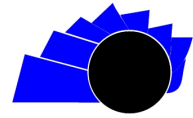







UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS



Design of a module in Opensim for the development of competences in Information and Communication Technologies for university students

Diseño de un módulo en Opensim para el desarrollo de competencias en Tecnologías de la Información y la Comunicación para estudiantes universitarios

Roberto Ferro-Escobar ¹, Danilo Vera-Parra ², Astrid Ramírez-Valencia ³

INFORMACIÓN DEL ARTÍCULO

Historia del artículo:

Enviado: 21/11/2022

Recibido: 13/12/2022

Aceptado: 28/01/2023

Keywords:

OpenSim

Sloodle

ICT skills

Integral development

Metaverse

Sloodle

3D



Palabras clave:

OpenSim

Sloodle

Competencias TIC

Desarrollo integral

Metaverso

Sloodle

3D

ABSTRACT

This article proposes the implementation of a module within the Open Simulator platform, for the development of competencies in Information and Communication Technologies for the integral development of university students that will allow them to succeed in their labor field in this type of tools that allow the development of different skills in soft skills and strong skills, this tool allows the simulation of different training environments within virtual worlds.

RESUMEN

El presente artículo propone la implementación de un módulo dentro de la plataforma Open Simulator, para el desarrollo de competencias en Tecnologías de la Información y Comunicación para el desarrollo integral de estudiantes universitarios que les permita tener éxito en su campo laboral en este tipo de herramientas que permiten el desarrollo de diferentes competencias en habilidades blandas y habilidades fuertes, esta herramienta permite la simulación de diferentes ambientes de entrenamiento dentro de mundos virtuales.

1 Universidad Distrital Francisco José de Caldas, Colombia. E-mail: referro@udistrital.edu.co

2 Universidad Distrital Francisco José de Caldas, Colombia. E-mail: danieloverap@gmail.com

3 Universidad Distrital Francisco José de Caldas, Colombia. E-mail: aramirezv@udistrital.edu.co

1. Introduction

Today, one of the most relevant topics in computer-based content management research efforts is how to present and visualize content, moving beyond the simple static web delivery of documents, books and publications over HTTP on the Internet. Today, users demand better ways to access new sources of information, and proof of this is the current evolution of digital environments into virtual worlds, which can be used to implement different types of training within these 3D worlds.

Thanks to the dizzying progress of bandwidth and the high speeds of current networks, it is possible to implement virtual world simulators that implement avatars in these 3D environments, such as Open Simulator [1].

This simulator was created under the Secondlife guidelines, released in 2007 by Linden Labs [2] and placed under the GPL license [3]. OpenSim complies with the general client server model and has a modular structure written in C# and supports several viewers, such as the HIPPO viewer [4], the RealXtend viewer [5] and Secondlife's own official viewer.

The dynamic nature of Web 3.0 [6] combines various techniques of artificial intelligence, geospatial mobility, mashups, interactivity, multimedia and 3D transformations to build a semantic database capable of learning. Secondlife was initially used as a game in a virtual world built on a commercial server, and after some years, due to the large number of worlds that can be created, Linden Labs decided to release the code to allow an open construction of virtual worlds and private islands by programmers, nowadays it is not necessary to be a programmer to create these islands and create different applications that allow to simulate trainings in different fields of knowledge.

One of the most promising areas of research related to 3D virtual worlds technology is the study of the integration of these 3D virtual worlds with current virtual learning environments and content management systems, given the powerful emergent properties that arise from such a combination that can develop ICT competencies of university students and show the benefits of these virtual platforms, aspects that make the construction of these virtual universes very attractive and can help to reduce the university desertion of many students who consider very tedious the development of Web applications, the access to databases and the development of interactive 3D type platforms, besides this, virtual worlds can be applied to the development of future educational projects in them that integrate different subjects.

