Das ist ein archivierter Beitrag. Du wirst hier nicht abstimmen oder kommentieren können.

1084

Hey, this is kyren from Chucklefish, we make and publish cool video games. One of our two next projects is currently being written in rust, and I'd like to talk to you about it! (self.rust)

eingereicht vor 7 Monate * von kyrenn

Here's a little bit about Chucklefish if you're not familiar.

So, one of our next projects is codenamed "Spellbound", and you can read little bit about it here. It's still pretty early in development, but I got the go-ahead from the boss to talk a bit about development here, in case anybody is interested.

Mostly I want to talk about the technical aspects of development so far in rust, and some things that we've run into along the way. A big part of this is that one of the goals for development on Spellbound has been to focus on console development very early in the process, rather than focusing on PC and then porting at the end, as we've done in the past. Something that I've had working for quite a while now but so far had been mum about was that we have a nontrivial rust project working on all three major consoles, and I have just been *dying* to talk about this. I have to be pretty careful, because of course I can't talk about specifics due to legal agreements that all console developers are bound by, but I believe I can explain quite a lot of what went into it and how easy / not easy it was without running into trouble.

I sort of intend this as an AMA, so feel free to ask me anything you're curious about regarding rust gamedev, or getting rust to run on consoles, or anything technical really. First though, I'll try and talk about some of the pretty obvious stuff:

1) Who are you, and why are you talking at me and why should I care?

I'm "kyren", I was the original lead programmer of the game "Starbound", and I'm working as the technical lead on one of the two current Chucklefish projects, "Spellbound".

2) Why did you decide on rust for developing a new game?

I think Chucklefish falls into a very specific niche in game development where using "alternate" languages is viable, and after exploring the space of alternatives to C++, I chose rust.

3) But rust is too young, there are no game engines written in rust, why don't you use Unity, etc?

Like I said, Chucklefish just so happens to be well suited to push boundaries a bit, because we focus on 2D games and don't really use any existing game engines. I don't want to start a huge debate about the merits of game engines like Unity for 2d development, but *for us* it has never really seemed terribly attractive. YMMV.

4) Why not C++? Why not a different language?

We're very *very* familiar with C++, Starbound was written in C++, and Chucklefish's other current project "Wargroove" is also written in C++. I feel that rust solves a lot of the problems and matches a lot of the lessons that I learned making Starbound, and I'm more comfortable making new projects in rust rather than C++ from here on out. There are not TOO many languages to choose from though, because for practical reasons, you need a language that can has no runtime, no gc, and can more or less pretend to be C.

5) How did you get rust to run on three consoles, was it difficult? Are you using 'std' or 'no_std'? Is this something that is feasible for other people to do?

Spellbound is built as a static library using some high level interfaces that define just enough of an Application / Audio / Rendering API. On the PC, this is pretty easily implemented in SDL2, on consoles, it is implemented roughly half in C++ (to interface with the console APIs) and half in rust, with the specific balance depending on the specifics of console APIs that I cannot talk about. We patch 'std', 'libc', and 'rand' to build with custom targets for each console, so that we can more or less use stock rust with 'std' and a whole bunch of crates without difficulty. I can talk about this more in detail depending on what people want to know. I would estimate the total amount of *extra* time that I spent getting Spellbound running on consoles vs if this was a project in C++ rather than rust at around 2 weeks of just my time. It was actually easier than I expected, but it does require quite a lot of domain knowledge.

6) Rust is not ready for game development, this was a bad decision!

That's not a question :P For us, for this project, it honestly seems to be working out pretty well. The last real concern was platform portability, and that's no longer really a concern. There's not REALLY any more roadblocks related to choice of language, which is why I'm talking about it here now.

7) This means rust is 100% ready for game development for everyone!

Hey, that's not a question either! I would emphatically say NO to that, but honestly I'm not sure I would say yes to that about literally any platform. Everyone is different, every game is different, everyone's requirements are different. If you want to make something in Unreal 4, you might have a bad time? Maybe not, I don't know!

8) I think [insert other platform / engine] would have been a better choice!

Still not a question! That's very likely to be true *for you*, that may even have been true for us, who knows. That's just like, your opinion, man.

9) Does this mean that your next game will 100% for sure immediately come out on all three consoles on release day?

The game is *running* on all three consoles with input / audio / rendering, but that is not all that goes into releasing for a console. I'm not really allowed to talk about it in tremendous detail, but I can pretty much say that there shouldn't be anything *technically* in the way. We're still pretty early in the development process though, please do not construe what I'm talking about to be a promise about scheduling or releases or anything like that.

10) What crates do you use?

So far, in no particular order, at least lazy_static, log, rand, num, smallvec, serde (+json +yaml), strum, rental, regex, backtrace, itertools, error-chain, sdl2, gl, png, ogg-sys, vorbis-sys, vorbisfile-sys, twox-hash, rlua, and probably some I've missed. Cargo is a *godsend*. Edit: I also forgot 'smallvec', and there's a transitive list in the comments below.

