

Create a simple GUI app in C++ w/ ImGui in 5min

How to create a simple GUI app using C++ and Dear ImGui

Requirements:

- Some C++
- Visual Studio Community 2022 (C++ dev tools)
- Git
- 5 min

- [A C++ GUI app in 5 min](#)

A C++ GUI app in 5 min

Dear ImGui describe itself as a bloat-free graphical user interface library for C++. It outputs optimized vertex buffers that you can render anytime in your 3D-pipeline enabled application. It is fast, portable, renderer agnostic and self-contained (no external dependencies).

Dear ImGui is particularly suited to integration in games engine (for tooling), real-time 3D applications, fullscreen applications, embedded applications, or any applications on consoles platforms where operating system features are non-standard.

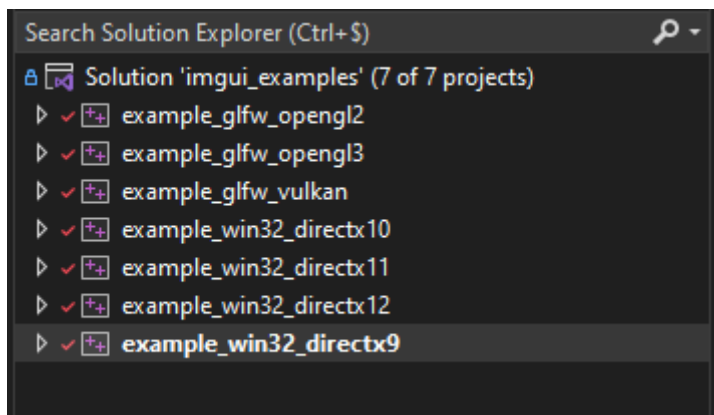
src: <https://imgui-test.readthedocs.io/en/latest/>

5min, let's start the Stopwatch.

Let's start by opening a terminal in a directory of your choice and cloning Dear ImGui repositories from Github:

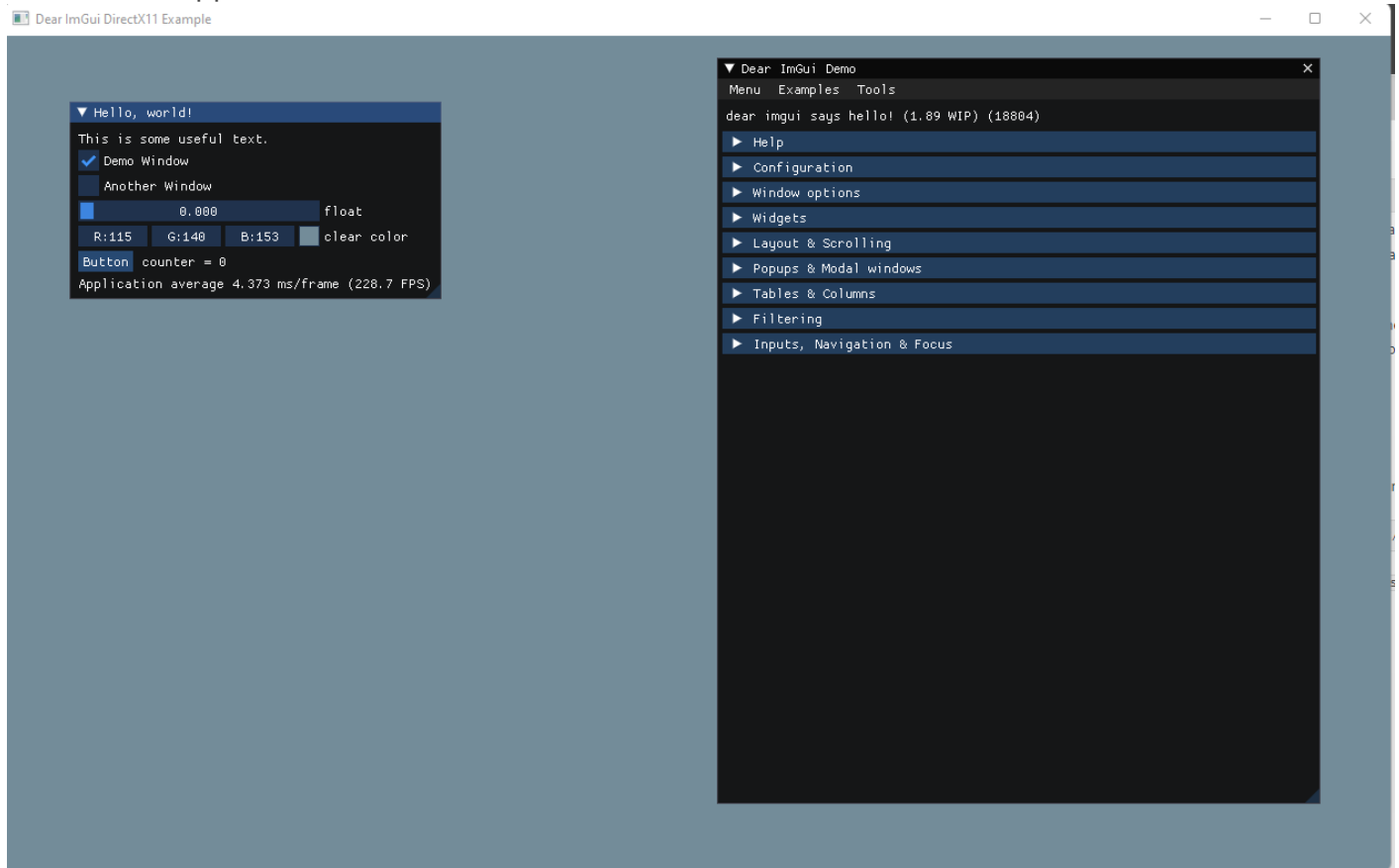
```
git clone --recursive https://github.com/ocornut/imgui.git
```

