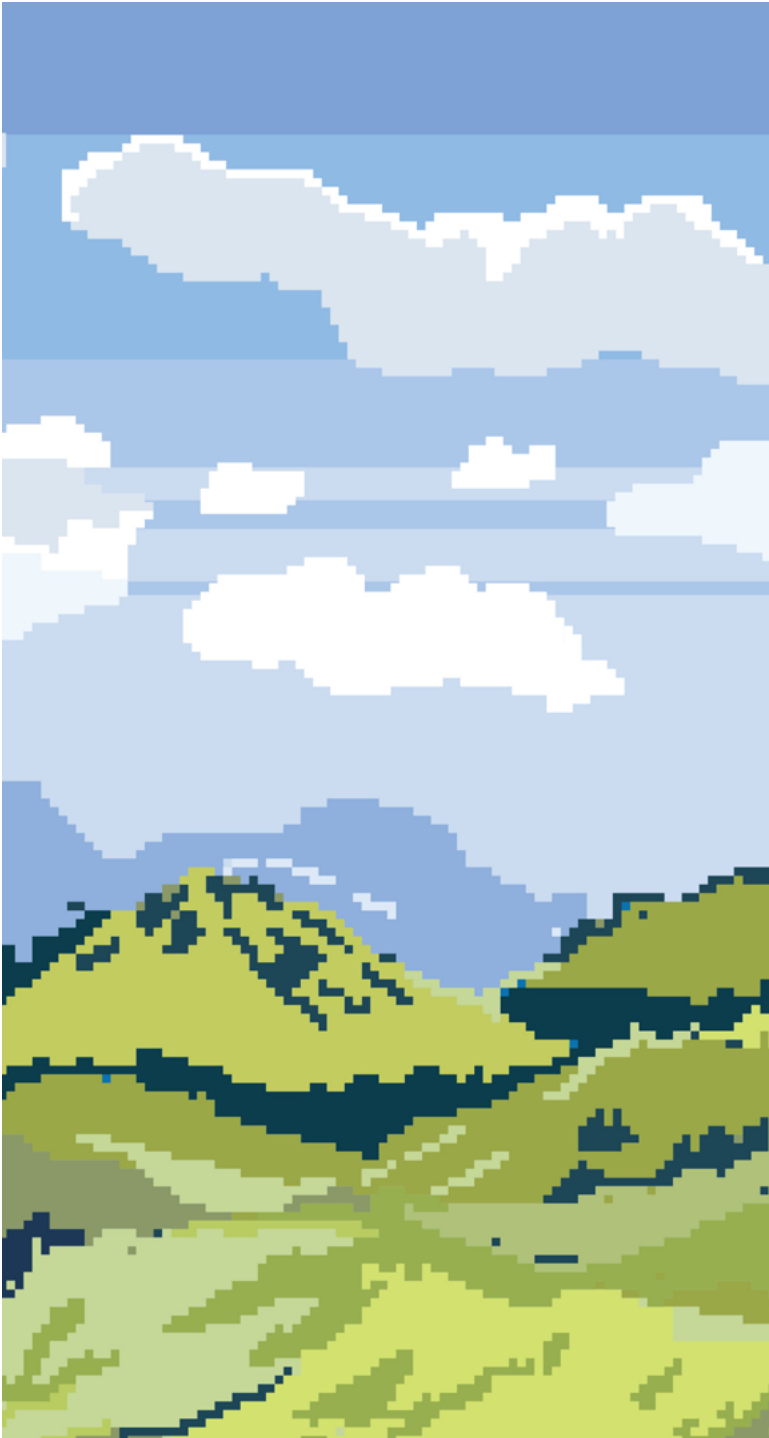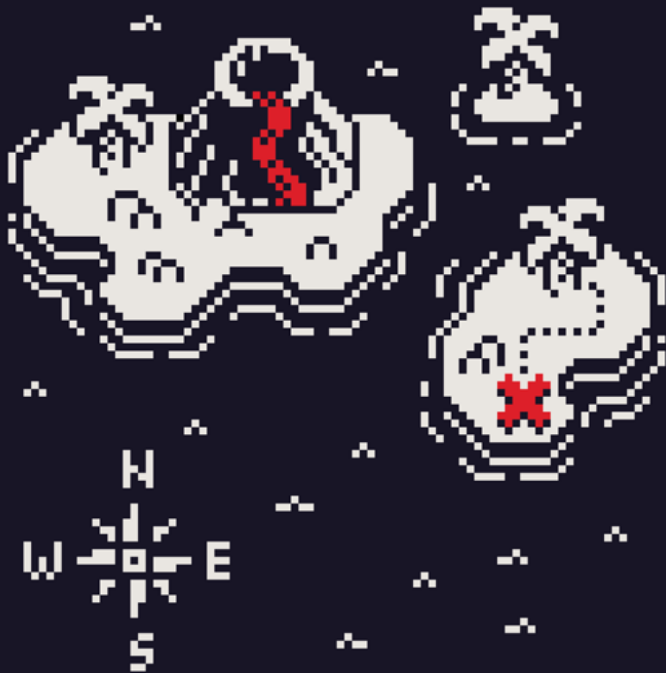# The True Costs of Creating, Launching and Operating a Game

