April 29, 2020

Mixed reality takes giant leap forward with Unreal Engine and HoloLens 2

Architecture (/en-US/feed?tags=architecture) **More Uses** (/en-US/feed?tags=more-uses) **MR** (/en-US/feed?tags=mr) **Product Design** (/en-US/feed?tags=product-design) **AR** (/en-US/feed?tags=ar) **Broadcast & Live Events** (/en-US/feed?tags=broadcast-live-events) **HoloLens** (/en-US/feed?tags=hololens) **Apollo 11** (/en-US/feed?tags=apollo-11) **Simulation** (/en-US/feed?tags=simulation) **Blueprints** (/en-US/feed?tags=blueprints) **Spotlight** (/en-US/feed?tags=spotlight)

Picture a team of engineers around a table. They're collaborating and interacting with the same ultra-high fidelity digital 3D hologram, streamed wirelessly to their HoloLens 2 devices from a high-end PC.

This is the technical capability showcased in the live Apollo 11 HoloLens 2 demonstration—and it's a giant leap forward for mixed reality in an enterprise context. The demo illustrates that by leveraging Unreal Engine and HoloLens 2, we can achieve the highest-quality visuals yet seen in mixed reality.

We previously took a look at this project (https://www.unrealengine.com/en-US/blog/microsoft-build-2019-epic-games-apollo-11-mission-hololens-2-unreal-engine-4)when it came to fruition last year. Now, we’re going to dive deep into some of the technical aspects of its creation.

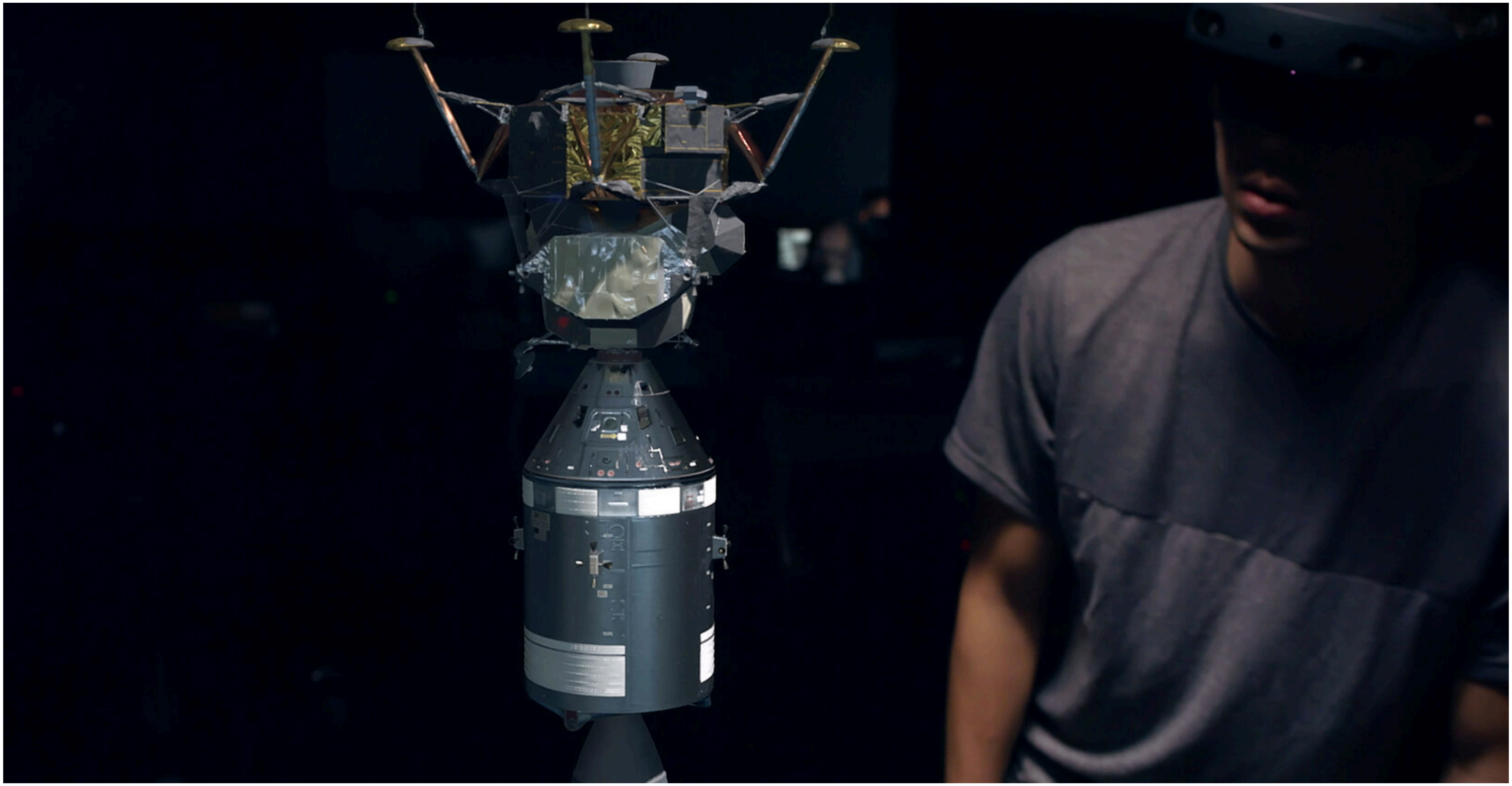
High-fidelity MR visuals streamed wirelessly

