

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

This episode is brought to you by AG1, the daily foundational nutritional supplement that supports whole body health. I view AG1 as comprehensive nutritional insurance and that is nothing new. I actually recommended AG1 in my 2010 best seller more than a decade ago, the 4-hour body, and I did not get paid to do so. I simply loved the product and felt like it was the ultimate nutritionally dense supplement that you could use conveniently while on the run, which is, for me, a lot of the time. I have been using it a very, very long time indeed.

And I do get asked a lot what I would take if I could only take one supplement. And the true answer is invariably AG1. It simply covers a ton of bases. I usually drink it in the mornings and frequently take their travel packs with me on the road. So what is AG1? What is this stuff? AG1 is a science-driven formulation of vitamins, probiotics, and whole food sourced nutrients. In a single scoop, AG1 gives you support for the brain, gut, and immune system. Since 2010, they have improved the formula 52 times in pursuit of making the best foundational nutrition supplement possible using rigorous standards and high-quality ingredients. How many ingredients? 75. And you would be hard-pressed to find a more nutrient-dense formula on the market.

It has a multivitamin, multi-mineral superfood complex, probiotics and prebiotics for gut health, an antioxidant immune support formula, digestive enzymes, and adaptogens to help manage stress. Now, I do my best, always, to eat nutrient-dense meals. That is the basic, basic, basic, basic requirement. That is why things are called supplements. Of course, that's what I focus on, but it is not always possible. It is not always easy. So part of my routine is using AG1 daily.

If I'm on the road, on the run, it just makes it easy to get a lot of nutrients at once and to sleep easy knowing that I am checking a lot of important boxes. So each morning, AG1. That's just like brushing my teeth part of the routine. It's also NSF certified for sports, so professional athletes trust it to be safe. And each pouch of AG1 contains exactly what is on the label, does not contain harmful levels of microbes or heavy metals, and is free of 280 bad substances. It's the ultimate nutritional supplement in one easy scoop. So take ownership of your health and try AG1 today. You will get a free one-year supply of vitamin D and five free AG1 travel packs with your first subscription purchase. So learn more, check it out. Go to drinkag1.com Tim. That's drink AG1, the number one, drinkag1.com Tim. Last time, drinkag1.com Tim. Check it out.

This episode is brought to you by Helix Sleep. Helix Sleep is a premium mattress brand that provides tailored mattresses based on your sleep preferences. Their lineup includes 14 unique mattresses, including a collection of luxury models, a mattress for big and tall sleepers, that's not me, and even a mattress made specifically for kids. They have models with memory foam layers to provide optimal pressure relief if you sleep on your side, as I often do and did last night on one of their beds. Models with more responsive foam to cradle your body for essential support in stomach and back sleeping positions and on and on. They have you covered. So how will you know which Helix mattress works best for you and your body? Take the Helix Sleep Quiz at helixsleep.com/slash/Tim and find your perfect mattress in less than two minutes. Personally, for the last few years, I've been sleeping on a Helix Midnight Luxe mattress. I also have one of those in the guest bedroom and feedback from friends has always been fantastic. They frequently say it's the best

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

night of sleep they've had in ages. It's something they comment on without any prompting from me whatsoever. Helix mattresses are American made and come with a 10 or 15 year warranty depending

on the model. Your mattress will be shipped straight to your door free of charge. Then there's no better way to test out a new mattress than by sleeping on it in your own home. That's why they offer a 100 night risk free trial. If you decide it's not the best fit, you're welcome to return it for a full refund. Helix has been awarded number one mattress by both GQ and Wired magazines. And now Helix has harnessed years of extensive mattress expertise to bring you a truly elevated sleep experience. Their newest collection of mattresses called Helix Elite includes six different mattress models, each tailored for specific sleep positions and firmness preferences. So you can get exactly what your body needs. Each Helix Elite mattress comes with an extra layer of foam for pressure relief and thousands of extra microcoils for best in class support and durability. Every Helix Elite mattress also comes with a 15 year manufacturer's warranty and the same 100 night trial as the rest of Helix's mattresses.

And you, my dear listeners, can get 20% off of all mattress orders plus two free pillows.

So go to helixsleep.com slash tim to learn more. That's helixsleep, H-E-L-I-X, helixsleep.com slash tim. This is their best offer to date and it will not last long. So take a look with Helix Better Sleep starts now. At this altitude, I can run flat out for a half mile before my hands start shaking. Can I ask you a personal question? No, I would have seen it for a good time.

I'm a cyber-nerdy organism living this year over a metal endoskeleton.

Hello boys and girls, ladies and germs. This is Tim Ferriss and welcome to another episode of the Tim Ferriss Show, where it is my job to interview and deconstruct world-class performers from all different fields. My guest today is John Romero. Who is John Romero?

Computer and video game legend John Romero has designed and published more than 130 games since

his first sale at the age of 16. A teenage programming prodigy, his major achievements include co-inventing a series of revolutionary computer games, Doom, Quake, Wolfenstein 3D, and Commander Keen that launched the industry's most popular genre, The First Person Shooter. The memoir Doom Guy, Life in First Person, is his first book. You can find him on Twitter at Romero and on Instagram at TheRomero. Without further ado, please enjoy this wide-ranging conversation with John Romero. John, nice to see you and welcome to the show.

Well, thanks for having me. I'm excited. I thought I would give a few caveats for people listening. Number one, I'm going to operate on the assumption that most people have very little context on your life story. So I will be asking a lot of very basic questions. I extend my apologies to you and the hardcore gamers out there, but I think that'll be helpful for folks. And some very, very long-term listeners may recognize your name because it came up in conversation with Alexis Ahanian, who is a friend of mine, co-founder of Reddit, who credits your story with helping inspire the creation of that company. And of course, your story and the products of your imagination and hard work have also inspired many, many others. The creation of Oculus VR and the cultural sort of touchstones and the influence of pop culture over decades is really remarkable. So in the process of doing my homework and due diligence for this conversation, the expansive nature of the impact is pretty mind-boggling. But I want to start in the beginning. And by

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

beginning, I mean childhood because I'd like to provide some connective tissue that then leads into later questions. Could you please just describe where you grew up and what your childhood looked like? So I grew up in Tucson, Arizona. I was born in 67. So I did a lot of my early years in the late 60s and the early 70s. Grew up right next to the Yaqui Reservation. And my families of my dad's side has Aztec and Yaqui heritage. And we'd call ourselves Mexicans. And Mexican food was the food that we all ate. So yeah, we lived in a pretty poor place. I only lived in that area until about fourth grade. But it was pretty culturally significant to me because I really learned a lot of my Mexican side at that time. And then we moved, my dad went to the grocery store, never came back. And so my mom got a divorce and met a guy that was a military guy and got married. And

then we moved to Northern California, which was very, very different than the desert in Tucson, Arizona. I would love to hear you expand a little bit on your time in Tucson and your experience with your biological father because I think it paints perhaps a picture. I was reading a piece in The Washington Post that certainly made it seem very challenging, that chapter of your life. It was, I don't know how many generations ago that is, but in human generations, it was a long time ago and people definitely acted differently back then. And I guess in the family and in just socially, the circles that my dad ran around in, drinking alcohol was like drinking water. It was, I never saw him actually drink water. It was always beer. And it wasn't just him. It was basically just about everybody. And you know, people that drink usually will surround themselves with other people that drink because they like doing similar things together. So everybody that I saw always drank. And that always led to all kinds of crazy stuff happening because when you've been drinking, you're going to be doing all kinds of random stuff.

