D, or What Go May Have Been

Adrian Kosmaczewski

2022-09-09

In my quest to learn more and more programming languages, I recently dipped my toes into the D Programming Language[1]. My reaction to it involves sadness; on the positive side of things, the language is undeniably brilliant.

On the other side, it is an almost completely ignored language, at least in the circles I use to move. In my 25 years of professional experience, I only met one developer using D for work; the language hardly (if ever) shows up in technical discussions. The official website says[2] that D is used by eBay, Netflix, Facebook, and others, so thankfully it has found certain niches where it shines.

But D deserved more.

As usual, I implemented my Conway[3] and Fortune[4] projects with it. The development experience is brilliant, on par with Go (hence the title of this blog,) C#, TypeScript, and other modern languages. The compiler generates fast code fast, the language has lots of modern features, and in general, it qualifies as a boring language[5]. Again, in my own scale of values this is a positive trait of a programming language.

Developers coming from C++, C#, Java, and Go will find D immediately approachable.

```d
import std;
import core.thread;
import conwaylib;

void clrscr()
{
    spawnShell("clear").wait;
}

void main()
{
    Coord[] alive = blinker(Coord(0, 1));
    alive ~= blinker(Coord(0, 1));
    alive ~= beacon(Coord(10, 10));
```

---

[1] https://dlang.org/
[2] https://dlang.org/orgs-using-d.html
[3] /blog/polyglot-conway/
[4] /blog/fortune-apps/
[5] /blog/dart-is-boring/

```d
    alive ~= glider(Coord(4, 5));
    alive ~= block(Coord(1, 10));
    alive ~= block(Coord(18, 3));
    alive ~= tub(Coord(6, 1));

    World world = World(30, alive);
    int generation = 0;
    while (true)
    {
        clrscr();
        generation++;
        writeln(world);
        writeln(format("Generation %d", generation));
        Thread.sleep(dur!("msecs")(500));
        world = world.evolve();
    }
}
```

In many ways, D is like Go. It uses garbage collection, has a nice library with everything you need out-of-the-box, is cross-platform (available for Windows, macOS, all major Linux distributions, Android, and even FreeBSD,) has functional programming features, has integrated unit testing, a package manager, and much more. It also adds some features that Go has not. For example, D integrates unit tests with documentation generation (a fantastic idea!) It also features class inheritance–because, after all, D was meant to replace C++.

Unfortunately, and quite obviously, D is not Go. I say unfortunately because such a language, with such an ecosystem around it, and with such luminaires at its helm (Walter Bright[6] and Andrei Alexandrescu[7] being just a few of them), deserved better. Better, for example, than *not even being mentioned* in the Stack Overflow 2021 survey[8] of most popular technologies. This is, simply put, not fair; *dura lex, sed lex.*

And now Google, after Go, Dart, and how many more languages, is positioning *yet another* one, this time called Carbon[9], as a successor of C++. **Such privilege belongs to D, not to Carbon.** But market forces are something to be reckoned, and at this point it is not obvious how the D Language Foundation[10] could curb them.

Politics and stock market value have a lot to say in the war for programming language supremacy. Go has had its "killer app" with Kubernetes, and as a result Go is 15th in the August 2022 TIOBE ranking[11], while D appeared 20 positions lower. Go is 11th in the IEEE Spectrum ranking[12], while D appears at the 27th position. Go is 9th in PYPL[13] at the time of this writing, and 16th

---

[6]https://en.wikipedia.org/wiki/Walter_Bright
[7]https://en.wikipedia.org/wiki/Andrei_Alexandrescu
[8]https://insights.stackoverflow.com/survey/2021#technology-most-popular-technologies
[9]https://9to5google.com/2022/07/19/carbon-programming-language-google-cpp/
[10]https://dlang.org/foundation/about.html
[11]https://www.tiobe.com/tiobe-index/
[12]https://spectrum.ieee.org/top-programming-languages-2022
[13]https://pypl.github.io/PYPL.html

in the RedMonk ranking[14], and D doesn't even appear in either list.

D is ready, powerful, easy to learn, compatible, cross-platform, fast, extensible, and fun. There's a lot to like in there. But it lacked a "killer app."[15]

---
[14]https://redmonk.com/sogrady/2022/03/28/language-rankings-1-22/
[15]https://en.wikipedia.org/wiki/Killer_application