

Cameron Schneider & Ben Hubner

Prof. Dan Buckstein

GPR-400 Adv. Rendering

12 February 2021

Annotated Research on Vulkan and Rendering Frameworks

Bauchinger, Matthias. "Designing a Modern Rendering Engine." *Diplomarbeit*, 9 Feb. 2018, www.cg.tuwien.ac.at/research/publications/2007/bauchinger-2007-mre/bauchinger-2007-mre-Thesis.pdf

This paper is about the conceptual design of a rendering engine from an academic perspective. While this resource does not go very deeply into the actual programming process of creating a rendering engine, it does have information about how a graphics rendering pipeline should be structured. It also goes into detail about why someone would want to create their own rendering engine, instead of using an already existing engine. We believe that this paper is a useful source; while we are going to be using Animal 3D for our project, so we do not have to build a rendering engine from scratch, we will be heavily modifying it, and having a helpful guideline for the structure of additions or modifications will be useful. This paper was published from the Vienna University of Technology, a major university in Vienna, so we believe that this is a reliable source.

Blackert, Axel. "Evaluation of Multi-Threading in Vulkan." *Digitala Vetenskapliga Arkivet*, Linköpings Universitet, 12 Sept. 2016,
www.diva-portal.org/smash/get/diva2:1037368/FULLTEXT01.pdf

This paper is about the uses and benefits of Vulkan when it comes to the topic of multithreading. Specifically, it compares the rendering performance of a scene rendered using both the Vulkan API, which can use multithreading, and the OpenGL API, which does not. For our project, we are interested in implementing multithreading in Vulkan, so having more research about that topic is very important. Also, since we are already familiar with OpenGL, it will be useful to look at a direct comparison between the two APIs. We believe that this source is reliable, as it was published by Linköping University, a large university in Sweden.

Gkeka, Maria Rafaela, et al. "Comparative Performance Analysis of Vulkan Implementations of Computational Applications." Computer Systems Lab. 9 Feb. 2021,
www.iwocl.org/wp-content/uploads/iwocl-2019-maria-greka-comparative-performance-analysis-of-vulkan-implementations-of-computational-applications.pdf

This source is a series of slides from a seminar about porting "Local Laplacian" filtering algorithm to the Vulkan API. The author(s) of this seminar go into extensive detail about the speed of different APIs working through this algorithm, and which API is best suited for their task. Since our project is about the usage of multiple graphics API for comparisons between them, we believe that this source will be useful for understanding what to expect when looking at different APIs. This is a published source from the University of Thessaly in Volos, Greece, so we believe that it is a reliable source.

Sellers, Graham, and John Kessenich. *Vulkan Programming Guide: The Official Guide to Learning Vulkan*. Addison-Wesley, 2016.

This textbook, written by Graham Sellers and John Kessenich, is a full guide to learning about the Vulkan API. It is full of information, from how Vulkan handles resources and memory, all the way to how it renders to the screen for viewing. Since we will be using the Vulkan API for our project, getting as much information as possible on how it works and how to set necessary pipeline features up will be exceedingly helpful. Also, since this book is on our class's list of recommended textbooks, and the authors have also written similar books on other API (such as OpenGL) in the past, we believe that it is a very reliable source.

Plateau, Alexandre. "The Beginning of My Journey: Making a Rendering Engine." *DEV Community*, DEV, 8 Oct. 2018, dev.to/superfola/the-beginning-of-my-journey-making-a-rendering-engine-18af

This article is a personal anecdote about a game developer's experience creating a rendering engine for his game. While we are not going to be making our own rendering engine, instead modifying Animal 3D to work with Vulkan and suit our needs, we believe that the experiences will be comparable, and this source will better inform us of what to expect during the process. Also, while this article is about OpenGL, the basic ideas should still be useful to us especially since, as we discussed in our API Showdown, Vulkan is compatible with OpenGL, so this will hopefully be of use. This article was written by a systems and network administrator for The Debian Project that has also created his own games and rendering library, so we believe that this source is informative and reliable.