

Interactive virtual world for games

Dana Simian, Petrică Bota, Robert Sandică

Abstract

The aim of this paper is to present a system that implements virtual world technology in the gaming design. The main idea was to develop, with a lower price, an environment which the user can explore and interact with. We have built a system that makes use of virtual reality glasses to project realistic surroundings where the player uses a controller designed by us in order to interact with the world. This system represents a first step in creating a new platform for users as well as for developers.

Keywords: virtual world, game, programming

1 Introduction

The concept of virtual reality (VR) refers to the computer simulation of the reality and the artificial creation of sensory experiences. The idea of VR and some prototypes first appeared in the 50's but the technology at the time was not powerful enough to sustain a stable virtual world [3]. The recent breakthroughs in processor power and screen resolutions gave the possibility to develop devices able to create viable virtual experiences. VR technology can be implemented in many fields like healthcare, education, aeronautics, military, entertainment, etc. and can bring forth benefits [3, 6]. Virtual reality can also be used to present information that cannot be comprehended otherwise [5].

In the virtual environment systems the actors are a human operator, a computer and a human-machine interface [1, 2]. The human-machine interface is multimodal, the human actor is immersed in the computer-generated environment and the reality is perceived in three dimensions [1]. There are many software for creating VR, but most of them are dedicated to professional users, require expensive devices and are not free [11, 12].

The aim of this paper is to design a system that implements VR technology in the gaming design. The main idea was to develop, with a lower price, an environment which the user can explore and interact with. A 3D visual display is the output of the user interface, while the interaction is done through manipulation and navigation.

We created a virtual 360° open world which can be explored with the help of a VR headset. The headset immerses the user in the virtual world while the movements of the user as well as its interaction with the environment are controlled using a virtual rifle designed by ourselves. The rifle acts as a controller through which the user controls its actions.

For the beginning we created a frame for the virtual world. It is based on a simple map and contains landscapes such as mountains, a lake and a grassy field. The basic action of the user consists in controlling a character (a tank) with the help of the rifle as a controller. The rifle is meant to add to the realistic experience given by the virtual reality headset.

The paper is organized as follows. In Section 2 we present the technologies used to design and implement the virtual world. Section 3 is dedicated to the new input device: the virtual rifle (the controller). Details of implementation of our virtual VR platform can be found in Section 4. Section 5 reveals new directions of study for further development of our work.

2 Technologies

2.1 Unity

We developed the frame of our VR world using Unity, due to its features. Unity is a powerful game engine that allows developers to easily create 2D and 3D games [9]. The engine allows the creation of games from the simplest ones to the most extensive intricate ones developed by big companies.

Unity puts accent on the portability of its applications by supporting the following APIs: Direct3D on Windows Xbox 360; OpenGL on Mac, Windows and Linux. The engine provides game developers with a multitude of tools such as: automatic shadowing, object collisions or map creation.

2.2 Zeiss VR One

The immersion of the user in our virtual world is realized using a VR headset. There are many headsets from which to choose. A list of VR headsets can be found at [10]. Our choice was Zeiss VR One, a mobile and system-independent headset. It works via apps installed on smartphone devices. The main reason why we chose Zeiss VR was the fact that the headset was relatively cheap compared to others of the same quality available at the time. Another important reason was the fact that Zeiss provides a free SDK for developers in order to aid them in the creative process. The VR One SDK [13] is a SDK for Unity with the purpose of helping developers to create virtual reality apps for the VR. As of now it supports apps for mobile created with Unity3D for Android and IOS. The SDK splits the screen of the smartphone into two video feeds and applies a pre-distortion field along the edges so as not to create discomfort to the user. The SDK also uses the smartphone's sensors to track head-movement and reduce latency as much as possible [14, 15].

3 The controller

We felt that a keyboard or a joystick do not transmit the sensory/immersive feeling that we were looking for. Thus we created our own controller that simulates a rifle. It is based on the Arduino Leonardo board for data transmission and processing the signal from sensors. It uses an AT MEGA 32 microcontroller and has available 14 PWM pins and 8 ANALOGIC pins. It has a 3V or 5 V power source.

The rifle is the device that controls the movements of your character (i.e. in fact of the user) as well as its interaction with the environment.

The rifle can work in two ways: analogical and digital. In the analogical form you move the target on the screen at the exact position. In the digital form you only transmit in which direction the rifle is moved.

The controller is not represented by just the device; it is the whole system with two parts: the Rifle and the AIM library.

3.3.1 The Rifle

The first part, the rifle itself is intended more for the general user. You simply plug in the rifle and it acts as your input device. In Fig.1. are presented the components of the Rifle in early stages of construction [4].

