HOW BOSSA STUDIOS CREATED HIGH-FIDELITY UNDERWATER GRAPHICS IN

I AM FISH

Artist-friendly tools enabled tight collaboration and efficient workflows to drive the studio’s most immersive game yet.

→ **THE CHALLENGE**
Achieving top next-gen quality with greater artist/developer collaboration

→ **PLATFORMS**
PC, Xbox One/Series X|S

→ **PROJECT STAFF**
10 artists, 5 programmers, 3 technical artists, and 2 level designers

→ **LOCATION**
London, U.K.

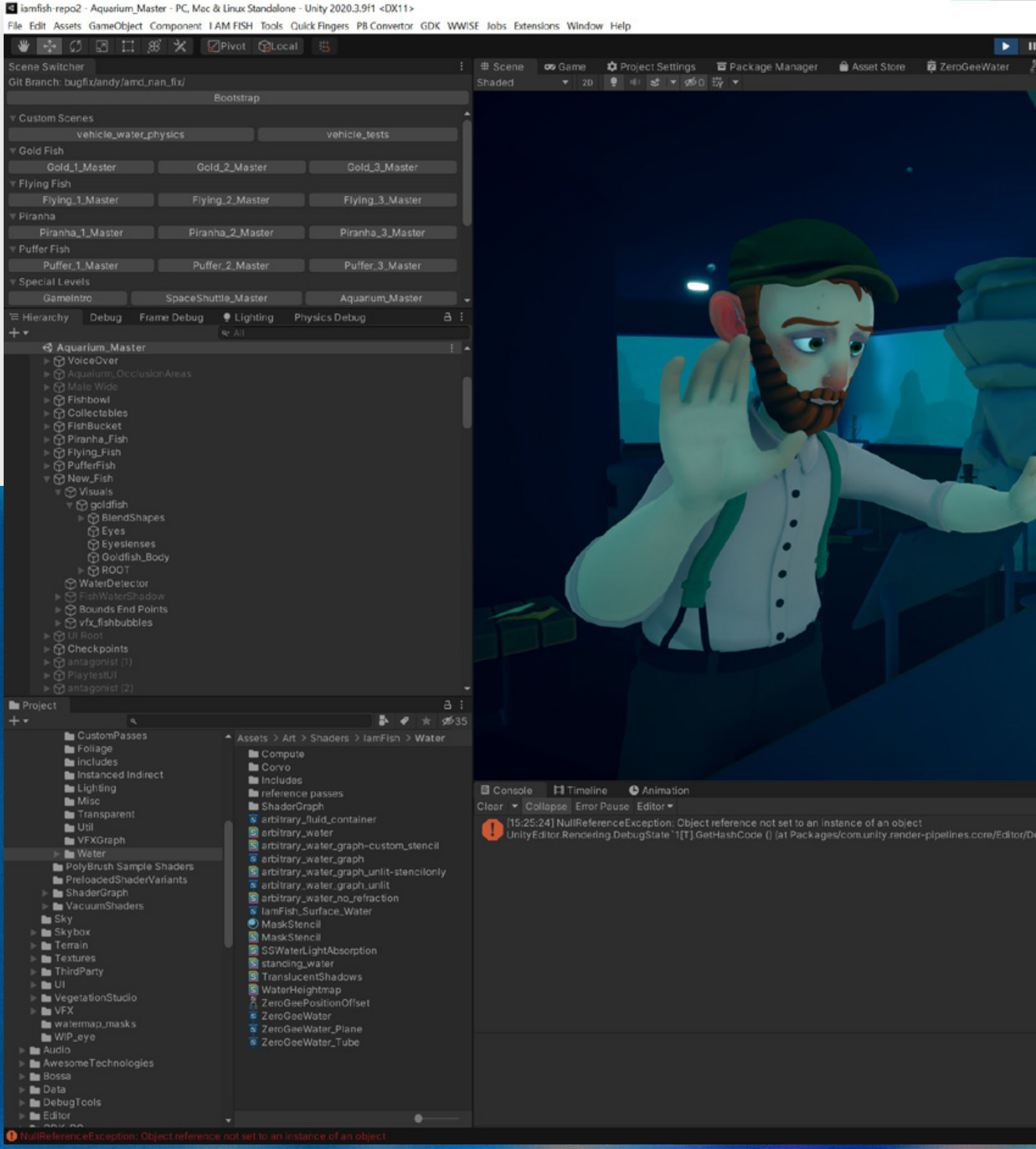
BOSSA STUDIOS: A UNITY CASE STUDY

Water – in fish bowls, oceans, even downtown streets – is the star of Bossa Studios's latest puzzler. And water rendering and simulation is normally tough to get right, particularly if you're angling for next-gen, AAA quality. Yet talented Bossa artists and programmers pulled it off in I Am Fish using the Unity High Definition Rendering Pipeline (HDRP) to refract and reflect light in a world of water and achieve stunningly realistic effects.

Combined with Unity's artist tools for in-Editor creation, their team improved their collaboration, solved complex technical tasks, and rapidly iterated to produce their most powerful game yet.

USING UNITY FOR HIGH-END 3D ON MULTIPLE PLATFORMS

Bossa Studios started off strong – in 2012, their first creation, *Monstermind*, was nominated for a best debut game BAFTA, and it won for best online browser game. It was a 2D, Flash-based Facebook offering, but as Bossa grew into 3D production for other platforms, they had to leave Flash behind. They chose to develop subsequent games with the extensive artist tools in Unity. According to Senior Technical Artist Andy Green, "Unity is a flexible and versatile engine that lets the whole team collaborate on a wide range of games and platforms." Bossa hasn't looked back.



Git Branch: bugfix/andy/amd_nan_fix/

```
[15:25:24] NullReferenceException: Object reference not set to an instance of an object
