Erfan Momeni Yazdi

Website | LinkedIn | GitHub | Twitter

Location: Tampere, Finland
Email: erfanmo98@gmail.com | Mobile: +358 407385966

Software Engineer

Experienced professional with a strong passion for programming and solving challenges. Advanced knowledge of C++, C# and graphics programming. Always eager to learn and explore new concepts. Logical thinker with excellent problem-solving and communication skills.

Work Experience

Research Assistant — Sep 2022 – Present
Tampere University — *Tampere, Finland*

- **Master's thesis: Cloud powered VR rendering:** Contributed to a large-scale project focused on photo-realistic rendering of digital twins for VR. Was responsible for **Developing an open-source, Linux-based system** to offload rendering tasks of a standalone VR headset to a remote headless server, achieving a low latency of 20 ms with path tracing. This work involved extensive collaboration and communication within our research group at Tampere University. (video link) .
- **Optimized and Enhanced TauBench 1.0**: Improved a specialized benchmarking tool by resolving redundancy issues, achieving a **2x improvement in render time** (reduced from 36s to 12s) and a **3x reduction in file size**(from 2.7 GB to 900 MB) (Link)
- **Designed and developed TauBench 2.0** dynamic benchmark. TauBench 2 was designed to further stress test real-time renderers with much more dynamic lights and fast moving objects (paper in progress)

Education

Tampere University — Tampere, Finland
Master of Science in Computing Sciences: Signal Processing and Machine Learning — *Sep 2022 – Oct 2024*

K.N. Toosi University of Technology — Tehran, Iran
Bachelor of Science: Computer Engineering — *Sep 2016 – Aug 2021*

Projects

Syndra — *C++, OpenGL, Git, RenderDoc, ImGui* — Source Code

- Designed and developed a physically based renderer and **Game Engine** using **OpenGL API** and **C++**
- Implemented Rendering Algorithms such as **Deferred and Forward Plus Rendering** and **dynamic soft shadows**
- **Data oriented** Entity Component System (ECS) with an editor to modify the components
- Implemented PBR material models (Demo video link)

Shader Programming — *GLSL, ShaderToy* — Source Code

- Practiced shader programming by implementing ray-tracing concepts like **SDF**, **ray marching** and **Monte Carlo estimation** (Link)

Ray Tracing — *C++, Git, Path-tracing, Multi-threading* — Source Code

- Implemented a multi-threaded **CPU Path Tracer** with C++

Technical Skills

Languages	:	C, C++, C#, Python, Java, GLSL, x86 assembly
Graphic APIs	:	OpenGL, Vulkan, DirectX(familiar), OpenXR
Libraries	:	CMake, ImGui, Assimp, SDL, GLFW, Premake, Monado
Dev Tools	:	Git, Visual Studio, Linux, Github, Gitlab, CI/CD RenderDoc, Tracy, Nsight
Softwares	:	Blender, Unity, Unreal Engine

Selected Courses

- Computer Graphics (5/5)
- Algorithms Design (A)
- Linear Algebra - Foundations to Frontiers by UTAustinX (Certificate)