DOI: 10.15439/2017F310
ISSN 2300-5963 ACSIS, Vol. 12

Virtual Tour for Smart Home Developed in Unity Engine and Connected with Arduino

Erich Stark, Erik Kučera and Oto Haffner
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology in Bratislava
Bratislava, Slovakia
Email: erich.stark@stuba.sk, erik.kucera@stuba.sk

***Abstract*—Nowadays, virtual tours are very popular and many people would like to see a virtual house before the acquisition of the real one. The paper demonstrates a creation of a virtual tour for smart home developed in Unity 3D engine. This virtual tour is connected with Arduino microcontroller which has attached several sensors and actuators. These electronic devices react to the events in the virtual tour and vice versa.**

I. Introduction

MODERN forms of visualisation are now realized on the basis of the development of new ICT technologies (e.g. interactive applications made in 3D engine [1], virtual reality or mixed reality). Visualisation of process modelling, identification and control of complex mechatronic systems, elements and drives using virtual and mixed reality allows students to get a much better and quicker understanding of the studied subject compared to conventional teaching methods.

Nowadays, there is a trend of using interactive 3D applications and virtual reality in virtual tours for houses, cars and other products. Also, many interactive 3D applications for education are being developed.

Toyota offers modern virtual showroom [2] for their customers. This showroom was developed using Unreal Engine.

There are also interactive applications from Animech Technologies. This company offers many education modules like Virtual Car, Virtual Truck or Virtual Gearbox [3]. Using these applications students can understand the functioning of mentioned devices and they can look into their interior and detach their individual components in detail.

Very interesting project is a virtual clinic [4]. This project is supported by the University of Miami or Charles R. Drew University of Medicine and Science in Los Angeles. This interactive application offers an insight into the actual functioning of a larger clinic, and they can also try to diagnose patients. Students are thus trained through a real experience with the health system, but this complex system is modelled and simulated in virtual reality.

An absolute novelty is Microsoft HoloLens [5], the arrival of which has led to the emergence of a completely new segment of mixed reality. Mixed reality has unquestionable advantages over virtual reality, as the user perceives a real world and also a virtual world in the same time. The use of this feature is in practice undisputed and it is assumed that mixed reality will become a new standard in many areas such as education, marketing, modeling of complex mechatronic systems, etc.

Fig. 1. Microsoft HoloLens - mixed reality application (Volvo) [6]

For Microsoft HoloLens there are more education and virtual tour applications.

Application HoloTour [7] provides 360-degree spatial video of historical places like Rome or Peru. The application complements 3D models of important landmarks that have not been retained or supplementary holographic information about elements in the scene.

II. Main aspects of proposed application

This paper describes an interactive 3D application that simulates virtual tour of the smart house and its exterior. The application is implemented in Unity engine. As it is the interactive application that responds to the perceptions and changes from the environment, it is necessary to connect it with external hardware which captures the signals from the environment and sends the data to the application. As the best candidate to solve this problem, Arduino family microcontroller has been chosen. Arduino will be connected to the computer via the USB port and connection will be established through the serial port. Through this port, the data from sensors will be sent to the application. It is important to note that communication will not run in only one direction (from Arduino to the computer) but also from the computer to Arduino. So it is possible to control actuators connected to Arduino.

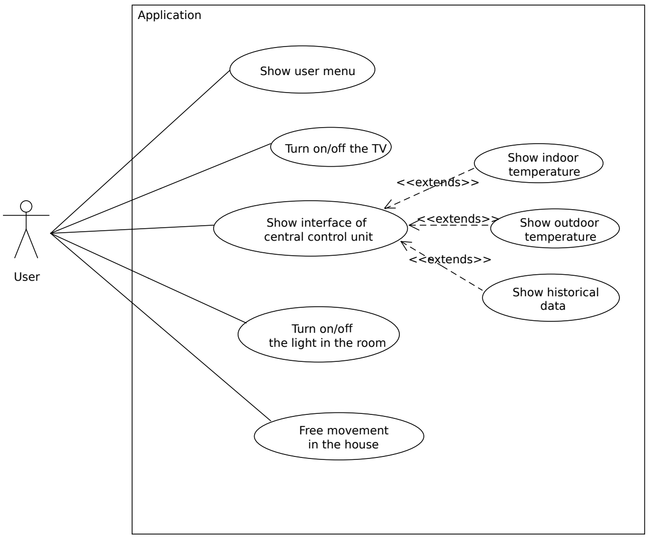


Fig. 2. Use-case diagram

The proposed application has its own data storage. This storage can be used for statistical evaluations or retrieval of historical data. The database was created using cloud service Microsoft Azure [8].

