



▼



r/cpp



Search Reddit



Advertise



▼

r/cpp

Posts



Posted by u/pgroarke 5 years ago



429 A pretty big list of C++ GUI libraries



So I have to choose a UI library for my game engine. Since I won't use popular options, I went down a rabbit hole looking for libraries. I thought there weren't many GUI solutions for C++, I was quite wrong. The result is a list of 35 "healthy" C++ UI libraries.

I also ended up unearthing a bunch of old, dead UI libraries. It is quite entertaining to go back memory lane.

I've compiled the libraries [here](#) with screenshots, short descriptions and pros/cons I could quickly gather. Let me know if I missed anything (alive or dead)!

Here is the list of healthy projects I could find, Cheers.

- [Qt](#)
- [Sciter](#)
- [wxWidgets](#)
- [GTK+](#)
- [gtkmm](#)
- [CEGUI](#)
- [Dear ImGui](#)
- [Noesis GUI](#)
- [Juce](#)
- [Fox Toolkit](#)
- [MiniGUI](#)
- [Nuklear](#)
- [NanoGUI](#)
- [LittlevGL](#)
- [neoGFX](#)
- [morda](#)
- [U++](#)
- [dlib](#)



v



r/cpp



Advertise



- [libui](#)
- [Agar](#)
- [FLTK](#)
- [IUP](#)
- [Boost.UI](#)
- [LCUI](#)
- [GLUI](#)
- [TGUI](#)
- [ivtools](#)
- [GuiLite](#)
- [Ultralight](#)
- [Chromium Embedded Framework](#)
- [CopperSpice](#)
- [FlatUI](#)
- [SFGUI](#)
- [Lgi](#)
- [Verdigris](#)



187 Comments



Share

...



u/invented2020 · Promoted

Safe Deal: Your AI Shopping Assistant. Uncover the best prices and reviews on AliExpress, eBay, and Amazon. Shop smart and secure!
Join the conversation

joinsafedeal.com

Install

Comment as [myNameIsMe1981](#)

Markdown Mode

Comment

//



▼



r/cpp



Advertise



jherico · 5 yr. ago

This list would be more useful if it specified supported operating systems, whether or not they support direct GPU rendering and supported GPU APIs

55 ...

thebruce87m · 5 yr. ago

Licensing would be interesting too for us embedded folk.

43 ...

pgroarke OP · 5 yr. ago

note taken

11 ...

mikeblas · 5 yr. ago

The [Wikipedia list](#) is smaller, but has that information for the libraries it does include.

13 ...

pgroarke OP · 5 yr. ago

TBH, most of them are compatible with Windows, Mac, Linux. I'll consider it for a future iteration, I'm a bit burnt right now.

10 ...

jherico · 5 yr. ago

iOS and Android compatibility are important as well. I use the Qt QML system for rendering UI for compositing into 3D VR scenes and its support for Android is critical for me.

9 ...

Pieralinovski · 5 yr. ago

Good job! Very interesting!

15 ...

pgroarke OP · 5 yr. ago



Advertise



mayumichan · 5 yr. ago

I would list Qt widgets and Qt quick as 2 different GUI frameworks, even though both are part of the Qt project. Both are great in my opinion. Qt quick requires learning QML a declarative markup language with JS thrown in the mix.

14 Reply Share ...

disperso · 5 yr. ago

I would treat Widgets and Quick differently as well. Note that nowadays you can also use Qt Quick with QSkinny and QNanoPainter, which would be in C++ only. Those projects are third party, and might not be considered as stable as Qt Quick Controls.

5 Reply Share ...

kirbyfan64sos · 5 yr. ago

I really like the idea and the amount of research you've put into this, but some of the cons (and a few pros) can seem really...superficial, like:

- Sciter actually uses a JS derivative language as part of the UI; it's not entirely C++.
- GTK's website screenshots are super old, and GIMP actually uses an older version (it doesn't even really stand for GIMP ToolKit anymore). It supports CSS for theming the UI and can be made to look really gorgeous (IMO the new version of the Adwaita theme looks pretty nice OOTB too.) IME it's not that hard to build for as well, and you'll primarily be using prebuilt binaries (e.g. from your distro or MSYS2). Also, Glade can be used on plain GTK, not just gtkmm.
- Again gtkmm would be primarily prebuilt or automatically built versions, so the autotools shouldn't have much of an impact.
- CEGUI...actually I have nothing to say here, building that crud is nightmare fuel.
- nuklear is actually a bit odd to use IME; it's by no means terrible but definitely requires quite a bit of glue, especially if you want to use system fonts. God forbid you want to use system fonts...
- If you're including NanoVG as a gui library you might as well include Skia and Cairo...
- "doesn't use cmake" isn't necessarily a con. Also again, you're rarely going to be building morda from source...
- Seems a bit unfair to put "not battle tested" under Nana when there are sever others here that don't seem to be used much more.
- "requires SFML", not really that big of a dependency...