2. Pedagogical theories and their application to constructivist learning

It is necessary to understand the following aspects related to pedagogical teaching related to the different theories organized chronologically according to the different authors who have worked on these approaches and the given approach:

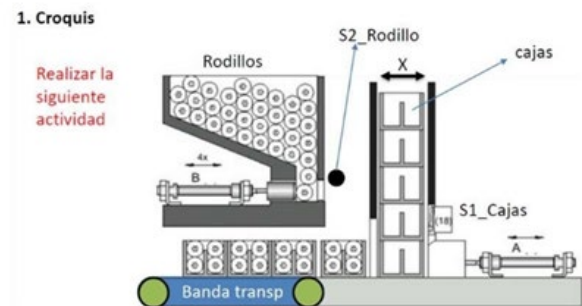
- Jean Piaget (1896-1980): Developed the theory of cognitive development, which emphasizes how children construct their understanding of the world through assimilation and accommodation of new experiences [7].
- Lev Vygotsky (1896-1934): A pioneer in sociocultural theory, he emphasized the role of the social environment and interaction in learning, and coined the concept of the "zone of proximal development" [8].

- Jerome Bruner (1915-2016): Proposed the theory of discovery learning, which emphasizes the importance of problem solving and active exploration in the learning process [9].
- Seymour Papert (1928-2016): Developed the concept of "constructionism," an extension of the constructivism that focuses on learning through creation and construction [10].
- David Ausubel (1918-2008): Known for his theory of meaningful learning, which is based on the idea that learning is most effective when new knowledge is meaningfully related to prior knowledge [11].
- Howard Gardner (1943): Although he is best known for his theory of multiple intelligences, he has also addressed constructivism by highlighting the diversity of learning styles and the importance of adapting teaching to individual abilities [12].
- Jean Lave (1942): Known for her work on situated learning theory, which focuses on how learning occurs in specific social and cultural contexts [13].
- Ernst von Glasersfeld (1917-2010): Developed radical constructivism, which holds that knowledge is a subjective mental construct and that reality is an individual interpretation [14].

For example, Seymour Aubrey Papert [15], through the programming language LOGO, was able to promote the active construction of solutions and the understanding of programming through the creation of applications by means of this program.

Figure 1. Example of PLC programming using the LOGO program used by Siemens [16].

Programación PLC LOGO Siemens Graficet a Ladder



3. Development of TIC competencies

Regarding the development of ICT competencies, it is important to address the following aspects:

Technical Skills:

- Programming: Proficiency in at least one programming language, with the ability to solve problems and create applications.
- Web development: Knowledge of front-end (HTML, CSS, JavaScript) and back-end (databases, servers) technologies.
- Networking and Security: Understand how networks work, protocols, configuration and cyber security measures.
- Databases: Know how to design, administer and query databases, including SQL and NoSQL databases.
- Application Development: Create applications for mobile and desktop devices.
- Artificial Intelligence and Machine Learning: Basic knowledge in machine learning algorithms and artificial intelligence techniques.

Conceptual Knowledge:

- **ICT Fundamentals:** Understanding the basic concepts of computer science, operating systems, computer architecture, etc.
- **Cybersecurity:** Awareness of cyber threats, security policies, encryption and best practices.
- **Cloud Computing:** Familiarity with cloud services, such as AWS, Azure or Google Cloud.
- **Big Data and Analytics:** Knowledge on handling and analyzing large volumes of data.
- **Internet of Things (IoT):** Understanding how devices connect and communicate in a network.
- **Virtual and Augmented Reality:** Awareness of these emerging technologies and their applications.

Soft Skills:

- **Problem Solving:** Ability to address technical challenges creatively and effectively.
- **Critical Thinking:** Evaluate information and make informed decisions.
- **Technical Communication:** Express technical ideas in a clear and understandable way to both experts and non-experts.
- **Teamwork:** Collaborate in multidisciplinary projects and communicate with other professionals.
- **Adaptability:** Be willing to learn and adapt to new technologies and approaches.
- **Ethics and Responsibility:** Understand the ethical and legal implications of technology and act responsibly.

