Image courtesy of Meta Immersive Synthetics

TRENDS AND BEST PRACTICES FOR **FLIGHT SIMULATION**

INFORMATION PAPER

UNREAL ENGINE

TABLE OF CONTENTS

Image courtesy of Meta Immersive Synthetics

Introduction **3**

Trends **3**

Content pipeline evolution 4

The influence of the digital twins trend 4

The geospatial world 4

Deployability versus immersion: a complex balance 10

Realism and training accuracy 19

The Unreal Engine simulation ecosystem **27**

Environmental content creation 28

Terrain/environment creation 29

Models creation 30

Flight dynamics and scenario 32

Preparation 35

Execution 35

Interoperability 36

Co-simulation 36

Behavior datasets/models 36

Deployment and infrastructure 36

Infrastructure 38

Rendering 38

Display devices 39

Haptics and user interfaces 39

Audio 39

Implementation examples **40**

A DIY flight simulator tutorial 40

Antoinette Project demo: behind the scenes 41

Summary **43**

Image courtesy of Meta Immersive Synthetics

Introduction

The use of simulation for training individuals engaged in high-risk or high-cost activities has grown exponentially in recent years largely due to the increased availability of highly realistic synthetic environments. Nowhere is this more true than in pilot flight training.

In this paper, we will outline the state of the flight simulation industry and discuss trends that are accelerating the way pilot training solutions are being developed. We'll also provide examples and insights from experts about how the industry is changing and its trajectory for the future.

The goal of this document is not to describe a unique formula to build a relevant flight simulator, but rather to deliver a blueprint illustrating what can be achieved with commercially available technologies, standard formats, and open source solutions. The content of this document is based on the state of the art, and the literature, in the fast-growing flight simulation domain as of spring 2022.

Trends

In recent years, the simulation industry has seen several global aerospace simulation trends that have led to the current state of the field. In this section, we outline some of these trends and hear from industry leaders regarding current and future technology, particularly how we can make use of it now to create more efficient and effective simulators.

Content pipeline evolution

Traditional content pipelines require bespoke datasets, such as terrain and 3D models, and highly customized software, hardware, development environments, storage solutions, and delivery methods.Traditionally, these pipelines were time-consuming and human-intensive to develop and maintain, and with the use of proprietary products, service providers, and protocols, the upgrade path for development limited.

Now, we see that the content pipeline that's worked for a long time for vehicle design and creation—easily accessible data that flows readily between systems and applications—is coming to the world of geographic content pipelines.The new paradigm gives us greater availability of high-quality world data, a proliferation of cloud-based storage offerings, and a commitment to open standards across the simulation industry.

This wide-open world of data, hardware, and software has given rise to new approaches to content development for simulation. The traditional approach of developing stand-alone terrain datasets or 3D models is giving way to on-demand availability of worldwide data and procedurally generated content.

With the technology convergence we see in the metaverse context, the data flow is evolving strongly with data and file formats transferring more easily between applications and systems. Access to information is becoming more and more open. For example, the creation of terrain for simulation offers many open and flexible innovations to accelerate the process. Most off-the-shelf CAD and visualization software packages can read 3D formats or GIS data in a variety of file formats, or generate procedural terrain. For those with more specific needs, multiple service providers are working on reproducing real-world environments with an astounding level of detail, all available in open standards.

The influence of the digital twins trend

Innovative industries with heavy data requirements—virtual production, architecture, manufacturing, and game development—have parallel needs when it comes to data ingestion. When aiming to achieve the same goal, these industries create a technology convergence that aligns with the world of simulation trends.

The digital twin industry, in particular, has paved the way for such democratization of data and processes and continues to drive it forward. The production of digital twins for aircraft is reaching a level of maturity where aggregating the data isn't the bottleneck anymore; engineers can instead focus on how to make it smart.

An aircraft's digital twin is concerned with its takeoff and landing locations, the appearance of the terrain it flies over, and the characteristics of the aircraft's movements. A flight simulator can easily leverage the same data ingestion pipeline and development tools to use this real-time data for a cockpit view.

Because of the rise of digital twins in several industries, developing such a system in the simulation context is almost as easy as mixing and matching existing tools and processes.

When moving from digital twins to flight simulation, the next step is to make sure these digital twins come to life, with data attribution and semantics connected to the simulation capabilities you plan to develop.

The geospatial world

Key associations such as the IEEE (Institute of Electrical and Electronics Engineers), the OGC (Open Geospatial Consortium), the SISO (Simulation Interoperability Standards Organization), the USGIF (United States Geospatial Intelligence Foundation), the Khronos group, and many others are ensuring that the simulation community continues to leverage open industry standards and modern pipeline approaches, in addition to the traditional labor-intensive process of providing GIS source data for simulation needs. These smart groups are also aiming to solve the complex problems of data attribution, enabling all types of industries that rely on simulations—defense, automotive, civil aerospace, and so on—to leverage a virtual environment that is not only a beautiful representation of the real world, but also a semantically correct and computer-vision-ready data set.

Brady Moore
Director of Mission Support Cesium

Rapidly developing world events, coupled with the proliferation of sensor data, demands that training simulations use the maximum amount of available data to create environments that reflect the real world. Today's simulations require the inclusion of multiple domains and data from partners around the world.

Data about the operational environment is being gathered and shared faster, in greater amounts, with better resolution, and at lower cost than ever before. Photogrammetry from satellites, aircraft, small drones, and even handheld devices are part of our 3D data stores. To make training simulations more effective, this data must be quickly delivered and integrated wherever users are being trained.

Historically, training simulations were limited in scope and scale, and took considerable time to build. Integrating with, or iterating upon existing simulations was equally time-consuming and complex. Training environments were constrained to what authoritative data sources could represent and often could only cover one domain or use case. Simulations were thus limited in effectiveness, lifecycle, and accuracy of the training environment, and in their extensibility to larger simulations covering many domains and geographic areas.

Training and flight simulations built using game engines and open source technology are more effective, more quickly developed, more cost-effective by being reusable, extensible, and integratable, and are delivered to the point of need. Moreover, the use of open source tools and open formats allows for rapid addition and combination of new data to create more accurate, up-to-date training environments.

Additionally, commercial aerospace companies are producing incredibly sophisticated flight simulations. With CAD models of the actual aircraft, engineering data, real physics engines, and aeromodels, they create flight simulations that are, from the computer's perspective, indistinguishable from a real flight. The combination of geospatial and game-engine technology provides the virtual world for the simulated aircraft to fly in. This is a major change from past flight simulators, which had artist-created landscapes for limited areas and confined pilots to the boundaries of the visualization. Using open source technology like the **Cesium for Unreal plugin**, pilots can take off from any airport on the planet and follow an actual flight path to any destination.

This has been the result of a number of new technologies that are being quickly adopted across the industry and by forces themselves. Streaming 3D geospatial data, as represented by the **3D Tiles open format**, enables organizations to deliver training and flight simulators with the most up-to-date geospatial data with market-leading performance for data transport speeds, bandwidth consumption, and runtime capability. This makes for visually realistic, current environments that can be used in simulations anywhere in the world, on any device. It also enables users to combine authoritative datasets with dynamically gathered geospatial data to create the most valuable, realistic simulations possible in short timeframes. An example of the speed to value for this data can be seen with **University of Tokyo's Cesium Story showing 2D and 3D models of the battlefield in Ukraine**.

Cesium for Unreal provides the ability to incorporate this data into new or existing simulations, which is key for a joint, all-domain data strategy and maximizing life cycle effectiveness. Two early examples of this evolution are Army Game Studio's aviation simulation of Yuma Proving Ground using Army One World Terrain data, and a game developer in Finland assembling his own flight simulator in under an hour. Both stream 3D Tiles, and thus can use the other's data to recreate a real-world environment with relative speed and high precision. Where proprietary formats and engines prevented this kind of interoperability in the past, game engines and open source technology have made it a reality today.

Product image courtesy of Immersive Display Solutions, Inc.

We see two key trends that are efficiently changing the creation process for simulation terrains and large environments:

- **Photogrammetry.** The ability to scan the world with drones or digital cameras, and automatically reconstruct the environment from these surveys, has sped up the acquisition of terrain data to an incalculable degree. Companies like Reality Capture are simplifying this process by providing all the necessary tools to make the reconstruction of the 3D elements available to all users and creators. Photogrammetry has never been as efficient as it is today.

- **Procedural content generation.** Weather patterns, heat maps, water flow, ground composition, and other non-visual or non-static data can be gleaned in real time via machine learning and AI. If such data is available offline, it can be pre-loaded to sync with the real-time data provided to the operator or user, with visuals of metrics or graphic elements as desired. This extra layer of semantic information delivers an unprecedented level of real-time information to the operator.

"

If a given scenario in real life would cause the pilot to sweat and stress, we want those same feelings happening in simulation. Realism is how we prepare pilots for the critical moments.

USAF Col. Scott Koeckritz
(Chief, Test & Training Branch)

Michael Putz
Co-founder and CEO, Blackshark.ai

Since their early days, image generators have been evolving and improving significantly. Virtual environments, however, are still mostly shown in a simplified form, far off an accurate depiction of the surface of our planet. Until today, there's been no viable economic answer to the need of agile mission planning in photorealistic environments based on rapidly changing theaters.

Imagine an end-to-end pipeline offering capabilities to automatically reconstruct critical mission theaters in close to real time, based on any input covering the entire range of available (sensor) data like satellites, planes, UAVs, and others. Details not provided by sensors but essential for perceiving the reality as truly immersive will be filled in by an advanced AI, ensuring the reconstructed replication of reality is always perceived as true to reality.

Blackshark.ai is providing a solution disrupting traditional tool pipelines as used in most image generators by including machine learning and large-scale cloud-based computation. The vision of this solution is to offer a capability to accurately portray and simulate our planet' surface in near real time.

A major focus is fast turnaround times in adapting to changing input data. Right now, Blackshark.ai can reconstruct the surface of the entire globe in less than 72 hours, resulting in more than 1.5B of buildings and over 30M square kilometers of vegetation. This patented approach is based on the automated large-scale analytics of reality captures, either 2D satellite images or 3D laser scans, and applying deep-learning techniques to understand the topology and semantics of the given input data. The results are categorized using an indexed hierarchical description of the surface of our planet and forming the foundation for any type of high-fidelity physical simulation. Designed for mission review or training scenarios, a set of integrated tools is allowing for manual editing if desired, or to provide automated permutations of the entire reconstructed scene for simulation and training of the same theater under various changing side-conditions.

Storage and handling of the massive data sets resulting from the depiction of reality requires innovative solutions, too. Data can no longer reside just on the client side. Next-gen flight simulators therefore need flexible asset management tailored for the available client bandwidth and latency. Predictive downloading, based on AI predictions or other heuristics, will help maximize the utility of every client connection and simulation scenario. In combination with bespoke level of detail management, this enables a requirement of multi-projection with very high frame rates—even in constrained environments.

