Design of an interactive game-based application for improving the resolution of common Computer Network issues

Mario Ciau[1], Lizzie Narváez[1] and Victor Chi[1]

**[1] Universidad Autónoma de Yucatán, Facultad de Matemáticas, Unidad Multidisciplinaria Tizimín
Tizimín, Yucatán 97700, México**

Abstract

The need to find new ways to help students in the learning process makes us more involved with technology which evolves every day. This background leads us to find new methods and forms of teaching that help students accelerate their learning skills in a different way from the traditional, as in the case of "learning through play", a concept that has been used in many games which have had great impact recently, becoming an innovative tool for learning. This paper shows the design of an interactive game-based application developed with Blender 2.49b tool to be implemented in the area of computer networks.

Keywords: *Game-based, Blender, Design, Network Addressing.*

1. Introduction

It is undeniably true to say that using educational games substantially supports the teaching/learning process [1]. For this reason, lately, the advances in technology have become apparent in diverse aspects of life in general and within the educative environment particularly.

Traditionally, the training received in school has focused on the transmission of knowledge. Schools do a reasonably good job of preparing students in mathematics, social sciences and other disciplines. However, they have not done well in developing students' skills in areas such as problem solving, communication, teamwork, and collaborative learning. The teachers have to find ways to provide students with meaningful experiences through which these students can develop skills in the context of a particular subject. Video games can provide opportunities to students to develop these skills and therefore teachers can take advantage of informal apprenticeships which are already developing through play experiences [2].

A game can be the starting point for a progressive approach to an issue, a problem or a question that interests everyone, or is part of the goals that teachers are interested in addressing. This approach should be based on the students' initial conceptions, and should pave the way for the construction of more elaborate problems, theories and examination or investigation practices. Video games facilitate these practices bringing us closer to the issues that we want to work on and enabling a progressive questioning by a visible discussion of the challenges, theories, scientific facts and research reports [2].

A lot of work has been done in the field of the educational videogame development, bearing in mind the benefits of the learning process while playing [3].

Video games are the games of the XXI century and are becoming increasingly present in diverse social and cultural environments [4]. A video game is a computer program created specifically for fun, based on the interaction between a person and a machine running the game. These recreate virtual environments in which the player can control a character or other element of the environment to achieve one or more objectives through certain rules [1].

To create and develop games, we must have a game engine which is a software system designed for this purpose. There are engines designed to create games for consoles, and others designed to create games for PC; some of them are multi-platform and are designed to allow multiple consoles and PC [5].

The typical functionalities provided by a game engine are [5]:

- A rendering engine, 2D or 3D to draw the game graphics.
- A physical engine and collision detection to simulate properties such as gravity and detect collisions between objects, contact with the ground, forces simulation, etc.
- Support for Sound (both editing and integration in the game), animation (skeletal animation), scripting, networking, and multithreading.
- Artificial Intelligence, streaming and memory management.

Game engines provide a working environment that integrates all the above features usually with a visual interface that makes it easier to design and program a video game. These engines abstract the logic of the game so much that several different games can be done with the same engine, thus reducing cost, complexity and development time, all critical factors in the video game industry.

The process of design and development of virtual environments can be supported by tools and frameworks, to save time in technical aspects and focus on the content. [6] Presents an academic framework which provides several levels of abstraction including integrated components adopting open-source solutions in order to face specific problems.

Constantine and Lockwood describe a collection of principles for improving the quality of user interface design. Between these principles are the structure principle and the simplicity principle. In the structure principle, the design should organize the user interface purposefully, in meaningful and useful ways based on clear, consistent models that are apparent and recognizable to users, putting related things, differentiating dissimilar things and making similar things, differentiating dissimilar things resemble one another. In the simplicity principle the design should make simple, common tasks simple to do, communicating clearly and simply in the user's own language, and providing good shortcuts that are meaningfully related to longer procedures [7]. Both principles are considered in the user interfaces of this work.

The software tool used in the development of this work is Blender 2.49b.

