

1 Files & Folders

- Media files (images, audio) follow `UpperCamelCase.<extension>`
- C++ source files follow `UpperCamelCase.cpp`
- C++ header files follow `UpperCamelCase.hpp`
- Folders/directories follow `lower_underscore`
- `.md` files follow `CAPS_UNDERSCORE.md`

2 C++

2.1 General

The ORG engine's priorities for C++ code are generally ordered as follows: **1.** portability, **2.** maintainability, **3.** clarity, and **4.** performance. Therefore if a segment of code is clear, but not easily maintained, it should be changed. Performant but not clear? Should probably be changed. Maintainable but not portable? Change it. Etc., etc.

STL containers are preferred when applicable. Similarly, C++ paradigms are preferred over C equivalents; e.g. `enum class` over `enum`, `struct` over `typedef struct`, `constexpr` over `#define`. Standard OO best practices should be followed; member data should be not be externally visible (accessor functions should be used instead),

`auto` is a useful tool for writing maintainable code, but be weary of using it in complicated snippets as it can hinder readability by hiding information.

2.2 Naming / Identifiers

- Class and struct names follow `UpperCamelCase`
- Member variables follow `_lowerCamelCase`
- Static class variables follow `s_lowerCamelCase`
- Local variables follow `lowerCamelCase`
- Constants follow `CAPS_UNDERSCORE`
- Enum values follow `CAPS_UNDERSCORE`

Lastly, short forms in identifiers should generally be avoided. So, if you're writing a class that represents a **regular expression**, `RegularExpression` is much preferred over `RegExp`.

3 C

4 Python

5 CMake