ARNOLD (WEI-NO) WU

Unity Gameplay Programmer | Game Systems Design
Redmond, WA · (425) 979-6289 · arnoldwu2012@gmail.com
weinowu.com · github.com/WEI-NO · linkedin.com/in/wei-no-wu

PROFILE

Gameplay programmer with a B.S. in Computer Science and Game Design from DigiPen. Builds data-driven mechanics and designer-facing tools in Unity and C#, with additional experience in Unreal Engine, UEFN/Verse, and custom C++ engine development. Experienced in debugging, playtest-driven iteration, and working on cross-disciplinary game teams.

SKILLS

Unity & C# Unity 2D/3D, C#, Physics 2D (Rigidbody2D, hinge joints, colliders, raycasts), Particle System, Prefabs, ScriptableObjects, Unity UI, Addressables, Tilemap, Animator, mobile touch input, Unity Profiler

Gameplay Systems Finite state machines, A* pathfinding, procedural generation, grid & spatial partitioning, weighted loot tables, progression & economy, save/load persistence, event-driven architecture

Game Design Systems and technical design, reward-loop and gacha odds tuning, progression pacing, designer tooling, playtest-driven iteration, QA and bug triage

Development Practices Agile development, code reviews, debugging, playtesting, QA, bug triage

Other Engines Unreal Engine (C++, Blueprints, UMG), UEFN / Verse, custom C++ engine, Roblox Studio

Languages & Tools C#, C++, Python, Lua, Verse, Java · Git, GitHub, Perforce, Visual Studio, Aseprite

UNITY PROJECTS

Clawmi - Cozy Claw-Machine Idle Game - Unity, C# · Mobile

Gameplay Programmer & Systems Designer · 2-person team · github.com/WEI-NO/ClawMachineIdle · weinowu.com/clawmi

- **Diagnosed** that Transform-driven claw movement bypassed Unity's physics solver so grabs never registered collisions; rebuilt the rig on Rigidbody motion and hinge joints, tuning angle limits and motor strength for natural dangle and believable grab failure.
- **Designed** a three-state claw controller (Relax / Expand / Grab) driving joint motor parameters, turning an unpredictable physics rig into a tunable, designer-readable grab sequence.
- **Built** an isometric building and decoration system on a dictionary-keyed grid for constant-time tile lookup, with placement validation, edge detection, rotation, and storage.
- **Designed** data-driven claw machines and gacha eggs - each theme takes a prize list with weighted odds, so new content ships as data.

D.O.P.C - Pixel-Art RPG Platformer with Procedural Maps - Unity, C#, Aseprite

Gameplay Programmer · github.com/WEI-NO/DOPC · weinowu.com/d-o-p-c

- **Built** a procedural map generation system assembling rooms/chunks into playable runs, with bitwise validation rules guaranteeing every layout is completable and traversable.
- **Implemented** the 2D platformer character controller and game feel: coyote time, variable jump height, combat particle system with persistent and impact layer, and screen shake.
- **Built** RPG progression systems including stats, inventory, leveling, enemy scaling, elemental type effectiveness/resistance and an enemy behavior state machine.

EXPERIENCE

Gameplay & UI Programmer - TideTurners (Indie Studio) | UEFN / Verse | Apr 2026 - May 2026

- **Shipped** gameplay and UI features into an existing Verse codebase, inside an established architecture and team workflow rather than greenfield code.
- **Built** UMG/WBP interface flows bound to live gameplay state through event bindings, widget references, and runtime state updates.
- **Integrated** class-based systems into existing spawn, respawn, team-state, and ability flows; turned weekly playtest notes into fixes via pair programming and live debugging.

Gameplay Programmer & QA Lead - Neon Vengeance: Atavysm | Aug 2024 - Apr 2025

3D hack-and-slash, Unreal Engine · 22-person cross-disciplinary team

- **Implemented** core player character systems in Unreal - stamina with sprint drain and regen, sprint-to-walk transition handling, and the Spark Charge ultimate gauge that gates the character's ultimate ability.
- **Implemented** the player-facing layer where design intent becomes readable: HUD, menu flow, tutorial support, settings, and combat feedback.
- **Collaborated** in an Agile team environment.
- **Led QA** — owned the bug pipeline, triaged reports across disciplines, verified fixes, kept weekly builds playtest-stable, and turned findings into concrete tuning and design feedback for feature owners.

Programmer - Astro Odyssey | Custom C++ Engine, 2D Action | Sept 2023 - Apr 2024

- **Built** a 2D physics system in C++ with collision detection, resolution, and Quadtree broad-phase partitioning.
- **Implemented** weapons, companions, enemy behavior, and combat logic across the C++/C# boundary on a 7-person team.

EDUCATION

DigiPen Institute of Technology - B.S. Computer Science and Game Design, Minor in Mathematics | Graduated Dec 2025

Coursework: AI for Games, Computer Graphics, Low-Level Programming, System Design, Linear Algebra, Probability & Statistics