

# Mohamed Gadalla

Game Engine Developer

## CONTACT

Phone: +201097450184

Address: Giza, Egypt

Email: mohamedag202@outlook.com

Portfolio: <https://frodoalaska.github.io/projects>

## PROFESSIONAL EXPERIENCE

### C++ Game Engine And Tools Engineer

Freelancing: Jun/2022 – Present

- Upgraded and refactored a legacy proprietary game engine from DX9 to DX11.
- Converted a custom proprietary 3D animation binary format into FBX for use with the Unity Game Engine.
- Wrote a scalable CMake build system with test integration, replacing legacy MSBuild, enabling full cross-platform deployment.
- Wrote Python, Bash, and PowerShell scripts to automate tests and build pipelines across Linux and Windows, ensuring ease-of-use and efficiency.
- Integrated the NuGet package manager with CMake to enhance usage of third-party libraries.

## PROJECTS

### Nikola – 15<sup>th</sup> November 2025 / 10<sup>th</sup> 2026

A custom game engine made using C++ and OpenGL specifically for level-based games with emphasis on speed, efficiency, and ease-of-use

- A fully-featured and modular OpenGL-based renderer, supporting PBR shading, post-processing effects, simple shadow maps, particles, and instancing.
- Designed a proprietary binary format for optimized asset loading, reducing load times by almost 40% across all assets.
- An integrated 3D physics system, using the Jolt physics library.
- A robust 2D UI system featuring animation for each widget (sliders, checkboxes, buttons, more).

### Freya – 15<sup>th</sup> Jan 2026 / Present

A fully-featured 2D game engine

- Cross-platform window creation with OpenGL 4.5+.
- A flexible and configurable 2D renderer, using modern OpenGL (4.5+).
- A fully-fledged audio system with both 2D and 3D spatialized audio.
- A robust and easy-to-use game UI system, using HTML and CSS.
- A robust 2D physics engine, using the fantastic Box2D library.
- A flexible Entity Component System (ECS) module, using the EnTT library.

## SKILLS

### Languages

- C++
- Python
- C#

### Graphics

- OpenGL
- Direct3D 11
- ImGui

### Build Systems

- CMake
- Make
- MSBuild

### Tools

- RenderDoc
- Visual Studio
- Unity