Open Fest 2022

Is The C++ Desktop Dead?

ImGUI: Cool, multiplatform, C++ GUI Library

Doncho Angelov, VP Engineering, Pliant.io

@doncho in/donangel

So, you know today’s C++

- Firmware
- Drivers
- Backend
- Compilers
- Command-line stuff

…

- But GUI? React? Angular? MFC?
    - (except for some great car manufacturers’ UI)

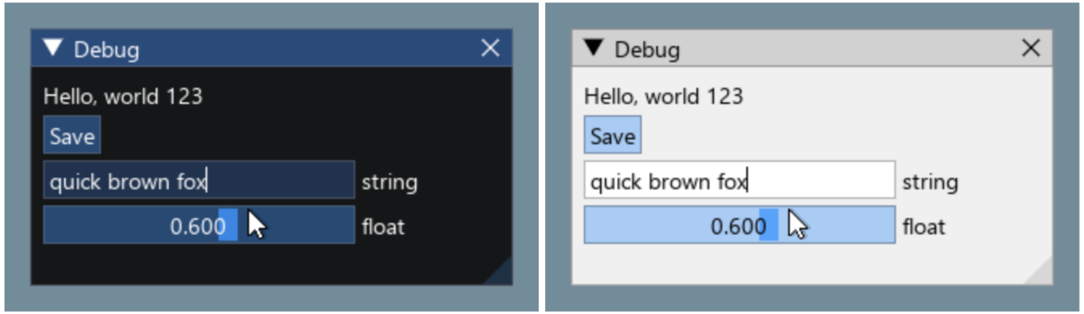
Pliant

Can I display something different than…

```
Enter the number of elements: 5
You will have 5 elements in your queue. Now enter
each element!
Element 1: Blah
Element 2: Moo
~/>
```

Meet “Dear ImGui”: An immediate mode GUI for C++

```
ImGui::Text("Hello, world %d", 123);
if (ImGui::Button("Save"))
    MySaveFunction();
ImGui::InputText("string", buf, IM_ARRAYSIZE(buf));
ImGui::SliderFloat("float", &f, 0.0f, 1.0f);
```



```
// Create a window called "My First Tool", with a menu bar.
ImGui::Begin("My First Tool", &my_tool_active, ImGuiWindowFlags_MenuBar);
if (ImGui::BeginMenuBar())
{
    if (ImGui::BeginMenu("File"))
    {
        if (ImGui::MenuItem("Open..", "Ctrl+O")) { /* Do stuff */ }
```

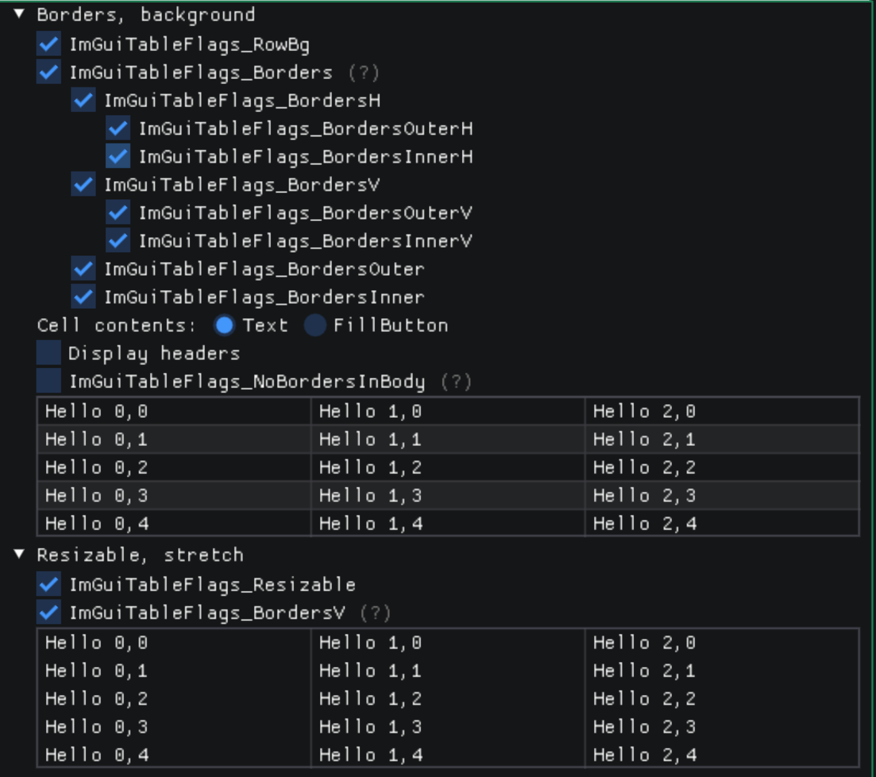
The “Immediate Mode” Paradigm

Disclaimer: Lots of religious arguments: “immediate” vs. “retained”

Immediate (as in ImGUI) means:

- Do Your Job Yourself: you construct the interface!
- No References to UI Objects in your code (well, sort of...)
- The app state is kept in your code, not in the library
- You own the “app loop”
- …and some other weird stuff, more typical for games!

Plethora of Widgets



▼ Borders, background

- [✓] ImGuiTableFlags_RowBg
- [✓] ImGuiTableFlags_Borders (?)
  - [✓] ImGuiTableFlags_BordersH
    - [✓] ImGuiTableFlags_BordersOuterH
    - [✓] ImGuiTableFlags_BordersInnerH
  - [✓] ImGuiTableFlags_BordersV
    - [✓] ImGuiTableFlags_BordersOuterV
    - [✓] ImGuiTableFlags_BordersInnerV
  - [✓] ImGuiTableFlags_BordersOuter
  - [✓] ImGuiTableFlags_BordersInner

Cell contents: [● Text] FillButton

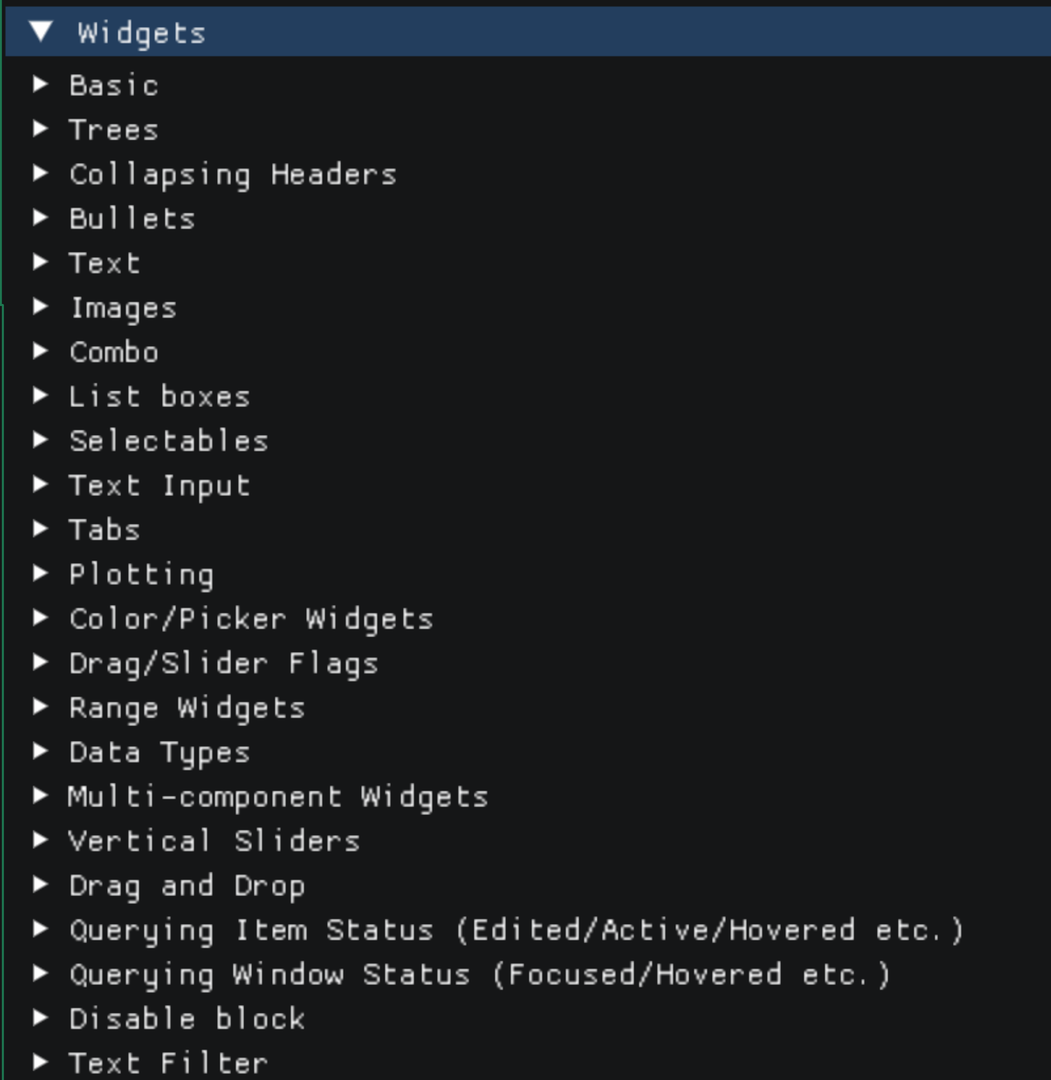
- [ ] Display headers
- [ ] ImGuiTableFlags_NoBordersInBody (?)

