GVRf and Blender: A Path for Android Apps and Games Development

Bruno Oliveira(✉), Diego Azulay(✉), and Paulo Carvalho(✉)

SIDIA (Samsung R&D Institute Brazil), Manaus, Brazil
{b.araujo,diego.a,p.alexandre}@samsung.com

Abstract. Virtual reality on mobile devices has been used in many areas such as entertainment, health care, training and simulations, although it still has limitations in terms of graphics quality and application performance. The ease of portable devices that can be used for virtual experiences in other "worlds" provides its fascinations, however, this technology can not match the virtual reality for personal computers, with high levels of quality in its graphics, with multiple features and high superior processing power while offering simplicity in use and being cheaper than its competitors on other platforms, such as personal computers and video game consoles. Thus, the purpose of this essay is to present a solution that GVRf, together with Blender, the GVRf Exporter, that makes the process of developing applications for VR technologies more optimized.

Keywords: Virtual reality · Gear VR · GVRf · Blender · Exporter · 3D

1 Introduction

In recent years, the improvement of new technologies such as Virtual Reality (VR) and its capabilities to transport users to immersive environments has been fascinating, researches to explore the potential of these technologies have allowed the creation of innovative experiences that have excited the interest of emerging technology enthusiasts [1]. The combination of these factors stimulates the search for efficient alternatives on the part of the developers for the optimization of resources, and for the process of development of VR systems, Gear VR framework (GVRf) with Blender present as an effective possibility.

The process of developing an application for VR systems basically consists of organizing the assets created by artists (concept, modeling, texturing, animation, etc.) into a set of libraries in the game engines where the application will be generated by adding the logical part of the game, in that all of these objects receive programming, as well as creating artificial intelligence and all the functionality for each application element. However the environment of these artistic tools like Blender [13] and Z-Brush [14] etc., are different from the engine used in the rendering and execution of the application. As they render materials, shaders and other elements show discrepancies from one platform to another. When artists are creating assets for an application, although they finish their work, in Blender, for example, needs to see how this feature behaves on the final platform and make adjustments to leave the scene as intended. In

J. Y. C. Chen and G. Fragomeni (Eds.): HCII 2019, LNCS 11575, pp. 329–337, 2019.
https://doi.org/10.1007/978-3-030-21565-1_22

addition, they need to see how the scene is on the device and how the end user will experience it. It is quite important to have this validation of the built scene to make sure that the user will have the best experience possible with the use of the application. However, one of the unwelcome problems in the development for VR mobile devices it's the necessity to always build an application (APK) to have a correct perception of how the application will perform, that is, create a new version of the app, this having a very significant impact on software development and art pipeline, making them both slow down because of this additional step. Generating a new APK build for any changes, even for small changes, can impact the pipeline as a whole, whether for use by an artist asset or a change in the code line, those small changes can make the workflow more difficult and tedious. As often as there are changes, such as 3D assets that can happen all the time, but this testing process is not ideal and even optimized built scenes creates one more step to be taken into consideration.

This was the problem that motivated the creation of a Blender add-on called GVRf Exporter, a Blender add-on that exports 3D scene components from Blender to a GVRf client application running on a mobile device. GVRf exporter, export the objects of a Blender scene to a mobile device so that the artist can preview the scene and see exactly what the user of the application will see. Thus, the 3D artist can view and validate animations, sharing animatics - a visual resource widely used for the preview of scenes with narrative structure, visual effects and other elements that after a previous analysis, will be produced later [17] - to other android devices connected to a VR and scenes where the application of physics to objects occurs and make adjusts as well as see in real-time the modifications done, making the creation process much faster, mainly for developers and artists who are looking for a VR mobile software solution and more flexibility to their pipeline. Also for who want to have an open platform that allows people to make any kind of change in their source code, to create some application for your own company without the need to purchase licenses from other development platforms, such as Unity or Unreal. Artists that use Blender and want to test a scene in VR, by doing in the cheapest way possible, using only Gear VR and their mobile device, that way anyone can create VR content only using Blender and GVRf.

2 Gear VR as an Alternative to Resource Optimization

Project development processes use Unity 3D – a cross-platform game engine designed to support and develops 2D and 3D video games, simulations for computers, virtual reality, consoles and mobile devices platform [2] - and GVRf - a lightweight and powerful open source rendering engine with a Java interface for developing mobile VR games and applications for Gear VR and Google Daydream View [3]. These development platforms require certain protocols for their operation, be it a game or a simple application. These protocols, in turn, have as their starting point the development recommendations of VR device manufacturers, such as Oculus [4], suitable for developers and content creators.

