EPISODE 838

[INTRODUCTION]

[00:00:00] JM: Augmented reality applications can be used on smartphones and dedicated AR headsets. On smartphones, ARCore from Google and ARKit from Apple allow developers to build for the camera on a user's smartphone. AR headsets, such as Microsoft HoloLens and Magic Leap, allow for a futuristic augmented reality headset experience. It's still early days for augmented reality. The most prominent use of augmented reality today is gaming. With a notable example being Niantic's Pokemen GO.

Tony Godar is a software engineer who works on augmented reality and virtual reality applications. Tony joins the show to talk about his day job working on virtual reality experiences and an AR game he built called ARhythm. Tony was the winner of the FindCollabs Hackathon, our first hackathon, and we also discuss his experience working on this ARythym project on FindCollabs. We are having the second FindCollabs Hackathon.

You can go to findcollabs.com/open and find out more about that hackathon. We're giving away $2,500 in prizes for projects like machine learning, music, visual art, podcasting, data visualization, cryptocurrencies, computer games. These different categories are things that I'd love to see cool projects built with on FindCollabs. FindCollabs is a place to build projects and find collaborators. I hope to see you there, and if you want to find out more, go to findcollabs.com/open.

[SPONSOR MESSAGE]

[00:01:46] JM: When a rider calls a car using a ridesharing service, there are hundreds of backend services involved in fulfilling that request. Distributed tracing allows the developers at the ridesharing company to see how requests travel through all the stages of the network. From the frontend layer, to the application middleware, to the backend core data services, distributed tracing can be used to understand how long a complex request is taking at each of these stages so the developers can debug their complex application and improve performance issues.

LightStep is a company built around distributed tracing and modern observability. LightStep answers questions and diagnosis anomalies in mobile applications, monoliths and microservices. At lightstep.com/sedaily, you can get started with LightStep tracing and get a free t-shirt. This comfortable, well-fitting t-shirt says, "Distributed tracing is fun," which is a quote that you may find yourself saying once you are improving the latency of your multi-service requests.

LightStep allows you to analyze every transaction that your users engage in. You can measure performance where it matters and you can find the root cause of your problems. LightStep was founded by Ben Sigleman, who is a previous guest on Software Engineering Daily. In that show he talked about his early development of distributed tracing at Google. I recommend going back and giving that episode a listen if you haven't heard it. If you want to try distributed tracing for free, you can use LightStep and get a free t-shirt. Go to lightstep.com/sedaily.

Companies such as Lyft, Twilio and GitHub all use LightStep to observe their systems and improve their product quality.

Thanks to LightStep for being a sponsor of Software Engineering Daily, and you can support the show by going to lightstep.com/sedaily.

[INTERVIEW]

[00:03:57] JM: Tony Godar, welcome to Software Engineering Daily.

[00:04:00] TG: Thank you for having me.

[00:04:01] JM: Describe the state of augmented reality applications today, including gaming and nongaming.

[00:04:09] TG: It's still quite early. I guess the big breakout was the Pokemon game that went out and then that kind of made it mainstream. But I think that kind of gave everyone a false sense of security that it was a mature technology. I didn't even realize until I kind of did this hackathon really how immature or how early it is in AR, AR altogether. So it's got a huge amount of potential.

[00:04:37] JM: Okay. So there's a lot of potential, but there's not really any applications today. Can you help me understand the golf between those two?

[00:04:49] TG: There's pieces. So we have like – Essentially, we have a lot of tech demos out there. We have a lot of viral videos out there, that really the viral videos are just little few second snippets. There isn't really – I guess there's some business applications or like the IKEA app, which they've actually pulled it off. There's some measuring tapes and so forth, but really no big breakthrough yet, and it's going to take time. I think phones are evolving to be able to handle it seamlessly and people are getting more used to it.

[00:05:24] JM: Do you think it's a technology issue, or do you think it's more – I mean, you're holding out your phone. It's kind of a weird situation, we have to do that. I guess it kind of makes sense for something like gaming where you're willing to kind of bend over backwards in order to play the Pokemon game and catch the Pokemon, especially because everybody else around you is doing it. But is there something wrong with the form factor? Are we going to need the AR glasses in order to make this a reality?

[00:05:54] TG: We're looking out. I definitely think we're going to need the glasses. Right now, kind of the big popular AR app or killer app right now is kind of the masks that you have on Facebook, Instagram and Snapchat. Especially with cameras and the phones evolving are really good, and those are ones that are heavily used. I think from there, they're going to evolve that in other direction. By that point, glasses are going to show up.

[00:06:23] JM: Tell me about the state of the AR tooling. If you want to build an augmented reality app today, what is the tooling?