Hello 0,0	Hello 1,0	Hello 2,0
Hello 0,1	Hello 1,1	Hello 2,1
Hello 0,2	Hello 1,2	Hello 2,2
Hello 0,3	Hello 1,3	Hello 2,3
Hello 0,4	Hello 1,4	Hello 2,4

▼ Resizable, stretch

- [✓] ImGuiTableFlags_Resizable
- [✓] ImGuiTableFlags_BordersV (?)

Hello 0,0	Hello 1,0	Hello 2,0
Hello 0,1	Hello 1,1	Hello 2,1
Hello 0,2	Hello 1,2	Hello 2,2
Hello 0,3	Hello 1,3	Hello 2,3
Hello 0,4	Hello 1,4	Hello 2,4



Quite a Few Supported Platforms

List of Platforms Backends:

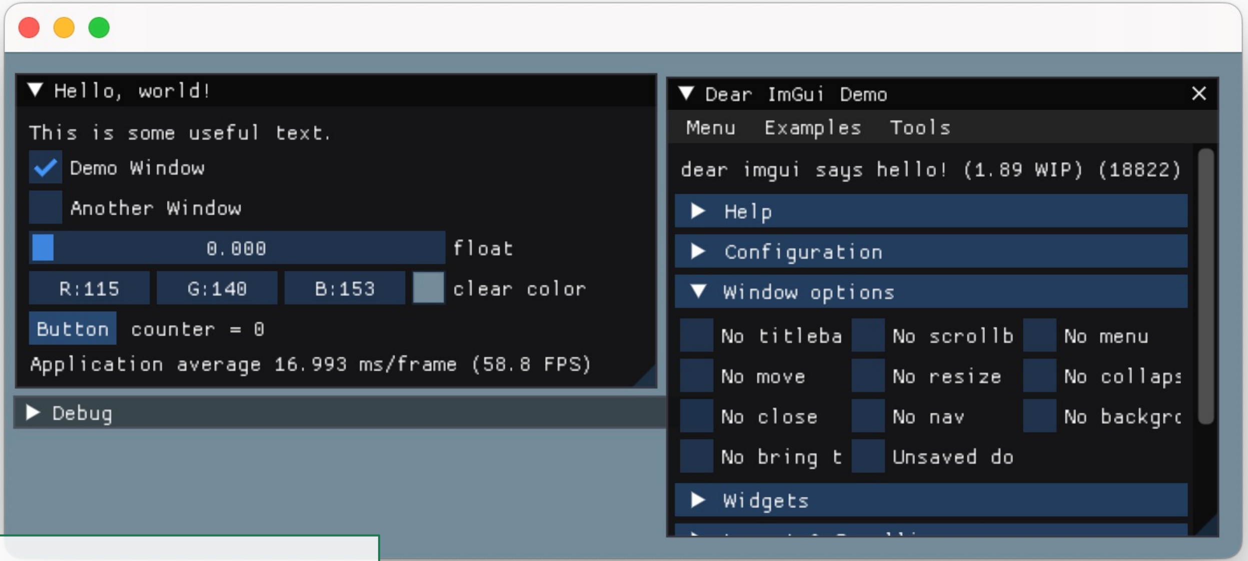
```
imgui_impl_android.cpp    ; Android native app API
imgui_impl_glfw.cpp       ; GLFW (Windows, macOS, Linux, etc.) http://www.glfw.org/
imgui_impl_osx.mm         ; macOS native API (not as feature complete as glfw/sdl backends)
imgui_impl_sdl.cpp        ; SDL2 (Windows, macOS, Linux, iOS, Android) https://www.libsdl.org
imgui_impl_win32.cpp      ; Win32 native API (Windows)
imgui_impl_glut.cpp       ; GLUT/FreeGLUT (this is prehistoric software and absolutely not recommended today!)
```

List of Renderer Backends:

```
imgui_impl_dx9.cpp        ; DirectX9
imgui_impl_dx10.cpp       ; DirectX10
imgui_impl_dx11.cpp       ; DirectX11
imgui_impl_dx12.cpp       ; DirectX12
imgui_impl_metal.mm       ; Metal (with ObjC)
imgui_impl_opengl2.cpp    ; OpenGL 2 (legacy, fixed pipeline <- don't use with modern OpenGL context)
imgui_impl_opengl3.cpp    ; OpenGL 3/4, OpenGL ES 2, OpenGL ES 3 (modern programmable pipeline)
imgui_impl_sdlrenderer.cpp; SDL_Renderer (optional component of SDL2 available from SDL 2.0.18+)
imgui_impl_vulkan.cpp     ; Vulkan
imgui_impl_wgpu.cpp       ; WebGPU
```

Pliant

The Demo Code / App



▼ Hello, world!

This is some useful text.

✓ Demo Window

Another Window

0.000 float

R:115 G:140 B:153 clear color

Button counter = 0

Application average 16.993 ms/frame (58.8 FPS)

▶ Debug

▼ Dear ImGui Demo

Menu Examples Tools

dear imgui says hello! (1.89 WIP) (18822)

▶ Help

▶ Configuration

▼ Window options

No titleba · No scrollb · No menu · No move · No resize · No collaps · No close · No nav · No backgrc · No bring t · Unsaved do

▶ Widgets

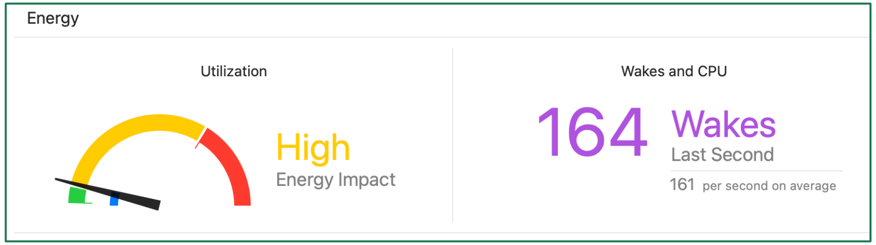
```
// 1. Show the big demo window
// (Most of the sample code is in ImGui::ShowDemoWindow()!
// You can browse its code to learn
// more about Dear ImGui!).
if (show_demo_window)
    ImGui::ShowDemoWindow(&show_demo_window);
```

