Porting Your VR Title to Oculus Quest

Eugene Elkin
eugene.elkin@survios.com
Survios, Inc.
Culver City, California, USA

ABSTRACT

Survios, a virtual reality (VR) game developer dedicated to building active, immersive experiences that push the limits of VR innovation, ported their VR boxing title, Creed: Rise to Glory, to the newly anticipated Oculus Quest. Porting to this new mobile VR platform is complex and demands extra creativity from developers when compared to porting to the previous generation of consoles. In this paper, Eugene Elkin, Senior Software Engineer at Survios, will share insights and learnings from the process, covering target hardware and its capabilities, rendering techniques, game optimization, and more.

CCS CONCEPTS

• **Human-centered computing → Virtual reality**.

KEYWORDS

virtual reality, VR, Oculus Quest

ACM Reference Format:
Eugene Elkin. 2019. Porting Your VR Title to Oculus Quest. In *Proceedings of SIGGRAPH '19 Talks.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3306307.3328202

1 INTRODUCTION

Survios, a virtual reality (VR) studio dedicated to premium software development, recently ported their VR boxing title, Creed: Rise to Glory (CRTG), to the newly anticipated Oculus Quest, the first all-in-one gaming system in VR. The decision to port CRTG to Oculus Quest, with its unfamiliar hardware and its studio's lack of mobile development experience, presented a significant challenge because the process did not begin until after the game had already been shipped on PSVR, Oculus and Vive. Target hardware and its capabilities are normally determined in the very early stages of a game's development cycle. A game's mechanic decisions and visual art style sometimes are not achievable with certain hardware, so these considerations are front-loaded during the stages prior to production. With Survios' previous three titles, PSVR was chosen as the lead skew, due to team's familiarity with the hardware and confidence that achieving the desired visual style at the proper frame rate was possible.

2 EXPOSITION

Developers for the Oculus Quest should become familiar with the "tiled-based rendering" techniqueâĂŤthe first rendering stage is batching up all submitted triangles into small, screen-aligned tiles. During the rasterization of these tiles, additional textures might be required by the pixel shader and thus, will be fetched; rinse and repeat for all other tiles. It's important to setup an effective debugging environment before further getting into the details of game optimization.

At Survios, Visual Studio (VS) is the preferred integrated development environment (IDE), however, debugging Android with VS can be problematic. The Nvidia Tegra plugin allows for Android remote process attaching to the Oculus Quest, but has proven to be quite slow when stepping through code or monitoring variables. As a solution, Survios suggests building the UE4 project with Gradle instead of Ant; a Gradle build can be loaded up into Android Studio IDE for a much quicker debugging process; EU4 provides a python script that should be set up as an LLDB Startup command to help with engine-specific data formatting during debugging. UE4 native graphics processing unit (GPU) debuggers will not be accurate due to tiled-based rendering, so RenderDoc is the preferred tool for GPU captures. It's important to note that pixel durations should still not be taken at face value, but can be used to determine the relative cost of the entire frame.

CPU profiling can still be done with UE4 native tools: command "stat startfile/stopfile" will record analysis of the CPU work and the data file can be pulled from the device back to the PC.

It's important to get a build running on the new hardware before any kind of profiling can be attempted. If this is not done, it's likely that a developer's first roadblock will be running out of memory due to the large texture assets and complicated materials. At the start of the porting process, Survios had a nimble team, so stripping all the materials in the level and replacing them with a cheap default shader served the purpose of getting the build to runâĂŤforcing the highest level of detail (LOD) on all the meshes put the GPU within the desired operational framerate.

Once the game is running with temporary visual reduction, developers can begin to inspect the initial bottlenecks: HDR, post-process and dynamic lights and shadows should be disabled from the start because they can be prohibitively expensive with tiled-based rendering. It is worth noting that post-processing requires a second render of each tile, with additional fetches of the frame buffer and possible depth buffer.

Fill rate of the GPU should, hopefully, not be an issue and framerate should be governed by the CPU or the draw calls. The UE command "stat unit" will display the frame time and should give a good indicator as to which task is taking the longest. If frame time is over 13ms, pause the game thread and inspect the frame rate; if frame time is under 13ms, then the frame is bound by the CPU.

Otherwise, if no change in frame time was observed, the number of objects being drawn is responsible.

There is no magic number for the maximum number of draw calls but, generally speaking, they should be kept under 200. In CRTG, Survios kept the draw calls under 150 by heavily merging game assets – the entire environment was merged down with 3-4 textures shared between them. Packing as many textures into atlases is recommended to minimize cache thrashing and normal maps should ideally be eliminated in many areas to reduce fetches. ASTC compression was applied to speed up the texture transfers, while sometimes, masked materials were converted to translucent as an optimization strategy made necessary due to the hardware's design. Non-essential materials were changed to fully rough because specular calculations can add a significant amount of instructions to the shaders. With a lack of post-processing, such as Bloom, which greatly influenced the original visual style, fake translucency blur objects were introduced into the game. The dynamic spotlights, which are crucial to the mood of the fight arenas, were simulated with material shaders (Figures 1 and 2).

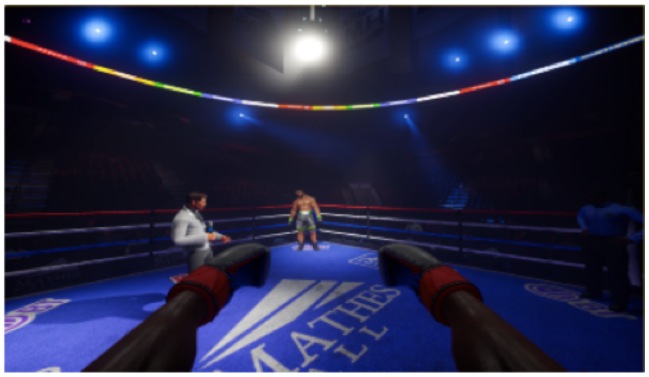

Figure 1: Oculus Rift Version. Image by Survios, Inc.

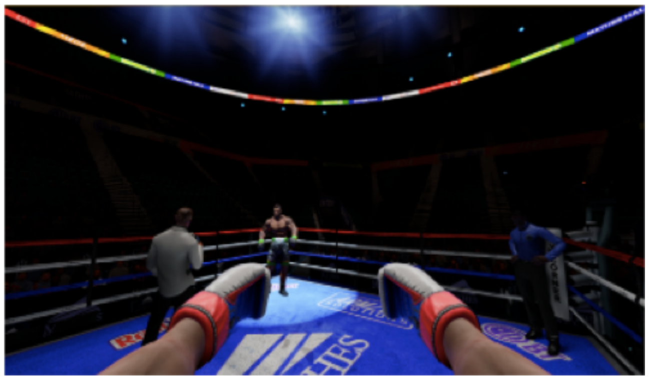

Figure 2: Oculus Quest Version. Image by Survios, Inc.

3 CONCLUSIONS AND FUTURE WORK

The final results had a similar look and feel when compared to the original game. The biggest takeaway from our porting journey is not to take any shader instruction for granted. Just 10 extra instructions on pixels that might occupy a large area of the screen can significantly impact performance; simplify the lighting model, avoid branching and keep texture sampling to a minimum. Survios will continue porting to Quest in anticipation of the headset's market success.