unity

Introduction

So, you've decided to make a game. You've got some idea of what you want to create and what kind of people you need to do it. Ready, set, go! Seems easy, right?

We all know it's really not that simple or easy. If it were, every video game would be a success, but that's clearly not the case (ever play E.T. for Atari?). There's actually a lot more to making a game than just pulling your team together. Some considerations are obvious, while others tend to lurk unseen. We're going to focus on the latter.

The hidden costs of developing video games can be intimidating. Numerous decisions made during the development process can cascade back on one another in ways that amplify their negative effects. Setbacks can occur at any stage of a game's lifecycle, from the first twinkling of an idea to release – and beyond.

Even if these costs don't ravage your game's bottom line, they carry real consequences in terms of wasted time, damaged reputations, and the loss of other fulfilling or profitable work.

Yet these stumbling blocks don't have to stay unseen if you ask the right questions. By considering every phase of the development cycle, this e-book will help you unearth some of the hidden costs you might encounter – and, more importantly, to identify the questions you should ask yourself if and when they arise.

As you start your game development journey, we're going to guide you through the five milestones you need to hit in order to ensure you're on the right path to success.

Let the journey begin!

Concept/prototype

1

Inspiration can arrive at any time, but without knowing the best way to transform this into a reality, it can fizzle out just as fast – and everyone from first-time game designers to industry veterans can run into problems right out of the gate.

Meet Chris. He's a gamedev who's just beginning his quest. Chris has a great idea, but he's missing a map – what will he need to find success on his journey?

Key considerations

"The Tilemap and spritesheet support have saved us hours and hours of work every week… if we didn't have those 2D features, we simply couldn't have made a game of this scale."

– EpicHouse Games cofounder and developer Tim van het Kaar

Do my tools let my team do their work efficiently, regardless of location?

Remote work is expected to continue trending upward in the coming years, so flexible solutions for game development are a must.

EpicHouse Games cofounders Tim van het Kaar and Joshua Boren are a perfect example of how a team can collaborate remotely, having met through a web forum. They knew it would be fun to work together and had a solid vision for their "world-shifting" 2D platformer, *Phased*, but they lived half a world apart: one in the U.S., the other in the Netherlands. And with one of them going to school and the other working a full-time job, time was a precious commodity.

They needed tools that allowed them to determine very quickly whether their ideas were viable. By choosing software that helped them to ideate and iterate quickly, EpicHouse estimates they saved 10–20 hours a week – which may not sound like a lot, until you consider that the team was just two people with busy lives outside of game development.