- Keep it handy: displays “Dear ImGui Demo” window with a single call
- Use it to see “how the professionals do ImGUI”

How the Demo App Runs on my Desktop

MacBook Pro M1

- Quite high CPU, but won't get higher
- Low memory footprint
- Energy hog:


Energy

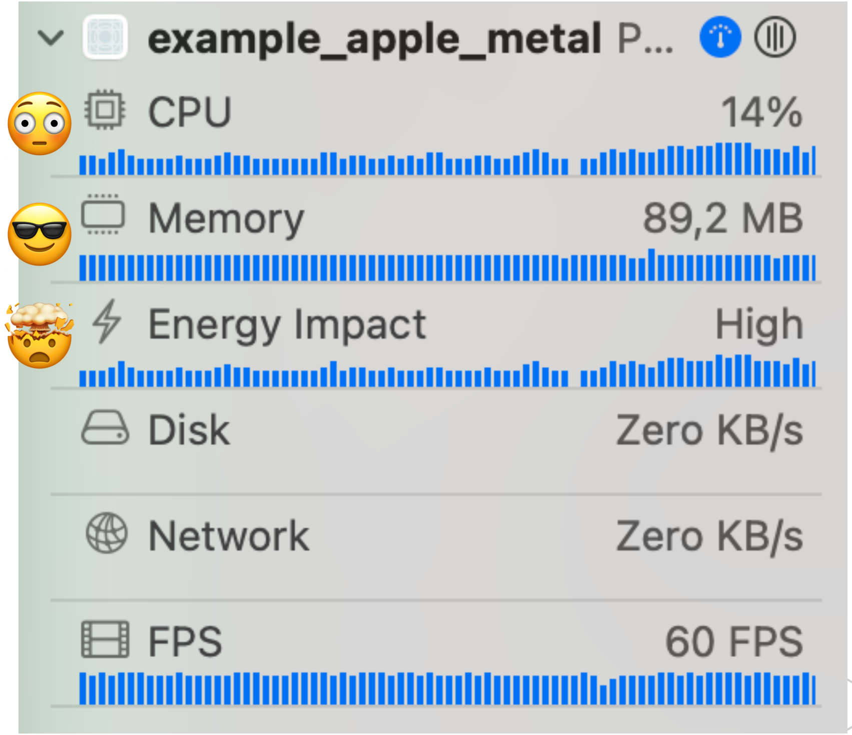
Utilization

High
Energy Impact

Wakes and CPU

164 Wakes
Last Second

161 per second on average

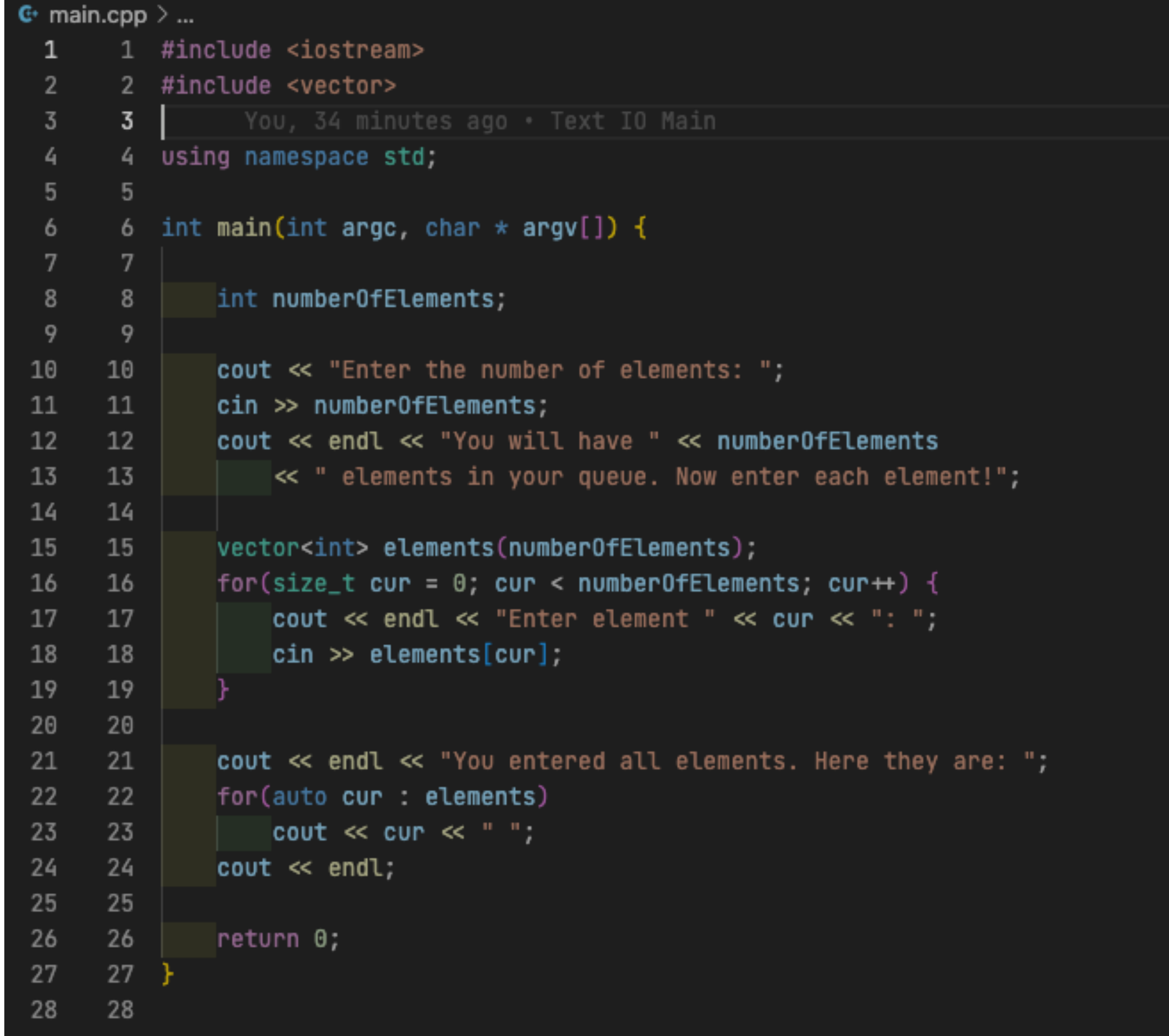

example_apple_metal P...

CPU	14%
Memory	89,2 MB
Energy Impact	High
Disk	Zero KB/s
Network	Zero KB/s
FPS	60 FPS

Pliant

Time To Get Our Hands Dirty!

1. Simple C++ program
   Level 1 Warlock at "The Deathknell Graves"
2. Let's convert it to GUI with ImGUI!



main.cpp > ...