11) Open source your engine, I want to use it!

I wouldn't consider the engine spellbound is being made in to be *general purpose* exactly, but it may be general purpose if you limit yourself to 2d games. Closer to release, I think I may be able to swing open sourcing more of the engine than is currently, but right now our main open source contribution is the 'rlua' crate.

I have left out a TON I'd like to talk about, because otherwise this post might go on forever. If you're interested in more specifics, let's talk about it in the comments!

Edit: Okay, I have to go, I tried to answer as many questions as I could, and I still have a bunch to answer and I'm losing the battle against sleep. I'll try and answer any remaining questions tomorrow, so if I didn't get to something you really wanted answered, hopefully I'll get to it tomorrow. Thank you for the gold, and thank you all for being so supportive and positive, it really means a lot to me! Good night!

Edit 2: Well, I answered a bunch of questions from this morning / afternoon, and I tried to answer basically everyone. I'm sure I've missed some, and I'm sorry if I did! I'll check back occasionally, but I think I need to take a another breather for a while. This has been really amazing, thank you all for the wonderful questions! I learned a whole bunch actually reading some really good, deep discussions that were started. <3 you all :D

337 Kommentare Weitersagen Speichern Ausblenden Schenke Gold melden crosspost

Alle 337 Kommentare

sortiert nach: **beste**

Suchen

Dieser Beitrag wurde eingereicht am 23 Oct 2017

1.084 Punkte (98 % gefällt das)

shortlink: `https://redd.it/78bowa`

submit a link

submit a text post

abbestellen

39.051 Rustaceans

905 Mitglieder anwesend

Set your user flair

kellerkindt

The Rust Programming Language

A place for all things related to **the Rust programming language**, an open-source systems programming language that emphasizes zero-overhead memory safety, fearless concurrency, and blazing speed.

Rules

- **Respect our Code of Conduct.**
- **Constructive comments only.**
- **No memes.**
- **No zealotry.**
- **Submissions must be on-topic.**
- **Chill out!**
- **Be Safe**

Resources

- **Learn**
- **Install**
- **Discuss**
- **Contribute**

Nachricht an die Moderatoren schicken

MODERATOREN

azth
kibwen
dbaupp rust
flaper87 rust
mrmonday libpnet · rust
howellnick rust
wrongerontheinternet
Manishearth servo · rust · clippy
sho_kde
llogiq
clippy · twir · rust · mutagen · flamer · overflower · bytecount

...und 3 mehr »

KÜRZLICH ANGESEHENE LINKS

Ignition, another classic game re-released on GOG with Linux support
31 Punkte | 23 Kommentare

VP9 encoding/decoding performance vs. HEVC/H.264
308 Punkte | 42 Kommentare

Video games website Steam fined $3 million for refusing refunds
23.0k Punkte | 3141 Kommentare

This might just be the worst I've ever seen
4711 Punkte | 317 Kommentare

My triumph in Rocket League and why you need to own it on Linux
2 Punkte | 6 Kommentare

Löschen

Account-Aktivität

[–] **Manishearth** servo · rust · clippy 210 Punkte vor 7 Monate*

Wow, this is amazing! I love your games! Stardew Valley is one of the most relaxing games I've played and I really love it. Starbound is super fun too!

Someone linked to this on Twitter and I was like "omg omg they're probably making a new game gotta see this", *before* I realized what sub this was on.

Then I opened the link, started reading, and realized that the theming of the subreddit was suspiciously similar to that of ... OH, OMG.

:)

> We patch 'std', 'libc', and 'rand' to build with custom targets for each console,

I understand you can't talk about the details here, but it would be nice if these could be eventually upstreamed somehow (at least, in part?). Rust on consoles is something folks have made work but I don't think it's something that's well documented or "easy" to do, yet. Edit: Oh, I see; this is not something that you can't talk about till the game release, it's because of console NDAs. Oh well :(

Some questions:

- What was the *worst* thing about using Rust? (i.e, what can we improve upon?)
- Not a question, but: OMG OMG OMG
- How was the experience ramping up your team on Rust? (again, what can we improve upon?)
- It seems like you're doing a nontrivial amount of FFI to existing C++ code. Are you using bindgen? How challenging was building this portion of the codebase?
- Do you use clippy? :) [Given you've forked the stdlib and clippy is strongly tied to nightly versions I suspect this to be a no]
- Would you like to be on https://www.rust-lang.org/friends.html ? Once the game is released (pre-release is also fine, "available to consumers" basically), if you follow the instructions at the bottom of the page, you can be!
- Also not a question: OMG
- We (the Rust community team) often organize chats with "production" users to get an idea of folks' needs, and to help them in any way we can. It should also be possible to do these with an NDA (we haven't had to yet, so I'm not sure) if you wish to more freely discuss things. We've found these very helpful in the past; it helps us get a better picture of things and prioritize. Would you like to at some point set one of these up? If so, please contact us at community-team@rust-lang.org (or talk to me here)

Permalink embed Speichern Schenke Gold

[–] **kyrenn** S 134 Punkte vor 7 Monate*

> Oh, I see; this is not something that you can't talk about till the game release, it's because of console NDAs. Oh well :(

I can't talk about it *at all*, afaik, even after release. You can't discuss or share information on console APIs, the rules are very strict. I would in a heartbeat if I could, but of course if that was true, rust would already be 100% supported because it is not that hard and other people would already be working on it in the open.

