Sam Stålhandske

samstalhandske.com

samstalhandske@hotmail.com

github.com/samstalhandske

+4673 980 91 66

1999-12-21

Summary

Outgoing and friendly Swedish intermediate generalist programmer who is eager to work on interesting projects and to collaborate with my colleagues. I'm a strong programmer with a drive to become an important part of whatever project I'm on.

Experience with

C++, C#, Odin, Lua, Python, Networking, ECS, Unity (+ DOTS), CCC, Nintendo Switch, Xbox, MonoGame, Custom frameworks/engines, Profiling, Game AI, OpenGL, DirectX 11, Raylib, SDL, WWise, FMOD, PlasticSCM, Git, SVN.

Professional Experience

Aug 2020 - Jan 2025: **Thunderful Development**, Malmö

Worked on two cross-platform titles for PC (Windows/macOS), Xbox, and Nintendo Switch. Worked in C#, Unity (+ DOTS). Part of two multidisciplinary teams (15–30 people), supporting both "engine"/framework development, gameplay systems as well as tools for programmers and other disciplines.

Designed and implemented:

- A domain-specific scripting system for level designers (dialogues, triggers, quest logic, set up entities)
- A custom AABB collision system with no external physics engine

- Deterministic online multiplayer architecture:
    - Fixed timestep simulation
    - Client-side prediction and rollback
    - Replay tool for debugging and presentations, including desync detection
    - Steam integration
    - Dynamic audio system integration (FMOD/Wwise) with real-time RTPC and spatial audio triggers
    - Built and maintained modular UI frameworks (menu systems, save-slot selectors, settings panels, gamepad support)
- Procedural level generation system

Owned and implemented key gameplay features:

- Core player mechanics (movement, interaction, feedback, traversal)
- Health/damage/death components
- Inventory and shop systems
- Skill trees and upgrade systems
- Interactive world-map and minimap components
- Leaderboard and progression tracking
- Optimized performance across systems:
    - Created automated profiling camera spline to capture performance data over time
    - Reduced garbage collection and CPU spikes
    - Worked with technical and graphical artists on LODs, shader optimizations, and mesh tuning

Acted as right-hand to the lead programmer:

- Mentored junior developers
- Coordinated planning sessions (Hack’n’Plan, Slack)
- Upgraded Unity and package dependencies
- Managed source control with PlasticSCM

Education

Sep 2018 - Sep 2020: **The Game Assembly,** Malmö
Built a game engine from scratch in C++ and developed eight games during the program— four with a permanent group, four in rotating teams. I gained a deep understanding of core concepts like memory management, stack vs heap, pointers and so on.

I spent a lot of my time on the engine/game architecture, together with my team, as well as on the differerent games' state machines, player/enemy mechanics, importing/exporting data, input, UI systems, camera work and general gameplay logic. I also collaborated very closely with the other disciplines to try to make their visions for the games come true.

We used TortoiseSVN for version control; I helped onboard others as well as resolve merge conflicts.

My final exam project was an intelligent third-person camera that dynamically avoided obstructions and reframed contextually.

Some of the features in our engine:

- Entity–Component System (ECS)
- Mesh loading and rendering via DirectX 11
- Serialization/deserialization with JSON
- Input system with action-binding (supporting rebinding)
- 2D/3D animation systems with event-triggering
- Basic networking implementation
- Multi-threading
- UI rendering system

Recent Personal Projects

3D Game Engine:

- Written in Odin (low-level C-like language with manual memory management)
- Features ECS architecture, OpenGL renderer (Vulkan planned), SDL2 backend for input and window creation
- Hot-reloading assets, shader/material system, shadow rendering with PCF, render-passes, render-commands
- Level-editor package
- Fixed tick simulation with adaptive vsync rendering
- One of my long-term goals is to build an online multiplayer arena shooter with rollback networking

2D Framework:

- Built in Odin using Raylib + Emscripten (web export)
- Sprite animations, AABB & circle collision, and immediate-mode UI

Online Tile-Laying Strategy Game:

- Turn-based, deterministic netcode
- Flexible action system with undo/redo, event hooks for animations/sounds

GameBoy Emulator:

- Developed on top of custom 2D framework
- Emulates CPU, memory, and PPU with scanline rendering
- Passes many test ROMs but not fully compliant yet

Digital Audio Workstation:

- Built in Odin with Dear ImGUI frontend
- Hot-reloadable Lua-based plugin scripting
- Designed for both modular sound design and structured music composition