Using Tilemap, "we soon realized that we could extend [its tools] with brushes and features that matched our own workflow, and it was making it very easy to prototype and iterate on new levels," van het Kaar says. "That meant we could just try out a feature in a separate scene and then once everything worked, put it in the main level, and we didn't have to break things before th were ready."

More importantly, choosing tools that worked well in networked situations meant that the creators did not have to physically be together to work on *Phased*. In an industry that is increasing its reliance on remote teams, this can be the difference between a published game and a promising concept that dies on the vine.

4

Is my game targeting the correct platform?

Choosing between an established audience or breaking new ground can be a tricky decision. The answer here is not always easily apparent; it's great if it makes sense to be everywhere in today's cross-play-obsessed, multiplatform environment, but that's not always the case. Are you making a game that's for mobile or PC? Xbox or Playstation? Nintendo Switch? If you plan beforehand, it could make it easier to port to other platforms when the time comes.

Sometimes you may not know until some preproduction work has already been done. When Game Insight first dreamt up *Guns of Boom*, it had an eye on the PC and console market – a logical move for a 4v4 multiplayer shooter. But along the way to its alpha release, the studio decided that the absence of high-quality, competitive team shooters on mobile meant it might be worth switching gears to concentrate on iOS and Android instead.

There was a risk involved in the change – especially since it meant switching engines as well – but the *Guns of Boom* team found it was well worth the effort. The game saw five million downloads in just its first two weeks, and is now a staple of the mobile esports community.
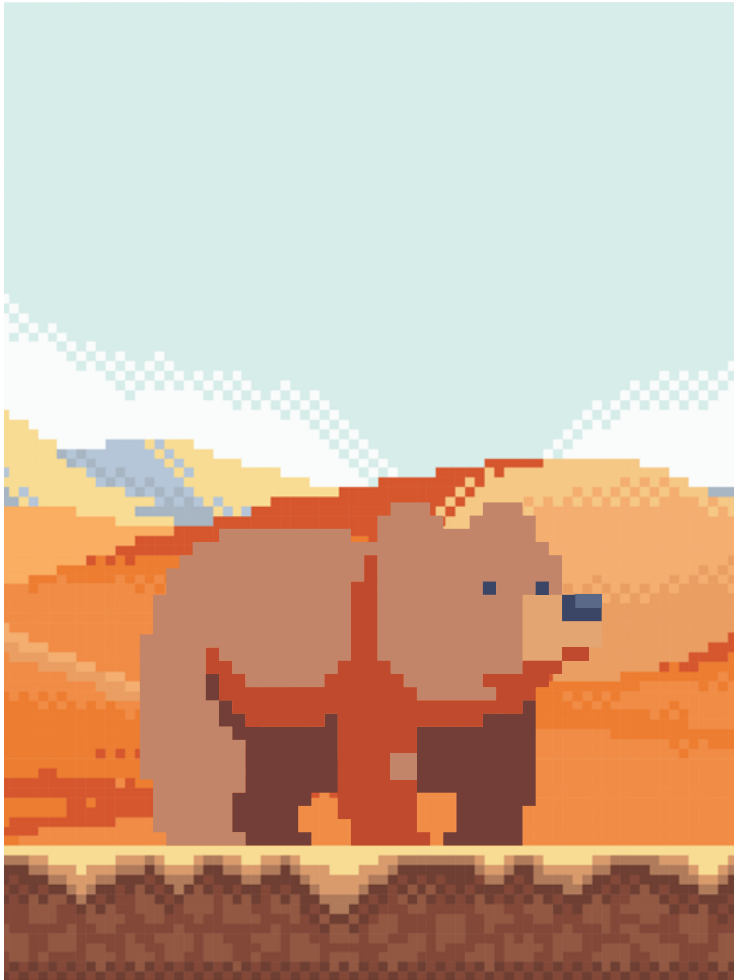
Help your game hit a bullseye by choosing your platform strategically, like Game Insight's savvy release of *Guns of Boom* for mobile.