To develop an Information and Communication Technology (ICT) portal, a student must acquire a variety of technical skills and conceptual knowledge related to web design, programming, content

management and usability. Here is a guide on what a student should learn to make an ICT portal:

Web Design and User Experience (UX):

- Visual design fundamentals and UX design principles.
- Use of design tools such as Adobe XD, Sketch or Figma.
- Responsive design so that the portal works well on different devices.
- Interaction design and intuitive navigation.
- **Front-End Programming:**
 - Markup languages: HTML for structure and CSS for style.
 - JavaScript for interactivity and dynamism on the page.
 - Frameworks and JavaScript libraries, such as React, Angular or Vue, to speed up development.
- **Back-End Programming:**
 - Server-side programming languages, such as Python (with Django or Flask), Ruby (with Ruby on Rails), PHP or Node.js.
 - Database connection and data management.

Databases:

Relational (e.g., MySQL, PostgreSQL) or NoSQL (MongoDB) database design and administration.

Queries and data manipulation using SQL or other tools.

- **Content Management:**
 - Content Management Systems (CMS) such as WordPress, Joomla or Drupal to facilitate content creation and management.

- Customization and development of modules or add-ons.
- Security and Data Protection:
 - Implementation of security measures, such as data encryption and user authentication.
 - Awareness of vulnerabilities and good web security practices.
- Optimization and Performance:
 - Performance optimization techniques to ensure that the portal loads quickly and efficiently.
- Testing and Debugging:
 - Ability to test and debug the portal on different browsers and devices.
- Identification and correction

This article proposes the fundamental bases for the design of a module in Opensim for the development of competences in Information and Communication Technologies ICT for university students.

4. Tools to create

Metaverses

OpenSim is currently used to easily design, create and implement 3D virtual worlds by a large base of programmers around the world. With this tool it is possible to create content aimed at different areas, such as e-learning, virtual tours, simulations, among others. Moreover, being a free software tool itself leads to low-cost development and allows developers to contribute to the development of new features.

A. OpenSim 3D Modeler

OpenSim has three main operational instances, defined as "modes": Standalone, Grid and Hypergrid.

The OpenSim code is written in C#, which is more generally compatible with Mono, an open source .NET platform. These modes are described below:

- Standalone mode: the client interacts with a server and the client user authenticates to that server before being placed in the world. The server contains all the basic services included in an executable image that invokes dynamically linked libraries as needed. These services are described separately as follows.
- Grid mode: This mode includes specific abbreviated services such as UGAIM. Each letter in this sentence represents a separate service: User, Grid, Active, Inventory and Messaging. The User service handles authentication and knows which users log in to the network. The Grid service knows the general layout of the rectangular grid, including the Internet addresses of the endpoints for each region. A grid is composed of regions in the same way that a floor contains a grid of tiles. The entire grid comprises the "world". The asset service manages all assets in the regions. Examples of assets include basic geometry, texture maps, audio files, and terrain geometry. The inventory server manages which assets are linked to a particular agent, so that when a user authenticates as an agent, that agent will "own" a set of inventory items. The messaging server handles text chat messages.
- Hypergrid mode: A hypergrid is a collection of loosely connected simulators without an overall network manager. In many ways, it resembles the hypermedia of the Internet. The web hypergrid equivalent for OpenSim is teleporting from one region to another. Hypergrid concepts are more complex than on the web, as teleportation requires significantly more message exchange and updating of region information.

Figure 2. View of an OpenSim Avatar virtual world simulator.



Source: Own.

B. Moodle for Web courses

Moodle (Modular Object-Oriented Dynamic Learning Environment) is an open-source software package for the creation of courses and web sites based on PHP and designed to work with different database engines, especially MySQL [9].

Moodle is intended to help educators easily create and manage online learning communities, and is designed to encourage knowledge construction by students.

