Artem Kirienko

(813) 709-9515 – artem.kirienko1999@mail.ru – 5009 N Sheridan Rd, Chicago, IL 60640

CAREER OBJECTIVE

Aspiring Software Engineer with a strong foundation in graphics programming, real-time systems, and multithreaded development. Adept at designing and implementing custom game engines, optimizing performance, and creating scalable frameworks for complex systems. Seeking to leverage my expertise in DirectX, HLSL, C++, and Software Design Patterns knowledge to deliver cutting-edge real-time technology.

SUMMARY OF QUALIFICATIONS

- C++ Software Engineer for a NASA-funded project on Drone Traffic Management Ecosystem.
- Proven experience in Real-Time Game Systems and Multithreaded Development.
- Experience working with large-scale projects (150+ classes).
- Competitive character, striving to be the best of the best and compete against smarter people.
- Demonstrated analytical and quantitative problem-solving skills.
- Proven capacity to manage workloads under extreme time pressure and deadlines.
- Fluent in English and Russian(native speaker).

EDUCATION

DePaul University (Chicago, IL).....**Jan/2023 – March/2025**
Master of Science in Software Engineering

- *Concentration on Real-Time Game Systems*

University of South Florida (Tampa, FL).....**Aug/2017 – Aug/2022**
Bachelor of Science in Computer Science

RELEVANT DEVELOPMENT EXPERIENCE

Additional information and videos of the projects below can be found at: artushdeveloper.github.io/portfolio/

Game Engine Development `[C++/DirectX 11/Google Protocol Buffers]`

- Designed and programmed a custom game engine from scratch using **DirectX**, **HLSL**, and **C++**, showcasing expertise in graphics programming and real-time rendering.
- Engineered a robust framework to encapsulate high-level abstractions such as meshes, textures, cameras, lights, etc., enhancing scalability and modularity.
- Developed **skeletal animation** and **mesh skinning** systems, enhancing the engine with realistic character animation through matrix transformations and efficient execution through DirectX Compute Shaders.
- Compressed animation clip data by a factor of ~3x with no perceivable visual degradation, optimizing memory usage and improving runtime performance.
- Programmed HLSL shaders to handle multiple dynamic and static lights in the scene including Point, Directional, Spot and Area lights based on Blinn-Phong BRDF.
- Implemented a **Volumetric Deferred Lighting Pipeline** capable of efficiently handling up to **10,000 lights** (~360 lights with forward lighting) in a scene with no noticeable drop in framerate, enabling realistic and scalable lighting effects.
- Programmed custom **static libraries** to support the engine framework, including a **Math library** for vector, matrix and quaternion operations, a **Memory library** for efficient memory management, and an **Object library** for game objects hierarchies and resource handling.
- Created an **offline mesh conversion tool** to process and convert .glb files into an engine-compatible format, ensuring seamless integration and reducing runtime overhead using Google Protocol Buffers.
- Conducted thorough debugging and performance profiling using tools such as **RenderDoc** and **Visual Studio Graphics Debugger**, enhancing engine stability and efficiency.

Artem Kirienko

(813) 709-9515 – artem.kirienko1999@mail.ru – 5009 N Sheridan Rd, Chicago, IL 60640

Drone Traffic Management Ecosystem `[C++/Unreal Engine 5]`

- Collaborated with a team of six programmers to develop a **NASA-funded** drone traffic management system, using **C++** and **Unreal Engine 5** to create a scalable and efficient ecosystem.
- Owned the implementation of a **video streaming feature** using **GStreamer**, enabling real-time transmission from drone cameras to a dedicated server using the **RTP protocol** and **H.264 video format**.
- Optimized the project's **multithreading architecture**, improving responsiveness under high-load conditions.
- Performed local code optimizations to boost overall performance and ensure efficient system operations.
- Contributed to the development of the system's **networking architecture**, ensuring reliable communication between drones and servers for real-time data exchange.
- Delivered a robust and optimized solution for drone traffic management, demonstrating expertise in real-time video streaming, networking, and multithreaded programming.

Omega Race Multiplayer Networking Layer `[C#/.NET/Lidgren Network]`

- Enhanced an existing codebase for Omega Race game by adding a networking layer, transforming it into a multiplayer game with a **Client-Server architecture** for two players.
- Developed robust communication systems using the Lidgren networking library, enabling seamless packet transmission between clients and a central server.
- Implemented **data-driven queues** for efficient packet management, ensuring reliable message exchange over the network.
- Achieved precise game state synchronization with **Cristian's algorithm** for clock synchronization and a **Lock-Step protocol**, ensuring deterministic and consistent gameplay across clients.
- Designed and implemented **Client-Side Prediction** and **Dead Reckoning** techniques to compensate for network latency, enhancing responsiveness and player experience.
- Created a debugging Record/Playback tool that records game sessions in real-time and allows playback to reproduce and diagnose errors, improving the debugging process and system reliability.
- Delivered a fully functional multiplayer networking layer, showcasing expertise in real-time networking, latency compensation, and debugging tools.

