



## Intel Assists With Optimizations in Epic's Fortnite\* and Unreal\* Engine

Intel® Developer Zone (Intel® DZ) interviewed Bob Tellez, technical lead on Fortnite\* at Epic Games. As you probably know, Epic's game Fortnite is an amazing showcase for their Unreal Engine\* technology, and the development team behind Fortnite is continually improving the game, increasing the number of supported platforms, and feeding those insights back into the engine. We wanted to learn more about how their process impacts Unreal Engine – as well as the role that Intel played in assisting with optimizations. Below are some of the key takeaways from our discussion.



Figure 1. *Fortnite* artwork courtesy of Epic Games

Intel DZ: *Fortnite* is known for its accessibility, and we especially see this with cross-system play. When you set out with *Fortnite*, was accessibility always top of mind?

Bob: We've always worked to make the game run on as many machines and platforms as possible. Originally, when we were building the game, we were targeting PC first, in terms of

design and core requirements. It wasn't until we started engaging with the public that we started expanding to work on consoles. But it had been the plan from the start to reach a wide audience and make the game accessible to as many people as possible.

*Intel DZ: Given that you guys work extensively in your own engine, both in terms of Fortnite's development as well as for the engine itself, were you able to leverage some techniques or discoveries around optimization for Unreal Engine?*

Bob: Definitely – we have a culture here of using our technology to its fullest extent. We had used Fortnite over the last several years as sort of a flagship proving ground for Unreal Engine features as they came out, and we worked on platform support as they were added. Basically, when we had a need to do something in the game and it coincided with other people's needs for Unreal Engine, we focused on making sure we wrote code in such a generic way that other people could use it for their projects as well. We did that for every single platform we released on, going through with a fine-toothed comb to make sure everything was working in a fast and efficient way.

Working on the engine behind Fortnite is exciting for us because all of the work we do benefits every other Unreal Engine licensee. Although we have a lot of problems to solve, all of our solutions help everybody out. Everyone benefits from this work.

*Intel DZ: Which features or improvements are you particularly proud of?*

Bob: We completed a refactor for how network replication in Unreal Engine works to make it more event-based, which saves time for dedicated servers. There's less lag and latency between actions, making it a better experience for everyone playing the game. That feature is included in Unreal Engine 4.20 and subsequent releases.

*Intel DZ: Since the game has been released, would you say that most of your efforts have been over network issues and latency, improving the experience there? Or are you still working on graphics, assets, and other aspects of the game?*

Bob: We're always making optimizations for all aspects of the game across all platforms – and to Unreal Engine in general, too. One major effort was getting consoles to run in 60 FPS mode, as well as introducing dynamic resolution scaling. So, yeah – network bandwidth usage, server CPU utilization, CPU/GPU threads, server memory – we're always working on every aspect of the game.

*Intel DZ: How do you combine all those optimizations? It sounds pretty challenging to coordinate.*

Bob: We end up wearing a lot of hats. We also have separate teams to handle separate items; we use different engineers to optimize for certain consoles or for dedicated servers, for instance, to make sure their performance is really strong. In our position, we're mostly trying to

coordinate strategic efforts to identify things that can be optimized and to help the individual teams come up with possible solutions. So, we distribute the effort.

Intel DZ: *In the early days of Fortnite, were the efforts on making the high end or low end more performance-friendly?*

Bob: We actually started by identifying what our target machines would be – basically analyzing who our potential player base was. After figuring out what kind of PCs they'd be using, we determined what each type of target machine would be. We've kept that up as we've gone through development, occasionally re-analyzing to figure out what new machines might be popping up. Optimizing for only PC was much easier because there are fewer settings and combinations to track, but our decisions were largely data-based from analytics on our potential players.

Intel® DZ: *How did creative input factor into the design process?*

Bob: The artists make the decisions around how the game looks; engineering then reviews the game design to ensure it is doable. Regarding the scalability of it – we see a target and we go for it. It doesn't have to do with the art style – whether it's realistic or cartoony, the optimization process is the same for both.

