Experiential Learning VR System for Studying Computer Architecture

Maria-Iuliana Dascalu, Sergean Bagis, Melania Nitu, Oana-Maria Ferche, Alin Dragos Bogdan Moldoveanu

University POLITEHNICA of Bucharest
400084, Cluj-Napoca
E-mail {oana.ferche, *alin.moldoveanu}@cs.pub.ro*

Abstract. As (Experiential Learning Lectures) points out, there are several downsides to classical lectures that can be extended to a lot of areas of education: the fact that students tend to lose their attention as soon as lectures start (Improving Listening Skills) or that they do not take notes rigorously (Taking Notes). Both the concentration on the taught subject as well as the need for taking notes can be improved by using experiential learning techniques. According to (Experiential Learning, Illinois), these techniques require the student to use his critical, reflective and analytical abilities during the learning process, requiring them to take initiative and to learn through experiments, asking questions, while being curious and creative. The role of the teacher is therefore shifted in experiential learning compared to classical lectures, as the teacher must mainly raise attention over various problems, support students and maintain an adequate environment for the learning process. The advancements of technology lead to new opportunities for systems increasing the interactivity during classes (Ubiquitous Learning). One such system is proposed by the current paper, which presents a Virtual Reality system designed to support experiential learning for computer architecture classes by creating an environment in which the user is immersed where he receives information and indications through textual cues while he assembles virtual components of a virtual computer.

Keywords: **:** virtual reality, augmented reality, e-learning, experiential learning.

1. Introduction

Experiential learning means "*learning by doing, learning through experience, learning through action and learning through discovery and exploration*" (Vargas-Hernández, 2014). Also, experiential learning activities allow students to apply theoretical knowledge as well as practice the newly learned skills within real-world situations (Glaser-Segura, Mudge, Bratianu, & Orzea, 2014).

Experiential learning is not a new concept, it was proposed by Kolb in

1984 (Kolb, 1984), but obtained added value from the exploitation of emerging technologies in learning environments and settings (Huang, Chen, & Chou, 2016), such as Virtual Reality (VR) (3D VE Chemistry, Mixed Reality (MR) (3DUPB MR))(3DUPB MR Campus)(MR Learning), Augmented reality (AR), semantic web. According to Kolb, there are four stages in the experiential learning cycle (Kolb, 1984): (1) concrete experience (in which the learner must not have a passive role, but he/she has to practice), (2) reflective observation (in which the learner has to do one step back and discuss about what he/she did with one's peers), (3) abstract conceptualization (in which the student makes connections based on one's experience and becomes aware of what he/she learnt); (4) active experimentation (in which the learner considers how to apply the newly learnt concepts in real life situations). Technologies may play an important part in each of the four stages, e.g. the discussions from the second phase can be facilitated by web technologies – via blogs, forums, social networks, the modeling from the third phase can be done with concept maps and ontologies, while simulations from first and last stages can be implemented using VR or AR.

The current paper highlights the opportunities brought to experiential learning by VR and AR, offers a concrete example of such a VR application which can be used to learn computer architecture, and which was developed by the authors and tested in an university environment and, in the end, underlines the benefits of experiential learning in business and education.

2. Technological opportunities of enhancing experiential learning

Over the years, a new wave of VR concepts has brought a lot of improvements for display technologies and input devices, completely modern technologies being designed and developed every day. In addition, new solutions to deal with existing problems on the hardware and software side of VR technology are constantly being introduced. The enhancements of these devices lead to the development of more accessible systems using VR technologies in various areas such as medicine (Architecture Neuromotor Rehab).

At the moment, the most popular and most accessible VR systems are Head-Mounted Displays (HMDs) and their specific input devices. The most