Its first version was released in August 2002 and to date the current version is 2.3. Moodle currently has more than 6,856,000 courses offered through 72,164 registered sites in 223 countries, with Spain occupying the second place in the world.

C. Sloodle Project

The SLOODLE (Simulation-Linked Object-Oriented Dynamic Learning Environment) Project

[10] is a successful code that integrates multi-user virtual environments such as Second Life and OpenSim with the MOODLE learning management platform. Sloodle provides a wide range of tools (some still under development) that enable virtual learning worlds and immersive education, and are fully integrated with a web-based learning management system used by thousands of educators and students worldwide.

D. Operating System D.Ubuntu

Ubuntu is an operating system that uses a Linux kernel and based on Debian [10], it is maintained by the British firm Canonical and the community of developers. This distribution aims to improve the end-user experience, facilitating the interaction with the operating system through an improved GUI. It is one of the most popular distributions today, with approximately 50% share among Linux distributions.

For the development of this project, it should be noted that Linux is widely known as an operating system designed from the beginning as both multithreaded and multiuser, so it does manage very securely the permissions of each user over processes and files, preventing users from accessing each other's information. Ubuntu has disabled the "root" user by default, so the system repeatedly asks for the user's password to install components and forces the user to take special care with permissions to modify files and folders.

5. Installation of metaverse in OpenSim

The following is a brief summary of the steps required in the Ubuntu distribution to assemble a metaverse in OpenSim and link it to the Moodle platform.

The server can be built on different platforms such as Windows, Mac and Unix. The procedure for Windows is described below:

- a. Get Visual C# Express, which can be downloaded for free from the Microsoft website. Mono can also be used for Windows but is required for Mac and Unix installations.
- b. Get the OpenSim distribution from the instructions at: <http://opensimulator.org/wiki/Download>
- c. Follow the compilation instructions at: http://opensimulator.org/wiki/Build_Instructions.
- d. The default installation runs a server at the address

127.0.0.1 (localhost). To run the server so that other clients can access the server, it is necessary to use the external IP address of the server machine.

More detailed instructions for building OpenSim can be found on the web pages above. In practical terms, OpenSim runs well on most computers, including laptops, and performance problems arise mainly when there are too many objects stored as assets or when there are too many agents, or clients, trying to access the server; most regions can support 20 avatars at a time. Differences in computer architecture and machine capabilities certainly have an effect on performance. For example, more memory can provide a significant performance improvement with more avatars and objects possible in the virtual world being created.

A. Installation of the XAMPP platform

Since we need a web server to distribute Moodle contents and it is recommended to configure MySQL as the default database engine for an OpenSim server,

we will use the XAMPP [11] Apache distribution, which includes the MySQL installation. The XAMPP installation files for Linux are available from the official "Apache Friends" website. Once the file is downloaded, you must deploy it uncompressed in the "opt" directory. Finally, you must start the server using the command `"/opt/lampp/lampp start"`.

One of the most powerful utilities found in XAMPP is the PHP-based administration tool for MySQL databases "phpMyAdmin", which can be accessed by typing in a browser "http://localhost".

By default, the administration page does not have access control enabled. Therefore, anyone who has access to the server can change this valuable data. We recommend setting up passwords. The XAMPP documentation describes how to do this.

B. Installing the Metaverse world with OpenSim

In the OpenSim download page we can find the necessary files for the installation of the server. It is recommended to install the server on a 32-bit Linux version, since 64-bit versions usually have compatibility problems with OpenSim and its dependencies.

OpenSim is designed to run on the NET from Microsoft. As the code and scripts used in the metaverse are based on C#, we should install the Mono platform on Linux, which has the necessary libraries to compile those languages. The files needed to install Mono on Linux are available on its official download page or can be downloaded directly using the terminal installation commands (e.g., Ubuntu: `sudo apt-get install mono-complete`). After having installed mono opensim and unzipped the archive, we must open a Terminal. To start the server, simply type the command `"mono OpenSim.exe"`, with root permissions.