All together, this integrated solution is paving the way forward towards a vision where the entire planet surface can be reconstructed and simulated in close to real time for use cases in many industries and scenarios going beyond traditional flight simulation. Users of Unreal Engine will be among the first to get their hands on this technology via a custom developed plugin solution.

(Automated reconstruction of San Jose Airport (SJC) in Unreal Engine 4)
Product demonstration imagery courtesy of Blackshark.ai Inc.

As a result of this technology evolution, the data collections coming from different domains (including consumer markets) are growing, and becoming easily accessible. We see companies such as ESRI, Cesium, Luxcarta, Aerometrex, and many others starting to provide large open world datasets and detailed visual content to the simulation community.

ESRI

Rex Hansen
Principal Product Manager,
ArcGIS Runtime and Maps SDKs at Esri

Our world today is becoming increasingly digital. Many organizations are pursuing the use of digital twins as virtual representations of physical objects, processes, and relationships in the real world. GIS is critical for any digital twin solution because it provides the geospatial foundation to accurately locate and represent digital twins and the digital environment in which they exist and operate.

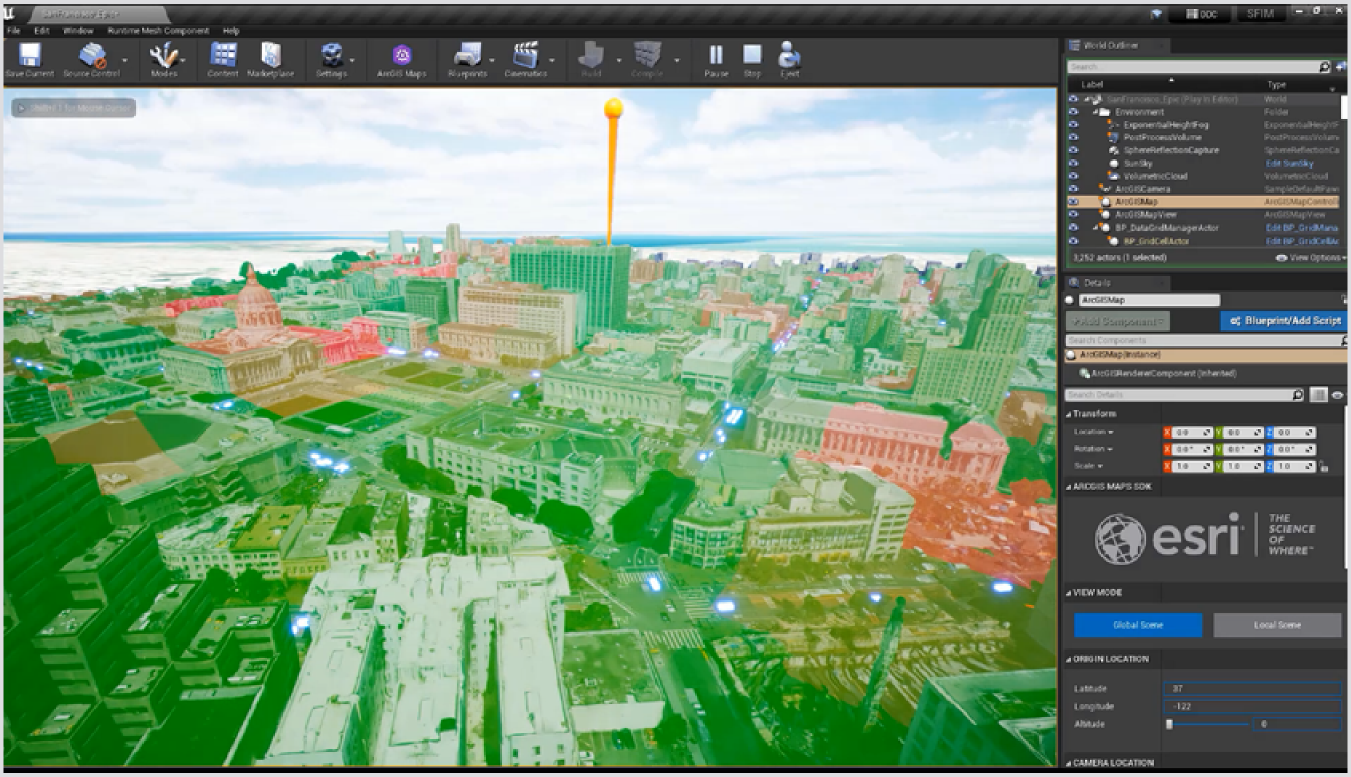
Esri is the global leader in GIS and delivers ArcGIS, the geospatial technology to connect digital information, environments, and assets within a spatial context. We provide the system to capture, integrate, manipulate, and distribute highly accurate digital representations of natural and built environments.

Many industries seek to use digital twins to improve the physical world, which requires accurate geospatial data and context, geo-specific output from intelligent simulation engines with industry-oriented models, and flexible developer tools to deliver photorealistic visualization experiences.

ArcGIS provides the geospatial system of record to support responsible delivery of digital twins and real-world environments to machine-centric simulations with industry-oriented models for training and testing. For example, simulation engines can be used to iterate and test vehicle traffic flows on a network and movement of oil and gas through a refinery, check long-term effects of climate change, or optimize distribution of humanitarian aid within a geographic region.

Unreal Engine provides the developer tools to deliver custom, immersive, interactive, photorealistic visual experiences with which you can combine digital twins, their environment, and simulation results—so organizations can improve processes, enhance decision-making, predict outcomes, increase revenues, and reduce risk.

Esri provides the ArcGIS Maps SDK, a plugin that integrates with Unreal Engine to provide developers with ready access to a geospatial canvas and ArcGIS services and data. I'm excited to see the vast opportunities for organizations who seek to improve their workflows with a foundation on GIS provided by ArcGIS, intelligent simulation solutions from a multitude of providers, and beautiful, engaging visuals powered by Unreal Engine.



Using Unreal Engine, ArcGIS Maps SDK, and Presagis simulation engine to visualize navigation of vehicles through an urban landscape overlaid by cells of various colors that indicate traffic density

Deployability versus immersion: a complex balance

Traditionally, simulators have been of very defined and locked designs, which limits the upgrade path. With the ability to output to many types of display devices, from enormous domes to AR glasses, and even mobile devices, and commercially available XR equipment (Oculus, Vive, HoloLens, etc.) and other readily available hardware, users have a lot more options for developing new training systems and improving existing solutions. Advantages of such a wide range of broadcast options include automatic asset quality-scaling based on the hardware's features, and the ability to adopt a "create once, deploy many" strategy for a number of possible training solutions on different media.

The tools you choose should also support a wide variety of display hardware so that you may choose the most effective medium for developing and delivering your training without the constraint of limited hardware compatibility. The rise of OpenXR is bringing a level of standardization between the devices, removing the burden to adapt to any single set of hardware drivers and interfaces while applications are being developed.

Likewise, Unreal Engine is designed to run smoothly on a wide range of hardware and OS configurations, taking advantage of the most powerful and modern systems while still performing adequately on cheaper or aging hardware. If you have access to VR goggles and a PC, chances are you can already create an efficient VR-based training simulation with your existing hardware.

Advances in deployability of training systems have created greater access and availability of critical training systems, while also decreasing overall development and maintenance costs of training systems.

Image courtesy of Meta Immersive Synthetics

With the current state of technology, we have the option to use a game engine to create experiences and export them to any modality of training distribution, from traditional PC base monitors to large domes but also mobile handhelds or XR—augmented, mixed, or virtual reality—for immersive training. Such development environments are designed to ingest files from a variety of standard formats, and include programming tools to customize environmental and user actions/reactions as well as the user interface.

The variety of supported platforms doesn't restrain the developer's outcome to a single target. The training needs and efficiency can now be replaced at the core of the curriculum definition, and based on the budget or immersion needs, a target platform of any scale (and cost) can be chosen.

Because they were originally designed for consumer hardware systems, game engines can run smoothly on a variety of configurations without requiring expensive state-of-the-art hardware. With a consumer PC and GPU, a pair of consumer VR goggles, and a game engine, you can develop an efficient VR-based training solution for many use cases without further investment in hardware.

John Burwell
Global Head of Simulation and Training at Varjo

Flying an airplane requires a high degree of precision, attention, reliability, and spatial awareness. Because of the demanding nature of flying, flight training has traditionally relied on highly realistic simulators that closely replicate the operational environment of the actual aircraft. As a result, today's high-fidelity simulators rely on a range of complex display systems that use a variety of screens, domes, projectors, beam splitters, and mirrors to support the visualization of scenes which is essential in creating the suspension of disbelief.

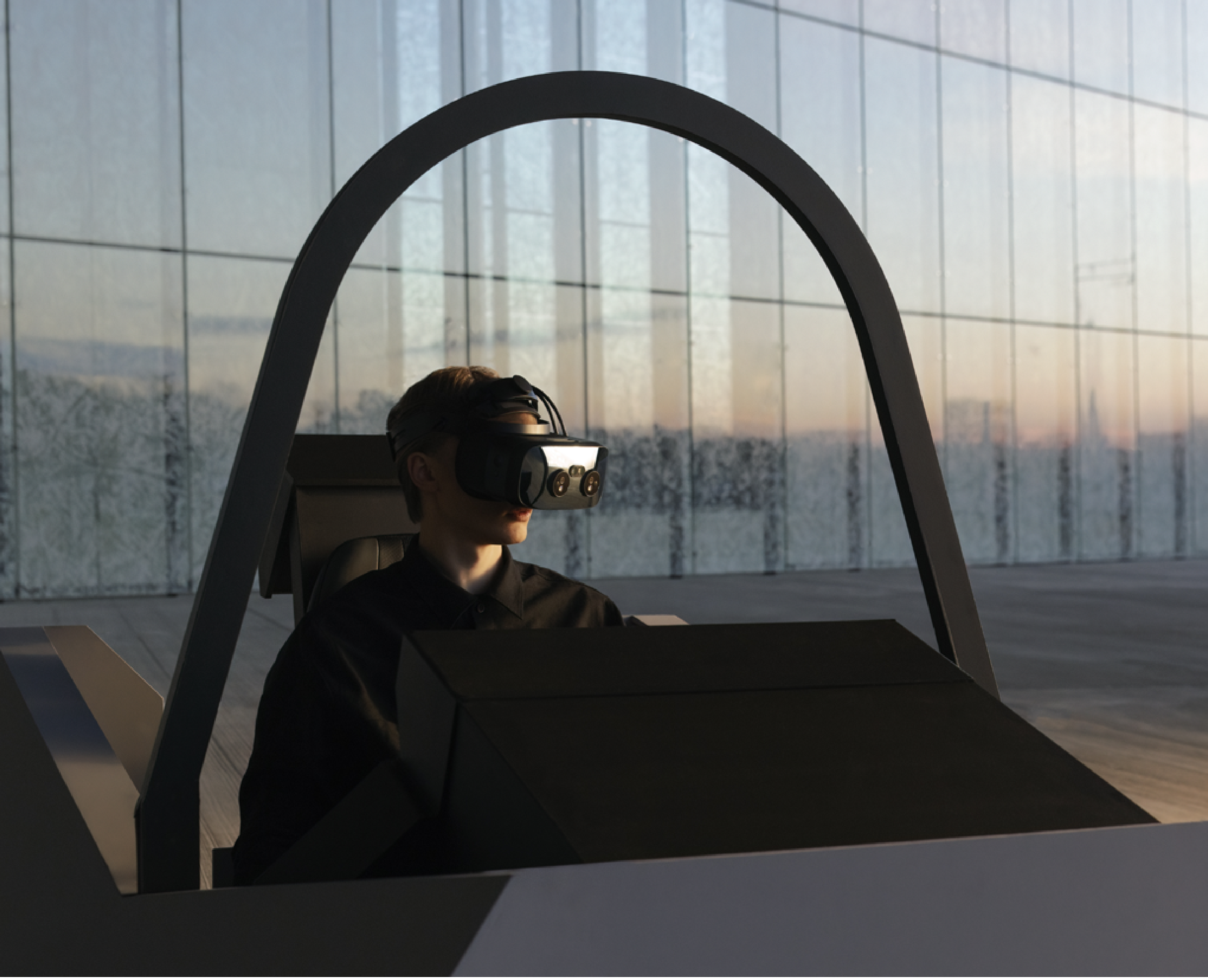
These devices produce high-resolution out-the-window scenes and allow pilots to freely interact with physical cockpits as they do in the real world. Where the degree of realism achieved with these devices has never been higher, the acquisition, maintenance, facility cost, and lack of portability is a growing challenge for those developing training systems.