Then browse to `imgui/examples/` and open `imgui_examples.sln` you might need to retarget the project solution. As you can see there are many examples using opengl, vulkan, directx.



Right click on the solution you want to try, personally I choosed DirectX11 (`example_win32_directx11`), then click on **Set as Startup Project**.

To check that everything is working let's try to build our app by pressing **Ctrl + F5** ! You should see a demo app like that:

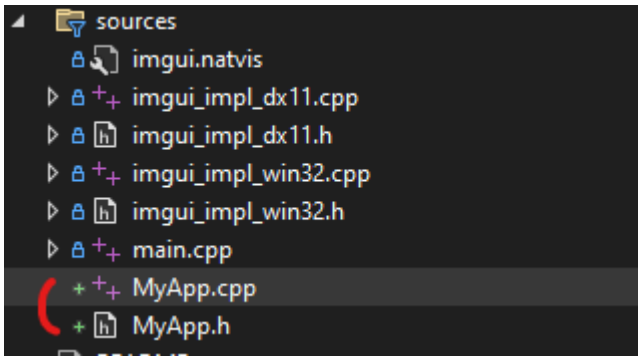


Okay this is cool, but what if we want to create our own app not this demo window ? How to get rid of the console that starts with our app ? How to use the full viewport to make a fullscreen window ? Let's see how !

First you should know that everything in the **imgui/** folder contains Dear ImGui files needed for all examples. It could also be used as a doc, if you search how to implement a feature you can browse the file to see how it is done in the demo window. In the **sources/** folder you will find only the ImGui files needed for the example you chose and a **main.cpp** creating the standalone example application for DirectX 11 using the Win32 api and ImGui.

Let's create our custom window

Create a new header and source file in the chosen example project. Right click on the **sources/** folder and click **Add > New Item** then create a .cpp and .h files and call them MyApp.



Open the header file *MyApp.h* and paste the following under `#pragma once`. We are creating a namespace and a procedure for rendering the UI.

```
namespace MyApp
{
    void Render();
}
```

In our source file *MyApp.cpp* let's create our UI, first paste the following to import and implement `Render()`

```
#include "MyApp.h"
#include "imgui.h"

void MyApp::Render()
{
    // Create our UI here
}
```

Note that every ImGui window are created using almost the same process. In your program loop you start the window by calling, `ImGui::Begin("WindowName")`. For each call to `ImGui::Begin()` you need a matching `ImGui::End()` when you finished creating your window.

