

Arnav Kumar

✉ a8kumar@uwaterloo.ca 🌐 arnavcs 🐙 arnavcs.github.io in Arnav Kumar

Skills

Programming: Python, C++, Rust, Go, JavaScript, TypeScript, Haskell, Racket, Java, Bash, L^AT_EX

Tools: WebGL, Three.js, Pandas, Matplotlib, NumPy, Tensorflow, Scikit-learn, OpenCV, Linux, Unix, Godot, Protobuf

Experience

Software Engineer (Graphics, 3D Geometry) @ Arcol.io

Sep 2025 - Dec 2025

San Francisco, USA

- Designed and implemented a Rust terrain triangle mesh generation algorithm based on Delaunay triangulation and interpolation between terrain contours.
- Performed prototyping, iteration, and proof-of-concepts to choose and implement the most suitable approach.
- Rendered generated terrain and contour lines, sent over the Wasm ABI to TypeScript, with Three.js and WebGL.

Software Engineer @ Trend Micro

Jan 2025 - Apr 2025

Kanata, ON.

- Engineered utility functions for Alicloud's Object Storage Service and Parameter Store, optimizing cloud workflows and enhancing secure data management.
- Developed serverless functions to send scan metrics and results to the backend, helping identify details about customer errors and significantly cutting customer costs.

Combinatorics Researcher @ University of Waterloo

May 2024 - Aug 2024

Waterloo, ON.

- Authored a paper (The dimension of sparse random graph orders) in pre-print in arXiv (<https://arxiv.org/abs/2504.19029>).
- Worked on novel research exploring random graph orders and poset dimension utilizing configuration models, approximation techniques, convergence in probability, devising constructions, and computationally searching for counterexamples.
- Developed two theorems about the nature of poset dimension for bipartite posets based on the dimension of induced subposets and for random graph orders.

Blockchain Software Engineer @ Dandelion Networks

May 2023 - Aug 2023

Online

- Developed and implemented an enhanced lattice syncing and node discovery algorithm in Go, querying peers with Protocol Buffers to identify and address missing blocks in the local lattice with secure and concurrent updates.
- Created and deployed a locally hosted debugging website in Go to be run by each node of the blockchain network.
- Designed multiple mock servers, clients, and services with Ginkgo and Gomega to test the algorithms.
- Created Jenkins pipeline which automatically build, test, vet, and format pushed code to catch regressions.

Achievements

- 335th (top 10%), with score 26, **William Lowell Putnam Mathematical Competition** 2022
- 10th in Canada, **Asian Pacific Mathematics Olympiad (APMO)** 2022
- 18th in Canada, 2 time qualifier, **Canadian Mathematical Olympiad (CMO)** 2021 - 2022

Projects

4D Raytracing Pathtracer | C++

- Implemented SDF raymarching in 4D space with CPU pathtracing to construct GIF renders of the scene.
- Encoded the render such that every frame in the GIF, the camera sends out rays into a 3D affine subspace of the 4D space, which are scattered into 4D space upon colliding at a surface.
- Designed and wrote a powerful material system with microfacet BRDFs, glass, volumetric fog, and more with the ability to assign more than one material to a geometry.

Rigid-body Particle Simulation | C++

- Simulated gravity, spherical constraints, and collision physics for rigid body particles.

Software Rasterizer | C++

- Recreated the whole rasterization process on the CPU, including perspective transformation, triangle rasterization, depth buffering, and perspective correct interpolation.

Software Raytracer | C++

- Implemented the Möller-Trumbore algorithm for fast ray-triangle intersection
- Programmed Lambertian diffuse, specular refraction, and specular reflection behaviour, and support for spherical environment mapping

Renovating the Labyrinth (Game) | JavaScript

- Solo submission made in 72 hours with vanilla JavaScript on an HTML canvas for the UW Game Dev Club's fall 2024 game jam; voted winner of the technical achievement award.

- Built a real time optimized ray caster with ordered Bayer matrix dithering.
- Programmed 2D rigid body collision behaviour and a randomized Prim's algorithm for map generation.

InterPlanetary File Explorer (IPFE) | *Go, Python, Scikit-learn, Estuary, Co:here, Three.js*

- Created vector embeddings for files with their headers using Co:here's NLP embeddings to facilitate classification of files.
- Performed principal component analysis of the vector embeddings to reduce the dimensionality from 4096 to 3 to be plotted and displayed interactively in 3D space using Three.js.

Prognosing Idiopathic Pulmonary Fibrosis (IPF) | *Python, Tensorflow2, Pandas, Scikit-learn*

- Implemented an auto-encoder, linear regression, dense neural network, and bayesian model in order to accurately predict future lung capacity and give a confidence value using initial lung capacity data, age, sex, smoking status, and more.
- Obtained a Laplace Log Likelihood score of -6.9 (much better than the baseline score -8.1) with $\sigma \approx 200\text{mL}$.

Education

Candidate for BSc. Computer Science (4th year) @ University of Waterloo
Waterloo, ON.

2022 - Present

- 4.0 GPA, all advanced level courses, and term distinction.
- Received the Ronald G. Dunkley National Scholarship (\$18,000), the President's Entrance Scholarship (\$3,500), the NSERC undergraduate research scholarship (\$6,000), and the Cenovus Energy STEM Scholarship (\$18,500).