Design and optimisation methods for interactive mobile VR visualisation

S Skulimowski, M Badurowicz, M Barszcz and J Montusiewicz
Lublin University of Technology, Department of Computer Science, Nadbystrzycka 38, Lublin, Poland
s.skulimowski@pollub.pl

Abstract. The article describes the process of creating a VR mobile application for Google VR platform. The application presents a virtual museum exhibition as part of cross-cultural competence and user interface usability research. The process of creating the application requires taking into account the hardware and software limitations and the specificity of mobile devices. In addition, a higher level of optimisation helps build positive user experience, which is essential in VR applications. The process includes, among others: planning activities, preparing assets, implementing VR vision and interaction modules. The paper includes descriptions of individual steps, problems and other non-standard situations that occurred during the creation and testing of the VR application. In addition the results of the initial performance tests and the methods used to optimise and maintain the stability of the application are presented. By taking into account all of the mentioned factors and the fact that many solutions are yet not well described, the description of the process is expected to provide a point of reference and help for future developers of mobile VR applications.

1. Introduction
In recent years, VR (Virtual Reality) is developing dynamically, becoming a very promising technology. Computer games and art are its primary scope of application [1]. However, VR is equally important in medicine [2], education (training of pilots, doctors, surgeons [2], and other specialist [3,4,5]), engineering (design, simulations, repairs) or business communication (where VR combined with telepresence can replace expensive employee journeys in many cases) [6]. More and more often VR technology is used to prepare virtual tours of historic buildings [7].

Companies producing VR equipment (Steam and HTC, Oculus or Sony) are competing to outdo the parameters of the rival products. These are devices that offer rising opportunities each year, however, they are still very expensive. That is why new low-cost solutions are being sought. The market offers relatively cheaper mobile solutions compared to available professional ones (i.e. VR glasses - Oculus Rift, HTC Vive, as well as PlayStation VR). Their equivalent on the mobile market is, among others, Samsung Gear VR and Google Cardboard. Instead of the built-in screen, these solutions use only the user's phone, which takes over all the work and allows the image to be processed and displayed [5]. These solutions are, therefore, much more accessible financially than their prices established high-end competitors. In addition, the entire system is completely wireless and can be used anywhere. No wiring allows free head rotation. These solutions are the most forward-looking, nevertheless, today they offer far worse parameters.
According to literature analysis, many VR applications are currently being developed for mobile platforms [8,9]. Some of them serve to broaden the experience of visiting the museum [10,11], while others support the cross-culture competency education process [12,13,14]. These are areas similar to the one intended to explore in the present paper. In no other case are methods of proper development and optimisation of application operation described. This confirms the authors’ belief that the prepared article will be useful for future developers.

In the process of creating proprietary mobile applications in VR technology, the Unity environment is often used. Although it is usually used to create games, it may be often found in scientific experiments, such as: testing the behaviour of a homogeneous robotic hive in a space with defined obstacles [15], generating digital landscapes built of trees with variable structures for VR [16], or simulation of nuclear decommissioning devices in VR [17].

The purpose of this research is:
- to develop an optimised application in the context of operational stability (fluency) and minimisation of resource consumption (energy and operational memory) for mobile devices,
- to organise knowledge in the field of procedures for designing and producing interactive VR presentations for selected mobile devices.

The authors have developed the application and have optimised it to become a low-cost mobile VR platform, due to the need to reduce the costs of using VR, to make it possible to use low-cost VR goggles [18], instead of stationary VR platforms such as Oculus Rift or HTC Vive.

2. The subject of study
The subject of the research is the VR application created within the framework of the work [18], along with the equipment of Mirza Ulugbek's madrasah located in Samarkand, Uzbekistan. The room model and its equipment were developed through the usage of reverse engineering technology, i.e. 3D scanning and classical modelling based on photographic documentation. Several models have been developed: among others, a hexagonal table typical for Uzbek culture, a book stand - lauch and a book placed on the stand. In addition, textures from photographs were used, representing objects such as turban and chapan - an Uzbek coat, carpet or kilims. The developed application was made as part of a project implemented by the Institute of Computer Science of the Lublin University of Technology “3D Digital Silk Road” [19]. One of the objectives of the application was to present museum objects without the necessity of going on an actual trip to a real museum, and as a way to improve the attractiveness of museums.

In addition, the application is used for:
- testing user interfaces in virtual space,
- educating students of the Lublin University of Technology in the field of cross-culture competencies [20].

A stereoscopic view from the user's perspective of the presented application is presented in Figure 1.

