many driving simulation platforms to serve this purpose have been developed [3].

Traffic accidents are one of the traffic problems that result in life and property loss. When the cause of these accidents is examined, the primary cause has been affiliated with driver behaviors and mistakes [4-6]. In addition, when researchers examined accidents around the world involving the loss of life and injuries, they found that young drivers were involved more often than middle-aged or older drivers [7]. Similarly, the accident rate of new driving license owners was higher than that of experienced drivers [8]. Moreover, a great percentage (80%) of traffic accidents generally takes place in city traffic [9]. Therefore, providing new driving license owners, in other words novice drivers or young drivers, with secure driving skills in city traffic may contribute towards decreasing traffic problems. Simulation systems built with this purpose provide drivers with live controlled conditions and help them to experience and acquire the habits of obeying the rules through real-time traffic simulations [10].

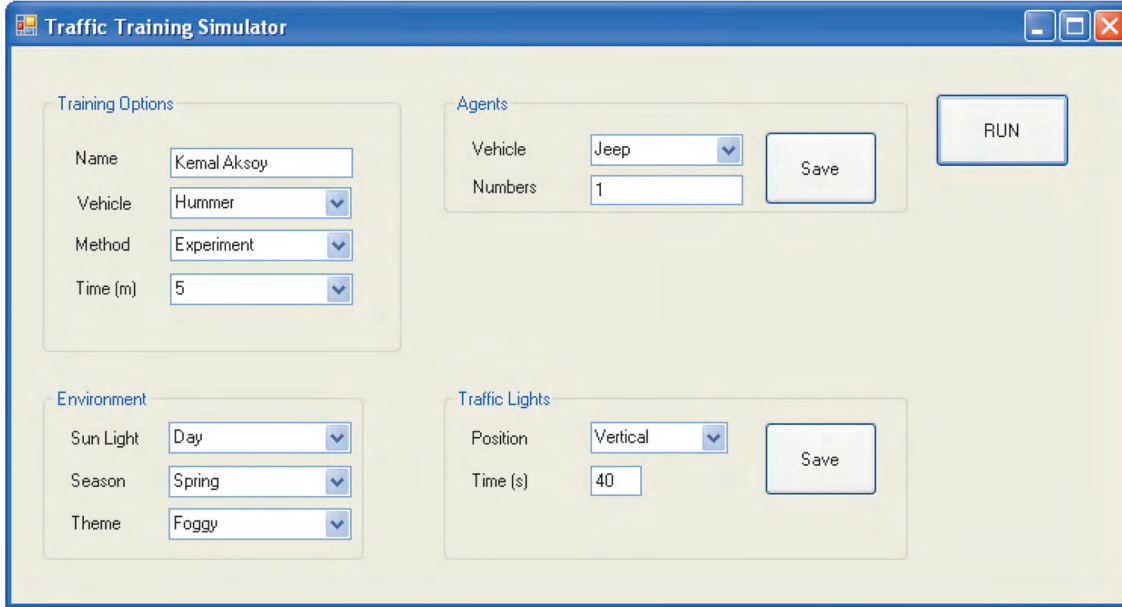
In a dynamic environment, we may find vehicles, pedestrians, traffic lights, etc. To develop a large-scale simulation system and to ensure an effective and realistic training environment, the other vehicles (i.e. factors in the traffic environment) must have a certain level of autonomy within the framework of certain rules and behaviors. This provides an original and realistic attitude, which in turn helps the system to demonstrate its quality. In microsimulation applications, for simulating the driver behaviors the cellular automata [11], hierarchical concurrent state machines [12,13], are used. Vehicle driving activity is an operational matter that needs to be taken care of with dynamic and complex theoretical aspects. In a traffic environment, balance is required to have the system in a constantly stable situation. This naturally requires a process of adjustment and adaptation [14]. For the sake of realism, with a moving vehicle, the acceleration and slowing down of operations must use a dynamic behavioral model, and the behavior of the power transmission components and tools, especially the use of gear boxes and the clutch mechanism, must be taken into account and used in the simulations [15].

In this study, we have developed a system to address this problem. One of the targets of our system is to provide realistic driving conditions. The Delta3D library [16], developed by the Modeling, Virtual Environments, and Simulation Institute at the Naval Postgraduate School in Monterey, California, is employed in our system for developing our simulation environment. Delta3D uses many well-known, open-coded libraries. In this study, the following applications are used: for monitoring operations, OpenSceneGraph [17]; for physics, the Open Dynamics Engine [18]; for character animation, the Character Animation Library [19]; for multichannel and 3D sound operations, OpenAL [20]; and for devices like the steering wheel and joystick operations, the Portable Game Library [21].