Intel DZ: *So it's not a question of "can we do this?" when it comes to art and creative direction, but rather "how do we make this possible?"*

Bob: Absolutely, even those of us responsible for performance want this game to be as fun and crazy as possible. Sometimes it doesn't work, but we always put our heads together to see what happens. It's important to note that, as we're talking about optimization, our direction really comes from features. Early in the project, we built RPG-like systems for Player vs. Environment (PvE) – a.k.a. "Save the World" mode – and we decided to add what we call the gameplay ability system. It's more data-driven, where we don't have to make individual items; instead, we refer to a spreadsheet of attributes. This spawned a whole set of features that filtered into the main branch of Unreal Engine that is now available to all developers. On the rendering front, because players can build walls and destroy buildings in a dynamic way, we developed distance field ambient occlusion and distance shadowing effects to give the game higher fidelity graphics. Those features, which have also been released, were a requirement because the game is so fluid that we couldn't really pre-bake anything.

Intel DZ: *Did your optimization process and workflow change as you worked on the title?*

Bob: Workflow certainly changed over time as the number of people involved in the process fluctuated. We optimized as we developed the features, but as we got closer to release we threw a lot more people on it to get it ready for the public. The process has definitely improved and become formalized over time. Automating performance tests and getting constant feedback

on game performance helps a great deal, as does being able to get performance feedback from our in-house daily playtests.

For example, if we see we're dropping frames for a certain console or spec, we can call that out in our daily report. Quality Assurance (QA) will have already gathered profiles – consoles or PC – and using our tools they'll tell us what the frame breakdown is, so we can dig in and optimize it. We'll either implement something new or identify a change that has led to a drop in performance.

*Intel DZ: So as you test each day, you know that only a few things have changed if you see a drop in performance, and that lets you narrow down possibilities?*

Bob: Yes. We're a fast-paced company and we want to get our technology out there as fast as possible. Manual testing was proving to be too slow, so we devised a system to automate these tests, and we get results not only same day, but sometimes within a couple of hours. We went from playtests every day to multiple tests per day – we'd like to get it to every ten minutes if possible.

*Intel DZ: Do you find the mix of hardware challenging or fun? Mixing up on the same platform, distributing your work across CPU/GPU compute, that sort of thing?*

Bob: Oh yeah, it's super fun to get a new piece of hardware, figure out how it works, then figure out what its strengths and weaknesses are. After that, we leverage our platform-agnostic code to change the underlying mechanisms to take advantage of how the hardware works.

*Intel DZ: How was Intel able to help with the optimization process?*

Bob: Intel has been very helpful during this project. Intel reached out to us early on and told us that they'd like to help us make the game accessible to a large audience. Occasionally they have come over for a week or so to headquarters, working with us to analyze Intel-specific hardware to understand our bottlenecks and how to improve them.

A couple of years ago, we worked with them on the rendering side and they helped us analyze Fortnite on Intel GPUs to give us deep information on where we were bound; were we using too many draw calls, were we vertex-bound, were we bandwidth-bound, etc.

One time, our screen space ambient occlusion path was incredibly bandwidth-bound, so Intel made suggestions on how we might improve that, along with prototyping different options. We worked with them on how to include that in Fortnite and Unreal Engine, which let us stay focused at a higher level around features and optimization without worrying too much about Intel-specific hardware.

We also got some great insight as to how the hardware works – usually with every Intel visit we'd get one big improvement. For instance, we were doing a lot of texture fetching in our

terrain behind conditional expressions, which was extremely expensive for certain pieces of hardware. We probably wouldn't have paid attention to it, but Intel pointed it out and it gave us a lot of performance back. In addition, a portion of the work Intel did on poly targets in Unreal\* Tournament informed some of our decisions around Intel GPUs while building Fortnite.



Figure 2. *Fortnite* Season 7 artwork courtesy of Epic Games

Learn more about how you can use Intel tools in your game at [Intel Developer Zone](#).

For more personalized game developer content, join the [Intel® Game Developer Program](#).