

£3
ALL FORMATS

Wireframe

LIFTING THE LID ON VIDEO GAMES



Bullet hell

Balletic chaos in
My Friend Pedro

Crash Bandicoot

How to remake a
console classic

Breaking through

BAFTA Brits explain
their methods

OUTFOXED

Why Tunic's shaping up to be
the Dark Souls of cute

Issue 5 £3
wfmag.cc



Subscribe today

12 weeks for £12*



Visit: wfmag.cc/12weeks to order

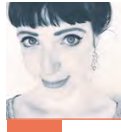
* UK Price. 6 issue introductory offer

Game-making's sense of loss

Following the last, wheezy gasps of 2018, I've been thinking about endings. Fewer players seem to 'finish' titles than ever before. There's a growing expectation that all games should be updated with new content after release. Pete Hines promised *Fallout 76* would run forever. There was a sigh of relief when Obsidian promised *The Outer Worlds* wouldn't be that big.

Developers have issues with endings. Designers, I'm learning, are the most doomed of the lot of us, cursed by an unkind god to see only the flaws in their games, no matter how rapturously the title is received. But all devs find it difficult to let go: making games is such a life-consuming delight that when it's over we don't know what to do with ourselves. I wonder if that's – partly – why there's such a fear of committing to release dates, aside from lacking confidence in your production estimates. When your game's out, it's finished – isn't it? The whole disastrous affair? And then you move on and do it all again?

As anyone who's launched a game before knows, this isn't how it works. You have this gigantic hopeful bubble which pops and, well, sometimes you have a buzz for a week because everybody loves it and you're suddenly seeing a huge spike in sales and a burst of positive reviews. But even that isn't the end – it often takes months to receive your first payment from distributors, and by then the glow of launch



LOTTIE BEVAN

Lottie Bevan is a producer and co-founder of the experimental narrative microstudio Weather Factory, known for its Lovecraftian card game, *Cultist Simulator*. Before founding Weather Factory, she was producer at Failbetter Games where she worked on *Fallen London*, *Sunless Sea*, *Zubmariner*, and *Sunless Skies*.

week has worn to a duller patina of user reviews, support tickets, bug fixes, and the sense I imagine new parents feel when they step outside the hospital with their baby and just, well, go home?

A wise gentleman I used to work with once gave me a pep talk about post-launch blues, that gloom which descends on developers in the aftermath of release. He said: 'I spend years working on a game, and then marketing comes in and turns it into something else. They turn it this way and that so it catches the light. And then people respond to it in ways I didn't intend, and people have opinions, and form personal experiences that have nothing to do with me. Something that was mine has gone out into the world, and it isn't mine anymore.'

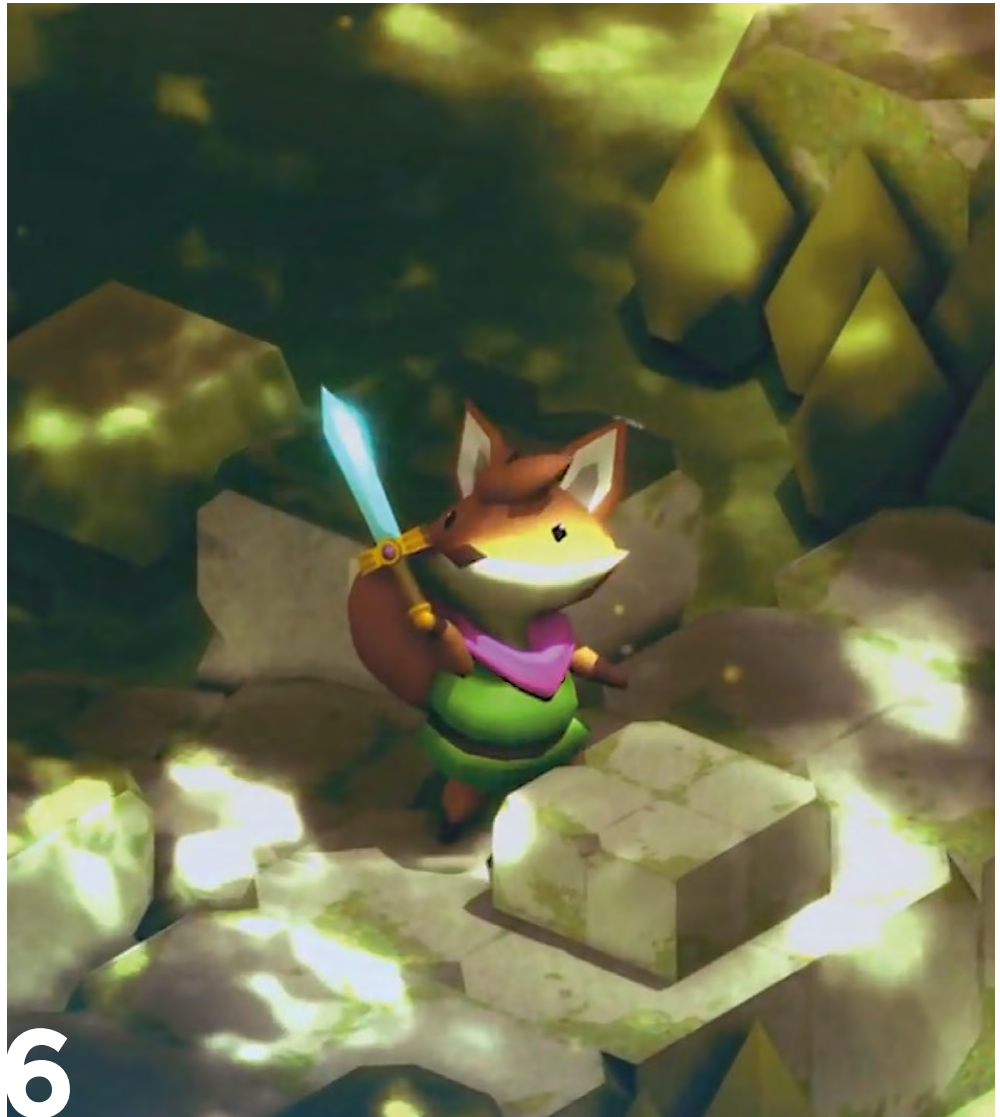
This is the closest I think developers get to an ending. A game after launch is a consumable, eaten by a thousand mouths. Developers eventually stop developing, but their game slots into storefronts for people to play. It's likely that never before has that game had so many people playing it, which is its *raison d'être*.

This is a wonderful thing. There's a sense of loss in game-making, but your loss is a player's gain, and it all balances out in a vast, impersonal ecosystem. At this chilly time of the year, when one calendar makes way for another, take solace in this. Toast yourself, the games that aren't yours anymore, and the games you'll lose in the future. It's a bittersweet, but marvellous, destiny. 🍷

#05



Contents



Attract mode

Interface



- 06. Tunic**
Creator Andrew Shouldice on the art of keeping secrets
- 10. My Friend Pedro**
Looking ahead to a manic, balletic platform shooter
- 12. Mythic Ocean**
An immersive undersea adventure about gods and conversation
- 16. Incoming**
Mystical eagles, lonely monsters and futuristic cab drivers

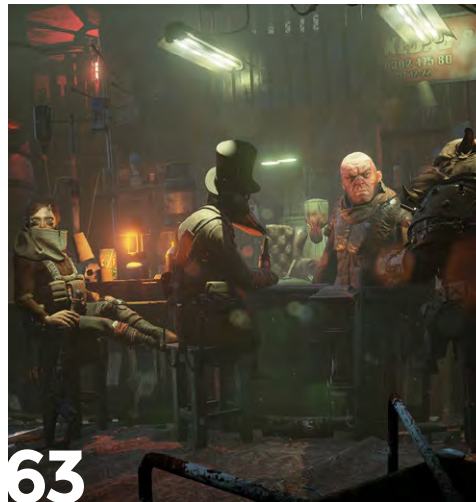
- 18. Breakthrough Brits**
Two BAFTA-winning game devs talk about their industry paths
- 22. Suda at 51**
Suda51 talks *No More Heroes*, Kafka and wrestling
- 24. Interactive: Transmission**
Back to the eighties with a soothing drive-'em-up from a solo dev
- 44. Crash Bandicoot remade**
Vicarious Visions on the console classic's revival



44



18



63

Toolbox

28. Design principles

Why being prepared to deviate from established design ideas is essential

30. CityCraft

Three ways of making a truly immersive video game city

32. Source Code

How to create your own arcade-style high-score table

34. Improving your Unity FPS

Part two of our guide to making a first-person shooter

Rated

56. Smash Bros. Ultimate

More of the same, but another punchy delight on Switch

58. Ashen

Anonymous matchmaking enlivens a solid action RPG

61. Desert Child

Relax to the sensation of racing hoverbikes on Mars

63. Mutant Year Zero

Think of it as XCOM, but with post-apocalyptic ducks

WELCOME

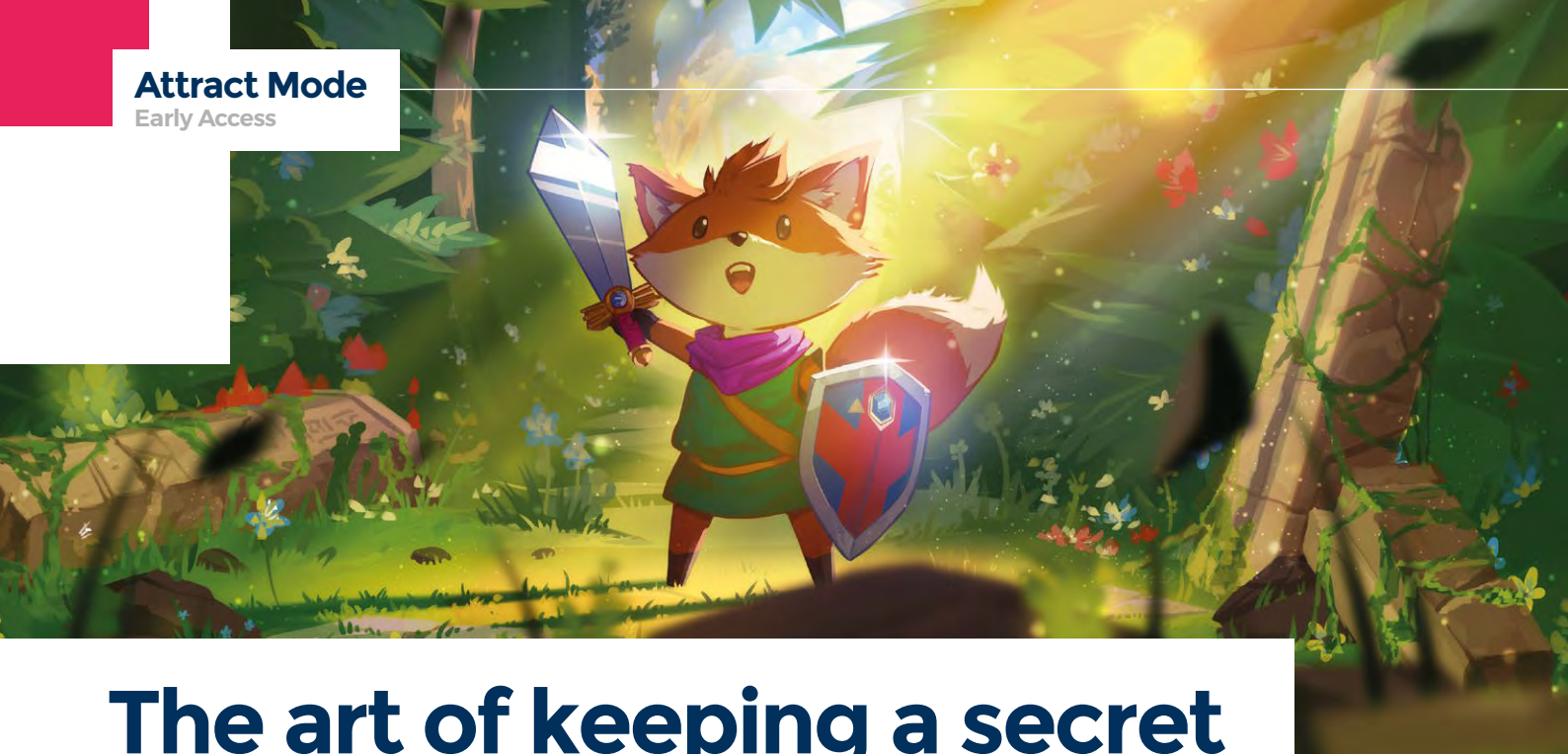
Christmas may be a distant memory now, but I've still fond memories of how I spent most of it: lying on the sofa, playing Konami's original *Castlevania* from 1986. For some reason, I've long used the festive break as a chance to load up an old game I've either never played, or, in *Castlevania's* case, only tinkered with briefly.

It goes without saying that *Castlevania* remains a classic; not just because of its influence, but because its Gothic platforming action still feels so fresh. What's really striking, though, is how much atmosphere *Castlevania's* graphic designers managed to wring out of the Nintendo Entertainment System's meagre hardware. Even today, *Castlevania's* setting has a pleasingly dank, grimy feel, while the protagonist, with his determined march and lashing whip, packs a lot of personality into just a few frames of animation.

As *Transmission* developer Jon Dudley explains on page 24, making indie games is partly about doing the most with the time and resources you have. It's why classics like *Castlevania* are so important: not only are they fun to play, but they're also worth studying to see how designers can use a few kilobytes of memory and a tiny colour palette to such a dazzling effect.

Ryan Lambie
Editor





The art of keeping a secret

Developer Andrew Shouldice talks to Wireframe about *Tunic*, the adventure of a tiny fox in a big world

▲ Erica June's hand-painted key art does much to convey *Tunic*'s atmosphere in a single image.

Keeping secrets isn't easy these days. Thanks to the internet, leaked details of a major franchise sequel can spread around the world within seconds.

Videos of a game's ending can appear on YouTube or Twitch within hours of release. For now, though, isometric adventure *Tunic* has an enticing air of mystery surrounding it. Its lush world is full of ancient ruins and hidden pathways; in-game text is written in an unintelligible script, not unlike the exotic language of Fumito Ueda's classic, *Ico*; even the identity of its protagonist, a plucky fox in a green tunic, is currently unclear.

Like its fox, who boldly lopes into battle with his tail bobbing along behind him, *Tunic* can maintain its sense of mystery because it's so small. For the past four years, Canadian developer Andrew Shouldice has been working on the game largely by himself, with occasional YouTube dev diaries and Twitter updates offering only tantalising glimpses of his work

in progress. Indeed, the sense of mystery is something that has formed a core of *Tunic* from its inception.

"The concept of the game was really an old idea that's been in my head for a long time," Shouldice tells us. "But the more I think about it, the more I realise what a vague idea it was. It was this feeling – this type of experience, where you feel like you're a stranger in a strange land. You're entering a place where you don't belong. As this vague idea turns into an actual game, we're still managing to hit that feeling of a sprawling, unknowable world with a bunch of strange rules."

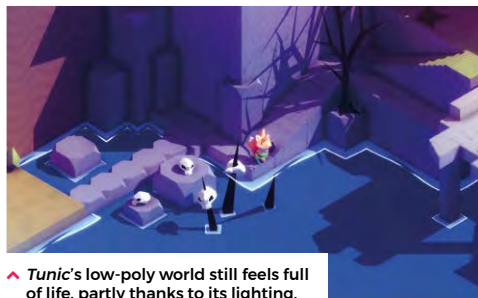
It's a description that vaguely recalls Shigeru Miyamoto's now legendary inspiration for *The Legend of Zelda*, in which he attempted to capture the sensation of exploring the Kyoto countryside as a child. And, as you've probably noticed, *Tunic* has more than a touch of *Zelda* about it; the fox, with his sword and shield, is a hero firmly in the mould of Link, while the lock-on combat system handles similarly to the one in *Ocarina of Time*.

TREASURE HUNT

But while *Tunic* might positively invite comparisons to *Zelda*, it has its own twists on the Miyamoto formula, from the isometric perspective to an overarching air of self-awareness that reminds us a little of *Fez*. Just as Phil Fish's indie masterpiece constantly reminded us of the artificiality of its sumptuous 2D world, so *Tunic* throws in little postmodern ideas. Open up a treasure chest, and you'll



▲ Hunt in the right places, and you'll find pages that look like they're torn from a video game manual circa the early nineties.



▲ *Tunic*'s low-poly world still feels full of life, partly thanks to its lighting.



^ The player's character model has been reworked at least once, Shouldice says, from an angular early iteration to this smoother version.



occasionally uncover a page torn from a video game manual – a manual written for *Tunic* itself.

“One of the things I want people to have is the feeling of sitting down and playing a video game that you don't understand the boundaries of,” Shouldice explains.

“I think of myself playing a video game aged five, leafing through the manual, not understanding any of it, bumbling around and not really understanding what the rules of video games are, let alone what the rules of this video game are. Part of that, I think, is having text that feels like it's not made for you.”

Appropriately, *Tunic* has been a journey of discovery for Shouldice himself. Back in 2014, he quit his job as a developer at Nova Scotia's Silverback Productions to work on the game, then called *Secret Legend*. Armed with a degree in computer science and experience of shipping games, Shouldice already had plenty of skills to fall back on, but going indie has nevertheless presented a number of challenges – including a crash course in 3D animation.

“It's that feeling of a sprawling world with a bunch of strange rules”

“I did a lot of 2D animation in my old job, just sliding stuff around,” Shouldice says. “I remember the first time I used Unity's animation system, and I tried to make a ball bounce, and I was like, ‘This is awful. This is terrible. Burn it! I'm never doing this again.’ And then of course I got used to it.

So really, it's been over the course of this project that I've learned how to do more 3D modelling and animation.”

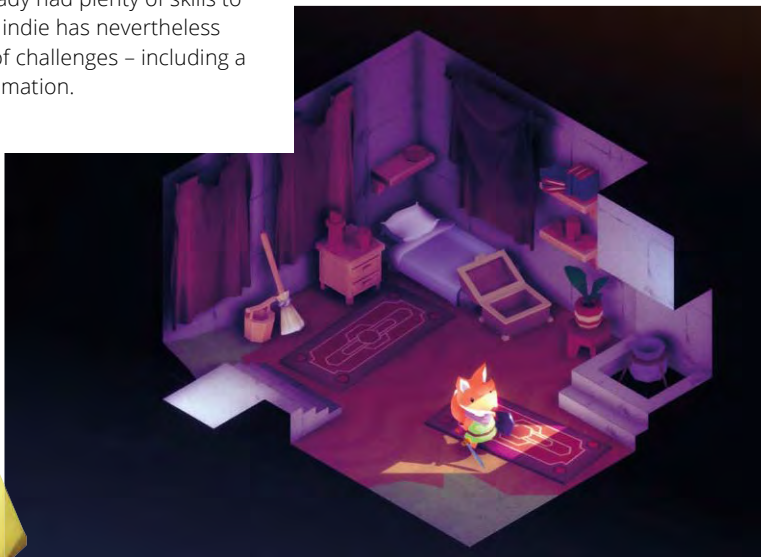
Cannily, Shouldice designed *Tunic's* world around his own limitations: he found ways of constructing assets like bushes and tufts of grass from as few vertices as he could, resulting in the low-poly style that makes *Tunic* so eye-catching.

“I wanted to keep everything simple originally, not being especially good at doing modelling →



^ Inspired by his childhood experiences of playing video games, *Tunic* is clearly a personal work for Andrew Shouldice.

> The isometric viewpoint results in some captivating little interiors like this one.



POINTS OF VIEW

One of the most appealing aspects of *Tunic's* design is its isometric perspective, which provides a sense of depth and, fitting with the theme, a sense of secrets hiding around corners. “I like the isometric perspective for a number of reasons,” Shouldice says. “The world becomes a static image, sort of... it means there's this old-school illusion, where it's almost a picture that you're scrolling around, like a top-down adventure game. There are a bunch of artsy-fartsy reasons why I really like it, but it's very pleasant. Also, *Monument Valley* is the most beautiful thing that has ever been created by human hands. That's a fact.”





▲ *Tunic's* opening area is full of places that are tantalisingly out of reach at first.

RIGHT ON TARGET

While *Zelda's* a clear influence on *Tunic*, you'll also find traces of *Dark Souls* in its combat. Irrespective of the game's cuddly aesthetic, beating some of *Tunic's* enemies, from hog-like warriors to mechanical bosses, will require memorisation of attack patterns and precisely-timed dodges. "A combat sequence [in *Tunic*] can vary in a much more different way than a *Zelda* game could," Finji's Harris Foster says. "*Dark Souls* games are hard until we start learning the patterns, and then the guy that seemed really hard before is now quite easy because we know how he works. So I think technique is going to be a big part of that."

and animation," Shouldice says. "I charted out where vertices should go on graph paper, because that was easier for me to conceptualise what the character should look like. Trying to do it with as few vertices as possible so I didn't have that much geometry to deal with."

Rather than wrestle with high-poly character models, Shouldice has instead focused on getting little animated details just right – the tactile way the grass bends as the player moves through it; the puff of red hair that bounces on the fox's head as he runs.

"One of my favourite things to do is just go in and add polish – little tiny things that might be a fraction of a second, but you see them a lot. I feel like that's the maximum pay-off – I can tweak some curves just here, and then all of a sudden, every time the player moves around, they see this little bounce. That's a good cost-value trade-off sort of thing."

Lighting is another detail that adds atmosphere and depth to Shouldice's low-poly world: flecks of light shimmer along the ground in a woodland, suggesting a canopy of trees above, while on the beach, grey cliffs are bathed in diffuse sunshine.

"Very early on, I knew I wanted to have this soft light, where if a surface is strongly illuminated, the object itself isn't just bright, but other walls near it are bright," Shouldice says. "Like when sunshine falls on a hardwood floor, the room is filled with this light golden glow. It's really pleasant and makes the whole place feel bright and shiny. Then when you go through a forest, that feel is characterised for me by that dappled light on the floor. It's trying to get to the

core of, 'what can I do to communicate a feeling [without] a team of artists working on it?' How can I deliver that emotional payload with an abstracted representation of these things?"

Not that Shouldice has attempted to tackle everything in *Tunic's* development himself. For sound design, he turned to Power Up Audio, the Vancouver-based team who recently won an award for *Celeste*; musician Terence Lee – also known as Lifeformed – is behind the ambient soundtrack. He's also found a valuable ally in Finji, the developer and publisher that distributed such games as *Night in the Woods* and the forthcoming *Overland*.

DISCOVERY

"It was just this natural collaboration that turned into something more official as we got to know one another," Shouldice says. "If this game is going to have a publisher, then Finji is going to be publishing it – it just makes sense. That's when I met Harris [Foster, Finji's community manager], who is helping out with all kinds of stuff, like running shows and setting up interviews and managing our community."

It's a partnership that quickly bore fruit; in June 2018, *Tunic* was one of the games prominently displayed in Microsoft's E3 Indie showcase. Now a console exclusive for Xbox One, Shouldice's little adventure game has a wider potential audience than ever.

Says Shouldice, "It's the sort of trajectory that, if you'd told me this five years ago, I'd be like, 'That's the dream, isn't it?' Ha. Anyway... I started posting Vines – RIP Vine – on Twitter a few months after development started, and people just picked up on it and got enthusiastic. It fuels me to this day, really, that there are actually people who are excited about this."

▼ *Tunic's* chests: some are out in the open, others are hidden in deviously obscure places.





▲ "It was either use right stick in order to do facing, or have some sort of lock-on system," Shouldice says of the combat system. He ultimately chose the latter.

Despite the growing anticipation surrounding *Tunic*, Shouldice remains intent on retaining his game's air of mystery. He draws the distinction between the feeling of solving a clue in a point-and-click adventure and that of stumbling on a hidden secret that was never really meant to be found.

"It's the satisfaction you get when you get to the end of a chapter in a book and there's a big cliffhanger or the murderer is revealed," Shouldice says. "That's cool, but on the other end of the spectrum, there's something that you actually discovered: a new fact about the world that was deeply hidden. It's like reading that mystery novel and being like, 'That was good', and then realising that if you take a razor blade and cut the page down the middle, there's a secret page in between there that has something more on it. It feels like you really weren't meant to find it."

"One of the things I've learned is that building content takes a long time"

WORLD BUILDING

Creating those hidden mysteries, however, takes time. While *Tunic's* core is pretty much done, Shouldice still has several months – perhaps a year or more – of development before the game is finished.

"One of the things I've learned is [that] building content takes a long time," Shouldice

tells us. "The feel of a game, the mechanics – you can fine-tune and polish those a lot. But giving the player something to do that is compelling, and hand-crafted, and feels new, takes a lot of time; going back and trashing stuff and reiterating and redoing it. There's no specified release date at this point, but that's the thing that I've been spending a lot of time doing –

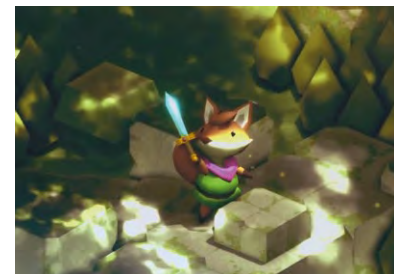
basically building new areas for the player to explore."

The question remains, though, whether *Tunic* will be able to keep its aura, especially once it's released

and all those content creators start creating their YouTube and Twitch videos. "Perhaps it's inspirational," Shouldice admits. "I don't know if people are going to come away with that feeling, but I desperately hope that it's the case."

Finji's Harris Foster, on the other hand, seems confident that we'll all be talking about *Tunic's* secrets for some time to come.

"It has a feeling of mystery that I haven't felt in 15 years," Foster enthuses. "We have the internet now, which makes these things way easier, but *Tunic* makes you want to get out there and talk to your friends about it. I think it's a game people will want to scratch their heads over, out loud, with other people." 🗨



▲ You start the game waving a stick, but you'll soon find a sword to keep enemies at bay.





▲ A shaven-headed Jason Statham-type in the Flash game, the antihero in *My Friend Pedro*, is now a mysterious, masked avenger.

Bullet ballet

My Friend Pedro designer Victor Ågren tells us all about his forthcoming platform-shooter

Info

GENRE
Action platformer

FORMAT
PC / SWITCH

DEVELOPER
DeadToast
Entertainment

PUBLISHER
Devolver Digital

RELEASE
TBA 2019

F or those of us who can barely walk up a flight of stairs without doing something clumsy, movies like *The Matrix* or John Woo's *The Killer* offer an additional layer of fantasy: not only being a hard-as-nails hero, but also being able to do somersaults in slow-motion while also wielding a pair of semi-automatic pistols.

My Friend Pedro, the forthcoming 2D platform-shooter from developer DeadToast Entertainment, delivers on that fantasy, albeit with its own wild twists. Here, you're able to leap and roll through the air, ricochet a bullet off a frying pan and into a bad guy's head, do a backflip off a wall and then maybe kick an ashtray at another bad guy's sternum. All in glorious slow-motion.

