

TOM SMURDON: [video] That's me backing away from the player like this, and then walking forward again, and here's me just walking out of the frame, and I'm walking back in, and then I'll play it again without me talking. [video] So the good thing about that, it was a soft notification, and it wasn't jolting or jarring in any way. Next, we're going to have Pete talk about dynamic room modeling.

PETE STIRLING: Yeah, so dynamic room modeling is a new feature we developed last year, and we shipped it shortly after Oculus Connect 4, so we didn't get a chance to talk about it then, but I wanted to take a chance to talk about it now because I think this is really valuable and I think a lot of people would be interested in how this works. So dynamic room modeling is this feature where we use the existing system that we already have for early reflections and the late reverb, and we often refer to this as the shoebox model, so we have this rectangular prism that we use to model the reflections, and what we add with dynamic room modeling is we're actually resizing that shoebox in real time to match the game space, so whatever virtual environment you're in, we're fitting the shoebox to that environment, so it's resizing in real time, and this is really great because it matches the same performance profile of the early reflections and the reverb system that we already have and we know works well across all the platforms, but it gives you this extra ability to have the space match where you feel like you are, so when you're in an open space, it's louder, longer reverbs, when you're in a tight corridor, you're hearing that sound bounce back and forward, and now I've got a video of that now to show you. [video] And you'll see as I get into this corridor, it gets much narrower. Now I'm going to walk into this bigger space, and you can hear that reverb really open up as you get into that space. So this is a feature, it's available now, it's in the Audio SDK. It works with Unity, we've also got it available for UE4 as well, and it's going to draw this overlay in engines, so you can see what's happening in real time. And there's some parameters you can use to dial it in, so how many rays it's casting and how fast the smoothing is, and the performance of the actual ray casting and stuff like that is really minimal, and it just leverages the game engine's ray caster, so it's just shooting a handful of rays out from the listener position, just to kind of work out where the walls are. And it also allows you to get near a wall, so as you get close to a wall, that wall's there with the audio modeling, and you're hearing a slap back reflection off that wall, so it's a really effective little trick. That it's a minor feature performance wise, but I think it can add a lot of bang for buck.

TOM SMURDON: So that works in Quest too, right?

PETE STIRLING: Yeah, it works on Oculus Quest as well, but again, you don't want to be using early reflections on everything, but it will work with the reverb, so the RT60 and the decay properties of the reverb will be matching if you've got this on your Oculus Quest tile, but this is really just scratching the surface of a much more complex problem, and this is one that multiple people have started working on in different ways, and that is propagation, and I wanted to talk a little bit about our approach to this. This is something that Facebook Reality Labs have been working on for a while, and they've come up with something really special, so I'm going to talk a little bit about the process we went through and talk a little about the problems we're solving and how we see it. So the state of the art for reverb modeling in AAA games would typically be something like this, where you've got your environment with a few separate rooms, and then the sound designer would dial in reverb presets for each of those rooms, and then you would fade between those reverbs as you move through them, and maybe in a really sophisticated theme,

you might have multiple reverbs feeding into each other to try and hear sound propagating from one room to the next. You might have a portal system

[00:05:00]

and some visibility to handle occlusion, and those kinds of problems, but there's a lot of work involved in doing this. There's a lot of engineering effort that's involved, and then the sound designers need to dial it in, they need to work together, and then you've got to step through and debug the whole thing, so we wanted to find a solution that is much simpler to implement and more physically accurate. So we wanted to just take the actual game geometry and then have some materials to describe the acoustic properties of the scene, so you're going to have material properties that are going to be describing say the scattering coefficients, the transmission coefficients, absorption of all the different objects in the scene, and then do some modeling of that in software to accurately propagate the audio through that scene, and so this combines a whole bunch of different problems that have been solved in different ways. So you've obviously got the direct path from the source to the listener, as we always have done, and you've got those early reflections, but instead of bouncing off the shoebox model, they'll now be bouncing off the actual game geometry, and you've got that reverb, but that reverb as well is going to be informed by the geometry, and the reverb itself is going to be directional as well. If you've got a cave over on your right side, you want the reverb from the sounds in that cave to be over on the right, you don't want that all around you, so there's going to be a lot of asymmetry and things like that with the way the actual physical audio model would work. Another separate problem is occlusion, so that would usually be some game engine side thing where you're doing some ray casting to see visibility of the object, but with this sort of a system, it should all be included, and it's also going to be affecting the early reflections, so even though the sound is not visible, you still want those early reflections to be bouncing around the object, and obviously the reverb would not get occluded by a small box in the middle of the room either.