JOE SCHWARTZ
joseph.robert.schwartz@gmail.com | 414-559-3102

PROFESSIONAL SUMMARY

Senior AI / Gameplay / Full Stack Engineer specializing in AI-driven game development, runtime content generation, gameplay systems, editor tools, and full-stack game technology. Experienced across Unity, Unreal Engine, C#, C++, Python, Java, web frontends, backend services, cloud deployment, and generative AI pipelines. Known for rapidly turning ambiguous ideas into playable systems, reducing production costs, improving designer workflows, and building AI tools that generate game-ready assets, characters, quests, dialogue, and interactive agents.

JOB HISTORY

Senior AI / Gameplay / Full Stack Engineer (Contract) November 2025 - Present
Upheaval AI, Remote, Middleton, Wisconsin

- Crafted gameplay music for a game demo.
- Built an Unreal Engine shader plugin that outlines selectable meshes.
- Added runtime downloading and caching to two Unity mobile games, cutting FTUE wait time by ~60 seconds and significantly improving first-time user experience.
- Generated custom-tailored art assets in minutes, saving thousands of dollars in production time and external asset costs.
- Identified and fixed hard locks in a tutorial created by a previous engineering team.
- Added trial modes to two previously paid-only games, including limited-access systems, feature gating, subscription-directed exploration, and conversion-focused user flows.
- Streamlined UI and user flow to create a cleaner, sleeker, and more intuitive experience.
- Fully shipped two Unity games to both Android and IOS store
- Used Codex to assist with Unreal C++ and Unity C# development.
- Built a React frontend for a banking-style app targeting web, Android, and mobile platforms.
- Implemented and managed an Azure backend with deployed web services, databases, and servers.

Senior AI Game Engineer (Contract) August 2025 - November 2025

Azra Games, Remote, Middleton, Wisconsin

- Built a generative social media manager in Python and HTML within a few days.
    - This system generated story-driven conversations between different characters, scheduled the posts, and published them automatically from each character account.
- Ported an HTML/Python clicker game into a deployed Discord Activity within days.
- Developed a Quest Visualizer tool in Unity using Odin Tools.
    - Built an interactive zone map with clickable quest steps.
    - Used Unity Gizmos to display player navigation paths between quest steps in the editor.
    - Engineered data scraping and aggregation from multiple sources to improve quest visualization accuracy.
- Upgraded the Quest Editor with search, duplication, and batch-creation workflows.
    - Designed a future LLM-driven workflow to automate quest object creation and prefab setup from natural-language quest descriptions.
- Debugged and improved tools using an AI debugging system to benchmark tool performance for other contractors.
- Built a Save Data Editor Tool to accelerate designer iteration and improve productivity.
- Used Claude Code to assist with Python, HTML, Unity, and tooling development.

Senior AI Game Engineer (Contract) August 2025 - September 2025

Running Bull, Remote, Middleton, Wisconsin

- Led a team of two engineers to deliver an MVP XR shooter in Unity.
    - Integrated the Unity game with a Next.js, React, and TypeScript frontend website and a Django backend.
- Developed an AI agent with real-time voice capabilities using OpenAI across both web and Unity environments.
- Built the agent architecture and implemented memory storage in a database.
    - Designed and created the agent's web chat UI to improve user interaction.

Senior AI / Gameplay / Full Stack Engineer June 2024 - August 2025

Upheaval AI, Remote, Monona, Wisconsin

- Built an Unreal plugin and Unity project that connected to a custom C# stack to generate game data, images, 3D models, and animated rigged assets, then imported them directly into game engines.
    - Implemented a pipeline where generated rigged assets auto-retargeted to Unity and Unreal skeletons, allowing generated characters to use engine-native animations and support possession.
    - Example video: https://www.linkedin.com/feed/update/urn:li:activity:7326535328557674497
- Built AI tools in C# MAUI and web technologies, including HTML, JavaScript, and CSS, hosted on Azure and connected to a backend for generating images, 3D models, rigged models, game data, videos, and video-game characters.
- Built LLM-driven NPCs using AI-generated data, including memory, thoughts, dialogue, NPC-to-NPC communication, player interaction, and decision-making.
- Built an interpretive layer that allowed NPCs to create other NPCs, communicate with each other, and control local computer actions.
- Built an API layer to run custom ComfyUI workflows for image, 3D model, and video generation.
- Implemented Unreal Blueprints and C++ code when needed.
- Developed Python API backends, web frontends, and custom Blender automation pipelines.
- Delivered the work as part of a small two-engineer team with Twain Martin, an early World of Warcraft team member, 30+ year industry veteran, and technical mentor.
- Additional examples are provided at the end of the resume.
- Implemented a RAG vector-based knowledge system for agentic memory.
- Worked with vector databases and semantic search technologies, including Pinecone and NVIDIA RAG tooling.
- Used OpenAI vector embeddings.
- Implemented and managed an Azure backend with deployed web services, databases, and servers.
- Developed agent-based AI workflows that autonomously performed tasks, orchestrated services, and supported decision-making processes in game development and gameplay.
- Implemented AI agents that interacted with APIs, databases, and enterprise systems to create runtime assets and store them in databases and CDNs.

