EPISODE 856

[INTRODUCTION]

[00:00:00] JM: Niantic is the company behind Pokémon Go, an augmented reality game where users walk around in the real-world and catch Pokémon which appear on their screen. The idea for augmented reality has existed for a long-time, but the technology to bring augmented reality to the mass market has appeared only recently.

Improved mobile technology makes it possible for a smartphone to display rendered 3D images over a videostream without running out of battery. Pokémon Go was the first game to come out of Niantic, but there are other games on the way. Niantic is also working on the Niantic real-world platform, a planet scale “AR platform” that will allow independent developers to build multi-player augmented reality experiences that are as dynamic an entertaining as Pokémon Go.

Paul Franceus is an engineer at Niantic and he joins the show to describe his experience building and launching Pokémon Go as well as abstracting the technology from Pokémon Go and opening up the Niantic real-world platform to developers.

A few updates from Software Engineering Daily land. FindCollabs is a place to find collaborators and build projects. FindCollabs is the company that I’m building and we’re having an online hackathon with $2,500 in prizes. If you’re working on a project or you’re looking for other programmers to build a project or start a company with, check out FindCollabs. I’ve also been interviewing people from some of these projects on the FindCollabs Podcast. So, if you want to learn more about that community, you can hear it on that podcast.

We have a new Software Daily app for iOS. It includes all 1,000 of our old episodes as well as related links and greatest hits and topics. You can comment on episodes. You can have discussions with other members of the community and you can become a paid subscriber for ad-free episodes at softwareengineeringdaily.com/subscribe. The Android app is coming soon, and if you use the iOS app, I would love to get your feedback on it.

With that said, let's get on to today's show.

[SPONSOR MESSAGE]z

[00:02:19] JM: Capital One makes technology to impact real people. Capital One is a bank, and today's banks are becoming technology companies. Capital One is an engineering-focused organization that puts technology at the center of everything they do. Engineers within Capital One work with new technologies and develop modern systems for banking customers.

Capital One is led by the founders and they're exploring new possibilities and working to build a bank for the future. Capital One is hiring for New York, Dallas, D.C. and many other cities across the United States. Capital One works in small cross-functional teams, which embrace a lean, agile approach to cloud native software development.

At Capital One, you can bring your ideas to life. Go to capitalonecareers.com/sed and find out careers that might fit your job requirements. You can also check out the episode that we did with Capital One. We explored the idea of digital transformation in the realm of a bank. You can find that episode on Software Engineering Daily .

Capital One is an equal opportunity employer across gender, disability, veteran status, and sexual orientation. To find opportunities within Capital One, go to capitalonecareers.com/sed.

[INTERVIEW]

[00:03:56] JM: Paul Franceus, welcome to Software Engineering Daily.

[00:03:58] PF: Hi, Jeff. Good to talk to you.

[00:04:01] JM: We are talking about Niantic and augmented reality today. The idea for augmented reality has existed for a long-time. We have our definitions for what we see as augmented reality. How would you define the term augmented reality?

[00:04:20] PF: So, that's a really good question actually. So, I think that people think of AR as this – Like you're placing a vase, a flower vase on a table, right? But we kind of like to go beyond that. So, one thing that we would like to say is if I put a flower vase on the table, that you would see the flower vase in the same spot with the same flower. If I were to pick a flower out of that vase, that you would see the flower get picked out of the vase. If you walk in front of the vase, I would not see the vase anymore.

I mean, currently, with most of these AR stuff, it doesn't include the background at all. But beyond that, let's get more interesting here. That flower is going to be depending on the native plants in that area of the world where you currently are. If it happens to be raining or the wind is blowing, you will see that effect on the actual device that you see. AR doesn't have to be even a camera thing at all. Our game Ingress that came out in 2012 doesn't actually have any in-camera AR, but it is an augmented reality app because the world that you see through the game is not the world that you see around you. It has things in it that are unique to the game, but that you have a different sense. That sculpture in the park next to suddenly become something that means something in a game as supposed to being just a sculpture.

[00:05:46] JM: What are the limitations of AR today? We have these great smartphones. We have these robust and ever-improving cloud provider tools, but where the limitations. What do we need in order to make AR a more widely accessible, broadly adapted reality?

[00:06:09] PF: So, first of all, it would be nice if there were glasses that people would actually wear, and I don't know how far that is away. But, I mean, you're not going to see people walking around with a HoloLens or a Magic Leap on their face maybe in a small – Those who work indoors perhaps, but we're looking for something that works across the whole world. I think that maybe people need to expand their idea of what AR is.

For example, the in-camera kind of things, we don't want the whole game to be based on that just because then what you end up with is a bunch of people walking around and holding their phones up all the time and people, first of all, look really strange doing that. Second of all, you don't really want to do that. What we're trying to do is to decrease the amount of time people actually look at the screen and have the game be something that is more integrated into your life where maybe you're walking around and your phone notifies you that there's something

interesting nearby that you should check out and maybe you'll be willing to pull your phone out for that purpose. But I don't know what it's going to take to make this thing widespread. I mean, we've had pretty good success with Pokémon Go as you're aware, and I think we're going to hopefully keep being successful like that.

[00:07:23] JM: Today, most of the augmented reality applications that we see our games, there are some utility applications of augmented reality. When do you think we'll see more widespread utility applications, or are we already seeing them?

