

FlyWithLua V2.7 NG Reference Manual

William Good

October 09, 2018



This manual is trying to document the added features that FlyWithLua NG provides and is meant to be used with the original manual.

Two new features that were added is floating windows and support for ImGui. Floating windows allow you to create a window that can be used in 2d or VR that supports OpenGL just like the original FlyWithLua. ImGui is a graphics widget library that also supports 2d or VR. In 2d mode you can move the window to a second monitor.

In the “FlyWithLua/Scripts (disabled)” folder there are example scripts for both features. The floating windows examples are prefaced with “floating_wnd_” and the ImGui are prefaced with “imgui_”.

Reference

Predefined variables

SYSTEM_DIRECTORY

Full path to the X-Plane install directory with trailing slash.

INTERNALS_DIRECTORY

Full path to the FlyWithLua Internals directory with trailing slash.

MODULES_DIRECTORY

Full path to the FlyWithLua Modules directory with trailing slash.

This is how we create a floating window. Please look in FlyWithLua/Scripts (disabled) for examples prefaced with “floating_wnd_”.

demo_floating_wnd = float_wnd_create(800, 450, 1, false)

The first two parameters specify the size of the window in boxels. Boxels are scalable pixes. If the UI scale is set to 100% in the settings, a boxel equals a pixel. However, if the user sets the UI scale to more than 100%, a boxel will be scaled to span multiple pixels.

The third parameter specifies the window decoration. The following decorations are possible:

- 0: "X-Plane will draw no decoration for your window, and apply no automatic click handlers. The window will not stop click from passing through its bounds. This is suitable for "windows" which request, say, the full screen bounds, then only draw in a small portion of the available area."
- 1: "The default decoration for "native" windows, like the map. Provides a solid background, as well as click handlers for resizing and dragging the window."
- 2: "X-Plane will draw no decoration for your window, nor will it provide resize handlers for your window edges, but it will stop clicks from passing through your windows bounds."
- 3: "Like 2, but with resizing; X-Plane will draw no decoration for your window, but it will stop clicks from passing through your windows bounds, and provide automatic mouse handlers for resizing."

The last parameter configures whether you want to use imgui for the floating window.

The return value is a handle to the window that can be used to configure additional things.

This function sets the title of the window if it has a decoration.

float_wnd_set_title(demo_floating_wnd, "floating window Demo")

The first parameter must be a handle to a window previously created with float_wnd_create and the second parameter is the title of the window.

This function sets the initial position of the window.

float_wnd_set_position(demo_floating_wnd, 775, 650)

The first parameter must be a handle to a window previously created with float_wnd_create and the second and third parameters set the initial position of the window.

This function sets the name of the ondraw function.

```
float_wnd_set_ondraw(demo_floating_wnd,  
"on_draw_floating_window")
```

The first parameter must be a handle to a window previously created with `float_wnd_create` and the second parameter set the name of the ondraw function.

This function sets the name of the onclick function.

```
float_wnd_set onclick(demo_floating_wnd,  
"on_click_floating_window")
```

The first parameter must be a handle to a window previously created with `float_wnd_create` and the second parameter set the name of the onclick function.

This function sets the name of the onclose function.

```
float_wnd_set_onclose(demo_floating_wnd,  
"on_close_floating_window")
```

The first parameter must be a handle to a window previously created with `float_wnd_create` and the second parameter set the name of the onclose function.

This function gets the handle of the window created above to be used with ffi.

```
xplmtest_wnd =  
float_wnd_get_xplm_handle(demo_floating_wnd)
```

The first parameter must be a handle to a window previously created with `float_wnd_create`.

function on_draw_floating_window(demo_floating_wnd, x3, y3)

x and y are the origin of the window, i.e. the lower left
x increases to the right, y increases to the top

function on_click_floating_window(demo_floating_wnd, x3, y3)

x and y are relative from the origin of the window, i.e. the lower left

function on_close_floating_window(demo_floating_wnd)

When on_close it called, it is illegal to do anything with the wnd variable.
It is also not allowed to create new windows in on_close!

