The Preparation of Graphic Models for a Virtual Reality Application in Unity

Pavel Pokorný[1] and Michal Birošík[2]

[1] Department of Computer and Communication Systems
Tomas Bata University in Zlín, Faculty of Applied Informatics
Nad Stráněmi 4511, 760 05 Zlín, Czech Republic
pokorny@utb.cz
[2] Department of Informatics and Artificial Intelligence
Tomas Bata University in Zlín, Faculty of Applied Informatics
Nad Stráněmi 4511, 760 05 Zlín, Czech Republic
michal.birosik@gmail.com

Abstract. Although Virtual Reality has existed for decades, it is currently experiencing a boom. This is mainly due to the new powerful hardware devices available to many people. Accompanying this greater availability, the fields of use are also currently expanding – medicine, military, business, engineering, education, entertainment, etc. The main purpose of Virtual Reality is to provide the user with a real-looking simulated environment, including interaction. Specialised developer software tools are used in order to create Virtual Reality applications that exploit suitable 3D graphics models. This paper deals with the creation and preparation of such graphical models - including textures, and the problems of the conversion process, to a developer application. Blender is used to create models and UV maps. The export of these models and UV maps from Blender is set up for easy implementation in the Substance Painter software, where the textures are drawn - and later, into the Unity developer engine for Virtual Reality application.

Keywords: Virtual Reality, Modelling, Visualisation

1 Introduction

The term Virtual Reality is sometimes interpreted in different ways. Resource [1] defines it as “Immersive Computer-simulated Reality that creates a physical environment that does not exist”. Resource [2] mentions that “Virtual Reality is the computer-generated simulation of a 3D environment, which seems very real to the person experiencing it, using special electronic equipment”.

This term can also be defined based on its purpose. The main objective is to achieve a strong sense of being present in the virtual environment. [3] It enables cognitive activities in the artificial, digitally-created world, which can be a simulation of many aspects of the real world. [4]

The key elements of Virtual Reality are its participants, creators, virtual worlds, immersion and interactivity. [3] The participants are probably the most important element because the whole Virtual Reality experience happens in their mind. Each participant gains individual experiences that are given to creators, who design and implement the virtual reality application. A virtual world forms the content and exist solely in the mind of its originator - or is manifested in such a way that can be shared with others. Immersion means a perception of being physically present in a non-physical world. The user needs virtual interaction with the virtual environment in order to be able to feel fully immersed.

Although Virtual Reality was previously generally associated only with the gaming industry, at the present it is amazingly applied in a wide variety of fields of human activities. [5] This includes robotics and engineering, (robots can be applied as an intuitive human user interface with virtual reality); training and education, (these can develop or increase skills without the real-world consequences of failing) [6]; architecture design, (to simulate buildings or whole cities); entertainment and leisure, (video games, cinema, music, virtual travel); medicine, (it can be used for some therapies) [7], and many more.

Because the virtual reality world has to be very realistic-looking, this environment must be created carefully, thoroughly, consistently and greatly detailed. This includes high-poly 3D graphic models and high-resolution more-layered textures. All these entities are then imported into specialised developer software, where the final Virtual Reality application is created.

This paper describes these actions. Chapter 2 presents the Virtual Reality technology in more detail. Chapter 3 contains more details about the software used. Chapter 4 describes the creation of 3D models and the preparation of “uv” maps for textures.The final chapter (Chap. 5), contains information about how to export uv maps and fast and easy texture creation in order to prepare them for use in a Virtual Reality application

2 Virtual Reality Technology

Virtual Reality has one main goal - to convince the user that they are located in a different environment than they really are. It mainly works on the principle of deceiving the human brain - through vision, and those parts of the brain that perceive movement. It is necessary to combine a number of technologies in order to create an illusion,. These technologies include stereoscopic displays, motion-tracking hardware, suitable input devices, and computing platforms. Without these technologies, it is not possible to achieve a fully immersive Virtual Reality Experience. [9]

Stereoscopic displays (also known as head-mounted displays or headsets – an example is shown in Figure 1), use a combination of special lenses, multiple images and realistic optical distortion to produce a stereo image that a human’s eyes interpret as having three-dimensional depth. Motion-tracking hardware includes accelerometers, gyroscopes and other components that can sense one’s body movements and head turns and this information is then provided to the application that can update the view into the 3D scene.

Virtual Reality also uses these new types of input devices - including game controllers and hand and body sensors that can recognise basic motions and gestures. Computing platforms include computer hardware (desktop or mobile), operating systems - and various software packages that understand the interfaces of the above-mentioned devices and allow one to develop and run a Virtual Reality application.