▼



r/cpp

Comment



Advertise



A few more things to note:

- Few of these likely support Wayland, sadly.
- Not all of these have OOTB system font support.

29 ...

pgroarke **OP** · 5 yr. ago

There was an attempt at humor ;)

I was split with NanoVG, but yes I think I'll remove it.

"doesn't use cmake" means they hardcoded all platforms IDE projects, which is 100% code smell in my book. I'll bet you one of those projects is out-of-sync.

"requires SFML" is a big deal as it is an OpenGL context loader, and others may use SDL/GLFW to load the context. It means you can't use it in that case.

The post isn't meant to be super serious research, and its certainly not trying to be "objective". Just what I could spot looking at their project in ~10 minutes. If you do have experience with some libraries and have extra pros/cons, let me know! I will happily add info. Good day :)

6 ...

c-smile · 5 yr. ago · *edited 5 yr. ago*

[Sciter](#) has compact and stable plain C API that allows to do UI without scripting at all.

Scripting in Sciter can be treated as a DSL that defines interaction of UI layer (events) with native backend. Pretty much in the same way as HTML is a DSL for defining DOM tree and CSS as language that defines how that tree needs to be rendered.

UI in general has pretty complex ownership graph that usually contain loops so scripting and its GC helps quite a lot here.

Native backend (code) is more regular in this respect - ownership is trivial and `std::shared_ptr<T>/std::unique_ptr<T>` are pretty adequate in most of cases.

3 ...

Tobblo · 5 yr. ago

amount of research



▼



r/cpp



Advertise

[Up](#) -6 [Down](#) [Reply](#) [Share](#) ...

itsarabbit · 5 yr. ago

OP never claimed that they were being purely factual, this is simply a list with GUI libraries and short notes from him.

But yes, "ugly af" and "smells like build hell" does imply that the author has done research. It's hard to be more than superficial when you are looking at 35 completely different implementations of the same or similar feature set.

[Up](#) 22 [Down](#) [Reply](#) [Share](#) ...

pgroarke OP · 5 yr. ago

I would like to give you the "understands internet" trophy today, cheers :)

[Up](#) 12 [Down](#) [Reply](#) [Share](#) ...

Comment removed by moderator · 5 yr. ago

AutoModerator MOD · 5 yr. ago

Your comment has been automatically removed because it appears to contain disrespectful profanity or racial slurs. Please be respectful of your fellow redditors.

If you think your post should not have been removed, please [message the moderators](#) and we'll review it.

I am a bot, and this action was performed automatically. Please [contact the moderators of this subreddit](#) if you have any questions or concerns.

[Up](#) 2 [Down](#) [Reply](#) [Share](#) ...

lithium · 5 yr. ago

JUCE is incorrectly listed as closed source.

[Up](#) 9 [Down](#) [Reply](#) [Share](#) ...

pgroarke OP · 5 yr. ago

It isn't? My bad, will update. Thank you!

[Up](#) 3 [Down](#) [Reply](#) [Share](#) ...



v



r/cpp



Advertise



What the...? No ImGui? @ [/u/pgroarke](#)

ImGui (for prototypes and debug UIs) and Qt (for full-fledged GUI applications) are pretty much the only GUI libraries that matter in C++, IMHO.

EDIT: I'm blind.

28 Reply Share ...

pgroarke OP · 5 yr. ago

Of course ImGui! It is listed as Dear ImGui, its real name ;) Right there at the top.

25 Reply Share ...

bikki420 · 5 yr. ago

Ohh, my bad! I completely missed it.

9 Reply Share ...

DustVoice · 5 yr. ago

For me and my clients JUCE is perfect. You definitely have to get used to it and it doesn't have the shallowest learning curve but neither has Qt! JUCE is a great framework with a nice and flexible developer team that help you solve your bugs and problems. The community is great too and I have to say mobile development isn't of course as easy as developing in the native language of iOS/Android but it's truly cross-platform!

7 Reply Share ...

ComposerShield · 5 yr. ago

Not the only ones, JUCE is great!

5 Reply Share ...

FlibbleMr · 5 yr. ago

I plan on destroying the current "least worst" option (Qt) with neoGFX. :D

<https://neogfx.org>



▼



r/cpp



Advertise



bikki420 · 5 yr. ago

Nice, keep it up!

I hope those disgusting Vista-esque gradients are 100% optional though. :p

3 ...

FlibbleMr · 5 yr. ago

I like gradients but yes there will be a flat skin too.

1 ...

bikki420 · 5 yr. ago

Nice.

How does it perform on Linux, Android, macOS, and iOS?

1 ...

FlibbleMr · 5 yr. ago

It should perform the same no matter what the OS.

1 ...

bandzaw · 5 yr. ago

Looks interesting. Too bad it is licensed under GPLv3 though since that will not work for me...

4 ...

FlibbleMr · 5 yr. ago

When 1.0 is released I will move to a dual LGPL/commercial license.