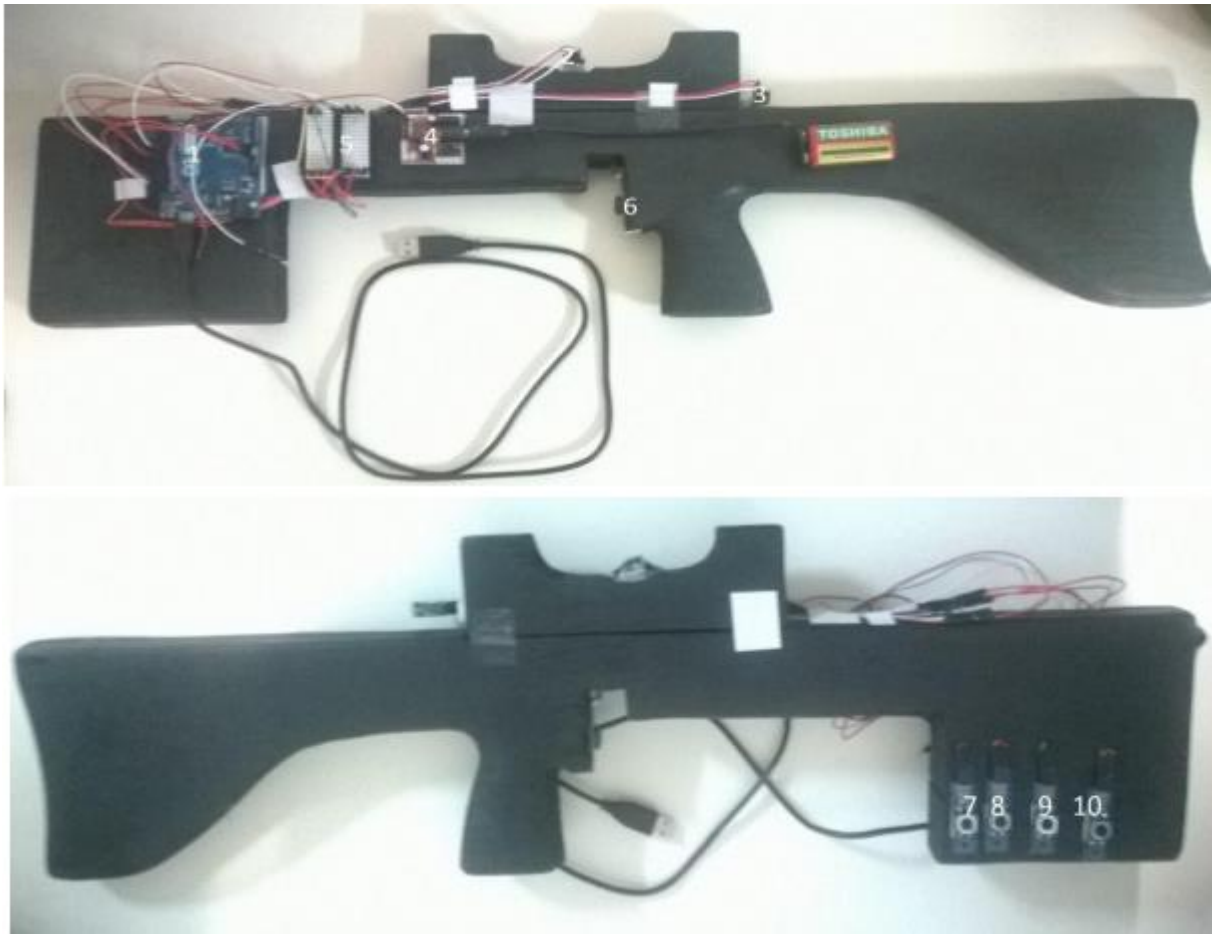


Fig. 1 - Rifle's main components: 1. Arduino board; 2. IMU sensor; 3. Proximity sensor; 4. Power supply; 5. Breadboard; 6 - 10. Buttons.

Arduino is an open source kit composed of a program and a single board microcontroller [7]. Arduino platform is based on the C and C++ programming language. Arduino Leonardo board is based on the AT Mega microcontroller and it is capable of controlling the mouse and keyboard with a set of libraries called Mouse and Keyboard [8].

We used the IMU sensor Pololu MinIMU-9, with 9 degree of freedom. It returns the exact orientation of the device to the user. It is equipped with an accelerometer, a magnetometer and a gyroscope, all three having 3 axes of orientation.