5

GAMEDEV PHASE:

Preproduction

2

Most of the big decisions get made in preproduction. Each one is a risk – the wrong choice can cost you production time down the road, but being paralyzed by indecision will get you nowhere.

Chris is so worried about the dangers that he almost didn't set out on his quest at all! Roadblocks are inevitable – to succeed, he'll need to stay nimble and seek a better path when problems arrive.

Key considerations

**Are there existing resources we can use to
help with preproduction builds?**

Changing directions when your game is already underway can be daunting. The clear hazard is wiping out the work your team has already done, and the time spent recovering – which can lead to dreaded launch delays.

It's an even trickier calculus when external stakeholders are involved, as was the case for Ludia when developing *Jurassic World Alive*. The pressure came from managing one of the world's most recognizable IPs, along with a large number of preregistrations. The studio put in six months of work using its own in-house tech before realizing that they needed to switch engines for something better suited to location-based AR.

Instead of holding back progress, the developers found that this change helped claw back some of the time spent on the initial work, thanks to things like improved rendering speed: a matter of hours compared to half a day for the previous prototype. On top of that, the team's workflows improved as people became empowered to make contributions that previously required them to wait for colleagues to assist.

Making big changes carries the risk of sinking the initial effort's costs in terms of time and money, but finding the best possible tech to create *Jurassic World Alive* proved to be worth it since it allowed the game to release on time, earning $25 million in revenue in less than four months.

At first, Chris tried to do everything himself – even when it wasn't his strongest skillset (like, er, game art). Creating something awesome doesn't mean doing everything from scratch – sometimes the best way forward is the path of least resistance.

7

Try a lot, and see what sticks. At first, Chris thought this 8-bit tree owl looked dumb – but it ended up being the cornerstone of his game's storytelling! Hoot, hoot!

Are there existing resources we can use to help with preproduction builds?

In the preproduction phase, rapid iteration can be a huge asset. Determining what works and what can be discarded as quickly as possible helps you save time and money, but only if you're able to limit the work needed to create each new version.

Gameloft does this as part of its standard preproduction procedures. The team puts together a working version of a game's core concepts as quickly as possible, decides whether or not it's viable, and then moves on to something else if it isn't. In just one year, they completed over 30 projects and saw 12 games greenlit for production.

Since the Gameloft team never really knows what kind of initial builds it will be assembling from month to month, it needs a source for game-ready assets it can rely on, which in their case means purchasing existing assets from an online marketplace.

"When you just need models or controllers or something similar, the research team can go to the Asset Store to create the first look and feel, and it can save them up to six months of development time," says Gameloft Montreal's director of gameplay Renaud Forestié, noting that his studio has found and used everything from characters, environments, and music to billing add-ons and localization tools.

8

Production

3

Once production is underway, most of your crucial decisions have been made – but certainly not all of them.

Imagine Chris's surprise when he discovers that issues he'd thought were done and dealt with have come back to haunt him.