The Apollo 11 demo—originally intended to be presented onstage at Microsoft Build—is a collaboration between Epic Games and Microsoft to bring best-in-class visuals to HoloLens 2. “When Alex Kipman, Microsoft’s technical fellow for AI and mixed reality, invited us to try out early prototypes of the HoloLens 2, we were blown away by the generational leap in immersion and comfort it provided,” explains Francois Antoine, Epic Games’ Director of Advanced Projects who supervised the project. “We left the meeting with only one thought in mind: what would be the best showcase to demonstrate the HoloLens 2’s incredible capabilities?”

It just so happened that last year was the 50th anniversary of mankind’s biggest technical achievement—the first humans landing on the moon. This provided the context they were looking for, and so the Epic team set out to create a showcase that retold key moments of the historic mission.

To make sure they were as faithful as possible to the source material, the team called upon some of the industry’s foremost experts on the Apollo 11 mission. The live demo was presented by ILM’s Chief Creative Officer John Knoll and Andrew Chaikin, space historian and author of Man on the Moon. Knoll provided much of the reference material, advised on the realism of the 3D assets, and explained the whole mission to the Epic team working on the project.

Diving into many aspects of the Apollo 11 mission, the demo offers an unprecedented level of visual detail. “To date, there is nothing that looks this photoreal in mixed reality,” says MinJie Wu, Technical Artist on the project.

