

EnduringPlay Season 2 Episode 7_ Jo Cronk

Sun, Apr 12, 2026 7:37AM 1:09:21

SUMMARY KEYWORDS

Game development, technical design, prototyping, game engines, game education, DigiPen Institute, game development strategy, game design, game development process, game development tools, game development challenges, game development trends, game development industry, game development careers, game development future.

SPEAKERS

Enduring Play Computer, Jo Cronk, Cheryl Platz

E Enduring Play Computer 00:00

Initializing Enduring Play podcast season two engine. Decompressing audio. Synchronizing waveforms. Reticulating splines. Launching podcast lobby.

C Cheryl Platz 00:16

Welcome back to enduring play, the podcast, where we explore what it takes to create video games that don't just survive, but thrive. I'm your host, Cheryl Platz, video game designer, director and author of The Game Development Strategy Guide from Rosenfeld media and during play, season two is about scaling our game development ideas beyond the individual. A common thread through most of our conversations is the collective through communities of game developers, through game education, through games research, and through the events and systems that support the releases and people making our games. Few games are released in a vacuum. So how do we scale together? Jo Cronk is a successful video game engineer with past credits at studios like Crystal Dynamics, snow blind studios and PopCap Games. After over a decade of experience in the video game industry, they started a new chapter of their career as an instructor at DigiPen Institute of Technology, where they now teach technical design as an assistant professor, Joe applies everything they know about game development and design, not just to their course content, but to the design of their classes and the way they interact with students. In this conversation, you'll learn more about the emerging specialty craft of technical design. Explore firsthand the benefits of pre production from an engineering perspective, get deep insights into video go game engines and explore the world of modern video game education at a school ranked consistently in the Princeton Review top 10 for game design programs.

E Enduring Play Computer 01:36

Loading episode preview

J Jo Cronk 01:38

The first place where I worked as a technical designer, in practice, was at Crystal Dynamics on Tomb Raider Legend, and a lot of what I was doing day to day was working with the main physics programmer and implementing physics puzzles in the game. But the other part was generally meeting with the designers and the artists and trying to figure out what their normal sort of like workflows were, and coming up with tools to help automate some of that kind of stuff. And the next thing I knew, I had one of the company executives like in my cubicle wanting to know who I was, and, you know, thanking me for helping them save a lot of money. So that was kind of cool as a, you know, fresh faced junior engineer.

C Cheryl Platz 02:29

Let's get into the game.

E Enduring Play Computer 02:31

Player one, Cheryl Platz, pronouns, she her. Player two, Jo Cronk, pronouns, they them. Podcast, level start.

C Cheryl Platz 02:43

Welcome back to Enduring Play, the podcast where we discuss what it takes to make games that don't just survive, but thrive. And I am so excited to have Jo Cronk here with me today. Fantastic background in video gaming, who now brings that experience to the next generation of gamers, one of the most well respected gaming schools in the country, right here in the Seattle region. Jo, thank you for being here with me today. Thank you for having me. As with all of my guests, I like to give folks a chance to tell my audience about what's important to you right now. How would you introduce yourself to somebody at, say, a Game Developers Conference, if you were meeting them for the first time.

J Jo Cronk 03:25

I don't know. Usually if I'm like, meeting somebody kind of in a casual situation, I don't it always feels really super awkward to talk about myself, so I'll just downplay it and be like, Yeah, I've been a software engineer for a long time, and now I teach computer science and technical design at a video game related College, and kind of leave it at there. But yeah, I've been working in games a very, very long time. I was a software engineer for about 20 years before I came to DigiPen, and most of my background is in gameplay programming and tools engineering, and right now, what I primarily teach is technical design, which is kind of a weird thing in and of itself, because no one really has - like, a I don't know. There's not really a standardized description of exactly what technical design is, which is sometimes a bit of a problem. But I mostly try to trick design students into enjoying computer programming without them realizing what I'm doing, using tools like visual scripting

C Cheryl Platz 04:34

As somebody who runs design teams. Yes, that technical design role is such a unicorn and so misunderstood. And so thank you for doing this work and helping those early career tech designers find their feet and for being here with me today so we can maybe help the rest of the industry understand what that role is or should be, how we can set those early graduates up for success. Because it's been hard to recruit for that very specific set of skills. So excited to talk to you for so many reasons.

E Enduring Play Computer 05:08

loading podcast level one: defining the role of technical designer.

C Cheryl Platz 05:13

Let's just start there. Let's start there. Let's go with the technical design stuff, and we'll work backwards to all the engineering stuff and all the other pieces. What does technical design mean to you?

 J

Jo Cronk 05:23

Technical design, I'm kind of looking at it through the lens of just my previous work experience, for the most part, and a lot of times when I was doing technical design type work, I was expected to be kind of this hybrid programmer designer who was mostly just writing a ton of code, but then also having, like, some say in design decisions, even if I wasn't the person who was, like, predominantly responsible for the design of something. And I don't know I really enjoyed doing that type of work. Rapid Prototyping is something that I would get tasked with pretty often. And usually what we would end up doing is there would be some random design document somewhere that would be, hopefully mostly complete, or at least complete enough to create a prototype from. And then I would just create this really fast prototype with, you know, not the most optimized code or anything, and then we would just try to get it done quickly so we could put it in front of people and figure out if we're even going to keep that feature or what. And I learned, like, really early on as a younger person, not to spend a ton of time trying to make the code nice, because half the time we would just turn around be like, You know what? No, we're deciding to go in a different direction. Just throw all that away, and the number of times I had spent, like, so many hours ever gonna live to see the light of day? Yeah, it made me a lot faster to realize that. But the other thing that I feel like technical designers have to do, and something that I ended up doing at most of the companies that I worked at was kind of becoming like a an interpreter between the engineering department and the creative departments. You know, whether that was design or art. Not to disparage my previous co workers, because I love all of them, but you know, sometimes the engineers just really don't want to talk to artists or designers and vice versa. And so having someone who could speak in both of their sort of languages to them and translate for each other was super valuable.

 E

Enduring Play Computer 07:38

Tutorial level: prototyping as a game development tool.