[00:06:32] TG: So there's ARKit and ARCore, and function-wise very similar. The Apple ecosystem is a lot more mature, which I discovered. The Android one, because of the random handsets you have out there, some support ARCore, some don't, and it doesn't really hit me with this hackathon. I was trying to find people to collaborate who had an Android phone that supported this. It was really hard. Because Apple had its standards and success. Pretty much everyone with an Apple phone supports it nowadays.

I develop primarily, I guess, initially to Android device, and it was okay. We had some frame rate issues and it worked okay. I was quite shocked when I got it working on an iPhone, even a lower-end iPhone, how well it worked. So I think Android's got to catch up a bit software and hardware. Yeah, I think I –

[00:07:28] JM: So ARCore. ARCore is the – Or ARKit. I'm sorry. That's the iOS version. What's the API? So I developed an ARKit application with somebody else a while ago. It was about a year and a half ago, and I remember it was pretty cool, but there were a lot of annoyances, like in order to get 3D models, we kind of had to like make the 3D models ourselves. It was kind of hard to figure out how to put those 3D models into the world and he workflow for building them. There were something involved with Unity. I don't know. The story required a lot of Googling and stack overflow searching, as any early technology does. But give me a little bit more detail on what ARKit requires of you today.

[00:08:14] TG: So I got lucky and that I discovered early on in the hackathon about Vue or React, which kind of sits over and allows you to easily access both ARKit and ARCore depending on which device you're on. So that was a big win. Also, it's in React Native. I'm new to the language, but it's easy one to kind of figure out, and it worked really well. So I think short timeframe, getting something proof of concept. That was a big help.

Realistically though, there are still a lot of challenges. So a lot of – I guess the 3D geometry, the physics, a lot of it has to be coded. Importantly, the model is quite a pain. I don't have a Mac or I didn't have a Mac at the time, and the tools to import them were only available on Mac. So I just pretty much took an existing model and retextured it because of the lack of time.

However, I think things are evolving. There's a tool that Facebook is coming out with or had ultimately evolved a lot. So Facebook ARStudio.

[00:09:17] JM: ARStudio. I remember seeing this thing.

[00:09:20] TG: Yeah. So I didn't use Spark for the hackathon, because it was, again, only Mac only and quite limited. But even – I think right now, today and tomorrow, is the big F8

Conference, where Facebook is announcing what they're doing in the next year. I saw some news that they're going to add quite a bit more functionality to Spark and kind of make it a lot more approachable. One thing that really interests me was the ability to bring in music and actually code to the music. This one, it's a lot less coding and it's a lot more kind of a node base connect the box type system. It looks quite impressive.

[00:10:02] JM: It's surprising to me that a framework built by Facebook that sits over both ARCore and ARKit, which are themselves quite nascent technologies. It's surprising to me that Facebook's technology that sits over those nascent technologies is useful. Because, I mean, usually, in order to get a point where you have this additional API layer on top of existing API layers, usually only happens when the underlying API layers are a little bit more mature. Did it surprise you that Facebook's thing that sits over the two immature things actually worked well for you?

[00:10:44] TG: Well, I think the Facebook's Spark side of things, their big advantage is there can be – They are embedded in Instagram, embedded in Facebook and embedded in Facebook Messenger. So the Vue core that I used has nothing to do with Facebook. It's that they used React Native language, which is loosely – Well, it is originating from Facebook. But ARStudio doesn't use a coding language. It's all visual.

I think one roadblock for the small little proof of concepts is kind of – If I use Vue or React, a small 30-second experience for someone to download from the store is very unlikely. However, if I did the same thing in Facebook's ARStudio and all my friends and all their friends link to it and could actually try it from within Instagram or Facebook, that would reach way more people. I think that's the big advantage on their side.

[00:11:46] JM: So you developed this game, ARhythm. Describe your idea for ARhythm.

[00:11:54] TG: Essentially, I wanted to – So we were in Easter break, and had some free time. I was off work. Kids were off school, and your hackathon came along and I wanted to kind of do something with them. So I wanted to bring music into some sort of a game. This was kind of like the first thing that hit me is kind of music beat type game with AR, and it just kind of snowballed from there. Seeing which technology is available and the potential. Yeah, it just kind of naturally – The natural choice for me to go for it.

[00:12:31] JM: The market for music-based games has always struck me as – I mean, it's just funny that Dance Dance Revolution was such a popular game, but it seems like that was really the only music-based game that really took off. Are there any other examples of music-based games that really took off?

[00:12:51] TG: Oh, many, many.

[00:12:54] JM: Or I guess there's Guitar Hero also. Sorry.

[00:12:56] TG: The Guitar Hero. Those are massive on the consoles. There's a lot of singing and dancing. The Sing Stars Series that Playstation came up with. I think just Dance Series that Ubisoft came out with. Yeah, my kids, when they go to their friend's house, they still play those games with their friends.

There really is still the social nature of them. They're always going to be around and they're evolving. In the VR world, kind of by far the most popular VR game right now is a music game.