The proximity sensor is an infrared based sensor detecting if an object blocks the ray of light at distances of 2 to 10 cm.

The power supply ensures that the energy is provided in the range of 3 to 5 Volts.

The breadboard is a wire connection pad.

The buttons have a brick form, i.e. they are some sort of plug and play ready to use. The button 6 is used to shoot. The other four buttons control the movements. They replicate the W, A, S, D keys used for forward, left, back and right.

The inertial sensor (IMU sensor) returns to Arduino board the two axes coordinates: the X axis, given by magnetometer reading the North and the Y axis given by accelerometer [4]. Using these two axes coordinates we have developed a formula which converts the position of the sensor into the target cursor on the screen. Our formula reveals a linear dependence of them.

In order to use the setup the controller needs to be connected to the smartphone via a OTG cable. The battery has to be connected and the voltage regulator must be turned on. Controller calibration in the virtual reality world is made by keeping the rifle aligned with the headset in a

comfortable position and pressing the first button placed on the handle of the controller (button 6 from Fig. 1).

3.3.2 The AIM Library

The second part of the controller is the AIM Library. This library is written in C#. This part of the project is intended for the developers. It realizes the connection between Arduino and the code that you are writing to create the application that will use the Rifle.

The library has two main uses: one is intended to receive values from the proximity sensor and the buttons and the other one is intended to set the actions that take place when you press any of the buttons set on the rifle. Using the AIM library you can change the buttons and proximity sensor configuration giving them any value that you want.

4 Implementation

As stated earlier our desire was to create a simple frame for a virtual reality platform onto which we can add further development. Our present simple frame is used in order to get users accustomed with the new controller as well as with the virtual reality world. In order to achieve this we designed first a simple map in such a way that the user has no other alternative but to steer a tank downhill while he explores the map. For improving the user's dexterity in handling the rifle we stacked red boxes which the user can aim and shoot at. The game's goal is to hit the red boxes with the tank's cannon while exploring new parts of the map. What separates this game from others is the fact that you are actually inside the game and the way you move and shot is transmitted from the controller that resembles a rifle.

In order to create the terrain we used the terrain builder that Unity offers. The character was imported as an asset but we modified it so that it suited our needs. A screenshot from the game design process is presented in Fig. 2.

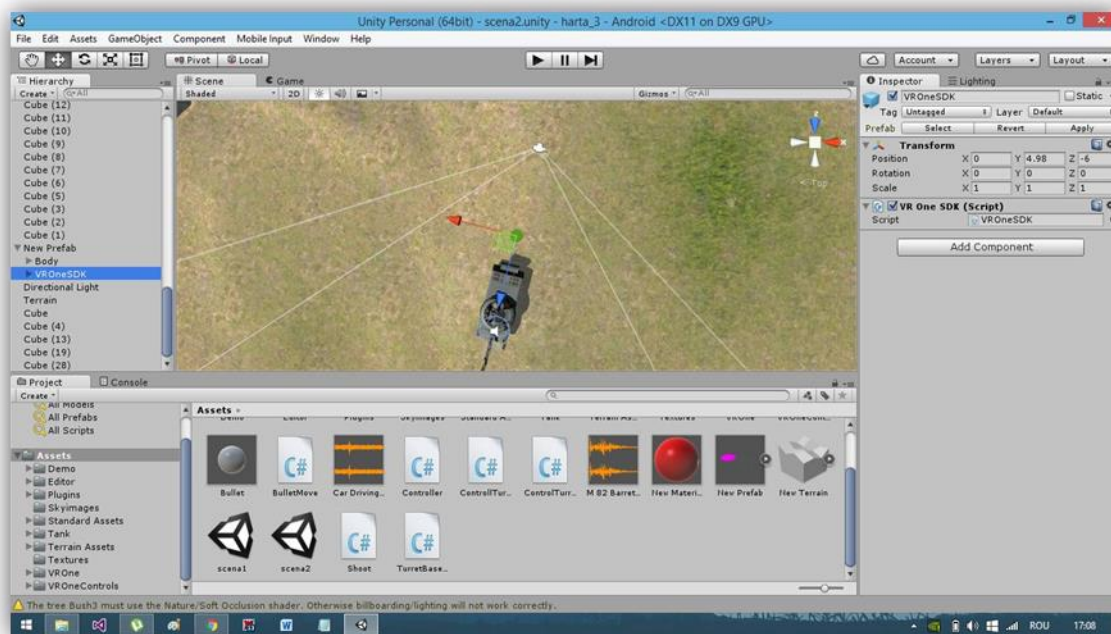


Fig. 2. – Image from the game design using Unity

The scripts that control the character were written by us in C#. At the moment the only options that the user has are to explore the map, test his accuracy by shooting the boxes and get used with our setup.

