Kai Schneider Junior Engine Programmer

Skövde, Sweden · kai.schneider@gmail.com · +46 70 555 0142 · linkedin.com/in/kaischneider

PROFILE SUMMARY

- **Junior Engine Programmer** with **3 years** on a **factory-building game** in **Unreal Engine 5**, working on **game-thread performance, save memory and editor tools** for the PC and console versions.
- Writes **C++20** and **Python**, profiles with **Unreal Insights** and **Superluminal**, builds tools in **Slate**, and works in **Perforce** with **Unreal Build Tool** and **Jenkins**.
- Strongest in **late-game performance**, **memory leaks in saves** and **asset validation**, after chasing slowdowns that only showed up in player saves with huge factories and thousands of conveyor belts.
- Works with **Gameplay**, **Tech Art** and the **console team**, joins the weekly performance review, and pairs with QA on crash repros from player bug reports.
- Wrote the studio's **Unreal Insights quick-start guide** for new programmers and helped **2 interns** land their first engine changes.

TECHNICAL SKILLS

Programming Languages:	C++20, C++17, Python, C#
Game Engines:	Unreal Engine 5, Unity (DOTS)
Low-Level Systems:	Data-oriented design, multithreading, Unreal memory tags, SIMD basics
Profilers & Debuggers:	Unreal Insights, Superluminal, Visual Studio, PIX
Platforms & SDKs:	Windows, PS5, Xbox Series X|S, Steam
Build & Version Control:	Perforce, Unreal Build Tool, Jenkins, Git
Editor & Pipeline Tools:	Unreal Editor (Slate), Python editor scripting, Dear ImGui

EDUCATION

University of Skövde — **B.Sc. in Computer Game Development (Programming)** — Skövde, Sweden · Aug 2020 to Jun 2023

WORK EXPERIENCE

Coffee Stain Studios — Junior Engine Programmer — Skövde, Sweden · Aug 2023 to Present

- Built an in-editor checker in **Slate** and **Python** that catches building meshes with missing collision or LODs before check-in, stopping **300+** broken assets in its first 4 months.
- Profiled late-game player saves in **Unreal Insights** and moved conveyor belt item updates into one batched, cache-friendly loop, cutting game-thread time by **2.1 ms** on the heaviest test save.
- Traced a save-system leak in object serialization with Unreal's memory tags, which stopped long sessions from growing by about **1.4 GB** over 6 hours of play.
- Split the starting world's level streaming into smaller cells with soft asset references, and brought the first load on a mid-range PC from 48 s to **31 s**.
- Fixed **27** certification issues in save data, suspend and resume, and controller handling for the November 2025 **PS5** and **Xbox Series X|S** launch, as the junior on a 4-person console team.

Stunlock Studios — Engine Programming Intern — Skövde, Sweden · Jun 2022 to Dec 2022

- Rewrote a projectile collision check as a **Burst**-compiled **Unity** job across worker threads, cutting its CPU cost in 40-player server tests from 3.2 ms to **0.9 ms**.
- Added a gate and door state component and system to the game's **ECS** code, with save support, which **3** gameplay programmers reused for later content.
- Set up a nightly **Jenkins** job running **120** automated play-mode tests on the server build, so regressions showed up the morning after they landed.