Even before the pandemic, both the aviation industry and government organizations were under increasing cost and operational tempo pressures. While the world is rapidly returning to a new normal, the challenges facing aviation training have not gone away. As a result, the aviation training community is looking to virtual and mixed reality technologies to help tackle growing pilot shortages and budget challenges.

Virtual reality training, where the pilot is fully immersed in a computer-generated environment, has been found suitable for training basic tasks such as cockpit familiarization, checklist training, and basic flight skills. Higher-fidelity mixed reality (XR) training where the pilot can see and interact with physical hardware in his/her vicinity has been found more suitable for training tasks that require development of muscle memory and where complex switchology and systems are involved. In virtual and mixed reality training, high visual fidelity is critical for pilots to see cockpit displays and objectives clearly.

Mixed reality or XR training can be achieved with advanced head-mounted displays with digital pass-through cameras that seamlessly blend video of the outside world with computer-generated content. Mixed reality simulation enables tactile feedback so pilots can see their hands and feel the stick and throttle, buttons, and switches that populate the crew station while flying in a fully virtual environment. XR training setups, unlike traditional displays, can cut the cost of training devices by half to a full order of magnitude with little loss of fidelity. The physical size of resulting devices is dramatically reduced, saving space as well as electrical and cooling needs.

The lower device and operational costs brought by VR and XR equate to greater availability of training tools that allow trainees to achieve more reps and sets, engaging training that encourages trainees to repeat tasks until they achieve mastery, and all-around better scalability. Thanks to the portability and reduced simulator size of immersive solutions, VR and XR can also be used to complement traditional simulator training in scenarios where travel is not possible. When training organizations successfully implement immersive solutions, they will be able to train more pilots faster, provide more flexibility in the training, and with less cost than ever before.



© Varjo

A series of small-footprint VR simulators can train a large number of junior pilots in a smaller space, with trainees building muscle memory and mastering basics before lining up for their time in the dome. VR systems deliver efficient, immersive, and very deployable training capabilities for a large spectrum of use cases. Some other use cases, while they have a lesser need for deployment, are leveraging more traditional simulator configurations to generate immersion. We believe one is not meant to replace the other, and that in the training continuum different technical solutions are serving different use cases and needs. Large visual projection configurations and high-end projectors are generating a level of innovation that is making them essential hardware of the ecosystem.

Product image courtesy Immersive Display Solutions, Inc.

George Forbes
CEO IDSI

In the world of flight simulation, Unreal Engine is a welcome addition. The current pilot shortage is being exacerbated by the advent of the electric Vertical Take-off and Landing aircraft (eVTOLs) market space. This evolving eVTOL market is driving demand for high-quality, low-cost VR and direct-view simulators. Fortunately, the worldwide community of Unreal Engine developers now allows a wider range of simulator solutions to be developed to suit specific customer needs and budgets.

Modern game engines have quickly matured, benefiting from the massive economy of scale of the $100B+ consumer gaming market. Game engines are now robust enough to become the foundation on which flexible, cost-effective military-grade simulators can be designed and delivered. The rendering engine in a commercial simulator needs to be open and flexible, to seamlessly integrate with Best-in-Class analytics, entity simulations (Computer Generated and /Semi-automated Forces), and physics engines. Through APIs and plug-ins, customers can now leverage a large universe of add-on solutions from developers who specialize in solving specific problems. The benefit is a much higher quality of visuals, such as volumetric clouds and high-fidelity ground details, all at a lower cost.

In the case of direct-view immersive visual systems for simulation, such as domes and cylinders, it is a hard requirement that rendering engines are capable of distributed rendering because the rendering workload must be scaled across a cluster of computers and projectors. Unreal Engine is now capable of distributed rendering and has proven to function very well in these complex simulation environments.

A specific use case for simulators is Joint Terminal Attack Controller (JTAC) training systems. This training curriculum is particularly challenging because it presents “the sniper problem” as a matter of the training curriculum. The challenge for the rendering engine is to manage the physics of a bullet or missile trajectory as it travels across multiple terrain grids across multiple computers while maintaining synchronization with all the entities in the simulation. Advanced networking and fast servers help to overcome this historically difficult use case.

Neil Wittering
Simulation Segment Marketing Manager,
Barco Simulation

From our point of view, the goal of any simulator is to fully replicate real-world scenarios and feedback in a way that allows the trainee to suspend their disbelief that they are not "live". The real-time visual system is one of the most critical elements in creating a fully immersive training environment. Selecting a projector that can match the processing power of the IG frame for frame, pixel for pixel, is an absolute requirement.

Another important factor is how the dynamic visual environment is perceived—a pixel is not simply a pixel: How the projection system renders pixels will determine the quality of the image being viewed by the trainee. A high-quality projection system will have higher rendering capacity, making sure the dynamic environment the pilot sees delivers a real-world experience, independent of movement of the aircraft or other visible objects.

Consider the out-the-window view during an aircraft's banking turn or the acceleration during a takeoff roll, where signage, indicators, and objects on the ground need to be observed. It's essential to reproduce these images at a higher refresh rate so the trainee can experience a seamless image without artifacts and distractions from motion blur.

As the performance of IGs continues to increase, this drives demand for higher performance visuals pushing beyond current standards. To keep pace with these changes, Barco is committed to developing projection and other display technology to meet the needs of higher resolution and higher refresh rates.

If we take one more step in this direction, ultimately a larger set of senses will be stimulated in the simulation domain. Because modern projection and HMDs are the descendants of our TV sets and stereos, the eyes and ears in many virtual experiences have inherited their technology from the great world of AV (audio-visual).

Beyond audio and visual inputs, new use cases are bringing more importance to the kinesthetic cues for augmenting training transfer. Motion-cueing systems have been in use for ages in training systems, but micro-haptic devices, which produce heat variation, impact impressions, and pressure or electrostimulation of our muscles are now also being used to further immerse trainees in the simulation environment.

Advances in these haptic and motion-cueing systems used in other training domains have started to become more prominent in the simulation domain. Between small vibration systems or motion-cueing devices and the massive hexapod able to support dozens of tons, the field of motion cueing, and more recently, the haptic devices trends coming from the healthcare domain, have started to become more prominent. The openness of Unreal Engine's architecture allows for easy integration with these systems, which are bringing immersion to the next level.

Roger Klingler,
CEO, Brunner

Going from good to great training requires a level of realism that fully immerses trainees. In order to do so, the trainees' engagement needs to be stimulated from all of their senses. There are many cues already available in common simulators, but new inventions are needed to enhance the experience for total immersion.

Immersion is a simple word very often used nowadays in combination with VR or MR applications. But how is "total Immersion" possible? How can we reach our goal of total immersion or total realism? Cues which are mainly used in VR and MR applications nowadays are visual. High-fidelity graphics and high-resolution HMDs with brilliant 360° views are achieved and the person feels as though they are diving into a completely different world. Brilliant 3D sound offers another cue and helps to immerse even deeper.

Last is haptic feedback from controls or gloves which give certain feedback to the user, enabling them to dive even deeper into the illusion.

Brunner's expertise in haptic and motion feedback is creating the right level of kinesthetic stimulation to fully immerse pilots and trainees. This system offers a high-dynamic, latency-free 6-DOF motion platform that utilizes advanced motion-cueing algorithms in combination with high-resolution VR/MR simulation software, high-resolution HMDs, and our own high-fidelity control-loading units.

The kinaesthetic cues given by the platform let the pilot feel the aircraft he's sitting in. The feedback given by the aircraft not only immerses the pilot much more, but also trains muscle memory and his feeling for the aircraft's movements and forces.

Realistic 3D sounds, a radio connection to the instructor operator station, a seat shaker that simulates the aircraft's engine, and structure vibrations—all these sensations add up to "total immersion," the feeling that you're piloting a real aircraft.

Image courtesy of Brunner

Image courtesy of Meta Immersive Synthetics

The transformative value of these factors—readily available commercial devices, and the visual realism and fast-paced innovation of these development environments—has been very high and has been one of the driving forces for innovation in flight simulator development. Training can now happen anywhere from the trainees' homes, to in the air, or at a flight training school.

"Bringing the training to the trainees, and not inverse, has been a motto for simulation creators for years," says Sébastien Lozé, Unreal Engine Simulation Business Director at Epic Games. "These technologies, whether integrated with software or hardware, and with the right innovation spirit and following a proper methodology, will not stop surprising us in the coming years. The sky is no longer the limit."

The last piece of the puzzle, one that is accelerating the deployment of simulation solutions across the board, is the adoption of powerful cloud-based solutions as part of the training system infrastructure. With the use of Google Cloud, Amazon Web services, Microsoft Azure, and other cloud application solutions, development teams no longer need to acquire storage media, computer equipment, or even pricey GPUs. Running the simulation from the cloud means training can happen anywhere there is electrical power and a high-bandwidth internet connection.

Having the simulation applications in the cloud also makes the training or analysis solutions easier to maintain, as updates can be pushed to trainers in any location simply by updating the cloud-deployed application.