The application must meet these functional requirements:

- option to move in the house and in outdoor areas.
- ability to view the current temperature
- user menu and ability to set COM port for Arduino microcontroller
- ability to turn on/off light in rooms by loud sounds like clapping
- fan rotation on room ceilings when temperature is higher than a certain value
- triggering of fire alarm when detecting the presence of fire in a real environment
- stretching the curtains in the living room in low light conditions and vice versa
- option turn on/off a TV using IR controller when the user is at a sufficient distance from the TV
- option to view historical data about indoor and outdoor temperature
- alerting the user of the unfavourable state of the application

A use-case diagram is in Fig. 2.

III. Sensors and Actuators

The application is based on a number of the necessary sensors and actuators connected to the microcontroller:

- fire sensor
- sound sensor
- light sensor
- temperature sensor
- IR receiver

These sensors will be mapped in the application for a certain functionality. Also, few actuators will be used:

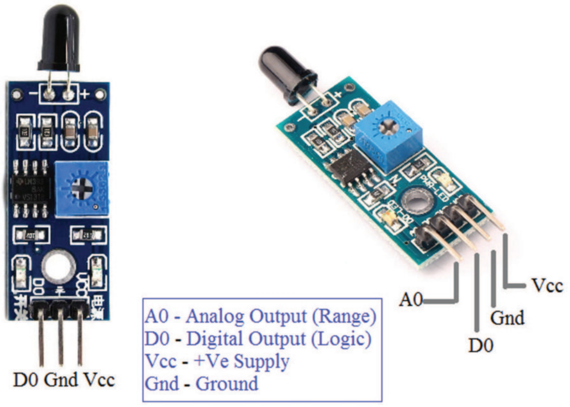


Fig. 3. Fire sensor

Fig. 4. Sound sensor

- LED diode
- buzzer

A. Fire sensor

One of the basic sensors of the proposed system is a fire sensor (Fig. 3) that detects the presence of a flame. In principle, it is a detection of infrared light with a wavelength in the range of 760 to 1100 nm. Its core parts include an infrared sensor, a potentiometer, an operational circuit amplifier and a LED. There are different types of these sensors, but two most well-known are three-pin and four-pin sensors. Four-pin sensors have one pin for the analog connection.

B. Sound sensor

The sound sensor (Fig. 4) is a small board with a microphone that enables sound detecting from the environment. By connecting to the analog pin, it is possible to detect the intensity of the incoming sound.

C. Light sensor

Light sensor (Fig. 5) is also called a photoelectric sensor because it converts light energy into electrical signals. The more light it gets on the surface of the light-sensitive part, the resistance decreases. Normal value ranges from 8 to $20k\Omega$.

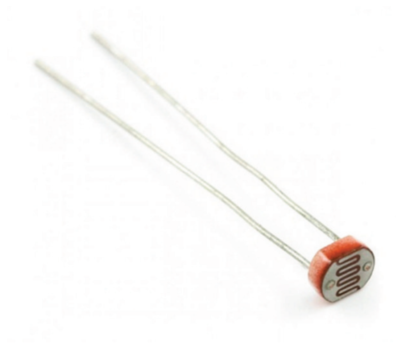


Fig. 5. Light sensor

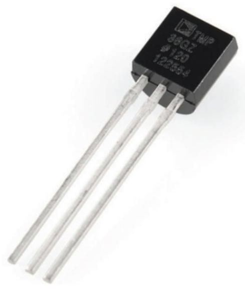


Fig. 6. Temperature sensor

D. Temperature sensor

Temperature sensor (Fig. 6) of type TMP36 is used. It is a low-voltage thermal sensor that provides a voltage output that is proportional to the sensed temperature. This device is also very easy to use and requires no external calibration. The temperature range is from $-40°C$ to $+125°C$ is a very decent result for such a small and simple device, although the accuracy of the measured values may have a deviation of up to $2°C$.

E. IR receiver

The last important sensor in proposed project is the infrared receiver (Fig. 7). It has also a build-in infrared transmitter but it is not used. As the infrared transmitter, a modern smartphone can be used.

