High Performance Graphics

Zoe Wall (40182161)
BSc Games Development
Supervisor: Dr Kevin Chalmers
Second Marker: Prof Jon Kerridge

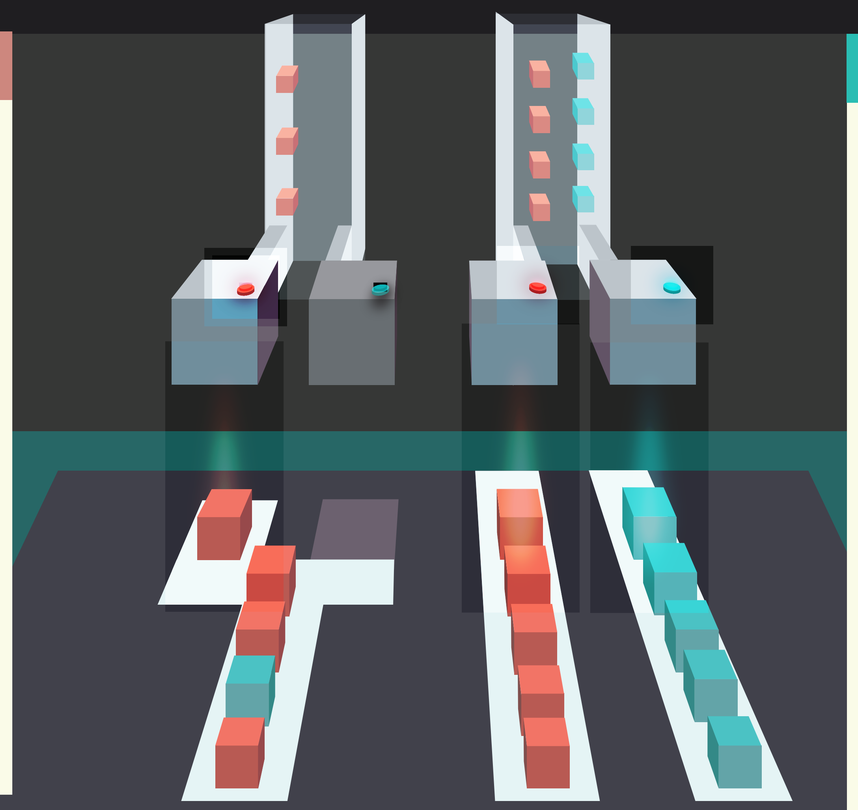
Edinburgh Napier UNIVERSITY

Introduction

The aim of this project is to develop a 3D rendered application to analyse the effect of **Asynchronous Compute** on rendering performance.

The objectives of the work are as follows:

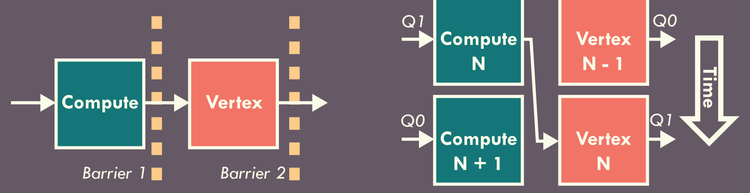
- Undertake a critical evaluation of the current literature and technology surrounding Asynchronous Compute.
- Gain in-depth knowledge and understanding of a new low-level graphics API, **Vulkan**.
- Design and implement an application, using Vulkan, that attempts to leverage the use of *asynchronous work queues* to improve performance.
- Gather experimental results of this application, following a scientific methodology.
- Analyse and evaluate the results and implementation with regards to performance, and ease of implementation.



Methodology

Modern hardware allows for a novel improvement on the current GPGPU techniques, by allowing for concurrent processing of graphics and general tasks on the GPU - this is known as Asynchronous Compute. The Vulkan API was used to develop a **N-Body** simulation with 3 separate modes; a baseline compute example, and two different async techniques.

By scheduling commands differently, the physics (compute) and rendering (e.g vertex) processing could be executed at the same time.



Experiments were conducted - altering parameters such as the number of particles rendered, and the resolution of each mesh. This formed the basis of an evaluation into the affects of varying workloads on the performance of each simulation.

Conclusions

Overall, the project was a success. The results show an increase in performance using certain Asynchronous Compute techniques.

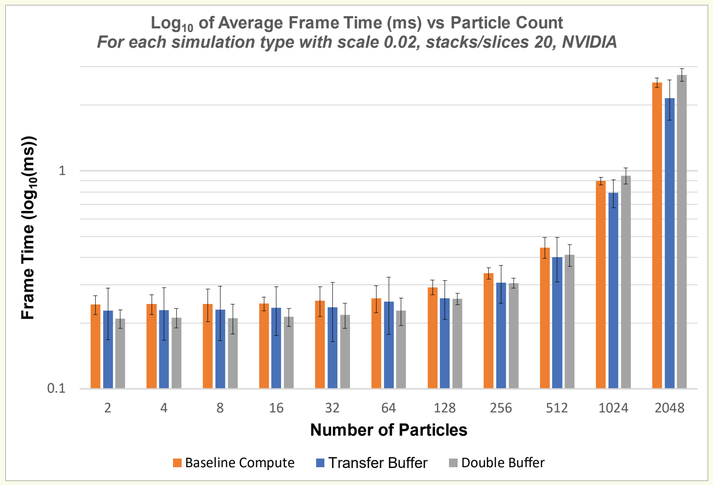
The experiments confirmed that there is a distinct difference between vendors in how the propriety drivers and hardware handle asynchronous commands. Even though the GPU on average, didn't perform the tasks asynchronously, when the graphics time for a frame reached over 4ms an overlap of the shaders was recorded. This is significant, as it could be proof that NVIDIA finally has support for true concurrency across queue families.

Although promising results, the study's biggest limitation is that the application was CPU bound. Unfortunately, the developed scene was not sufficiently detailed, and therefore not expensive to begin with.

Future work includes: a more realistic benchmark simulation, (e.g frame times ≈30ms), more detailed study into the overheads of Async and if possible mitigate them.

Experimental Results

The results for the Asynchronous Transfer simulation show a performance increase, ranging from 1.063x-1.175x speed-up. This is an interesting result, as it shows **that the more work completed by the GPU - the more speed-up is achieved.** This means that with this technique, the GPU is working **more efficiently as the workload increases.**



Whereas the results for the Asynchronous Double Buffer simulation, *are the complete opposite*. The speed-up ranges from 1.158x to 0.920x, which actually shows a **decrease** in performance for 1024 particles and above. This result could be due to the overheads of implementing Async.

Profiling Results

Current industry standard profilers, have **no support** for both Vulkan and NVIDIA GPUs. However, by running the simulations on an AMD card, results showed positive for Async compute.

As the profilers were not conclusive enough to prove that NVIDIA was asynchronous, time stamps were also recorded at the start and end of both the compute and graphics launches to compare. These timings showed that for the Transfer Simulation, NVIDIA performed some frames asynchronously.

The results show a **correlation between performance loss and Async Compute for higher workloads.** This is for the Double Buffer simulation on AMD hardware.