- Developed LLM-powered multi-step reasoning workflows for automated asset development.
- Built a Docker container to automate deployment of serverless AI image generation.

Senior Game Engineer (Full Stack) December 2022 - June 2024
Tempo Storm, Remote, Monona, Wisconsin

- Built an AI tool that created game assets from text input and piped them directly into the game.
    - Reduced playable-card prototyping time to seconds, achieving a 99.38% reduction in art and design iteration costs.
    - Generated card art and design data, then packaged the results into Unity-ready playable cards.
    - Replaced a multi-day concept-art, game-data creation, and Unity integration pipeline with a near-instant AI-assisted workflow.
- Built a client-side combat system that used server-provided seeds to prevent cheating.
    - Reduced server costs by 98.57% while increasing available bandwidth and maintaining robust anti-cheat protections.
    - Pre-simulated entire battles on the client in under a second so Unity could focus on fight playback and user experience.
- Built event-driven infrastructure that allowed a shared C# DLL to run core game logic on both server and client.
    - Converted Unity into a visual playback and user input layer while routing game systems through shared DLL code.
    - Extended the same seed-based architecture from combat into other game systems to further reduce server costs.
- Rebuilt the game from the ground up to be event-driven, lean, cost-efficient, fair, and designer-friendly.
- Shipped the Early Access version of the game along with multiple updates.

C++ Game Engineer August 2022 - December 2022

Infinite Games, Remote, Monona, Wisconsin

- Wrote C++ UE4 and UE5 plugins that made HTTP calls to backend services for features such as login and item retrieval.
- Integrated Unreal Engine plugins with the account system.
- Worked with data-driven systems while prioritizing speed and efficiency.
- Added features, debugged, and maintained C++ code for a MySQL server backend infrastructure and database.
- Built a development automation tool with a C# Windows Forms user interface to generate repetitive code through a pipeline.
- Built a C# front-end admin tool using the Blazor framework.
- Developed a logging system for a .NET framework using C#.
- Used RESTful APIs to build a web manager with the .NET Swagger framework.
- Worked with Azure to deploy builds, make HTTPS calls, and publish from Visual Studio.
- Wrote batch scripts to automate compiling, running code, and generating Visual Studio files for Unreal Engine.
- Collaborated with engineers to choose the best language, tool, and architecture for each problem.

System Engineer April 2017 - August 2022

UW Computer Aided Engineering, Madison, Wisconsin

- Solved a wide variety of technical issues using strong troubleshooting and problem-solving skills.
- Created and deployed packages for Windows-based applications using App-V, SCCM, and GPOs.
- Upgraded and maintained Windows-based applications and servers.
- Monitored and troubleshot new and existing software installations.
- Configured Windows application packages for deployment across a lab environment of 700+ computers.
- Automated server builds using templates in VMware vSphere.
- Automated software installations through batch files and PowerShell scripts.
- Deployed machine images through SCCM Task Sequences.
- Documented work through knowledge base articles and ticketing systems.
- Worked directly with customers to customize installation requests and resolve issues.
- Mentored and trained new hires.

TECHNICAL SKILLS

AI Skills and Tools:

- OpenAI (ChatGPT, DALL-E 3, Whisper, Text To Voice) — 4 years
- Stable Diffusion (Image Generation) — 4 years
- GitHub Copilot — 4 years
- Agent-Based AI Workflows — 3 years
- ComfyUI (Image Generation) — 2 years
- Eleven Labs (Text To Voice/Sound Effects Generation) — 2 years
- Various Local LLMs (LM Studio, Oogabooga) — 2 years
- Video Generation (Wan 2.2, LTX 2.3, Seedance 2.0) — 2 years
- Suno AI (Music Generation) — 1 year
- Cursor (Code Generation) — 1 year
- RAG (Retrieval Augmented Generation) — 1 year
- Claude Code — 1 year
- Codex — 1 year
- Play HT (Text To Voice Generation) — 4 months

Programming Languages:

- C# — 5 years
- C++ — 4 years
- Python — 4 years
- Java — 2 years
- C — 2 years
- Node.js — 2 years
- React — 1 year
- Assembly — 1 year