For the creator behind *My Friend Pedro*, Victor Ågren, it all began about 14 years ago while he was still at college in Sweden. Back then, his gun ballet shooter was a little Flash game he'd created as a final-year project. Initially, there was a prototype called *Extreme Russian Gymnastics*, which first explored the jumping and spinning

mechanics – to which Ågren would add all the shooting in later builds. For about eight years, *My Friend Pedro* lay dormant, a Flash file languishing on a hard drive, until Ågren polished it up and put it online in 2014.

"It got well-received and people kept asking for more," Ågren tells us, "and that gave me the confidence to make it into a full game."

Since then, Ågren has formed his own studio, DeadToast, and is now in the latter stages of making a new, improved version of *My Friend Pedro* in Unity. Ågren has also built up some vital industry experience: after finishing college, he moved to the UK for a job at Media Molecule, developers of the hit *LittleBigPlanet* series. He was just 19 years old at the time.

BULLET TIME

"It was really good to learn the process of how a game is actually made," says Ågren. "What priorities to have at what stage of development. Ideas of keeping things modular and knowing the hook of what you're trying to create. And just being around amazing people and being able to absorb a bit of knowledge about everything."

When Ågren saw the reaction to *My Friend Pedro*, though, he decided to leave Media Molecule to work on his new iteration full-time; a process that has meant getting to grips with Unity, and figuring out how to realise his vision of a gracefully chaotic platform-shooter.

"There's always a bit of a learning curve when picking up new tools," Ågren tells us. "The biggest thing for me was probably getting used to a 3D engine, as I've never done any 3D



▲ *My Friend Pedro's* grimy interiors are created with a mix of monochrome textures and tinted lights.

▼ *The Matrix* and *Half-Life* mod *The Specialists* inspired *Pedro's* gracefully violent action.





^ "I think I just like the juxtaposition of a bloody, smiling banana next to a pair of guns," Ågren says of his game's mascot, Pedro.

modelling or the like before. A lot of the physics in the game is sort of only half-using Unity's physics engine. I generally find that relying on pure physics too much for gameplay can make things a bit unpredictable at times, which can lead to frustration for the player."

For *My Friend Pedro*, Ågren wanted to move on from the evocatively hand-drawn 2D graphics of the Flash game. For inspiration, he turned to the work of Liam Wong – a celebrated game director

at Ubisoft, and also a photographer noted for his colourful, neon-infused images of Tokyo at night. In the process, he also came up with a way of creating *My Friend Pedro's* assets by himself.

"I kept seeing Liam Wong's amazing photography popping up all over the internet, and that inspired part of the visual direction," Ågren tells us. "What I ended up doing actually saved me a lot of time. So all the textures for the backgrounds in the game are actually black and white, and things get coloured with post-processing and lights. This meant I could reuse a lot of textures and it also helps making the game more readable, bringing emphasis to the things the player will interact with."

What will likely become *My Friend Pedro's* unique selling point is its array of movement options. As the game progresses, the masked hero's sidekick – incidentally, a grinning, talking banana – relates a growing roster of moves that players can use as they see fit. Like the classic *Max Payne*, the hero is able to slow down time as he rolls, dives or twists through the air; the wrinkle here, though, is that the pair of pistols held in the protagonist's hands can be aimed and fired independently. You can therefore barge your way into a room, shoot one bad guy down on some stairs and another standing on a catwalk above; indeed, the constant kill bonuses

"The biggest thing for me was probably getting used to a 3D engine"

actively encourage you to pull off as many daring and graceful assassinations as your skill allows. The 2D perspective also plays a part in *My Friend Pedro's* appeal; with a plan view of an upcoming room full of enemies, the player can take a moment to consider their attack.

"I like keeping the gameplay to two dimensions, since it makes it easier for the player to process what's actually going on," Ågren says. "There are sections where you might want to plan ahead a bit before diving into a room full of baddies, while other sections might

force you to just push on through and deal with each situation from moment to moment. I found that trying to keep the action super-

intense all the time can be a bit too much, so it's important to bring down the energy level occasionally, only so you can ramp it up again from there."

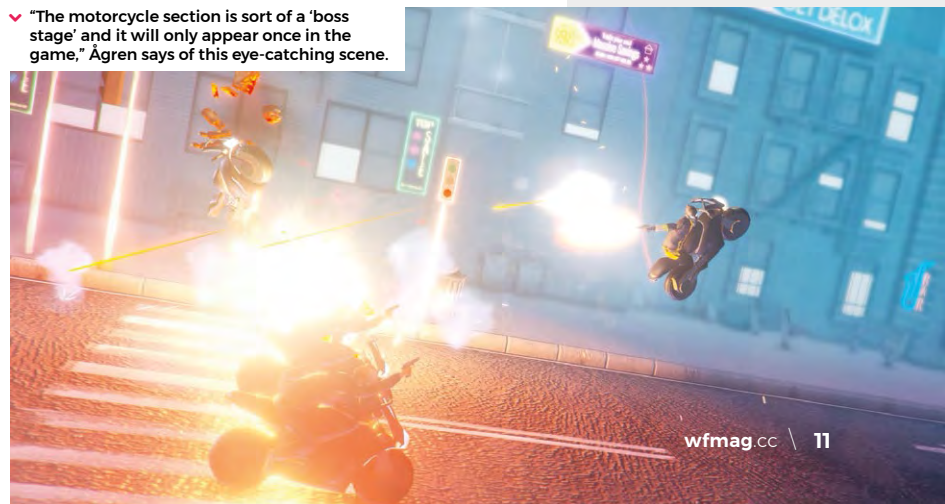
As for life as a solo developer, the experience has been rewarding but similarly intense, Ågren says.

"It can be quite overwhelming at times, but also liberating sometimes too, as I can take the thing I'm making in whatever direction I want. I love the hype the game is getting, but at the same time I sort of have to not take it too seriously, in order to not crumble under the pressure to deliver. Solo indie development is a great way of finding your limits!"

SWITCHING IT UP

On PC, *My Friend Pedro* is played with keyboard and mouse; to use its split-aim mechanic, you're asked to hold the right mouse button to lock onto one target, before moving your crosshair and hitting the left button to lock onto the second. It's a system that, with practice, will have you diving around and pulling off violent stunts like Neo in *The Matrix*. On the Nintendo Switch version, meanwhile, the controls will work a little differently, Ågren explains. "On gamepad, it's like a twin-stick shooter where you point your right stick in the direction you want to aim. To split your aim with the gamepad, you aim at an enemy and as long as you hold down the left trigger you will lock on one arm to aim at that enemy, allowing you to aim freely with the other arm."

^ "The motorcycle section is sort of a 'boss stage' and it will only appear once in the game," Ågren says of this eye-catching scene.





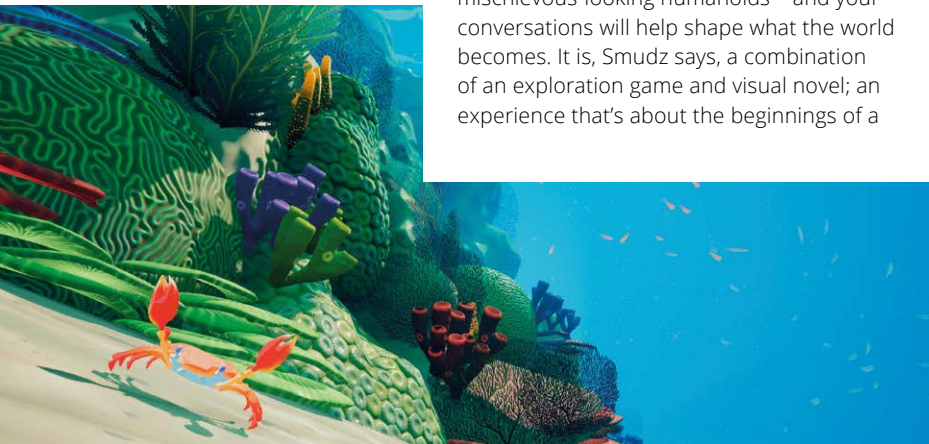
Total immersion

Mythic Ocean is an upcoming adventure game about exploring and conversation. Wireframe finds out more

Info

GENRE
Adventure
FORMAT
PC
DEVELOPER
Paralune
PUBLISHER
Paralune
RELEASE
TBA 2019

♥ Researching the game library, Paralune were struck by how commonly the ocean features in ancient myth – and thus, *Mythic Ocean's* setting was born.



Imagine you're paddling on a seashore, but looking out to the deeper waters beyond, you get the unsettling feeling that something unknowable and maybe a little monstrous awaits beneath the surface. That, according to environment artist Matt Smudz, is what he wants players to feel when they're playing *Mythic Ocean*. It's an adventure game that, with its cel-shaded graphics and gentle pace, looks inviting enough, but there are hidden, darker complexities here, too: there are even parts, Smudz says, that players may find somewhat scary.

The basics are these. A new world has just been formed – one sort of like our Earth, but governed by godlike, aquatic creatures. The player traverses the depths, talking to these gods – which range from fish to sea otters to mischievous-looking humanoids – and your conversations will help shape what the world becomes. It is, Smudz says, a combination of an exploration game and visual novel; an experience that's about the beginnings of a

new world, rather than saving an existing one from some kind of apocalypse.

"We want the experience of playing the game to be very inviting, intuitive, approachable even to people who maybe haven't played a first-person exploration game before," Smudz says. "But under the hood it's pretty complex. We really want your choices to have a big impact."

The studio behind *Mythic Ocean*, Paralune, was set up in May 2016 by Smudz and co-founder Darren Malley; at the time, they were developers at Vicarious Visions, the Activision-owned firm that developed *Skylanders*, *Guitar Hero* and the *Crash Bandicoot* remakes (see page 44). Deciding to strike out on their own, the pair set up their indie studio with a third member, artist and animator Robyn Haley, with the aim of making more personal kinds of video games.

"I think our studio mantra is that we make games about feelings," Smudz says. "Whatever we make, we want our games to really dive deep into the inner world of the characters within them – to their inner emotional life. Regardless of genre, that's something we want to accomplish. Our second mantra is, we say that our players are storytellers, too. Meaning, whatever narrative we create, we want the player to become part of it through playing the game. And we want them to be able to have an effect on the outcome."

Given that Paralune comprises a composer and a pair of artists, it's little surprise that visuals and sound are a major part of *Mythic Ocean*; Smudz cites the films of Japanese artist Hayao Miyazaki as an inspiration when it comes to the



^ Who you are in the game is a mystery, Smudz says – and while you’ll be able to learn more, “there may be consequences” for finding out.

game’s aquatic gods. The premise, meanwhile, was dreamed up while playing *Dark Souls*. What if, Smudz thought, rather than having to slay these mythical beings, you had to converse with them?

“Your identity isn’t known to you early in the game,” Smudz explains. “One thing I like about that is hopefully it makes the game feel a little bit more accessible – you’re free to be who you are in this world. Nobody ever comments on your appearance or anything of that nature... I wanted to have a game where the player isn’t the hero around whom everything revolves. You’re not the star of the show – it’s about these gods and the world they choose to create.”

While *Mythic Ocean* will be comparatively short – Smudz estimates that players will be able to complete it in around four hours – it’ll contain branching conversations that will, ultimately, have a far-reaching effect on how the game ends. Whether those endings are good or bad, Smudz says enigmatically, will depend on how the player interprets them. But from a technical point of view, he adds that the sheer volume of branching dialogue has represented a challenge for such a small team.

“Normally, working on a larger game in the past, I’ve had access to a QA team – a large staff of people who are paid to just test the game around the clock,” Smudz explains. “On a smaller team like ours, we’re kind of our own QA, at least at this point in time in development. We have to plan very diligently. We do have to keep track of the aspects of the game that might require balancing later on. Our writer [Malley] is also a systems designer, so he’s very tied up in that whole area. One of the challenges for sure is that it requires a lot of

testing. It’s difficult to test when there are so many potential outcomes.”

Mythic Ocean, then, is a comparatively gentle game; one about relationships and story rather than action.

“I’d really like more games that systemically incentivise, or encourage, people to slow down and just take in their surroundings. Games that transport you to another place. I think, if a game can make you feel like you’re in another place, convincingly, then that’s quite an accomplishment. That’s something that I appreciate in any genre of game.”

Just because *Mythic Ocean* is non-violent, though, doesn’t mean that there isn’t room for mischief. And it’s here, perhaps, that players will find some of those darker depths Smudz was talking about.

“Oh yes,” Smudz says, when we ask whether we’ll be able to lie to the gods. “We’re absolutely leaving room for players who want to be a little bit mischievous, or cause a little chaos. You can spread misinformation. The way certain storylines play out, you can get certain gods to trick others. You can definitely take a somewhat destructive path in this game.”

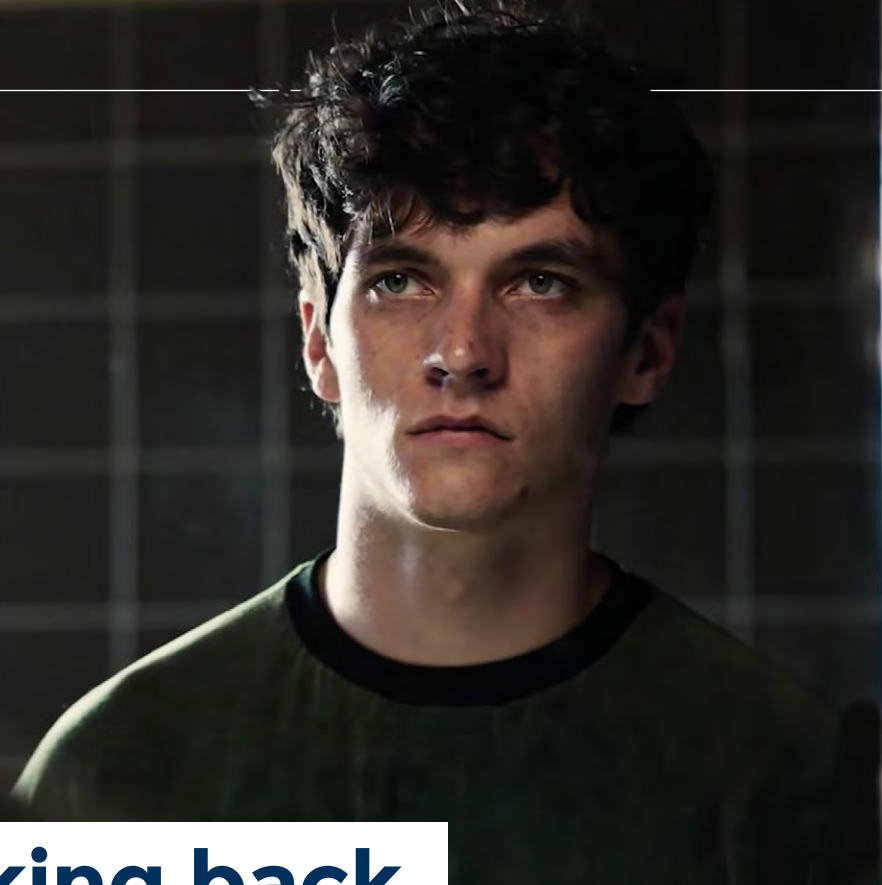
“We want our games to dive deep into the inner world of the characters”

^ “I’m not the hero who’s going to save the day. What is important is to influence other people,” Smudz explains.



PICTURE PERFECT

In creating *Mythic Ocean*’s cel-shaded look, Paralune have taken inspiration from old Disney films and classics of Japanese anime, like Studio Ghibli’s much-loved *Ponyo* and *Princess Mononoke*. But Smudz, who has been working on the background art, was inspired by a less obvious influence: the hit shooter, *Overwatch*. “For me, doing the backgrounds of the game, *Overwatch* was a big inspiration,” Smudz tells us. “It’s a first-person shooter, a very different kind of game, but it showed how you can create materials and surfaces that are somewhat true to life, but not photorealistic. There’s a simplicity to them – the noise and the chaos of nature and reality have been subtracted in a very artful way.”



^ *Black Mirror* brought back the concept of nineties FMV games with gusto.

Looking back

And looking forwards, too – it's 2018 and 2019



^ *The Walking Dead* will be finished, at least.

Really we should focus a look back at 2018 on the fact we enjoyed a wonderful launch of the very magazine you hold in your hands (or PDF you read on your screen), Wireframe. But we can't, because oddly enough this recent past year was full of all manner of ups and downs in gaming news – and we've put together a bit of a highlights (and lowlights) reel, along with a bit of a look at what is, and what could be, coming up in 2019.

The fall of Telltale dominated the last quarter of 2018, with the erstwhile developer of the modern choose-your-own-adventure games

collapsing under the weight of its own bad management. Ninety percent of the team was let go, and while the final *Walking Dead* season was rescued by creator Robert Kirkman's own company, the damage was impossible to ignore.

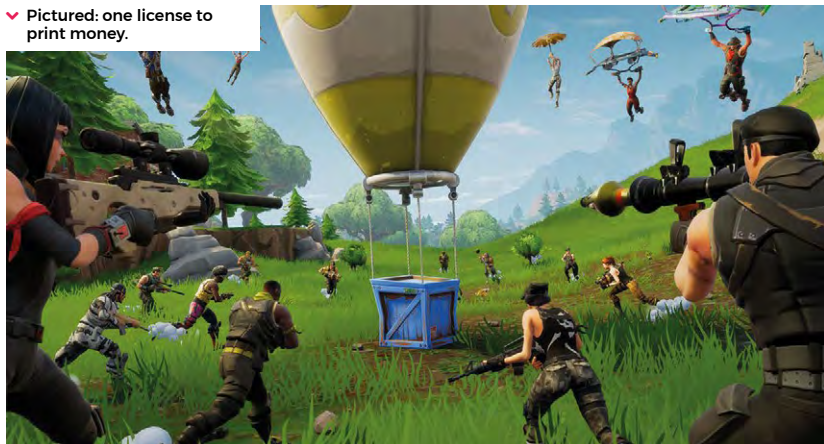
On the happier side of things, another choose-your-own-adventure showed up in the form of the Netflix production *Black Mirror: Bandersnatch*. Taking the anthology series that asks 'what if machines are actually bad?' and turning it into an FMV adventure game from the mid-nineties seemed to hit all the right notes with those who didn't own a PC with CD drive back in the day. Could it mean a revitalisation of the genre, only this time with real actors? We shall see.

EPIC'S EPIC

The biggest news – or more accurately the biggest ongoing saga – has been that of Epic's *Fortnite*. Yes, the studio is being sued by people who innovated dances before the game 'borrowed' them to sell to players, but that's nowhere near as important as the fact this is a game now making around £75 million per week. Just to restate that: per week. *PUBG Mobile* might have done well in 2018, but *Fortnite* is in charge, and it'll be hard to see how that will change any time soon.

Your traditional massive game taking years to make – and 100-hour weeks, at least for

^ Pictured: one license to print money.



▼ Red Dead Redemption 2: proof there's life in the old single-player horse opera yet.



some people – isn't dead though, and the huge success of the 76%-rated *Red Dead Redemption 2* is proof there's still a market for the one-shot blockbuster. We'd fully expect Rockstar to keep supporting the game – and bringing in cash – via the game's online mode, aka *GTAV*, but that initial performance is something to bear in mind next time someone says the only game that will do well is a battle royale game-as-a-service.

Speaking of making money, there was an interesting shunt in 2018 away from nickel-and-diming behaviour in favour of – dare we say it – a more consumer-friendly approach. It obviously operates hand-in-hand with games as a service and in-app purchases becoming the norm, but we shouldn't overlook the fact that a) incidences of loot boxes dropped dramatically, both thanks to public outcry and official restrictions in some countries, and b) studios such as Ubisoft announced future updates to some of their games would be completely free, rather than demanding a few extra notes every time a minor cosmetic update goes on its respective storefront.

SPACE YEAR 2019

As we made the jump into 2019, one thing was looking ever more certain: the battle lines of old are starting to blur, and the future is one in which we will all, in one way or another, be beholden to the power of the stream – and even if we're not, we'll all be able to play everything together anyway... okay, maybe not during 2019, but things are heading that way.

Many of us remember those arguments of the playground – the Nintendo versus Sega, Sony versus Sega/Nintendo/Microsoft, PC versus nobody, the usual. To think we'd live in a world now where (after some heel-dragging from Sony) we can play the most popular game in the world, *Fortnite*, on any home computer or console format against one another is incredible. And that's just the start, as this exercise has shown publishers the format mattering ever-less compared to the game.

With titles increasingly opting for a game-as-a-service format, the actual machine you play them on doesn't really factor in so much. Kids are less PS4 or Switch fans than they are

“Kids are less PS4 or Switch fans than they are Just *Fortnite* Fans”

Just *Fortnite* Fans. And with this drive towards decentralised gaming comes the moves we've seen being teased and pushed for in recent weeks and months. Hand in hand with that is the streaming future, and 2019 should see some early signs from Microsoft's big push there, Project xCloud.

Another company aiming to shake things up in a big way is Improbable, with its distributed server tech SpatialOS promising previously unheard of player counts and world sizes – one battle royale title based on its tech, *Mavericks*, promises 1,000-player matches, incredibly. 2019 should finally see some fruit from these particular seeds, and while it'll take something truly impressive to blow away a cynical, buzzword-weary crowd, the company (and so the tech) has some serious money behind it, and again it's something that's platform agnostic. It does mean we'll likely see a bunch more battle royales coming in the near future, of course, but what can you do?

None of this look backwards or forwards matters when placed next to the biggest of the recent news-hits, of course: Valve re-hiring two of *Half-Life's* writers, Jay Pinkerton (who returned around July 2018), and more recently Erik Wolpaw, in the latter stages of 2018. Say it with us now: '*Half-Life 3* confirmed'. That'll be the day... 🗣️

NEW CONSOLES? SLIGHTLY MAD

The new Xbox and PlayStation will likely see the light in 2019, as will an updated Switch, we assume. But the real surprise arrived in the shape of tweets from Slightly Mad Studios chief exec Ian Bell, claiming the Project Cars dev is making its own console. Apparently capable of running VR titles at 120fps (60 per eye), pumping out 4K with ease, and all manner of other vast promises, the Mad Box (as it's known) won't see the light of day for a few years. It could be vapourware, it could be a glorified Steam Machine, it could actually be what Bell is promising – a new console from a very surprising source. And... well, it could just be a joke.



Attract Mode



The Pathless ↑

A mystical world, an archer and eagle companion, a curse to be dispelled – *The Pathless* sounds pretty pretty by the numbers, but it has a pedigree that makes it worth paying attention to. Giant Squid Studio's previous effort, *Abzû*, had some truly great elements to it – not least of which its superb atmosphere – and it looks like that ambience makes itself known again in *The Pathless*. The addition of combat could be a make-or-break factor, though – we're hopeful it can be more *Shadow of the Colossus* than *Extinction*.

Release date: 2019



The Waylanders ↑

The Waylanders isn't shy about harking back to RPGs like *Dragon Age* and *Neverwinter Nights 2*, and it seems the buying public isn't averse to this approach either – the game raised over £130k on Kickstarter. There's a lot still to come from this one, but early opinions on *The Waylanders'* combat systems – tactical, involved and layered with strategic elements – show a ton of promise.

Release date: June 2020



Killer Queen Black ↑

Enjoying an extended beta is the kind of thing you want to see from a multiplayer strategy/ combat game, so it's good to see *Killer Queen Black* taking that route to release, making sure any potential issues are ironed out as much as possible. A tweaked version of the eSport-focused original *Killer Queen Arcade*, *Black* is coming to PC and Steam with eight-player support and optimisations for the non-cabinet format. There's a lot of competitive multiplayer potential here.

Release date: 2019

Far Cry New Dawn ↓

Far Cry's *Blood Dragon* and *Primal* set the tone, so it's of little surprise to see a spin-off to the fifth core game in the series. *Far Cry New Dawn* is (spoiler alert) set after the bombs have dropped and the world has ended – but rather than a grey/brown and lifeless post-apocalypse, this is one riddled with bright colours and, somewhat ironically, life. Still, *Boomer is dead*.

Release date: 15 February, 2019





Neo Cab ↗

A futuristic Uber simulator might sound... well, actually, that sounds quite interesting. Anyway, *Neo Cab* is an 'emotional survival game' in which players control a driver-for-hire as she ferries around passengers, learning – and impacting on – the stories of those who sit in the back.

Players will make their way around Mexico City, San Diego and Dubai-influenced Los Ojos, with routes procedurally generated from prefab 'chunks' of the city. While the world is created on

the fly, the stories running through *Neo Cab* – both those of individual passengers and that of the main narrative – have been crafted by a team of writers. With multiple decisions to be made and paths – both on the road and elsewhere – to be taken, you'll also have to make sure you look after your own emotional well-being along the way.

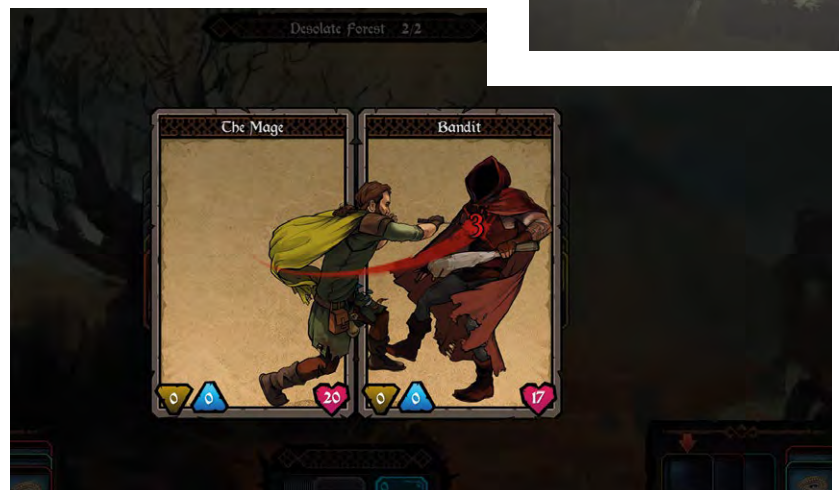
Neo Cab has some serious scope, and could well be the game to put the visual novel in front of more players than ever before.

Release date: 2019

Ancient Enemy ↘

Developed by Grey Alien Games, *Ancient Enemy* sees a host of other names joining the card-party – Big Robot Games' Jim Rossignol, artist Jen Pattison, and 'support' from Bithell Games. The studio's previous attempt at card combat/RPGing arrived in the form of *Shadowhand*, a challenging strategy game with a few issues. Said issues are being tackled directly in *Ancient Enemy*, though, with the dev team highlighting changes like faster combat from the get-go. With such a team of talent behind it and a willingness to learn from past mistakes, *Ancient Enemy* is certainly heading in the right direction.