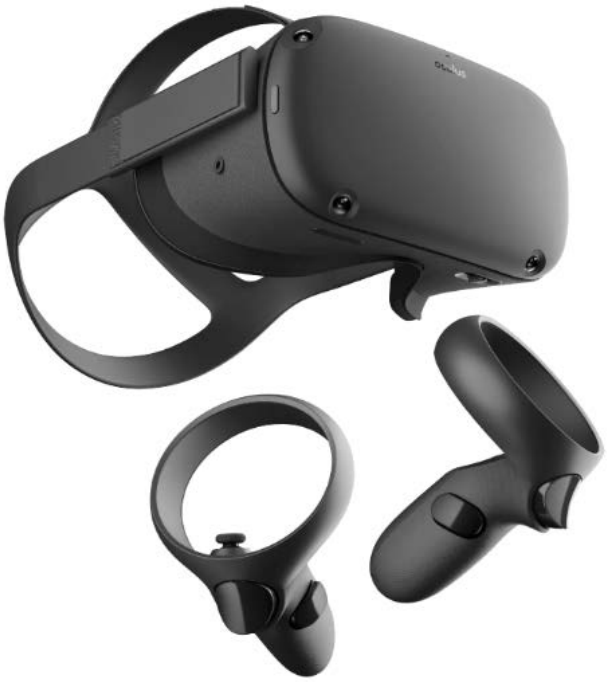


Fig. 1. An example of a Virtual Reality headset with input devices. [8]

Technological advances made it possible to expand and divide Virtual Reality into four main categories - according to the use of hardware devices and components: desktop, mobile, console and stand-alone. Desktop Virtual Reality refers to computer programmes - (on a powerful personal computer or laptop), that simulate a real or imaginary world and compute the output images. These images are then sent to the headset, which is connected to the computer. Nowadays, Mobile Virtual Reality is one of the most popular and widespread forms of (not only) entertainment, since it is highly accessible for anyone who has a smartphone. With this, one can use Virtual Reality anytime and anywhere. Console Virtual Reality is performed by means of a game console, which computes output images - (the Sony Playstation VR for example). Stand-alone Virtual Reality does not require any other computational hardware devices. Its headset has all such devices built-in (CPU, memory, hard-disk, GPU, displays, and more). In addition, these headsets have no connective cables, so users do not have limit themselves to a restricted space. Figure 1 shows an example of an Oculus Quest stand-alone Virtual Reality headset [8].

Virtual Reality Modelling language was developed in order to create Virtual Reality applications. Later, when the Web3D consortium was founded, WebVR technology was established - which provided support for Virtual Reality devices and allowed developers to create web applications. This technology has recently been replaced by WebXR - which represents a group of standards that support the rendering of 3D scenes for most modern headsets. [10] Currently, Virtual Reality is also massively supported in modern software engines – like Unreal [11] or Unity [12].

3 Software

We need to use some software tools to prepare models for Virtual Reality applications. This chapter describes them briefly.

The Blender software tool was used for 3D modelling and uv map preparation [13]. There are a lot of 3D software packages around. Blender was selected because we have a lot of experience with it and it offers a lot of benefits. It is a free and open source 3D creation suite. It supports the entirety of the 3D pipeline - modeling, rigging, animation, simulation, rendering, compositing and motion-tracking, video editing and a 2D animation pipeline. In all of above-mentioned fields, Blender offers a lot of sophisticated tools that can be further extended via plug-in modules and, in addition, a customisable interface (Figure 2), significant upgrades, workflow speed and very quick launch speed. [14]

