ISSN 1870-4069

Implementation of a Reactive Model for Responding to a Trembling Earthquake: A Perspective from Virtual Reality and Multiagent Systems

Miguel Pérez-Ramírez[1], Benjamin Eddie Zayas-Pérez[1], Rodrigo Beltran-Labra[2], Diego Márquez-Salazar[2], Carlos Alberto Ochoa-Zezzatti[3]

[1] National Institute of Electricity and Clean Energies, Cuernavaca, Morelos, Mexico

[2] Polytechnic University of the State of Morelos, Jiutepec, Morelos, Mexico

[3] Autonomous University of Juarez city, Ciudad Juárez, Chihuahua, Mexico

{mperez, zayas}@iie.org.mx, {blro130524, mado140055}@upemor.edu.mx, alberto.ochoa@uacj.mx

Abstract. We present a virtual reality simulation of an evacuation caused by a trembling earthquake in a secondary school. The simulation is based on artificial intelligence for which a mathematical model is implemented that controls the behavior of agents that represent the school students in the virtual stage. The simulation allows the interaction with the agents on stage by using the gyroscope of a smartphone while roaming around the stage and thus observing the behavior of the agents when the human stampede occurs. The simulation can be executed either on a computer or on a smartphone. Once again, the importance of artificial intelligence to prevent losses of human lives is demonstrated.

Keywords: Artificial Intelligence, Virtual Reality, Simulation, Multi-agents Systems, Paranoia, Behavioral Patterns.

1 Introduction

We can have an interactive virtual world behind the screen of a computer, however, the challenge is to make that world to looks real and feels like such. Virtual reality (VR) has been evolving throughout time and has been accepted as a viable technique in different fields such as visualization [5], simulation, design and research. Our senses allow us to experience the reality we live in, VR aims to allow interaction with a scenario in third dimension where this reality is emulated even if this is not available for being dangerous

pp. 25–35; rec. 2016-08-06; acc. 2016-10-08

or physically inaccessible; people will be even able to see and interact within physically inexistent virtual worlds, but from their perspective these will be perceived as real. VR has a wide variety of applications in a number of different fields, such as medicine, architecture, design, sports, humanitarian logistics and even games, just to name a few of them. It is within humanitarian logistics that this research is based on. Here we describe how we developed a VR simulator for a humanitarian logistics, where we attempt to make predictions in the unpredictable natural disasters such as earthquakes in order to reduce the possibility of human losses and injuries.

Specifically, by integrating Virtual Reality and Artificial Intelligence, we seek to find out those specific obstacles that could prevent the opportune evacuation of students and teachers from a three-level building of a secondary school, and observe at the same time the behavior of all agents involved in real time.

2 Materials and Methods

For the development of this research, a Smartphone Samsung J7 was used together with a computer with a Operating System X Yosemite version 10.10.5 (Intel Core i5 processor, 6MB in Cache and 3.20 GHz, graphics card NVIDIA GeForce GTX 1070 with 8GB of RAM).

In order to model the structure of the building of the secondary school, we took as a reference the plane defined in a previous work [2], which was modeled and programmed (Fig. 1) in the framework "Menge" [6].

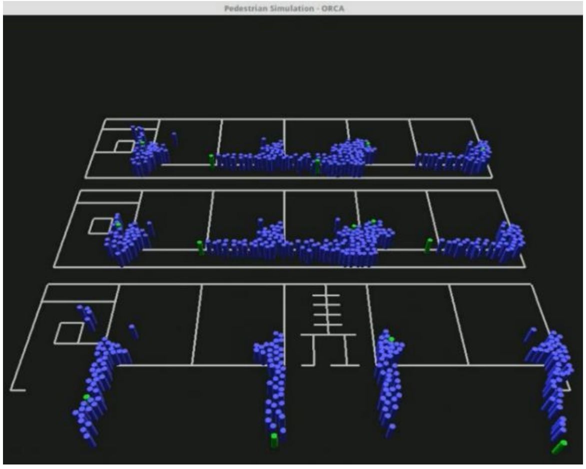


Fig. 1. Simulation of evacuation of the structure using multi-agents system.