If there are no compatibility problems, the server should start correctly, loading one by one all the necessary dependencies to the metaverse. If this is the first time you are running OpenSim on the computer, you have no metaverse to start with, so the simulator prompts for the creation of one, requesting the following information:

- New region name []: alphanumeric character. Enter the desired name for the region.
- RegionUUID [default_UUID]: UUID. Unique identifier for this region. OpenSim proposes a default UID, so just press Enter unless you prefer a specific UUID.
- Region Location [1000.1000]: x, y coordinates. It is the point on a coordinate plane that you want for the region.

The mode must be specified for gridless overlay in the region of existing regions. Otherwise, you can accept the default point by pressing the Enter key.

- InternalAddress [0.0.0.0]: Wildcard mask (IP address).

This address is actually a wildcard mask that allows you to restrict the range of addresses that can access the server. In most cases, it should be left at the default value, so that all hosts can access our server.

- InternalPort [9000]: # port (0-65535). Internal server port to be used for server access.
- AllowAlternatePorts: True or False. This feature is still in experimental stage. The default value is false.
- External host name: IP address. This is the address used by clients to access the server, which must take into account some other considerations that will be discussed later on. To begin with, you can exit "SYSTEMIP".
- Do you wish to join an existing region to the state of Udin (yes / no)? [Yes]: This is used to

attach the newly created region to an existing state. If there is a previously created state which you want to attach to the region you should type its name, or else type no" and proceed to create one.

The parameters required to create one are:

- Name of the estate [MyEstate]: alphanumeric character. Any name.
- The following data create a new user on the server and be the first user to log in:
- Name of the owner of the property [Proof]: alphanumeric character.
- The name to use for the owner of the property.
- Surname of the owner of the farm [User]: alphanumeric character.
- The name to be used for the owner of the property.
- Password: alphanumeric character. Password of the created user.
- E-mail []: alphanumeric character. E-mail of the new user created.

Now we have a region, a state and a user. Then, we can run the server and it can be accessed from any viewer. These parameters can be modified at any time by accessing Regions.ini located in the bin/Region's directory within OpenSim. The text file appears as shown in Figure 2.

It is possible to change the customer directly from the screen.

The only important parameter to change is the login URI (the socket through which you access the server), it also allows viewers to save configurations and assign names and other descriptive parameters that are not of interest here. Here are some

configurations for the server and client, and network topologies.

Topology 1: the client and server are hosted on the same computer.

The client and server configuration shown in Table 1:

Table 1. Parameter settings for topology 1

Client	Internal Address	0.0.0.0
	ExternalHostName	SYSTEMIP
Server	Login Uri	http://localhost:port

Source: Own.

Figure 3. Regions view with location, internal address, port and the name of the UDIN server, the Regions ID is composed by a hexadecimal number unique in the world.



Source: Own.

C. OpenSim configuration for different network topologies

OpenSim is a simulator that runs on a server, i.e., it is not a local program like a video game in which the user starts a process on his computer and ends when he finishes using it.

The situation resembles that of an interactive web site where the user makes requests to the server and initiates or terminates sessions, accesses databases or performs specific actions where the processing occurs on the server (which may be in another location or on the same computer from which you connect the client). If you have a server, you must have a client with the necessary application to access its services. In case you have OpenSim Hippo viewers, such as OSV, RealXtend or

simply Second Life Viewer, these allow users to connect to the servers running the simulator. Since the client and server may be located on the same network or on separate networks, we will continue with a brief explanation of the basic configurations of the above parameters, so that the clients can access the servers.

Depending on the topology, we can have multiple address assignments (static or dynamic - depending on whether the server has DHCP enabled or not).

Changes to the server are made in the Regions.ini file, located in the bin/Regions folder within OpenSim. The only parameters that need to be modified for access are "ExternalHostname" and "Internal Address". The network topology is shown in Figure 4.

Figure 4. Topology 1, this computer hosts the server and client software, the virtual world development generator is inside.