important wired HMDs are the Oculus Rift (Oculus Rift, 2017), the hTC Vive (hTC Vive, 2017) and the PlayStation VR (PlayStation VR, 2017). There are also the so-called mobile HMDs, which carry a common smart phone as an entire unit for display and data processing and are much more accessible in terms of price. They provide a simple box that contains an optical system, it is designed so that the phone display is correctly positioned to create the appropriate image for the user. In 2014, Google launched one of the first devices of this kind - Google Cardboard, which follows the philosophy to act as a very basic viewer (Google VR, 2017). Following the success of Cardboard, Google worked to develop a more advanced mobile VR solution - this time aimed at more stringent hardware requirements and taking full advantage of its total control over the Android mobile operating system. Announced in May 2016 and launched in November of the same year, the new platform is called Google Daydream (Google VR, 2017). Its compatibility is limited to high-performance phones with high-resolution, low-latency and low-persistence screens, working under the specially designed Android 7.0 or later, while aiming to deliver a virtual reality experience that is as good as possible, compared to the dedicated cabled HMDs. Google has also specifically designed a lightweight and comfortable headset called Daydream View. An important addition is the inclusion of a controller. This controller can be used to interact with the virtual world through button presses or moving the device. On-board sensors are used to track the direction of the controller and approximate the position of the user's hand.

Below are listed some of the existing implementations that use VR for educational purposes in general and experiential learning in particular:

- Engage (by Immersive VR Education Ltd.) (Engage, 2017) is an educational platform that gives students and teachers the means to communicate in a virtual environment: accessing lessons or private meetings from any location, teachers having the ability to use "Immersive FX" presentation tools, and to record their lectures using the "projected presence" feature and virtual interactive whiteboards. It is intended for Oculus Rift and hTC Vive VR systems (Engage, 2017). The Engage system allows teachers to record their actions in the VR that will later on be presented to the student whereas he system presented in this paper is mostly dedicated to the student and allows

Maria-Iuliana Dascalu, Sergean Bagis, Melania Nitu, Oana-Maria Ferche, Alin Dragos Bogdan Moldoveanu

him to interact with the virtual environment, while receiving information and indications.

- Public Speaking VR (by VirtualSpeech Ltd.) is a learning solution designed for public speaking. It was published for Google Cardboard. Users can customize their experience and train for their own discussions and specific presentations. The application provides feedback on the quality of their speech based on criteria such as rhythm, hesitation, volume and tone. Other interesting features are that it provides a heatmap of the user's look during its virtual presentation, the live audience, the ability to record speeches and advanced analyzes (Public Speaking VR, 2017).This tool is dedicated to help users overcome the fear of public speaking. The system that is the subject of the current paper can also use Google Cardboard to help the user achieve skills but in the field of computer architecture, unlike Public Speaking VR that helps improving public speaking abilities.

- Virtual Reality Training (by eLearning Studios) which offers a portfolio of design services for learning technologies and development services including eLearning, mobile learning, gameful eLearning design, VR, AR, serious games and mixed solutions. Part of their activity is to create customized learning experiences at the request of their customers. (Virtual Reality Trainin, 2017). The virtual environments created by eLearning Studios are dedicated to safely learning new techniques, for training for real-life situations that involve hazardous environments, emergency situations, all while providing an environment where it is safe to fail. The present system is not dedicated to training sessions that would otherwise be hazardous but provides a virtual environment in which the user is immersed to be able to learn in a risk-free manner the processes and stages of computer architecture.

- Unimersiv is a learning platform for virtual reality, which includes virtual tours of places such as cities, museums and the International Space Station, virtual courses presented by virtual teachers and interactive lessons such as those on the human body (Unimersiv, 2017). The ideas behind Unimersiv include that the user is much more likely to remember what he learns if it is presented to him in an

immersive manner. This idea is shared by the currently presented application for experiential learning.

- A VR-based application to learn how to drive (Muoh, Abdul-Karim, Rosu, Dascalu, & Stanica, 2017). Such a system would also test the abilities of the user in an otherwise dangerous situation, without the associated risks.

- A mixed reality university campus (Moldoveanu, Gradinaru, Ferche, & Stefan, 2014). This kind of system is usually dedicated to socializing while also involving the student in the campus activities and assisting the students and the professors in communicating aspects regarding the university.

All the studied systems have used VR to assist the user in either learning, socializing, dealing with stressful situations, etc. The purpose of the currently presented system is mostly dedicated to assisting the user in learning elements of computer architecture, to present him with the necessary information, all while allowing him to make mistakes in an environment where there are no risks or dangerous consequences.

The opportunities brought by VR to simulation environments and experiential learning (in universities, in learning centers or in organizations) are tremendous and it depends only on teachers, instructional designers and students to exploit them. VR and AR offer a safe environment for experiments and even if the initial costs of development of VR environments can be high, the cost of exploitation is much lower than the case in which experiments take place in real set-ups.