Key considerations

10
simulations

165
million playthroughs

600
hours of development

Used just 10 simulations to efficiently run 165 million playthroughs, saving at least $80,000 and 600 hours of development time along the way (*Death Carnival*, Furyion Games)
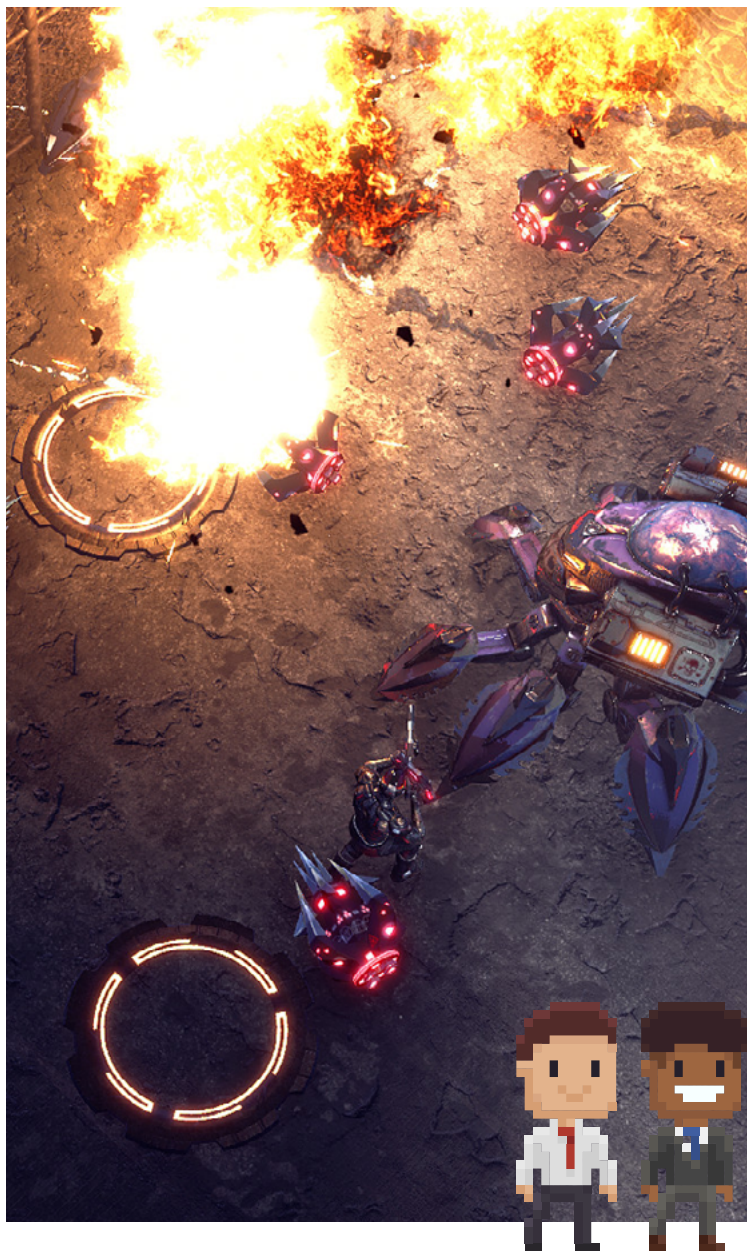
EN GARDE!

Can we capitalize on proven strengths in new ways?

Unless your goal is specifically to push yourself in a new direction and step outside your comfort zone, there's no need to reinvent the wheel. Sometimes the recipe for success is already there and only needs an additional ingredient or two.

That doesn't necessarily mean a sequel, but rather reframing what works in a new way to broaden your audience. When Flyclops was considering a follow-up to *Dominos!*, it didn't want to stray too far from what made it the #1 dominoes game in the App Store. With a proven revenue stream in hand, the question became how to bring in more money and heighten retention without jeopardizing it.

Flyclops leaned into what it knew for *Dominoes Gold*, but with an esports twist and a partnership with Skillz. In just one week, a single developer integrated an SDK into the game that added new features like prizes, large-scale tournaments, and leagues, while maintaining the core gameplay that made the original game a hit.

"We're actually showing fewer third-party ads and making more money," Flyclops coowner Dave Martorana said after seeing an increase in ARPDAU. The new game also had a D1 retention rate of 41%, proving that the studio's decision to build on top of what was already working was a wise one.
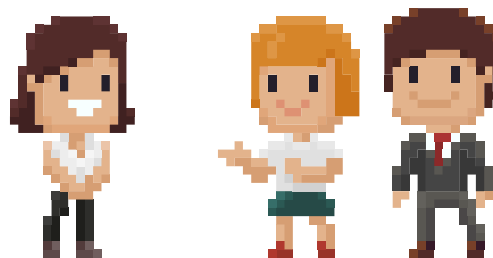
Is my team spending its time on the right things?