## 2. Working architecture

In this study, a platform designed to aid driver candidates in gaining safe driving skills in traffic is presented. The system may also be used as a gaming platform to educate the pupils about the traffic rules involved in driving. The platform also helps the users to become acquainted with the behavior of the cars under certain conditions. The traffic simulation system (TSS) was prepared using Visual C ++ 9.0 and Delta3D 2.4 [16]. When the system is started, the software interface expects the user to enter the necessary parameters. In Figure 1, the TSS entry screen is shown. In the screen, the name of the trainee, the type of the vehicle to be used

in the training, the method of the training (e.g., testing), the type and number of agents to be involved in the training session, the driving conditions including day or night driving and seasonal values, and traffic lights parameters are entered.



**Figure 1.** The TSS settings interface.

The system is operated in 3 different modes, namely the orientation, training, and testing modes. The candidate driver can do driving practice in the orientation mode, which is set up with a predetermined time with a chosen car, and the flow of traffic and other environmental conditions are also determined. When the TSS is in the training mode, the candidate has no time limit; in other words, the driver may use the system as long as he or she wishes to do so. When the system is operated in this mode, if an error occurs the user is warned orally every 5 s. The aim here is to help the driver in real-time about the errors that he or she is making during the driving session. For example, if the driver turns to the right at a junction with no turning signal, this is reported to the driver with an oral warning. In the training mode, at the end of the driving session, a written report is also produced.

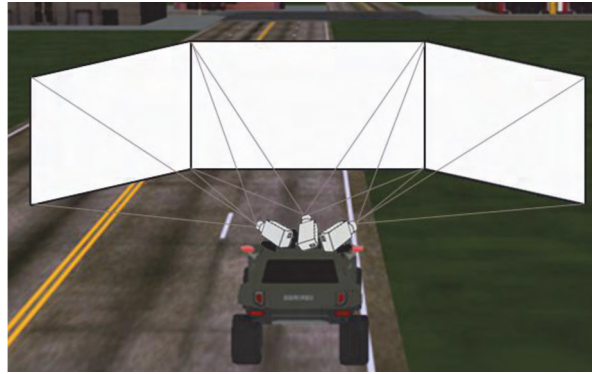
Another operating mode of the TSS is testing. In this mode, the target is to evaluate the behavior of the driver. Different durations and testing options in the system are possible (1, 2, 3, 5, 10, 15, and 30 min). In contrast to the training mode, in the testing mode, errors and oral feedback are not reflected immediately. However, for whatever the driver is doing in terms of mistakes during the session, a relevant error code is processed and saved in the background. When the test is complete, the driver can see the mistakes that he or she made during the test period. Each test drive is saved with the driver's name and the testing time in order to evaluate the performance and generate feedback information. In all 3 modes, a rule file containing the traffic rules is employed to evaluate and report behaviors back to the driver [22].

### 2.1. Setting up the traffic environment

Traffic environments are risky, involving subelements such as vehicles, pedestrians, traffic lights, and road markers. Giving a realistic simulation environment to the driver during training to prevent them from partaking

in risky activities in the real environment is always desirable. There are 2 different vehicle types in the TSS, which constitute the vehicle and the traffic environment for the driver candidates, including proxy tools. The control of the candidate driver's vehicle is selected by the candidates during the training mode. The vehicle is geared; during the motions of slowing down and accelerating, the vehicle's speed is achieved by changing the gear. The TSS provides for the slowing down of the vehicle by cutting off the power without using the brakes, which is defined as "engine brake". The driver performs the driving using the steering wheel and pedals.

The driver is able to see the environment ahead of the moving vehicle via three 3D synthetic cameras on top of the vehicle. As can be seen from Figure 2, a realistic view is produced on 3 screens positioned as in the Figure. For the display, 2 graphics cards (Quadro FX1800, GeForce 9500 GT) and 3 monitors are used.



**Figure 2.** The display platform of the TSS.

Safe driving skills such as autonomous lane tracking, vehicle tracking of proxy vehicles, and behaviors such as obeying the traffic rules are evaluated by the TSS [23]. Proxy vehicles moving in traffic autonomously may be categorized in terms of their behavior: some vehicles may be set as agents obeying the rules, some as agents disobeying the rules, and some as agents partially obeying the rules.

