

ALL FORMATS
116 PAGES MONTHLY

Wireframe

LIFTING THE LID ON VIDEO GAMES

CREATIVE MODE

Make a block-building
sandbox game in Unity

MEGABLAST

Follow our guide to make
a bullet hell shoot-'em-up

FATAL FRAME

The allure of taking
photos in games

SPEED DEMON

INSIDE DEMON TURF: AN INDIE
PLATFORMER WITH ATTITUDE

Issue 53 £6 | wfmag.cc





G-MASTER™

MONITORS 4 GAMERS

by **iiyama**

UNLEASH YOUR FULL GAMING POTENTIAL



AMD
FreeSync
Premium

Get the competitive edge you need to unleash your full gaming potential with the 24" and 27" G-Masters offering 0.8ms MPRT and 165Hz refresh rate. Armed with FreeSync Premium you can make split second decisions and forget about ghosting effects or smearing issues. The ability to adjust brightness and the dark shades with the Black Tuner delivers greater viewing performance in shadowed areas and the IPS panel technology guarantees superb image quality.



Fixed stand versions:
24" G2470HSU-B1 & 27" G2770HSU-B1
Version with height adjustment:
24" GB2470HSU-B1 & 27" GB2770HSU-B1



Find your match at
gmaster.iiyama.com

3D platformers ruined everything, but that's OK

In *The Salmon of Doubt*, Douglas Adams outlined rules describing people's reactions to technology. You've probably read them. In short, he suggested what's in the world when you arrive is normal, and what's invented when you're relatively young is new and exciting. However, whatever rocks up when you're past it is against the natural order of things. That last one is how I felt about 3D platform games.

I know: old man yells at cloud. I'm aware this response was down to technology moving on without me, decades ago; but there was a tangible feeling during the mid-1990s that every game suddenly had to be in 3D, by law. It was like everyone's design brains stopped working. Gone was any debate regarding the suitability of presentation formats. The notion of whether a game would be better in 3D was superfluous. Gamers were hungry for 3D, we were told – and they had snazzy new hardware to prove it. Therefore, all games would now be in 3D! The end.

This felt prescriptive and wrong, eroding what I loved about the genre. Much of the appeal of platform games was down to their clockwork nature – an emphasis on placement, precision, and timing. Perhaps too much emphasis at times, as evidenced by my younger self's screams of anguish when Monty Mole yet again abruptly exploded across the screen after lightly brushing a patrolling foe. Or that deep resignation on realising Miner Willy would never get a good night's sleep. But there was a feeling of total control – and that these titles felt right in having a healthy disinterest in the third dimension.

As gaming hardware and creator capabilities evolved, games in this space became increasingly expansive and clever. *Jet Set Willy* and *Impossible Mission* gave way to the masterpiece in design that was



CRAIG GRANNELL

Craig Grannell has been writing about tech and games for more years than he cares to remember. He shares his time between black rectangles, too many games of *Polytopia*, and rediscovering one-button classics to play with the youngling. Tweet him: [@CraigGrannell](https://twitter.com/CraigGrannell)

Super Mario Bros. I gawped at the astonishing technical prowess of *Turrican II*, casually throwing screen-sized enemies about on an ageing 8-bit Commodore 64. I was exhilarated by the sheer speed of Sonic.

Then some idiot ruined everything, because platform games had to be in 3D. Suddenly, I had to grapple with depth and cameras. Everything felt woolly, like directing a jelly. Nothing clicked, whether I was playing a bear, a bandicoot, a newly 3D speeding hedgehog, or a moustachioed man somehow still claiming he's a plumber. (There has got to be some dodgy tax stuff going on with that.)

Having happily regressed into 'old man' mode, I was delighted when 2D platform games made a triumphant return on handhelds. All was right with the world again! But today, 3D platformers – having been out of vogue for a while – are on the rise, making a comeback of their own. Yet having thought about this for a while, that doesn't fill me with horror this time around.

Why? Largely because there's a feeling in gaming today that everything is now acceptable. Any shift towards 3D is not being demanded in response to technical evolution, and is instead driven by a combination of nostalgia and suitability regarding the games in question. Design brains aren't turned off this time, and people will carry on creating 2D platformers for old farts like me as well. Joy!

We're in a place where the best of this generation's 3D and 2D platform games can happily coexist – and folks like me can stop worrying that industry overlords will decree all future iterations of *Manic Miner* must only be in 3D. Still, it could have all been worse: a maniac recently made a Sonic POV gameplay video on Twitter ([wfmag.cc/sonic-pov](https://www.wfmag.cc/sonic-pov)) and it's genuinely terrifying. At least 1990s platform gaming only went 3D rather than full POV. 🐼

#53



Contents



Attract mode

Interface

06. Demon Turf

Fabraz and Playtonic Friends on their sassy 3D platformer

12. Gloomwood

This first-person sneak-'em-up is more than just a Thief clone

14. Sable

Shedworks talk about their open-world game's final stretch

18. GRIME

Come for the surreal monsters, stay for the brutal combat

20. News

Another wry glance at the month's video game happenings

24. Incoming

Future wars, dark fantasy, and a suspiciously familiar hedgehog

28. Life through a lens

Exploring the indie games with photography at their core

34. Kim Justice

Why can't all sports games be as fun as Sensible Soccer?

36. 3D platformers

A whistle-stop tour of the genre's rise, fall, and welcome return

70. Italy's early game studios

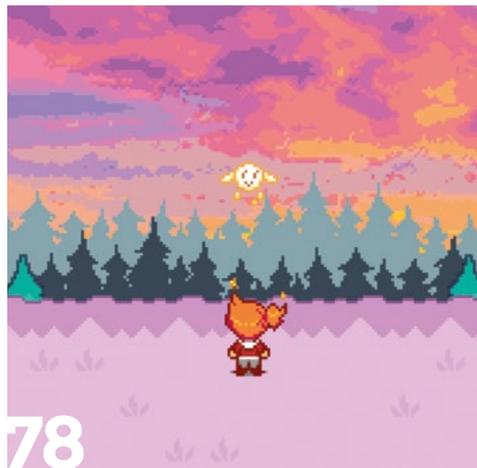
The software houses that blazed a trail in the 1980s and 1990s

78. Crystal Story

A solo developer's quest to make the ultimate 2D Zelda-like

90. Steve McNeil

Roguebook, and the right way to do downloadable content



WELCOME

As I write this, a copy of *Super Mario 64* has sold for more than \$1.5 million. Yes, a game that's about as rare as a Ford Fiesta just went under the hammer for roughly the sum of money it'd take to build a new ward at your local hospital. I bring this up because a fair chunk of this month's issue is given over to celebrating the 3D platformer: a genre that rose in the mid-1990s thanks to the added processing power of a new generation of consoles, and fell the following decade as interest waned. But the 3D collectathon has made a triumphant return in more recent years, thanks in part to heavy hitters like *Super Mario Odyssey*, but also indie titles like *Yooka-Laylee* and *A Hat in Time*.

Then there's this month's cover game, *Demon Turf*: a sprightly platformer that combines 2D characters and 3D environments to charming effect. What's more, the game's developer Fabian Rastorfer reckons this is only the start of an indie renaissance for the genre, and that there are plenty of other unique takes on the 3D platformer in the works that have yet to be publicly shown off.

Does the rising popularity of this style of game explain why *Super Mario 64* – admittedly, one of the genre's key titles – sold for the price of a house in Knightsbridge? No, there isn't a logical explanation for that. It's just more mind-boggling strangeness from the increasingly febrile world of game collecting.

Enjoy the new issue!

Ryan Lambie
Editor

Toolbox

- 46. Sim cities**
Tips for making your own systems-based metropolis
- 48. Creative Mode**
Our guide to making a Minecraft-style block-builder
- 58. Bullet hell**
Code your own shoot-'em-up, complete with area boss
- 64. Source Code**
How to recreate Nintendo's 1984 take on Pinball

Rated

- 92. Biomutant**
This strange, flawed action game isn't without its charm
- 95. Knockout City**
The sport that asks you to dodge, duck, dip, dive, and dodge
- 98. Slipways**
The superb space strategy game that's also a puzzler
- 104. Backwards compatible**
The Nintendo 64's birthday, GBA mods, and more

BORN TO RAISE HELL

Fabraz and Playtonic's Gavin Price tell us how *Demon Turf* is bringing back the 3D platformer in fiery style

When a bunch of former Rare devs vouch for an upcoming 3D platformer, it's worth taking notice. And that's essentially what's happened with *Demon Turf*, the upcoming game from the creators of 2017 indie hit, *Slime-san*; it's being championed by none other than Playtonic, the studio behind 3D platformer throwback *Yooka-Laylee* and founded by former Rare developers who once worked on such pivotal titles as *Banjo-Kazooie* and *Donkey Kong Country*.

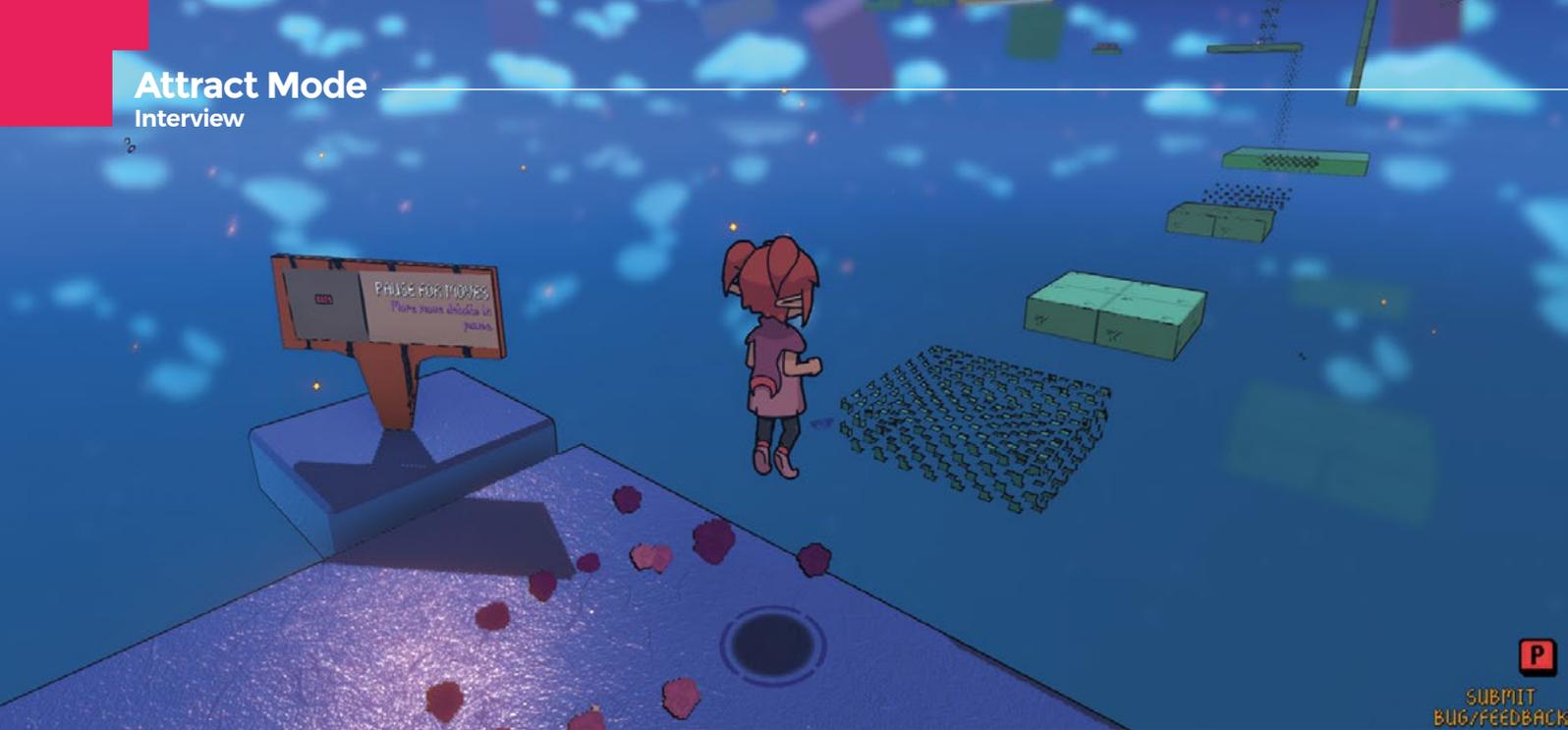
It's not hard to see why Playtonic Friends – the studio's new publishing arm – warmed to developer Fabraz's work-in-progress, either. Its use of 2D sprites in 3D environments give it a fresh look, while the traditional 3D action – solve puzzles, collect items, fight boss battles – is married to some fun, original ideas. Cast as a

plucky young demon named Beebz, whose mission in life is to seize the Demon King's throne, you're armed with an impressive array of moves to help you on your quest. Super-jumps, flutter-spins, and charge-punches can all be used to traverse chasms and batter enemies into submission, and if the going gets a little too tough, you can pay a gang of demonic revolutionaries to complete a level on your behalf. It's all part of a game designed to honour those 3D platformers of the past – not least Nintendo's breathtakingly polished output – while also taking the genre somewhere new and more accessible.

With all this in mind, we caught up with Fabraz founder and game designer Fabian Rastorfer, plus Playtonic co-founder and all-round design legend Gavin Price to talk about the making of *Demon Turf*, and the keys to crafting a classic 3D platformer. ➔







P
SUBMIT
BUG/FEEDBACK

WE ARE LEGION

Despite *Demon Turf*'s scale, compared to Fabraz's previous title, *Slime-san*, the game's still largely being developed by the same core team of three. "We're essentially three people working on this full-time," Rastorfer tells us. "But we work with a lot of part-timers or commission-based artists to help out. Playtonic is helping a ton, and you can essentially count them as team members. So it's been a dynamic growth. Beebz's sprite sheet was done by an artist we commissioned and worked with closely. All of Luci's and Midgi's were from a different artist. We work with three composers outside of our team... The credits end up looking really long, which is nice because I like making sure every single contribution, no matter how small, gets acknowledged. It's a beast of a project, and every little bit helps."

An obvious place to start: how did you hit on the mixture of 2D/3D aesthetics?

Fabian Rastorfer: There's a trend in the indie scene of getting more 2D visuals in 3D games, so I think we're riding the wave as it's started to become popular. But basically, we were prototyping a bunch of projects and one of them was a 2.5D platformer, so a camera that wouldn't be movable but would still have our 2D characters in a 3D space. We experimented with that and we really dug where it was going, but gameplay-wise, it felt too constrictive. It didn't have a thing that made it feel unique, especially after having worked with *Slime-san*, where the main thing was the morphing and all that. So we said, 'Let's unlock the camera and see what happens'.

We immediately realised it'd cause an avalanche of new problems to make 2D sprites work in a 3D world, especially with a free camera, but it also immediately and suddenly opened up the floodgates of inspiration. Eventually, we found a point where we're like, 'This is working; I don't

think there's a moment where this feels less precise than with a 3D model.' And we can give our characters so much charm with an illustrative approach. So that's how we ended up with Beebz and her nearly 1000 sprites.

Your engine's Unity. So does 2D and 3D mesh well in the engine, or do you have to do some customisation to get it to work?

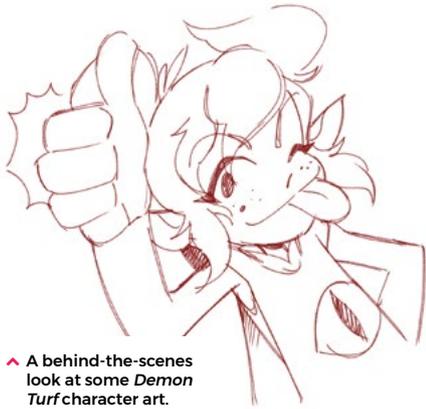
FR: Yeah, there was some custom stuff involved. A lot of it was making the 2D sprites work in 3D space with lighting, so that the sprite gets affected by being in shadow. And there were also a lot of things in terms of obstruction, because technically, the central character's essentially a cardboard cut-out, right? So if the sprite's too close to the camera, that's obstructing so much of your view, we have to [find] a way to figure out when it's too close to the camera. We had to come up with a whole system for our projector renderer that provides the drop shadow beneath the character, which is essential for the feeling of placement. So yeah, that was a lot of custom work.

At what point did Playtonic come in, Gavin, and what role have you been playing in development?

Gavin Price: We spotted the game quite a while back. We have our internal team channels, and we were just looking at things we find inspiring or like the look of. *Demon Turf* certainly caught our eye: when we saw this 2D character in a 3D platformer, we thought it looked really different. And I remember the one screenshot that really got us all falling in love with the character



> Once you've defeated a world's boss, you'll find that the levels have changed the next time you visit them, with different collectables to pick up.



^ A behind-the-scenes look at some *Demon Turf* character art.

was when she goes to take a selfie. That's gonna be so nice, taking your character around and taking selfies. She has so much character, and because she's a sprite, she can do so much.

Around that time, we were asking, what do we want to do as a company? How do we want to grow? We were really falling in with the idea of becoming a publisher and meeting other developers, which is a great part of the job. So we played a build of *Demon Turf* and it was ready to ship, pretty much. We thought, 'Do you even need a publisher?' We were asking them, 'What can we do for you that makes sense?'

From a gameplay perspective, Playtonic offered nothing – nothing – and got a free game to play in return! What we do have though, is experience in porting. Unity projects can be quite taxing in terms of console ports, so we've been able to help do some tidying up. We've just come in at the end and helped with the kinds of tasks that no team loves doing. We don't go creating our own IPs and think, 'I can't wait to start porting this.' It's not really high up there.

FR: As Gavin says, it's not fun to port and optimise, but you want to get it right. With our team size, the help was vital, because at this point, it's ludicrous. We're in between generations, so we've ended up with seven platforms to port to. It's just, like, 'What?'

GP: And there's physical releases as well.

3D platformers feel like they're coming back into vogue after kind of fading out for a while. I wonder if you had any thoughts on why they slipped out of fashion, and what brought them back.

FR: I think there are two things. Obviously, there are trends: there's always genres that are more popular for a time. Deck-building games and roguelike games are at the top right now. So that comes and goes. 3D

platformers had their height during the N64, PSone era, and I think now it's just rising back up again. The second thing is that it needed some time to re-evaluate itself as a genre. It's like how *Borderlands* said, 'What if we combine an FPS with an RPG?' And that kickstarted the whole thing of RPG-ifying almost every genre.

I think 3D platformers, too, are saying, 'Let's look at other genres and see what can make us unique.' I'm friends with a bunch of other indie devs – we have a little Discord server where we talk. And what's so awesome is that I can list five or six games that look amazing – they're all 3D platformers, but all entirely different. I think that's why the genre's becoming popular again, because it has the potential to be awesome – it's just really difficult to pull off.

Of all the genres I've worked on so far, 3D platformers are the hardest thing that I've had to work with. What I find fascinating about them is that moving is an act of expression for the player, so you have to create something that's approachable and easy to learn, that flows, but also has depth so people can start exploring and do wild skill combinations. I don't think there are many genres that do that, where expression comes through in the act of just simple movement.

Can we talk a bit about the game's story? It seems to me that these quirky, outsider protagonists are a hallmark of your studio. →



^ Each world (or Turf) has its own boss, who'll begrudgingly become your underling once you defeat them.

JUST THE TONIC

"When they reached out to me initially by email, I was double-checking because I was thinking, this is probably a scam," Rastorfer says when he recalls the day Playtonic first got in touch. "But I was immediately enamoured by the idea because, obviously, a good chunk of the team are giants when it comes to 2D and 3D platformers. For example, Jens [Restemeier], who's helping us with the PlayStation ports, worked on the *Donkey Kong* Game Boy ports, which I played as a kid. I was thinking, this is surreal that somehow little Fabian on his Game Boy was playing a game that he loved and now he's working with that person on the port of his game. It's just a crazy loop of fate. I grew up with all of it: *Banjo-Kazooie* was always at the top of my Nintendo 64 lists, and there's no doubt that all of those games had an impact, not only on *Demon Turf* specifically, but just my game design sensibilities and even my interest in becoming a game developer."



DEVILISHLY DIFFICULT?

Demon Turf will offer a number of mechanics for newer or younger players. A small army of helper characters, named Revolutionists, will be on hand to complete levels on your behalf in exchange for cold, hard cash; if you're finding boss battles tough, the same rabble will support you on the sidelines, throwing trash at your opponent. "I hate nothing more than a player enjoying your game and then not finishing it because they can't," Rastorfer says. "I think that's a failure as a developer, because it's just not fair – if you're enjoying it, you should be able to complete it. So we very seriously thought about all these potential solutions."



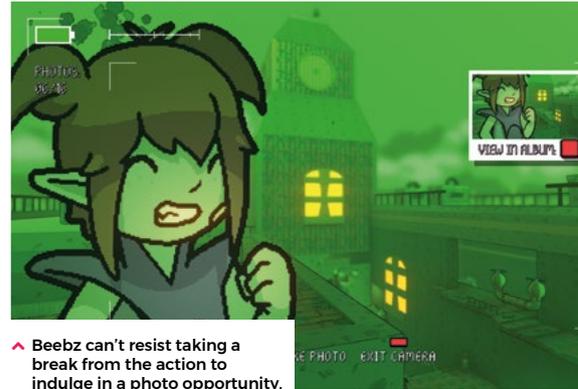
You've had a gooey blob, a skeleton, and now a demon.

FR: Our humour is super-weird, I think. Which is why we also mesh really well with Playtonic, because their humour's weird. But we just love going into the quirkiness; with *Slime-san*, we managed to create a simple character that still had a lot of personality. We looked at what people enjoyed about *Slime-san*, and it was the character and also the world. We're not necessarily grand storytellers – we're not going to be telling the next *The Last of Us* epic, but we are capable of creating a world the player wants to get lost in because of its humour, its charm, its quirkiness.

One thing about *Slime-san* that was popular was the hub town – it had a lot of goofball characters in there. And so we extended that with *Demon Turf*, with a hub town that's even bigger and that has even more goofy stuff to explore and find. And let's also inject more of a narrative without it being overbearing. Beebz has a bit of an attitude; she has a protective shell. But at her core she's a soft, friendly person who's really looking for recognition.

Is there a key to really good 3D platformer level design?

GP: I'm gonna agree with Fabian in terms of it being the most difficult genre to make a game in. You have all the problems, and then everyone points at everything Nintendo



▲ Beebz can't resist taking a break from the action to indulge in a photo opportunity.

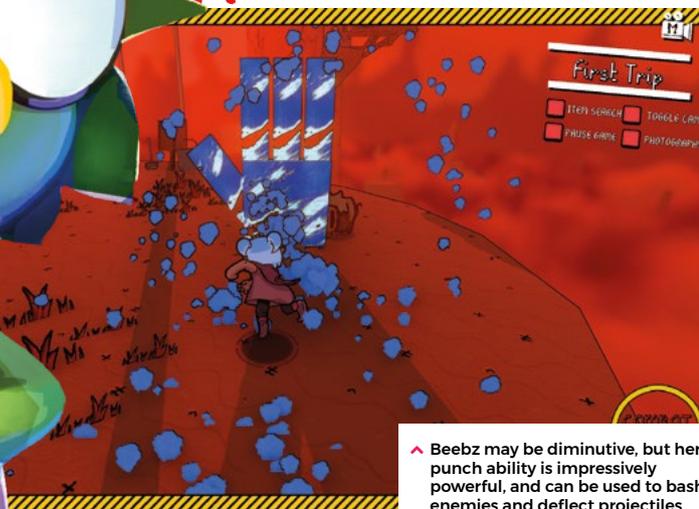
makes, saying, 'But look, it's really easy. Look, they're doing it time after time. Come on, you lazy devs!' [Laughs]

Fabian spotted something we also saw in *Super Mario Odyssey*. If you look at all the major routes through the game, and break them down, they're very well-hidden *Super Mario 3D World* challenges. *Super Mario 3D World* is fantastic at creating these A-to-B linear courses which test players' platforming skills. So, when you look at [*Odyssey*], they kind of disguise these courses in a complete, 360-degree roaming world space, which is brilliant. And the world is really there as a playground for your moveset, and it's very metrically driven. We had one of our lead designers look into *Super Mario World*, at how metrically driven they all are – they're so precise, even down to the textures applied to the buildings and paths, floors... You can see... 'Wow, even the textures are really easy to read'. You don't spot this stuff as you're playing, but you can see that there's one unit in the texture of a wall, so that when you stand next to it, you can see that the wall is four units high.