C

Cheryl Platz 07:44

What role does prototyping play in the game development process? In his book, "A Playful Production Process," Game designer Richard LeMarchand defines four phases of a typical game development process: ideation, pre production, production and post production. If you want to go more specific, there's some sub phases in my book, The Game Development Strategy Guide. In the ideation phase, you're looking to find ideas worth exploring, and the evidence that proves they're worth exploring. In my experience there's infinite ideas, so you need something pretty compelling, and that's where prototypes often come in. Richard goes on to define a concept of player activity as how a player uses a particular verb, like running, jumping or collecting. Prototyping is typically an exploration of one or more player activities sometimes collected into repetitive game loops. The secret sauce of the prototype is sometimes determining whether this activity is even achievable technologically or otherwise, but sometimes it's more the nuance, timing, game feel emotional resonance, mood, esthetics, scalability, etc. But importantly, both Richard and Chaim Gingold, a former co worker of mine from Maxis, who wrote the essay "Catastrophic Prototyping And Other Stories", agree that most prototypes should be pointed at one or more specific questions to answer. Once those questions are answered, prototypes can be set aside and work can begin on the next prototype, or perhaps planning for that production game. It is this laser focused mindset that allows us to work quickly and avoid being too precious about discarding the ideas that no longer serve us or our eventual game.

E

Enduring Play Computer 09:13

Podcast level two: Technical design on Tomb Raider.

C

Cheryl Platz 09:18

What was the day to day like as a technical designer, what sort of tasks were you doing, or what was the project you were really proud of?

J

Jo Cronk 09:26

Well, the first place where I worked as a technical designer, in practice, even though my job title was something else, was at Crystal Dynamics on Tomb Raider Legend, and a lot of what I was doing day to day was working with the main physics programmer and implementing physics puzzles in the game. But the other part was generally meeting with the designers and the artists and trying to figure out what their normal sort of like workflows were, and if they were doing any. Anything where they, let's say they're editing spreadsheets or doing other things manually, by hand, repeatedly, and coming up with tools to help automate some of that kind of stuff. I think my first week there to help me get onboarded on our level editor, which actually at the time, was 3d Studio Max. They had me just sit and watch the artists and come up with some 3d Studio Max macros for them to help, you know, streamline some things. And they were doing a bunch of rigging, skeleton targeting by hand over and over. And so I just made a few really fast things to help them out with that. And the next thing I knew, I had one of the company executives, like in my cubicle wanting to know who I was, and, you know, thanking me for helping them save a lot of money. So that was kind of cool as a, you know, fresh faced junior engineer, amazing.

C

Cheryl Platz 10:57

It speaks to your versatility, because they're basically asking you to do a form of user experience research too, like you come in as an engineer, also just watch human beings and have insights about them, and then go do technical things. But so well done. That's very rock star all of it, but it's a great story.

E

Enduring Play Computer 11:15

Loading narrative cutscene: The Tomb Raider franchise.

C

Cheryl Platz 11:19

Tomb Raider is one of the most enduring video game franchises of all time. It started with a bold idea from early career designer Toby Gard make a 3d action adventure game with a female protagonist on a Sega Saturn. Toby successfully communicated this vision to a team of other talented creators and managed to create a game that defied the standards of the day with 3d Lara Croft navigating freeform environments that original 1996 release leans into the constraints of the technology at the time, outdoor spaces weren't well supported by the original graphic section, but those tombs certainly were. Several sequels followed over the course of an intense five years, each selling millions of copies, until the shocking twist during Tomb Raider, the last Revelation, the first attempt at rebooting the series from original publisher core design on the PlayStation two called Tomb Raider Angel of Darkness was deemed unsuccessful from a franchise standpoint, there were high standards. Parent Company Eidos Interactive sent the entire product to Crystal Dynamics in 2004 where today's guest Jo had the opportunity to contribute to the series. Tomb Raider Legend sought to retain puzzle solving and Lara's agency as an intelligent explorer, while ensuring that the world felt like a platformer with light combat and elements of self expression through acrobatic flair, while Crystal Dynamics remains the primary developer of Tomb Raider games, there are occasionally other developers and a variety of publishers involved in their products. Over 100 million Tomb Raider games have been sold as of 2024 Tomb Raider Legacy of Atlantis is a remake of the original slated for upcoming release alongside Tomb Raider catalyst in the years to come.

E

Enduring Play Computer 12:47

Loading podcast level three, when does prototyping stop?

C

Cheryl Platz 12:53

You also talked about prototyping and the role prototyping played. And one thing that struck me when you talked about that was the cognitive dissonance we have because we talk about prototyping and like, don't, don't be too precious. Don't spend too much time in the code. But then also, like on other podcasts, I've been like, "Don't hard code your strings. Build tools." And how do we know? Like, how do we help our early game desk know when they're crossing the precipice? When do you know that it's no longer time to bring the prototypers mindset at its time. Is it? Is it like a production gate? Is it a vibe? Because, because it's, it was very clear that you knew when you were in a prototyper mindset. But how do we know?

J Jo Cronk 13:32

Yeah, it's, that's a really tricky question, because I think ultimately it depends on the size of the team and what the individual roles are, because at a lot of the like, at least midsize to larger companies that I worked at, roles were extremely well defined. And so I just knew for a fact that I was never going to be writing like engine code, and that someone else, somewhere would eventually just take my slop and rewrite it properly in C++ so I was able to kind of focus on the low drag, you know, fast iteration. But if you're at a smaller company, especially something like a, you know, an indie studio or startup, I could see something like this happening where you have the best intentions and you're just, you know, you're going to do this, like, really fast prototyping, and then at some point you're going to take, like, a tech debt week and just rewrite this stuff. And I have never really seen that happen in practice, because there's always, like, you know, some milestone to hit or some demo that needs to happen for a you know publisher or potential. A, you know, funding person. I can't think of what that word is right now.

C Cheryl Platz 15:07

Investor?

J Jo Cronk 15:07

Investor, yeah, so if you're doing a demo for a potential investor, I think I kind of like took a side step into academia to get away from, like, VCs and stuff. Oh my gosh, all the dog and pony shows. So I, you know, I don't really know what the best solution is for that kind of thing, except that you just kind of have to be constantly vigilant and make sure that where your tech debt is, so that when you do have time to come back to things, you can find it and deal with it.

E Enduring Play Computer 15:44

Loading, podcast level four, the struggle at small studios.