The methodology used to calculate the pedestrian speed is made up of several sub models that are used to estimate the level of paranoia, the density in the classroom, and the impact of the socio-cultural training of the student [2]. The equation that calculates the level of paranoia considers the following variables: noise (R), temperature (Temp), lack of sleep (Fs), intensity of hunger (Af), density of people in the classroom (D) and adjustment variable (α=4).

Two variables are considered to calculate the density of students: living room surface (S) and the total amount of people in the classroom (T).

$$D = \left(\frac{T}{S}\right) \tag{1}$$

Equation that calculates the decibel scale from number:

$$dB = (R * 90) + 30 \tag{2}$$

Equation that calculates the number scale from the decibel:

$$R = \frac{-30 + dB}{90} \tag{3}$$

The lack of sleep is defined on a scale of 0 to 1 and is calculated by taking into account sleepiness as the maximum value of the scale and 8 hours or more of sleep as the lowest value. The equation that defines the sleep-failure scale was designed in such a way as to represent a person's fatigue exponentially.

$$Y = \begin{cases} 1 \; if \; x = 0 \\ x > 0 \rightarrow \dfrac{1}{x} - \alpha \end{cases} \tag{4}$$

$$Domain = [0, \infty) \quad , \quad Range = \; (0\,, 1]$$

For the intensity of hunger are taken three states: fasting, breakfast and lunch. When the student is in a state of fasting, it is harder to react in a timely manner because his body does not have the necessary nutrients to perform optimally. By contrast, if the student breakfasted before the accident, can physically respond faster; even more, if the student breakfasted and then had lunch, she/he is in a good condition to face obstacles that are presented to evacuate the building [2]. For hunger states, we have the following values:

- Fasting = 1.00; Breakfast = 0.50; Lunch = 0.01

Equation that calculates the paranoia:

$$P = \frac{(R + Temp + Fs + Af)^D}{\alpha} \tag{5}$$

For the equation that defines the socio-cultural model the following factors are taken: level of altruism (A), gender (g) and maturity (m).

The maturity factor takes into account the maturity of mental development of the student, which is usually is determined by age.

Equation of sociocultural model:

$$C = A * m * g \tag{6}$$

Finally, to carry out the calculation of the pedestrian velocity in each student the following factors are used to evaluate the probability of vital loss: earthquake magnitude (G), level of the structure (N), index of paranoia (P), time (t), sociocultural model (C) and adjustment variable (X=0.3).

The magnitude of the earthquake is not adjusted or represented by any other value, rather than the original Richter scale, therefore in decimal and can range from 4.0 to 9.0 on the Richter scale. It takes 4.0 as initial reference value because it is a scale in which the earthquake can be felt. Since there is no record of any major 8.6 earthquake in the history of our country (México), it is taken as value limit reference 9.0 [2]. The level of the structure can take the values 1, 2 or 3 depending on the level of the structure where the student is located. The index of paranoia is calculated using equation (5) and their possible values range are from 0 to 1. The time is measured in seconds and takes into account the range of 0-90 seconds, this is because it is the maximum time evacuation before the structure collapses.

In order to calculate the socio-cultural level of the student, we apply the equation (6) and as the same as paranoia, values range are from 0 to 1.

Equation that calculates the pedestrian speed:

$$V(t) = \left[\frac{G + N}{P(t)} * C * X\right] \tag{7}$$

2.1 Displacement with Gyroscope

The gyroscope installed on the Smartphone measures the movements of the player and makes our device follow the gestures and movements (Fig. 2).