At this point, I think it might be helpful to provide a little bit of context related to the hardships that John endured and overcame in his childhood because I think it informs who he is and speaks to a toughness and resilience that is worth underscoring. So what I'm going to read is an excerpt from a very good piece in The Washington Post titled Why Revered Developers John and Brenda Romero started a game studio in Ireland. And here is that excerpt. John grew up in Tucson, Arizona. His biological father was an alcoholic. Quote, at that point in time, John would have had burn marks on his arms, would have seen his mother getting beaten up and quote, Brenda

said, Brenda is his wife. Money and food were tight. John's father robbed stores at gunpoint on multiple occasions to get diapers. One day when John and his younger brother Ralph made a mess in the kitchen, his mother stressed, told John's father to quote, get rid of them. It wasn't meant to be literal. He just put us in the truck and drove us out into the middle of the desert and told us to get out and then took off. John said he was six. Ralph was four. John recognized where they were and decided to follow the truck spending a half hour in the desert until his mother made his father bring them home. So this is one of quite a number of examples. But I think it is worth including here simply because it is part of the context that makes John who he is. And with that, back to the interview. It seems like when your mom met this military man that it was a real significant chapter shift and shift of environment, shift of experience. And I'm not sure how far ahead we'd be skipping, but could you explain what aggressor squadron was and how you became involved with that? The aggressor squadron was one of the divisions on the military base that he

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

was stationed at in 1983. That was a place where the pilots, this is the Cold War, by the way, the pilots are learning Russian flight maneuvers, combat maneuvers. And the aggressor squadron is where they all learned all of the flight training, I guess that they needed in case they got in contact with Russians. And the point was not to ever get in contact because this space was primarily a reconnaissance space where they were flying U2s and TR1s from to do reconnaissance over

Russia to try and find out what they're doing. But in case something happened, they would know what

to do. So they flew a lot of other planes out of there. I think F5s, a lot of F5s. Yeah, that was the place where they did that. When I started school, the beginning of 1983, but was halfway through the school year of my sophomore year in school, I could actually choose to take computer programming, which was pretty great in the 80s. Because it was halfway through the year, the teacher had to kind of approve that it was okay for me to join because I was halfway through learning basic. But I already knew basic, I was finished with basic, and I was learning assembly language. And when I talked to the teacher and showed her that yeah, I knew assembly and that's what I'm coding. And I know everything about this machine and all that, she was really impressed the

first day that I got into school. And the second day when I came to school, she told everyone in the class what to work on. And then she told me to follow her and we went to her car. And she drove me to the aggression squadron on base and introduced me to Captain Spencer. He was like top gun dude. He was a total rocket jockey guy. He ejected out over the Mediterranean. This guy's done so much stuff. He went to a phone and he said a bunch of different code words into the phone. And then this giant bank vault looking door opened big round one, you know, it opened and we went inside of this vault, a bank vault looking thing. It was in the middle of the aggression squadron where they keep classified stuff. And they had already put away the stuff because they knew that he was bringing someone in that didn't have clearance. And he took me to the back and I saw this mini computer. It was a Chromimco and he said, here's our mini computer. I'm just wondering

if you knew how to program it. And so I got on it and I noticed that there it was using the CPM operating system and that you could program it in basic. So I said, yeah, yeah, I know CPM and I knew basic. So he's like, you got yourself a job, you know. So basically, I got a job the second day I got to school just because of connections, you know, my teacher saw that I knew how to code and then she ran into this guy later that day and told him about me and he's like, bring him on over. So it was just really, really random. How did you get your very first taste of computers? When did you get bitten by the bug? Given the humble beginnings where one would assume, especially at that time, that it would be very hard perhaps to come across or get access to computers.

Getting access to computers in the 70s was basically not going to ever happen in someone's house. Computers weren't even in the home yet. The personal computer revolution was just barely beginning. So only thousands of people in the United States may have had a computer and they most likely built it themselves. So computers were very early. They weren't really in houses yet.

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

But in 1979, I was addicted to arcades. I played arcade games all the time or pinball or anything because it was so much fun and so different. But it costs money. And it's hard for a kid, you know, who's 11 or 10 or 11 to actually have the money to spend on arcades. But my friend came to my house and said, I just found a way for us to play games. It doesn't cost any money. And I'm like, Oh my God, I jumped on my bike and I followed him. And about a half an hour later, we pedaled up to the Sierra College computer lab. It's a summertime in 1979. So I'm 11 years old. And I just follow him. We go into this computer lab. It's really nice and cold. And we basically, he shows me this computer. It's a mainframe in a room. There's a room that's just got a giant computer in it. And there's these terminals, these black and white TV screens with keyboards connected to them. And I mean, there's like one unit. And so he starts showing me these really early text games. And these text games are Hunt the Wumpus and Star Trek and Poison Cookie and these just really early, early games. And I was just like really impressed because I didn't need to pump in quarters and I could take my time with every move. I could just like think about it and then move. That was very, very different. So I played for a little while. I noticed that there were only probably two students in the room. And I said, well, what are they doing on the computer? Because it doesn't look like they're playing. He goes, oh, no, they're programming. And I'm like, what is that? And he said, well, that's actually what you do to make these games. You have to learn how to tell the computer how to put stuff on the screen and what to do, how to think. And I'm like, I want to do that. So I just went over and asked us, one of the students, what does that mean? You know, and it's like print, print statement and so other stuff. And he tells me what it is. And I start asking him some more questions. What does it do what does that mean? And so he's kind of explaining it. And eventually, because I'm asking too many questions, he just gets me the book for basic that's there. And I just take it over to the computer that I was on. And I basically start learning how to code with nobody kicking me off the computer. I'm sitting there with a book and I'm starting to teach myself how to program like a really basic adventure like which goes from room to room, just with a text description. On a high level, it's kind of crazy. 11 year old goes to computer lab at the college and teaches himself how to code. Yeah, it's incredible. And I didn't have a computer for three years after that. So in the meantime, I would go to any computer store in my town or the next town. And if they had a computer there, like Radio Shack has a TRS 80 or something, I would sit down and I would find one of the books that has programming in it. And I would just enter the code into the computer and play around with it and figure out like if it doesn't work, usually why it doesn't work, and try to fix it and then change it. And that's like really slow learning. That was how I learned how to code within my first introduction to computers was at the college because I never saw him anywhere else. Pedaling over and was the main interest at that point, I would assume the creation of games. And I guess the follow on to that is was there something about programming that sort of tickled your brain in a particular way? Was there something else sort of intrinsic to programming that appealed to you beyond the output, which would be the games? So gaming was the only reason why I wanted to start learning how to code, because thinking about arcade games, it was like, I have no idea how I would even make that. But looking at the

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

games that I had played on the mainframe was like, I could probably do that because that's just like really simple stuff on the screen. So when I could get things up on the screen quickly, and they kind of looked similar to what I was playing, I was like, I am totally on the right track. This is going to work. The interest was only to make games. But one of the really interesting things about programming, especially back then, was that the programming experience was extremely immediate. On computers back then, you could turn the computer on and it was ready to code right then. There was no no programs to run to code a computer, it was ready. So the iteration speed of learning was as fast as it gets, I could type something in and run it.

Either it's going to bomb and tell me there's something wrong, or it's going to work, and I'm like, I just learned something new. So that iterative, that fast feedback, iterative nature of programming was really, really conducive to keep my interest in learning how to code and get better at it. It was just like building on stuff very, very fast, learning all the keywords for you and how to put them all together to do certain things. And then it's like, I'm not just solving problems, but I'm starting to architect what I want to do. And so there's these phases of learning where you start to understand that like anything that you can think of, you can make it happen. If you only you had the right words and the right description to like, how do I put a thing on the screen? Well, that's with print. How do I put it over there on the screen? There's some commands for saying exactly

how to get to that point on the screen. So when you start breaking down all of these little problems, and you know how to solve every one of these little problems, you start to have a lot of confidence that you can do what you're hoping to do. And that just makes you want to keep on going. If you're a person who likes to solve problems, programming is absolutely for you because your whole life is solving problems when you are a beginning programmer, you're climbing these walls

constantly to get over them and learn something and go do the new thing. So I'm going to leap forward

a bit. And then we're going to go back and fill in some of the gaps. And I'm getting very much in the deep end of my ignorance pool. So I'm going to need you to lead me through what I'm about to read. But I was talking to my brother last night, who codes is involved with technology. And he pointed me ultimately to a rabbit hole that led me to a paragraph. This is going somewhere on game rant.com. And I was hoping you could explain what this means. So John Romero ended up making the biggest advancement in gaming since the invention of the video game itself.