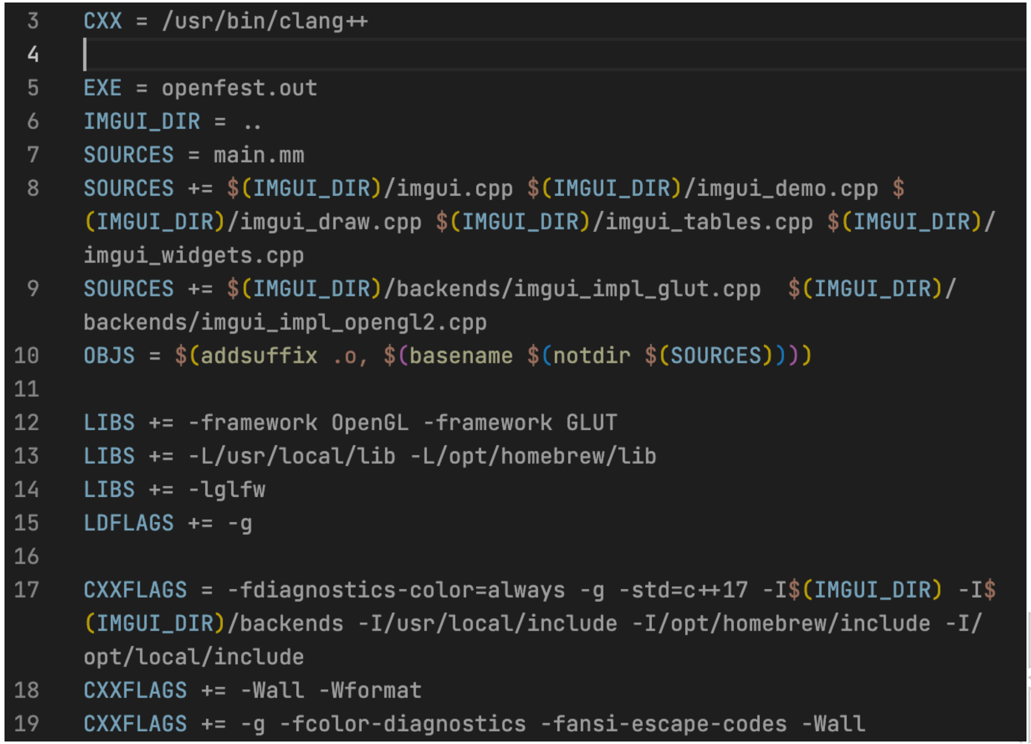
```cpp
#include <iostream>
#include <vector>
        You, 34 minutes ago • Text IO Main
using namespace std;

int main(int argc, char * argv[]) {

    int numberOfElements;

    cout << "Enter the number of elements: ";
    cin >> numberOfElements;
    cout << endl << "You will have " << numberOfElements
        << " elements in your queue. Now enter each element!";

    vector<int> elements(numberOfElements);
    for(size_t cur = 0; cur < numberOfElements; cur++) {
        cout << endl << "Enter element " << cur << ": ";
        cin >> elements[cur];
    }

    cout << endl << "You entered all elements. Here they are: ";
    for(auto cur : elements)
        cout << cur << " ";
    cout << endl;

    return 0;
}

```

The Obsolete(st) example_glut_opengl2 Makefile

$(IMGUI_DIR)

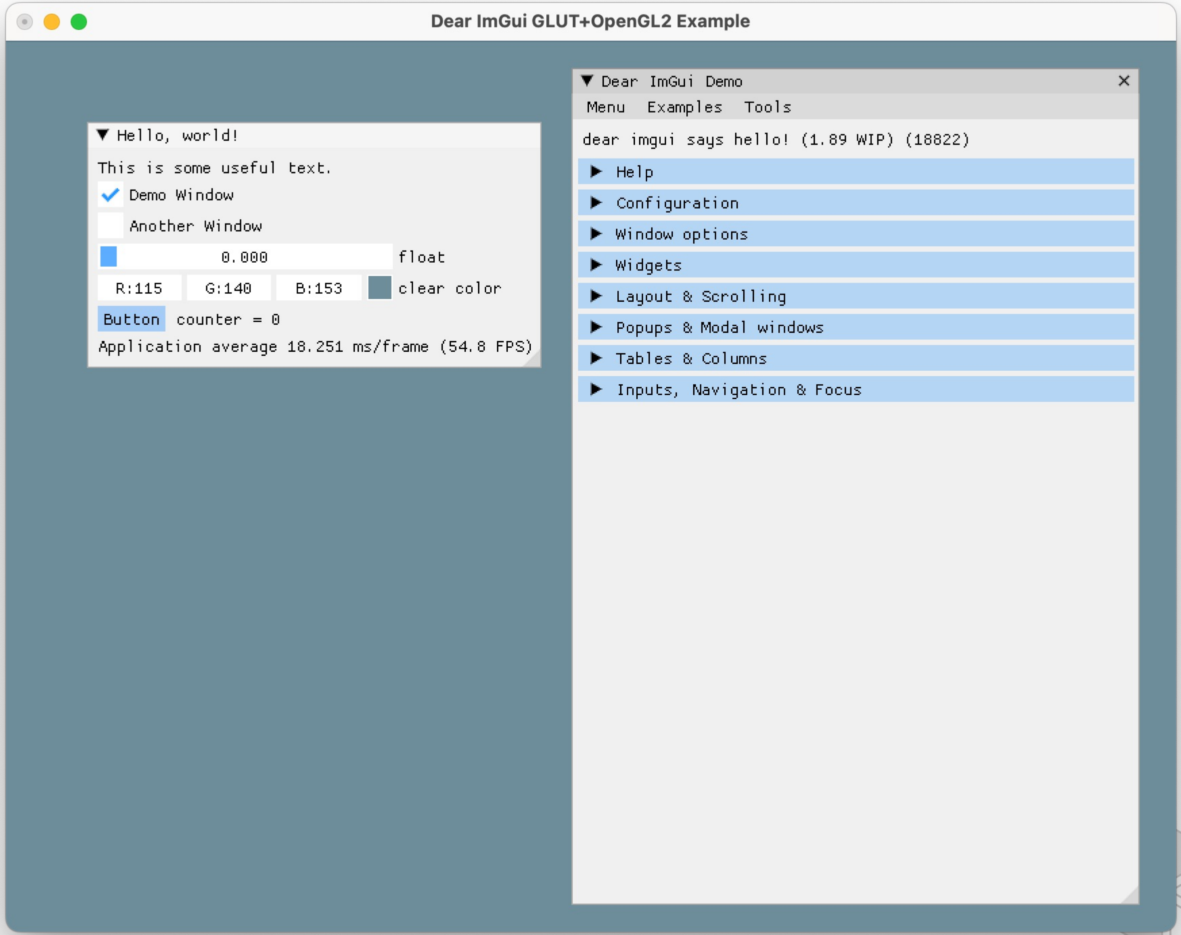
/imgui.cpp
/imgui_demo.cpp
/imgui_draw.cpp
/imgui_tables.cpp
/imgui_widgets.cpp

$(IMGUI_DIR)/backends/
imgui_impl_glut.cpp
imgui_impl_opengl2.cpp



```
CXX = /usr/bin/clang++

EXE = openfest.out
IMGUI_DIR = ..
SOURCES = main.mm
SOURCES += $(IMGUI_DIR)/imgui.cpp $(IMGUI_DIR)/imgui_demo.cpp $(IMGUI_DIR)/imgui_draw.cpp $(IMGUI_DIR)/imgui_tables.cpp $(IMGUI_DIR)/imgui_widgets.cpp
SOURCES += $(IMGUI_DIR)/backends/imgui_impl_glut.cpp  $(IMGUI_DIR)/backends/imgui_impl_opengl2.cpp
OBJS = $(addsuffix .o, $(basename $(notdir $(SOURCES))))

LIBS += -framework OpenGL -framework GLUT
LIBS += -L/usr/local/lib -L/opt/homebrew/lib
LIBS += -lglfw
LDFLAGS += -g

CXXFLAGS = -fdiagnostics-color=always -g -std=c++17 -I$(IMGUI_DIR) -I$(IMGUI_DIR)/backends -I/usr/local/include -I/opt/homebrew/include -I/opt/local/include
CXXFLAGS += -Wall -Wformat
CXXFLAGS += -g -fcolor-diagnostics -fansi-escape-codes -Wall
```

First Blood!


Dear ImGui GLUT+OpenGL2 Example

▼ Hello, world!

This is some useful text.