3. Experiential learning using VR: a case study on learning computer architecture

LeProVe is a virtual reality system for experiential learning: it was designed and developed by the authors in order to offer students from University POLITEHNICA of Bucharest a virtual 3D environment to learn a computer architecture by simulating its assembly. So, instead of making a computer in a real lab, the students can build one in a virtual lab, without the fear of making a mistake or breaking components. All actions taken by the user in the virtual environment are guided by dynamic additional text description for each computer component that indicates the place where that hardware should

be placed in the computer, thus facilitating the learning by doing. Furthermore, all the learning data are meant to be stocked and analyzed, in order to allow the students guidance for future improvements.

3.1. General description of an experiential learning application to learn computer architecture

The main functional specifications of the application are: the use of virtual reality platform, the means of locomotion in the virtual environment, the full control the user always has over his virtual reality position, the possibility to add new experiences and environments, the communication with the analysis service, the fact that the system reacts realistically and without inconsistencies, the possibility for the user to use his preferred hand, the possibility to start and stop the application when requested and stopping the application when the user exits the VR.

As non-functional specifications we can identify the design, the performance, stability and security. Regarding the design, the users can focus on objects and read the necessary explanations by holding the head tracker in a steady position. The controller is used as a laser pointer when user clicks on the interface elements, the cursor is being displayed at the same depth as the targeted objects. All scenes are realistically build, the objects and environments are using a real scale, keeping the visual style as realistic as possible within performance margins, maintaining the visual clarity. The application is stable: it installs and runs without crashing, provides real-time feedback to the user in case the application takes more than 3 seconds to load, rendering at a high and constant frame rate. For security reasons, the application is built to not compromise the user's data.

3.2. Technical details

Hardware

As hardware equipment, LeProVe is implemented for Daydream, the new Google VR platform and a smart phone connected to it (Google VR, 2017). As Daydream was not available at all times during the testing and experimental phases we also used the Google Cardboard (Figure 1).

Figure 6. Hardware used by the application

Software

One of the main tasks of the project was to create the virtual environment itself. This involves the creation and use of 3D models, which should prove realism, proper scaling, appropriate visuals, while taking into account platform and peripheral performance constraints. To create the virtual environment, we exploited Unity which is a multiplatform game engine and is mainly used to develop video games and simulations for PCs, consoles, mobile devices and websites (Unity, 2017). Based on the fact that they are the most popular and powerful methods of creating 3D resources, we used photogrammetry (with Agisoft Photoscan) and 3D modeling for the virtual environment (with Cinema 4D software) – see Figure 2.

In developing the application, we followed the popular model of Agile Software Development, particularly the extreme programming method (XP), which uses iterative cycles**.** The application architecture is based on our engine choice, Unity having an architecture of Entity-Component type. The design pattern that is the root of this architecture, commonly known as Entity Component System (ECS), is very often encountered in video games architecture and other highly interactive experiences. The principle is that the entity does not have its own logic, being implemented by its components. An entity contains several components, each with its own purpose. This design pattern is a base principle in the gaming industry, where every object in the virtual environment is actually an entity. When we want to add a certain behavior to an object, we actually add components to its entity. This is an easy and flexible approach, necessary to adopt because an interactive game

or experiment is created by continuously testing, so the code must be very accessible and easy to modify.
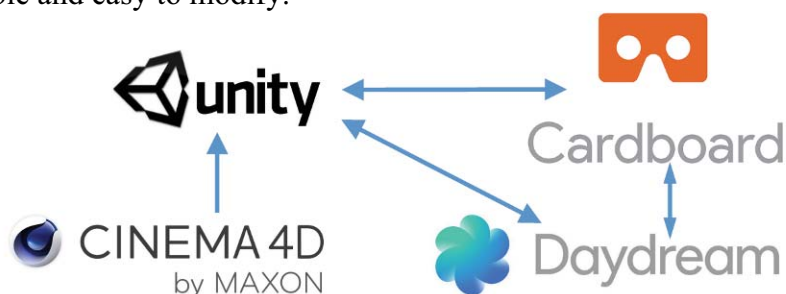

Figure 7. Software used by the application

In order to create the environment for Daydream using Unity, special prerequisites must be met. First, the chosen version of Unity must be compatible, which, at the time of writing, means that it must be 5.6 or more recent. Most of the development of our application was performed using Unity 2017.1. Another prerequisite is that the Google VR SDK must also be used. This is a special Unity package that provides all the basic compatibilities and functionality needed to run an application in virtual reality mode. The Android virtual reality mode is compatible only with Android 4.4 and later versions, so that the project's build settings must be configured to target this version at a minimum.

