Virtual Reality in High School Education

Bc. Nikola Šenovský

Master's thesis
2018

Tomas Bata University in Zlín
Faculty of Applied Informatics

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2017/2018

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Nikola Šenovský**
Osobní číslo:	**A15351**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Učitelství informatiky pro střední školy**
Forma studia:	**prezenční**
Téma práce:	**Virtuální realita ve vzdělávání na středních školách**
Téma anglicky:	**Virtual Reality in High School Education**

Zásady pro vypracování:

1. **Vypracujte literární rešerši o technologii virtuální reality.**
2. **Popište software Blender a Unity a jejich využití ve virtuální realitě.**
3. **Charakterizujte strukturu vzdělávací aplikace.**
4. **Vytvořte vzdělávací aplikaci implementovanou ve virtuální realitě, za použití zmíněného softwaru.**
5. **Vyzkoušejte vytvořenou vzdělávací aplikaci na studentech střední školy a vyhodnoťte výsledky.**

Rozsah diplomové práce:
Rozsah příloh:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **MURRAY, Jeff W. Building virtual reality with Unity and Steam VR. Boca Raton, FL: CRC Press, 2017, 352 s. ISBN 978-1138033511.**
2. **BLAIN, John M. The complete guide to Blender graphics. 4. vydání. Boca Raton, FL: CRC Press, 2017, 552 s. ISBN 978-1138081918.**
3. **POKORNÝ, Pavel. Blender: naučte se 3D grafiku. 2., aktualiz. a rozš. vyd. Praha: BEN – technická literatura, 2009, 286 s. ISBN 978-80-7300-244-2.**
4. **11.127x Design and Development of Games for Learning. MITx [online]. edX Inc, 2016 [cit. 2017-11-11]. Dostupné z: https://courses.edx.org/courses/course-v1:MITx+11.127x_2+3T2016/course/**
5. **BLENDER. FAI UTB [online]. Vimeo, 2011 [cit. 2017-11-11]. Dostupné z: https://vimeo.com/user8620872.**
6. **Tutorials. [online]. UNITY Technologies, 2017 [cit. 2017-11-11]. Dostupné z: https://unity3d.com/learn/tutorials.**
7. **WENDERLICH, Ray. Unity. [online]. Razeware LLC, 2017 [cit. 2017-11-11]. Dostupné z: https://www.raywenderlich.com/category/unity.**
8. **Tutorials. [online]. BLENDER Foundation, 2017 [cit. 2017-11-11]. Dostupné z: https://www.blender.org/support/tutorials/.**

Vedoucí diplomové práce: **Ing. Pavel Pokorný, Ph.D.**
Ústav počítačových a komunikačních systémů
Datum zadání diplomové práce: **1. prosince 2017**
Termín odevzdání diplomové práce: **16. května 2018**

Ve Zlíně dne 11. prosince 2017

doc. Mgr. Milan Adámek, Ph.D.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.
①

prof. Mgr. Roman Jašek, Ph.D.
garant oboru

I hereby declare that:

- I understand that by submitting my Master’s thesis, I agree to the publication of my work according to Law No. 111/1998, Coll., On Universities and on changes and amendments to other acts (e.g. the Universities Act), as amended by subsequent legislation, without regard to the results of the defense of the thesis.
- I understand that my Master’s thesis will be stored electronically in the university information system and be made available for on-site inspection, and that a copy of the Diploma/Thesis will be stored in the Reference Library of the Faculty of Applied Informatics, Tomas Bata University in Zlín, and that a copy shall be deposited with my Supervisor.
- I am aware of the fact that my Master’s thesis is fully covered by Act No. 121/2000 Coll. On Copyright, and Rights Related to Copyright, as amended by some other laws (e.g. the Copyright Act), as amended by subsequent legislation; and especially, by §35, Para. 3.
- I understand that, according to §60, Para. 1 of the Copyright Act, TBU in Zlín has the right to conclude licensing agreements relating to the use of scholastic work within the full extent of §12, Para. 4, of the Copyright Act.
- I understand that, according to §60, Para. 2, and Para. 3, of the Copyright Act, I may use my work – Master’s thesis, or grant a license for its use, only if permitted by the licensing agreement concluded between myself and Tomas Bata University in Zlín with a view to the fact that Tomas Bata University in Zlín must be compensated for any reasonable contribution to covering such expenses/costs as invested by them in the creation of the thesis (up until the full actual amount) shall also be a subject of this licensing agreement.
- I understand that, should the elaboration of the Master’s thesis include the use of software provided by Tomas Bata University in Zlín or other such entities strictly for study and research purposes (i.e. only for non-commercial use), the results of my Master’s thesis cannot be used for commercial purposes.
- I understand that, if the output of my Master’s thesis is any software product(s), this/these shall equally be considered as part of the thesis, as well as any source codes, or files from which the project is composed. Not submitting any part of this/these component(s) may be a reason for the non-defense of my thesis.

I herewith declare that:

- I have worked on my thesis alone and duly cited any literature I have used. In the case of the publication of the results of my thesis, I shall be listed as co-author.
- That the submitted version of the thesis and its electronic version uploaded to IS/STAG are both identical.

In Zlín; dated: ....

Student´s Signature

ABSTRAKT

Cílem této práce je navrhnout a implementovat vzdělávací aplikaci pro virtuální realitu. Teoretická část obsahuje literární rešerši o technologii virtuální reality a současných možnostech jejího využití v oblasti vzdělávání mládeže. Následuje popis softwaru používaného pro vývoj ve virtuální realitě, jmenovitě Blender, se zaměřením na přípravu 3D modelů pro virtuální realitu, a Unity, vývojářský engine.

Praktická část se zabývá procesem vytváření vzdělávací aplikace zaměřené na učení prožitkem. Jsou použity 3D modely vytvořené v programu Blender a samotná aplikace je vyvinuta ve vývojářském engine Unity. Vytvořená aplikace byla otestována na studentech vybrané střední školy a výsledky tohoto testu byly vyhodnoceny.

Klíčová slova: virtuální realita, vzdělávání, školství, výuka, Blender, 3D model, Unity

ABSTRACT

The aim of this thesis is to design and implement an educational application for virtual reality. The theoretical part contains literary research about virtual reality technology and current possibilities of its utilization in the field of secondary education. A description of the software used for virtual reality development follows, specifically Blender, which focuses on preparing 3D models for virtual reality; and Unity, a creation engine.

The practical part deals with the process of creating an educational application which focuses on experiential learning. 3D models created in Blender are used and the application itself is developed in the Unity creation engine. The created application was tested on students of a selected high school and the results were evaluated.

Keywords: virtual reality, education, school, teaching, learning, Blender, 3D model, Unity

ACKNOWLEDGEMENTS

I would like to express my most sincere gratitude to everyone who ever supported me over the course of my studies at the Faculty of Applied Informatics of Tomas Bata University in Zlín and especially during the work on this Master's thesis. Namely my wonderful parents, Ing. Lukáš Urbančok, who gave me the idea for the topic of this thesis in the first place, my supervisor Ing. Pavel Pokorný, PhD, the developers who made the software used and its easily comprehensible tutorials, the students who tried the educational application after it was finished, my employer CN Group for being understanding and giving me enough space to work on this thesis as well, and last but not least the artists whose music I listened to in order to concentrate better and do a great job.

Unthinking respect for authority is the greatest enemy of truth.

— Albert Einstein

CONTENTS

INTRODUCTION

With the success of the Oculus Rift at the crowd-funding platform Kickstarter in 2012 (raising US$2.4 million, almost tenfold of the initial goal of US$250,000 [7]) and with Google breaking new ground in 2013 by pioneering the Google Glass (an augmented reality solution), the world suddenly started paying attention to the technology of virtual reality that had been around for decades yet had failed to be commercially viable. [1]

The Oculus Rift research and development team progressed to something quite special, releasing the Oculus Rift headset. This was followed by the HTC Vive, which uses the Steam platform, and by the Microsoft HoloLens. Let us not forget Samsung Gear VR, Google Daydream headset and other cheaper headsets that allow the users to use a mobile device to immerse themselves into the world of virtual reality.

However, the most cost effective of them all is the Google Cardboard platform launched in 2014 at the Google I/O. This is the virtual reality solution that finally made this technology available to the masses, instead of keeping it in the hands of the somehow wealthier parts of population. The newfound availability is attractive not only for the private sector, but also for the field of education (either private or public). This Master's thesis explores the idea of virtual reality in education, specializing on high schools (while not omitting other levels of education) and describing the process of creating a virtual reality application.

I. THEORY

1 VIRTUAL REALITY

Virtual reality (VR) is a computer-generated scenario that simulates a realistic experience. The immersive environment can be similar to the real world in order to create a lifelike experience grounded in reality or sci-fi. Augmented reality systems may also be considered a form of VR that layers virtual information over a live camera feed into a headset, or through a smartphone or tablet device. [5]

Current VR technology most commonly uses virtual reality headsets or multi-projected environments to generate realistic images, sounds and other sensations that simulate a user's physical presence in a virtual or imaginary environment. A person using virtual reality equipment is able to "look around" the artificial world, move around in it, and interact with virtual features or items. The effect is commonly created by VR headsets consisting of a head-mounted display with a small screen in front of the eyes, but can also be created through specially designed rooms with multiple large screens.

VR systems that include transmission of vibrations and other sensations to the user through a game controller or other devices are known as haptic systems. This tactile information is generally known as force feedback.

1.1 History

The adjective "virtual" has been in use by the late 14th century. Since the middle of the 15th century, it has had the meaning "being something in essence or effect, though not actually or in fact" The term has been used in the computer sense of "not physically existing but made to appear by software" since 1959. [6]

In 1938, Antonin Artaud described the illusory nature of characters and objects in the theatre as "la réalité virtuelle" in a collection of essays, *Le Théâtre et son double*. The English translation of this book, published in 1958 as *The Theater and its Double*, is the earliest published use of the term "virtual reality". It was first used in a science fiction context in *The Judas Mandala*, a 1982 novel by Damien Broderick. [5]

At the 1939 New York World's Fair, the View-Master was introduced, the first precursor to today's head-mounted displays. It was a device for viewing photographs based on the principle of showing a different image to each eye to form a stereoscopic 3D picture. [1]

Figure 1: A View-Master Model E of the 1950s

In 1979 Eric Howlett developed the Large Expanse, Extra Perspective (LEEP) optical system. The combined system created a stereoscopic image with a field of view wide enough to create a convincing sense of space. The LEEP system provides the basis for most of the current virtual reality head-mounted displays available today.

When it comes to consumer virtual reality, Sega announced a head-mounted display add-on for the Sega Genesis game console called SegaVR in 1993. However, after technical difficulties and reports of headaches and nausea during playtesting, it never launched. Two other companies tried to penetrate the VR market in 1995, Nintendo and Forte Technologies. Alas, Nintendo Virtual Boy failed to reach sales targets because of its low quality monochromatic display, low processor speeds, lenses that caused eye strain and headache, slow head tracking and widespread nausea and sickness.

Figure 2: Sega VR and Virtual Boy

The VFX1 from Forte Technologies did not fare much better. The VFX1 was a head-mounted display that could play PC games like Doom, Magic Carpet, Descent and Quake on a regular IBM PC 386. It had built-in headphones, a microphone, head movement tracking on three axes and its display rendered 256 colors out to dual 0.7” 263×230 LCD displays at 60 Hz. It is the closest to today’s headsets, but it just did not sell, likely due to the lack of capable home computers. Neither the VFX1 nor its successor the VFX2 were popular enough to keep their maker in the business.