UnityEditor.Rendering.DebugState`1[T].GetHashCode () (at Packages/com.unity.render-pipelines.core/Editor/De
```

```
NullReferenceException: Object reference not set to an instance of an object
```

THE RESULTS

→ Designed, blocked out, dressed, and lit an entire level in 6–8 weeks

→ Saved 1–2 weeks per level using ProBuilder and Polybrush

→ Cut around three months of shader programmer work with Shader Graph

→ Reduced dev time for writing tools and behaviors by 3–4 weeks with Cinemachine and Timeline

PET S
RESULTS → THE RESULTS → THE RESULTS → THE RESULTS → THE RESULTS → THE RESULTS → THE RESULTS →

CREATING A FUN PHYSICS PLATFORM FOR FOUR FISHY FRIENDS

Bossa has a reputation for comedic, physics-based games, and their follow-ups *Surgeon Simulator* and *I am Bread* proved to be steady performers. Green is a Bossa veteran and, delighted with the team, says, “Where else can you debate how an inflated puffer fish should sound as it bounces down a stairwell?”

It’s a perfectly appropriate question given Bossa’s recent release of the *I Am Fish* puzzle platformer. Inspired by Disney’s *Finding Nemo* and *Dory* films, it stars a plucky pet fish who tries to escape the fishbowl and get to the Atlantic past one absolutely absurd obstacle after another. The game creates a sense of adventure, peril in an oversized world, and camaraderie. Targeting younger audiences with cute characters, it also attracts adults with touches of dark comedy.

Watch the official trailer for *I Am Fish*.

LIGHTING WATERWAYS WITH HDRP

The concept art of *I Am Fish* featured dramatic lighting of beautiful waterscapes to explore and perilous cityscapes to survive. Bossa built their first prototypes with the built-in rendering pipeline, but when they tested the look with the specific capabilities of the Xbox Series X, they decided to move forward with HDRP to get the AAA quality they needed.