3 Realistic vs. Stylized

3D graphics have certain complexities that involve in certain artistic choices with regard to the visual of the applications/games. These choices often direct artistic style to a realistic look or stylized look, but portable virtual reality on mobile devices does not yet have the capabilities to support highly realistic graphics such as PCs can handle. This carries a simpler approach in artistic terms that translates into products or applications that largely use the low poly style, such as Look to the Sky [5] - an app developed with GVRF that gives to users a view of the night sky, the user can look at the stars according to his location - (Fig. 1) as a choice for a solution that blends artistic refinement with the optimized application/gaming capability.

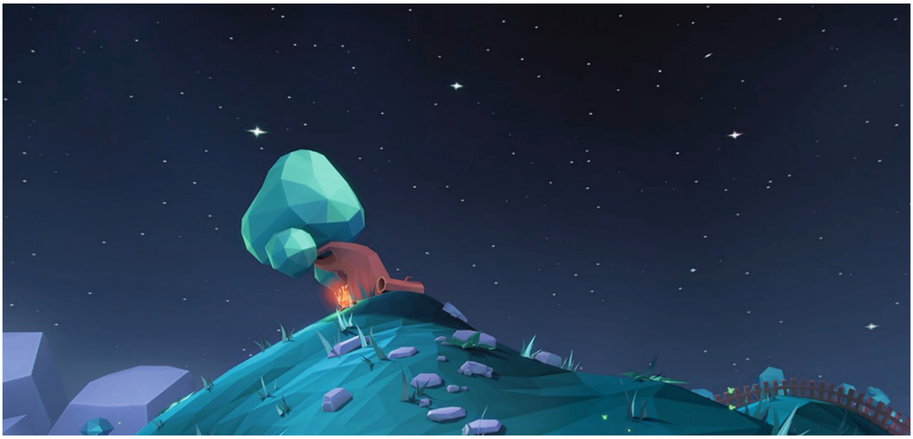


Fig. 1. Look into the sky [5]

4 GVRf

The Gear VR framework (GVRf) is a lightweight and powerful open source rendering engine with a Java interface for developing mobile VR games and applications for Gear VR and Google Daydream View [3].

The project GVRf is driven by Samsung Research America (SRA) [18] e SIDIA [19] collaborate creating news features and giving support to improvements other features.

5 Blender

Blender is an open source software that performs 3D computing tasks such as modeling, animation, texturing, composing, rendering using different techniques, such as raytracing, radiosity, ambient occlusion, or scanline rendering, video editing and creation of interactive 3D applications such as games through Blender Game Engine, an integrated game engine. Blender is available for all major operating systems under the GNU General Public License and is being actively developed under the supervision of the Blender Foundation [7].

5.1 General Application/Game Production Pipeline

A application/game production pipeline is basically a concept of workflow management for use in the game development process. The phases of this pipeline are certain tasks need to be fulfilled until the release of an application/video game [15]. The tasks are well defined for each phase, as The tasks are well defined for each phase (Fig. 2):

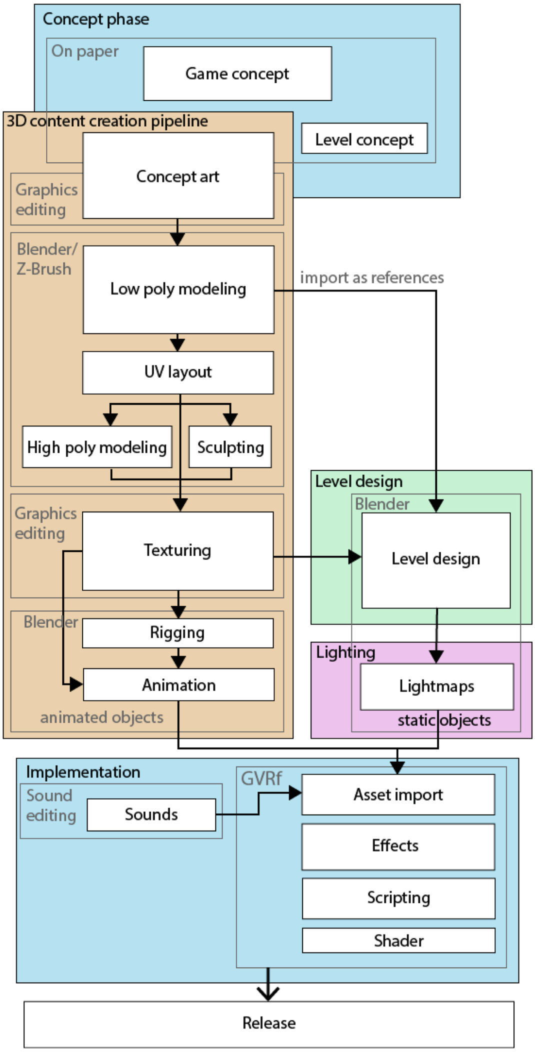


Fig. 2. Application/game production pipeline overview [15].