[00:13:26] JM: Really? What is it?

[00:13:27] TG: Beat Saver.

[00:13:28] JM: Oh! So that's the most popular game in VR.

[00:13:31] TG: Really.

[00:13:32] JM: What do you do in Beat Saver?

[00:13:34] TG: You have two lightsabers and there are objects coming at you and you slash away at them at the beat, and it's a lot of fun.

[00:13:44] JM: And your idea for ARhythm was kind of like an augmented reality version of this.

[00:13:51] TG: Exactly. So that one kind of sparked my interest in how would you do this in AR, and I learned a lot from trying to implement a very similar sort of mechanics. Yeah, it's got potential. I think the mechanics are very different, because you're not immersed. You don't have, I guess, controllers in your hands. It's got a lot of potential.

[00:14:14] JM: Tell me about the spec required for ARhythm. When you sat down and you're like, "Okay, this is something I'm going to build." Tell me about the process of scoping it out.

[00:14:25] TG: Essentially, because it's quite limited time, a lot of it was really the learning curve. So it was really up until the two days or last two days of the hackathon, I was just kind of learning the language, learning how to do things. So I had a bunch of little test cases. So I knew what it is capable of, so to bring in a 3D model, to be able to move it and some sort of sound interaction. Then I guess from there, the whole goal was to have kind of a 3D object coming towards you and to be able to touch it or slash it and for it to explode. Then next phase would be kind of to time that with the music and see if it's fun. Overall, yeah, it worked quite well. Not as fun as I expected. But I think it's got potential. Since then, I've made some tweaks and checked that into the GitHub repo and it's a little bit finer now.

[00:15:17] JM: Do you find Beat Saver fun?

[00:15:19] TG: Oh, yeah. It's a lot of fun.

[00:15:21] JM: Okay. So what was the hardest part about getting to a functional MVP?

[00:15:29] TG: I guess the fine details. I struggled a lot with controlling the 3D shape multiple instances of it in real-time and for it to react quickly so that when you touch it, it explodes and vanishes. So that took a lot of back and forth. Also, doing that in a way that it performs well. It was just down to my lack of knowledge of using React Native. Once I learned a little bit more about React Native, it fit in quite well. I guess the challenge at the end was when I saw the potential of where I could do more with it, but there just wasn't enough time.

[SPONSOR MESSAGE]

[00:16:18] JM: DigitalOcean is a reliable, easy to use cloud provider. I've used DigitalOcean for years whenever I want to get an application off the ground quickly, and I've always loved the focus on user experience, the great documentation and the simple user interface. More and more people are finding out about DigitalOcean and realizing that DigitalOcean is perfect for their application workloads.

This year, DigitalOcean is making that even easier with new node types. A $15 flexible droplet that can mix and match different configurations of CPU and RAM to get the perfect amount of resources for your application. There are also CPU optimized droplets, perfect for highly active frontend servers or CICD workloads, and running on the cloud can get expensive, which is why DigitalOcean makes it easy to choose the right size instance. The prices on standard instances have gone down too. You can check out all their new deals by going to do.co/sedaily, and as a bonus to our listeners, you will get $100 in credit to use over 60 days. That's a lot of money to experiment with. You can make a hundred dollars go pretty far on DigitalOcean. You can use the credit for hosting, or infrastructure, and that includes load balancers, object storage. DigitalOcean Spaces is a great new product that provides object storage, of course, computation.

Get your free $100 credit at do.co/sedaily, and thanks to DigitalOcean for being a sponsor. The cofounder of DigitalOcean, Moisey Uretsky, was one of the first people I interviewed, and his interview was really inspirational for me. So I've always thought of DigitalOcean as a pretty inspirational company. So thank you, DigitalOcean.

[INTERVIEW CONTINUED]

[00:18:26] JM: Seen the videos of the game. Basically, wherever you're standing, you have these cubes flying at you while music is playing and then you kind of like take actions in the AR world to make those cubes explode. Just tell me about what is required to program augmented reality cubes flying at you in the current world of AR APIs.

[00:18:54] TG: Essentially, the key was giving, I guess, creating one cube with a texture, lighting it in a way that it was natural for an environment. So if you look at the video, when I'm indoors, it looks a bit odd. But when I'm outside, it looks really good. Just simply the lightings fixed to one environment, and that was the outdoor. In the future, you could have it take the lighting to the room, and that lighting will be matched. From there, I chose I think about six of those and it just takes one 3D model and it uses it six times. That performs quite well. The spinning animation on it, and then it's about moving it over a period of time. So I'm not updating every frame. It's just got a start point, got an end point. Then at the end point, it has an event. Then the event is just to bring it back to the front of the scene. Then at a point where it's close enough, you check whether it's been touched. If it's been touched, you explode it and then trigger a sound effect.

[00:19:57] JM: When you're programing a cube to move through space, is it a 3D Cartesian model. What's the model of describing where an object is in space around you in an AR application?