3 ...

fundorin · 5 yr. ago

ImGui

Not a good sign when the test app crashes immediately -

<https://i.imgur.com/EeIhXp6.png>

4 ...



▼



r/cpp



Advertise



Next time read the README. It tells you what to do (in this case install OpenSSL).

3 Reply Share ...

TankorSmash · 5 yr. ago

I think there's a better way to respond to someone when they're reporting a crash, even if it is something they could have read in the docs.

5 Reply Share ...

fundorin · 5 yr. ago

I'd rather find a better, less dependent GUI library.

Good luck with your work, anyway.

6 Reply Share ...

FlibbleMr · 5 yr. ago

The OpenSSL dependency is there due to Boost being dependent on it. It can be built without that dependency though. Thanks for letting me know that you can find something "better". :/

1 Reply Share ...

meneldal2 · 5 yr. ago

Is there any reason why a GUI would end up needing to transitively include OpenSSL?

6 Reply Share ...

ppetraki · 5 yr. ago

```
ppetraki@vanguard:~/Sandbox/krillia/build-debug$ conan sea
Existing package recipes:
```

```
imgui/1.53@bincrafters/stable
imgui/1.61@bincrafters/stable
imgui/1.62@bincrafters/stable
imgui/1.63@bincrafters/stable
imgui/1.64@bincrafters/stable
imgui/1.65@bincrafters/stable
imgui/1.66@bincrafters/stable
ppetraki@vanguard:~/Sandbox/krillia/build-debug$ conan ins
Configuration:
```



▼



r/cpp



Advertise



```
os_name=Linux  
arch=x86_64  
arch_build=x86_64  
compiler=gcc  
compiler.version=7  
compiler.libcxx=libstdc++11  
build_type=Release  
[options]  
[build_requires]  
[env]  
  
imgui/1.66@bincrafters/stable: Not found in local cache, looking in remote  
imgui/1.66@bincrafters/stable: Trying with 'conan-center'.  
  Downloading conanmanifest.txt  
[=====] 166B/  
  Downloading conanfile.py  
[=====] 1.7KB/  
  Downloading conan_export.tgz  
[=====] 764B/  
  Decompressing conan_export.tgz: 772B [00:00, 214kB/s]  
  imgui/1.66@bincrafters/stable: Installing package  
    Requirements  
      imgui/1.66@bincrafters/stable from 'conan-center' - Do  
    Packages  
      imgui/1.66@bincrafters/stable:66c5327ebdcecae0a01a8639:  
  
  imgui/1.66@bincrafters/stable: Retrieving package 66c5327e  
  Downloading conanmanifest.txt  
[=====] 3.0KB/  
  Downloading conaninfo.txt  
[=====] 456B/  
  Downloading conan_package.tgz  
[=====] 705.4KB/  
  Decompressing conan_package.tgz: 689kB [00:00, 11.0MB/s]  
  imgui/1.66@bincrafters/stable: Package installed 66c5327ebdcecae0a01a8639
```

↑ 0 ↓ Reply Share ...

pgroarke OP · 5 yr. ago

I recommend using conan for your dependencies, it makes all these sort of problems go away ;)

↑ 2 ↓ Reply Share ...



▼



r/cpp



Advertise



A random thought I've had going around my head for a while - and be aware I don't really know what I'm talking about here - would it be possible to pull out the C++ classes Chromium uses to represent elements on a web-page, and then expose a C++ way to connect those up into a UI?

That could be a fascinating way to develop cross-platform apps.

(Feel free to rip this idea to shreds. I've never looked at the Chromium code)

5 [Reply](#) [Share](#) ...

hopfield · 5 yr. ago

So you want to manipulate an HTML DOM in C++? Why not just use Electron at that point?

7 [Reply](#) [Share](#) ...

spinicist · 5 yr. ago

That's not quite what I'm imagining. I think. I don't really know.

I think what I'm imagining is using the cross-platform rendering capabilities of Electron, but being able to do that from C++ with a standard set of widgets to avoid the Javascript layer. But this is probably based on a fundamental misunderstanding (on my part) of how Chromium actually works.

Allow me to go off topic for a minute - I can see a future where Electron becomes a default cross-platform app engine. I'm not saying I will like this future, I'm saying it's a clear possibility. In fact I take Microsoft dropping Edge and rebuilding it with Chromium as a big push in this direction. MS already have web versions of all their main products - why not go the whole hog and drop the native versions and bundle Electron versions instead? They could take it one step further and have a shared instance of Chromium on Windows that ran all of Office to avoid some of the bloat.

I'm wondering how we can avoid that future and the many wasted CPU cycles it will entail.

2 [Reply](#) [Share](#) ...

hun_nemethpeter · 5 yr. ago

Look at this presentation: [Life of a Pixel by Steve Kobes](#)

There is a problem with chromium although. It uses a Garbage Collector internally called Oilpan for DOM nodes as those objects are exported out for JavaScript which doesn't do manual memory management.

