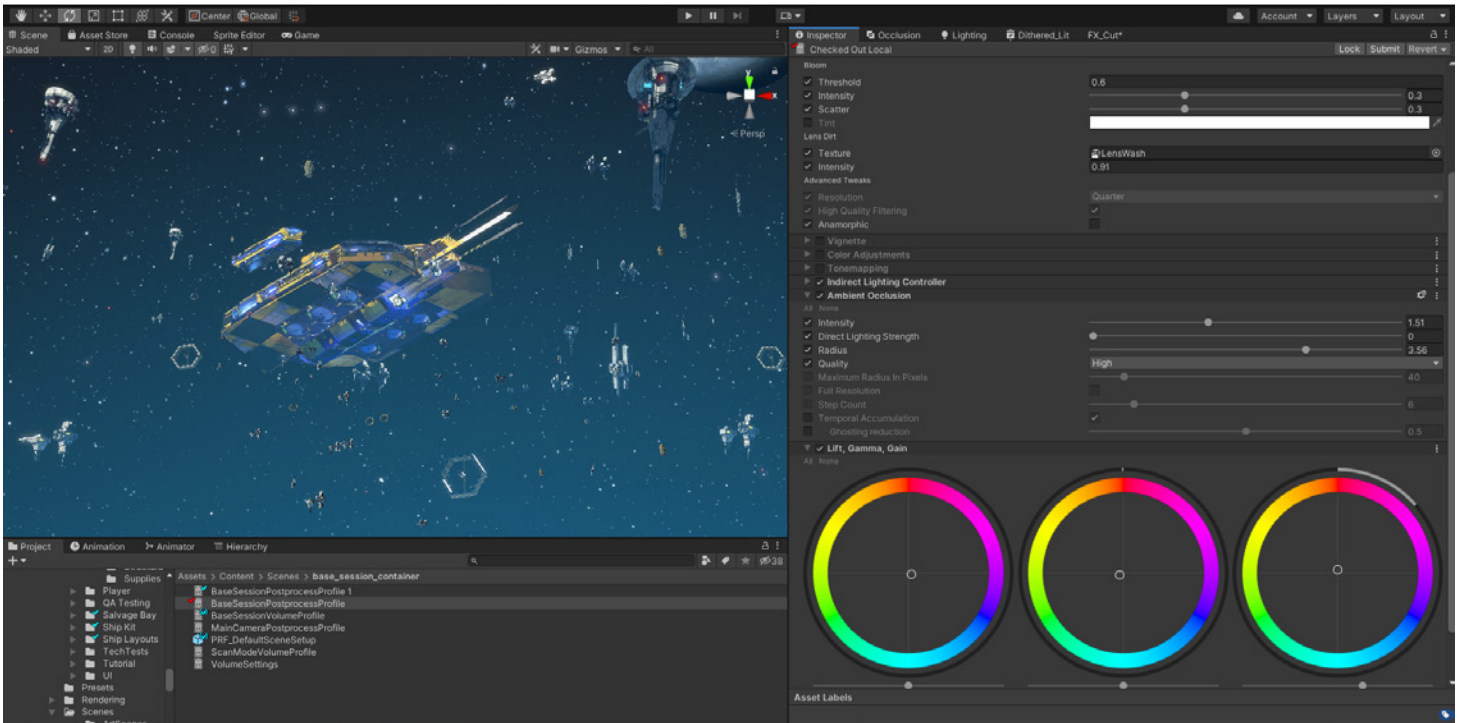




Hardspace: Shipbreaker

Learn how Unity's artist tools and workflows help create the stunning visuals of *Hardspace: Shipbreaker*.



Post-processing lighting setup for a salvage bay

How do you create heart-pumping excitement in a sci-fi game where there are no aliens or enemies? By remembering that in cold, dark space, the entire environment is the antagonist. In [Hardspace: Shipbreaker](#) you're salvaging junk spacecraft to pay off your debt to Lynx Corporation. But watch out – there's a good chance you'll set off an explosion and have to respawn, increasing your debt. To bring this futuristic and suspenseful game full of highly detailed, immersive visuals to life, Blackbird Interactive (BBI) chose Unity.