You're getting this information all the time as a player, not realising you're taking it all in. It's unbelievable. I think, in a 3D world, people are looking for that blend of something legible: they can see where they should be going and where they shouldn't bother trying to go.

The camera can make or break a game like this. How have you refined it?

FR: Yeah, it's a huge task. It was basically the second most important part after getting the platforming right. There are so many little tricks. So, for example, when the camera bumps against the wall, we don't want it to go through, so it'll start sliding across the wall



▲ Beebz may be diminutive, but her punch ability is impressively powerful, and can be used to bash enemies and deflect projectiles.



instead. But we don't want that to happen with thin objects, so if there's a telephone pole, the camera will bump against it and then go back into its original position. This creates a jarring effect, so what you do is build a system that either detects the width, or have a tagging system that tells the camera, 'ignore that pole'. So then the camera will be behind the pole, but you've got a new problem: your character might be obscured by the object, so then you have to build a system of figuring out how to show a character obscured by an object. So you create a silhouette that shows through it. And that's how it goes. You solve one

we then put them in our 'nice to have' pile, which then usually ends up becoming free updates or DLC. So most of *Slime-san's* DLCs were made up from these ideas we had during main development.

Were there ideas that you hadn't initially planned for that made it into *Demon Turf*?

FR: Yeah. We expanded the boss fights a lot. They're really intricate fights with dynamic music. One of the main things in the boss fights that I like is that, a lot of times in video games when you beat a boss, you unlock a new ability, right? In our case, you get that ability when you start the boss fight and then



▲ The game's 3D, which means characters need to be considered from every angle.

“Everyone points at Nintendo, saying, ‘Look, it’s really easy’”

problem, it introduces a couple of new ones, and you have to solve those until you get to a point where you don't think about the camera too much anymore.

Is overscoping a danger? Getting carried away trying to fit in too many ideas?

FR: The Fabraz philosophy is that we overscope a little bit, which sounds like an oxymoron, but that's essentially what we go for. If an idea comes up that wasn't part of the initial plan, but everybody in the team just absolutely adores it, then we consider it. How much more time will it add? How many more issues will it produce? And then we sometimes add them, but other times,

the fight is centred around it. You're learning how to use this new thing during the boss fight, master it, and then move on with it. Bosses have been expanded to a much larger importance than we'd initially planned.

Playtonic Friends is a new publishing arm for the business. Is it just going to keep growing, this side of Playtonic?

GP: Not really. We haven't entered into this to become a huge publisher. We're still a developer that can help people publish rather than a publisher which can help people develop. And we're not looking to do huge numbers of titles year in, year out. If anything, we might go for fewer games per

year. We want to help other people achieve more by working with us than otherwise they could. We're trying to be the publisher that we dreamt of coming across ourselves with regards to our terms, how we can help, how flexible we are. And in the background we've still got 90% of the company working on the games we've got in development.

So finally, how much have you got left to do on the game?

FR: Basically, the game's done. It's content complete. We're now in the trenches finishing all of the porting for consoles. And we'll then have to go through QA and all of that. In terms of future content, we have such an insanely cool list of ideas, and we'll basically pick the size of those ideas based on how well the game does, really. We have a couple of ideas that are so hugely scoped that we'll need to hear enough of an enthusiastic roar to make that happen! 🗣️



(Not a) Thief in the night

Stop comparing it to Thief: Gloomwood has a lot of its own ideas

Info

GENRE
First-person sneaker

FORMAT
PC

DEVELOPER
Dillon Rogers,
David Szymanski

PUBLISHER
New Blood
Interactive

SOCIAL
@NewBlood

♥ It's great to see New Blood engaging with retro-style FPSes that aren't just about pace and gunplay - patience and forethought can be fun too!



It's influenced by Looking Glass Studios' classic tale of Victorian-ish, sort-of steampunk-ish first-person sneaking and stabbing, there's no doubt, but *Gloomwood* can't just be defined by the *Thief* that came before it. In part, because just a few years ago, *Thief* received its third sequel, so it's not like this is a long-dead series brought back by committed fans. Instead, *Gloomwood* is its own thing: an FPS mixed with immersive stealth, survival horror elements, and a brooding, claustrophobic atmosphere that serves to make the Victorian-ish setting feel just as foreboding as you'd hope.

"As a marketing tool, comparing *Gloomwood* to its inspirations like *Thief* is an excellent way to grab attention," says Dillon Rogers, lead developer on the game. "However, there's a lot

more DNA in *Gloomwood* than just *Thief*. It has a good deal of traditional survival horror elements like *Resident Evil* and *Silent Hill*. It has elements of other immersive sims like *Arx Fatalis*. We didn't want to 'make *Thief* again' because that game already exists. We wanted to make something that feels like your first time playing a game like *Thief*, but unfolds as its own unique experience."

Co-developer (and *DUSK* creator) David Szymanski concurs: "Forget an elevator pitch, we condensed *Gloomwood's* pitch down to 'Thief with guns'. Three words and someone already has an idea what to expect, and whether or not they're interested in it. On the other hand, 'Thief with guns' is hardly a complete summation of what *Gloomwood* is." An attaché case-style inventory comes from *Resident Evil 4*, an introductory level influenced by *Call of Cthulhu: Dark Corners of the Earth*, the manner in which players check their ammo riffling on similar systems in *Condemned: Criminal Origins* - and that's just nods and references to other games. "There are plenty of things in *Gloomwood* that are just... *Gloomwood*," Szymanski continues. "They aren't taken from anything else, they aren't trying to reach the standards set by anything else, they're just our own ideas. So if someone were to come into the game and be like, 'I expect this to be exactly like *Thief* down to the last detail', they're going to be disappointed."

On playing the early demo of *Gloomwood* that's been available a few months now, it's



- ◀ With the explosion and blood and viscera everywhere, it's fair to assume this is a scene of wanton carnage.
- ▼ If the buildings were a bit wonkier, this might look like Worcester on a cold winter's evening.



clear most people playing the game won't be disappointed, regardless of what they expect. This is a retro-styled title showing hints of tight design, engaging mechanics of sneaking and shooting, and that aforementioned atmosphere. How it holds up through extended play is, of course, the real test, but the early impressions are decidedly positive. This, in part, comes from the game's time in development and its redesigns along the way – Rogers explains the idea came about while he was in college, but the original plan for a procedurally generated world was binned and the immersive sim approach emerged from his dissatisfaction with the original design.

That shift from procedural general to handcrafted levels is an important one. "As I developed the more core survival horror/immersive sim elements of the game," Rogers says, "I came to an understanding that level design is one of the most important parts of both those genres. It dictates how players explore the world and how they choose which routes to take, encounters to approach, and resources to take. It also gives the world its own feeling of permanence – the player learns over time how the city is laid out and how best to traverse its alleyways and buildings. It was definitely the right way to go with this game."

Sticking with procedural generation would have robbed *Gloomwood* of much of its control over the player's encounters and interactions with the entities within the city. The robed, traffic-light-eyed sentries wandering the cobbled

streets, ready to swarm and blast you with their shotguns should you stray into their line of sight without a plan to escape or dispatch them – they would certainly still carry a slight air of dread regardless of how levels were designed, but the more cultivated approach makes their positioning – their threat – all the more effective.

"Maybe you could argue that a systemic game could benefit from having an algorithm randomise the way elements interact," says Szymanski. "There are certainly titles that run with this idea – *Noita*, *Spelunky*, *City of Brass* – but that's not really the experience we're going for

with *Gloomwood*. The point is to immerse you in the world and the journey you're on, and for the emergent systems to aid in that, not to just

give you a lot of emergent systems for their own sake. That's not really something that would benefit from procedural generation."

Plus, he adds, it's actually probably less work to make levels by hand than it is to code a procedural generation algorithm that could make "natural, interesting, varied levels with good flow and an interesting journey".

There's a lot of work going into *Gloomwood* – it's the sort of project driven by a clear passion, and with New Blood backing it (and Szymanski helping out), the signs and portents are definitely pointing in the right direction. Early Access looks like it will happen later in 2021, so there's still a fair bit of time to polish everything off here – and, honestly, it's going to be a difficult wait. 🗨

"The player learns how the city is laid out and how best to traverse it"

BAKE OFF

Development is never plain sailing, and it can get harder when you're taking requests and listening to feedback. "*Gloomwood* uses baked lighting to give it that late 1990s look and feel that games like *Quake* and *Thief* had," Rogers explains. "After the demo, one of the most frequent requests we got was to allow lights to be breakable. However, because our engine expects developers to use dynamic lighting for... well, dynamic lights, we hit a huge snag when we tried to implement baked maps that can dynamically change. We had to implement a rather hacky system to bake the maps – to allow them to be swappable at runtime – that is still clunky and prone to errors. It's been a big headache."



It came from the desert

Developer Gregorios Kythreotis tells us how Star Wars, Studio Ghibli, and Zelda have all inspired the upcoming *Sable*

Info

GENRE

Adventure

FORMAT

PC / Mac / XBO /
XB S/X / PS4

DEVELOPER

Shedworks

PUBLISHER

Raw Fury

RELEASE

23 September 2021

The last time we covered *Sable* was two years ago – then midway through its development, it was already looking truly special even back then. So special that, when it was shown off at E3 in 2018, interest in the game exploded, and *Sable* rapidly became one of the most anticipated indie games on the horizon. It's not hard to see why: set on a desert planet that may or may not be Earth, *Sable* is an open-world adventure about exploration and discovery: cast in the role of a young woman

on a quest to find five lost masks, you're free to roam the landscape as you see fit. You can hurtle along sand-dunes on your hoverbike, climb cliffs in search of abandoned buildings filled with secrets, or chat to the world's masked inhabitants, who'll provide quests and hints as to where to go next.

Sable recalls the design of such video game staples as *Ico*, *Shadow of the Colossus*, and *Breath of the Wild*, but its sci-fi world reaches further afield – the depiction of a ruined world where remnants of a previous society's tech is treated with almost holy relevance recalls Walter M. Miller Jr.'s novel, *A Canticle For Leibowitz*. And isn't the intrepid heroine roaming post-apocalyptic vistas akin to the one in master animator Hayao Miyazaki's manga and animated feature, *Nausicaä of the Valley of the Wind*? Yes, agrees *Sable*'s creative director Gregorios Kythreotis, who reaches up to grab a book of *Nausicaä* artwork from a shelf. "I picked this up from the Studio Ghibli Museum in Japan," he tells us, referring to the Tokyo-based attraction dedicated to the work of Miyazaki and his collaborators. "At the beginning, before we kicked off with *Sable* in 2017, I went to the museum. I was going through some personal stuff at the time – a few of my family members had just died – and I think, going there, experiencing that, and then coming back to start





work on this project, something about it was always going to just stick with me and always be part of it.”

Echoes of Miyazaki’s clean, delicate style are easy to spot in *Sable*’s gorgeous cel-shaded environments, and the artful minimalism of the visuals also extends to the world’s design: unlike other titles of its type, *Sable* doesn’t contain combat or harsh survival mechanics – instead, it offers a calmer experience where the player is rewarded rather than punished for gliding off into the unknown. “We don’t have a main narrative that the player’s required to follow,” Kythreotis says. “We have a beginning and end, but the middle is completely loose, like a series of short stories. So people can play it and pace it how they want – if you get stuck on a puzzle, we don’t ask you to finish it, you can just walk away, and the same’s true for a piece of narrative content... we tried to have an element of freedom in the game, like *A Short Hike* for example, or *Breath of the Wild*. But then the flip side of *Breath of the Wild* is that there’s combat; there’s always things that can kill you. We don’t really want to do that – we wanted to give players the confidence to just explore, so in that sense, it’s quite a relaxing game.”

Back when *Sable* was in its earliest stages, studio co-founders Kythreotis and Daniel Fineberg thought carefully about their individual

“We wanted to give players the confidence to just explore”



skills and how they could use them to make a video game. At this point, the pair were still working out of a garden shed belonging to Kythreotis’ parents (hence the studio’s name, Shedworks) and had mostly worked on smaller-scale projects like websites and apps. With Fineberg’s degree in Comparative Literature and Kythreotis having studied architecture, the duo asked themselves: what kind of project would suit their strengths? And equally important: what kind of game could they make as a tiny independent team of two?

Many of the design decisions that make *Sable* so distinctive sprang from those conversations: the clean, cel-shaded rendering style, for example, meant that the world could be constructed from simpler, low-poly assets. “A lot of what we do revolves around making sure we can produce this,” Kythreotis says. “Like the masks that the characters wear – they’re quite an integral part of the culture and mechanics of the game, but originally they came about ➔

▲ *Sable*’s come a long way since it was unveiled in 2018, and so have Shedworks. For one thing, they’ve now moved from a garden shed to a proper office space.

◀ One of *Sable*’s simplest pleasures comes from exploring old buildings and wondering what their function might have been in this alien world.

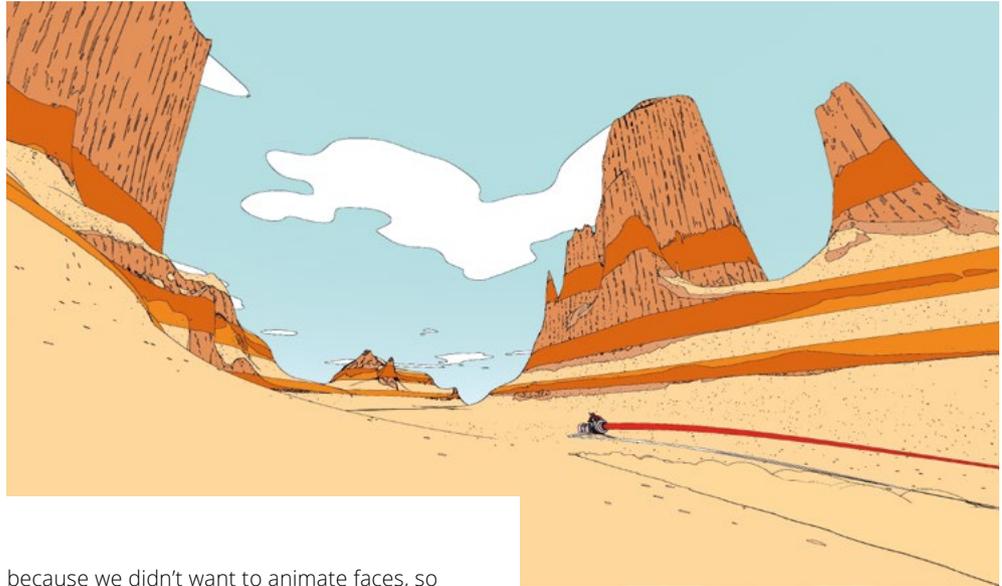
THE GHIBLI EFFECT

One of the names that often comes up when journalists talk about *Sable*’s style is Jean ‘Moebius’ Giraud, the influential French artist. Moebius, Kythreotis readily admits, is one ingredient in the stew of touchstones that *Sable* draws from. “It’s kind of circular, right?” Kythreotis says. “*Nausicaä*, specifically, was inspired by *Arzach*, Moebius’ work. At the start of *Sable*, the initial idea came from watching *Star Wars: The Force Awakens*... We started working on *Sable* in 2016, so a year before *Breath of the Wild* came out, but obviously you can’t ignore it, and we haven’t ignored it. But if you look at that game, it’s more of a Studio Ghibli game than *Ni no Kuni*, actually, in terms of atmosphere. So it’s all circular – a lot of the things we make are inspired by other things we’ve looked at. I think it’s unavoidable.”

➤ *Sable's* cel-shaded look began long before the game itself, and emerged from Kythreotis' rendering experiments while he was studying architecture.

GREAT EXPECTATIONS

When *Sable* was greeted with a wave of enthusiasm in 2018, the media interest must have been exciting for Shedworks – but did it also bring an element of pressure of meeting the public's expectations? "Definitely," Kythreotis says. "We've been really wary of over-promising, so everything we've shown publicly has tended to be at least a year old. But that's a dangerous thing to do as well, because you can promise something and then it just doesn't work for the game, and then you feel maybe you want to cut it then you can't... the flip side is if you're ambiguous then people start projecting stuff onto the project. So the demo [released on Steam] was a big part of that expectation management – to communicate exactly what the game is."



because we didn't want to animate faces, so we said, 'How can we turn this weakness into a strength?' A lot of decisions have been made like that. The lack of combat for example: we love combat in other games, but we thought, 'We don't think we can do it to the level that *Dark Souls* can'."

Kythreotis' architectural training, meanwhile, allowed him to focus on one of *Sable's* most immediately striking elements: the vast buildings that jut out of the desert landscape. At once ancient-seeming and technologically advanced, they're the hiding place for many of the game's mysteries and puzzles. "I think one of the cool

things that we've tried to do with the world-building is make sure it feels like a special place, but these are just ordinary people living here," Kythreotis says. "It feels somewhat grounded in that sense – it has a stronger sense of life... also, we give the player such a freeform set of skills to explore, the buildings have to be fully realised – they can't just be a shell. So having a background in architecture has helped with that, and the way I try to approach it is by thinking about the function, the context, but also trying to create buildings where it's less of a structured experience. Instead, we're nudging players towards opportunities where moments can occur rather than having anything specific in mind. But level designers do a lot of the things architects do in terms of using openings and windows to create viewpoints, or using spaces to create psychological effects and emotions."

Exploration and mystery are so key to *Sable* that one of the early design decisions Shedworks made was to avoid giving the player any kind



▲ *Sable* contains lots of puzzles, but there isn't pressure on the player to complete them.





of mini-map to follow. “We didn’t want to have mini-maps in our game, and I feel that way quite strongly,” Kythreotis says. “I feel like it’s the equivalent of going abroad somewhere and just staring at your phone the whole time. Because it optimises the way you navigate the environment, so it kills the experience. I feel like a lot of games suffer from that. So we try to make sure we designed the game to solve that problem. Like, even *Skyrim*, they had the compass on the top of the screen, and that solution’s very elegant.

In that regard, we looked at *Ico* and also the *Souls* games, because those environments are constructed so that you understand them without needing a map at all. *Shadow of the Colossus* has a map, but it’s really zoomed out, so it’s more about locating yourself in relation to other landmarks.”

As careful as Shedworks has been to keep *Sable’s* scope under control, making an open-world game as a two-person studio still presented a challenge – not least because the landscape’s studded with dozens of NPCs that all needed to be designed, animated, and written for. “I definitely understand why ruins are a useful video game tool,” Kythreotis observes with a rueful smile. “It’s so much easier to make a building that’s uninhabited and implies a history than inhabit it, because then you

“It’s the equivalent of going abroad and then staring at your phone”

start introducing the uncanny valley into your architecture. That’s such a hard problem to solve, and so much effort goes into that.”

The sheer scale of the game – not to mention the stresses of Covid-19 on production – have meant that development has taken rather longer than Shedworks had originally envisioned. But with *Sable’s* release mere weeks away (it’s slated for 23 September), the largest hurdles have now been cleared, and Kythreotis is already anticipating a well-earned rest. “Covid made

production much harder, but we’re on track, and I’d say the reception to the demo was really good. And again, I’d say we’re setting expectations – we

want people to realise we’re a small team making a pretty ambitious project. Maybe we shouldn’t have made such a big project, in retrospect... I think we’re ready to move on, creatively, and have been for a while. Maybe we’ll want a smaller, palate-cleanser project before we dive into another long one. Because we never thought this project would end up being four years of production. You’re constantly moving that end date, so you push yourself to get to that, and then that gets moved back. It’s hard to know how to pace yourself when the distances start changing. It’s hard to go from a 100 metre sprint to 800 metres, you know? You’re going to suffer. So yeah, we need a break! I think we deserve one.”



▲ Like *Journey* before it, *Sable’s* lack of combat leaves you free to explore and soak up the world’s ambience.



Black hole sun

Director Yarden Weissbrot talks us through the stresses and inspirations behind his team's stunning-looking debut, GRIME

Info

GENRE
Metroidvania

FORMAT
PC

DEVELOPER
Clover Bite

PUBLISHER
Akupara Games

RELEASE
TBC 2021

SOCIAL
@Play_GRIME

Ideas can come from all kinds of familiar places, but still combine to create something different and unexpected. *GRIME* is a tidy example: it's the work of five self-described "Dark Souls enthusiasts and *Hollow Knight* fanboys" from Israel, takes elements from such games as *Bloodborne*, *Sekiro: Shadows Die Twice*, and *Salt and Sanctuary*, and yet looks very little like the titles that inspired it. In essence, *GRIME* is a side-scrolling hybrid of RPG and Metroidvania, a bit like *Hollow Knight*, but with a detailed combat system that allows you to absorb your enemies' abilities and adopt them as your own. Also: the central character is a deathly looking humanoid with a swirling vortex in place of a head, and the levels you roam are surreal hellscape populated by various ghoulish and twisted creatures.

"I've always been a huge fan of strange and surreal designs," says *GRIME* director Yarden Weissbrot. "The very first designs I did were of the Rockhead [enemies] and face-shaped rocks in the desert, followed by an enemy with a giant hand as a torso. At first, I drew these for no other reason than I find those kinds of designs cool. The world and story were non-existent at that point; all I knew was that it was likely going to be a world with an attachment to distorted human shapes since that's what I like drawing."

GRIME is the debut game from Clover Bite, a team of former students who got together at Israel's Tiltan School of Design and Visual Communications. Not long after the project began, the basic side-scrolling combat was in place ("*Salt and Sanctuary* was our main reference for the core combat and movement mechanics," Weissbrot says), but there was a certain nervousness about pushing the game's scope too far. "Relatively early into development, and after playing a [lot] of *Bloodborne*, I became really fond of the parry mechanic used to spice up the *Souls* combat," Weissbrot tells us. "But, since this was our first game and I had the added responsibility of directing the project, I was terrified of deviating from the established formula. To be blunt, I just didn't think we'd have the ability to pull it off and create something entirely our own from a gameplay perspective – especially with me at the helm and with basically

▲ *GRIME*'s inhabitants may be creepy, but they aren't just monsters – they have the makings of a society, as these simple cave paintings prove.

▶ The game's built in Unity, along with tools "we made ourselves as we needed them, such as a level streaming system, enemy AI management, and so on," Weissbrot says.



zero game-making experience. I didn't want to risk the game and everyone's faith in it by chasing a mechanic I just recently grew to like on a whim."

All of that changed, however, when another game by FromSoftware emerged – *Sekiro: Shadows Die Twice*. Says Weissbrot: "The *Sekiro* announcement trailer came out, and I became obsessed with analysing it and figuring out how parrying works in it. I realise this may be pretty silly, but it honestly felt in a way like I have permission to make the parry as a central mechanic for the game now that FromSoftware did just that. After focusing my attention on designing what I thought would be a unique and cool look for a parry, I landed on using a black hole to sort of catch enemy attacks with and then pull them in."

The influence of FromSoftware's action output runs deep in *GRIME*, then, from the combat to the Trait system, which is where those purloined abilities can be upgraded up to five times. "In order to level up traits, you need to get Hunt points, which are typically dropped from challenging enemies that don't respawn on defeat," Weissbrot explains. "The idea was to have the player want to hunt down enemies, both weak and strong, for the challenge and for their added power, like the predator they are."

