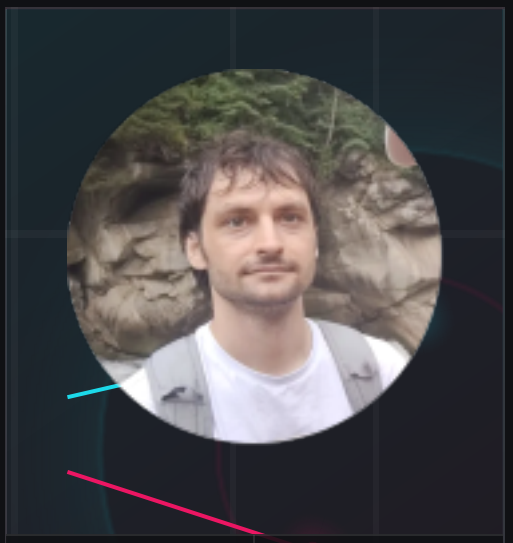
GAME ARCHITECTURE SINCE 2010

Jes Obertyukh



Game Architect & Technical Director •
Principal Unity/C# Engineer •
Architecture for Fast Iteration at Scale

I design game architecture and build gameplay systems that keep production fast as the project grows.

View LinkedIn ↗

~20 years in game development	**14+** years in architecture & leadership
Unity primary engine	**C#** primary language

Contact

jesobertyukh@gmail.com

linkedin.com/in/jes-obertyukh-20528a50 ↗

Telegram: @JesObertyukh ↗

Time zone: CEST

Languages

Ukrainian	Native or Bilingual
Russian	Native or Bilingual
English	Professional Working

Core Skills

- Game Architecture
- Multiplayer & Networking
- Performance Optimization
- Tools, CI/CD & Content Pipelines
- Unity
- Client/Server Architecture
- Rendering & Technical Art
- Technical Design & Game Rules
- C#, C, and C++
- Backend & Distributed Systems
- Voxel Systems
- Technical Leadership & Mentoring
- Gameplay Systems
- Reliability & Fault Tolerance
- ECS, DOD, Unity Jobs & Burst
- Architecture Audits & Project Rescue

01 / ABOUT

The whole game
One technical system

For nearly 20 years, I've worked hands-on across the full technical stack of game development

I design technical foundations and complex gameplay systems from architecture through production support, connecting game design, networking, rendering, technical art, QA, content, and delivery into clear, testable systems. Game-design study since 2007 helps me turn intent into coherent rules; I profile and optimize games for Apple, Android, and PC hardware, adapting to platform-specific GPU and memory constraints.

Architecture reviews are a regular part of my lead and consulting work. I trace global game flow, ownership, dependencies, testability, and hot paths, then propose focused changes that restore boundaries, reduce coupling, and recover iteration speed without unnecessary rewrites.

Across lead and CTO roles, I have mentored programmers and technical artists and led hiring. I also provide independent architecture consulting and long-term technical mentoring, from project setup and rescue to production workflows and technical decision-making.

1–2 sec	3–4 devs	2–3 days
Launch directly into the required gameplay state or test case.	Maintained exceptionally high feature throughput in a small engineering team.	Many gameplay features in about two days; complex UI screens in two to three.

My work has eliminated recurring integration and reliability failures and produced resilient client/server flows with seamless reload and reconnect. Production experience spans multiplayer and single-player FPS, Battle Royale, voxel and sandbox games, racing, strategy, runners, and roguelike shooters, among others.

I'm open to Game Architect, Technical Director, CTO, Principal Unity Engineer, and hands-on gameplay or systems roles, as well as architecture audits, project rescue, consulting, and technical mentoring.