1.2 The Current technology landscape

During the first decade of the 21st century, the gaming industry seemed to have lost interest in the technology of virtual reality and considered it a bad business decision. [1] That changed with the arrival of the second decade of the third millennium when the crowdfunding website Kickstarter was taken by storm by the Oculus Rift in 2012, raising almost ten times its initial target of US$250,000 after receiving donations from 9,522 backers totaling US$2,437,429. [7] This success was not left unnoticed by the rest of the gaming industry and several other companies started developing their own virtual reality solutions, such as HTC, Sony, Google and Samsung. Oculus VR, LLC itself was bought by Facebook in 2014 for US$2.3 billion. [5]

1.2.1 Desktop virtual reality headsets

Three major desktop virtual reality headsets were released in 2016, two of them for PC and one for a video game console. The overview of the technical specifications of those virtual reality headsets follows, including their release date and MSRP.

Name	**Oculus Rift**	**HTC Vive**	**PlayStation VR**
Release date	28 March 2016	5 April 2016	13 October 2016
Display	2160×1200 px, OLED	2160×1200 px, OLED	1920×1080 px, OLED
Refresh Rate	90 Hz	90 Hz	90–120 Hz
Field of view	110 degrees	110 degrees	100 degrees
Weight	470 g (17 oz)	470 g (17 oz)	600 g (21 oz)
Platform	Oculus Home	SteamVR, VivePort	PlayStation
Tracking area	3 sensors-6 m^2 (64 ft^2)	21 m^2 (225 ft^2)	5 m^2 (50 ft^2)
Built-in audio	Yes	Yes	Yes
Built-in mic	Yes	Yes	Yes
Controller	Oculus Touch, Xbox One controller	Vive controller, any PC compatible gamepad	PlayStation Move controller
Sensors	Accelerometer, gyroscope, magnetometer, Constellation tracking camera.	Accelerometer, gyroscope, Lighthouse laser tracking system, front-facing camera	Accelerometer, gyroscope
Connections	HDMI, USB	HDMI, USB	HDMI, USB
Minimal requirements	NVIDIA GeForce GTX 960 or AMD Radeon RX 470 equivalent or greater	NVIDIA GeForce GTX 970 or AMD Radeon RX 480 equivalent or greater	PlayStation 4 with PlayStation Camera
	Intel Core i3-6100 AMD FX4350 or greater	Intel Core i5-4590 equivalent or greater	
	8GB of RAM	4GB of RAM	
	Compatible HDMI 1.3 video output	1× HDMI 1.3 or 1× DisplayPort 1.0	
	2× USB 3.0 port, 1× USB 2.0 port	1× USB 2.0 port	
	Windows 7 SP1	Windows 7 SP1	
Release price	US$400	US$800	US$400

Table 1: Desktop virtual reality headsets comparison [14] [13]

1.2.2 Mobile virtual reality headsets

Three major mobile virtual reality platforms were released since 2014, two by Google and one by Samsung in cooperation with Oculus. The overview of their specifications follows, including the release dates and MSRP of the headsets.

Name	**Google Cardboard**	**Samsung Gear VR**	**Google Daydream**
Release date	25 June 2014	27 November 2015	10 November 2016
Field of view	90–100 degrees	96–101 degrees	90–100 degrees
Weight	Depends on viewer, starting at 88 g (3 oz)	318 g (11 oz)	220–260 g (8–9 oz)
Platform	Cardboard	Oculus Home	Daydream
Controller	One hardware button	Trackpad, back button, volume rockers, handheld controller since 21 April 2017	Handheld controller with a touch pad, two buttons and volume rockers
Supported phones	Any phone with a gyroscope and Android 4.1+ or iOS 8.0+	Samsung's flagship models, starting with Galaxy S6	Requirements met by 15 Android phones as of May 2018
Release price	Depends on viewer, starting at US$5	US$100	US$79

Table 2: Mobile virtual reality headsets comparison [5]

The display parameters depend on the phone that is inserted into the viewer. In case of Cardboard, it can be virtually any display with the size between 4.5” and 6”. Gear VR and Daydream have higher demands on display quality and the resolution has to be at least 1920×1080 pixels.

Neither of the platforms supports area tracking (as opposed to the desktop VR platforms) because there is no base station to serve as a beacon for tracking the position of the user in a given area.

When it comes to audio, neither platform integrates the audio output nor a microphone into the headsets because both of those things are already integrated in the phones used with the headsets.

More detailed description of the Google Cardboard follows since it is the platform of choice for developing an educational application for virtual reality as a part of this Master’s thesis.

1.3 Google Cardboard

Google Cardboard is a low-cost virtual reality platform developed by Google. It was introduced at the Google I/O 2014 developers' conference. In 2015, the scope of the platform was extended by adding support for iOS devices, supporting wider range of Android devices, starting promotional partnerships and introducing related initiatives (filmmaking ecosystem Jump, educational program Expeditions). [5]

1.3.1 Head-mounted viewer

Google published instructions how to manufacture a head-mounted viewer from cheap materials, such as plastics (for mass production) or a literal cardboard (for DIY purposes), hence the name Cardboard. The user only needs to scan a QR code (which includes the parameters of a barrel distortion that should be applied to the rendered image in order to compensate for the lenses' pincushion effect) and insert their smartphone into the viewer.

1.3.2 Software support

Cardboard demonstration app was released for Android and iOS devices. The stereoscopic 3D effect is created by splitting the phone's screen into two halves, each rendering the image for one of the user's eyes. Google also released Cardboard software development kits (SDKs) for Android operating system (uses Java), Unity game engine (uses C#) and iOS devices. Plenty of third-party developers used it to create their own VR apps, published on Google Play and AppStore.

1.3.3 Limitations of the platform

As opposed to other VR platforms, the capabilities of Google Cardboard are more limited.

As with other mobile VR platforms (Google Daydream and Samsung Gear VR), the main problem is the refresh rate of the screens. Desktop VR platforms (Oculus Rift and HTC Vive) use screens with a refresh rate of 90 Hz but current smartphones almost exclusively make use of 60 Hz screens. This results in decreased fluency of movement, making it more likely for users to experience motion sickness while using the Cardboard headset. The only exception from this rule is the Razer Phone released in November 2017, featuring 5.7" LCD display with a resolution of 2560×1440 px and a refresh rate of 120 Hz.

The fluency of movement is also impacted by the level of precision of the phone's gyroscope. The gyroscopes in desktop VR platforms generally react more swiftly to user's movements.

Another visible drawback of most smartphones is their screen resolution. Only expensive high end models use screens with higher resolutions than FullHD. However, the prices of smartphones that do use a FullHD screen are currently starting at around €120 and the screen door effect (visible lines between pixels on a screen) with this resolution is tolerable.

Nevertheless, the biggest gap between the desktop and mobile VR platforms creates the hardware that powers them. Desktop PCs could easily be ten times more powerful than current smartphones. The level of detail of the rendered graphics is at equally different levels.

The second but last thing that is missing from the Cardboard platform is a pointing device independent on the movements of user's head. Desktop VR platforms, Samsung Gear VR and Google Daydream all offer such pointing device. However, only desktop VR platforms can maintain the calibration of the pointing device because of their base stations. On mobile VR platforms, the calibration of the pointing device can be progressively lost, resulting in different direction of pointing than intended by the user.

The last feature that all mobile VR platforms are lacking is the support of base stations. Desktop VR platforms detect the user's movements using those base stations and also allow them to move around the room. Whereas in mobile VR platforms, user's movements around the room are left undetected. Those base stations also keep the calibration of user's head movements and pointing devices stable during the session.

1.4 Virtual reality in education

1.4.1 Introduction of VR to lessons

Virtual reality allows the students to move away from simply 'learning' a subject or topic to 'feeling' the content. This is not simply an engagement tool or a gimmick, it facilitates the exploration and experiencing or being involved in something, as if the students were actually present in the virtual environment or place. [11]

1.4.2 Benefits of VR to learning

It is often more fruitful to see and hear something than have it explained. Students sometimes just need to be taken out of the school environment and put into an immersive world where they can experience a performance in a concert hall or in a theatre, watch dinosaurs walk around, or live stream with others around the world in virtual social spaces. The number of educational applications for virtual reality being developed and published is rising rapidly, and with every new piece of software, another avenue for learners opens up.

1.4.3 Barriers to adoption

The technology of virtual reality is still evolving. Even though there are many virtual reality videos and applications, not all of them are genuinely useful in the classroom. Furthermore, the cost of most virtual reality platforms is prohibitive and the set-up more or less clunky. There is not a device out there which does not have a significant constraint attached for school acquisition and inclusion in the curriculum, with the Google Cardboard being the closest to the ideal platform of choice due to its low-cost nature.

1.4.4 Examples of schools using VR

Washington Leadership Academy in Washington D.C. uses virtual reality and computer science to supplement the studies. The school uses technology and online courses to meet students at their level, rather than teaching an entire grade the same content. [10]

At Sevenoaks School (a co-educational day and boarding school in the United Kingdom), there is ongoing research into the technology of virtual reality. The researchers are experimenting with using VR for educational purposes and share their findings with other teachers and schools to further progress the use of this technology at school grounds. [11]

A Japanese corporation specializing in entertainment and technology by the name of Kadokawa Dwango has established a virtual reality high school called the N High School in April 2016. The school itself is a combination of online and physical teaching methods, with most lectures available via online video streaming. This allows the students to watch them when they find it convenient and then take tests after each session. [9]

1.4.5 Existing educational VR applications and their structure

Expeditions is an educational application made by Google for Cardboard. Students move through an experience that their teacher controls from a tablet. The Expeditions program has distributed the equipment to classrooms and worked with teachers to figure out lesson plans. Now any teacher with a tablet and access to VR viewers can use it. With lessons loaded on the tablets, teachers and students do not need to have internet access. [12]

Once students put on the VR headsets, they are immersed in a 3D version of any historical sight. They can look around, and the teacher can share information about things they are seeing. For example, Google built a Great Wall of China experience for a fifth grade math class, to give the students a more tactile lesson about multiplication.

VR Solar System is another example of an educational VR application for Cardboard. Available for free from Google Play, it takes the user on a virtual journey through the solar system. The voyage begins at Earth and ends at Pluto (even though that is no longer considered a planet since 2006, as mentioned in the app). At every planet, a narrated overview of the basic information about that planet is played back as the user gazes upon its realistic 3D rendition.

Mozaik Education is a developer publishing free educational VR applications at Google Play. They have published several applications, each presenting a certain topic, such as the human body, the four-stroke engine, the Acropolis of Athens and others. All the apps have similar structure – the user selects through a menu which part of the topic they want to explore and a 3D presentation of that part then begins, with a narrated voiceover providing useful information.

2 BLENDER

2.1 General characteristics

Blender is a multiplatform open source application dedicated to creating 3D models and animations, rendering, post-production activities and interactive applications. Its source code and builds for all three major platforms are available for download from the website of the Blender Foundation. [4]

Multiplatform means that Blender can be run on Microsoft Windows, Linux, Mac OS X and other platforms. Open source means that the program is not only completely free of charge for commercial use, but also that its source code can be downloaded, compiled with any given platform’s compiler for better performance, customized and anyone can participate in Blender’s development.

Blender is suitable for animators, graphic artists or designers looking for a powerful tool to visualize their ideas and projects while not spending money on commercial 3D applications. Over the course of its development, Blender has been offering tools that are comparable to midrange commercial 3D software.

New versions are generally released to public every 3-4 months, and in addition to a number of minor enhancements and bugfixes, they also bring new tools and features that follow the current development of users’ needs.

In addition to the tools for modeling, animation and rendering, Blender also includes Game Engine that can be used for creating interactive presentations, walk-through visualizations (e.g. of building interiors) and computer games. It is built into Blender’s internal graphical editor with the ability to add code written in the programming language Python.

In addition to internal scanline and pathtrace renderers, Blender also offers direct output to the YafaRay external raytrace renderer, which is also open source and hence for free.

Blender can be complemented by a number of Python script extensions, which include very complex plugins, such as trees, grass, animal fur and import and export filters for compatibility with other applications.

Other extensions are possible via material or sequential (postproduction) plug-ins provided in the form of library files (such as .dll).

2.2 Interface

The Blender interface is relatively original and especially for users coming from other 3D applications, it may be somewhat confusing. However, getting used to the philosophy of its layout does not take too much time since it is quite intuitive, effective and allows to create models and animations quickly, naturally and without the need to think intensely about where to find a feature hidden in a few submenus.