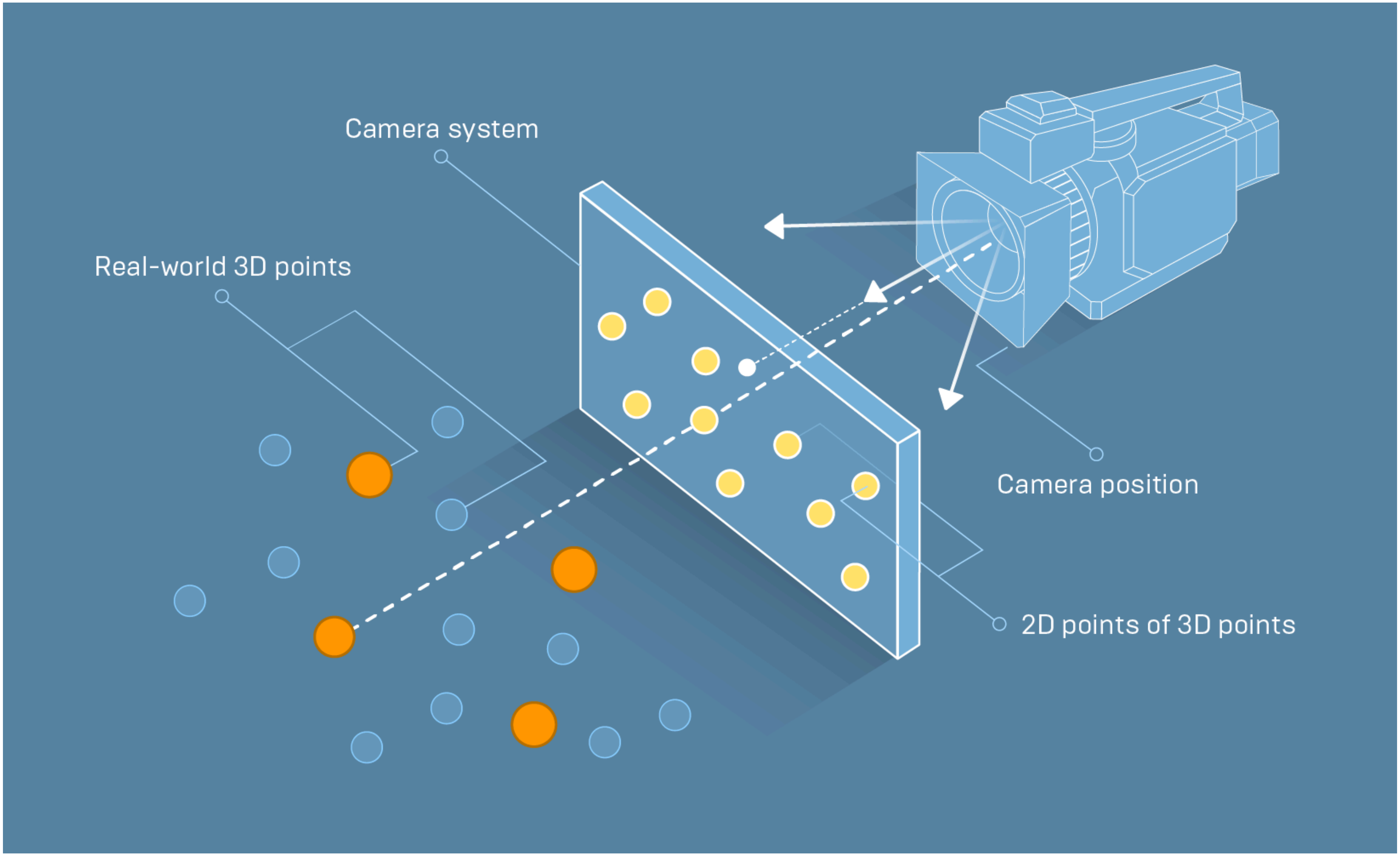


Visuals are streamed wirelessly from Unreal Engine, running on networked PCs, to the HoloLens 2 devices that Knoll and Chaikin are wearing, using a prototype version of Azure Spatial Anchors (https://azure.microsoft.com/en-us/services/spatial-anchors/#features) to create a shared experience between the two presenters.

By networking the two HoloLens devices, each understands where the other is in physical space, enabling them to track one another. A third Steadicam camera (https://en.wikipedia.org/wiki/Steadicam) is also tracked by bolting a HP Windows Mixed Reality headset onto the front of it.

Calibrating this third physical camera to align with the Unreal Engine camera was particularly challenging. “We shot a lens grid from multiple views and used OpenCV to calculate the lens parameters,” says David Hibbitts, Virtual Production Engineer. “I then used those lens parameters inside of Unreal with the Lens Distortion plugin to both calculate the properties for the Unreal camera (field of view, focal length, and so on) and also generate a displacement map, which can be used to distort the Unreal render to match the camera footage.”

Even if you are able to get the camera settings and distortion correct, there can still be a mismatch when the camera starts moving. This is because the point you're tracking on the camera doesn't match the nodal point of the lens, which is what Unreal Engine uses as the camera transform location, so you need to calculate this offset.



“To solve this, you need to know some known 3D positions in the world and some known 2D positions of those same points in the camera view, which lets you calculate the nodal point's 3D position, and if you know the tracked position of the camera when you capture the image, you can find the offset,” explains Hibbitts.

For getting the known points and known camera position, the team made sure to have the tracking system running while it shot the lens grids using the points on the grid as the known points. This enabled it to get the calibration of both the camera and the tracking offset in one pass.

Three Unreal Engine instances are required for the setup: one for the camera and one for each HoloLens. They all network to a separate, dedicated server. “They're all talking to each other to figure out where they are in the physical space, so everybody could look at the same thing at the same time,” explains Ryan Vance, XR Lead on the Apollo 11 project.

Shifting the computing process away from the mobile device and onto a high-powered PC is a significant step forward for mixed reality. Previously, you’d have to run the full Unreal Engine stack on the mobile device. “The advantage of that is that it’s the standard mobile deployment strategy you’d use for a phone or anything else,” says Vance. “The disadvantage is that you’re limited by the hardware capability of the device itself from a compute standpoint.”

Now, by leveraging Unreal Engine’s support for Holographic Remoting, high-end PC graphics can be brought to HoloLens devices. “Being able to render really high-quality visuals on a PC and then deliver those to the HoloLens gives a new experience—people haven’t had that before,” says Vance.

