Things to know about window

Here is the brief list of what you can or cannot do with the SFML windows.

You can create multiple windows

SFML allows you to create multiple windows, and to handle them either all in the main thread, or each one in its own thread (but... see below). In this case, don't forget to have an event loop for each window.

Multiple monitors are not correctly supported yet

SFML doesn't explicitly manage multiple monitors. As a consequence, you won't be able to choose which monitor a window appears on, and you won't be able to create more than one fullscreen window. This should be improved in a future version.

Events must be polled in the window's thread

This is an important limitation of most operating systems: the event loop (more precisely, the `pollEvent` or `waitEvent` function) must be called in the same thread that created the window. This means that if you want to create a dedicated thread for event handling, you'll have to make sure that the window is created in this thread too. If you really want to split things between threads, it is more convenient to keep event handling in the main thread and move the rest (rendering, physics, logic, ...) to a separate thread instead. This configuration will also be compatible with the other limitation described below.

On macOS, windows and events must be managed in the main thread

Yep, that's true; macOS just won't agree if you try to create a window or handle events in a thread other than the main one.

On Windows, a window which is bigger than the desktop will not behave correctly

For some reason, Windows doesn't like windows that are bigger than the desktop. This includes windows created with `VideoMode::getDesktopMode()` : with the window decorations (borders and titlebar) added, you end up with a window which is slightly bigger than the desktop.