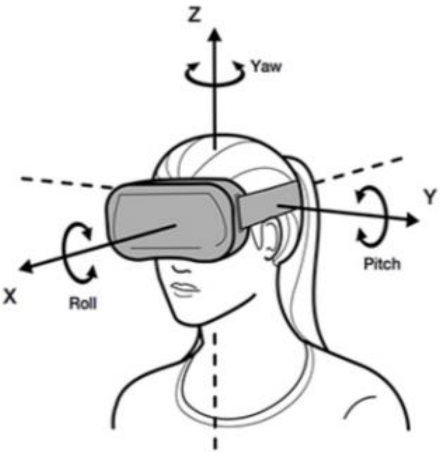
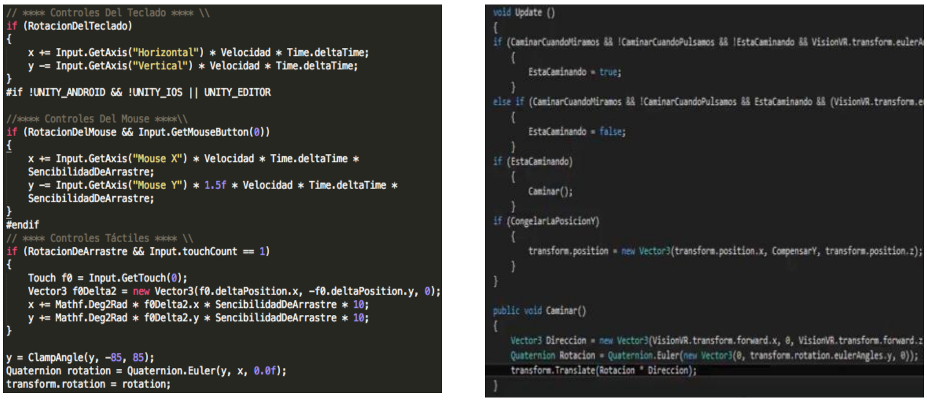


Fig. 2. Movements detected by the gyroscope [8].

A movement algorithm is proposed, so that the player can move around the stage regardless of the terrain type (Fig. 3a); the main objective of this algorithm is to detect whether the player is touching the ground or not (Fig. 3b).



```
// **** Controles Del Teclado **** \\
if (RotacionDelTeclado)
{
    x += Input.GetAxis("Horizontal") * Velocidad * Time.deltaTime;
    y -= Input.GetAxis("Vertical") * Velocidad * Time.deltaTime;
}
#if !UNITY_ANDROID && !UNITY_IOS || UNITY_EDITOR

//**** Controles Del Mouse ****\\
if (RotacionDelMouse && Input.GetMouseButton(0))
{
    x += Input.GetAxis("Mouse X") * Velocidad * Time.deltaTime *
    SencibilidadDeArrastre;
    y -= Input.GetAxis("Mouse Y") * 1.5f * Velocidad * Time.deltaTime *
    SencibilidadDeArrastre;
}
#endif
// **** Controles Táctiles **** \\
if (RotacionDeArrastre && Input.touchCount == 1)
{
    Touch f0 = Input.GetTouch(0);
    Vector3 f0Delta2 = new Vector3(f0.deltaPosition.x, -f0.deltaPosition.y, 0);
    x += Mathf.Deg2Rad * f0Delta2.x * SencibilidadDeArrastre * 10;
    y += Mathf.Deg2Rad * f0Delta2.y * SencibilidadDeArrastre * 10;
}

y = ClampAngle(y, -85, 85);
Quaternion rotation = Quaternion.Euler(y, x, 0.0f);
transform.rotation = rotation;
```

(a)

```
void Update ()
{
if (CaminarCuandoMiramos && !CaminarCuandoPulsamos && !EstaCaminando && VisionVR.transform.eulerA
    {
        EstaCaminando = true;
    }
else if (CaminarCuandoMiramos && !CaminarCuandoPulsamos && EstaCaminando && (VisionVR.transform.e
    {
        EstaCaminando = false;
    }
if (EstaCaminando)
    {
        Caminar();
    }
if (CongelarLaPosicionY)
    {
        transform.position = new Vector3(transform.position.x, CompensarY, transform.position.z);
    }
}

public void Caminar()
{
    Vector3 Direccion = new Vector3(VisionVR.transform.forward.x, 0, VisionVR.transform.forward.z
    Quaternion Rotacion = Quaternion.Euler(new Vector3(0, transform.rotation.eulerAngles.y, 0));
    transform.Translate(Rotacion * Direccion);
}
```

(b)

Fig. 3. (a) Script that contains the movement algorithm for the player, written in C#; (b); script that defines the walk of the player, written in C#.

The controls are defined in the scripts, so the player can walk on the stage. The movement of the player is defined as follows: walk forward (down); stop player (up), rotate left (left) and rotate right (right). The control movements work to handle with mouse or with Smartphone.

The implementation with Google VR SDK in the simulation gets the virtual view for the player (Fig. 4). The Google VR SDK for Unity provides additional features like spatialized audio, Daydream controller support, utilities and samples [3].