The project structure meets Unity standards. It is based on a regular file tree on which we generally have complete access, with the exception of certain reserved file names such as "Assets", "Resources", "Materials", etc. Each technical aspect of the project can be managed by Unity from the input, audio, physics, lighting, quality, graphics, construction options, etc. Aspects such as realistic rendering (number of pixels, texture quality), shadows (resolution, display distance, cascades) or the time required for the central processor to download textures for the graphics processor are taken into account when quality is chosen. The graphics settings influence the shading of objects of the virtual world, in relation to the distribution of light. All this had to be carefully chosen and adjusted to ensure the best performance on our target devices.

Google also recommends imposing other configurations for how the application is built, most of which relate to graphics quality and some other performance optimizations. The application must also be lightweight and, as

such, all unused functionality must not be compiled or packaged in the executable.

Input

The Daydream platform offers two input sources: the headset and the controller. The headset, using the sensors of the phone, provides rotation and acceleration data that can be used to determine the rotation of the user's head. Sometimes an additional input form can be provided using the touchscreen or the magnetic sensor of the phone.

The controller has a touchpad and three buttons, one of the buttons being integrated into the touchpad. The controller also provides information about its rotation, which can be used for various purposes, including a virtual pointer or implement to turn it into a virtual tool with which the user can interact. The headset input is standard for VR experiments, but the controller brings something new to the table and has great potential for enhancing interactivity. One of the main objectives the controller serves in our application is a means of locomotion of the user. Given the fact that the platform does not provide position tracking meaning that, if the user walks in real life, it will not be detected in the virtual environment. Considering this, we use the controller as a pointer to where the user can teleport into the virtual world. We chose teleportation because it is somewhat intuitive and does not cause any motion sickness as other means such as motion systems offered by regular applications or games.

Another way in which the controller input is used in our application is a way for the user to interact with the virtual objects, we have provided a way for the user to bring them, throw them, place them, install or interact with each other – see Figure 3. Since positional tracking is not available, the application uses the actual position of the controller to increase immersion. It must be as close as possible, because the user feels it in the real world, so there is an invisible virtual arm that tries to move it as realistically as possible, taking into account the angles between that a human shoulder, elbow and the wrist normally have.
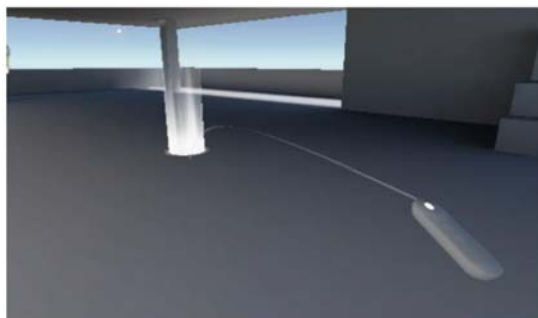
Figure 8. User input

Physics

Physics is essential for creating a game: an object must have the correct acceleration, the player must collide with the objects of the world, gravity or other forces must influence the events. Our application simulates the physical interaction between different objects.
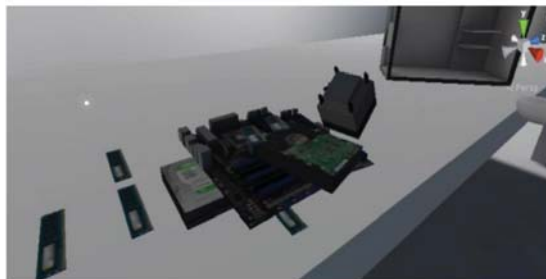

Figure 9. Physics engine in action

The Unity physics engine (see Figure 4) uses certain components that entities must have: rigid body and collision. The rigid body allows the physical behavior - that is, the impact of gravity and other forces. Collider defines a simplified mesh that allows to calculate intersections and collisions with other objects. The engine of physics is also used for radiation, a technique that is essential for the implementation of the interaction. The ray brails throw a line from one object to another and can be used to determine the distance, the line of sight that can be used to program almost all interactions. In our application, raycasting is used for interaction between the controller and user objects, environment and interface elements.