[00:20:14] TG: Your camera, when you start off – So it's calibrating to one spot, and that's essentially 000 and XYZ coordinates. Then you're based on that. So once it's calibrated, the objects are flowing towards you. So you can sidestep and watch them come past you or you can stand right in the middle and have them come right at you.

[00:20:36] JM: Were you able to get the cubes to move in a way that was syncopate to the rhythm?

[00:20:43] TG: No. So they're pretty much coming at you at a random speed right now. In the future, yes, I need to get that in there, because that's very important to it. However, the whole touching isn't really the direction I'm hoping to go. I think it's more of avoiding and catching would be a lot more interesting, or is a lot more interesting.

[00:21:08] JM: Yeah. Do you have any engineering ideas about how to make a general model for synchronizing a moving object in a game to a piece of music?

[00:21:21] TG: At this point I haven't really planned it out. So not at this point. So, essentially, the data I'm using – So Beat Saver allows people to submit their own levels. So they're publicly available. So they include the music and also the timings. Right now, we're using those timings with those songs, because they work pretty good already.

[00:21:44] JM: That's cool. Wait. But how do you get access to that data? Did you say it's open source?

[00:21:50] TG: They have on their website. You can download a ZIP file. The ZIP file contains a JSON file and the music file.

[00:21:56] JM: That's pretty cool.

[00:21:57] TG: Yeah, and there's a lot of songs and a lot of people submitting songs.

[00:22:00] JM: That's really cool. I think I told you this, but just for the listeners, when I was in college, I made a game and had kind of a similar inspiration, where I would have a song and then I wanted the song to generate the stock prices that would move according to – Like syncopated with the time of the music. My way of doing it was – Actually, somebody else told me about this. Originally, I was thinking, "Okay, you could take the sound wave and then you could make a bit map from the sound wave, and the bit map from the sound wave could be used to translate into intensity levels, right? Because it would be an image of the sound wave, and at the more intense parts of it you could say, "Okay. This is a more intense moment," and then you kind of have an up and down graph of intensity overtime and you can derive the movements of the game from that. But my brother actually told me to use a Fourier transform, which can be used to modify like a highly-dimensional space into – I don't know if that's exactly how you would say it, but I think a highly dimensional space into just kind of a lower dimensional up and down motion perhaps. Anyway, it's kind of a random and interesting domain. Like how do you translate music into a more workable function to have gaming syncopated to the music.

[00:23:27] TG: So I guess part of this hackathon, the research part of it I've played as many I guess music and beat games as I could find on the Android and iOS store. What I found is a lot of them, you're allowed import your own music and they use similar sort of algorithm, where it's generating I guess what you go towards or what you clicked on or whatever. What I found was if it isn't 100% accurate, it's horrible.

[00:23:59] JM: It's horrible. I mean, that's how my game ended up. It was like it just looks like – The things that were generated were completely random and had nothing to do with the music.

[00:24:09] TG: So I really like the idea. For something like this to add new levels, you'd probably need a level editor, where you're going through and repeating, repeating, repeating until you really perfect it. The timing has to be exact. Once it's exact, then it's fun. I think if you make one or two mistakes, then you're totally going to get someone out of flow and then it's not fun anymore. These music games, it's essential that someone gets into flow.

[00:24:32] JM: That's really cool. So I guess you get there and Beat Saver. You're able to really just syncopate yourself with the music as you're slashing these objects.

[00:24:42] TG: Yeah, totally. It also helps a lot, because you're wearing a VR headset. So you're completely immersed. You're completely surrounded, and what you have in your hand are glowing lightsabers, which is good fun.

[00:24:53] JM: To give people more context for who you are, you have spent a long time in the gaming industry and you currently work at a company called MelodyVR. Explain what MelodyVR is.

[00:25:05] TG: So MelodyVR is – Essentially, we have an app for the Oculus Go and Gear VR and I guess the Quest that's been just announced today. What we do is we have live music concerts from huge amount of artists where we've gone in and recorded up to 10 jump spots. I guess purchase these shows and watched them from end-to-end. Go around to different spots. Yeah, it's quite exciting.

[00:25:37] JM: How would you compare the maturity of the virtual reality world to that of the augmented reality world?

[00:25:44] TG: Very different. Very different. I think virtual reality lends itself to very different experience, and virtual reality definitely – Because it's immersive, you don't have to worry about – So a lot of the headsets right now don't' have six degrees of freedom. So you're kind of stuck in one spot. That's not a problem. That would be impossible in AR. So that bought it a lot of time. So the whole immersion factor, once you block out all of your field of view, you can totally get into it and you don't have those distractions that you would in AR.

I think AR – Once we have some headsets out there, once we have some glasses, which don't look really dorky, then there's a lot of potential for those. Until then, there's going to be a lot of improvements in phones and cameras that will make the phone experience slowly better and better.

