

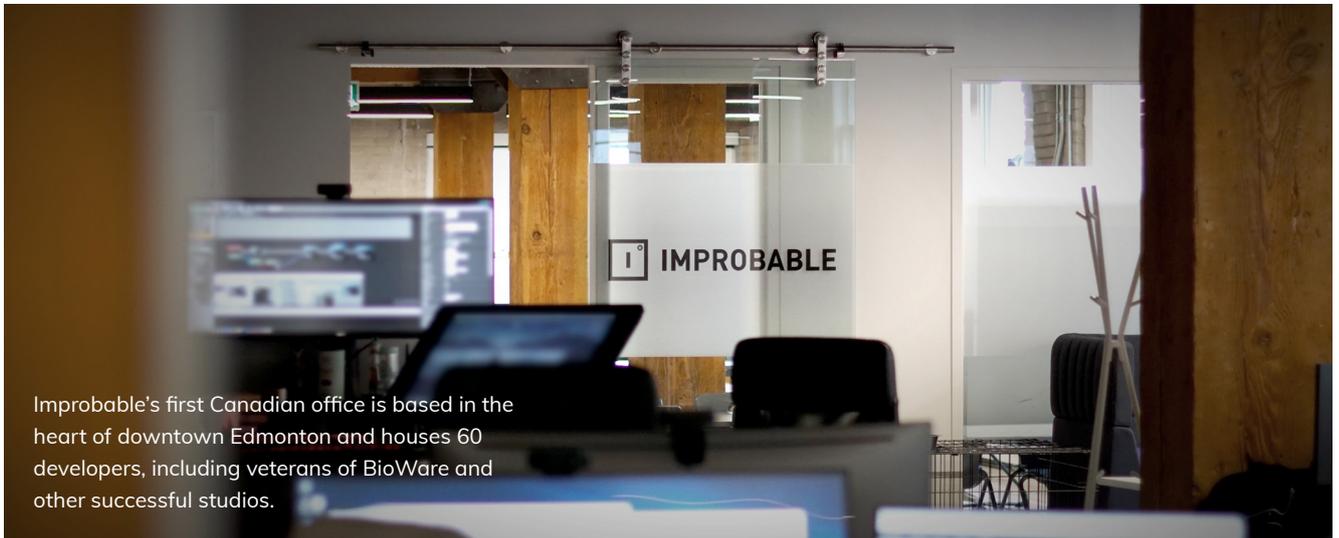


# Improbable Games Canada

Increasing the speed of  
iteration and innovation in  
multiplayer development

▣ SpatialOS

# Increasing the speed of iteration and innovation in multiplayer development.



**SpatialOS's capabilities and workflows have impressively sped up our Canadian team's game development.**

In September 2018, Aaryn Flynn founded Improbable's first studio in Edmonton, Canada. Previously, general manager Flynn was the GM of BioWare, overseeing 500 staff, working on games from *Dragon Age* to *Mass Effect* and *Star Wars: The Old Republic*. Today, his ambition hasn't dimmed and he's making games at a similar scale – but with a team less than 15% the size of his team at BioWare. The question is how is he adapting to life without a AAA headcount?

First, to create Improbable Games Canada, he's brought together a diverse group of talented developers, from AAA studios (including BioWare), smaller independent teams, and highly technical new graduates. "We are a team with years of experience in game development, working on some of the most iconic role-playing games of the past 20 years", says Flynn. They're building an online RPG using Unreal Engine 4 and SpatialOS, with the aim of pushing the technical boundaries of modern games.

Second, they're using Improbable's unique SpatialOS technology, which has enabled the team to test and iterate its designs faster than ever before. "When we were building *Anthem*, it would take a whole month and two dedicated engineers to set up a playtest for the team, which might only be stable for ten minutes. Yet the workflows that we are embracing to take full advantage of SpatialOS make a team of 60 feel like a team of hundreds." Anyone on the development team – not just the most seasoned engineer – can set up a playtest, and iteration speeds are so fast that the team easily manages two or multiple playtests with different code in a single day.

|                       |   |
|-----------------------|---|
| Platforms:<br>TBA     | Developer:<br>Improbable                    |
| Genre:<br>Online RPG  | Team size:<br>60                            |
| Release:<br>TBA       | Built with:<br>Unreal Engine 4<br>SpatialOS |
| Location:<br>Edmonton |   |

That iteration speed has given Flynn and his new team the ability to hit the ground running, especially given the complex game they're making. "Online role-playing games have always been ambitious projects," says Flynn. "They're about giving as much freedom to players and removing as many barriers as possible, allowing people to express themselves as individuals while also coming together in a shared living world. This project is that game in our imagination, and SpatialOS is the technology freeing us to make a virtual world unlike anything experienced before." Turning that ambitious vision into a reality is happening faster than any of the team have experienced and it comes from three core elements: live playtesting, effective debugging and easy comprehension of the game.