The application allows the user to interact with objects through the VR interface. After interacting with the exhibit, its description is presented. Depending on the situation for which the digital room will be used, its content is supplemented with additional objects. For this purpose, other objects belonging to different cultural circles have been made, such as the Roman Catholic cross, a Torah in the form of a scroll or the Star of David. These objects were entirely made using classic 3D modelling in the 3ds Max programme and incorporated in the application. In addition, a model of a virtual sculpture belonging to the Gallery of the Art of Socialist Realism, which was obtained using the 3D scanning technique using the Artec Eva scanner, was placed in the student's room.
From the dominant mobile platforms, i.e. Android and iOS, the Android platform was selected for the purposes of the presented application as more common and available. Among mobile platforms in Uzbekistan, up to 87% of the market belongs to Android [21], whose additional advantage is the availability of free tools for programmers.

The choice of mobile technology as a base for VR applications is, however, associated with the lower computing power of mobile devices [22]. For the best results of VR usage and to prevent motion sickness occurrence it is necessary to achieve the proper application frame rate [16] - the minimum value is 24 frames per second. This caused initial problems with application performance that had to be resolved through proper application optimisation.

3. Application development process

The application development process included the use of many Information Technology tools (IT tools) - Figure 2. These were tools in the field of visualisation design and testing (Figure 2 L1), 3D modelling (Figure 2 L2), and communication with external devices (Figure 2 L3).

The Unity engine was used to integrate the interaction mechanisms and the virtual environment. The choice of the environment was justified by the possibility of porting the created project to many distribution platforms, which fits in with the mission of increasing accessibility. It soon turned out, however, that although the ability to port project means that it is possible, it by no means makes any implication of its simplicity. The steps necessary to complete the VR application in the Unity environment and the problems that arose during the work on the project are described in Sections 3.1 and 3.2.

3.1. Programming environment configuration

In order to assume control over and perform stereoscopic projection on Android mobile devices, it was necessary to use the Google VR SDK - GVR SDK (E1) Unity extension. This required additional configuration determining the exact version of the Android API that is used in the project and launching support for VR Cardboard and Daydream (W0) technologies.

Further problems with the Unity and GVR SDK environments were:

- the need to separately install the Android Debug Bridge (ADB) tool (E2),
- the need to use the ADB tool only in a specific version compatible with Unity (E3),
- the inability to use two versions of the ADB tool in parallel on one development machine (E3),

Figure 1. Stereoscopic view of the scene from the perspective of the user (the first-person perspective).
bullet failure to detect Android SDK and Android Tools on the Unity side (E6, E7),
bull an incorrect version of the installed JDK environment, i.e. JDK9 (E5) – unsupported by the Unity community, which resulted in an enigmatic message “Resource compilation failed! Failed to recompile Android resource files” (E4).

Figure 2. Diagram of dependencies between tools used in the project, with division into layers (Ls) and considering the areas where errors (Es) and warnings (Ws) occurred.

To sum up, every further issue related to the application building itself revealed a defect in programming, resulting in a loss of integrity with tools operating at ever-lower levels of abstraction (E1, E2, E4, E6, E7). The information contained in the official Unity documentation did not take into account the scenario that was described above, and most of the suggestions regarding potential solutions came from online forums and blogs of people who faced similar obstacles in the past. For example, the lack of detection of Android SDK and Android Tools has been solved by downloading custom tools for Android and a separate manager for Android SDK components (E6, E7).

3.2. Mobile VR application development

After dealing with the problems described above, the authors proceeded to the proper development of the application. The process included:

bullet creating C# scripts describing all planned types of moving in virtual space and interacting with selected objects on the stage,
bull creating shaders, describing the way of displaying objects in key moments of the projection, such as: the moment of focusing the participant's attention on the object, the initiation of interaction, the interaction effect, the termination of interaction and ending focus on the object,
bull the arrangement of developed 3D models on the stage and assigning methods and interaction effects to them.
Due to the specificity of the Unity environment and the application, it was impossible to introduce automatic tests. This significantly extended the application development time – searching for potential errors and verifying the operation of scripts during the time of further modifications.

Based on the created prototype, manual tests were carried out on a group of IT students of the Lublin University of Technology in order to identify the initial, repeated errors. The prototype contained only the basic mechanisms of interaction and imported models.

Most of the participants had clear problems with concentration or balance. This was due to the low frame rate. That is why work on the optimisation of applications was required to be carried out later, to improve the performance by increasing the number of frames per second of the displayed VR scene.

4. Application optimisation procedure

4.1. Tools for testing application performance and evaluation criteria

The basic element that assisted programmers in optimising the presented application were the tools for profiling the performance of the Unity platform (Figure 3).