2. Blender

Blender is an integrated suite of tools that provides everything needed to create animated videos, and video games using a graphical interface that abstracts the designer from the program, allowing him/her to do everything in a visual way. It is totally free, open source, and among its main characteristics we can find the following [5]:

- Interface: Based on non-overlapping windows. Completely configurable.
- Modeling: very powerful tool, with capabilities similar to the best commercial tools. It has all the features needed for a professional modeling.
- Animation: A very useful feature is that you can store the "frame" that is walking in an animation and access it from a Python script to perform actions at different points in the execution of the animation.
- UV Texture mapping: This tool allows us to unroll a mesh over an image, using different methods of unwinding. Subsequently it is possible to move the vertexes of the unrolled mesh to fit the pattern of the texture image.
- Physical and particles: Allows you to simulate fluids, soft and rigid bodies, and alter its physical properties for different visual effects. The physics engine includes collision detection and particle field effects like wind and vortices.
- Real-time 3D game creation: Provides a graphical interface for inserting logic in video games without programming. For more advanced control of the game, the graphic interface allows users to incorporate Python scripts as controllers, and has the library "GameLogic.py", which is a very complete implementation of these drivers.
- File management: You can save all the information for a scene in a file with the extension ". Blend" including the images used as textures. Blender files provide forward and backward compatibility; support compression, encryption, and can serve as a library for other ".Blend" files.
- Multiplatform support: Blender supports several platforms and its interface remains the same for each, such as Windows 2000, XP, Vista, Mac OS X, Linux (i386 and PPC), FreeBSD 5.4 (i386), SGI Irix 6.5 and Sun Solaris 2.8 (sparc).

2.1 Advantages and Disadvantages when compared to Commercial Software

Among the advantages of Blender when compared to commercial software (Inventor, Rhinoceros, 3D Studio etc.) are the following [8]:

- The quality of both the final model and the animations is similar or superior to many commercial tools.
- No cost per use or per license, i.e. we can download the full version from their website and modify the software for our needs, because its source code is also available.
- It has an Armor-making tool, which allows human movements to be executed naturally.
- It has a powerful engine for games.

However, Blender also has some disadvantages compared to commercial software; among them we can mention the following:

- It does not perform simulations.
- The user interface is not very friendly at first, but this disadvantage can be overcome by practicing with the software.
- It does not work with Reverse Engineering.

- It requires learning of several key combinations.

Despite the above, these disadvantages do not negatively affect the final product due to the design features of the games described in this paper. The usefulness for improving the process of student's learning for certain concepts in the area of computer networks is not lost.

3. Design of the game-based tool

3.1 The concept

One of the factors involved in the evolution of game design is the progress in the technology that supports them. You can easily illustrate this by following the evolution of design alongside the evolution of graphics in video games: they began by being extremely simple, text-based or with very low resolutions (highly limited vector graphics), they evolved over time to be capable of expressing two-dimensional shapes and increasingly complex animations up to expressing three-dimensional forms and making a progression in terms of the complexity of these forms [9].

Seen from the design perspective, this can be understood as a release from creative restrictions: initially all games were limited in their expression to the use of certain types of forms, then to the use of more expressive but simple forms, then to the use of more complex but still two-dimensional forms, then to the use of three-dimensional but still realistic forms, and currently capable of expressing almost-photo-realism [9].

The traditional way of teaching students in the area of networks, specifically on the issues of converting decimal numbers to binary and vice versa, identifying the class of an IP address, identifying the type of IP address (public or private), and determining the network type of an IP address (host, network or broadcast) can frustrate the student and make him/her feel overwhelmed by handling many numbers to understand the concepts.

We also know that games can be a great tool for promoting learning by providing alternative methods to help us overcome the barriers imposed by traditional learning. Studies in Chile, Mexico and Spain show the results of applying this technology to the learning process [10].

Bearing in mind these studies, we decided to make use of this technology, in particular we proposed the design of games to help students to make the learning of the subjects more entertaining and dynamic, and to help them achieve the central purpose of the game: practicing, identifying and the expediting of the diverse activities that will lead to mastery of the subjects mentioned above.