Release date: March 2019



Sea of Solitude ↘

The next title to arrive under the EA Originals banner comes from Cornelia Geppert and her Jo-Mei Games studio. It tells the tale of a young woman who, as a result of her loneliness, has turned into a monster – and the quest to regain her humanity. It sounds fascinating, and looks splendid.

Release date: 2019





^ Harry Nesbitt (left), creator of *Alto's Adventure*, and Adrienne Law (right), producer of *Monument Valley 2*.

Breakthrough Brits in conversation

BAFTA-recognised devs Adrienne Law and Harry Nesbitt share their thoughts on making games, work-life balance and more

It's certainly woolies and scarf weather now, but the low hanging sun provides a beautiful backdrop as Adrienne Law and Harry Nesbitt make their daily short walk from home to the ustwo games office. In late 2018, Adrienne and Harry were both recognised by BAFTA as Breakthrough Brits – an award given by BAFTA to new and emerging talent across a variety of art and entertainment industries. But that's not the only thing they have in common – as well as working in the same office, the pair are housemates, too. Adrienne is a producer at ustwo games, most recently on the acclaimed puzzler, *Monument Valley 2*. Harry doesn't work for ustwo, but he's a regular fixture there, taking a spare desk to work as the lead developer

and artist for *Alto's Adventure* and its sequel, *Alto's Odyssey*.

As two professionals early in their careers in an ever-evolving industry, Adrienne and Harry find themselves with much in common, but the routes that led them to working and living together were very different. The pair agreed to take an hour out of their work schedules to speak to Wireframe, and to each other, about their personal experiences of game development, how it feels to release a game, work-life balance, and the potential of games to affect and enrich lives.

Adrienne Law: My route into the games industry was semi-accidental. I played games a lot when I was a kid but didn't know there was



I went to an all-girls' school that specialised in maths and science, so there was no reason that I would have thought I couldn't work in tech, but the school didn't push the idea of working in tech and coding. I think if I had been aware of it from a younger age I would have been a programmer.

an industry as such to go and work in. I did an English degree thinking that might possibly set me up for going into some kind of creative, story-driven field, which was what interested me. After that, I spent a few years working different jobs – I was a teaching assistant, I worked in finance, retail, marketing, and was circling around trying to get into film and TV industries.

Eventually, I got to the point where I went onto job sites and searched for 'production assistant' and that's where I found a production assistant role going at ustwo games. I thought, "Oh! Production is a thing in games! I didn't know that." I decided to just

go for it. I ended up having a few interviews with ustwo – I think they were worried because I was quite

quiet and they weren't sure how much I would step into the role – but they let me through the door and gave me a chance. I've been here ever since. I never set out to be in the games industry, but I think I'd been gaining a lot of skills and had an awareness of the medium, so those things combined into making me a good candidate for the role.

"I played games when I was a kid, but didn't know there was an industry to go and work in"

Harry Nesbitt: I've always thought about working in games. From a young age I had an interest in how games were made from an artistic standpoint. I would always look up who was responsible for the concept art. Concept art as a job was something I was aware of from a very young age.

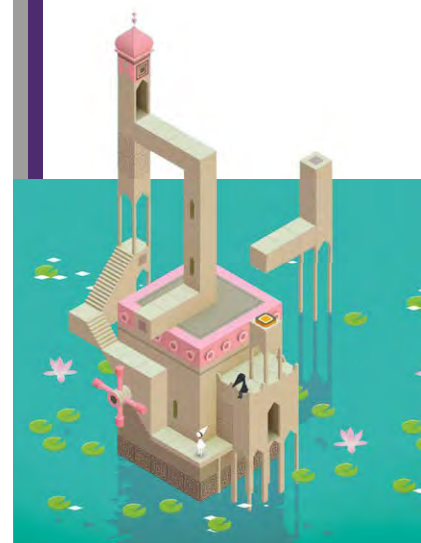
Around 2006, when I started at university, indie games weren't in the mainstream, and making games in your own bedroom wasn't as popular an idea. When I discovered Unity, I thought "Oh, I can download this for free and I can learn all the basics online." I saw examples

of illustrators who were downloading it and making cool, interesting little projects – almost like little art pieces

– bringing their illustrations to life. It made me realise I could have a play with that. My knowledge of the basics of JavaScript and web development helped me pick up the coding side of things a little bit more easily.

When it came to making *Alto's Adventure*, I knew a little bit of Unity and had been playing with it for about 12 months, so I ▶

▼ Produced by Adrienne Law, mobile puzzler *Monument Valley 2* has been nominated for dozens of awards since its release in 2017.





realised I could at least be playing around with it, seeing what's possible and using it as a way to demonstrate certain ideas. Within a very short space of time, less than a week maybe, I'd been able to put together a basic prototype of the core systems, such as the terrain generation, basic player physics, even some effects such as particles and Alto's scarf. It took another year and a half from there to get it finished, but online resources gave me what I needed to eventually get the game made. It's not necessarily an experience I'd want to repeat though!

How did it feel to release your first game?

AL: With *Monument Valley 2*, we announced and released it on the same day at WWDC in June 2017. Technically, the first game I worked on at ustwo was *Land's End*, but when I joined as a production assistant for the final stages it had already been running for 12 months. It was lovely because the team congratulated me on my first game when it was finished, which felt pretty good, but when you join a project that already existed, it doesn't quite feel like it belongs to you in the same way.

When *MV2* came around I found out I was producing it, so I was there from day one. It's a very different feeling to joining a game halfway through – you take a lot more personal ownership over it, you're very emotionally involved in the game. Even though we were using the same mechanics as *MV1*, it felt full of potential, so I was excited to put my own stamp on *MV2*.

At the same time, it was kind of strange, because nobody knew we were working on it due to our plan to announce and release on the same day. You start feeling like you're a bit crazy, thinking, "People are gonna care, right? People are gonna play this, right?". When we actually

released it, it was like a big exhale. You could see the news spreading out and people playing it. Because it's a relatively short game we had people telling us how much they loved playing it by the end of the same day.

HN: Did it feel quite surreal? When we've launched our games, there's a transition period where the game is ready and people are expecting it, and even then it's very surreal. Suddenly it's gone from being yours to being everybody's, in a sense, and it's a very difficult thing. For you guys that happened in the space of 24 hours, and suddenly you're talking about this thing you've literally not said a word about.

AL: I was on a panel at BAFTA Guru, two or three months before *Monument Valley 2* came out, and I was having to constantly check myself in case I accidentally said '*Monument Valley 1*'. Internally, we had to make that distinction, but we hadn't announced anything publicly. It would be such a small slip-up, and I didn't do it, but it's those kind of pressures you're dealing with. How did you find releasing *Alto's Odyssey* as opposed to *Alto's Adventure*?

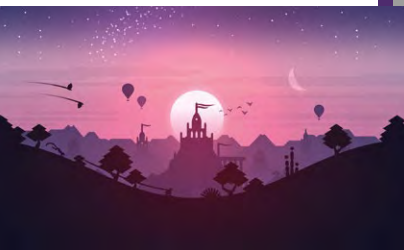
HN: *Odyssey*, our second game, was very different. I had a much better idea of what to expect and was able to prepare emotionally as well as logistically. Working in the ustwo office made a huge difference, being surrounded by supportive people who understood what goes into making games and how it can be when

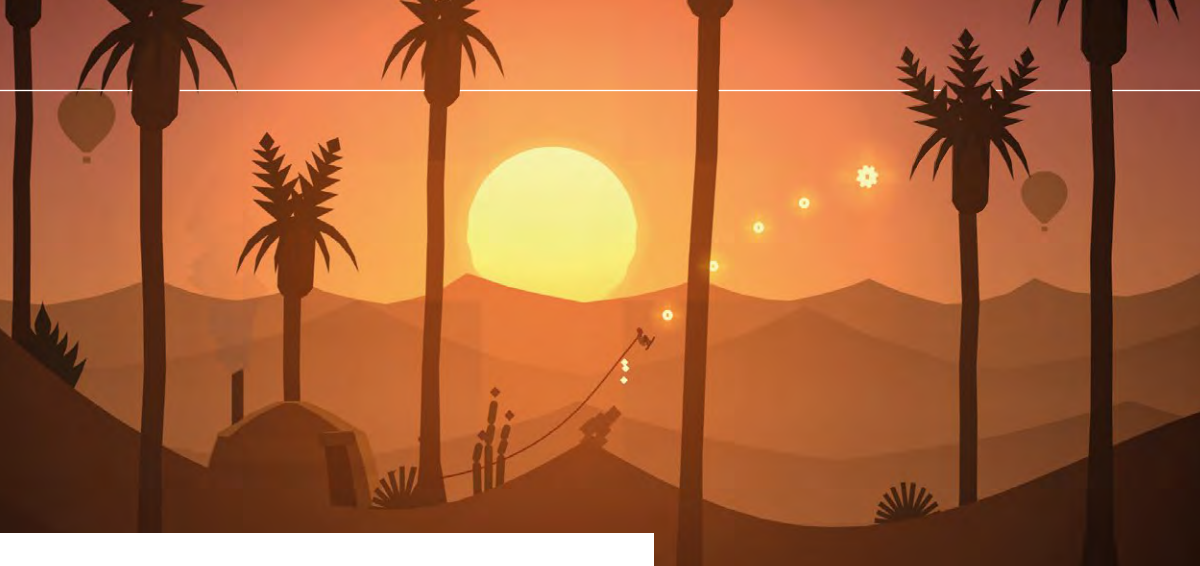
"Within a very short space of time, I'd been able to put together a basic prototype"

they're in the public eye. For the first game I didn't have any of that – I was working by myself from home in a quiet part of Devon in the southwest of England. I had no idea what to expect.

Launching *Alto's Adventure* was a whole mix of feelings because you have this thing that is incredibly personal to you that you've been nurturing for a long period of time. *Adventure* was very much my creation and I was the only person who had my hands on it in any meaningful sense. Then suddenly you put that out into the world and it's being – thankfully – enjoyed by a lot of people, but also heavily scrutinised. Suddenly feeling solely responsible for hundreds of thousands of people when previously your only responsibility was to do

▼ "Seeing how people have responded to the *Alto* games as a peaceful escape has been great," Nesbitt says.





your laundry and remember to buy toilet paper [laughs] really threw me for a loop.

The focus for the second game was to make it in a healthy and sustainable way. We built the team out and we prioritised everyone's health and happiness. So comparing the two launches, that was the main difference – with *Odyssey* we were much more prepared, structurally and emotionally. With *Adventure*, we were just trying to figure all that out in real time whilst it was happening and it was kind of chaotic.

AL: For me, work-life balance is the most important thing. I feel a lot of responsibility for that as a producer, for example, managing people's schedules so you're not suddenly putting a load of extra work on people. On *Monument Valley 2*, I ended up putting in extra hours partly in moral support of friends who wanted to do stuff on a weekend, and I'd just come and be there. Partly it was anxiety too, because it was the first time around for me being a producer, and I was still working out where I was and wasn't adding value.

It wasn't that all my time was at work, it was that all my thoughts were at work. Even if physically I was at home with my friends, emotionally I was at work and my entire life had been sucked into the job. What I've tried to do in the last 12 months is to withdraw from that and take all my thoughts and feelings back, splitting them out across my life. I think everyone should

^ Harry Nesbitt's warm, minimal artwork did much to make the snowboarding mobile games *Alto's Adventure* and *Alto's Odyssey* (pictured) so atmospheric.

have the right to do that. Fortunately we have a very anti-crunch culture at ustwo and I think it's really hard to do that in companies that force people to work 80 hours a week.

What is the most important thing in your games to you?

AL: Our target platform is mobile and it's by far the most accessible medium for games. It's not a console and you don't need a super-powered PC. Previously you'd only be catering to an audience who are very familiar with games already.

Film and TV are seen as mediums – you rarely meet people who say "I don't like films," but you will meet people who will say "I really like romcoms but I don't like action movies," and there's a difference between medium and genre. I feel like games are getting to the point where people who aren't already gamers might be starting to understand that 'games' is not a genre, it's a medium, and there are many different genres that you can be exploring within that medium.

When people think of mobile, they think quick puzzle games, free to play, a couple of minutes to fill in time on the train. We want to say, "actually, that's a genre within mobile, not the medium." We want to say, "let's not treat games as a single genre anymore."

HN: The games that have had the most profound effect on me are where the medium almost disappears and it's the experience that sticks with you. I think games are uniquely positioned to do that because you have agency in those worlds. It gives you the potential for creating empathy and looking at the world slightly differently. I'm not saying I could hope to make games on that level, at least not right away, but I want to try and consider a little bit of that in everything I do. 🐾



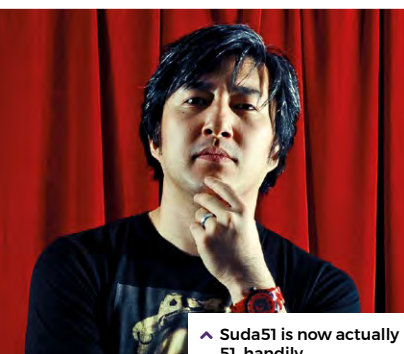
^ "It was kind of strange, because nobody knew we were working on it," Law says of *Monument Valley 2*.





Grasshopper's middle age

Suda51 on plans for the future, pro wrestling, and truly becoming his nickname



▲ Suda51 is now actually 51, handily.

For two whole decades, he's sat atop a studio pumping out a mix of the weird and wonderful; games picking up plaudits as much as they were dismissed by the buying public. Even in an industry as boom-and-bust as the world of gaming is, Goichi 'Suda51' Suda is still soldiering on with Grasshopper Manufacture. And he's almost as bewildered as we are about that fact.

"Looking back, I feel the same way you do," he laughs. "One unique thing that's allowed us to survive so long is the fact that unlike other developers who usually stick with one publisher and work with them for a very long time, Grasshopper has had the opportunity to work with lots of different publishers." Sega, Bandai Namco, Level-5 – even western outfits like EA and Deep Silver have all partnered with Suda's studio for releases over the past 20 years. "That

has really empowered us to be more flexible, to be able to adapt to different work styles," the CEO continues, "and that's enhanced our game development overall."

The most recent publisher Grasshopper has worked with is Nintendo, partnering up for the physical release of *Travis Strikes Again: No More Heroes*. Suda himself was one of an early batch of Japanese developers invited to see the Switch, and opted to make something exclusive for the format, attempting to take full advantage of the hardware's idiosyncrasies.

You'd think all of this is typical behaviour for an indie studio, but Grasshopper was purchased by GungHo Online Entertainment in 2013, and has been directed by its Japanese parent ever since. Even so, Suda does see his team as independent in spirit. "And I intend to keep moving in that direction," he says. "We used to be a team of around 50 people, but we've

gone down to 20 now. We're pretty much close to being fully independent. I don't see that changing any time soon."

That ethos comes from the top, with Suda impossible to separate from Grasshopper, and vice-versa. So when asked about how the man sees his career over the years, it's easy to draw parallels between it and the company he started. "As far as my career is concerned, there are different eras – different styles I've adapted, and different styles I've stuck with," he explains. "Currently I feel as though I'm in the fourth era with this iteration of Grasshopper.

"The first era – the first 10 years of my career, up to *No More Heroes* – I was pretty much in full director mode the entire time, pushing out games one after the other. Then, in the next five-year era, my company got a lot bigger thanks to the success of *No More Heroes*." Around the same time the PS3 and Xbox 360 came about, along with the Unreal Engine on said consoles.

"That led to *Shadows of the Damned*, *Lollipop Chainsaw*, and *Killer is Dead* – what I consider the HD trilogy," Suda says. "During that period I stepped back from a director position, I was more of a creative director and producer... For the third era, that's when GungHo came in, and I started making games more closely with them for the next five years.

"Finally now in this fourth era I want to return to the original Grasshopper roots that I started with in my first 10 years. That's what led to the changes in the studio you see now."

Those changes see a smaller team, but the scope for projects has widened significantly. When courted by the likes of EA a decade ago, Suda and Grasshopper were stymied: if the paymasters weren't happy, the game wouldn't happen. See *Kurayami*, a planned game based on Franz Kafka's *The Castle*. Conceptualised but ultimately cancelled, it ended up – in some form – in the finished *Shadows of the Damned*.

Would Suda go back to this pet project now? "I feel like if it were a bit earlier it would be harder to sell [*Kurayami*], but now I feel the time might be good," he laughs. "*Shadows of the Damned* originally started out as the *Kurayami* project. I feel very satisfied with what *SotD* ended up becoming, but at the same time there's a frustration that I wasn't able to realise the



▲ Lollipop Chainsaw: imaginative, silly, rough.

original vision I had. Just talking about it with you now, I feel it's rekindled the flame of *Kurayami* inside me."

But the near future won't see Kafkaesque games from Grasshopper, at least until after one other project is finished. Suda began his career working on the *Fire Pro Wrestling* series, and his love for pro wrestling has endured – so naturally one of the next titles he wants

from Grasshopper involves a bemused pantomime artiste in spandex. "I really want to make a game where the main character is

a pro wrestler," Suda explains. "Hopefully, I'll be able to make that in the next 10 years or so. I'm working on a project for it right now, actually... I want to make an action-adventure pro wrestling game where a pro wrestler near retirement goes to save his grandchild."

Whatever's next, there are no plans of slowing down just yet – in fact, there's a bit of an omen in the air.

"I actually turn 51 this year [on 2 January], so starting in 2019 I will finally be Suda51. That will be the true starting point to my career!" 🗨



▲ Travis Touchdown is almost a mascot of sorts.

VISION OVER COMPROMISE

It would be rude not to ask a man with decades in the industry for his advice for the young and hungry devs of the world – and Suda is happy to oblige: "I feel there are a lot of chances floating out there these days – it's so different from what I grew up with," he says. "There's Steam and other PC platforms, more and more ways to release games every day. So people shouldn't be afraid.

"I feel successful indie games are ones where developers only focus on what they want to create – the noise and the passion.

"They don't try to bend or compromise their vision. That's what ends up making them shine in the end."



^ *Transmission* began as an "on-rails infinite runner" before evolving into a driving game, says Dudley.

Interactive

Back to the 1980s with Transmission

A driving game designed to soothe rather than exhilarate? Developer Jon Dudley tells us more

Are you a solo developer working on a game you want to share with Wireframe? If you'd like to have your project featured in these pages, get in touch with us at wfmag.cc/hello

Think about the driving genre, and things like fast reactions and split-second decisions tend to spring to mind. Even an old classic like Sega's *OutRun*, which is about the fantasy of driving an Italian sports car on a sunny day, asks players to hit a checkpoint before the time runs out. Sea Green Games developer Jon Dudley's forthcoming driving game *Transmission*, meanwhile, has a very different philosophy behind it.

While Sega's arcade racers are influences – the game's even set in 1986, the very year *OutRun* came out –

Transmission is, Dudley says, meant to be calm and soothing where those games were fast-paced and frenetic.

"There's a lot of different inspirations for the project," Dudley tells us, "but the key it revolves around are memories of being driven along the motorway at night with my family when we were going on holiday. That's responsible for the aesthetic, atmosphere and setting."

The premise is that you're a courier, tasked with driving the highways of a city at night, collecting packages from customers and delivering them to the appropriate destination.

In this respect, *Transmission* is a bit like another Sega classic, *Crazy Taxi* (Dudley once dubbed his work-in-progress '*Cozy Taxi*'), in that there are items to be taken from one point to the next, optimum routes to navigate, and penalties for late deliveries. The difference here, though, is in the overall pace and tone; you'll drive to a mellow synth soundtrack, rather than something spiky ground out by a guitar band like The Offspring; this is complemented by the imagery,

which is all shimmering city skylines and neon streaks from car tail lights.

There's a hint of wistfulness to

Transmission's sights and sounds: the distinctive tick-tick of indicators and angular dashboard immediately evoke the boxy hatchbacks of the eighties. In fact, a certain percentage of *Transmission's* challenge springs directly from its setting: sat-nav is still years away, so you'll have to navigate your way from one point to another by reading signs and old-fashioned road maps. A handy slow-motion button will help with this: triggering it will give you a few extra seconds to read a road sign as it whizzes past.

Master your routes and get your packages delivered on time, and you'll be able to use

"Euro Truck Simulator meets Jalopy on a rainy eighties summer night"



^ *Transmission* makes cunning use of lighting effects and depth of field to create the illusion of a huge city at night.

your income to build a courier empire. Cash can be spent on upgrades for your car, or if you're feeling really flush, you can hire staff and open up new branches elsewhere on the map, allowing you to tackle other, more lengthy delivery jobs. It all goes into an experience that Dadley describes as "*Euro Truck Simulator* meets *Jalopy* on a rainy eighties summer night."

What might come as a surprise, though, is that Dadley isn't a driver himself.

CRUISE CONTROL

"Getting the car handling to feel right has been extremely tough," Dadley explains. "I don't actually drive myself, so it's not something I have an innate knowledge of. Secondly, I don't really play racing games, so again, I don't have that built-in knowledge of how a car game should 'feel'. That said, *Transmission* isn't a racing game, so I've made the handling feel more impressionistic – it's heavy and weighty, reflecting the relaxed feel of the game; it's not a twitchy racer. Luckily, I've made some friends who have experience working on some huge driving franchises like *Forza Horizon* who've helped me get the feeling right."

With a 10-year background in software engineering and design, Dadley has plenty of experience in the sometimes fraught process of making triple-A games. But when working as a solo dev – albeit with friends who've helped with things like music, 2D art and writing – Dadley stresses the importance of staying focused on a central game concept.

"It's very seductive to think your latest idea is the key, but that leads to feature creep and a messy, incoherent project that never ships," Dadley says. "More practical, especially as a solo dev, is to have a core of a few systems or ideas that serve what you want the player to feel. I try and relate everything I add to *Transmission* back to the idea of making the player feel relaxed and cosy. If I have an idea that doesn't fit that central design pillar, it's generally thrown out."

Dadley began working on *Transmission* just under a year ago, and he's cannily worked out a number of ways to help keep the project



manageable. How, for example, does a solo developer go about creating the illusion of a 1980s skyline, without a huge budget?

"Smoke and mirrors are essential when you don't have the resources to make something on the scale of *GTA V*," Dadley says. "One of the reasons the game makes such heavy use of depth of field and has a night setting is that it allows me to be more impressionistic. Glowing lights on the horizon create a convincing impression of a city without the need to build them. The player fills in the gaps in their head."

And while Dadley's used Unity as a basis, he's also had to come up with a few bespoke solutions when creating his sprawling map.

"It would be super-time-consuming to name each city, town and road in the world and then create road signs for every road junction. So I built a set of tools to procedurally generate location names and then create the textures for each road sign to correctly point to them all."

At the time of writing, Dadley still has about 18 months of development left, but already, *Transmission* looks refreshingly different: a driving game designed to calm the nerves rather than set them racing. 🎮



^ Customising your car, gradually turning it from a run-of-the-mill hatchback to a miniature hot rod, will play a major role in the game.



^ Dadley's self-built tools allow him to create location names and signs procedurally, saving hours of development time.

THE BENEFITS OF TWITTER

It can be lonely at times as a solo developer, but for Jon Dadley, Twitter's been essential – not only in terms of talking to like-minded designers, but as a sounding board for new ideas. "I spent a few months developing a series of prototypes and posting videos on Twitter," Dadley says. "None really got a reaction until my first video for *Transmission*. That blew up and gained a tonne of attention. Straight away, I knew it had to be my next project – there was something there people connected with."

"I'd definitely recommend using Twitter as a sounding board for prototype ideas. Try and keep videos short and to the point – get to the juicy, flashy heart of what you're showing as soon as possible. People are scrolling through their feed and aren't going to hang around."

The power of ideas



JANE WHITTAKER

Jane Whittaker has been in game development for over 30 years, rising from the ranks of trainee programmer to Vice President levels in the industry.

“I saw technical wizardry and jaw-dropping graphics, but beneath the surface, great ideas were scarce”

W

hat makes a great game? Every developer asks themselves the same question at some point. And while the games industry might seem like a place where you need millions of dollars and huge development teams to make a properly successful game, this isn't always true. *Red Dead Redemption 2* was, of course, an acclaimed game with a budget akin to the income of a small country. Yet, for every *RDR2*, there are others – like *Fallout 76* – that struggle, despite all the money thrown at them.

In reality, a great game isn't linked to how many artists you can hire or how much money you can muster. As history's proved time and again, a classic game can be by a single developer with almost no budget at all – just look at *Tetris*.

The best games, then, have the best ideas – it's really that simple. Be it a challenge of reflexes and skill, or an original story that envelops you, the idea is everything. If you have a great idea at the core of your game, you're already on the path to success.

During my time in product acquisition for publishers, I spent hours sitting in stuffy cubicles at shows like E3 as a procession of hopeful developers lined up to pitch their games: literally a queue of people, forming at breakfast and finishing at bedtime. I can count on one finger how many great ideas I saw during those sessions; I saw technical wizardry and jaw-dropping graphics, but beneath the surface, great ideas were scarce.

On another occasion, I was once sitting at my desk when some team members excitedly showed me a new submission. Was it a game with a vast open world? Was it a cinematic extravaganza like *Uncharted*? No, it was a puzzle game. It was, however, a hopelessly addictive, very clever puzzle game that was full of fun. I cancelled my appointments immediately and flew halfway across the world that same day to meet with the developers. That game was *Tetrisphere*, which became a sizeable hit on the Nintendo 64.



▲ *Tetrisphere* on the Nintendo 64: an example of a solid idea, well executed.

This month, I'm judging a competition for indie developers. Essentially, I'll be choosing the ten best game submissions, which will then be introduced to publishers as part of the event. I'm not judging games on their graphics or scale, but on their underlying ideas. If you have a great idea, anything else can be fixed; if your game lacks a spark of genius, then even the finest marketing won't necessarily save it.