## 2.2. The XML rule file

Which rules a vehicle is obeying or disobeying depends on the location at which the driving occurs. The vehicle is responsible and needs to obey the traffic signs and rules depending on its location. The 3D virtual city used in the TSS is produced [24] using an OpenFlight [25] 3D model. By analyzing all of the roads in the city, intersections and traffic rules are evaluated and the vehicles are made to behave accordingly. City highways are divided into roads, and the roads are divided into segments covering lanes with the starting and ending position information (i.e. the x, y, and z axes), which are then recorded in an XML file [23]. The city roads are connected to each other by junctions. The last segment of each road is connected to the initial segment of a different road at the junction. The location information of a vehicle is kept in a file called 'intersections.xml'. While advancing toward a junction, the relevant information (presented in Figure 3) kept in the file is used.

There are rules to be obeyed by the candidate driver while moving in the virtual city, as in real life. The rules can be classified as visual and nonvisual rules. Visual traffic rules contain previously reported limitations in the form of alerts, such as traffic signs and traffic lights. Nonvisual traffic rules are the rules that drivers should know and obey without any warning. For example, drivers should know the maximum speed limit on urban roads (which is 50 km/h) and should not exceed the limit on such roads. The behaviors of the drivers against the traffic rules in the TSS are monitored using a rule set including these 2 types of rules. The information for

visual and nonvisual traffic rules at each segment (i.e. regarding the traffic signs and traffic lights) are processed into a file (such as trafficrules.xml). Table 1 contains the elements used in the rule file.

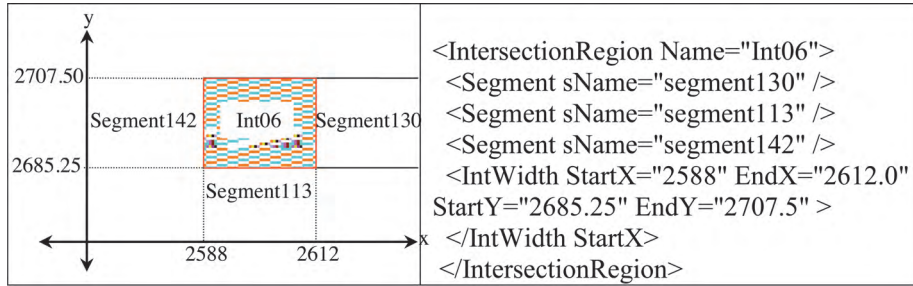




Figure 3. Converting the intersection data from the model to XML codes.

Table 1. The trafficrules.xml file, showing the aspects of the components.

Element	Root	Value	Traffic sign/comment	Basic description	
Segment	Rule		Traffic rules		
sName	Segment	"Segment100"	Segment name		
Turn	Segment	"3"		Value	Comment
				0	Free
				1	No left turn + no U-turn
				2	No right turn + no U-turn
				3	No left turn
				4	No right turn
TRL	Segment	"1"	There is a traffic light.	Value	Comment
				0	There is no traffic light.
				1	There is a traffic light.
TLNumber	TRL	"0"	Traffic light number is "0"	System consists of 8 traffic lights.	
TLDirection	TRL	"1"	It is interested in direction "1"		
Velocity	Segment	"1"	30 km/h speed limit 	Value	Comment
				0	Free
				1	30 km/h speed limit
				2	50 km/h speed limit
				3	70 km/h speed limit
				4	90 km/h speed limit
Parking	Segment	"1"	No parking	Value	Comment
				0	Free
				1	No parking
				2	No stopping

### 2.3. The control of the traffic rules

The TSS controls the status of the driving candidates at their segments. The trafficrules.xml file contains the data belonging to the whole city. For the control of the driver candidate, traffic rules affecting the relevant segment are required. A data structure was formed to hold the rules of the related segment. The vehicle that the trainee uses is evaluated in terms of the values in this data structure at the related segment. The rule data set is renewed when changing the current segment by motion. The 'segment shift' of the vehicle is determined

with the calculated driving finish point according to the main coordinate axis and moving direction parameters of the vehicle. The driving target point of a vehicle moving towards the (x+ or y+) axis direction or the (x- and y-) direction is calculated from the initial point of the related segment. When a vehicle ‘shifts a segment’ or enters a junction point, the LoadSegmentRules() function is executed. This function is a set of rules that requires monitoring at the current segment and finds the end point of the driven vehicle. Figure 4 shows the LoadSegmentRules() routine and the pseudocode for the function.