✓ Demo Window

Another Window

0.000 float

R:115 G:140 B:153 clear color

Button counter = 0

Application average 18.251 ms/frame (54.8 FPS)

▼ Dear ImGui Demo

Menu Examples Tools

dear imgui says hello! (1.89 WIP) (18822)

▶ Help

▶ Configuration

▶ Window options

▶ Widgets

▶ Layout & Scrolling

▶ Popups & Modal windows

▶ Tables & Columns

▶ Inputs, Navigation & Focus

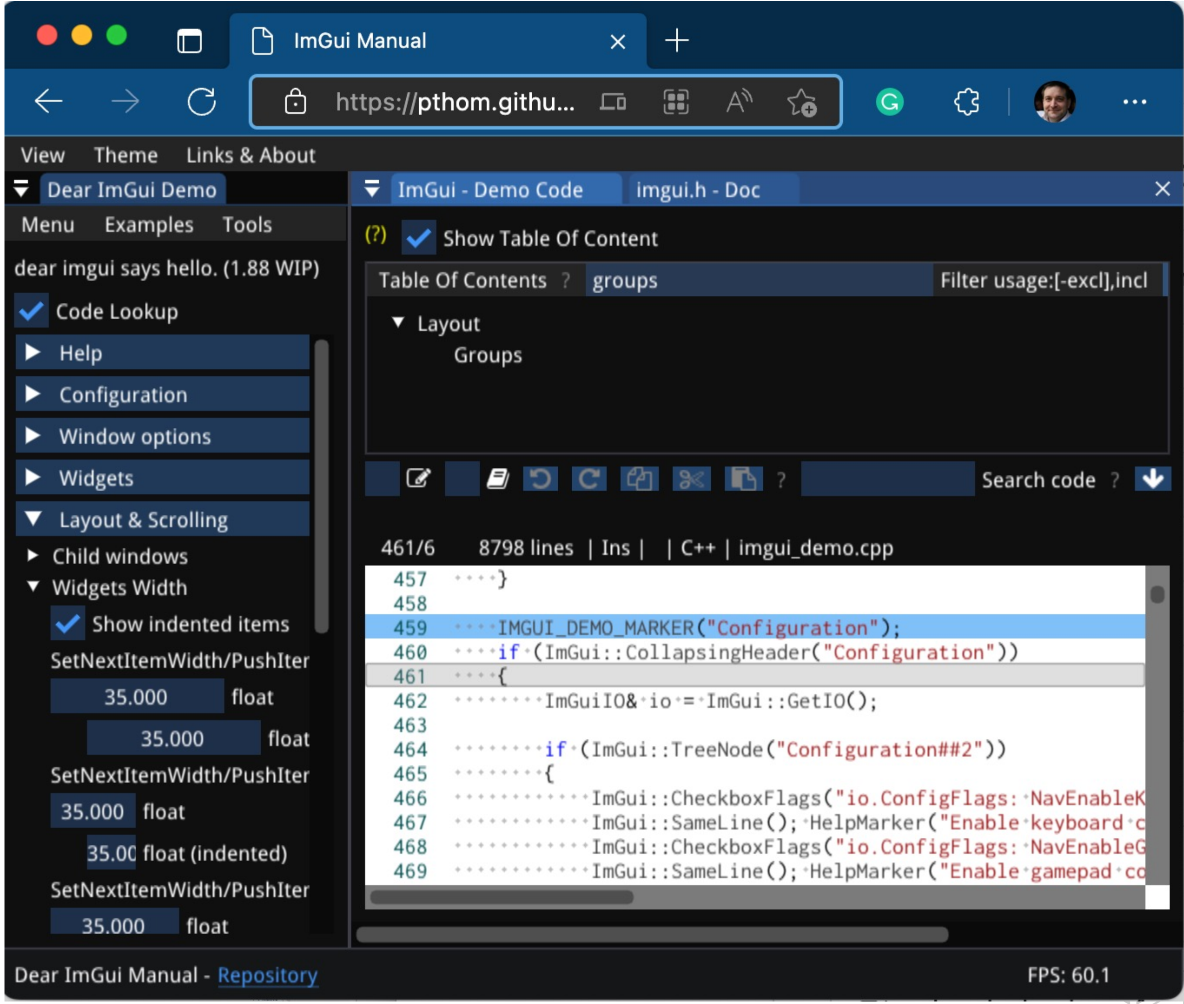
Pliant

First Docs!

FAQs:
https://github.com/ocornut/imgui/blob/master/docs/FAQ.md

Great interactive Docs:

https://pthom.github.io/imgui_manual_online/manual/imgui_manual.html

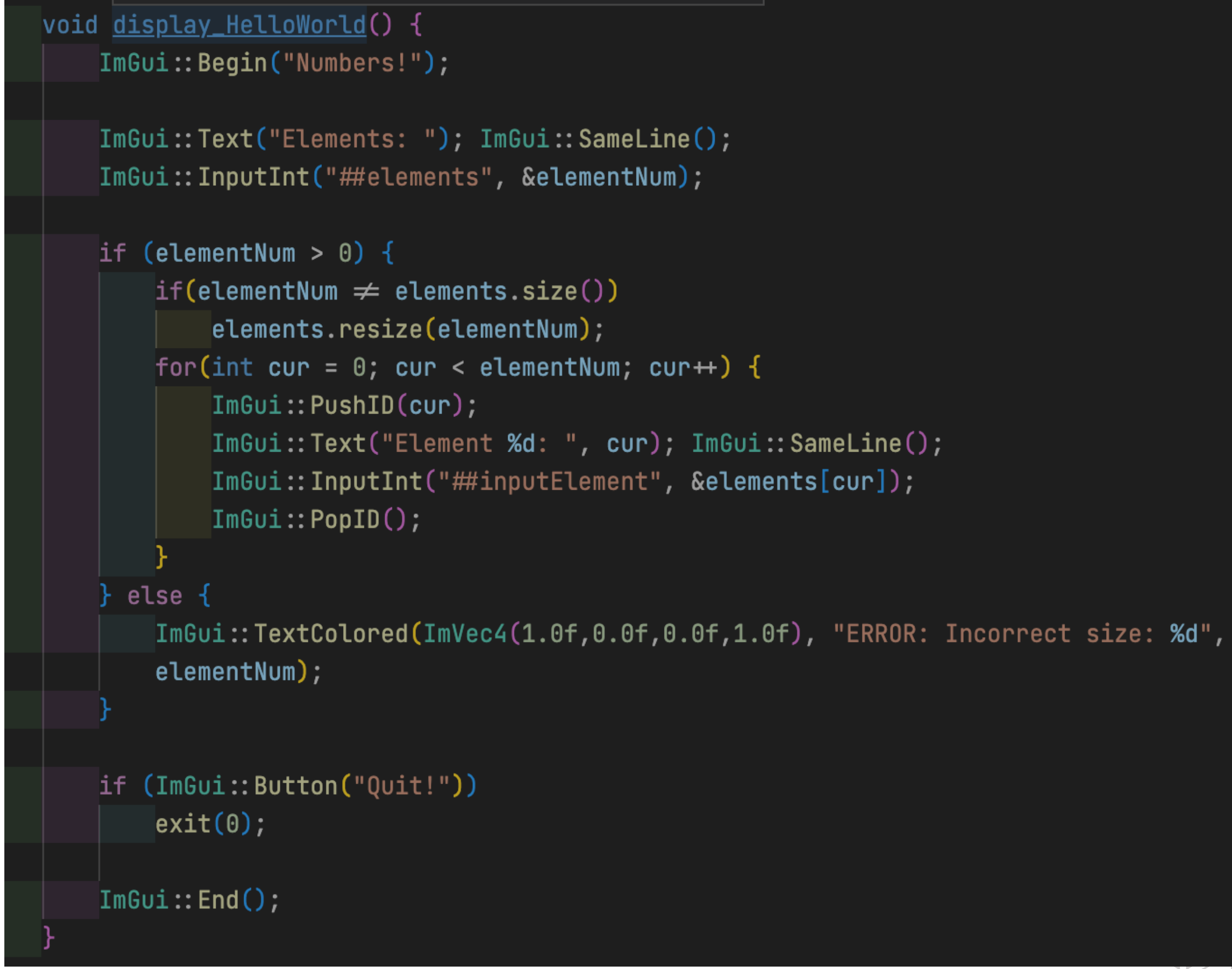


```
461/6    8798 lines  | Ins |   | C++ | imgui_demo.cpp
    }

    IMGUI_DEMO_MARKER("Configuration");
    if (ImGui::CollapsingHeader("Configuration"))
    {
        ImGuiIO& io = ImGui::GetIO();

        if (ImGui::TreeNode("Configuration##2"))
        {
            ImGui::CheckboxFlags("io.ConfigFlags: NavEnableK
            ImGui::SameLine(); HelpMarker("Enable keyboard c
            ImGui::CheckboxFlags("io.ConfigFlags: NavEnableG
            ImGui::SameLine(); HelpMarker("Enable gamepad co
```

Pliant

Our reworked work of art!