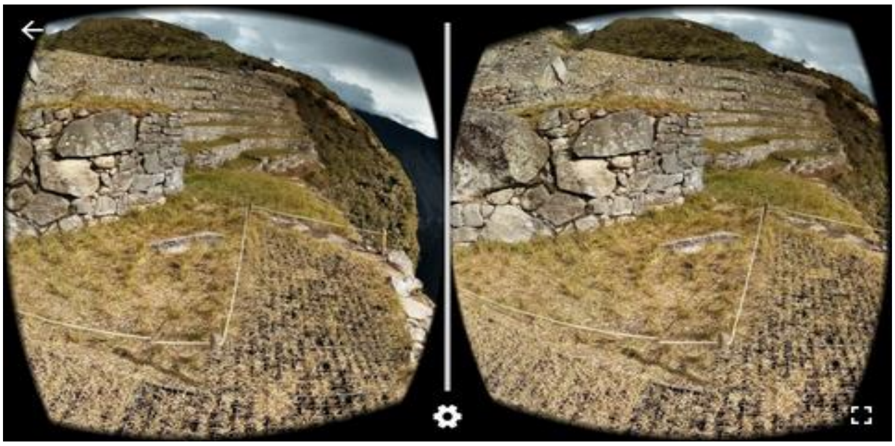


Fig. 4. The Cardboard mode allows the user to see the panorama in a Cardboard device like any other Cardboard app [4].

3 Problem Statement

The problem addressed in this research is to develop a more realistic visual environment of the evacuation of a three-level structure in a secondary school. We aim to have a more accurate observation of the behavior of the agents when they are in a state of human stampede. By using the methodology that controls the behavior of the agents used by Beltran et al. (2016) in their article "Characterization of an undulatory earthquake in a three levels building of a high school using multi-agent systems" characterizes the agents as students of the school and they are applied the model in the simulation in third dimension using "Unity". The information gathered from this study will be used to ensure that the personnel in charge of security in structures determine the factors that hinder the evacuation of the building with the exact tracking of the trajectory of each of the students. This intends to save as many human lives as possible.

3.1 Models of the Interfaces

"Unity" is an image processing and 3D development toolkit [1] that allows us to represent the scenario of the secondary school in a more real and accurate way, for that reason we take as reference the model developed in the previous work constructed in three dimensions (Fig. 5).

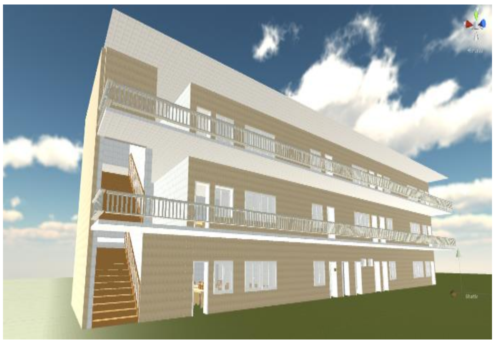
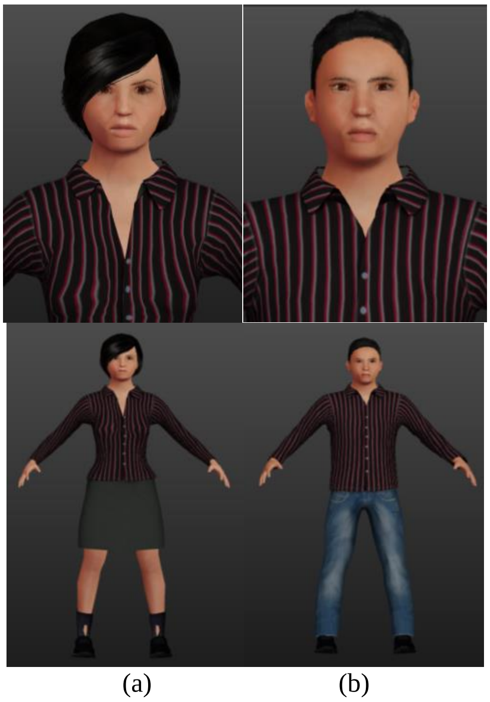


Fig. 5. Representation of the three-level structure using Unity 3D.

MakeHuman is a free and open source software for the creation of more realistic third-dimensional humans [7] to characterize high school students (Fig. 6).

