STACK4 and the D Programming Language (Part1)

As of last year I am the proud developer of a mobile game app for android (and blackberry as of now). STACK4[1] is a rather simple little game based around the concept of connect-4 but adds 3D and consumable extras to the mix.

This post is about the biggest addition to the game since its launch last summer. In December 2013 I added a realtime multiplayer mode to the game (version 0.3.0).

The server backend for this endeavor is written in the D Programming Language[2]. You may ask why this is… Well this was kind of a journey! This is gonna be a series of articles about how I developed that server and what kind of challenges occurred on the way.

Part1 elaborates roughly what other possibilities were evaluated and a brief not-so-technical overview about the development in D-land over the last year.

Typescript / Nodejs

I used nodejs a lot over the last 2 years and I really like it… performance-wise. What totally annoys me is Javascript. So naturally the first straw that I grabbed was Typescript which is basically JS with types that compiles down to bare JS. Using this, I implemented a first small browser game with interactive multiplayer gameplay called Beachwars-Online. Unfortunately TS is still JS and all the libraries out there are for JS and writing bindings annoyed the hell out of me. So I dreamt about having a framework as rich as node (+all the npm packages) but with a statically typed language.

Java / Vert.x

Next thing I evaluated was Java! Typesafe, strict OOP and an insane huge framework for all the needs, why not? Vert.x promised the event-driven performance I was used to from Node. In the end the VM and/or Vert.x were too ressource hungry for my commodity hardware server. And by the time I was evaluating Java I got help by a friend who was no java coder nor was he willing to become one. So I thought to look for yet another solution.

PHP

My friend is very proficient in PHP and put that on the table too. But since I wanted to do realtime server connections and wanted to leave scripting languages behind PHP got sorted out pretty quickly too.

C/C++

I am a C++ coder at day and my friend also learned C in the days. But trying to find object-oriented wrappers for stuff like libuv was not as easy as I expected. Writing something ourself seemed unreasonable for the task and not being able to use C++11 for it (remember the commodity = dated hardware) made me expect the worst. That is when I remembered something that cannot deny its C++ roots: D.

D

2013 was also a good year for D. The first real dconf was held in San Fransisco[3] with a load of interesting talks and people! I recommend everyone to watch the video recordings!

The D reference compiler “dmd” matured a lot, stability-wise and by finally supporting x64 and shared libraries. The toolchains got orders of magnitudes better through the dub package manager[4] that lent a lot of ideas from npm for nodejs (which is a good thing, since this works extremely well). Also in IDE land mono-d[5] made my work with D a lot easier. It got very stable by now and the developer is insanely responsive (lets hope he never finishes university).

In october Andrei Alexandrescu announced the first commit of D code to the Facebook source repository[6]. In November, they even started to support D financially by putting bounties on compiler bugs[7]. Others followed.

Facebook is also the sponsor of the location for the D conference in San Fransisco. And I think it was the dconf talk of Vladimir Panteleev[8] that first introduced me to vibe.d. The Vibe.d framework offers asynchronous I/O and all you need to write a webserver in D.

TL;DR

I know D for a long time, have used it a lot years ago, wrote my B.SC Thesis about it. Vibe.d[9] offers me what I like in node.js albeit being written in a usable language for serious sized projects. Despite being asynchronous like node it also looses the callback hell by using a co-routine like design. It is more memory efficient than Java/Vert.x, PHP was no option, bare C not even close and D is the better language compared to the competitors anyway.

Part2 is going to describe a little more in detail what the STACK4 server backend looks like and how I implemented it in D… stay tuned

[1] https://play.google.com/store/apps/details?id=com.Extrawurst.FIR [↩]
[2] http://dlang.org/ [↩]
[3] http://dconf.org/2013/ [↩]
[4] http://code.dlang.org/ [↩]
[5] http://mono-d.alexanderbothe.com/ [↩]
[6] http://forum.dlang.org/thread/l37h5s$2gd8$1@digitalmars.com#post-l37h5s:242gd8:241:40digitalmars.com [↩]
[7] http://forum.dlang.org/thread/l65mvq$du0$1@digitalmars.com#post-l65mvq:24du0:241:40digitalmars.com [↩]
[8] http://dconf.org/2013/talks/panteleev.html [↩]
[9] http://vibed.org/ [↩]