This is how we create a ImGui window. Please look in FlyWithLua/Scripts (disabled) for examples prefaced with “ImGui_”.

demo_wnd = float_wnd_create(800, 450, 1, true)

The first two parameters specify the size of the window in boxels. Boxels are scalable pixels. If the UI scale is set to 100% in the settings, a boxel equals a pixel. However, if the user sets the UI scale to more than 100%, a boxel will be scaled to span multiple pixels.

The third parameter specifies the window decoration. The following decorations are possible:

- 0: "X-Plane will draw no decoration for your window, and apply no automatic click handlers. The window will not stop click from passing through its bounds. This is suitable for "windows" which request, say, the full screen bounds, then only draw in a small portion of the available area."
- 1: "The default decoration for "native" windows, like the map. Provides a solid background, as well as click handlers for resizing and dragging the window."
- 2: "X-Plane will draw no decoration for your window, nor will it provide resize handlers for your window edges, but it will stop clicks from passing through your windows bounds."
- 3: "Like 2, but with resizing; X-Plane will draw no decoration for your window, but it will stop clicks from passing through your windows bounds, and provide automatic mouse handlers for resizing."

The last parameter configures whether you want to use ImGui for the floating window.

The return value is a handle to the window that can be used to configure additional things.

This function sets the title of the window if it has a decoration.

float_wnd_set_title(demo_wnd, "Demo Window")

The first parameter must be a handle to a window previously created with float_wnd_create and the second parameter is the title of the window.

This function sets the initial position of the window.

float_wnd_set_position(demo_wnd, 775, 650)

The first parameter must be a handle to a window previously created with float_wnd_create and the second and third parameters set the initial position of the window.

imgui GUIs are built each frame. You will need to supply a function that is called every frame.

`float_wnd_set_imgui_builder(demo_wnd, "build_demo")`

The first parameter must be a handle to a window previously created with `float_wnd_create` and the second parameter must be the name of a function. Note that unlike other function in FlyWithLua, you can only pass the name of a function here, not an arbitrary lua string. This function will be called for every frame while the window is visible, so you don't need an additional `do_every_frame` unless you also need to do things while the window is closed.

If the window has a decoration with a close button (like decoration 1), you might need a way to know when the users closes the window. This function can be used to setup a function to be called when the user closes the window.

`float_wnd_set_onclose(demo_wnd, "closed_demo")`

The first parameter must be a handle to a window previously created with `float_wnd_create` and the second parameter must be the name of a function. Note that unlike other function in FlyWithLua, you can only pass the name of a function here, not an arbitrary lua string. This function will be called when the user closes the window. After the window is closed, its builder function will not be called again and it is illegal to use the window handle variable returned by `float_wnd_create`, e.g. it is illegal to set the window title after it was closed.

imgui supports drawing images, so let's load one. The images must always be loaded globally to make sure they are only loaded once per script!

```
image_id = float_wnd_load_image(SCRIPT_DIRECTORY ..  
"/imgui_demo.jpg")
```


imgui widgets have no state, that means we need to store states of checkboxes, radio buttons etc. globally

```
makeRed = false  
sliderVal = 0  
choice = 1  
angle = 0  
text = ""
```

function build_demo(wnd, x, y)

This function is called for every frame. Use this function to create your GUI. The first parameter (wnd) is the handle of the window. It is the same handle as the one returned by `float_wnd_create`. It is required if you have multiple windows that use the same builder function so you know which of these windows is currently being built. The x and y parameters are the current position of the window in OpenGL coordinates, i.e. the position of the lower left corner in global screen coordinates. These coordinates are only needed for using the graphics module on top of the imgui GUI. If you only use imgui, these coordinates are not required because imgui has its own coordinate system inside the window.

function closed_demo(wnd)

This function is called when the user closes the window. Drawing or calling imgui functions is not allowed in this function as the window is already destroyed.