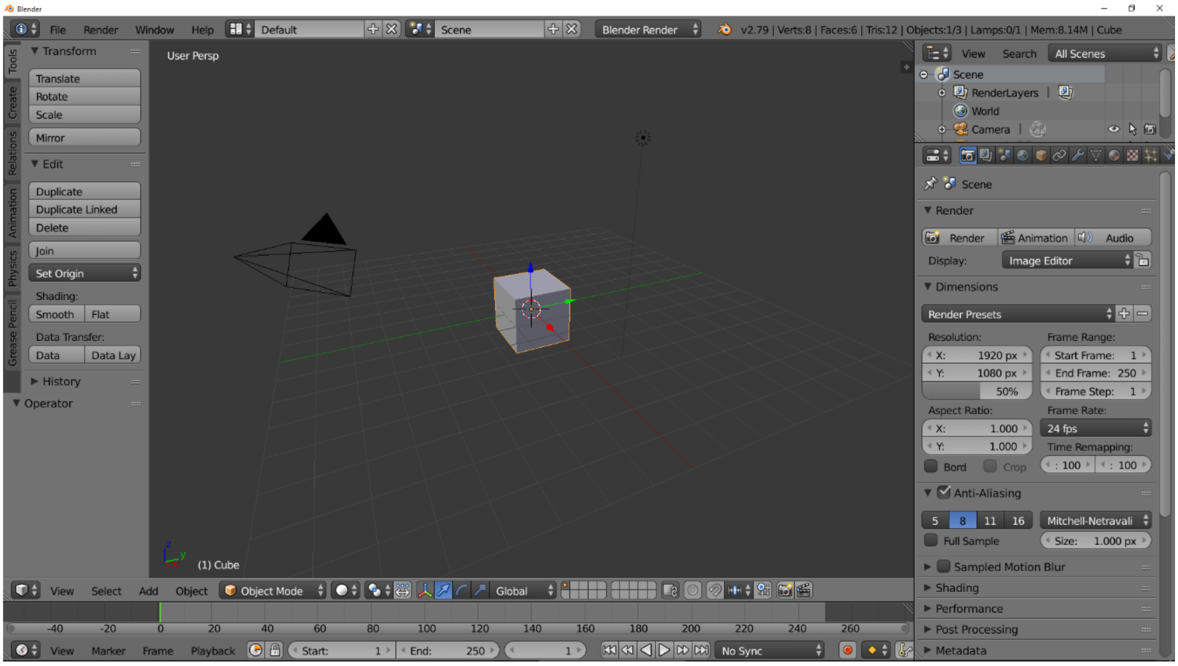


Figure 3: Main window of Blender, version 2.79

The main attributes of the Blender interface are:

- fully adjustable work area
- divided into windows for modeling, animation curves, outliner, nonlinear video editing, UV map editing, animation of characters (pose editor, NLA editor), file manager
- database system for optimal management of scene, instance and dynamic linking of projects in different files
- localization into multiple languages, including the ability to turn on partial localization (e.g. a button label in English with the button's explanation in another language) and the ability to create your own language sets
- built-in text editor for taking notes and programming Python scripts
- interface is the same on all platforms

2.3 Modeling

- supports polygons, Nurbs surfaces, Bezier and B-spline curves, metaballs and vector fonts (TrueType, PostScript and OpenType)
- Catmull-Clark surfaces (meshsmooth equivalent) with editable edge sharpness
- polygonal mesh editing with optional selection of vertices, edges or faces
- Boolean mesh operations
- editing functions such as extrude, bevel, cut, spin, screw, warp, subdivide, noise, smooth… the list continues and expands with new releases
- the ability to program Python modeling tools as needed

2.4 Animation

- armature deformation (bones, skeletons) with both forward and inverse kinematics, autoskinning and interactive weighing of deform groups using the WeightPaint tool
- several types of constraints for rigging, pose editor
- editor of nonlinear animation (NLA), automation of movement of a character with a walkcycle along a defined path
- vertex keys and relative vertex keys animation (like morph targets) with sliders
- particle effects with deformers by wind, gravity, magnetism and collision detection
- SoftBodies (e.g. cloth simulation) with collision detection
- animated lattice deformation
- support for “motion curve” and traditional key-frame editing
- audio support and tools for audio/video synchronization
- possible to program animation tools or “controlled animations” in Python as needed

2.5 Render

- a choice of 3 rendering engines – Blender render (internal scanline render), Cycles (internal pathtracer, can use GPU) and direct access to YafaRay (external raytracer)
- oversampling, motion blur, postproduction effects (glow, zBlur), non-square pixels
- environment maps, halo, lens flare, fog…
- several material shaders for diffusion and specular shaders – Lambert, Oren-Nayar, Minnaert, Fresnel, Cook-Torrance, Phong, Blinn, Toon, Ward Isotropic
- Edge rendering for cartoon effect
- Procedural textures, ambient occlusion, radiosity

- A number of export scripts to other raytracers, such as PovRay, Renderman, Lightflow or Virtualight
- UV editor with several methods for unwrapping (e.g. very effective LSCM)

2.6 Interactive applications – realtime 3D and gaming

- graphical editor for programming the logic of an application or a game without the need for programming
- collision detection and dynamics simulation
- accessible engine via Python scripts for more complex logic or artificial intelligence
- support for all OpenGL surface modes, including transparency, animated reflex maps
- playing games and interactive 3D applications without compiling and pre-computing
- audio using SDL toolkit
- multi-layering of scenes for a floating interface

2.7 Files and supported formats

- all data in the scene is stored in a single file with the extension ".blend"
- .blend format supports compression, digital signatures, encoding, forward and backward compatibility and can be used as a library accessible from another file
- reads/writes TGA, JPEG, PNG, Iris (+ Zbuffer), SGI Movie, IFF, AVI, GIF, TIFF, PSD, MOV…
- native support for importing and exporting DXF, Inventor and VRML files, import and export to a number of other formats (OBJ, LWO, COB…) is made possible via Python scripts, the main ones are already included in Blender's base package
- interactive 3D applications and games can be saved as executable files (.exe) or played in a web browser using a plugin

2.8 Supported platforms

- Windows Vista, 7, 8.1 and 10
- Mac OS X 10.6 (Snow Leopard) and later
- Linux
- FreeBSD
- compilable on other platforms as well

3 UNITY (GAME ENGINE)

Unity is a cross-platform game engine developed by Unity Technologies, which is primarily used to develop both three-dimensional and two-dimensional video games and simulations for computers, consoles, and mobile devices. First announced only for OS X at Apple's Worldwide Developers Conference in 2005, it has since been extended to target 27 platforms. Six major versions of Unity have been released. [5]

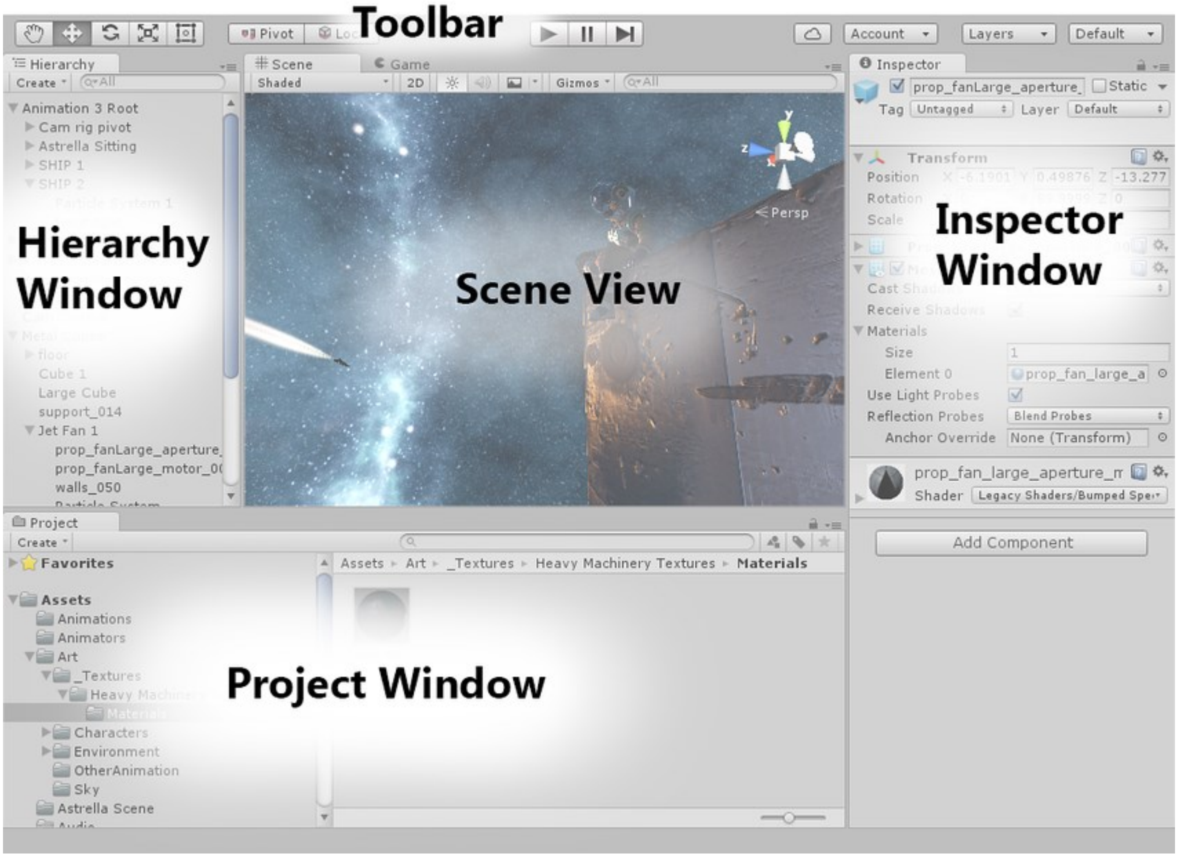


Figure 4: Unity's interface overview

3.1 Overview

Unity is a multipurpose game engine that supports 2D and 3D graphics, drag-and-drop functionality and scripting using C#. Two other programming languages were supported: Boo, which was deprecated with the release of Unity 5 and JavaScript which started its deprecation process in August 2017 after the release of Unity 2017.1.

The engine targets the following graphics APIs: Direct3D on Windows and Xbox One; OpenGL on Linux, macOS, and Windows; OpenGL ES on Android and iOS; WebGL on the web; and proprietary APIs on the video game consoles. Additionally, Unity supports the low-level APIs Metal on iOS and macOS and Vulkan on Android, Linux, and Windows, as well as Direct3D 12 on Windows and Xbox One.

For 3D games, Unity allows specification of texture compression, mipmaps, and resolution settings for each platform that the game engine supports, and provides support for bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render-to-texture and full-screen post-processing effects. Unity also offers services to developers, these are: Unity Ads, Unity Analytics, Unity Certification, Unity Cloud Build, Unity Everyplay, Unity IAP, Unity Multiplayer, Unity Performance Reporting and Unity Collaborate.

Unity supports the creation of custom vertex, fragment (or pixel), tessellation, compute shaders and Unity's own surface shaders using Cg, a modified version of Microsoft's High-Level Shading Language.

3.2 Editions

Unity comes in three editions, Unity Personal, Plus and Pro. [8] Unity Personal is intended for beginners, students, and hobbyists and can be used for free if the following criteria are met:

- The users do not make more than $100k in annual gross revenues, regardless of whether Unity Personal is being used for commercial purposes, or for an internal project or prototyping.
- The users have not raised funds in excess of $100k.
- The users are not currently using Unity Plus or Pro.