> Good architecture should ensure that the fast solution is also the correct one

02 / ARCHITECTURE IN PRACTICE

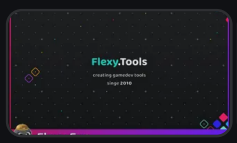
Experience, materialized

FLEXY.FUN

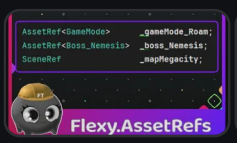
A reusable Unity architecture that materializes nearly two decades of engineering experience

Flexy.Fun is my long-term personal engineering initiative. Since 2010, I've been developing an original, general-purpose game-architecture approach. Its current Unity-native implementation feels like an extension of Unity, preserves modularity, testability, and iteration speed, and leaves game-specific development unconstrained. Flexy.Bricks / ToYs defines the methodology; Flexy.Framework turns it into practical systems, templates, and publicly available packages.

Flexy.Bricks / ToYs	An architectural methodology refined since 2010 to preserve modularity, testability, and iteration speed without fighting Unity or restricting game-specific development. Each unit can run independently in dedicated development and test scenes for fast implementation, debugging, manual verification, and automated testing.
Flexy.Framework	Flexy.Framework turns this methodology into a Unity-native architectural foundation. It defines how systems are composed, owned, initialized, connected, and tested across the entire game, while leaving game-specific design unconstrained. This gives projects clear boundaries, predictable flow, and fast iteration as they grow. Documentation, templates, and packages are publicly available on GitHub and the Unity Asset Store.

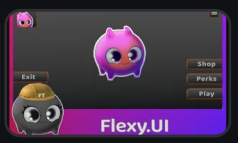


Flexy.Tools



Flexy.AssetRefs

Flexy.GameFlow



Flexy.UI

03 / WORK EXPERIENCE

From programmer to whole-product ownership

Mar 2025
Aug 2026
1 yr 6 mos

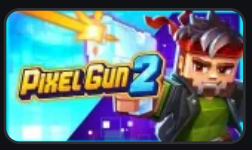
CUBIC GAMES

Game Architect & Principal Engineer

Owned the core client/server and gameplay architecture for Pixel Gun 2, a voxel-based multiplayer FPS built with Unity and Photon Fusion

- Designed the end-to-end game lifecycle covering initialization, metagame, core gameplay, transitions, reloads, reconnects, and shutdown, and structured game services and dependency injection around GameContexts
- Built controllable Photon Fusion simulation and presentation pipelines with explicit server-only, client-only, and shared-simulation boundaries, using ECS/DOD principles and message-bus communication
- Optimized gameplay and voxel-destruction synchronization, movement prediction, input latency and precision, hanging voxels, and destruction-driven NavMesh rebuilding
- Decoupled game modes from maps and enabled direct Play Mode, host, and client startup from gameplay maps with synchronous frame-0 initialization
- Optimized low-end mobile performance with algorithms, Jobs, Burst, and DOD; introduced ECS thinking and ECS/NetCode-ready boundaries without a project-wide rewrite
- Modernized supporting systems with C# 10, FEPM, an Addressables-based on-demand asset pipeline, and a flexible audio subsystem

Pixel Gun 2



Pixel Gun 2 on Steam

Mar 2023
Mar 2025
2 yrs 1 mo

ALFA BRAVO INC

Game Architect & Principal Engineer

Developed and optimized gameplay, Photon Bolt networking, rendering, and content-delivery systems for Combat Master, an online multiplayer Battle Royale FPS

- Designed the complete Content Update architecture covering asset layout, bundle versioning and assembly, and delivery through Google Play Asset Delivery
- Moved content-bundle assembly out of the monolithic build script into a modular task-based pipeline shared by local and CI builds
- Built a runtime depth-aware semi-automatic impostor system that kept the entire detailed Battle Royale map visible during aerial drops, with visual differences noticeable only on close inspection
- Reduced mobile GPU bandwidth and used texture arrays with a single material per map to keep standard maps near 52 rendering batches and the large map below 100
- Implemented networked moving platforms and a networked train, alongside vehicle systems
- Implemented perks, fully rebindable gamepad input, a player-profile editor, and ScriptableObject client-side tamper protection
- Continued supporting the team through periodic consulting on architecture, technical art, reusable Unity systems, and scalable backend solutions suitable for a team without dedicated backend specialists

Combat Master: Season 5

Combat Master: Welcome to Combat Zone!