6 Blender Add-on

GVRf Exporter [6] is a Blender add-on that exports 3D scene components from Blender to a GVRf client application running on a mobile device.

GVRf exporter was created based on the necessity of accelerate the creation and exportation pipeline of assets for VR applications. There are some tools that makes a job similar to the GVRf exporter, like the Blender2Ogre [10] and Godot [11] blender exporter. But these tools are limited to export the scene from a 3D software, in this specific case, from Blender, in a way that the render engine will be able to import and render the content exported from the Digital Content Creation (DCC) Tool [12]. GVRf Exporter gives a step ahead because, not only it can export the content from DCC Tool, but shows immediately after the export the result on the final platform.

The idea for the creation of the add-on arose from a request of SRA for the development of a tool of creation of content with physics for GVRf, because until then, the only way to have a simulation of physics in a GVRf application was programmatically. That way Blender came into the scene, which, because it is open source, does not require the need to get a software license.

During the resolution of this case, a question arose was whether it was possible to perform the combination of work between Blender and Gear VR: how to simplify certain procedures for generating deploys of scenes created in the graphical environments of development platforms, such as Unity 3D [2] - where the pipeline consists of creating the scene with the assets in Blender and then exporting the set of assets to Unity 3D?

Based on this issue, several features were created so that you could export multiple objects between different Blender files, as well as scenes directly to an Android application installed on some device, so that you could benefit from the flexibility of a wireless device without the need to connect to another platform, such as a PC. One of the references on the market was Blinking Dalai Plug-In for Blender VR: That Concerning a preview of VR for the viewport in Blender 2.77 [16], only changed the fact that, instead of being rendered inside the computer was rendered on a mobile device via file transfer.

The GVRf Exporter is composed of three parts: remote scripting application, a http file server and the Blender add-on.

6.1 Components

Remote Scripting

It's a GVRf application capable of executing JavaScript commands (which are converted to Java) sent from a remote source. This is the application that shows the scene exported from Blender (Fig. 3).

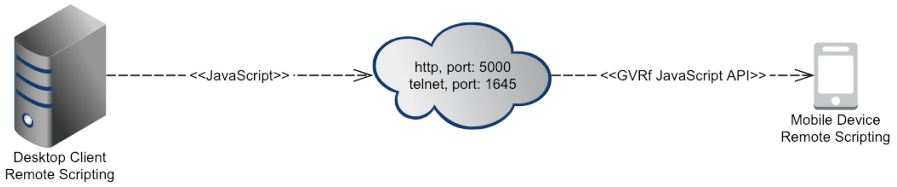


Fig. 3. Remote scripting.

Http File Server

The remote scripting application uses the file server to get the objects (.fbx) and textures exported.

Blender Add-on

The add-on provides a interface from Blender to the http server and the remote scripting client. The user can configure and export through its UI.

The Blender add-on is responsible for exporting files, sending JavaScript commands to the remote scripting application and maintain the http file server (Fig. 4).

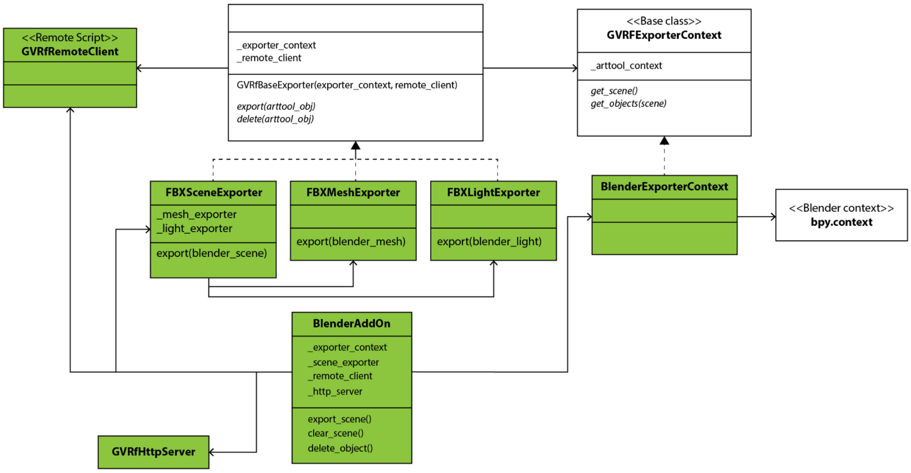


Fig. 4. Blender add-on.

