Education of Students Using Virtual Reality

P. Pokorný and N. Šenovský

***Abstract*—** This paper describes the development of an Educational Virtual Reality application that could be used in technically-oriented subjects - in this instance, Micro-computer Programming. This is taught at FAI, TBU in Zlín, and its practical part takes place in a specialised laboratory that contains some controllable real models. The article describes the design and realisation of the multi-platform Virtual Reality application in a selected real model´s environment. By using Virtual Reality hardware and this application, students can then experience the virtual world that they have created as if was the real world – thus allowing them to gain a better understanding of these subjects.

***Keywords*—** Computational Techniques, Computer Simulation, Education, Software Engineering, Virtual Reality.

I. Introduction

IN contemporary literature, we can find a number of different interpretations of the term Virtual Reality. Resource, [1], - defines this as “Immersive Computer-simulated Reality” that creates a physical environment that does not exist. Resource, [2] - mentions that the term Virtual Reality is mainly determined on the basis of its purpose: “The purpose of Virtual Reality is to enable sensorimotor and cognitive activity for a person - (or persons), in a digitally-created, artificial, world; which can be imaginary, symbolic - or a simulation of certain aspects of the real world”.

In general, Virtual Reality is the computer-generated simulation of a 3D environment, which seems very real to the person experiencing it, using special electronic equipment. The main objective is to achieve a strongly-held sense of being present in that virtual environment. [3] There are five tied key elements of the Virtual Reality realm: creators, participants, the virtual world, immersion and interactivity. [4]

It is necessary to use special software and hardware in order to fully exploit Virtual Reality Technology. The Virtual Reality Modelling Language (VRML), was developed for this purpose - and later - so too, the WebVR (JavaScript Application Programming Interface), was produced for various Virtual Reality devices. Virtual Reality is currently massively supported in modern software engines - like Unreal or Unity 3D [3]. Modern Virtual Reality hardware includes a headset in particular, based on smartphones with small and fast computers, small HD screens, gyroscopes and motion sensors.

As the Virtual Reality medium matures, so too will its use increase. Customer-level Virtual Reality started with video gaming and entertainment applications (like 3D cinema). Its development meant that it was possible more and more to use it in non-gaming applications, like travel and tourism (visiting faraway places without leaving home), industrial design and mechanical engineering (AutoCAD, SolidWorks), medicine (enhanced visualisation of medical diagnoses), architecture, real-estate, training (like driver instruction) or education. [3]

Education is one of those areas where Virtual Reality is applied more and more frequently. Companies and organisations increasingly incorporate Virtual Reality technologies into the marketing, training and skills development and product and service design fields. [5]

This was the reason why we wanted to design and develop a multiplatform Virtual Reality application that could be used in the tuition processes at our faculty. This paper describes the progress of its creation.

II. Application Design

Initially, it is necessary to decide what should be the content of the Virtual Reality application. This decision is based on its practical use in practice in specialised subjects and to offer some benefits.

Our faculty (FAI – Faculty of Applied Informatics) [7], is professionally focused on new software creation, robot and intelligent systems design; computer and data security, web-programming, office administration, modern building security and the automatic management of technological processes. After considering several options, the final decision was to apply this to the Microcomputer programming subject. Specialised laboratories form part of these subjects, which include interesting real models, including an intelligent house, a bottle-filling line, and an automated warehouse or a fountain with two sets of lights and two water circuits, and others.

The model of an intelligent house – from the above-mentioned list was selected. Its photo is shown in Figure 1. This house is connected to a SAIA PCD 2 programmable logic controller, where the students design control algorithms for the automation of processes inside the house model.

Pavel Pokorný is a senior lecturer at the Department of Computer and Communication System, Faculty of Applied Informatics, Tomas Bata University in Zlín (full address: Nad Stráněmi 4511, 760 05 Zlín, Czech Republic), e-mail: pokorny@utb.cz.

Nikola Šenovský was a student at the Faculty of Applied Informatics, Tomas Bata University in Zlín (full address: Nad Stráněmi 4511, 760 05 Zlín, Czech Republic), e-mail: senovsky@gmail.com.