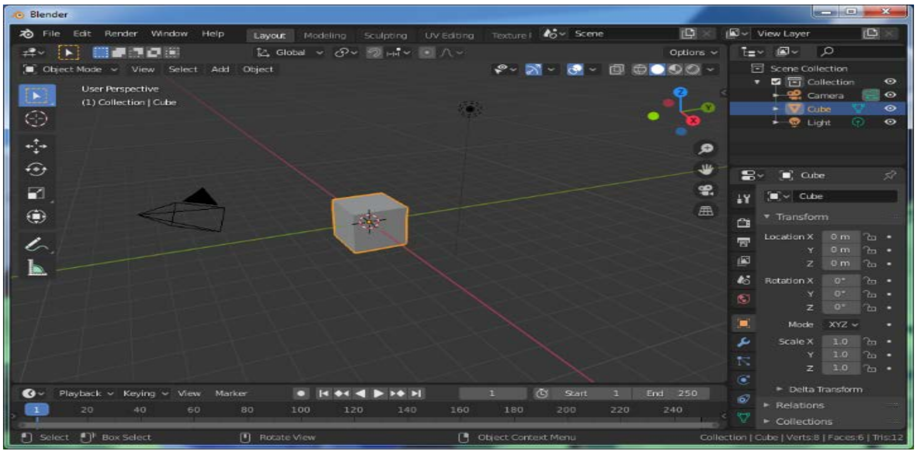


Fig. 2. The Blender 2.80 Interface

Textures were created in Substance Painter [15]. This tool encapsulates all of the necessary commands for texturing 3D assets. It performs real-time painting in 3D space using smart brushes, projection tools - and even particles, supports smart materials and smart masks that are adapted to user’s models, extracting and baking any material from any 3D object with a GPU accelerated baker, fast renderer, imports assets from many commonly-used graphic formats and exports them to any game engine. [16]

Although the development a Virtual Reality application is not the main subject of this paper, some knowledge of the development environment for this application is needed in order to ensure that 3D models are best prepared for easy import. The Unity engine [12] was selected for Virtual Reality development. Unity provides an excellent entry point into game and 3D application development, balancing features and functionality with price. The free version of Unity allows people to experiment, learn, develop, and sell games before committing any of their hard-earned money. Unity's very affordable, feature-packed, Pro version is royalty free, allowing people to make and sell games and other applications with the very low overheads essential to the casual games market. With its huge user community, Unity removes the elitist separation between programmers, artists, and game designers that is typical of high-priced game engines. It makes it possible for anyone to take the first step in bringing their ideas to life. [17]

4 3D Model Preparation

We tried to follow the basic rules for common 3D model creation in the modelling phase in Blender - considering the type and use of these models in third generation applications. This chapter describes these rules, the modelling process and uv map creation methods.

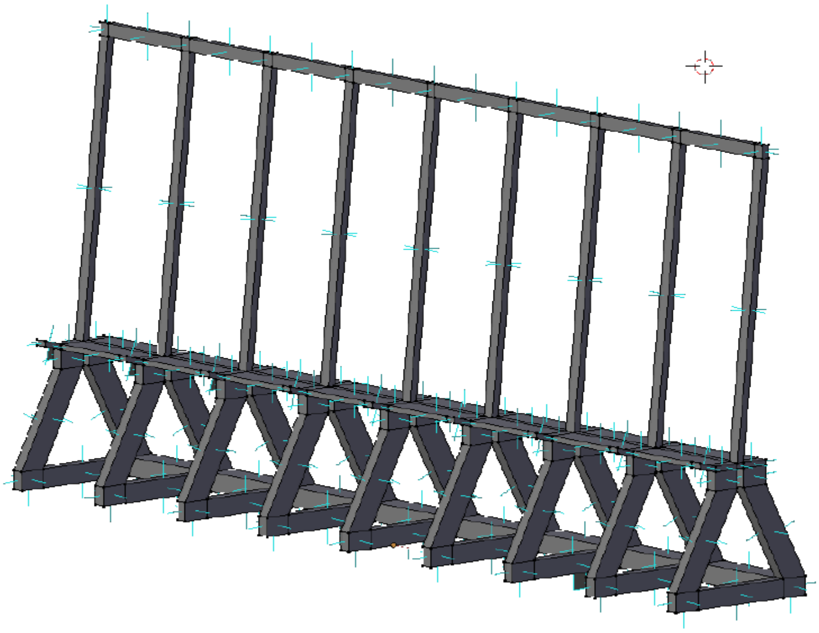


Fig. 3. An example of a 3D model with highlighted normal vectors on faces

4.1 Modelling

All models were created using polygonal representation. This means a collection of vertices, edges and faces that define the geometric shape of an object. The face topology can be represented by triangles (three-sided polygons), quads (four-sided polygons), or N-gons (polygons having more than four sides). The choice depends on the purpose of the prepared 3D models – if the model is going to be used for animation or deformation (then, then the preference is to use quads), or if it is to be a static model (then, it is possible to use triangles). We avoid the use of common N-gons, since they can cause technical complications in the future; for example, when exporting from one software package to another, or when they are subdivided. Since we expect to use models and their parts for interaction, animation and transformation, the preference is to use quad topology.

For model creation, we used the so-called box modelling technique. Box modelling is a technique, where primitive shapes are used at the beginning (Circles, Planes, Cubes, Tubes, Spheres, etc.), these objects are then shaped to the final model using modelling tools.