Their greatest lighting challenges were imposed by the game's primary setting, which was in many ways also a main character – the omnipresent water. HDRP's Screen Space Refraction override let Bossa execute their vision with higher-fidelity lighting setups. Getting refraction just right is typically very challenging and can consume a lot of memory. Green says, "Unity HDRP has the screen space refraction buffer built in, it's so easy to apply transparency with different types of refraction in a performant and stable way."

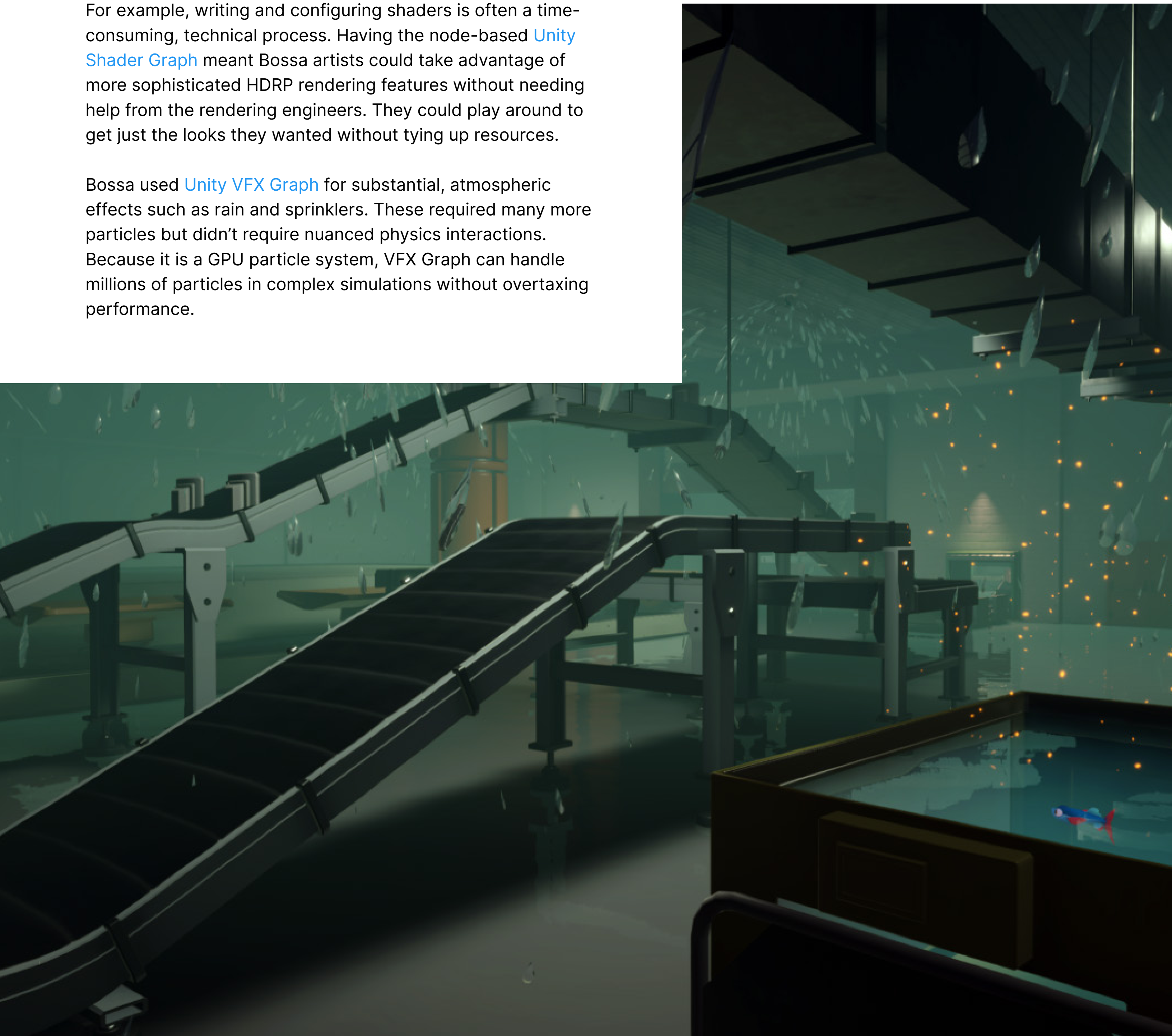
Other built-in HDRP features such as Volumetric Lighting, Volumetric Fog, and Screen Space Reflections were capabilities that previously required either custom scripts or third-party Unity Asset Store tools. According to Green, "With Unity HDRP, our lighting artists made *I Am Fish* our most high-fidelity game ever. And its flexibility let us add in-house scripting for water sorting, screen space caustics, underwater fog, and translucent shadowing features."