One area where *GRIME* will diverge from the *Souls* games and *Hollow Knight*, however, is its difficulty – it's an aggressive, violent game, sure, but it won't completely beat you down from the outset. "I think *GRIME* overall will be a lot less punishing than both, since dying in the game doesn't drop all of your currency," says Weissbrot. "The goal with removing that penalty was to encourage players to play recklessly and

aggressively, or allow them to take different routes whenever a certain one is too hard, rather than forcing them to travel back to where they died, in order to return what could be an hour or more of hoarded progress."

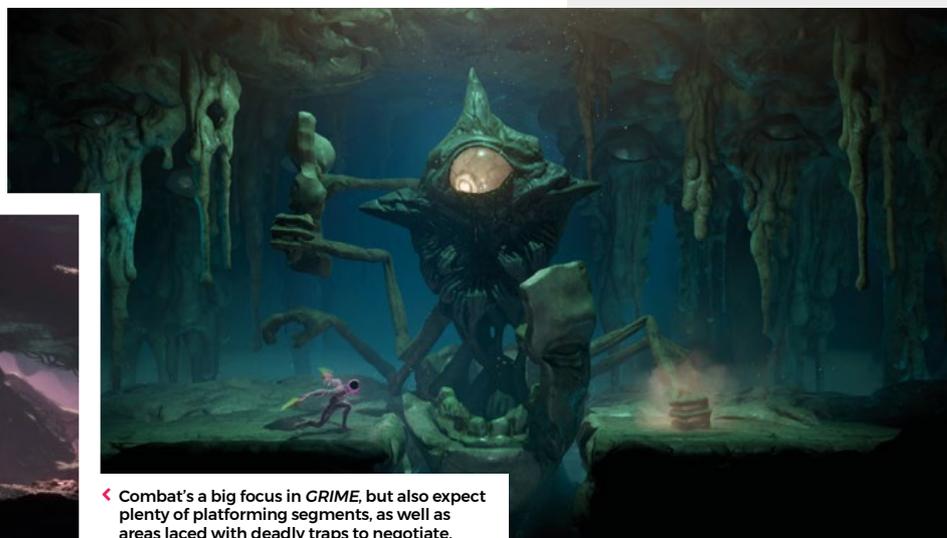
Behind the scenes, development has thrown up some *Souls*-like challenges of its own. "As I've mentioned, this was our first game," says Weissbrot. "In many ways, everyone ended up wanting to be in charge and doing something different with *GRIME*'s gameplay. This hadn't immediately changed after I'd been chosen to lead the project, as everyone still had little reason to trust my judgment when I, like them, had no experience either... It was a difficult period, filled with self-doubt. I had to trust that I was making the right decisions to keep everything from losing focus and going in different directions. We've had a lot of clashes, and [I feel] bad about rejecting ideas I didn't think fitted. I'm not a total saint... but eventually, we [understood] each other's faults and merits. I like to think we've all grown as people. These days, we joke about these kinds of things openly."

GRIME's development is on the home stretch now, and it looks remarkably polished for a small team's debut. Due out later in 2021, it could prove to be one of the year's most darkly enjoyable Metroidvanias. For more of its surreal artwork, head to page 68. 🗨

"It's an aggressive game, but it won't completely beat you down"

WEAPON OF CHOICE

"My favourites are the maul weapons from a visual perspective, since they have this clawing hand that shapeshifts during their special attack," Weissbrot says, when we ask which of *GRIME*'s weird armaments he likes best. "But from a gameplay perspective, I definitely think the lanterns have the most unique playstyle. They're all about stacking certain debuffs on enemies, and then either unleashing that debuff or interacting with it via another weapon. For example, the Wisp lantern puts stacks on enemies that turn into homing projectiles once that enemy is destroyed, targeting other nearby enemies. However, you can combine that with another weapon that can tear out those projectiles while that enemy is still alive, dealing additional damage for each torn projectile."



◀ Combat's a big focus in *GRIME*, but also expect plenty of platforming segments, as well as areas laced with deadly traps to negotiate.

That was the month that was



02



01

01. Inevitable GTA 6 rumours

Grand Theft Auto 6 confirmed? Not really, but it's clearly inevitable – and a leaker with a solid track record of getting big games right has claimed a few things that have pricked up the ol' ears. First up, we shouldn't expect *GTA 6* until at least 2025, which is an extremely sci-fi-sounding year. Second, the game is to be a modern-day version of *Vice City*, rather than going back to the 1980s again, in order to give Rockstar lots of options when it comes to the next *GTA Online*. Finally, the leaks claim *GTA 6* will see an ever-evolving map similar to that employed by Epic with *Fortnite* – a foundational landmass that changes and gets tweaked over time, though you'd assume that'd be limited to the *GTA Online* section rather than single-player. Anyway, rumours – aren't they great?



03

02. Steamed

Suddenly, Steam announced the Steam Deck – a handheld gaming device based on custom AMD Zen 2 architecture, with an array of different storage options for your pennies. Said pennies ranging from £349 to £569 for the model with a 512GB NVMe SSD and some other gubbins. It also features a touchscreen, trackpads to make up for a lack of mouse, and gyroscopic controls if you want to use those too, *and* it's compatible with regular PC peripherals, *and* you can hook it up to your TV, *and* it's getting a dock. It's basically a mix of the various GPD portable PCs and a Switch, and frankly, it looks fantastic. The Steam Deck is planned to start shipping at the end of this year.

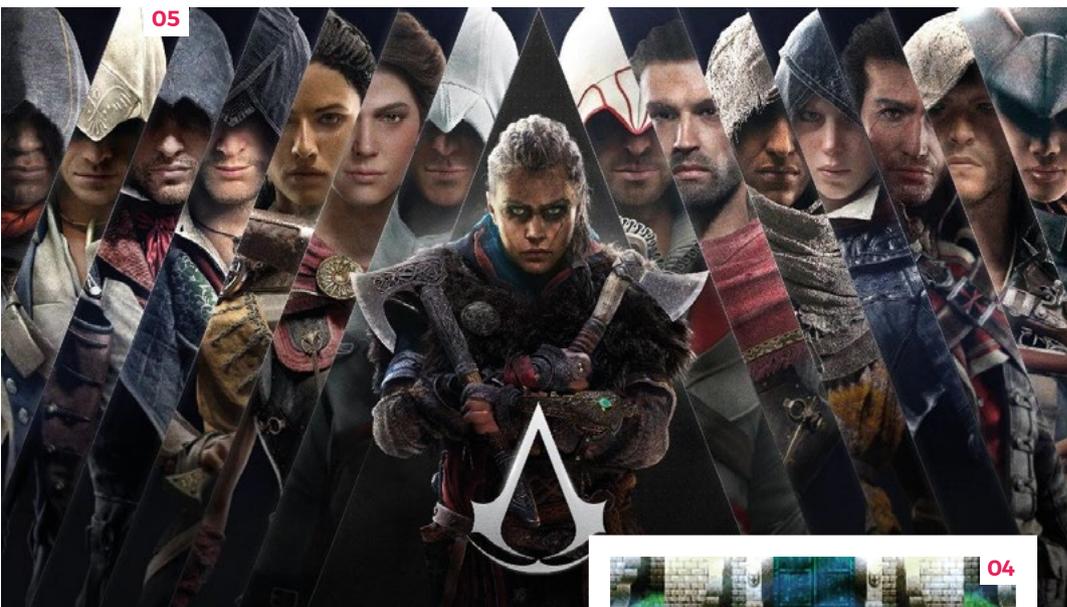
03. Netpiz(ell)

Netflix is set to dip its toe into the world of video games, as it announced the hire of Mike Verdu – the company's new (and first) vice-president of game development. Verdu previously worked at EA, as well as at Facebook on the Oculus side of things. A Bloomberg report claims Netflix's plan is to have games available on its platform within the next year, with video games sitting right there alongside your usual TV series, stand-up specials, and terrible sci-fi movies from 1995 you never knew existed. It's fair to say few are expecting triple-A titles to pop up on the platform, but it's going to be interesting to see how this all works out.



Yep, someone's made a Greggs in *Far Cry 5* (h/t PC Gamer)

Bloober Team rumoured to be working on new *Silent Hill*; doesn't deny it



04. RIP Near, champion of emulation

Near/Byuu, a much-vaunted member of the emulation and homebrew scene, has taken their own life. The news arrived via friends of the developer late in June, and followed a prolonged and targeted harassment campaign originating from the Kiwi Farms message board. Near's work included creating bsnes – the first SNES emulator with 100% compatibility – and higan, a multi-system emulation platform that pushed for accuracy in its emulation over anything else. It didn't begin and end with emulation though, with their focus on translation projects culminating in the excellent *Bahamut Lagoon* release earlier in 2021; a passion project decades in the making, it brought a title that never saw an English release to a much wider audience. A full list of international suicide helplines can be found here: wfmag.cc/helplines.



05. To Infinity

Ubisoft was prompted into revealing more about the next step for its time-travelling stab-'em-up franchise, announcing the existence of *Assassin's Creed Infinity* following a Bloomberg report revealing the project. It's actually more a platform than a game, with the series heading down the 'games as a service' path, with development split across multiple Ubisoft studios for different, vague things that Ubi didn't explain much. It's... inevitable, really. Just like Ubisoft not getting shot of a bunch of senior members of staff accused of various negative behaviours. But hey.



Image: Heritage Auctions, HA.com

06. 64 (million)

Potentially the most ridiculous thing in all of history transpired partway through July, as a mint, sealed copy of *Super Mario 64* sold for \$1.56 million (£1.126m) via Heritage Auctions. The copy was rated 9.8/10 condition-wise, or A++, and... nope, that's about it – nothing special about it beyond being a classic game in fantastic condition. Over a million quid. A copy of the original *Zelda* on NES sold for \$870,000 (£628,000) days prior, but at the very least that auction was for a version of the game that was limited, coming from an early production run lasting just a few months in 1987. This million-pound *Mario*... it's just *Mario 64*? The game's good – great, even (check out page 36 for more on that) – but worth over a million quid? Well, time to get rooting through our cupboards for other mass-selling classics that might be worth a few bob.

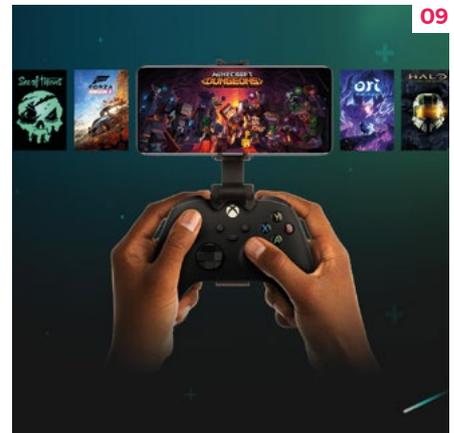
Sony acquires *Returnal/Stardust* developer Housemarque



Final Fantasy IX to be turned into animated series



08



09

ACTIVISION®

BILZARD

07

07. Acti-Blizzard sued

Activision Blizzard has been sued by the California Department of Fair Employment and Housing following two years of investigations into the company, primarily focusing on Blizzard. Allegations of a 'frat boy' culture surfaced, with claims women faced harassment in the workplace alongside other inappropriate behaviour. Female employees were overlooked for promotions, undermined, and one female member of staff took their own life following intense sexual harassment involving multiple male members of staff, according to the suit.

Activision Blizzard has strenuously denied the claims. Read the full details here: wfmag.cc/ActiBliz.

08. WB wnb sold

That's 'will not be', by the way. Rumours of WB Games' death have been exaggerated it seems, as the company reported it will only be letting go of Playdemic, which was sold to EA in June. The rest of the publisher and its stable of dev teams is staying put at Warner HQ, with the games division included in the \$43 billion (\$31bn) purchase of WarnerMedia by Discovery. Earlier rumours had all manner of suitors queuing up to purchase the *Batman*, *Back 4 Blood*, and *Mortal Kombat* publisher – turns out, in the end, Discovery *discovered* keeping gaming under its wing was a good idea.

09. Head in the

Xbox fairly quietly gave its Xbox Game Pass Ultimate members quite the treat as it released its cloud gaming service for all subscribers. At the time of writing, it works on Windows 10 PCs and Apple phones and tablets, allowing Ultimate subscribers to stream hundreds of titles from the Game Pass library. It's been in beta a while, so the launch isn't a complete surprise, but it did sort of slink out there without a gigantic amount of fanfare – and having tested it briefly, it works well. It's a great addition to the subscription. And Sony, once again, looks decidedly old-fashioned with its approach to subs services.



Portal creator Kim Swift joins Xbox Game Studios' cloud gaming division

Phil Spencer: 'Xbox controller might get an update; PS5 controller has done a nice job'



10. S(um)old

The purchase of Sumo Digital was agreed in July, with Chinese giga-vestment and mega-internet firm Tencent valuing the company at around £919 million. This marks the second billion-pound (or thereabouts) purchase of a UK-based video games company this year, following Codemasters' acquisition by EA for around £875m. The Sumo purchase will see no changes in management and will help the company bring its titles to a broader audience, with the door to the Chinese market now held wide open. The only question it has us really asking is: who's next? You can buy a few subs from us if you like, Tencent.

11. All dogs...

The inspiration and physical model for *Fallout 4*'s Dogmeat, a German Shepherd named River, passed away. The announcement came via previous *Fallout* staffer/current Capybara Games studio director, and owner of River, Joel Burgess, who tweeted out a tale of how the pooch's visits to the Bethesda office inspired the design of Dogmeat beyond mere looks.

Her companionship pushed developers to pivot from a more aggressive, attack dog-style of character to the one we ended up with: a tenacious, inquisitive, and helpful little doggo with wonderfully expressive eyebrows. Even if Dogmeat did often set off traps, he was still a good dog (they're all good dogs). Xbox donated \$10,000 to the Montgomery County Humane Society in memory of River, which is a nice little break from all the bleakness. Adopt, don't shop, and all that jazz.



12. Not that one

No, this isn't about the cheeky, deer-killing pub-lover from *Location, Location, Location* – it's just too brilliant a crap joke to miss. Anyway, the other Phil Spencer – the Xbox one (of Xbox One fame) – was doing the rounds in recent weeks and months, talking up the future of gaming. One future of gaming involves nothing but multiplayer games, but that's not the future that Phil (not from Channel 4) sees. Speaking to the Guardian, Spencer the Xbox boss said of traditional, narrative-led releases: "We're probably building more of those now than we've been in the history of Xbox." Kirstie Allsopp was unavailable for comment.

PlatinumGames offers *Bayonetta 3* update: still exists, still being made. Phew



Sean Murray's head mod adds Sean Murray's head to *No Man's Sky*

Preview

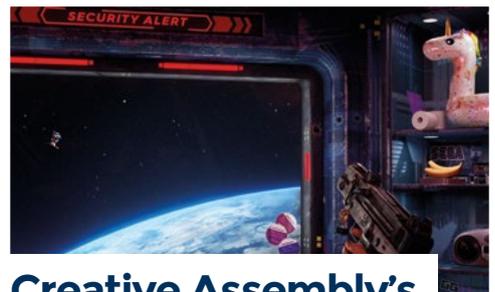


Lost Ark ↑

Amazon Games is taking a different tack with this release – it's an already-established free-to-play MMO that's won plenty of awards in its home nation of South Korea. Amazon has jumped in to help spec the game up and bring it to the west, and frankly it's looking very pretty indeed already. A contender to the *Genshin Impact* hegemony? We shall have to wait and see.

Pathfinder: Wrath of the Righteous ↓

Russian studio Owlcat Games returns with its follow-up to breakout CRPG success *Pathfinder: Kingmaker*. *Wrath of the Righteous* is more of that good and pure isometric role-playing that worked so well before, based on Paizo Inc's Pathfinder tabletop series. And, like before, it's gone via the Kickstarter route, bringing in an incredible \$2 million (£1.5m) to help fund development. It releases 2 September, so really not long to wait now.



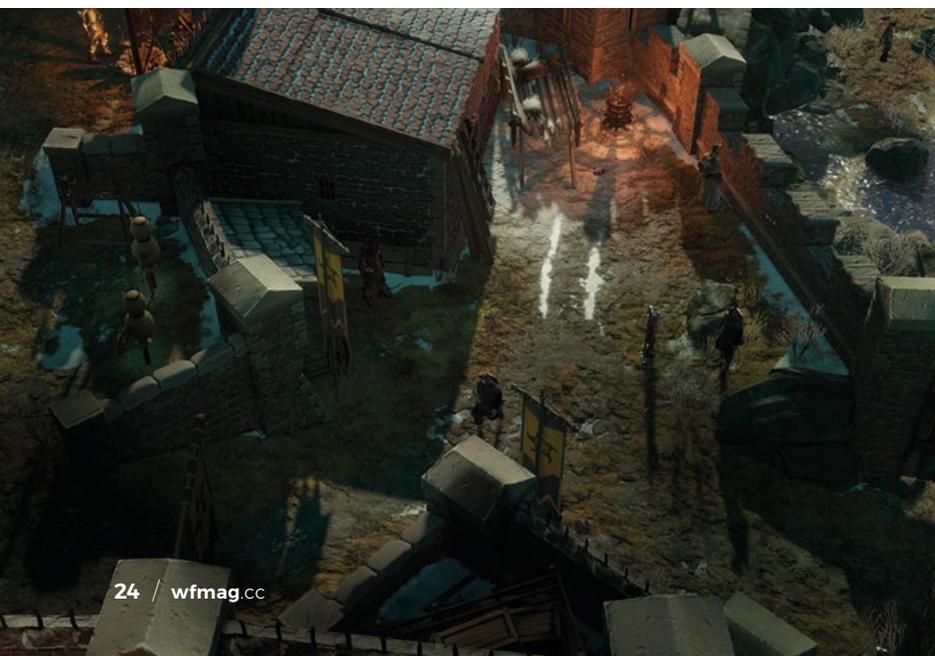
Creative Assembly's 'New Sci-Fi FPS' ↑

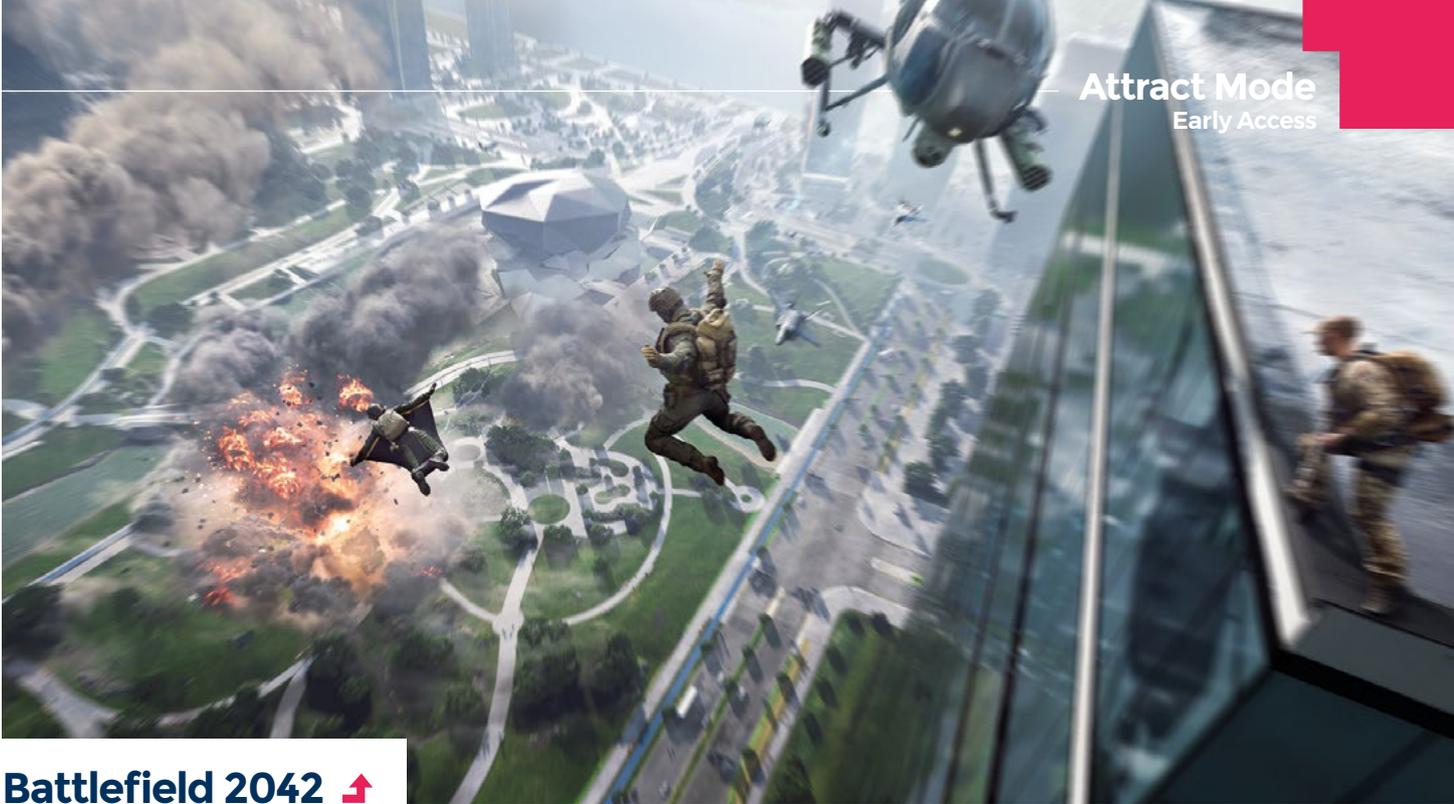
The team which worked on the pretty good *Halo Wars 2* and the utterly stupendous *Alien: Isolation* is back with another sci-fi offering: this time it's a multiplayer first-person shooter, and... that's roughly all we know about it. The very early blurb says it'll be a 'fresh take' on the genre and that players will be able to 'embrace a variety of creative playstyles'. Honestly, it's already enough to get us salivating.



Wirewalk() ↑

One of those where sticking with the blurb is actually helpful, *Wirewalk()* is an 'adventure where you raid virus-riddled systems for the lols'. The original Game Boy aesthetic is what helps this one stand out, as you make your way through a top-down RPG mixing in multiple elements – combat, challenges, even a bit of football along the way. Also, there's a heavy emphasis on dogs and the act of stroking said dogs, so there's a bonus for us all.





