Judicaël CLAIR

clair.judicael@gmail.com
+44 758 593 3800
32 Stanford Road, W8 5PZ, London, United Kingdom
Portfolio: https://www.doc.ic.ac.uk/~jsc18/ (C++ web app)

Education

- Bachelor of Engineering in **Joint Mathematics and Computer Science** at **Imperial College London** 2018 — 2021
  - **3.8/4.0 GPA** (averaging **77.39%** overall. — grade conversion to GPA endorsed by Emeritus Professor Susan Eisenbach).
  - Awarded the **Olav Beckmann Project Prize** for outstanding second year project work.
  - **Object Grasping with Haptic SLAM** (final year individual research project)
    Supervisor: Professor Andrew Davison
    Developing a novel method for **grasping objects** with a humanoid hand. The goal of this project is, given no knowledge of the surroundings and **relying exclusively on haptic feedback**, to locate, identify and subsequently hold a simple object.
- **A-Level** (Lycée Français, London, United Kingdom) 2017 — 2018
  - **3 A***s (top grade) Maths, Further Maths, Physics.
  - **2 A**s Chemistry, French.
- **GCSE** (Lycée Français, London, United Kingdom) 2016
  - **8 A***s (top grade) Maths, Chemistry, Physics, Biology, French, Spanish, Chinese, Computing.

Work Experience

- **Arm — Part time Undergraduate** December 2020 — Present
  Cambridge, England, United Kingdom
  Developing accelerated computer vision and machine learning algorithms for augmented reality on Arm mobile platforms.
- **Arm — Software Engineer Intern** June 2020 — September 2020
  Cambridge, England, United Kingdom
  Key Skills: **C++**, **Python**, **SYCL**, **OpenCL**, SPIR-V, Bash, CMake
  - **Successfully ported DPC++**, a **C++ compiler** for **heterogeneous compute**, to an Arm CPU + Mali GPU platform.
  - Modified DPC++'s build system and C++ code base to cross-compile from an x86 host architecture to Arm.
  - Augmented LLVM's testing infrastructure to support remote execution of tests on development boards — used Python.
  - Fixed bugs that occurred when running tests on the remote physical systems.
  - Added **support** for **32-bit devices** (e.g. clvk), collaborated in fixing memory layout issues and SPIR-V device code generation.
- **Cleanuc — Software Consultant** July 2017 — February 2018
  Villebon-sur-Yvette, France
  Developed software in **Python** from the ground up that automates **modelling nuclear installations for safety analysis**. Implemented **machine learning** algorithms that manipulate and output Excel data with the whole process being abstracted away by a carefully designed graphical application.
- **Fire Tech — Volunteer Teaching Assistant** July 2016 — August 2016
  London, England, United Kingdom
  Taught kids between the ages of 9 and 13 to code in **Python** and **Java** as well as helping them with their Arduino electronic projects.
- **Imperial College London — Summer Research Assistant** June 2015 — June 2015
  London, England, United Kingdom
  Learnt about the Caspian Sea and ways researchers identify climate change. Interpreted a large data set to estimate the depth of the sea using **MATLAB**.