Really, that's what makes this business so exciting: a tiny developer can still compete, to some degree, with major studios – as *Minecraft* proved not so long ago. We're in the midst of a digital gold rush, where indies and triple-A studios jostle for the public's attention. So the next time you worry about your lack of resources, take a seat in your favourite chair, grab a drink, a pen and some paper, and spend some time thinking about that next great idea. Those are the only tools you need for success. 🌐

Toolbox

The art, theory and production of video games

28. Design principles

The importance of deviating from your own design ideas

30. CityCraft

Three easy processes you can use to create a game city

32. Source Code

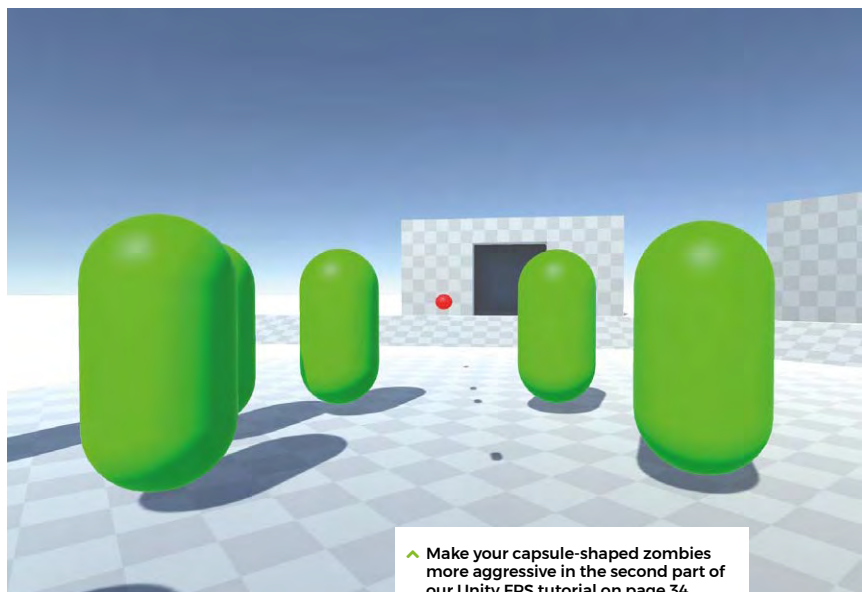
Make your own retro arcade high score table

34. Improving your Unity FPS

Add AI, hit detection and more in part two of our FPS guide

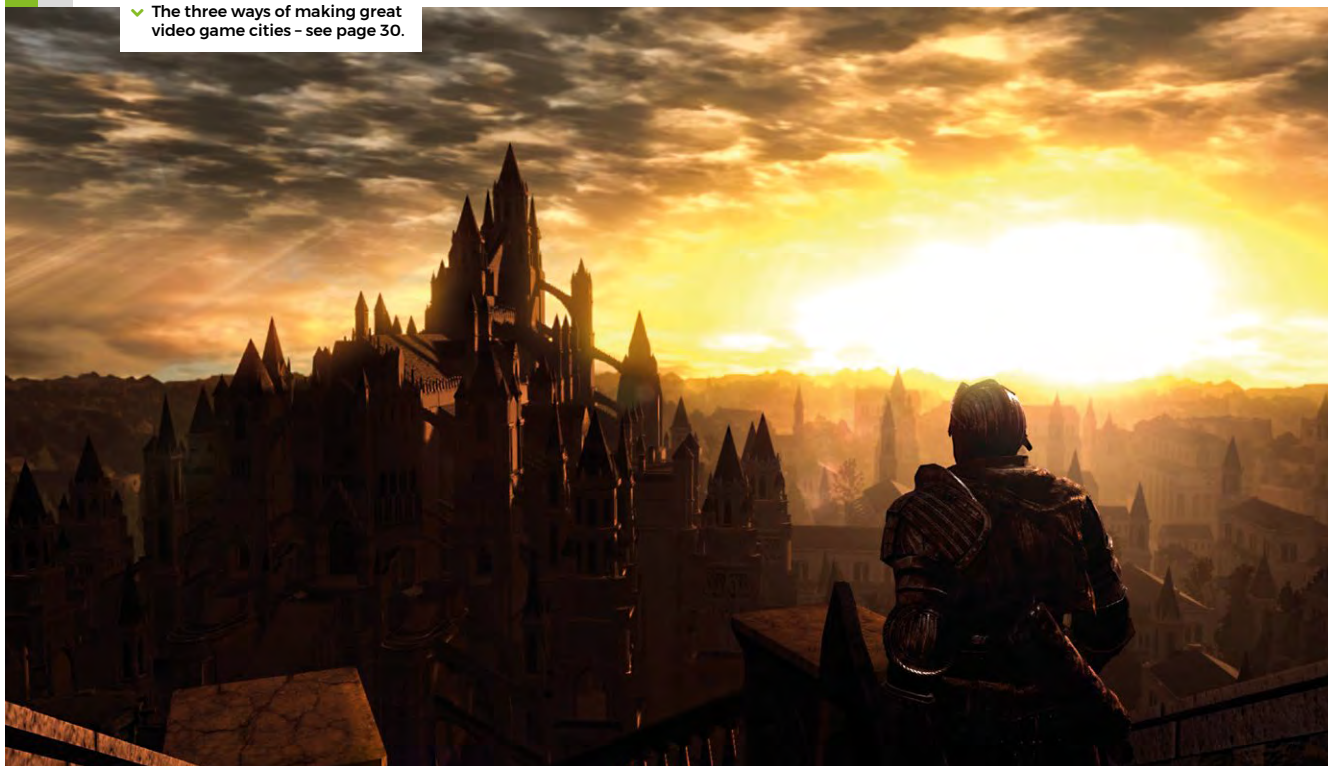
42. Directory

Tutorials, courses and resources for budding pixel artists



^ Make your capsule-shaped zombies more aggressive in the second part of our Unity FPS tutorial on page 34.

^ The three ways of making great video game cities - see page 30.



The principles of game design

Here's why listening to feedback and being willing to deviate from early ideas are vital for great game design



AUTHOR
HOWARD SCOTT WARSHAW

Howard is a video game pioneer who authored several of Atari's most famous and infamous titles
onceuponatari.com

In my last column, I explained how I laid the conceptual basis for my 1982 game, *Yars' Revenge*. Now it's time to see what happens when a concept meets reality. Let's dig in.

I write the core display routines without much problem, and I'm getting the basic game elements on the screen. These being: the player avatar (Yar), the antagonist (Qotile), the antagonist's super-weapon (Swirl), the shield guarding the antagonist (Shield), the drone chasing the player (Drone), the player's basic munition (Missile), and finally the player's big weapon (Zorlon Cannon). The shiny thing in the centre, put there for visual excitement (called the Ion Zone), did not exist yet in my head.

First, I have to get the game to first playable and see if what I have is worth pursuing. The polish and animation will come later. My primary implementation goal is to make a game I'd enjoy playing. That's the great thing about action games – I can have (and honestly evaluate) the player experience.

In *Star Castle* (the arcade game that formed the basis for *Yars' Revenge*) the player's missile is enough to kill the monster once you penetrate the shield. That didn't seem challenging enough, which is why I added a second player weapon with a unique aiming mechanic. But in game design, each choice has its consequence. If you add an extra weapon, you may need an extra activation mechanism.

One major decision in every video game is the control scheme. I was using the 2600 joystick, which is a simple four-switch stick with one button. The fact that you could close adjacent

switches simultaneously yielding eight different stick positions was the gravy!

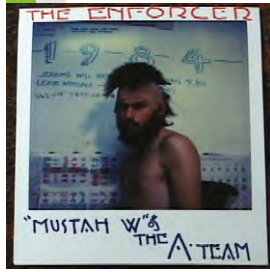
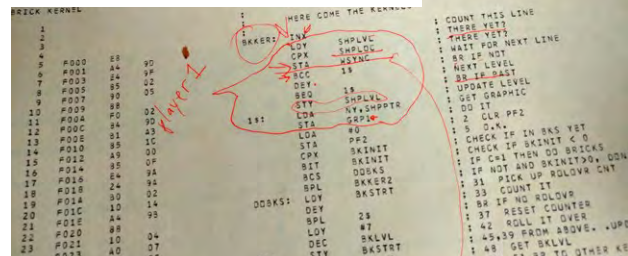
Star Castle used the standard four-button *Asteroids* setup: rotate right, rotate left, thrust and fire. Choosing the fire button, then, is easy. Then I'll just use stick right and left for rotation and stick forward for thrust. That leaves stick back free, which will come in handy for activating the Zorlon Cannon. Now I have all my game mechanics covered.

After months of programming, I got to first playable, wherein all game elements are reasonably represented and all rules implemented. Finally, I got to play my first game for the first time. I tried it out for a while, and I asked a few other people to play it, and here's what I learned:

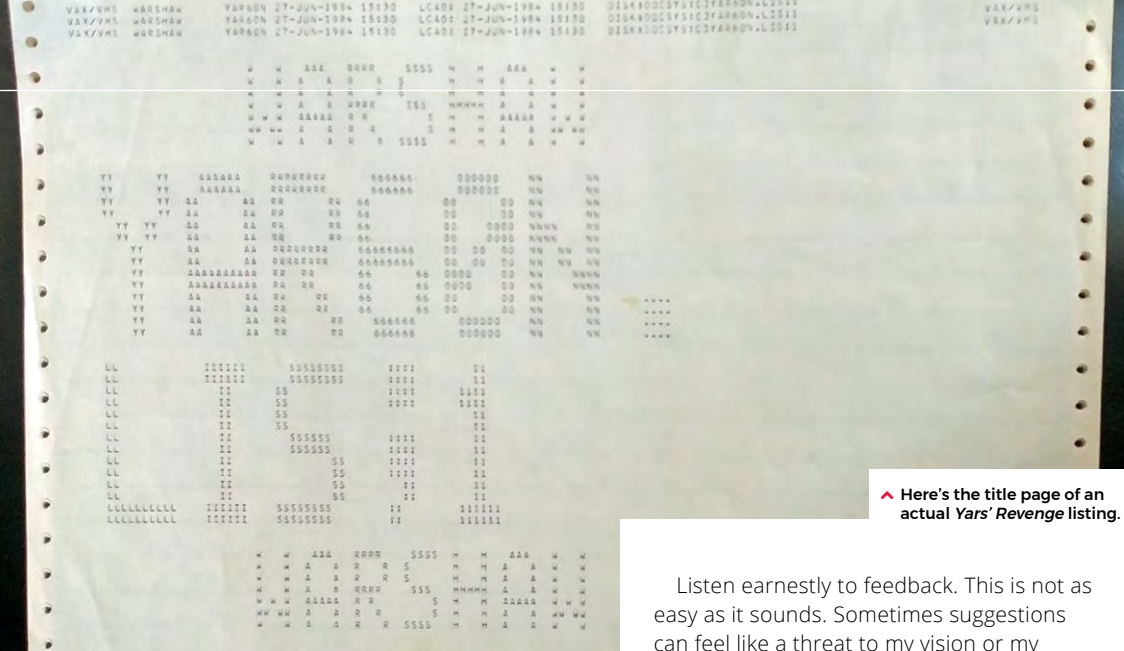
I needed this game to be great... and it wasn't. Something had to change, and I didn't know what. This was a dark time for me.

The control scheme, though true to the original coin-op game, was cumbersome and didn't translate well to my new screen geography. Steve, my manager at the time, suggested simply allowing the player to move in the joystick direction. It felt much nicer, but it presented two problems: how to control speed and how to produce the Zorlon Cannon.

That's machine code on the left, assembly language instructions in the middle, and my running commentary on the right.



Howard Scott Warshaw, back in Atari's halcyon days.



^ Here's the title page of an actual *Yars' Revenge* listing.

If you can't get the cannon, you can't succeed in the game, which is a problem. But clearly the current way sucks and must change. Interestingly, this poor controller scheme led me to a conceptual breakthrough in game design.

RISK AND REWARD

The speed problem was no big deal – I just made a flat speed when you move, and when you release the stick, you stop. It's more responsive, it allows the player to be accurate with motion, and I don't have to program any physics. The larger issue was this: if I take the cannon off the controller, how do I activate it? I'm out of buttons.

Then it hit me: Why not play your way to getting the cannon? Ordinarily, you shoot the shield with missile and cannon to expose the Qotile. What if there was another way? Bump up against the shield to remove a brick, and that activates the cannon. What if you run out of shield? You can also activate it by touching the Qotile. This meshes beautifully with another design trade-off I'd made: in order to attack the player, the Qotile occasionally morphs into the super-weapon, sparing me the need for another graphic element (which I don't have). This forces you to increase your risk any time you want to activate your big weapon, creating a lovely and well-motivated balance within the play: want bigger returns? Take bigger risks!

Those two changes brought the game to life. Suddenly people were excited about it. The transformation was amazing, as was the enormity of my relief. Like most designs, my basic concept was solid but raw. It needed a few magic ingredients to realise its potential. This taught me several key concepts of video game design.

"I needed this game to be great... and it wasn't. Something had to change"

Listen earnestly to feedback. This is not as easy as it sounds. Sometimes suggestions can feel like a threat to my vision or my implementation skills. Beware over-commitment to your concept! It helps to remember it's not your job to create everything, but rather to make the best choices from all the possibilities.

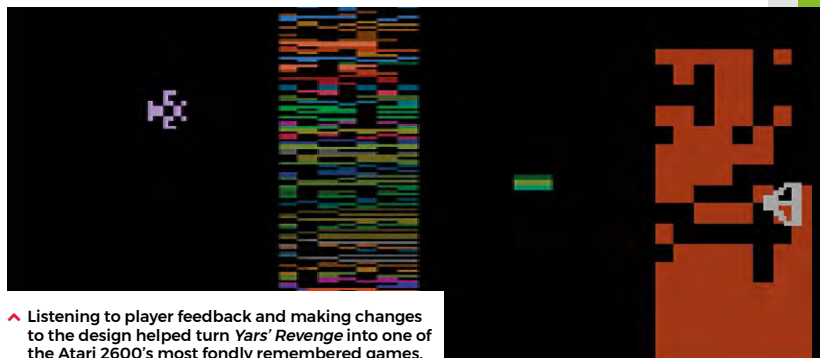
Any time you take a game mechanic off the controller and put it in the play, the game gets deeper and better. Today, button mashing *is* gameplay in some cases, but generally speaking, it can be more fun and rewarding to use a play sequence to accomplish a task.

Rumination time is essential to positive design evolution. First playable is a drive to meet the original spec, as well it should be. But thereafter, time is needed for nothingness. To be able to play a game until you're sick of it, let it go, then come back to it again with fresh eyes. Sometimes to move something forward you need to release it. It's a lesson which would come into sharp relief on one of my later projects.

This is how the game went from basic concept to solid play experience. Now it was 80 percent done – but with a long way still to go. Next time, I'll share with you some of the short cuts, long shots and finishing touches which helped *Yars' Revenge* go from a fun game to an enduring contribution to gaming history. 🍷

Keep it simple

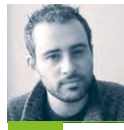
When it comes to creating solutions on the 2600, I adhered to Einstein's advice: "Make it as simple as possible, but no simpler." The glittering Ion Zone in the centre of the screen is the code from the cart, literally. I grabbed a byte of code and used it for both graphics and colour. No extra space needed, no extra fetch cycles and it's already pseudo-random. Atari's lawyers balked at displaying code, but it shipped. Sometimes the answer is right in front of you, even if you can't see it for the ones and zeroes.



^ Listening to player feedback and making changes to the design helped turn *Yars' Revenge* into one of the Atari 2600's most fondly remembered games.

CityCraft: Three ways of making game cities

Imaginary, real, fantastical or inspired by history, all virtual cities should be created equal



AUTHOR
KONSTANTINOS DIMOPOULOS

Konstantinos Dimopoulos is a game urbanist and designer, currently working on the *Virtual Cities* atlas, and consulting on several games. game-cities.com

The city of the future

Extrapolating from existing technological and societal trends is a method commonly used by science fiction writers and artists to build futuristic worlds, and it's something you can do yourself when designing a city. You too should decide on a socio-economic foundation – was your sci-fi settlement constructed by a classless society, a feudal system or something entirely new? – and then imagine the cities it would need, and their functions. Also consider how technologies would shape such a city; New York's skyline, for example, was made possible thanks to a single invention: the elevator.

A single, universally applicable approach to crafting game cities doesn't quite exist yet, but thankfully, the three ways I'll be discussing here should cover most kinds of virtual city building. They're broad methods that can tackle settlements in any type of setting, and any game genre. They're flexible enough to be employable in 3D or 2D environments, open worlds, adventures and action games, and of course also in fantasy, contemporary, sci-fi or historical settings.

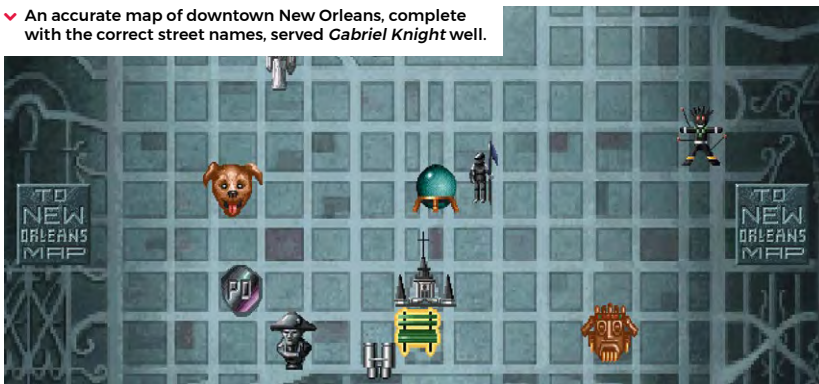
CHOOSE AN EXISTING CITY

The first option is to choose a richly documented, thoroughly mapped contemporary or historical city, and try to believably recreate it. This is what the *Assassin's Creed* series has

been doing with places such as Victorian London and ancient Athens, what the original *Gabriel Knight* masterfully achieved with New Orleans, or how the *Blackwell* adventures tackled New York. In these cases, detecting what the defining elements of each city are (or were) is crucial. Is it the distinctive architecture and atmosphere of New York? The flora, music and customs of New Orleans? The parks, factories and tense class relations of London? The landmarks or street patterns? It is up to the designer to correctly distil the essence of each city in as economical a manner as each genre's restrictions demand, recognise its main functions, recreate its defining characteristics and thoroughly research it.

Obviously, a 3D open world will be more demanding in terms of assets than a map and a selection of static, hand-painted locations, but in all cases, the city in question has to be abstracted to a manageable level. Recreating every building, person, road and festival is impossible. We have to apply a sort of cartographic generalisation to every aspect of reality. Just as cartographers decide what goes into a map based on its scale, so must we as designers. In a map, less information is presented as we zoom out. A city is thus initially shown as various buildings and roads, then as empty city blocks, then as an outline, and finally as a dot. At every level, only critical information is shown, and this is exactly the logic that needs to be applied to every layer of a game city.

✓ An accurate map of downtown New Orleans, complete with the correct street names, served *Gabriel Knight* well.





▲ Rubacava from *Grim Fandango*: the Land of the Dead is rarely believable, but it's a wonderfully dreamlike place.

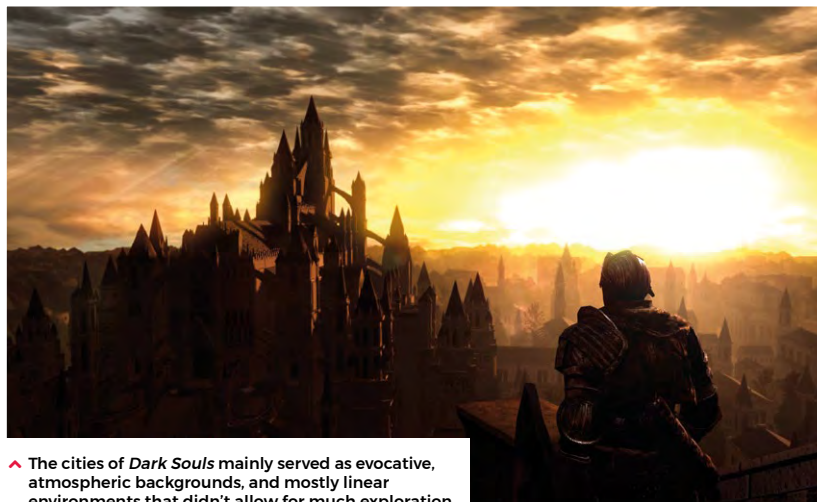
MODIFYING AND COMBINING CITIES

There's always the alternative of freeing the creative process from historical constraints while still keeping reality's convincing elements intact. In this case, renaming, repainting, heavily modifying and occasionally combining existing cities is the way forward. *Half Life 2*'s City 17 is a fine example of this, as it picked and mixed elements such as the Budapest Western railway station, and Sofia's courtyards, to create a new city. Even more famously, Terry Pratchett's fantastical Ankh-Morpork was based on Tallinn and Prague, but also incorporated elements of 18th century London, 19th century Seattle, and early 20th century New York. *Mafia III*'s New Bordeaux, on the other hand, is a modified, abstracted, subtly disguised New Orleans.

Keep in mind, though, that civic elements lifted from different eras and areas do not always play well with each other. Copying the architecture of a snowy mountain city, and applying it to a tropical island would simply be a mistake, whereas a Parisian boulevard would feel out of place in a medieval Germanic town. On the other hand, combining the twisty roads and unexpected plazas of Florence with the size and natural geography of New York – with the monumental scale of its dominant architecture – might lead to something spectacular indeed.

STARTING FROM SCRATCH

The most intriguing and demanding option would be to create a city entirely from scratch. A truly imaginary city should still borrow some of its elements from historical reality, literature or science, and artistically combine them to create something new and believable. For this you'd have to initially decide on the mythology and lore of your setting, and the universal rules that apply to it. Provide your creation with context, a wider geography and a hinterland. Is this a world of magic, 19th century Earth,



▲ The cities of *Dark Souls* mainly served as evocative, atmospheric backgrounds, and mostly linear environments that didn't allow for much exploration.

a futuristic universe of mutated humans, or a weird fantasy version of the pirate-infested Caribbean? Do nations exist? Wars? Also, how's the climate and the economy? Above all, answer the three crucial questions: when, where and how big the city will be.

Then comes the truly interesting part: make a detailed list of all the urban functions you imagine to be necessary (production, consumption, shelter, transportation, access to water and breathable air). Start crafting, and layering the place's history. Research real-world cities and dream up defining events for all its major eras. Has the old infrastructure survived? Is the cathedral still standing? Was it once safe to live by the river? Go on by attempting to roughly map the whole city, and its environs. Iterate. Write a paragraph or two describing the place. Possibly even commission some concept art, but always work from the general to the specific, and always consider urban structures, population densities and land uses. Follow this process, and even the most outlandish video game city will still feel like a living, breathing place. 🗺️

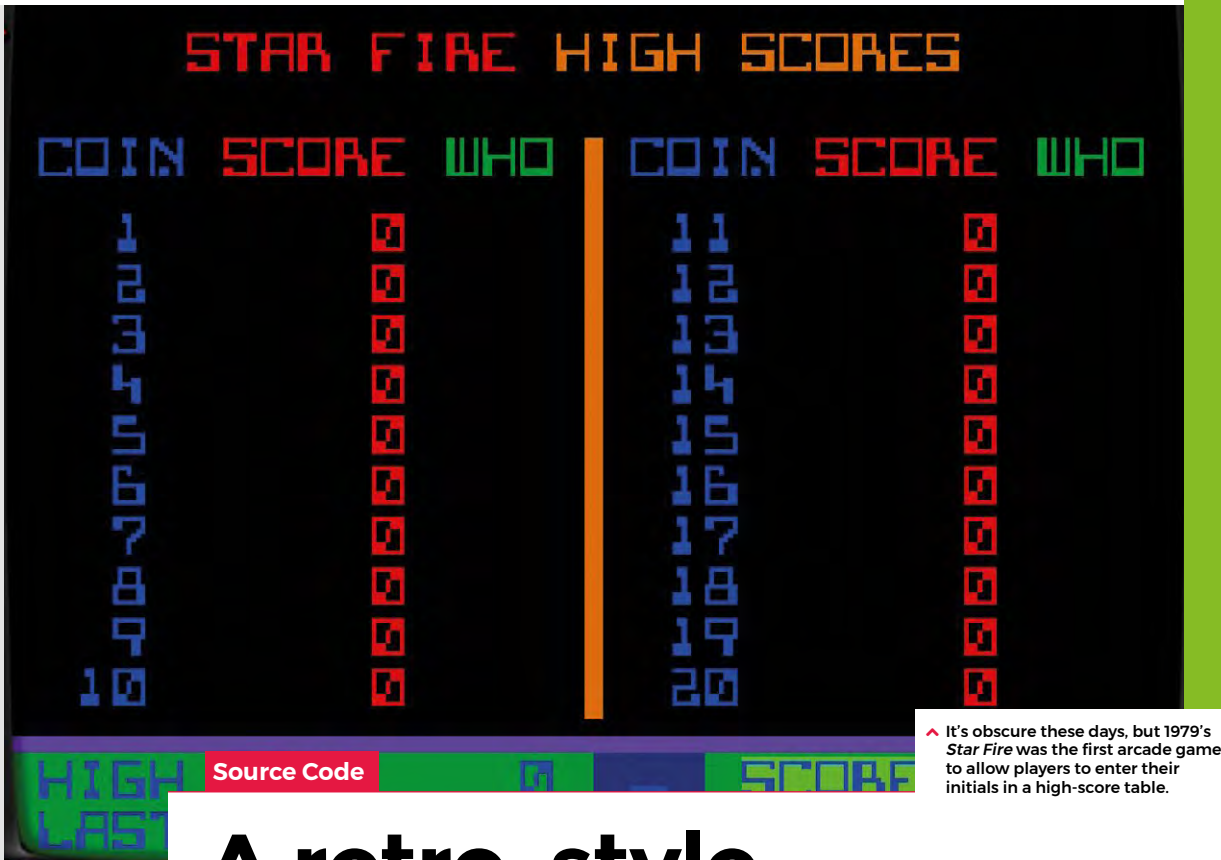


◀ It may have only been a backdrop, but the squalid Metro City did much to make Capcom's *Final Fight* so evocative.

“It is up to the designer to correctly distil the essence of each city”

The power of a backdrop

At first glance, *Final Fight*'s Metro City doesn't look like a crucial aspect of the game. It could easily be swapped for a spaceship filled with aliens, and the brawling action would remain essentially unchanged; *Metro*, like many settings in action-focused games, is effectively an elaborate background. And yet this background matters. It enhances the game by giving it an exaggerated yet familiar backdrop, a cohesive if simple story, and opportunities for violent encounters in elevators, city streets and grimy underground trains.



A retro-style high-score table

Here's how to create a high-score table that you can easily add to your own games



AUTHOR
RIK CROSS

First appearing in arcade games back in the late seventies, high-score tables add an additional challenge, and keep players coming back for more. In this article, I'll show you how to create a high-score table that you can add to your games.

A simple way to track a top score is to create a `highscore` variable. The `highscore` is initially set to 0, and at the end of each game a sufficiently high `score` becomes the new `highscore`.

The principle for creating a high-score table is the same, except now you need to store a list of scores and names, and at the end of each game you need to add the new `score` and `name` to the table if the score is high enough. The high-score table is stored as a list, containing `(score, name)` tuples. (In Python, a tuple can be thought of as a list whose elements cannot be changed.)

So a `highscores` list containing three `(score, name)` tuples might look like this:

```
highscores = [(100, 'Rik'), (86, 'Becci'), (45, 'Steve')]
```

Square brackets `[]` are used to access a particular element of a list, so `highscores[0]` is `(100, 'Rik')`, and `highscores[1]` is `(86, 'Becci')`, etc (remember we count lists starting at position 0). `score` is the first item (item 0) of each tuple and `name` is second item (item 1), and we use the same square bracket notation to access elements of a tuple. Therefore, you can access the first score in the list using `highscores[0][0]`, and the first name by using `highscores[0][1]`. It's best to have some scores to aspire to, so when creating the `highscores` list I'll populate it with ten low scores:

```
highscores = [(0, 'Player') for i in range(10)]
```

The code for checking a score and (optionally) adding it to the table will be put in a function called `addscore(score)`. This function will take a `score` as a parameter, and add to the `highscores` list if the score is higher than the lowest score in the list.