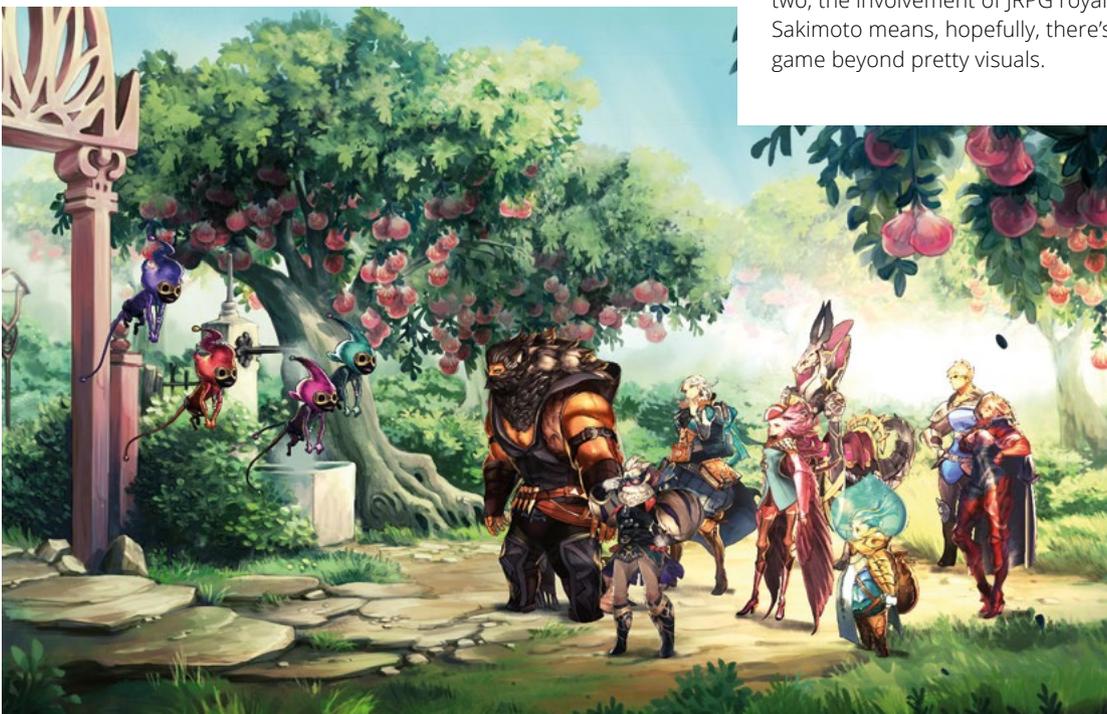
Battlefield 2042

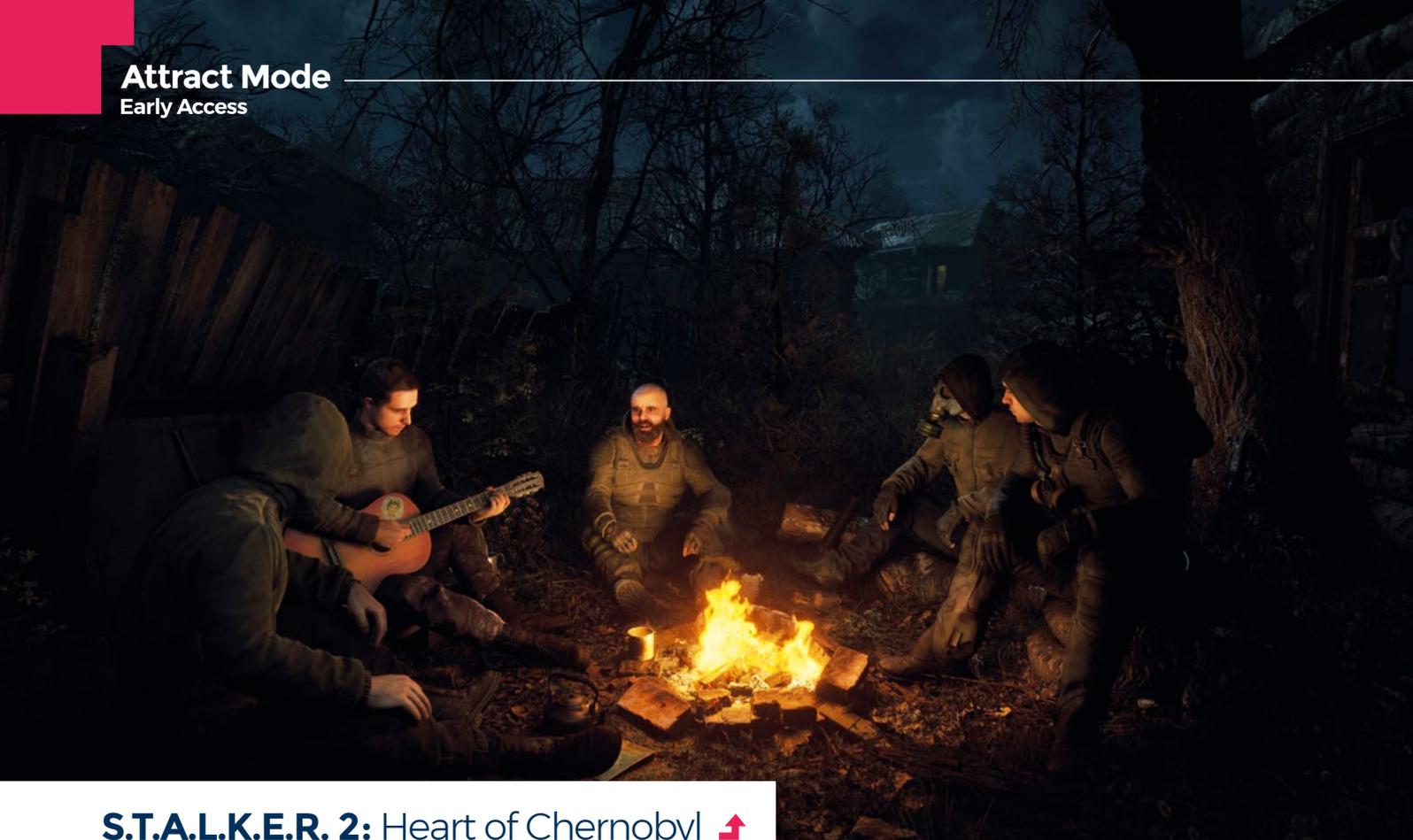
We're heading into the distant sci-fi future of... well, 21 years away. Oh god, time, what are you doing? Anyway, *Battlefield 2042* is bringing us a future war that looks surprisingly similar to the modern conflicts the series has so proudly shown off over the years. Except with wingsuits. The headline attraction so far is the introduction of 128-player matches – and the sub-headline attraction within that is DICE's intention to pad those player-lists out with up to 64 bots where necessary. This means that you won't run into any empty matches, and that you might actually be able to take out some tangos. Soldiers. Whatever.

All the battle passes and seasons plans and whatever else are already being explained in detail by EA – to be expected given live service titles earn the company a *lot* of money and, yes, *Battlefield 2042* is multiplayer-only. There'll also be betas and test versions showing up in the run-up to the game's 22 October release, so plenty of opportunities to get involved in yet another massive online war (and get shot in the face).

Astria Ascending

The luscious-looking JRPG from Artisan Studios is being made in the most part by the Quebec-based studio, but ears are perking up at the sound of two names involved in the project – Hitoshi Sakimoto is composing the score (as seen in *Final Fantasy XII* and *Vagrant Story*), while some writing duties are coming from Kazushige Nojima (of *Final Fantasy X* and *Final Fantasy VII Remake* fame). It doesn't automatically make *Astria Ascending* a winner, but it shows two things: one, it's a project with a lot of ambition, and two, the involvement of JRPG royalty like Nojima and Sakimoto means, hopefully, there's something to the game beyond pretty visuals.





S.T.A.L.K.E.R. 2: Heart of Chernobyl ↗

Technically, *S.T.A.L.K.E.R. 2* will have been announced *twelve years* before its eventual release – should that come to pass – on 28 April 2022. Originally announced back in 2010, this extremely long-awaited sequel has been through the wringer in the interim – officially cancelled at one point with developer GSC Game World closing down, several clones and offshoots of the ambitious sequel have appeared in various guises along the way. But here we are, eventually, but also suddenly, not too far removed from the sequel to one of the finest survival/difficulty/terror-based FPS games ever made. OK, and another technically – this is *technically* *S.T.A.L.K.E.R.*'s third sequel as there were two others back in the day.

Anyway, what do we expect from *S.T.A.L.K.E.R. 2*? More dour-faced survivors making whatever they can of a life inside the exclusion zone, dogged by mutants and generally strange goings-on. More fearing other humans more than the creatures in the zone. More having the absolute bejesus scared out of you as a hitherto-invisible irradiated cloud of doom appears right on top of you. If this has just 10% the impact of the first game, it'll be a winner.

Dying Light 2 Stay Human ↘

It feels like it's been a while, but the date is now approaching: *Dying Light 2* will release on 7 December. At the end of this year, we'll finally be able to see how accurate the hype was – just how much your actions will impact the story of the zombie-riddled civilisation. The original *Dying Light* was a slow burn, but ended up getting the respect it deserved as an energetic, atmospheric mix of parkour and brutal melee combat. If the second game keeps on that track, we'll be onto a winner at the year's end.





Hot Wheels Unleashed ↗

Milestone isn't averse to licensed tie-ins – in fact, the majority of its releases are based on one automotive brand or another. That said, it's still a surprise – and a pleasant one – to see the Italian team involved in bringing a game of wacky toy cars to the world. There's plenty of focus on the accuracy – and collectable nature – of the Hot Wheels cars in the game, but also: there's loops!

Advance Wars 1+2: Re-Boot Camp ↘

Two of the best turn-based strategy titles of all time are getting the Nintendo do-over treatment, and are heading directly for Switch at the end of 2021. Both of the initial *Advance Wars* games are bona fide classics in the genre, so it's bound to be great fun heading back into them – the only question rests on the change of artistic style, moving from sprites to 3D models in a fashion that can only be described as "...oh".



Dragon Quest XII: The Flames of Fate ↘

Not long after turning 35, *Dragon Quest XII* was announced. Fittingly, it's going to be a dark fantasy tale 'for adults' – though probably not in the same way *Razze* is for adults – and it'll be the first game in the long-running series to get a simultaneous worldwide release. Not much else to go on for now, but definitely one to keep an eye on, it being a JRPG legend and all.



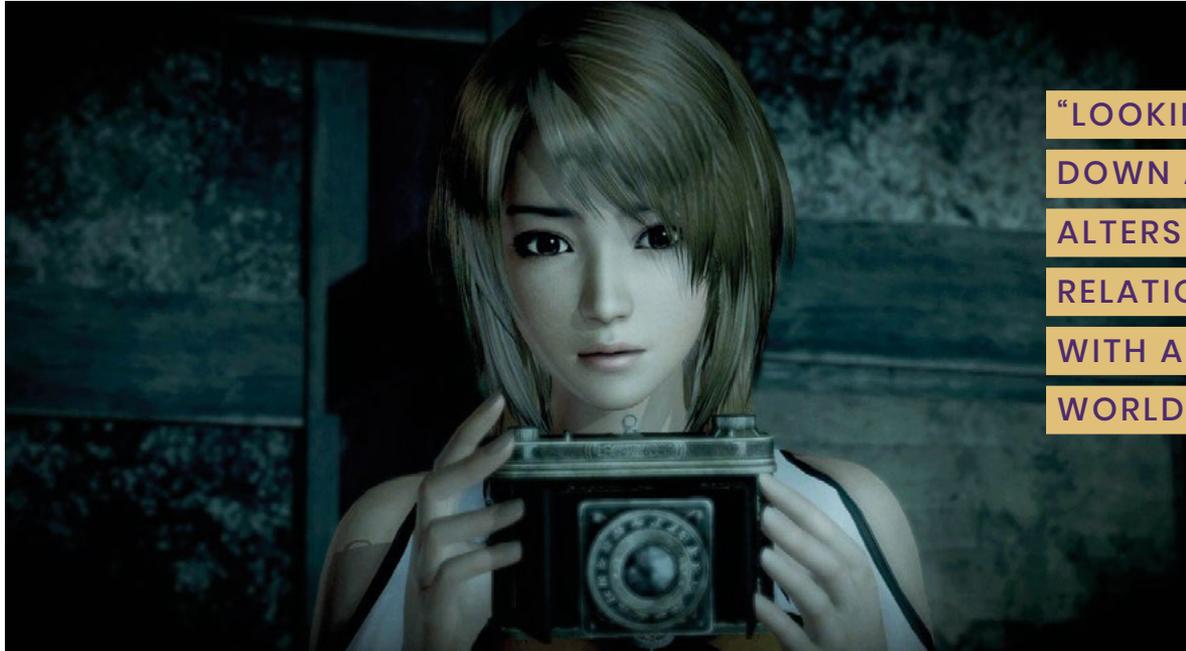
Sonic the Hedgehog ↗

There's absolutely nothing to go on here beyond three key aspects: Sonic Team is working on a new Sonic game, it's coming in 2022, and there's a teaser trailer in which our little blue friend is 'going fast'. Hope beyond hope it isn't a stinker like so many 3D Sonic games have been – if we all just hope together, maybe the dream will come true.



WITH SEVERAL PHOTOGRAPHY-BASED INDIE
TITLES ON THE WAY, WE FIND OUT WHY TAKING
PICTURES IN GAMES IS SO SATISFYING

WRITTEN BY DAN COOPER



“LOOKING
DOWN A LENS
ALTERS OUR
RELATIONSHIP
WITH A GAME’S
WORLD”

From the earliest cave-art to digital photography, making pictures has long been an intrinsic part of our being. When the world feels too big or chaotic to comprehend, photography can act as a form of curation, allowing us to extract meaning from chaos by choosing and capturing a chosen instant.

In their purest form, video games serve a similar function. Whether wielding an axe in *God of War* or a hoe to tend your pumpkin patch in *Stardew Valley*, games are a simulation of humanity's basest instinct: the need to bring order to the world. And, as game worlds have become larger and more complex, players have been granted increasingly sophisticated cameras which they can use to document their own unique experiences within these spaces.



▼ *Gekibo* got a PlayStation sequel called *Polaroid Pete* in the US. Oddly, its camera isn't actually a Polaroid.

Many games are content to include a photo mode as a 'bolted-on' feature, and do so brilliantly, providing players with an endless array of creative options to capture thrilling or scenic moments. Over the past few decades, though, some developers have chosen to tightly weave the camera into their narratives, redefining the way each game's protagonist, and to some degree, us as players, perceive that world.

Looking down a lens alters our relationship with a game's world, and even the medium's earliest attempts to capture this relationship reflected this. *Gekibo: Gekisha Boy*, released in 1992 for the PC Engine, was a side-scrolling snap-'em-up where the lead character – a recently orphaned, depressed teenager – salvaged his spiralling academic career by navigating peril-filled levels and taking photos along the way. Whether it realised it or not, *Gekibo* developer Tomcat System had hit on one of the fundamental tenets of photojournalism: that a single image can find meaning in a random, sometimes senseless world.

Other games soon used cameras as a way to explore a virtual environment; 1999's *Pokémon Snap*, for example, allowed players to observe and appreciate fantastical creatures in their 'natural' habitats, rather than engaging in the morally dubious practice of trapping them in Poké Balls and occasionally forcing them to fight each other. Titles like *Fatal Frame* (2001) and →

▲ Shuddering with every shutter click in the underrated *Fatal Frame: Maiden of Black Water*.

PHOTO FINISH

Our cover feature in Wireframe #49, *Martha is Dead* is an upcoming period thriller that uses photography to heighten the game's sense of realism: with your trusty box camera in hand, you can adjust the focus and aperture size, then you can head into your basement dark-room to process the film and develop your photos on paper. For LKA, the camera is a means of pulling the player more deeply into the game, as they photograph clues and record the world around them. "There are many elements that are important to the gameplay, but the camera is the most important one," creative director Luca Dalcò told us. "The idea is to steer away from these old concepts of the walking simulator – there's different stuff to do. I personally love walking simulators, but I also feel that the traditional walking simulators have said what they have to say, so one of the challenges is to deliver a strong narrative."

"WHEN YOU TRAVEL AND VISIT FOREIGN CULTURES, THERE'S ALWAYS THAT TENSION BETWEEN BEING PART OF A MOMENT OR RECORDING IT"

Beyond Good & Evil (2003), meanwhile, made clever use of the camera as a game mechanic for busting ghosts and uncovering conspiracies.

One of the most innovative photography-themed games of the past decade was, regrettably, never released at all. In 2011, Australian developer Defiant sought to explore the way photojournalism could change the course of history, as it had in the Vietnam War. Called *Warco*, the game would have cast the player as a photojournalist dropped into the middle of a violent conflict in North Africa, armed with only a camera and the hope of making a



▲ Uncovering conspiracies in *Beyond Good & Evil* (above) and capturing memories in *Life is Strange* (below).

difference. While games about viewing the world down the sights of a firearm have long been ubiquitous, *Warco* offered a different mission: to document the human face of war, and uncover the true cost of conflict rather than simply gunning down hordes of identical enemies.

Warco never emerged from the development dark-room, but its concept was still a noteworthy step in the in-game camera's evolution. Other games emerged with their own unique takes on the in-game camera, though, including 2015's *Life is Strange* and *Firewatch*, released a year later. Each made the camera essential to understanding – or appreciating – their respective worlds.

Five years on, and a newer crop of indie studios are working hard at evolving the camera within their narrative-led titles, each of them seeking, in one way or another, to prompt players to examine their worlds more closely, while also giving them the freedom to use the camera to capture the world as they see it. One recent example is *Alba: A Wildlife Adventure*, a charming eco-fable we covered back in issue 47. About a little girl attempting to save her Mediterranean island from being paved over



▼ *Warco* was designed with a noble aim: to teach photojournalists how to prepare for a real war zone encounter.



by property developers, it offers a free-roaming take on *Pokémon Snap*: you can go wherever you like, capturing photos of the local wildlife on your phone, while the results are logged in an app. Photography's only one aspect of *Alba*, but it's a pivotal one: it both allows the player to explore and better appreciate the game's carefully animated creatures, and also helps further the plot. As designer David Fernández Huerta told us, "Grandad shares his hobbies of birdwatching and hiking with [Alba, the game's heroine]. It's a way for them to be close to a person they have limited access to, but that they love with all their hearts."

In *Alba*, photography forms one of the pillars of a deeply personal open-world adventure, and Scavengers Studio's upcoming *Season* uses camera mechanics to a similarly heartfelt end. A gorgeous-looking indie adventure game set in the face of a mysterious cataclysm, *Season* is a bicycle road trip where you use your camera and other media to document your surroundings before the oncoming catastrophe washes the world away. It's a bitter-sweet premise that puts choice at the heart of the player's journey, and extends *Firewatch*'s camera mechanic, which



▲ *Pokémon Snap* was an on-rails photographic safari that depicted Pokémon in their natural habitats.



only gave you a finite number of shots via a disposable camera. In that game, players had to think carefully about which of the beautiful vistas to capture as their shots were limited. In *Season*, the player faces some difficult decisions about which aspects of the world merit preserving on film and which will be lost to time.

"Your goal in *Season* is to bear witness to a world that is about to disappear, in order to record and try to transmit what's important to future generations," the team at Scavengers Studio tell us. "As such, we wanted the camera to play a central role in the experience. When you travel and visit foreign cultures, there's always that tension between being part of a moment or recording it. When you pull out a camera, you're not into the moment anymore; you have already extracted yourself from experience to record it. In building the game, we're trying to capture that tension: 'Should I enjoy the moment, or should I record it?'"

In our Instagram era, this is undoubtedly a relevant subject, and one thrown into relief by the game's pre-cataclysm setting, where an air of finality pervades every decision. For every picture you choose to take, you're forsaking everything beyond the frame of the image to memory. As the team tell us: "Capturing an image fixes it in place. It's a way to bypass the erosion of time, at least for a little while. It's also a prosthetic of memory. Photos we've taken sometimes seem to replace our actual memories of an event. The underlying forces and themes at work there are what *Season* is all about."

A perfect counterpart to *Season* is *Forever Ago*. Beautifully crafted, it's another road trip →

▲ *Firewatch* offers some of the most breathtaking views a game can offer, all ready to snap on your camera. Your photos play over the end credits.

PICTURE PERFECT

We asked each of the dev teams which in-game camera mechanics have inspired them. Here's what they had to say:

Scavengers Studio:

"The camera in *Zelda: Breath of the Wild* is interesting because you can play with the generative aspect of the game to create and capture fun and unique compositions."

Something We Made:

"I like *Pokémon Snap*, the Nintendo 64 one. It was a cool thing, and they really stepped it up in the next one where you can share your photos."

Third Shift:

"I really like the phone camera in *GTA V*. It was the first game where you take selfies and I think that was an interesting commentary on the state of photography in general."

DOG'S BOLEX

If you want to satisfy your inner photographer, then here are three (relatively) recent titles with great in-game cameras...



Life is Strange (2015)

The Polaroid instant camera in DONTNOD's Oregon-based adventure perfectly complemented the rewind time function, providing an artistic and creative flourish that underscored the game's theme about the passing of time.



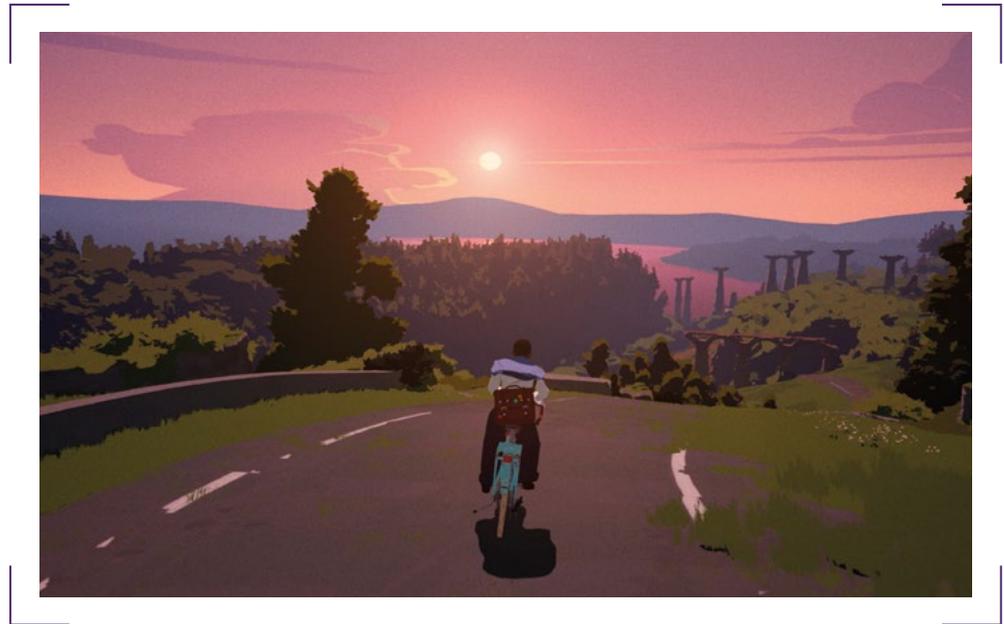
Firewatch (2016)

It wasn't integral to the core game, but *Firewatch* gave you a disposable camera, which meant there were only so many shots of beautiful vistas you could take. Players could even send their in-game photos away to be developed into physical prints.



Dead Rising (2006)

Perhaps not one of the best, but certainly one of the most fun. As photojournalist Frank West, getting dropped into a game version of 1978's *Dawn of the Dead* was terrifying, wacky, and fertile ground for some great photo opportunities.



^ Snapping birds in *Alba: A Wildlife Adventure* (above) and capturing a dying world in *Season* (top).

adventure game with a camera mechanic at its heart. Created by the two-man team at Third Shift, *Forever Ago* adopts a subtly different narrative perspective from *Season*. Where *Season* tasks players with documenting the world before a catastrophe occurs, in *Forever Ago*, the tragedy's already happened. Its hero's journey is one of redemption, with his trusty camera along for the ride. "We want to tell a story that took place a couple of decades ago, and we want to visit these places in *Forever Ago* – this was the core idea of the game," says developer Fabian Denter. "We thought a lot about how we could tie the past and the present together. The camera came as a way to do that pretty late in the process, but it turned out really well."

"It was a late but obvious choice," adds Kai Brueckers, the other half of Third Shift. "Before that, we thought about a lot of other mechanics such as the 'ghost people' that a lot of walking simulators use, but then we really thought

about how to capture the topic of memories. Photographs are a good tool to dive into [memories], relive them, revisit places where you took them. We also thought about flashbacks [and] voice-overs, but none of them had that core gameplay mechanic, like a camera does."

The camera offered an ideal means for the developers of both *Season* and *Forever Ago* to tell stories in a way other game mechanics wouldn't allow. For the developers behind *TOEM*, a hand-drawn indie title in development at Swedish studio Something We Made, an in-game camera became a unique means of letting players appreciate the game's captivating visuals.

"We always had the art for *TOEM*, that's how it started," says CEO Niklas Mikkelsen. "We had this really cool art and we [weren't sure how to proceed]. It was cool to walk around the game world, but we didn't know what to do with it. It took four different versions of *TOEM* to arrive at the camera mechanic. It came from a telescope mechanic that we turned into a camera. It allowed you to really appreciate the art which has always been at the forefront of the game."