[00:26:36] JM: The developer tooling for VR, how does that compare to that of AR?

[00:26:41] TG: I guess with VR, you have a lot more power in general. Initially, you're doing your development on a desktop PC. So way different than a mobile phone. Later on, when the Oculus Go and the Gear VR came in, there was a lot less power, but it's quite optimized like a game console, especially the Go. Whereas as a phone, you're sharing your resources with a lot, and the updates and the different variations, it's much trickier to target. Such a wide range of advices. But when an actual headset device comes out, that can hit the mass market and be one fixed piece of hardware. There's a lot of potential there.

[00:27:24] JM: Do you think that both of these technologies, AR and VR, will these both eventually be mainstream technologies?

[00:27:32] TG: Oh, definitely. Definitely.

[00:27:33] JM: Definitely.

[00:27:34] TG: Yes.

[00:27:36] JM: What makes you so sure about that?

[00:27:38] TG: I've been involved with VR and 3D for so many years now. So probably two, three years before Sony announced that even. It's been really part of my life in that aspect. Yeah, I just can't imagine. Because it's evolved so much and because it's evolving at such a high pace, it's just going to integrate into our lifestyles. AR is going to be a major step to take a lot of the tricky bits in VR where it was too isolated and not social enough and that it will really bridge the gap. By that point, the hardware is going to be advanced enough that it will be able to push – Yeah, it will be able to have enough horsepower to push everything.

[00:28:20] JM: How deeply have you looked into the hardware? What are the hardware bottlenecks to getting – I guess, device is small enough, or do you just know this on the sense that it's a general trend?

[00:28:31] TG: Well, I guess the HoloLens is getting there, but it's still a massive device and it's still expensive to buy. I think about a year ago, I think Facebook announced one of their AR headsets that they're looking into. One of their head researchers talked a lot about it, but really said it's not until going to be the size of a pair of glasses that it will really take off. I kind of agree with him on that.

[00:28:56] JM: What do we need to get there? I guess we need like super small chips? How hard would it be to get to that size of chip? Do you have any idea in the state of the chip design?

[00:29:10] TG: Essentially, it's going to be a wireless connection to your phone in your pocket, and will be a display embedded in your glasses with enough batter power and it will also going to have to have some gyros and some cameras which don't creep people out. So I guess a lot of challenges to overcome.

[00:29:30] JM: I don't know. The wireless –

[00:29:30] TG: But it's definitely close.

[00:29:31] JM: The wireless connection. Do you think that's okay? Because, I mean, if you have some kind of – Like my Bluetooth headphones. There's hiccups all the time. Whenever there's a Bluetooth hiccup and my podcast starts to sound weird or choppy, it's like it's so disorienting. I compare that to how disorienting it would be if I was wearing AR glasses or VR glasses. I think it would just too problematic for me to want to even engage in the experience.

[00:29:57] TG: Yeah, I guess if the battery life is a few hours, it's going to be very painful. Kind of like the early smart watches, where it could barely last a day. It was really painful.

[00:30:06] JM: But it's not even batter life. It's like short field network connectivity. But maybe that's just an engineering problem that remains to be solved.

[00:30:13] TG: Yeah, I think it's just a matter of time. It's a matter of time. I guess, likely, the first few generations will have a wire coming out of the arms of the glasses.

[00:30:24] JM: Sure, and maybe that's not so bad. I mean, we, for a very long time, or at least even people who still don't have Bluetooth headphones, all due respect to them. They still have a cord coming out of their ears and going into their pocket. So maybe that will be a fine form factor if the glasses were good enough.

[00:30:42] TG: Yeah, I think that's the next logical step. Once it's wireless and once the batteries and I guess everything is in there, then it's really going to be everywhere.

[00:30:51] JM: You spent – What was it? 20 years of your career in the gaming world before you started working at MelodyVR, or was it 10 years?

[00:31:02] TG: Yes. So I was at Playstation R&D for a good 14 years. Then before that, I was working on Playstation 2 game for a couple of years before that.

[00:31:11] JM: What has drawn you to the gaming industry?

[00:31:15] TG: It's kind of accidental. I never really had the – I guess, expected that I'll be going to the game industry. But after finishing college, I went to Japan and ended up getting a job there and not really game related. But at the time, PS2 was at its peak. So everyone wanted to make a PS2 game. Kind of like when the iPhone was at its peak and everyone and their granny were making iPhone apps. At the time there, it was PS2 games. So I jumped on that and really never looked back.

[00:31:47] JM: So your experience building this thing was part of the FindCollabs Hackathon. You actually won the FindCollabs Hackathon for your ARhythm application, and I thought that was well-deserved, because you're really grinding on it and the end result was something that was quite cool. I would love to get your take on both the FindCollabs platform and the hackathon. I mean, I don't want this to turn into an ad for the product I'm building. But I really just like to get from a feedback perspective, what you think of the platform.