The GVRf exporter can export the following objects:

- Meshes (with textures and armature);
- Lights (all the types supported by GVRf, which are Direct, Point, and Spot);
- Camera (GVRf supports only one camera per scene).

The available configurations are:

- Choose to export all objects or just selected objects of a scene;
- Choose the directory where to export the files;
- Set the scale of exported objects.

6.2 How to Use

Working with GVRf Exporter requires a number of steps. First, run the GVRf-remove-scripting client application on a device. Then, enable the GVRf-exporter add-on on Blender through user preferences menu (Fig. 5) and click on Import-Export tab of toolshelf (Fig. 6). Choose the directory to export (default dir is GvrfExportWorkspace located on user's home). Set Client's IP field to reflect the client's device IP address (Fig. 7), then click on Export button, then it will be possible to view the scene created in the Blender viewport directly on the mobile device.

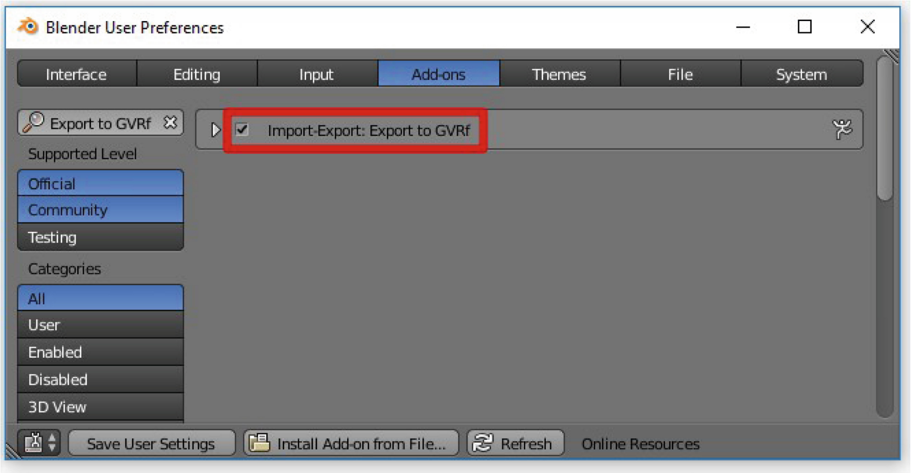


Fig. 5. Export to GVRf menu.

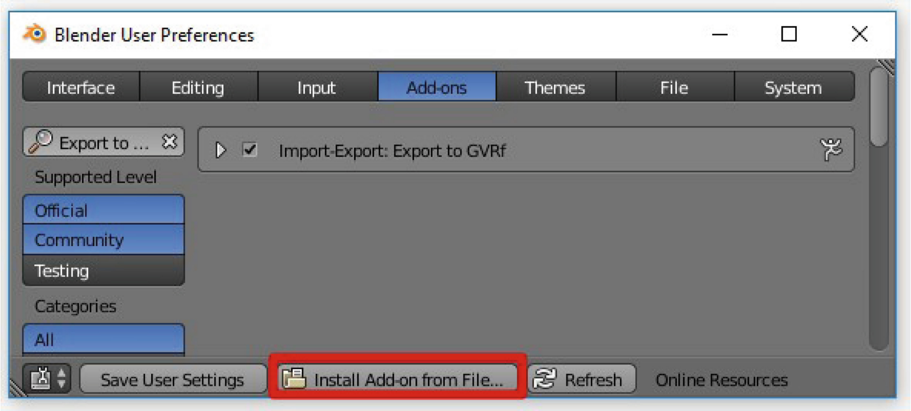
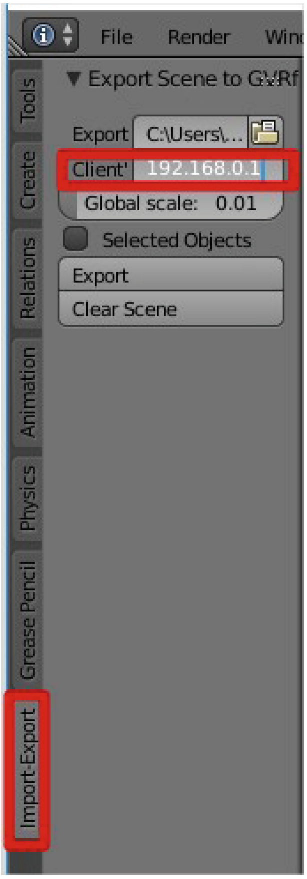


Fig. 6. Export to GVRf menu.