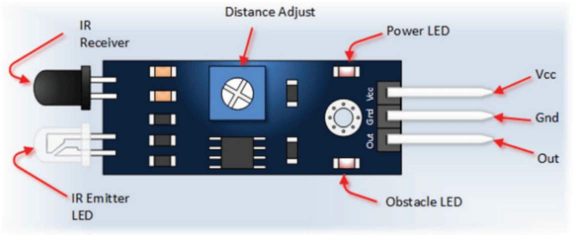


Fig. 7. IR receiver

TABLE I
IDENTIFIERS AND POSSIBLE VALUES FOR SENSORS

Type of sensor	Identifier	Measured value
Light sensor	light	from 0 to 1024
Temperature sensor	temperature	from -40 to 120
Sound sensor	sound	from 0 to 2014
Fire sensor	flame	fire / calm
IR receiver	ir	signal

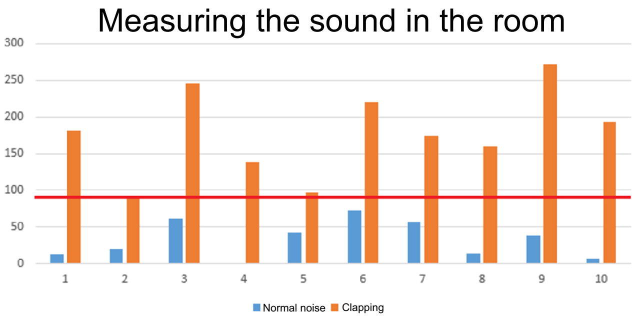


Fig. 8. Measuring the sound in the room

IV. IMPLEMENTATION OF THE APPLICATION

A. Processing data from sensors

As it was mentioned, the data from sensors will come from Arduino in a continuous stream. It is, therefore, important to determine the data format so that we can easily recognize the sensor and what value the sensor has captured. On the Arduino side, an infinitely uninterrupted cycle will take place, and on the Unity side, the C # programming language will provide parsing functions, which will process the information and perform the necessary functionality. The data format is: `{sensor}+"_"+{type_of_sensor}+"_"+{measured_value}`

The first part is a characteristic string that will let us know that there is some data from the sensors. It is important to start with a particular string because if we also have other data from other sources in the application that we would like to send through our application, it might happen that we are simply mixing the data. This is a situation we are, of course, trying to avoid. The second part will be a unique identifier for individual sensors connected to Arduino. The last part will be the measured value we get from the connected sensors. See details in Table I.

A very important part is the definition of boundary values captured on sensors when a system will perform a certain function corresponding to the measured values. On of these values is the volume for the sound sensor which will allow the system to turn on or off the lights in the room. It is important to set this value sufficiently sensitive to clapping near the sensor but at the same time high enough to filter out any ambient noise. Few test has been made (Fig. 8) and it was found that good boundary value would be 90.

Another boundary value in the system is the value of light in the room. If we capture values of light under the certain value

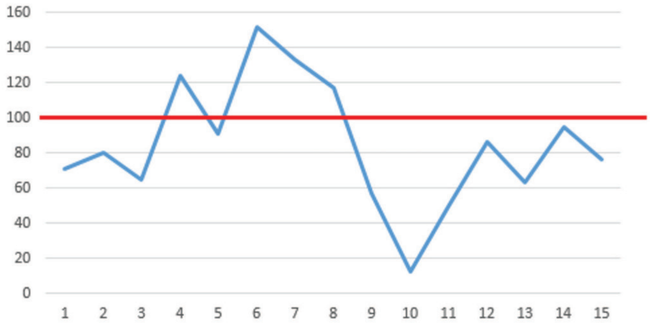


Fig. 9. Measuring the light value

TABLE II
PROPOSED API FOR MYSQL DATABASE

URI	HTTP method	Response data type	Description
uri_of_server/all	GET	JSON	all records
uri_of_server/{id}	GET	JSON	One record with id
uri_of_server/add	POST	boolean	Positive/negative answer according to the success of the operation

(boundary value) the curtains on the ground floor will spread out. To determine the boundary value it was necessary to make several measurements (Fig. 9) in different light conditions. The chosen boundary value is 100.

The last boundary value is the temperature that was set to $27°C$.

B. Classes and data

Class diagram can be seen in Fig. 10.

Used MySQL database is closely linked to the API that provides the interface between the database and the application. Our API carries the RESTful service characteristics but we can not fully say that this endpoint meets all REST properties like all CRUD operations but at least it has the proper use of HTTP methods. For our needs, it is not necessary to use all CRUD operations. We have defined three unique URIs through which it is possible to access database data (Table II).