[00:07:40] PF: So, our focus at Niantic is about the real-world applications. So, our mission in our company is to get people to go outside, to explore the world around them, to exercise and tend to be social with each other face-to-face. So, that we're working on. I don't think that from Niantic's perspective, we're not going to be doing you're in a factory and you put on the pair of glasses that help you to know what part you need to replace on the machine that is currently broken. I think that those things are starting to happen in the industry, but I'm not that plugged into it myself.

[00:08:18] JM: So, we've got these libraries like ARKit and ARCore. Do these augmented reality libraries, do they do everything that you want as an augmented reality developer, or are there some things that the augmented reality tooling ecosystem for developers doesn't satisfy yet?

[00:08:39] PF: Well, yeah. So, the ARKit and ARCore are both very similar. Basically, what they do is they give you set of planes and then you can place anchors on those planes. The Apple system currently has some ability for two devices to share their view of the world. But one thing that it doesn't do is it's not cross-platform, right?

So, if I'm on an Android device and you're on an iPhone, you're not going to be able to do any kind of sharing with that. Also, it doesn't have any real support for shared experiences. So, we actually have a team here that is working on our own internal AR system, and we make use of ARKit and ARCore on the devices themselves, but then we go far beyond that.

So, we actually have the ability for multiple people to all share the same view of a set of AR things and to interact with those things and in real-time and to make sure that all the devices are

in sync with each other. So, if you go on to our YouTube channel, there're a couple of videos there, one called codename Neon, which is a multiple player game where you can see 8 to 10 people I believe running around in a room, picking up blobs of energy off the ground and then shooting them at each other and everybody can see that view of those things as they're happening in real-time.

[00:10:00] JM: And that's whether they're on an Android or an iOS device.

[00:10:04] PF: Correct. Because our games – Our software needs to work on both platforms. You don't want to exclude – Those are the majority platforms right now and we don't want to exclude anybody. So, we want to make it work cross-platform.

[00:10:17] JM: Is the round-trip time to the server? Because in order to do that kind of synchronization, I assume you need to basically like go to the cloud, write something to the cloud and then the other augmented reality user needs to read from the cloud. Is the latency low enough in that loop to have an experience that feels fully interactive?

[]00:10:42PF: So, I'll give you two part to answer that question. First of all, the AR system that we've built has about a 10 millisecond latency round-trip between the clients. We do hole punching through the mobile network to be able to have the phones be able to talk to each other directly, and there's like a multicasting type of protocol that the devices can all communicate with each other.

But this also leads to why we're so interested in 5G and that we have partnerships with Deutsche Telekom in Germany and EE in England right now, and we're helping launch their 5G, because we're going to put servers in the edge so that the players on 5G can have like a 1 millisecond round trip latency.

[00:11:25] JM: How does 5G improve the viability of AR applications? What is 5G actually mean in a meaningful sense to you?

[00:11:35] PF: From our perspective, I mean, it's a higher performance network. But really what's important for us is not speed, but latency. The way we get benefits from latency is the support for edge computing, which we've actually now demonstrated.

This codename Neon, which I mentioned previously, we actually did a demonstration at Mobile World Congress along with our partners there. Samsung, I believe, and Deutsche Telekom. So, for us, I think these companies are seeing us as being the example of this is what we can do with 5G, because I think they've created this kind of awesome technology with this edge computing and don't really know what to do with it yet. I think we've come along at a good time and can provide them with an application.

[00:12:21] JM: Okay. You drew a distinction there between speed and latency. Can you distinguish those further? How are those different?

[00:12:28] PF: So, I'm downloading a movie. I'm watching Netflix. I don't care if it takes five seconds before the stream starts coming, because I need the bandwidth for that. But it doesn't matter. I'm not interacting with the film, I mean, except for maybe pausing, but that can be handled locally with buffering, right? But I need a round-trip, quick round-trip, because if, let's say, I fire a bullet, an energy blob at another player, if that energy blob leaves my phone and it takes a half a second before you see it, I'll think that I fired at you but you will have moved out of the way before it actually goes and it ruins the illusion of reality.

So, you don't necessarily or not necessarily transferring a lot of data in that situation. It's just really, "Hey, there's a blob coming. Here is the position of it. It's moving like this. Where is it currently and so that both devices can render it in the same virtual location?" But we're not transferring like I'm sending you – I'm not sending you a giant video file.

[00:13:27] JM: Got it. You worked on Pokémon Go.

[00:13:32] PF: Yes.

[00:13:32] JM: What lessons of software engineering did you learn working on Pokémon Go?

[00:13:39] PF: Oh, gees! Lessons of software engineering. I mean, it's really just like any large project. We had some issues with – I think when we started things out, that we build things in such a way that then we realize that we made some mistakes. So, for example, in a game, you typically have a lot of different pieces of data that you need that all the different parts of different scenes and all that need to have access to, and we had this kind of monolithic thing, and it became pretty clear relatively quickly that that wasn't going to work first of all, because all that stuff is loaded in-memory at the same time and we're on a limited memory device.

Also, it made it so that when the game launched, it took a much longer time to load because you had to create all these objects. So, we quickly kind of went to a much more – We do something called game state. So, when you're on the map, that things that you need to talk to the map are loaded into memory. When you add an encounter on top of that, we load the encounter state, and those states will stack with each other. Because the encounter, you're still essentially on the map. So the map state is the base, kind of the base state and then the encounter state comes on that and you have access to those things. Then when you're done with the encounter, I don't need a 3D Poke ball anymore. I don't need the scene of the forest that comes behind the – Or the AR, the AR stuff that's loaded when you're playing, doing an encounter in AR, and that stuff can be loaded out of the game. So it's a way of keeping the things more efficient.