Source: Own.

Topology 2: Client and server on the same network.

In this case, we have the client and server using the same network serving different IP addresses (usually within a network with private addresses).

D. Construction of objects in OpenSim

Creating and modifying objects in OpenSim is the basis for building a metaverse. The first thing to do is to increase the size of the island, this is achieved by

using the long "terrain fill" command with the number of the desired size. The region can have a maximum size in the grid, because if it is set to a larger size, it will overlap with other regions in grid mode.

- Vision (Ctrl + 1) is used to change the angle of view, focus and camera orbit.
- Move (Ctrl + 2): With this option you can move different objects in the region.
- Edit (Ctrl + 3): this is one of the most important tools; this tool can easily modify the position, rotation and size of objects.
- Create (Ctrl + 4): This tool creates different primitive objects such as cubes, cylinders, cones, spheres, prisms, toroids and others.
- Terrain (Ctrl + 5): With this tool you can raise, level, smooth or reset the selected field. With this tool you can also change the properties to specific "graphics", including the type of texture to be used for playing media.

Figure 5. OpenSim viewer, basic tool menus and tabs for the design of objects in virtual worlds.



Source: Own.

When you open any of the above-mentioned tools, you can make specific modifications to the objects you have just created, using the tabs at the bottom of the pop-up window.

The tabs available in the tool's menu are:

- General: here you can make a description and edit the permissions and ownership of objects

and define the action that is performed by default when an avatar touches it.

- Purpose: In this tab you can set "physical" properties of the object in the metaverse, such as position, rotation, size or even if the object is material, phantom, etc. The detailed description of the management of the client tools for building an OpenSim metaverse is not the subject of this paper.
- Features: edit the features associated with the Physics Engine Simulator. OpenSim accepts different physics engines that allow more realistic simulations of objects, but for the purpose of this project they are not important.
- Texture: each object can also have different textures associated with it, which create more realistic environments and are also used for multimedia playback within the metaverse.

The complexity of building a metaverse from scratch using basic objects and the time it takes will depend on the user's ability to use these tools. A detailed description of metaverse construction is not the subject of this document.

6. Link to the metaverse created with Moodle

If you have a Moodle platform installed on a client-accessible server, it can be linked to the OpenSim metaverse created with the Sloodle tools, so that users can use some of the Moodle features directly from within the metaverse. For this project, we installed Moodle on the same machine that hosts an OpenSim server, to perform some tests.

Sloodle Installation

The files necessary for Sloodle to work must be implemented on the server and on the user/client

machine. The server files are available for download on the official project wiki.

The unzipped folders should be copied into the following subfolders. It is important to remove any previous versions of Sloodle directories installed in Moodle before copying the new one.

- Module (required): You must copy the "Sloodle" folder to the "mod" folder inside Moodle.
- Block (Optional): You must copy the "Sloodle_menu" folder to the Moodle "blocks" folder.
- Sloodle plugin object assignment (optional): you must copy the "sloodleobjct" folder to the "mod/assignment/type" folder inside Moodle.

User files for OpenSim virtual objects are required to use Moodle functionality within the metaverse.

This is called the Sloodle set. Once downloaded, it must be loaded into the metaverse via the "load rem" command using the command line on the OpenSim server. Note that this loads a new region, so before doing so, it is advisable to save the region you are working on.

Once SLOODLE is loaded, the objects appear.

Therefore, Sloodle objects must be stored in the user's inventory. With the above, the modules are installed and Sloodle is ready to use with OpenSim.

7. Results

Seymour Aubrey Papert, is known for applying constructivism in the teaching of software development, specifically through the programming language LOGO, which promotes the active construction of solutions and the understanding of programming through creation. It is from this type of

pedagogical exercise that the foundations are laid for the use of OpenSim as a tool for learning ICT in a 3D virtual environment where imagination has no limits.

