

ERDI GÜLMEN

Game Developer · Unreal Engine & C++

PERSONAL DETAILS

+90 554 166 2002

gulmenerdi@gmail.com

SOCIAL LINKS

LINKEDIN

linkedin.com/in/erdi-gulmen-305a85201

GITHUB

github.com/danteexe

FAB

fab.com/sellers/Dante%20Dev

YOUTUBE

youtube.com/@erdigulmen5305

DOCUMENTATION

dante-3.gitbook.io/dante-docs

SKILLS

ENGINE & LANGUAGES

Unreal Engine 5, C++, Blueprints

UNREAL SYSTEMS

Enhanced Input, CommonUI, Gameplay Tags, Game Features, Data Assets, PCG, World Partition, Mass / City Sample

DEVELOPMENT

Gameplay Systems, Editor Tools & Plugins, UI Systems, Performance Optimization, Async Asset Loading, Localization, Automated Testing

TOOLS

Git, PlasticSCM, Perforce, AWS, Blender, SketchUp, Twinmotion

LANGUAGES

English

Turkish

PROFILE

Unreal Engine developer specializing in C++ gameplay systems, editor tools and real-time interactive applications. Builds modular, data-driven frameworks for input, inventory, UI and game settings, optimizes rendering performance, and creates production pipelines and update systems. Publishes documented Unreal Engine plugins on Fab, with a focus on clean architecture and designer-friendly workflows.

WORK EXPERIENCE

Freelance Game Developer (Localization)

UnderBadger

Jan 2025 – Present

- Fully localized Unreal Engine projects, integrating all in-game content in 12 languages.
- Managed localization assets, string tables and translation workflows to keep in-game text organized, consistent and scalable.
- Supported project performance through optimization tasks and workflow improvements during localization passes.

C++ Developer

Archinord Software

Jun 2025 – Jun 2026

- Built real-time architectural visualization and presentation projects in Unreal Engine 5.
- Built a custom DLC / patch / update system on AWS cloud services, allowing client projects to be updated without full rebuilds.
- Developed a custom launcher with argument-based startup flows to open projects securely from outside the engine and drive the update mechanism.
- Created systems that fetch and display live residence data from CRM / cloud sources, including prices, availability, media and metadata.
- Optimized foliage, lighting and rendering setups, improving average FPS by approximately 30% while increasing visual quality.
- Built asset pipelines to prepare, optimize and import Twinmotion and SketchUp models into Unreal Engine 5, reducing manual setup.
- Developed custom editor tools and integrated World Partition, PCG, Mass / City Sample and CommonUI into project-specific features.

Unreal Engine Tools & Plugin Developer

Independent · Published on Fab

Present

- Developed and published reusable Unreal Engine plugins for gameplay experiences, attachments, input, HUD indicators and runtime screenshots.
- Built modular, extensible C++ / Blueprint systems focused on clean architecture, maintainability and usability for developers and designers.
- Wrote documentation, setup guides and usage examples on GitBook and YouTube.

Game Developer Intern

Tigloos Games

Aug 2023 – May 2024

- Developed core shooter gameplay mechanics, including weapon system foundations, inventory structure, health, damage and knockback.
- Built extensible Blueprint / C++ gameplay systems designed for reusability, maintainability and future development.
- Organized project assets for chunk-based downloading and scalable content delivery.

EDUCATION

Mechanical Drawing and Construction

Eskişehir Technical University
2022 – 2024

REFERENCES

Ataberk Palacıoğlu

Founder & Creative Director,
UnderBadger
ata.palacioglu@gmail.com
Relationship: Client

PROJECTS

Selected Unreal Engine plugins and game projects. Project titles link to product pages.

Modular Experience System | Unreal Engine Plugin · Fab

- Built an experience framework where gameplay experiences are composed through parent-child definitions, with Game Feature Actions activated and deactivated per experience.
- Added a user-facing experience workflow (map selection, display data, extra arguments, loading screens) and automatic Experience Manager setup for GameState-based projects.
- Fully Blueprint-exposed, with a modular loading screen system, custom editor asset factories, async Blueprint nodes and ready-to-use base classes.

Modular Attachment System | Unreal Engine Plugin · Fab

- Built a data-driven attachment system with a three-layer architecture (Fragments, Definitions, Components) for composing visual and gameplay elements on actors.
- Implemented a three-phase async loading pipeline with soft references and a ref-counted cache that prevents duplicate loading and reduces memory usage, plus in-editor preview.

Modular Input System | Unreal Engine Plugin · Fab

- Designed a data-driven framework on top of Enhanced Input with Gameplay Tag-based routing through configurable Input Modes, Processors and Gates, decoupled from the Character and Player Controller.
- Implemented an input mode stack for context-dependent controls, plus input blocking, filtering, consumption and buffering for combo-style actions.
- Provided C++ and Blueprint APIs and the Modular Input Monitor, a real-time debugger for allowed, blocked and buffered input.

Indicator System | Unreal Engine Plugin · Fab

- Built a screen-space HUD indicator system that projects world-space actors into UI markers, with off-screen edge clamping, directional arrows, multiple projection modes, distance-based visibility and overlap prevention.
- Created a data-driven setup with Data Assets and reusable C++ / Blueprint setup actions, fully configurable in Blueprint.

Async Screenshot | Unreal Engine Plugin · Fab

- Built a lightweight plugin with an async Blueprint node that captures screenshots at runtime, including in packaged builds.

Modular Inventory System | In Development

- Building a game-agnostic, composition-based framework for items, inventories, equipment and weapons, where a new item is a data asset made of fragments rather than a new class.
- Designed multi-container inventories keyed by Gameplay Tags and a transactional add / remove / move API with tag-based failure reasons and a preview mode.
- Implemented a weapon layer with 6 fire modes, hitscan, ammo, heat-based spread and pattern recoil; zero plugin dependencies, Blueprint-extensible, with console commands and a debug overlay.

Modular Game Settings | In Development

- Building a data-driven settings framework where each setting is composed of a value source, effects, conditions and a widget, so new settings need no C++.
- Shipped a CommonUI settings menu with tabs, search, gamepad navigation, live preview, apply / revert / reset and per-user and per-machine saving with version migration.
- Implemented Enhanced Input key rebinding with conflict handling; 60+ settings across 10 pages, passing strict and Shipping builds with automated test coverage.

Modular UI System | In Development

- Developing a data-driven CommonUI framework with UI Contexts, UI Metadata, Button Actions, async Modal Dialogs, Widget Mapping Tables and layered Game Layout workflows.

T.E.G | Game Jam · itch.io

- Developed all gameplay systems (movement, combat, weapon switching, abilities, enemy interactions) and the full in-game UI / HUD, working with 2D/3D art and narrative teammates.
- Delivered a fully playable Windows build within the game jam period.