[00:15:10] JM: Wow! Okay. So you're saying, the V1 of Pokémon Go, I download Pokémon Go, I open it up. All of the static assets for all the Pokémon are on my phone. V2, you say, "Okay. What Pokémon are in a 3 mile radius? Let's just download the static assets for those Pokémon," and then as I walk to a different circumference, then I'll download static assets for a different set of Pokémon.

[00:15:41] PF: Yeah. So, that's something a little different, which I'll talk about in a minute. This is more like I need the AR camera object in-memory only when I'm in encounter. But when I'm in the map, I don't need it. So, it's basically loading and allocating.

This was before the game actually launched. I mean, this happened before we ever launched V1. It was just while we're in the process of developing a natural thing, "Let's just build this thing. Let's make it work. Oh, crap! It's taking away too much memory. Let's try and come up with another way of doing it."

As far as assets – So, we don't download the 3D – So, this is for 3D assets of the Pokémon. You have none of those on your device when the game is first installed. First of all, it would make the binary size much too large and app stores have limits on how much they'll have. For example, downloading over Wi-Fi. I think Apple just raised 250 MB or something like that. We would like for people to be able to download over their cellular connection, just because when we upgrade, it makes a smoother experience for people. So, we actually have an asset bundle system that when a certain Pokémon is nearby, we will then go and request from the server, "Hey, send me the asset for this thing," and those then are maintained on your device.

So, first of all, it makes it so that you don't – Somebody can't pull apart the app on the phone and pull all the assets out and steal them. Although, I mean, once you've encountered everything, you have them all, and it just makes for a more efficient kind of experience. The other thing is that as the game gets upgraded, we'll change assets. The new versions of unity may have some change with the way the 3D models need to work. So, this whole thing is versioned. So it's like Android or iOS, and then which version of the game do you have and which version of the asset? So it knows when to download assets in the background on-the-fly.

[00:17:35] JM: So, the first thing that you mentioned that I misinterpreted was actually like I download the app, it gives me the things I need. But loading AR camera applications, so like the application infrastructure that's sitting over my camera essentially and taking that imagery that's coming in and doing all the necessary AR magic on top of it. I guess that is a memory-intensive application is what you're saying?

[00:18:03] PF: So, that library or that object takes up memory. If the game is running and you have a gig of memory in your thing, and let's say that thing takes up a certain amount of memory. That's memory that's not available for other things. When I'm in the store or if I'm in the player profile or if I'm in the map itself, I don't need that. So, we kind of load and unload those things from memory as they're needed by this thing we call game states.

So, I'm in the state of the map. So, what do I need in the map? I need the load map tiles. I need to be able to see the world. I need the gestures that let me turn around. I need whatever things

that are rendered on the screen. I don't know like off the top my head all the things that are there, but there's an object that I can look at.

When I go into an encounter, now I suddenly need other things to be loaded into memory, but when I leave the encounter, those things need to go away, because I don't want to – When I'm in the store, I don't want the stuff that I need for encounter to be loaded in-memory at the same time.

[SPONSOR MESSAGE]

[00:19:14] JM: ExpressVPN is a popular virtual private network. ExpressVPN is useful for getting a private, secure, anonymous connection for your internet browsing. It encrypts your data and it hides your public IP address. You've got easy to use apps that run seamlessly in the background on your computer or your phone or your table, and turning on the ExprsesVPN protection only takes a single click.

If you use ExpressVPN, you can safely surf on public Wi-Fi without being snooped on or having your personal data or your Bitcoin account information stolen. For less than $7 a month, you can get ExpressVPN protection.

ExpressVPN is the number one VPN service rated by TechRadar, and it comes with a 30-day money back guarantee. You can also support Software Engineering Daily if you check out expressvpn.com/sedaily. You would get three months free. Everybody needs at VPN at some point in their life. So if you want to get your ExpressVPN subscription for free for three months while also trying out ExpressVPN and supporting Software Engineering Daily, you can kill all those birds with one stone by going to expressvpn.com/sedaily.

Thanks to ExpressVPN for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:20:53] JM: So, eventually we're going to get to talking about the real-world platform, which is the Niantic planet scale augmented reality platform for people who want to develop games on

top of it. I'd like to understand to what extent which Pokémon Go something that was built on top of this abstract platform infrastructure, or was Pokémon Go more built specifically just for the Pokémon Go use case without the ideas around like going beyond Pokémon Go and having more abstract interfaces?

[00:21:27] PF: Sure. So, if you go back further in time, we have Ingress, which came out in 2012. We always had an idea that we would build a platform.

[00:21:37] JM: Explain what Ingress is for people who don't know.

[00:21:38] PF: Okay. So, Ingress is augmented reality game that we released in 2012. It has a much more science-fictiony kind of theme to it. Basically, what you have is two – I'll deep for you. There is an alien presence called the Shapers that have been influencing humanity for thousands of years, and the Shapers have these things called portals, which are the place that they enter our dimension and there's this stuff called exotic matter that leaks into humanity and some people are sensitive to XM, and it causes them to do things like create artworks and things like that.

So, all the public statues and sculptures an interesting architecture that you see is because of the influence of XM on humanity. There're two teams in Ingress. There's the Enlightened that believe that the shapers, the creatures that create this XM and have been influencing our humanity for thousands of years are here to help us, and the Resistance who think that they don't think that this is a good idea and that humanity should be left alone to their own devices.