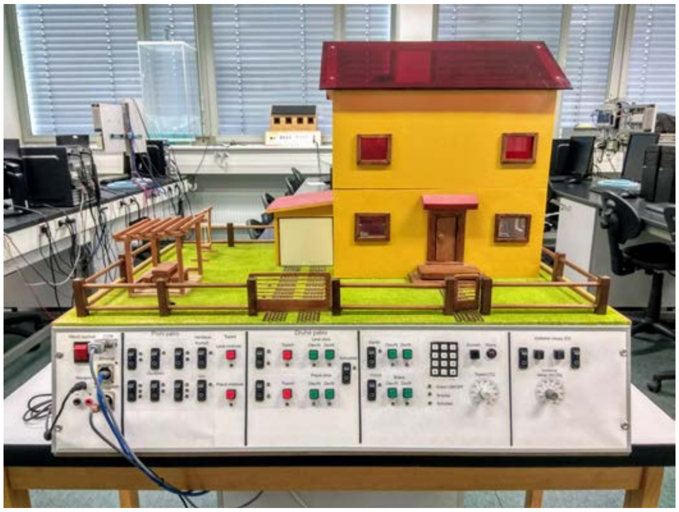

Fig. 1 A selected model of an intelligent house in the faculty laboratory [6]

Students can see this model from the outside and dismantle it in the laboratory. But with the Virtual Reality application, they would be able to study this model without dismantling it. They can see how the house appears inside and then judge how the intelligent automated systems inside the house behave. With this information, students could be better able to design and debug their control algorithms.

The first phase was to compile detailed documentation and measurements, because it would be very necessary to create the most accurate possible 3D model of the house. For this reason, some photos in high resolution were captured – and, in the following step, all of the needed dimensions were manually measured and documented.

The structure of the planned application was designed very simply. Its starting point was positioned in the front of the house. Students can change their position in other parts of the house - (including rooms inside), by clicking on one of a number of floating light-blue cubes that are equally scattered over the whole 3D area. After focusing and clicking on this cube, the student would be "moved" to its position. This solution was designed with a view to minimising the risk of Virtual Reality Sickness. [8]

III. Hardware and Software used

When the content and functionality of the application has been determined, the ensuing step is to select suitable hardware and software.

A. Hardware

The market offers a variety of headsets that can be used for Virtual Reality solutions. Complex solutions designed for desktop computers - (Oculus Rift and HTC Vive), were excluded because they are expensive, require a powerful computer and connector cables. Better solutions are offered by Samsung Gear VR and Google Daydream because they are designed as mobile Virtual Reality platforms. They are cheaper and support remote-control, but using this often requires maintaining calibration.

In the end, the Google CardBoard device, the ColorCross VR BOX type was selected, which is one of the cheapest solutions (approximately 12 Euro) – shown in Figure 2. Another advantage is the top head strap, which ensures the stability of the image on the user's head, has adjustable lenses for sharpness correction of the image, and supported by a smartphone diagonal display of up to 6 inches. [9]

The Xiaomi Redmi Note 4 was used as the chosen smartphone [10]. It offers a very powerful octa-core processor, 32 GM internal memory, a dual LED flash for taking photos, a large full HD screen (5,5 inches), and an Android 6.0 Marschmallow with a user-friendly MIUI interface.

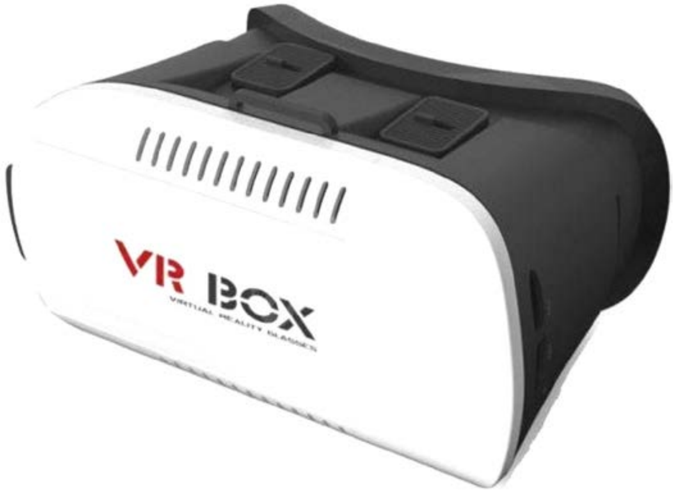

Fig. 2 The ColorCross VR BOX – VR Headset

B. Software

All 3D models used in the application were made in Blender. These models were then imported into the Unity engine, where the Virtual Reality application was created.