Jun 2018
Mar 2023
4 yrs 10 mos

D.A.W.STUDIO

Chief Technology Officer

Led a four-person engineering team with end-to-end ownership of technical development across a studio of about 20 people—from architecture and technology selection to gameplay, backend, rendering, tools, production systems, QA integration, and delivery—while remaining hands-on in the architecture, implementation, and production trade-offs of every major subsystem

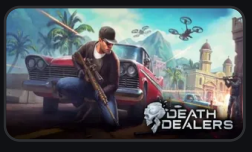
Designed and built from scratch the architecture for four games: two online multiplayer sniper games, a single-player runner, and a single-player roguelike shooter

- Across all four projects, a team of three to four programmers delivered many gameplay features in about two days and complex UI screens in two to three days, making content and configuration the primary production constraints
- Enabled direct startup into the required gameplay or test state in about one to two seconds
- Designed a Scrum delivery model using relative complexity, historical velocity, and Value/Time prioritization to improve planning predictability
- Built a Jira and Confluence workflow unifying feature specification, implementation, QA acceptance, and permanent documentation, combining User Stories, Job Stories, and JTBD with testable game rules and QA-owned final acceptance
- Separated department backlogs from work assigned to individuals, making ownership, available capacity, and unallocated work visible
- Partnered with game designers on story and gameplay across all four projects, translating design intent into consistent rules and resolving conflicts between features
- Simplified UI/UX by reducing unnecessary clicks and screen clutter while making interaction rules more consistent and learnable
- Selected and hired key specialists, supported the growth of junior and mid-level programmers, and coached artists in technical production and optimization
- Built a reproducible CI/CD pipeline around a Unity-owned build process, allowing developers and the build server to produce the same builds with the same configuration
- Owned asset and content pipeline architecture across studio projects, including content organization, build workflows, and delivery
- Kept complete game scenes within roughly 50–100 rendering batches through rendering, shader, and technical-art optimization
- Optimized water rendering for an island map while preserving its visual quality
- Consolidated layered rendering into a single-camera URP pipeline, replacing auxiliary cameras with Render Features
- Improved lightmap quality without increasing texture resolution, eliminating visible stepping artifacts
- Adjusted UI art production to achieve hardware-assisted antialiasing with negligible runtime cost
- Designed the communication architecture spanning ASP.NET Core metagame services and UDP battle servers, with offline operation and seamless reconnect; the same resilience model was later adopted independently by Combat Master and preserved player continuity during a prolonged backend outage
- Designed a custom networking layer over Unity Transport for both online multiplayer sniper games
- Designed battle-server orchestration, built a Python supervisor for managing processes on battle-server machines, and owned backend scaling decisions
- Built an ASP.NET Core player-profile service and matchmaker backed by MongoDB and Redis
- Created MC-VmV, an MVVM-derived architectural pattern supporting the same application structure in single-player and server-backed games
- Built a Game Design Information system that translated gameplay rules into structured, reusable configuration models, giving designers clear control over gameplay systems
- Built reusable client-side systems for player profiles, UI windows, asset bundles, battle quests, and game-designer-friendly IAP
- Built the entire CoreGame on an ECS-based foundation using Unity Entities, spanning movement, weapon firing, ballistics, controllable bullet-sprites, network synchronization, and other gameplay subsystems
- Built level-design and pathing tools with integrated animation and camera controls


Cyberstrike


Animals Happy Run


Sniper League: The Island



Death Dealers

Dec 2016
Jun 2018
1 yr 7 mos

TELLURION MOBILE GAMES

Lead Unity Developer

Improved performance and gameplay foundations for RealmCraft, a released mobile voxel sandbox

- Moved voxel lighting from CPU to GPU, optimized shaders and fog, and contributed to delta-based lighting updates and culling, reducing recalculation, rendering cost, and battery usage
- Designed extensible mob AI and accelerated pathfinding
- Rebuilt voxel-world movement for stability, performance, and customization
- Built working RealmCraft VR versions for Gear VR and HTC Vive, optimizing rendering and implementing VR-specific camera movement and controls