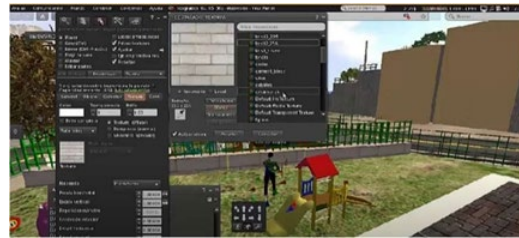
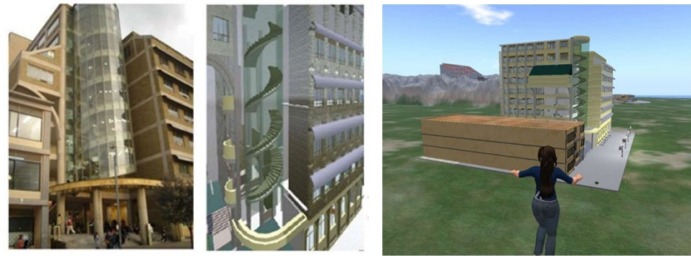
The application of constructivism in software development involves the creation of active learning environments where students build their understanding and skills through problem solving and active construction of solutions. OpenSimulator, as an open-source virtual worlds platform, provides an ideal setting for applying constructivism in this context. Developers can use OpenSimulator to create interactive simulations that allow students to tackle programming and problem-solving challenges in an immersive 3D environment.

This encourages active participation, hands-on learning and collaboration, which is essential in the formation of software development skills and knowledge acquisition in the field.

The main result is a metaverse that will be applied to the teaching of university students Universidad Distrital "Francisco José de Caldas" is hosted on a server accessible from the Internet, and tested for Video Streaming using Open Sim textures and mobile networks within the university campus.

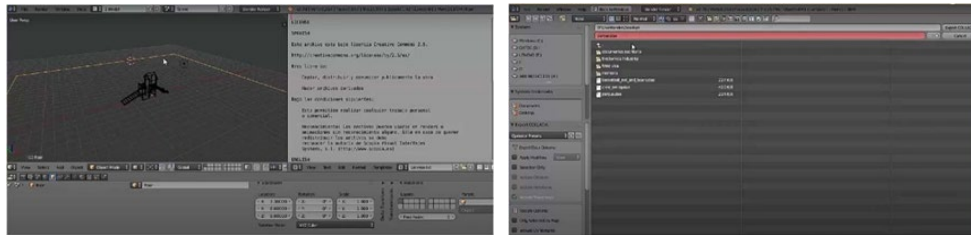
In Figure 6 you can see the photo of the real building of the Faculty of Engineering and, to the right in OpenSim the same building was constructed but inside a computer island, the experience on the part of the students was so successful that they were already interested in learning about network connectivity, installation of databases, creation of elements such as houses, classrooms, buildings, examples of this was the creation of Maloka, Monserrate, different branches of the university and even a park in a neighborhood in Bogota, this can be seen in the following figures.

Figure 6. Example of metaverse created at the Universidad Distrital, Engineering Department.



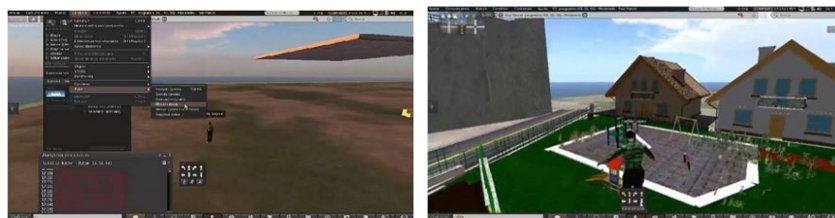
Source: Own.

Figure 7. Commands used within the 3D virtual metaverse in OpenSim.



Source: Own.

Figure 8. Development of El Paraíso zonal park in OpenSim.



Source: Own.

Figure 9. Maloka construction project in the 3D world of OpenSim photo downloaded from Google.



Source: Own.