Anyway, so the game turns out to be – Essentially, a global game of capture the flag. You try and capture these portals for your team and then link them together into triangles to cover the surface of the earth with the triangles to influence that human population under those triangles for your side.

[00:23:01] JM: Okay. So that's Ingress. Now, take us through the evolution of Niantic's platform.

[00:23:07] PF: Right. So we built Ingress, we always had in mind that we are going to build a platform, but Ingress was just kind of built as a game and it turned out to be extremely useful in the sense that we learn what to do and what not to do.

So, when we're going to start doing Pokémon Go, we said, “Okay, let's take all the lessons learned. Let's start building a – The real Niantic platform.” So, Pokémon Go is built on the Niantic real-world platform, but you could think of it as being built on version 0.1.

So, over the course of the three years that we've been with Pokémon Go, we've learned a lot more things and we’ve been able to continue to grow the platform, and that's ongoing. So, now we’re getting to the point where we’re going to start inviting third parties. So, actually, Harry Potter: Wizards Unite, which is coming out sometime soon. We are actually doing the server development and Warner Bros. is doing the client. So that was kind of our first venture into having a second party player working.

We’re also doing something called the Beyond Reality Development Contest, and we've invited a bunch of teams to give us proposals and we have selected some number of those teams to come and actually hand them a platform. So, it's one thing to build something yourself and the guy next to you is the guy that wrote the API, and it's a different thing to hand somebody a pile of documentation and expect them to be able to build something. So, that's a process, a learning process that we’re going through right now.

[00:24:37] JM: Tell me little bit about that. So, you get these kind of exemplar applications, like Ingress and Pokémon Go. You learn a ton about how to build augmented reality applications. Then you say, “Now, we’re going to abstract this out and build generalized AR platform.” Tell me about that platform. What are the things that you want in every AR application that you can build into a platform?

[00:25:09] PF: So, I would change the word AR to real-world. So, what is it that we provide? So we have a single worldwide instance. So what that means is that you're not going to be assigned to one server or the other. Whenever you pull your phone out, in this case, all things are mobile games at the moment. Whenever you pull your phone out in the world, you're going to be in the same game world as everybody else. So we have built a system that can do that.

An example of that is in Ingress again. Pokémon Go doesn't really have any kind of worldwide interaction, but in Ingress we actually have players that have made these giant fields that I mentioned that have literally covered the entire Arctic Circle, and everybody who's playing the game can see that.

So, we have the ability to do that in a way that everybody has the same view of the universe. We are incorporating things like all our points of interest that we've developed. So, over the years, Ingress players have submitted interesting art, architecture, local businesses, things like that that they have found interesting, and those things have been incorporated to the game. So, it's a giant crowd-sourced database of the interesting points in the world. That is the basis for the portals in Ingress, the pokestops and gyms in Pokémon Go.

We also have things like climate data, so we know that where I'm sitting right now in San Francisco, it's wet, it's cold. In Arizona, it's hot, it's a desert, and those things can be used to influence what generates in the game. We have live weather data now incorporated into the platform.

Obviously, you need maps. So we're using open street maps, but we provide those map tiles to you that you can then decorate to look however you want. If you looked at Ingress and Pokémon Go, you'd see two vastly different views, and when Harry Potter comes out, you'll see it looks completely different.

Then, like even to the point of our platform, I mentioned the health and fitness data. That health and fitness data is going to be a platform service as well. So, then the players activity, walking, how they move in their life can also be incorporated into the game.

[00:27:14] JM: Amazing. So, if I was to develop an augmented reality game on this platform, what kind of experience do I want from the developer experience point of view? I've already got these tools like ARKit in ARCore. What kinds of APIs do you want to offer me? Like, do I need – Is this just a library I'm importing? Is it a cloud service? What am I doing here?

[00:27:43] PF: Sure. So, there's client and server piece. Initially, you're going to get – With a contest, we have kind of various versions of the client server. So the initial version of what you'll end up getting is just a process that runs on your local machine, right? So, it doesn't have any kind of cloud component to it, but essentially what you're doing is you're going to be building – We support Unity at the moment. So, building a Unity app. We have a native plug-in that plugs into that Unity app that provides things like your map tiles, the location information, secure RPCs. So we do a lot of work to prevent cheating, which I can't talk about. But there's a lot of stuff going on inside the native plug-in that assures that you are who you say you are and that where you say you are to try and fight against the spoofers and such and provides other kind of services to the Unity side.

That then communicates with the server and the server is is – You would then write code on the server. I need to catch a Pokémon. I need to recycle something in my inventory, etc., etc., and those things become what we call game actions and so you would then build those game actions on the server and the server know how to talk to our backend that would be able to give you the spatial entities and other things like, and I can't talk too much about how that works, because that's pretty much our secret sauce. But, yeah. So, it's pretty standard client server kind of development, but we give you a bunch of the pieces and we have a special way of doing this stuff. Really, the key is if you build this naïvely, and I would say when we built Ingress, we did it naïvely. It will cost you a lot of money to run your server.

With Pokémon Go, we were able to do this in such way that it was very inexpensive to run. So, therefore, we could actually make a profit, because as you probably know, only a very small percentage of players actually pay money and we can't run the service unless we can pay for it.

[00:29:43] JM: Brings up an interesting point. Thinking back to the days of Pokémon Go when it was just spiking, it sounded like Niantic has some crazy firefighting stuff that was going on. I mean, not the worst application to have a firefighting thing to contend with, because it's a game, and as important as it was, it's like not the end of the world if something goes down or if there's an outage. But tell me about the firefighting experience. Was it an all hands on deck kind of thing, or was it kind of a – Like, let's just keep our cool and kind of not stay up too late with on-call tickets. What was that on-call experience like?