[Blink GC API reference](#)



v



r/cpp



Advertise

<https://skia.org/>

You can reuse the layout engine from Chromium also.

https://github.com/chromium/chromium/tree/master/third_party/blink/renderer/core/layout/ng

4 Reply Share ...

spinicist · 5 yr. ago

Thanks, that's all really interesting.

1 Reply Share ...

c-smile · 5 yr. ago

to pull out the C++ classes Chromium uses to represent elements on a web-page

[Sciter](#) does that: [sciter::dom::element](#).

1 Reply Share ...

khaki0 · 5 yr. ago

I've been using Qt for about 6 months and so far am pretty happy with it. Just curious, what do you mean by Qt uses "ancient paradigms" and is "slow"?

19 Reply Share ...

airflow_matt · 5 yr. ago

QtWidgets is slow. Unfortunately. The content of entire window is rendered in software (entire backing store framebuffer is in RAM) and then pushed to video memory. On Windows, this is fine, however on OS X pushing pixels from system memory to video memory is quite slow. Just try scrolling in Qt Creator on a 4K screen, you get nowhere near 60fps. It feels rather sluggish. There was an improvement about a year ago where unnecessary color conversion was removed when blitting those pixels, but it's still quite slow.

Qt Quick Scene graph on the other hand is fully accelerated, but functionality wise Qt Quick really lags behind Qt widgets for developing desktop applications.

13 Reply Share ...

Adequat91 · 5 yr. ago

on OS X pushing pixels from system memory to video memory is quite slow



time graphics on 5K screens (about 20 fps) with no problem (something I thought would never be possible). However, this is with a good Qt expertise and experience of writing fast code. This being said, this might not be enough for a game app. And secondly, GPU graphics are obviously faster in any case. But this is also possible with Qt.

4 Reply Share ...

airflow_matt · 5 yr. ago

But 20fps on 5K screen *is* slow. Anything cocoa, Qt scene graph or even electron based will do 60fps without breaking a sweat. That's night and day difference.

6 Reply Share ...

Adequat91 · 5 yr. ago

Yes 20 fps is slow compared to 60 fps. I just want to mention that widget painting is as fast on Mac as on Windows, nowadays. And that 20 fps on a 5K screen is decent anyway, especially when you consider that each pixel is painted individually by the CPU (but SIMD used a lot).

2 Reply Share ...

airflow_matt · 5 yr. ago · *edited 5 yr. ago*

The code for widget painting is exactly same on Windows as it is on Mac, and so is the performance. Which is generally quite good.

But in my experience the bottleneck is flushing the backing store. In OS X this involves either creating CGImageRef from backing store and updating CALayer with it, or updating opengl texture with new backing store content. Both of these unfortunately take lot of time. Even if you have empty window with no widgets, you still won't be able to do this at 60FPS on a 4K display.

In my experience this is still significantly faster on Windows, where the backing store simply does BitBlt, which seems way better suited for this use case.

3 Reply Share ...

Adequat91 · 5 yr. ago

But in my experience the bottleneck is flushing the backing store



▼



r/cpp



Advertise



least a 2x or 3x time speed up.

1 Reply Share ...

airflow_matt · 5 yr. ago

I mentioned that in my very first comment. It was improved by removing [color space conversion](#). Even with this it is still slower than Windows.

1 Reply Share ...

khaki0 · 5 yr. ago

Agree. Btw have you used Qt Quick? It's obviously more fluid but in my limited experience it eats a ton of RAM.

2 Reply Share ...

krum · 5 yr. ago

I'm not a huge Qt fan myself, but it's not slow by any means.

17 Reply Share ...

pgroarke OP · 5 yr. ago

Uses inheritance instead of composition. Slow compared to GPU rendered UIs, especially if you need custom widgets (for example bezier curve editors). You probably will never hit that point in non-intensive applications however.

1 Reply Share ...

jonesmz · 5 yr. ago

What's wrong with inheritance?

Qt has built in opengl handling, you seem to be implying that it doesn't support GPU rendering at all?

22 Reply Share ...



Heuristics · 5 yr. ago



Comment deleted by user · 5 yr. ago

peppedx · 5 yr. ago



▼



r/cpp



Advertise



↑ 3 ↓ Reply Share ...

pgroarke OP · 5 yr. ago

Reading the docs, it seems only QML uses their GPU scene graph.

QPainter

The QPainter engine uses software to paint, and is used when drawing Qimages or Qwidgets. Its advantage over OpenGL is the high quality when antialiasing is enabled, and a complete feature set.

The QPainter can use an OpenGL engine, but as the document mentions it is more susceptible to state changes. And has to be used carefully.

AKA their OpenGL implementation isn't up to par ;)

Now, I don't keep up with Qt news. Last time I tried using their painter, perf degraded after 3 bezier curves with circles and squares. Maybe things are different now.