The important process of playtesting and balancing can also bog down any game prior to launch. The more complicated a game's systems are, the more critical the balancing act becomes.

For Furyion Games's top-down multiplayer online shooter *Death Carnival*, the game's deep weapons systems made it a priority to get the game balance just right. The developers found that each weapon typically took five test iterations to balance by hand, each one averaging six hours from build to observation and feedback to tweaking.

Automation turned out to be the key. A third-party service allowed the studio to use just 10 simulations to efficiently run through 165 million playthroughs, trying out every possible combination available to players. Furyion estimates it saved at least $80,000 and 600 hours of development time along the way.

Less time spent balancing also freed Furyion's programmers and artists to spend more time leading up to release on polish, getting additional details just the way they envisioned. Those types of prelaunch improvements don't necessarily translate directly to the bottom line, but they're the kind that players and critics notice.

There is constant pressure to produce higher-quality content in ever-shortening timelines. Look for opportunities to automate repetitive tasks so your team can focus on creating the best game possible.

Launch

4

Launch day is a huge milestone – but it's not the end of your game development journey. With your game in players' hands, the next challenge is making sure they can play it with as little friction as possible.

Chris just slew a big gamedev dragon – getting to launch day – but he's still got a lot of work to do before he gets to collect his treasure.

Key considerations

Is our game optimized for the widest range of player experiences?

Creating a console game means you'll know exactly what kind of hardware will be used by players, and that gives you a good deal of certainty about the experience people will have. In contrast, developing for platforms like PC and mobile – particularly for Android – may as well be the Wild West.

Creating an Android game means taking into account the fact that the audience will play it using a wide range of devices, all at different points on the scales of power and capability. The last thing you want is to compromise on quality just to get a game to run properly, but coding for the full spectrum of Android devices can eat away at valuable time and resources.

That dilemma popped up for Yodo1 during the creation of *Rodeo Stampede*. Because it's exactly the kind of game that appeals to casual gamers who don't always have the latest phones, the developers needed to keep the lowest-possible range of devices in mind during development.

Even though Yodo1 had previously crafted its own solution to optimize for all Android hardware, doing so required a lot of work on both the game and its servers. The studio discovered it could save time and money by utilizing a third-party tool for device optimization called LiveTune, reducing load times across all devices to 10 seconds or less. Not coincidentally, Day 1 retention rates as low as 29% on certain low-end Android handsets saw a boost to rates of between 40–45% across the board.

Yodo1 VP of global publishing Vincent Diao explains: "It took us around a week to get everything fully integrated and upgraded. It would have cost us hundreds of hours of work time with our previous solution, and we still wouldn't have been able to achieve the kind of results we have with LiveTune."

Welcome to the Wild West, where big risks mean big rewards.

Be ready for anything like the team behind *Apex Legends*. I mean, a Day 1 rush is a good problem to have, right? Just make sure you're armed with the right resources to solve it.

Are we prepared if launch proves much more successful than anticipated?

This sounds like the kind of problem any developer would love to have. You've made projections and have reasonable expectations about how many people to expect at launch, and your game ends up shattering all of them. Nothing but good times ahead, right?

Not necessarily. If you're not ready for an unexpected crush of players, it's possible that everyone playing will have a bad experience. And sometimes success comes at a level that's simply impossible to predict, as was the case with *Apex Legends*. Respawn's surprise battle royale release proved so popular that 50 million people downloaded it within the first 24 days, and concurrent server loads reached eye-popping levels: two million players at once about a week after launch – more than 10 times the most optimistic studio estimates.

Trying to figure out the maximum server capacity needed is a difficult exercise for any multiplayer online game. Respawn's solution didn't involve nailing the right numbers for server capacity, but instead using hybrid cloud technology to be able to scale right past their own projections when needed.