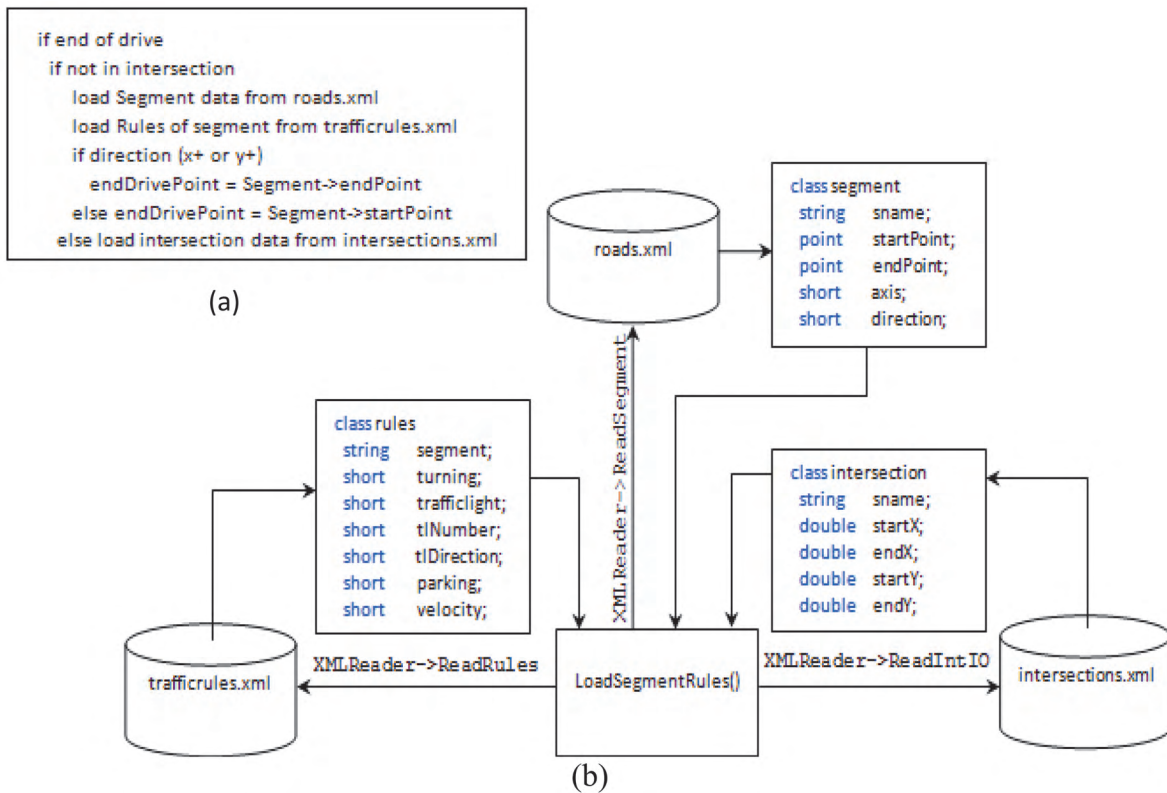


Figure 4. a) The pseudocode and b) the block diagram of the LoadSegmentRules() function.

### 2.3.1. Nonvisual traffic rules

Nonvisual traffic rules in the TSS are city and intercity speed limits, signaling behaviors at lane changing, or turning or using the roads properly on 2-way roads. The control of these rules in the TSS is implemented as follows.

1. Speed limits, either in the city or on intercity roads, are kept in the trafficrules.xml file as shown in Table 1. The speed values corresponding to each segment are found here. The control process is carried out using the same method as with the visual traffic rules.
2. Driver behaviors, such as signaling and moving along the roads, are controlled by the dedicated functions.

In the case of violations detected while controlling both the visual and nonvisual sets of traffic rules, a specific error code is generated for each false activity. In addition, this code and its description are saved.



### 2.3.2. Turning

With traffic signs forcing the drivers to turn right, turn left, or make a U-turn, a turning control for the vehicle is performed in the case where the turning value is different from 0. With traffic signs forcing the driver into right/left turning at junctions, a control procedure to catch the possible mistakes is executed. For this purpose, the movement direction of the vehicle, the lane, and the orientations of the vehicle are used. The vehicle direction information consists of  $(\{x, y\}, \{+, -\})$  sets. For example, in the case where a vehicle moving in a  $(y+)$  direction turns left at a segment with left turns prohibited, the direction of the vehicle turns to  $(-1, 0)$  from  $(1, 0)$  and an error occurs. When the vehicle moves in the other directions, no error occurs, because the direction values are valid. Table 2 illustrates the representations of the directions.