Blender is a free and open-source 3D creation suite. It supports the entirety of the 3D pipeline – modelling, rigging, animation, simulation, rendering, compositing and motion tracking - even video editing and game creation. [11]

It has a wide variety of tools that make it suitable for almost any sort of media production. People and studios around the world use it for hobby projects, commercials, feature films, games and other interactive applications in kiosks, games and scientific research. [12]

Custom architecture is the basis of the Blender suite – its user interface, shortcuts and layout can be fully customised and is the same on all platforms. It also supports localisation into many languages and an extensive Python API; this means that every tool is available for scripting and customisation. In order to cooperate with other applications, Blender performs import and export options into many graphic formats, including images (jpeg, jpeg2000, png, tga, tiff), videos (avi, mpeg, quicktime) and 3D formats (3ds, dae, dxf, obj, svg, stl, vrml, x3d) and many others. [13]

Blender's modelling tools include N-gon support and many editing commands (Extrude, Edge Slide, Knife, Bevel, Collapse, modifiers, transformations, sculpting tools, etc.). Its animation feature set offers a character animation pose editor, forward/inverse kinematics, automated walk-cycles along paths and fast rigging tools. Some different rendering algorithms (Blender Render, Cycles Render, FreeStyle, Blender Game and more) perform different output visual styles. All rendering algorithms support Multi-core CPU rendering, GPU rendering and multi-GPU.

Blender also supports powerful simulation tools (particles, hair, fire, smoke, cloth, fluids and destructible/collidable objects), and a fully-fledged compositor with camera and object motion tracking, masking and compositing commands. [11]

Unity represents a cross-platform real-time engine that is used to create many games and 3D visualisations in the world. Its real-time platform, powered by tools and services, offer incredible possibilities for creators and game developers across industries and applications. [14]

The Unity engine offers primary scripting in the C# programming language, but many scripts are prepared in templates and are easier to use without a deeper knowledge of this programming language. Unity also supports many programmable interfaces, which predetermine its use in many platforms. These interfaces include Direct3D, OpenGL, OpenGL ES, WebGL, as well as Vulkan (Windows, Linux, and Android platforms) and Metal (iOS platform). [15]

Newer Unity versions support all main graphics and game technologies, including different resolution settings for each platforms, mip-maping of images, many variable shading algorithms, texture compression, different mapping techniques (bump, parallax, reflection), ambient occlusion, and dynamic shadowing or many post-processing effects.

Unity requires the Java Development Kit, Android SDK and Android Build Support in order to build applications for the Android platform. The Java Development Kit can be downloaded from [16], Android SDK and Android Build Support can be additionally installed from Unity.

IV. 3D Model Preparation

The 3D model of an intelligent house and its parts were created in the Blender creation suite. A standard polygonal representation for all these models was used (called Meshes in Blender – all objects are composed from vertices, edges and faces). Blender supports a large number of modelling tools for mesh objects.

Partial 3D models were created from mesh primitives (like planes and cubes), that were inserted into the 3D scene. After that, these 3D meshes were transformed (moved, rotated, scaled), according to their corresponding original real parts. The Snap command was used in order to attain exact locations (it can align the objects according to their origin points). Other frequently used tools included editing commands –copying, duplicating, deleting, and the joining of vertices, edges, faces or whole objects.