3.3 Supported Platforms

Unity supports building to 27 different platforms, including native support for virtual and augmented reality. The platforms are listed in the following: iOS, Android, Tizen, Windows, Universal Windows Platform, Mac, Linux, WebGL, PlayStation 4, PlayStation Vita, Xbox One, Wii U, 3DS, Android TV, Samsung Smart TV, tvOS, Nintendo Switch, Fire OS, Facebook Gameroom, Vuforia and those virtual and augmented reality platforms: Oculus Rift, HTC Vive, GoogleVR (Daydream and Cardboard), Steam VR, PlayStation VR, Gear VR, Apple ARKit, Google ARCore, and Windows Mixed Reality devices.

II. ANALYSIS

4 APPLICATION DEVELOPMENT PREPARATIONS

4.1 Content

First, the content of the educational application had to be decided. The content should provide some added value to the lesson where it is used. After considering several options, the final decision was based on a personal experience with a 1:30 model of a house connected to a Saia PCD2 in a programmable logic controllers (PLC) lab.

Figure 5: 1:30 model of a house connected to a Saia PCD2

During the initial exploratory phase of learning how the model could be programmed for the purposes of house automation, the students can look at the model from the outside as much as they want, but since it is currently not possible to shrink a human being to one 30th of their size, they will never be able to walk through the model of a house. This makes it difficult to imagine what it is like to be inside of the house and assess how the house automation system should behave. The educational application facilitates the exploration of the house and the planning phase of programming of the house automation system.

Other models that are installed in the PLC lab include a fountain with two water circuits and two sets of lights, bottle filling line, automated warehouse and others. Those models could be used in the educational application by giving the students the ability to disassemble them and learn how the separate parts have to work in order for the model to function as a whole.

Even though the PLC lab in question is located at the Faculty of Applied Informatics of Tomas Bata University in Zlín and university students work with the equipment, similar PLC labs can be found at technical high schools, with similar (sometimes even more advanced) equipment and degree of difficulty of the educational tasks performed with the models. Therefore, the educational application can be used at both high schools and universities.

4.2 Structure

The structure should be simple enough in order not to burden the students with additional need for learning how the application works. Hence, the application starts directly at the front yard of the house in question. There are floating blue cubes scattered around that work as teleport points. After aiming at a cube and clicking on it, the user is teleported to the cube's location. This mode of transportation was chosen in favor of continuous movement to lower the risk of VR Sickness. This kind of nausea can occur when the real-world user's body is stationary while their virtual point-of-view moves around a virtual environment.

4.3 Location documentation

Second, the location that was recreated into virtual reality had to be well documented and measured. Combination of Asus Zenfone 2 and Xiaomi Redmi Note 4 was used to take 24 photos and two videos. The dimensions of the real lab and its equipment were measured by a laser measuring device and the dimensions of the real house model were measured by a retractable tape measure. Those dimensions were written by hand into the taken photos.

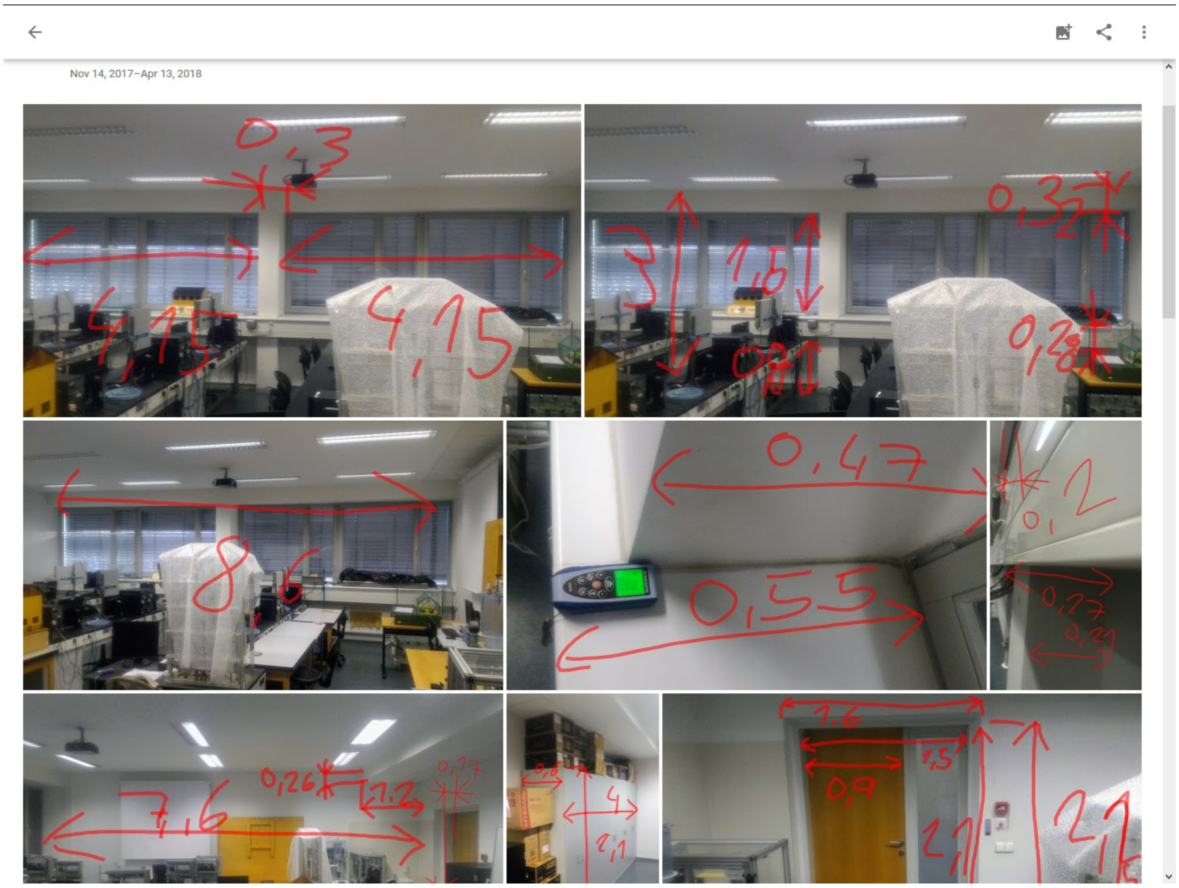


Figure 6: Google Photos collection of location photos with written dimensions

4.4 Target platform

The selection of the target platform was based on multiple criteria. The most important one was the cost of the platform since high schools usually have a rather limited budget. When the fact that there would have to be several headsets purchased for the whole class is taken into account, the significance of cost effectiveness is even more apparent. This rules out the choice of desktop virtual reality headsets (HTC Vive and Oculus Rift) since they are not only expensive themselves, but also require a powerful (hence expensive) PC to run on. These headsets are also not very user friendly because of the cables that connect them to the PC and put the user in a risk of tripping over.

The mobile VR platforms Samsung Gear VR and Google Daydream are not exactly cheap either. The combination of a headset and a cheap compatible smartphone costs around €400 as of May 2018. While they do offer the advantage of supporting a handheld remote control that allows the user to point the cursor at a different direction than where they are looking, this remote control has to be paired with the smartphone first, its batteries have to be maintained, the calibration might get lost over runtime and it could also easily get missing in a school environment with high fluctuation of users.

All those reasons led to the choice of Google Cardboard platform. It offers an abundance of cheap viewers, such as the VR BOX VR-X2 that was bought for €10 in October 2017.

Figure 7: VR BOX VR-X2 white

Save for the low price, another advantage of this viewer is its top head strap which contributes to the stability of the viewer on user’s head, complemented by the low weight. However, that is where the advantages end. The wearing comfort is worsened by an insufficient nose notch and the usability is significantly hindered by a lack of hardware button (can be solved by connecting a Bluetooth mouse to the smartphone) and too much of a distance between the lenses and the smartphone’s screen. That compromises the user’s feeling of being immersed – present in the virtual environment. The lenses are adjustable which might sound like an advantage but is actually the opposite because it requires the user to spend some time finding the appropriate position of the lenses instead of relying either on their healthy eyesight or on an already existing sight correction (via dioptric glasses or contact lenses).

4.5 Hardware used

4.5.1 Smartphone

The smartphone that was used during the course of development of the educational application is Xiaomi Redmi Note 4. It was bought new for €140 in February 2018. The technical specifications are as follows. [15]

Display	5.5 inches, 1080×1920 pixels, 401 PPI density
System on a chip	Snapdragon 625 (8×Cortex-A53@2.0 GHz, Adreno 506@650 MHz)
Memory	32 GB storage, 3 GB RAM
Operating system	Android 7.0 (Nougat), ROM MIUI 9.5
Battery	Li-Pol 4100 mAh, non-removable
Size and weight	151×76×8.5 mm (5.94×2.99×0.33 in), 165 g (5.82 oz)

Table 3: Technical specifications of Xiaomi Redmi Note 4

4.5.2 Headset

For the purpose of the development of the educational application, Xiaomi Mi VR Play 2 headset was bought new for €30 in March 2018. All the disadvantages of VR BOX VR-X2 (described in chapter 4.4) were redeemed while Xiaomi Mi VR Play 2's only disadvantage, the absence of the top head strap, was not found to be an issue. The superior overall comfort and user-friendliness of this headset justify the €20 price difference.

Figure 8: Xiaomi Mi VR Play 2 dark grey

5 MODELLING IN BLENDER

Online video tutorials that were created in 2011 at the Faculty of Applied Informatics of Tomas Bata University in Zlín were used to learn how to create 3D models in Blender v2.79. [2] To make the workflow more user-friendly, left mouse button and right mouse button were swapped (File → User Preferences → Input → Select With: Left) and a plugin called rRMB was installed. [16] It adds a context menu with numerous handy commands that are otherwise not be easily accessible in Edit Mode, such as “Move Origin To Selection”.

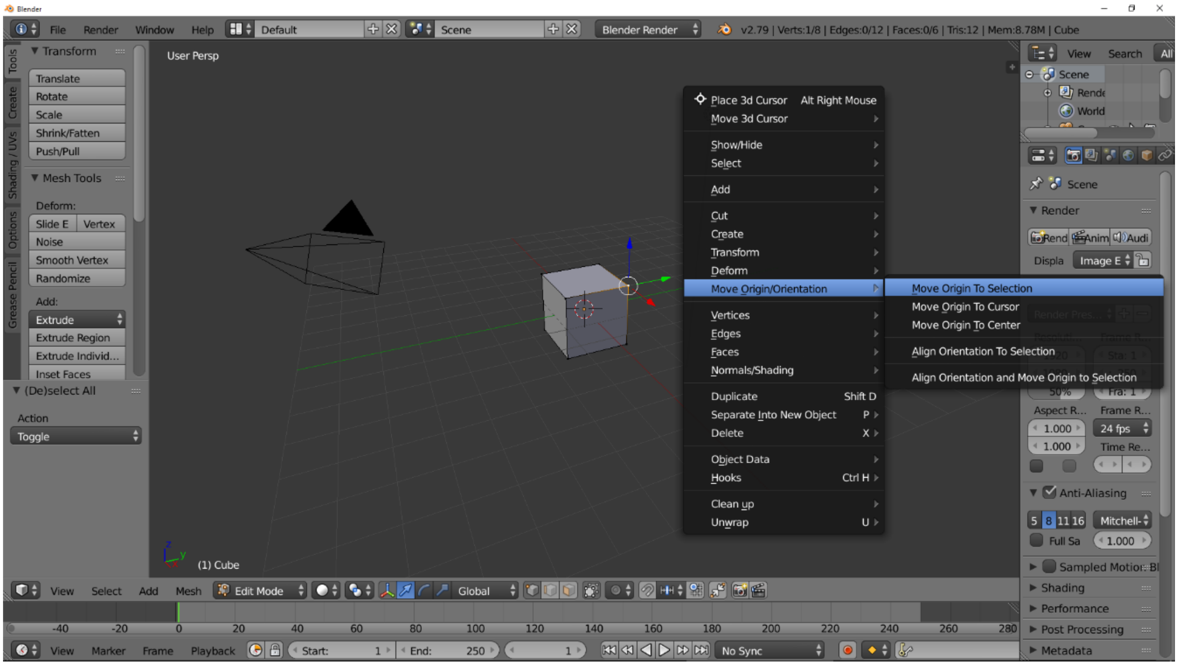


Figure 9: rRMB plugin’s context menu in Blender

5.1 Modelling and manipulation techniques

The 3D models in Blender were created by combination of a number of modelling techniques. Mesh primitives (such as planes, cubes and cylinders) were created via the Add Meshes menu on the left side. Those meshes were then scaled according to the dimensions of the real-world lab and its equipment. The translation of those meshes (either as a whole or of their vertices, edges or faces) was usually done only in one of the three axes by pulling the arrow of the axis and typing the desired numerical value on the numerical keypad. Another useful way of moving the objects around is by snapping them to the 3D cursor which was moved to another object first (right mouse button → Move 3D Cursor → To Selected). Then the object that needs to be moved can be selected and snapped to the 3D cursor (Shift+S → Selection to Cursor). Note that the origin of the object will be snapped to the 3D cursor.