[00:32:25] TG: I think for me it was really timing and just kind of showed up. I was listening to your podcast. You mentioned it. I check it out and I really liked the idea. So, so many of these collaboration sites out there are more towards someone trying to make a quick buck and not really towards learning, or even just the experienced developer who wants to do some coding on the side for fun. This one really fit that niche. Timing-wise, I had free time, and there was a good momentum there. You guys had your meet up and I could join in remotely and talk to you guys and that sort of thing, and I think there's – Yeah. Definitely, as you get more people on to this site – I guess the most frustrating thing for me was trying to get people to join. I think it's easy for someone to start their own idea and bring some of their friends on it. But to kind of convince a stranger to join their idea, it's still not quite there yet. But it will get there. It will get there. I think once some universities and once I guess some clubs, coding clubs and so forth start getting involved with this sort of thing, it could really take off.

[00:33:33] JM: That's my sense too. Since you tried out FindCollabs, and I think you embody the kind of the spirit of developer that I really wanted as an early user of the platform, maybe the idea kind of resonates with you. But like there have been numerous times where I've just been – Back when I was a software engineer in industry, like I think every weekend, almost every weekend, I would be hacking on something. Whether that was like a music project or an artistic project or an editorial project or a podcast or something in software, a Rails app or something, I always wanted to work with other people, but I could never find people to collaborate with.

What I wonder is, is that like an inherent aspect of like the way that we just exist as software developers? Do we just want to be working in isolation on these kind of weekend projects? Or is it like is there a problem with the social networks and the social tools that are just inhibiting us from doing this?

[00:34:41] TG: Good point, yeah. Because I probably want to go on Facebook and say I'm working on this project for the next 48 years. It's not really something people go around showing off to their general social friends. Yeah.

But overall, always when I've hired younger engineers, one of the main things was kind of do they code for fun or do they code because they have to? Those that do it for fun, it's massive. The quality of their output, and I guess they're just enjoying coding night and day. It's night and day. So having somewhere where people who enjoy coding used to have fun with coding. It's a good break. So, for me, the whole fact that I could learn and play with React Native was actually a lot of fun.

Pretty much, if I join another hack – Well, I will join the next hackathon, and what I'll do is try and find something out of my comfort zone again, but even out of my comfort zone.

[00:35:46] JM: Very cool. So how much of your – You could be honest with me here. How much of your entry was motivated, or how much of your use of FindCollabs was motivated by this hackathon semi-competitive element?

[00:36:00] TG: Competitive was helpful, definitely. Having a prize is always great. But I think what motivated more was the fact that others to compete against and others who are also wanting to win. So back when I was at Playstation, we did quite a lot of hackathons in the last five years or so, and I really enjoyed doing those.

When I moved to a startup, we don't have that sort of thing, but kind of the whole startup culture is a little lot more high-paced. So it's almost essentially constant hackathon. But, overall, a safe place where you're competing against people who are kind of the same mindset where they're learning and enjoying it is kind of a rare thing.

[SPONSOR MESSAGE]

[00:36:54] JM: Whenever someone asks me where they should start if they're trying to learn to program, the answer is easy; freeCodeCamp. FreeCodeCamp is the best place to start your programming journey. You can learn to code for free, and there's a support of community of millions of coders. There're projects that you can work on. You can get experience by coding for nonprofits. Again, it's entirely free, making it easily the first place that you should start your programming journey as you decide to learn to code.

If you want to go to a boot camp from there or take some online courses that cost money, those are options as well, but there's really no downside to starting with freeCodeCamp. We've done several shows with Quincy Larson from freeCodeCamp, and his true mission is to make coding accessible and free.

FreeCodeCamp is also open source and there is a nice onramp to working on open source if you want to start with freeCodeCamp by just taking the courses and then eventually you can become an open source contributor by working on freeCodeCamp itself. So it has that kind of cool meta element to it.

Thank you to freeCodeCamp for existing and being my go-to source of referrals to get started with your coding career or your coding education. You don't necessarily have to be a career programmer. Thanks, freeCodeCamp.

[INTERVIEW CONTINUED]

[00:38:39] JM: So we just built this enterprise thing, where basically people can log in with their enterprise Gmail and then they get a private instance of FindCollabs, and I'm trying to figure out if this is something that people want. I mean, we're trying a lot of things right now, and I'm not really sure if I should just focus on one thing and double down on it. We've kind of got this open network idea. We've sort of got the hackathon thing that has some traction, but then that's paid acquisition and is that really real users. Is that a legitimate user? Is that a long-term user? Maybe we're a hackathon platform. I don't really know. Do any of these use cases sound like particularly things that you would double down? If you were on my position, entrepreneurially speaking, would you focus on any one of these particular verticals?

