

ROBERT BIAULET

Gameplay & Tools Programmer

📍 Lille, France ☎ +33 6 81 51 52 57 ✉ biauletr@gmail.com [in](#) [LinkedIn](#) [🌐 Portfolio](#)

SUMMARY

► I have 5+ years of experience on both in-development and live titles, mostly Unity and C#, with time in [Haxe/Heaps](#) and two custom ECS codebases. I'm at home iterating on gameplay features or building the tools that accelerate everyone else's work.

EXPERIENCE

-  **Motion Twin:** Programmer —  [Windblown](#) ----- February 2025 • September 2026
🇫🇷 Fully remote · 👤 10-12 people
- Jack of all trades to alleviate pressure and keep a good development speed
 - Taking care of hard to fix bugs on the multiplayer or ECS's views layer
 - Supported development of various features: meta-progression, gameplay, cinematics...
-  **Leikir Studio:** Gameplay programmer —  ----- August 2024 • January 2025
🇫🇷 On site · 👤 10 people
- Project refactor for better iteration and productivity
 - Simplification of an existing ECS model to better leverage Unity's strengths
-  **Shiro Games:** Gameplay programmer —  [Northgard](#) ----- March 2023 • July 2024
🇫🇷 On site · 👤 8-10 people
- Creation of DLCs end to end (*Quick prototyping followed by polish*)
 - 3 new playable factions (*Stoat, Owl, Hounds*)
 - ~12 new levels for the Conquest mode of the game
 - Creation of various new tools to improve the team's visibility on the state of the game
 - Design and implementation of Analytics using Amplitude
 - Debug interface to better balance fights
-  **BetaDwarf:** Programmer —  [Vaultbreakers](#) ----- June 2021 • March 2023
🇬🇧 Fully remote · 👤 20-30 people
- Design, creation and maintenance of features or tools in their entirety
 - Bridge logic between an ECS gameplay layer and Unity visuals
 - Set of localization tools to edit, import and export text entries
 - Prototyping and polish of gameplay features for a MOBA A-RPG
 - Active and Passive abilities usable by players and enemies controlled by AI

SKILLS

- **Highly skilled using Unity and C#**
I know my way around the engine: UI, VFXs, localization, profiling, editor tools... And I'm really comfortable with C#: unsafe, async, custom exceptions...
- **Software development knowledge**
To make my code more robust, I learned a lot of useful concepts often brushed over in regular gameplay programming: memory management, design patterns, threads, allocations lifetimes...
- **Art integration**
I make art myself using: Gimp, Blender, Figma... This allows me to know what the constraints and pipelines for art are and makes it easier to integrate into the engine.
- Quick to adapt**
I often had to learn new technologies on the spot: Haxe/Heaps, ECS, PlasticSCM, CI/CD pipelines like with TeamCity...
- Game design**
I studied and still practice Game Design. It helps me get a better understanding of what I'm working on and also communicate efficiently with designers.

EDUCATION

- **[Rubika SupinfoGame](#)** — 2018 • 2021
- 3 years: Bachelor in Game design
 - 1 year: Master in Game Programming