This game was developed to support the learning process in the area of computer networks for the Bachelors of Computer Science at the Tizimín Multidisciplinary Unit (Mexico). The intention is that the game will be as attractive as possible for the students, we thought about creating a kind of casino using games as slot machines, target shooting, cards and billiards. The game consists of 4 different components which are described below:

- Game 1: *Binary-Decimal Converter*, allows the player to enter a decimal number and a binary number to help him verify whether the entered decimal number corresponds to the binary number. This procedure is also done in reverse and its design corresponds to a machine similar to a casino slot machine.
- Game 2: *Classes*, allows the player to pick an IP address and choose its corresponding kind of network through target shooting game, which displays a result of right or wrong.
- Game 3: *Public or Private*, gives the player an IP address using a random function. He must choose whether the address is public or private through cards like in poker.
- Game 4: *Network-Host-Broadcast*, allows the player to pick an IP address and choose its address type (network, host or broadcast) by hitting billiard balls to their corresponding holes.

These games can be chosen from a main menu. Each of them has its own options that allow the player to start the game, see the help, know about the game, or exit. See Figure 1.

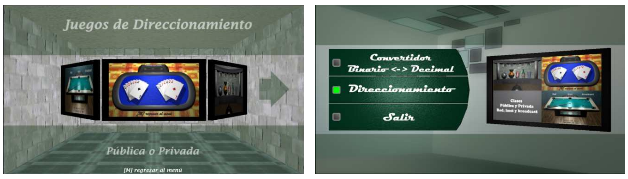


Fig. 1 Main menu.

3.2 Game 1

Game 1 is the Binary-Decimal Converter. This game is a slot machine with two sections: one to insert a decimal number and one for the binary number. Three positions were placed on top of the slot machine to select the decimal digits (this decimal number must be in the range 0-255).

The bottom of the machine was used to form a binary number placing eight dice (0-255 binary numbers can be formed with 8 bits) with the digits 0 and 1 on the faces of the dice. Players choose between these two digits and have the corresponding binary number.

To assist the player we placed a decimal number below each die according to its position. This game has a control panel which placed the letters on the keyboard and assigned them with their corresponding die. See Figure 2.

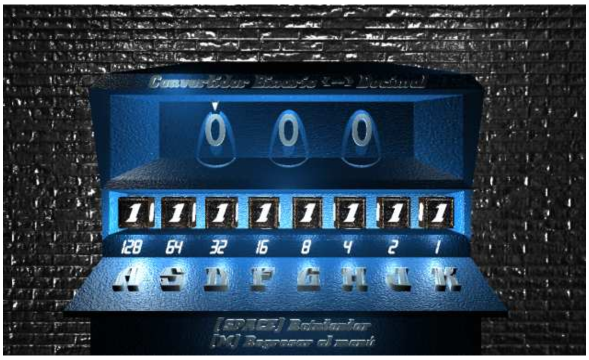
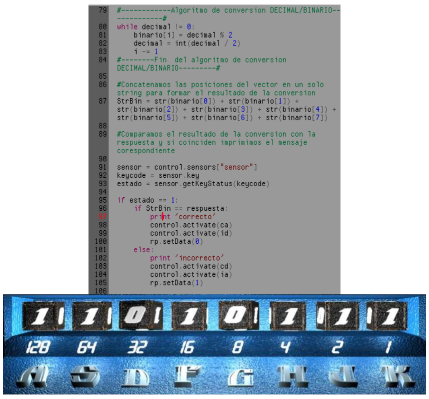