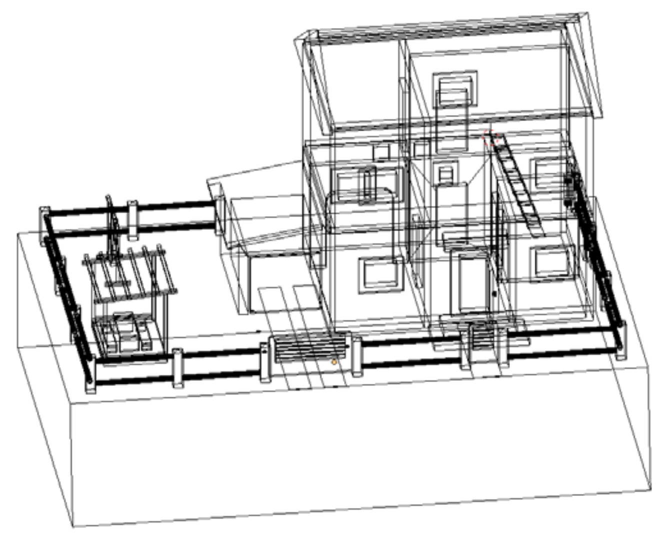

Fig. 3 A 3D model of an intelligent house in Blender (wire-frame shading)

In the modelling phase, some more advanced modelling tools were also used. The Extrude command allows one to create a new face on selected faces. In order to create smaller detailed parts of objects, the Subdivide and Knife tools can be used. They can simply split any selected faces into their smaller composite parts.

Modifiers are another useful tools in Blender. They can improve/transform any shapes of selected objects into other ones. The Boolean modifier was used - which can make new shapes using logical operations among selected objects. Finally, the Decimate modifier was used for the optimisation (reduction), of the number of object vertices.

Figure 3 shows the final shape of the 3D intelligent house model in Blender created by using the tools described above. This model uses wire-frame shading - (the faces are invisible), in order to see inner and reverse parts of the objects.

When the final model of the intelligent house was created, this was followed by making the materials and textures. Since textures are linked to materials, and materials are linked to object faces, materials were made first. Only two settings were used – specified diffused and specular colours for each face. The accuracy of these colours was unimportant since both colours could simply be changed later in the Blender or Unity environments. If the colour assignment was sufficient, it was the only surface setting. Textures were used in other cases.

The textures were made, linked and then set in Unity. The reason was the elimination of problems during the export process from Blender to Unity - (in some cases, texture settings are not exported correctly). The preparation for textures was performed in Blender – for removing doubled object faces, correcting the direction settings of normal vectors and assigning UV maps to every object that needs to be textured.

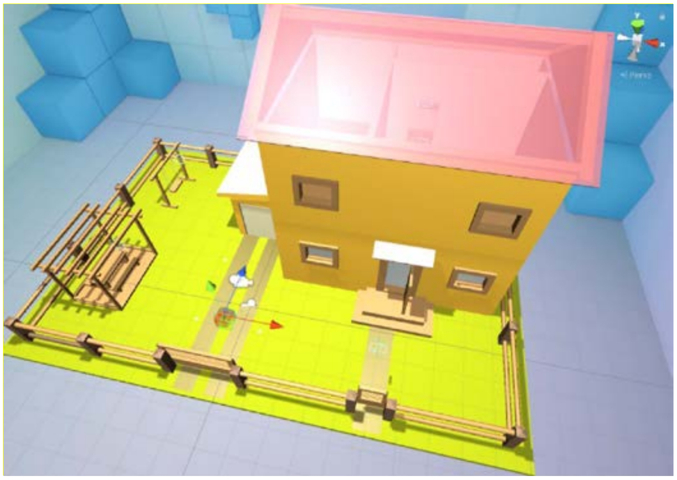

Fig. 4 A 3D model of an intelligent house with materials in Unity [6]

V. Creation of the Application

After the creation of the new project in Unity, the complete 3D model of an intelligent house was imported. The .blend format – a native Blender format was used for traffic purposes. In order to speed up the rendering process, all non-triangular faces of the imported model were triangulated automatically.

In the import process, all 3D models needed to be moved into the Hierarchy window in order to create a new instance. The 3D models can make multiple instances at any time, and each instance can have different parameters. Only one instance was used in this project.

After the instance was created, all parts of the house model were transformed as necessary – this correction was made because Blender and Unity use different 3D scene concepts.