Performance profiling tools present primarily the time that the application allocates to the execution of individual stages of its operation, such as the time of rendering a single frame, the duration of scripts (Figure 3 C), the order of rendering of the elements of the scene in each frame (Figure 3 D) and similar.

![Figure 3. Unity environment during local testing of application operation: A – scene editor with preview of the user avatar position in space, B – FPP view simulation with a list of basic projection statistics, C – profiler window with time-varying computer resource allocations, D – a debugger frame window with a view of the order of rendering objects on stage in each frame.](image)

The use of profiling tools has shown that the basic operation that takes up the application's operating time is the rendering of the scene itself, not the interaction script. Hence, there was no need to modify the scripts' code; however, it was necessary to focus on the stage itself. That is why the authors used the tool provided by Unity, Stats (Figure 3 B), which shows the number of triangles and vertices in the current view of the scene, the number of operations on the graphics card – dynamic and cached batches and occupancy of VRAM.
This mechanism proved to be effective in searching for 3D objects that had too complex a grid. According to preliminary experiments, it emerged that the stage must contain less than 100,000 triangles and this value was set as a base value for subsequent stages of the optimisation process.

In addition, a script developed by the authors presenting the number of presented frames per second displayed in the application based on the Time.unscaledDeltaTime property of the Unity platform, responsible for determining the time elapsed between successive frames generated in the application was used to assess the optimisation process. Displaying the frame rate indicator (FPS) was placed outside the user's field of view and served only for the development purposes. However, this mechanism has been abandoned in favour of mechanisms provided to programmers directly by the Google VR platform presenting overlay on the application displaying the current frame rate along with the graph of history, which is presented in Figure 4. This allows programmers to evaluate the decrease in the number of frames generated depending on the user's movement in the virtual world, which allows spots inefficient system components.

![Figure 4. Application statistics screen provided by the Google VR platform.](image)

An important element presented by the tools of the Google VR platform is also the temperature remaining until reaching the threshold value when lowering the processor clock speed of the mobile device occurs (thermal throttling), which can further affect the decrease in the number of frames generated per second.

4.2. Optimisation methods implementation

Some of the problems have been solved by using a suitably efficient mobile device. The performance increase was achieved thanks to the transition during testing for Samsung A5 (W2) and later Samsung S8 instead of the initially used Blackberry Priv (W1).

Some of the optimisation methods have been developed in accordance with the recommendations of the Unity platform manufacturer [23], i.e. such steps as:

- baked lighting (pre-calculated illumination that is not included in any run-time lighting calculations) and resignation from dynamic lights and using only static ones,
- shadows resolution reduction to minimise processing overhead,
- the use of occlusion culling (disabling rendering of objects not currently seen by the camera because they are obscured by other objects),
- the use of the level of details (LOD) technique (loading meshes in different resolutions depending on the distance from the camera),
- adding a mesh-combine mechanism (to reduce the overall number of draw-calls).
Apart from the changes in the Unity configuration, the application design and the virtual scene, it was necessary to modify the 3D models. The number of triangles for the general simplification of scene construction was limited (specialists working on creating models for the presented application had no previous contact with the limitations required for this type of application (W1)).

Three types of 3D model optimisation procedures were produced, depending on their size, complexity and origin. Objects are obtained using 3D parametric modelling (constructed of surfaces reflecting primitive fragments, e.g. bowl, sphere, cylinder, cone, wedge) were automatically simplified. 3D objects obtained using manual 3D modelling using mesh surfaces (including irregular shapes) were also manually requested – the designer decides on the local density of the mesh to correctly reproduce the shape of the object (however, this method is considerably work-intensive). Objects created from 3D scanners (correct reflection of dimensions and shapes) owe their accuracy to the mesh surface created in the triangulation process, containing a large number of points, and therefore triangles. Automatic reduction of the mesh of objects created from 3D scans can easily lead to excessive simplification (e.g. small rounding). In addition, the available mesh reduction tools do not allow interaction with selected areas of 3D objects. For these objects, a compromise must be made between reflecting the actual shape and the simplicity of the object.

Table 1. The comparison of the 3D model parameters before and after optimisation.

| Description        | Textured model | Object mesh before optimisation | Object mesh after optimisation |
|--------------------|----------------|---------------------------------|--------------------------------|
| Example            | ![Example Image](image1.png) | ![Before Optimisation](image2.png) | ![After Optimisation](image3.png) |
| Polygons           | Not applicable | 73914                           | 30031                          |
| Vertices           | Not applicable | 37041                           | 15100                          |
| Size without texture| Not applicable | 4.68 MB                          | 1.19 MB                        |

The 3DS Max’s Modify> Opti-mise tool was used, which allows to change the values of vertices and faces in a simplified model relative to the original model (Table 1). The tool was used in several repetitions to test different settings of different values so as to obtain the mesh of low, medium and high (but smaller than the original) number of vertices. In the case of a net with a low value of vertices, the desired value would not significantly affect the deterioration of the quality of the model.