Image courtesy of Microsoft | HoloLens 2 | Epic Games

Holographic Remoting streams holographic content from a PC to Microsoft HoloLens devices in real time, using a Wi-Fi connection. “Not being dependent on native mobile hardware to generate your final images is huge,” says Wu.

The interplay between the presenters and the holograms illustrates the difference between designing interactions for MR and VR. “In VR, you set up your ‘safe-zone’ and promise you’re only going to move around in that,” says Jason Bestimt, Lead Programmer on the project. “In MR, you are not bound in this way. In fact, moving around the space is how you get the most out of the experience.”

Image courtesy of Microsoft | HoloLens 2 | Epic Games

Addressing roundtrip latency with reprojection

To create the Apollo 11 demo, the team leveraged features that have already been used in games for many years. “The cool thing is, you can take standard Unreal networking and gameplay framework concepts—which are pretty well understood at this point—and use them to build a collaborative holographic experience with multiple people,” says Vance.

Whenever you add a network connection into an XR system, there’s latency in streaming the tracking data. One of the biggest challenges on the project was ensuring the tracking systems aligned despite at least 60 milliseconds roundtrip latency between the PC and the HoloLens devices. “The world ends up being behind where your head actually is,” explains Vance. “Even if you're trying to stand extremely still, your head moves a little bit, and you'll notice that.”

To address this, Microsoft integrated their reprojection technology (https://docs.microsoft.com/en-us/windows/mixed-reality/hologram-stability#reprojection) into the remoting layer—a standard approach for dealing with latency issues. Reprojection is a sophisticated hardware-assisted holographic stabilization technique that takes into account motion and changes to the point of view as the scene animates and the user moves their head. Applying reprojection techniques ensured all parts of the system could communicate and agree where a point in the virtual world correlated to a point in the physical world.

Experimenting with mixed reality interactions

After the live event project proved the HoloLens 2-to-Unreal Engine setup viable, the team created a version of the demo for public release. “We wanted to repackage it as something a bit simpler, so that anybody could just go into deploying the single-user experience,” says Antoine. The Apollo 11 demo is available for download (https://www.unrealengine.com/marketplace/en-US/product/missionar) on the Unreal Engine Marketplace for free.