1. IDs/Labels/ are no joke! PushID/PopID
2. Refreshes the rendering each time when there’s an interface change.
3. Colorful error message

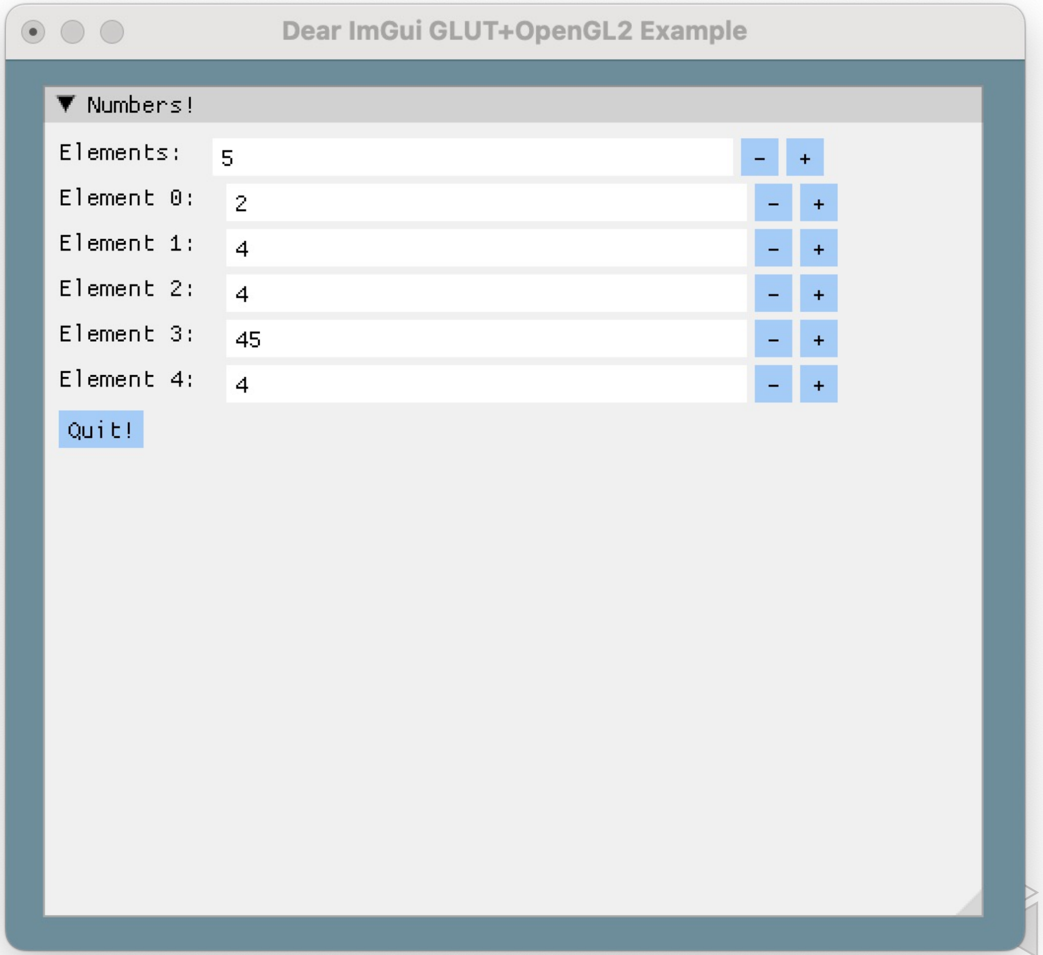


```
void display_HelloWorld() {
    ImGui::Begin("Numbers!");

    ImGui::Text("Elements: "); ImGui::SameLine();
    ImGui::InputInt("##elements", &elementNum);

    if (elementNum > 0) {
        if(elementNum ≠ elements.size())
            elements.resize(elementNum);
        for(int cur = 0; cur < elementNum; cur++) {
            ImGui::PushID(cur);
            ImGui::Text("Element %d: ", cur); ImGui::SameLine();
            ImGui::InputInt("##inputElement", &elements[cur]);
            ImGui::PopID();
        }
    } else {
        ImGui::TextColored(ImVec4(1.0f,0.0f,0.0f,1.0f), "ERROR: Incorrect size: %d",
        elementNum);
    }

    if (ImGui::Button("Quit!"))
        exit(0);

    ImGui::End();
}
```

Our reworked work of art!

1. Tested for up to 100,000 elements (well, did not enter them all!)
2. Maybe a group to hold all elements, so that "Quit" and "Elements" remain on the screen during scroll
3. Resize handling
4. Smarter window proc

…….


Dear ImGui GLUT+OpenGL2 Example

▼ Numbers!

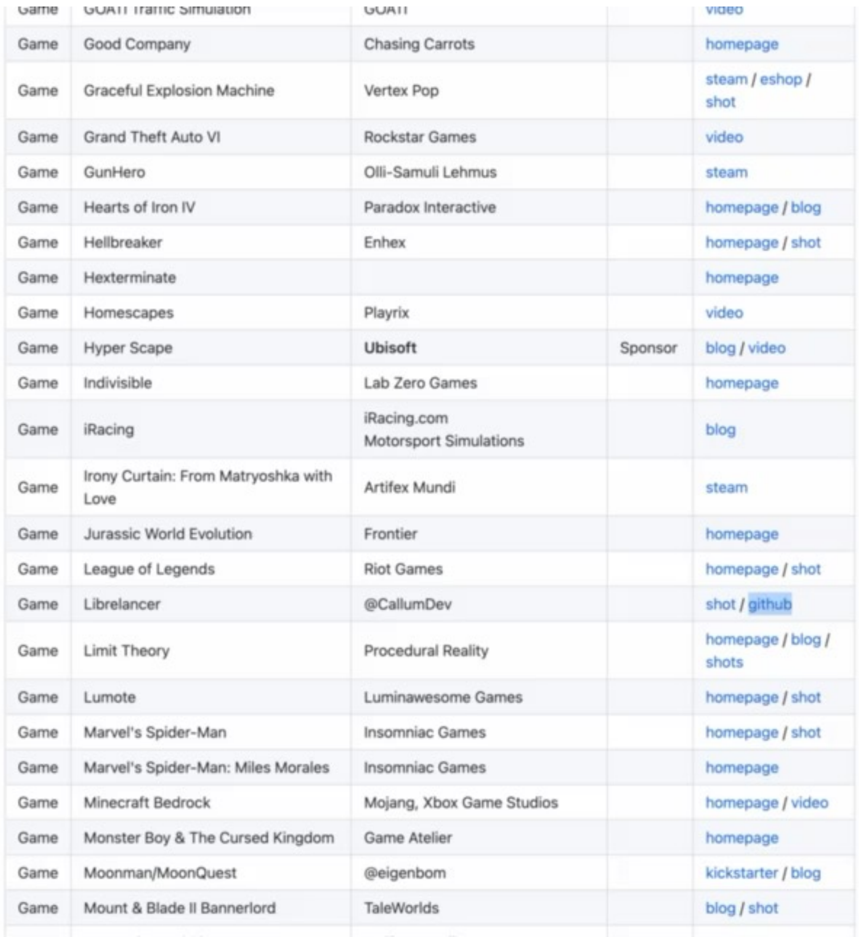
Elements:	5	-	+
Element 0:	2	-	+
Element 1:	4	-	+
Element 2:	4	-	+
Element 3:	45	-	+
Element 4:	4	-	+

Quit!