Graphics

One of the great advantages of the Unity engine is that it offers great visual possibilities. Light, cameras, hardware, textures, nuances and visual effects are only a few essential elements of the Unity graphic part. Light is important to obtain nuances on an object: brighter or more shady. Light can have different directions, colors or intensities. When creating a new scene, a directional light exists by default. This illuminates all the objects in the scene on the same single direction, while the distance to the object is not taken into account. In the main scene white point lights were used, to simulate the ceiling lights.

This type of light is located in a certain place in space and emits light also, in all directions. The light also provides shadows, which can be more or less detailed, visible at a closer or farther distance. The skybox also contributes to lighting, adding subtle lighting signals that add greatly to the realism of the scene.

Due to performance constraints, the lighting could not be fully realized in real time for our application. As a result, lighting must be calculated in advance. This adds a bit to the iteration time because, depending on the quality parameters, the calculation of a small environment can last from 5 to 30-45 minutes. However, pre-calculated lighting greatly improve the level of performance and is still visually realistic. Unfortunately, pre-calculated lighting affects only static objects.

Since our application makes extensive use of dynamic and interactive objects, this greatly detracts realism and immersion. Fortunately, we can use some of the pre-calculated lighting data to approximate the lighting that the dynamic objects would receive. Unity offers a system of "light probes" that store lighting information for many volumes in the environment. Dynamic objects that traverse these volumes automatically use lighting information while interpolating data from other nearby volumes to achieve as high a realism as possible

Models and environments

All models have been created using the before mentioned Cinema 4D software (see Figure 5 and Figure 6) sometimes using models processed using photogrammetry as a basis.

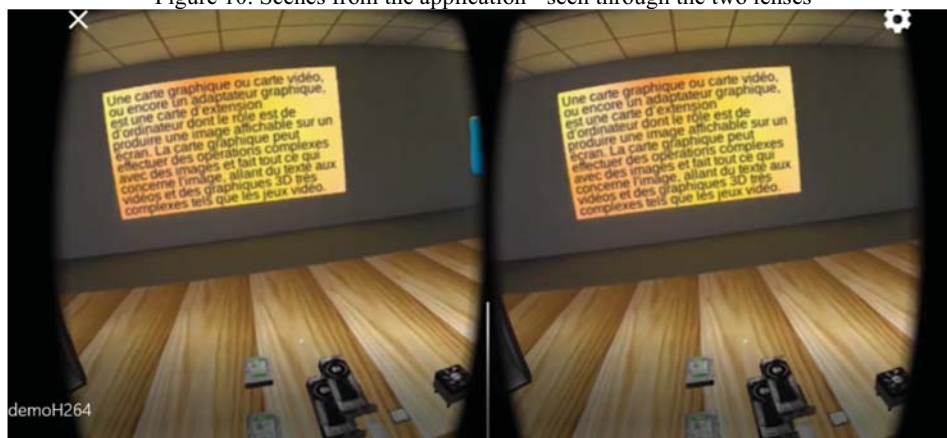Figure 10. Scenes from the application - seen through the two lenses

Figure 11. Text tips and tutorials in the application

Unfortunately, although there are many 3D models available on the internet, most of them are very detailed and far exceed the performance capabilities of our targeted devices. The modeling process included obtaining real-world measurements (which included research), making a more detailed model of the target object, with the appropriate images and textures, that resembles the real ones as closely as possible .Because of performance and memory constraints, models have had to undergo a simplification process of manually eliminating unnecessary faces and edges with the final objective of having the fewest possible polygons while maintaining a good level of detail.

The scripts

The script is an essential element in any Unity application. They have the purpose of managing the behavior of the player or other objects, managing user input or creating artificial intelligence systems in the game. In Unity, two languages are available to write scripts : JavaScript and C #. We used C # because it is very similar to Java and is more often used in the implementation of games. Each new script created in Unity builds a class that is derived from the built-in class called MonoBehaviour - this one provides important functions and events that are available to the objects in the game. The classes are models and/or controllers that can be used to make components and attach them to each object of the game (GameObject). When creating a class derived from MonoBehaviour, there are two predefined functions: Start () (useful for initialization) and Update () (called at each update of a frame of the game). Each component of an object can also be accessed inside a script.