Christopher Covert
Director, Mixed Reality Government at Microsoft

Digital ecosystems are awesome! Over the last few years alone, we've seen analog systems going digital at a remarkable pace: workplaces are going remote, organizations are leveraging simulation/IoT to generate massive data-driven ontologies, and cloud-centric operations are stronger than ever. All of this change has helped platform companies like Microsoft innovate with confidence through the development of cross-functional, extensible product offerings, and as this digital transformation drives the accessibility of our products, we feel the empowerment of an incredibly strong creator community in return.

Of all the different ways Microsoft supports the simulation industry, none is more closely tied to our DNA than flight simulation: three years before the first release of the Windows OS, we released Microsoft Flight Simulator on MS-DOS and Classic Mac OS. That was almost 40 years ago, and we've continued to set the bar high with the new FlightSim released in 2020. For the gamers out there, Microsoft and our awesome partners have created a living, breathing world that contains over 37 thousand airports, 1.5 billion buildings, and over 2 trillion other assets like trees, roads, and rivers all procedurally generated from 3.5 petabytes of Bing Maps and partner data.

And for the researchers out there, we have AirSim—a simulator for unmanned aircraft systems (UAS) and autonomous ground vehicles—built as an **Unreal Engine** plugin. Since it's an open-source, cross-platform project, anyone can clone it straight from GitHub and run their own software/hardware-in-the-loop simulations with popular off-the-shelf flight controllers, ideal for deep learning, computer vision, and reinforcement learning algorithmic training.

But this industry is way more than just the application that the pilot interfaces with, so we are actively supporting the accessibility of these experiences with Epic Games through new innovations like **Pixel Streaming** on Azure, which allows developers to host Unreal Engine apps on the cloud and stream rendered frames/audio down from cloud GPUs directly to a desktop/mobile web browser without the need for users to have a state-of-the-art GPU on device. Also, to bring the best of Unreal Engine to the world of mixed reality, we have the plugins, code samples, and documentation like the Mixed Reality Toolkit (MRTK) for Unreal that help developers turn their Unreal Engine environments into the perfect mixed-reality applications.

And we are not alone. We are all collectively working to arm simulation developers/users with the right tools to drive deeper insights and create unparalleled experiences. To do this, we are driving incredible advancements in:

- Ingesting and computing massive amounts of data at the hyperscale
- Handling sensitive, secure, (and in some cases classified) data
- Using AI/ML to streamline user experiences and make quick decisions
- Creating, simulating, and visualizing 3D worlds
- Fostering collaboration between users
- And generating insights from real-time and post-event analyses

This simulation industry—one that is built on the decades of investment and advancements of global infrastructure, data-driven world-building, artificial intelligence, and immersive 3D experiences—is certainly one that I am excited and proud to be a part of today.

Image courtesy of Meta Immersive Synthetics

Realism and training accuracy

The flight simulation domain is probably one of the most demanding and challenging environments for real-time 3D. In years past, game engine-based solutions, while they could provide unmatched quality of photoreal imagery, lacked too much on the scalability and accuracy front to be adopted by simulatorists.

In order to generate beautiful images in real time, games engines have, in the past, provided only limited precision with regard to the calculation of several elements in their core mathematical models. While this limitation didn't cause insurmountable problems for discrete, relatively small environments, any small, insignificant errors in geolocation would multiply exponentially as distances and speed increased. This made the creation of aerospace training applications using game engines challenging, as such applications require precision over hundreds or thousands of virtual miles.

This problem goes back to the fact that game engines have traditionally been limited to 32-bit calculations. While various means can be employed to correct geolocation errors on the fly, these corrections can also be faulty.

We are pleased to say that Unreal Engine supports 64-bit calculations in its latest release, Unreal Engine 5. The integration of 64-bit precision means that the engine is now suitable for simulations that take place over long distances, retaining accuracy through thousands of miles of virtual travel without the need for error-correcting methods. Accuracy over long distances implies taking earth curvature into consideration, and ellipsoidal datums like WGS84 are fully supported through a native georeferencing plugin.

Nick Giannias
Director, Advanced Technology & Innovation and Chief Architect at CAE

CAE, as the global simulation leader, has been creating synthetic immersive worlds for decades. In the early years, we leveraged computers and operating systems and essentially created a stack of custom-made software and custom-build electronic hardware to bridge the real and synthetic worlds. That technology yielded fantastic innovations which changed, and improved, the world we live in. The level D full-flight simulator is but one example of this history of innovation contributing to the safe growth of aviation in our skies.

Across the last 40 years of simulation industry history, we have witnessed several disruptive technological tipping points. Six-degree-of-freedom electrical motion systems displaced hydraulic actuators, generating a substantial reduction in operating power requirements. Commercial GPU cards and advanced graphics drivers displaced the large, printed custom-designed and built circuit boards in image generators. New projector technology based on LCD, LCOS, and lasers displaced the large and heavy cathode ray tube projectors, simultaneously shrinking the size of the visual display solution and increasing both resolution and operating life.

Game engines have been with us for over 20 years. Essentially, they are a software toolbox for game creators to ease the work in producing the final product—a game.

They provide common services and functionalities which are needed by game developers. These range from low level functions to communication and interface services, image rendering services, and all the way up to landscaping and 3D model editing, including topography and behaviours such as physics and AI.

The simulation industry has a similar software stack providing similar capabilities, but with a focus on real-world environments and accurate physics. Our earth is round, and force always equals mass times acceleration. Our goal all along has been to provide a faithful and verifiable representation of reality in these synthetic worlds. Nevertheless, our industry has been watching, with envy, the rapid technological progress of games and game engines. The quality and sophistication of image rendering, the ease and speed in creating complete scenarios, the out-of-box delivery to a wide range of media—from mobile devices, to AR/VR headsets, to high resolution projectors—were all significant achievements.

Approximately five years ago, we began to use game engines to construct and deliver real-world training solutions. Aircraft maintenance trainers, walkaround familiarization trainers, and medical procedure trainers were amongst the first successes. The massive investments in game engines driven by the ever-growing global gaming market had obvious benefits in terms of speed, efficiency, and quality. A technology tipping point had occurred. And there was no turning back.

In the last two years, new capabilities in game engines have further disrupted the state of the simulation industry. Cloud deployment and scalability options opened the doors for large-scale distributed simulation supporting multiple interfaces. Open, flexible APIs and even source code availability has made it possible for us to add our own pieces as part of the solution, only reusing necessary parts from the game engine. The large marketplace of contributors in the game development community, continuously creating assets which can be leveraged by any application, yields incredible synergies.

We believe the simulation industry is going through another technology tipping point where game engines and their supporting ecosystems will displace some of the technologies and processes we have built over the years.

CAE Prodigy IG leverages Unreal Engine at its core.

ENTROL

Enrique Silvela
Visual Systems lead engineer, Entrol

The development of game engine technology has far surpassed traditional methods used in simulation, both in performance and realism. We soon realized that developing a flight simulation IG is quite similar to developing a video game. This is not a novel idea; some of the most popular commercial IG are (or started as) video games, such as Xplane and Prepar3D. While the simulation industry has viewed game engines as a real possibility for small, detailed databases, it was disregarded for flight simulation due to, primarily, database extension.

It is true that it was hard in the past to create very large worlds in commercial game engines, due to technical (32-bit position) and production (lack of database generation support) limitations. However, this is not true for modern game engines—most notably, Unreal Engine has made the leap into simulation territory with Unreal Engine 5, boasting 64-bit precision support with its Large World Coordinates and the new World Partition tool for very large databases.

One of the biggest advantages of developing our own IG is the freedom we have to pursue new ideas and features for our customers. Previously, if we wanted a new feature (or even a bug fix) we would have to wait months for our IG provider to implement it. Open standards means we can look directly at the source code to solve bugs and develop new features faster than ever.

The ability to see changes to a database in real time is a key feature of game engines that greatly increases production speed and reliability. Being able to add a new model to a database and perform all the necessary quality checks, without having to recompile the whole database, has increased our team's performance substantially. All of this enables us to customize our databases to each customer's training objectives.

Many of our customers are operators that perform mission operations such as Search and Rescue (SAR), Emergency Medical Services (EMS), or Fire Fighting, which are complex and dangerous. They need to recreate the most demanding situations that pilots and other crew members can encounter during these operations: engine failures, wild weather conditions, and so on.

Simulator manufacturers must develop databases that replicate all these conditions. That way, all the crew members can practice emergency procedures and train on communication and phraseology, which is a critical part of the mission.

Image courtesy of Entrol

The evolution of flight simulators has been a collective effort, and we look forward to more collaboration with the industry. While it is important to develop applications following appropriate verification, validation, and accreditation processes, we can accomplish these goals while also bringing a high level of innovation, if we work together.

Niclas Colliander
Managing Director of Meta Immersive Synthetics

It will not be the first, or the last, time we go through these transitions. Our culture of innovation, and our quest towards bigger, better, and faster, has always driven our industry.

Traditionally, when looking at the use of game engines for military simulation and training, it is important to understand that game engines were not originally designed from the ground up to support all things needed in these use cases.

As an example, military scenarios often involve kinetic, optic, or electromagnetic interaction over significant distances, often on the order of tens, hundreds, or even thousands of kilometers. This requires accurate 64-bit physics as well as proper code supporting electromagnetic propagation, transmission, and reception and kinetic interactions between different materials at different speeds.

Classically, simulators have solved this by separating the simulation from the visualization. Hence, the term "image generator" was born.

In the game world, visualization and other parts of the game such as physics have grown closer and closer and today most game engines cater to physics, rendering (visualization), animation,and other things needed to interact with the game. There are many benefits to combining accurate physics and visuals in the same platform. If you have a good understanding of the physical world, you can make better visualizations. Conversely, if you have an accurate visual system, sensor modeling can be based on the same data and technique as the rendering pipeline, saving performance and increasing quality.

By leveraging Unreal Engine, MIS with its NOR-platform can take what is already world-class in terms of rendering technology and create the right applicative layer to augment it with simulation systems. This creates a platform that is as capable as large, expensive, and dedicated simulators in terms of accuracy and precision while also being able to produce life-like visuals on par with the latest games.

The need for accuracy and precision in the aerospace simulation space have typically been isolated to mean accurate physics and systems modeling. However, if photorealistic visuals are also introduced, many more use cases become viable to train synthetically.

An example use case that has typically been avoided in simulation is Close Air Support (CAS). This is a complex mission type for military pilots that includes cooperating closely with ground forces in order to help them achieve objectives on the ground. Since most flight simulators have been very limited in terms of ground visuals, effects, and other environmental factors, this has typically been deemed a scenario not worth training for in the simulator. With NOR, we can provide high-fidelity flight simulation while also providing a rich world down at eye level, complete with dust, weather, sound, and other factors that impact the mission.