An out-of-this-world, skills-based game

From the game's conception, the studio wanted *Hardspace: Shipbreaker* to awe players with its high-fidelity vision of life in space. The game also had to feature physically accurate, zero-gravity object behaviors and collisions. This presented significant challenges such as creating baked-quality artwork for completely dynamic environments, making processes integrated and efficient, and constantly finding better ways to develop and collaborate. Since the project start over four years ago, the BBI team has blasted through numerous technical barriers, innovated countless artistic techniques, and shared many "oh wow" moments on their way to delivering one of the most exciting space games in eons.

CHALLENGE

Empower Blackbird Interactive's artists with the tools and workflows to achieve their visual goals while simplifying collaboration and speeding up development.



Platform

PC



Project staff

~25 Unity artists and developers



Location

Vancouver, Canada



A Cutter salvaging a junk spaceship

Drawing players in with immersive, personalized storytelling

From the start, the BBI team wanted players – called “Cutters” – to fully feel the challenges and loneliness of space, under pressure to build the skills necessary to succeed against tough odds. But this is not a superhero game. As players move and work within the rich visual environment, they each create their own story.

“That’s a big part of the immersive experience for us,” says Trey Smith, *Hardspace: Shipbreaker*’s Game Director. “You play as yourself and become part of the story. For example, we designed the hands so they feel like they’re your hands and not some random bot hands. Highly realistic animation and effects were crucial too, as they ground you in the gameplay, making you more invested in what’s going on.”

The result is one of the most dynamic, rich gameplay environments imaginable. As players maneuver and cut, electricity sparks and fire erupts in the highly charged settings. There are realistic tools to manipulate, signs to read, tethers to attach, tight spaces to navigate, deep shadows and flying particles to adjust to, and much more. All in a zero-gravity, topsy-turvy environment. And because each player’s journey is unique, great individual stories unfold.

RESULTS

Quick grey-boxing with ProBuilder and Cinemachine to nail the proof of concept

Faster workflows with virtually immediate round-tripping for 3D digital assets in FBX format

Explosive, real-time particle effects courtesy of VFX Graph, with thousands of unique objects, high visual fidelity, and no performance lags

Greater artist involvement in “under the hood” processes with nested Prefabs

Unprecedented area and volumetric lighting effects with Unity’s High Definition Render Pipeline (HDRP)

“Our founders are all art directors and space enthusiasts, yet we also know that gameplay is king. So we want people to be both drawn into our unique, painterly environment and play within it. And Unity helps us combine those two key elements seamlessly.”