More than anything, all three development teams we spoke with used a camera mechanic to give players the freedom to interact with the games' worlds on their own terms. "The camera's an incredible tool," adds the dev team at Scavengers Studio, "because a lot of information and knowledge couldn't be captured through words in *Season*; they're captured in the walls, the sculptures, the animals, etc. Every single detail tells you something new about that world.

What's especially interesting about the camera is that it allows you to zoom in on those details, record them, and put them side by side with other recordings to create your impression of the world."

"There are a lot of things you can do with the camera as a mechanic in terms of just capturing specific objects or items," says *Forever Ago's* Brueckers. "Or you can try and play with the viewfinder to reveal something or unlock additional narrative. That's something we're playing with."

Although their games are very different, the developers we spoke to agree that giving the player free rein with the camera allows them to form a meaningful relationship with that world. This is especially true when a game's story is tightly scripted, since the freedom to explore and interpret the world through a lens balances out the more restrictive elements of an authored narrative. "The story [in *Forever Ago*] is pretty linear," continues Brueckers. "There are some story beats we have to get right, and that includes some photos that we intend the player to take. But we allow the players to immerse themselves in the world by taking pictures at almost any time. They can review them

inside Alfred's [the game's protagonist] photo collection, and that's what makes the camera interesting. Those players can shape how they experience the story by capturing parts of it that are important to them."

It's a sentiment echoed by Mikkelsen, who also reminds us that perhaps the greatest advantage of the in-game camera is its universality. After all, the still image is a form of visual language just about all of us can immediately understand. "We have a goal, and that's to get our mums to play TOEM, understand it, and say 'That's so cute' – and they've never played a game in their life! So right now, it's zoom, snap, and you also have a honk button so you can wake people up or hurry up snails, which is a far more important upgrade for our camera."

Whether an in-game camera includes a 'honk' function or not, giving the player the means to form their own interpretation of a virtual world is undeniably powerful. It is, after all, a form of expression we understood before we even developed the power of words. As Scavengers Studio puts it: "What you choose to record or not record, and what you decide to save, is based on what you think is important. What you believe matters in this life." 



^ The line art style is integral to *TOEM's* atmosphere – and makes for great photographs.

▼ In *Forever Ago*, Alfred's camera is clipped to his bag, meaning it's always ready for a quick photo.

"WE ALLOW
PLAYERS TO
IMMERSE
THEMSELVES
IN THE WORLD
BY TAKING
PICTURES"



The struggle to simplify sports games



KIM JUSTICE

Kim Justice is a YouTuber, streamer, and writer who specialises in the world of retrogaming. If she isn't making lengthy documentary videos about old games and companies, she's probably chatting and mouthing off about them live to a dedicated handful of people.

Few games are quite as maligned critically as current sports titles, especially when compared to how many units they usually shift. The list of problems these titles have is considerable, even compared to a decade ago when the main issue was that constant annual updates made any version of a game that wasn't the latest one feel thoroughly disposable. Now, these bloated monsters are often filled with bugs. The single-player has been neglected for years in favour of online multiplayer, and the promotion of loot crates and ultimate teams make the games feel, at best, like a blatant cash grab, and at worst, a way to make children addicted to gambling. It's little wonder that everyone who streams the latest *FIFA* seems to be perpetually angry – always on the verge of smashing their controller in a frothing rage.

It's also no surprise that there are often loud voices who pine for a return to something simpler, a game that harks back to the likes of *Sensible Soccer* or the 16-bit *NHL* titles, something that's more about gameplay than ensuring that Gareth Bale's man bun is accurate, and certainly

"It's little wonder everyone who streams *FIFA* seems to be perpetually angry"

not something that's only packed in with a new game à la EA's recent *NHL 94 Rewind*. A game that transcends the season it was released in and is mercifully bereft of squad building and spending piles of cash in the hope of packing highly rated players. Is that too much for people to ask? Why can't someone just recreate the ideas that worked so well before? A lot of people seem to shout for a game like this whenever the latest sports game-based controversy rears its ugly head, and yet even the original creators of such games have found making them – and

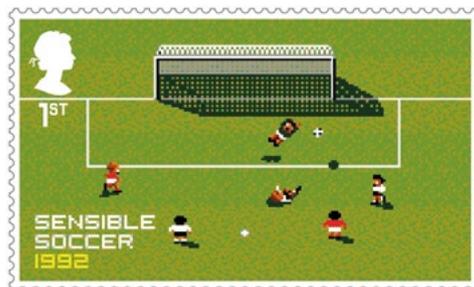
getting people to play them – tough going.

The truth is that there are a lot of simpler sports games out there, but even the best of

them don't seem to find an audience beyond a small niche. This is true even of the genre's most decorated developers – *Sociable Soccer*, the spiritual follow-up to *Sensi* from original creator Jon Hare, hasn't had the smoothest of rides over the years, and that's nothing compared to the brutal treatment that *Dino Dini's Kick Off Revival* received on PS4. It seems sports games have an image problem that runs even deeper than the problems that plague the current big titles, and it's tough to separate the genre's core audience from the annual roster updates put out by EA Sports with something that eschews graphics and realism in favour of simple, addictive gameplay.

It's a shame, as the best of these new games deserve a lot more than to be sullied with the age-old generic dismissals that usually greet sports games.

The likes of *Basketball Classics* and *Mutant Football League* are exceptional, and they make for experiences just as fun as anything from back in the day, and it'd be even better if there were people around to play with. 🗨️



▲ *Sensible Soccer* is so unmaligned, it received an official Royal Mail stamp.

sinclair



Commodo

the

COMPUTERS

THAT MADE

BRITAIN



OUT NOW

"The Computers That Made Britain is one of the best things I've read this year. It's an incredible story of eccentrics and oddballs, geniuses and madmen, and one that will have you pining for a future that could have been. It's utterly astonishing!"

- **Stuart Turton**, bestselling author and journalist

.....



Buy online: wfmag.cc/ctmb



The rise, fall, & rise of the **3D platformer**

Inspired by this month's cover game, we take a look back over the 3D platformer's glory days, underrated entries, and recent return to the fore

WRITTEN BY RYAN LAMBIE & IAN DRANSFIELD

From the middle of the 1990s to the start of the 21st century, the 3D platformer was on the ascendency. Adding more freedom of movement than the side-scrolling ancestors of previous hardware generations, the genre exploded thanks to the likes of *Super Mario 64* and *Crash Bandicoot*. But by the 2000s, audiences had apparently begun to tire of bouncing around colourful environs, collecting trinkets; the FPS was on the up, and even Nintendo began to shy away from the 3D platformer – a genre it had a hand in popularising in the first place.

Over the past few years, though, the 3D platformer has begun to sneak back into the spotlight: Playtonic's *Yooka-Laylee* is a conscious throwback to *Banjo-Kazooie*, designed by former Rare devs who worked on that series. Meanwhile, *Super Mario Odyssey* saw Nintendo openly channel the spirit of *Super Mario 64* in spectacular style. Enough time has passed to make gamers yearn for

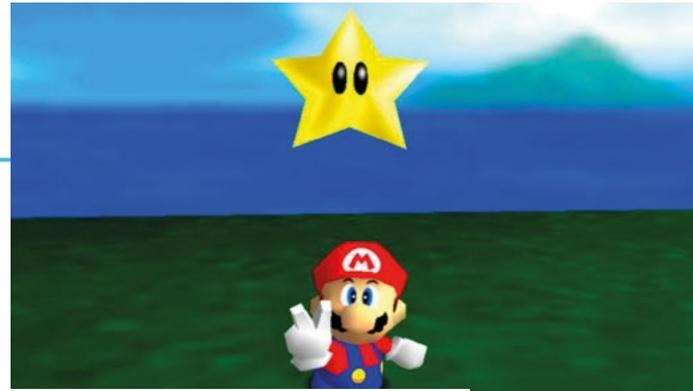
“The genre exploded thanks to the likes of *Super Mario 64* and *Crash Bandicoot*”

those simpler, late 1990s days, certainly, but according to Fabian Rastorfer, developer of our cover game *Demon Turf* (see page 6), there's more to the revival than pure nostalgia. “It needed some time to re-evaluate itself as a genre,” he tells us. “It's like how *Borderlands* said, ‘What if we combine an FPS with an RPG?’ And that kickstarted the whole thing of RPG-ifying almost every genre.”

What's more, games like *Super Mario Odyssey* and *Demon Turf* are just the tip of the iceberg when it comes to the genre's revival. According to Rastorfer, there are several indie studios working on their own platformers, all combining the staples we'd expect with modern ideas. “I can list five or six [forthcoming] games that look amazing – all 3D platformers, but they look entirely different,” he says. “I think that's why the genre's becoming popular again, because it has the potential to be awesome.”

To illustrate the rise, fall, and rise of the 3D platformer, here's a brief history of the genre, as seen through a selection of its most prominent, noteworthy titles.

The Rise



Super Mario 64 (1996)

In mainstream terms, it's the game that changed pretty much everything. Here, Shigeru Miyamoto and his team didn't just transplant what had been a resolutely 2D franchise into three dimensions – they completely changed the boundaries of what a *Super Mario* game could be. There was the freedom of exploration, certainly, but there was also Mario's fluid freedom of movement: jumps and dashes were joined by cartwheel flips, ground pounds, crawls, punches, and more besides. The move to 3D wasn't entirely flawless – the camera system, while innovative, took a bit of getting used to – but *Super Mario 64* was nevertheless a pivotal game in a nascent genre.



Crash Bandicoot (1996)

It hasn't aged as well as *Super Mario 64*, but *Crash Bandicoot* was nevertheless a key 3D platformer. First, it gave the then-new PSone a mascot of its own, endearing Sony's newcomer to a younger generation of players. Second, it was a pivotal smash hit for Naughty Dog, years before *Uncharted* and *The Last of Us*. And third, there's some genuine technical innovation going on under that gaudy hood: as studio co-founder Andy Gavin related to *Ars Technica*, getting long, flowing 3D levels with varied textures was a new thing in 1996, and required some ingenious programming and problem-solving to pull off. But beyond all that, *Crash Bandicoot* is also plain, old-fashioned fun: a chunky obstacle course of pitfalls, fast-moving hazards, and juicy items to grab.

Banjo-Kazooie (1998)

The British masters of upbeat action games, Rare, threw their hat in the 3D platformer ring with this charming entry. Like *Super Mario 64* and *Crash Bandicoot*, the emphasis was firmly on charging around and collecting stuff, but its ingenious level design, technical brilliance (this was a far more detailed-looking game than those earlier rivals), and Rare's warped humour made it a true standout. It's a glimpse of the studio at its creative, comic height.



Spyro the Dragon (1998)

Insomniac entered the 3D platformer game soon after the studio's formation, with the original *Spyro* marking the team's second-ever release. And what a game it was – a creative, atmospheric, and truly charming take on the formula, focusing on a fine blend of challenging platforming, varied exploration, and whimsical characters. Followed by two superior sequels, *Spyro* was a series that flew somewhat under the radar, and only received the plaudits it deserved following the *Spyro Reignited Trilogy* remaster in 2018.



Sonic Adventure (1998)

Fans were demanding a true 3D Sonic game for ages, but we had to wait until the Dreamcast era before we finally got one. Two points worth noting: *Sonic Adventure* hasn't aged too well, and it led directly to some of the most egregious examples of the genre over the following decade-plus. But there's no denying this was a spectacle when it first appeared; a brand new take on Sonic, a system-seller, and a glimpse of a bright future for both Sega and its blue mascot. If it had ended right there... well.



Rayman 2: The Great Escape (1999)

Rayman gets a bum rap, honestly – while *Rayman Legends* is one of the best 2D platformers ever made, the series tends to be overlooked as a whole. This is especially true of *Rayman 2: The Great Escape* – one of the absolute best 3D platformers, but one that's rarely mentioned in the same breath as other genre greats like *Mario's* three-dimensional treats. Superb character designs, genuine laugh-out-loud moments, and captivating minute-to-minute play alongside the genuine desire to go back and replay things just to get every single lum you could – *Rayman 2* is an overlooked classic.

Jak and Daxter: The Precursor Legacy (2001)

With *Crash* passing through the hands of different developers, creator Naughty Dog was left to its own devices for its first 3D platformer on PS2 – and what a rebirth it was. *Jak and Daxter* nailed so much: a characterful, bright, and cheery take that flew in the face of where gaming as a whole was heading. The formula for the genre was well-established at this point, so the core concept might have looked overly familiar to some, but it's arguable that *Precursor Legacy* is one of the best examples of a 3D platformer to date.



Conker's Bad Fur Day (2001)

The naughtiest outing on N64 pitted the foul-mouthed squirrel against challenges such as a hangover and a singing pile of poo. That the game looked out of place on a Nintendo platform and arrived so late in the console's life meant that it was missed by many the first time around – subsequent re-releases on Xbox formats have given *Conker's Bad Fur Day* the broader audience it deserved. At the same time, *Conker* worked as a portent for the 3D platformer: the old, cutesy style was on its way out, the future was in 'mature' games, and these two concepts didn't necessarily mix very well.



The Fall



Super Mario Sunshine (2002)

Even *Mario* wasn't immune from the power of 'meh', with this GameCube entry still the most divisive in a series of games that, by and large, are unquestioningly praised to the hilt. It's hard to put a finger on what was really *wrong* with *Sunshine* – the emphasis on gimmicky extras, the feeling it was (shockingly for Nintendo) a bit rushed, all contributed to a general unease with the game. But, all the same, a slightly worse *Mario* game is still significantly better than most other 3D platformers. It just seems this one didn't quite have the audience for it at the time.

Jak II (2003)

Probably the most egregious example of what happened to the 3D platformer, *Jak II* – it's fair to say – was not a bad game. It was a good game. Some might say great. But what Naughty Dog did after the prior entry to the series – the whimsical, light-hearted, fun, and pure platformer – was to turn it into *GTA III*. No, but really: *Jak Theft Auto* would have been a more fitting name. It retained platforming elements, of course, but far more focus went into the 'flavour of the month' style of play: driving vehicles, taking on missions in an open world, and gunning down hordes of enemies. This should have been a game, definitely, but it shouldn't have been a *Jak and Daxter* game.



I-Ninja (2003)

It's not that there weren't good 3D platformers during the fall period for the genre; it's more the interest of players was elsewhere, and plenty of titles were hugely overlooked as a result. *I-Ninja*, from Argonaut Games, was just one of those games. It had all the hallmarks of a quick cash-grab, it's fair to say – super-cutesy main character, reference to ninjas, the sometimes-unreliable Namco on publishing duties – but it would be doing the game a huge disservice to say it was anything other than: really good fun. A classic? Not at all. A rush job hacked together and contributing to the downfall of an entire genre? Well, no, not that either.



Sly 2: Band of Thieves (2004)

Sidestepping the original, Sucker Punch's mix of noir, thievery, and raccoons was easily the best in this overlooked foursome. A visual treat from start to finish, the stylish sneaking and mixture of stealth and classic 3D leaping worked together in perfect harmony, resulting in yet another must-play title in the genre on PS2. After *Sly*, Sucker Punch moved on to *Infamous*, arguably another blend of *GTA* and 3D platformers (but with more superheroes) and the universe of criminal creatures looks to be finished.



Kameo: Elements of Power (2005)

We wouldn't argue if you were to tell us Rare's opening salvo on the Xbox 360 was indeed *not* a 3D platformer. It's more an action-adventure game than anything else, just with elements of gap-jumping thrown in every now and then.



But it has to go in here, because it highlights a huge turning point for the studio once synonymous with the best the genre had to offer: Rare had been gobbled up by Microsoft, and now it was pumping out mediocrity like *Kameo*. A fair effort, a bit of fun, with some good ideas in it – but not the stone-cold classic we were used to, that we *expected* from one of the UK's finest dev teams.



Psychonauts (2005)

Psychonauts was meant to be the next big hope for the 3D platformer: a world of fantasy and wild imagination, humour, which carefully trod the line between classic, tight platform mechanics, and a more modern *feel* to things. It hit every note, it really did, but *Psychonauts* was an abject failure. Selling around 100,000 copies on its first release, it nowhere near hit the lofty goals envisioned for it. *Psychonauts'* struggle helped – very publicly – send the genre into a tailspin. It's all the more shocking when you remember how good a game it was.

Super Mario Galaxy (2007)

Both this game and its sequel arguably represented the pinnacle of the 3D platformer to date: their mix of bite-sized, spherical levels, sense of experimentation, and sheer imagination made them



truly world-class *Super Mario* games. The first game did well, but with sales of around six million, *Galaxy 2* was a muted success compared with the 2.5D *New Super Mario Bros.* on the DS and *New Super Mario Bros. Wii* (both circa 30 million). Tellingly, Nintendo didn't attempt to make a *Super Mario* game as expansive and ambitious as the *Galaxy* titles for another seven years.

Mirror's Edge (2008)

Debatable if it is actually a 3D platformer in the purist's sense, we include it here more as an example of experimentation with the genre, and to give an idea of what was going on in this period. DICE was keen to play about with something other than



Battlefield and put together an elegant mix of first-person shooter (without much in the way of gunplay) and parkour. It worked as a game, but it didn't really find an audience – the first-person 3D platformer renaissance is one we're still waiting on to this day.

The Return



Ratchet & Clank: A Crack in Time (2009)

Seven years into the series, *Ratchet & Clank* wasn't going to win any awards for its vanilla platforming. Meanwhile, the emphasis on gunplay was still fun, but it no longer felt fresh. What *A Crack in Time* did, though, was bring in outside inspiration from the time where the 3D platformer had been sat on the sidelines – it mixed in the classic playstyle with time-based puzzles reminiscent of indie darling *Braid*. Throwing this new element into the long-running series served to mix things up wonderfully, resulting in what is arguably the series' best release – somewhat ironically because of its *lesser* emphasis on shooting and platforming.

Super Mario Odyssey (2017)

Nintendo's triumphant return to the 3D platformer, and a game that both harks back to *Super Mario 64* and improves on it in every way possible. As Playtonic co-founder Gavin Price put it to us this month, *Super Mario Odyssey* beautifully nests linear courses and challenges within larger, 3D environments, creating the perfect blend of focused and free-roaming play: "The world is really just there as a playground for your moveset," Price points out. Nintendo not only took a notoriously tricky genre and honed it to perfection – but they did it with such low-key precision, and made it look effortless.





A Hat in Time (2017)

This charming throwback to the 1990s 3D platformers listed earlier flew under the radar somewhat, perhaps because of its proximity to a certain Italian plumber. Still, this is a sparky title from indie developer Gears for Breakfast, and it's more than deserving of a second look if you haven't already sampled its hat-based delights.

Yooka-Laylee (2017)

One swallow doesn't make a summer, as the saying goes, but three 3D platformers in one year is a sure sign the genre's back on the ascendency. *Yooka-Laylee* was, of course, the debut game from Playtonic, the studio formed by several ex-Rare devs. While it isn't perfect, *Yooka-Laylee* captures the spirit of the *Banjo-Kazooie* series, both in its anarchic lead duo and the sprawl of its vibrant levels. With the series bolstered by 2.5D spin-off, *Yooka-Laylee and the Impossible Lair*, here's hoping we'll see Playtonic continue to refine the 3D platformer genre in the years to come.



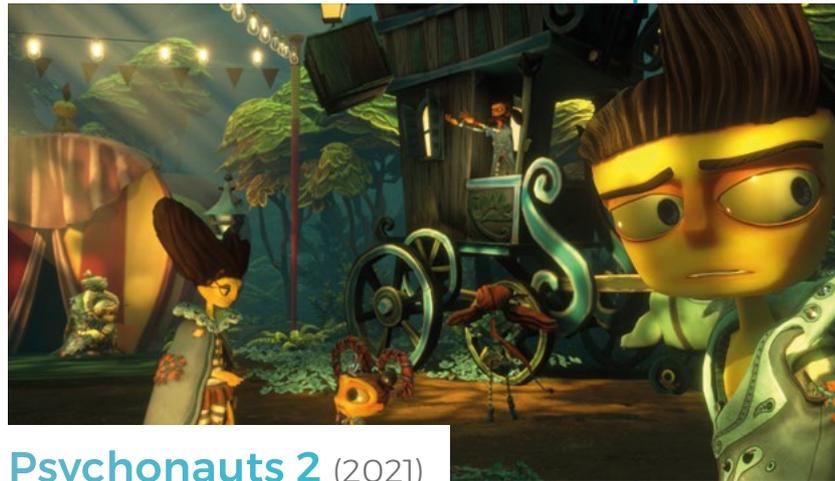
Crash Bandicoot 4: It's About Time (2020)

The remastered trilogies of both *Crash* and *Spyro* arrived in 2017 and 2018, respectively, with both acting as litmus tests for players' appetites for nostalgia, and for the viability of 3D platformers in the contemporary world. Turns out Activision thought the *Crash* trilogy's performance was good enough to justify a full new entry in the series: *Crash Bandicoot 4* was born. Handily, the sequel managed to please old-school fans while also adding exciting new elements to the franchise. Outside of Nintendo, Crash's fourth mainline adventure is one of the biggest – and best – steps back into the realm of 3D platformers by a major publisher.



It Takes Two (2021)

Another fine example of the modernisation of the genre, *It Takes Two* blends aspects of 3D platforming with some superb co-op mechanics and a hearty chunk of action-adventure. This push toward modernisation is extra surprising considering where it came from – EA's Originals label – with the publisher not really having very much to do with the genre outside of a couple of *TY the Tasmanian Tiger* titles. It's good, then, that *It Takes Two* is a fabulous little game, both fun to play and awash with empathy in its narrative – a perfect game to while away the hours with another player.