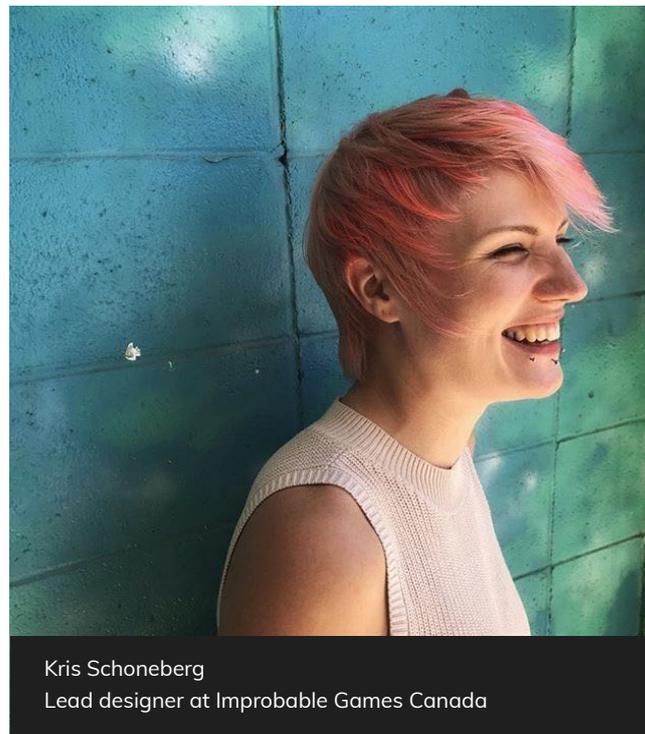
### Always-online playtesting

"In my previous experience, having a live multiplayer environment to test in has been borderline impossible," says Kris Schoneberg. She's lead designer at Improbable Games Canada, having previously been creative director on Anthem. "Given the friction that often comes with iterating in a multiplayer environment a lot of features were created in a single-player environment - obviously not great when you are creating a multiplayer game."

**"I have never experienced that kind of turnaround time for iterating on designs in a multiplayer environment in my 15 years of experience as a game developer. Ever."**

– Kris Schoneberg [Lead Designer]

However, this time they're building on SpatialOS – so they have easy access to always-on multiplayer playtests of their latest code. Rather than running code around the office on a USB stick or, worse, sending it over insecure networks, the team uses the SpatialOS Launcher tool to generate a single shareable link for anyone to play their latest build running in the cloud.



The Launcher has enabled them to set up and run playtests in a matter of minutes. "Something that can be challenging is co-ordinating people, making sure that everyone can run the latest build," says Deanna Dombroski, QA gameplay analyst at Improbable Games Canada. "So, being able to send a link and they click two buttons and they're in? It means we can focus more on the content of what we're testing, or test multiple times with different versions."

"I see the development team working on multiple builds at the same time," reports Flynn. "Different A/B tests are happening, different experiments, and all that's completely disambiguated for the development team – it's just frictionless functionality that lets them do exactly what they want to do." This has had a big impact on the team, enabling it to dramatically reduce its iteration cycles and spend its time creating the game it's aiming for.

"I've never experienced that kind of turnaround time for iterating on designs in a multiplayer environment in my 15 years of experience as a game developer. Ever," says Schoneberg. "We can prototype a feature on Monday morning, have it in the multiplayer build by the afternoon and have action items ready to go on Tuesday."

**“We aren’t thinking, ‘How do I get this up and running?’ We’re thinking, ‘How can I make my design fun?’ And it allows us to find that fun really quickly.”**

– David Holmes [Gameplay Designer, Improbable]

### Efficient and effective debugging.

Testing that fast is great, but it needs something else to unlock faster iteration – an equally fast and reliable debugging loop. This is even more vital, because the team has adopted a ‘zero bugs policy’. This prevents the team from accumulating technical debt as it rapidly prototypes new features, and avoids long periods of downtime with the game unplayable – but it means the team needs to be able to easily diagnose and fix problems.