**Table 2.** Vehicle directions.

Moving direction	X	Y
x+	1	0
x-	-1	0
y+	0	1
y-	0	-1

In the case of U-turns, a different method is used. In this method, the source segment of the car and the target segment information are compared. When the target segment is reached, and a previously recorded value of the segment that has just been left is the same as the current segment, then this means that the vehicle has made a U-turn.

### 2.3.3. Speed limits

The speed limit control is performed according to the velocity values reported in Table 1 at the TSS speed limit segments. If the speed limit of the vehicle exceeds the preset value, then the error procedure is executed. The tolerance value determined by the local authorities is 10% of the allowed limit. Therefore, for example, in the case of a 50 km/h speed limit segment, if the vehicle exceeds 55 km/h, a speeding error takes place. Regardless of the direction of a vehicle at the segment, the speed limit value is the same.

```

if(rules->velocity!=0)
  find the speed limit for segment (SVS)
  if(vehicle->speed>svs*1.1)
    errCode=SVS+"RED LIGHT"
    Add the Error to List with errcode
    
```

**Figure 5.** The pseudocodes of traffic light control.

### 2.3.4. Traffic lights

One of the traffic rules that drivers have to obey in the system is the traffic lights. Normally, a car may only be mobile when the green light is on. The other 2 states force the cars to stop or wait. When a vehicle arrives at a segment with traffic lights, the driver’s behavior is monitored according to the traffic light status. If the vehicle moves when the red or yellow lights are on, this is recorded as a mistake. In the system, traffic lights are controlled by a finite state machine, which is responsible for the synchronization of the lights [23]. Traffic lights are the elements that control the traffic at the junctions. For this reason, the control process is carried out when the vehicle enters a junction. There are many traffic lights in the system. Which traffic light controls

which roads depends on the vehicle’s progress direction. For this reason, in the data structure formed for the traffic lights, the rules and the directions are needed. In Figure 6, the pseudocodes for the yellow and green lights are presented.

<pre> if(rules-&gt;trafficlight==1)   if(rules-&gt;tDirection==true)     if(trafficlight[rules-&gt;tNumber]=='RED')       if(inIntersection)         errCode="RED LIGHT"         Add the Error to List with errcode                     </pre>	<pre> if(rules-&gt;trafficlight==1)   if(rules-&gt;tDirection==true)     if(trafficlight[rules-&gt;tNumber]=='AMBER')       if(inIntersection)         errCode="AMBER LIGHT"         Add the Error to List with errcode                     </pre>
--	--

**Figure 6.** The pseudocodes for the traffic lights.

### 2.3.5. Parking or stopping in the lanes

The stopping behaviors that affect the flow of traffic are also controlled by the TSS. Under normal circumstances the driver is not expected to stop unexpectedly in the lanes. At the segments where stopping is not allowed, a speed check is carried out (the speed should be 0 km/h); if this check produces an abnormal result, an error is immediately produced. If a vehicle stands still for more than 5 min, then an error message is generated. In Figure 7, the pseudocodes representing these conditions are shown.

<pre> if(rules-&gt;parking==1)   if(vehicle-&gt;velocity&gt;0)     start timer(parking)     calculate elapsed time     if(elapsed time&gt;5 m)       errCode="PARKING"       Add the Error to List with errcode                     </pre>	<pre> if(rules-&gt;parking==2)   if(vehicle-&gt;velocity&gt;0)     errCode="STOPPING"     Add the Error to List with errcode                     </pre>
--	---

**Figure 7.** The pseudocodes for parking.

### 2.3.6. Changing lanes

Drivers should signal when changing lanes or at the junctions before turning right or left, to warn other drivers about the maneuvers. One of the habits that the driver candidates should acquire is signaling. The drivers are monitored for this as well. If their signaling is not proper, a relevant error message is produced. For the lane change control of the system, the information of the present lane and of the lane to which the vehicle is moving is checked.