Psychonauts 2 (2021)

It's not the biggest game on this list, and frankly, nothing can approach *Super Mario Odyssey* in terms of quality – sorry, everything else – but there's definitely a poetry in Double Fine's *Psychonauts 2* finally releasing here in 2021. The original game was emblematic of the decline of 3D platformers. Abandoned by Microsoft and (eventually) sent out to die, it's entirely fitting that this long-awaited sequel arrives with the wholehearted backing of publisher/studio owner Microsoft, welcomed warmly into an era where appreciation of the genre is back on the rise. The phoenix, she rises. 🐦

Toolbox

The art, theory, and production of video games

44. Design Principles

The fine line between partying and productivity at Atari

46. CityCraft

Tips for designing an immersive city simulation

48. Creative Mode

How to make an engrossing block-building game in Unity

56. Narrative Design

Cinematic storytelling in Ratchet & Clank: Rift Apart

58. Homegrown Bullet Hell

A guide to making your own side-scrolling shooter

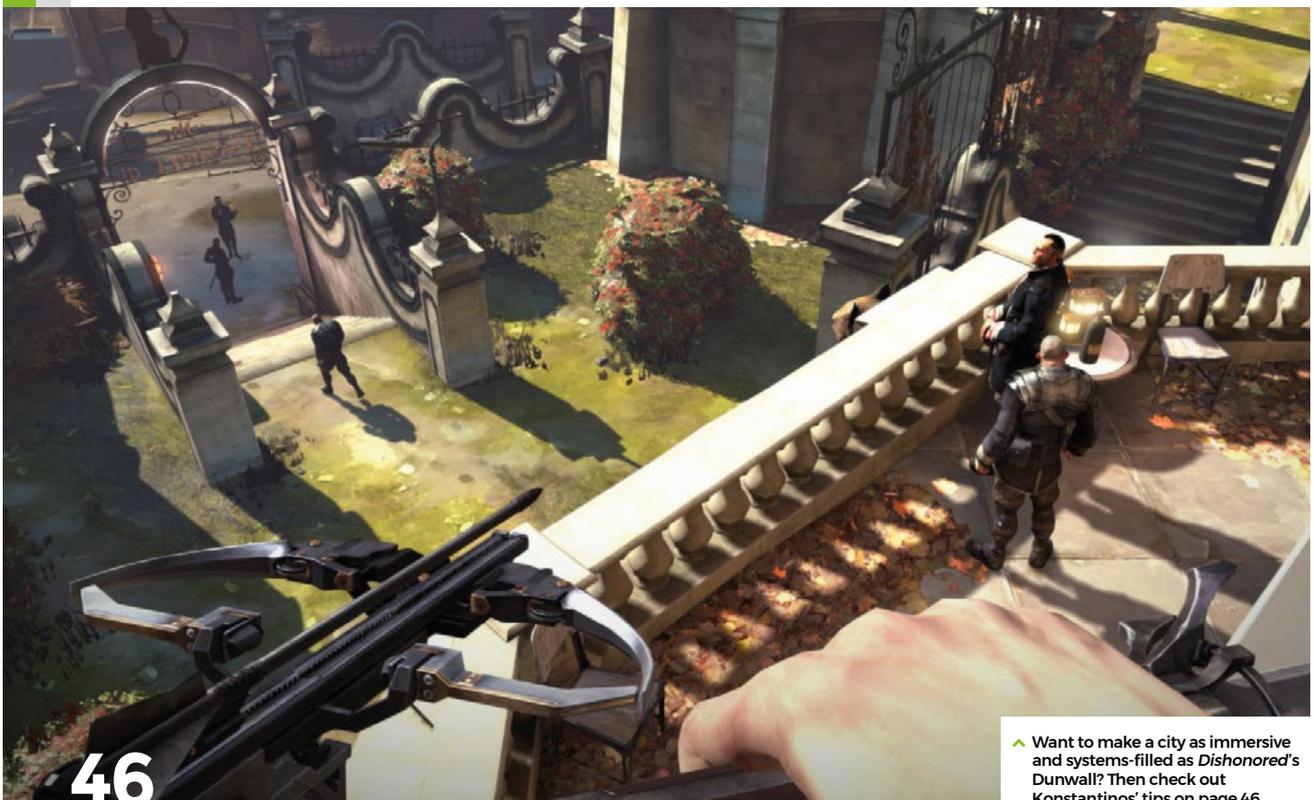
64. Source Code

Make a ball-flipping pinball game in Pygame Zero



▲ It's party time at Atari on page 44. Design legend Howard Scott Warshaw tells all.

44



46

▲ Want to make a city as immersive and systems-filled as *Dishonored's* Dunwall? Then check out Konstantinos' tips on page 46.

48



▲ Making your own creative block-building game is easy – just follow our guide on page 48.



58



56

▲ How does *Ratchet & Clank: Rift Apart* tell a cinematic story in a video game context? Antony lays it all out on page 56.

◀ Love shoot-'em-ups as much as we do? Then make one of your own in Python! See page 58.

The principles of game design

This month, Howard provides a snapshot of the hell-raising that went on behind the scenes at Atari...

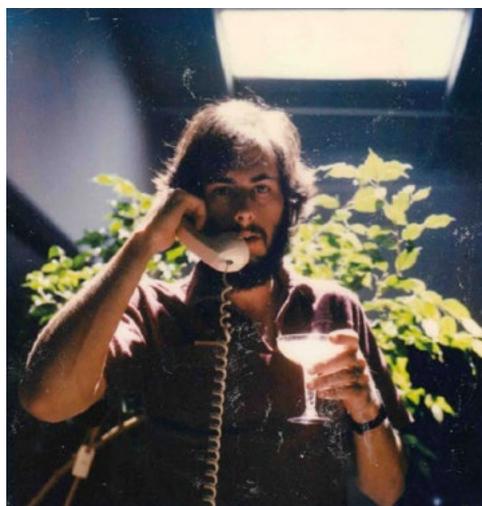


AUTHOR
HOWARD SCOTT WARSHAW

Howard is a video game pioneer who authored several of Atari's most famous and infamous titles. His new book, *Once Upon Atari: How I Made History by Killing an Industry*, is out now. onceuponatari.com

Video game creation is unusual in that developers need to be focused intently on achieving design goals while simultaneously battling tunnel vision and re-evaluating those goals.

It's a demanding and frustrating predicament. Therefore, a solid video game creator needs two things: a way to let ideas simmer (since rumination is how games grow from mediocre to fabulous) and a way to blow off steam (since frustration abounds while trying to achieve fabulous). At Atari, there was one place where things both simmered and got steamy... the hot tub. The only thing we couldn't do was keep a lid on the antics cooked up inside.



➤ Refreshments were readily available at Atari in its heyday.

The hot tub was situated in the two-storey engineering building. This was ironic, because the hot tub generated way more than two stories in that building. The VCS/2600 and Home Computer development groups were upstairs. The first floor held coin-op development, a kitchen/cafeteria, and an extremely well-appointed gym. The gym featured two appendages: a locker area and the hot tub room. Many shenanigans were hatched and/or executed in the hot tub. One from the more epic end of the spectrum comes to mind: the executive birthday surprise.

It was during the birthday celebration of a VP who shall remain nameless, but it might have been the one who used to keep a canister of nitrous oxide and another of pure oxygen in his office. The nitrous oxide was for getting high and laughing some time away, while the oxygen was used for rapid sobering up in the event a spontaneous meeting was called (which happened regularly at Atari). As the party raged on, a small crew of revellers migrated to the small but accommodating hot tub room. Various intoxicants (well beyond the scope of nitrous) were being consumed in celebration of the special event (although by this standard, nearly every day was a special event at Atari).

As the party rolled on, inhibitions were shed along with numerous articles of clothing. At one point, the birthday boy was adjudged to be in dire need of a proper tubbing as he hadn't lost sufficient layers to keep pace with the party at large. The birthday boy disagreed, and the ensuing negotiation took the form of a lively chase around



▲ A moment of relative calm in Atari's offices, circa the early 1980s. There's Howard nearest the camera on the right.

the area. The VP ran out of the hot tub room and headed for the workout area with a wet posse in hot pursuit, all in varying stages of undress.

It's important to note here that although refreshments and revelry were widely available at Atari, one item in short supply was conference rooms. Consequently, meetings could pop up in odd locales. Any place an aggregation could be achieved was a potential meeting spot. The sensitivity of the subject matter would determine the level of privacy required on a case-by-case basis. Since people weren't always working out, the gym had enough places to sit that it could serve as a decent host for gatherings. And as for sensitivity, the hot tub room was well sound-proofed, so intruding ears weren't a concern.

As the crew of rowdy revellers followed the VP into the workout area, they were confronted by just such a collection of executives who happened to be meeting at the time. I don't think the birthday party was on the agenda. However, they may have been pleased that the absentee VP had ultimately decided to join their number. It was embarrassing for some, entertaining for others, and nearly career-ending for a couple. The moral of this story being that Atari executives should never go anywhere without their oxygen tanks in tow.

But morals aside, there was work to be done at Atari. In a place where work can lead to antics and antics can lead to work breakthroughs, it's difficult at times to suss out the precise boundary

between work and antics. It takes passion and commitment to pursue side quests productively and yet remain on task when necessary.

The main reason this was a challenge comes down to the fact there are so many distractions constantly going on. Creative people tend to be creative frequently and spontaneously. Also, their creativity is much more motivated by fascination and interest than it is by task lists or project plans. Fun can break out at any moment, and answering

the call isn't always the right choice, no matter how compelling the siren.

Rob Fulop, creator of *Missile Command* and *Demon Attack* for the Atari

2600 (among many other hits) isn't only a great game maker, he's also a keen observer of human nature. We used to chat about just where the edge is between work and play at Atari. Those who misjudge it can easily fall off the cliff.

Likewise, we explored the concept of what makes a good game designer. Rob said it's just the right combination of silly and anal. He believed that the people who did well at Atari (and as game makers in general) were the people who could be silly enough to recognise fun, and anal enough to get all the minutia and details aligned correctly in order to deliver the fun. Of course, Rob (being the poet he is) created a wonderful phrasing to describe those with the right stuff. He put it like this: the people who did well at Atari were the people who could goof around as much as possible but still go to heaven. ☺

“We used to chat about where the edge is between work and play”

Insert Disc

The hot tub wasn't the only form of recreation we had at Atari, nor were video games for that matter. We would occasionally actually leave the building and do something outside. A game or two of ultimate Frisbee used to be good sport now and then. Sometimes, while sitting in the hot tub, I would look down and think, “This stomach used to be much smaller and better defined.” In fact, there was a time when I was actually sporting a solid six-pack. Of course, I don't know how to phrase that in the UK. What's the metric conversion for six-pack?

▼ Between developing games, Howard and Atari's other programmers found time to play a bit of Frisbee.



Designing cities for immersive sims

How can we design an immersive city as impressive as Dishonored's Dunwall? Konstantinos has some tips



AUTHOR
KONSTANTINOS DIMOPOULOS

Konstantinos Dimopoulos is a game urbanist and designer combining a PhD in urban planning with video games. He is the author of the *Virtual Cities* atlas, designs game cities, and consults on their creation. game-cities.com

Immersive sims – games like *Dishonored*, *System Shock*, or *Thief* – put their emphasis on player choice, and provide us with a seemingly endless array of stuff to do. Such games aim to create a sense of granularity and cohesiveness in worlds packed with overlapping systems. They attempt to spatially immerse their audiences in convincing, believable spaces able to react to any in-game action, and allow new gameplay possibilities to emerge. Warren Spector, one of the genre's fathers, famously said immersive sims create the feeling of there being nothing standing between players and the game, and make them believe they really are in an alternate world.

PALPABLE ENVIRONMENTS

Ultima Underworld: The Stygian Abyss was a groundbreaking RPG widely considered to be the first immersive sim. Both it and its sequel mostly took place in architecturally convincing dungeons, but despite the games' success, the genre quickly moved above ground, and showed its love for urban environments like the ones in the *Thief* series. Cities do, after all, handily come with built-in complexities, familiar and interdependent systems, intriguing geometries and verticalities, and are full of people. Often, city life itself requires creative solutions to complex problems.

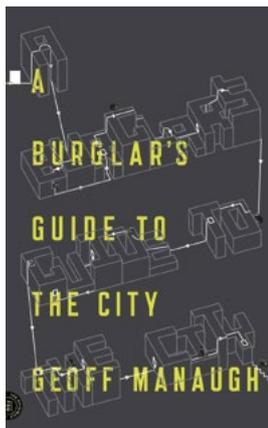
Immersive sims can cover a variety of genres, ranging from stealth games and RPGs, to platformers, shooters, and survival horror offerings. The multipurpose, interconnected spaces of cities, their networks, societal

webs, and districts can both accommodate genre needs and inspire designs. The internal differentiation of urban space can be taken advantage of to create areas and missions with specific focuses, whereas the density and complexity compressed into relatively small areas allows for all kinds of storytelling and gameplay ideas in the same area. Game cities additionally offer countless opportunities for side missions or secondary objectives, and allow players to use familiar spaces in unfamiliar ways.

Immersive sim spaces need to feel textured and tangible, and so attention to detail is crucial. Such games must convey a strong sense of in-world realism, if only to suspend players' disbelief, and allow them to understand how the virtual world surrounding them works. They force players to pay attention to all the qualities of their environment and learn to take advantage of its functions, as even the slightest detail can determine the outcome of a mission. In *Thief: The Dark Project*, for example, light and sound must always be taken into account if players are to remain hidden, whereas walking on stone rather than on a carpet can tip a guard off.

Architectural qualities also play an important role in *Thief*. The combination of building volumes and torch placement creates everything from safe hiding spaces to completely exposed positions. Even the urban designs around core levels force players to carefully traverse ill-lit roads while looking for accessible and hopefully unguarded entry points, avoiding patrols or entertaining the option of entering via handily placed wells and sewer gates. Similarly, the

▼ *A Burglar's Guide to the City*: an essential read for budding sim designers.





▲ *Ultima Underworld* (1992) was the game whose cohesive spaces, multiple play style options, and impressive physics gave birth to the immersive sim.

urbanism of *Dishonored* ensured Dunwall came with well-placed vantage points, accessible balconies, sewage systems, bridges, yards, and roofs allowing creative players to discover several access points and entry paths to their target buildings. Back-alleys, utility connections, maintenance infrastructure, and small service doors often provide an overlooked but important set of level details to any game employing acrobatics, stealth, and breaking-and-entering.

DESIGN CONSIDERATIONS

The game design philosophy guiding immersive sims is focused on the interactions with reactive and consistent game systems that can allow for emergent gameplay to happen, and heighten the sense of player agency. Consistency and the legibility of space should therefore guide all relevant level design choices. Systems and rules must be stuck to wherever possible, and though urbanism and environmental storytelling shouldn't get in the way of the game itself, they have to be treated with care. Players will thoroughly examine our worlds, so they must stand up to scrutiny. This is exactly why immersive sims tend to come with fully realised settings featuring religions, societies, fashions, technologies, economies, and histories. Dunwall's whale-oil economy and its related technologies, for example, drove much of the world-building, but also much of the level and game design.

A video game city, with all its NPCs, multiple paths, and density of detail, can often mean that a purely open-world approach is all but impossible

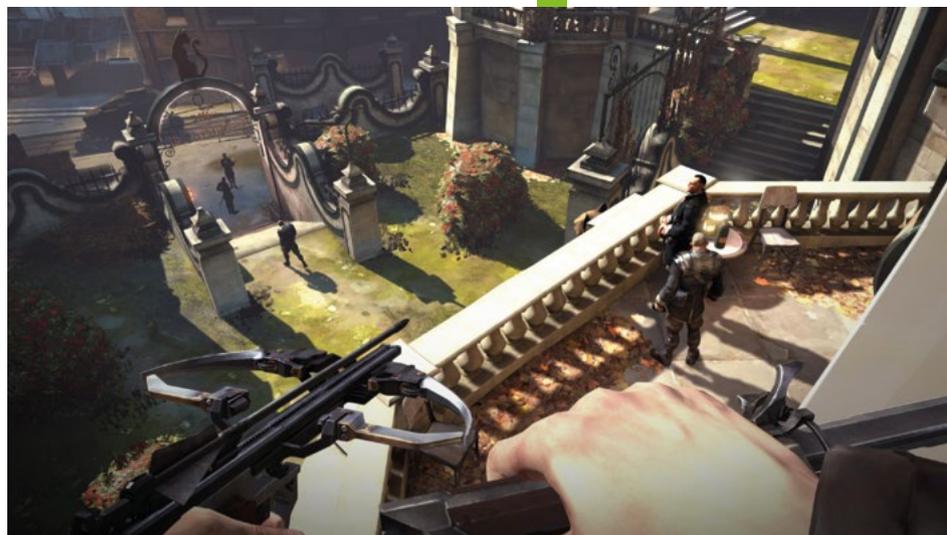
“Game cities offer countless opportunities for side missions”

– unless we create it in a procedural or semi-procedural way, that is. Quality, handcrafted levels can effectively form smaller parts of much larger cities: a road, a neighbourhood, a building, a block and its environs, a palace, or a park. Levels are the nodes in an implied urban tissue, often connected to a hub-like structure. Travelling to and from a hub can help imply the size of a larger city, as can maps, dialogue references, well-selected views, in-game paintings or descriptions, and even obvious suggestions of repeating urban planning patterns.

The opening level of the original *Deus Ex* is an excellent example of immersive sim level design. It's a legible, tiny bit of New York City where the possibilities felt vast, and the lights of the Manhattan skyline provided all the civic atmosphere we could ask for. The level itself offers

clear gameplay alternatives when it comes to going down the stealthy, hacking, or shooting paths, and is surprisingly easy to navigate, as your goal (the Statue of Liberty) is a constantly visible landmark, the boundaries absolute, and its distinct areas highly recognisable.

Finally, when dreaming up ways urban systems and networks can be incorporated into your level design, just think about how something as simple as a power cut could create a dramatic mission: the player could be asked to cut off the electricity to a particular district in order to silence alarms and turn off CCTV cameras. With a bit of imagination, the design possibilities of a video game city are almost endless. 🌐



▲ The architectural choices that shaped the striking image of *Dishonored*'s city of Dunwall were also crucial in providing gameplay options to players.

A BURGLAR'S POV

Published in 2016, Geoff Manaugh's *A Burglar's Guide to the City* (burglarsguide.com) isn't just an interesting, fun read about criminals, cities, and architecture. It's also a completely novel look at the spaces and volumes of cities and their buildings; an approach to urban space that ignores formal entrances, instead focusing on unexpected entry points into private spaces, and imaginative navigations and uses of the urban fabric. Ultimately, it's a book packed with ideas and inspirations for immersive sim designers.

Craft your own Creative Mode in Unity

Dig into the design genius of Minecraft by building your own take on its Creative Mode



AUTHOR
RHETT THOMPSON

Rhett Thompson is an indie game developer by heart, and a web developer by trade. He's currently working with SentrySoftSolutions to help bring Texas community corrections facilities to the cloud. [keybored.app](#)



For many young (and not so young) minds alike, *Minecraft* is the perfect creative sandbox: an open medium that allows us to experiment, create, and play. For all these reasons, it's the perfect example to recreate as a learning tool.

Before we jump into coding, let's first get the project setup out of the way. The code itself should run on most versions of Unity, but I'm using Unity 2019.4.10 LTS. If you're unfamiliar with how to install Unity, you could head to Wireframe #3 and read the Unity tutorial on page 34 ([wfmag.cc/3](#)). The only other tools

we'll need are a 3D modelling program and a 2D image editor. You can use whatever you like, but I've used the newest versions of Blender and GIMP. They're both free, work well, and the installation (from [blender.org](#) and [gimp.org](#)) is straightforward.

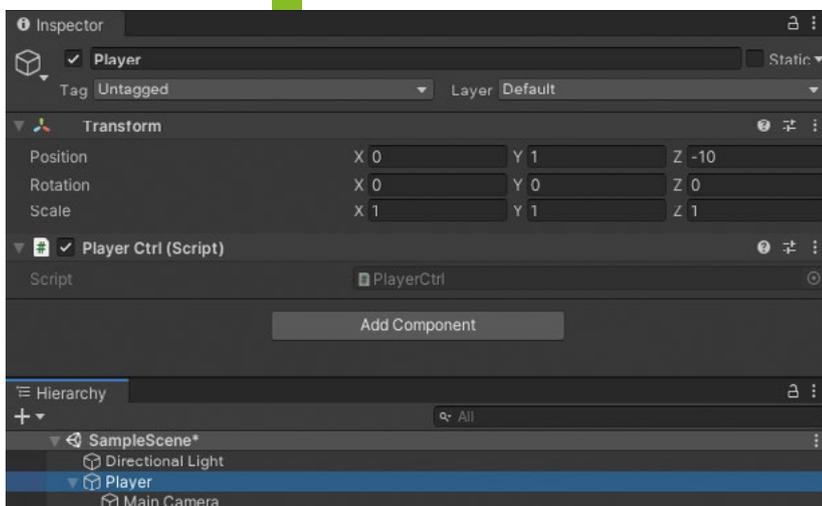
MOVEMENT AND CONTROLS

With the setup out of the way, make a new Unity project with the 3D template so we can get started. First, let's work on creating the player GameObject. Inside the default game scene, right-click in the Hierarchy and select Create Empty. Rename that object from GameObject to Player. In the Inspector panel, change its Transform Position to X: 0, Y: 1, Z: -10. Drag and drop the Main Camera so it's a child of Player. Then, right-click inside the Project panel and select Create > C# Script. Name it PlayerCtrl. Drag it onto the Player GameObject. Your Hierarchy should now look like **Figure 1**. Next, open the PlayerCtrl script in your favourite code editor and add the following:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

```
public class PlayerCtrl : MonoBehaviour {
    public float flySpeed = 10f;
    public float rotateSpeed = 2f;
    Camera cam;
```

▼ **Figure 1:** If your Player object looks like this, you're good to go!





```

Vector3 camRot = new Vector3(0, 0, 0);

void Start() {
    Cursor.lockState = CursorLockMode.Locked;
    Cursor.visible = false;
    cam = Camera.main;
}

void Update() {
    float mouseX = Input.GetAxis("Mouse X");
    float mouseY = Input.GetAxis("Mouse Y");

    transform.RotateAround(transform.position,
        Vector3.up, mouseX * rotateSpeed);

    camRot.x += -mouseY * rotateSpeed;
    camRot.x = Mathf.Clamp(camRot.x, -90, 90);
    cam.transform.localEulerAngles = camRot;

    float vertInput = Input.GetAxis("Vertical");
    float horiInput = Input.GetAxis("Horizontal");
    float jumpInput = Input.GetAxis("Jump") -
        Input.GetAxis("Fire3");

    Vector3 move = Vector3.zero;
    move.x += horiInput * flySpeed;
    move.y += jumpInput * flySpeed;
    move.z += vertInput * flySpeed;
    transform.Translate(move * Time.deltaTime);
}
}

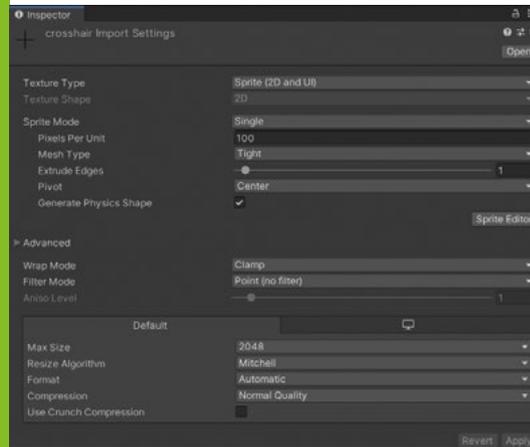
```

The code starts by defining some variables: flySpeed and rotateSpeed can be adjusted inside the Unity Inspector to suit your control preferences. cam will store a reference to the Main Camera. And camRot will save the Player's up or down rotation between frames. In the **Start()** function, lock the mouse in the middle of the screen, hide it from the user, and find the Main Camera in the game scene. In **Update()**, first, read the current mouse movement and then rotate the Player GameObject according to that movement left/right using **RotateAround()** and up/down by adding the values to camRot.x. Be sure to adjust sensitivity based on rotateSpeed, use -mouseY to prevent an inverted axis, and clamp/limit the rotation so the player can't turn all the way upside-down. Remember, cam is a child to the Player GameObject, so we apply the new rotation to its local transform. Then, read keyboard input from the **WASD**, **SPACE**, and **LEFT-SHIFT** keys through the Vertical, Horizontal, Jump, and Fire3 axes. Note, there isn't a default input axis that will read both **SPACE** and **LEFT-SHIFT**, so we combine two different axes to work together.

Next, save everything in both Unity and the code editor. Every time we make a code change, be sure to save the file. This lets Unity know it needs to recompile the script. If you press Play now, you won't see much of anything. Don't worry: the script is working fine, but we need to add a cube to the scene so we can actually tell that the Player is moving. Right-click in the ➔

^ At a glance, you might mistake this for the real deal!

► **Figure 2:** Be sure your crosshair image has these settings.



WHAT'S A MESH?

To find the normal vector that tells which direction to place a new block in, we based all the maths on the mesh of the previous block. But if you're a bit newer to game development, you might be wondering, "What's a mesh?" I'm glad you asked! A mesh is simply all of the 3D model data for a particular GameObject. In this case, that would be the cube model used for a block. You might further ask, "What are 3D models made out of?" The answer's triangles. Every 3D model can be represented as a bunch of triangles, which allows us to calculate all sorts of useful things. To find out more about 3D models and triangles, check out this video: wfmag.cc/triangles.

► **Figure 3:** Make sure your Hierarchy has this layout, and your UI image has these settings.

Hierarchy and select 3D Object > Cube. Now press Play again, and you'll find testing the movement much easier.