Should any object need rotation in only one of the axes, it can be done by pressing the corresponding letter on the keyboard first and then typing the rotation degrees on the numerical keypad.

Extrusion was also used to create new faces on some meshes. It works by selecting an existing face or at least an edge and pressing E in the Edit Mode. The newly created face or faces can be then translated in only one axis pressing its letter on the keyboard.

When two or more objects need to be joined, they need to be selected first in the Object Mode and then Ctrl+J is pressed. It can often happed, especially after snapping one object to another, that there are going to be overlapping vertices in the joined objects. Those can be removed in the Edit Mode by the command Remove Doubles in the left side Tools menu.

Another useful feature of Blender is the Fill command. It can be invoked by pressing the F key after two or more vertices were selected in the Edit Mode. If only two vertices were selected, an edge is created between them, otherwise a face is created between three or more vertices.

When it comes to sculpting the shape of an object by the shape of another one, Modifiers can do the job. The object that needs to be modified can be assigned the Boolean Modifier, where the modifying object is then selected. If it is going to subtract from the modified object, Difference Operation is selected. If it adds faces to the modified object's mesh, Union Operation is chosen. Intersect Operation as also available, but was never used during the works within this Master's thesis.

After using the Boolean Modifier, unwanted vertices sometimes came into existence. Those could not be deleted just by selecting them, pressing X or Delete and choosing Vertices though because that would lead to the deletion of the edge they belong to. Instead, the command Dissolve Vertices has to be used, also available after pressing X or Delete.

In the Delete menu, another important command is found, Only Faces. As the name suggests, it deletes only the selected faces, not the edges that surround them. This is particularly useful when getting rid of faces that will never be seen and are therefore counterproductive to be rendered, as explained by the Unity Tutorial on VR optimization. [3]

At times, the quality of some objects needed to be lowered. The Decimate Modifier serves this purpose, lowering the number of faces of an object to a number that can be controlled by the user in multiple ways.

5.2 Materials

The models created in Blender were not textured because texturing was done in Unity. However, not every surface needed a texture, some of them were only colored. This was done both in Unity and Blender.

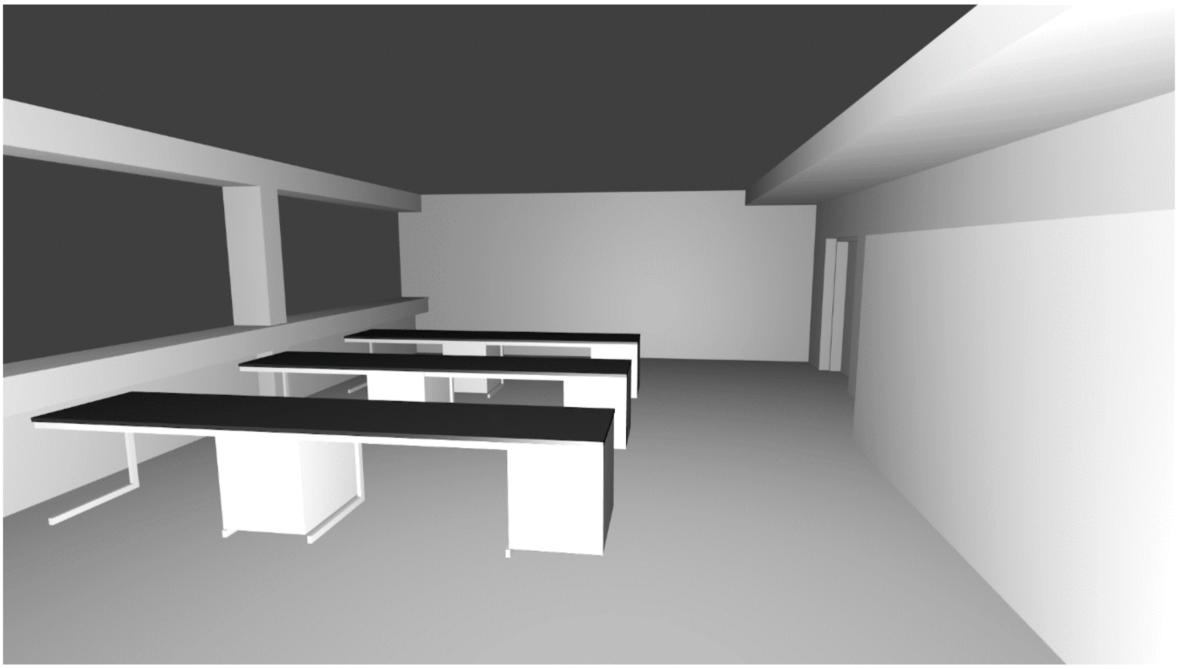


Figure 10: Early render of the 3D model of the PLC laboratory

Every mesh created in Blender is automatically assigned a material with an 80% white color. This color can be easily changed, which was done with the walls, desks and LCD monitors found in the lab. The walls were brightened to 88% white, desk were colored black, grey and white and the monitors were made black.

5.3 Preparation for texturing

In order to prepare the 3D models for texturing, some more alterations had to be made.

Faces that were supposed to receive a texture that the rest of the object was not supposed to have, were selected in Edit Mode and separated as another object by pressing P and choosing Selection.

Another problem that arose was incorrect orientation of some of the faces. In order to flip a normal of a face, the key W has to be pressed and Flip Normals has to be selected. [17] Or the rRMB plugin can be used: right mouse button → Normals/Shading → Flip Normals.

The last and very important thing is to assign a UV map to every object that needs to be textured. This can be done by selecting an object in Object Mode, pressing space, writing “UV map” into the search field and choosing Add UV Map from the results.

5.4 Readymade models

The internet is full of 3D models already made in Blender and available for download, some of them even for free. One of the websites offering such services is BlendSwap.com. The 3D model of LCD monitor used in the virtual lab was downloaded from this website. [18]

It was simplified to contain only 111 faces and also complemented by a keyboard made previously during learning how to use Blender by following online video tutorials. [2] The number of faces rose from 111 to 125 after the keyboard was included.

Figure 11: 3D model of LCD monitor from BlendSwap.com before simplification

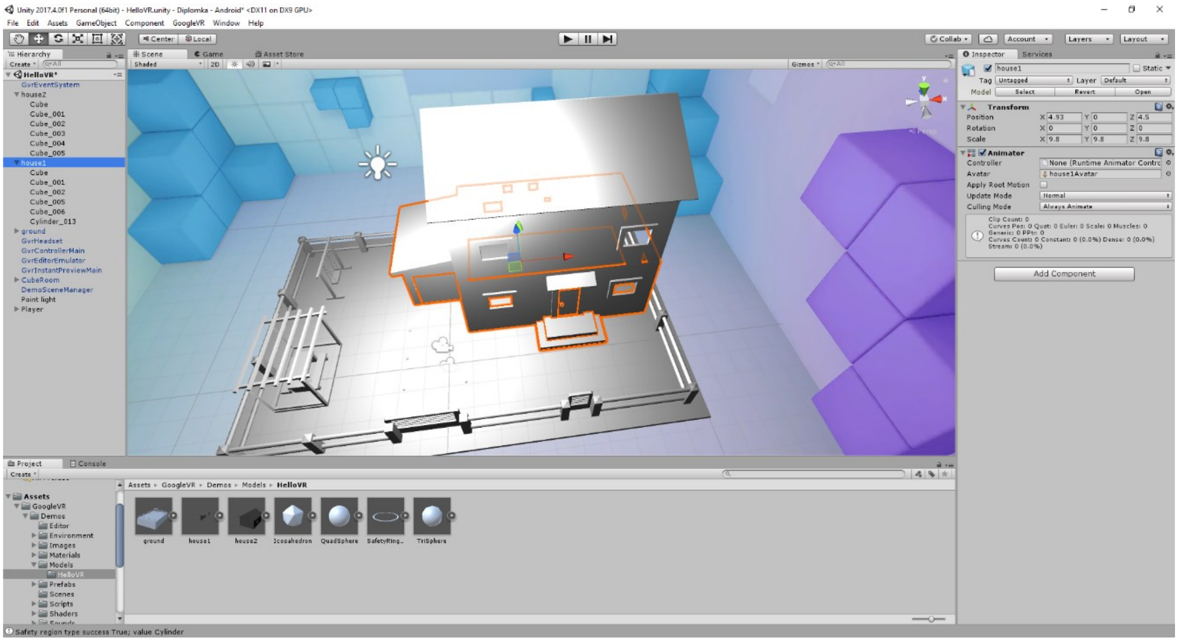
6 APPLICATION CREATION IN UNITY

6.1 Developer tools installation

Unity Personal can be downloaded from the Unity webpage. Versions 2017.4 and 2018.1 were used during the development of the educational application. The conversion of the project from version 2017.4 to 2018.1 did not break anything, the only thing that needed to be done was reopening the working scene.

Unity requires Android Build Support, Android SDK and Java Development Kit in order to be able to build apps for Android. Android Build Support has to be selected during the installation of Unity, Android SDK can be installed either using Android Software command line tool or as a part of Android Studio and Java Development Kit can be downloaded from Oracle's website. The installation paths to Android SDK and Java Development Kit have to be provided to Unity in Edit → Preferences → External Tools → Android. [19]

6.2 Importing 3D models from Blender



house1 ☐ Static
Tag: Untagged, Layer: Default

Transform
Position X 4.93 Y 0 Z 4.5
Rotation X 0 Y 0 Z 0
Scale X 9.8 Y 9.8 Z 9.8

Animator
Controller: None (Runtime Animator Contro
Avatar: house1Avatar
Apply Root Motion: ☐
Update Mode: Normal
Culling Mode: Always Animate

Clip Count: 0
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 0
Generic: 0 PPtr: 0
Curves Count: 0 Constant: 0 (0.0%) Dense: 0 (0.0%)
Stream: 0 (0.0%)