Leah Summers was a producer at BioWare and is now senior producer on the project. “One of the biggest restrictions of previous technology I’ve worked with is that it would take weeks before you would understand where a bug was coming from.” With rare playtests and limited data, bugs can take weeks or months to resolve.

The team combats this by using the SpatialOS suite of production-ready logs, metrics and inspection tools. These online tools enable the team to quickly spot and understand any performance bottlenecks or gameplay

bugs, and flag them for resolution. Bjorn Taylor, Improbable Games Canada’s senior technical designer, has found pairing the logs and metrics particularly useful: “We’ve already done lots of performance optimisations, because when bad things are happening, we can see the logs and metrics, work out what was happening, and go fix the problem.”

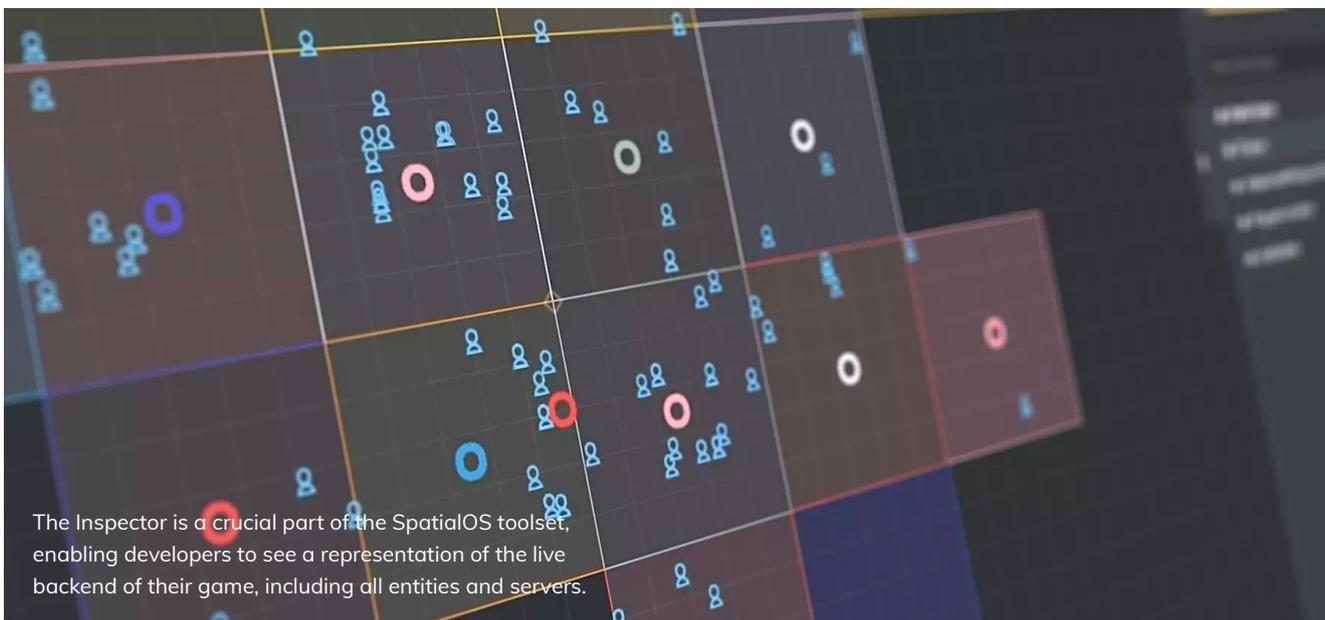
SpatialOS also provides systems that store all your logs and metrics data, a favourite feature of software developer Devon Sigurdson: “These provide tons of logging support. We can get unique counts: how often did something occur, when it started occurring, as well as great crash reporting.”

