CASE REFERENCE · TECHNOLOGY & INNOVATION

Vector Gaming

FDX - Apex (GPU)

Vector Gaming	Technology & Innovation	Develop · Operate	• C#	• Vulkan
CLIENT	INDUSTRY	DELIVERED AS	• 350M-parameter LLM	• Distributed GPU Training

DISTRIBUTED MODEL TRAINING

Cutting a two-month NPC model training run down to size for Vector Gaming

Vector Gaming needed serious GPU horsepower to train a custom 350-million-parameter model for smarter game NPCs. We put our private GPU cluster behind their Vulkan-based distributed trainer and cut a two-month run down to a fraction of that.

01

THE CHALLENGE

Vector Gaming had built its own open-source LLM training framework to create a small 350-million-parameter model for game development, the kind of model that gives in-game NPCs real intelligence and makes a game more immersive. The framework is deliberately game-friendly: written entirely in C# with no reliance on Python, and built on Vulkan rather than NVIDIA CUDA so it fits the way game hardware actually works. The problem was time. Training the model on local hardware was going to take around two months of solid running, which is a long time to tie up a workstation and a long time to wait to find out whether the model was any good.

02

THE APPROACH

Vector Gaming's framework is built to scale out. A server app assembles the model file in chunks, much like a BitTorrent file, while training nodes run on whatever hardware is available, using the Vulkan SDK to soak up all the GPU on each machine, train their piece, and send the chunks back. What it needed was more GPU to run those nodes on. We put our private GPU compute cluster behind it, running the training nodes across the cluster so the distributed framework could do what it was designed to do at a scale a single workstation never could. Because the framework is Vulkan-based and pure C#, it dropped onto the hardware without dragging a CUDA or Python toolchain along with it.

03

THE OUTCOME

The two-month local training run collapsed to a fraction of the time once the framework had a cluster of GPUs to spread the work across. Vector Gaming got its 350-million-parameter model trained and testable far sooner, which means game developers can get their hands on smarter, more immersive NPCs sooner too. The framework stays open-source and game-friendly, pure C# on Vulkan, and Vector Gaming now knows it has somewhere to turn for the compute when it is time to train the next, bigger model.

CASE REFERENCE · TECHNOLOGY & INNOVATION

Vector Gaming

FDX - Apex (GPU)

Vector Gaming	Technology & Innovation	Develop · Operate	
CLIENT	INDUSTRY	DELIVERED AS	• C# • Vulkan • 350M-parameter LLM • Distributed GPU Training

04 SUMMARY & BENEFITS

- ✓ A custom 350-million-parameter model for smarter, more immersive in-game NPCs.
- ✓ Our private GPU cluster ran the BitTorrent-style distributed training nodes at scale.
- ✓ Vector Gaming's open-source, pure-C# framework runs on Vulkan, with no CUDA or Python dependency.
- ✓ A roughly two-month local training run cut down dramatically.

Want a result like this?
Talk to us about your own challenge — info@firsttech.digital or +27 10 501 0800.

Let's Talk →