

Re: Why Lisp supposedly "sucks for game development"

Source: <http://coding.derkeiler.com/Archive/Lisp/comp.lang.lisp/2004-10/1901.html>

From: Kenny Tilton (ktilton_at_nyc.rr.com)

Date: 10/23/04

Date: Sat, 23 Oct 2004 19:35:53 GMT

Jack Palevich wrote:

> *Wow Kenny, good one! Pretending to be swayed by my arguments, but*
> *actually cleverly rebutted each one. It really makes me want to*
> *consider your point of view!*

Oh, gosh, if I had only responded in good faith you would have stopped spouting nonsense such as "Lisp cannot call C"? [Duane, you were right! I should have reasoned with him!]

>
> *Oh wait, no it doesn't. It makes me think you must be a smug Lisp*
> *weenie who doesn't know anything about the problem domain I'm*
> *operating in.*

Good for you, don't get sucked into actually responding to the points made. I think it was brilliant how your Sucks II simply did not address any of the telling things said here in response to Sucks I. You have been watching Dubbya debate, haven't you?

Anyway, I /am/ pretty stupid, but one of my monkeys was curious and went over to Gamedev.net (a known hotbed of SLWs) to learn about the alleged slowness of Direct3D vs OpenGL. If Gamedev is right, that is wrong. I am surprised you did not pounce on that. Oh, wait, I see why: "Performance is no longer an issue! <snip> performance is in the hands of the guys who make graphics cards and drivers". He is talking about the race between D3D and OG, but the point is still that performance is not in the hands of either the application or even d3d/og, let alone C++ or Lisp.

Here is the full excerpt, from the d3d/og comparison:

<http://www.gamedev.net/reference/articles/article1775.asp>

"Performance and Quality Factors

I've carefully avoided mentioning performance for most of this article. Why? There is a huge debate on the problem. There are many people who claim that OGL is faster than D3D, and of course, the reverse as well. I've been studying performance results: I've considered my own, my friends', and the results of major companies. I have come to the following conclusion: Performance is no longer an issue! The speed for both APIs has come out exactly even for well written programs. The performance can only be gauged per machine, and that by testing. There is no way to predict which will run faster...<snip>.

I feel that I should still address what exactly it is that affects the performance. If a game is well written, by people who know how to properly use the API, then performance is in the hands of the guys who make graphics cards and drivers. No one else has anything to do with it anymore. What determines performance is how well the API, driver, operating system, and hardware interact. On one of my computers, for example, OpenGL runs somewhat faster. I do know, however, that this is due to an incompatibility between DirectX and WindowsXP and my GeForce MX100/200 card/driver.

The above applies a great deal to quality as well. The APIs don't really have an edge over each other at all. The most important factor in quality, however, is your artwork. A good artist will make even a bad game look fantastic, but the converse applies as well. If you want high-quality, get a high-quality artist. "

A high quality artist, and as someone else noted, an HQ story. But wait, what do I know? Let's check in again, this time with those SLWs over at Gamespot, on how they rate games:

<http://www.gamespot.com/misc/reviewguidelines-2.html>

"Graphics

This includes technical use of graphics, as well as the aesthetic quality of a game's appearance and presentation. Games that look impressive in still images do not automatically earn high graphics scores, because the graphical performance of a game weighs heavily into our consideration of its visuals.

Sound

This includes the quality and use of sound effects, voice acting, and music in the game. Any of these may potentially have a significant impact on the overall sound score. For instance, a game with an excellent soundtrack and sound effects may still lose points for sound if the voice acting is of low quality.

Gameplay

This includes everything, from the game's interface to its control to how well balanced it is. Basically, this represents how well a game plays and how enjoyable it is to play. Games very rarely earn a 10 in this category, due to how elusive games with perfect or near-perfect

gameplay actually are.

Value

For the most part, a game's value score is an indicator of the game's longevity. It represents how long you'll be able to both play and enjoy the game, and it also signifies how much replay value you'll get out of it. Breadth of gameplay options and overall volume of content both weigh heavily into this score. The retail price of a game can also figure into the value score, as can the presence or lack of similar, competitive products in the market. Finally, the overall quality of the game has some impact on its value, so, for instance, a bad game that's extremely long is still probably not worth playing.

Reviewer's Tilt (Tilt)

This score basically lets a reviewer sway the final score—either higher or lower—based on the reviewer's overall experience with a game. Here's an example: A game might have really good graphics and sound but only mediocre gameplay. As a result, it gets a low tilt score to keep the overall from being boosted too high, since the game ultimately isn't fun. Likewise, a game might have mediocre graphics and sound, but it might include a really good story and a lot of original ideas. As a result, it might get a high tilt to boost the overall score a bit, which suggests to you that you should look beyond the game's production values."

You were right! Performance (which is now only a factor of the OS and graphics card and machine power) is an important /part/ of the graphics score, which is a /part/ of the overall score. That was your point, right?

[I have a letter into Gamespot asking them exactly how the sub-scores are weighted. They do that, but I could not find the specific weights.]

By the way, I love that "head them off at the pass" move in which you pretended you were looking in earnest at using Lisp for games development. You work for Microsoft and already are a C++ whiz. But earlier in that gamedev d3d/og comparison we see this:

"DirectX also provides something called the Direct3DX function library, introduced with DirectX 7.0. This is a static library that provides gobs of useful functions that help make your life easier. There is one catch, though: many of the things that have been added are C++ specific. D3DXMATRIX is derived from D3DMATRIX, for example, and many of the operators are overloaded as well. Of course, if you're using C++ anyway, this really isn't a problem."

So unless you also have your letter of resignation from Microsoft open in another window, there is no way you would be considering anything other than C++. But, hey, what chance is there that a bunch of SLWs would know any of the above? None!

Don't worry, your secret is safe with me. Just don't blow it by actually responding to anything of substance — not that I have to tell you that!

comp.lang.lisp: Re: Why Lisp supposedly "sucks for game development"

kenny