DEVELOPING SHADERS AND VFX WITH SPEED

Collaboration at Bossa takes several forms. Sharing is one, and Unity Nested Prefabs let artists create a base source asset for other team members to modify or directly place into other scenes in other levels. Another form of collaboration is unlocked when artists can do more with user-friendly tools, freeing engineers to attend to other challenges.

For example, writing and configuring shaders is often a time-consuming, technical process. Having the node-based Unity Shader Graph meant Bossa artists could take advantage of more sophisticated HDRP rendering features without needing help from the rendering engineers. They could play around to get just the looks they wanted without tying up resources.

Bossa used Unity VFX Graph for substantial, atmospheric effects such as rain and sprinklers. These required many more particles but didn't require nuanced physics interactions. Because it is a GPU particle system, VFX Graph can handle millions of particles in complex simulations without overtaxing performance.



In the airport level, the fish sets off the sprinkler system in the baggage hall. The sprinkler particle systems are made in VFX Graph, along with the orange "checkpoint reached" particles you can see above the fish on the right.

MAKING CINEMATOGRAPHY SIMPLE

I Am Fish is Bossa's first game to use cutscenes, and they found that Unity Timeline made the process straightforward. Bossa uses Maya to create their core animations. After importing them into Timeline, they sequenced in other assets, tweaked camera movements, and added transitional animation sequences as needed. Green says, "It was convenient being able to marry the artist's work with more real-time-dependent work in the same tool."