## 2.4. Behavior types of virtual vehicles according to traffic rules

It is impossible to say that all drivers obey the traffic rules in their daily lives. The most important cause of traffic confusion is the behaviors of the drivers, which can be generalized as some drivers obeying the rules while others do not. To realistically exhibit these types of behaviors, virtual agents representing such drivers are added to the TSS system. The TSS virtual agents behave either normally or aggressively. The degree of aggression is also parameterized. A virtual agent with “normal” behaviors is expected to behave according to the written traffic rules at all times. For example, they follow the speed limits, they keep a safe following distance, etc. An agent with a certain level of “aggression” is not expected to follow the traffic rules. Some of the well-known actions (driver patterns) that take place in Turkey’s city traffic are simulated within the system.



In the TSS, an artificial intelligence module is employed to control the behaviors according to the features when predetermined characteristics are used at agent formation. The behaviors of the virtual agents in the simulation environment with respect to the traffic rules are shown in Table 3.

**Table 3.** Behaviors of virtual agents against the traffic rules.

Rule	Normal	Aggressive	Outside of the rules
Speed limit	Obey all speed limits!	Don't obey nonvisual speed limits!	Don't obey speed limits!
Amber light	Stop!	If on the move, keep moving. If waiting, start to move!	Keep moving!
Red light	Stop!	Stop!	Keep moving!
Safe following distance	Sum of the deceleration distance and security distance	Half of the sum of the deceleration distance and security distance	Half of the sum of the deceleration distance and security distance
Vehicle speed approaching the junction	Decrease!	Decrease at intersections controlled by traffic lights, keep moving at the others!	Keep moving!
Visual traffic rules (no parking, no left/right/U-turns, etc.)	Obey rules!	Obey rules!	Don't obey rules!

### 2.5. Messaging system

In order to teach the traffic rules to the driver candidates during the training, a messaging system is employed. As was mentioned above, each violation produces a relevant violation/error code. These messages are predetermined and recorded voice messages used to warn the user. A sample list of the error codes is shown in Table 4.

A written error message is displayed on the screen for a period of 5 s during the training sessions. However, they are not displayed in the case of test driving, but a test report is produced at the end of the session. In the case where audio messaging is enabled, the same activities are repeated for this type of messaging. In other words, if the candidate makes an error, he or she is warned immediately, except in the case of testing. The voice message is played once in the case of each error. To administer voicemails, the OpenAL [20] application programming interface (API) was used. The OpenAL is a multichannel, 3D positional audio effect used to run a sound API that is used in many Windows games. The names of the audio files corresponding to the error codes are written into a file called errorfiles.dat. In the case of an error, the CreateAudioMessage() function is called with the relevant parameter.

**Table 4.** Error codes and voice messages.

Error code	Message	Error code	Message
1	Right turn error	9	50 km/h speed limit error
2	Left turn error	10	70 km/h speed limit error
3	U-turn error	11	90 km/h speed limit error
4	Red light error	12	120 km/h speed limit error
5	Amber light error	13	Left signal error
6	Parking error	14	Right signal error
7	Stopping error	15	Signal error
8	30 km/h speed limit error	16	Wrong way error

### 3. Experimental results

The TSS platform was used in the training sessions for a group of 24 people in Ankara in a private driving school. First, a test drive was carried out with the participation of all of the candidates and the results were recorded. Next, the candidates were asked to use the system in training mode with different types of vehicles. Afterwards, the candidates were asked to take the test again and their performances were recorded. The drivers were given an evaluation sheet about the TSS. The results of the evaluations are represented in Figure 8.

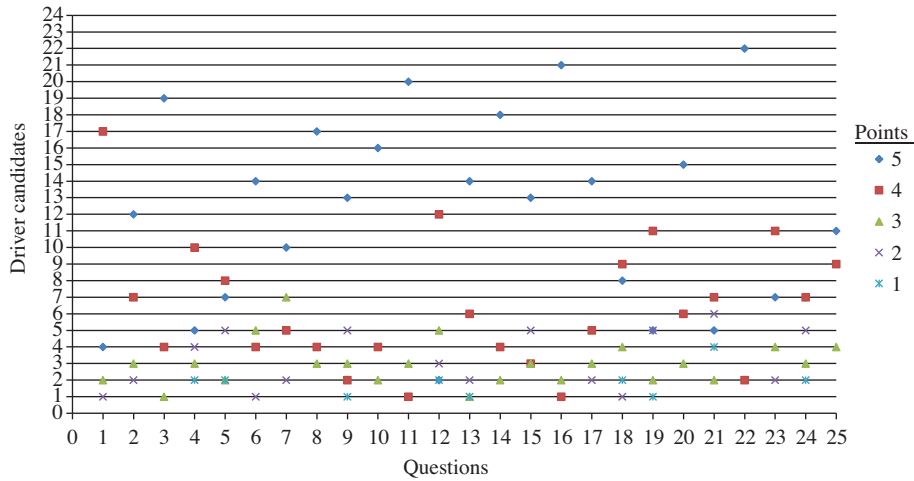


Figure 8. The results of the test assessment questionnaires.