8. Conclusions

Constructivism is a theory of learning that holds that individuals actively construct their own knowledge through interaction with information and their environment. There have been numerous authors and theorists who have contributed to the development and understanding of constructivism over the years.

This type of metaverses can be aimed at educational virtual environments and the development of ICT skills for university students. In order for a student to develop an Information and Communication Technology (ICT) portal, he or she must acquire a diverse set of skills and knowledge. This includes mastering web programming languages such as HTML, CSS and JavaScript, as well as understanding the creation of interactive interfaces using frameworks such as React or Angular. In addition, you should learn about databases, cyber security, user experience design (UX/UI), agile development methodologies, project management tools, and optimization and deployment techniques. Familiarity with content management systems, APIs, hosting, testing and debugging, as well as ethics and legal compliance, completes the skill set needed to successfully create and maintain an ICT portal.

OpenSimulator provides a hands-on environment for developing ICT skills through the creation and operation of virtual worlds. Students can learn programming, design, communication, collaboration and other essential skills while working on interactive and creative projects in this virtual environment.

References

- [1] G. P. Fishwick, "An introduction to Opensimulator and agent-based M&S applications of virtual environments," in Simulation Conference (WSC), Winter 2009 Proceedings, December 2009, pp. 177-183,64.
- [2] Linden Research, <http://lindenlab.com>
- [3] M. Barbulescu, M. Marinescu, O. Grigoriu, G. Neculoiu, V. Sandulescu, and I. Halcu, "GNU, GPL in the study of programs in the field of systems engineering," in Roedunet International Conference (RoEduNet), June 10, 2011, pp. 1 -4.
- [4] Hippo OpenSim viewer, available: <http://mjmlabs.com/viewer> Viewer RealXtend. <http://realxtend.org>
- [5] M. Pattal, Y. Li, and J. Zeng, "Web 3.0: a truly personal web! More opportunities and more threats," in Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST '09. Third International, Conference on, September 2009, pp. 125 -128. <https://doi.org/10.1109/NGMAST.2009.82>
- [6] McLeod, S. A; Piaget "Cognitive Theory". Simply Psychology. 2023.
- [7] J. P. Bronkard et al., "Vigotsky aujourd'hui," Paris: Delachaux & Niestlé, 1985.
- [8] J. Bruner, "Research on cognitive development," Spain: Pablo del Río, 1980.
- [9] S. Papert and I. Harel, "Situating constructionism," Alajuela: INCAE, 2002.
- [10] D. P. Ausubel, "Knowledge acquisition and retention. A cognitive perspective," Barcelona: Ed. Paidós, 2002.
- [11] N. Athanassopoulos, "Chapter 7: Comparative study of the development of multiple intelligences in students who do and do not study dance in a conservatory," Innovating in Education.
- [12] J. Lave, "Situating learning in communities of practice," in H. Resnick, S. Levine, & S. Teasley

- (Eds.), Perspective on socially shared cognition (pp.63-82). Washington, USA: American Psychological Association, 1991.
<https://doi.org/10.1037/10096-003>
- [13] E. Von Glasersfeld, "An introduction to radical constructivism," in P. Watzlawick, "The invented reality," New York: Norton, 1984, pp. 17-40.
- [14] MIT Media Lab. "Professor Emeritus Seymour Papert, pioneer of constructionist learning", dies at 88. MIT News, 2016.
<http://news.mit.edu/2016/seymour-papertpioneer-of-constructionist-learning-dies-0801>
- [15] Development of an application with Siemens PLC,
<https://educatia.com.co/programacion-plc-logo-siemens-grafcet-a-ladder/>
- [16] J. D. Ospina Carreño, G. Tarazona Bermúdez, and L. A. Rodríguez Rojas, "Virtual Platform as an E-Government Mechanism Between Young Population and Local Administration of Mosquera, Colombia", ing. Solidar, vol. 14, no. 24, pp. 1-12, Jan. 2018.
<https://doi.org/10.16925/in.v14i24.2162>