Adding new mission types like CAS put us on a path where the line between different types of simulators start to blur, and eventually will disappear. All mission types should be available in the same software for networked training. From our perspective, there is no reason a modern force should not have access to a fully immersive combat cloud where units can simply log on to train as they fight, at any time.

Image courtesy of Meta Immersive Synthetics

We see established stakeholders in the simulation community, as well as new entrants, stimulating the pace of innovation in flight training. More and more, we see organizations moving away from closed, proprietary systems and toward a more collaborative and open approach.

This very dynamic environment is fostering the development of flight simulation solutions for training centers as well as bringing training to the trainees at the point of need; to their homes, their offices, and even in some cases bringing training to the pilots, in their cockpits, during flight.

One example is the **in-flight aviation training system developed by Red6**. While in flight, the pilot is able to see the actual environment while one or more computer-generated enemy aircraft are projected onto the AR goggles in stereoscopic 3D. Pilots can run the same scenario multiple times to hone their skills, without the cost and risk of sending up other pilots to pose as adversaries.

Derek Cedarbaum
Product Strategy, Red 6

Training pilots is expensive. Getting a pilot from day one of pilot school to fully ready in the aircraft they might fly their entire career can cost anywhere from $3 million to $11 million. In addition, pilots must train for several dozen hours per month just to maintain the skills they have worked so hard to attain.

Flying newer fifth-generation aircraft such as the F-22 can cost upwards of 500% more per flight hour than older aircraft such as the F-16. If you are an F-22 pilot spending hundreds of hours per year in constant training to the tune of more than $60,000 per flight hour, the tax dollars add up!

To further complicate matters, the way most pilots train today does not fully represent what they will see in the battlefield should they ever have to see the day. Between this and **too few actual flight hours**, pilots are facing something known as "negative training."

During flight training, one pilot usually flies with or against one or more aircraft to simulate an actual scenario they may one day face. However, an F-22 flying a combat scenario against an F-16 does not represent the kind of environment they would be flying in if faced with a near-peer adversary's fourth- or fifth-generation aircraft in actual combat.

For the fifth-gen F-22, more complex training requires the pilot to face off against 10 enemy aircraft at once, as this is one of this aircraft's design intents. However, this has not been possible to consistently execute due to the inherent complexities of planning and flying so many jets in a constrained airspace. This means that our pilots are not able to train against threats they may face in the future.

Thankfully, these limitations in pilot training are becoming a thing of the past due in large part to advancements in augmented and virtual reality hardware technology and a variety of game-based software solutions.

Commercial game engines such as **Epic's Unreal Engine**, coupled with military networks for enabling multiplayer environments have made it possible for an entire metaverse to digitally materialize 30,000+ feet above our heads.

Now, a pilot can get into his or her jet wearing an augmented reality headset and fly scenarios that they may encounter one day. This means complete, within-visual-range training against modern threats and adversaries.

Further, with advancements in high-speed networking, a ground-based VR simulator 100 miles away can connect to the aircraft in the sky and pilot an aircraft that appears in the pilot's augmented reality headset.

This new capability means more of the right kind of training while saving potentially billions of taxpayer dollars.

Most importantly, perhaps, is how networked augmented reality will act as the new user interface to the digital universe. Humans have difficulty processing massive amounts of data, thus the rise of big data analytics platforms. However, to shorten the time to and improve the quality of decision-making, data must be more intuitively presented to humans.

Image courtesy of Red6

Being able to overlay the right data at the right time in 3D, integrated with the physical world, will greatly reduce the working memory requirements in high-stress situations (such as piloting an F-22 at 1.5 Mach during combat missions). With improvements in understanding this 4-dimensional digital UX/UI, driven by high-speed 5G networks, Low-Earth Orbit satellite networks such as Starlink, and AI data platforms such as Palantir's, we can now design the tools necessary to improve the decision-making capacity of any human on earth.

When people ask what the metaverse means, perhaps the epiphany of this technological revolution will be that the metaverse improves the cognitive capacity of humans. It will be the interface layer between the physical world, and the cumulative combination and use of all of the digital tools we've been developing since the invention of the transistor.

The Unreal Engine simulation ecosystem

While game engine technologies are solving lots of the problems at the core of the development process, they are just one of the elementary bricks to creating next-generation solutions. We believe that there are several technologies and approaches that can coexist and carry us collectively toward excellence in the flight simulation domain. The efficiency of a simulator is not coming from the tools used to build it, but from the methodology used to create it.

This section of the document focuses on the ecosystem of simulation-focused modules built for the Unreal Engine by our ever-expanding community of users. These modules help simplify the process of building reliable, performant, and engaging simulation solutions for a myriad of training requirements. We will explore content creation, flight dynamics, and deployment as three primary components of building any simulation environment.

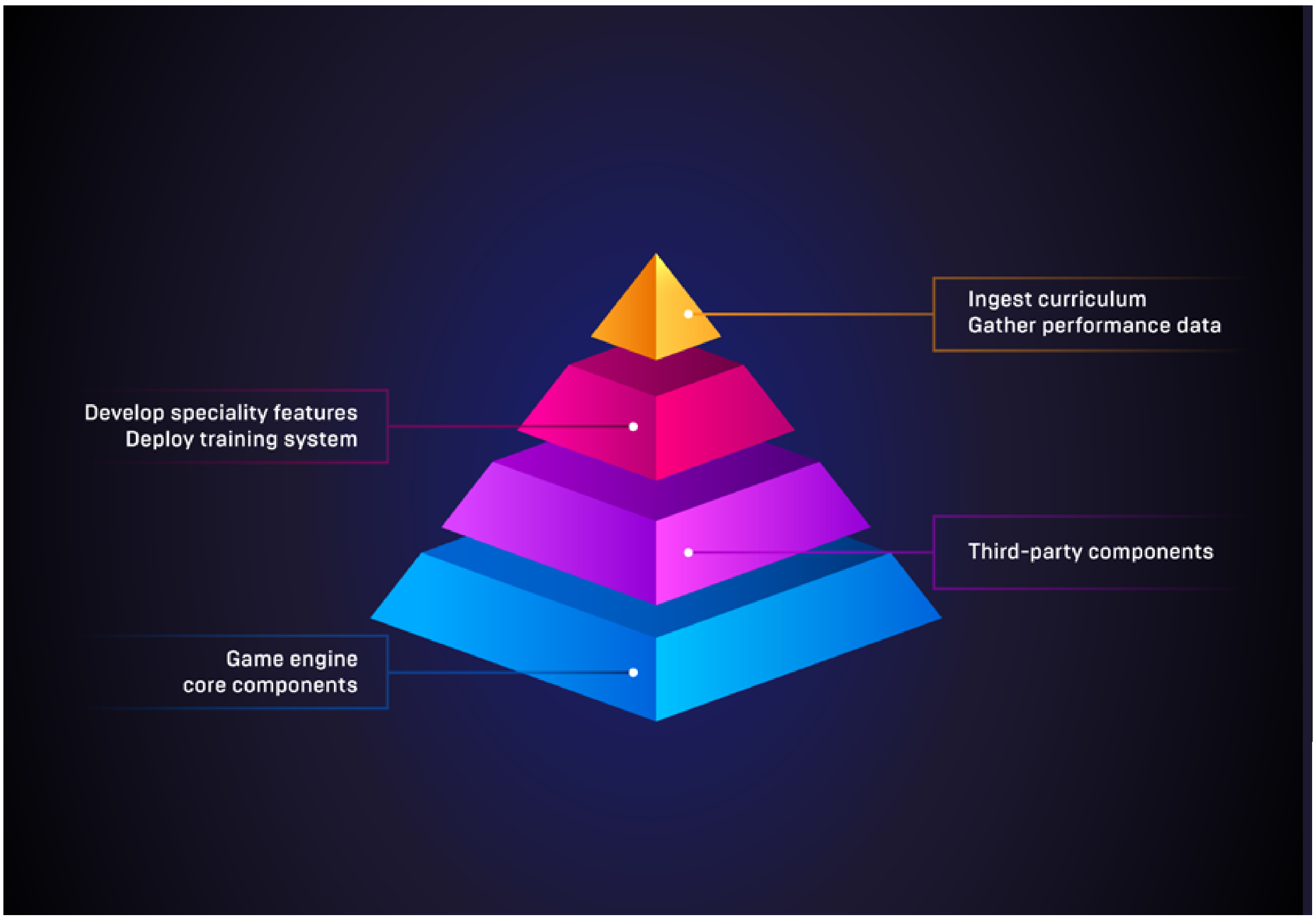


Figure 1: Overview of key components for simulator development

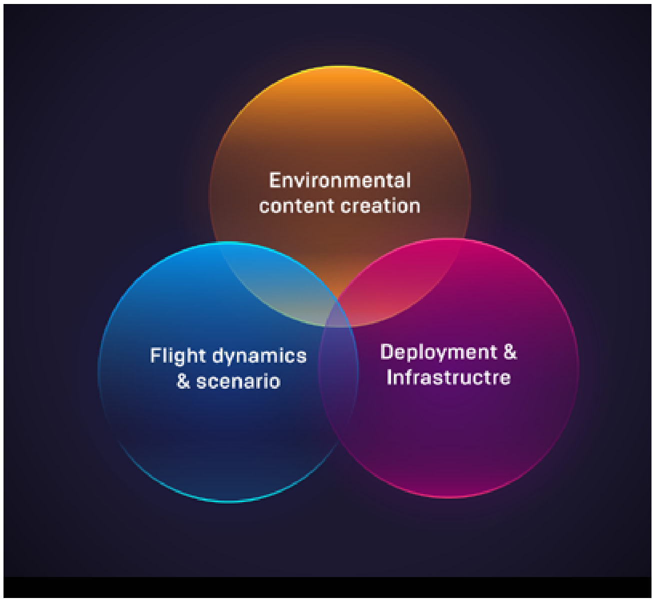


Figure 2: The three phases of flight simulator development with Unreal Engine

Environmental content creation

The initial step in developing any simulation system is to construct or obtain the 3D environment within which your simulation takes place and to populate the environment with other 3D content. This could include buildings, airports, roadways, vegetation, and other objects that make up the world.

As you think about pipelines in the simulation domain, some key beats from the ecosystem around Unreal Engine come to mind: 3D Tiles integration, ArcGIS integration, and the CDB plugin, among others. It is easy to also establish a natural link with the work done by our Epic Games colleagues in the photogrammetry pipeline optimizations of Quixel and Reality Capture—they are creating a simple way to inject content into your simulation applications by leveraging new techniques from aerial-captured videos to **handheld-device photography**.

World Partition is providing a customizable mechanism to simplify the creation of massive environments in your application and establish persistent worlds.