C

Cheryl Platz 15:50

Think there's something really important in what you just sort of insight you just talked about there, because you said at the mid size to large companies, my role was very clear. It's two things about that. One, I've worked at lots of mid sized, large companies where worlds were not clear, but so we're just gonna put that in a box, but the gift of clarity, because you were given a clear role, clear responsibilities, you were able to just run with it. And that speaks to the often dismiss, like soft skills, like Job architectures, all that stuff for like, Oh, we don't have time for that. But because they did that for you at those companies, you were able to be the best technical person you could be. And I think that's really cool. But then, like, the implication for indies is like, "Well, yeah, maybe you do have to be a little bit more careful about your stuff, because you are all that there is."

J

Jo Cronk 16:37

Exactly, yeah, it's like, I get kind of scared sometimes when it's like everyone on the team is wearing, like, five different hats. You know, I have a love hate relationship with that, because I have ADHD, and I kind of need a certain amount of novelty in my like, general life, and the idea of working at a small studio where you do have just this, you know, wide range of responsibilities, appeals, but at the same time, you know, there's a cost there where you will not always have the time that you want or need to deal with things that probably need to be dealt with, but, you know, get sacrificed in the name of shipping something on time. And you know, not to say that larger studios don't have some of those kinds of problems too. But it just feels like if you have a certain amount of structure and organization, especially, you know, if you can afford things like, oh, pre production. Then, you know, you can kind of go into a project with a more realistic schedule and maybe a better idea of, you know, how everything needs to be planned out and worked out.

E

Enduring Play Computer 17:59

Loading podcast level five, the value of slowing down to speed up.

C

Cheryl Platz 18:06

Let's talk about that a little bit. Because you are an experienced engineer, you've been through this, and you're also teaching. One thing I've experienced as a creative leader is when I try to help my teams slow down to speed up. Like when you put in creative ideation processes, there's a lot of pushback, especially at first, right? Like, "wow, there's so much process. Why are we doing it? Why can't I just make the things?" And I am blessed to be at the point in my career where I have enough confidence, like, "you're just gonna have to trust me on this one." But I'm curious, like, is- Has that been your experience that, like the slowing down to speed up, like the pre production pays off in the end? That you get benefit from that alignment?

J

Jo Cronk 18:43

I mean, even from my own personal projects, I see the benefits if I just take, you know, some time to hand sketch user interfaces and hand draw class architecture. It definitely saves me time when I sit down and I'm doing the actual implementation, because I don't have to really think about it in the same way. I kind of like have created my own task list, and I can just run through it. So one of the things that I do in my prototyping classes at DigiPen is I do make them do documentation upfront, and they hate it. A lot of people don't really see the value in it at first, until, you know, they go through it, and they go through the entire process. And then, you know, I'll talk to them, usually, like, as freshmen, they they don't necessarily like, see it right away, but usually I'll have conversations with the students in their junior year or senior year, and they're like, oh, you know, some of those things that I learned way back when, wow, that was really helpful for me later on, when I was working on these much more complicated projects than the things we were doing in those introductory classes. And that always, I don't know, I just hope they can, you know, take that into the industry, and if they work somewhere that doesn't value pre production, you know, maybe they can be champions for that cause, out in out in the world somewhere.

C

Cheryl Platz 20:07

So many things I want to unpack there. First of all, thank you for doing such important work, because I'm a huge fan of knowledge management. But one of the most impactful things, but the people in my education did for me was had us lean into documenting our process when I was studying user experience design at Carnegie Mellon. And then I had a portfolio, you know, and then I had documentation of everything I had learned. So, like, even, even if they don't like doing it, even if people don't care, like, that's that. It's like cementing the learning right, and it gives you, it sets you apart as an interviewer, so you're giving people that gift even if they don't want to acknowledge it. But let's put that aside. Yes, documentation and game development. No. People want to say like, "oh, agile, we don't need it." No, no, no, no, no, no, we're putting a movie and a cloud service together, and we have to be aligned so that those things are the same. I'm so glad that you are helping them with that. And documentation is power. It amplifies your impact and your ideas, and so you're giving your students this ability to, like put the idea out once, and have 80 people go and read it and do the same thing. And that's such a gift.

E

Enduring Play Computer 21:14

Podcast level, paused, begin host commentary.

C

Cheryl Platz 21:18

I'm not just passionate about documentation because I like words, documentation creates the kind of shared understanding that drives a group of diverse creatives to create groundbreaking work like Tomb Raider or Flow. Tracy Fullerton, from the University of Southern California's Game Innovation Lab, has coined a number of influential terms and frameworks used by game designers around the world to drive that shared understanding through their teams. At the core of that work is experience goals, a concept which resonates deeply with my training and user experience. Experience goals are fundamentally different from project goals, which are also important, but might be more like "drive 100,000 players into the game" or something more concrete. An experience goal talks about player impact on a human level. Maybe your goal is players have an experience so emotionally moving they have to talk to a friend about it, and that brings those friends into the game. Project goals are usually about the what or why for the business and the player experience. Goals are the desired outcomes for the player. By keeping those goals free of implementation details, you leave yourself free to adapt if constraints come up during implementation, and they always do. It's much easier to align on a desired feeling or emotion than one specific implementation. Start with experience goals, align on them and document them. Slow down to speed up.

E

Enduring Play Computer 22:29

Podcast level six using templates to jumpstart creativity.

C

Cheryl Platz 22:36

Do you have any go to references or like tricks, or is it just, kind of, like, generally, just whatever you want, just write it down, or, like, there anything that works to kind of break through to students who are trying to document their own work.

J

Jo Cronk 22:48

I mean, usually I will give them templates that they can start with that have, you know, especially in those introductory prototyping classes, you know, they're doing a lot of design work from scratch. So it will. The templates will include things like, you know, core game loops and feedback planning tables and, you know, things that you might not necessarily have to deal with if you're doing, like, the design of a smaller system, you know, within a company or something. Yeah, the templates are pretty generic. And, you know, a lot of it, especially the freshman year, semester one, semester two. I'm just trying to get them started thinking about those sorts of things. And then they do go into their individualized design classes and do a lot more work that's a little bit more complicated. So one of the things, just to give a little context about DigiPen and how the Bachelor of Arts in game design program works, there are multiple different tracks that students can choose to specialize in, and the students have to pick at least two different specializations so they don't end up like solely specializing in, say, narrative design, where there are like two jobs,

E

Enduring Play Computer 24:07

loading narrative cutscene: educational video game programs.