For example:

```
// Create a window called "MyApp)" and append into it only if begin succeed
if (ImGui::Begin("MyApp"))
{
    // Create specific UI here
}
// Call a matching End() regardless of the return value of Begin().
ImGui::End();
```

Then let's add some widgets. Note that for every input value you will mostly need to use static variables. Copy the following in our `Render()` function.

```
// Static variables for input
static int counter = 0;
static bool isChecked = false;

if (ImGui::Begin("MyApp"))
{
    ImGui::Text("This is some useful text."); // Display some text (you can use a format
strings too)
    ImGui::Checkbox("CheckBox", &isChecked); // true/false state stored into isChecked

    if (ImGui::Button("Button")) // Buttons return true when clicked
        counter++; // Increase the counter
    ImGui::SameLine(); // Keep the next widgets on the same line as our
button
    ImGui::Text("counter = %d", counter);
}

ImGui::End();
```

Then in our **main.cpp** let's replace the demo window by ours ! Locate this line of code:

```
// 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);
```

Then delete everything from `if (show_demo_window)` to the second `ImGui::End()` before `ImGui::Render()`. You should have something like that:

```
...
ImGui::NewFrame();

// Our code goes here

// Rendering
ImGui::Render();
...
```


All we have to do now is to import `MyApp.h` at the top of our `main.cpp` by adding `#include "MyApp.h"` and replacing the lines we deleted by a call to our Render function.

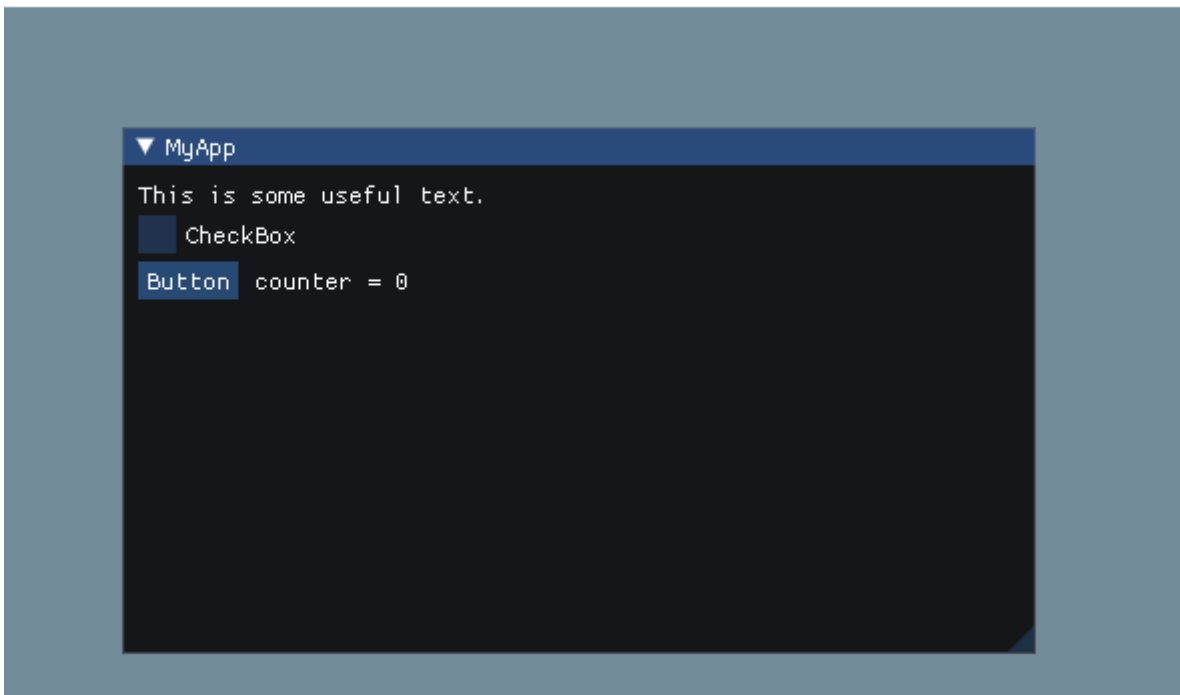
```
...
MyApp::Render();

// Rendering
ImGui::Render();
...
```

We could also delete the variable `bool show_demo_window` and `bool show_another_window` as we are not using them anymore.

Then let's run our app !

 Dear ImGui DirectX11 Example



Voilà we have a basic GUI app in C++ ! Now that you know how to create a basic app using ImGui you could create a new project and build your own ImGui app from scratch based on the example you choosed. As you saw ImGui is really easy to use, and you can do almost anything you want with it !

Covering the entire screen/viewport

If you want your app to fit with the viewport all you have to do is to pass some flags to `ImGui::Begin()` and set the position of our ImGui Window.