He was the first person to make a navigable maze appear on a screen giving gaming a third dimension of movement. So we were just talking about problem solving. And this may require some explanation on your part, but could you walk us through the significance of that and how it came to be? Games have been around for long to computer games have been around since the 70s. And 3D games have also been around since the 70s. So there were some really primitive games. First person if you're talking about first person, which is where people really start to get into 3D, there was a really, really early games in the, you know, say 1974 is where 3D games really start on mainframe computers, where it's rendering a hallway scene and maybe a character on the screen

in front of you. And when you move, you don't move smoothly, you move by a trunk of the maze,

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

right? So it's really primitive movement, but it's like it's rendered in 3D. And not until we get to say battle zone in 1980, does it become very smooth and battle zone is line drawn pyramids and stuff in the scene and say another tank, you're in a tank and there's another tank and they're smoothly moving and coming towards you and you're having to use tank controls to aim and shoot at it and everything's coming for you. But it was a smooth moving 3D, but it was line drawn. So the advancements in 3D eventually get to the point where you have solid filled walls. So it looks like a color, you know, walls look a certain color. So when you turn it, it's not like seen through an invisible wall. And then they start getting textures in the mid 80s. So if you play, say, might magic, you'll see it. I think some wizardry games also had it in even Ultima, you know, when you go into the dungeons, you'll see a rendered 3D view that has texture on it. Like someone drew a wall, like an actual bricks or whatever, but it doesn't move smoothly. It's moving by chunks, you know, still on computers, it was a lot slower to get to that point of smooth movement. So when we started doing 3D in 1991, we did do a filled wall, you know, solid color walls, but we could move through that maze perfectly smooth and turn any direction we wanted to and see that maze in fluid, full, high speed 3D, which nobody had seen at that point. So our focus was on trying to make that happen. The speed was the focus for us getting 3D moving quickly became at least at that point with that one early game that was really important for us. It wasn't to do something really amazing with the 3D, but it was just to get a 3D scene drawing at high speed so you can turn around in one second, you can just do a spin. That took a lot of computer knowledge of the hardware of the computer and also how to draw a scene, not using all of the 3D equations that you normally would use to compute a 3D scene, restricting the way that you could move to reduce the amount of computation that you needed to do to turn the scene into a scene that is visually complex in 3D, do that very quickly by not having to worry about certain angles that you would see or turn your head or roll forward or whatever. So we restricted you to only one plane of turning where you're just rotating your head around. You can't look up and you can't look down and you can't tilt your head sideways. So by reducing your motion that way, it reduced the kind of computations that we needed to do. So if I'm hearing you correctly and I may not just because of my limitations as a Luddite or maybe not a Luddite but more of a Muggle in this capacity, but it seems like you applied positive constraints, a number of constraints so that you could use, let's call it an elegant code to dramatically improve the speed of the player experience. And I'd love for you to explain two things for people listening. Number one, when you're saying us, who is us and why do you think you guys were able to create this breakthrough instead of perhaps other teams or were other people simply not looking at it the same way you were in terms of optimizing for speed and using that as the primary driver behind the development? Okay, so us is id Software, which is a company that I co-founded in 1991. We started in 1990, but we officially started a company in 1991. And that's me and John Carmack, Tom Hall and Adrian Carmack. Adrian and John are not related. And it was just us four that founded this company. And in 1991, which is the first year that we were officially in our own office, basically, to make games, full-time all day long at our own company, which is a major achievement for us because like everybody, we have jobs, like we had day jobs, but we had created a game in our off time, and it did really well and allowed us to basically start the company formally. We were doing side scrolling games at that time using a

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

technique that no one had ever used before on a PC. It hadn't been discovered for nine years since the introduction of the PC until John Carmack figured out how to get pixel smooth scrolling horizontally on a PC. Can you give an example just for people listening who may not know the terminology? I know it seems self-explanatory, but a side scrolling game, what would be a well-known

side scrolling game so people can visualize? Yeah, Super Mario Brothers is a side scrolling game, and there were thousands of them throughout the 80s. But on a computer, on a PC, the PC came out in August of 1981, on the PC, no side scrolling game could move a dot at a time horizontally. It always moved by eight pixels. It just chunked across the screen, so it was very chunky and blocky when it moved, and that's because even moving the whole screen itself took a lot of time.

To move it by a pixel was kind of unimaginable because there was no way to control that at a high speed. So when John made the breakthrough of 1990 to figure out how to do that horizontal movement, like a Nintendo, like Super Mario Brothers on a Nintendo, how to do that on the PC, that was basically the moment that idSoftware was born, because it was like, we are doing this in our own company, oh my god, no one has ever seen this before, I'm just blown away. And when we finally

started the company, I think it was like half a year after that, when we started it, in 1991, we made 13 games that year. And we made them at high speed. A game took us two months, but we would make two games at a time. We were so busy. The first 3D game that we made was called Hover Tank

One. And that was just to get our feet wet in 3D, because we knew that 3D is really where games are going, eventually, because at that point, there were very few 3D games, and they were very, very slow. None of them made a wall look like there were bricks on it or rocks or whatever. It was always some kind of solid color, or some little pattern of dots, two different colored dots to make it look like another color that doesn't really exist. It's called dithering when you do that. It wasn't really advanced. So our focus was to make the screen appear like other games, except make it so we can move through the level smoothly, not in chunks, and to be able to turn in any direction and do that at a very high frame rate, because no one had ever done that before. And that was the focus of Hover Tank One, and that game was made in two months. That was John's big breakthrough, was just creating the architecture of that engine and getting the rendering for 3D, going at that speed. 13 games in one year. I know nothing about game development, but I imagine with a relatively small crew, that sounds like a lot of games. What do you think enabled you guys to do that that maybe was not present for other people working on games who had much lower output?

What were the ingredients or the approaches of the principles? Anything at all? Maybe just tolerance for sleep deprivation? I don't know. What were the ingredients that allowed you guys to do that? So the way that we could achieve this game development at such high speed back then was that

when we got together, as in software, we had already individually, other than Adrian, who just learned how to put pixels on a computer screen, but he was a fine artist on paper already for his whole life. Each one of us four had 10 years of game development experience, like all the time. And so we'd already made dozens of games each. Tom Hall, who was our creative

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

director, he was a programmer, an assembly language programmer. He made tons of adventure games.

He did action arcade games in assembly language. So he was already a very fluent programmer. John Carmack, of course, coded since he was a kid. I had coded since I was 11.

So we'd been making games for 10 years, and the amount of games that we'd made was dozens each.

And when you make that many games, that quickly, you learn how to scope and scoping is basically defining exactly what that game should be and not adding to it. Subtracting from it most likely, if you're going to try and get it done. So scoping is like, this is the amount of time that I have.

This is probably the kind of game that we can make and how big we can make it. Let's do that.

And as you get closer and closer to that deadline, you start cutting things away

that you don't need to hit that deadline. When we did it before meeting each other,

when we did it on our own time, we set ourselves our own goals for finishing our own games.

And we got really good at scoping. So when we got together and we saw what we could do together, we also scoped together how big those games should be. And we had to deliver within two months.

So

we had to know how good every one of us is, how fast we can all work, and then scope a game to that limitation. And because we had 10 years of game dev experience, we could just slam games out super fast, come up with ideas, bam, start doing the graphics, doing the animation, programming the engine, making the levels, making the level editors to create levels, you name it, all the tools that are in the background that no one sees to build the game data, put it together in packages that the game can access. There's so much technology that goes on before you get to gameplay programming. But we were used to doing that already for 10 years. So it was not just the experience that we had, but it was also the focus that we had. We were just so happy to be able to make games all day long as a job, which is really just like the luckiest position ever.

So we could focus on doing that. We could scope our games down really, really well. We knew what to cut when it was time to start cutting stuff if we get we're getting closer to our deadline.

And because there was no internet back then, and no one had cell phones, there was complete and total focus because our phone never rang. There was a phone on the wall, and that thing never rang.