The player can observe a virtual scenario as shown in Fig. 7 and 8, in which she/he can walk through the entire structure and observe the behavior of the agents.



(a) (b)

Fig. 6. (a) Characterization of students' faces in MakeHuman. (b) Characterization of the body of the students in MakeHuman.

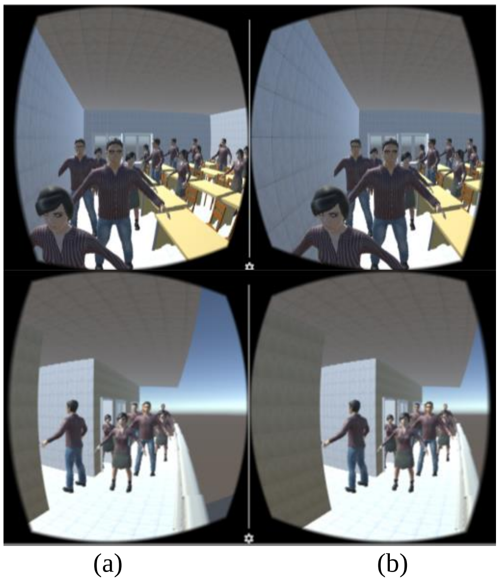


Fig. 7. (a) Students starting to leave the classroom implementing Virtual Reality. (b) Evacuation seen from the corridor of the third level of the structure.

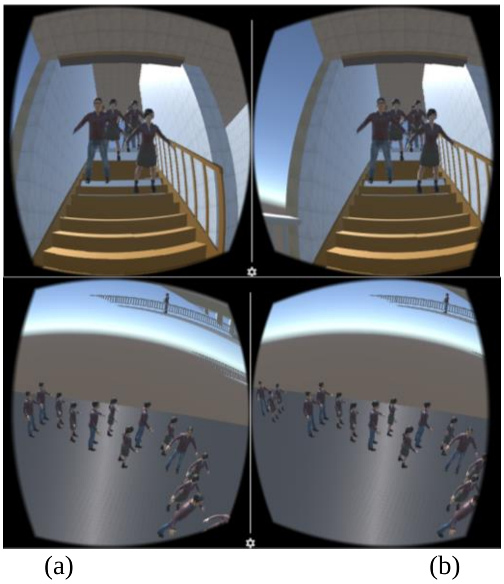
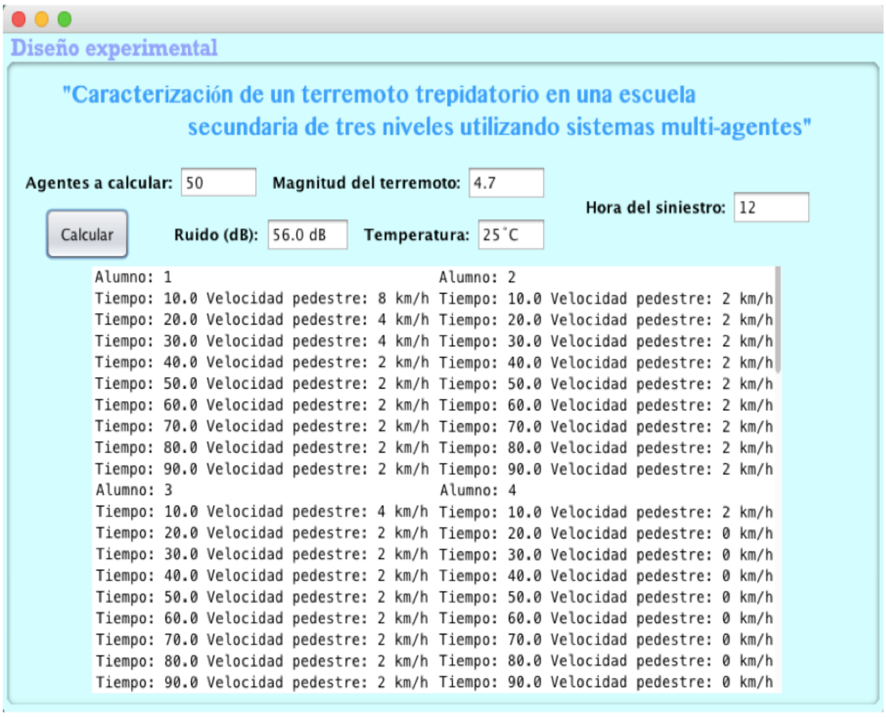


Fig. 8. (a) Seen from the stairs of the second floor of the structure, this place usually occurs the human stampede. (b) View of the first students going out to the patio.