C

Cheryl Platz 24:13

The DigiPen Institute of Technology is one of many specialty gaming schools that have sprung up, particularly around the US. Founded in 1988 DigiPen was the first school in the world to offer a bachelor's degree in video game development, and its Redmond Washington, proximity to Seattle area studios like Microsoft and Valve, mentioned in episode five of this season is no accident, as Jo points out, the game design major at DigiPen includes six concentrations, level narrative, systems, technical, user experience, and user research. All of these specialties are mentioned in some way in my book, *The Game Development Strategy Guide*. In 2026 DigiPen was ranked the number six undergraduate Game Design program by the Princeton Review. Joining them in the top six were New York University, the University of Southern California, the University of Central Florida. And the Rochester Institute of Technology in the US and Abertay university in Scotland. The 2026, rankings for graduate and undergraduate programs will be linked in the show notes, as well as a USC podcast I did recently. If you want my advice as an educator and former grad student, you'll want to look for programs that have strong interdisciplinary elements, dedicated project work, strong case studies emerging from that classwork, the strong fundamentals in non gaming fields that give you something to fall back on in times of layoff between games and of course, check out the faculty. Make sure you've got folks with real industry experience like Jo. A mix of academic and industry perspectives is fine, but you will need that industry perspective.

J Jo Cronk 25:40

Usually, you know, the messaging that we give the students is like, hey, you know, if you're trying to maximize your ability to land interviews and jobs after graduation, you might want to consider a combination of technical design with user experience design, because there are a lot more jobs outside of games in those fields than something like, say, level design, so they'll go into, say, I don't know, systems design too, or something like that, where they'll be doing documentation and pre production that's specific to that type of project

C Cheryl Platz 26:19

That was such helpful context, and I'm certain that a couple of listeners heard that, are intrigued and probably going to want to learn more. But also, it's very pragmatic, right? Some of those paths are very games oriented, and some are a little bit broader and but I love the sort of interdisciplinary perspective, because games are so interdisciplinary, that's one of the things about writing the book was, you know, needed a chapter where I just listed as many disciplines as I could think of a game development, because I felt like so many people on my teams didn't feel like they understood their peers and felt judged because of that, or didn't know how to start conversations. And so it's great that like at least there's this exposure to these other disciplines.

E Enduring Play Computer 27:00

Loading podcast level eight, navigating interdisciplinary teams.

C Cheryl Platz 27:07

What's some advice you have for game devs who are navigating these interdisciplinary environments? Since game development is so interdisciplinary,

J

Jo Cronk 27:18

you know, just thinking about the students that I've had over the last handful of years is that there are always people who like come into the school and kind of expect to be able to just work individually in a little siloed bubble somewhere, and never talk to anybody else and never get feedback from anybody else. And you know, one of the things we all work on as faculty there is to kind of wean them out of that mindset, especially the design students, because just being able to work collaboratively on a team is such an important part of design. And I don't know, I'll make some flip it comments every once in a while too, like, hey, you know, if you just want to sit in a cubicle somewhere and not talk to anybody. Maybe you should go into one of the CS degrees. But even then, you know, I mean, like the engineers are gonna have to talk to other people too, other engineers, you know, and depending on how much cross disciplinary communication any specific studio needs, the school itself also has a big group game project class called Game class that, honestly, I think, is the main reason why a lot of students go to Japan. And usually what it is is that, especially junior year and senior year, the students are on these multi disciplinary teams, so you'll have engineers and designers and artists and usually sound designers, all working together on some kind of larger, multiple semester long project that a lot of the students now are just publishing to steam right away, you know, and already kind of like making a name for themselves, and that Class is where a lot of the students actually learn how to deal with people in other disciplines and how to talk to different people with different skills. We do have some game project classes that are a little bit earlier, that are just for the designers to work with other designers to get some practice talking to other humans and thinking about other humans before they get thrown into game class with all the other degree programs. And usually in those classes, if I remember the structure correctly, there will be like teams of three designers, where, technically, one of them is the quote, unquote programmer, who is usually, you know, the most technical of the designers on the team. And then there will be the other two people can kind of choose what role they want to play on that specific team. Usually there will be, like, a systems person, a level person, or maybe level and UI and systems, or what there's it's kind of up to them how they want to self organize for that kind of stuff. So they do get a lot of practice. With that.

C

Cheryl Platz 30:02

I like the contrast, too, between the working with within design and then working across disciplines, because there is that dichotomy. On the teams I work on, there's the within craft or across similar crafts. You're kind of doing critique, you're tearing apart ideas, you're testing things. Then when you go across disciplines, that's a different skill set you're trying to build shared understanding. Often it's less about like tearing things apart, or you're asking that person to bring a different perspective. So it's cool that the system brings both of those types of collaborations because not all collaboration is the same.

J Jo Cronk 30:36

And the types of projects that the students can work on at the school are so varied. So every student's gonna have, like, a different experience with all that kind of stuff. Because some of the game teams will be working on projects in commercial engines like Unity or Unreal, and some teams will actually be a little bit more programmer heavy and be working on entirely custom student made engines which are much more challenging to work on, because usually there isn't really, like, a usable editor for any of those until they're supposed to be, like most of the way done with the game. Yeah, a lot of the teams do end up kind of defaulting to the commercial engines just to have an editor that they can use right away and start prototyping for play testing right away,

E Enduring Play Computer 31:25

Loading narrative cutscene: Video game engines.

E Enduring Play Computer 31:30

A video game engine is a framework that usually consists of both a developer focused set of tools and a player focused set of libraries and runtime engines that let game developers build games quickly without starting from scratch. Many of the problems game studios used to solve are common. You need to be able to represent physical space as a map. You need a heads up display. You need menus a player character. You need a system to interpret player input via keyboard, mouse, touch or controller. You need a physics system and an asset management system for the art and animation of the game. And of course, you're going to need to program the logic of the game in there. Somehow Unity and Unreal are the big two commercial engines, and both have free tiers available, exactly for the use case Joe describes, because it is advantageous to the business to let students in small studios get familiar with their products at scale before making the huge investment to launch with their engine. Monetized games using a commercial engine usually have to pay per seat license fees as well as royalties on revenue. There are also open source engines, and the most popular option right now is Godot. Open source comes with its own benefits and challenges. It's a compelling choice for indie studios who are worried about accidentally scaling past the monetization thresholds on the big commercial engines, but there are always extra Legal and Security complexities when shipping on open source software, and no matter what, using pre existing game engines means you are sacrificing future flexibility for speed. Now, some students and devs prefer to stay close to the hardware and fully explore that flexibility for the greatest gains building their own engines. But as we heard in season one, episode seven, for most studios, using an off the shelf game engine is the right choice.