So there's no interruptions. Nobody's coming to the house and knocking on the door and interrupting our thoughts. So we could just focus for 12 hours a day at least, and just code, design, you name it, constantly. To build on the emphasis you put on scoping, which I think is certainly applicable to so many different projects, maybe all projects. I mean, that certainly applies to book projects and so on that I've been involved with, podcasts, right, thematically choosing where you go and where you will not go. I was doing some research. And if I'm taking us into a dead end or a cul-de-sac, that doesn't make sense, you can tell me. But I read a discussion of different, I suppose, principles at id Software, and a comment came up, which was no prototypes, which was attributed to you, and talks about you guys being your own best testing team and so on. And I was wondering if, I can't believe everything you read on the internet, of course, but if this is true, if you could elaborate on what you mean by that.

So to get that many games done that fast, you can't be playing around with making prototypes.

Prototypes are like, let's test this and see if it works. And let's test that and see it.

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

There's no testing. It's like, here's the game that we're going to make. Here are all the characters that go in the game. Here's how many levels we have. Here are all the pieces of the levels that need to be drawn. Here's all the animations that need to be created. All the tools that need to be programmed, all that stuff. There's no way to do any prototyping if you're going to get done in two months with four people. So we had no prototypes. We knew how to code already. So there was like, because we can just see it in our minds, we can just visualize it and we communicated with each other just going, okay, it's going to look like this. On the screen, we're going to see mountains in the background. On the foreground, we're going to have another plane and we're going to have trees

there. And we're going to have up in the trees, we're going to have these characters and their AI is going to be limited to blah, blah, blah, blah. And so we just kind of describe what the game is. And we can do that within, let's say an hour. Yeah, the very beginning of a new game. In one hour, we can describe that game extensively for what we're going to create in the next two months. There wasn't like, well, then we have to figure out this. It was like, there's no figuring it out. We're just making it. So there weren't prototypes. It was just, we're making the game.

Later on, with bigger games that we could spend more time on, we could do R&D, but we still knew what we're making. We just needed to figure out how to make that part of it. It wasn't like we'd build a piece of the game and then throw it away. We would build this functionality in, because that was the original design. And if we found out that that functionality actually is a detriment to the game itself, to the soul of the game, then we would remove it. But it wasn't like we're prototyping it. We're making that piece of technology in the game. It was going to be part of the game. But we find out that that is not where the game is really. And then we pull it out.

So let me ask a, I'm sure, naive question. But if you are going to ensure that, for instance, you can take an element like that and remove it later, if it is not contributing to it, reinforcing the soul of the game, do you need to think about that design spec in advance so you don't end up with a piece you want to remove, but that is sort of interdependent with other elements in the game? In other words, just like the architecture, how you approach it need to be decided in advance so that you have the capability of removing something in that way.

When we're coming up with the design of the game, let's say in this case, it's Wolfenstein 3D. The inspiration for Wolfenstein 3D was a 1981 game called Castle Wolfenstein. And it had a lot of really great gameplay features in it that we wanted to put in Wolfenstein 3D, such as when you shoot a guard, you could drag them and hide them around a wall, you could search their bodies, you could go up to storage chests and try and open them and maybe shoot them open, but that makes noise, my call guards, all these little pieces of design that we wanted to replicate, but replicated in 3D because Castle Wolfenstein was just a top-down game from 1981. It wasn't 3D, it didn't move, the screen didn't move around, it was just like every other character's are marching around on the screen and that's it. Like they might leave the screen and disappear, but your screen was static like many games back in the 80s. So when we were making Wolfenstein 3D,

we start programming in the dead body dragging and searching and that was the design. So we already designed that we're doing it, we already got the code in to do it and we're now doing it, we're playing the game and we're going, you know, shooting somebody, going up and searching

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

them

and as we're playing the game, it feels like that doesn't match now what this game feels like because the original game, it was perfect, it matched perfectly with the way that you moved around and the feeling of the original game was more of a stealth game. So Castle Wolfenstein was really the first stealth game made. It was an action arcade game, but it was really primarily about stealthing around and secretly getting out of this castle and escaping it and we were doing that in our game as well. It was a recreation, but when we got to the point where we're doing some of the similar actions that were not just running gun, we knew that that is not what this game strength is. The strength of this game is doing this never before seen high-speed rendering and high-speed movement through this maze and no one had seen that before in slowing you down and making

the player stop to do something else was anti the focus of the game, right? So the focus became like this is the soul of the game. The soul of this game is high speed run and gun. Anything that stops you needs to come out. So even though we programmed in that stuff, we pulled it out because it wasn't

true to the new spirit of the game, the one that we were making. It was like the original game's idea was transferred to our game, but all of the details of that those two games are different. The way that you shoot enemies in the original Castle Wolfe side was not the way that you do it in our game. Our game is much more fluid and simple and easy to control and very, very fast. And so when we started focusing our design on that part of it, we started removing everything that took away from that speed. But there were some things that we had to leave in like to open a door and wait for the door to open. That's actually suspense that we need in the game. If we didn't have any doorways, you're just running through a maze just like blowing everything away constantly and nonstop,

but you have no suspense really. So those doors provided the suspense. One of the original game's focuses was the suspense of wondering if you've made a noise and alerted the guards to come after you or if they heard you trying to open a chest in the next room and you can see them in the next room

and you're just like trying to open a chest and if you shoot it, then they're coming in the room and they're going to try and arrest you. So we had a new focus of our design and that's why we start removing and changing it. Just a quick thanks to one of our sponsors and we'll be right back to the show. This episode is brought to you by LinkedIn Jobs. These days, every new potential hire can feel like a high-stakes gamble for your small business. So you want to be 100% certain that you have access to the most qualified candidates. That's why you should check out LinkedIn Jobs. LinkedIn Jobs helps you find the right people for your team faster and for free. Add your job in the purple hashtag hiring frame to your LinkedIn profile to spread the word that you're hiring. Simple tools like screening questions make it easy to focus on candidates with just the right skills and experience so you can quickly prioritize who you'd like to interview and hire. It's why small businesses rate LinkedIn Jobs number one in delivering quality hires versus leading competitors. LinkedIn Jobs helps you find the qualified candidates you want to talk to faster. So post your job for free at [linkedin.com slash tim](https://www.linkedin.com/jobs). That's [linkedin.com slash tim](https://www.linkedin.com/jobs) to post your job for free. Terms and conditions apply.

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

So let me build on Wolfenstein 3D and I have a question from a friend of mine because I mentioned to him that we were going to be speaking. His name is Alon Lee. He's the co-founder, co-creator of Exploding Kittens, not just the single game but the company that produces many, many tabletop games and card games. And he said, holy crap, amazing. Ask him how the pressure of promising the world a game when he didn't have one somehow could shoulder the cells of his body to do the impossible and will one into existence. And we went back and forth a little bit and he said, let me know if he needed any links to early Romero development stories. That guy literally did the impossible. So could you expand on what he means by promising the world a game when he didn't

have one and how that could shoulder the cells in your body? I mean, that may not be the right way to put it, but to do the impossible just for people that don't have any of that background. We made Wolfenstein 3D. For me personally, that was my 87th game that I'd made. Right after Wolfenstein

3D, we made Spear of Destiny. That took us two months. That was number 88. There was a game called Shadowcaster that views our technology that we had made. So all the tools that I wrote, any gameplay code, menu code, all the stuff, we license our technology and give it to another company and then they make a game. That's game number 89. It's called Shadowcaster. And then game number 90, not that I even was counting back then. It was just like, next game, you know, we're going to make the next game. At the time, at the very end of 1992, after making all of these games, and really when Wolfenstein came out, everything changed because now we have high speed 3D. No one had ever seen this before. Ultima Underworld was out and it was a really great role playing game set in 3D, but the 3D let you turn your head and look up and down and all that stuff, which really slowed the game down. So it wasn't a high speed game. You had stepped slowly through a dungeon, but it was 3D. It was like the 3D is now happening at this point. Like, if it's not ours, it's probably slow, but it is 3D because the world is now going to become 3D.

