Prasurjya Pratim Talukdar

Assam, India | omnimistix@gmail.com | linkedin.com/in/prasurjya | github.com/omnimistic | omnimistic.itch.io

Education

[IIT Dholakpur] [Dholakpur]
[B.Tech in Computer Science] *[2031]*

Experience

Software Developer & Game Developer (Independent) Jan. 2021 – Present
Omnimistic *Assam, India*

- Developed and published multiple interactive projects and tools via Itch.io and GitHub, focusing on systems programming and game mechanics.
- Engineered custom UI solutions and language parsers to bridge the gap between low-level performance and high-level ease of use.
- Managed full-cycle development including architecture design, asset creation, and cross-platform deployment.

Projects

Rosemary | *C++, OpenGL, LuaJIT, Python, IPC*

- Architected a standalone UI Server using a Godot-inspired node-tree, decoupling visual presentation from logic via a "Signal-over-Pipe" model.
- Developed a custom C++ engine utilizing OpenGL 3.3 and MSDF for resolution-independent text rendering and hardware acceleration.
- Implemented an IPC bridge using STDIN/STDOUT, allowing external controllers to drive UIs without the memory overhead of Chromium/Electron.
- Built Rosemary Studio, a native C++ editor for visual layout, property inspection, and scene serialization into custom .rose configurations.

PAIN | *Python, CLI, CMake, vcpkg*

- Developed a zero-friction C++ Project Scaffolder and Dependency Manager built entirely in **Python**, designed to automate modern C++ environments without manual CMake configuration.
- Engineered a CLI tool that abstracts **vcpkg** and **CMake** workflows, enabling users to create, manage, and build projects in seconds across Windows, Linux, and macOS.
- Implemented cross-platform build orchestration, handling compiler detection (MSVC, MinGW, GCC) and automated boilerplate generation for high-performance native applications.
- Optimized developer experience (DX) by providing a unified, Python-driven interface to manage complex C++ toolchains and external library integrations.

AlgoLang | *C++, Compilers/Interpreters, Memory Management*

- Developed a high-level interpreted programming language from scratch using **C++**, implementing a robust **Recursive Descent Parser** to handle complex arithmetic with full BODMAS/PEMDAS precedence.
- Engineered a custom memory management system to support dynamic variable allocation for multiple data types (Integers, Floats, Strings) and managed scoped execution states.
- Implemented structured control flow including nested `WHILE` loops, `IF/ELIF/ELSE` blocks, and a modular execution engine capable of both legacy `GOTO` jumps and modern block-based syntax.
- Authored comprehensive technical documentation and released the project under the **MIT License**, ensuring modularity for future expansion into algorithmic visualization and logic execution.

Technical Skills

Languages: C++, Python, Lua, GDScript, GDShader, HTML/CSS, JavaScript
Engines & Frameworks: Godot Engine, Flutter, Tailwind CSS
Libraries & APIs: OpenGL 3.3, SFML, Pygame, OpenCV, Three.js, p5.js, MSDF Rendering
Developer Tools: Git, Neovim, CMake, vcpkg, CLI, Linux
Graphics & Multimedia: Blender, Figma, Krita, LibreSprite, GIMP, Kdenlive, Audacity