

MONTAL — commercial 2D game with Qt/QML & C++ performance paths

Domain: Consumer product / games **Duration:** 2013 – 2014 **Stack:** C++ · Qt · QML

A shippable 2D game product — QML for presentation and flow; C++ for movement, collision, and AI where frame budget matters. Not a prototype: production iteration across gameplay and UI polish.

CHALLENGE

Frame-budget gameplay with polished QML UI

Balance fluid QML screens with performance-critical game loops in C++ — object scaling, mechanics, and asset integration without sacrificing frame stability on target hardware.

APPROACH

QML + C++ split with adult supervision on hot paths

- QML for menus, flow, and presentation layers
- C++ for physics, collision, AI, and performance-critical loops
- Iterative gameplay and UI polish cycles to shippable quality
- Pattern reused in later Qt industrial and trading programmes

TECHNOLOGY

Stack & delivery surface

Qt QML C++ 2D game Performance tuning

OUTCOMES

What the programme proved

- Early proof of Qt/QML + C++ delivery depth (founder portfolio)
- Referenced on sevendyne.com/projects/qt/ as representative delivery
- Shows product-grade Qt beyond enterprise forms and dashboards