

# Gavin Schmidt

Gavinschmidt47@yahoo.com | gavinschmidt47.github.io | linkedin.com/in/gavinschmidt47 | github.com/gavinschmidt47

## Education

### University of Central Florida

Bachelor of Art in Digital Media – Game Design | Minor in Computer Science

Orlando, FL

August 2021 – May 2026

## Experience

### Lead Gameplay and Systems Engineer

August 2025 – Present

Head or Tail Studios

Orlando, FL

- Designed **10+ core gameplay/systems modules** across **2 projects**, defining ~95% of the technical design surface area (data flow, extensibility points, and implementation standards)
- Document and implement **60+ scripts** containing **2,500+ lines of code** with **automated tests/tools** for 65% of gameplay code, **reducing** designer implementation time by **25%**
- Planned sprint milestones for **3 engineers** and unblocked **5 high-priority issues/week**
- Set coding standards and conducted reviews for a **3-person** engineering team, improving consistency and reducing regressions

### Sole Programmer and Mechanics Lead

August 2025 – Present

Mystical Die Studios

Orlando, FL

- Implemented **3 gameplay systems** (combat, grid-movement, dialogue) using **C++** and **blueprints**
- Created detailed documentation for **20+ scripts and blueprints**, enabling designers to build **10+ features** without engineering support
- Ran **weekly** training sessions, supported **7 designers/artists**, resolved **5+ support requests/week** with a **3 hr** average turnaround
- Exposed systems to designers via Blueprint-extensible APIs, cutting engineering involvement per new feature from **full implementation** to **targeted review-only (<35%)**
- Ran extensive **profiling** and **debugging** on all new features in weekly builds

### Front of Store Team Member

August 2025 – Present

Gamers Inc.

Orlando, FL

- Provided product guidance across console/PC accessories and trade-ins
- Became familiar with gaming hardware and the effects that has on video game production and maintenance
- Handled **30-40** customer interactions/shift
- Performed inventory counts for **500+ SKUs** with **>99% accuracy**

### Assistant Scholar

May 2025 – July 2025

Limbitless Solutions LLC

Orlando, FL

- Built and integrated **player-facing mechanics** in Unreal as part of a **25+ person** interdisciplinary team; delivered changes on tight playtest deadlines
- Translated playtest feedback from **Bionic Kids** into concrete gameplay changes (controls clarity, pacing, feedback/UI), improving moment-to-moment readability and engagement
- Shipped **weekly** iteration builds and partnered with design to rapidly scope, implement, and validate changes in short loops

## Projects

### Solar Scavengers | Unity 6, VS Code, Musescore Studio 4, Git

January 2026

- Created for **FIEA Game Jam 2026**, designed for **PC**, made in **Unity 6**, coded in **VS Code**
- Core mechanics center around **procedural chunk-based generation**, **resource management**, and **momentum-based controls**
- The generation is designed around an adaptive **chunk-loading** and **object pooling** system
- Difficulty, spawn rates, music, and in-game lighting are **responsive** to in-game progress

### Fearosis | Unity 6, VS Code, Android APK, Musescore Studio 4, Git

September 2025 – Present

- Created at **Head or Tail Studios**, designed for **mobile Android**, made in **Unity 6**, coded in **VS Code**
- Core mechanics center around **stat-based simulations**, **adaptive upgrade trees**, **object pooling**, and **DataAssets/DataTables**
- Character movement was achieved through an **A\* Pathfinding algorithm**
- Player data is saved and protected using an in-depth **binary serialization** system

### Reconnection | Unreal 5, Visual Studio, Musescore Studio 4, Git

September 2025 – Present

- Created at **Mystical Die Studios**, designed for **PC(Enhanced Input)**, made in **Unreal 5**, coded in **Visual Studio**
- Core mechanics center around **UE5 Gameplay Framework**, **turn-based combat**, **dialogue-responsive buff system**, and **enemy AI Behavior Trees**
- Combat is designed using Dungeons & Dragons' **roll-to-hit** and **roll-for-damage** systems
- Player movement uses an intuitive compliment of **Navmesh** and simple **2D grid mathematics**