[00:30:27] PF: So, thankfully I'm a client engineer, although I did have some things that I needed to do with on the client at the time. So, remember that we spun off from Google in 2015. There were 35 people at the company. By the time the game Pokémon Go launched, I think we had about 20 engineers. We had four server engineers that were working on Pokémon go. We did some projections. We did some planning. We looked at other MMOs. We thought about what we thought the traffic was going to be like. We did five times over provision from what we thought we would have.

The game launched in Australia and New Zealand a few days before the US. I don't remember what day it was, but it launched in Australia and New Zealand first. We exceeded our worldwide traffic for the whole world in just Australia and New Zealand. It turns out that the real traffic was 50 times our worst case scenario.

So, we were going crazy. Thank God, we had Google cloud to support us. They literally were moving other people out of the data center that we were in so that they could give us more space. We found all sorts of interesting problems. Like, example, I don't know if you knew this, but the Apache HTTP library has a bug in it. If you have more than 10,000 open connections, it stops working. That's not something that we could load test.

So we were fixing things like that. We found all sort of weird data store contentions and other kind of the things like that on-the-fly and we were kind of like rewriting the server on-the-fly as we went and we had to also – We kind of naïvely launched with no geographic blocking. We just kind of like, "Okay, it's going to be available in Australia and New Zealand, but the whole world is actually populated." So we had so many people side loading the app. So we were kind of like on-the-fly, adding geographic blocking to the server to prevent those areas from being able to get into the game just to control the load. It was totally insane.

I still have a little bit of like – One of our engineers, the art director of engineering gave a talk about the beginning of Pokémon Go just last week, and I was just kind of like, "Oh my gosh! I totally had forgotten about half of this stuff." But it was really surreal. I mean, you flip on – So the day after the game launched, I flip on the TV in the morning before I go to work and the Today's Show was on and that workers is catching Pokémon and I'm like, "Oh my God!" So I changed the channel to CNBC and John Ford is interviewing the CEO of Microsoft and the CEO of GE

and like within 10 seconds of me turning the channel he was like, “So what do you guys think about this Pokémon Go?” I was like, “Oh my God! What have we done?” and went into the office.

Several days later there was this giant somebody on Facebook had organize a giant Pokey crawl here in San Francisco and the servers went down that afternoon before this thing started, and we managed to get the whole thing back up like 15 minutes before the thing was supposed to start. We were just down the street from where the starting point of this thing was. So we are kind of worried that the giant crowds of thousands and thousands of people were going to storm the office if we didn't get things going again. So, it was pretty stressful.

[00:33:39] JM: Yeah, as an added degree of hilarity. Pokémon continues to surprise me with its perennial reemergence as a cult phenomenon. I played Pokémon cards back in like elementary school and it was my on-ramp to playing magic, learning about a ton of new stuff from there. So I will be forever grateful for Pokémon. What do you think – I mean, you sound like you're a games player yourself. What is it about the Pokémon, this totally unrelated software engineering? What is it about the Pokémon brand and the Pokémon game space that is so compelling?

[00:34:16] PF: You know, I don’t really know, because I'm actually a little too old for Pokémon. But if you think about –

[00:34:21] JM: You’re never too old for Pokémon. That’s like saying I’m too old for Pixar.

[00:34:24] PF: Yeah, anyway. Well, I'm a level 40 player now. But the thing about it is that if you think about it, so many people had such nostalgia for it, and there's a lot of depth to it. I mean, all the different types and this guy is good at that and the history, if you read about the different creatures and what their history is and all that kind of stuff. That’s kind of compelling.

But then if you think about the nature of what Pokémon was and is, it was all about someone exploring the world and searching for these creatures. So, when we talk to the Pokémon company about doing this, it was just a totally natural fit. Now we actually will have Pokémon actually in the world, but the beauty of it – So there's a couple of things about this. One,

because people were outside doing it, it's not like a normal videogame. A normal videogame, you're in your living room. Nobody sees you playing.

With Pokémon Go, you see 100 people standing around outside looking at their phones. You walk up to them, "What are you people doing?" "Oh! We're playing. We're catching Pokémon." Then because of the single world instance, if I saw a rare Pokémon on my phone, everybody else saw the same Pokémon in the same place and was able to catch it. So you had this kind of social phenomenon.

Another thing that I think that I'm happy for Niantic to take credit for, but going outside, walking around, getting exercise, fresh air, sunshine, is something that makes people feel good. But people don't necessarily realize that, but then they're playing our game and all those feelings of being outside and all that stuff, they get those, but then they give us the credit for it.

[00:36:03] JM: That's a great point. I love the positive spin on the exercise side of things. Coming back to the engineering – By the way, if that engineer, whoever gave a talk about the early days of Pokémon Go. If they can have an outstanding invitation to come on Software Engineering Daily and tell those stories, I'd love to hear those.

[00:36:21] PF: I think that talk was considered highly company confidential.

[00:36:26] JM: Totally understand. I'm sure there will be some day where the – It is declassified and he or she can come on then. So, what you said about thank God for Google Cloud. Wasn't Google Cloud also saying, "Thank God for Pokémon Go," because this was like the stress test for Kubernetes, right?