In our `Render()` function you could add the following and pass it to `ImGui::Begin()`. Note that depending on what you need you should use the full viewport or work area as ImGui explain in

comments.

```
static bool use_work_area = true;
static ImGuiWindowFlags flags = ImGuiWindowFlags_NoMove | ImGuiWindowFlags_NoResize |
ImGuiWindowFlags_NoCollapse;

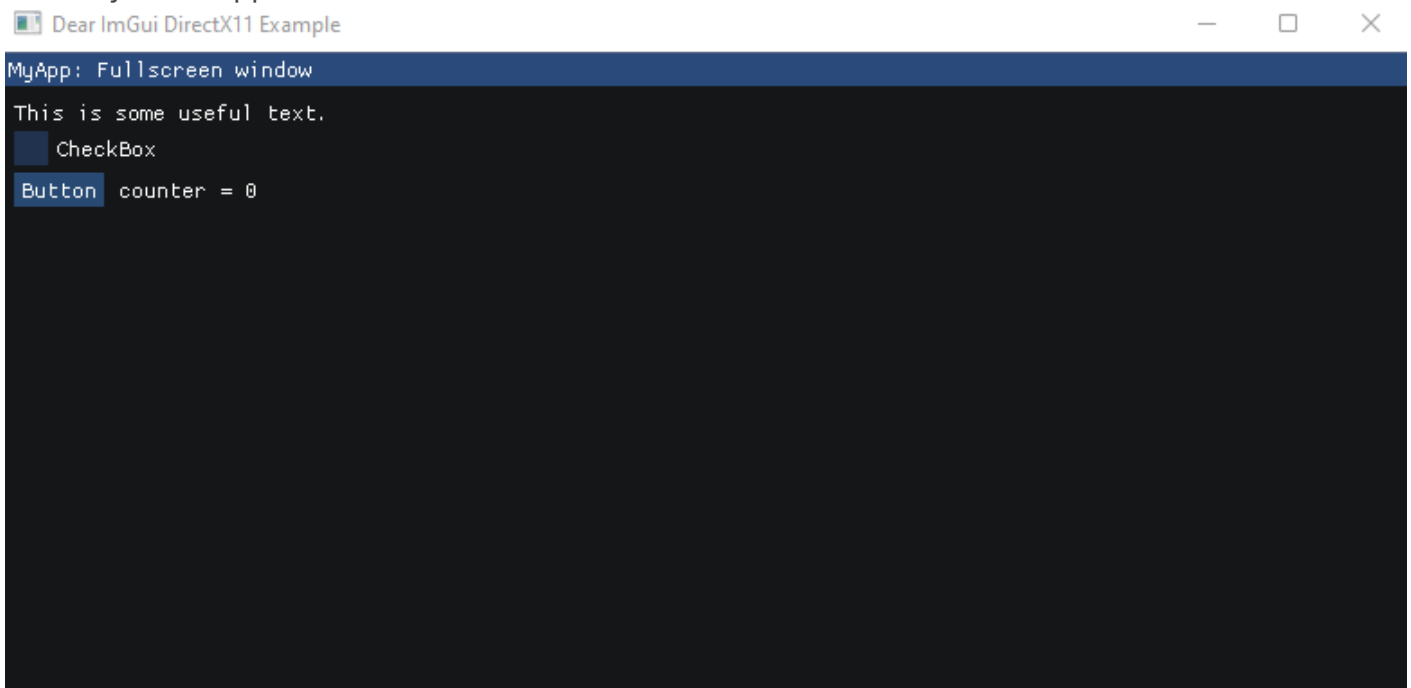
// We demonstrate using the full viewport area or the work area (without menu-bars, task-bars
etc.)
// Based on your use case you may want one of the other.
const ImGuiViewport* viewport = ImGui::GetMainViewport();
ImGui::SetNextWindowPos(use_work_area ? viewport->WorkPos : viewport->Pos);
ImGui::SetNextWindowSize(use_work_area ? viewport->WorkSize : viewport->Size);

...

if (ImGui::Begin("MyApp: Fullscreen window", nullptr, flags))
{
    ...
}
ImGui::End();
```

You could also change the size of the background window in `main.cpp` by editing the arguments passed to `CreateWindow()` by default ImGui examples are creating a window of 1280 x 800.

