

R-Type

Server (C++): Multi-threading, must handle multiple games at the same time and in a row.

Client (SFML): Players, monsters, power-ups, missiles, obstacles, background....

Cross-platform:

- **Conan** (dependencies): [Conan Dependencies](#) – [Epitech Dependencies](#) – [Bincrafter Dependencies](#)

- **CMake** => Build the project:

mkdir build && cd build && conan install && .. -build=missing && cmake .. -G "Unix Makefiles" && cmake -build .

➔ On Windows the build result to a Visual Studio Solution.

Protocol: Client-Server communications

➔ **!!! UDP !!!** (if justify we can use TCP)

➔ [Documentation Protocols](#) in RFC: RFC [Keywords](#) + **ASCII format** + **postscripts** (in English)

Libraries:

- Use **Boost**
- If needed **Boost:ASIO** that is NOT RECOMMENDED ! It's better to make our own server.

Engine: Re-usable (+++), extensible, abstracts

Our engine must be able to make Battlefield, Counter-Strike, Call of Duty, League of Legends etc.

- **Run-time extend**: can load dynamic libraries at run-time to add systems
- **Compile time safe**: beautiful code!
- **Easy** to use (someone can make a Snake very fast)

General:

- ➔ Starfield: our background! Must scroll horizontally (with **TIMERS**).
- ➔ Players move with arrow keys.
- ➔ **Multi-thread** server.
- ➔ If client **crash the server MUST works and NOTIFY other players**.
- ➔ Player 1, 2, 3, 4 = Blue, Red, Yellow, Green.
- ➔ **Monsters must be dynamics libraries** loaded by server at run-time
- ➔ We must have an **API interacting with them**.
- ➔ **Monsters spawn randomly** on the right of the screen.
- ➔ ++++++ BONUS ++++++

Restrictions:

- ➔ If **usage of libc: that's means that it has no C++ equivalent** and we **MUST wrap it**.
- ➔ If **lib not specified: it is forbidden**.
- ➔ If we use copies instead of **pointers and references**: **JUSTIFY IT**.
- ➔ If we don't modify a variable it must be **const**, if not: **JUSTIFY IT**.
- ➔ **!!! PERFORMANCE + CLEAN CODE + NO USELESS CODE !!!**