C. Menu and interface

The interface of the application should be simple and understandable. For the proposed application, it is best to use the first person view as it offers the most realistic experience. It is possible to rotate with the mouse and use arrows for basic movement. For future purposes, such control is easy to map in virtual reality. That is why we decided to use a point at the center of the screen instead of a mouse control. Using *Escape* the user turn on/off the main menu (Fig. 11). Another important menu is GUI (12) for temperature inspecting. This menu shows when the user is close to the central control panel in the virtual smart house.

The exterior of the home has been designed to match the overall home visualisation to create the atmosphere of a luxury smart house with all the equipment from the collection of cars, a swimming pool and trees.

In the interior there are many interactive points that interact with the user in a certain way. These are televisions, lighting, fans, or curtains.

In Fig. 13, there is a living room of the presented smart house. It is possible to see many interactive elements. The first one is a television that can be turned on and off by an external controller. The second one is the curtains that pull and stretch automatically depending on the intensity of the light in the home (i.e. light sensor connected to Arduino) and the third one is the ceiling fans that are spinning at an excessive interior temperature.

Fig. 14 shows the light in the house that can by controlled by clapping. If the user is close to the light and claps, the light turns on or off. For this functionality, the sound sensor is used as it was stated.

On each floor, there is a control unit on the wall (Fig. 15). When the user focuses and clicks on it, the menu (Fig. 12) of the central unit opens.

In Fig. 16 there is the exterior of the smart house.

V. CONCLUSION

Nowadays, there is a trend of using interactive 3D applications and virtual reality in virtual tours for houses, cars and other products. This paper describes an interactive 3D application that simulates virtual tour of the smart house and its exterior. The application is implemented in Unity engine. As it is the interactive application that responds to the perceptions and changes from the environment, it is necessary to connect it with external hardware which captures the signals from the environment and sends the data to the application. In future research, it would be interesting to use this experience and develop the application for mixed reality (e.g. Microsoft HoloLens) that will communicate with real sensors and actuators. In this way, it will be possible to control a real smart house using a mixed reality application.

ACKNOWLEDGMENT

This work has been supported by the Cultural and Educational Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic, KEGA 030STU-4/2015 and KEGA 030STU-4/2017, by the Scientific Grant Agency of the Ministry of Education, Science, Research and Sport of the Slovak Republic under the grant VEGA 1/0819/17 and by the Tatra banka Foundation within the grant programme Quality of Education, project No. 2016vs046 (Support of education in mechatronics through virtual reality).

REFERENCES

[1] Triseum. (2017) Variant: Limits. [Online]. Available: https://triseum.com/calculus/variant/

[2] K. Sloan. (2016) Rotor brings toyota showroom 360 to life with unreal engine. [Online]. Available: https://www.unrealengine.com/showcase/rotor-brings-toyota-showroom-360-to-life-with-unreal-engine

[3] A. Technologies. (2014) Virtual gearbox. [Online]. Available: http://www.animechtechnologies.com/showcase/virtual-gearbox/