Pliant

And now – some cool applications!

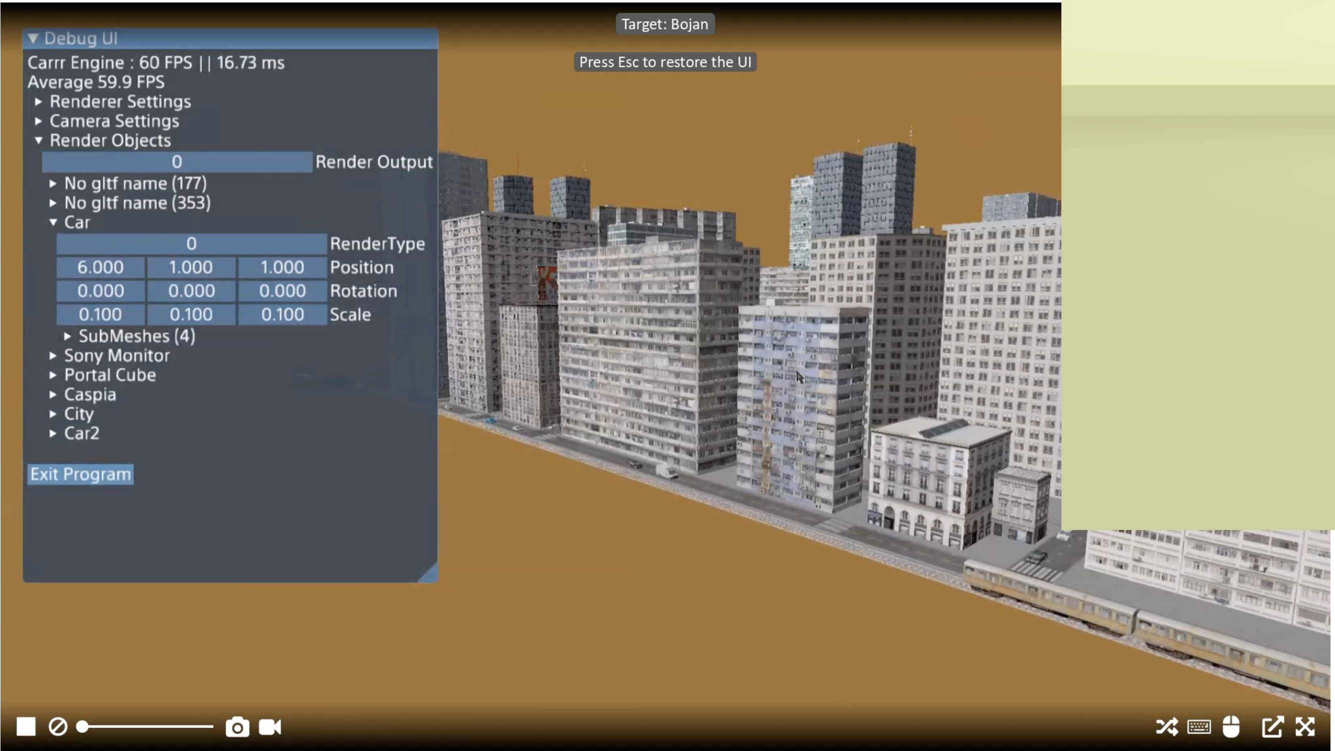
Who's Using it: A Non-extensive list from the ImGUI site



Game	GOATI Traffic Simulation	GOATI		video
Game	Good Company	Chasing Carrots		homepage
Game	Graceful Explosion Machine	Vertex Pop		steam / eshop / shot
Game	Grand Theft Auto VI	Rockstar Games		video
Game	GunHero	Olli-Samuli Lehmus		steam
Game	Hearts of Iron IV	Paradox Interactive		homepage / blog
Game	Hellbreaker	Enhex		homepage / shot
Game	Hexterminate			homepage
Game	Homescapes	Playrix		video
Game	Hyper Scape	**Ubisoft**	Sponsor	blog / video
Game	Indivisible	Lab Zero Games		homepage
Game	iRacing	iRacing.com Motorsport Simulations		blog
Game	Irony Curtain: From Matryoshka with Love	Artifex Mundi		steam
Game	Jurassic World Evolution	Frontier		homepage
Game	League of Legends	Riot Games		homepage / shot
Game	Librelancer	@CallumDev		shot / github
Game	Limit Theory	Procedural Reality		homepage / blog / shots
Game	Lumote	Luminawesome Games		homepage / shot
Game	Marvel's Spider-Man	Insomniac Games		homepage / shot
Game	Marvel's Spider-Man: Miles Morales	Insomniac Games		homepage
Game	Minecraft Bedrock	Mojang, Xbox Game Studios		homepage / video
Game	Monster Boy & The Cursed Kingdom	Game Atelier		homepage
Game	Moonman/MoonQuest	@eigenbom		kickstarter / blog
Game	Mount & Blade II Bannerlord	TaleWorlds		blog / shot

Pliant

ImGUI on PS5 and Pi➡️⬇️



▼ Debug UI
Carrr Engine : 60 FPS || 16.73 ms
Average 59.9 FPS
▸ Renderer Settings
▸ Camera Settings
▾ Render Objects

