Topics for licență final projects – Simon Cowell – Fall 2022

1) Fast primality testing

Some internet security protocols require fast generation of large numbers that are prime, or that are "probably prime".

You will implement in code both the Agrawal-Kayal-Saxena (AKS) polynomial-time primality testing algorithm from 2002, which is deterministic, and the Miller-Rabin primality test from 1980, which is probabilistic. You will compare their theoretical time complexities, as well as their speed in practice.

In 2002, Agrawal, Kayal and Saxena published the first ever deterministic algorithm of polynomial time complexity for determining whether an arbitrary given number is prime, whose proof of correctness does not depend on the generalised Riemann hypothesis, nor on any other mathematical conjecture. This answered, in the affirmative, the long-standing question of whether the primality decision problem was in the class "P":

https://people.csail.mit.edu/vinodv/COURSES/MAT302-S13/AKSpaper.pdf

In this project you will:

a) Implement the AKS algorithm in code.
b) Implement the Miller-Rabin probabilistic primality test in code.
c) Carry out empirical measurements of the time complexity of both programs and plot the results.

2) The Burrows-Wheeler Transform

The Burrows-Wheeler transform is a simple transform which has several applications, notably in genomics. That is, in DNA sequencing, in particular, in efficient pattern matching for sequence alignment, and in lossless compression of DNA sequences. It is also useful for lossless compression of text, and of certain types of image. You will implement the Burrows-Wheeler transform using a simple algorithm of time complexity $n \log n$, and using a more sophisticated algorithm of only linear time complexity, and you will compare the timing results empirically, on large texts and on large DNA sequences.

3) Programming the Graphics Processing Unit (GPU) using OpenGL

As a pre-requisite for this topic, you should complete the first 7 exercises in the pdf file "Exercises for Computer Graphics Fall 2022", which I have posted on the SUMS platform.

OpenGL is a widely used Application Programming Interface (API) for programming graphics. It is commonly used to control the GPU in order to achieve hardware-accelerated rendering. You will control the GPU using OpenGL in C to generate fast visualisations of interesting graphics. You will do either one of the following two projects:

a) Implement the 3-dimensional geometry processing algorithms described in Keenan Crane's notes "Discrete Differential Geometry: An Applied Introduction":

https://www.cs.cmu.edu/~kmcrane/Projects/DDG/paper.pdf

for example, for simplification of high polygon-count meshes used as 3d assets in computer games etc., or

b) Implement some Physically-Based Rendering (PBR) techniques, including Global Illumination (GI), which are often used in modern photorealistic Computer Generated Imagery (CGI).