– Chris Williams, Art Director,
Blackbird Interactive

A studio founded on art and outer space

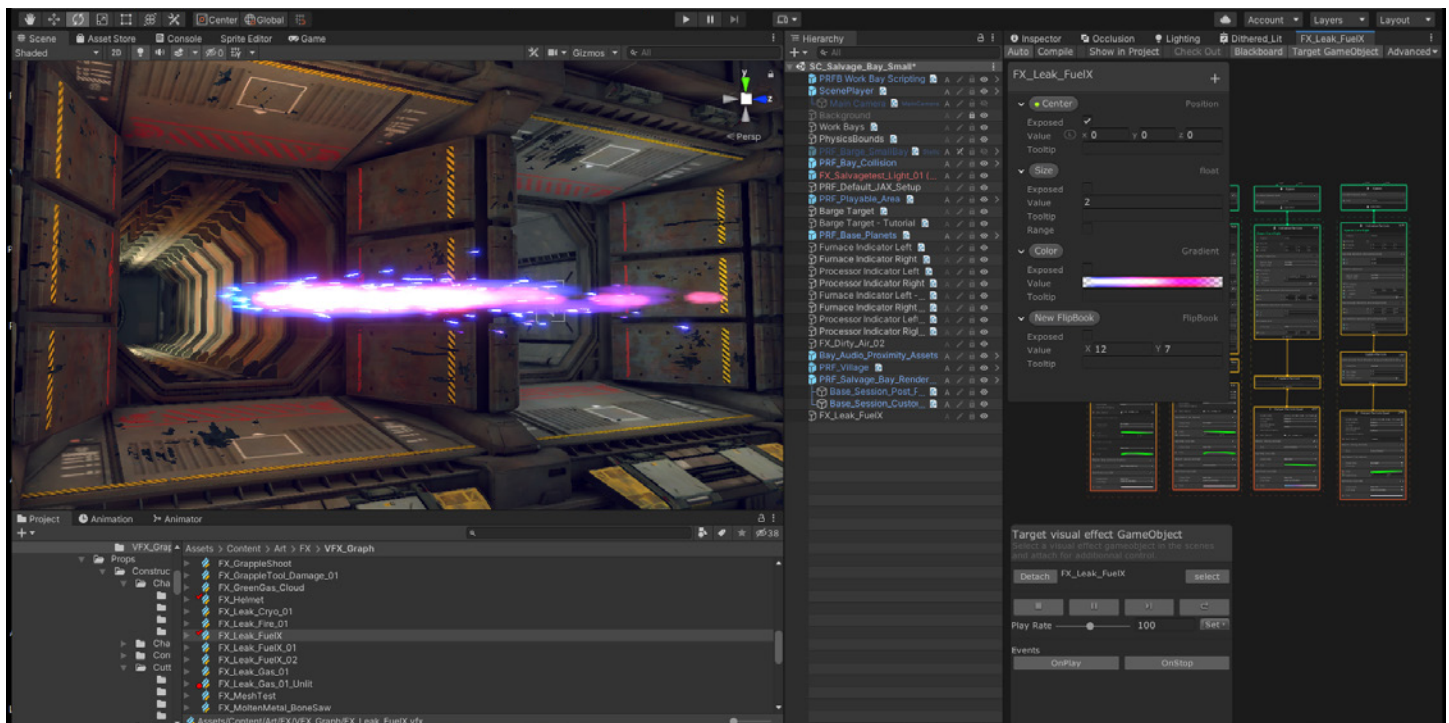
Putting this heightened drama together hinges on deeply realized visuals, and the studio was up for the challenge because, at its heart, it's visually driven. "Our founders are all art directors and space enthusiasts, yet we also know that gameplay is king. So we want people to be both drawn into our unique, painterly environment and play within it," according to Art Director Chris Williams. "And Unity helps us combine those two key elements seamlessly."

Virtually everyone on the team had worked with Unity according to *Hardspace: Shipbreaker's* Tech Director Richard Harrison, "It's a triple-A engine, but Unity also caters to developers who want to get something off the ground and reach that proof of concept really quickly." This was especially important considering the small size of the team when the game was born (during an internal game jam) and the need to solve some substantial artistic challenges.

Lifting off with grey-boxing and cinematic tools

BBI leverages [ProBuilder](#) to determine how to properly manipulate, test, and craft the plethora of ships and other 3D models. "ProBuilder enables our designers to prototype ship layouts and gameplay props. Since object dimensions and other metrics are extremely important in our game, it lets us quickly iterate on rough blockouts to nail down an object size before asking Art for the final mesh," says Elliot Hudson, BBI's Associate Creative Director. "It really helps our artists build an efficient grey-box pipeline."

To create many of the highly detailed and realistic animations, BBI used [Timeline](#). For example, when a player begins their workday, Timeline is behind the power-up of all the lights in the hangar bay and the fire-up of the ship's power source. "The first time our animator used Timeline, he loved it and immediately saw the potential for enhancing the realism of the game with cool details like that," says Harrison.



Setting up a dramatic fuel leak in the Visual Effect Graph

They also used [Cinemachine](#)'s advanced camera tools for important design stages. "Our developers were able to quickly create proof-of-concept segments with Cinemachine that captured the vibe and tone of what we were trying to do. Being able to scrub a timeline and see the camera in the view was particularly useful," says Trey Smith, BBI's game director.

Creating custom tools within Unity

Over the course of the project, BBI also built a set of Unity-friendly tools. Much of their core shader technology, for example, was hand-coded for the game, but the developers anticipate greater use of [Shader Graph](#) to take advantage of Unity's [High Definition Render Pipeline](#) (HDRP) capabilities.