Firstly, there's no need to do anything if the score isn't high enough to make it into the table. If the `score` isn't higher than the tenth score in the list (the first tuple element of list element nine – `highscores[9][0]`) then the function returns without updating the table:

```
if score < highscores[9][0]:  
    return
```

If the function hasn't returned, then the score is to be added to the table, and next we need a name to attach to the score:

```
# get the player's name  
name = input('High score! What is your name?')
```


Next, the **score** and **name** need to be inserted into the **highscores** list at the correct position. To do this, we start with a variable called **pos** which is set to 0 (the first item in the list). This variable is then incremented until either we reach the end of the list, or until the player's score is higher than the next score in the list. Here, the value of **pos** is where we place the new score.

```
pos = 0
while pos < len(highscores) and score
<= highscores[pos][0]:
    pos += 1
```

To add the new **score** and **name** to the list, we next need to split the **highscores** list into two parts; all scores before the position to add, and all scores after the position to add. In Python, the shorthand syntax for all elements up to and including **pos** is **highscores[:pos]** and all items from **pos** to the end of the list is **highscores[pos:]**. We then add the **(score, name)** tuple to the list:

```
highscores = highscores[:pos] +
[(score, name)] + highscores[pos:]
```

We only need the top ten scores in the list, and the addition of another score means there are now eleven. To save just the top ten scores, we can use the same shorthand:

```
highscores = highscores[:10]
```

To print the scores, we first need to print the table headings, separated by a tab (**\t**). Then, we can loop through each tuple in the **highscores** list and print the **score** and the **name**, again separated by a tab. Using tabs rather than spaces keeps the data lined up.

```
def drawtabletext():
    # print the table headings
    print('Score\tName')
    # print score and name pairs in order
    for s in highscores:
        print("{0}\t{1}".format(s[0],s[1]))
```

As an additional challenge you could save the table to a file when the game is quit so it can be retrieved later. For a retro feel, you could also limit the player's name to three characters, using 'up' and 'down' to cycle through the alphabet! 🎮

High-score table in Python

Here's the full high-score table Python code. To get it running on your system, you'll first need to install Pygame Zero – you can find full instructions at wfmag.cc/XVzieD

```
# highscore list is initially filled with low scores
highscores = [(0, 'Player') for i in range(10)]

def addscore(score):
    global highscores
    # only add the score if it is greater than the
    # current lowest score in the highscores list
    if score < highscores[9][0]:
        return
    # get the player's name
    name = input('High score! What is your name?')
    # starting at 0, increment the 'pos' variable
    # until it's at the position to insert the score
    pos = 0
    while pos < len(highscores) and score <= highscores[pos][0]:
        pos += 1
    # add the (score, name) tuple
    # at the correct place in the list
    highscores = highscores[:pos] + [(score, name)] + highscores[pos:]
    # only store the top 10 scores in the list
    highscores = highscores[:10]

def drawtabletext():
    # print the table headings
    print('Score\tName')
    # print each score and name pair in order
    for s in highscores:
        print("{0}\t{1}".format(s[0],s[1]))

# prints the table in Pygame Zero
def drawtablepygame():
    # print the table headings
    screen.draw.text('Score', topleft=(50,50), fontsize=40)
    screen.draw.text('Name', topleft=(150,50), fontsize=40)
    # using 'enumerate()' gives the position of each tuple in the list
    # which is used to calculate the vertical draw position of the data
    for pos, data in enumerate(highscores):
        screen.draw.text(str(data[0]), topleft=(50,100+(pos*50)), fontsize=40)
        screen.draw.text(data[1], topleft=(150,100+(pos*50)), fontsize=40)

def draw():
    drawtablepygame()

# use the 'addscore()' function to add some scores
addscore(64)
addscore(30)
addscore(87)

# print the populated table
drawtabletext()
```



Improving your Unity first-person shooter

If you followed our guide in Wireframe #3, you'll have the basis for a first-person shooter. Here's how to add enemies and more



AUTHOR
STUART FRASER

Stuart is a former designer and developer of high-profile games such as *RollerCoaster Tycoon 3*, and has also worked as a lecturer of games development.

In this second Unity tutorial, we're going to look at advancing the simple shooter we developed in issue three. In the first part, we focused on getting our player character in motion and adding the ability to shoot a basic projectile. Now we're going to add some basic AI, have a way to detect damage to them, and have points displayed. In addition to this, we'll add some basic 'quality of life' elements to the experience.

IMPROVING THE FIRST-PERSON AIMING

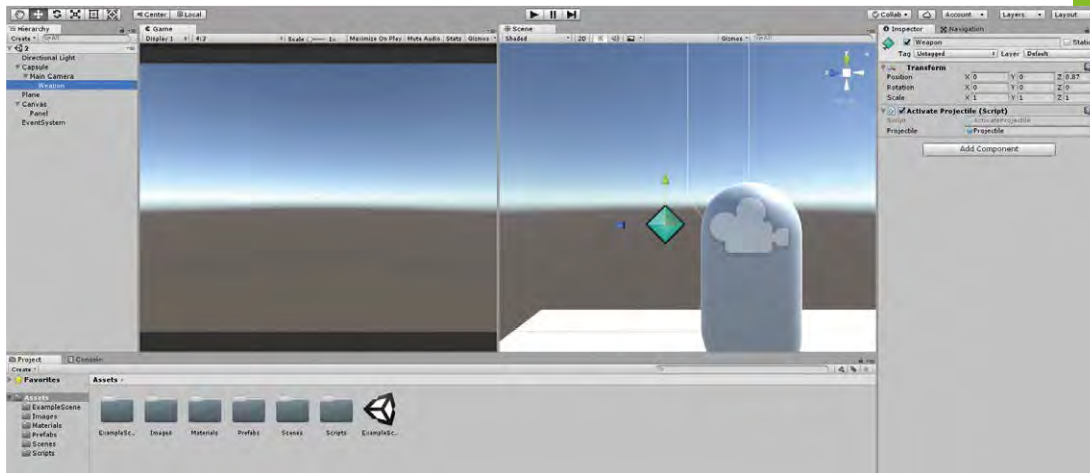
First, we're going to add a crosshair to the first-person camera – this will help the player with their aiming as they move and fire. We'll also make it so that the player can aim up and down. The first thing to do is to open the project; if you are using the Unity Hub then you should see a list of all the projects as soon as you load into the launcher. Select the applicable project, and then this should load the Unity Editor and the last scene you worked on.

Select the Hierarchy window to the left-hand side of the screen, and then right-click in an empty space and select UI > Canvas. Next, select the Canvas object in the Hierarchy, right-click, then select UI > Panel. You'll see that the Panel is linked to the Canvas game object. With the newly created Panel selected, look to the right-hand side of the editor and find the Inspector. In your Inspector, you should see the first pane is called Rect Transform. Select the icon that looks like a blue cross with arrows at each end. A new window will be shown called Anchor Presets. Select the icon in the centre that looks like a red cross with a red dot in the middle.

You should see there are now values in the Width and Height boxes; we want to type the

▼ Use the Anchor Presets to set the relative positions your UI element will be anchored to. This will be unaffected by changes to the screen resolution.





◀ To allow for vertical movement of our projectile weapon, we need to make the Weapon game object a child of our camera.

value of 3 into both boxes. You then need to change some parameters in the Image script component. First, select the box to the right of the word Color. This will bring up a colour selection tool similar to one in a photo manipulation package. We want to choose a bright colour – for example, a nice red. We also want to adjust the bottom slider prefixed by the letter A (for alpha) to be fully to the right-hand side. This will make the red dot be fully opaque, so you can easily see it in the centre of the screen. We now have a simple crosshair that should be rendered to the screen.

“We’re going to effectively make our shooter into a wave-based game”

The next change is to make the projectile fire from the centre point of where we’re looking. For this, go to the Hierarchy and select the Weapon game object we created last time, and drag this onto our Main Camera. If you can’t see these objects, you may need to click the down arrow next to your Capsule object.

The final change is to keep the Weapon game object selected, and then in the Inspector, set the Position for the X and Y to 0 for both. This will effectively set the object to be aligned to the Camera position. Feel free to try out the current iteration of what you’ve done by pressing the play button. You should see the crosshair and the projectile spawn from that point. Don’t forget to exit the preview by selecting the play button again.

SETTING UP OUR ENEMY

We’re going to effectively make our shooter into a wave-based game. For simplicity, we’ll have ‘zombie’ enemies who’ll do you damage on contact. We’re going to use the AI pathfinding

in Unity to create this logic, so the first thing to do is create a separate capsule to represent our enemy. In the Hierarchy window, right-click and select 3D Object > Capsule. With the new capsule selected, go into the Inspector panel, and rename the Capsule to a unique name. I’d suggest Zombie.

We’re also going to add a unique colour to the capsule to help differentiate it. First, we select the Project window and then right-click

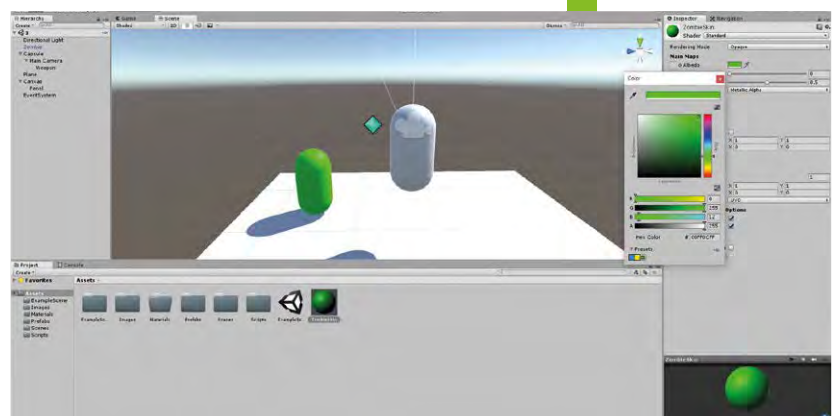
and select Create > Material, then add a unique name to the material – for example, Zombie Skin. With this material still selected,

go to the Inspector panel, and click on the box next to Albedo colour to open the Colour Palette window. You can try adjusting the colour on your material; I’ve gone with a bright green. Once you’ve set your material colour, select it from the Project window and drag onto the Zombie game object in the Hierarchy window. ➔

LOOKING FOR PART ONE?

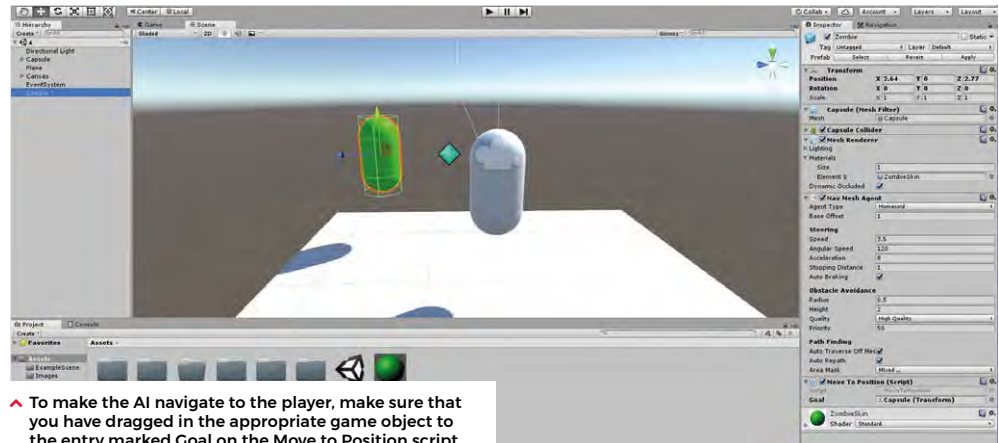
If you missed out on the first part of our tutorial, you can download a free PDF of Wireframe #3 from wfmag.cc/3, or you can download the tutorial as a Unity project from wfmag.cc/FPS001

▼ Once you have created a material, selecting the Albedo will allow you to set a colour of your choice via the colour wheel, sliders or a hex value.



A TIP FOR PREVIEWS

If you can't tell whether you're in a preview of your game or not, then there's a great setting you can use in Unity preferences. From the taskbar select Edit > Preferences... and from the new window choose Colors. There are lots of customisation options for colours here, but we want to select Playmode tint. Select the colour to open the colour palette window and simply choose an obviously different colour from the standard Unity grey. From now on you will always see this colour tint in a game preview.



▲ To make the AI navigate to the player, make sure that you have dragged in the appropriate game object to the entry marked Goal on the Move to Position script.

The final thing we need to do is to use a component on our enemy to make it navigate to the player.

To do this is easy enough: select the enemy in the Hierarchy, and then in the Inspector, select Add Component and choose Navigation > Nav Mesh Agent.

While we now have our navigation agent, we will also need to tell it where we want it to go. It's just a case of creating a simple script to drive the AI character.

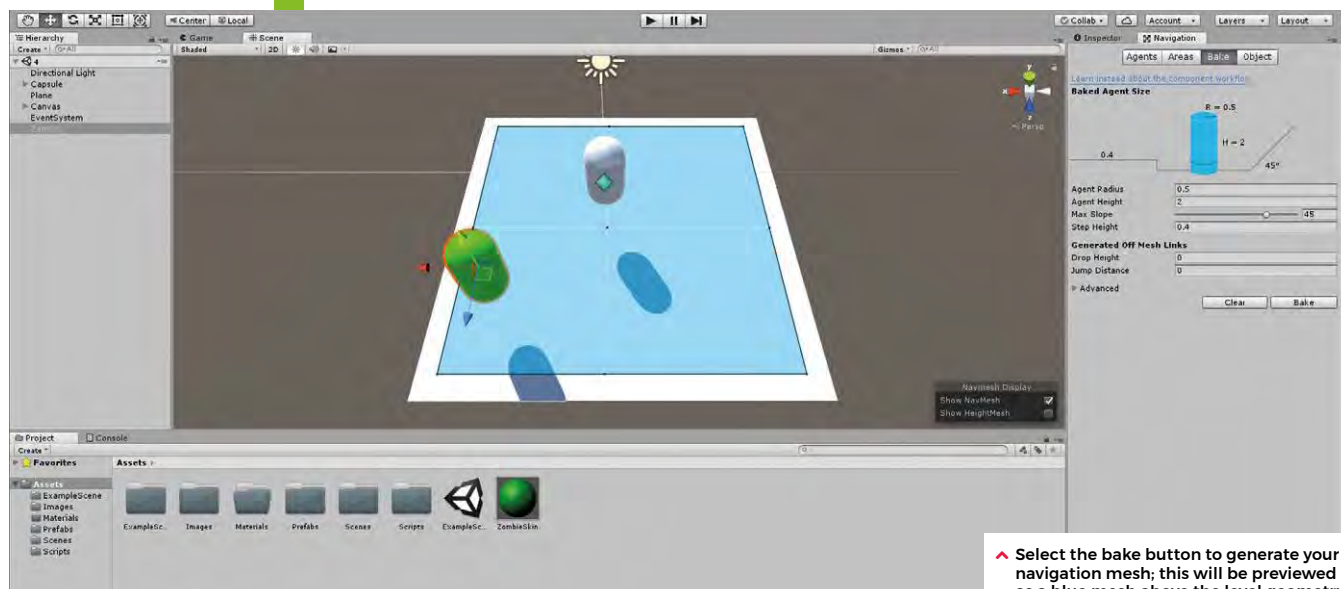
First of all, select Add Component and scroll down to the New Script entry. Then, in the next window, set the script name to **MoveToPosition**, double-click the new script on the Inspector and the script editor should load. Now, simply replace the template script with the code below.

```
using UnityEngine;
using UnityEngine.AI;

public class MoveToPosition : MonoBehaviour {

    public Transform goal;
    private NavMeshAgent agent;

    void Start()
    {
        agent = GetComponent<NavMeshAgent>();
    }
    void Update()
    {
        agent.SetDestination(goal.position);
    }
}
```



▲ Select the bake button to generate your navigation mesh; this will be previewed as a blue mesh above the level geometry.

Once you've saved the script and navigated back to the Unity editor, you'll see that the script will have a slot called Goal; this was made publicly accessible in our script above. What we need to do is set the Goal as our player. This is simply a case of selecting our player capsule in the Hierarchy window and dragging it to the slot to the right-hand side of the word Goal.

The final step is to select the Zombie in the Hierarchy window, and then drag this into the Project window to create our second Prefab object. The reason to do this is so we can drag in or spawn more copies of the same enemy.

GENERATING OUR ENEMY NAVIGATION

So, we have our enemy, and it has the navigation component and a script to tell it where to go. There is, however, one final and critical step that we must perform to make this all work. We need to create a navigation mesh. This is an invisible mesh that tries to contour itself to the floor of a game level. This is something that most modern games will use as a way to create AI navigation. In the example, we only have a flat plane, but let's imagine we have a sprawling level with steps, slopes, and so on. This will work out if the AI could acceptably make it to these places.

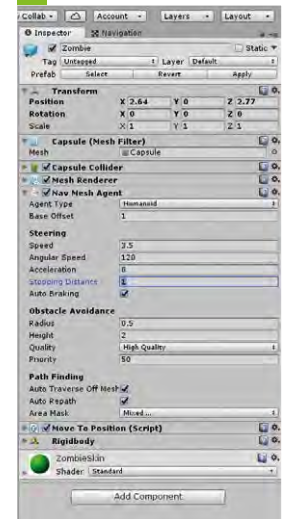
To use this feature, go to the taskbar and select Window > Navigation. It will open a new

tab which will appear where the Inspector is usually shown.

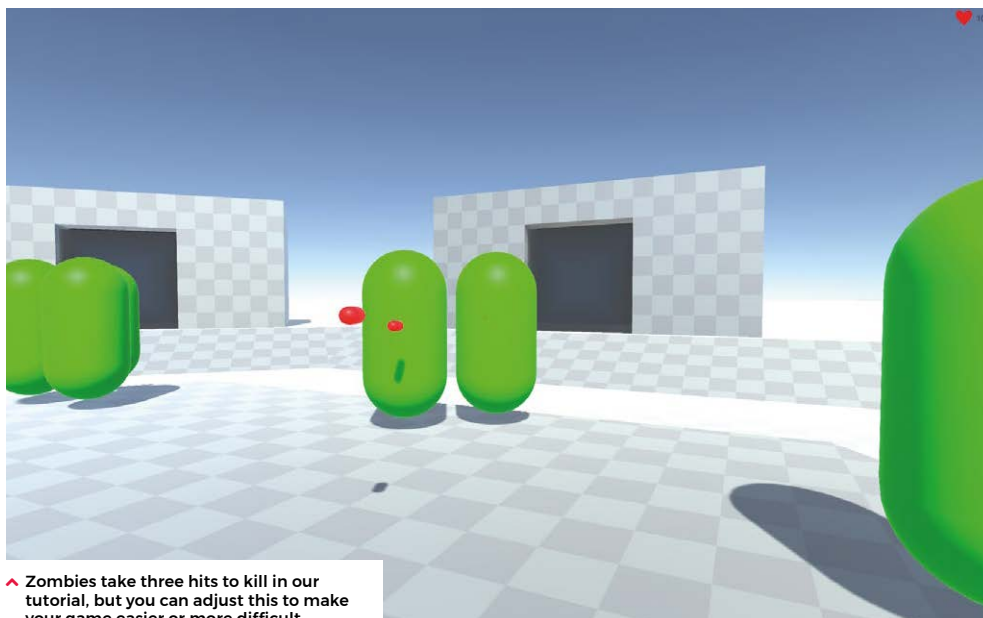
Now the issue is that we need to have a static object to bake the navigation. So, in the Hierarchy, select the Plane we created for our floor. We actually need to select the Inspector tab, so switch away from the Navigation tab for a second. In the Inspector for the Plane game object, you need to select the checkbox to the top-right called Static. We can now move back to the Navigation tab and from the row at the top of this window, select the Bake option. Finally, select the Bake button to the bottom-right of the window. You will now see the navigation mesh in a blue colour that will render just above the Plane mesh.

We're ready to test our enemy character navigation. All we need to do is use the transform gizmo tool to move the position so that it doesn't start on top of the player character. Simply select the red or blue axis and they should highlight. Drag these until you're happy with the position. When we press play to preview, you'll see that the AI will navigate to our player's position.

One issue we can see straight away is that the enemy navigates to the same point as the player, and we can see some odd behaviour in this interaction. To fix this, we need to stop the game from playing and then select the enemy game object. In the Inspector, look at the Nav



▲ Change the stopping distance for the Nav Mesh Agent to stop the AI character moving onto the same spot as the player.

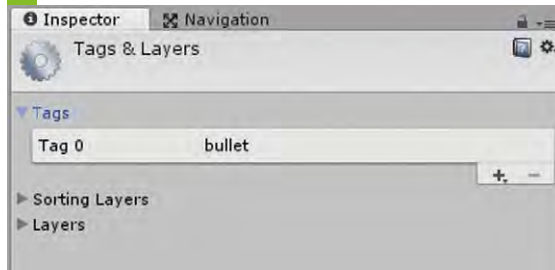


▲ Zombies take three hits to kill in our tutorial, but you can adjust this to make your game easier or more difficult.

OBJECT AVOIDANCE

While we're using static objects to generate navigation, we can do the same for dynamic objects. To set an object to have object avoidance, you need to select the game object and then use Add Component to add the Nav Mesh Obstacle component. There are various settings here, but you must remember to check the carve option to make it work with your navigation. Do remember that this is costly in terms of performance, so use it sparingly in your scenes.

► We want to add our own unique tag to our bullet by first creating a new tag in the Tags and Layers window.



Mesh Agent component, and then set the Stopping Distance value to around 1.0, as this will make the enemy stop just before it touches the player.

UNDERSTANDING TRIGGERS AND COLLISIONS

We now want to set up a way of testing for damage when the enemy collides with our player, and when the projectile hits the enemy. We're going to use an event trigger in both cases to achieve this. An example of a trigger in video games is when you walk into an area and this starts a cutscene. Effectively, there's an invisible box around the area, and by walking into it, the player sets off the event.

“The first challenge is to work out if the projectile has hit the enemy”

While I say we are using a trigger, we're actually going to test if we're colliding with an object, and set our events based on that. By default, in Unity, anything with a collider can be used to check if something is interacting with it. The only limitation with the event is that one of the two colliding objects needs a Rigidbody. As a rule of thumb, I would add a Rigidbody to any dynamic object.

We already added a Rigidbody to the player in the first tutorial, so let's add one to the enemy. In the Hierarchy, select the enemy game object. We then go to the Inspector and select Add Component and then select Physics > Rigidbody. I'd also expand the Rigidbody component and check the option Is Kinematic. This means while it has a Rigidbody, the Physics interactions of motions won't be applied, as we want this driven by our AI script.

Remember, we created the enemy as a Prefab. To make sure all the prefabs have the same properties we need to select Apply; this is to the top-right of the Inspector options, and should be below the Static checkbox and Layer drop-down.

We want to do the same with the projectile we created last time; in fact, we'll also be adding a

script to this. The script will check that the bullet object has hit an object with a collider; this will then disable or 'destroy' the game object. The first thing is that we never have our projectile in the scene. It's a prefab that is created when we are pressing the fire button.

We can still make changes to it by looking for it in the Project window. We do want to, however, expand the object by selecting the arrow to the right-hand side of the prefab icon. Now select the bullet mesh and you should see the capsule mesh listed in the Inspector. We want to select Add Component and repeat the process of adding a Rigidbody to the object. I recommend disabling the Use Gravity option on the Rigidbody, as while there would be some

gravity applied to a real bullet, this is much larger, and we want an arcade feel to the game.

As mentioned above, we want to add a script

to the object, so select Add Component and scroll down to New Script. Give this script the name **BulletHit**, then open it in the script editor. We can then copy the code below over the template Unity provides.

```
using UnityEngine;

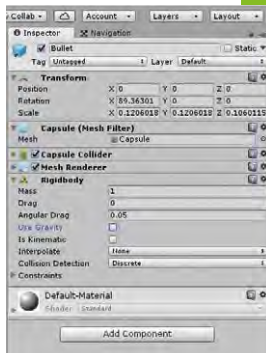
public class BulletHit : MonoBehaviour {

    //When we touch the collider we disable this object.
    void OnCollisionEnter()
    {
        gameObject.SetActive(false);
    }
}
```

Once you've saved the code and you're back in Unity editor, try previewing the game. You should be able to fire the projectile and it will disappear when hitting the enemy or the ground plane.

HANDLING OUR ENEMY DAMAGE

We now have our bullet understanding that if it collides with an object, we want it to be removed from the scene. Let's say we want to apply this logic to the enemy, but we want it only to be



▲ On the bullet prefab we want to disable the Use Gravity checkbox on the Rigidbody component.

damaged by the bullet, and we only want it to be destroyed after three bullets have hit it.

The first challenge is to work out if the projectile has hit the enemy. Unity has provided us with a way of marking up certain game objects to help us identify them. This feature is known as a tag, and we're going to add it to our projectile prefab. We need to go back to the Project window and find our Projectile prefab; if you need to expand it then do so. You should then select the Bullet mesh that we added the script to earlier.

Look at the top-left of the Inspector, and you should see the words Tag, and in the drop-down, it will show Untagged. Select the drop-down and then select Add Tag... and the Inspector will change to show Tags & Layers. We want to expand the Tags element by clicking the down arrow, and then you should see a + icon to the bottom-right of this element. Click the + icon and add your tag name, and then Save. I would type in 'bullet' for the name, but do note that this is case-sensitive, so it must be exactly the same in the code.

We need to re-select the Bullet mesh in the Project window to get back to our original Inspector view. You'll notice that, annoyingly, the tag wasn't added even though we just specified it. We just need to make sure that we select the Tag drop-down and change it to our bullet tag we set up.