[00:36:43] PF: Yeah. I think we were the largest Kubernetes cluster in the world for a while. In fact, it only supported the number of – It didn't support enough instances, and at some point that we got an upgraded version that supported more instances because we needed to be larger, yeah. Obviously, we're below that peak at this point. But –

[00:37:04] JM: Yeah. Well, anyway. I think I heard some interesting stories about that at the time.

[00:37:10] PF: Oh, yeah. The Google people. My friends at Google were like, “Oh my gosh!” Google does this meeting every week internally called TGIF, and they were all like, “Oh my God! You guys were the topic of TGIF this week.”

[00:37:22] JM: Okay. So, getting back to this real-world platform. You laid out a whole lot of detail in this real-world platform. You talked about infrastructure for doing fitness tracking and detecting cheating, and I think you said your secret sauce was kind of like – If I remember correctly, kind of like the rendering of objects in the world and like doing the occlusion stuff, right?

[00:37:47] PF: I mean, that's part of it, right? So the real secret is to be able to do the single system view of the world and do it inexpensively.

[00:37:55] JM: What does that mean? Single system view of the world?

[00:37:56] PF: So in other words, every player in our game – So we don't have separate servers. We play World of Warcraft, like you and I, if we want to play together. We have to say which server we want to be on. With our games, everybody's in the same view of the world.

[00:38:10] JM: Ah, right. So you need to like maintain a consistent experience and it needs to be highly responsive. That's cool. I mean, this sounds like something that's kind of like the spanner, or I don’t know how much you go into like the backend side of things, because you said you’re a client developer. But I remember like the Spanner project from Google is kind of trying to solve the database.

[00:38:32] PF: Yes. So we actually do use both the cloud data store and Spanner in our backend, but we have a layer of our own on top of that that enables us to do this.

[00:38:42] JM: Because spanner probably has some – A little bit too much latency.

[00:38:46] PF: Yeah. I mean, like I said, I don’t want to go into that.

[00:38:49] JM: All right, fair enough.

[00:38:51] PF: I like working here.

[00:38:53] JM: No, fair enough. But, I mean, just talking abstractly, like the problem of global replication at low latency, that's a deep engineering problem and it sounds very interesting. It sound like a worthwhile oat.

[00:39:06] PF: And the key is to do it cheaply, right? So, if it costs you more than you can collect in fees from your players, then you don't exist anymore, right?

[00:39:17] JM: Right. What an interesting engineering problem. That is a series backend engineering problem and edge computing engineering – Okay. You said the edge infrastructure, that's part of what makes 5G appealing to you. Okay. I don't want to drill down to that. I don't want to get you fired. You did talk about the Niantic server a little bit. So I think you said the Niantic server is written in Java. The client is in Unity C#. Can you tell me more about these language choices and what they bring you?

[00:39:46] PF: Yeah. So, I mean, I'm not sure that it was really thought through that much. I think that our server engineers, the initial server engineers had been using Java for a long-time and we're comfortable in it. I mean, okay, so the Google Cloud stuff APIs are Java-based, but you could use Scala or something else, I suppose. Java just has such a vast library of things that if you need to use for whatever purpose you have, that's always there.

I don't know how much far beyond that it went. As far as C# and Unity, well, Unity is the most popular game engine. So if we're going to use Unity, I think we could use JavaScript to C#, and I don't think we're going to use JavaScript.

[SPONSOR MESSAGE]

[00:40:36] JM: Software Engineering Daily is a media company, and we run on WordPress just like lots of other media companies, although it's not just media companies that run on

WordPress. I know of many organization that manage multiple WordPress sites and it can be hard to manage all of these sites efficiently.

Pantheon is a platform for hosting and managing your WordPress and Drupal sites. Pantheon makes it easier to build, manage and optimize your websites. Go to pantheon.io/sedaily to see how you can use Pantheon. Pantheon makes it easier to manage your WordPress and Drupal websites with scalable infrastructure, a fast CDN and security features, such as disaster recovery.

Pantheon gives you automated workflows for managing dev, test and production deployments, and Pantheon provides easy integrations with GitHub, CircleCI, Jira and more. If you have a WordPress or a Drupal website, check out pantheon.io/sedaily.

Thanks to Pantheon for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[00:41:58] JM: People who would not develop in the game world don't really understand how Unity fits into the stack, because they're thinking, "Okay. All right, you're on iOS. You're on Android. So that means you're like doing Android Java development and iOS, Objective-C or Swift, right." "No. It's Unity." Can you help bridge that lack of understanding?

[00:42:21] PF: Sure. There are these things called game engines right? So Unity and Unreal are like two of the most common ones, but there's a bunch of other ones out there. What that does is it essentially provides a set of libraries that make game development easier. So, like rendering of 3D models and shaders and things like that. Then like collision detection and consistent 30-frame per second clock that you get and the ability to do things in a frame. Games are generally written as in you renderer a frame and then you figure out what you need to do next in the next 16 milliseconds, and you render the next frame. So you're in that mode of just trying to do something relatively quickly, and all the things they need to do, and Unity provides that framework for you.

Also, they take your C# code and then compile it down for you into an Android APK and an IPA for iOS through various methods. In the case of – They actually have something called IL2CPP that takes the –takes the C# byte code and converts it into C++ and then they can compile that for each platform. So, it's basically like a vast library of stuff that you need to do to build games. GameUI and 3D rendering and modeling and all those other kinds of stuff that you need to do. Stuff like, "Okay, the artist works in Maya and they built some sort of 3D model of something. You need to import that into the game." Unity knows how to do that and coordinating and syncing things all that kind of stuff.