4 Design of Experiments

Once the pedestrian speed model has been mathematically tested, it was encoded and implemented in an application created in Java to generate random values (Fig. 9). These values are used to characterize the behavior of each of the students in the building. For instance, it can be seen that the velocity of student number 4 is zero at the second 20, which means that the student remained paralyzed or trapped within the structure and is interpreted as a human down.

According to the simulation, all agents spend 1 minute with 28 seconds to evacuate the structure. In order to verify to what extent the pedestrian speed model approached the real life situation with respect to the students' behavior, students were trained to perform a simulacrum, assuming that the alarm alerted them in the opportune moment. When observing the behavior of the students and later comparing it with the simulation, many similarities were detected. The most notable is the way to go down the stairs and the moment of the human stampede.



"Caracterización de un terremoto trepidatorio en una escuela secundaria de tres niveles utilizando sistemas multi-agentes"

Agentes a calcular: 50
Magnitud del terremoto: 4.7
Hora del siniestro: 12
Ruido (dB): 56.0 dB
Temperatura: 25˚C

```
Alumno: 1                                Alumno: 2
Tiempo: 10.0 Velocidad pedestre: 8 km/h Tiempo: 10.0 Velocidad pedestre: 2 km/h
Tiempo: 20.0 Velocidad pedestre: 4 km/h Tiempo: 20.0 Velocidad pedestre: 2 km/h
Tiempo: 30.0 Velocidad pedestre: 4 km/h Tiempo: 30.0 Velocidad pedestre: 2 km/h
Tiempo: 40.0 Velocidad pedestre: 2 km/h Tiempo: 40.0 Velocidad pedestre: 2 km/h
Tiempo: 50.0 Velocidad pedestre: 2 km/h Tiempo: 50.0 Velocidad pedestre: 2 km/h
Tiempo: 60.0 Velocidad pedestre: 2 km/h Tiempo: 60.0 Velocidad pedestre: 2 km/h
Tiempo: 70.0 Velocidad pedestre: 2 km/h Tiempo: 70.0 Velocidad pedestre: 2 km/h
Tiempo: 80.0 Velocidad pedestre: 2 km/h Tiempo: 80.0 Velocidad pedestre: 2 km/h
Tiempo: 90.0 Velocidad pedestre: 2 km/h Tiempo: 90.0 Velocidad pedestre: 2 km/h
Alumno: 3                                Alumno: 4
Tiempo: 10.0 Velocidad pedestre: 4 km/h Tiempo: 10.0 Velocidad pedestre: 2 km/h
Tiempo: 20.0 Velocidad pedestre: 2 km/h Tiempo: 20.0 Velocidad pedestre: 0 km/h
Tiempo: 30.0 Velocidad pedestre: 2 km/h Tiempo: 30.0 Velocidad pedestre: 0 km/h
Tiempo: 40.0 Velocidad pedestre: 2 km/h Tiempo: 40.0 Velocidad pedestre: 0 km/h
Tiempo: 50.0 Velocidad pedestre: 2 km/h Tiempo: 50.0 Velocidad pedestre: 0 km/h
Tiempo: 60.0 Velocidad pedestre: 2 km/h Tiempo: 60.0 Velocidad pedestre: 0 km/h
Tiempo: 70.0 Velocidad pedestre: 2 km/h Tiempo: 70.0 Velocidad pedestre: 0 km/h
Tiempo: 80.0 Velocidad pedestre: 2 km/h Tiempo: 80.0 Velocidad pedestre: 0 km/h
Tiempo: 90.0 Velocidad pedestre: 2 km/h Tiempo: 90.0 Velocidad pedestre: 0 km/h
```

Fig. 9. Random data generation with application created in Java.