↑ 3 ↓ Reply Share ...

peppedx · 5 yr. ago

Qml qt quick is more or less the preferred way of writing a qt gui nowadays. Anyway I dont usually do bezier guis :D Ps have a look at this project...
<https://github.com/uwerat/qskinnny>

↑ 1 ↓ Reply Share ...

t3685 · 5 yr. ago · edited 5 yr. ago

You might want to have a look at QOpenGL* related classes ;)

↑ 1 ↓ Reply Share ...

jcelerier · 5 yr. ago

ossia score



Comment deleted by user · 5 yr. ago

Red-Portal · 5 yr. ago



v



r/cpp



Advertise



though.

[↑ 8](#) [↓](#) [Reply](#) [Share](#) ...

ShillingAintEZ · 5 yr. ago

A big downside of Nana is that it uses some sort kind of hacked together text markup for layout. It is hard to get things to work right and hard to debug.

[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...

Red-Portal · 5 yr. ago

Some of my colleagues actually like nana because of that markup. I guess it's about preference to some degree?

[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...

ShillingAintEZ · 5 yr. ago

I can see why someone would say that initially. At first I wondered if it was better, worse, or just different. The fundamental aspect though, is that an invented markup is the worst of all worlds.

You can't see or affect any sort of state or layout data directly. If you could pass a data structure to it or use some function calls it would be much different, but because it is text that gets embedded into a program, it just creates an opaque layer between the program and the layout for no reason.

[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...

mjkclaim · 5 yr. ago

QuantStack/mamba + gamedev/embededd/robotics

A few issues:

- Some of these GUI libraries do not have pros listed, when they do have pros (Dear ImGui isn't perfect, although it's good) which immediately smells like it's a very opinionated list, which is not useful for people who look for tools by comparing with their context see if the pros are acceptable.
- Immediate mode gui libraries and the others should be separated, they imply a widely different way to code.
- Some of the libraries only work inside your own display system (like Dear ImGui) which is another massively different category.
- The "ugly af" comments are not helpful at all. Same for most of the comments you make on the libraries you didn't try.



▼



r/cpp



Advertise



- How "javascript" is an issue for tools that work with HTML/CSS/JS???
- The build system these libraries use is not a problem, except if you must tweak it.

8 Reply Share ...

feherfarkas · 5 yr. ago

Nice list. Would be more enjoyable without the "af", and "cancer".

15 Reply Share ...

pgroarke OP · 5 yr. ago

Well you can't have your cake and eat it too XD

-3 Reply Share ...

feherfarkas · 5 yr. ago

This is where you are mistaken. It's not about my cake. People won't judge me based on your style. They will judge you.

4 Reply Share ...

pgroarke OP · 5 yr. ago

It is your right to judge me ;)

5 Reply Share ...

feherfarkas · 5 yr. ago

Nowhere did I say I judge you. I gave you feedback. You can think about it or be pissed about it. It's your call.

3 Reply Share ...

mkngry · 5 yr. ago

<http://wtl.sourceforge.net/>

3 Reply Share ...

johannes1971 · 5 yr. ago

It seems you have listed every non-dead, bright-colored GUI as "not pretty". If that didn't happen by accident, I invite you to do an image search for the phrases "cool GUI" (mostly



v



r/cpp



Advertise



pgroarke OP · 5 yr. ago

valid criticism, fixed

[Up ↑ 1](#) [Down ↓](#) [Reply](#) [Share](#) [...](#)

IloveReddit84 · 5 yr. ago

CopperSpice is literally a fork of Qt 4.8, but with less features (I remember qmoc wasn't still possible)

[Up ↑ 3](#) [Down ↓](#) [Reply](#) [Share](#) [...](#)

Red-Portal · 5 yr. ago

The whole point of copperspice was to ditch qmoc.

[Up ↑ 9](#) [Down ↓](#) [Reply](#) [Share](#) [...](#)

FonderPrism · 5 yr. ago

I've compiled the libraries here with screenshots, short descriptions and pros/cons I could quickly gather. Let me know if I missed anything (alive or dead)!

This is great, I've been looking for something like this. Finding examples/screenshots of GUI's can be hard sometimes.

[Up ↑ 3](#) [Down ↓](#) [Reply](#) [Share](#) [...](#)acmd · 5 yr. ago · *edited 5 yr. ago*

Nice! Probably many of us had done a similar research sometime but haven't published it. IMO, trying to do/emulate native look in a cross-platform UI library is a bit of wasted effort nowadays, since users seem to not care about it and happily use bloated electron-based apps. Web and desktop interfaces are converging, whether we like it or not, so we might as well use anything decent.

Also [I've tried](#) using [Windows Composition Engine](#) to get multiple Dear ImGui windows drawn on a transparent background with GPU acceleration, but the only blending mode that Windows supports is [DXGI_ALPHA_MODE_PREMULTIPLIED](#), so we are out of luck(or I'm missing something). There's an easier option to just create titleless and borderless window if your software doesn't need to have multiple windows simultaneously and you want to use render-to-texture library.