Game Performance Optimization `[C++]`

- Reengineered core data structures to improve memory usage and data caching, achieving a **10x speed boost** in system performance.
- Optimized the **vector** and **matrix math library** by leveraging **SIMD Intrinsics**, resulting in a **5x performance improvement** in mathematical computations.
- Developed a custom heap-based memory system, providing a **3x speed increase** over the default Microsoft memory management system.
- Implemented a **Load-In-Place** technique to accelerate data initialization, delivering a **7x speed improvement** in startup times.
- **Final Project Achievement**: Refactored a **200,000-particle game system** for a class competition, achieving a **20x speed boost**, marking the best performance in the 16-year history of the course.
  - Removed costly matrix multiplication operations from the game loop, integrating a **SIMD-optimized math library** for real-time performance.
  - Designed a **project-specific memory scheme** to minimize and consolidate memory allocations, enhancing resource efficiency.
  - Applied **Return Value Optimization (RVO)** to eliminate unnecessary temporary constructions, reducing runtime overhead.
  - Transitioned the project from **double** to **float** data types, significantly reducing computational weight and improving processing speed.

Artem Kirienko

(813) 709-9515 – artem.kirienko1999@mail.ru – 5009 N Sheridan Rd, Chicago, IL 60640

Multithreaded Game Audio Engine `[C++/XAudio2]`

- Engineered a real-time multithreaded game audio engine leveraging the **Actor Model** architecture, with **7 independently running threads** handling distinct tasks across a codebase of over **7,000 lines**.
- Harnessed the XAudio2 framework to extract, process, and play raw audio data, implementing extended functionality using XAudio2 Callbacks and Attributes.
- Designed and implemented a **thread communication system** using the **Command** design pattern and a **Circular Queue** data structure to efficiently pass commands between threads.
- Built a custom **Handle Library** utilizing **C++ mutexes** to ensure thread-safe access to shared resources.
- Developed a linked list-based **Manager Library** for efficient asset management, ensuring memory integrity and preventing leaks.
- Created an **Error Handling System** that processes and resolves runtime errors on a dedicated thread, enhancing engine stability and reliability.

Space Invaders Game Remake `[C#/.NET]`

- Implemented the legacy arcade game Space Invaders, employing **10 Object-Oriented Design Patterns**, such as **Singleton**, **Observer**, **Proxy**, **Null Object** etc., to achieve a modular and scalable codebase.
- Constructed the game with **149 classes** across more than **9,000 lines of code** (2,550 lines of executable code), demonstrating advanced software design and development skills.
- Programmed core gameplay mechanics, enabling the game to **draw**, **move**, and **animate sprites**, process keyboard inputs, and handle **sprite collisions** for interactive gameplay.
- Integrated the irrKlang Audio Library to produce dynamic sound effects for alien movement, missile shooting, and background music, enhancing player immersion.
- Implemented an **Object Pooling Method** to minimize dynamic memory allocations, significantly improving game performance and resource efficiency.
- Developed an **Object Collision System** leveraging early-out techniques and **Visitor** and **Composite Design Patterns** to optimize collision detection and eliminate unnecessary computation during gameplay.
- Created a **Font System** using the **Flyweight Design Pattern** to efficiently render the player's score, level labels, and menu options, reducing memory usage while maintaining visual quality.

Other Research Areas `[C++/C#/.NET]`

- Real-Time Collision Detection.
- Real-Time Multithreaded Architecture.
- Applied Algorithms and Data Structures.

WORK EXPERIENCE

C++ Software Engineer for a NASA-funded project at *DePaul University* (Chicago, IL)....**9/2024 – Present**

TECHNICAL SUMMARY

Programming languages: C, C++, HLSL, C#, CUDA C/C++.
Libraries and API: DirectX 11, Unreal Engine 5 API, CUDA, GStreamer, Win32, .Net, XAudio2.
Networking Protocols: TCP, UDP/RTP, IP.
Software and IDEs: Unreal Engine 5, Visual Studio Enterprise.
Software Engineering: UML, Test-driven and Data-driven development.
Version Control: Perforce.
Platforms: Windows.