Safety region type success True; value Cylinder

Figure 12: Positioned 3D model imported from Blender into Unity

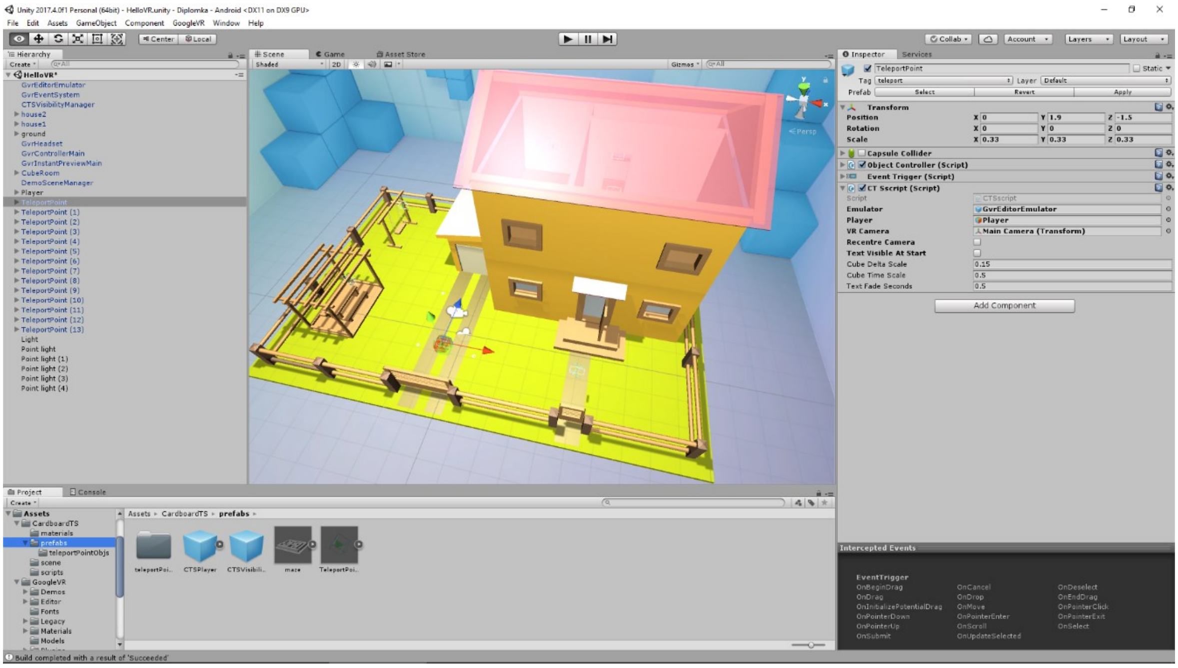
The 3D models from Blender were imported into the project simply by copying the .blend files into a Models folder found in the project's Assets folder. Unity automatically triangulates any non-triangular faces of the models without any change of the model's shape (this can also be done manually in Blender by pressing Ctrl+T in Edit Mode with non-triangular face(s) selected). A very useful feature of Unity is that it automatically reimports any files that have been modified after initial import into a project. Therefore, should the necessity arise, the user can open any .blend file for editing by simply locating it in the Project window, double-clicking on it and after the file is saved in Blender and the user returns to Unity, the 3D model is automatically reimported, but it retains the parameters assigned to it (such as position, rotation, scale, materials etc.).

Upon import, the 3D models need to be dragged & dropped into the Hierarchy window to create an instance. Only the instances can be used in the scene. One 3D model can be instantialized multiple times and each instance can be assigned different parameters. However, if the source file changes, all of its instances are going to change as well.

All of the 3D models were positioned, rotated and scaled as necessary. When a part of any 3D model needs independent positioning, rotating or scaling that can be done in Unity, but would require breaking the instance, it is recommended to do it in Blender instead.

6.3 Materials and texturing

6.3.1 Solid color materials



☑ TeleportPoint ☐ Static
Tag: teleport Layer: Default
Transform
Position X 0 Y 1.9 Z -1.5
Rotation X 0 Y 0 Z 0
Scale X 0.33 Y 0.33 Z 0.33
☐ Capsule Collider
☑ Object Controller (Script)
Event Trigger (Script)
☑ CT Sscript (Script)
Script: CTSscript
Emulator: GvrEditorEmulator
Player: Player
VR Camera: Main Camera (Transform)
Recentre Camera ☐
Text Visible At Start ☐
Cube Delta Scale: 0.15
Cube Time Scale: 0.5
Text Fade Seconds: 0.5

Build completed with a result of 'Succeeded'

Figure 13: Solid color materials applied to the imported 3D models

Any materials created and assigned to the 3D models in Blender are imported to Unity as well. However, they are not editable and only the color of the materials is imported, not any other parameters. Therefore, the Blender materials were used only for some 3D models, namely the walls, desks and LCD monitors.

Some other 3D models (windows of the lab, walls and roof of the model of a house) were assigned a Unity material. It has to be created in the Project window, named and then edited via the Inspector window. If it is going to be only colored and not textured, the Shader type should be Standard. Rendering mode should be either Opaque or Transparent if transparency is sought after. Then the Albedo color has to be chosen, the levels of Metallic and Smoothness can be adjusted. Different values of Tiling than X = 1 and Y = 1 does not make sense because the tiling effect would not be visible on a solid color surface.

6.3.2 Texturing

Wooden materials, grass, roads, carpet and control panel on the model of a house, keyboards and lab floor were textured. The procedure for texturing also begins with creating a new material in the Project window. A Mobile/Diffuse Shader shall be selected. Then a source image file has to be located. Both the X and Y dimensions of the image have to be a 2^n number, where n is an integer. Unity will automatically enlarge any images that do not conform to this requirement. Tiling can be used to create an illusion of a high resolution texture while keeping the application's memory footprint low. However, usage of seamless texture is imperative for this illusion to work. This was successfully done with the carpet, grass and roads at the model of a house and the lab floor. The only adjustment that had to be done on the source images was dimming their brightness to 50% because they appeared too bright in Unity thanks to the lighting technique used.

6.3.3 Skybox

Skybox is a special kind of texture spherically encompassing the scene, creating an illusion of a sky. The skybox used was downloaded from Unity Asset Store, imported and Skybox_Daytime.mat was used as Skybox Material in Environment settings, found in Window → Lighting → Settings. Main Camera's Clear Flags were also set to Skybox. [23]

6.4 Lighting

The scene is lit by seven purely white (RGBA 255, 255, 255, 255) point lights, with the range set to 333. They are all baked because realtime lighting costs more GPU time. The intensity of the lights is set to 1, except for two lights closest to the window, their intensity is 2 to simulate the light coming by the windows from the outside. Indirect Multiplier is always set to 0.

The lights cast no shadows because it turned out to be rather difficult to create realistic looking lighting and realistic looking shadows at the same time. It is not that difficult to create realistic lighting when the Render Mode of the lights is set to Not Important, but then the lights do not cast shadows. When their Render Mode is switched to Important, they do cast shadows, but it is hard to create life-like lighting conditions in Gamma color space (set in Edit → Project Settings → Player → Settings for Android → Other Settings → Rendering).

This color space was chosen in favor of Linear color space which makes it relatively easy to make both lighting and shadows appear believably realistic, even with the Render Mode of the lights set to Important, but is not supported by the graphic API OpenGL ES 2.0 nor WebGL 1.0. Absence of targeting those two APIs would leave legacy Android devices and legacy internet browsers unable to render the educational application. However, despite the absence of shadows, all objects still does not seem to be floating above the ground (common problem caused by the absence of shadows in 3D graphics) because there is a good sense of space awareness in the educational application created by the stereoscopic way of rendering.

6.5 Mode of transportation

There are several available modes of transportation in virtual reality. They differ in two distinct features: continuity of movement and freedom of choice where to go. As mentioned in chapter 4.2, continuous movement can cause motion sickness because the human brain gets confused by conflicting information. It is visually told that the person is moving while their body is actually stationary. Usually, people who get sick in a car also get sick from continuous movement in virtual reality. Therefore, teleporting was chosen as a way of moving around the virtual environment.

Since the students are supposed to learn about specific parts of the models explorable in the educational application, there is no point in letting them go wherever they want to. Instead, there are fixed clickable teleport points that keep the students' attention where it is needed.

Both of those requirements were met by an asset called Cardboard Teleport System that was bought from the Unity Asset Store for US$12 (incl. VAT) and implemented into the educational application by creating 14 teleport points according to the video tutorial provided by the author of the asset. [22] Also, the 3D model of a cube, which the user can click on in order to teleport to a particular position, was changed. Simplification in Blender brought down the number of its faces to 15% – from 624 to 96 (both values after triangulation).

6.6 Building the application

6.6.1 Android build

The development of the educational application can happen entirely in Unity since it can be previewed in the Game window after pressing the Play button from the Toolbar (mouse look simulating head movements can be enabled by holding down the Alt key). However, the command Build & Run (Ctrl+B) from the File menu was regularly used to build, install and run the application directly at the target device. In order for this to work, USB debugging has to be turned on. On Xiaomi phones, this is done by entering Settings → About phone, tapping MIUI version 7 times to enable Developer options and entering them (they will appear in Additional settings). USB debugging has to be enabled, along with Install via USB.

In Unity → Edit → Project Settings → Player → Settings for Android → Other Settings → Identification, the Package Name has to be changed from the default name to something else, in this case com.senovsky.virtualab. The Company Name and Product Name in Inspector → PlayerSettings can be changed accordingly as well. Senovsky and VirtuaLab names were used for the educational application (the Product Name is also used as a name of the application once installed on an Android phone). Then a compatible Android phone has to be connected via USB (drivers should be installed automatically by the operating system upon first connection) and the Build & Run command shall be issued.

On Xiaomi Redmi Note 4, an install prompt will appear when Unity will attempt to install the application on the phone over ADB for the first time. [20] This prompt has to be confirmed only once, unless the Product Name will change. Note that if the Android SDK gets reinstalled, the application has to be uninstalled from the Android phone in order for the Build & Run command to work again. The correct setting for barrel distortion was found not to be the one contained in Mi VR Play 2’s QR Code [21], but simply the default setting called Google Cardboard. It is also recommendable to turn screen brightness to maximum.