Another very useful feature of Unity scripts is the debug command. It is used to display a message to the console and to check if certain functions are called properly or in the scenarios where the position of the player in the world was needed. Depending on the coordinates displayed in the console at each step, it was possible to determine which task should be performed in a certain location. Scripts are used for all intents and purposes, from the input and display to the behavior of the objects and the network component of the application.

Audio

Unity contains many features for the audio, such as 3D spatial sounds or real-time mixing. The principle of audio in Unity is based on the idea of transmitter-receiver: one has the audio source, which generates the sound and the audio listener that behaves like a microphone that receives the sound. The transmitter is attached to a certain object of the scene, while the receiver is usually linked to the camera. In order to add realism, google has provided an advanced sound system. This system is similar to how human ears influence the sound received and the subtle differences that help our brain determine the position of a sound. This is referred to as a "head-linked transfer function" audio system. This greatly improves the hearing experience and immersion. However, this only works correctly if the user is using stereo headphones. We used audio to add subtle audio clues related to the environment, such as environmental sounds that would be specific to the actual counterpart of

virtual environments. We also used audio to add sound clues to the user's actions, when teleporting, resuming an item, or successfully executing a task.

Building the application

After the creation of the scenes and the realization of the connection between them, the application can be tested by exporting it into an executable file. This can be done using the "Build settings" menu, where the scenes are added in the desired order. This order is very important because the SceneManager.LoadScene (int Index) function uses as index the position of the desired scene to be downloaded from this list. The application can then be created using the "Build" option for several platforms (PC, web, mobile), for a 32/64-bit architecture, to be able to validate the software. The Unity editor eliminates at the time of build any asset that is not used: objects, scripts, animations, to ensure that the dimension of the generated file is as small as possible.
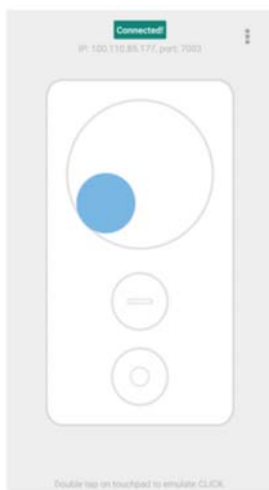

Figure 12. Daydream Controller Emulator Interface

Controller Emulator

Controller Emulator is an application provided by Google to help develop and use Daydream virtual reality applications. This application uses a phone equipped with a gyroscope and its touch screen in order to imitate the Daydream controller. It can be used with any Daydream compatible application. Our application is fully compatible with this emulator and has

been widely used because we did not have access to a Daydream headset or a Daydream compatible phone during development.

Regarding the availability aspect, the application is also compatible with cardboard devices, which are even weaker than those compatible with Daydream. LeProVe can run also on the cardboard devices, making full use of the input features, allowing a comparable user experience.

3.3. Validation

The application has been tested by 34 students and 2 teachers from University POLITEHNICA Of Bucharest, Faculty of Engineering in Foreign Languages and the feedback has been extremely positive. The students had the possibility of using the application for several minutes in a laboratory setting. After that, they were asked to fill in a questionnaire which is available in Table 1.

The students were from 1st to 3rd year of faculty, thus some of them (the ones from the 1st year) did not study computer architecture so far: 79.4% of the participants study computer architecture in school, the rest didn't. Most of them considered LeProVe system to be useful (79.4% of them). From them, 66.7% motivated that they like to learn by experiencing and LeProVe gives them that, 11.11% considered LeProVe to be useful, because they are interested in anything related to VR, while 40.7% were afraid of breaking something in a real laboratory and they are glad that LeProVe is just a simulator.

Almost half of the students who considered our system not to be useful explained to us that learning in a real laboratory can't be replaced by any simulator, even if it is closed to reality. 55.9% of the students would buy the system, if it is affordable, while 41.2% would use the system only if the teacher provides it. 70.6% of the students would like to build such a system, so, using VR based simulators in learning and teaching might increase the students' interests towards emerging technologies. The users did not user the application more than 30 minutes, so there were no problems with the battery.