E Enduring Play Computer 33:01

Loading podcast level nine: the why of engine programming.

C

Cheryl Platz 33:07

So tell me about what might drive someone to make the choice to do engine programming. And I know when we had kind of talked about getting set up this, when I asked there was something you would do differently, engine programming came up. And so I think this is an interesting segue, because there's still students choosing to do it, even though there's a lot of stuff that's available off the shelf. Which, obviously someone has to develop those off the shelf engines. So I'm curious, like, your take on like, what role that lower level programming plays now, and like, what skill sets are well matched to it, and how it's showing up for you and your students.

J

Jo Cronk 33:41

Yeah, so all of the custom engine stuff happens in the different CS related degree programs, so I don't actually get to see a lot of it personally, but you know, just thinking about where development was 20, 30 years ago, and kind of where things are now. I just remember, you know, being so excited when commercial engines became widely available and free. One of the first game projects that I ever worked on. We were actually licensing the Unreal Tournament 2003 engine. And if I remember correctly, I think our publisher had to pay, like, I don't know, 100 grand or 200 grand or something, to license that thing. And, you know, then we were expected to make enough of a profit to, you know, make it worth that expense. Oh, triple A, okay, you know, the idea of writing your own engine to me back then, as a as a junior engineer, just seemed like this insurmountable kind of project. But there's so much more information out there now, and with these commercial engines, you know, different conversations about different types of object architecture, and there's just so much information available. Online about different programming languages, and if you are going to be doing a lot of stuff in C++.

To me personally, like the draw for creating, like a custom engine would be, I don't know, maybe it's because I'm type A but just having complete control over the structure of the engine, and like your update loops and the memory usage, and being able to really, really optimize the engine for the exact game and exact game objects that you are creating really appeals to me. Now I don't necessarily know that a lot of that is necessary. You can make great games with these commercial engines that don't require this, this type of work.

E

Enduring Play Computer 35:51

Podcast level 10: Development deep dive, update loops,

C

Cheryl Platz 35:57

You know, I wanted to go back and maybe unpack a little bit of what you were talking about, because you gave some really cool really cool, specific examples of some of the benefits of potentially customizing your own engine. And I think one of the things I like about doing this podcast is helping people understand across disciplines, you mentioned update loops. And let's, let's, like, double click on that a little bit. How does an update loop... Having control over an update loop maybe differentiate when you have a custom engine over, over the shelf engine, like, what might you not be able to do in the off the shelf engine that you think you might be able to do in a custom engine?

J

Jo Cronk 36:28

I mean the timing of things and like, where things happen. Like, if, if you, you go online and you do a search for something like Unity physics problems, and you see all the people raging about just how, like their physics engines work. You know, if you don't like how a specific engine is handling something, you have the ability to do it yourself in a different way. Is law work. But the thing that I think is, I just, I don't know, I kind of feel like everybody should make at least, like, just a really crappy little 2d engine, because when you get everything set up, and you are dealing with things like rendering sprites on screen, and when that happens, and you know how you're going to move them around and how you're going to resolve collisions, All of that teaches you about the inner workings of engines in a way that you won't necessarily get by just using something like Unity and Unreal to like create a game with and whether or not that information is necessary for anything I don't know. I just feel like it makes you understand engines in general, in a much deeper way.

E

Enduring Play Computer 37:45

Podcast level 11, the lost art of optimization.

C

Cheryl Platz 37:50

That's very fair. I mean, it's been very interesting to me, the sort of sprawl of games over the years, like when our career started, I was shipping games like 1632 megabytes, and now they're gigabytes, right? And some of that, I feel like, is, if you don't understand how to do things gracefully or efficiently, you could just do more of a thing, doing less of a thing, or doing more a thing more efficiently, or doing doing it for more devices, requires that more innate knowledge of the technology or the approaches.

J Jo Cronk 38:19

Yeah. And I think that's one of the things too, is that now that we have basically unlimited RAM and giant hard drives, there isn't really this pressure to optimize things in the way that there was when everyone was trying to ship to like, 10 different console platforms. And some of those consoles, you had to know this, like, weird, tribal knowledge, dumb, like, how to actually, like, get certain types of performance out of, like, certain platforms, and I don't know a lot of that is kind of a lost art at this point. Since we've moved to mostly, like, PC gaming, and I think, you know, a lot of times it does kind of encourage a certain amount of laziness when it comes to things like asset sizes and load times and things like that, because we don't really have to worry about it as much unless you're going to be doing something like mobile development, or, you know, some serious cross platform development.

E Enduring Play Computer 39:27

Podcast level paused, begin host commentary

C Cheryl Platz 39:32

When you're just starting out. Scale seems like a distant problem someone else's problem, but it's so much easier to make smart, scalable decisions at the framework level, once your game is full of giant textures you don't need and inefficient animations, it can be near impossible to extract them without destabilizing or destroying your game. The result of a bloated game is a success ceiling. You will not be accessible to certain consoles, low spec mobile platforms or PCs with older hardware or limited. Disk space and RAM and for games that live and die on metrics like daily active users or concurrent users, what seems like an obscure engineering decision can have massive consequences on the health of your game design and player community down the line, when you're facing cutting off large numbers of players due to device incompatibility, if you can remember to slow down, to speed up, take time in pre production to get Opto pessimistic and plan for the worst consequences of your success your future self will thank you.

E Enduring Play Computer 40:28

Podcast level 12, the hardware replacement cycle.

C

Cheryl Platz 40:33

Although I think it is interesting when we look at the issues with RAM and hard drive availability right now, because there is a little red flag way before me where I'm like, we've already seen console replacement cycles pushed out a little bit, and now I find myself like, I have like, award screeners games, and I'm like, I don't have space for you. You are 80 gigs, and also you are 80 gigs, and you are 80 gigs. There's not space. What am I supposed to do with you? And I wonder that's going to play out at scale, where people are just going to run out of space, and it's going to be, you know, heard the term friend slap thrown around derisively, but I think it's it, first of all, that's people often discount, like, connection with friends and things like that. But I think it's also just like people appreciate things that are lightweight and meet them where they are and fit their life, which right now is what's left of the hard drive they can afford.