Tools:

- Googling — 20 years
- Slack — 7 years
- Object-Oriented Programming (OOP) — 5 years
- Unity — 5 years
- Git — 5 years
- REST APIs — 5 years
- CI/CD Pipelines — 5 years
- GitHub — 4 years
- Azure — 4 years
- Agile — 4 years
- Visual Studio — 3 years
- Data Oriented Programming (DOP) — 3 years
- Unreal Engine 5 — 2 years
- Bash — 2 years
- Linux — 2 years
- Jira — 2 years

- Unreal Engine 4 — 1 year
- Unit Testing — 1 year
- Amazon Web Services (AWS) — 1 year

EDUCATION

University of Wisconsin-Madison — Madison, Wisconsin
B.S. in Computer Science, Physics, and Theatre (Acting Option)
Triple major — September 2014 - May 2021

Yonsei University — Seoul, South Korea
Studied abroad — September 2016–December 2016

PROJECTS

Unreal Engine MMO — **2026**

- Ported Unreal Engine's Game Animation Sample Project motion-matching character to C++.
- Built a fast and efficient C++ MMO server backend.
- Connected replicated multiplayer systems to the C++ motion-matching character.
- Conducted multiplayer testing with remote users across the state using a custom C++ server and Unreal client.
- Developed expansive procedural open-world terrain generation in Unreal Engine for both editor and runtime use.
- Generated large-scale open-world landscape including terrain features like volcanoes, mountains, canyons, fissures, rivers, plains, and hills.
- Built configurable generation controls that allow terrain settings, world-shape parameters, and procedural feature behavior to be tuned.
- Built a generated-world preview map tool that displays the world layout, shows the player's current location, and supports right-click teleportation to selected map locations.
- Added an LLM chat interface that allows procedural generation settings to be adjusted through natural-language prompts.
- Built a 3D asset generation tool that creates 2D art assets from text prompts, converts them into 3D assets, and rigs humanoid assets when applicable.
- Automated importing of generated 3D assets into Unreal Engine and enabled spawning both at runtime and in the editor.
- Integrated generated assets into the procedural generation pipeline so they can be placed as procedurally spawned world assets.

Smaller Voxel Game **2026**

- Built a custom voxel game engine with procedural generation, finer voxel resolution, C++, and Vulkan.

Voxel Game **2026**

- Built a custom-engine Minecraft-style project with procedural map generation in C++ and Vulkan.
- Generated chunks at runtime and animated them falling down.
- Implemented weather systems and GLB rendering.
- Integrated LLM-driven runtime procedural generation using prompts such as “Generate me a dense forest with mountains” or “Generate a barren desert.”

3D Raptor Creation, Rigging and Animation

- Generated a 3D raptor, rigged it, and animated walk, run, and idle animations in Blender.

Upheaval AI Work **2024 - 2025**

Monona, Wisconsin

- Built runtime generation systems for functioning NPCs and agents.
- Built these systems as part of a two-engineer team.
- Recorded, edited, and assembled the videos, and generated the audio and music used in them.
- Example: putting toys away
  - *https://www.linkedin.com/feed/update/urn:li:activity:7327964508977557505*
- Example: making a playable Orc at runtime
  - https://www.linkedin.com/feed/update/urn:li:activity:7326535328557674497
- Examples: generating assets in a tool and importing at runtime
  - Werewolf *https://www.linkedin.com/feed/update/urn:li:activity:7244074085423456256*
  - Dragon Egg *https://www.linkedin.com/feed/update/urn:li:activity:7243009835456233472/*
- Example of generating assets at runtime in game
  - https://www.linkedin.com/posts/twainm_agentic-character-creation-imagine-creating-ugcPost-7243003061193900033-tXdT?utm_source=share&utm_medium=member_desktop&rcm=ACoAACsG-B8BE4NltI7FIKJy1cw-TXRVcc9dsDs

AI-Generated Music and Interactive Audio **2025 - 2026**
Monona, Wisconsin

- Created and published AI-generated radio-style songs using generative music tools.
- Demonstrated prompt-driven composition, rapid audio iteration, creative direction, and AI-assisted production workflows.
- Portfolio: https://open.spotify.com/artist/1KN97H12k1tUgy1tSMaa2b?si=-1TQa64LRBOJTGjfFkOyXQ

Choose Your Own Adventure App **2023**
Monona, Wisconsin

- Built a Unity app in C# using OpenAI text generation.
- Created a novel-like interactive experience where users choose from three options or enter custom text to navigate the story.
- Presented eight starting genres plus a custom genre option.