But we actually do have a substantial amount of native code in addition to Unity. So, like my product – So, I'm the tech lead on something called Adventure Sync, which is the health kit and Google Fit integration into the game. So you could actually think of that as being like a totally separate app that lives inside the game, and that thing is written partly in native Objective-C and, in our case, Kotlin, for Android, but then has a core of C++ code that knows how to do the logic. That happens to run in the background.

So that while you're walking around even if you don't have the game open, it will periodically wake up, sample the fitness data on your phone, send an RPC to the server and so that you would get notifications later, which say, "Oh! You hatched an egg, or your buddy found a candy," which would hopefully, from our perspective, get you to pull the phone out and open the game. But also people enjoy it because they get credit for their activity even when they're not staring at the screen. So that code is written as a native way and is integrated into the game. so we have kind of like stuff going on at multiple levels.

[00:45:05] JM: How much friction is there in building a resource-intensive app that is cross-platform? Because I've written mobile applications before that are meant to be cross-platform and it is just a nightmare even for like a regular crud application. From what I hear, even if you're doing React Native. But if you're doing something like Pokémon Go where you've got really intense resource requirements, is it just layers and layers of difficulty or can you architect the team structure to make this a little bit more feasible and not so difficult?

[00:45:42] PF: Yeah. So a lot of the developers don't really ever touch the native code, rights? So they're their working in Unity, and Unity does a pretty good job of generating something that

will perform on both platforms. The difficulty comes in – So, remember, I mentioned our native plug-in that provides a security in RPC and all that kind of stuff. That now has parts that are written cross-platform and then native code that's written for both environments. So you have to imagine, the building of that is fairly complicated. You have to like pull the – Like in the case – Adventure Sync is the same thing where there's this native C++ core, but then there's like, "Oh, here's the Objective-C parts. Here's the Kotlin parts. So the Objective-C parts, the integration between C++ and Objective-C is relatively straightforward." But for the Koltin parts, now I have to go into JNI. I have to build like an interface. Okay, I have to decide – I'm getting health samples from both platforms. Those health samples look vastly different.

So now I have to take those things. I have to convert them into some common format that the C++ code can ingest, and then I have to control separately, "How does this thing launch? How does it know when to stop? What does it mean to be done? What is it mean for the app to be launched in the background? Android services versus iOS background execution," all these kind of stuff? So it turns into being a pretty complicated problem.

I feel like I've done a pretty good job with it, because our server-side folks, when I gave a talk about architecture for this thing they said, "Huh! I thought this was a simple service?" I said it's a simple service because you get two RPCs. It's not a simple service of all the stuff I have to do to make it work.

[00:47:27] JM: I want a little bit more about the real world platform. We really just glossed over it. I'd like to just revisit it and tell me, again, what you're trying to offer on a high-level and what kinds of game development opportunities you see. What kind of games would you like to see – If you could name the games that you would really like to see people build on your AR –

[00:47:52] PF: Yeah. Yeah. So I have a personal favorite, and that I've kind of pitched internally a little bit and I don't know if it's ever going to happen. So, if you think about Ingress, we use only the POI, there's a map, but we don't use anything on the map. There's just the points of interest. Pokémon Go, we use climate and weather and all these kind of stuff. But the roads, nobody use – We don't use the roads for anything, and I really want to build a tower defense game that is in the real- world.

So you walk around and you place your towers down and then you launch the attack against the other members – The other players, and they use the road network as the wave for these creatures to – Whatever they are, robots, creatures, tanks, whatever it is, to move around the world. I would love that game.

I don't know if we're going to build that. I would love to see somebody try that. I don't know what the teams that we have. We have several teams that are selected. I think the contest is starting pretty soon, but I don't know exactly what they're working on. But if somebody is building that game, I would love to have it.

[00:48:51] JM: What are the parameters of the contest?

[00:48:53] PF: We requested input, right? So people got to make a proposal for what they were going to do. Then from that list of people, we selected some number. I think we're trying to go beyond just games, right? so it's going to be – We're trying to select a variety of different types of experiences –

[00:49:06] JM: These are like external people that are throwing ideas at you?

[00:49:09] PF: Right. Totally external people that gave a proposal, and we have some funding, and then there's prices at the end, like a million-dollar prize.

[00:49:17] JM: Okay.

[00:49:17] PF: Yeah.

[00:49:18] JM: Okay. Very cool. So with the tower defense example, walk me through what I would get out of the Niantic platform. If I wanted to build that, what would my workflow be?

[00:49:29] PF: Right. So you would you would get a map. We would tell you this is a road. This a park. This is whatever on the client. But on the server, you would also maybe have that information as well, and you would have to build some sort of graph. It would be up to you to figure how to do that given the map tiles that you get. You would have to be able to kind of

analyze it and figure out where the roads were so that on the server – Because the server is going to be responsible for creating, generating the creatures or whatever it is. I place a defense tower at a certain spot in the game and that sends of RPC to the server and one of the game actions that you build, it says, “Look in my inventory Place this thing at this location, and then activate it.” Then something else in the backend would say, “Oh, yeah. There's this creature moving along in this path,” and you would have to do some action that would cause it to at the thing or whatever. Then there would be more game actions.