*"World Partition is an automatic data management and streaming system used both in the Editor and at runtime, which completely removes the need to manually divide the world into countless sublevels to manage streaming and reduce data contention. Using World Partition, the world exists as a single persistent level. In Editor, the world is split using a 2D grid and data is partially loaded based on your area of interest using the World Partition editor window. This makes it possible to handle massive worlds that would otherwise not fit in memory or take a long time to load. When cooking or running in the Editor, the world is divided into grid cells optimized for runtime streaming, which become individual streaming levels." Find all the details on the **World Partition** page in the Unreal Engine documentation.*

Lighting is great in Unreal Engine 5 thanks to Lumen. But this global illumination solution is only as good as the geometry you leverage in Unreal Engine. Lumen works in conjunction with the engine's new virtualized geometry system, Nanite. Nanite is known already by many of the Early Access adopters as one of the biggest game changers introduced in Unreal Engine 5.

Nanite not only helps you get better terrain and environment detail into your environment, but it also enables you to load the most detailed engineering-level data for devices, vehicles, or buildings into Unreal Engine without forcing you to degrade your 3D sources.

And what is coming is even more exciting, as we start to see AI solutions augmenting captured geographic data, such as that developed by Blackshark.ai. Exciting and essential too, is the work done in the community by large-world open standards like 3D Tiles Next. The 3D Tiles Next open standard not only contains geometric and geographical data for your environment but also provides attribution to this geometry, enabling the simulation to interact more intelligently with its environment.

Recently you may have read articles about Megascan trees, volumetric clouds, or water capabilities. Implementing these elements will become easier with Unreal Engine 5. For example, your Megascan assets will be just one click away from you as the Quixel Bridge gets integrated into the Unreal Editor. Virtual Shadow Maps, Lighting, Temporal Super Resolution, or Animations also bring you to the right level of realism without having to be an expert animation specialist.

All the options that exist for content creation within the Unreal Engine simulation ecosystem provide users with a set of tools that drastically simplify what was once a time-consuming and often expensive process.

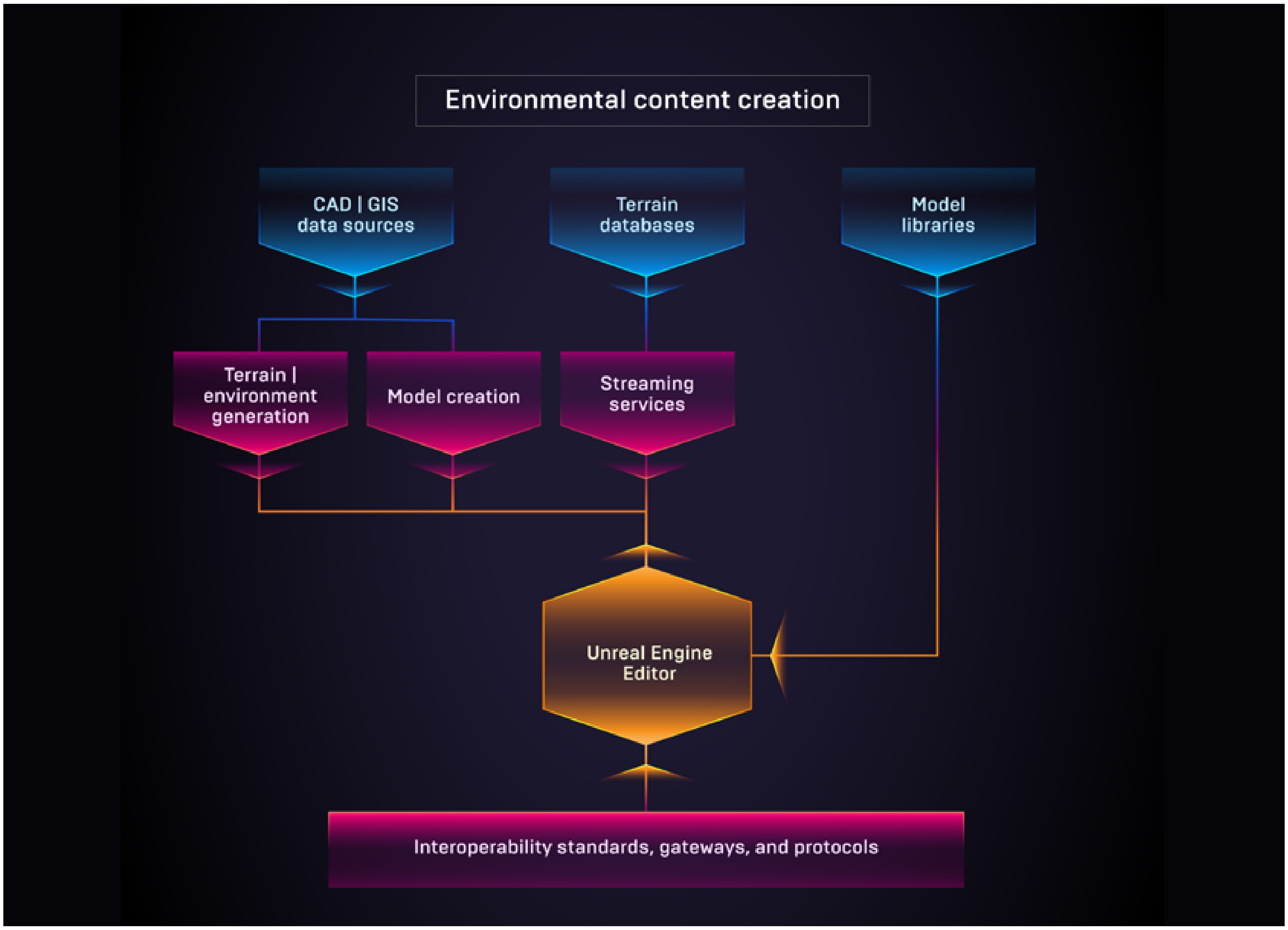


Figure 3: Workflow for environmental content creation

Terrain/environment creation

Terrain formats and data streaming providers
Openflight format - **Presagis: Openflight to Unreal**, **Simblocks.io: OpenFlight Importer SDK for Unreal**
CDB format - **Simblocks.io: CDB Datasmith Exporter for Unreal**, and **CDB Runtime Publisher SDK for Unreal**, **Presagis: VELOCITY 5D**
3D Tiles format - **Cesium: Cesium for Unreal**
ESRI format - **ESRI: ArcGIS Maps SDK for Unreal Engine**
Supermap: SuperMap Scene SDKs for Unreal Engine
VBS Terrain - BiSim: VBS World Server for Unreal
Aerometrex : Photogrammetry models

Large open world creation
UE: Open World Tools
UE: Nanite
TrianGraphics: Trian3DBuilder
Presagis: Terra Vista
Mathworks: Road Runner
Terraform Pro

Models creation

Models libraries
Quixel: Megascans
UE: Marketplace
Vigilante
Simthetiq

Tree Libraries
Quixel: MegaScans trees
Speedtree for UE4

Character Libraries
UE: MetaHumans
Reallusion iClone for Unreal
Autodesk Character Generator

Models Creation
UE: Unreal model editing tools
UE: BIM/CAD Support, Import tools (with FBX / Datasmith)
3DS Max
Maya
Blender

Loading Pipeline / Optimization
UE: Datasmith / DataPrep pipeline (LODs, jacketing, merging, HLODs)

© Varjo

Flight dynamics and scenario

A classic misconception about game engines is that they are only for creating beautiful images. It is often linked to the fact that their “scene graphs” are the visible part of the iceberg. Chaos Physics is a highly customizable, lightweight physics simulation solution available in Unreal Engine 5, built from the ground up to meet the needs of next-generation games.

Large World Coordinates (LWC) introduces support for double-precision data variant types in Unreal Engine 5 where extensive changes are being implemented across all engine systems to improve floating-point precision. These systems include those set up for architectural visualization, simulation, rendering (Niagara and HLSL code), and projects with massive world scales. In Unreal Engine 4, 32-bit float precision types would restrict the size of the world. LWC vastly improves the size of your projects by providing a 64-bit double to your core data types. These new changes will enable you to build massive worlds and greatly improve Actor placement accuracy and orientation precision.

Once your 3D environment is constructed, you will need to focus on the entities that will populate your environment and how they will behave and interact with the world around them. For a flight simulation, the accurate modeling of flight dynamics is crucial to ensure realistic representation of the aircraft.

Image courtesy of Epic Games

JSBSIM

Bertrand Coconnier
Main Developer for JSBSIM

"Open source and publicly available source codes are enabling the flight simulation community to collectively improve the way we train and the way we play. In this ecosystem, people from all sorts of backgrounds are collaborating to improve the software and are bringing flight simulation to new levels of immersion and reality.

As an open source flight dynamics model, JSBSim is no exception and has benefited from the contribution of hobbyists, academics, and professionals in flight dynamics to bring a rigorous development framework that allows developers to focus on vehicle datasets rather than reimplementing flight dynamics equations. Flight dynamics is essentially the physics and math that define the movement of an aircraft, rocket, helicopter, etc., under the forces and moments applied to it using the various control mechanisms and from the forces of nature. JSBSim also uses the WGS84 rotating Earth model and the International Standard Atmosphere to provide a high-fidelity environment to airspace vehicles.

For more than 20 years, JSBSim has evolved from being the only flight dynamics model of the FlightGear flight simulator to being a library that has multiple applications such as being used in flight simulation, helping develop UAV autopilots, or training neural networks to pilot an airplane. JSBSim has recently been used in a virtual air combat simulation where one of the AIs went undefeated in five rounds of mock air combat against an Air Force fighter. It has also been compared by NASA against six other flight dynamics software and JSBSim was found to be in agreement with the other software libraries.

Basically, the library propagates the simulated state of an aircraft given inputs provided via a script or issued from a larger simulation application. The inputs can be processed through arbitrary flight control laws, with the outputs generated being used to control the aircraft. Aerodynamics, aircraft control, and other systems, engines, etc. are all defined in various files in a codified XML format. As a design goal, JSBSim has attempted to find a balance between high simulation fidelity and data file simplicity, so the task of simulating the flight of any aerospace vehicle can be done with the minimum specific input possible. JSBSim is versatile enough to simulate anything from a simple falling ball to a detailed model of the Space Shuttle.

JSBSim is written in C++ and can be interfaced with MATLAB/Simulink or used as a Python module. Our MATLAB/Simulink interface is used in the industry and by academics to develop flight control systems while our Python module has brought much interest from the machine learning (ML) community as a number of leading edge ML software are available in Python. As a C++ library, JSBSim can be interfaced with virtually any software using any language.

With the addition of the Unreal Engine plugin created initially by Epic Games, we are happy to see a new community being able to leverage JSBSim for the next generation of flight simulators. I am looking forward to seeing what the open source community will do with it and where it will take this plugin moving forward."

Image courtesy of Epic Games

In addition to the primary aircraft, a simulation will frequently be populated with additional entities for relevant realism for the pilot. These include other aircraft, ground vehicles, and potentially pedestrians.

In Unreal Engine 5, users can now run the physics simulation in its own separate physics thread as opposed to running on a game thread.