Export C:\Users\...

Client' 192.168.0.1

Global scale: 0.01

○ Selected Objects

Fig. 7. Client's IP field.

7 Conclusion

The research made possible the understanding of the content creation process for GearVR, as well as its development process in the Android environment, using GVRf as a flexible tool that optimizes the development time in the experimental stage of experiments with the help of Blender and GVRf Exporter. At the same time it became practical for modeling artists during the artist validation process, where they could observe how the assets would be in the scenes created by them, without necessarily having to send all the assets to the programmers to set the scene in development environment to be validated.

GVRf Exporter has some room for improvements. As future works, its UI can show more informations, including draw calls and the FPS (frame per second) of the running scene on the mobile device. Another function that will be befit its user is to export physics using the .bullet file format. This file is used both by Blender and the Bullet physics engine, which is the physics engine of GVRf. With this feature, it will possible to create physics for the objects inside the DCC tool and the device will run the scene with physics simulation without any additional code. Another issue to be solved is to maintain its stable performance after exporting and cleaning the scene successively. Finally, GVRf can be ported to another DCC tools, like Maya [8] and 3DSMax [9].

Acknowledgments. This essay was the result of the combined efforts of the SIDIA Solutions Team and SRA (Samsung Research America). It is also important to highlight the company's performance in supporting and promoting the research and development of systems that are present in the main technology products in the domestic market. We thank Samsung Electronics da Amazonia Ltda., as part of the results presented in this study were obtained from Project SXR SDK (Gear VR Framework), financed by the company under the Law 8387 (art.2)/91. For all the people involved in this project, our sincere thanks.

References

1. Psotka, J.: Educational games and virtual reality as disruptive technologies. Educ. Technol. Soc. **16**(2), 69–80 (2013)
2. Unity 2018. Unity Documentation – 3D projects. https://docs.unity3d.com/. Accessed 30 Jan 2019
3. Gear VR Framework. http://www.gearvrf.org
4. Oculus Developer Center. https://developer.oculus.com/documentation/
5. Look to the Sky VR App on Oculus Store. https://www.oculus.com/experiences/gear-vr/1236186189841229/
6. Blender Add-on for GVRF. http://www.gearvrf.org/blog/blender_plugin_guide
7. Hatka, M., Haindl, M.: Advanced material rendering in blender. Int. J. Virtual Reality **11**(2), 15–23 (2011)
8. Maya - Computer Animation & Modeling Software – Autodesk. https://www.autodesk.com/products/maya/overview
9. 3DSMax - 3D Modeling, Animation & Rendering Software – Autodesk. https://www.autodesk.com/products/3ds-max/overview
10. blender2ogre: Blender exporter for the OGRE 3D engine. https://github.com/OGRECave/blender2ogre
11. Godot-Blender-Exporter: Add-on for Blender to directly export to a Godot Scene. https://github.com/godotengine/godot-blender-exporter
12. Importing 3D Scenes - Godot Engine latest documentation. http://docs.godotengine.org/en/3.0/getting_started/workflow/assets/importing_scenes.html
13. Blender - Free and Open 3D Creation Software. https://www.blender.org
14. Pixologic: ZBrush - The all-in-one-digital sculpting solution. http://pixologic.com/
15. Aquino, M., Grashäftl, F., Kohl, S., Labschütz, M., Krösl, K.: Content Creation for a 3D Game with Maya and Unity 3D. CESCG (2011)
16. Felinto, D.: Virtual Reality Viewport in Blender 2.77 (2015). https://github.com/dfelinto/virtual_reality_viewport
17. Negrete-Yankelevich, S., Morales-Zaragoza, N.: e-Motion: a system for the development of creative animatics. In: Proceedings of the Fourth International Conference of Computational Creativity (2013)
18. Samsung Research America. https://www.sra.samsung.com/
19. SIDIA. http://www.sidia.org.br/