J

Jo Cronk 41:23

Yeah, I haven't put together a new machine in a while, so I am blissfully ignorant of the current prices for everything. In the past, I used to just build a new desktop machine. Anytime Bethesda would release a new game, my only goal was to be able to run it at, like, super high graphics, and just be able to, like, brag to all of my friends, for the first you know, however, ready come it has been a very long time since I've done anything like that, and these days, honestly, since I am going back and forth and traveling quite a bit, I do a lot of my development on a very small laptop, even, like a lot of my game playing, I don't play anything that requires serious hardware. Favorite game right now that I haven't even touched like over a year, is Final Fantasy 14. I just don't have the time for MMO RPGs with everything going on. But I love that game, and it just doesn't take that much to run. So that's part of the part of the appeal.

C

Cheryl Platz 42:27

I think there's something to that not taking that much to run. And for the games that decide that they want really big mainstream success, that is an art that they have to figure out is, how do you keep targeting the devices that have scraps of hard drive left all of that stuff. And it sounds like they're doing it. I remember being at Riot and still supporting, like, Windows eight or something, and I liked that we were still doing that for as long as we did. There is an art to that. And that was, of course, Hex tech, that was their own cast of engines, so they built in that era. So probably easier to do than if you just pull something off the shelf.

E

Enduring Play Computer 43:00

Loading podcast level 13: Biz Dev dog and pony show.

C

Cheryl Platz 43:06

You mentioned the dog and pony show, which means that you've, you've been exposed to the indie, entrepreneurial game lifestyle, and I'm not going to ask you to, like, air your dirty laundry about any of that, but I did want to ask if you have any advice for game devs navigating the cross company relationships, because there's, you know, there's developers, there's publishers, there's platforms, there's investors. Is there anything you wish you had known, or you feel like you wish your students would know as they go out into that complex space?

J

Jo Cronk 43:38

I'm toying with the idea of starting up my own indie studio at some point, so I'll have to actually do some dedicated research into how to run a business. You know, as a software engineer, I just really don't have any experience with that, necessarily. I was a very early employee at a mostly a hardware startup that I thought was going to go all the places, and one of our major, major problems, and part of the reason why we, you know, never really hit the jackpot, was the fact that there were basically three engineers trying to run this company with absolutely zero sort of business knowledge, and we would meet with these sharks who would just eat us alive. And there was a period of time where I just remember constantly we had a handful of prototypes that we were taking around to all the venture capital folks and doing our little show, trying to demonstrate that we were worth funding. And those meetings were always just humiliating and dehumanizing, and it was a game to see, like, how much we would give them and how little they could get away with giving us. And I don't know i. Kind of burnt out on that work, and we did eventually hire a CEO to run sort of the business side of things, but it happened much, much, much, too late to really make a big difference for us. So I think that there are different types of skills and the top producers, well, okay, not even producers, but the the top people at your company with the highest productivity in certain areas are not necessarily going to be good managers or good business people or anything like that, and you kind of need people in those roles with those types of skills very specifically in order to have the best outcomes? Yeah, I would say, like, I don't know. I know a bunch of younger people who are trying to start their own indie game studios right now, and my recommendation for them would just be, like, okay, whoever is going to be the figurehead, or, you know, your chief whatever you need to do some research into how to run a business and how to do biz dev types of things, and potentially just have someone who is dedicated to that type of work, very specifically, because when we were doing a lot of our dog and pony shows, I just remember, there was a span of six months where the three of us, we did zero engineering work like zero. So we were getting even farther behind in our own scheduled plans, because we were just constantly, constantly, constantly doing these traveling and doing demos. It was such a waste of time.

C

Cheryl Platz 46:39

Well, thank you for your candor and for sharing that. There's a lot to unpack about these sort of specialty role of business development. One thing I've seen that I don't know if it's pandemic or I don't know if it's like generational, but it seems like we maybe aren't preparing this generation with enough skills to understand like when we need to shift how we communicate external versus internal communication, or things like that. I often think about biz dev and partner communications a lot like Dungeons and Dragons, right? Like having to maintain multiple different mindsets, like, there's the GM knows some things, and I know some things, and my character knows some things. Everybody has different like maintaining all those different, state, states, and trying to figure out how you get all those groups to a shared understanding, while maybe not everybody has all the data, where I've seen a lot of teams fall down, is where they just assume everybody is working from the same goals and the same rules, and just blurting things out that like accidentally randomize a company that a vendor company that interprets a thing as like a business deal or getting fired, or a piece of advice that was recently given to be like, hey, that needs to be a negotiation. Most things in biz dev need to be negotiations. Like, okay, okay, yeah, thank you. Thank you for that. That's a good reminder. Good reminder.

J

Jo Cronk 47:52

That's right, yeah, I prefer to have somebody in those positions who's good at that kind of stuff and can just, you know, the cognitive load for them to do that type of work is much lower, and outcomes are better. And yeah.

C

Cheryl Platz 48:06

We are better together than we are apart, right? That's the beauty of having a studio and not a one person shop that has to do all the things and more power to you if you think you could do all those things. But when you have a well balanced studio, and you have people playing clearly defined roles - to our point earlier, make sure you know you've clearly defined that engineering doesn't have to go on the dog and pony show, but biz dev folks do, and clearly defines, like, how each role supports each other - but then then you can go more places, farther, faster,

E

Enduring Play Computer 48:37

loading podcast level 14: disruptive behavior in the classroom.



Cheryl Platz 48:43

You have also done some work internationally, and I'm wondering how that work changed your perspective, because I'm always looking to expand my horizons and as a teacher myself, some of the best lessons I've learned from teaching are the perspectives my international students bring in. How does your international educational experience impact your work?