Fig. 2. Interface of game 1

Una vez definido lo anterior se procedió a agregar colores y texturas con relieve para simular volumen a algunos objetos mientras que a otros solamente se les aplicó color. Activando la opción Blender GLSL Materials y usando el conjunto de lámparas para la iluminación se logró imprimir un toque de realismo al juego y contar con una mejor interfaz. Al tener listo el diseño del juego lo siguiente fue programar el algoritmo que permita recibir los datos introducidos por el jugador, realizar la conversión y comparación de éstos y arrojar un resultado donde el jugador pueda comprobar si lo que hizo fue correcto o incorrecto. Los dados debían tener movimiento para poder mostrar el 1 ó 0; así que se les agregó una animación y con la programación de los bloques lógicos se logró controlar los dados con el teclado, de igual manera para elegir la posición y el dígito se creó un script que permita manejar los objetos y mostrar el dígito correspondiente para que luego por medio de los bloques lógicos hacer que ese script funcione al pulsar las teclas de dirección. Ver figura 3.



```
#------------Algoritmo de conversion DECIMAL/BINARIO--
-----------#
while decimal != 0:
    binario[i] = decimal % 2
    decimal = int(decimal / 2)
    i -= 1
#--------Fin  del algoritmo de conversion
DECIMAL/BINARIO---------#

#Concatenamos las posiciones del vector en un solo
string para formar el resultado de la conversion
StrBin = str(binario[0]) + str(binario[1]) +
str(binario[2]) + str(binario[3]) + str(binario[4]) +
str(binario[5]) + str(binario[6]) + str(binario[7])

#Comparamos el resultado de la conversion con la
respuesta y si coinciden imprimimos el mensaje
corespondiente

sensor = control.sensors["sensor"]
keycode = sensor.key
estado = sensor.getKeyStatus(keycode)

if estado == 1:
    if StrBin == respuesta:
        print 'correcto'
        control.activate(ca)
        control.activate(id)
        rp.setData(0)
    else:
        print 'incorrecto'
        control.activate(cd)
        control.activate(ia)
        rp.setData(1)
```

Fig. 3. Script and Interface of binary numbers (Game 1)

3.3 Game 2

The interface of this game has the form of target shooting (Figure 4). First a stand was modeled and different objects (ball, jar, can, bottle and glass) were placed on it. Two columns were added to this stand using cubes and cylinders. Then each of the mentioned objects was modeled using a sphere, a cylinder and bezier curves and was put in its place adding a label with its corresponding letter indicating the network class to allow the player to choose which of them to throw. Next, a gun was modeled and planes were added to form the walls and floor, at last the IP address was added with an appropriate size at the base of the stand so that the player can view it. Colors and textures were added to objects to make them more eyecatching. Transparency was applied to some objects like the bottle and the glass to simulate real glass, and a normal texture was applied to the walls to simulate volume (with the 'Blender GLSL Materials' option activated for realism). Finally bright lights were added to this game which helped obtain extra realism. See Figure 5.

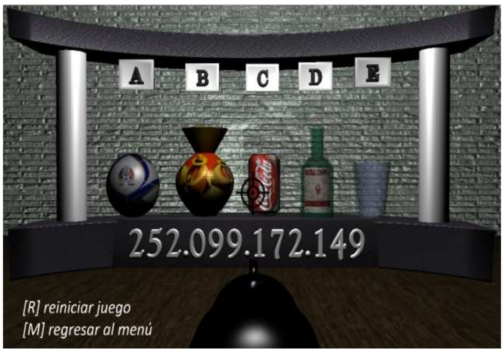


Fig. 4. Interface of game 2.

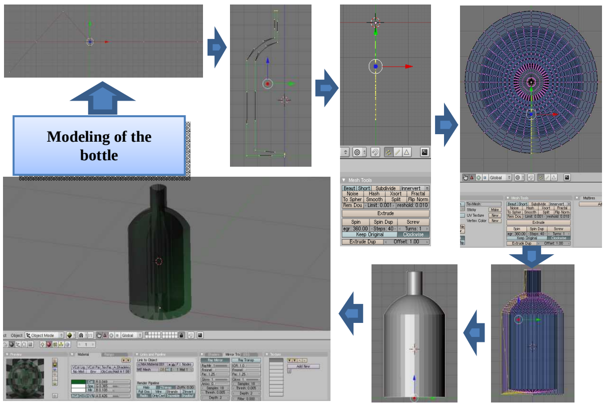


Fig. 5. Modeling the bottle

After finishing the design the game scripts were developed. These include the script that controls the display of IP addresses each time the player press the ENTER key, the script that collects the information, and another that compares and returns a result.

The objects (ball, jar, can, bottle and glass) are active actors because when they make contact with the bullet they have to simulate a falling motion. See Figure 6.