Most users had never used such an application before and were mostly impressed. Regarding the educational aspect, most users who were unfamiliar with the chosen topic said they had learned something new. However, some constructive comments were also received - a couple of users found that the tips of the tutorial could be more detailed. Others, being new to virtual reality, felt that they needed more detailed instructions on how to use the controls.

Table 1. Feedback questionnaire for LeProVe users' validation

Questions addressed to LeProVe users	Possible answers
Did you study computer architecture so far?	(Only one answer is possible.) Yes/ No
Do you consider LeProVe to be useful?	(Only one answer is possible.) Yes/No
(If you consider LeProVe useful) Why you consider LeProVe useful?	(Multiple answers are possible.) I learn better by experiencing things and LeProVe provides me that. I will be able to make a computer without the fear of making a mistake. Virtual reality makes me curious and I want to try LeProVe. Other (please state)
(If you don't consider LeProVe useful) Why you don't consider LeProVe useful?	(Multiple answers are possible.) Nothing compares with learning in real life. I'm afraid to put on the head-mounted device. I feel sick when using a head-mounted device. Other (please state)
Would you be willing to buy LeProVe?	(Only one answer is possible.) Yes, for sure. / No, I will use it only if my teacher provides it for free./ It depends on the cost.
Would you like to build a system like LeProVe?	(Only one answer is possible.) Yes, for sure: it makes me curous about virtual reality technologies./ No, I am not interested in the domain.
Do you have anything to add about the system? If yes, write it down.	(Open question) The user could add anything.

The application has also been tested from a technical point of view, on a variety of devices and virtual reality tools, all results being in the expected performance. All users stated that they would like to make greater use of the application and encourage its continued development.

4. Benefits brought by experiential learning in education and business

According to (Experiential Learning, 2017), experiential learning has great advantages over theoretical learning.

First of all, it maintains the interest of the student. Due to its clear practical application, the student is more likely to find that the knowledge he gathers while performing the practical task will also be applied in the future. He sees the immediate results of the actions he takes – whether a success when

performing correctly or errors when making mistakes - and is therefore more involved.

Mistakes are also an important part of experiential learning – trying something that doesn't work leads him to learn from mistakes instead of fearing them, thus supporting creative thinking, and improving the attitude towards learning. Making decisions, taking responsibility, critical thinking also create opportunities to grow beyond the immediate purpose of the respective class or lesson, are most often missing in theoretical learning processes but are most important when performing a task in the real world (Experiential learning, Texas, 2017) (Experiential Learning, 2017).

5. Conclusions

The experiential learning process is one that helps the student develop important aptitudes, regarding the theoretical aspects as well as outside of the scope of the lesson: critical thinking, taking responsibility, learning from mistakes, and can also contribute to the involvement of the student, through increased interactivity and direct feedback.

This paper presents the development process and tools used to create a VR application dedicated to learning computer architecture lessons by building virtual computer parts using virtual components modeled after real ones. The components must be placed in the right position and order, by following text instructions on screen.

The development process used the Unity game engine for creating the Virtual Environment with all of its functionality, including scripting, physics, audio and input controls. The system was developed to use Google Daydream Head Mounted Display, that makes use of a smartphone to immerse the user in the Virtual Reality. For testing purposes, due to the unavailability of the Daydream we used the Google Cardboard, that can simulate the same behavior.

The system was validated through tests with 8 students and 2 teachers from the Faculty of Engineering in Foreign Languages at the University POLITEHNICA of Bucharest. All the users managed to use the system successfully, despite some of them not being familiar with VR equipment. The received feedback was mainly positive, the users stating that they learned something from the practical VR lesson, but also included constructive feedback regarding some features that they thought could be improved.

Reference text and citations

Experiential Learning Lectures (2017). https://uwaterloo.ca/centre-for-teaching-excellence/resources/integrative-learning/experiential-learning#Wolvins

Improving Listening Skills. (1983) Wolvin, A.D. "Improving Listening Skills" In R. B. (ed.), Improving Speaking and Listening Skills. New Directions for College Learning Assistance, no. 12. San Francisco: Jossey-Bass, Inc. 1983.