We used the default Blender user interface setting in the modelling phase. The 3D scene appearance can be set to represent perspective or an orthographic look. The Orthographic look was set, because the same metrics are shown for all dimensions and this property is very suitable for modelling the final shapes.

Blender offers many editing, transformation and modelling tools and these tools can be used in Object or Edit Mode. These names suggest their uses – the Object Mode is designed for working with whole object models - whereas the Object Mode is intended for work with any part of a selected object. In the Edit Mode, one can therefore work directly with any vertices, edges or faces, edit or transform them and/or, use other modelling tools. These tools often generate new vertices, edges and faces, so they allow one to make more detailed and better shaped models. Extrude, Subdivide, Bevel, Loop Cut, Knife or Insert Face are examples of these tools [14]. An example of the modelling phase is shown in Figure 3.

In case we needed a rounded shape, Blender does this by performing the Sculpt Mode. This mode is used for shape forming, using different brushes that do not work directly with vertices, edges and faces like other tools. The Brushes manipulate an object´s whole geometry in the brush´s region of influence, and shapes it dependent on the Brush type and its set parameters.

4.2 UV maps

When the shapes of the mesh models have been created, the UV mapping phase begins. The main goal in this phase is to make UV maps for all of them and their entire faces. The UV mapping process defines how to display any 2D image files on the faces of 3D models. This process is usually reversed – 3D models are processed into component parts and shown in 2D images. These images are called UV maps and displays, where the texture must be painted in order to correctly be displayed on the models´ surfaces.

There are more ways regarding ways to make UV maps in Blender [14]. We preferred to use edge selection where the faces must be ripped in order to obtain a 2D image. These edges are determined, selected and then marked as so-called “seams”, (this operation is performed using the Mark Seam command in Blender). After the edges are marked, the Unwrap command will create the uv map. Another way, (without seams marking) was used to separate more complex models into several independent objects - (this is performed in Blender using the Separate command), and then by running the Smart UV Project command, which ensures the generation of seams and UV maps automatically. The Smart UV Project cannot be successfully used for complex-shaped models since it often depicts wrong edges as a seam- and the uv map then has an unnecessarily complicated appearance.

Because some shapes are often repeated, it is possible to save work and time through UV map creation. Only one UV map was made in these cases and the UV map was then applied - (duplicated), to all other identical shapes or models.

The important thing regarding UV maps is to use image space effectively in order to obtain the highest possible texture quality, the uv map should cover the image area to the greatest extent possible. Figure 4 depicts an example of a created uv map.

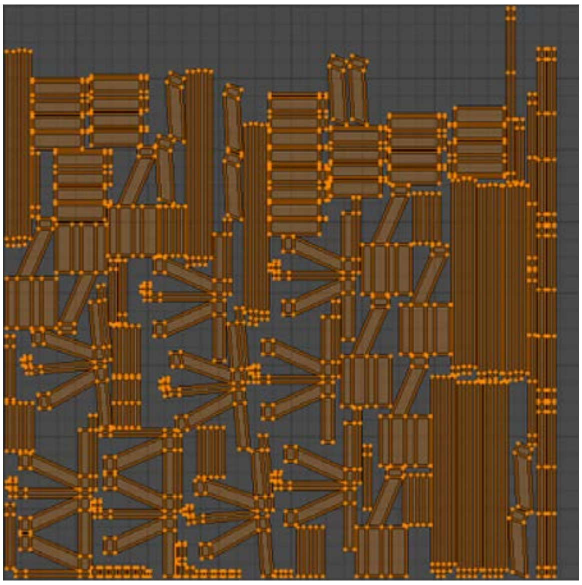


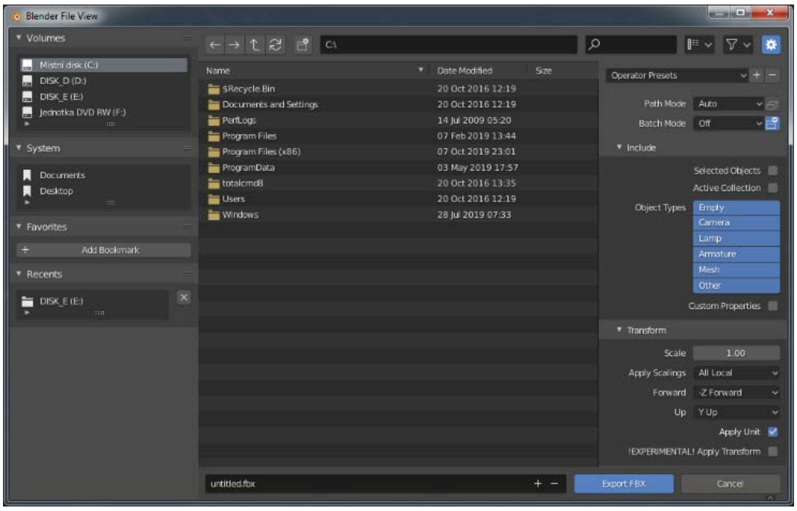
Fig. 4. An example uv map for the model shown in Figure 3.