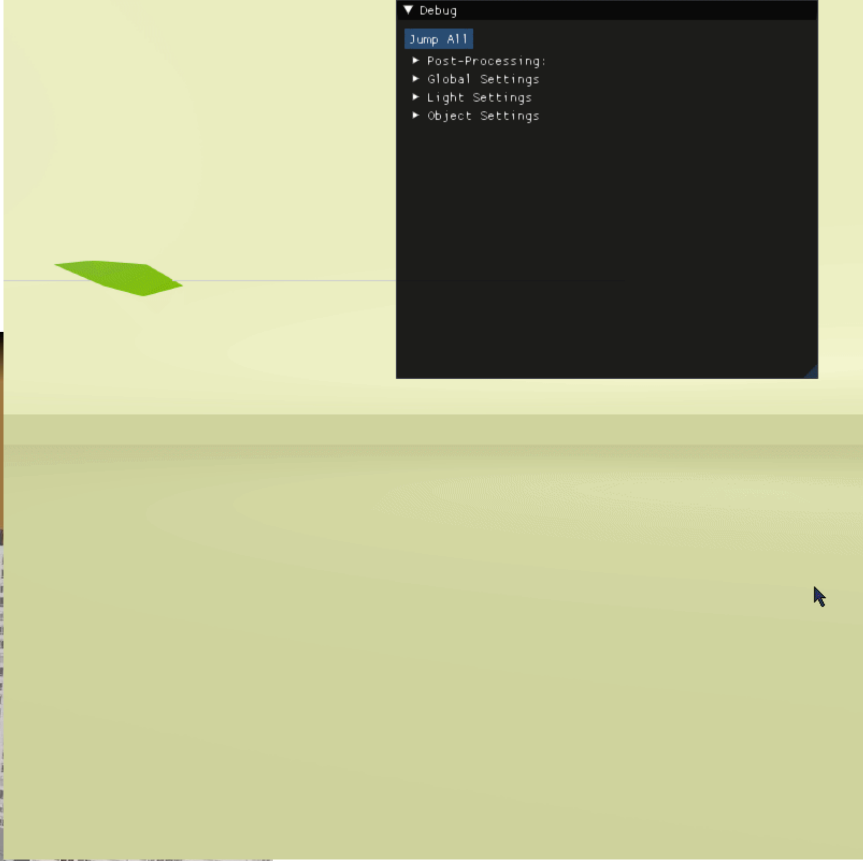
0			Render Output

▸ No gltf name (177)
▸ No gltf name (353)
▾ Car

0			RenderType
6.000	1.000	1.000	Position
0.000	0.000	0.000	Rotation
0.100	0.100	0.100	Scale

▸ SubMeshes (4)
▸ Sony Monitor
▸ Portal Cube
▸ Caspia
▸ City
▸ Car2

Exit Program

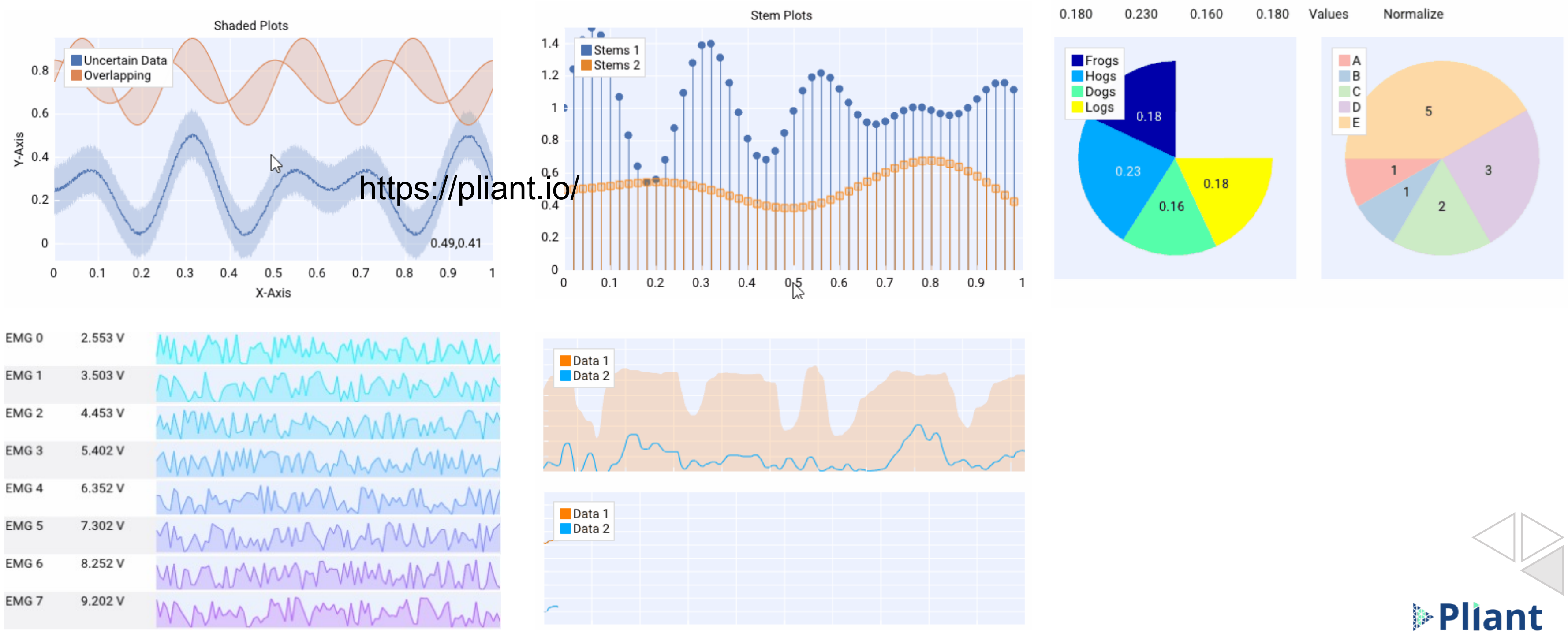

▼ Debug
Jump All
▸ Post-Processing:
▸ Global Settings
▸ Light Settings
▸ Object Settings

Pliant

Cool Stuff: ImPlot

implot

"immediate mode, GPU accelerated plotting library for Dear ImGui"



Pliant

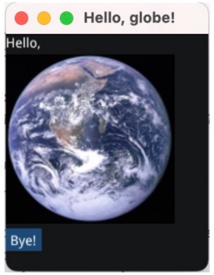
Cool Stuff: Hello, Dear ImGui

hello_imgui

Dear ImGui "`Hello, world!`" in one single call.

```
HelloImGui::Run(
    []{ ImGui::Text("Hello, world!"); }, // GUI code: this lambda will display a single label
    { 200.f, 50.f },                     // window size
    "Hello!" );                          // window title
```

A slightly more complex multiplatform app, including assets and callbacks is also extremely simple to write. The "Hello Globe" app shown below is composed with three simple files. It will run with no additional modifications (including in the cmake code) on iOS, Android, Linux, Mac, Windows and Emscripten_



```
└── hello_globe.main.cpp   // main file, see below
├── CMakeLists.txt         // 2 lines of cmake, for all platforms!
├── assets/
│   └── world.jpg          // assets are embedded automatically, even on mobile platforms!
```

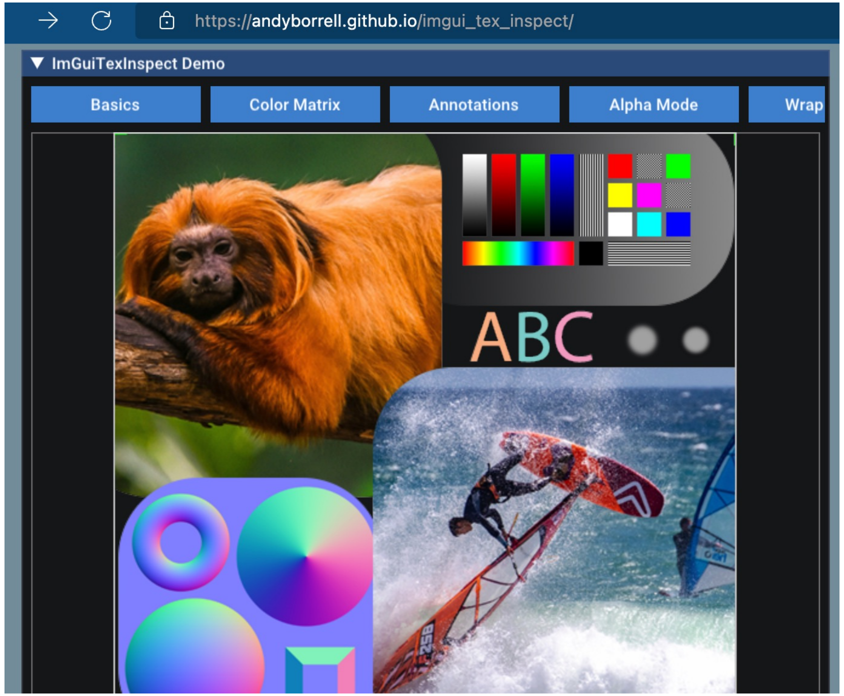
Pliant

ImGUI in the web

imgui_tex_inspect

- Yes, it's possible.
- The imgui_tex_inspect demo is amazing (andyborrell @ github).
- Maybe one day I'll be web developer, again?

```
example_emscripten_opengl3
example_emscripten_wgpu
```


https://andyborrell.github.io/imgui_tex_inspect/

Thank you, OpenFest!

…it was a great experience preparing for this talk;

…and it proved to me that C++ is far from dead: tons of stuff and libs there!