J

Jo Cronk 49:05

It's interesting to see the differences in the different areas. We do actually have a campus in Singapore, and one of the requirements of graduation for the Singapore students is that they have to come to Redmond for at least like one summer semester and take classes with us on campus and seeing the differences between the Singapore students and the US students and how they approach the assignments and approach the course material, and kind of like getting a sense of their mental models when it comes to learning has been really interesting to me. You know, as someone who's trying to become a better educator, one of the things that I have done recently is I have flown to China a few times to teach for a school that the Japan. Was trying to kind of forge a relationship with and a handful of the faculty went out there and taught some workshops to middle school age students who are preparing for competitive high schools. And it is just mind blowing to me, like how competitive everything in China is because of the size of the population and how much competition there is for things like positions in a school or entry level jobs and things like that. Even teaching in China is kind of a wacky thing. I would be unable to teach at a university there without a PhD. It's very, very different. I am just starting to get used to teaching college students. So going into kind of basically like a K through 12 environment was a little rough for me. The very first time I went, I had a very small group of students, and they were a little bit older, I would say they were maybe like kind of junior high age, and very tedious, hard working, did everything that I asked them to do, and the workshop was great, and I feel like we had some good presentations at the end of it. And the second time that I went out there, because the first time had been a success, they shortened the duration of the workshop to one week instead of two, which trying to teach Unreal to a bunch of middle schoolers in a week was just almost impossible, but the group was also almost three times the size. So then I started having social issues in the classroom that I hadn't anticipated, because I didn't see that the first time around, and I've never really taught people that age before, but the second group was a little bit younger, and yeah, there were almost some fist fights in the classroom, and I don't know how to deal with that. So thankfully, I had a local ta who was like a younger person who is, is going to school for education, and he had great advice for me on, like, how to modify the workshop to work with this group of students. I was super lucky to have him there and his knowledge. Yeah, it's been really, it's been a big learning experience for me and thinking about, you know, ways to make the content more approachable for kids that age, there's still a lot of value in those sorts of things to make the content more approachable for people of all ages. In my opinion, I feel like every time I run a course or a workshop with a group of people, I am learning about how to be a better teacher and craft lessons that help better with things like recall and retention. Like I said earlier, I have a scorched case of ADHD, and I need a lot of novelty in my work life. And one of the things that really interests me about teaching is the fact that there are so many different things to do, like curriculum design, then the lecturing part. And I would say probably my least favorite thing, is grading. That has driven some research and development on my part to figure out how to possibly incorporate some machine learning in to my own like tools development, to automate as much grading as I possibly can. So I feel like I'm just also constantly learning new things and new technical things and improving my skills as a like a tools engineer, as a programmer in general. You know, being at a school like it depends. So it's just, it's so much fun.

C

Cheryl Platz 53:49

Thank you for sharing that. It's interesting the story about the younger students, because it brings to mind some of the challenges we face with player safety and live-service Games, because you see all this disruptive behavior, and a lot of times it's kids in that age range. Some of them are, some of them kind of got past the filter, right? Like they're they're not actually 13, and it's that age group with that energy, and they're bringing it into a group, in a room of people who think they're dealing with adult and then everything just starts to devolve, right?

E

Enduring Play Computer 54:17

Loading, podcast level 15: inspiring students with game design techniques in the classroom.

C

Cheryl Platz 54:25

I'm curious, because you mentioned kind of you pick some things up all the time and you develop some techniques. One of the motivators of play that's evolved for me is education. And so I think people might be able to benefit and learn how to be better game designers from some of the things you've learned when you talk about better recall and retention, are there things that you've learned that you think might help people who are looking to build games for folks who are looking to play games that also make them better people?

J

Jo Cronk 54:53

One of the things I've been kind of looking into is just the different learning styles. And one thing that's. Seems to be pretty much universal across all of the design students I've ever had, regardless of age, is that a lot of people just tend to learn best with hands on practical application. So I love theory, and I love thinking about theory, but a lot of times, if I spend too much time lecturing on theory. I lose the students, especially the younger ones, and so one of the things I've been trying to do in the last year or so is that I actually don't lecture in class anymore. I pre record tons of videos, like I think last semester, I made 80 different tutorial videos just for this one class that basically cover all the possible topics that I would want to share with them. And the students can kind of pick and choose which ones are the most relevant to them, and the types of work that they're doing, or the types of prototyping that they're trying to do, and then all of the class times are basically hands on labs where they're supposed to be making progress on whatever our current project is, and what everyone is working on the class is kind of self paced based on their interests and their previous experience in a setting where They can get really easy help from me and from my small army of TAs that I usually hire to help kind of coordinate a lot of the in class help. And that what I'm seeing happen from that is that the students themselves are, especially the current cohort that we have, they are a lot more open to looking things up and doing a certain amount of like self study and self research and self directed activities in a way that I think is really, really going to help them. When they go into, like a professional work environment and they're expected to have a certain amount of self motivation and the ability to like work independently without constant supervision in order to get things done, but are also still interacting with the other students in the room and getting help from each other and helping each other and learning how to be on a team and talk to other people and Be respectful to other people in like, a group environment where you might not necessarily agree with the design decisions of everyone else around you, but you know, learning how to be able to have those conversations in a very respectful way. So all of those things kind of come together in these classes in this really, like organic way that I try to curate to a certain extent, but it's always different. Every group of students is completely different every semester, so I kind of have to tune things on the fly to customize the experience for that specific group of students. And it feels like I don't know. I'm basically running like a live services game. It's amazing, though. I laugh with the students because I tell them that they're my current batch of play testers, and that, you know, we're designing the design of the design program in this iterative way that we try to get them to do with their games. And so I think they kind of enjoy that.

C

Cheryl Platz 58:17

What a great way to model growth mindset. And I love the way you're thinking about it, helping people be self directed, because that is something I've seen at several places where people sort of are stalling out more like they like. If something isn't clearly delineated, folks just like stop. And so I love that you're seeing when you shift your mindset, folks are able to work past not having clearly defined steps for something that is a huge gift. Because if you want to go work in this crazy creative industry, there will not be an answer key, especially if you want to push to senior, you want to push the principal, you want to push the creative director. There is no answer key. No one's going to tell you what the right path is. Like these skills you're giving them to be able to like, I'm gonna go research. I'm gonna go find some best practices. I'm gonna go find six games that failed. I'm gonna go do all these things. That's a gift. That's a big gift. Thank you. I love that they're lucky to be play testing, continually iterating the program. And that's the other thing, right? Like, if it's continually iterating, you know you're getting the best version of whatever the program is. It's the current best. It's great.

E

Enduring Play Computer 59:26

Loading podcast level 16, teaching while indie.

C

Cheryl Platz 59:31

I know you gave a talk about teaching while indie at GDC, and how did that go? What lessons did you impart, and what was the reaction?