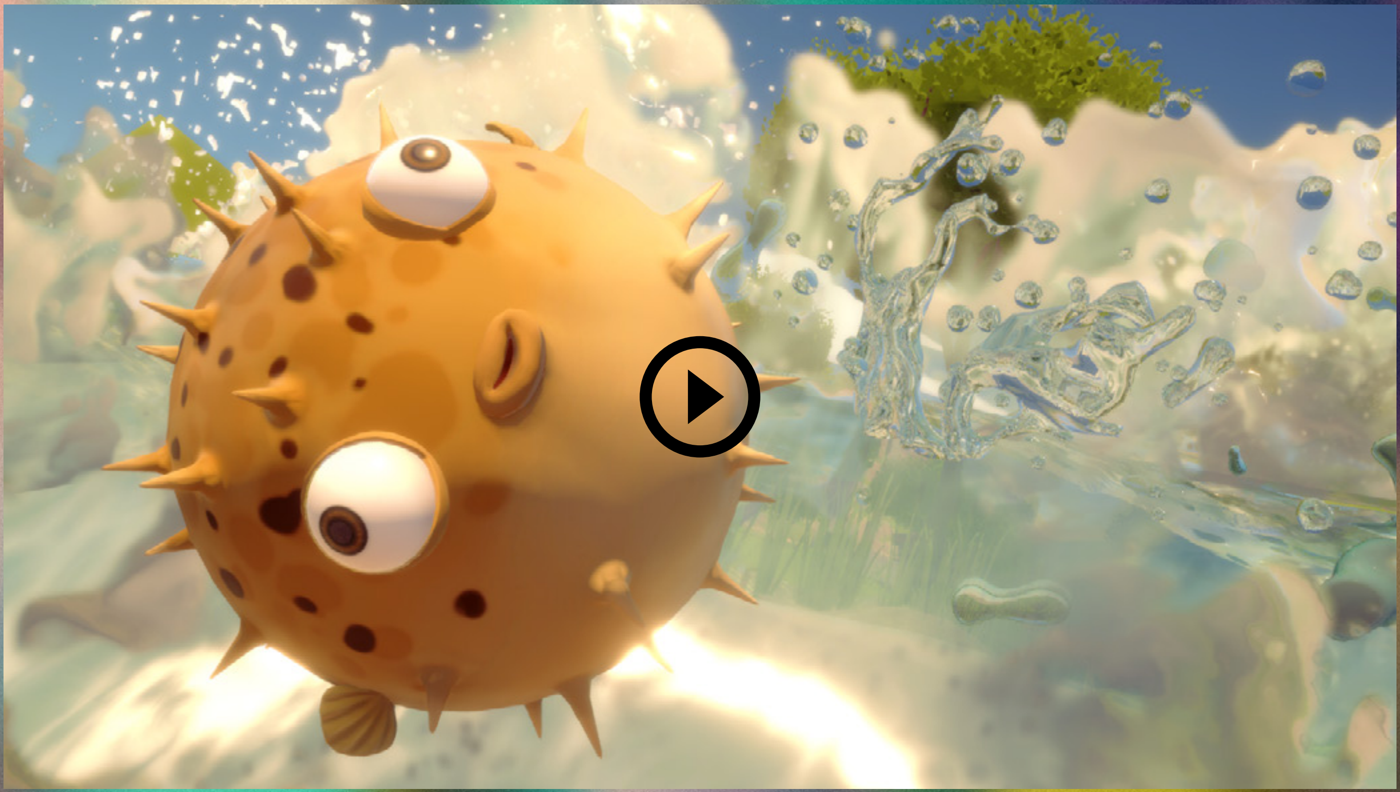
Unity Cinemachine let Bossa transition between camera behaviors seamlessly, reducing the programming they'd normally have needed for complex effects. For example, a free-orbiting camera would follow a player-controlled fish and when the fish went through a tunnel, Cinemachine's 3rd Person Follow mini-rig setup would let the camera move smoothly, keeping a consistent velocity, into the appropriate place and constrain the camera as needed.