5 Exporting and Texturing

When the modeling and UV map were complete, all created assets were exported from Blender into the Substance Painter software package. The models were exported by means of the .fbx graphic format and uv maps were exported in .tiff graphic format with a 1024 x 1024 pixel resolution.

FBX was designed as a 3D asset exchange format that facilitates higher-fidelity data exchange between many 3D graphic software packages - including Blender, Substance Painter and Unity and other platforms (including mobile devices). [18]

When exporting 3D models from Blender to the .fbx file format, one obtains the dialogue window shown in Figure 5. In order to obtain correct conversion, one should know the differences of 3D scene interpretation between the Blender and Substance Painter (Unity) suites. [19] The first thing is to export only pure models (i.e. no cameras, lamps, armatures and others object types). The second thing is that Blender and Substance Painter (Unity) use different coordinate systems (the Z and Y axis are flipped). So in order to transform coordinate work in these applications, the Apply Transform button must be enabled.



C:\

Name	Date Modified	Size
$Recycle.Bin	20 Oct 2016 12:19	
Documents and Settings	20 Oct 2016 12:19	
PerfLogs	14 Jul 2009 05:20	
Program Files	07 Feb 2019 13:44	
Program Files (x86)	07 Oct 2019 23:01	
ProgramData	03 May 2019 17:57	
totalcmd8	20 Oct 2016 13:35	
Users	20 Oct 2016 12:19	
Windows	28 Jul 2019 07:33	

untitled.fbx

Path Mode: Auto
Batch Mode: Off

Include
Selected Objects □
Active Collection □
Object Types: Empty, Camera, Lamp, Armature, Mesh, Other
Custom Properties □

Transform
Scale: 1.00
Apply Scalings: All Local
Forward: -Z Forward
Up: Y Up
Apply Unit ☑
!EXPERIMENTAL! Apply Transform □

Fig. 5. The FBX export dialogue in Blender

5.1 Texturing

All textures were drawn in the Substance Painter application. We used the Relief Mapping technique because it performs accurate and effective renderings of the surface detail of 3D objects. Relief Mapping included Projection Painting (the combination of multiple projectors together on a surface for creating details); Colour Map Creation (to define the base diffuse colour of materials) ;and Bump Mapping (a technique for simulating bumps and wrinkles on the surface of objects without having geometric details). An example of a mapped Bump Map on the surface of an object is shown in Figure 6.

They were separated into multiple layers depending on texture effects, (e.g. colours, bumps, transparency, gloss, etc.). And then the final appearance of the surface was specified by a combination of multiple layers of materials or textures using the Mask

Map tool. This allows one to specify which texture is drawn on which part of the surface using black - (the invisible part of a texture), or white colour - (the visible part of a texture). Using a mask map is a very flexible and non-destructive way to work with textures because it allows one to change the surfaces´ appearance without actually modifying the maps themselves. [14]

