# Assignment P1 – Low Level Optimization

Formal assignment description for P1 – INFOMOV – Jacco Bikker, 2024


**Introduction**

This document describes the requirements for the first assignment for the INFOMOV2024 course. For this assignment, you will apply low level optimizations to a small application that uses a genetic algorithm to approximate a color image with a set of lines. Profiling indicates that the (anti-aliased) line drawing code is the bottleneck.

*You can find the base code for this assignment in **Teams -> General -> Files -> Class Materials**; the name of the archive is evolution24.zip.*

**Wu Antialised Lines**

The application uses Xiaolin Wu's line algorithm to produce antialiased lines (see Wikipedia: https://en.wikipedia.org/wiki/Xiaolin_Wu%27s_line_algorithm). The original implementation is from Codeproject member '.Suchit' and can be found on https://www.codeproject.com.

**Low Level Optimization**

Your task is to make the line rendering faster. Overall application speed is measured as 'ips' (*iterations per second*). The application caches part of the drawing, to save on the number of lines. As a result, the 'lps' (*lines per second*) figure fluctuates, but a 'peak' lines per second is also reported.

For the purpose of this assignment, several limitations to the optimization process must be enforced:

- Do not use multithreading to speed up the code.
- Do not use the GPU to speed up line rendering.

So: Just low-level optimization, 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**

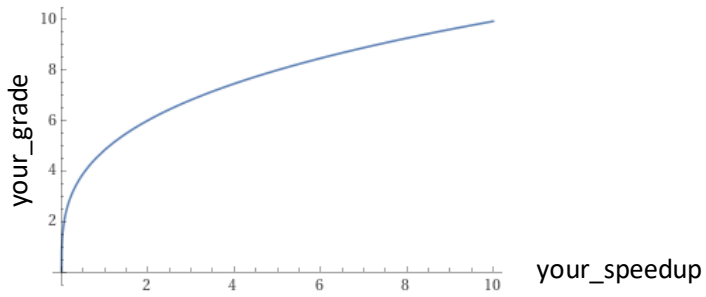
For this assignment, you only have to hand in the optimized project, in source form, via Teams. Make sure the code compiles out-of-the-box in Visual Studio. Please include a **tiny readme**: Use it for build instructions, and: **The score you expect** (see below).

**Grading**

Your final product will be graded on relative performance alone. For this, I will compare against my "best effort". Handing in the original code not faster than the original scores a 1.0. Meeting my result scores an 8. All other grades are calculated using the simple formula:

```
your_grade = clamp( 8 * pow( your_speedup / my_speedup, 0.3125 ), 1, 10 )
```

Assuming I can do 5x, the graph looks like this:




If one or more students score a perfect 10, there will be treats for the whole class.

**Deadline**

The deadline for this assignment is **Friday May 10, 17:00**. Please submit your work through Teams. If you fail to meet this deadline, you may submit one day later, i.e. on May 11, 17:00. One point will be subtracted from your grade in this case.

**The End**

Have fun! Questions and comments:

bikker.j@gmail.com.


**INFOMOV 2024**