All materials created and linked in Blender are automatically imported into Unity with their corresponding objects. However, only diffuse and specular colours are available, and these cannot be modified. Due to this, some advanced materials were created directly in Unity - especially with regard to the transparency of the mirrored materials. They had to be created in the “Project Window”, and all of the requisite parameters were set in the “Inspector Window”. The appearance of the model in this phase is shown in Figure 4.

Textures were applied to object surfaces, where material settings were not sufficient – for instance, roads, grass, carpets, etc. Texture settings is very simple to perform in Unity. The source image file must be set here. Its dimensions must have power 2, and if the texture is “tile mapped”, it has to be seamless. Where necessary, the texture colours were adjusted. Skybox represents a special kind of textural spherically encompassment of the scene, creating an illusion of a sky above/around. The Skybox used in this educational application was downloaded from Unity Asset Store [17].

In conclusion, the lights must be set in order to finalise the 3D scene in Unity. Seven pure white light sources were used - regularly placed within the whole scene. They all were “baked” - (pre-calculated), because their real-time calculation is time/hardware consumptive. The light sources cast no shadows - because it turned out to be rather difficult to create realistic-looking lighting and shadows at the same time.

The creation of the application core was when the 3D scene was completed. Movement - in the virtual world can be resolved in several ways. There are however, two criteria – movement continuity, and a freedom of choice relating to where to move. Teleporting movement was selected due to the fact that continuous motion can cause motion sickness in individuals. Since the students are supposed to learn about the specific parts of the models exploitable in the educational application - 14 teleport points were created. They were positioned with respect to points where is a need to retain students’ attention. The model of a simple cube with transparent faces was inserted in these places. The edges colour was set as light-blue. Its appearance inside the Educational Virtual Reality application is shown in Figures 5 and 6.

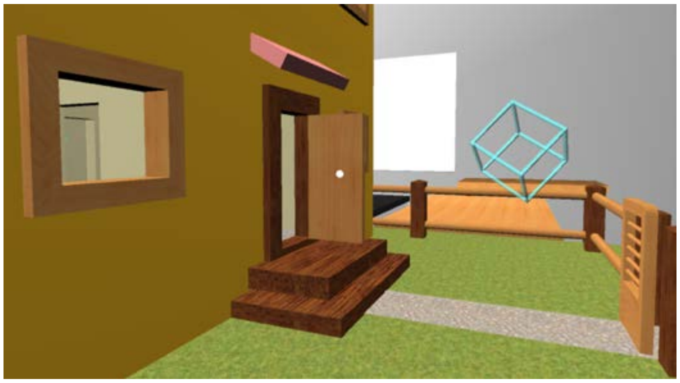

Fig. 5 A snapshot taken from the created Virtual Reality application (in front of the house)

Other requisite algorithms - for example, navigation in the scene; and linkage to a Virtual Reality headset in order to react to user movements), can be performed in two ways. Firstly, to program it manually using Unity Script [18]; the second, is to buy a complete solution via Unity Assets Store [17]. The Virtual Reality application described in this paper also made use of the C.T.S. Cardboard Teleport System solution [19].

The whole development and debugging processes of the designed application can take place in the Unity environment because it can be pre-viewed - and in a separate window. Three builds were compiled: Android, Windows, and WebGL. The Windows and WebGL application source codes had to be modified since the main camera must react to mouse movements and on clicking - instead of head movement with a Virtual Reality headset suite.

When the entire educational virtual application had been completed, it was student volunteer tested in Microcomputer programming subjects. They visited all 14 teleport points with mounted Virtual Reality headsets. After that, they gave their impressions - which were evaluated. These reactions led to the creation of methodological guidelines for teaching staff who want to exploit Virtual Reality in their subjects.