We now need to turn our attention to the enemy and the script to track whether it's been hit by a bullet, and how many times it's been hit. If we count enough hits, we'll destroy the enemy. As usual, select the Zombie enemy in the Hierarchy and then in Inspector, select Add Component and choose New Script. We then name the script as **EnemyDamage**, and then add the code below.

```
using UnityEngine;

public class EnemyDamage : MonoBehaviour
{
    //Private means only this script can
    access the variable.
    private int hitNumber;

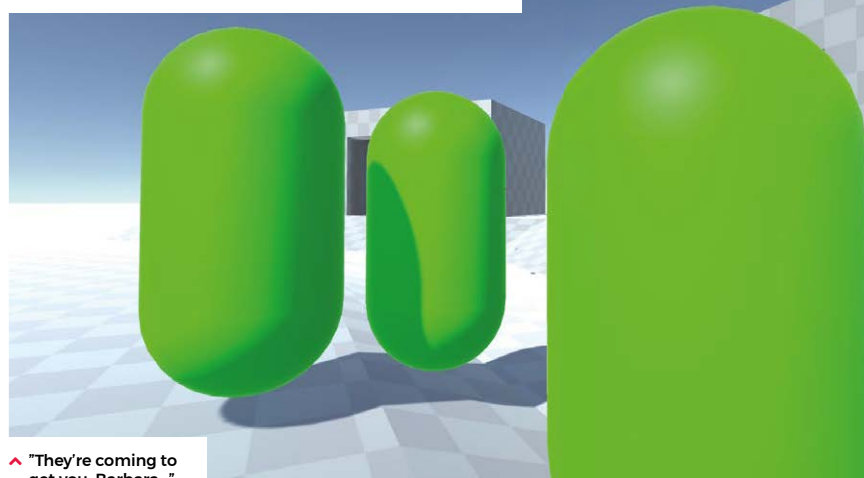
    //Unity stores the collider it hits and we
    can access it via the name other.
    void OnCollisionEnter(Collision other)
```

```
{
    //We compare the tag in the other
    object to the tag name we set earlier.
    if (other.transform.
CompareTag("bullet"))
    {
        //If the comparison is true, we
        increase the hit number.
        hitNumber++;
    }
    //if the hit number is equal to 3 we
    destroy this object.
    if (hitNumber == 3)
    {
        Destroy(gameObject);
    }
}
```

Do remember to save the code and then move back to the Unity Editor. If we play the game in preview mode now, we'll see that when we successfully hit the enemy three times, it'll be destroyed. Once you're happy that you've tested the feature, exit out of the play mode so we can continue.

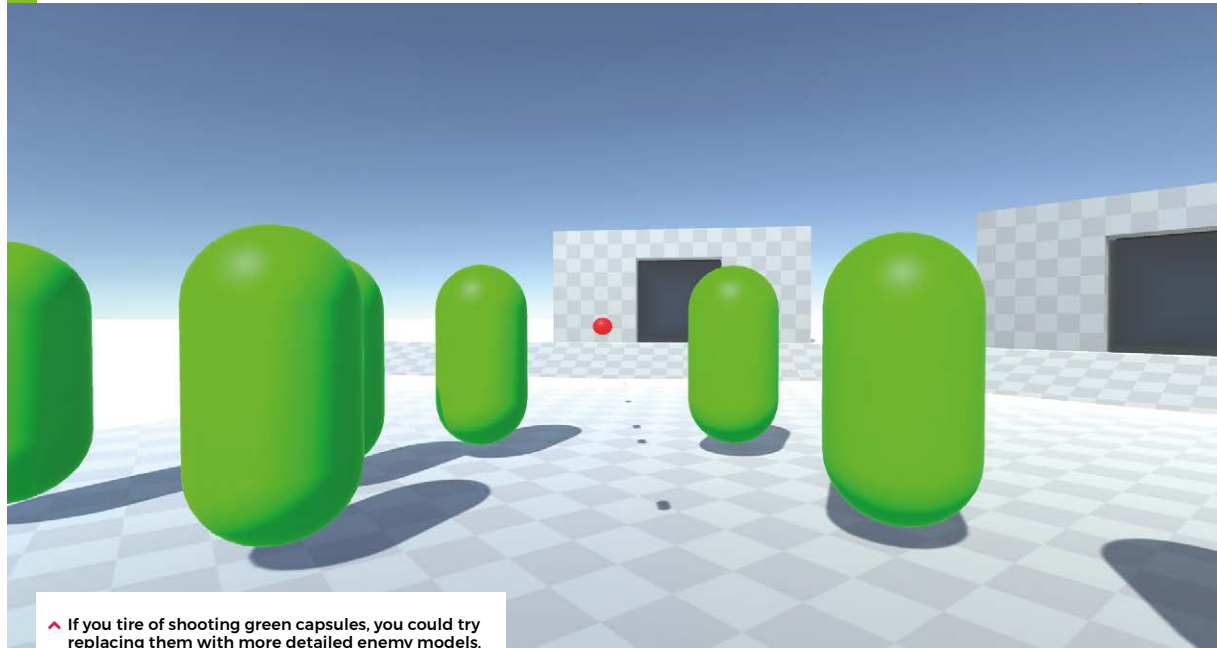
DISPLAYING DAMAGE TO THE PLAYER

We want our zombies to do damage when they touch the player game object, but we need a way of expressing the damage to our player. We're going to expand our canvas that we created →



COLLIDERS AND TRIGGERS

Games engines like Unity all have a concept of a collider, which is often an invisible, simplified version of the more detailed visual mesh you see in-game. The reason we use a collider is that it's processor-intensive to do collisions, so a simplified mesh – for example, a box – is less costly to calculate. You can find more about colliders and triggers in the Unity online documentation (wfmag.cc/RYXD). This also has a matrix of which components are required to create a collision event.



▲ If you tire of shooting green capsules, you could try replacing them with more detailed enemy models.

THE POWER OF TAGS

Tags are a powerful tool in Unity and help to improve performance. They allow you to quickly look for all objects with a tag and perform an operation on those objects. You could find all the objects individually, but this is an inefficient way of working and will slow performance of your game.

for the crosshair and add the player's health to the interface. We're also going to communicate between the enemy and the player. Effectively, we'll send a message from a script on the zombie that they're biting the player. This means that if we have multiple zombies, each one will apply damage at the appropriate times.

We're also going to reuse the tag functionality, but this time we don't need to create a new tag. First, select the player capsule in the Hierarchy window, and then in the Inspector, select the Tag drop-down, and select Player from the list. While we're in here, rename this object from Capsule to Player. It'll make it easier for us as we expand the game and make it clear to us which is our Player object. Another change I would make is ensure you also check the Freeze

“We'll send a message from a script on the zombie that they're biting the player”

Rotation for the Y rotation on the Rigidbody under Constraints.

Now we've done those changes, let's go back to our enemy object and add a new script to send a message from the zombie to our player. We use a command called **SendMessage** that Unity created to talk between game objects. It is by no means the only way to do this and, arguably, not the best way. However, it's simple to understand and effective enough for our needs.

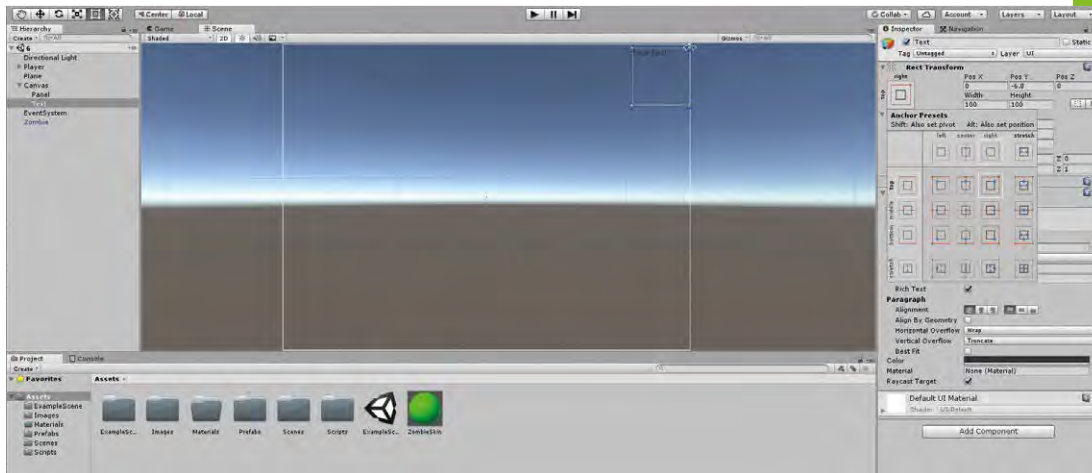
So, select the Zombie game object in the Hierarchy, and then in the Inspector, select Add Component, and then New Script. We will call our new script **SendDamage**, and then we can open the script in our code editor and replace with the script shown below. Do also remember to apply the prefab changes.

```
using UnityEngine;

public class SendDamage : MonoBehaviour {

    void OnCollisionStay(Collision other)
    {
        //We compare the tag in the other
        object to the tag name we set earlier.
        if (other.transform.
        CompareTag("Player"))
        {
            //If the above matches, then send
            a message to the other object.
            //This will also pass a value of 1
            for our damage.
            other.transform
            SendMessage("ApplyDamage", 1);
        }
    }
}
```

As I said above, we want to add some sort of output to the Canvas object we created to show the player health. We'll keep this really simple for



◀ We should use the Anchor Presets again, but we want to align the pivot for the text in the top-right corner.

now, and just display a health value of 100 to 0. In the Hierarchy we need to find the Canvas, and then right-click and select UI > Text.

You should see the default text string that says New Text in the viewport. We don't need to change the text as we will overwrite it in code, but we do need to reposition it. We'll use the Anchor Presets we used when setting up our crosshair. With the Text object selected, go over to the Inspector and once again select the icon that looks like a blue cross with arrows at each end.

We need to look for, but not select, the icon that is on the second row down and has a red dot on the top-right outer box. As we select this, hold the **SHIFT** key, and this will set the pivot rather than moving the position of the text. Now, back in the RectTransform, replace Pos X and Pos Y values with 0. Also, you need to set the Alignment option for the Text component by clicking the right-hand icon on the first row. The text is now neatly positioned in the top-right corner.

The final steps are to add some code to the player, and then link our Canvas to it so it correctly updates. Find the newly renamed Player object and select it in the Hierarchy. Move to the Inspector and select Add Component, and then select New Script. We'll call this new script **PlayerDamage**, and then open it ready for editing with our code.

```
using UnityEngine;
using UnityEngine.UI;
public class PlayerDamage : MonoBehaviour {
    //Use this to reference the text in the
    canvas
    public Text healthPanel;
    //Sets default health to 100
```

```
public int health = 100;
private void Start()
{
    //Sets the health text at the start,
we pass 0 as we don't want to remove health.
    ApplyDamage(0);
}
void ApplyDamage(int damage)
{
    //Checks we has attached a health
panel and our health is greater than 0
    if(healthPanel!=null && health>0)
    {
        //Stores the current health and
subtracts the damage value
        health = health - damage;
        //Sets the text on our panel.
        healthPanel.text = health.
    ToString();
    }
}
```

As usual, you'll want to save and return to the Unity Editor. We still, however, need to link our Canvas. Select the Player object and look in the Inspector – you'll see the script has two entries, one called Health Panel and one called Health. The Health entry is self-explanatory and has a default value of 100. The other is where we need to drag in our Text object, so go ahead and do that.

We should be ready for a test, but you may want to duplicate your Zombie enemy a few times and move them around. Once you're happy with the set-up, hit play and start your new zombie-blasting challenge. 🎮

WHY NOT TRY...

You may want to tweak things like the player health or number of Zombies to make for a better-balanced game. This is now in a stage that you can go to town on the level layout, too; feel free to expand the plane or even build in additional structures to the Unity scene using 3D Objects. Don't forget to bake your navigation and set the geometry to be static if you do modify the level structure.

GET INVOLVED

Do you have an online tutorial you'd like to share with readers? Have you created an online resource that other game developers might find useful? Maybe you have a local code club you're keen to promote? If you have something you'd like to see featured in the Directory, get in touch with us at wfmag.cc/hello

✓ Daniel Kvarfordt's *Pyxel Edit* is a solid and affordable application for creating sprites, and is particularly useful for generating tilesets.



Directory

Pixel art: tutorials and handy resources

If you're just starting on your first 2D retro game, here's a selection of tutorials, tools and resources to help you along

➤ **OpenGameArt.org**

You'll find a wealth of sprite and tile art to download here – just make sure you read the site's terms of licence, though, especially if you're planning on releasing your game commercially.

➤ **Spritters-resource.com**

Copyright laws mean you can't actually use the wealth of classic sprites archived here, but it's a great learning resource all the same; what better way to get a greater understanding of sprite design than to take a frame-by-frame look at the classics?

➤ **Lospec.com**

Although not solely devoted to pixel art, you'll find a range of tutorials, advice and utilities on the subject here. The palette list – a database of colour palettes from the likes of the NES and Commodore 64 – is incredibly useful all by itself.

➤ **Free pixel assets on itch.io**

As well as a platform for buying and selling indie games, itch.io hosts a range of assets that you can use in your own games, including animated pixel characters and background tilesets. wfmag.cc/rDMLsg

➤ **Derek Yu's pixel art tutorial**

It's years old now, but like a great piece of sprite design, Derek Yu's tutorial is pretty much timeless. Besides, Yu's the designer of *Spelunky*, so you're learning from a true expert. wfmag.cc/grKxK

➤ **Pixel art tilesets course**

Whether you're making a platformer or top-down RPG, this online course takes you through some of the techniques you'll need to get the most out of your pixel art. wfmag.cc/NXAH

➤ **Pyxel Edit**

Just about any free image editor can be used to create sprites, but Pyxel Edit offers some useful specialist features, including Tilemap exporting. It's available for PC or Mac, and will currently cost you \$9. wfmag.cc/GdXij

➤ **Aseprite**

Another sprite editor, Aseprite is particularly well suited for 2D animation. It's available for PC or Mac at \$14.99, or you can download the source code and compile it yourself for free. wfmag.cc/FQCZR

Wireframe

LIFTING THE LID ON VIDEO GAMES

Download the app

Out now for smartphones & tablets



Save
45%
with an annual
subscription

£1.99 or **£34.99**
rolling subscription subscribe for a year



CRATE

EXPECTATIONS:

THE REMAKING OF

Crash Bandicoot

WRITTEN BY
AARON POTTER

With 3D platformers on the rise again, Vicarious Visions reveal how they remastered a trilogy of Crash Bandicoot games for a new generation

It was in 2015, at Sony's second PlayStation Experience event in San Francisco, when rumours started to rumble that a certain platform-defining mascot might spin his way back from obscurity. Gleefully rubbing his hands together, SIE Worldwide Studios chairman Shawn Layden emerged on stage not donning his usual sharp suit, but a PlayStation-branded t-shirt featuring Crash Bandicoot. For most in the outside world, it was the first clue that, thanks to a partnership between PlayStation and Activision, Naughty Dog's platform hero was set to return. ▶





“We didn’t hesitate to say yes when they approached us to develop it,” says Dan Tanguay, creative director at Vicarious Visions, the developer tasked with re-creating the PS1’s original three *Crash Bandicoot* games: *Crash Bandicoot*, *Cortex Strikes Back*, and *Warped*. As a remastered package dubbed the *N. Sane Trilogy*, the collection eventually sold over 2.5 million copies worldwide by September 2017 – just three months following its initial release exclusively on PS4.

Restorations of older games – be they a straight port, scaled-up remaster, or full-blown remake – are plentiful today, thanks in small part to valid concerns surrounding

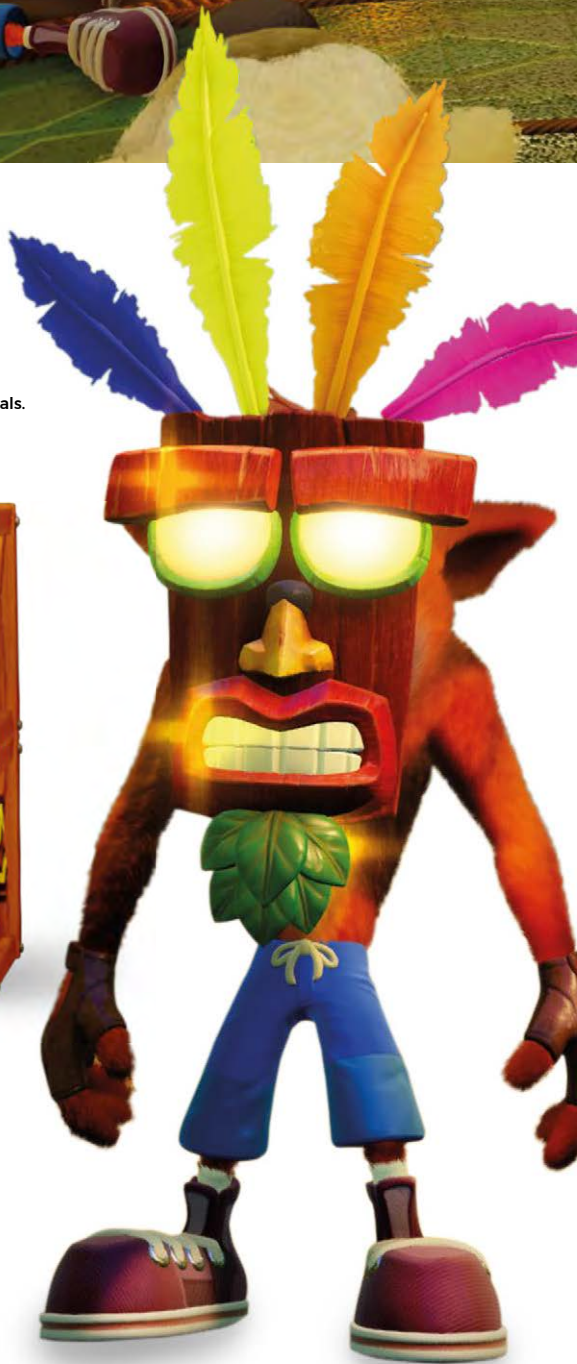
“We managed to get hold of all the games’ untextured 3D geometry”

game preservation, but mostly because of the gaming public’s seemingly unending hunger for nostalgia. In the case of *Crash Bandicoot N. Sane Trilogy*, meanwhile, the team themselves have a strong nostalgic connection to the character and his antics... “We’re huge Crash fans and have a long history working with him, dating back to *Crash Nitro Kart* on PS2 and even portable games like *Crash Bandicoot: The Huge Adventure*,” Tanguay says.

More recently, the studio had a hand in the mobile iterations of *Skylanders*, Activision’s toys-to-life series; it saw Crash emerge for the first time in five years by way of an in-game add-on pack. With the *N. Sane Trilogy*, however, Vicarious Visions faced the tough task of remaining true to the fun, frantic platforming action that old-school fans remembered – all while providing an experience that could be enjoyed by new players who might not have any pre-existing knowledge of how 3D platformers in the 1990s should operate.

This remit to stay as authentic as possible to the PS1 trilogy did at least give the team a

› Vicarious Visions coined the term ‘Fur K’ to celebrate the games’ transition to HD visuals.





▲ Dan Tanguay, the *N. Sane Trilogy*'s creative director.

starting foundation to work from. "We did two things right off the bat: figure out assets we had access to and research. That research helped guide any decisions we made," Tanguay reveals.

"A remaster is a different beast compared to a new IP – and no two remasters are built quite the same way either. For a new IP, you spend a lot of development time at the beginning of the project concepting the world, characters, and gameplay ideas. With a remaster, you have all of that at the start of the project. The trick is recapturing the spirit of the original."

FREE SPIRIT

For Tanguay and the rest of the team, this 'spirit' was distilled into three core areas: beauty, humour and charm, and fun. Speaking about the first category, producer Kyle Martin is quick to state just how graphically advanced the PS1 trilogy was considered at the time – emphasising the team's ambition to do the same almost 20 years later. "Simply increasing the resolution of the textures wouldn't do *Crash* justice," he says. "All of the *N. Sane Trilogy*'s art assets were lovingly created from the ground up."

It helped that Vicarious Visions were lucky enough to secure a limited amount of source files. This meant they were able to "access some of the original level terrain and some concept art and design documents," says Martin.

Nailing the eponymous bandicoot's wily personality and charm in the *N. Sane Trilogy* was another challenge. Whether it's the way Crash moves his eyebrows when psyching up for a polar bear ride, or the many taunts delivered by villains during boss stages, advancements in tech offered Vicarious Visions the chance to reassess and emphasise the series' wackier aspects.

"We wanted to ensure the trilogy captured the charm, humour, and zaniness of *Crash Bandicoot*," Martin explains. "Crash, his friends, and his foes needed to feel like they've been ripped out of a cartoon. We took great care



in studying the original *Crash* concept art and game design documents, as well as cartoons that inspired Naughty Dog when creating *Crash Bandicoot*. We knew that nailing the expressiveness of the characters, the exaggeration in their animations, and their unique personalities were all crucial to bringing the world of *Crash Bandicoot* to life."

Vicarious Visions' third focus, fun, similarly, led it to stay true to the classic PS1 trilogy's risk-reward approach. As Martin states: "It's always asking the players to try a little harder, while at the same time teaching them that it's okay to fail – just keep trying." It was clear to them that what gamers enjoyed about *Crash Bandicoot*'s sometimes brutally hard levels was the knowledge that they'd fail a lot in pursuit of that perfect run, but be rewarded with tons of unique death animations while doing so."

EXPERT MODE

With their goals set, Vicarious Visions got to work remaking three of the PS1's most beloved games. But knowing that people's expectations would be high, being meticulous at a technical (as well as visual) level proved vital. "We managed to get a hold of all the games' untextured 3D geometry," Tanguay reiterates. "Once we decoded it, this geometry became our Rosetta Stone. From it, we derived scale, layouts and rough collision. Most importantly, it gave us the metrics to remaster *Crash*'s controls and handling."

This led to a rinse-and-repeat process of constant iteration internally; boiling down mechanical factors such as *Crash*'s jump height and run speed to a precise degree so they appeared identical to the original games. "We'd run side-by-side video comparisons of our build alongside the original game to confirm animation states, timing, and so on. We also had a team of *Crash* experts on staff, and they tested every single aspect of the gameplay." →



VICARIOUS VISIONS

In an industry where studios come and go, Vicarious Visions has enjoyed a long and busy history. Founded in 1990, the New York-based studio has steadily worked away on a variety of licensed titles; in 2005, it was purchased by Activision, and has worked on a number of its titles since, including adaptations of *Guitar Hero* games and a variety of entries in the *Skylanders* series. After finishing the *N. Sane Trilogy*, though, the studio embarked on something a little more hard-edged: the PC version of Activision's multiplayer blast-fest, *Destiny 2*.

▼ The *N. Sane Trilogy*'s *Future Tense* is a wholly new stage made from the same mould as the PS1 games' levels.





^ VV recaptured Crash's zany personality using Naughty Dog's original cartoon influences.



^ Kyle Martin, producer at Vicarious Visions.

Arguably, you can't get better experts than those of the original developer, and while Martin admits that interaction with Naughty Dog staff was "quite limited", its seal of approval served as a strong vote of confidence during development. "Through Sony, some at Naughty Dog did have a chance to play the *N. Sane Trilogy* at certain points in development," he says. "We were ecstatic to hear that they were impressed and gave it two thumbs up."

Encouraging feedback aside, Tanguay confesses that it wasn't always plain sailing. It's no secret that this is true for the development of most games, whether they're a remake or not. So while some might assume that remaking a game is less taxing than creating something from scratch, Tanguay assures us that this isn't the case.

"The biggest challenge was remastering the controls," he states. "There was an entire team dedicated to that over the course of the entire project. And it wasn't just the Crash controls. Crash rides hogs, polar bears, hoverboards, jet packs, motorcycles, biplanes..."

Each variable presented its own development hurdles to overcome. "While one team worked on remastering Crash himself, other teams got to work on recreating the rest of the game, including character models, level design, enemy behaviour, the user interfaces, and of course the gorgeous art and audio that populated each level."

ROOM FOR IMPROVEMENT

One obstacle that can be particularly tricky, in part due to its exclusivity to remakes and remasters, is knowing which original elements to leave in and new elements to introduce. After all, even though Vicarious Visions were keen



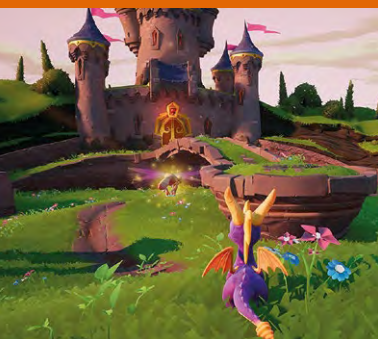
to remain faithful to how the *Crash Bandicoot* games looked and felt, the studio couldn't resist sprinkling in the odd feature that enhanced the core experience.

Tanguay highlights a small yet specific example: in the original three games, the pause menu varied between each; Tanguay's aim was to create a menu that was common to all of them. "We explored a lot of different ideas, including some very modern takes that would feel at home in any PS4 game. But we ultimately realised that the charm of having the menu slam together like a crushing block hazard was too important to get rid of. Once we came to that decision, we arrived at the final design pretty shortly after that."

Another way Vicarious Visions approached the trilogy was to ensure that any changes or additions they made were additive rather than destructive – in service of the original trilogy and not in spite of it. An overhauled menu system aside, this inspired the team to make Coco, Crash's beloved girlfriend, playable for the first time; in addition, time trials were made available in all three games, whereas on PS1 they were only in *Crash Bandicoot: Warped*.

One of the clearest steps to take for Martin was to realise some ideas that were considered but never executed in the original games' releases. "Once we had the core titles recreated, we couldn't resist the opportunity to give the fans new levels to play," he says.

Stormy Ascent was a notoriously difficult level that was cut from the original *Crash Bandicoot*, precisely due to Naughty Dog's fear that it proved too challenging. The *N. Sane Trilogy* acted as the perfect opportunity for Vicarious Visions to reinstate this missing piece of the Bandicoot puzzle. "We rebuilt it from the ground up and hand-crafted a brand-new bonus area for it as well," Martin says.



PLATFORMING PARTNER

Following Activision's success with *Crash Bandicoot N. Sane Trilogy*, its eyes quickly turned to another platforming icon from the PS1 era: *Spyro The Dragon*. Developed by long-time *Skylanders* developer Toys for Bob, *Spyro Reignited Trilogy* completely overhauls the visuals of the original game, *Ripto's Rage*, and *Year of the Dragon* – featuring the return of *Spyro* voice actor Tom Kenny and composer Stewart Copeland. Like *Vicarious Visions'* pursuit for authenticity, Toys for Bob took special care to recreate every memorable detail and asset bar a new menu track by Copeland, which mashes together the music of all three games.



▼ Whereas only the PS1 original featured time trials before, the *N. Sane Trilogy* added them to the sequels.



▲ Crash now looks better than ever thanks to HD textures and improved lighting.

CRASHING AHEAD

Following a year's exclusivity on PS4 due to the agreement between PlayStation and Activision, *Crash Bandicoot N. Sane Trilogy* eventually saw the light of day in June 2018 on Xbox One, Switch and PC – the first time the original three games have gone multiplatform. To mark the occasion, Vicarious Visions sought to one-up itself yet again.