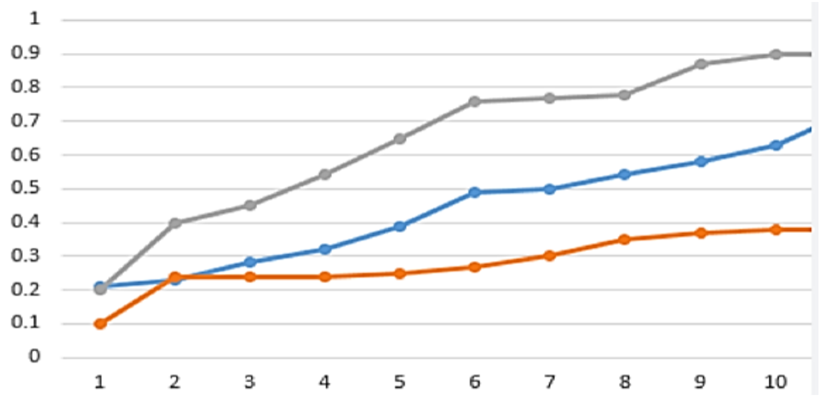


Fig. 10. The graph shows the behavior of three different students. The X axis represents the time in seconds, while the Y axis represents the pedestrian speed scale. The student represented by the orange line had a level of paranoia greater than the others while the student represented by the gray line had a lower level of paranoia, where the pedestrian speed is not very affected.

It should be noted that an accuracy of slightly more than 90% in the behavior of the agents in the simulations was obtained since in the simulacrum the students took 1 minute with 33 seconds to evacuate the building and the agents in the simulation take 1 minute with 28 seconds, a difference of 5 seconds.

5 Conclusions and Future Research

The implementation of the VR in this research has been used to detect in a more real way the possible factors that influence the safe evacuation of the building since the players can be moved throughout the building, even more, it can be perceived certain emotions similar to the adrenaline experienced in a real situation, this simulator seeks to face the challenge of saving the greatest number of human lives and eliminate obstacles to optimize evacuation routes.

As future work, it is intended that the mathematical model developed in this research can be used in different scenarios of catastrophes or natural unexpected events or even provoked, in which an increment in paranoia is caused by the situation and the environment. Where Paranoia is the main factor that intervenes in the process of escape and human movement routes, the model and the methodology can be adapted to diverse situations in which it is necessary to analyze and to propose solutions of humanitarian logistics in addition; it can be also used to perform security tests of existing properties. In near future, we are planning to apply this methodology in different scenarios with augmented reality.

Acknowledgments. Dr. Francisco Javier Bermúdez Alarcón (Director General in State Coordination of Civil Protection of the state of Morelos), for the supervision and validation of the investigation. Lic. Luz Elena Espino García (Director in General Secondary School # 4, "Valentín Gómez Farías", Cuernavaca, Morelos), for the attentions provided to carry out the simulacra in the institution. Eng. Peter Savier Oropeza. (Polytechnic University of Morelos), for the technological support offered in this research.

References

1. Autodesk Inc.: Autodesk. Autocad 2017, Website: http://www.autodesk.mx/campaigns/ (2016)
2. Beltran, R., Márquez, D., Ochoa-Zezzatti, A.: Characterization of an ondulatory earthquake in a three levels high school using multi-agent systems. In Humanitarian Logistics based on Multiagents Systems for the prevention of a plethora of disasters related with crowds, Mexico, Espinosa, ISBN: 978-136-553, pp. 29–41 (2016)
3. Google Developers.: Overview Android/NDK iOS Unity Unreal Get Started VR View Controller Support Spatial Audio VR Entry Flow VR Android Manifest VR Video Samples and Tutorials Downloads Release Notes Content Features Google VR SDK for Unity (2017)
4. Google Developers.: VR view Sample Code. Website: https://developers.google.com/vr/android/samples/vrview (2016)
5. Johnston, L. E.: A Graphical Input Device and Shape Description Interpretation Routines. In Memorandum to Prof. Mann, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, Mass (1960)
6. Van den Berg, J P., et al.: Optical Reciprocal Collision Avoidance. University of North Carolina, Website: http:// gamma.cs.unc.edu/ORCA (2010)
7. MakeHuman team.: MakeHuman for education. MakeHumanTM, Website: http://www.makehuman.org (2016)
8. Olivera, C.: Augmented Reality and Virtual Reality: The king is naked! Creative Commoms, Website: http://carlosoliverablog.com/2016/08/realidad-aumentada-y-realidad-virtual/ (2016)