The dev team went into the *Apex Legends* launch believing it could handle uncertainty, and they were right. Pulling it off protected Respawn's reputation for providing a reliable multiplayer experience, despite achieving the kind of enormous popularity that might have crushed other games.
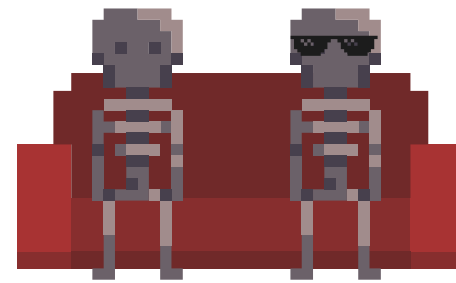
Live ops

5

Your work isn't done after launch – you still need to take care of potential problems for players and anticipate their needs.

Chris's game is out and performing well – but live ops offers new, next-level challenges every day.

Key considerations

Am I using the best tools to help players communicate?

In-game channels to keep players in touch are crucial to many different genres of games in the current market. Chat was once a nice extra but has now become an expectation, with studies highlighting its importance in helping players win at online games and even improving their ability to form social bonds.

Sometimes you won't even need to dig into any data to see when in-game communication has to be improved because the lack becomes obvious. In the first part of 2017, it wasn't hard to find people playing *Rainbow Six Siege* posting about their poor experiences with in-game chat and looking for help from within the community. Complaints about being unable to hear other players sent an undeniable message to Ubisoft that a change was needed.

Rainbow Six Siege's custom-built, peer-to-peer voice chat software simply wasn't as reliable as it needed to be – an even more significant consideration given the game's status as an esports staple. Ubisoft turned to an existing product from Vivox and found a level of stability that suited everyone from casual to pro gamers, and without having to take apart too much of what they had built.

"When we were evaluating solutions for our voice needs, Vivox immediately jumped out with their proven stability and notable partner list," explained Alexandre Remy, brand director for *Rainbow Six Siege*. "What surprised us was the ease of the integration and how the Vivox tech just worked from the first day of our testing."
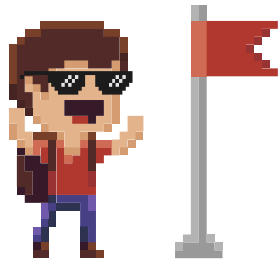

16

Can I identify the most important and valuable players?

Player engagement has proven to be linked to behaviors like a willingness to upgrade to premium versions of games and make in-app purchases. However, your ability to identify and target the right people is just as important. Think of it as a virtuous cycle: Finding the gamers who are most engaged, giving them reasons to keep playing, and improving their likelihood of spending money to increase their engagement.

Without insight into which players are the true backbone of your ongoing success, you face the prospect of creating content without a purpose. Mindstorm Studios wrestled with this consideration when it released the mobile strategy title *War Incorporated*. After four million downloads across iOS and Android, there was no question it had found an audience, but sifting through the data was a hurdle for a relatively small team.

By choosing an analytics solution that met their specific needs, the studio responded to the demand for new content and engagement almost in real-time. The *War Incorporated* team also learned more about how players differ by region, such as how North American players tended to make their first IAPs sooner than their European counterparts – the kind of realization that can directly improve a game's profitability.

17

We're here to help throughout the game lifecycle.

Game development is a marathon, not a sprint, but if you keep these five checkpoints in mind you'll be on the right path to success:

1. **Concept/prototype** – What are we building, and for whom?

2. **Preproduction** – How are we going to do it?

3. **Production** – Are we focused on the right things?

4. **Launch** – Are we ready for success?

5. **Live ops** – What should we adjust?

The obvious expenses of creating, launching and operating a game can be formidable on their own, but the additional costs in time and missed opportunities are no less relevant – they're just harder to anticipate. Unity is here to help you with tools and services to make sure you are supported throughout your game development journey, from concept to commercialization.

If you're ready to dive in deeper, you can get started with Unity Pro today or talk to one of our experts to learn all the ways we can assist you.