"We wanted to try our hands at crafting an entirely new level for the *N. Sane Trilogy*," Martin reveals. This new level, Future Tense, marked the first time a Crash fan could play the original trilogy, load up a new level, and not know what to expect. "Future Tense provides players with many different layers of challenges; players have to dodge rockets, destroy robots, and leap over lasers while ascending a massive futuristic skyscraper. But it didn't feel right to create a brand-new level without at least a small nod to *Crash Bandicoot* development lore, so Future Tense pays homage to the unreleased Waterfall level from the original *Crash Bandicoot* game."

As for its reception, Martin is clearly happy with the reaction from players. "The response to Future Tense has been incredibly positive, so taking the chance to craft a new level for the original trilogy was totally worth it," he says.

Like it or not, the *N. Sane Trilogy's* success feeds into the games industry's continuing fascination with remakes – a trend that, for

now at least, shows little sign of slowing down. None of this is lost on the team at Vicarious Visions, whose sister company Toys For Bob is responsible for this year's equally-ambitious *Spyro: Reignited Trilogy*. When pressed on his thoughts on the matter, Martin insists that, when handled carefully, remasters and remakes can be a force for good. "It's important to keep gaming history alive and accessible to new players, especially if it is one of gaming's greatest heroes," he says.

"Developing the *N. Sane Trilogy* wasn't just about remastering the visuals of the original trilogy," he continues, using his own experience as a case in point. "It was about lovingly

recreating every aspect of the original trilogy from the ground up."

Ultimately, it'd be wrong for players who may have missed

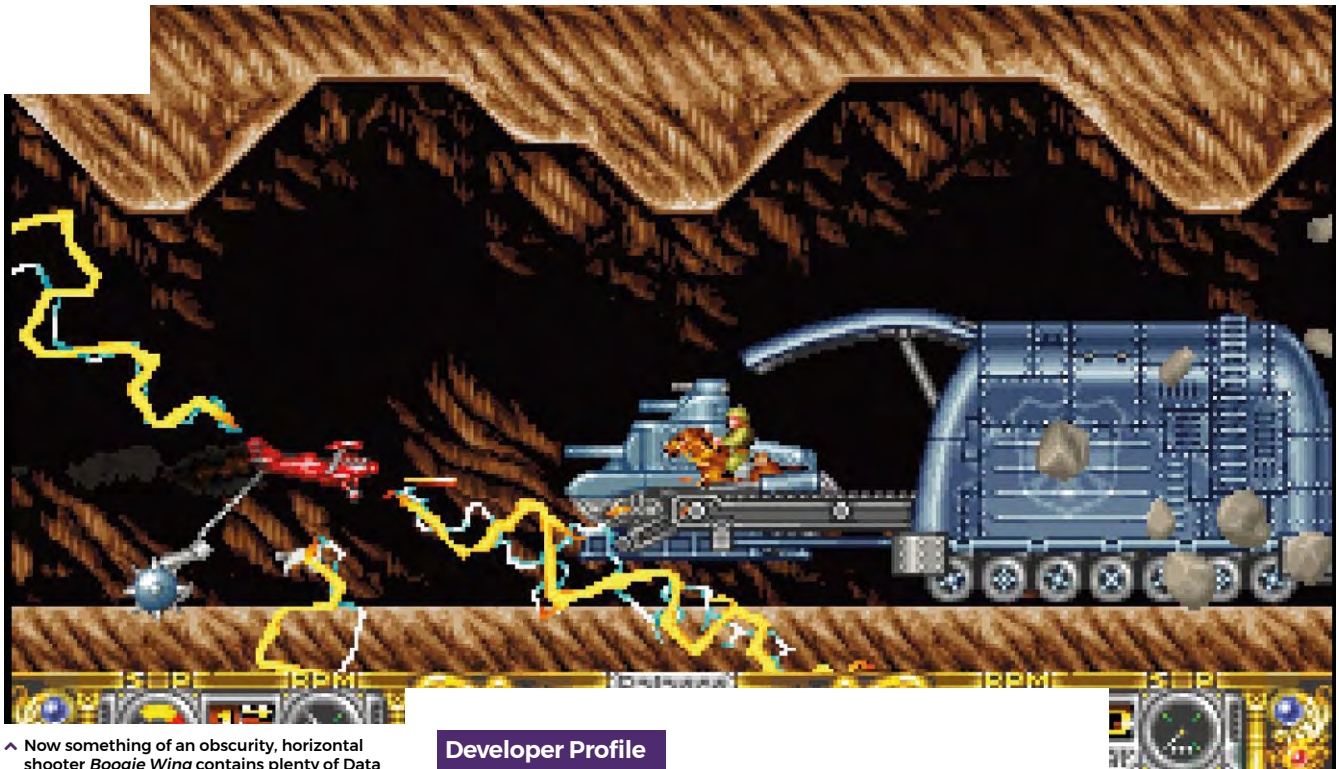
out on the classics to be denied the ability to easily experience them; especially those – like *Crash Bandicoot* and *Spyro* – from the 64-bit generation where they're currently tricky to emulate. Besides, there's something to be said about seeing a much-loved game updated to look and feel just how players remember it.

Reflecting on *Crash Bandicoot N. Sane Trilogy's* reception, Dan Tanguay remains humbled by the scale of it all. "It still doesn't seem real sometimes. We put our hearts and souls into this remaster, and it's been amazing for the whole team to see all the hard work pay off." 🗨



▲ For the first time ever, Crash's girlfriend Coco was a playable character in all three games.





^ Now something of an obscurity, horizontal shooter *Boogie Wing* contains plenty of Data East's eccentric design ideas.

Developer Profile

Data East

BurgerTime, Karnov and Bad Dudes' creators were a plucky mainstay of the 1980s and 1990s arcade scene

There are occasions when a developer's oddness doesn't become apparent until much later. In its 1980s pomp, the output of Tokyo-based Data East (or DECO, as it was also known) felt of a piece with most other arcade games emerging from Japan at the time. Largely action-focused, they ranged from side-scrolling brawlers to space-themed shooters, but Data East's games were frequently garnished with a hint of surrealism or playful mischief; 1988 brawler *Bad Dudes Vs. DragonNinja*, with its bizarre roster of characters and save-President-Reagan plot, feels in retrospect like a parody of the whole eighties era of straight-to-video martial arts movies. Then there was *Karnov*, an action platformer that featured a bald, rotund fire-breathing Russian as its protagonist – according to

a 2002 developer interview translated by Shmuplations.com, *Karnov's* hero was dreamt up solely to poke fun at one of Data East's bosses. Understandably, that boss wasn't too impressed when he discovered that the burly lead was based on him.

Nor was this the only time Data East's odd creative flourishes landed it in trouble. In 1988, the company released *Chelnov* – also known by the more ungainly title *Atomic Runner Chelnov: Fighting Human Power Plant*. With a backstory about a Russian miner caught in a nuclear accident, and the word Chelnov so closely resembling 'Chernobyl' when rendered in Katakana, the game was condemned by at least one Japanese outlet for trivialising the Russian nuclear disaster that hit the headlines two years earlier. Data East tried to distance themselves from the

^ Released in 1988, the *RoboCop* arcade tie-in was one of Data East's bigger hits.





^ Capcom were less than impressed by *Fighter's History*, Data East's answer to *Street Fighter II*.

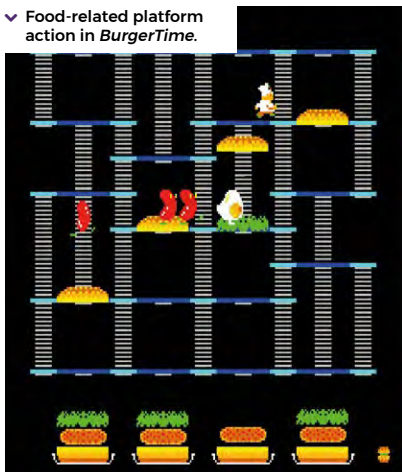
accusations, and western releases of the 1992 Mega Drive version downplayed the arcade game's Soviet themes.

SPARKING GENRES

In a way, games like *Bad Dudes*, *Karnov* and *Chelnov* were the reflection of a small company largely populated by young designers. Data East had been around since the arcade's first flourish in the 1970s, but even at its eighties peak, the firm only had around 200 staff – not a tiny company, but still far smaller than many Japanese competitors of the time, such as Konami, Sega and Capcom. This might explain why, even though the quality of its games varied, Data East's output always had a distinct flavour – whether it was the idiosyncrasies of its character design, its games' curious plots and dialogue, or the speed and difficulty of their action.

Data East's seventies and early eighties games were frequently clones of existing games, but *BurgerTime*, a single-screen platformer released to arcades in 1982, was one of its most fondly-

^ Food-related platform action in *BurgerTime*.



^ *Bad Dudes*, perhaps the most eighties of coin-ops, saw you rescue Ronald Reagan himself from the villains.



^ *Karnov* wasn't the best Data East game, but its hero became an unofficial mascot for the company.



remembered early hits, and perhaps the first game to show flashes of Data East's humour. The company's more significant contribution to gaming from the era, though, was arguably 1984's *Karate Champ* – featuring a pair of (for the time) large opponents who fought one another with a variety of punches, kicks and feints, it sparked an entire genre of one-on-one fighting games.

In its wake, a number of other studios made similar martial arts games of their own; Konami released *Yie Ar Kung-Fu*, with its roster of colourful opponents, in 1985, while designer Archer Maclean

“Even in its heyday, Data East was something of an underdog”

created *International Karate* for the Commodore 64 that same year. This latter game, quickly ported to other computers and released by publisher Epyx as *World Karate Championship*, would prove to be a pretty major bone of legal contention; Data East attempted to sue Epyx for copyright infringement in 1988, but the case was eventually dismissed when the court concluded that the two games were not substantially similar. Whether the ruling was fair or not, Data East could hardly have held back the wave it had inadvertently started. Later games like Capcom's *Street Fighter* series owed a debt to *Karate Champ*, but evolved its ideas almost beyond recognition. (Ironically, Capcom would later attempt to sue Data East when it noted how

Hard-wearing

Although better known for its games, Data East also dabbled in hardware – most obviously a string of popular licensed pinball machines – and also an earlier, more obscure venture called DECO Cassette System. Making its debut in 1980, the DECO system was an early attempt to make games easy and more affordable for arcade owners. With a standard cabinet and games that could be switched in and out on cheap-to-produce audio tapes, the DECO system sounded like a solid idea, but it failed to gain long-lasting traction – the tech was, sadly, less than reliable and prone to piracy. Flawed though it was, the DECO system nevertheless predated SNK's conceptually similar Multi Video System by almost a decade.

similar the latter's *Fighter's History* was to *Street Fighter*; this case was also dismissed.)

Having ceased making games entirely by the end of the 1990s, Data East finally closed in 2003 – a victim of a shrinking arcade market and changing audience appetites. Even in its heyday, Data East was something of an underdog, but it was uncannily adept at making distinctive, quick-fix action games that entertained in the moment and stuck in the mind for years afterwards. 🍷

Art DECO

10 of Data East's most memorable

Turns out DECO was behind a lot of fondly remembered titles



01

Atomic Runner (Chelnov)

Arcade / Mega Drive /
Sharp X68000 – 1988

An endless runner – of sorts – conjured into being well before we even had a name for the genre, *Chelnov* saw players endlessly running to the right (apart from in boss fights), leaping about and shooting enemies. While great fun in the arcade, it was actually bettered in the Mega Drive's home port – a rarity in the late eighties.



02

Bad Dudes Vs. DragonNinja

Arcade / Amiga / NES /
C64 / PC / others – 1988

A legend more for its out-of-game antics than the in-game action, *Bad Dudes* became a legend more because of both its name and the all-too-subtle use of 'President Ronnie' as the damsel in distress. Are you a bad enough dude to eat burgers with the commander-in-chief?



03

Midnight Resistance

Arcade / Amiga / Mega Drive / C64 /
Spectrum / others – 1989

A neat take on the Contra-alikes flooding the market in the late 1980s, this arcade shooter allowed players to shoot in different directions to the way they were moving – progress! *Midnight Resistance* also marked Data East's first attempt at a Mega Drive conversion with its port of *Chelnov* arriving later, in 1992.



04

RoboCop

Arcade / Amiga / NES / Game Boy / C64 /
Spectrum / others – 1988

Oddly enough, Data East had little to nothing to do with the home versions of this automaton policeman movie tie-in – all of which were infinitely more popular than the arcade original. Still, without its original for Ocean and co's ports to be based on, we wouldn't have *that* theme by Jonathan Dunn.

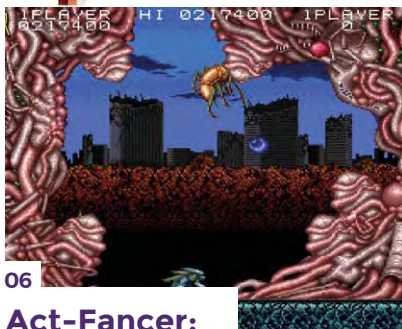


05

Sly Spy: Secret Agent

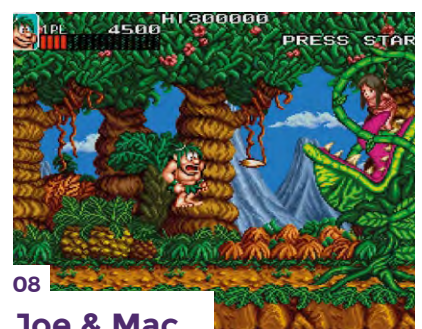
Arcade / Amiga / C64 /
Spectrum / others – 1989

While it might not feature someone acting as either a spy or in a sly fashion, *Sly Spy* still proved a solid, fun way to waste 15 minutes of your life. Trotting around as a Bond rip-off blasting invading terrorists on foot, on bikes, even in freefall was always going to be fun (if shallow), and... well, it was.



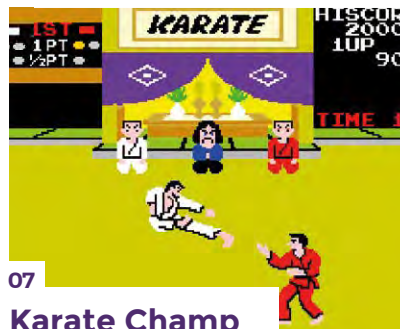
06 Act-Fancer: Cybernetick Hyper Weapon
Arcade – 1989

A side-scrolling shooter featuring a cyborg which can be powered up to blast its enemies is one thing. The fact it's one of the few Data East games only to appear in arcades is another. But really, you have to focus on that name. It's just... blissful. With some gloriously fleshy pixel art, this is pure eighties Japanese game joy.



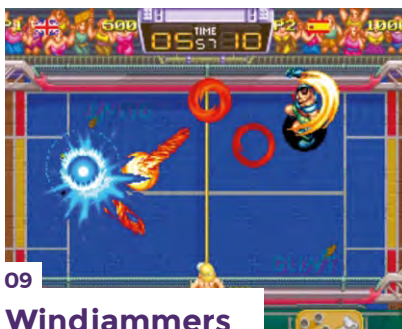
08 Joe & Mac
Arcade / SNES / Amiga / Mega Drive / NES / others – 1991

The cavemen ninjas rocked up on loads of machines back in the day, but it's the arcade and SNES versions that stayed under Data East's wing. Colourful, over the top platformers with just enough early nineties radditude to make them memorable, there's a special place in the hearts of many for these two prehistoric dudes.



07 Karate Champ
Arcade / C64 / NES / Apple II – 1984

It's hard to say if it started it all, but *Karate Champ* was definitely one of the earliest one-on-one fighting games released, kickstarting the trend towards *Street Fighter* and everything else that arrived in its wake. And we can't forget, of course, *Kung-Fu Master* – one of the first scrolling beat-'em-ups in existence, developed by Irem and put out by Data East.



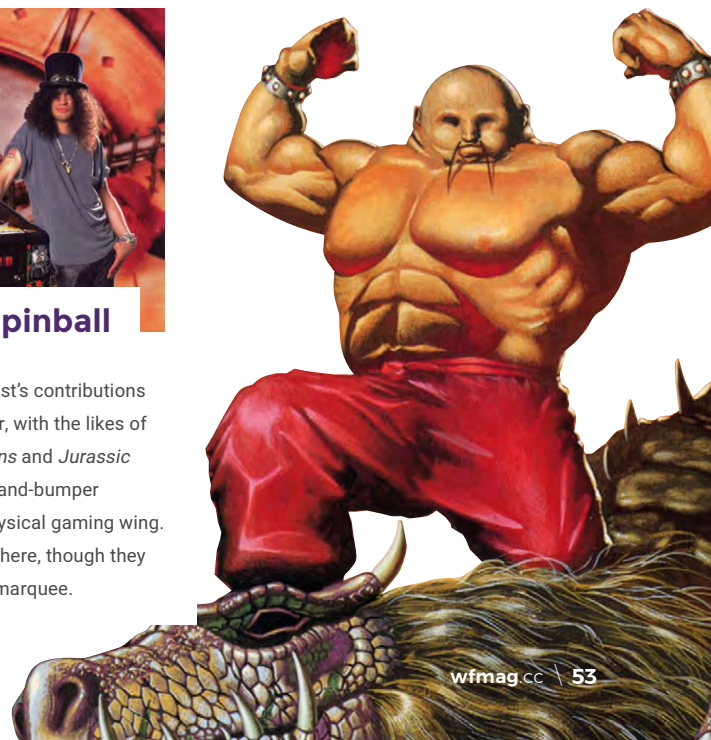
09 Windjammers
Arcade / Neo-Geo / Neo-Geo CD – 1994

One of the most beloved Neo Geo games, *Windjammers'* blend of ridiculous action and – brace yourselves – flying discs proved an instant hit with all who played it. In 2017 we finally saw the western re-release we'd begged for, and 2019 will see a – brace yourselves again – belated sequel. Data East, of course, will not be involved.



10 Guns N' Roses pinball
Pinball – 1994

We shouldn't ignore Data East's contributions to the world of pinball, either, with the likes of *Guns N' Roses*, *The Simpsons* and *Jurassic Park* all getting the chrome-and-bumper treatment by the studio's physical gaming wing. There were some greats in there, though they didn't all have Slash on the marquee.



Subscribe today



SAVE
49%

wfmag.cc/subscribe

Try 12 weeks for only £12

Subscriber benefits

- > **Free delivery**
Get it fast and for free
- > **Exclusive offers**
Great gifts, offers and discounts
- > **Great savings**
Save up to 49% compared to stores

Introductory offer

wfmag.cc/12weeks

- > **Low initial cost** (from £12)
- > **Cancel at any time**
- > **Free delivery to your door**
- > **Available worldwide**

Subscribe for 12 months

wfmag.cc/12months

£40 (UK) £75 (USA)

£65 (EU) £75 (RoW)

Offers and prices are subject to change at any time



Digital subscriptions from

£1.99



wfmag.cc/subscribe

Info

GENRE
Fighting

FORMAT
Switch

DEVELOPER
Bandai Namco /
Sora Ltd

PUBLISHER
Nintendo

PRICE
£49.99

RELEASE
Out now

Super Smash Bros. Ultimate

Sometimes, bigger does mean better

There's a lot riding on this latest release in the *Smash Bros.* franchise. The title alone promises the definitive incarnation of a series that's famed for being utterly stuffed with a staggering array of options, characters and stages. How on earth can this hope to meet the expectations of its... let's say, more 'dedicated' player base, and bring in newcomers?

Let's get the main criticism out of the way first here. The heart of the game remains unchanged from the previous instalment on the Wii U. It's still the same chaotic blend of fast-paced combat, random items and shifting stages. There are tweaks under the hood, but to be blunt, these will only be noticed by those dedicated players we mentioned earlier.

The other issue with *Ultimate* is that the Switch's Joy-Con are just too fiddly to really get the most out of the game. There are better suited controllers out there, including a clone of the GameCube controller that diehards absolutely swear by. It's still a pain

having to shell out for a new controller just for one game, though.

Now that the nitpicks are out of the way, let's talk about just how joyous this game is. Everywhere you look there's something to put a smile on your face. Ryu facing off Captain Falcon in Green Hill Zone. Mario dodging a spurt of acid on Brinstar while Mega Man charges up his laser.

For anyone new to the series, *Smash Bros.* flips the rules of the standard fighting game a little: instead of beating each other up until there's only one standing, the aim is to bash your opponents to the point where they can be knocked off the stage. Each opponent you despatch earns you a point, while the more damage you take, the more vulnerable you are. *Smash Bros'* defining characteristic has long been its level of chaos – especially when you're

REVIEWED BY
Jake Laverde

HIGHLIGHT

If you're of a certain age, then the number of character cameos and stages based on classic games will surely bring a smile to your face. There's so much here that only the truly dedicated will unlock everything.

Using items against your opponents can be super-effective.

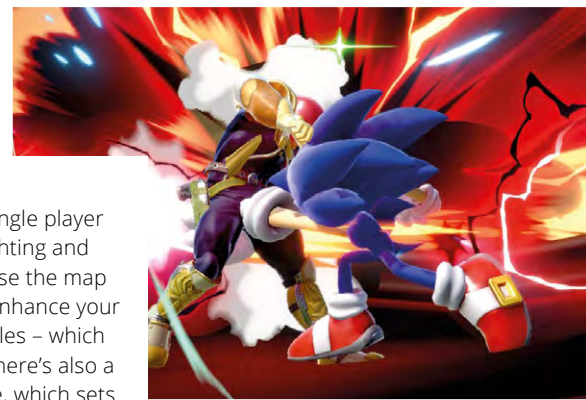


It's not just smashing – there's rolling, too.





▲ Live out all your nineties Nintendo vs Sega playground fantasies.



▲ Low blow Sonic. Low blow.

fighting seven other players at once. A typical match is like watching a firework display: fighters zip around the screen on a stage that's constantly shifting, with attacks and bonus items flying everywhere, so you can be forgiven for falling to your doom in your first few matches.

Smash Bros. Ultimate sticks to that basic formula. Its main draw, at least for long-time players, is that it contains every character and stage from all previous instalments, along with 11 new fighters, for a combined total of 74 characters at launch. And that's not including forthcoming DLC characters, the first of which is Joker from *Persona 5*. With this amount of content available, you might expect to see the odd corner cut here and there, but what's impressive about *Ultimate* is the level of craft on display. Each stage is a wonderful recreation of its respective game: karts race by on the Super Mario Kart stage; players take part in familiar mini games on the WarioWare stage. Every screen has something new and surprising to offer.

“Ultimate still remains true to its accessible core”

On top of all this, there's a new single player mode, World of Light, that mixes fighting and *Pokémon*-style collecting: you traverse the map collecting Spirits – characters that enhance your chosen fighter's abilities during battles – which in turn unlocks yet more content. There's also a new multiplayer mode, Squad Strike, which sets two teams of three or five against each other.

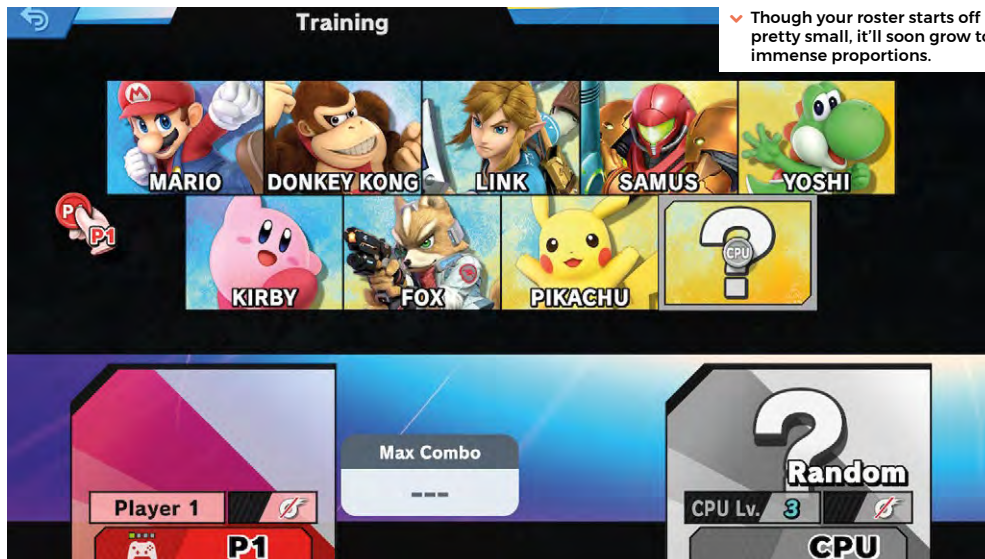
What has always set *Super Smash Bros.* apart from other fighting games is its playful nature – the average game is akin to watching kids mashing their favourite action figures together.

And while the series' brand of chaos isn't necessarily for everyone, it's a fun and accessible brawler where, not unlike *Mario Kart*, even beginners can sometimes win by grabbing a random item just at the right moment. Despite all the characters and modes to choose from, it's pleasing to see that *Ultimate* still remains true to that accessible core.

Playing *Smash Bros. Ultimate* is like sinking into a vat of nostalgia, such is the range of games featured here; as well as Nintendo's roster of characters, you'll find early arcade icons like *Pac-Man* and more recent indie heroes like *Shovel Knight*. That *Ultimate* not only brings all these eras of games together, but does so in the context of a brawler as exhilarating as this, is quite an achievement. 🎮



▲ Each character has their own unique fighting style.



VERDICT

No innovations here, but an absolutely dizzying array of options makes this the ultimate package.

81 %



Review

Ashen

Info

GENRE
Action RPG

FORMAT
XBO (tested)
/ PC

DEVELOPER
Aurora44

PUBLISHER
Annapurna
Interactive

PRICE
£33.49 (Xbox),
£35.99
(Epic Store)

RELEASE
Out now

A timely reminder that it's dangerous to go alone

Like other imitators that have attempted to rise to the occasion, it would be easy to pick apart *Ashen* with regards to its similarities to *Dark Souls*, from the stamina management to the near-identical controller button-mapping, but New Zealand-based studio Aurora44 has also created something refreshing in its own right.

In a sunless world that's been shrouded in darkness for eons, you're on a quest to find and protect the titular phoenix-like creature that has recently awoken to bring back light. It's a journey you never undertake alone – indeed, a moustached, pipe-smoking chap called Jokell is with you from the start, helping you defeat a band of hostile stragglers to establish your hub, Vagrant's Rest. Before long, you continue encountering more allies to rally behind your cause, from a mysterious seer to a giantess. More interestingly, they may in fact be other players.

Taking inspiration from *Journey's* multiplayer, *Ashen's* seamless and anonymous matchmaking is a fascinating idea that creates an organic and unpredictable dynamic between you and a stranger – the trick being that, for the other player, it's you who is role-playing as their companion while they are in control of their own avatar. That any stranger could inhabit these

roles suddenly explains the characters' lack of a face.

As unique as this might sound, the concept can also be easily nixed in the menu, where you can opt to use a password filter to ensure you're matched only with a friend, or just stick with AI. The latter became useful when I just couldn't deal with unreliable players during a difficult point or wanted to selfishly run on ahead – issues you don't have to worry about with compliant AI, who also has a knack for warping over to you if they need to catch up.