J

Jo Cronk 59:38

That was my first talk at GDC. Thankfully, it was a micro talk. I was part of a panel that we're all talking about related topics under kind of this bigger umbrella of teaching while indie and my part of the panel focused on how to keep your skills sharp while teaching. Yeah. Because one of the things we're constantly doing is we're trying to find part time adjuncts to come and help us teach, especially things like electives at schedules that are totally can work around day jobs. And I have had the weirdest time trying to recruit basically where I will talk to developers about the idea of part time adjuncting, and I can see they get really nervous about that idea. And one of the things that I think people are afraid of is that if they start teaching, they'll just atrophy, and all of their skills will kind of fade away with time, and so I basically just kind of covered how I approach teaching. I've been doing it full time now for about seven years, and I feel like my skills are the sharpest they've ever been. And I put a lot of work into that, but it's very the design of my classes is very intentional to help maintain all of these skills. So I spent some time talking about that. And basically also, you know, it was a little bit of a warning to educators who are in that situation where maybe they, you know, they have some course material that they've been recycling for a while because, you know, it works, and they get good course evaluations at the point where you've been teaching the same thing for multiple years. Engines change, pipelines shift. Anything that they're teaching is basically history at that point. And I think the students, especially students at technical schools, are very sensitive to that sort of thing, and they know that they're being taught like outdated information, so basically showing people how to avoid that situation. And then also it was partially like a thinly veiled recruiting pitch to the underemployed devs in the audience to be like, Hey, if you're currently looking for work, you know you can always do some part time adjuncting While looking for your next studio gig. I think it went over pretty well.

C

Cheryl Platz 1:02:12

The buzz seemed pretty good. I remember how energized you seemed afterwards. Our schedules were at odds, but I am excited to see your talk on the Vault. And it was my first time speaking at GDC as well, so I am anxious to see my talk on the Vault. I will post both of those links in the show notes when they're available, although note to listeners: GDC Vault links usually require a subscription.

E

Enduring Play Computer 1:02:38

Final Boss level: artificial intelligence and game development education.

C

Cheryl Platz 1:02:43

As we wrap things up, tell me about the role artificial intelligence is playing in your class and assignments and your class policy.

J

Jo Cronk 1:02:53

Yeah, as far as like AI goes in my design classes, I allow my students to freely use llms or any other generative AI that they want to use, because I feel like these are tools that will help them prototype things faster, and most of them will never be writing production level engine code. So their CS skills aren't necessarily something that I need to scrutinize. Now in my CS classes, it is a completely different story, where I do not want them using llms, and it's really tricky because there's no way to enforce this. They're going to do whatever they're going to do outside of class. So then part of the design challenge, as far as like course materials goes is like how to design assignments and assessments in a way where you can find out if there is clear core understanding of the material. And yes, they can put stuff into Claude code or, you know, chatGPT and they could just generate a bunch of like functional code, but what I like to do is, and I do find those assignments useful to have them, you know, produce something very specific. But then I want to sit down and talk to them about their process and make sure that if they did copy and paste a bunch of code, they at least understand what it's doing, and they can, like, walk through the code and say, this does this, this does this, this does this. And I have found that the students who are just blindly copying things without that understanding cannot talk their way through their own code.

C

Cheryl Platz 1:04:35

I think that talking is a really interesting point. Like, that's a technique I use in my class too. Is like having a yeah, I'll have you write your take on a game, but I'm also gonna have you speak to it, so that even if you you gave me a whole, whole piece there, you're gonna have to connect with it enough that you can get up in front of people and then talk to it, that there is game design even in teaching now, to get it comprehension and engagement and Oh, yeah. People seeing the value and the mastery of the material enough that they want to do it instead of the AI,

J

Jo Cronk 1:05:07

yeah, I find trying to gamify everything. I mean, gamification works on me. I will absolutely do things more often, brushing my teeth and bathing and all of the normal adulting things if I give myself, like gold stars. So I try to, like, do some of that in my classes too, and try to make it a little bit more fun. Yeah, I have like, systems where certain assignments passing them unlocks other assignments. And so, you know, a lot of the students will just try to hit that sequence as hard and fast as possible, and I love encouraging that sort of thing.

C Cheryl Platz 1:05:44

Well, thank you for this. This has been a really fun conversation, and selfishly, it's been wonderful to learn more about how you approach your curriculum and your students and and the craft of your teaching, in addition, in addition to the craft of your game development, your students are lucky to have you.

E Enduring Play Computer 1:06:02

loading final podcast level links and follow up.

C Cheryl Platz 1:06:07

You're building great futures for the next generation of game devs. But what do you see in your personal future of game development?

J Jo Cronk 1:06:14

One of the things that kind of keeps me at Japan too is that I love being in a place where I can just take random classes, and I am pursuing a Master's of Science in Computer Science at DigiPen right now. So I'm I'm taking classes as a student, which I think the students find super amusing, because sometimes I will be in classrooms with some of the CS students, and yeah, it's I just always want to be learning, if I was independently wealthy, I think I would just have, I would be pursuing like 10 different degrees, or just get all degrees for fun and maybe learn some biz dev skills, because maybe I'll need that at some point. How can people follow you in your work? I try to be off social media as much as possible for my own sanity and mental health, but I have found that I do most of my like work related posting on LinkedIn.

C Cheryl Platz 1:07:17

Thank you again for this wonderful, expansive conversation, and I know I will be a better teacher for having haven't had it. So thank you for your time. This has been super fun, and I hope our paths cross much more frequently in the future in Game Dev.

J Jo Cronk 1:07:32

Excellent. Thanks so much for having me again

C

Cheryl Platz 1:07:37

continue exploring the craft of game development with my book that inspired this podcast, The Game Development Strategy Guide available worldwide from your favorite online bookseller or from my publisher, rosenfeldmedia.com enduring play. Listeners can get 15% off@rosenfeldmedia.com through July, 31 2026 using the code enduring play s2 check out season one of this podcast for the interviews that helped inspire the book, or tune in next time for interviews that help expand our understanding of what it takes to create games that don't just survive but thrive. I'm your host. Cheryl Platz. Until next time, keep thriving.

E

Enduring Play Computer 1:08:23

Initiating final credit sequence. Podcast: Enduring Play. Created and produced by Cheryl Platz. Copyright IdeaPlatz LLC, All Rights Reserved. Explore the podcast at enduringplay.com. Podcast theme composed by Cheryl Platz. To learn more about Cheryl's books, visit rosenfeldmedia.com. Follow idea Platz on Instagram, Youtube or blue sky, I, D, E, A, P, L, A, T, Z. To learn more about idea Platz, the design education consultancy, visit ideaplatz.com. Thank you to David Demma for episode quality assurance. For book and podcast merch Go to ideaplatz.dashery.com.