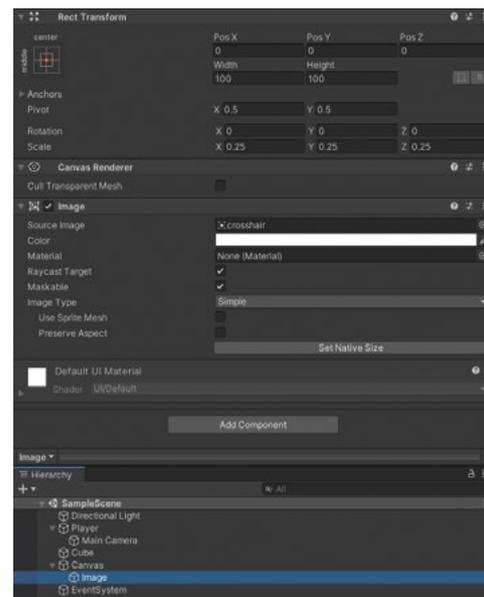
CROSSHAIR UI

What first person game is complete without a crosshair in the middle of the screen?

Let's add this now. Either make your own crosshair sprite using GIMP or download mine from the GitHub link (wfmag.cc/wfmag53).

Import the image file by dragging it from your File Explorer into the Unity Project panel. Click on the crosshair image to bring it up in the Inspector. Change the Texture Type from 'Default' to 'Sprite (2D and UI)'. Change Filter Mode from 'Bilinear' to 'Point (no filter)'. Click Apply (which is at the bottom of the Inspector panel). Confirm your Inspector settings by checking **Figure 2**.

"Minecraft wouldn't be Minecraft without blocks!"



Now, let's add the crosshair to our game scene. Right-click in the Hierarchy and select UI > Canvas. Right-click on the Canvas that was just created and select UI > Image. Click on the Image object that was just created. From the Project panel, drag the crosshair file onto the 'Source Image' property of the Image object. Make sure the Pivot property is set to X: 0.5, Y: 0.5 (this will keep the crosshair in the middle of the screen). Depending on how you made the crosshair image, you might need to adjust the Scale property to make it smaller (I keep mine set to: X: 0.25, Y: 0.25, Z: 0.25). Confirm your settings by checking **Figure 3**.

BLOCKS

Minecraft wouldn't be *Minecraft* without blocks! Let's add the building functionality now.

First, let's modify the Cube we added to the scene earlier so it's more *Minecraft-y*. Select the Cube inside of the Hierarchy. Rename it to Block. In the Inspector, right-click Box Collider and select Remove Component. Click 'Add Component' and search for 'Mesh Collider'. Click

on 'Mesh Collider' in the search results to add it to the Block.

For the texture, either create one in GIMP or use mine by downloading the

file named **wireframe.png** from GitHub. Import the texture by dragging it into the Unity Project panel. Right-click in the Project panel and select Create > Material. Name it BlockMat. Select the newly created material and in the Inspector change its Shader property from Standard to Unlit > Texture. Drag the wireframe (or your texture) image onto the property of the material that says None (Texture). Drag the new BlockMat material from the Project panel and drop it on top of the Block GameObject in the Hierarchy panel. Finally, drag the Block GameObject from the Hierarchy into the Project panel to save it as a prefab. If the Block in your game scene changed to have the image texture of your choice, you've done these steps correctly. To verify your Block settings, check out **Figure 4**. Do the same for BlockMat via **Figure 5**.

Next, make a new C# script that will control block placement. Right-click inside the Project panel and select Create > C# Script. Name it **BlockCtrl**. Drag it onto the Player GameObject inside the Hierarchy. Open the BlockCtrl script and add the following:

```
using System;
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

public class BlockCtrl : MonoBehaviour {
    public float buildReach = 10f;
    public GameObject[] blockCatalog;
    int activeID = 0;
    GameObject blockGroup;
    Camera cam;

    void Start() {
        cam = Camera.main;
        blockGroup = new GameObject("---Block
Group---");
        GenBlock(Vector3.zero);
    }

    void Update() {
        RaycastHit pointRay1;
        if (Physics.Raycast(cam.ScreenPointToRay(Input.
mousePosition), out pointRay1, buildReach)){
            Mesh hitMesh = pointRay1.collider.gameObject.
GetComponent<MeshFilter>().mesh;
            Vector3[] verts = hitMesh.vertices;
            int[] tris = hitMesh.triangles;

            int triangleStartIndex = pointRay1.
triangleIndex*3;

            int vertIndex0 = tris[triangleStartIndex];
            int vertIndex1 = tris[triangleStartIndex + 1];
            int vertIndex2 = tris[triangleStartIndex + 2];

            Vector3 point0 = verts[vertIndex0];
```

```
            Vector3 point1 = verts[vertIndex1];
            Vector3 point2 = verts[vertIndex2];

            Vector3 side1 = point1 - point0;
            Vector3 side2 = point2 - point0;
            Vector3 triNormal = Vector3.Cross(side1,
side2);
            triNormal = triNormal.normalized;

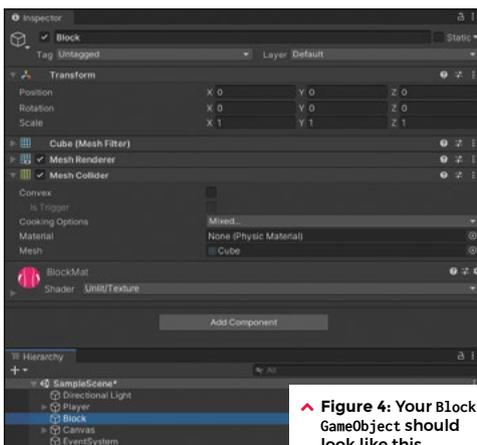
            Vector3 newBlockPos = pointRay1.transform.
position + triNormal;

            if ( Input.GetButtonDown("Fire2") ) {
                GenBlock(newBlockPos);
            }

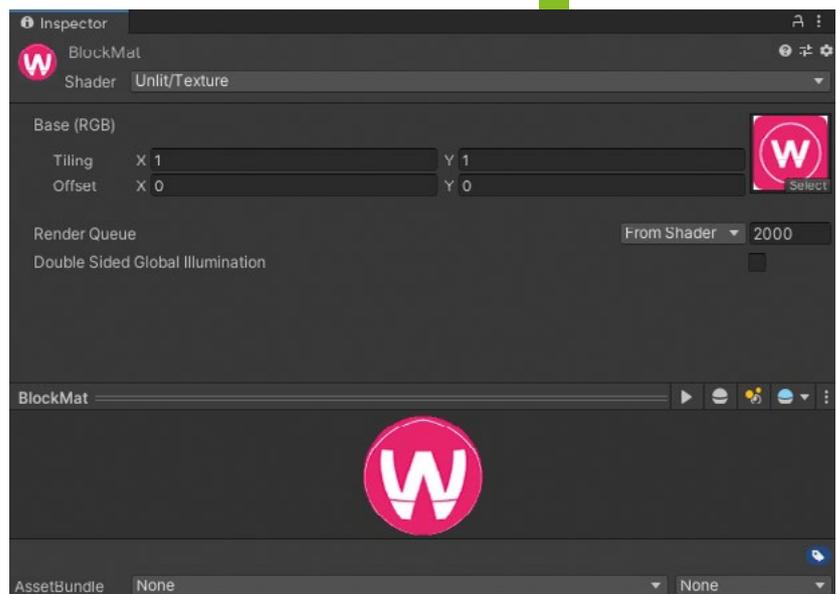
            if( Input.GetButtonDown("Fire1") ){
                Destroy(pointRay1.collider.gameObject);
            }
        }
    }

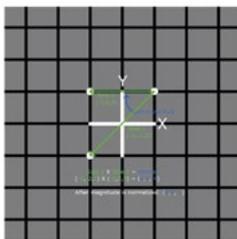
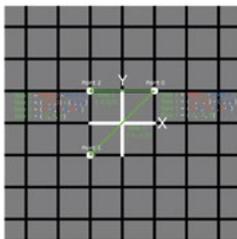
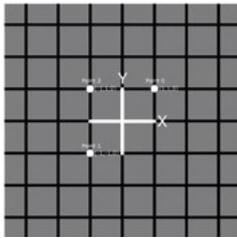
    GameObject GenBlock(Vector3 pos){
        GameObject blockObj =
Instantiate(blockCatalog[activeID], pos, Quaternion.
identity, blockGroup.transform);
        blockObj.name = "Block";
        return blockObj;
    }
}
```

✓ Figure 5: Your BlockMat material should look like this.



▲ Figure 4: Your Block GameObject should look like this.





▲ **Figure 6:** To calculate a triangle normal, first find two vectors that represent sides of the triangle. Then, take the cross product of those vectors.

This code begins by defining some variables: buildReach will determine the max distance the player can place/destroy blocks. blockCatalog is an array that will hold all of our different block prefabs. activeID controls which block prefab will be placed once the player clicks. It maps to an index in blockCatalog. blockGroup will hold an empty GameObject that will act as a parent to hold all the blocks inside of the game scene.

The **Start()** function initialises blockGroup and calls **GenBlock()** to generate a default block at the coordinate system's origin. Without this, no other blocks can be placed.

In the **Update()** function, first check if casting a ray from the mouse position, with buildReach as its length, will hit anything. If the ray did hit a block, grab a reference

from the RaycastHit data to the mesh of that block. Then copy its vertices and triangles.

RaycastHit.triangleIndex is the index of the whole triangle hit by the ray. In other words, it's not an index in any of the arrays, but rather the number that maps a mesh triangle to the tris array. Mathematically, that means the index of the first point of the triangle can be found by multiplying triangleIndex by 3. Now that we know the triangleStartIndex, we can look up each triangle point's Vector3 position using the value of the tris array. Remember, the tris array just holds integer values that map to indices (which hold Vector3 objects) in the verts array.

Next, we have to do a normal vector calculation. The normal of a triangle tells the rendering system which direction the triangle's

facing. In turn, this tells us in which direction a new block should be placed. It would be nice if Unity stored this information for us, but Unity stores mesh normals per vertex, not per triangle. So, we'll have to calculate this the old-fashioned way (see **Figure 6** for a visualisation of this process.) Once you find triNormal, be sure to normalise the vector. We only need the direction, not the magnitude. Remember, the normal tells which direction the new block should be placed in. To find the new block's position, simply add the normal to the position of the old block.

Handling player input is easy. If the player clicks the right mouse button (Fire2), place a new block. If the player clicks the left mouse

button (Fire1), destroy the block the player's looking at.

The **GenBlock()** function will create new blocks for us. It accepts

a Vector3 position and returns a new block GameObject. When you instantiate the block, be sure to use the activeID to get the right prefab from the blockCatalog, and make it a child of the blockGroup GameObject.

We're almost ready for a good playtest, but first, head back to the Unity Hierarchy and do the following. Select the Block GameObject and delete it from the game scene. Select the Player GameObject and, in the Inspector, drill-down Block Catalog to reveal the array's options. Change Size from 0 to 1. Inside the Project panel, find the Block prefab and drag it over to the Element 0 property of Block Catalog. To verify your Block Catalog settings, check out **Figure 7**.

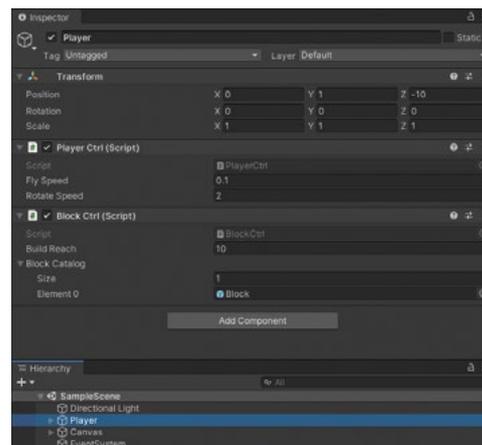
Now, press the Play button and watch the magic unfold! Aww, isn't there something wonderfully soothing about freely placing blocks? Once you've played to your heart's content, open the BlockCtrl script. Now, let's add the ability to place different blocks.

Insert the following at the beginning of the **Update()** function:

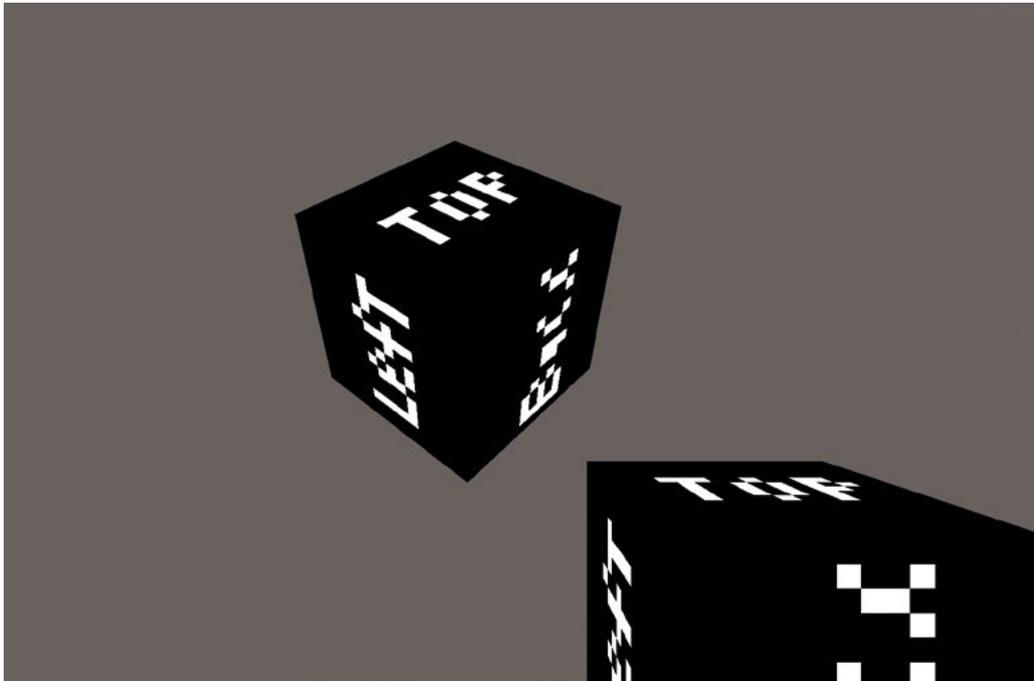
```
void Update() {

    float scrollInput = Input.GetAxis("Mouse ScrollWheel");
    if(scrollInput > 0) {
        activeID++;
        if(activeID > blockCatalog.Length-1) {
```

“Open the BlockCtrl script, and let's add the ability to place different blocks”



▶ **Figure 7:** You'll add more blocks inside the Block Catalogue soon! Remember these steps.



◀ A proper UV map allows different images to be mapped to each side of the block.

```

        activeID = 0;
    }
}
else if(scrollInput < 0){
    activeID--;
    if(activeID < 0){
        activeID = blockCatalog.Length-1;
    }
}
}
...

```

This bit of code starts by grabbing the player's scroll wheel input. If the player scrolled in the positive direction, increment to the next block ID. If activeID reaches a number greater than the max catalog index, loop back to the first index. If instead the player's scroll is negative, decrement to the previous block ID. If activeID reaches a number less than the first index, loop to the max index.

Let's head back to the Unity editor and create a second Block prefab. Select the Block prefab in the Project panel. Press **CTRL+D** (or **COMMAND+D** on macOS) to duplicate the prefab. Also, select and duplicate the BlockMat material in the same way. Then, select the new material (which is named BlockMat 1) and in the Inspector, change its texture to something different from the first block's. Apply BlockMat 1 to the new prefab (which is named Block 1) via drag and drop in the Project panel. In the Hierarchy, select the Player GameObject. In the Inspector,

increase the Block Catalog's Size by +1. Drag and drop the new Block 1 prefab from the Project panel into the newly created Element 1 property of the Block Catalog. You can repeat these steps to add as many new blocks as you want!

COSMETIC CRAFT

Now that the core functionality is finished, let's add some cool stylistic touches. Let's start by adding a fairly small change in the code of the BlockCtrl script that will show what block the player currently has selected:

```

public class BlockCtrl : MonoBehaviour {
    ...
    GameObject blockPreview;

    void Start() {
        ...
        RenderPreview();
    }

    void Update() {
        ...
        if(scrollInput > 0) {
            activeID++;
            if(activeID > blockCatalog.Length-1) {
                activeID = 0;
            }
            RenderPreview();
        }
        else if(scrollInput < 0){
            ...
        }
    }
}

```

♥ I like using Unity's Default-Checker-Gray as a test texture. You can find this by clicking Select at the bottom-right corner of the material's texture property in the Inspector panel.





Figure 8: The Unity animation panel can be a bit tricky, so check these settings.

```

activeID--;
if(activeID < 0){
    activeID = blockCatalog.Length-1;
}
RenderPreview();
}
...
}

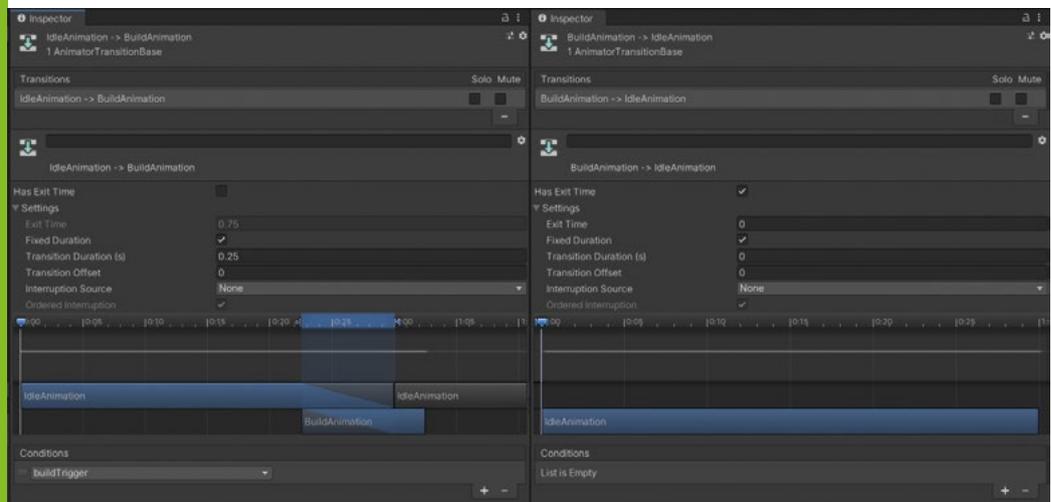
void RenderPreview() {
    Destroy(blockPreview);
    blockPreview =
Instantiate(blockCatalog[activeID], cam.
transform);
    blockPreview.transform.localPosition = new
Vector3(1,-1,2);
}
...
}
    
```

The code above does the following: first, add a new field named blockPreview. This will hold a GameObject showing the current block to the player. In **Start()**, call **RenderPreview()** to spawn the initial blockPreview. In **Update()**, re-render the blockPreview each time the player scrolls to a different block ID. Do this for both the positive and negative scroll directions.

In the **RenderPreview()** function, first destroy the old blockPreview. Then, instantiate a new blockPreview as a child of the cam GameObject. Finally, set the new blockPreview's local position so it will look like the player is holding it in their hand. The player now has a nice preview of what block they're using before placing it.

Next, let's add a build animation. Right-click inside the Project panel and select Create > Animator Controller. Name it BlockAnimator. Drag it onto the Block prefab to add an Animator component. Do this on every Block prefab. Right-click on the Block prefab and select Open. This will open 'Prefab Mode'. From the menu bar at the top of the screen, open Window > Animation > Animation. Make sure Block is selected in the Hierarchy, then click the Create button in the Animation panel. Save the animation with the name IdleAnimation. Click 'Add Property'. Drill-down Transform, and select the '+' next to Rotation. Create a new clip and save it as BuildAnimation. Add the Rotation property again. Drag the timeline scrubber to 0:30. Drill-down Block : Rotation. Change Rotation.x to 45. Exit Prefab Mode by clicking the arrow at the top-left corner of

Figure 9: Left image shows the transition settings for IdleAnimation > BuildAnimation. Right image shows the transition settings for BuildAnimation > IdleAnimation.



the Hierarchy. Check out **Figure 8** to verify all your key frames. Now, set up the Animator Controller. With BlockAnimator selected inside the Project panel, click Window > Animation > Animator from the menu bar. Making sure Parameters is highlighted at the top left-hand corner, click the '+', select Trigger, and name it buildTrigger. Right-click on IdleAnimation, select Make Transition, and click on BuildAnimation to set a transition. Select the new transition arrow, and add to it a condition set to buildTrigger. Then, create an opposite transition from BuildAnimation to IdleAnimation, but leave it without any conditions. (Note both of these transitions require extra settings that are shown in **Figure 9**.) Finally, the playback speed of the animation might be a little sluggish. So, select **BuildAnimation** and set its Speed to 3. When it's all said and done, your Animator panel should look like **Figure 10**. Now, add a tiny bit of code in the BlockCtrl script:

```

...
if ( Input.GetButtonDown("Fire2") ) {
    GenBlock(newBlockPos);
    blockPreview.GetComponent<Animator>().
SetTrigger("buildTrigger");
}

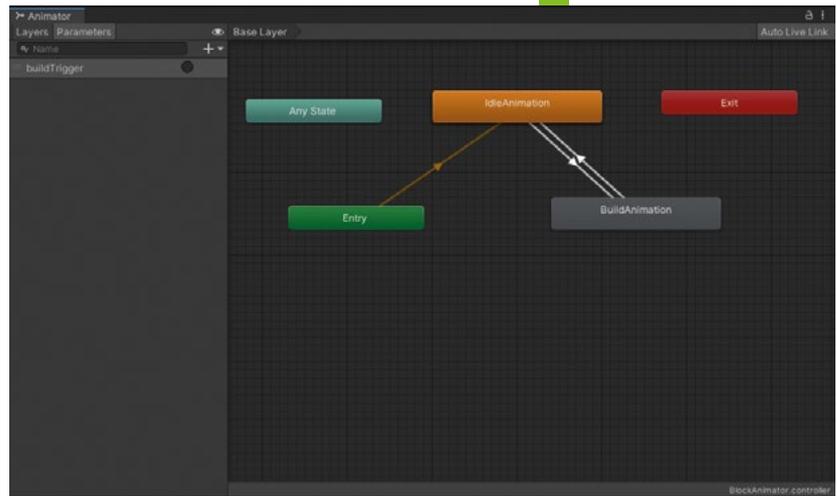
if( Input.GetButtonDown("Fire1") ){
    Destroy(pointRay1.collider.gameObject);
    blockPreview.GetComponent<Animator>().
SetTrigger("buildTrigger");
}
...

```

In the code that checks for mouse clicks, this tells blockPreview to trigger the build animation. It triggers the same animation when placing or destroying blocks.

The final feature our game needs is block UV texture mapping. Currently, the block prefabs only have a single image on every side. To change this, we'll need to import a different block model with a correct UV map. If you're familiar with 3D modelling, feel free to create your own cube UV model. If not, don't worry: I've set up a properly mapped model you can download from GitHub.

Go ahead and download/import the file now. Drag the **CubeUV.fbx** file from your File Explorer window into the Unity Project panel. Select



^ **Figure 10: BlockAnimator should have this configuration.**

CubeUV in the Project panel. In the Inspector, change Read/Write Enabled to true (or check the checkbox). Click 'Apply'. Now, download **UVOutline.png** from GitHub. Then Import **UVOutline.png** by dragging it into the Unity Project panel. Select UVOutline in the Project panel. In the Inspector, change Filter Mode to 'Point (no filter)'. Click 'Apply'.