▼



r/cpp

Comment



Advertise



c-smile · 5 yr. ago

Not clear what you are trying to achieve.

DirectComposition/Direct2D + WS_EX_NOREDIRECTIONBITMAP can give you transparent windows. Yet also [the Acrylic](#) if needed.

1 [Reply](#) [Share](#)

acmd · 5 yr. ago

Yes, but the blending mode is incorrect. [That's what it draws.](#)

1 [Reply](#) [Share](#)

c-smile · 5 yr. ago

Not sure I understand what do you mean by blending mode there.

It simply draws [dst-over](#)

Instead of white notepad window put there some window with black content.

1 [Reply](#) [Share](#)

acmd · 5 yr. ago · *edited 5 yr. ago*

Please refer to the source link in my top comment. I'm using [WS_EX_NOREDIRECTIONBITMAP](#), but then I [CreateSwapChainForComposition](#) which could only use [DXGI_ALPHA_MODE_PREMULTIPLIED](#). [Here's another link](#) showing the problem. Even though I'm setting #0074FFFF for FrameBg, it still shows underlying visual studio window. So I guess I need another API for that. Layered windows maybe?

1 [Reply](#) [Share](#)

c-smile · 5 yr. ago

[DXGI_ALPHA_MODE_PREMULTIPLIED](#) has absolutely nothing with blend modes. It just defines [format of alpha values](#) in RGBA representation.

1 [Reply](#) [Share](#)

acmd · 5 yr. ago

Alright, thanks for the info. I still don't understand why my fully opaque black text is invisible over white notepad window. Can you explain why is that the case?



v



r/cpp



Advertise



dacap · 5 yr. ago

I would like to add my own dead project: [Vaca \(screenshots\)](#)

3



Reply Share ...

airflow_matt · 5 yr. ago

> I would like to add my own dead project

This brings back some memories. We've used Vaca for a Win32 GUI of a small project few years ago. It was abandoned even back then. We end up internally forking it and fixing some bugs down the way, but the result wasn't bad. It was very lean and as far as winapi wrappers go, I have to say that Vaca was very, very sane and nice to work with :)



3



Reply

Share

...
...

dacap · 5 yr. ago

Glad to hear that it was useful for someone else out there ^-^



1



Reply

Share

...
...

pgroarke OP · 5 yr. ago

roger roger



1



Reply

Share

...
...

cleroth · 5 yr. ago

Game Developer

I quite like [turbobadger](#). The only problem I find is it doesn't have much documentation, so you'll have to look at the examples, mostly.



2



Reply

Share

...
...

pgroarke OP · 5 yr. ago

It looks very decent. I listed it in the dead category since it hasn't been updated in years though.



2



Reply

Share

...
...

cleroth · 5 yr. ago

Game Developer

Also [Ultralight](#), though licensing is still unclear.



▼



r/cpp



Advertise



pgroarke OP · 5 yr. ago

Yep, I put it at the bottom since they don't have screenshots :)

[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...

idea4good · 5 yr. ago

How about the smallest(5 KLOC) GUI library -- [GuiLite](#)[↑ 2](#) [↓](#) [Reply](#) [Share](#) ...

pgroarke OP · 5 yr. ago

ty! Adding it :)

[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...[deleted] · 5 yr. ago · *edited 5 yr. ago*

libui seems to be missing, although I'm not sure how active the author still is, the previous update was in september I believe

Edit: I just noticed the last commit on [libui](#) was on April 1st, so it's still active[↑ 2](#) [↓](#) [Reply](#) [Share](#) ...

pgroarke OP · 5 yr. ago

:+1:

[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...

the_real_m0radin · 5 yr. ago

Maybe Slate? The GUI lib of UE4

[↑ 2](#) [↓](#) [Reply](#) [Share](#) ...

lsc4719 · 5 yr. ago

(And nobody use)

[↑ 2](#) [↓](#) [Reply](#) [Share](#) ...



v



r/cpp



Advertise

What about [Verdigris](#)?[Up](#) 2 [Down](#) [Reply](#) [Share](#) ...

pgroarke OP · 5 yr. ago

will add, thx!

[Up](#) 2 [Down](#) [Reply](#) [Share](#) ...

msadeqhe · 5 yr. ago

thx, also there is [cppgir](#) which is a tool to generate GTK binding.[Up](#) 1 [Down](#) [Reply](#) [Share](#) ...

tabana_minamoto · 5 yr. ago

Great work!

I have one that I'm not 100% sure it qualifies. There's the Slate UI Framework <https://docs.unrealengine.com/en-us/Programming/Slate> In theory you don't need the entire Unreal engine to make an application with the GUI framework, but I don't really know why someone would do that, beside for the Unreal Editor.

