Timothy Trimble

41674 Greenbriar Ln, Plymouth, MI 48170 | (801) 837-0433

tim.d.trimble@gmail.com | www.trimbletim.com

Summary

Highly skilled Gameplay Programmer with extensive experience in creating immersive gaming experiences and data-driven gameplay systems. Proven expertise in systems architecture, Entity Component Systems (ECS), and parallel programming for multiplayer strategy games. Adept at bridging engine-level systems with gameplay mechanics, empowering design teams to craft compelling content through robust technical frameworks. Background includes AI implementation and simulation programming.

Education

Master of Science in Entertainment Arts & Engineering
University of Utah, Salt Lake City, Utah | August 2021 – May 2023

Bachelor of Science in Computer Science – Entertainment Arts & Engineering
University of Utah, Salt Lake City, Utah | August 2015 – December 2019

Technical Skills

- **Programming Languages:** C++, C#, Lua, C, Python, Swift
- **Game Engines:** Unity, Unreal Engine, MonoGame, Godot, OpenFrameworks
- **Programming Paradigms & Architecture:** Object-Oriented Programming (OOP), Entity Component System (ECS), Data-Driven Design, Parallel Programming
- **Design Patterns:** Observer, Factory, State, Command, Singleton, Service Locator, Component, Flyweight, Prototype, Subclass Sandbox, Type Object
- **AI-Assisted Development:** GitHub Copilot, Claude Code (Agentic Coding), AI Pair Programming, Prompt Engineering for Code Generation
- **Gameplay Systems:** Player Movement, Combat Systems, Level Design, Animation Integration, UI Systems
- **Version Control:** Git, Perforce
- **Project Management:** GitLab, GitHub, Jira, Bitbucket, Trello
- **Development Tools:** Visual Studio, Visual Studio Code, Unreal Engine Profiler
- **AI Implementation Experience:** Behavior Trees, Finite State Machines, Pathfinding (A*, Dijkstra), Goal-Oriented Action Planning

Professional Experience

Game Developer
Stardock Entertainment | July 2024 – Present

• Architected and implemented data-driven gameplay systems enabling designers to rapidly prototype and iterate on gameplay mechanics and content without engineering dependencies

• Developed Lua-driven scripting systems empowering designers to create and modify gameplay logic without C++ code changes

• Engineered robust integration layers connecting core engine systems (animation, UI, physics) to gameplay logic, ensuring seamless communication across system boundaries

• Developed scalable ECS (Entity Component System) architecture for multiplayer strategy games, optimizing for parallel processing and network synchronization

• Implemented high-performance parallel programming solutions to maintain responsiveness in complex strategy game simulations with thousands of entities

• Leveraged AI-assisted development tools (Claude Code, GitHub Copilot) to accelerate feature implementation and refactoring workflows

• Collaborated closely with design teams to translate gameplay vision into flexible, maintainable technical frameworks

• Optimized critical gameplay systems for performance, achieving target frame rates in intensive multiplayer scenarios

Simulation Software Programmer

Hill Air Force Base, Hill AFB, Utah | January 2020 – June 2024

• Collaborated with stakeholders to implement high-fidelity simulated models of F-16 Falcon aircraft systems

• Designed and optimized A* pathfinding algorithm with custom heuristics for large-scale graph navigation

• Led scrum team as technical lead, facilitating cross-functional workflows and sprint planning

• Maintained large-scale codebase integrating modern and legacy systems across 4+ years of active development

• Ensured cross-platform compatibility (Linux, Windows, proprietary systems) through rigorous testing and abstraction layers

• Developed and maintained launcher software managing multiple simulation programs and dependencies

• Integrated physics engine for realistic flight dynamics and environmental simulation

• Executed rapid prototyping initiatives, delivering proof-of-concept features under tight deadlines

• Conducted thorough code reviews, providing technical mentorship and ensuring code quality standards

Computer Science Teaching Assistant

University of Utah | January 2019 – December 2019

• Instructed undergraduate students in C++ programming fundamentals and advanced concepts

• Taught memory management techniques including pointers, dynamic allocation, and RAII principles

• Developed and delivered lesson plans for Software Practice II Lab, adapting teaching methods to diverse learning styles

• Provided one-on-one mentorship and debugging assistance to students

Published Games

Slime Knight - Gameplay Programmer

Unreal Engine 5 | Published May 2023

- Engineered universal behavior tree architecture supporting diverse enemy AI with modular attack systems
- Implemented comprehensive save/load system preserving game state and player progression data
- Created interface system for identifying and validating grabbable objects across level environments
- Managed Perforce version control infrastructure, providing technical training to art team members
- Diagnosed and resolved critical performance bottlenecks through profiling and code optimization
- Optimized spline mesh rendering pipeline, achieving 100% performance improvement (30 to 60 fps)

Pathos – UI Programmer

Unity | Published May 2015

- Implemented complete UI system and integrated all UI art assets with gameplay systems
- Integrated FMod audio middleware, developing custom wrapper class for streamlined audio implementation

AI in Games

Flow Field AI Pathfinding Plugin - Solo Developer

Unreal Engine 5 | In Progress

- Developing production-ready flow field pathfinding system for large-scale crowd simulation
- Implementing parallel pathfinding algorithms to support thousands of simultaneous agents
- Optimizing graph representation using hierarchical pathfinding and A* portal navigation