But it's hard to get to that point. Our focus was speed. That was where we wanted to live. So with the next game, we had had such massive success with Wolfenstein and Spear of Destiny that we jumped

basically to the front of the industry with those games. Like immediately, bam, we are now in the lead

because nobody was making this and this was the new thing. High speed 3D is the future. We're there

first and we need to keep our lead. The planning for our next game after that, and then our planning was happening at the end of 1992, was the creation of the engine that was a Wolfenstein Plus engine after Spear of Destiny. It led to Shadowcaster, but that engine's architecture was too limiting for what we had in mind for our next game. So that's why we licensed that technology out and decided we're not using that 3D engine architecture anymore. And so at that point, we'd made four games in full 3D, high speed, all that. For our fifth game, we needed a new architecture and to come up with a list of all of the things that this new game would do. And we were thinking like, we need to go beyond anything we've ever seen. This next game, and this was the only time we ever did this. The next game needs to be the best game we could imagine playing, that there can't be a better game in the world than this game. What would that game have? So we made a list, a bullet

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

point list of all the things that this new game should have. Obviously, it needs to have light in dark, it has to have light levels, it needs to be like spooky, it needs to have bright areas, which we had no lighting in Wolfenstein. It was just full bright, you know, like the lights were always on. So it's like, we need lighting control, we need height control, we need to be able to go up and down stairs, we need to have giant tall rooms, really cramped rooms, we need walls to be drawn at any angle, which no game in history had ever done at that point, where you can have a wall that's at an angle and like a wall that's jagged and all that. Every game before this next game was basically a maze game. And then when you look at a maze from above, it's just 90 degree wall terms. It's like those classic mazes, labyrinths, you know, where you're just like drawing a line through the maze. That's what every single game from the 70s until that point looked like.

Wolfenstein

3D was a high speed maze game. It was the same thing, 90 degree corners all over the place. So this next game would break that barrier. So we're going to have what we call non orthogonal walls that could create a space that no one had seen before. We were going to make this game multiplayer so people could play with each other and against each other for the first time, because right then local area networks were just barely like they were just barely starting to sell network cards and the cables and all that stuff in all the computer retail stores. So we would make a game that would take advantage of a network if you ever had it, and it would take advantage of the modem to play against or with your friends. And then we would also make the game open so people could change the game to do whatever they wanted and make it their own. Basically, they could make levels, they could make new sounds, new graphics, they could change the whole game basically in mod

it. It's called mods modification. So we would open it up to modders who wanted to create a new experience using the tech that we created. And we would also give away the first piece of this game for free into the world. And you don't have to pay for it. You just get it. And if you like it, you had fun. If you want more, then you buy it from us. That's called shareware. We had this whole list of everything we're going to do. And we were very confident that we could do it. Because we already, John was already planning the architecture of the engine, he already was very confident in how he was going to be able to do that. And at that point, we were just on top of the world, like we had just conquered everything with Wolfenstein 3D and with the spirit of destiny. And so we're just like invincible feeling. So what we did was we took our two week vacation at the end of 1992. And we came back in January of 93, right at the beginning of the year. And we're going to start working on this new game that we called doom. And we decided for some reason that we were going

to put out a press release and tell everybody what's going to be in this game. And it will be the greatest game that Earth has seen. And we did, we put a press release out before we even start writing the game, we put a press release out saying this will be the best game anyone has ever played

on a PC. And this is what's going to be in it. And it's a breakthrough in game technology and all this crazy stuff. So we needed to fulfill that. And so we started working on it.

Okay, crazy, crazy. Yeah, it's nuts. There are, for instance, as thought exercises, some companies that will write press releases in advance of developing a product technology

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

product

to help sort of hone their focus on various elements, right? But that is for internal use only, you guys actually put out the press release. What was the conversation around the press release? Because it's a smart team. There was some thought, I'm sure, that went into that decision.

What was the logic if there was one behind putting out the press release?

I don't know why we would have put out a press release. It doesn't make any sense. But I think that what happened is Wolfenstein 3D came out. Remember at this point, we were so used to making games extremely fast. With Wolfenstein, that was the very first game that we could make in any amount of time that we wanted. There was no deadline on Wolfenstein. We made the shareware version of Wolfenstein in four months and uploaded it. So there was four of us making Wolfenstein. We had Kevin Cloud in there for the very last month to add graphics and do a bunch of other cool stuff. But we were very fast and we didn't have a deadline on that game. And it released in May, the shareware. So we had the rest of 1992 just building up this world domination of Wolfenstein.

There was an internet top 100 list of games every week. And Wolfenstein was at the top for the entire

year after it came out. It was huge. We were just on top of the world. But after we released Spear of Destiny, it came out, I believe, in September of 1992. And no one had heard anything from us in three months. Which back then, with these accelerated schedules and people wanting to know what we're doing, like we're finally now doing interviews with places. We're doing radio interviews. We have the local TV station in our office interviewing us and recording our screens. We have them interviewing our publishers like people wanted all kinds of information about what we are doing. So I think at that time, it was like, let's put out a press release telling people what we're going to start working on. Because everyone is asking us constantly what we're doing next. So let's just do a press release and just like, it's equal information for everyone. It's not like there's an exclusive with this one computer magazine, you know, computer gaming world or something. There's no exclusive. It was here, everybody, this is what we're working on. And now we have to dive back down in the ocean and start making it like, don't bug us, you know. So I think the press release is probably to let everyone know what we're going to do next.

It's incredible. All right, so I want to take a pause here just for a second to ask you about tell me if I'm pronouncing this correctly. Hyperthymia, am I getting that right?

Okay, so could you just define this for folks? And my question is also,

to what extent this has been an asset and to what extent this has had maybe a shadow side or been detrimental in any way. So if you could just explain what it is and then hop into that.

Yeah, hyperthymia is like a memory disorder where the personal hyperthymia has a superior autobiographical recall and it ranges from really good recall of significant events and things that happen in your life that most people would forget over time. With hyperthymia, you never forget those things. They just don't go away. You wouldn't know it if you're young, but as you get older and you start to lose decades of memories because you don't use them and they're

not significant anymore, hyperthymia, you don't lose them. They're always there. And the extreme end of hyperthymia is where you remember every second of your life since you got hyperthymia.

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

And that could be born with, but sometimes it is an induced memory disorder. So people have been hitting the head with a baseball and boom, they have hyperthymesia and they have like, they record every second of their life, which gets to be kind of insane because everything that you've ever done is always present in your brain. And having all of these thoughts, like all of these memories constantly, it's a big hindrance, I guess, in just normal operation, like just living. My hyperthymesia is, I remember basically everything that has happened to me, mostly all the significant stuff and things that are connected to it. So like music and movies and programming and games and that kind of stuff. I would remember basically all of that stuff through all the years and I would, you know, things that I cared about, I would remember those things. I would remember the names of people who wrote games when those games were published, the year that they were published, all of that stuff. So I would remember everything. So it makes it a lot easier to just say, hey, what happened in 1983 when you were doing blah, blah, blah? It's like, well, I was actually programming this stuff, but I didn't have my computer yet. So I had to do it on paper in the kitchen and, you know, it was a soy busy or I could tell people what I did whenever. Have you had that to the best of your knowledge since birth or was that induced at some point? I think I've had it mostly since birth, but it got stronger when I started to learn how to code at really high speed after getting the computer in the house. I was learning so much so fast, I actually had to take breaks from learning because my head was just packed full of stuff and it needed to settle. Yeah. And I had to like stop myself from learning for a day or two before I could jump back in and fill my head full of stuff again. And due to the nature of what I was learning in the early 80s, which is learning every bite of memory in an entire computer, the knowledge of language wasn't as important as the knowledge of hardware at that time. So I needed to learn all of the hardware of the computer that was programming and then the language is on top of that and then start creating off of the top of that information. So that was also a forcing function to induce memory of everything basically that was happening. I have one family member who has almost certainly, I had never had heard this term before, but Hyperthymesia. He can recall just about every fact, every experience, significant or trivial from his past. So he's much older than I am, but can recall the seating arrangement in every class that he was ever in, the every word of every foreign language he's ever studied, etc. And what I've seen is that that is a superpower on so many levels, but also as an example has been challenging for him because negative experiences, arguments, let's just say, slights, kind of moral injuries, all of these things are also as ever present for him. And it's very challenging. And I'm wondering if you have seen any, I don't want to say trade offs, but challenges related to having such an enhanced memory besides what you've mentioned. I'd say that's totally true for a lot of people that have that. For me, the say negative slash hindrance would be that I think a lot of things are similar in importance, even if they aren't. So I would think that this possibly insignificant detail is more important than anyone else would think. And the reason why I think it is because the way it's connected to other things. If that didn't happen, then these other things wouldn't have happened as well. So it's like not knowing like, well, say for this book, what's important, what isn't important. It was really important that I went to the heavy metal concert with this person, you know. Has that been challenging in group environments, like team environments,

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

where you're sort of trying to weigh the relative significance, but because in your mind, some of these elements that might not be deemed important by other people are sort of importance adjacent. So you weigh them more heavily. Has that been challenging in team environments at all?