"We used to re-UV the entire ship but then we started leveraging more things like property blocks that allow us to dynamically change properties on materials on the fly. Beyond that, we use a cleverly put-together shader in Unity that uses the HDRP back end. Karl Kent, our art tech wizard, created it to go hand-in-hand with the HDRP tech," says Harrison.

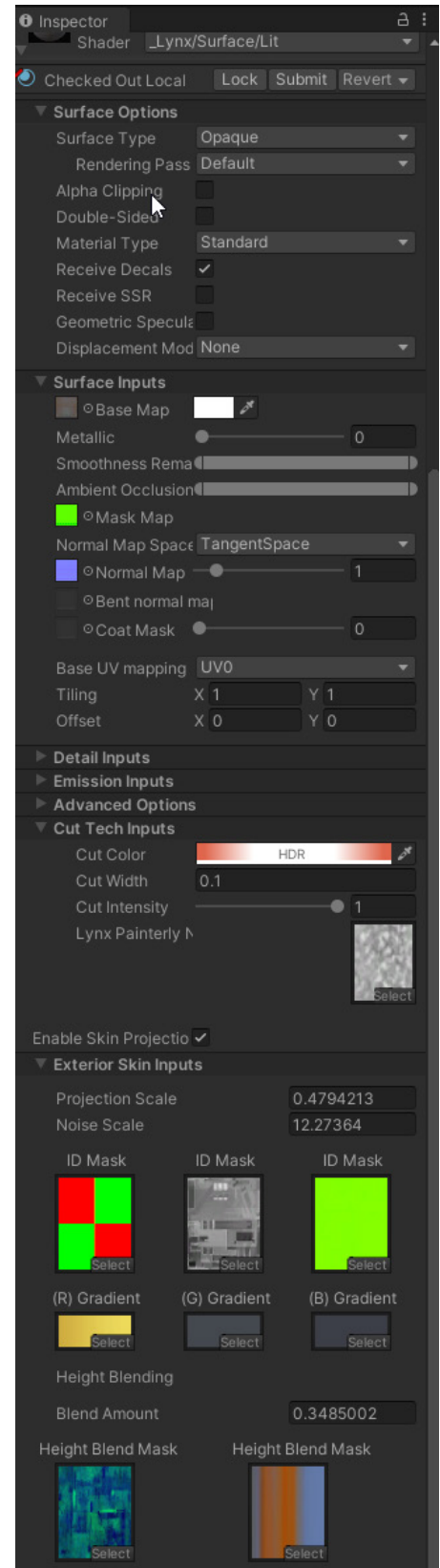
BBI's projection tool piggybacks on several aspects of Unity, including HDRP, to increase dynamism when they paint the ships at runtime. They also use a custom projection decal system to produce cool warnings in the ships, such as Danger signs on the bulkheads, for even more suspense.

Adding animations for additional realism

According to Harrison, the ships don't have a lot of animation, just a few here and there, such as the handles you use to open the doors and the doors themselves. "A lot of the animations that we do are baked into texture animations, things that you see as you work, such as the lights on the computer screens. The [MaterialPropertyBlock](#), which lets us modify things in real-time, has been super helpful."

The most complex physical animation they tackled was for the one-and-only character: the Cutter. "We had our animator rig up the character and animate the whole thing dynamically through [Mecanim](#). It's entirely state driven and parameter driven. It takes into consideration player velocity – where you're moving, and how you're drifting."

Of course, it's a first-person game, so players can only really see their hands and legs, as well as reflections and shadows from time to time. "It's a pretty powerful moment the first time you notice that when you drift off to the left, your legs lag behind you to the right. We feel that adds a sense of weight and presence to your character – helps you "feel" you're in a zero-g environment," adds Harrison.



Seamless merging of BBI's custom shader with HDRP's Lit Shader

“It’s a triple-A engine, but Unity also caters to developers who want to get something off the ground and reach that proof of concept really quickly.”