**“Since iteration times are so quick, we usually know whenever a bug gets introduced within a few hours rather than waiting weeks or months between playtests.”**

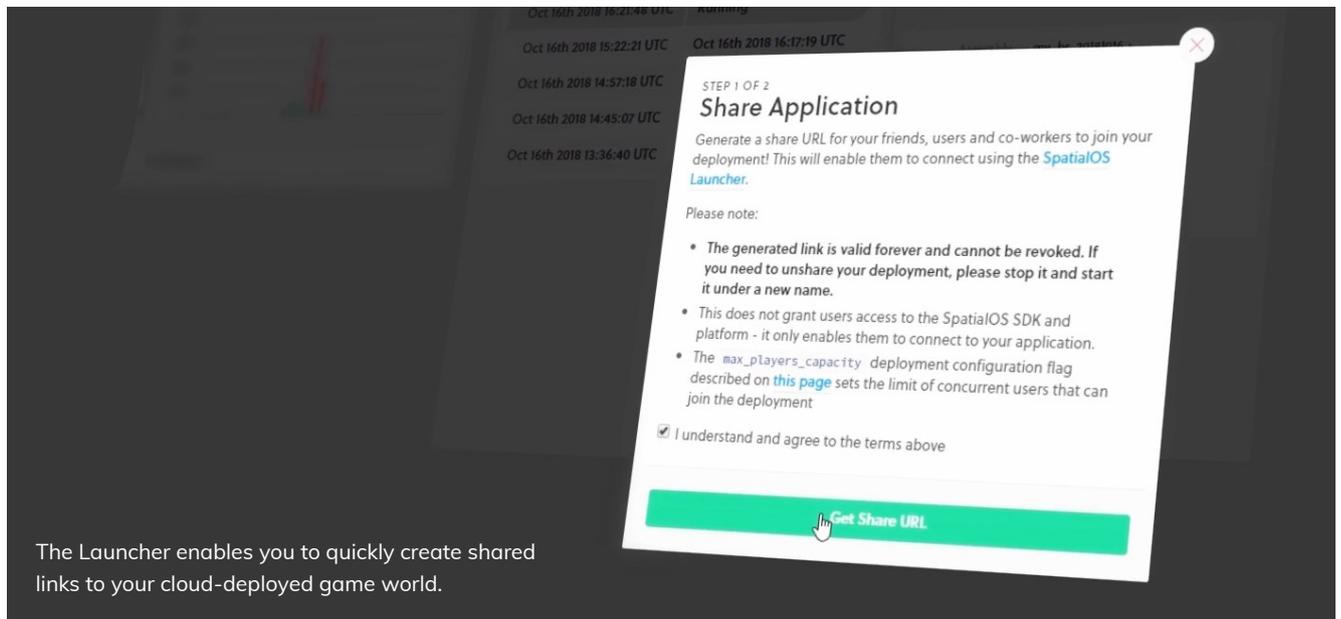
– Devon Sigurdson [Software Developer]

### Rapid comprehension of the game world

Another driving force behind the studio’s quick progress is being able to quickly understand what’s going on in its game world. The teams can then incorporate this understanding into their design and



The Inspector is a crucial part of the SpatialOS toolset, enabling developers to see a representation of the live backend of their game, including all entities and servers.



technical decisions, enabling them to get ahead of problems or optimisations to which they'd otherwise be blind.

For example, one tool is the Inspector. This provides a real-time top-down view of the game world, showing you exactly what the server is seeing. With this, developers can identify problems in online tests, both live and after the event, with loadable snapshots. "I was amazed the first time I saw this," says Summers. "The number of problems I could foresee it solving that were otherwise nearly impossible to diagnose – the possibilities were endless."

The studio uses the Inspector as a high-level view to see what people are doing when they're playing and to get a sense of the general state of the game. "We can spot problems that we might have seen months too late without this tool!" says Sarah Mainwaring, lead gameplay programmer. "It allows us to see in real time where all of our entities are – we can really drill down into what they're doing, what data they have, and a lot more that's not usually accessible on the server side. It's really easy for us to get a feel for what the whole population is doing, not just on the micro level."

Taylor agrees – he's found it particularly useful to locate quirks where the game isn't broken but is just being weird, which is hard to find in a normal game world without large-scale QA. "I love this thing for AI. I spend as much time watching AIs in this in our playtests as I do running around the game. For example, we had a random number generation issue that was causing all the AIs to randomly wander towards the right-hand side of our map. Finding that in-game is very tricky, but in the Inspector it was immediately obvious."

### Dreams into reality — fast

With this combination of live playtesting, effective debugging and easy comprehension of the game world, Flynn is confident that his team can make its ambitious vision into a reality – faster and with a much smaller team than before. "In the old days, multiplayer build tests would take days or even weeks to set up," said Flynn. "It's like sending a rocket to the moon; you're over-prepared because you only have one shot. Now, we get to go around and around, experiment and test, and try ambitious things because we know we can do it quickly."

**Visit [Improbable.io](https://improbable.io) to learn more about SpatialOS**