What are the differences between the OpenGL, GTK and QT libraries?

Asked 15 years, 9 months ago Modified 4 years, 7 months ago

Viewed 29k times

31

I understand the purpose of GTK, QT or other graphic toolkits. But I don't understand the role of OpenGL. Is it just another GUI-library or does it refer to something more fundamental? If so, what is it and when should I use it in day-by-day hacking?

opengl graphics

Share

Improve this question

Follow

edited Dec 14, 2011 at 18:13

casperOne

74.5k ● 19 ● 188 ● 259

asked Mar 7, 2009 at 23:29

prinzdezibel

11.1k ● 18 ● 57 ● 64

4 Answers

Sorted by: Highest score (default)

OpenGL is a library for drawing graphics on a low level. It excels mainly at 3D graphics. The most important

37

competitor to OpenGL is Direct3D (*not* the whole DirectX).

GL is only really powerful if you have it hardware accelerated. Today, almost every desktop or laptop solution has GL acceleration, but on some systems where the graphics card is integrated into the chipset, it is still rather slow.

For day-by-day hacking, GL can serve many unusual purposes. The reason is that you can leverage the GPU's processing power for graphic related tasks. For anything non-graphical, newer languages like CUDA exist as well to access the GPU.

While GL's real strengths lie in 3D scene rendering, it is often also used for 2D manipulation or composition. The reason is that 2D acceleration architectures of graphics cards often don't reach the power (freedom) or performance of GL. That's why sometimes GL is used for video output (mainly scaling) or blending tasks, even if specific 2D acceleration for these tasks would be available. Mac OS X is a prominent example of a whole desktop (optionally) being rendered through GL to allow appealing effects without high CPU load. But it can also be used for example to blend anti-aliased text glyphs onto a textured background.

As soon as you do a graphical, performance critical task, you will probably want the GPU to help out (possible even if you don't draw on screen) and then OpenGL is the clean, standardized, cross-platform answer to your needs. Today's GUI toolkits mostly don't offer you GPU-accelerated drawing, but the inclusion of GL rendering into their widgets. So you can use them for UI drawing and include your performance critical realtime graphics where it suits. Building a whole UI inside GL is a pain, but there are also solutions for that available (typical use cases are games with in-game UI).

Share Improve this answer Follow

edited Mar 7, 2009 at 23:47

answered Mar 7, 2009 at 23:33

ypnos
52.2k ● 14 ● 102 ● 148

1 Why is it that 2D acceleration is generally slower? Wouldn't it make sense for things to go faster with only 2 axes? – mk12 Jul 31, 2010 at 23:11

2 First, most of the additional operations in 3D, like the more complicated matrix operations, are basically implemented in hardware, therefore they are 'for free' as compared to more simple 2D operations. The drawing itself is then done in 2D for both cases (see Scanline algorithm). Second, 3D acceleration is generally faster because the GPU vendors put much more effort in optimizing drivers for GL and DirectX instead of 2D hardware acceleration; especially for platforms other than Windows. Most GPU benchmarks are 3D gaming benchmarks. That's where the competition in the market takes place. – ypnos Aug 2, 2010 at 8:36 ✎

22

It's something more fundamental. OpenGL lets you perform low-level drawing (think "draw a line from here to here" rather than "draw a button"). OpenGL operations are also often hardware accelerated and thus very efficient, and the OpenGL API allows you to easily work with 3D.

Choosing between OpenGL and a GUI toolkit like QT or GTK really depends on whether you're trying to draw UI elements for your application or want very fine control over how your graphics are drawn.

Sometimes, GUI toolkits employ OpenGL under the covers in order to draw their on-screen elements.

Share Improve this answer Follow

answered Mar 7, 2009 at 23:34

MandyK
1,221 ● 10 ● 11

10

OpenGL is graphics API upon which more complex user-facing graphics libraries are built. If you are interested in graphics development, and in particular 3D graphics, then it is well worth making it's acquaintance. It's and extraordinarily powerful system that can do everything from draw a line through render Quake to handle photo-realistic atmospheric rendering.

Your starting point for all things OpenGL related is OpenGL.org, which has links to everything you need.

If you prefer a textbook, and that's not a bad idea with OpenGL, then the definitive text is 'The OpenGL programming guide', colloquially known as 'The Red Book' on account of it's cover. You may wish to supplement this with 'OpenGL Shading Language' (or The Orange Book) which does what it says.

One of the nice things about OpenGL is that there is a simple test toolkit - GLUT - which has been implemented in most languages with bindings and is extraordinarily easy to get up and running to experiment and generally hack around with.

So go get rendering your teapot!

Share Improve this answer Follow

answered Mar 8, 2009 at 0:03

Cruachan

16k ● 6 ● 62 ● 112

8

GTK, Qt, MFC, wxWidgets, etc are windowing toolkits - they provide controls such as dialog boxes, widgets, buttons, etc to create GUI apps.

One of the controls they generally supply is a canvas that you can draw on with graphics commands (draw line, draw circle etc). They probably also supply an OpenGL canvas as well that you can draw on with OpenGL commands.

OpenGL (in simple terms) is a language for displaying 3D objects, it is very powerful, cross-platform, and somewhat complicated (especially if your maths is weak!) BUT it is far better than trying to do any of this 3D stuff yourself! It can be frustrating at times especially if you do something wrong and it just draws a blank screen.

Just to make things a little more complex there are also windowing toolkits written in OpenGL where OpenGL provides a set of buttons, dialogs, etc in the scene AND there are times when (behind the scenes) all the GUI elements used by something like Qt are actually being drawn by OpenGL!

Share Improve this answer

Follow

edited May 4, 2020 at 14:15

Aykhan Hagverdili
29.8k ● 6 ● 48 ● 103

answered Mar 8, 2009 at 0:39

Martin Beckett
96k ● 28 ● 195 ● 268