Assignment P2 – Vectorization

Formal assignment description for P2 - INFOMOV
Dionysis Alexandridis, Jacco Bikker, 2023

Introduction

This document describes the requirements for the second assignment for the INFOMOV course. For this assignment, you will improve the performance of an existing application using SIMD and GPGPU.

Base Code

The code for this assignment can be found on the course website and on Teams. It implements a basic cloth simulation on a 256x256 grid using a technique known as *Verlet integration*.

The goal of this assignment is to improve the performance of the application using vectorization. This can be accomplished with two main technologies discussed during INFOMOV: SIMD and GPGPU.

Your main optimization target is the function `Game::Simulation()`. This function executes the Verlet integration in three passes using the `for` loop on line 117. Please do not change the number of passes. A pass consists of:

1. Applying gravity and wind: lines 120 – 126
2. Satisfying constraints: lines 129 and beyond.

Not part of the optimization assignment is function `Game::DrawGrid` on line 87. Please leave this code unmodified.

Tasks

For the code in `Game::Simulate` we have formulated four tasks that let you practice your SIMD and GPGPU skills.

1. Improve the performance of the loop on lines 120-126 using SIMD. Doing this right probably requires a reorganization of the application data. You score up to 2 points if your SIMD code improves the overall performance of the application.
2. You can also score 2 points if you execute lines 120-126 on the GPU. For this, send the relevant data to the GPU once per pass, execute the GPU code, and copy the code back to the GPU for further processing. It is not necessary to improve the performance of the application this way; merely getting it to work is sufficient. In fact, we expect this to slow down the application.
3. For the third task we ask you to complete the SIMD conversion of the `Game::Simulate` function. This is a significant task and requires more data reorganization. Be aware that processing a point affects its neighbours; you may want to carefully bundle jobs to avoid concurrency problems. When done correctly this should yield at least a 2x speedup (using SSE) or 4x (using AVX); if you achieve this – without breaking the simulation – you score up to 4 points.

4. The final task takes the entire simulation to the GPU. For four points, we ask you to implement `Game::Simulate()` on the GPU. Note that concurrency challenges are amplified now that 32 threads run in lockstep.

You may execute any or all of these tasks for the final result. Your grade will be clamped to the range [1..10] as usual.

Team

You may work on this assignment alone, or with one partner. You may team with one partner for all assignments, but it is also allowed to change teams per assignment. You cannot change your team halfway an assignment; if for whatever reason you don't want to finish the project with your partner, both of you will work alone. Both team members may continue working with the code that was produced up till the split.

You may exchange information about the project with other students, online or in real life. Do not share code snippets; limit the exchange to ideas, hints, and concepts. This is in fact encouraged.

Deliverables

Your submission will consist of a brief **report** plus **project files**. Make sure the code compiles out-of-the-box in VS2019 and/or VS2022. If any other tools are required to produce the intended executable, please add a `readme.txt` that contains build instructions.

Deadline

The deadline for this assignment is **Friday June 2, 17:00**. If you fail to meet this deadline, you may submit one day later. One point will be subtracted from your grade in this case. Please submit your work using Teams.

Academic Conduct

The work you hand in must be your own original work, or properly referenced. If you used materials from other sources, please specify this clearly in your report.

Do not store your work in a publicly accessible location (this includes github!). If other students use your work (now or in the future), you may be reported along with the perpetrators.

The End