As the results suggest, the users gave 4 and 5 points to the TSS for the adequacy of training in simulated traffic, with a rate of 63%–88% in terms of capability of training and realism. The candidates believe that voice messages provide more effective learning than written messages. On the issue of realism in the case of driving maneuvers, their responses correspond to 4–5 points, while 30% gave the score of 1–2. When the candidates were asked to answer regarding the effectiveness of individual items in the TSS, 100% was given for the traffic lights and 88% was given for the traffic signs.

The candidates complained about the virtual agents that show the same behavior all of the time while obeying the written traffic rules, as well as other driving skills like lane tracking in the written comments, and they stressed that the realism of the system must be improved. According to the test assessments of the drivers, their capabilities of understanding and obeying the traffic rules have been improved when compared to their performances before the training sessions.

### 4. Conclusion

In this study, a simulation platform utilizing both hardware and software components has been developed to provide a cheap and easily accessible medium for new driver candidates who wish to experiment safely with this simulated environment. The system may also be regarded as a gaming tool, teaching young people the traffic rules as well as aiding driver candidates who wish to improve both their driving skills and traffic knowledge. Several driving skills, such as speeding up, slowing down, lane changing, and traffic light behavior, can be acquired by using and experimenting with the system. Since the system is cheap and affordable (a standard PC with a relevant graphics card is enough), it is more attractive than any other system that is commercially available. In the modeling environment, the behavior of different types of driver agents (e.g., normal, aggressive)

helps to create a more realistic traffic environment. However, the observation and feedback that we received from the users show that this is still not enough and more parameters are needed in this respect. The modes in which the system operates differ in terms of the expectations of the user from the system. The users are provided with facilities such as 3D orientation, training, and testing modes. Feedback messages are either reported in text format or as voice messages. The evaluations of the drivers are saved and kept in the system to be used afterwards. We suggest that this system could be improved and could become more attractive to the novice user if the animations and the behavioral aspects of the virtual agents are improved.

### References

- [1] T. Hirata, Development of Driving Simulation System: MOVIC-T4 and its Application to Traffic Safety Analysis in Underground Urban Express Ways, PhD dissertation, Tokyo Institute of Technology, Department of Civil Engineering, Tokyo, pp. 2–3, 20–22, 157–160, 2005.
- [2] S. Das, B.A. Bowles, C.R. Houghland, S.J. Hunn, Y. Zhang, “Microscopic simulations of freeway traffic flow”, Thirty-Second Annual Simulation Symposium, p. 79, 1999.
- [3] Y. Miao, U. Hoppe, N. Pinkwart, “Naughty agents can be helpful: training drivers to handle dangerous situations in virtual reality”, Proceedings of the 6th IEEE Computer Society International Conference on Advanced Learning Technologies, pp. 735–739, 2006.
- [4] N. Sümer, “Trafik kazalarında sosyal psikolojik etmenler: sürücü davranışları, becerileri ve sosyal politik çevre”, Türk Psikoloji Yazıları, Vol. 5, pp. 1–36, 2002.
- [5] Y. Yasak, “Trafik psikolojisi”, TŞOF-Psikoteknik: Sürücü Değerlendirme Eğitim ve Araştırma Merkezi, pp. 3–11, 2002.
- [6] L. Evans, Traffic Safety, Bloomfield Hills, Michigan, Science Serving Society, 2004.
- [7] M. Demir, Trafik Eğitimi İçin Zeki Etmenler Geliştirme, PhD dissertation, Gazi University Institute of Information Technology, 2008.
- [8] N. Sümer, A.B. Ünal, “Acemi sürücülerde tehlike algısibecerisi”, Üçüncü Trafik ve Yol Güvenliği Ulusal Kongresi, pp. 177–182, 2005.
- [9] H. Ece, “Trafik kazaları özeti”, T.C. Bayındırlık ve İskan Bakanlığı Karayolları Genel Müdürlüğü, pp. 3–5, 2003.
- [10] N. Sümer, A.B. Ünal, A. Birdal, “Assessment of hazard perception latencies using real life and animated traffic hazards: comparison of novice and experienced drivers”, Proceedings of the 4th International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, pp. 488–494, 2007.
- [11] T. Sun, J.F. Wang, “A traffic cellular automata model based on road network grids and its spatial and temporal resolution’s influences on simulation”, Simulation Modelling Practice and Theory, Vol. 15, pp. 864–878, 2007.
- [12] J. Cremer, J. Kearney, Y.E. Papelis, R.A. Romano, “The software architecture for scenario control in Iowa driving simulator”, Proceedings of the 4th Computer Generated Forces and Behavioral Representation Conference, pp. 523–532, 1993.
- [13] A. Okutanoğlu, V. İşler, “Simülasyon sistemlerinde esnek senaryo altyapıları”, Birinci Ulusal Savunma Uygulamaları Modelleme ve Simülasyon Konferansı, pp. 115–126, 2005.
- [14] J. Archer, I. Kosonen, “The potential of micro-simulation modelling in relation to traffic safety assessment”, Proceedings of the ESS Conference, pp. 427–443, 2000.
- [15] A. Laschet, “Computer simulation of vibrations in vehicle powertrains considering nonlinear effects in clutches and manual transmissions”, Society of Automotive Engineers Technical Paper 941011, 1994.
- [16] Delta3D Open Source Gaming & Simulation Engine, September 2010, <http://www.delta3d.org/>.
- [17] OpenSceneGraph, September 2010, <http://www.openscenegraph.org/>.