AI Strategy Independent Study - Solo Developer

Unity | December 2022

- Researched strategy AI approaches for partially observable, non-deterministic environments
- Analyzed environmental properties to optimize agent perception and decision-making frameworks
- Implemented strategy AI for capture-the-flag gameplay with goal reasoning and delegation systems
- Designed tools for describing AI objectives using GFI (Goals, Feedback, Interpretation) and MDA frameworks

AI Engine Framework - Solo Developer

OpenFrameworks | April 2022

- Built comprehensive AI framework implementing movement algorithms (Seek, Wander, Flocking)
- Developed pathfinding systems (Dijkstra's, A*) with customizable heuristic design
- Implemented decision-making architectures including Decision Trees, Behavior Trees, and GOAP

NPC Interactive Dialogue – AI Programmer

Unity | April 2022

- Integrated Tracery language-generation framework for dynamic NPC dialogue
- Designed relationship system driven by emergent conversational interactions

Gameplay Engineering Portfolio

The following projects were completed in weekly sprints as part of graduate-level coursework, each demonstrating specific engineering patterns and gameplay mechanics.

Kingdom Builder 3D - Solo Developer | Unreal Engine 5 | December 2022

- Transformed 2D gameplay to 3D while preserving core design pillars
- Implemented observer pattern for type-agnostic resource management system

Dungeon Walker - Solo Developer | Unreal Engine 5 | November 2022

- Generated procedural dungeons using binary space partitioning subdivision method
- Created musical puzzle mechanics enhancing exploration-based gameplay

Parkour Ranger - Solo Developer | Unreal Engine 5 | October 2022

- Developed climbing mechanics responsive to surface tangent angles
- Designed crafting system that dynamically modifies player movement capabilities

Castle Builder - Solo Developer | Unreal Engine 5 | October 2022

- Built UI-heavy gameplay following established UI/UX design frameworks
- Implemented command pattern enabling undo/redo functionality for building mechanics

Experimental Gameplay Portfolio

One to two-week rapid prototypes exploring unconventional gameplay mechanics and experimental design concepts.

Scoville Chopper - Solo Developer | Unreal Engine 5 | April 2023
- Utilized post-processing effects to create dynamic vision-blurring mechanics
- Integrated mesh slicing as core gameplay interaction

Uncertainty Principle - Solo Developer | Unreal Engine 5 | April 2023
- Implemented camera-based visibility system dynamically altering unseen geometry
- Applied UX principles to cultivate atmospheric tension and unease

After Wars - Solo Developer | Unreal Engine 5 | April 2023
- Designed strategic gameplay where suboptimal tactics yield superior outcomes
- Implemented turn-based system within Unreal Engine's real-time framework

Line World Shifter - Solo Developer | Unreal Engine 5 | March 2023
- Generated procedural level layouts with looping spatial topology
- Developed gravity manipulation mechanics enabling 360-degree traversal

Die Attack - Solo Developer | Unreal Engine 5 | February 2023
- Created time-limited AI movement system with physics-driven gameplay elements

Like Clockwork - Solo Developer | Unreal Engine 5 | January 2023
- Designed rotation-matching puzzle mechanics with state-dependent behavior

Additional Projects

Battle Bard - Solo Developer | Unreal Engine 5 | December 2022
- Integrated Arduino hardware interface with Unity Engine for custom controller input
- Designed and fabricated inclusive wearable drum controller for rhythm gameplay

Bouncer Mage - Solo Developer | Unreal Engine 5 | April 2022
- Analyzed and replicated Hollow Knight's bouncing mechanics in third-person perspective
- Implemented procedural upper-body animation responsive to player input

Kitty Kitty Bang Bang – Gameplay Programmer | MonoGame | August 2021
- Built rhythm-based metronome system evaluating player timing accuracy
- Developed level creation tools for rapid difficulty prototyping and testing

Personal Projects

Quest Board – Programmer/Designer | Godot 4 | GMTK Game Jam | July 2023
- Designed, implemented, and published complete game within 48-hour jam constraints
- Developed AI decision-making system for NPC behavior and quest assignment

- Created dynamic UI framework handling non-linear gameplay progression

Custom Game Engine - Solo Developer | SFML | In Progress

- Building custom 2D game engine using SFML for graphics and windowing
- Architected component-based entity system with serialization support
- Developed custom JSON parser for runtime object and level initialization
- Implemented Separating Axis Theorem collision detection system
- Applied advanced C++ techniques including meta-programming and template specialization

Personal Interests

- Passionate about exploring emerging trends in AI and game development through personal projects
- Primary creative inspiration: *Hyper Light Drifter*
- Classic gaming enthusiast, especially *The Legend of Zelda: A Link to the Past* (SNES)
- Active participant in game jams, fostering creativity and collaboration within the gaming community
- Game Master for tabletop RPG group, experienced in board games and card games
- Multi-instrumentalist: piano, guitar, ukulele, and clarinet
- Eagle Scout (earned 2013)
- Outdoor enthusiast enjoying hiking, camping, and swimming with family and pets