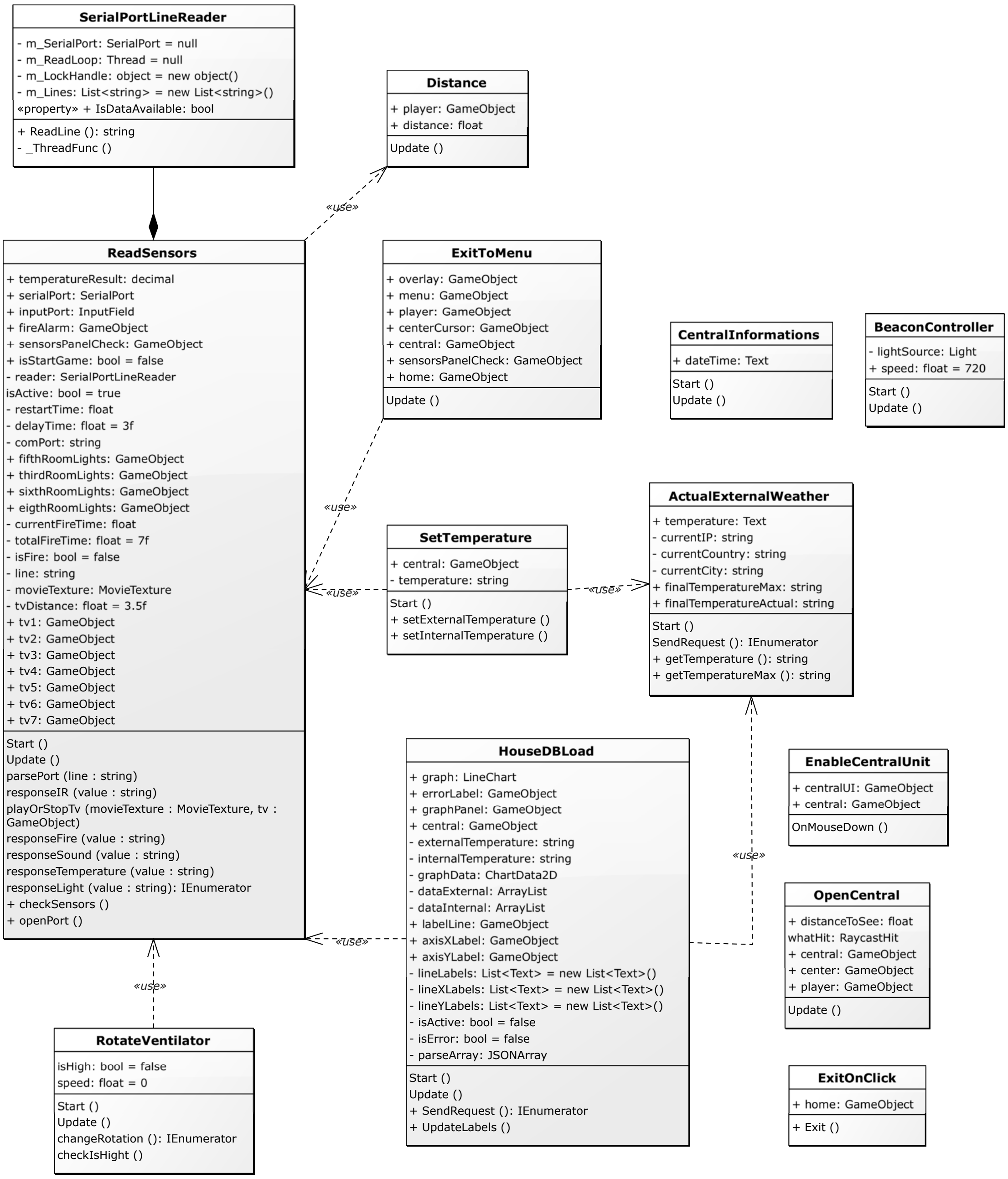


Fig. 10. Class diagram

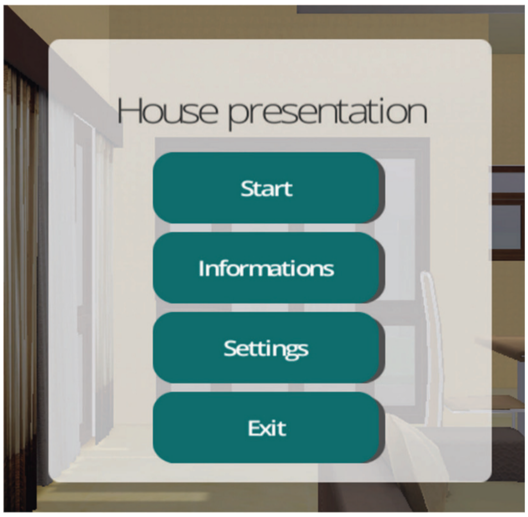
[4] D. Parvati, W. L. Heinrichs, and Y. Patricia, "Clinispace: a multiperson 3d online immersive training environment accessible through a browser," *Medicine Meets Virtual Reality 18: NextMed*, vol. 163, p. 173, 2011.

[5] P. A. Rauschnabel, A. Brem, and Y. Ro, "Augmented reality smart glasses: definition, conceptual insights, and managerial importance," *Working paper, The University of Michigan-Dearborn, Tech. Rep.*, 2015.

[6] E. Uhlemann, "Connected-vehicles applications are emerging [connected vehicles]," *IEEE Vehicular Technology Magazine*, vol. 11, no. 1, pp. 25–96, 2016.

[7] M. Corporation. (2017) Holotour. [Online]. Available: https://www.microsoft.com/en-us/hololens/apps/holotour

[8] M. Copeland, J. Soh, A. Puca, M. Manning, and D. Gollob, "Microsoft azure and cloud computing," in *Microsoft Azure*. Springer, 2015, pp. 3–26.



