JAVIER RODRIGUEZ VALENTIN

Gameplay Programmer - C++ / Unreal Engine

Stockholm, Sweden · jrvalentin@valenvier.com · Portfolio · GitHub · LinkedIn · itch.io

Game Programming student at Forsbergs Skola (Stockholm), focused on gameplay and multiplayer systems in C++ and Unreal Engine 5. Shipped a full online co-op multiplayer game and collaborated with an external studio on a networked prototype built on Epic Online Services. Full project videos and source at valenvier.com.

TECHNICAL SKILLS

Languages: C++, C#, JavaScript

Engines & Tools: Unreal Engine 5, Unity, Raylib, Git

Focus areas: Multiplayer & networking (EOS, replication, session management, host migration), AI (Behaviour Trees, perception), algorithms & data structures

FEATURED GAMES

Networked Coop Interactions - *source on request - NDA*

Team project · Unreal Engine 5.6 / C++ · built in collaboration with an external game studio

- Implemented peer-to-peer connection via Epic Online Services: device ID login, session hosting and joining.
- Built host migration - a surviving player takes over if the host disconnects - and game-state persistence (world, inventory, player position) across level reloads.

Hollow Maze: A Cooperative Escape - code *source on request* · PLAY ↗

Team project · Unreal Engine 5.7 / C++

- Built the first-person player character and the full multiplayer implementation for a two-player online co-op escape room.
- Implemented the session system (create / find / join), replicated character movement and state, a server-authoritative interaction and 9-slot inventory system, and checkpoint respawn.
- Tested across same-network and cross-network play using a VPN.

Abuelo Rodrigo: Furious Bingo - CODE ↗ · PLAY ↗

Solo project · C++ / Raylib

- Built a top-down survival shooter with 7 weapons with distinct firing styles and 5 enemy types, including a 3-phase boss.
- Structured enemy AI, weapons and game state around 8 design patterns (State, Object Pool, Strategy, Factory, Observer, Singleton, Command, Component).
- Designed a bingo-card win condition and a weapon-confiscation mechanic tied to player HP thresholds.

USS Calliope - code *source on request* · PLAY ↗

Team project · Unity v6000.2.9

- Built the enemy AI system: an FSM state controller with a vision-cone/raycast perception system and three infection phases with escalating behaviour.
- Implemented enemy call/coordination and ranged projectile systems, using ScriptableObjects for data-driven attacks, infection phases and patrol zones.
- Wrote the game's main story and lore.

WORK EXPERIENCE

ServiceNow Consultant & Developer - Freelance · 2021 – Present

Delivered enterprise software implementations and integrations for professional clients while studying game programming part-time.

EDUCATION

Game Programming - Forsbergs Skola, Stockholm · 2024 – 2027

Master's Degree in Full-Stack Development - UNIR, Spain · 2023

↳ *full-stack web development: React, Angular, Node.js, Next.js, MySQL/MongoDB*