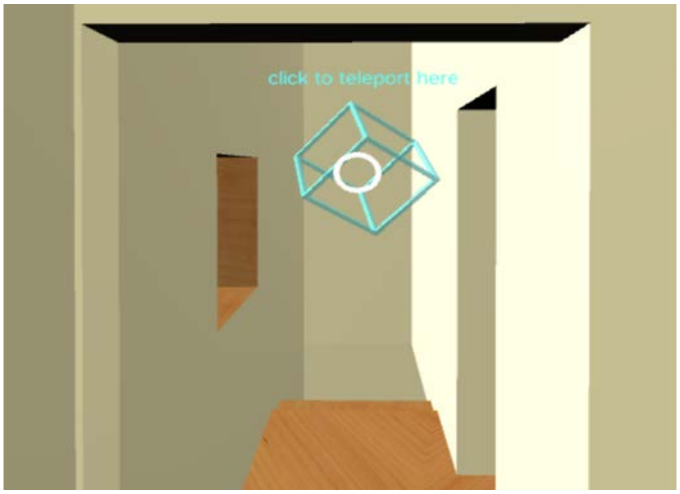

Fig. 6 A snapshot taken from the created Virtual Reality application (inside the house)

VI. Conclusion

This contribution describes the use of Virtual Reality in student tutorial processes in technically-oriented subjects. The Virtual Reality application - developed in the Unity real-time engine, was created for this purpose.

The application contains the environment of a selected real model of an intelligent house - taken from a Microcomputer programming laboratory - this subject is delivered at our faculty. Specialised Virtual Reality hardware, and this application, were then tested on FAI. TBU in Zlín students. Methodological guidelines for teachers were then created on the basis of student reactions.

A huge potential is offered by Virtual Reality implementation in the educational process. However, it may cause some health problems for some individuals – like eyestrain, headache or stomach-ache. For these reasons, a great deal of research and development is still needed – even required, in this research area.

References

[1] P. Mealy, *Virtual & Augmented Reality for dummies*, John Wiley & Sons, Inc., 2018. ISBN 978-1-119-48134-8.

[2] P. Fuchs, *Virtual Reality Headsets*, CRC Press, 2017. ISBN 978-1-315-20824-4.

[3] J. Linowes, *Unity Virtual Reality Projects*, Packt Publishing, 2015. ISBN 978-1-78398-855-6.

[4] W. R. Sherman and A B. Craig, *Understanding Virtual Reality: Interface, application and design*, Morgan Kaufmann, 2018. ISBN 978-0-12-818399-1.

[5] S. Minocha and A.D. Tudor, *Virtual Reality in Education*, Learning Design and Technology Enhanced Learning Special Interest Group, The Open University, Walton Hall, Milton Keynes MK7 6AA, September 2017.

[6] N. Šenovský, *Virtual Reality in High School Education*, Diploma Thesis, Tomas Bata University in Zlín, Faculty of Applied Informatics, 2018.

[7] FAI contributors (2019), Faculty of Applied Informatics – Tomas Bata University in Zlín [Online]. Available: https://fai.utb.cz/en/

[8] J. Munafo, M. Diedrick and T. A. Stoffregen, The Virtual Reality head-mounted display Oculus Rift induces motion sickness and is sexist in its effects, *Experimental Brain research*, Springer, vol. 235, March 2017, pp. 889–901.

[9] Google (2019), Google VR [Online]. Available: https://vr.google.com/

[10] Xiaomi (2019), Mi Global Home [Online]. Available: https://www.mi.com/global/

[11] Blender (2019), Home of the Blender Project [Online]. Available: https://www.blender.org

[12] Blender contributors (2019), Blender 2.79 Reference Manual [Online]. Available: https://docs.blender.org/manual/en/latest/

[13] J. M. Blain, *The Complete Guide to Blender Graphics*, 3rd Edition, CRC Press, 2016. ISBN 978-1-4987-4647-2.

[14] Unity Technologies (2019), Unity [Online]. Available: https://unity.com/

[15] S. Blackman and A. Tuliper, *Learn Unity for Windows 10 Game Development*, APress, 2016. ISBN 978-1-4302-6758-4.

[16] Oracle (2019), Java SE Development Kit 11 - downloads [Online]. Available: https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html

[17] Unity Technologies (2019), Unity Asset Store [Online]. Available: https://assetstore.unity.com

[18] Unity Technologies (2019), Unity – Scripting API [Online]. Available: https://docs.unity3d.com/ScriptReference/

[19] Unity Technologies (2019), CTS [Online]. https://assetstore.unity.com/packages/tools/gui/c-t-s-cardboard-teleport-system-11