Jeremy Kleve

Software Engineer

SUMMARY

Senior programmer with 4+ years of business level programming experience and 7+ years as a hobby game developer. I pride myself in taking the initiative and solving problems early on, working on projects from the design phase all the way through to delivery and beyond, providing exemplary support to clients and teammates, and continuously learning new skills. Programming and game development is much more than just a career for me, it is a passion.

jkleve1986@gmail.com
(626) 543-4239
Fullerton, CA
linkedin.com/in/jeremykleve
jeremykleve.github.io

SKILLS

- **Languages:** C/C++, C#, Javascript, Lua, HTML / CSS
- **Frameworks:** .Net Core, ASP.NET, ASP.NET Core, Blazor, gRPC
- **Libraries/APIs:** DirectX, OpenGL, Dear ImGui, SQLite
- **Software:** Visual Studio 2019, Visual Studio Code, Eclipse, Git, Blender, Unix/Linux
- **Game Engines:** Unity, Unreal 4, Godot
- **Modding:** Assetto Corsa and BeamNG
- **Non-technical:** Project management, teamwork, self-motivated learning

EDUCATION

California Polytechnic University
Pomona, CA
B.S. Computer Science
2016 - 2018

Mt San Antonio College
Walnut, CA
transferred
2014 - 2016

WORK EXPERIENCE

Game Programmer // **Sigma Integrale**, Pomona, CA
May 2018 - PRESENT

- Participated in a small team of software, electrical and mechanical engineers developing vehicle simulator systems for high profile clients including Dodge, Volkswagen, Kia, Alfa Romeo, Polaris, and Waste Management.
- Developed the DK motion system from design to production, providing motion haptic feedback for professional racing simulators. Highlights of the system include:
    - PC software complete with network layers, tools and UI capable of extracting and processing game physics telemetry in real time..
    - Embedded system with dual microcontroller PRUs running a real-time OS (TI-RTOS) working in unison to calculate and output motion commands.
    - In-house manufactured motion actuators driven by Teknic electric servo motors providing powerful and responsive motion feedback for simulators.
    - Total system latency from game output to motion averages from 9-72 ms depending on variable data rates and levels of motion smoothing .
- Created and maintained server/client software to host websites, collect data, manage event participants, and remotely controlled software through webUI.
- Performed in-game modifications for various simulator software including 3D modeling of vehicles and objects, tuning of physics engine, and scripting of game logic using Javascript, Python, and Lua.
- Separated business logic and views through Model-View-ViewModel software design principles.
- Created inter-process-communication between applications using gRPC and shared memory to optimize efficiency and debugging.