TIAGO JOSÉ SOUSA MAGALHÃES

Website: tiagojosemagalhaes.github.io
Email: tiagojsmagalhaes@gmail.com
Phone Number: (+351) 911785903

HIGHLIGHTED PROJECTS & EXPERIENCE

Software Systems Engineer at BOLD — November 2020 - Current

· Employed by BOLD and subcontracted to Efacecs platforms team in the R&D Division as a software systems engineer. As a part of this team, my various functions include:

- Development, feature backporting, and, Maintenance of the operating system kernel using in Efacec's products. This includes the base Linux kernel and any required hardware drivers.
- Development, porting, and, Maintenance of software packages to be used in the operating system distribution used in Efacec's products.
- Research and development of optimized software solutions for the needs of Efacec's products and clients' needs.
- Supporting other product development teams with developing solutions specialized for the various types of hardware used in Efacec's products.

Real-Time Ray Traced Voxel Global Illumination

· My master's dissertation proposes a technique to be used in real-time computer graphics that mixes hardware-accelerated ray tracing with voxelization techniques already used in rasterized computer graphics. This technique aims to achieve a balance between performance and graphical fidelity.

- Completely decouples ray-tracing resolution from video output resolution.
- The presented solution utilizes a 3-stage pipeline: rasterized-based surface voxelization, ray-traced radiance injection, and final image synthesis.
- Briefly investigated the option of building the ray-tracing BVH acceleration structure based on the results of the voxelization stage, however, between transferring data from the GPU to the CPU and intersection performance, this approach was discarded early in development.
- Implemented a resource descriptor memory allocator which attempts to minimize stalls from locks in multi-threaded scenarios.
- Implemented using a fully custom rendering engine written in C++17 using the DirectX 12 rendering API and the Windows API. This engine is my own creation, except for the usage of tinyobjloader for model loading and Dear ImGui for a debug interface.
- Debugged using mainly the Nvidia Nsight Graphics debugger and the visual studio C++ debugger.
- Utilizes a CMake-based build system.

Embedded Demotics Networked Device System

· This project developed a system that was meant to emulate the capabilities of the KNX protocol in an open and easily extendable manner. Developed as a part of an embedded systems class that taught us about scheduling and schedulability, general embedded development, and development standards and practices for critical systems (such as MISRA C).

- Developed for Espressif's ESP32 platform using C++17, the FreeRTOS API, and ESP32' API.
- Developed using a hierarchical dispatcher task architecture.
- Required the development of a virtual device system that enables developers to write fully custom device drivers.
- Required the development of facilities for simplifying interactions with network devices.

TECHNICAL SKILLS

Programming Languages	C++, C, C#, x86 Assembly, Java, Python, PHP, Javascript, NodeJS
IDEs	Visual Studio, Jetbrains Suite, 4coder
Profilers	CodeXL, Intel VTune, NVidia Nsight Graphics & Compute
Database Management Systems	PostgreSQL, SQLite, MS SQL Server, MongoDB
CI/CD Platforms	Gitlab CI, Circle CI, Jenkins
Notable APIs	DirectX 12, OpenGL, WINAPI, UNIX APIs

LANGUAGES

Portuguese	Native
English	C2 Level

EDUCATION

University of Porto - Faculty of Engineering (FEUP) *September 2016 - July 2020*

Integrated Master's Degree
Computation and Informatics Engineering (Software Engineering) - Dissertation Final Mark: 18/20

Polytechnic of Porto - School of Engineering (ISEP) *September 2015 - September 2016*

Bachelor's Degree
Computer and Electrical Engineering
At the end of the 1st year, I transferred to FEUP's software engineering degree.