Microsoft deserves some kudos here for pushing things forward https://docs.microsoft.com/en-us/windows/uwp/xbox-apps/

I'm going to go through all your questions!

> What was the worst thing about using Rust? (i.e, what can we improve upon?)

Hmmmmmmm, that's a good question. I'm not sure *exactly* if I would answer this way, but I know exactly what the other two Starbound programmers who are currently part time on Spellbound would say, and I have to say I don't disagree. *Floating point comparisons*. I know that there's not really a good answer here, and we do have *most* of a handle on it now, but it is just super super painful. The thing is, after programming rust for a while, if I go back to another language I'm *terrified* of floats, because I know that it never really *worked* there the language just doesn't *throw the problems in your face*.

The absolute 100% killer best feature of rust for me is lack of UB, and floats are like kryptonite for this :(. I don't even know how to solve this or anything, but if I could just wave a magic wand, I think I would go for a NotNan or Finite type that was built into the language or something and was Ord. I'm sure there are problems there I'm not thinking of though.

Our current solution is something like the ordered_float crate in combination with something like the ord_subset crate.

> Not a question, but: OMG OMG OMG

>:D

> How was the experience ramping up your team on Rust? (again, what can we improve upon?)

There are two other programmers who are working on both Starbound and Spellbound at the same time, and they've been taking this time to get up to speed with a new project and engine etc. They've honestly not had *too* bad of a time with things, partially because they're really really capable and all around awesome, but also because it's honestly not too bad. I think that a lot of the design pain was more early on, and hopefully it helps to learn coming into an existing project rather than something green field. I think the biggest gripes and confusion have been things that would be solved by NLL, and also again floating point ordering. I think the question of what is the most painful and what is the hardest to learn are really two sides of the same question, they're smart enough to get the *concepts* quickly, it's when all the designs that they would naturally write seem to generate errors and they don't know where to go is where it is really painful.

> It seems like you're doing a nontrivial amount of FFI to existing C++ code. Are you using bindgen? How challenging was building this portion of the codebase?

I SHOULD be using bindgen, it's actually on my list of things to do. For all three consoles, there are input / rendering / audio APIs that are C that wrap the platform API, but when you look at them they're actually kind of small, so I didn't use bindgen. I SHOULD be using bindgen there though, and if I have to change them again I probably will do that.

I'm gonna try and say this without running afoul of NDAs, but most of the consoles have like, a huge chunk of them that are just modifications of some existing platform, and so there's a lot of C functions available, and they're super close. That stuff, generally it's just a matter of modifying 'std' to fit it and mostly working around bad implementations, so that stuff doesn't need bindgen because it just all lives in libc or similar. The rest of the APIs are the rendering / input / audio APIs I mentioned above.

> Do you use clippy? :) [Given you've forked the stdlib and clippy is strongly tied to nightly versions I suspect this to be a no]

We use clippy! I LOVE clippy, it's especially helpful when learning. We generally do development on the PC version, and just make sure that the console versions still build as we go. In fact, we use CI really heavily at chucklefish, so we can just hit a button and out pops a [REDACTED] file ready to be tested on [REDACTED]. We don't used patched libstd on the PC version or anything, it's just normal everyday rust.

> Would you like to be on https://www.rust-lang.org/friends.html ? Once the game is released (pre-release is also fine, "available to consumers" basically), if you follow the instructions at the bottom of the page, you can be!

We would love to be on that list! I can't say ANYTHING about how long it will be before pre-release or release, or whether or not there will be a pre-release, but once we do I'll look into that!

> Also not a question: OMG

<3

> We (the Rust community team) often organize chats with "production" users to get an idea of folks' needs, and to help them in any way we can. It should also be possible to do these with an NDA (we haven't had to yet, so I'm not sure) if you wish to more freely discuss things. We've found these very helpful in the past; it helps us get a better picture of things and prioritize. Would you like to at some point set one of these up? If so, please contact us at community-team@rust-lang.org (or talk to me here)

That would be great, actually. What would be the next step to set that up?

Also thank you for all the love and support, it means a huge amount to me, and thank you for all your hard work on rust and servo!

Edit: I thought of a good answer to the first question, *self borrows*, or the lack thereof. I would *kill* for language supported self π