Personal Projects

- **Humanoid Robot Hand** (3D printed) 2019 — Present
  - Devised **custom**, compact **hardware** for high bandwidth position sensing in order to support high frequency sampling of dozens of sensors simultaneously.
  - Designed human-sized fingers (limited space) with precise joint position feedback using potentiometers and Hall effect sensors.
  - Key Technologies
    - Personally developed C++ framework **Clarity (Studio)** — see **below** — for visualisation and control.
    - **FPGA** programmed in **VHDL** and **MATLAB** for interfacing with peripheral hardware (sensors).
    - Fusion 360 for **3D modelling** / CAD.
- **Clarity — C++ multi-threaded application engine** 2017 — Present
  - **Built from scratch** with a continuing emphasis on the application architecture and **scalability**. For example, invoking `application::destroy()` will lead to a well-defined clean destruction of the entire application regardless of the complexity of the code base and calling context.
  - Feature-rich **work stealing thread pool task scheduler**:
    - Static and dynamic **task dependencies**.
    - Recurring tasks (keeps rescheduling itself until aborted).
    - **Task cancellation** from any context based on a task handle, tag or regex.
    - Robust **wait system** with notifications, timeouts and forced wake-ups due to state changes (e.g. task aborted).
    - Many edge cases had to be considered with one of the most challenging being: finding the most natural way of aborting a task group (set of inter-dependent tasks — a static dependency construct) midway through its execution without causing catastrophic unexpected behaviour or stalling.
  - **Utility library** contains various bits and pieces (e.g. plugin system) with a significant part of it being over-engineered template code — leads to an extremely elegant code base since there is significantly less code bloat.
  - **Graphics** utilities, which leverage open-source graphics engines (e.g. google's filament).
  - **Clarity Studio** is built on top of this library and offers an interactive and extensible development environment.
- **Ruined Kingdoms — 2D fantasy multiplayer online role playing game (MORPG)** 2012 — 2016
  - Adapted the open source **Python** project PyORPG (PyGame for graphics and Twisted for networking).
  - Rewrote the entire GUI from scratch for customisability (e.g. input boxes, scrollable text boxes and a window system).
  - Reworked the game logic and networking (e.g. item, NPC and user systems) - added new features & fixed flaws.
- **Breadboard Computer** 2015
  - 8-bit TTL computer built on a breadboard using discrete logic with a design similar to the SAP-1 architecture.
  - Can perform **basic addition** and **subtraction** with a seven-segment display as output (signed and unsigned integer format).
  - Programs can be written into the RAM using dip switches with one of the most complicated programs it can run to be the **Fibonacci sequence** because of its limited amount of memory (16 bytes).

Computing Experience

~**9 years programming experience** consisting mainly of **C++**, **C**, **Python** and **Java**.

- **C++** (see personal projects and Arm internship above)
  - **template metaprogramming** and latest C++ standard features.
  - **multithreading** (task-based and lock-free).
  - **heterogeneous compute** (**SYCL**, **OpenCL**, **CUDA**).
  - **graphics** (**Vulkan**, **OpenGL**).
  - Managed a group of 3 in developing a C++ **web app** (WebAssembly) for analysing & tracking C++ build statistics.
- **Python** (see personal projects and work experience above)
  - **PathBench** October 2020 — January 2021
    - Benchmarking framework for 3D **robot path planning** — coordinating a group of 6.
    - Classic algorithms (e.g. A* and RRT), as well as ML-based planners are both supported.
    - Rich visualisation of 2D and 3D simulations with Panda3D.

- **C** (predominantly school group projects)
  - Led a team of 4 in implementing a **primitive operating system**: multi-level feedback queue scheduler, synchronization primitives (lock, semaphore, . . .), user programs.
  - Cooperated on developing an Arm **assembler - compiler** and **interpreter**.
  - **Bare Metal arcade game** similar to Asteroids but on steroids, running on a Raspberry Pi (directed a group of 4).
    - Developed a **custom fixed-point library** (sin, cos, arctan, ...) — **no third-party libraries** were used for the project.
    - Interfaced with joystick using SPI (bit-banging) and video output (**GPU interfacing**) through memory-mapped I/O.
- **Java** (extensively used at university)
  - Team lead in designing and implementing a compiler for the WACC language, which outputs optimised (uses a graph colouring algorithm) Arm assembly code that is inter-operable with C code.
- **Hardware** (**VHDL**, **MATLAB**, **FPGAs** - Arduinos used for quickly verifying hardware correctness [e.g. wiring])
  - Implemented UART serial communication with error detection and correction.
  - Interfaced with sensors (IMUs, ADCs, Hall Effect sensors) using I2C & SPI.
  - Constructed Simulink models (i.e. using MATLAB) for signal processing (e.g. FIR filters) and then manually integrated the generated VHDL code with my hardware interface.

Other interests & skills

Languages

- **Native Speaker** in **French** and **English** (**dual citizenship British and French**).
- Intermediate proficiency in **Chinese** (enjoy travelling to China almost every year).

Extracurriculars

- Running almost daily.
- Intensive boxing (sparring) — member of the Imperial College Boxing Club.
- Cycling — climbed the highest road passes in Europe and North America among many others.
- Climbed Mount Kilimanjaro.