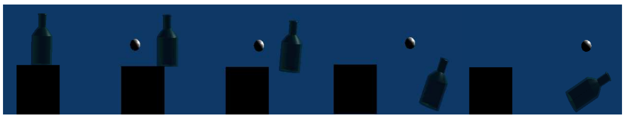


Fig. 6. Simulation of a fall.

3.4 Game 3

This game has the interface of a poker table with two hands of cards, retaining the approach that was taken from the casino theme. In the first instance the table was modeled using a cube to form the edge and a plane to form the background, we then created the cards that were basically planes scaled to an appropriate size and textured with edited images with a specific letter to form the words public or private when the hand is opened. At the end they were ordered and placed in their positions (in the center of the table). Also we needed to show an IP address for the player so a board was modeled where this address appears. The IP address was formed by adding sixteen digits (arranged in four groups of four elements) divided by a decimal point. This effect is handled by a script designed to show a different IP address when the player presses the ENTER key. Eventually we added the floor and a more colorful interface; also color and textures were added to objects with the option 'Blender GLSL Materials' enabled and lighting created realism. See Figure 7.

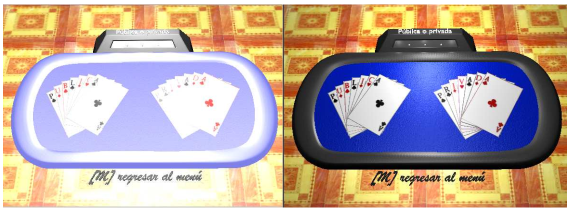
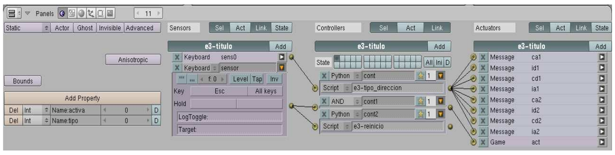


Fig. 7 Interface of the Game 3 without/with Blender GLSL Materials.

Subsequently, an algorithm was implemented to indicate whether the chosen IP address were public or private and tell the player if his/her answer was correct or incorrect. With ready-made scripts and using logic blocks the game's functionality was achieved. See Figure 8.



Del Int Name:activa 0 D
Del Int Name:tipo 0 D

Keyboard sens0
Keyboard sensor
f: 0 Level Tap Inv
Key Esc All keys
Hold
LogToggle:
Target:

State All Ini D
Python cont 1
Script e3-tipo_direccion
AND cont1 1
Python cont2 1
Script e3-reinicio

Message ca1
Message id1
Message cd1
Message ia1
Message ca2
Message id2
Message cd2
Message ia2
Game act

Fig. 8. Using logic blocks.

3.5 Game 4

In the same vein, a pool table was created as an interface for this game, where the player must enter an IP address and choose one of three options (Network, Host or Broadcast) hitting the ball with the cue to get it into one of the three holes. The first thing modeled was the table using cubes to form its edge and its base, also a plane for the background and cylinders were modeled and Boolean operators were used to form holes.

The balls were formed with spheres scaled to an appropriate size, the cue was made with a cylinder and the board and the IP address in the same way as in the previous game. As soon as the objects were built a texture was added to give them realism and to make the interface more attractive. Also the option 'Blender GLSL Materials' was activated and lamps were added to obtain the desired realism (Figure 9).

With the finished design an algorithm was implemented to know whether the presented IP address was from a network, a host or a broadcast. With this and other implemented scripts in the algorithm the required functionality of the game was obtained.

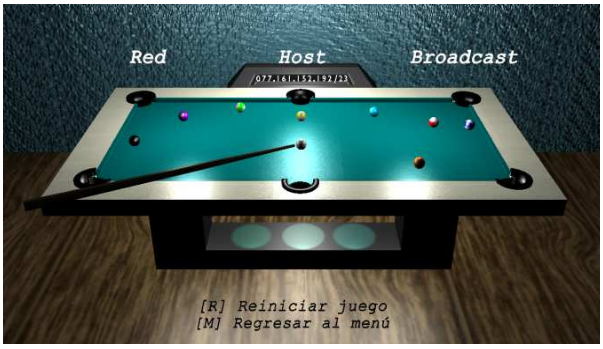


Fig. 9. Interface of Game 4.