You can use this feature to fine-tune your simulations to always behave in a predictable way. This behavior also serves as the foundation for networked physics, as it allows the server and its clients to tick physics at the same rate, making it easier to synchronize the results.

Flight simulation is about airflow movement, trajectories, and speed. The equations at play here have one goal: to make sure we can accurately describe and assess where you are and where you are going. All aviators have maps, all flight simulation creators have synthetic environments.

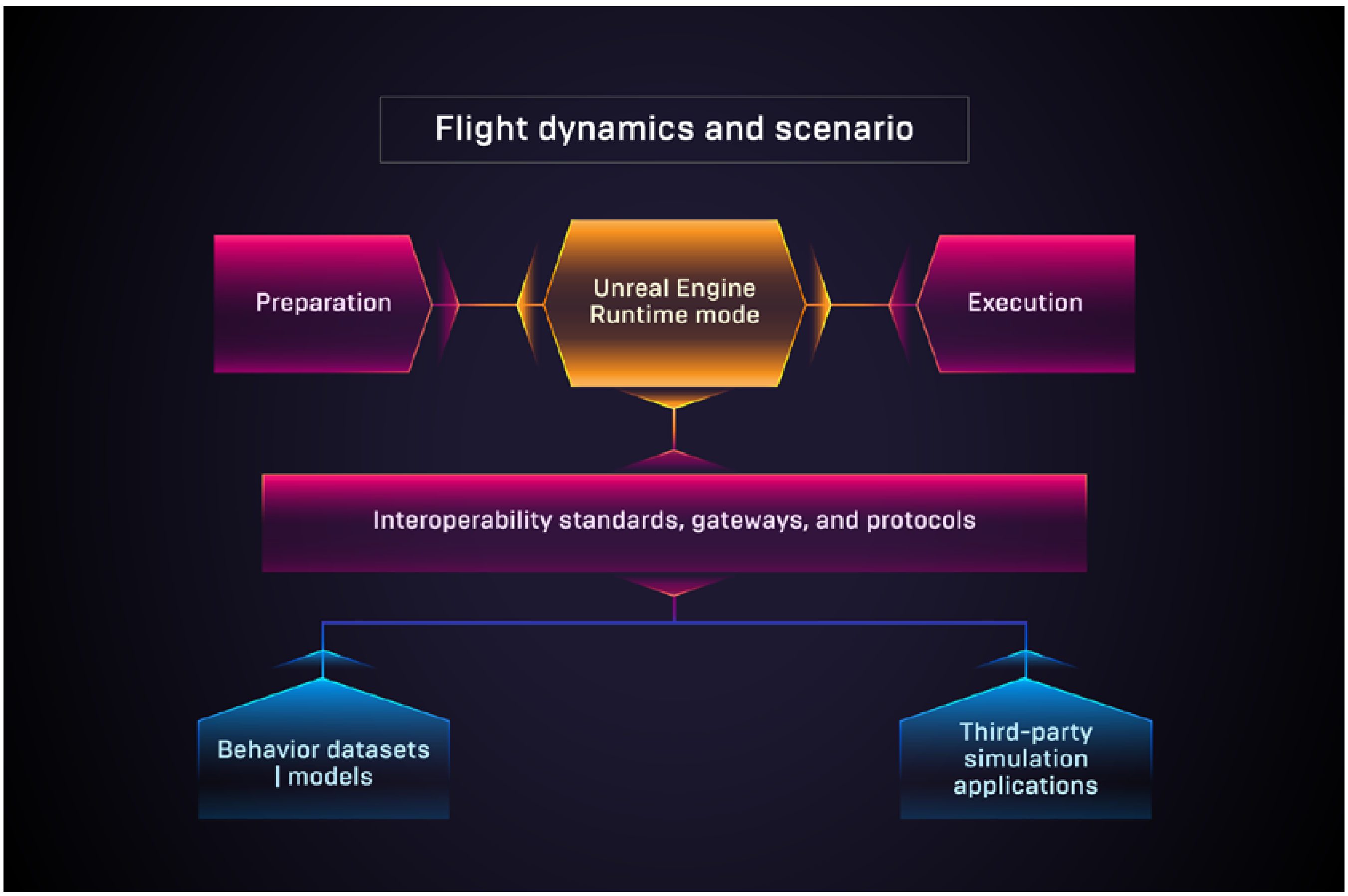


Figure 4: Components of flight dynamics and scenario development

Preparation

Scenario edition
UE: Tools and Editors

Initial conditions (weather, environment, time of day)
UE: Lumen
UE : SkyAtmosphere, Sunsky, Volumetric Clouds, Niagara Particles
UE: Water System
Simul: trueSKY for Unreal

Execution

Simulated objects' behaviors / artificial Intelligence
UE: Behavior Trees (Blackboard, state machines, tasks, environment queries)
UE: Mass Entity
UE: Navigation System (Pathfinding)
UE: Smart Objects
UE: Environment Query System
UE: AI Perception
Github Plugin: xAPI

Trainee scoring
After-action review
UE: Recording and playback
Asynchronous simulation
Data export
LMS integration

Simulated objects' physics
UE: Chaos physics (Rigid bodies constraints, destruction, cloth, vehicles)
UE: Full Body IK
UE: Projectile movement
And others plugins in the Marketplace

Large open worlds
UE: Double-precision data
UE: Geo coordinates (WGS84, ECEF, UTM and others)

Interoperability

Interoperability
Pitch Technologies: Pitch Unreal Engine Connector
ds.tools: coreDS Unreal
RealTime Innovation (RTI): DDS
MAK: Legion (40K live Unreal Engine entities demonstrated to CFT in 2020)

Co-simulation

Co-simulation
CM Labs: Vortex Studio
Algoryx: AGX Dynamics for Unreal
Mechanical Simulation: CarSim
Mathworks: Simulink
JSBSim

Behavior datasets / models

Behavior datasets / models
UE: Mass AI
UE: Mass Entity
UE: Mass Crowds

Deployment and infrastructure

Simulator manufacturers have always considered the ROI when designing a training system for a particular curriculum. Technology just for the sake of technology is a waste of resources if not aligned to a specific training need. Therefore, different scales of training simulators exist simultaneously: Small desktop trainers, with simple devices like a keyboard and mouse, or actual HMI replicas; trainers leveraging XR devices; vehicle cabins or cockpits, whether stationary, or with dynamic motion like that found in full-scale flight simulators. Each of these solutions have their own pros and cons, and are selected with the best efficiency/price ratio.

Because of the target system hardware's diversity, a single training application was often tailored for a specific platform. The lack of multi-system support in the core simulation components limited the reusability over platforms of different scales.

Conversely, game engines have been designed with a variety of potential target platforms in mind: consoles, computers, XR devices, and mobile devices. All of them have different hardware specifications, and assets have to be tailored to their capabilities for the experience to run smoothly.

To help that adaptation, Unreal Engine comes with an integrated scalability system that automatically takes care of optimizing the assets for each target platform.

Such technology makes it possible to adopt a "create once, deploy many" strategy to create different scales of training experiences from a single source.

Whatever the retained scale, giving students and trainees the most valuable training experience possible is at the core of the simulation applications we build. The stimulation of specific senses, with cues, triggers, and responses that trainees are likely to encounter in a real situation, are a significant part of "training transfer."

Unreal Engine includes tools for incorporating audio, visual, and kinesthetic cues to help you achieve a "you are there" training experience for your trainees.

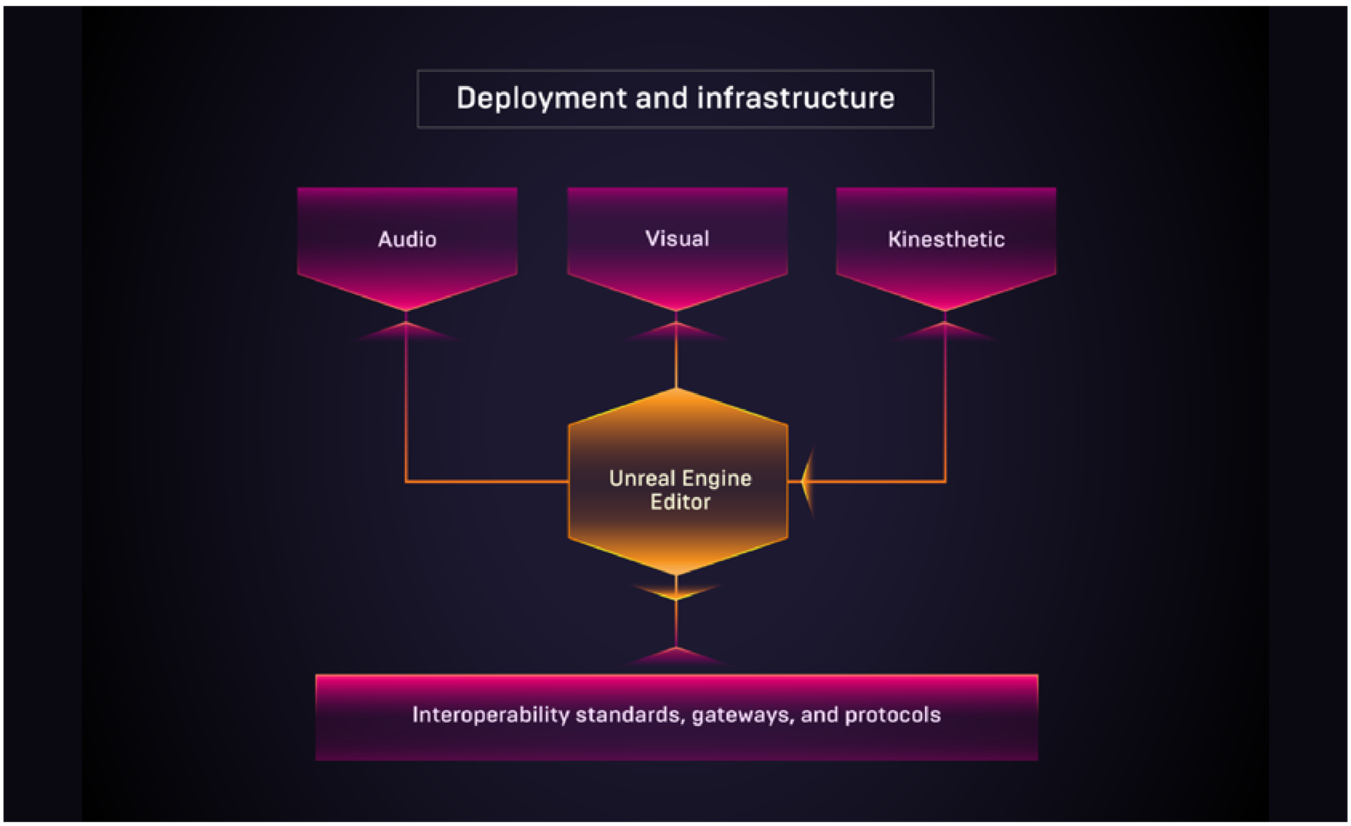


Figure 5: Workflow for sensory tools

Infrastructure

Networking
UE: Networking and Multiplayer
Epic Online Services
Hadean: Aether Engine
Improbable: Spatial OS