– Richard Harrison, Technical Director,
Blackbird Interactive

Speeding artist workflows with round-tripping and nested Prefabs

Since many of the game's 3D elements are created outside Unity, quick round-tripping to and from 3D modeling software, such as Maya, was particularly important to simplify and speed up repetitive processes. Williams says, "In some non-Unity environments, converting an FBX model could take half a day, but in Unity it's almost instantaneous." BBI built a custom editor, integrated within the Unity workflow, that lets their artists right-click on an asset and immediately turn it into a GameObject.

Another feature that significantly increased artist productivity was [nested Prefabs](#). It lets them build a ship as a Prefab, made up of many smaller Prefabs, which in turn comprise multiple smaller Prefabs, so when they want to change a characteristic, they only need to make the modification once and it's reflected throughout the model. In terms of workflow, this lets their artists get "under the hood" and assemble components on their own without having to code anything or rely on a programmer. According to Williams, "We were finally able to make a modular ship system with this handy feature. We wouldn't be where we are today without nested Prefabs."

In terms of always being in sync with the latest versions of Unity features, the team relies on regular updates via [Package Manager](#). "We use a lot of integrated packages so it's great to be able to sync when there's a new version that might have an enhancement or bug fix we've been waiting for. The Package Manager has been an important enhancement to our workflow," says Harrison.

Integrating particles into the game holistically

Considering the intense particle effects – from tiny electrical zaps to reactor meltdowns – that supercharge the entire game, it's not surprising that BBI tapped David Perdigon, their highly talented visual-effects artist, to create and control them. And when [Visual Effect Graph](#) (VFX Graph) became available, they adopted it immediately, finding it easy to use and tweak, and appreciate that it can handle far more digital input compared to legacy tools such as Flipbook or the Shuriken Particle System.

While it's exciting to add eye-catching pyrotechnics into a game, they almost always cause performance tradeoffs, which was a dealbreaker for the studio. However, when they implemented particles with VFX Graph, BBI found they could efficiently manage greater numbers of particles and include more stunning effects, more often.

This is largely because VFX Graph smoothly integrates particles into the [Post Processing Stack V2](#) and lighting. "There was a big performance increase when we made the switch to VFX Graph, and now particles feel like they're actually integrated into the world rather than being beside the geometry," says Harrison. They also tap Bloom and other Unity color-balancing tools to ratchet up the game's visual wow factor.



Check out our [Behind the Game](#) podcast with Blackbird Interactive's Trey Smith and Richard Harrison.

Realistically lighting up every environment

A key to producing believable and gorgeous light and shadows in space was the High Definition Render Pipeline (HDRP). For example, a hangar-bay scene might have four or five shadow-casting lights, and the developers needed to balance and align that bounce light with moving, directional sound. "With HDRP, everything is so responsive, and it was easy to understand right out of the box," says Williams.

HDRP also helped them create highly convincing area lighting in the abandoned-ship interiors, adding to the tension. For example, the team used volumetric lighting to create a "dirty air" effect when a Cutter pressurizes a ship, creating fog and drifting particulates. This makes a headlamp beam as vivid as a searchlight in the night sky. Pre-HDRP, BBI had tried using third-party tools to mimic these kinds of volumetric effects, but found them expensive and limiting. According to Williams, "The power of Unity's HDRP is that it's a systematic approach to the entire lighting model itself, and it beautifully suits our needs."

Final thoughts on getting to the launchpad

At BBI, they choose their game-development platforms carefully, asking themselves what tech and tools are best suited for the project, and then they do comprehensive technical, design, and art assessments. "I think it speaks volumes that we chose Unity for *Hardspace: Shipbreaker* right after we completed our first game – *Homeworld: Deserts of Kharak*. And now that *Hardspace* is live, we can definitely say that our initial assessments were correct – Unity was absolutely the right choice to meet our artistic, technical, and business needs," concludes Harrison. Suffice it to say, they're busy driving Unity to the max to craft new features and game content for the full release – and the results are sure to please the legion of Cutters writing their own stories in this beautifully realized "blue-collar" game.

Want to learn more?

At Unity, we empower the most innovative game creators to produce high-caliber games faster than ever. We help your studio achieve new levels of artistic productivity and creativity – from concept to commercialization. [You can learn more about our solutions here.](#)

For more information about building world-class games, check out [these resources](#) or [chat with an expert](#).