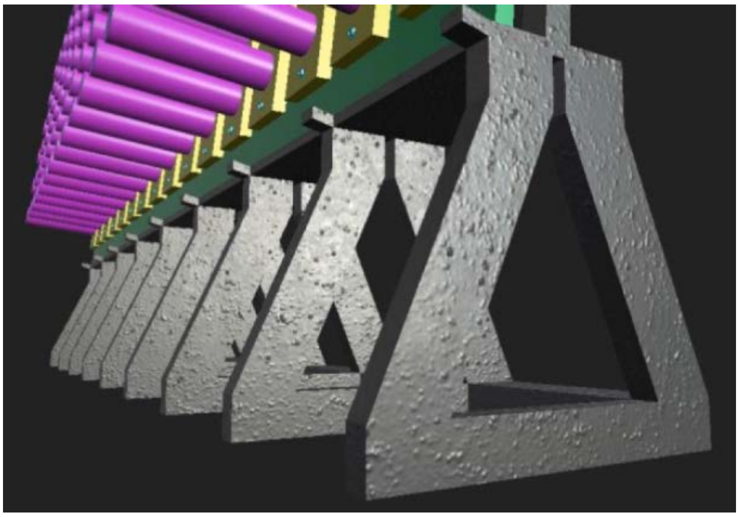


Fig. 6. A Bump Texture mapped on an object´s surface in Substance Painter

After creating all of the textures for the 3D model, only one step remains – the export of all of these textures into the Unity engine. First of all, the texture resolution was changed. The UV maps had 1024 x 1024 pixel resolution and this was increased to 4096 x 4096 pixels in order to obtain very high-quality 3D objects later, in the Virtual Reality application. Other texture optimisation for Unity is automatic – the Export Textures command displays a window with export configuration settings. One possible choice is Unity 5. By selecting this, all textures were automatic optimised for this software. The output texture format was set to .tiff.

6 Conclusion

Virtual Reality is currently a “hot” topic. This technology is closely linked to specific executive hardware and offers many advantages - like interaction with realistic-looking scenarios where is possible to try impossible or difficult-to-implement situations and environments in real life, in communicating with other people in the selected area, or convenience for one’s own work.

Since Virtual Reality requires high-quality 3D models with textures, this paper sets out to describe how to prepare them. This process began with models - and their UV maps creation in specialised software packages, (this paper describes Blender). After making the model and UV maps, the texturing phase started - this was done in the Substance Painter software. The last step was to export all of the completed models and textures to a developer software package where the Virtual Reality application is programmed – we assume the use of the Unity engine.

In order to practically exploit the acquired experience, our next task will be to use them for Virtual Reality application creation. The aim is to develop an application that can be practically used for new employee training process in industrial companies.

References

1. Mealy, P.: Virtual & Augmented Reality for dummies. 1st edn. John Wiley & Sons, Inc. (2018). ISBN 978-1-119-48134-8.
2. Linowes J.: Unity Virtual Reality Projects. 2nd edn. Packt Publishing, Birmingham (2018). ISBN 978-1-78847-880-9.
3. Sherman, W.R., Craig, A.B.: Understanding Virtual Reality: Interface, application and design. Morgan Kaufmann, Cambridge (2019). ISBN 978-0-12-818399-1.
4. Wexelblat, A.: Virtual Reality Applications and Explorations. Academic Press (2014). ISBN 978-1-48320-712-9.
5. Garbade, M.: 10 Amazing Uses of Virtual Reality, https://readwrite.com/2018/11/08/10-amazing-uses-of-virtual-reality/, last accessed 2020/01/18.
6. Peters, C., Postlethwaite, D., Wallace, M. W.: Systems and methods providing enhanced education and training in a virtual reality environment. U.S. Patent No 9, 280, 913 (2016).
7. Moro, C., Štromberga, Z., Raikos, A., Stirling, A.: The effectiveness of virtual and augmented reality in health sciences and medical anatomy. Anatomical sciences education, vol.10, issue 6, pp. 549-559 (2017)
8. Oculus Quest Homepage, https://www.oculus.com/quest/, last accessed 2020/01/18.
9. Parisi, T.: Learning Virtual Reality. 1st edn. O'Reilly Media, Inc. (2015). ISBN 978-1-491-92283-5.
10. WebXR Device API, https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API, last accessed 2020/01/18.
11. Unreal Engine, https://www.unrealengine.com/en-US/, last accessed 2020/01/18.
12. Unity Real-Time Development Platform, https://unity.com/, last accessed 2020/01/18.
13. Blender.org – Home of the Blender project, https://www.blender.org/, last accessed 2020/01/18.
14. Guevarra, E.T.M.: Modeling and Animation Using Blender:Blender 2.80:The Rise of Eevee. 1st edn. Apress (2020). ISBN 978-1-4842-5339-7.
15. Substance Painter, https://www.substance3d.com/products/substance-painter/, last accessed 2020/01/18.
16. Toudou: Substance Painter for Blender user. 1st edn. Amazon.com Services LLC (2017).
17. Blackman, S.: Unity for Absolute Beginners. 1st edn. Apress. (2014). ISBN 978-1-4302-6779-9.
18. FBX – Adaptable File Formats for 3D Animation Software, https://www.autodesk.com/products/fbx/overview, last accessed 2020/01/18.
19. How to import Blender models into Unity, https://gamedevacademy.org/how-to-import-blender-models-into-unity-your-one-stop-guide/, last accessed 2020/01/18.