- [18] Open Dynamics Engine, September 2010, <http://www.ode.org/>.
- [19] Character Animation Library, September 2010, <http://gna.org/projects/cal3d/>.
- [20] OpenAL, September 2010, <http://connect.creativelabs.com/openal/default.aspx>.
- [21] Portable Game Library, February 2011, <http://plib.sourceforge.net/>.
- [22] KarayollarıTrafik Yönetmeliği, 18.07.1997, <http://www.traport.com.tr/doc/trafikyonetmeliği.pdf>.
- [23] A. Çavuşoğlu, İ. Kurnaz, “Developing a parameterized simulation platform with intelligent synthetic agents for training driver candidates”, *Procedia Computer Science*, Vol. 3, pp. 727–731, 2011.
- [24] H.H. Göktaş, A. Çavuşoğlu, B. Şen, “AUTOCITY: A system for generating 3D virtual cities for simulation systems on GIS map”, *Intelligent Automation and Soft Computing*, Vol. 15, pp. 29–39, 2009.
- [25] G. Chen, S. Xiang, G.Q. Ji, Q. Ding, “Scene simulation platform based on data fusion of multiple format 3D models”, *IEEE International Conference on Computer Modeling and Simulation*, pp. 342–346, 2009.

**Appendix**

**Questionnaire for candidate drivers for the TSS**

Name :

Age :

Number	Question	Points				
		5	4	3	2	1
1	The length of the driving circuit is sufficient.					
2	The text feedback messages helped me learn the traffic rules.					
3	The voice feedback messages make it easy to learn the traffic rules.					
4	Traffic signs are sufficiently visual.					
5	The driving maneuvers are closely realistic.					
6	The TSS helped me to learn traffic rules easily.					
7	The test period was enough for my self-assessment.					
8	I have learned the meaning of the turning signs (right, left, etc.).					
9	I have learned the meaning of the parking signs.					
10	I have learned the meaning of the speed limit signs.					
11	I have learned city and intercity speed limits.					
12	I have gained the habit of signaling.					
13	I have learned the behavior of the driver to reduce speed.					
14	I have felt the diversity of driving different types of vehicles.					
15	The traffic signs show the curved paths.					
16	I have learned the necessity of using the vehicle speed to adjust the clutch and gears.					
17	The error report helped me to correct my mistakes at the end of the test drive.					
18	The lane following behavior of the agents is realistic.					
19	The vehicle following behavior of the agents is realistic.					
20	The agents obey the traffic signs.					
21	The behavior of the agents at intersections is realistic.					
22	The agents obey the traffic lights.					
23	Different types of drivers have increased the realism of the TSS.					
24	The display platform has increased the realism of the TSS.					
25	The steering wheel and pedals have increased the realism of the TSS.					
<b>Comments:</b>						

Copyright of Turkish Journal of Electrical Engineering & Computer Sciences is the property of Scientific and Technical Research Council of Turkey and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.