Not really, because I think the most helpful part of that is just knowing what has been made before, like what games and game design techniques, programming techniques, approaches to certain control issues, like that stuff. Luckily, just having played so many games over so long a time, that has been really helpful to be able to recall solutions or other approaches people have taken that we don't need to research and figure it out again, like reinvent the wheel. It's like, this game in 1984 did this on an Atari 800, and we might be able to do that on the PC or something like that. So it was more helpful to solving issues or being able to make sure that we're not replicating failed designs of the past, you know, and that we were always creating new things I hadn't seen before, which was really helpful. How important or how much of a contributor do you think to the ongoing success of the team was having two games and therefore two options for things to work on at any given point in time? I would say that having two games at a time, it did not impact really anything, because we knew what we were doing on both those games. One of those games came out a month before the other game, right? So that was like the priority. But what we did was kind of we divided our time into two roughly eight hour days or seven hour days. So we would go from 10 to six on one game, and then six to two on the next game. So we just all shift because it was really important to be in sync with each other when we needed to like ask questions or whatever, we were all on a project. And it wasn't like, oh, you're just making a level for the other game. Well, I guess I can't ask you, you know, it was like we're always in sync. So we would shift to the next game and start working on that. And then the next day, we start back on the other game. And the reason why we're doing eight hours and eight hours or seven and seven was because we love making games like we wanted to get these games done, we're learning a ton. It was super fun being together and making these games we'd be laughing. It would be new things that we're inventing. It was just like, it was like being in a lab where things are going right all the time. So it was super fun. And we could do those hours without a problem. And we did it for a whole year. All of 1991 was spent with all the hours that we were awake, working on games, two games at a time, at least, because we were in sync at the same time when we shifted projects, we didn't get creative block with something. It's like, ah, I can't even figure out what I'm going to do with this level. I'm going to just make a level for the other game. We would help each other if anyone got stuck. It's like, ah, you know, I don't know what I'm going to do for this one section. And I'd like, well, put a meat grinder in there or something. And he's like, Oh, yeah, good idea. You know, we would help each other. But most of the time we already knew what we're doing. I mean, we kind of prethought everything out before we started working on these games and we could do it at really high speed. So we just executed because that was the really important part. We got the ideas and we have lots of experience. And it was all about the execution of our ideas and in sync. And luckily, because we had such a small team of four people, that could happen very fast. The bigger the team gets, the harder and longer it takes to make one vision come out the right way. But with four people that have very distinct jobs to do, it works really, really well. Doom, we paused at the press release. You go on to create doom. When did you after its release or maybe in the course of developing it realize, holy

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

shit, this is a monster or it's going to be a monster. This is going to be bigger than even our grandest press releases could have predicted. Was there a moment or an event a day, anything that comes to mind? Several things happened during the development that were interesting in that respect. We knew at the very beginning of making the game that it was going to be the greatest game that we've ever played because that was our focus of, you know, like why we're making the game.

So we already like knew it was going to be awesome. And the best game that we had ever played, that was already like a given. But while we're making the game, we got to a point where Tom Hall left the team in August of 1993. And we brought in Sandy Peterson to replace him doing level design work. I basically took over as the game designer of the game. And that's how we started stripping a lot of stuff out of the game, like lives and score and getting rid of, you know, items that didn't matter. Really streamlining the design. Sandy was making levels. And at this point, the design language for the level creation was figured out because I needed to solve that problem with what are our levels going to look like that maximizes the use of our technology that no one has seen and create spaces no one has ever experienced before in a way that is kind of drastic and not a little bit better than Wolfenstein 3D. It needed to be drastically better than Wolfenstein 3D. So that took some work to kind of come up with this abstract level design style. When that happened, we started to, you know, make stuff. It's like, okay, this is looking really great. Sandy Peterson immediately started picking up that style. Because I like would sit and talk to him at the very beginning about like, here's what is important about using this technology that makes us stand out. And so he started putting those elements in his levels. And when I got to run Sandy's levels, not mine, because I when I make my own levels, I know everything that's in it. So I'm just like testing to make sure it looks good. It's working well. The enemies are doing what I want to do. The pacing is correct. When combat happens is when I want it to happen. When exploration happens, I have the area cleared for the player to do that. All of these things. But when I play Sandy's levels for the first time, I know nothing about it. And I remember the first time when I started playing these levels, I knew what people were going to feel like finally, right? Right. And it was like, wow, this is scary. You know, I'm going through this level and I'm hearing these monsters moving around. I don't know where they're at, but I don't like that they're coming for me somehow. And it was great because I finally got to kind of feel in a game experience way, what people would feel when they're playing it. Now, at that point, I'm so used to the technology that it's like whatever, even though it changed everything, like no one had seen a screen that looked this way. For us, it was just like another day at work, whatever. So I was just focused on the experience part of it. So that to me was really fun to play a level I didn't make an experience in a little way, what people were going to feel like. And then we got to a point where it's like early October. And I, for some reason, look at that press release again and look at the bullet points because I'm like, as a producer type person, I need to make sure we're putting everything in the game. We didn't promise stuff that we didn't deliver on and we need to mostly get that bullet point list done. And I saw that we were missing multiplayer, like we didn't even think about it. We thought about it in January, like, but we get to the point of the game where like the game is mostly done now and it's in it's early October. And I'm like, whoa, yeah, there's no

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

there's no multiplayer. So when I tell John, I'm like, you will not believe this. There's no multiplayer. Remember, we're going to do that. And he's like, Oh, yeah. So John's a genius. And so John figured out kind of immediately in his brain was like, Oh, I'll just solve it with peer to peer networking model. So he figured out how he's going to bolt multiplayer onto the game and make it feel like it's been there the whole time. And so it just took him a few hours, you know, he just needed to learn how to get network packets going between computers over our network.

And then how to get the game, like what data does he send between computers? He's trying to minimize the amount of data. So it's fast. And he learned within just hours, the techniques that he needed to do. And it took a couple days to hammer out some of the speed issues. But it was pretty fast before I'd seen him shoot a character, get two computers going, and he's moving a character

on one screen. The other one you see the guy running around inside the level. It's like, Oh, my God, like we've never seen this before in any game. And then you shoot them and the guy dies. And it's like, Oh, my God, this is going to be the most fun thing in the world. And I go back into my office. And I get that code running on my machine. And we just start shooting each other. And it's like, this is nuts. Oh, my God. And then I'm imagining the level I'm currently working on in seeing two people fighting in another room from my viewpoint. And I'm just like, there's nothing like this on earth. Oh, my God, this is changing everything. You know, so I got so excited at that point. So that was another holy shit, this is going to be so different. And blow everybody away to change everything. So multiplayer was when I saw that, I mean, in this October. And so that was really amazing. And then the lead up to when we're about to release the game, we're not online talking to anybody. But somehow, people knew we're getting close. We were starting to get random phone calls at the office. We have people online.

Coming up with there's like these weird subcultures going on in use net around doom, and people creating games based on doom being the best game that could ever be made. And then then like creating a really crude version of that in calling it smashing pumpkins into small piles of future debris. Because that was such a funny thing we put it as a cheat code in doom. Just there was a lot of internet activity around the fact that we're about to release the game.

