# Assignment P1 – Low Level Optimization

Formal assignment description for P1 – INFOMOV – Jacco Bikker, 2019



### Introduction

This document describes the requirements for the first assignment for the INFOMOV2019 course. For this assignment, you will apply low level optimizations to a small application that uses the CPU to render a spinning galaxy disc. Initial profiling indicates that rendering is the bottleneck. The current application struggles to reach 30 frames per second for 10,000 particles. The aim is to improve performance so that 50,000 particles can be rendered at 30fps, without affecting the quality of the animation.



## Sub-pixel

The application uses a modified version of the DrawScaled function that we encountered before. This time, the function is named DrawScaledAdditiveSubpixel:

- Additive refers to the fact that pixels are added to the screen buffer. This way, a pixel will
  never reduce the brightness of what has been drawn before.
- Subpixel indicates that the sprite will look different when drawn e.g. half a pixel to the left.
   To achieve this, bilinear filtering is used while sampling the source image.

The new functionality comes at a cost. It is much harder (maybe / probably impossible) to precalculate scaled sprites, and the calculations themselves are quite intense.

Apart from sprites the application also draws dots, also with sub-pixel accuracy. A single star in the system will smoothly move from one pixel to the next. Combined with the sub-pixel sprites, this yields a very smooth animation. Obviously, the optimized application should be just as smooth.

#### **Low Level Optimization**

Your task is to <u>make the application faster</u>. For the purpose of this assignment, you are asked to optimize without using the GPU, and without using multi-threading. Just low level optimization in other words, as discussed in lecture 2.

You may want to apply the 'rules of engagement':

- 1. Avoid costly operations
- 2. Precalculate
- 3. Pick the right data type
- 4. Avoid conditional branches
- 5. Early out
- 6. Use the power of two
- 7. Do things simultaneously (limited; no GPU, no multithreading!)

All, some, or most of these may apply; this is up to you.

#### 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.

### Deliverables

For this assignment, you only have to hand in the optimized project, in source form, by e-mail. Make sure the code compiles out-of-the-box in vs2017 and/or vs2019. If any other tools are required to produce the intended executable, please add a readme.txt that contains build instructions.

#### Grading

Your final product will be graded on absolute performance alone. For this, I will compare against my best effort. Handing in the original code yields a 1, meeting or surpassing my result yields a 10 (and treats for the whole class), all other numbers are linearly interpolated.

#### Deadline

The deadline for this assignment is **Tuesday September 24, 23:59**. If you fail to meet this deadline, you may submit one day later. One point will be subtracted from your grade in this case.

#### The End

Questions and comments:

bikker.j@gmail.com or room 4.24.



INFOMOV 2019