Performance-Aware Shader Authoring for Web-Based Virtual Reality Development

Alvis Misjuns, Arnis Cīrulis, Ieva Gintere

Vidzeme University of Applied Sciences, Valmiera, Latvia

WebXR allows users to create and experience virtual, augmented, and mixed reality content directly through popular web browsers, making these immersive experiences widely accessible. However, this ease of access introduces significant performance challenges, particularly on standalone VR headsets that have limited processing power. Maintaining high frame rates (typically 72–90 Hz) and low motion-to-photon latency (under 20 ms) is critical to preventing user discomfort. Yet, content creators, often artists without deep graphics engineering expertise face a performance knowledge gap, struggling to anticipate the performance costs associated with various shader functions. Current WebXR optimization workflows are predominantly reactive, involving a time-consuming cycle of authoring, deployment, profiling with tools like Chrome DevTools or Spector.js, and iterative refinement. This post-hoc approach provides limited guidance during the crucial design phase and is inefficient, especially given the performance variability across diverse device and browser combinations. Therefore, there is a clear need for preemptive performance feedback mechanisms integrated into artist-friendly shader authoring tools.

This research addresses this gap by establishing a benchmarking methodology and developing an intuitive, point-based scoring system for a shader authoring tool called FastShaders. The goal is to provide artists with actionable, real-time performance feedback during shader creation, enabling them to predictably balance visual complexity with runtime performance constraints. The primary contributions include: (1) ShaderCarousel, an automated WebXR benchmarking system built using A-Frame and Three Shading Language (TSL), designed to systematically evaluate shader performance on standalone HMDs; (2) a novel methodology to translate empirical benchmark data from ShaderCarousel into a predictive, point-based system representing shader operation costs; and (3) the FastShaders concept — an artist-centric, node-based shader authoring tool incorporating this preemptive scoring system.

The ShaderCarousel system presents a 3D environment featuring shaders applied to geometries positioned on a rotating circular platform aligned horizontally along the XZ-plane. As the carousel continuously rotates around the Y-axis, the system incrementally adds instances of a specific shader, evenly spaced around its circumference. A fixed-perspective camera oriented along the Z-axis toward the carousel's center captures the scene. This setup generates a dynamic visual effect wherein each geometry, as it rotates around the carousel, naturally transitions in size and visibility relative to the camera's viewpoint, allowing effective observation of shader performance across varying screen dimensions. ShaderCarousel logs key performance metrics, including frame rate (FPS), draw calls, and the maximum shader instance count achievable before significant FPS degradation occurs. Test scenarios include variations in camera proximity and shader instance loads. The collected data is exported as JSON for subsequent analysis.

The methodology for deriving the point-based scoring system involves identifying key shader operations, correlating them with measured performance data, and assigning initial point values based on their observed impacts. These point values are iteratively refined by comparing predicted aggregate scores for shaders against their actual benchmarked performance.

Initial results from ShaderCarousel demonstrate its ability to differentiate performance among various generative TSL texture shaders, with complex procedural shaders incurring significantly higher performance costs. A preliminary point system was developed, assigning costs to operations such as texture sampling, noise functions, and mathematical computations. This system provides a useful, albeit approximate, guidance mechanism. Early feedback helps artists optimize shader performance from the beginning, enabling them to explore efficient design options and understand performance implications proactively. This approach aims to minimize the need for extensive shader performance adjustments later in the development process.

This research establishes a viable methodology for providing preemptive performance feedback in WebXR shader authoring. The ShaderCarousel system and the FastShaders point-based scoring concept offer a practical approach to bridging the performance knowledge gap for artists, facilitating the creation of more predictable and performant WebXR content. Future work will focus on automating quantitative analysis for point derivation using statistical modeling or machine learning, expanding device and browser testing for broader applicability, conducting formal artist user studies to evaluate usability and effectiveness, integrating more granular GPU profiling data as WebXR standards evolve, and exploring dynamic adaptation of the point system to emerging hardware and runtime environments. Ultimately, this research contributes to democratizing performant WebXR development by empowering artists with tools for proactive performance management.

Keywords

WebXR, shaders, 3D, virtual reality, HMD, authoring