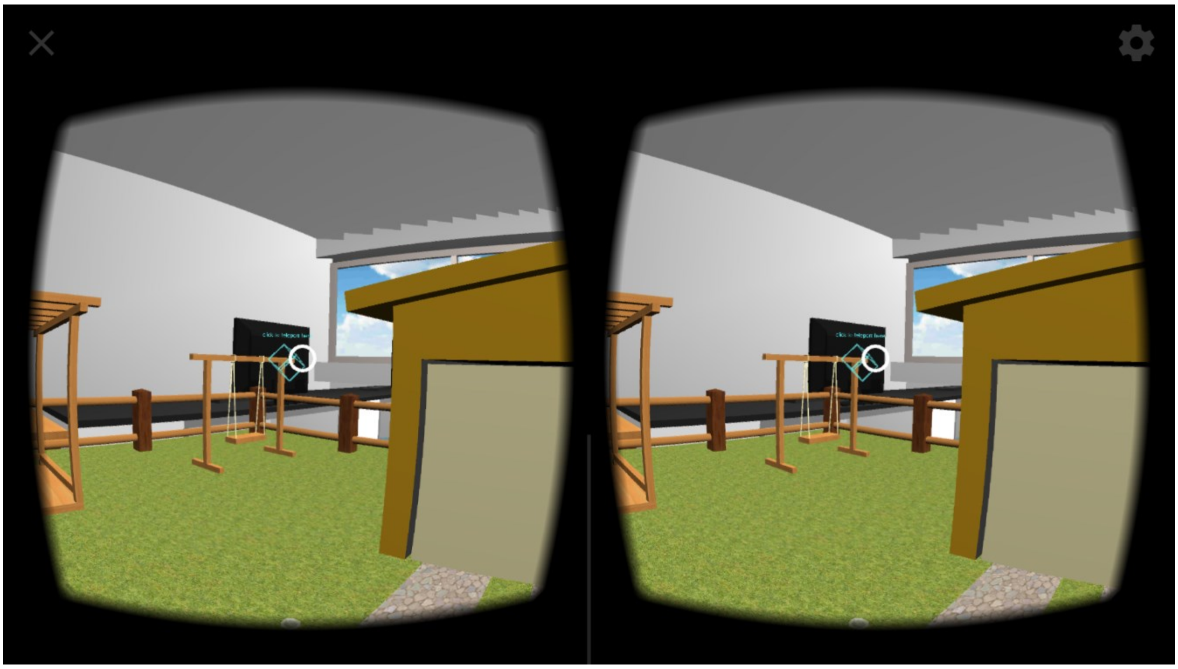


Figure 15: Default point of view after turning on the application

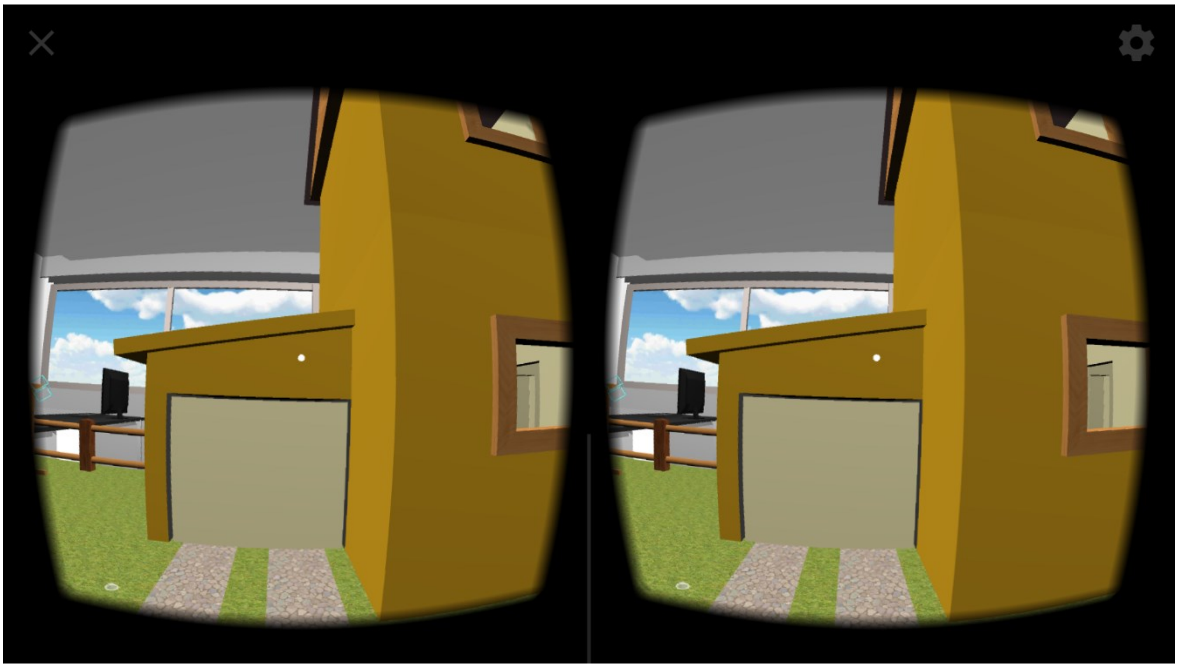


Figure 14: Slightly turning right from the default point of view

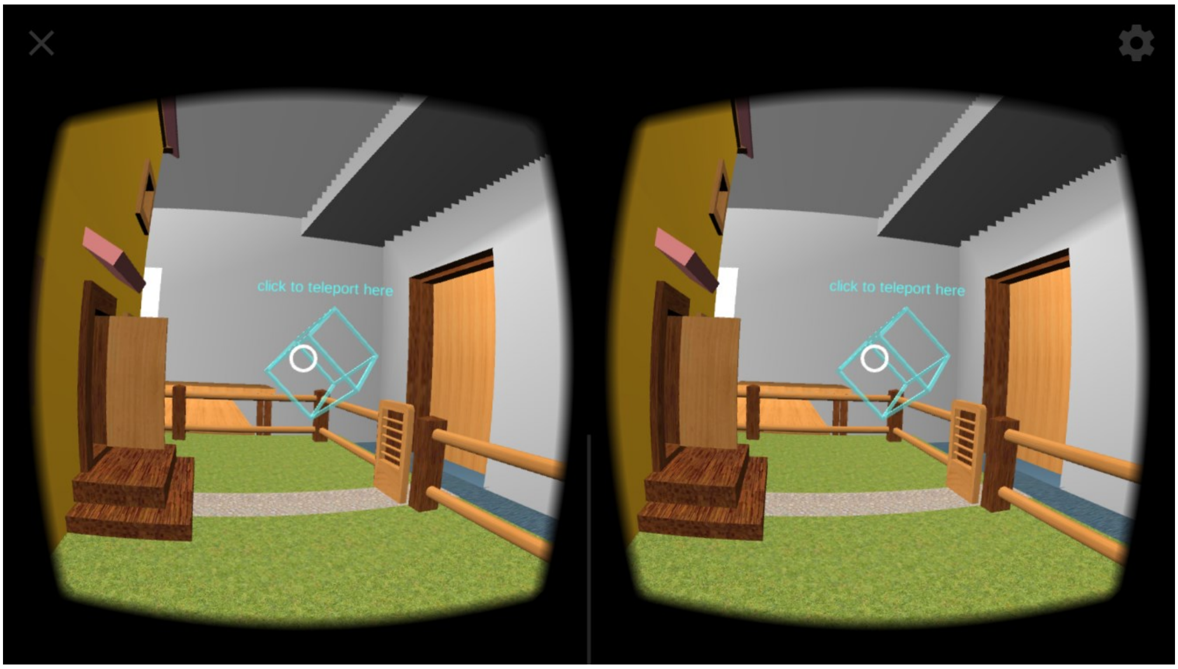


Figure 17: Looking at a clickable teleport point

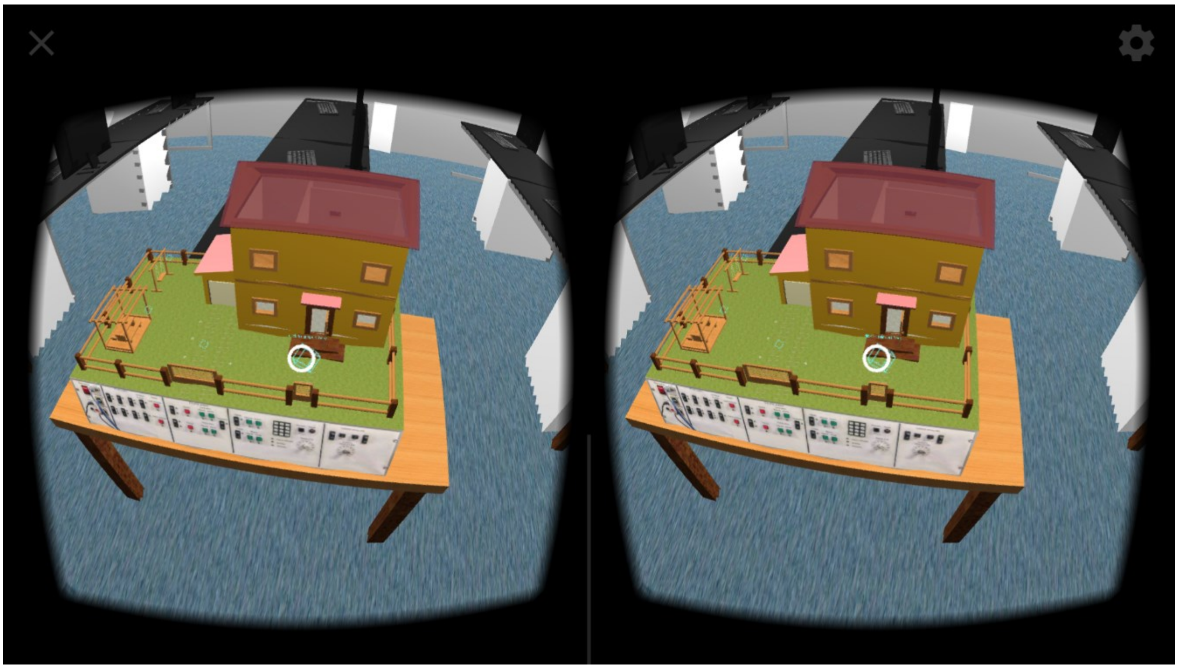


Figure 16: Looking at the same teleport point, but from a top point of view

6.6.2 Windows build

In order for the educational application to work on a Windows PC, a following CameraScript was used on the Main Camera which hid the mouse cursor and enabled mouse look instead of using head movements that are not available to the application on a Windows PC. [24] This script did not collide with the way the application processes input on an Android phone, so the same project was used to build the application for the Windows platform. Then the only thing that needed to be done was changing the Platform in File → Build Settings to PC, Mac & Linux Standalone. The Windows application was built by clicking the Build button.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CameraScript : MonoBehaviour {

      public float speedH = 2.0f;
      public float speedV = 2.0f;
      private float yaw = 0.0f;
      private float pitch = 0.0f;

      // Use this for initialization
      void Start () {
            Cursor.visible = false;
      }