[00:39:34] TG: I would focus on academia definitely. That's your best market. Mix academia with experienced engineers who are doing this for fun, and you're going to make a lot – They get to make the connections. They get to learn from each other and there'll be a good challenge back and forth. I'd definitely go that route. There's so many, I guess, groups of academia which already have that sort of hackathon experience against each other. You could bring a lot of them together.

[00:40:01] JM: How do you know that? I don't know of any academia hackathon things. How would you go to market? What would be your strategy? If you're on my shoes, how would you find those people?

[00:40:12] TG: I'll send you some links of some, I guess, universities in U.K. later on.

[00:40:16] JM: Okay.

[00:40:17] TG: So I've worked with sponsoring Ph.D. students in the past, and there's a lot set up where you have 5, 10 universities collaborating together with their Ph.D. students, and those Ph.D. students work together. But there's also kind of the standard engineering aspect of it together with the Ph.D. students and they do things like hackathons and so forth. It depends on the school. Some are doing it for their own intellectual property. I think you'll find the same with doing your finding collabs in a corporate environment. A lot of them are doing it to spark innovation ideas and to spark their own intellectual property. They may not necessarily want to share it, but school is generally a lot more open.

[00:40:58] JM: Right. Right. Good point. Yeah. Part of this – Another curiosity I have that I'm trying to evaluate in the midst of building FindCollabs is what is the gradient from project to product? In some cases, it's very deliberate. You have kind of the Y Combinator mentality where it's like you figure out a problem in your life and then you try to find a solution to that problem, or you're working at a big company and you see them build this internal solution and then you're like, "Oh! I want to bring that to the masses." There're all these kind of like classic prototypical stories. But I guess I'm just trying to figure out if there's a potential for like an on-ramp from side projects to products and what that on-ramp would look like. Does FindCollabs seem like a place that would be useful for that as well? Because there's like kind of the cofounder dating element.

That's a commonly cited problem. How do you find cofounders? That's another thing, and I'm kind of playing around with it. Have you ever had a desire to find cofounders or find other people to build products with?

[00:42:10] TG: I'm not sure, because I've been in the big company for so long, and then I'm only new to the startup game, and it's very different here. So I'm in London, England. So very far away from the west coast.

I don't know. The direction I went with this project – So the first thing I did was I tagged all the code as JBL. So that way sharing it makes complete sense and there's no real risk. Well, there's a risk, but it's more of an educational exercise. I think if it was a closed kind of aiming to be something to be sold, then there's going to be tension between the groups. So having something open and always open target makes more sense, because the value is what you learn. It's not what you create. Essentially, if I were to make an AR beat game, use this as a playground to learn stuff, throw away all the code and start from scratch and do it right. Just to be able to share that code and JBL it and not worry about – Pretty much ensure that anyone who is playing with it will just be playing with it to learn. Then they will go about and write it from scratch themselves too.

[00:43:27] JM: What caused you to go from a bigger company to a startup?

[00:43:32] TG: I really enjoy kind of the underdog challenge and kind of inside of a team being the underdog and trying to kind of succeed. So when I first started, we were doing heavily the Playstation Portable, which at that point there was no portable game console. So it was kind of to catch up Nintendo DS. Then the PS3 was a big challenge battling with Xbox 360. From there, the PS4, for that to be successful, kind of hit a pattern. Then I had this opportunity and really liked the mixture between VR and music just meeting with artists, being able to go and just – I'm working on hardware and software projects, which I really enjoy. That was really appealing, and so glad I made the move and I wish I made the move earlier.

[00:44:24] JM: What's your sense of the shifting business requirements for musicians? Because my sense is that social media and streaming have really changed the economics in a pretty dramatic way and changed the requirements for musicians. Tell me about some of your conversations with these people.

[00:44:47] TG: I guess they're all really excited in technology, and there isn't necessarily that many opportunities for them to get involved in technology. I guess on the music side of things, they don't have a group of coders working with them. So everything is kind of magic to them. To be able to give them a service and work with them and offer them a lot of technology to explore with, it's quite empowering and they're all really excited to work with us.

[00:45:17] JM: Does the VR experience, how does it compare to – So what you do, again, at MelodyVR. Basically, you put on a VR headset and then it transports you to the concert and it's not necessarily a live recording, I don't think. I think – Or do you do live recordings also or is it just like the post-concert recording, where you can kind of download it and then just beam yourself into it?

[00:45:47] TG: So when I first started, we were doing a lot of recordings. So we're going out to like many times a week to a lot of shows around Europe and North America. So we had a big back catalogue. Then at the end of last year, we focused a lot on live and we really, really got that really down path. Live is amazing in VR. Just knowing at that time that it's happening is massively different.

We saw some vide of some young girl watching this concert we did live before Christmas, and she was crying as Liam Payne came close to her, and she really felt like she was there. Live is really, really quite impactful, and we're doing a lot more of that now.