RealmCraft with Skins Export to Minecraft

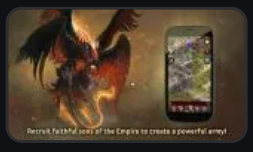
Jun 2016
Dec 2016
7 mos

SERIOUS CAKE

Lead Unity Developer

Led the Unity client for Age of Phoenix: Wind of War, a 4X strategy game backed by Akka.NET

- Improved runtime performance, reduced build size, and streamlined UI production for faster feature iteration
- Contributed selected Akka.NET backend features while retaining primary ownership of the Unity client
- Conducted technical interviews, mentored programmers and technical leads, and coached artists in technical production and optimization



Age of Phoenix: Wind of war

Jan 2015
Jun 2016
1 yr 6 mos

WARGAMING.NET

Lead Unity Developer

Led client development for REVOLT across architecture, client/server systems, gameplay, and optimization

- Rebuilt and stabilized the client architecture, restoring development flow and establishing reusable foundations for subsequent features
- Built a universal, fault-tolerant client/server command flow for simultaneous-turn gameplay, ensuring both clients replayed the same authoritative outcome
- Optimized rendering to run reliably on iPad 2 with substantial performance headroom
- Developed a custom stylized shader supporting the hand-drawn visual direction
- Established a collaborative technical-art workflow to translate art direction into the game's visual style and mentored the technical artist in rendering and production practices
- Worked in a mentor-guided Scrum team with rotating responsibilities and continuous improvement through retrospectives, later applying those practices as CTO

REVOLT Tank Battle

Sep 2014
Dec 2014
4 mos

PLARIUM

Technical Lead

Developed core technology for Motorplanet, a mobile multiplayer racing game

- Implemented network synchronization
- Built an in-editor track builder for rapid greybox iteration, allowing designers to place a car and drive any section before handing the track to the art team

Motorplanet

Aug 2014
Sep 2014
2 mos + advisory support

8D STUDIO

System Architect & Lead Programmer

Joined for a predefined two-month Unity engagement to design and implement the core architecture and part-based character-rendering system at the center of Sketch Tales, a single-player voxel adventure for Steam

Handed development to the internal team as planned and continued advising through most of the project's later production

Sketch Tales

Aug 2013
Jul 2014
1 yr

WEST-GAMES

System Architect & Lead Programmer

Owned technical direction and system architecture for Seaker, a client/server hidden-object game built around 3D panoramas

- Built the game's client/server functionality using Unity Legacy Networking
- Designed a unified shader and material approach that reduced the game's rendering to roughly seven batches

---

Feb 2012
Jul 2013
1 yr 6 mos

GOGAMES

System Architect & Lead Programmer

Led system architecture and hands-on C# server development for World of Fishing, defining the client/server communication protocol and supporting development of the Flash client

- Led the backend data migration from SQL to MongoDB and Redis, establishing a stack later reused at West-Games and D.A.W.Studio


World of Fishing

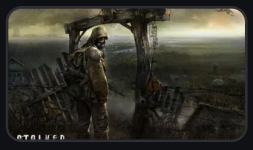
Dec 2009
Dec 2011
2 yrs 1 mo

GSC GAME WORLD

Tools & Game Logic Programmer

Developed game logic and production tooling for S.T.A.L.K.E.R. 2

- Worked across C and C++ engine runtime and gameplay code, C# editor tooling, and the Managed C++ interop layer connecting them
- Built an editor-tooling framework including a property grid, curve editor, and foundational node editor
- Implemented gameplay systems including proximity triggers and centralized projectile management



S.T.A.L.K.E.R. 2

Jun 2007
Nov 2009
2 yrs 6 mos

SONELLY

Programmer

Developed game systems and engine infrastructure in C# for a custom XNA-based .NET 3D engine

- Designed a JSON-based scene format supporting object hierarchies, components, loading, and persistence
- Implemented player save and load persistence
- Built scene-effect rendering and smooth transitions between scenes
- Developed hidden-object gameplay with an inventory system
- Implemented match-3 gameplay logic
- Developed a character-recoloring shader supporting multiple color variants

ARCHITECTURE FOR FAST ITERATION AT SCALE

Let's build the foundation right

LinkedIn ↗

GitHub ↗