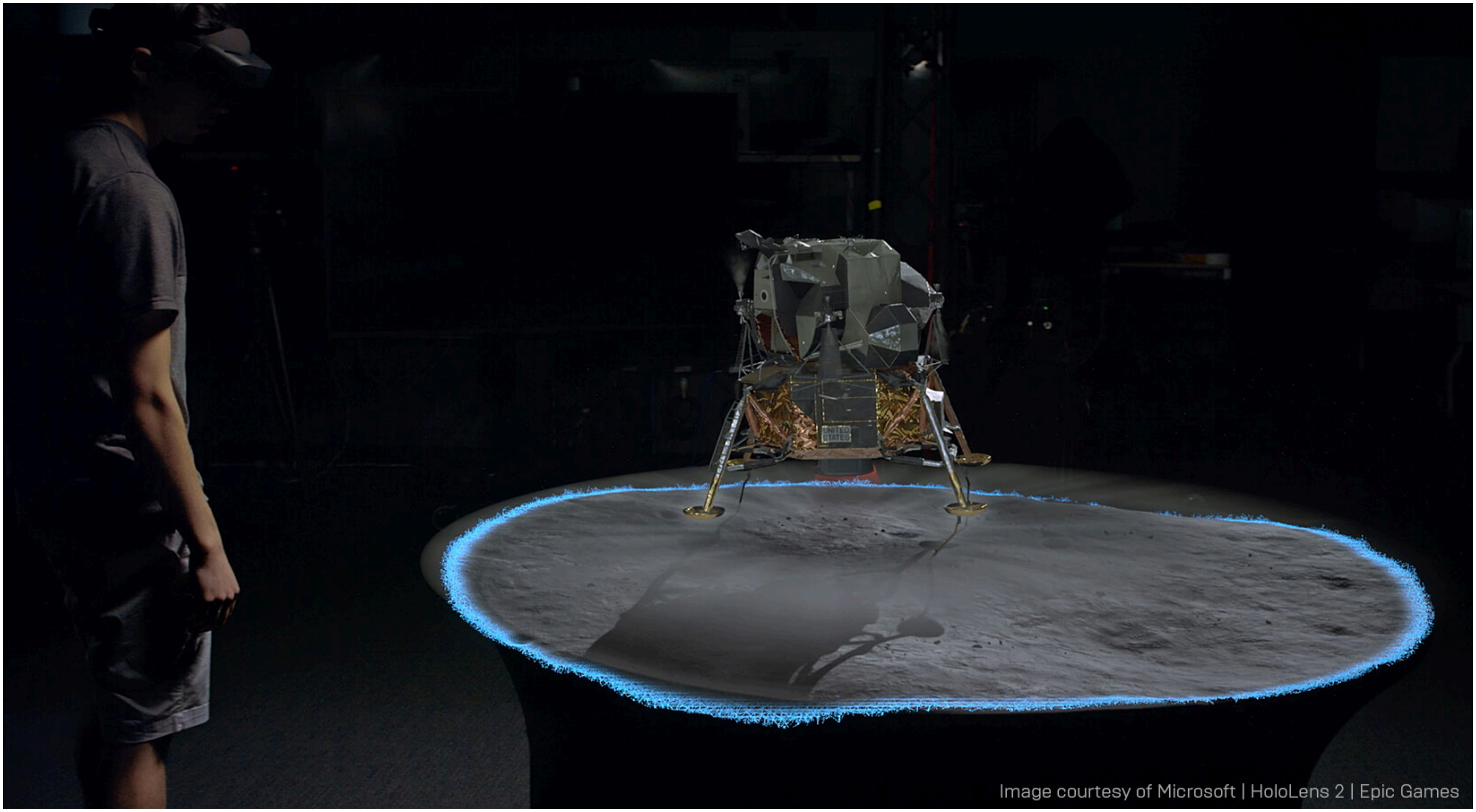


Image courtesy of Microsoft | HoloLens 2 | Epic Games

The team discovered much about the types of mixed reality interactions that were most successful while creating this public version. “At first, we created plenty of insane interactions—Iron Man-style,” says Simone Lombardo, Tech Artist on the project. “But while we thought these were really fun and intuitive to use, that wasn’t the case for everybody.”

Users new to mixed reality found the more complicated interactions difficult to understand, with many triggering interactions at the wrong time. The easiest interactions proved to be a simple grab/touch, because these mirror real-world interactions.

Based on this finding, the Apollo 11 demo leverages straightforward touch movements for interaction. “We removed all the ‘complex’ interactions such as ‘double pinch’ not only because demonstrators were accidentally triggering them, but also because we ended up getting a lot of unintentional positives when the users’ hands were just out of range,” explains Bestimt. “Many users at rest have their thumbs and index fingers next to each other, creating a ‘pinch’ posture. As they raised their hand to begin interacting, it would automatically detect an unintentional pinch.”

Other methods of interaction—such as using menus—also proved detrimental to the experience, resulting in the user unconsciously moving around the object far less.

The team also found novel ways to create a user experience that bridged the gap between the virtual and physical worlds. Many users’ natural instinct upon encountering a hologram is to try to touch it, which results in their fingers going through the hologram. “We added a distance field effect that made contact with real fingers imprint the holograms with a bluish glow,” says Lombardo. “This creates a connection between the real and unreal.”

Image courtesy of Microsoft | HoloLens 2 | Epic Games

The team have since been working on Azure Spatial Anchors support for the HoloLens 2, which is now available in Unreal Engine 4.25 (https://docs.unrealengine.com/en-US/Support/Builds/ReleaseNotes/4_25/index.html#new:hololens2improvements). Spatial anchors will allow holograms to persist in real-world space between sessions. “That makes it a lot easier to get everything to align,” explains Vance. “So it should be relatively simple to reproduce the capabilities we demonstrated in our networked stage demo, with multiple people all sharing the same space.”

Since the launch of Microsoft HoloLens 2, holographic computing has been predicted to have a seismic impact on industries ranging from retail to civil engineering. The setup showcased in the Apollo 11 demo represents a huge progression in the quality of visuals and interactions that will take us there.