The script that controls the tank movement is given below:

```

// Update is called once per frame
void Update()
{
    //Sound playing when a button is pressed to move the tank
    if (Input.GetKey(KeyCode.W)|| Input.GetKey(KeyCode.A) || Input.GetKey(KeyCode.D) ||
Input.GetKey(KeyCode.S))
    {
        if (!Source.isPlaying)
            Source.Play();
    }
    else
    {
        Source.Stop();
    }
    //Rotate the body of the tank
    transform.Rotate(new Vector3(0, Input.GetAxis("Horizontal") * RotationSpeed * Time.deltaTime,
0));
    //Accelerating tank
    Vector3 forward = Input.GetAxis("Vertical") * transform.TransformDirection(Vector3.forward) *
MoveSpeed;
    cc.Move(forward * Time.deltaTime);

    //gravity actioning
    cc.SimpleMove(Physics.gravity);
}

```

The split screen provided by the SDK is shown in Fig. 3.



Fig. 3. The split screen provided by the SDK

Other virtual worlds can be built on top of this frame by adding maps and by implementing more interactive actions.

5...Conclusions and further directions of study

By using Unity, which is a free game engine (up to a point), with the Zeiss VR One and a smartphone we created an open VR platform frame that can be accessed by many people relatively cheaply. Our frame enables further developments by adding new maps and actions of the users in the 3D environment provided by the game. More elaborate and interactive virtual worlds can be obtained starting from our frame. A further goal is to construct new controllers capable to improve the user experience in a VR world. In the future we intend to enrich the educational content of the games realized using our VR platform, developing actions that teach concepts or ideas to the user while they are playing the game.

Acknowledgement: Dana Simian was supported by the research grants LBUS-IRG-2015-01, project financed from “Lucian Blaga” University of Sibiu.

References

- [1] Nathaniel I. Durlach, Anne S. Mavor, Editors, *Virtual Reality. Scientific and Technological Challenges* The National Academies Press, 1995.
- [2] Roy S. Kalawsky, *Science of Virtual Reality and Virtual Environments*, Addison-Wesley Pub (Sd); 2nd edition, 2004.
- [3] Matjaz Mihelj, Domen Novak, Samo Beguš, *Virtual Reality Technology and Applications*, Springer Science + Bussiness Media, Dordrecht, 2014.
- [4] Robert Sandica, Marius Toma, Dana Simian, *AIM - Kit for app development*, Imagination, Creativity, Design, Development - Proceedings of the International Students Conference on Informatics, ICDD 2015, Sibiu, Romania, „Lucian Blaga” University Press, pp. 142-147, 2015.
- [5] Alan Wexelblat (Editor), *Virtual Reality: Applications and Explorations*, Academic Press Professional, 1993.
- [6] Applications Of Virtual Reality - <http://www.vrs.org.uk/virtual-reality-applications/>
- [7] Arduino description - <http://www.arduino.cc/>
- [8] Arduino Leonardo board picture and description - <http://www.arduino.cc/en/Main/ArduinoBoardLeonardo>
- [9] Unity homepage - <https://unity3d.com/>
- [10] Virtual Reality headset with review - www.aliexpress.com/
- [11] Virtual Reality Software, Augmented Reality, 3D Visualization, Knowledge Transfer, Training – www.eonreality.com/
- [12] Virtual Reality Software - www.worldviz.com/virtual-reality
- [13] VR One SDK - <https://bitbucket.org/vrone/unity3d/src/3c7395473a371ff755f89715a36ee532427c780d#markdown-header-what-is-the-vr-one-sdk>
- [14] Zeiss VR One homepage - <http://zeissvrone.tumblr.com/>
- [15] Zeiss VR One - Information – https://vronecontest.zeiss.com/index.php?controller=pages&view=about_vr

DANA SIMIAN
 “Lucian Blaga” University of Sibiu
 Faculty of Sciences
 Research Center ITI
 Dr. Ratiu Str 5-7
 ROMANIA
 E-mail: dana.simian@ulbsibiu.ro

PETRICĂ BOTA
 “Lucian Blaga” University of Sibiu
 Faculty of Sciences
 Dr. Ratiu Str 5-7
 ROMANIA
 E-mail: sandica_robert@yahoo.ro

ROBERT SĂNDICĂ
 “Lucian Blaga” University of Sibiu
 Faculty of Sciences
 Dr. Ratiu Str 5-7
 ROMANIA
 E-mail: bota.petrica2@yahoo.com