Using Vulkan for Graphics Research

Marco Castorina[1] and Gabriel Sassone[2]

[1]Senior Software Engineer at AMD
[2]Principal Rendering Engineer at Multiplayer Group

Abstract

The Vulkan API has has been released in 2016 and it has continued to evolve to include the latest hardware capabilities. Compared to OpenGL, it is a more verbose API that requires a deeper knowledge of the underlying hardware architecture. While this can make the API more difficult to get started with, it also rewards developers with finer grained control over resource management, multi-threading and work submission. This flexibility allows developers to achieve better performance over older APIs and opens the door to novel techniques that would have been harder, if not impossible, to implement before. In this tutorial we are going to provide an introduction to the core Vulkan API concepts and how they map to the underlying hardware. We are going to demonstrate how to leverage async compute to overlap graphics and compute work for better performance. We will provide detailed examples that make use of cutting-edge features like Mesh Shaders and Ray Tracing to achieve state of the art results in real-time rendering.

DOI: 10.2312/egt.20231031