Hosting
Supported platforms: Windows, Linux, MacOS, iOS, Android, AR, VR + Gaming devices and embedded systems

Cloud hosting
UE: Cloud-based solutions (Amazon, Azure, Google Cloud, remotely or on-premise)
UE: Pixel Streaming
NVIDIA: Cloud XR
Unreal Containers by Tensor Works

Rendering

Rendering quality
UE: Highly Realistic Materials system, with **PBR Materials**
UE: Global Illumination (Direct and Indirect Lighting, Baked or Dynamic)
UE: Lighting and Shadowing
UE: **Forward** or **Deferred** rendering

Performance
UE: Integrated LOD / HLOD system
UE: Performance and profiling tools
UE: Multi-threaded C++ Core
UE: Specific tools for large environments - **Nanite, Lumen, Large open worlds**

Animations
UE: State-of-art animation system (Skeletal meshes with LODs, vertex animation, retargeting, animation blending, full body IK)

Modes
UE: OTW, simple sensor modes through **Postprocess effects**
Presagis: Ondulus for Unreal
JRM Technologies
Cornerstone

Display devices

Domes and caves
UE: nDisplay (MPCDI, EasyBlend -ScalableDisplay-, VIOSO, and dome projection)

HMDs
UE: Native support for OpenXR, Oculus VR, SteamVR, Google VR, HoloLens 2, Magic Leap, Windows Mixed Reality Toolkit, ARKit, ARCore, Varjo, and custom

Display
UE: Pixel Streaming (Web browsers)
UE: High Dynamic Range Display

Haptics and user interfaces

Haptic / Kinesthetic devices
UE: Universal MotionCueingInterface on GitHub
3D Systems: TouchX and Phantom
TeslaSuit
Leapmotion: UltraLeap
Bruner: Motion platforms, NOVASIM VR

Input devices
UE: Extensible input system (Keyboard, mouse, touch, gamepads and Joysticks)
UE: Generic HID devices
UE: Force feedback

User interfaces
UE: Unreal Motion Graphics UI Designer (UMG)
UE: Slate (enables users to customize the editor)

Audio

Sound/chat
UE: Sound Cues
UE: MetaSound
UE: Dialogs
UE: Epic Online Services (in-game voice chat)

Image courtesy of Epic Games

Implementation examples

A DIY flight simulator tutorial

From this paper, and from the Antoinette Project itself, you have learned about capabilities and best practices when building a simulator with a real-time engine. You may naturally consider that while this information is valuable, you would like to know the specific steps that you can take to create your own simulator with Unreal Engine.

We have created a tutorial to illustrate how simple and fast it is to assemble the various building blocks of a basic flight simulator. You can find the **full tutorial on the Creator Hub**.

The tutorial will guide you through the process of developing a flight simulator using freely available assets and plugins for Unreal Engine.

The tutorial will provide detailed information on how to:

- Connect various input control devices for your pilot interface such as keyboard and mouse, GamePad, joystick, or specific flight-control device
- Integrate an aircraft model taken from the Unreal Marketplace
- Add an open-source flight dynamics plugin from JSBSim
- Simulate flying above world data using Cesium's Unreal Plugin

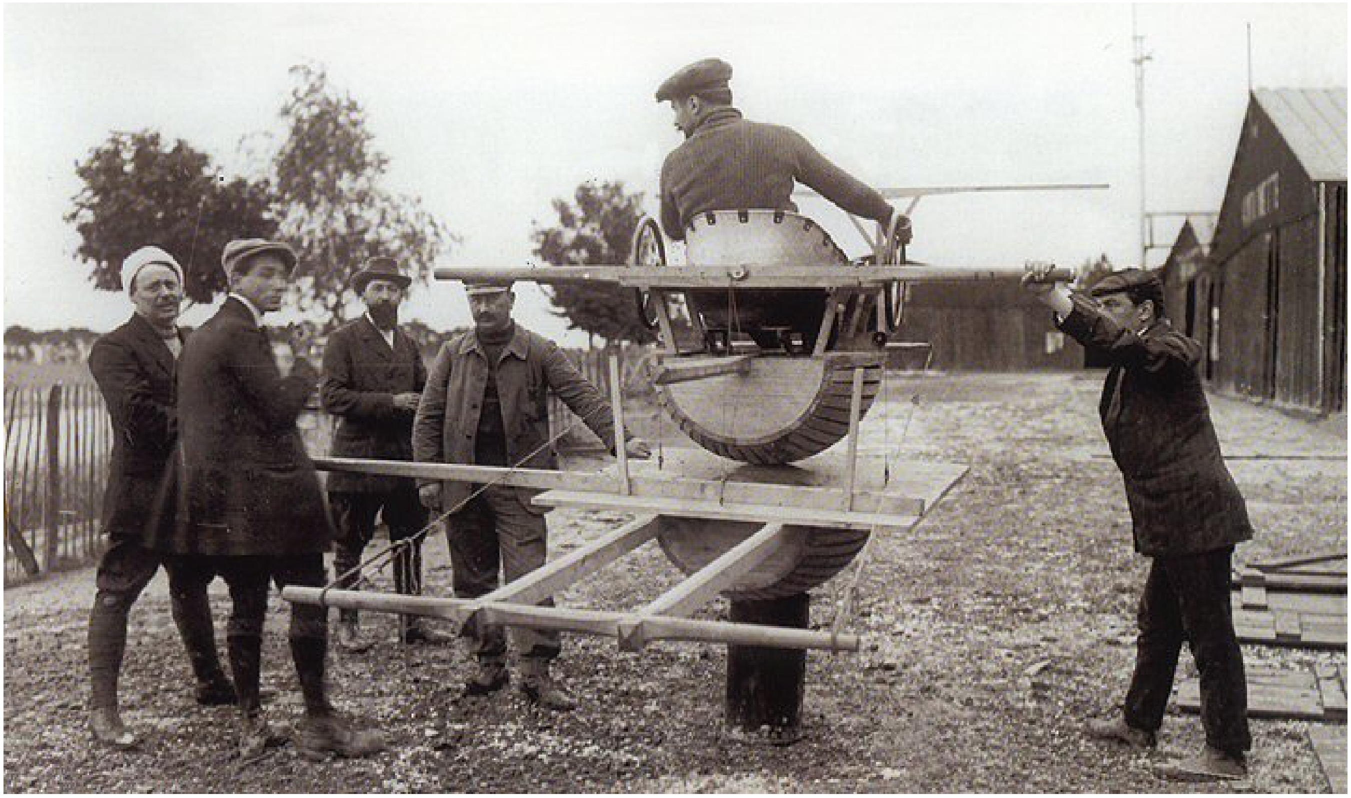
Image courtesy of Meta Immersive Synthetics

Antoinette Project demo: behind the scenes

The Antoinette Project is a flight simulator demonstration created by Epic Games as a proof of concept. The flight simulator was created with a combination of technologies, with Unreal Engine at its core to drive both the visuals and data, and additional technologies from Brunner, MIS, and Varjo.

To demonstrate what a flight simulation application can look like, we needed something easy to deploy that presents the right balance between immersiveness and deployability. The Antoinette Project was our answer to these questions.

Rather than reinventing the wheel to demonstrate these values, Epic Games worked with some key players in the field to build a portable demo to illustrate some of these trends and to create an inspiration reference for the community of developers; it will be demonstrated for the first time at WATS 2022.



Initially, the project was code-named “Antoinette Project” as an homage to **La Société Antoinette**, the French simulation pioneers—in 1906, they created the Antoinette Barrel as the first known method to demonstrate to pilots what they would sense when flying an airplane. We got attached to the code name, and decided it was fitting to keep this name for the publicly released demo.

The Antoinette Project demo consists of:

- NOR software framework from MIS, allowing developers to build training scenarios and serious flight simulation applications by leveraging all the expertise Meta Aerospace injected into this. You can read more about this on the **Unreal Engine blog**.
- **Brunner’s highly deployable 6DOF motion platform**, which uses advanced motion-cueing algorithms and high-fidelity control loading units. The kinaesthetic cues given by the platform let the pilot feel the aircraft he’s sitting in. The feedback given by the aircraft not only immerses the pilot much more but also trains muscle memory and his feeling for the aircraft’s movements and forces. Brunner worked on their Unreal Engine integration leveraging the **open-source motion-cueing interface developed by Epic Games**, you could develop your own plugin to any motion cueing solution you want.
- VARJO VR3 HMD which produces high-resolution out-the-window scenes in a very portable way. “OpenXR features include full support for Varjo’s photorealistic visual fidelity, eye tracking and real-time chroma keying. The resulting solution is likely to position Unreal Engine as the most advanced platform for enterprise mixed reality development.” You can read more about it on the **Varjo website**.

Summary

As described in this paper, content creation and production pipelines are evolving at an incredible pace. Not long ago we noted one common question from the simulation community stakeholders: "How big or how much terrain can Unreal Engine leverage?" These questions are gone now with the introduction of 64-bit precision support and the multiple examples of worldwide database environments that can be handled by Unreal Engine.

Because data is essential to the accuracy of the scenario executions, this community displays insatiable requirements for larger quantities of increasingly accurate terrain for training and real-world operations, using high-fidelity, dynamic content to populate training and analysis simulation applications.

Epic Games is further helping to lead the broader simulation community's fast-paced transition to synthetic environments, which can be observed in major training programs internationally. With photogrammetry, digital twins, and AI's ability to assist in content reconstruction/augmentation, it is becoming even more important to the design process.

Over the last several years there has been a convergence of simulation software capabilities that a game engine can provide. As Unreal Engine and the ecosystem of solutions around it have evolved, it has become clear that it provides the foundational capabilities that simulation providers require. Integrating UE into these environments provides an extremely powerful open-source set of technologies, one that enables simulation providers to focus on their core strengths and provides efficient, photoreal, and accurate training solutions to end users faster..

If you'd like to discuss the technologies mentioned in this paper or talk about how a game engine can help you create and deploy a training system, please contact the Epic Games Simulation team at simulation@ epicgames.com.

Image courtesy of Meta Immersive Synthetics

About this document

This document was prepared by Sébastien Lozé, Unreal Engine Simulation Business Director, and Daniel Williams, Technical Account Manager Simulation from the Unreal Engine Simulation industry group, with the help of the partners from the Epic Games ecosystem.

Authors

Sébastien Lozé

Contributors

Alban Bergeret

Daniel Williams

Editor

Michele Bousquet

Layout

Jung Kwak

Image courtesy of Meta Immersive Synthetics