[Up](#) 2 [Down](#) [Reply](#) [Share](#) ...pgroarke OP · 5 yr. ago · *edited 5 yr. ago*

Hmmm, good question. I'll think about it, but I'm leaning towards not including it as it would require a lot of work.

edit: Someone just mentioned libui, it looks very native as well.

[Up](#) 2 [Down](#) [Reply](#) [Share](#) ...

dvereb · 5 yr. ago

I was so excited to read what you wrote about Fox-Toolkit since that's what we use where I work. The cons list sums it up just about perfectly, lol!

[Up](#) 2 [Down](#) [Reply](#) [Share](#) ...

pgroarke OP · 5 yr. ago

XD

[Up](#) 2 [Down](#) [Reply](#) [Share](#) ...



v



r/cpp



Advertise



screeches about gtkmm documentation and the gnome foundation for 20 minutes

2 Reply Share ...

Dean_Roddey · 5 yr. ago

Charmed Quark Systems

Not that anyone is likely to use it, but mine includes amongst many other things a pretty extensive UI framework, though it only supports Win32 at this point.

<https://github.com/DeanRoddey/CIDLlib>

2 Reply Share ...

pgroarke OP · 5 yr. ago

Will add, thank you :)

edit : Oh, missed the Windows-only part. Let me know if it ever migrates to another OS.

1 Reply Share ...

Obloha · 4 yr. ago

FireMonkey is in my opinion best GUI for C++ if you want custom looking GUI. It is comparable with WPF for Windows, but F.M. is crossplatform. The big downside for me is, FM is only for Borland RAD Studio. The best alternative to FM is Noesis GUI (you have it on list) which is based on WPF. It uses XAML, you can build GUI inside VS Blend and use it in Noesis. But Noesis is good for game ui and maybe simple applications. It have some window system but not multiwindow (no more then one window), or you must use external window system like GLFW or SDL.

Btw. Nice list, I know almost all (when I was looking for C++ gui).

2 Reply Share ...

Misaka10782 · 2 yr. ago

cross-platform is actually a beautiful illusion, the true way to support different platforms is to divide your technical team into different groups and code in each platform's official tools 😊

2 Reply Share ...



v



r/cpp



Advertise



Great work. I really dig your summary of Qt as it is pretty much exactly what I feel.

As I am currently looking for a replacement of Qt, may I ask, which lib would you recommend that:

- supports HighDPI displays
- looks 'native' on Windows and Linux
- goes well with CMake
- BSD compatible license
- has an editor
- modern C++

3 [Reply](#) [Share](#) ...

pgroarke **OP** · 5 yr. ago · *edited 5 yr. ago*

If you are going for a native look, I think you would want to look at wxWidgets or boost.ui. Let me know if there is any other GUI lib with that focus in mind.

edit : libui was just mentioned, and it also looks native

3 [Reply](#) [Share](#) ...

Moose2342 · 5 yr. ago

That's what I figured after reading the post too. Boost.UI actually does look nice but the sparse docs and the fact that they are not 'ready' set me off a little. Also, I don't like how they call themselves a Boost lib which they are not. WxWidgets have been around for a long time and that's something to look at for sure.

2 [Reply](#) [Share](#) ...

RotsiserMho · 5 yr. ago

C++20 Desktop app developer

"Boost.UI" just wraps WxWidgets anyway. It's unfortunate that the author implies it's a Boost library when it's not.

1 [Reply](#) [Share](#) ...

c-smile · 5 yr. ago

Try my [Sciter](#), it uses H/W accelerated graphics on all platforms (DirectX and OpenGL) so it is pretty good on HighDPI monitors.

You can try [Sciter Notes](#) and [HTML-NOTEPAD](#) applications to get an idea of what can be achieved with Sciter.



▼



r/cpp



Advertise



DarkLordAzrael · 5 yr. ago

So, which of those requirements does Qt not meet? Why are you looking to replace it?

[↑ 1](#) [↓](#) [Reply](#) [Share](#) [...](#)ProRochie · 5 yr. ago · *edited 5 yr. ago*

windows.h or gtfo

jokes, I was searching google for something similar yesterday so thanks for this

[↑ 2](#) [↓](#) [Reply](#) [Share](#) [...](#)

awson · 5 yr. ago

OWLNext is missed.[↑ 1](#) [↓](#) [Reply](#) [Share](#) [...](#)

pgroarke OP · 5 yr. ago

ah it is Windows only, I don't list single platform libs unless they are dead.

[↑ 1](#) [↓](#) [Reply](#) [Share](#) [...](#)

Comment deleted by user · 5 yr. ago

kalmoc · 5 yr. ago

Did you have a look at cinder: <https://github.com/cinder/Cinder>?[↑ 1](#) [↓](#) [Reply](#) [Share](#) [...](#)

pgroarke OP · 5 yr. ago

I did, but I don't think you can extract their GUI lib to work standalone? Let me know if that's the case.