So, basically, what you're doing is you're building your game and your client. You’re deciding a set of –We use Google RPC and protocol buffers. So you essentially define your protocol in the protocol buffers and say, “Here's the list of things I need to do to make this game work as far as these 25, 30, whatever game actions that you build. On the server, you have to build those actions. On the client, you're going to send RPCs and consume the results you get back and render that stuff on the screen so the player can see it. Then there's probably going to be something like this tower defense or probably some sort of dataflow thing that needs to run to update the position of all these stuff and all that kind of thing. So that would be supported on the platform as well. You have some things that are real-time and other things that you kind of do in the background on the server.

[00:51:09] JM: So as we begin to wrap up, I guess I’d love to know just a few more kind of deep engineering problems that you've encountered or curiosities. I guess, more specifically, what is it deep engineering problem that you've had to solve recently? I just want to better understand what are the challenges in augmented reality development.

[00:51:32] PF: Right. I think a lot of these stuff that we've been doing, this is not me personally, but as a company. Our AR team has been doing a really good job of taking basically what the two platforms give us as far as ARKit and ARCore and turning that into something that’s a shared experience and that has low latency, and I know they’ve been working really hard on that.

This is basically a company that we acquired a few years ago, and I think the team is grown since then. But that's I would say a very hard problem that they’ve had to solve, and I don't have a lot of insight into it, because it’s not what I do. But that's a big deal for us.

We have another team that's working on the inclusion types of things, and that's a big machine learning problem where we're trying to be able to do occlusion, but only with a single camera. Also, a very challenging problem involving – I think as you move through the world, you can maybe stitch together the images that you see in such a way that you can kind of develop that 3D model on-the-fly, and we have a demonstration video. I mentioned the codename Neon demo, and there's a demo that as well on our YouTube channel if you are interested in taking a look.

Personally, I've been doing all these background stuff. So, my challenges have been both just dealing with the vagaries of the various platforms. As you know, there's documentation and there's how things actually work. So, a lot of it has been, "Huh! I didn't expect it to work like that." So then we have to spend a lot of effort to fix that.

A quick example from just this week, we are doing some – Because on Android we have multiple processes running the services that we have for doing background things running in a separate process, and we're using – So we use a content provider to do shared preferences between – Because if you use a regular shared preferences, it doesn't work across cross-processes. But it turns out, the way the content providers are implemented, the data actually survives app reinstall.

So, I had a flag that said this thing was initialized. The thing I'm working on now was initialized. I deleted the app and reinstalled it and instead of initializing it, I thought it was already initialized. So it didn't initialize. So you just have weird things like that that I don't know that that's documented anywhere, but you discover things like that as you're going.

[00:53:50] JM: Final question.

[00:53:52] PF: Sure.

[00:53:52] JM: We have this term mixed reality that I think suggests that augmented reality and virtual reality are on this gradient, and that there will be some applications where they're going

to merge, like maybe of AR for part of the application and then you enter a VR world. How do feel about this mixed reality stuff and virtual reality more broadly?

[00:54:14] PF: So, I'm personally not a huge fan of VR. I mean, I think it's kind of cool in some limited situations. I mean, I've seen this thing where you can become like a Terminator and play with your friends, and that seems really fun.

I think from Niantic's perspective, we want to get people out of their living room and go outside. So, it's not really going to work. I mean, you can't really work around. I don't know where the term mixed reality comes from. It seems like that's a thing that Microsoft talks about a lot. Augmented reality, mixed reality to me seem like the same thing.

Yeah, I think that you're going to see – Currently, if think about our progression as a company, so Ingress didn't really have any kind of camera-based AR. Pokémon Go right now is pretty limited. There's the Go Snapshot thing that we just did and the AR encounter things that we've had. There's more coming there.

I think when Harry Potter comes out, you'll see that we've gone a little further. Then as more products come out, you're going to see us heading more and more where there'll be more AR stuff. But we have to keep that kind of thing relatively small just because on the fact that we're on a mobile phone means that we don't want to make it so that people have to hold their phone up all the time.

So you have to kind of like curate when you do AR stuff in such a way that it's enjoyable for somebody, but it doesn't make it so that they have to be doing the whole time they're playing the game, or else you're going to lose a lot of people, at least in my opinion.

[00:55:44] JM: Well, I remember seeing Sergey Brin talk I think at TED a long time ago just making the same argument, like, "We should not be craning our next downward looking at our phones all the time. We should be out in the wild. Computing should be ambient, and it should be harmonious with our roots that make us want to spend time outside and exercise."

It's nice to see that embodied in Niantic. So, Paul, I look forward to seeing the continued developments of Niantic, and thanks for coming on the show.

[00:56:17] PF: Well, thanks for having me. It was great to talk to you.

[END OF INTERVIEW]

[00:56:22] JM: Deploying to the cloud should be simple. You shouldn't feel locked-in and your cloud provider should offer you customer support 24 hours a day, seven days a week, because might be up in the middle of the night trying to figure out why your application is having errors, and your cloud provider's support team should be there to help you.

Linode is a simple, efficient cloud provider with excellent customer support, and today you can get $20 in free credit by going to linode.com/sedaily and signing up with code SEDaily 2019. Linode has been offering hosting for 16 years, and the roots of the company are in its name. Linode gives you Linux nodes at an affordable price with security, high-availability and customer service.

You can get $20 in free credit by going to linode.com/sedaily, signing up with code SEDaily 2019, and get your application deployed to Linode. Linode makes it easy to deploy and scale those applications with high uptime. You've got features like backups and node balancers to give you additional tooling when you need it. Of course, you can get free credits of $20 by going to linode.com/sedaily and entering code SEDaily 2019.

Thanks for supporting Software Engineering Daily, and thanks to Linode.

[END]