House presentation

Start

Informations

Settings

Exit

Fig. 11. Main menu

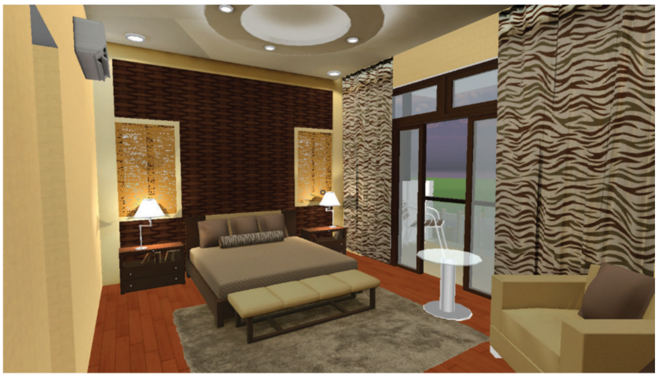
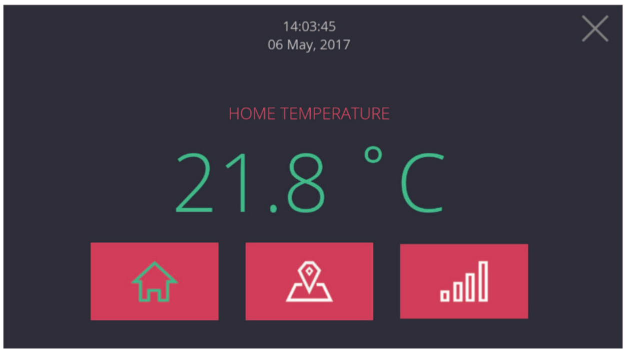


Fig. 14. Lights in the bedroom



14:03:45
06 May, 2017

HOME TEMPERATURE

21.8 °C

Fig. 12. Temperature menu

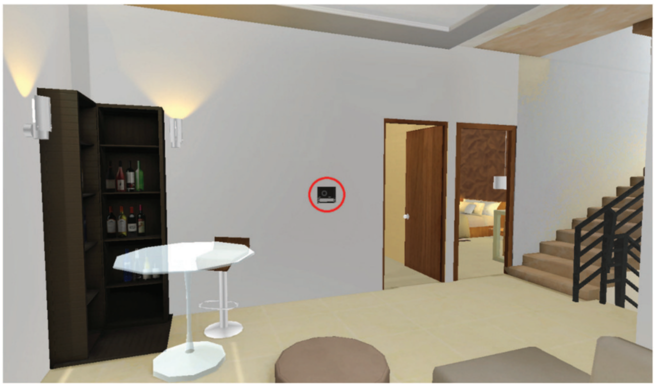


Fig. 15. Central control unit

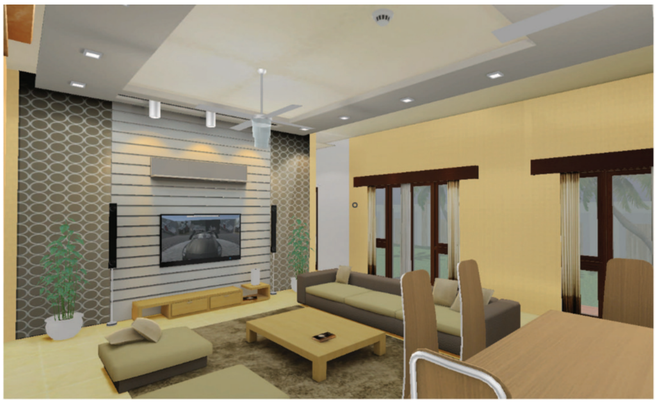


Fig. 13. Interior of the living room

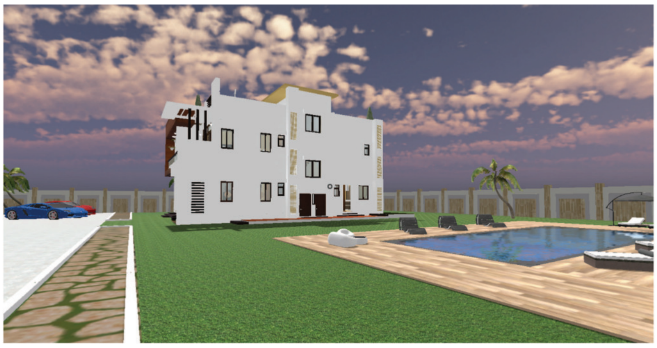


Fig. 16. Exterior