4. Conclusions

Through the practice of gaming the student acquires the necessary skills to work successfully in digital environments that are proliferating in all fields of our society. This puts gamers at an advantage over non gamers when facing interfaces for the different technological elements that are found in everyday life, enabling them greater success in the tasks they face and thus making them more competitive [2].

This research presents the design of a based-game application for improving the resolution of common Computer Network issues. This includes the design of each of the games that make the Casino game using Blender 2.49b as a development tool. We also demonstrated how video games have become involved in the teaching/learning process by making it more entertaining; additionally diverse functions of the free software called Blender are presented, this free software proved to be just as effective as its commercial counterparts and it has allowed us to develop projects like this videogame.

The use of video games in education is helpful for both teachers and students, but for the development of these there are several options to consider when choosing the most suitable software.

Applying technology in the teaching/learning process and exploiting the fact that most young people are enthusiastic about videogames we can obtain valuable tools that help us in the way we teach and learn a specific issue as shown here in the field of computer networks.

This game will be provided in the near future to students taking the Networking module of the Computer Science degree to support their learning process.

References

[1] Padilla, N, et al. “Diseño de Videojuegos Educativos Multijugador. Una Visión desde el Aprendizaje Colaborativo Visión desde el Aprendizaje Colaborativo.”, in Interacción 2008, 2008.

[2] Begoña, S. “Certezas e interrogantes acerca del uso de los videojuegos para el aprendizaje”. Comunicación, 1(7):251-264 p., 2009.

[3] Madera, Francisco, Uicab, Rocío and Basto, Luis, “Virtual Chichen Itzá”, in Segundo Encuentro Mexicano de Realidad Aumentada: ERA2011, 2011.

[4] González, J, et al. “De la Usabilidad a la Jugabilidad: Diseño de Videojuegos Centrado en el Jugador”, in Interacción 2008, 2008.

[5] Peral, Roberto, “Creación de un manual para el diseño e implementación de videojuegos en Blender 2.49b”, Universidad Carlos III de Madrid, Escuela Politécnica Superior. España, 2010.

[6] Catanase Salvatores, Ferrara Emilio, et al. A Framework for Designing 3D Virtual Environments. University of Messina, Italy. 2011.

[7] L.Jayasimman, A.Nisha Jebaseeli, Dr.E.George Dharma and J. charles. “Dynamic User Interface Based on Cognitive Approach in Web Based Learning”, International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011.

[8] Chung Alfonso, “Software libre aplicado al dibujo industrial: el caso Blender”, Sistema e Informática Vol 12, No. 1, Year 209, pp. 62-67.

[9] Moraldo, Hernán. “Desafíos y tendencias en el diseño de videojuegos”, Comunicación, Vol.1, No. 7, 2009, PP. 6-15.

[10] González, José, Cabrera, Marcelino and Gutiérrez, Francisco, “Diseño de Videojuegos aplicados a la Educación Especial”, in Interacción 2007, 2007.

Mario Ciau-Uitzil He studied Computer Science from the Autonomous University of Yucatan – Multidisciplinary Unit Tizimín. (UADY) in 2011. Currently he is developing systems and working in animation.

Lizzie Edmea Narváez-Díaz. Received a degree in Computer Science from the the Autonomous University of Yucatán (UADY) in 1997. She received a Master of Computer Science degree from Monterrey Technological Institute (ITESM), Campus Cuernavaca, in 2007. She has been a full time teacher at the Autonomous University of Yucatán since 2000 in the Networking department in Tizimín México. She has participated in software engineering development projects. Currently she is giving courses on networks in the professional programs in the UADY, she is CCNA.

Victor Manuel Chi-Pech. Obtained his degree in Computer Science from the Autonomous University of Yucatan (UADY) in 1996 and his M. Sc. degree in Wireless Network from Monterrey Technological Institute (ITESM), Campus Cuernavaca, in 2007. Victor Chi has worked since 2000 at the Autonomous University of Yucatan, as a full time professor. He has participated in software engineering development projects. He is currently giving courses on wireless networks and software engineering in the professional programs in the UADY, he is CCNA