[00:46:32] JM: You said Liam Payne?

[00:46:34] TG: Yeah.

[00:46:34] JM: Nice! That economics of MelodyVR, actually it reminds me a lot of the economics of Software Engineering Daily, because, basically, you kind of do a little bit of upfront work. Well, I mean, it's a little bit or a lot depending on how you look at it. I mean, I do my research before a show. Often times, I'm looking in documentation. I'm doing a lot of technical research. Ultimately, it's not that much work, and I produce a podcast artifact. Then the podcast artifact is pretty durable and it's an evergreen piece of content. That seems somewhat similar. You go to these concerts. You set up cameras. You record the concert. You're developing a workflow in that process. You're developing a physical workflow that's not necessarily easy to develop, but it's not rocket science. I mean, you can figure it out.

You record the content, and the content is evergreen. You and I were talking about this, but there are these famous concerts in history, like the Simon and Garfunkel at Central Park Concert. I'm sure there's plenty of other ones where if you would have like set up a VR concert there, that would have – I don't know, millions of downloads, and that's like really good unit economics on a content business.

[00:47:51] TG: The key though is really something that we had to evolve overtime is capturing it right. So for a camera to do 360 recording in almost pitch black with strobing lights, with smoke, with confetti, it's quite a challenge. So with our live shows, we ended up making our own cameras, which was, yeah, a good fun part of the challenge.

[00:48:12] JM: So as I've gone from working at bigger companies to working at my own very, very small company, the Software Engineering Daily Podcast, one thing I do like about the smaller size is I get a pretty intimate understanding of the financial inputs and outputs of the business and how those relate to technology. How those relate to like web services cost. How those relate to purchases of hardware and so on. It gives me a really comforting holistic understanding.

Now, you're a senior software engineer at MelodyVR. I don't know how many people are there. To what degree do you feel you have a holistic understanding of the inputs and outputs of the company?

[00:48:56] TG: Oh, it's night and day compared to being at Playstation. Always, the big appealing thing is kind of being able to work that directly impacts the customer and not going through so many layers. So, yeah, it's great in that. I could go to some of the concerts. I get to – When we're recording Liam Payne, he walks past and I get to show him on the VR headset how it looked and talked to him. Yeah, that sort of thing. Quite a big change, and yeah, I really enjoy it.

[00:49:24] JM: Just to wrap up, I'd love to get some perspective on what you would be building if not for any particular hackathon incentives. Let's say you had a free weekend, you wanted to build something. What would that project be?

[00:49:42] TG: To be honest, there's so much going on at work. If I have free time, I'd jump on some of that. So, I'm a bit of a workaholic in that aspect. So this was actually a great opportunity and that it allowed me to kind of reboot my – I was stuck on a few areas, and after kind of focusing solely on this hackathon, going back, I solved those problems a lot faster.

[00:50:07] JM: Fascinating. Going back to the problems at work, you mean?

[00:50:09] TG: Yeah.

[00:50:11] JM: So that brings a pretty interesting point, which is the fact that by taking yourself out of the day-to-day rigors. You can actually end up solving the problems of the day-to-day rigors more efficiently.

[00:50:24] TG: Yeah, I'm totally convinced, doing this every couple of months. Even if it's in total amount of time, it's cutting into some work time. I think, overall, efficiency-wise and kind of motivation and just kind of skillset. It's a massive boost.

[00:50:39] JM: Okay. Well, Tony, it's been really great getting to know you and I appreciate you coming on the show. I appreciate you competing in the hackathon. Congratulations again on winning, and I look forward to seeing you in the next hackathon once we announce it.

[00:50:54] TG: Great. Thank you very much. Thank you very much for the podcast. Yeah, thanks for doing all these. I really appreciate it.

[00:51:00] JM: Okay. Awesome.

[END OF INTERVIEW]

[00:51:04] JM: GoCD a continuous delivery tool created by ThoughtWorks. It's open source. It's free to use, and GoCD has all the features that you need for continuous delivery. You can model your deployment pipelines without installing any plugins. You can use the value stream map to visualize your end-to-end workflow, and if you use Kubernetes, GoCD is a natural fit to add continuous delivery to your cloud native project. With GoCD on Kubernetes, youdefine your build workflow. You let GoCD provision and scale your infrastructure on-the-fly, and GoCD agents use Kubernetes to scale as needed. Check out gocd.org/sedaily and learn how you can get started.

GoCD was built with the learnings of the ThoughtWorks engineering team, and they have talked in such detail about building the product in previous episodes of Software Engineering Daily. ThoughtWorks was very early to the continuous delivery trend and they know about continues delivery as much as almost anybody in the industry.

It's great to always see continued progress on GoCD with new features like Kubernetes integrations so you know that you're investing in a continuous delivery tool that is built for the long-term. You can check it out yourself at gocc.org/sedaily.

[END]