# Tejaswa Rizyal

tejaswarizyal4@gmail.com | 0786 432 12 98 | trizyal.github.io | in/tejaswarizyal | github.com/trizyal

## Personal Profile

A Masters of Engineering student in Computer Science with a specialisation in High Performance Graphics and Games Engineering. Experience in modern graphics techniques and high performance computing with a focus on real-time rendering. Developed a game engine using C++ and Vulkan targeting stylised rendering and PBR. As well as worked on a geometric mesh simplifier and repair toolkit. I also have a history in floating point arithmetic research and low level systems.

## Education

**University of Leeds**, MEng BSc in Computer Science 2020-2025

- Graduating in July 2025 with a First Class Honours.
- **Relevant Modules:** Computer Graphics, Advanced Rendering, Modelling and Animation, Data Structures and Algorithms, Parallel Computation

**Delhi Public School** 2016-2020

- **A Level Equivalent - 94.4%:** Computer Science, Mathematics, Physics, Painting, English, Chemistry.
- Appointed as the President of the Student Council in 2019-20 term.
- **GCSE Equivalent - 93.4%:** 7 subjects including Computer Science, Mathematics and English.

## Projects

**C++ Game Engine using Vulkan** 2025

- Wrote a game engine from scratch specialising in Vehicular games and unique stylistic effects.
- Developed the renderer which is capable of object specific rendering and runtime effect incorporation.
- Implemented PBR using Cook-Torrence BRDF and shadow maps for realism.
- Developed a object masking system for stylised rendering like toon shaded smoke using a unique algorithm.
- Created a built-in profiler for the renderer, and optimised it to run at 100fps for a highly stylised and object populated game.
- Added post processing effects such as bloom, chromatic aberration and glitch effects.
- Utilised Vulkan for MSAA and implemented additional sharpening filter and FXAA.
- *Tools Used:* Vulkan, C++, premake, GLFW, glm, GLSL.

**Car based 3D Bullet Hell Game** 2025

- Created a stylistic bullet hell where the player controls a car and demolishes enemy robots.
- Designed and integrated my own game engine to utilise different stylistic gameplay effects.
- Utilised the engine's pooling system to stay memory efficient and spawn 100s of enemy bullets all stylised to look like energy balls.
- Implemented different power-ups using the ECS and physics systems to provide speed boosts, sliding and health generation capabilities.
- Ensured target 100fps all the while utilising all postprocessing effects.
- *Tools Used:* C++, glm, UBQR Game Engine.

**Modern Rendering Visualiser in Vulkan** 2024-2025

- Developed a Vulkan based renderer, to evaluate and utilise different rendering techniques.
- Implemented a forward renderer with the ability to visualize depth, overdraw and overshading.
- Integrated render-to-texture and post processing functionalities with multiple render pipelines.
- Refactered and reused the program to implement deferred shading and shadow mappings.
- Examined GPU Optimisations and Data Bandwidth Reduction methodologies.
- *Tools Used:* C++, Vulkan, premake, GLFW, glm, gdb, obj files.

**Optical Physics Raytracer** 2024-2025

- Developed a CPU based raytracer, implementing projective and ray-traced image rasterisation.
- Implemented Blinn-Phong Shading with direct and indirect lighting including shadows.
- Adapted optical physics concepts and material properties such as reflection, refraction and total internal reflection using 3D Geometry and Vector Mathematics.
- *Tools Used:* C++, OpenMP, Linux, premake, gdb, GLFW, obj and mtl files.

**Mesh Processing Tools** 2024

- Developed tools to process triangle soups and convert them into .obj and/or .face files.
- Implemented further functionalities to check whether the mesh is manifold and calculate its genus.
- Improved on the functionalities and created algorithmic methods to fix given meshes, including holes, pinch points, and testing shared edges.
- *Tools Used:* C++, Qt, Linux, obj and tri files.

**Floating Point Accuracy in JuliaLang** 2023-2024

- Researched and developed a software to calculate all floating point calculation errors in the Julia Mathematics Libraries for 40 transcendental and elementary functions.
- Implemented functionality to be able to run parallel tests across all available threads to cover all 4 billion+ binary32 floating point numbers.
- Testing for accuracy against a higher precision library in 40 minutes per function, and faster in HPC systems.
- Implemented a binary tree style search strategy to find ranges that show the highest errors for binary64 floating point numbers.
- Presented the findings as the final year project for BSc, securing a high first class grade for the research.
- *Tools Used:* JuliaLang, GNU MPFR Library, OpenMP, Linux.

## Work Experience

**Research Assistant**, University of Leeds July 2024 – Aug 2024

- Researched further on the project written for the BSc with the supervisor.
- Redesigned and overhauled the interface to improve general usage aside from purely academic users.
- Increased functionality of the software by adding the ability to process and test functions with bounded domains, such as `asin` and `acosh`.
- Investigated and extracted from world leading research in floating point researches to have a holistic cover of the research area.
- *Tools Used:* JuliaLang, GNU MPFR Library, OpenMP, Linux.

**Software Engineer**, Honeywell July 2022 – July 2023

- Operated in a large C++ codebase with Qt Libraries for one of a kind fire panel solution in a low level Linux environment.
- Streamlined the HMI/HCI layouts to best serve both users and developers by reducing redundant UI applications and improving modularity of the codebase.
- Spearheaded a new feature to reduce start up inefficiencies and adding special UI features to improve the UX design as expressed by the test users.
- Developed in an Agile environment with sprints, and engaging with the project planning and feature implementations with developers across teams and countries.
- *Tools Used:* C++, Qt, Linux, Yocto Project

## Skills

**Languages:** C++, C, Python, Julia, Bash, GLSL, SQL, LaTeX, Java

**Technologies:** Vulkan, OpenGL, Linux, OpenMP, OpenCL, MPI, gdb, Qt Libraries, cmake, qmake, premake

**Skills:** Agile Software Development, Optimisation, Low-Level Programming, Research, Debugging, Documentation, DevOps, Linux CLI.