And we never said anything. So it was really weird that the world was starting to go nuts before we launched the game. And that was the third like, Okay, but up until that point, every game that we made, we would put it out. And then our plan is to just wipe that game out with the next one, like the next game is going to just destroy that thing. So it's like, boom, games out next, you know, so we never thought that doom was going to be living actively for 30 years, which is this year, it'll be 30 years old stuff is being made for the game on a daily basis all the time. And one of the greatest mods that has been created was released not that long ago this year. Incredible. It's called my house. What people are still making insane stuff for the game. But we never thought that was going to happen. We just thought we're going to make a really great game. And the next game is going to wipe it out. You know, like that was it. That was a goal. Let's hop back to something that you mentioned in passing that you left in 1996.

Could you tell the story of how that came to be? How you ended up leaving the company?

We started making the last game that I had worked on and did software was named Quake.

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

And we started working on Quake in 1995 as a team. John had already kind of started planning the technology at the end of 94 when we were finishing doom too. He was exploring the technology for the next game, which was going to be a massive, massive jump in technology. And so when we started as a team in 1995 to make Quake, it was like making doom, right? It was like,

here's this giant mountain, we got to climb again. There's a lot of tools. There's a lot of foundational work that has to get done. And let's do it. So we just started working on it. And the design of the game also was like, we started making Quake, and we couldn't because the tech wasn't going to be there. But this now was going to be the tech that we could use to make Quake, which is why we named it Quake. So it's like, we're going to do that game. That game was a medieval 3D adventure with combat. It wasn't a first person shooter, because that never existed

at that time. It was going to be a 3D action, medieval, we're going to recreate our D&D adventures that we had and put it in full 3D in a real full 3D game. So that was the design. As I'm making the level editor and kind of experimenting with 3D, there's a lot of other stuff going on at the company because doom changed the planet. And then doom too, again, was even better than the original doom. It was crazy because we only had at the time when we shipped doom where it was only six of us that shipped doom. And on top of that, we had three other people in the company to handle business, all of the managerial type stuff and all of the technical support type stuff that was going to happen. That's incredible. It was a very small company. And working on quake, we got our team up to 10 eventually to make quake. And still with the other three people, I think, on top of that, the company was 13 people when quake launched. So we had a very small team and we had the world coming at us. And there was so much to deal with doom's impact and the articles, the hint guides, the strategy guides, the other projects that we had going that it took up a lot of time, which wasn't bad because at that time, there was a lot of technology creation for quake going on. And I was creating the level editor for the level design team to start experimenting with. And it was really just two other people on the team doing level design in full 3D for the first time. And that was Sandy Peterson and American McGee. So it was still pretty small. We still had more people eventually to hire onto the team to get us to 10 by the time quake was shipped. But at this time, there was probably seven of us. It was pretty small. During that time, the design of that medieval game just kept getting stronger and stronger. It was getting reinforced with more things that could be done to reinforce the feeling of that medieval world and do new things that we hadn't seen the game do at that time in lots of different ways. So it was becoming a really strong design, really strong, really cool. And the problem is that we were spending all of that year kind of spinning our wheels, doing 3D levels, little pieces of levels and stuff. But we didn't have the gameplay programmer that could actually do anything in the game because the engine wasn't

ready. Like we're waiting for that engine to actually happen so we can start using it. John, can I pause you for one second? Just for people who hear this word, but may not understand how it applies within game development, could you just define engine? In simple terms, what is an engine? So at this point in our game development, we had created this term called engine. And the

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

engine is what drives the game. And that is the thing that puts the images up on the screen. It's the thing that loads files and saves files. It's the thing that takes input in and makes gameplay happen. It's what makes sound come out and music come out of the game. And so the core functionality that is needed for a game to run without any of the graphics and any of the actual sound effects themselves, the core technology is called an engine because you can put the data in and it will come out. It will play. You put sound effects in and then use in gameplay code. You just find when those will play and they come out. And so it's an entire set of code that any game can be made from. So it's called an engine. And we pioneered the idea of an engine in 1990 with our original Commander Keen game. We started calling it an engine even before we launched Keen. So

we started that idea in 1990. Now we're talking about 1995, which we've made a lot of games since then, since 1990, 20 something games. So with Quake, we're creating a new engine, one that can do full

3D, one that has a completely different look to it, has more advanced gameplay code.

Everything about it was different. When we got to November of that year, the engine was finally ready. But the team was burnt out because of the massive effort that it took to create the technology and the fact that they'd been making a lot of data that couldn't be used because the engine changed so much that what they made was too primitive. The engine could do more than what they had created.

Because every time we were making stuff for the game, the engine would be at a certain point of speed and we're modifying our levels to make it run well at that speed. But that means it's a little too primitive because the engine got faster and faster and faster. And by the time we reached November, we could render much more interesting views in the world, but we didn't build those worlds yet. We were slowly inching towards them. So it was a lot of like, wow, all the stuff I worked on can't really use it. So people were burned out from that. And the team did not feel like they were up to the task of doing research on a new style of game, which is what Quake was going to be. It was not going to be a first person shooter. They basically suggested that we should just make a shooter because we know how to do it. We can nail it. This is brand new tech that's awesome

already. If we just do a shooter, we can all just like ship a game and take a rest kind of thing. And I was just like, no, we need to do this. The whole world knows what this game is. It's in magazines. Everybody knows what Quake is. And this is what I'm all about is we're pushing our technology

to limit. Let's push design to the limit too. Let's go beyond what we've done before and make something new people haven't seen. I wanted to make this 3D game experience something no one had seen before that goes beyond our first person shooter. And so they just really didn't want to do another unknowable mountain to climb. When is it going to be defined? We don't have a scope that has an end on it now. We didn't either with the engine and we didn't change course while we're doing it either, which is what really harmed the team. So I basically just kind of gave in and said, I will design the first person shooter that will use all of this stuff. And what we've done to date to the best that we can and then release it as fast as we possibly can. So we started doing that in from the middle of November. We basically went into crunch mode at that time.

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

Even though everyone's burned out, they're mostly burned out because they weren't making something that they could see what it was going to be used for. Because we wanted to get it done as fast as possible, we went into what's called crunch mode, which at that time was seven days a week, all the hours that you can, which is at least 10 a day. And we did that for seven months, seven days a week. Oh my God. Yeah. And it was really hard. And after the game was released, it was like a month and a half before I quit because there was still some work for me to do to get the game mastered and created, go into stores and be shipped to customers, that kind of thing. I took a vacation, I came back and mastered all this stuff. And I was already planning for half a year to leave because if we're not going to commit ourselves to furthering design, it's just going to become a technology company. Then I want to leave and go create a design company. And so I already planned after that big meeting, about a month later, two months later, already planned to leave and start another company with Tom Hall. And he agreed that he would love to do that. So I said, after shipping Quake, then we'll start doing that. So that's what happened. After shipping, I started getting in touch with publishers and they also, John Carmack mainly, didn't want to work with me anymore. So they asked for my resignation. I was like, well, I'm going anyway. So let's sign it. It was a very tough game to get through. What was that experience of separating from the company like for you? Was it challenging? Was it strange to be on your own and to sort of leave behind maybe the identity that you had built with that team? What was that experience like just emotionally, psychologically for you? Leaving and starting another company, I was very excited about it. I was excited because I was going to be doing this with Tom, who's a really great friend, amazing creative force, someone super fun to work with. And we both had tons and tons of experience. So I was really excited about that. But I really liked working with the other people at it as well. So it was really like, it sucked, but it was also exciting. Like at the same time, it was like, that was a lot of really great work there. But I'm also excited about what I can possibly do without limitations on what, say, design could do. It was negatives and positives, but I think that it was mostly positive because we're starting to do something new. And that was really, really fun to do with Tom. And we spent about half a year building relationships with different publishers to start Eye on Storm. And finally, after six months, figured out who we're going to side with and signed up with Eidos. What led you to want to write your memoir, Doom Guy, Life in First Person? What were the reasons for wanting to write this book and publish this book? So several years ago, I went to do a speaking engagement in Canada. And they basically had heard the story of Doom's development and kind of already knew some of the stuff I talked about. So they said, can you do something different? Can you tell us about your early life and what made you the person that would make Doom? And instead of talking about Doom, talk about how you became the person to make Doom. So I was like, oh, kind of like my life in games, you know, early games. So I was like, all right, I'll write that talk. So I brought a talk that had a lot of the stuff that happened in my early childhood with all of the exciting things that had to do with drugs and murders and just the low income stuff, which kind of shocked people. They had never heard any of that before. And, you know, hearing the audience and the questions later was just like,

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

people saying, you should put that in a book. Like, nobody knows about that information. I was like, yeah, I didn't really ever talk about it before. And yeah, that would be a good idea. In fact, because I haven't seen anything out there that really documents the id software's, you know, ludography, what game started on what date, when did that game ship?