      // Update is called once per frame
      void Update () {
            yaw += speedH * Input.GetAxis(“Mouse X”);
            pitch -= speedV * Input.GetAxis(“Mouse Y”);
            transform.eulerAngles = new Vector3(pitch, yaw, 0.0f);
      }
}
```

6.6.3 WebGL build

The educational application was also built for WebGL and then published online at http://senovsky.wz.cz/virtualab/index.html, allowing anyone, not only people with access to Android phones with Cardboard viewers or to a Windows PC, to experience VirtuaLab's wonders. Both horizontal and vertical sensitivity of the invisible mouse cursor was changed: Edit → Project Settings → Input → Mouse X and Mouse Y Sensitivity was increased to 0.25. After this change, it no longer happens that a teleport point would be unreachable because the mouse cursor reached a border of the screen.

7 TESTING ON HIGH-SCHOOL STUDENTS

24 students of selected high school were asked to try the educational application under supervision. They were instructed to visit all 14 teleport points and then fill out a questionnaire with these 10 statements via Google Forms:

1. The experience made me physically unwell
2. The sense of space awareness was authentic
3. It was easy to find out how to move around
4. The fact that I didn't see my body was disturbing
5. I found the way of aiming the cursor user-friendly
6. The screen door effect was distracting
7. The response to my head movements was fluent
8. I was able to experience the interior better than just by looking at a model
9. I can better imagine how a house automation system would work than just by looking at a model
10. A class on teaching automation systems programming would be more attractive if the models were explorable in virtual reality

There was a five-point scale of agreement/disagreement available with the given statements: 1 – Strongly agree, 2 – Agree, 3 – Neutral, 4 – Disagree, 5 – Strongly disagree. Here are the counts of the responses to the statements, their average values and success percentages:

Statement	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**#8**	**#9**	**#10**
Count of 1	0	13	13	0	15	0	13	15	10	16
Count of 2	8	8	8	2	7	3	8	6	8	4
Count of 3	2	2	1	3	1	8	1	2	5	2
Count of 4	5	1	2	9	1	7	1	1	1	1
Count of 5	9	0	0	10	0	6	1	0	0	1
Average	3.63	1.63	1.67	4.13	1.50	3.67	1.71	1.54	1.88	1.63
Success %	65.63	84.38	83.33	78.13	87.50	66.67	82.29	86.46	78.13	84.38

Table 4: Overview of the responses to the questionnaire

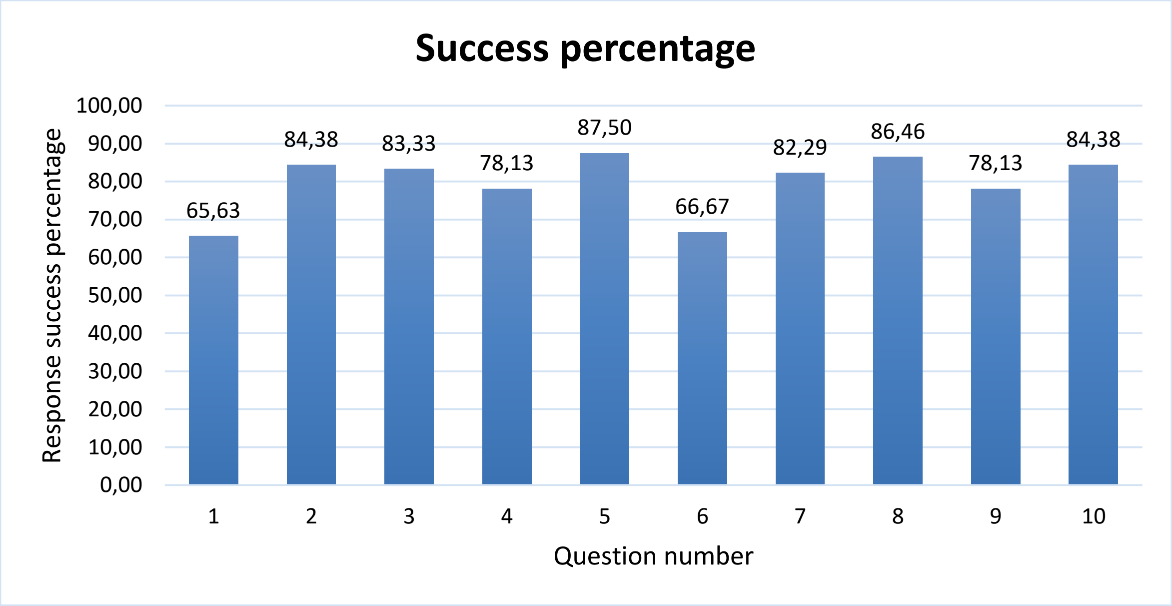


Figure 18: Success percentage overview of the responses

The success percentage was calculated using the following formula for statements where Strongly Agree was the most desirable response: $y = -25x + 125$ and for statements where Strongly Disagree was the most desirable response, this formula was used: $y = 25x - 25$. The variable x was always the average of responses for the given statement.

The success rate of the responses clearly indicate that the technology is still in its early stage because the two statements with the worst responses were about how physically unwell the students felt and how distracting was the screen door effect. However, the test subjects apparently felt quite immersed in the virtual environment and they found the application easy to use. Their belief that school lessons would benefit from the usage of this technology was also strong.

An average success percentage was then calculated from the individual success percentages of the statements. Its value is 80% which can be considered quite high, given the immaturity of the technology used and an intermediate level of complexity of the application.

8 METHODOLOGICAL SHEET FOR TEACHERS

Since the technology of virtual reality is a fresh addition to available educational tools, the teachers are going to need guidance on how to properly use it their lessons. It is also imperative to know how to mitigate the risk of students getting injured while using this technology. An example of a Methodological sheet for teachers follows:

Methodological sheet for teachers

Subject:

PLC programming

Lesson length:

45 minutes

Accessory needed:

A laboratory equipped with a model of a house connected to a PLC, Google Cardboard virtual reality headset, an Android smartphone with VirtuaLab installed

Lesson target:

Familiarize the students with the technology of virtual reality, warn them about its possible shortcomings and explain them how to use it during the lesson

Methodological instructions:

The students are going to see and will be able to use a model of a house connected to a PLC. However, since it is a model and it is not physically possible to go inside of it, the students can use a Google Cardboard virtual reality headset with an Android smartphone. The application VirtuaLab has to be installed on the smartphone, launched and then the smartphone has to be inserted into the headset. Before the students put it on their heads, they need to make sure that there are no objects in their immediate surroundings they could bump into. Sitting on a swivel chair is recommended – it allows 360° rotation while confining the students to a single spot (Google Cardboard does not track user’s movement around the room). It is also important to warn the students that they might feel unwell and dizzy in the virtual environment and instruct them to close their eyes and remove the headset from their head in case the unpleasant feeling persists. Those students can use an online version of the application instead: http://senovsky.wz.cz/virtualab/index.html. All of the students should be then told how to navigate through the virtual environment using the 14 teleport points and that they should envision how a house automation system would work.

CONCLUSION

During the work on this thesis, several important things were discovered and an overall conclusion about the pros and cons of implementation of virtual reality into high school education can now be drawn.

First of all, the lesson learned from literary research is that the history of virtual reality did not start in 2012 with Oculus's Kickstarter campaign. That was a moment when virtual reality finally became commercially viable after numerous, largely failed attempts to create some kind of commercial virtual reality device since the first half of the 20th century and today's technology takes a lot from those attempts. However, they laid out a very important groundwork and it can be only theorized where the technology is going to be in the future.

Regardless of the decades of research and development already done in the field of virtual reality, there is clearly still a huge room for improvement, especially in the category of low-cost virtual reality solutions. The experience could make less people feel sick, the price/performance ratio of the devices running the simulated environment should improve to allow more life-like experiences without draining budgets dry and creating content for the technology should not require developers to possess as much time and skills as it currently does.

When it comes to the educational application developed as a part of this thesis, it produced one surprising morale. The time required to develop the application itself was not as long as the time required to fill it with 3D content. Technology able to scan real environments and convert them to virtual ones is crucial if virtual reality is to become larger part of everyday life than it currently is. The team behind Unity and the authors of Google Cardboard Demo are to thank for allowing rapid development and steep learning curve stemming from very well prepared documentation, Unity tutorials and learning materials.

Making the 3D models in the virtual environment interactive would be a recommendable next step in evolution of the educational application. Albeit driving the time requirements even higher, interactivity would also increase students' involvement in using the application and teach them how individual parts of the shown models work.

One pleasant surprise happened when the educational application was installed on a Samsung Galaxy S4 mini. The resolution of the display and the computational power of the device are both one quarter of their counterparts found in Xiaomi Redmi Note 4, yet the experience was definitely not four times worse. The readability of text decreased, otherwise the sense of being immersed as only somehow weaker. This proves that some aspects of the technology available today are already at good level. What clearly hindered the experience was the reaction to head movements. This area needs significant improvements on mobile virtual reality in general, 60 Hz refresh rate of the screens is just not enough to make the response to head movements realistically fluent.

One technological aspect connected to creating more life-like experience is the availability of an efficient graphics API on both desktop and mobile devices called Vulkan. It allows higher level of detail than OpenGL ES on the same hardware and even though it is present and confirmed working on the Xiaomi Redmi Note 4, the educational application build by Unity crashed upon installation due to a segmentation fault when run on Vulkan instead of OpenGL ES. This kind of bugs will be hopefully resolved in the future, allowing users to enjoy more realistic imagery on cheaper hardware.

All things considered, especially the enthusiasm shown by the positive responses to the questionnaire statements regarding usefulness of virtual reality in lessons, there is a huge potential in implementing this technology into school education. However, a considerable amount of research and development still needs to be done in this area and the implementation process definitely should be careful, gradual and mindful of the obstacles that it has to overcome. After hearing several people complain of slight headache, stomachache or eyestrain from using the virtual reality headset for just a couple of minutes, and especially after seeing two test subjects hurtfully bump their knee into a table which they had clearly seen just one minute prior to the accident, I cannot stress enough how much emphasis has to be put on making sure that this promising technology brings more good than bad into classrooms.

BIBLIOGRAPHY

[1] MURRAY, Jeff W. *Building virtual reality with Unity and Steam VR*. Boca Raton, FL: CRC Press, 2017, 352 p. ISBN 978-1138033511.

[2] FAI UTB. *Vimeo* [online]. 2011 [cit. 2018-05-15]. Available from: https://vimeo.com/user8620872

[3] Tutorials. *UNITY Technologies* [online]. 2017 [cit. 2018-05-15]. Available from: https://unity3d.com/learn/tutorials/

[4] Tutorials. *BLENDER Foundation* [online]. 2017 [cit. 2018-05-15]. Available from: https://www.blender.org/support/tutorials/

[5] In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-05-15]. Available from: https://en.wikipedia.org/wiki

[6] Virtual (adj.). *Online Etymology Dictionary* [online]. 2018 [cit. 2018-05-15]. Available from: https://www.etymonline.com/word/virtual

[7] Oculus Rift: Step Into the Game. *Kickstarter* [online]. 2012 [cit. 2018-05-15]. Available from: https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game

[8] Unity Manual. *Unity | Documentation* [online]. 2018 [cit. 2018-05-15]. Available from: https://docs.unity3d.com/Manual/

[9] O'HARE, Tyler. Japan have created VRHS: Virtual reality high school. *Techly* [online]. 2016-07-07 [cit. 2018-05-15]. Available from: https://www.techly.com.au/2016/07/07/japan-created-virtual-reality-high-school/

[10] ROLPH, Amy. This high school wants to revolutionize learning with technology. *USA Today* [online]. 2017-04-03 [cit. 2018-05-15]. Available from: https://www.usatoday.com/story/sponsor-story/xq/2017/04/03/high-school-wants-revolutionize-learning-technology/99985812/

[11] LAWRIE, Graeme. How our school is using Virtual Reality to prepare pupils for a future dominated by technology. *The Telegraph* [online]. 2017-01-23 [cit. 2018-05-15]. Available from: https://www.telegraph.co.uk/education/2017/01/23/school-using-virtual-reality-prepare-pupils-future-dominated/

[12] HANSMAN, Heather. How Can Schools Use Virtual Reality? *Smithsonian.com* [online]. 2016-02-03 [cit. 2018-05-15]. Available from: https://www.smithsonianmag.com/innovation/how-can-schools-use-virtual-reality-180957974/

[13] Technical specifications. *PlayStation VR* [online]. [cit. 2018-05-15]. Available from: https://www.playstation.com/en-gb/explore/playstation-vr/tech-specs/

[14] MARTINDALE, Jon. Oculus Rift vs. HTC Vive. *Digital Trends* [online]. 2018-04-04 [cit. 2018-05-15]. Available from: https://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/

[15] Xiaomi Redmi Note 4. *GSMArena.com* [online]. 2017 [cit. 2018-05-15]. Available from: https://www.gsmarena.com/xiaomi_redmi_note_4-8531.php

[16] rRMB Menu. *Blender Wiki* [online]. 2016 [cit. 2018-05-15]. Available from: https://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/System/rRMB

[17] How do I flip a face? *Blender Artists* [online]. 2006 [cit. 2018-05-15]. Available from: https://blenderartists.org/t/how-do-i-flip-a-face/350980

[18] BHEEMA. Lcd_Monitor. *Blend Swap* [online]. 2013-06-21 [cit. 2018-05-15]. Available from: https://www.blendswap.com/blends/view/68813

[19] Quickstart for Google VR SDK for Unity with Android. *Google Developers* [online]. [cit. 2018-05-15]. Available from: https://developers.google.com/vr/develop/unity/get-started

[20] Adb install failure: INSTALL_CANCELED_BY_USER. *Stack Overflow* [online]. [cit. 2018-05-15]. Available from: https://stackoverflow.com/questions/37641670/adb-install-failure-install-canceled-by-user

[21] AASHISH314. Mi VR Play 2 QR Code. *MIUI* [online]. 2017-11-30 [cit. 2018-05-15]. Available from: http://en.miui.com/thread-1223485-1-1.html

[22] *Unity Asset Store* [online]. [cit. 2018-05-15]. Available from: https://assetstore.unity.com/

[23] Skybox Isn't showing in game view. *Unity Answers* [online]. 2016 [cit. 2018-05-15]. Available from: https://answers.unity.com/questions/1223256/skybox-isnt-showing-in-game-view.html

[24] LEARN EVERYTHING FAST. Rotate Camera with Mouse in Unity 3D. *YouTube* [online]. 2017-02-21 [cit. 2018-05-15]. Available from: https://www.youtube.com/watch?v=lYIRm4QEqro

LIST OF ABBREVIATIONS

HTC	High Tech Computer corporation
VR	Virtual Reality
AR	Augmented Reality
OS	Operating System
IMB	International Business Machines corporation
PC	Personal Computer
LCD	Liquid Crystal Display
US$	United States Dollars
LLC	Limited Liability Company
MSRP	Manufacturer’s Suggested Retail Price
OLED	Organic Light-Emitting Diode
HDMI	High-Definition Multimedia Interface
USB	Universal Serial Bus
AMD	Advanced Micro Devices, Inc.
RAM	Random Access Memory
DIY	Do It Yourself
SDK	Software Development Kit
API	Application Programming Interface
QR	Quick Response
2D, 3D	Two-Dimensional, Three-Dimensional
FullHD	Full High Definition (1920×1080 px)
D.C.	District of Columbia
UV map	The letters "U" and "V" denote the axes of a 2D texture
GPU	Graphics Processing Unit
CPU	Central Processing Unit
ES	Embedded Systems
PLC	Programmable Logic Controller
PPI	Pixels Per Inch
ROM	Read-Only Memory

LIST OF FIGURES

LIST OF TABLES

APPENDICES

CD with all Blender files, Unity project, assets, builds, source code and other material used.