5. Results and observations

Taking into account all the problems that arose during the configuration of the development environment for the purpose of creating a VR mobile application, and subsequent difficulties that occurred during the proper development, the authors state that the creation of such applications may turn out to be a very difficult and even daunting task even for experienced programmers who, however, had not previously created similar interactive visualisations. The correctness of the application depends on many factors (resources coming from many providers), which only a part
depends on the developer (e.g. optimisation of the application code, quality of the prepared assets). The remaining ones depend on many content providers (e.g. versions of the Unity environment, GVR-SDK versions, JDK versions, etc.).

At every stage of application life (creation, testing, implementation, and exploitation) one can notice the emergence of problems at various levels of abstraction, depending both on the software and hardware layer. A single error made during the configuration of the environment or a single error during the development may require a long backtracking search for sources of errors, effectively hampering the development of the project. In addition, the official documentation of the tools and software used is not always complete and does not provide information on the causes of the problems.

It has been noticed that in the optimisation process the models prepared by classic modelling have a greater advantage because they have a regular grid – this facilitated the use of manual and automatic optimisation methods. In the case of models obtained by 3D scanning, a large number of vertices that did not significantly affect the model's details were observed, which may be a problem during the process of automatic model optimisation [24].

Creating 3D models for use in VR mobile applications requires in-depth knowledge and experience in the field of modelling for projection with limited hardware resources. Otherwise, it will be necessary to modify existing models in such a way that they consist of fewer components and triangles, or in the worst-case scenario – creating models from scratch. The problem is the bigger the later it is identified and the more models it concerns. In both cases, the time of asset creation significantly lengthens and introduces additional work overhead.

As a result of the actions and tests carried out, it can be stated that:

1. The implementation of the optimisation methods described above reduced the average number of triangles in the scene view from 300,000 to 45,000 (over 6 times) and reduced memory usage from 25 MB to 5 MB (fivefold decrease), which resulted in a general acceleration of the application. The average value of fps was obtained at level 40, with an almost two-fold increase in value (21 fps). The value obtained is not fully satisfying because the recommendations of VR device manufacturers indicate the value of 60 fps as the minimum to ensure the proper level of immersion and reduce the risk of virtual disease.

2. As a result of the employed optimisation methods, a decrease in the temperature and an increase in the time of device heating during an uninterrupted VR session were noted. The device temperature elevation is due to the load on the components – processor, internal memory and screen, while the lack of an active cooling system (e.g. a fan), the device is not able to quickly remove the accumulated heat. Before entering the optimisation, the phone could warm up in a way that prevented its operation in a situation where it was necessary to remove it from the frame. Currently, this problem no longer occurs – the device heats up to a slightly higher temperature.

3. The changes introduced in the process of VR scene generation have contributed to the reduction of mobile device battery consumption. At the time when the VR application is running on the device, both the screen and some of the sensors remain in the mode of continuous activity, which directly translates into energy consumption. While these elements can be considered as a fixed part, the necessity of constant conversion of the scene view is variable and can be influenced by manipulating the computational complexity of the interaction and control algorithms used, or the complexity of the virtual scene, which will require the allocation of processor resources and operating memory for each frame, depending on the number and the complexity of the elements necessary to render. Before the introduction of optimisation, the battery discharged after 3 hours of uninterrupted VR sessions. After entering the described settings, this time has increased to 6 hours.

4. The studied group of students did not complain about the disruption of the application's fluency. All other comments of the research participants concerned specific situations and the way the application works. In most cases, they referred to personal aesthetic feelings and their own preferences regarding ergonomics.
The evaluation of the application's overall quality is subjective. It does not depend only on the quality of the software and the device, but also on the individual preferences and perceptions of users. Everyone has different preferences and their own quality thresholds determining the fitness and comfort of the application.

6. Future works
The application with various modifications will be expanded and used in further studies. In order to increase its functionality, it is planned to carry out new optimisation methods, including:

- size of textures mapped to generated models’ reduction,
- use of additional maps for materials to increase, compensate for the quality of models with simplified geometry.

The authors are also planning to carry out performance tests on a larger number of devices included in the group of low-end and high-end devices.