[↑ 1](#) [↓](#) [Reply](#) [Share](#) [...](#)

kalmoc · 5 yr. ago

What do you mean by stand alone in that context? I haven't used cinder for anything significant yet. I just was curious if you excluded it on purpose and if so why.



v



r/cpp



Advertise



pgroarke OP · 5 yr. ago

Well it isn't a GUI library, it is a multimedia/2d/3d library, no?

[Up](#) 1 [Down](#) [Reply](#) [Share](#) ...

kalmoc · 5 yr. ago

Yes, but you can just use the graphics part. But now that you mention it, I don't think it provides buttons and stuff like that out of the box

[Up](#) 2 [Down](#) [Reply](#) [Share](#) ...

Lastrevio · 5 yr. ago

Which one is the easiest to learn?

[Up](#) 1 [Down](#) [Reply](#) [Share](#) ...

DarkLordAzrael · 5 yr. ago

Qt is both very popular (despite people on this subreddit seeming to have something against it) and has great documentation and examples.

[Up](#) 5 [Down](#) [Reply](#) [Share](#) ...

pgroarke OP · 5 yr. ago

I'll guess Qt, simply because it is the most popular. Dear ImGui is also quite simple and fast to setup.

[Up](#) 2 [Down](#) [Reply](#) [Share](#) ...

Lastrevio · 5 yr. ago

do you necessarily need to learn a framework to develop a simple basic GUI for your C++ code like for example a tic tac toe game?

[Up](#) 2 [Down](#) [Reply](#) [Share](#) ...

pgroarke OP · 5 yr. ago

yep pretty much

[Up](#) 1 [Down](#) [Reply](#) [Share](#) ...

Lastrevio · 5 yr. ago

Rip



▼



r/cpp

Comment



Advertise



Lastrevio · 5 yr. ago

Isn't SDL2 a framework? If not then what do you call it?

1 ...

electric_machinery · 5 yr. ago

SDL2 is the simple directmedia layer, it doesn't have GUI functionality built in, but you could implement those features on top.

1 ...

Comment deleted by user · 5 yr. ago

crazdave · 5 yr. ago

I'm currently making a windows-only C++ service as a side project, and it needs a very simple window just to change some settings, anyone have a good suggestion of which library to use for this? Are any of them able to use with the standard createwindow/windowproc, or do they usually make their own window?

1 ...

pgroarke OP · 5 yr. ago

imgui is great for prototypes imho

2 ...

barfyus · 5 yr. ago

I've used [winlamb](#) for the same task (NT service configuration utility).

2 ...

Comment deleted by user · 5 yr. ago

cereagni · 5 yr. ago

Wow, this is amazing!



▼



r/cpp



Advertise



It seems like starting with Qt is not a bad option, as it is battle tested, mature and many programmers use it, but I'm a little overwhelmed by what seems to be the many "sub facilities" it has to offer (QtWidgets, QtQuick, QML...) Does anyone have a link to some summary of these different terms and/or a link to a tutorial with current-best-practices?

1 ...

onzelin · 5 yr. ago

QuickGUI

...

Cons

- mentions Ogre3D
- hosted by Ogre3D
- used with Ogre3D
- did I mention Ogre3D?

Do you have arguments against Ogre3D? (I mean, other than I don't like it)

1 ...

[deleted] · 5 yr. ago

Thanks for having set up that list. There is one point which I'm missing in the assessments of all the listed GUIs: DOCUMENTATION!! The more extensive a library is and the more cross-dependencies among its components it has the more important documentation becomes. There are huge differences between the libraries - consider e.g. these two extremes: nana vs FLTK.

1 ...

kamelEddine · 5 yr. ago

well idk for me i used Qt for long time very mature not that heavy if you install only the modules you need and if you didn't find any gui system that suit you you can create your's using SDL well god help you with that

1 ...



mercere99 · 5 yr. ago



v



r/cpp



Advertise



Actually_dumb_nerd · 2 yr. ago

Appreciate the list, some comments gave me a good laugh, thanks :L

[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...

Rogoreg · 2 yr. ago

You're missing one: MFC

[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...

Active-Tonight-7944 · 2 yr. ago

[AntTweakBar](<http://anttweakbar.sourceforge.net/doc/>) works very easily for OpenGL, DX9-11.[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...

Xinto_ · 11 mo. ago

I'd like to add [my own library](#) to this list, even though it's in a very early stage as of now.[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...

pgroarke OP · 11 mo. ago

[u/Xinto_](#) ping me when you're out of poc, i'll add it[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...

Xinto_ · 11 mo. ago

Sure

[↑ 1](#) [↓](#) [Reply](#) [Share](#) ...

Ikryanov · 4 mo. ago

Great and useful list! Do you mind adding [Molybden](#)?

It's an SDK for building cross-platform C++ desktop apps with HTML/CSS/JavaScript GUI. It includes CLI, rich API and functionality that let's you easily invoke C++ functions/objects from JavaScript and vice versa.



▼



r/cpp



1



Reply

Share

...