What's the authoritative book on the development of all of id software stuff? I could actually put that in this history as well. So it was kind of a two fold reason for writing the book was to get the whole story out there from when I was born until today. And to also be a really good chronicle for everything that happened during those years and when those things happened, and a bunch of the cool stories that people had heard before, and just put a lot of really good stuff in there as well. Like, there was a game that I was working on called Black Room.

And every once in a while, I get people asking, so what's up with Black Room? And I always told them, I can't talk about Black Room. Well, Black Room's in this book. So it's really, really people are going to get to hear everything about Black Room.

Is it true that Doom, the name comes from the color of money, or is the origin story a different origin story for the name? Yeah, the name for Doom, it came from John Carmack. He was watching Tom Cruise's The Color of Money movie. And when he was watching it, and he saw the guy walk up to Tom Cruise, and he's got his cue stick inside of his case on his lap. And the guy's like, what's in there? And Tom Cruise looks down at it and looks up at the guy and goes, it's Doom. He was like, wow, that's a really great name. We haven't heard of a game called Doom before. So he said, how about Doom? And it was like, that is awesome. Yeah.

And that was it. Rest is history. So John, people can find you on Twitter at Romero on Instagram at the Romero. Thank you for taking the time. And is there anything else you'd like to mention any other asks or requests of the audience, anything at all that you'd like to mention before we bring this to a close? Well, I mean, I'm really grateful to have had a really fun career making games. It's been a blast from the early days of learning how to program a computer all the way to today, using really advanced technology to make shooters with bigger teams. I've seen everything that's happened during that huge span of time. And it's been just a really, really, really great, a really great time overall. So I'm just really, just happy to have had that kind of a life. And that's all that is basically in this book. Excellent. John, thank you so much for taking the time. And I am looking forward to getting a copy of Doom Guy. People will be able to find it where books are sold. And we'll link to everything in the show notes as per usual to all my fine listeners at tim.log.com. And until next time, be just a little bit kinder to yourself and to others. And as always, thanks for tuning in. Hey, guys, this is Tim again, just one more thing before you take off. And that is Five Bullet Friday. Would you enjoy getting a short email from me every Friday that provides a little fun before the weekend? Between one and a half and two million people subscribed to my free newsletter, my super short newsletter called Five Bullet Friday. Easy to sign up, easy to cancel. It is basically a half page that I send out every Friday to share the coolest things I've found or discovered or have started exploring over that week. It's kind of like my diary of cool things. It often includes articles I'm reading, books I'm reading, albums, perhaps gadgets, gizmos, all sorts of tech tricks and so on that get sent to me by my friends, including a lot of podcasts, guests, and these strange esoteric things end up in my field. And then I test them,

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero — The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

and then I share them with you. So if that sounds fun, again, it's very short, a little tiny bite of goodness before you head off for the weekend, something to think about. If you'd like to try it out, just go to tim.blog/friday, type that into your browser, tim.blog/friday, drop in your email and you'll get the very next one. Thanks for listening. This episode is brought to you by Helix Sleep. Helix Sleep is a premium mattress brand that provides tailored mattresses based on your sleep preferences. Their lineup includes 14 unique mattresses, including a collection of luxury models, mattress for big and tall sleepers, that's not me, and even a mattress made specifically for kids. They have models with memory foam layers to provide optimal pressure relief if you sleep on your side, as I often do and did last night on one of their beds. Models with more responsive foam to cradle your body for essential support in stomach and back sleeping positions and on and on. They have you covered. So how will you know which Helix mattress works best for you

and your body? Take the Helix sleep quiz at helixsleep.com/slash/tim and find your perfect mattress in less than two minutes. Personally, for the last few years, I've been sleeping on a Helix Midnight Lux mattress. I also have one of those in the guest bedroom and feedback from friends

has always been fantastic. They frequently say it's the best night of sleep they've had in ages. It's something they comment on without any prompting from me whatsoever. Helix mattresses are American made and come with a 10 or 15 year warranty depending on the model. Your mattress was shipped

straight to your door free of charge, and there's no better way to test out a new mattress than by sleeping on it in your own home. That's why they offer a 100 night risk free trial. If you decide it's not the best fit, you're welcome to return it for a full refund. Helix has been awarded number one mattress by both GQ and Wired magazines. And now Helix has harnessed years of extensive mattress

expertise to bring you a truly elevated sleep experience. Their newest collection of mattresses called Helix Elite includes six different mattress models each tailored for specific sleep positions and firmness preferences. So you can get exactly what your body needs. Each Helix Elite mattress comes with an extra layer of foam for pressure relief and thousands of extra microcoils for best in class or in durability. Every Helix Elite mattress also comes with a 15 year manufacturers warranty and the same 100 night trial as the rest of Helix's mattresses. And you, my dear listeners, can get 20% off of all mattress orders plus two free pillows. So go to helixsleep.com/slash/tim to learn more. That's Helix Sleep, H-E-L-I-X, helixsleep.com/slash/tim. This is their best offer to date and it will not last long. So take a look with Helix Better Sleep starts now. This episode is brought to you by AG1, the daily foundational nutritional supplement that supports whole body health. I view AG1 as comprehensive nutritional insurance and that is nothing new. I actually recommended AG1 in my 2010 bestseller more than a decade ago, the four hour body, and I did not get paid to do so. I simply loved the product and felt like it was the ultimate nutritionally dense supplement that you could use conveniently while on the run, which is for me a lot of the time. I have been using it a very, very long time indeed. And I do get asked a lot what I would take if I could only take one supplement. And the true answer is invariably AG1. It simply covers a ton of bases. I usually drink it in the mornings and frequently take

[Transcript] The Tim Ferriss Show / #681: Doom Legend John Romero – The Path to Prolific Innovation and Making 130+ Games, How to Find the Soul of the Work, Audacious Ambition, and Building in Monk Mode

their travel packs with me on the road. So what is AG1? What is this stuff? AG1 is a science-driven formulation of vitamins, probiotics, and whole food source nutrients. In a single scoop, AG1 gives you support for the brain, gut, and immune system. Since 2010, they have improved the formula

52 times in pursuit of making the best foundational nutrition supplement possible using rigorous standards and high quality ingredients. How many ingredients? 75. And you would be hard-pressed to find a more nutrient dense formula on the market. It has a multi-vitamin, multi-mineral superfood complex, probiotics, and prebiotics for gut health, an antioxidant immune support formula, digestive enzymes, and adaptogens to help manage stress. Now, I do my best, always, to eat nutrient dense meals. That is the basic, basic, basic, basic requirement, right? That is why things are called supplements. Of course, that's what I focus on, but it is not always possible. It is not always easy. So part of my routine is using AG1 daily. If I'm on the road, on the run, it just makes it easy to get a lot of nutrients at once and to sleep easy knowing that I am checking a lot of important boxes. So each morning, AG1, that's just like brushing my teeth part of the routine. It's also NSF certified for sport, so professional athletes trust it to be safe. And each pouch of AG1 contains exactly what is on the label, does not contain harmful levels of microbes or heavy metals, and is free of 280 bad substances. It's the ultimate nutritional supplement in one easy scoop. So take ownership of your health and try AG1 today. You will get a free one-year supply of vitamin D and five free AG1 travel packs with your first subscription purchase. So learn more, check it out. Go to drinkag1.com slash Tim, that's drinkag1.com slash Tim, last time drinkag1.com slash Tim.

Check it out.