7. References
[1] Morrison A, Davies A, Brečević G, Sem I, Boykett T and Brečević R 2010 Designing performativity for mixed reality installations FORMakademisk 3
[2] Kim K H 2016 The Potential Application of Virtual, Augmented, and Mixed Reality in Neurourology Int. Neurol. J. 20 169–70
[3] Mateu J, Lasala J M and Alamán X 2015 Developing Mixed Reality Educational Applications: The Virtual Touch Toolkit Sensors 15
[4] Szymczyk T 2018 Interactive Educational Application using VR in a Smartphone 12th International Technology, Education and Development Conference pp 164–9
[5] Minocha S, Tudor A-D and Tilling S 2017 Affordances of Mobile Virtual Reality and Their Role in Learning and Teaching Proceedings of the 31st British Computer Society Human Computer Interaction Conference HCI ‘17 (Swindon, UK: BCS Learning & Development Ltd.) pp 44:1-44:10
[6] Wawrzyk M, Wesołowska K, Plechawska-Wójcik M and Szymczyk T 2019 Analysis of Brain Activity Changes Evoked by Virtual Reality Stimuli Based on EEG Spectral Analysis. A Preliminary Study Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology – ISAT 2018 ed L Borzemski, J Świątek and Z Wilimowska (Cham: Springer International Publishing) pp 222–31
[7] Agnello F, Avella F and Agnello S 2019 Virtual Reality for Historical Architecture ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLII-2/W9 9–16
[8] Ball C and Johnsen K 2016 An Accessible Platform for Everyday Virtual Education Everyday virtual reality (WEVR), 2016 IEEE 2nd workshop pp 26--31
[9] Innocenti E, Geronazzo M, Vescovi D, Nordahl R, Seran S, Ludovico L and Avanzini F 2019 Mobile virtual reality for musical genre learning in primary education Comput. Educ. 139 102–17
[10] Barbieri L, Bruno F and Muzzupappa M 2017 User-centered design of a virtual reality exhibit for archaeological museums Int. J. Interact. Des. Manuf. 12 1–11
[11] Carrozzone M and Bergamasco M 2010 Beyond virtual museums: Experiencing immersive virtual reality in real museums J. Cult. Herit. 11 452–8
[12] Moenning A, Turnbull B, Abel D, Meyer C, Hale M, Guerlain S and Brown D 2016 Developing avatars to improve cultural competence in US soldiers 2016 IEEE Systems and Information Engineering Design Symposium (SIEDS) pp 148–52
[13] Mayr S, Schneider S, Ledit L, Bock S, Zahradnicek D and Prochaka S 2017 Game-based cultural competence training in healthcare 2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH) pp 1–5
[14] Recupero A, Triberti S, Modesti C and Talamo A 2018 Mixed Reality for Cross-Cultural
Integration: Using Positive Technology to Share Experiences and Promote Communication

Front. Psychol. 9 1223

[15] Montusiewicz J, Skulimowski S and Lypovyi T 2017 Simulation of swarm behaviour as a method of examining students’ capability to conduct experiments - a case study 11th International Technology, Education and Development Conference pp 8655–64

[16] Kim J 2016 Modeling and Optimization of a Tree Based on Virtual Reality for Immersive Virtual Landscape Generation Symmetry (Basel). 8 93

[17] Chabal C and Soulabaille Y 2016 Virtual Reality: a way to prepare and optimize operations in decommissioning projects PREDEC 2016 pp. 2-10

[18] Montusiewicz J, Barszcz M, Skulimowski S, Kayumov R and Buzruk M 2018 12th International Technology, Education and Development Conference pp 353–63

[19] 3D digital silk road 2019 [Internet]. Available from https://silkroad3d.com

[20] Skulimowski S, Montusiewicz J, Badurowicz M and Barszcz M 2019 Virtual reality for cross-cultural education – a case study Preprint

[21] Mobile Operating System Market Share Uzbekistan 2019 [Internet]. Available from http://gs.statcounter.com/os-market-share/mobile/uzbekistan

[22] Lai Z, Hu Y C, Cui Y, Sun L, Dai N and Lee H 2019 Furion: Engineering High-Quality Immersive Virtual Reality on Today’s Mobile Devices IEEE Trans. Mob. Comput. 1

[23] Olly’s seven stages of optimizations for mobile VR 2019 [Internet]. Available from https://unity3d.com/how-to/optimize-mobile-VR-games

[24] Smaga M and Plechawska-Wójcik M 2017 Analiza zastosowania okularów 3D w symulacji przestrzennej na przykładzie wizualizacji architektonicznej J. of Computer Sciences Institute 10 187-192