Spulkan project plan

Mathias Plans, University of Tartu
MTAT.03.328

February 21, 2020

Introduction

Spulkan – short for Procedural Space Port with Vulkan and Rust – is a Rust port using Vulkan of my project from Computer Graphics course. The Space Generator project was written using Three.js. There are many reasons for the porting:

- Vulkan is high performance
- Vulkan and Rust support multi-thread programs
- Vulkan supports use of tessellation, geometry, computations and ray-tracing shaders.
- Vulkan is multi-platform. It works on Linux, Windows and MacOS

The last point is very important for me because I will be developing the project on Linux and the end of term expo will happen on a Windows machine.

This document will lay out the goals of this project, used technologies and forseen difficulties during the development.

Project Goals

The main goal of this project is to learn Rust programming language and Vulkan graphics library. Since most of the logic has already been developed, most of the focus will be on learning these new technologies.

There are many other goals for this project. Here is the list of all the goals:

- Learn Rust programming language
- Learn Vulkan graphics library
- Use tessellation or geometry shaders to get higher LOD on planets
- Use Newtonian mechanics for gravity calculations, instead of Kepler's laws
- Use GPU to do physics calculations
- Update the planet shaders so the surface looks more realistic

Required technologies

There are three main technologies that are used for this project:

- Rust programming languange. All the code that is ran on the CPU will be written in Rust.
- Vulkan graphics library. Vulkan has an API for Rust language. Vulkan also uses shaders.
- `shaderc` toolchain. This toolchain is used for compiling GLSL shaders into SPIR-V bytecode, which is used by Vulkan. Vulkan does not compile the shaders by itself.

Forseen difficulties

I have heard from many sources that Vulkan is really difficult. The success of the project will depend on the first few milestones. If I will not get Vulkan working by the end of the first milestone, all the other goals will have to be delayed, some of them even removed.

When using shaders with Three.js, the source code is modified. For example, attributes such as poistion, color, and normal are built into the library. When I start porting the project to Vulkan, I also have to implemt those attributes myself.

The last difficulti I am forseeing is the compatibility between Linux and Windows. Rust and Vulkan are multi-platform, but the building scripts and `shaderc` usage might not be.