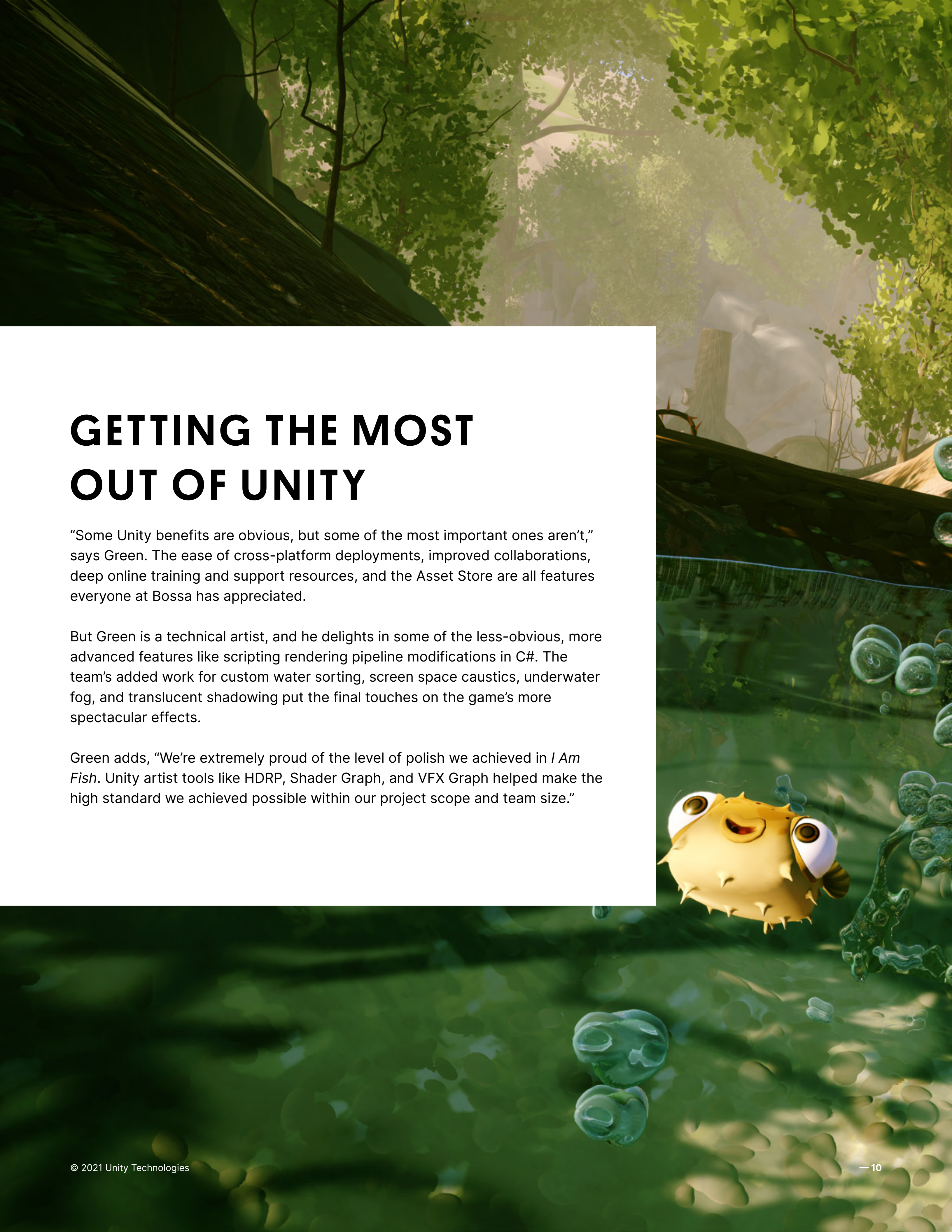
> "IT WAS CONVENIENT BEING ABLE TO MARRY THE ARTIST'S WORK WITH MORE REAL-TIME-DEPENDENT WORK IN THE SAME TOOL."
>
> — Andy Green, Senior Technical Artist, *I Am Fish*

REDUCING ROUND TRIPPING WITH PROBUILDER

Constrained to a nine-month production cycle, fast prototyping was critical for Bossa to develop the numerous and diverse environments needed for *I Am Fish*. With the game's minimalist-style art, using both Unity ProBuilder and Unity Polybrush let them move very quickly. "With Unity ProBuilder, designers could hand off meshes directly to artists," says Green. "We rapidly iterated through level design and transitioned quickly through to environment creation and set dressing." This accelerated development significantly, enabling more frequent iterations for the rest of the team to explore and review.

During the staging of the game, Unity ProBuilder enabled fast prototyping to build concepts from greybox to final render. It allowed designers and artists to work in-engine and enabled quick back and forth between the two disciplines. Watch the grey box to final art video of *I Am Fish* below.





GETTING THE MOST OUT OF UNITY

“Some Unity benefits are obvious, but some of the most important ones aren’t,” says Green. The ease of cross-platform deployments, improved collaborations, deep online training and support resources, and the Asset Store are all features everyone at Bossa has appreciated.

But Green is a technical artist, and he delights in some of the less-obvious, more advanced features like scripting rendering pipeline modifications in C#. The team’s added work for custom water sorting, screen space caustics, underwater fog, and translucent shadowing put the final touches on the game’s more spectacular effects.

Green adds, “We’re extremely proud of the level of polish we achieved in *I Am Fish*. Unity artist tools like HDRP, Shader Graph, and VFX Graph helped make the high standard we achieved possible within our project scope and team size.”

Unity®