Things get stricter once inside the game's dungeons, where you won't get matched with another player should you or your companion meet your demise. These are also the darkest environments in the entire game, requiring at least one person to sacrifice their shield arm to carry a lantern, as well as containing the most terrifying enemies, like the wraiths that rush for you then pin you down until your partner can fight them off. With no shortcuts, and only one checkpoint located before the boss room, they're perhaps the most nightmarish gauntlets I've run since *The Tomb of the Giants* in *Dark Souls*.

Fortunately, there's only several of these dungeons to contend with, making them more like rare challenges to relish instead of an overwhelmingly oppressive feature.

REVIEWED BY
Alan Wen



▲ Much of *Ashen's* lore and its characters were written by American-British fantasy writer Mark Lawrence.



^ The bosses tend to have a huge health pool, making it doubly important for both players to get hits in.

Leaving these aside, *Ashen* is on the whole a more accessible Soulslike. Naturally, having multiplayer by default alleviates the combat, but you're also free to equip any gear you happen upon, as nothing is restricted by character class or attributes. This equally applies to your companions, so it won't feel jarring that the huntress Vorsa is accompanying you with a spear in one quest then swinging a two-handed hammer the next.

You'll still be spending a lot of Scoria, the game's equivalent of Souls (acquired and lost in the same way as you'd expect) but these are focused on upgrading your tools, from your weapons' damage output to the potency of your Crimson Gourd (basically, this game's version of an Estus Flask).

The ultra-hardcore player needn't balk at these accommodations as they're similarly catered for. It's possible for you to fashion a talisman that lets you go through the game's dungeons alone, while about a third of the way in, you'll unlock a harder mode that you can start right away called 'Children of Sissna'.

Regardless of difficulty, the basics of *Ashen's* combat has much of the satisfying feedback as any veteran FromSoftware fan can hope for.

"Ashen's anonymous matchmaking is a fascinating idea"

What's more disappointing is that it doesn't leave as much mystery. While the wild open landscapes and ruins of forgotten civilisations can look wondrous from higher ground, on closer inspection it's also essentially a linear path you're journeying on (or rather a serpentine

path, conveniently surrounded by mountains), where quests are marked on your map, and without any eureka moments of

discovering an illusory wall or shortcut. And for all the unpredictable ways a player-controlled NPC might behave with you in the field, when it comes to speaking to them back at Vagrant's Rest, I found their delivery often slow and puffed, while the dialogue gets overly expository, lacking in ethereal ambiguity, mystery and humour – a far cry from *Dark Souls* lore.

Yet what *Ashen* has going for it most is that despite the hostile forces out to get you, it's also a very inviting place to spend time in. There are scenes and architecture that recall FromSoftware's masterpieces for sure, but when you get to take in the sweeping landscapes, it also conjures up the sparse natural beauty of *Breath of the Wild*. This world may have languished in darkness, but instead of a festering decay, it's undergoing renewal, where you and your fellow band of wanderers are fighting for a place to belong and to safeguard a hopeful future, instead of merely surviving. Ironically, for an adventure where no-one has a face, *Ashen* is a Soulslike that feels like it has its own identity. 🗨️

HIGHLIGHT

There's a genuine pleasure to routinely returning to Vagrant's Rest as it develops over time. Besides new characters setting up a useful workstation where you can craft talismans and potions, they're also building homes, making the place feel alive with hope. It's probably the most peaceful video game hub since Majula in *Dark Souls II*.

VERDICT

Ashen is a competent Soulslike elevated by a sense of optimism and community amidst the darkness.

79%

✓ Just walking up to a ritual stone replenishes your health and Crimson Gourd. It's only when you sit down that it also resets enemies.



Rated
Review

HIGHLIGHT

Each of the 12 controllable titans in *Override* features its own arena-decimating super-move. Unfortunately, without modifiers, it can only be triggered when your mech is down to 20 per cent health, making it unlikely you'll unleash it the more adept at piloting you become.

Review

Override: Mech City Brawl

Cancel the apocalypse... with a little help from your friends

Info

GENRE
Fighting

FORMAT
PS4 (tested) /
PC / XBO

DEVELOPER
The Balance
Inc

PUBLISHER
Modus Games

PRICE
£24.99

RELEASE
Out Now

REVIEWED BY
Aaron Potter

VERDICT

Override: Mech City Brawl is a great party fighter to break out for social gatherings, if little more.

67%

Rocket-powered punches. Sky-high elbow drops. 30ft katana swings. The list of mech manoeuvres *Override: Mech City Brawl* allows you to pull off is seemingly endless. As far as party fighting games go, this really is the stuff of kids' Saturday morning TV shows – particularly *Power Rangers* and *Voltron* – brought to life, complete with an over-the-top sense of campness, skyscraper-sized scale and a big heap of scrappy charm that comes with a slight lack of polish. Even if *Override* isn't the most in-depth or competitive brawler out there, hitting each other as big colourful robots is surprisingly satisfying.

It can't be understated just how much *Override: Mech City Brawl* begs to be played with friends. This isn't only due to the additional challenge that comes from taking on a real-life opponent, as you'd expect, but also due to the game's slight riff on co-op multiplayer, which sees you and up to three others pilot one mech in true *Pacific Rim*-esque fashion. This feeds into *Override*'s already-unique control scheme, where right and left triggers control your mech's conjunctive leg or arm. Both are cool gimmicks that successfully feed into the Kaiju vs Jaeger fantasy, ripe for causing hilarity whenever you invite a few mates over.

A lot of the fun found in *Override* comes from experimenting with the titans themselves. Ranging from everything including an Optimus Prime homage in Watchbot to the suspiciously Dragonzord-looking Metageckon, all 12 don't do much to hide their pop culture roots. But it's more than made up for with a suite of special movesets that sees all move and behave differently.

There's enough variety in each to keep you entertained for a few hours, and developer The Balance Inc has already started to add new mechs.

Sadly, it's when *Override: Mech City Brawl* is treated primarily as a solo experience that the AllSpark shines less brightly. There's a single-player campaign offering – of sorts – by way of the available arcade mode, but bar a few well-polished cutscenes, the story at hand plays it too faithful to its anime inspirations to keep things interesting. Whereas you're always fighting other mechs in *Override*'s Versus, the campaign's version of challenge seems to be to just have you clear out a number of grizzly (and admittedly cool-looking) beasts within a given time limit. It doesn't outstay its welcome, but unlocking new skins and mech modifiers only keeps you engaged for so long.

Override: Mech City Brawl won't become the next eSports fixture in fighting games – the lack of grapples and strategy required helps to solidify that somewhat. However, those looking for a schlocky brawler that pays homage to their favourite B-movie tropes of yesteryear will likely have fun stepping into the titan cockpit again and again. 🗣️

▼ *Override* sees you battle across various international locations from Morocco to the USA – even the Moon at one point.





HIGHLIGHT

Desert Child's soundtrack is exceptional, from the slow, funky hip-hop that accompanies your walks around town, to the more upbeat futuristic themes tailored to accompany each bike event. It all harmonises perfectly with the visual styling to convey the main character's personality and boost the overall experience.

Review

Desert Child

▲ Simple, pixelated backgrounds and characters are reminiscent of early Amiga and PC games, especially the old Sierra adventures.

Keep calm and race hoverbikes on Mars

Desert Child is a game about hoverbike racing. But, as you quickly learn from taking control of its anonymous protagonist, it equally wants you just to wander through its world and soak in the laid-back vibe.

The game's aesthetic expertly creates an atmosphere of detachment and indifference, from its washed-out colours and faceless NPCs, to the messy layouts of its streets and buildings. There's a muted, downtrodden feel to its Mars colony setting, with an underlying sense of corruption and poverty. Yet our nameless character seems unfazed as he tries to scrape by, always with a mellow hip-hop beat kicking along in the background.

The action alternates between casual strolls through the game's dozen or so locations and hoverbike events. Each day, you're free to wander and visit a range of places to repair and modify your ride, purchase (or steal) upgrades and eat different foods, before jumping onto your bike to earn funds. Once the event is complete, a new day starts, and the cycle repeats. There's an overall aim, but no rush to get there. Relax and take your time.

The bike events themselves range from straight races against single opponents to jobs such as pizza delivery, bounty hunts and kangaroo herding, or criminal dealings such as hacking bank systems or deliberately throwing a race. In practice, every event follows the same formula of driving from left to right for about a minute, then repurposes the basic move set –

shooting, boosting, dodging and collecting – to define its contextual objectives. Racing remains the most engaging task, as you manage your recharging boost bar, time reloads, shoot targets to gather speed and jockey for position.

While this combination of racing and ultra-light role-playing is certainly original and well-made, it's all a little too sparse and never blossoms into anything truly involving. The way the game allows you to endlessly walk its locations and choose any activity at almost any point accords with its relaxed attitude, but makes the experience loose and inconsequential – there's never much

at stake, and no real need to plan ahead.

Similarly, on the bike, all the activities are fun, but ultimately not substantial,

varied or challenging enough. In races, different environments and opponents provide little more than cosmetic variation, while the diverse objectives in other modes are actually very similar in execution. Even bike customisation, which introduces a neat puzzle element based on arranging modifications and power cells in a limited space, lacks sufficient depth.

Occasionally, an opponent makes you work for victory and it's tempting to wonder if the bike racing could have really come alive in a different structure. But keeping everything in tune with the easy-going feel is more important for *Desert Child*. It deserves respect for adhering to its stylistic vision, even if the result is merely a pleasant way to pass a few hours, rather than something truly compelling. 🗿

“There's an overall aim, but no rush to get there”

Info

GENRE

Racing

FORMAT

PS4 (tested) / PC / Mac / Linux / XBO / Switch

DEVELOPER

Oscar Brittain

PUBLISHER

Akupara Games

PRICE

£8.99

RELEASE

Out now

REVIEWED BY

Jon Bailes

VERDICT

Original and stylish, but too chilled to make the most of its content.

59%

HIGHLIGHT

Iris.Fall's cel-shaded aesthetic is an intriguing mix, mashing up anime and gothic illustration in the style of Tim Burton and Edward Gorey. When most games either want to dazzle you with photorealistic vistas or tickle your nostalgia zone with pixels, *Iris.Fall* is a breath of fresh air for the eyes.

Review

Iris.Fall

⚡ Ever get the feeling you're being watched?

Info

GENRE
Puzzle
FORMAT
PC
DEVELOPER
NEXT Studios
PUBLISHER
NEXT Studios
PRICE
£11.39
RELEASE
Out now

REVIEWED BY
Jake Laverde

VERDICT

An evocative trip through a series of wonderfully intuitive puzzles that doesn't outstay its welcome.

72%

A

t first, *Iris.Fall* looks like another in a long line of quirky indie titles with a mute young protagonist and... well, to be blunt that's exactly what it is. There's not a lot here that

hasn't been seen before. The monochrome visuals recall Playdead's *Limbo*, and it shares a similar doom-filled atmosphere expressed through an intentionally vague narrative.

But where in *Limbo* every step could be your last, *Iris.Fall* is more a modern take on surreal classic, *Alice in Wonderland*.

Taking control of Iris, you follow a black cat through a Victoriana-esque world filled with puzzles. One of the main mechanics here is Iris' ability to turn into a shadow via a book, and manipulating objects and light to get to places you can't reach in physical form. Again, it's something that's been done before, most notably in Compulsion Games' *Contrast*. (I must stress that this isn't meant as a criticism, though; after all, no one rubbishes *Super Mario* games for repeatedly using jump mechanics.)

It's rare and refreshing to see a game use simplicity so effectively, and *Iris.Fall* bucks the trend of filling the screen with a tangle of systems. There's a sharp focus here, fully exploring just a few mechanics rather than fobbing you off with busywork, like too many open-world games. If there's one complaint, it's that there's nothing here that *Iris.Fall* can really claim as its own.

So while *Iris.Fall* doesn't push any envelopes, it's all done with such confidence and charm that

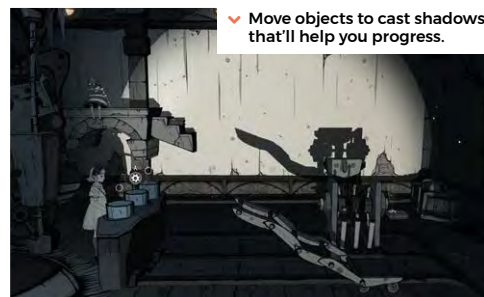
you can't help but be compelled to play it through. It's pleasing to see its story revealed through both the mechanics and hand-drawn animation, with themes of light and shade a constant in its two-to-three hour playing time. It isn't going to challenge the likes of *Red Dead Redemption 2* for longevity, but it's utterly stuffed with ingenious design touches. There's a huge range of variety in its challenges, from simple lock-and-key puzzles to more complex conundrums involving Rubik-esque cubes and light beams, all of which are instantly

intuitive – there are no tutorials to get in the way here.

In a way, it's a shame that *Iris.Fall* is only currently available on PC, because it's a wonderfully accessible puzzle adventure. Its simple controls

could easily be translated to a touchscreen, and its pace would be ideal for kicking back on a couch. Hopefully it'll get ported to more platforms in time, but for now, this is a perfect title for anyone wanting a bite-sized yet satisfying experience. 🗣

“It's rare and refreshing to see a game use simplicity so effectively”



HIGHLIGHT

While there are other teammates, Dux and Boarmin are easily the stars of the show. Boarmin is a hulking, intelligent veteran, while Dux is a sardonic, gritty Howard the Duck that somehow manages to be amusing without becoming the annoyance you'd think a talking duck would become.

▼ Cover will keep you alive, but not for long.

▲ Kicking back at the Ark lets you upgrade and buy new gear.

Review

Mutant Year Zero: Road to Eden

Think of it as Mad Max with pigs and ducks

Mutant Year Zero starts off in such a promising way. Two mutants, a duck and a boar walk home through the wastelands of post-apocalyptic Sweden.

Following the 'red plague', Sweden has been left an inhospitable hellhole for everyone but the most deranged and mutated. Sent on a rescue mission, Dux and Boarmin (your duck and boar, respectively) uncover that a missing genius has possibly discovered the apocalypse's version of heaven: Eden. On the surface, *Mutant Year Zero* looks like it has all its human-sized ducks in a row, but then some deeply ingrained problems break through the cracks and drag the experience down to a frustration.

There's a lot of love poured into this world, and the game's biggest strokes of genius come when it feels able to run wild with the setting. Much of the humour comes from the characters trying to understand our world through the objects left behind: boomboxes are literal bombs, cringeworthy 'kiss me quick' hats are signs of fertility, and the conspiracy theorists wrapping tinfoil around their heads must have been revered for their intelligence.

It's such a shame that moving through this world is such a pain. Much in the same style as recent XCOM games, positioning, using abilities and taking cover are all vital to survive the turn-based combat,

but even then, that's rarely enough. The game is maddeningly hard, even at its lowest difficulty setting. Fights are based on random chance, enemies usually outnumber your team two or three to one (even if you manage to ambush a few), tough med-bots revive them straight away, and reinforcements often flood in to make things worse. Surviving these encounters feels amazing in a way few games pull off, but all too often it's a matter of reloading a save and trying that lengthy fight all over again.

There's also a stealth angle, where creeping around enemies or taking them out from the shadows is framed as a viable option. During combat, those systems come together to some extent; if you remain quiet and out of sight of other enemies, you can easily get the drop of stragglers. Outside of combat, stealth is ludicrously basic, amounting to nothing more than avoiding circles of awareness that move around on the floor. All thought for cover and sound vanish, leaving behind the husk of what could've been an intriguing system.

The Bearded Ladies have built the foundations for a fantastic series. The Nordic-flavoured apocalypse and its colourful cast of characters deftly walk the line between dark and silly, offering a less suffocatingly bleak world than other Roadside Picnic-inspired games like *S.T.A.L.K.E.R.* The game itself lets it down big time, though, and requires more patience than most people probably have. 🐷



Info

GENRE
RPG / Strategy
/ Stealth

FORMAT
PC (tested) / PS4
/ XBO

DEVELOPER
The Bearded Ladies

PUBLISHER
Funcom

PRICE
£29.99

RELEASE
Out now

REVIEWED BY
Joe Parlock

VERDICT

Gives us a great setting and memorable characters, but slightly falls apart when it comes to action.

54%

Rated
Review



> Tapping buttons on the yellow sections of Alex's records increases the attack combo. Annoyingly, fumbling the inputs can result in a miss.



^ Fortunately, you meet more interesting characters than Alex. You meet Vella at the local arcade, and she's got quite the backstory.

Review

HIGHLIGHT

Given your protagonist's fondness for records, *YIIK* features a hip, eclectic mix of music, from retro-tinged chiptune and dreamy electronica to rock and disco, instead of one designated battle theme. Whether the game will establish an underground following remains to be seen, but it feels apt that *Undertale* creator Toby Fox also contributes a track.

YIIK: A Post-Modern RPG

Battling with a party like it's 1999

Info

GENRE
RPG
FORMAT
SWITCH (tested)
/ PS4 / PC
DEVELOPER
Ackk Studios
PUBLISHER
Ysbryd Games
PRICE
£15.99
RELEASE
Out now

REVIEWED BY
Alan Wen

VERDICT

While lacking in looks and execution, *YIIK* may be weird enough to garner cult status.

68%

Bearded college grad Alex Eggleston isn't your typical charismatic hero – he's a hipster, for one thing – but things soon take a strange turn when he becomes obsessed by the disappearance of local girl Semi 'Sammy' Pak. Recruiting like-minded misfits, you try to figure out the meaning behind Semi's disappearance, going down weird rabbit-holes, and fighting a bizarre gallery of enemies, like samurai tortoises, alien spaceships and sentient stop signs.

Taking place in 1999, *YIIK* (pronounced 'Y2K') recalls the small-town surrealism of *Earthbound* (even its signature enemies have a resemblance to the Starman) and the parallel dimensions of *Persona*. It's ostensibly a JRPG made by Americans, albeit with a cruder style and on a smaller budget – though fans of Adult Swim shows will feel right at home. Garish palette aside, though, *YIIK* has plenty of its own ideas.

Indeed, there's plenty of mechanics here that make it more than just another dungeon crawl and random encounter fest. Not only is there quite a bit of puzzle-solving and exploration required, including two deviously designed dungeons that take place inside the minds of two key characters, but even the turn-based battles have a lot of variation. Whether you're attacking, defending, fleeing or levelling up, everything functions as a mini-game, including QTEs,

rhythm-style button tapping, even a platformer paying homage to *Super Mario 2*. The best ones are for each character's attack and abilities, tied to their weapon of choice, such as vinyl records, a camera, or protest placards for one character.

Disappointingly, most characters rarely deal much damage, even with new gear, making even the most common encounters outstay their welcome once you realise every battle really is a series of mini-games. It soon transpires that Alex has the most powerful ability, so it's often a case of just spamming the same move while the rest of the party heals or tops up a PP meter with energy drinks. It's quite something when I suddenly appreciate the simplistic battles of *Dragon Quest*.

Taking only 25 hours to finish, *YIIK* is a shorter RPG than most, yet somehow still manages to drag in places, not helped by loading times before and after battles, and also Alex's tendency to indulge in ponderous monologues. Of course, when the discussions dabble in the surreal or metaphysical, *YIIK* can get pretty mind-bending, or perhaps just plain pretentious – something that Alex half-winks at in acknowledgement.

YIIK wears its love for JRPGs on its sleeve, but by doing something aesthetically and mechanically different, and going to some truly weird places, it manages to be more than just a cheap tribute act – even if it still does look more than a little cheap. 🎧



March of the picky penguins

The parrot definitely needs to go in front of the elephant

Sometimes, games designed for a solo player work even better when there's a second person helping you out. Take *Marching Order*, an adorably simple puzzle game available now for Android and iOS; you can happily play it by yourself, but in our experience, it's even more fun when you and a friend huddle around the same smartphone, trying to figure out the next move to make.

Marching Order dresses up some head-scratching logical problems with a parade of cartoon animals. There's a chameleon waving a flag, a fox playing a tuba, a lion nonchalantly bashing together a pair of cymbals, and out in front, a sweatily anxious rabbit leading the band.

Brilliantly, you're cast as the harried rabbit, whose job is to figure out what order the animals behind it need to march in. The animals are a picky bunch, and their moods change from day to day; sometimes, the parrot waving the banner prefers to hang out at the back of the parade; the fox might insist that it plays better when it's near an animal playing a drum; the chameleon sometimes decides that it doesn't like playing in front of creatures with beaks.

To figure out how to make all these animals play nicely together, you have to scroll through all the clues – "I should ask to be at the back", "I really want to be somewhere behind the duck" – and then drag and drop the band members into what you think is the best order. As you'd expect, the puzzles gradually increase in challenge, and there are bonuses to be had for setting the correct order of animals in as few moves as possible. It's all made doubly delightful by the presentation:

the hand-drawn animals march proudly with their respective beaks and snouts pointed to the sky while a jaunty (and infuriatingly catchy) tune pipes in the background. The way the clues are presented is another minor delight: the idea is that the rabbit is scrolling through the animals' various social media accounts in a vain effort to figure out what mood the little blighters are in.

Marching Order is the latest game from Tom and Adam Vian, whose studio SFB Games is perhaps best known for *Snipperclips* – one of the best puzzle games on the Nintendo Switch. According to Tom Vian's Twitter feed, *Marching Order* began life as a tool that could automatically generate logical puzzles; it then dawned on them that what they'd created was the basis for a new game in its own right.

We're so glad they did. *Marching Order* is one of those miniature experiences that feels perfect for mobile – and again, it really is strangely compelling when played with a friend. An extra brain comes in handy when the puzzles get really tricky, and the resulting conversations – "No, the parrot wants to be near a creature with fangs!" – are particularly fun to have on a crowded bus. 🐼

"I really want to be somewhere behind the duck"



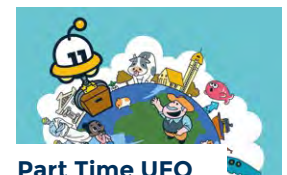
Wireframe Recommends



Mr Driller Drill Spirits

NINTENDO DS

Versions have appeared on iOS in the past, but this DS edition is the best handheld version for us. A joyous action puzzler, *Mr Driller* is well overdue a revival.



Part Time UFO

ANDROID / IOS

It's both an absorbing puzzler and (possibly) a surreal comment on the gig economy. HAL Egg's *Part Time UFO* is still one of our favourite mobile games of recent years.



Echochrome

PSP / PS3

It's over a decade old now, but Sony's minimalist puzzle game – with Hideki Sakamoto's elegant string quartet soundtrack – has barely aged a day.



The Legend of Zelda: Breath of the Wild

Exploring the vast Kingdom of Hyrule is hungry work: fortunately, Link's cooking skills are backed by a catchy tune

NINTENDO EPD / 2017 / SWITCH, WII U

It's not the overall experience we want to talk about today – *Breath of the Wild* is a phenomenal achievement, of that there's no doubt. It's not even the cooking in general we want to focus on, providing, as it does, boosts and boons across the board in a series of microsystems that are both simple and intuitive. No – Link's Switch outing bags the gong thanks to, of all things, the *sound* of its cooking, and we're willing to bet anyone who's played the game and cooked enough will be nodding along at the mere thought of it.

Tossing a few fruits or hunks of meat into the pan, Link quickly sautés his way through a skippable cutscene. It's seconds long, the kind of thing you're usually hammering the skip button to rush through (or holding it for an interminable amount of time, in the case of *Red Dead Redemption 2*). This stuff doesn't matter once you've seen it two or three times – it adds nothing and just contributes to time wasted *not making progress*. Usually. But not here.

No, Link's culinary pursuits are accompanied by a clackety-clack of rhythmic cooking sounds; the ingredients tossing themselves around the pan and rather magically forming into something far more aesthetically pleasing than it should be, considering the effort that goes into it. It's a tick-tock countdown before you can chow (or gulp) down, a boogie-

worthy tune limited to around five seconds, an eminently satisfying slice of SFX setting you up to receive another useful in-game item.

Of course, it's not always useful: you can make inedible, useless muck. And that's where another subtle little factor comes in – the sound cue at the end of the brief cooking scene gives you a little blast of triumph or a... well, non-triumphant pong. We're not going to claim this fundamentally

changes the game in the least; more that it shows how a brief, usually-forgettable sound cue was engineered to prove gratifying, to not outstay its welcome and – to an extent – be functional. It lifted

“Link's culinary pursuits are accompanied by a clackety-clack of rhythmic cooking sounds”

something as seemingly insignificant as a sound effect you would usually ignore, skip, or – at worst – get annoyed with to something some of us would sing along with. And we know that wasn't just us.

For all its features, killer or otherwise, *The Legend of Zelda: Breath of the Wild* maintains one constant: an extraordinary level of quality. It's easy to pull out those larger, more obvious elements and – rightly – celebrate them. But when you break it down, chip away at the surface, and see that even factors like what you hear when cooking an elixir or meal have been crafted to make something *feel* satisfying, it's a whole other level of class. 🍷

Next Issue

ON SALE 30 JAN

VOID BASTARDS

Don't
miss out!
Subscribe
PG54

Also

- ▶ The hellish creation of Pinstripe
- ▶ Behind the scenes on the gorgeous Sable
- ▶ The dark art of localisation
- ▶ GameSalad: inside the codeless game-making tool

Editorial

Editor

Ryan Lambie

Email ryan.lambie@raspberrypi.org

Features Editor

Ian Dransfield

Email ian.dransfield@raspberrypi.org

Sub Editors

David Higgs and Vel Ilic

Design

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Harriet Knight

Contributors

Jon Bailes, Lottie Bevan, Rik Cross, Konstantinos Dimopoulos, Gareth Dutton, Stuart Fraser, Jake Laverde, Joe Parlock, Aaron Potter, Howard Scott Warshaw, Alan Wen, Jane Whittaker

Publishing

Publishing Director

Russell Barnes

Email russell@raspberrypi.org

Tel +44 (0)7904 766523

Director of Communications

Liz Upton

CEO

Eben Upton

Distribution

Seymour Distribution Ltd

2 East Poultry Ave, London EC1A 9PT

Tel +44 (0)207 429 4000

Subscriptions

Raspberry Pi Press, Mann Enterprises, Unit E, Brocks Business Centre, Haverhill, CB9 8QP

To subscribe

wfmag.cc/subscribe

To get help:

rpiresshelp@raspberrypi.org



This magazine is printed on paper sourced from sustainable forests and the printer operates an environmental management system which has been assessed as conforming to ISO 14001.

Wireframe magazine is published by Raspberry Pi (Trading) Ltd., 30 Station Road, Cambridge, CB1 2JH. The publisher, editor and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised in the magazine. Except where otherwise



noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0).
ISSN: 2631-6722 (print), 2631-6730 (online).



wfmag.cc