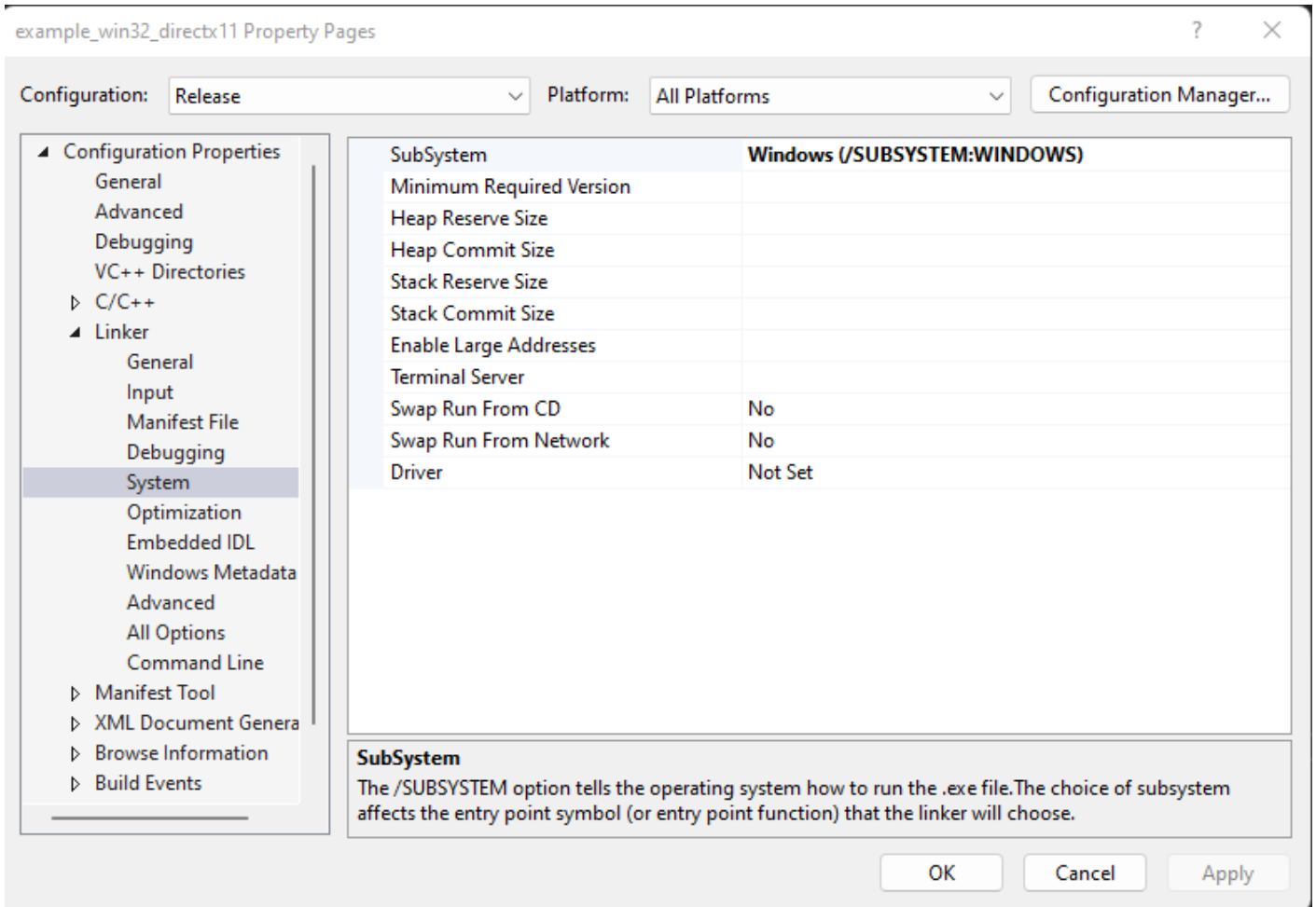
Here my basic app:



Getting rid of the console

There is nothing more easy than getting rid of the console ! For that open the **Project Properties**. (Right Click on the project then Properties). In **Configuration** choose **Release** and for **Platform**: **All Platforms**. The console won't appear when releasing our app but will still there in Debug mode.

Then go to **Linker > System** and choose `Windows (/SUBSYSTEM:WINDOWS)` under **SubSystem**



Then you need to replace the original `main` by `WinMain` as it is the conventional name used for the application entry point in Win apps. (<https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-winmain>)

In **main.cpp** replace `int main(int, char**)` line by the following:

```
int APIENTRY WinMain(_In_ HINSTANCE hInst, _In_opt_ HINSTANCE hInstPrev, _In_ PSTR cmdline, _In_ int cmdshow)
```

Then select Release mode and run your app ! The console is gone :D !