Taking Notes. Dezure, Deborah et al. Research on Student Note-taking: Implications for Faculty and   Graduate Student Instructors. CRLT Occasional Papers, No. 16. Center for Research on Learning and Teaching. University of Michigan.

Ubiquitous Learning. L. Stefan, D. Gheorghiu, F. Moldoveanu, A. Moldoveanu, "Ubiquitous Learning Solutions for Remote Communities - A Case Study for K-12 Classes in a Romanian Village", 569-574, 2013.

3DUPB MR. A Moldoveanu, A Morar, V Asavei. 3DUPB-The Mixed Reality Campus: A glimpse at how mixed reality systems can shape the future.

3D VE Chemistry. E. Shudayfat, F. Moldoveanu, A. Moldoveanu. A 3d virtual learning environment for teaching Chemistry in high school. Proceedings of the 23rd International DAAAM Symposium, Volume 23, No.1, ISSN 2304-1382

3D UPB MR campus. A. Moldoveanu, A. Gradinaru, O.-M. Ferche, L. Stefan, "The 3D UPB mixed reality campus: Challenges of mixing the real and the virtual", Proc. IEEE The 18th International   Conference System Theory Control and Computing (ICSTCC), IEEE Press, pp. 538-543, Oct. 2014.

MR Learning. M.-I. Dascalu, A. Moldoveanu, E. A. Shudayfat, "Mixed reality to support new learning   paradigms", Proc. IEEE The 18th International Conference System Theory Control and Computing (ICSTCC), pp. 692-697, Oct. 2014.

Architecture Neuromotor Rehab. S. Caraiman, A. Stan, N. Botezatu, P. Herghelegiu, R. Lupu, A. Moldoveanu, "Architectural Design of a Real-time Augmented Feedback System for Neuromotor Rehabilitation", Proceedings of the 20th International Conference on Control Systems and Computer Science (CSCS 2015), pp. 850-855, May 27–29, 2015.

Google VR (2017) https://vr.google.com/ Engage. (2017). http://immersivevreducation.com/engage-education-platform/

Glaser-Segura, D., Mudge, S., Bratianu, C., & Orzea, I. (2014). Convergence at What Cost?: A Quasi Experiment of Professional Identity under the Bologna Process. În A. Dima, Handbook of Research on Trends in European Higher Education Convergence (pg. 183-201). IGI Global.

hTC Vive. (2017). https://www.vive.com/eu/

Huang, T.-C., Chen, C.-C., & Chou, Y.-W. (2016, May). Animating eco-education: To see, feel, and discover in an augmented reality-based experiential learning environment. Computers & Education(96), 72-82.

Kolb, D. (1984). Experiential Learning: experience as the source of learning and

development. Englewood Cliffs: Prentice Hall.

Moldoveanu, A., Gradinaru, A., Ferche, O.-M., & Stefan, L. (2014). The 3D UPB mixed reality campus: Challenges of mixing the real and the virtual. The 18th International Conference on System Theory, Control and Computing (ICSTCC) (pg. 538-543). Sinaia: IEEE.

Muoh, N., Abdul-Karim, I., Rosu, S., Dascalu, M.-I., & Stanica, I. (2017). Training based on Virtual Simulators. Case Study for Improving Parking Skills. The International Scientific Conference eLearning and Software for Education (pg. 495-502). Bucharest: "Carol I" National Defence University.

Oculus Rift. (2017). https://www.oculus.com/rift/

PlayStation VR. (2017). https://www.playstation.com/en-us/explore/playstation-vr/

Public Speaking VR. (2017). https://www.oculus.com/experiences/gear-vr/1093617717347270/

Unimersiv. (2017). https://unimersiv.com/

Unity. (2017). https://unity3d.com/

Vargas-Hernández, J. G. (2014). Networking International Student Collaboration and Experiential Exercise Projects. În N. Baporikar, Handbook of Research on Higher Education in the MENA Region: Policy and Practice (pg. 208-229). IGI Global.

Virtual Reality Trainin. (2017). From https://www.e-learningstudios.com/virtual-reality

Experiential Learning. (2017) https://www.envisionexperience.com/blog/the-benefits-of-experiential-   learning (Last accessed October 2017)

Experiential Learning, Texas. (2017) https://facultyinnovate.utexas.edu/experiential-learning (Last accessed October 2017)