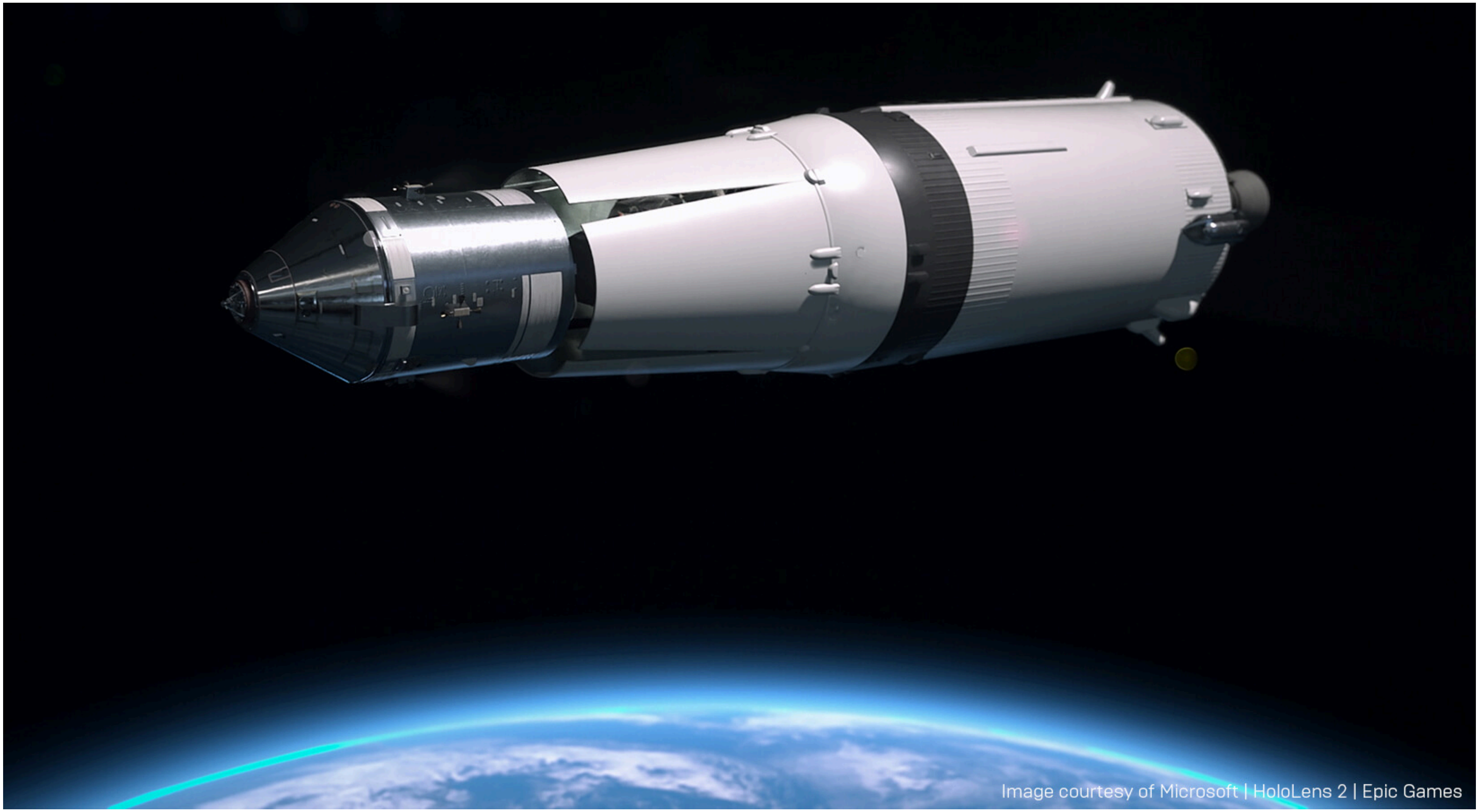


Image courtesy of Microsoft | HoloLens 2 | Epic Games

Want to create your own photorealistic mixed reality experiences? Download Unreal Engine (https://www.unrealengine.com/en-US/) for free today.

(/en-US/feed)

Games

Fortnite

Fall Guys

Rocket League

Unreal Tournament

Infinity Blade

Shadow Complex

Robo Recall

Marketplaces

Epic Games Store

Fab

Sketchfab

ArtStation

Store Refund Policy

Store EULA

Tools

Unreal Engine

UEFN

MetaHuman

Twinmotion

Megascans

RealityScan

Rad Game Tools

Online Services

- Epic Online Services
- Kids Web Services
- Services Agreement
- Acceptable Use Policy
- Trust Statement
- Subprocessor List

Company

- About
- Newsroom
- Careers
- Students
- UX Research

Resources

- Dev Community
- MegaGrants
- Support-A-Creator
- Creator Agreement
- Distribute on Epic Games
- Unreal Engine Branding Guidelines
- Fan Art Policy
- Community Rules
- EU Digital Services Act Inquiries
- Epic Pro Support

Back to top

Terms of service   Privacy policy   Safety & security