Now we've imported both the CubeUV model and a texture, let's create a new BlockUV prefab. In the Project panel, select and duplicate the BlockMat material. Rename the new material to BlockUVMat. Select BlockUVMat and, in the Inspector, change its texture to UVOutline. Drag CubeUV from the Project panel into the Hierarchy. Select it inside the Hierarchy and rename it to BlockUV. In the Inspector, click 'Add Component'. Search for 'Mesh Collider' and add it. Drag BlockAnimator from the Project panel and drop it onto BlockUV inside the Hierarchy. Do the same with BlockUVMat. Drag BlockUV from the Hierarchy into the Project panel to save it as a prefab. If a popup asks if you want to create a "new original Prefab or a variant", select Original Prefab. Delete the now unneeded BlockUV from the game scene Hierarchy, but keep the prefab in the Project panel. Add the BlockUV prefab to the Player GameObject's Block Catalog like we did before.

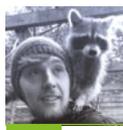
There we have it: a fully functional, *Minecraft*-style creative mode demo. Give it a playtest, then try adding more blocks by creating your own textures – or importing more of ours from GitHub at wfmag.cc/wfmag53.

VERTICES & TRIANGLES

Mesh.triangles is an array of integer values that correspond to indices in the Mesh.vertices array. So, in our code, every three entries in tris represent one triangle of the mesh. For example, if tris looked like [0,2,3, 0,3,1, 8,4,5, ...], that would mean the first triangle of the mesh is made of the indices 0,2,3 of the verts array. To put it another way, it is made of the points verts[0], verts[2], verts[3].

Ratchet & Clank isn't a movie

This month, Antony opens up Ratchet & Clank: Rift Apart to demonstrate some game storytelling fundamentals



AUTHOR
ANTONY DE FAULT

Antony is Wireframe's game writing and narrative design columnist. He's also creative director of Far Few Giants, and you can find his work on [default.games](#) or [@antony_de_fault](#) on Twitter.

Insomniac's latest game epitomises the qualities seen in triple-A 'prestige' games of the past decade. It's polished, merges the action/platforming and shooter genres, looks and sounds amazing, and tells a linear 'hero's journey' tale that feels like playing a blockbuster movie. But in my experience, that last point conceals heaps of nuance: many new writers I've taught have assumed that since these games *feel* straightforward, like a film, that the writing is simply a screenplay with breaks for game action. This ignores oodles of vital narrative work, so this month I'm breaking down all the various modes through which *Rift Apart* delivers its story content.

MAIN PLOT

Cutscenes, firstly, are relatively well-understood, so I'll keep this brief. They play exactly like a scene in an animated film, they have a linear script, precise timing and action, and the player can't exert any control over the game. Cutscenes are often written in screenplay format.

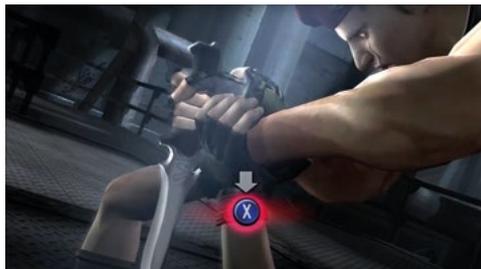
Critical dialogues are nearly unmissable and contain important tutorial, context, or character-building content which happen during play, while

the player's performing another action, such as exploring a new environment. Typically this consists of the player character talking with their sidekick about something they've seen: a recent plot event, a puzzle they've just discovered, etc. This usually doesn't occur during fights, presumably since the player would be too distracted to take it in, or because the dialogue would get in the way of combat audio cues. These conversations are important to the game, and are timed so the player will hear them in their entirety before triggering a fight or cutscene. There's an 'interruption' system in place in case the player enters, say, a shop interface during one of these conversations: each critical dialogue has secret 'checkpoints' throughout it, and if the player disrupts the conversation, the character will cut themselves off with an 'insert-line' such as "Hold on a minute!", then when the interruption is over (player leaves the shop interface etc) will begin again with another insert-line such as "Now where were we? Oh yeah!" before continuing the dialogue from the last checkpoint.

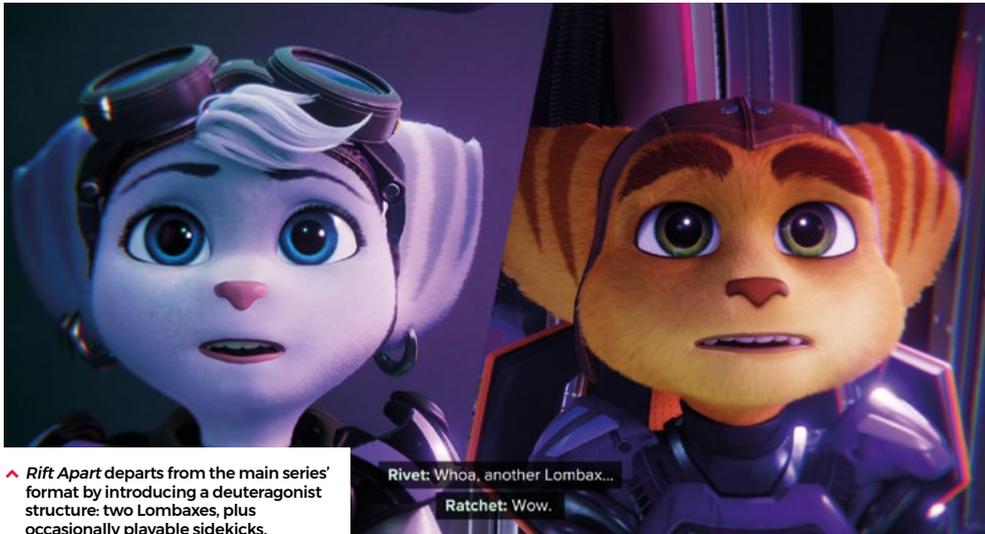
ADDING LIFE

Incidental dialogue is the less important dialogue that happens during gameplay. It doesn't contain critical information, doesn't resume if interrupted, and may or may not have anything to do with the plot. Generally, these dialogues occur between the protagonist and sidekick again, and provide world- and character-building. They are usually short: an exchange of quips, say, between the characters about some past adventure.

Barks are the best example of narrative polish. Essentially, a bark is yelled by an enemy when they're about to shoot. Writers have to come



► In other games, cutscenes might require a button press, known as a quick time event (QTE), to progress, but the principle is the same.



▲ *Rift Apart* departs from the main series' format by introducing a deuteragonist structure: two Lombaxes, plus occasionally playable sidekicks.

up with a number of variants: from the simple "Attack!" to more specific things like "For the glory of Emperor Nefarious!". But barks are so much more than that. Games are lengthy, and the player may easily go for long stretches between cutscenes or critical dialogues. Barks fill that void, and are triggered by or during common gameplay actions. They can be throwaway lines muttered by your character about the current destination, cries of pain or impatience, sales lines spoken by shopkeepers, or calls for help from quest givers. Barks are the chatter that make the game world feel consistently alive, rather than only alive at certain points. They're not plot-critical, but think of almost any triple-A game, and you'll find a vital part of its atmosphere is the quality of the repeated, enlivening barks.

Environmental dialogue is differentiated from incidental dialogue by the way it's located. Environmental conversations are generally short scenes played out between *other* characters and only 'eavesdropped' on by the player, who may choose to stop and listen, but will often simply blast past catching a snippet. For example, while exploring a bar location, if you approach a certain booth you will hear its inhabitants getting snippy with their robot waiter. While this doesn't add to the plot, many of these environmental dialogues intersect with the themes or events at play in the story, providing a kind of vox populi. Alongside barks, they contribute considerably to imbuing the fictional world with aliveness.

EXTRA CONTEXT

Collectable lore is probably the second most widely understood method of implementing narrative in a game (with cutscenes being #1).

Almost every major game in the last 20 years has had world, character, or story information attached to objects which can be found in the environment, from *Gone Home's* scraps of paper, to *BioShock's* audio logs, to *Dark Souls'* inventory item descriptions. *Ratchet & Clank* is no different, having no less than four different types of lore-collectable. Most on the nose are the 'Lorbs' (literally orbs filled with lore) and Spybots, which unlock audio logs in the pause menu. Then you have the Gallery, which has lengthy lore text about every character, unlocked through story progress or finding Gold Bolts. Lastly, each piece of collectable armour is accompanied by a sentence or two of context on its fictional origin.

So, even a 'straightforward' blockbuster like *Ratchet & Clank* has at least six distinct modes of narrative delivery, few of which would make sense to be written down in a screenplay. Instead, even games which appear filmic require a number of different systems and disciplines to tell a story. And this can be put to use: consider which of these you could use in your games to breathe more life into the fiction. 🗣️



KEEP IT FICTIONAL

I'd be remiss not to mention Zurkon Jr.'s preview videos, which are in-fiction advertisements for weapons. They function as a tutorial explaining each weapon's features, and I didn't find much lore in the scripts, but they're written and performed with such strong tone and character that I'd class them as actively world-building, reminding me of the interdimensional television adverts from *Rick and Morty* or the satiric ads in *Starship Troopers*.

▼ This particular sequence in *Uncharted 4* is the earliest example I can find of a game applying an 'interruption' system to voice performances.

Create your own side-scrolling shoot-'em-up

Code a classic horizontal shoot-'em-up in Python – complete with a bullet hell boss battle



AUTHOR
JORDI SANTONJA

Jordi currently works as a software engineer in Norway for a company that develops systems to design infrastructure: roads, railways, utilities... He originally comes from Spain and, in his spare time, codes video games.

MAKING ASSETS

To create the graphics, you need a graphic editor able to use and save transparency. I use the free, open-source image editor GIMP (gimp.org), but feel free to use any other image manipulation program you fancy.

I've always loved shoot-'em-ups, where you destroy waves of enemies in your trusty spaceship. Here, we're going to code the basic systems of a classic shooter in Python using two main libraries: Shapely to detect collisions between objects, and Pygame Zero for the graphics and everything else. Pygame only detects collisions between rectangles and sometimes, with irregular shapes, collisions can feel unfair. Shapely allows us to detect collisions between polygons of arbitrary shape, and lines and points. You can find more on Shapely, and how to install it, at pypi.org/project/Shapely.

Let's start with the background. We could do some fancy parallax scrolling, but we'll keep things simple, using only one scrolling image

covering the whole game area. The background image size is 1600×600, twice the game area, and is split into two files named **background_0.png** and **background_1.png**. This image moves right to left at a slow speed. **Listing01_Background.py** draws the background:

```
def draw(screen, frame):
    screen.blit("background_1", (800-
frame%(800*8)/4, 0))
    screen.blit("background_0", (800-
(frame+800*4)%(800*8)/4, 0))
```

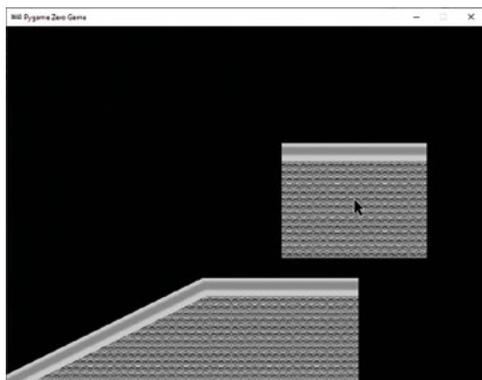
The input value **frame** is subtracted from 800 to obtain decreasing values to move from right to left, and after a modulo operation, the resulting value is divided by 4 to move the background at a speed of 1 pixel every 4 frames. **Listing02_TestBackground.py** shows how to call that function:

```
import Listing01_Background as background
globalFrameCount = 0
def draw():
    background.draw(screen, globalFrameCount)
def update():
    global globalFrameCount
    globalFrameCount += 1
```

Drawing the background uses the CPU heavily, so if your computer is a bit slow, you can disable it.



◀ The full background image, continuous from left to right. It's split into two images: **background_0.png** and **background_1.png**.



^ Run Listing04_CreateForeground.py to place foreground elements on screen and create the foreground map.

FOREGROUND

In the old days, maps, backgrounds, and foregrounds were composed of tiles, placed at uniform distances. Now we can use any size at any position. Using the image manipulation program GIMP, I drew some tiles and saved them as PNGs with transparency. The names of the foreground tiles are stored in the list `foregroundNames` in the file Listing03_ForegroundNames.py:

```
foregroundNames=['centreforeground', 'leftforeground',
                'leftvertforeground', 'rightforeground',
                ...
```

The code in Listing04_CreateForeground.py can be used to create our map by placing the tiles at any position. The controls for the program are: cursor LEFT and RIGHT to scroll the map. Cursor UP and DOWN to change the current tile. MOUSE CLICK to place the current tile on the map. SPACE to print the map in the Python console after sorting the tiles left-to-right.

The text of the map from the Python console must be copied to the `foregroundMap` list in the file Listing05_ForegroundMap.py. Every list element has three values: the index of the tile, the global horizontal position, and the vertical position.

```
foregroundMap = [(12,1634,571), (10,1831,571),
                (10,2072,571), (10,2314,571),
                ...
```

The foreground is an active object where different gaming entities can collide: the player spaceship, the bullets, and the lasers. Listing06_CreatePolygons.py is the program we use to draw the polygons around each object. It imports the list of background images to create a polygon for each. The controls are: cursor keys LEFT and RIGHT to change the current image. MOUSE CLICK to add a new point to the current polygon.



^ You can draw inspiration from existing games. For my shooter, I looked back at the Neo Geo's Last Resort, available to download on Nintendo Switch.

BACKSPACE or DELETE to remove the last point of the current polygon. SPACE to print all the polygon coordinates into the Python console.

We copy these polygons into a new file and assign them a variable, in our case the file Listing07_ForegroundPolygons.py contains `foregroundPolygons`, with the same number of elements as `foregroundMap`, and in the same order:

```
foregroundPolygons = [[(-122, -96), (120, -96), ...
                    [(-204, 96), (186, -96), ...
                    ...
```

Now we can draw the scrolling foreground stored in the map, one pixel every frame. Listing08_Foreground.py places the objects from the map into the list of active elements in a position beyond the right-hand side of the screen. Then it scrolls one pixel to the left every frame. When it disappears beyond the left-hand side of the screen, it's removed from the list. Here's the pseudocode for Listing08_Foreground.py:

```
import libraries
import foreground Names Map and Polygons
class Foreground:
    def __init__(self):
        initialize lists
    def draw(self):
        for element in self.elements:
            element.draw() →
```

MODULAR

The code is split into modules to avoid a huge file that contains everything. In the downloadable files, there are modules with data, classes, and functions; programs to test the code; and programs to create the content. Finally, everything is put together to run the first level of what could be a complete side-scrolling shoot-'em-up game. All files and assets are available at wfmag.cc/wfmag53.

GOOD BEHAVIOUR

The data that defines the behaviour and the graphics of the game is usually stored in files independent of the code. This way, the designer can test the changes without recompiling the code. To keep things simple, this tutorial's data is stored in the code directly into variables. As an exercise, you could replace that hard-coded data with text files read at the beginning of the program to fill these variables with the data.

```
def update(self, globalTime):
    traverse all elements
    delete elements beyond the left border
    compare globalTime with current
foregroundMap
    add to the list if the time has arrived
    rebuild the list of polygons to collide
```

I'd advise running `Listing09_TestForeground.py` to ensure the foreground is drawn and updated correctly.

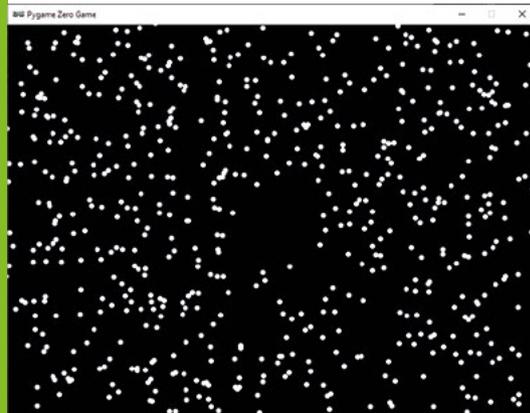
BULLETS AND ENEMIES

Bullets are shot by enemies and move in straight lines until they collide with something or go off the screen. `Listing10_Bullets.py` defines all the code to process the bullets. A `Bullet` object is defined by two members: an `Actor` with the bullet image, and a `speed`. Each frame the speed is added to the position of the actor.

Run `Listing11_TestBullets.py` to test the bullet behaviour. It adds bullets with random positions and speeds, and a collision object defined by a 100x100 square centred on the screen.

The enemies usually appear on the right of the screen and move along a set pattern. I've created my alien ships in GIMP – gun turrets, ships, and the like – then exported them as PNG files to keep the transparency. The names of the enemy files are stored in the list `enemyNames` in the file `Listing12_EnemyNames.py`:

```
enemyNames = ['ball', 'cannon', 'cannon_top',
              'flying_saucer', 'rocket',
              'ship_alien',
              'ship_cannon', 'ship_fighter',
              'ship_pistol', 'big_ship' ]
```



> Execute `Listing11_TestBullets.py` to test the bullets.

Every enemy needs a collision polygon. To create them, use the program `Listing06_CreatePolygons.py` but replacing, at the beginning, the line:

```
from Listing03_ForegroundNames import
foregroundNames as imageNames
```

...with the line:

```
from Listing12_EnemyNames import enemyNames as
imageNames
```

...and proceed as before to store the result in `enemyPolyCoords` in the file `Listing13_EnemyPolygons.py`.

`Listing14_EnemyHits.py` contains the list `enemyNumberOfHits`, which defines the number of hits every enemy must receive before it's destroyed. The bigger the number, the harder the enemy is to destroy.

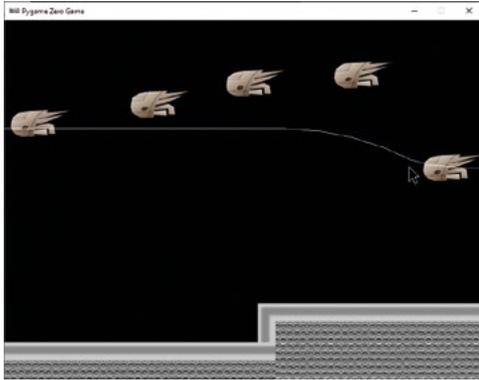
```
enemyNumberOfHits = [1,4,4,10,1,1,12,6,12,1000]
```

In our game, the paths are predefined, drawn manually with the program `Listing15_CreatePath.py`, using the following functions: PLUS key (+) to add a new path. MOUSE CLICK to add a new point to the current path. BACKSPACE or DELETE to remove the last point of the current path. Cursor LEFT and RIGHT to traverse all the paths. SPACE to print all the paths into the Python console.

It's difficult to create wavy trajectories by hand, so this program also creates predefined paths. Once we've created these, we can assign them to our enemies. We select a path for each enemy and store it in the `enemyPaths` list at the appropriate position to correspond with the same position on `enemyNames`. To adjust the starting position of every enemy, I've modified the first and last coordinates of every path to start and end just off the screen. `Listing16_EnemyPaths.py` contains the `enemyPaths` list:

```
enemyPaths = [[(800+32, 0), (790, 1), (780, 5), ...
               [(800+24, 0), (-160, 0)], ...
               ...
```

Every enemy must follow its path. In the file `Listing17_EnemyMovements.py`, the function `PositionFromLength`, from the input parameter distance on the path, returns two results: a Boolean, and the position x,y on the path. An



^ With Listing21_CreateStartEnemies.py, you can create the patterns of enemy waves to appear in the game, using the foreground as a reference.

input distance of 0 will return the starting point of the path, and an input distance of 100 will return the position at 100 pixels on the path from its starting point. The Boolean is **True** when the input length is on the path, and **False** when it's longer than the length of the path. This Boolean is used to finish the enemy movement.

This is only one type of movement, and we'd like to have different enemies with their own patterns. Python treats functions as objects, so we can create a list with different movements to be called for every enemy. The main code will remain the same, but every enemy can behave in different ways. The additional movement functions declared in Listing17_EnemyMovements.py are **Accelerate** and **BigBossMovement**. **Accelerate** is similar to **PositionFromLength**, but instead of following a constant speed, the speed increases. **BigBossMovement** is more complex: it follows a path, and when the path ends, it follows a never-ending sinusoidal vertical path.

These functions are stored in the **enemyMovement** list, to be called from every enemy update. We can create more types of movement and assign them to any enemy.

```
enemyMovement=[PositionFromLength,
PositionFromLength,
                PositionFromLength,
PositionFromLength,
                Accelerate, PositionFromLength, ...
                PositionFromLength, BigBossMovement
]
```

When the code **enemyMovement[index](parameters)** is run, the corresponding function will be called. Another property for each enemy is the ability to shoot bullets. This is achieved in a similar fashion to the movements: the bullet behaviour for every enemy is stored in a list, called **bulletStrategy**, in the file **Listing18_EnemyBullets.py**:



^ Every enemy, when hit by a laser, is changed to an all-white version of its image.

```
bulletStrategy = [NoBullets,
MoreBulletsToSpaceship,
                  MoreBulletsToSpaceship, ...
                  BulletsBigBoss ]
```

The bullet behaviours are: **NoBullets**, which fires no bullets; **SomeBulletsToSpaceship** and **MoreBulletsToSpaceship**, that fire bullets directly at the player's current position; and **BulletsBigBoss**, which shoots circular arrays of bullets spreading in all directions.

Certain enemies also release a power-up when killed. To keep things simple, there's only one type of power-up: increase the number of lasers. **Listing19_EnemyPowerups.py** contains the power-up strategies and behaviours for every enemy. Here, only one enemy uses **AddPowerup**, and the rest use **NoPowerup**.

```
def NoPowerup(powerups, posX, posY):
    pass
def AddPowerup(powerups, posX, posY):
    powerups.append(posX, posY)
powerUpStrategy = [AddPowerup, NoPowerup, NoPowerup,
                   NoPowerup, NoPowerup, NoPowerup,
                   NoPowerup, NoPowerup, NoPowerup,
                   NoPowerup ]
```

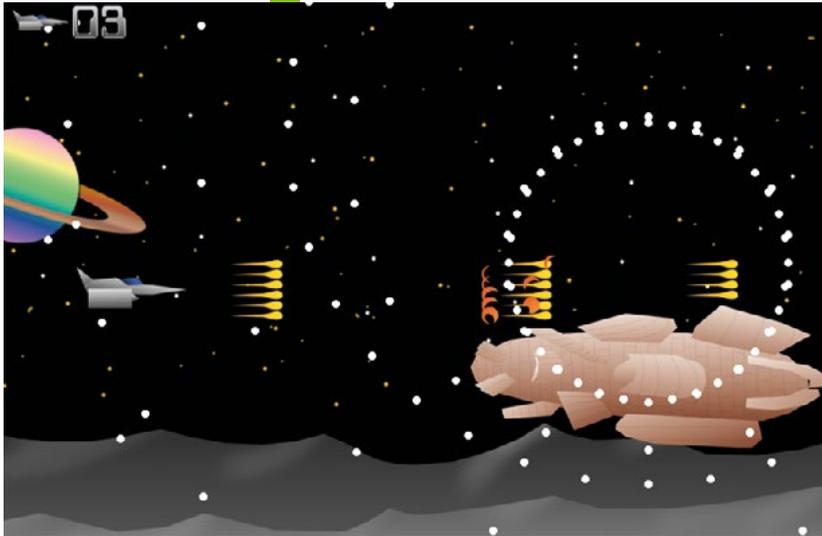
Now we have all the enemy properties and behaviours in place, we can code the **Enemy** class. **Listing20_Enemy.py** imports all the previous listings related to enemies, and implements some functions. Some enemies aren't killed with one shot, so to show they've been hit, they change to a white image, stored in another PNG file with the same name as the regular image →

MAGIC NUMBERS

We should avoid the use of 'Magic numbers' in our code. Instead of using, say, 206, we should use the variable `foregroundObjectHalfXSize`. Doing this avoids the possibility of returning to the code a few weeks later and not being able to remember what the numbers represented: if we see `foregroundObjectHalfXSize`, then the intent of the code is immediately clear.

Toolbox

Create your own side-scrolling shoot-'em-up



▲ The end-of-level boss has its own unique firing pattern to give the player the full bullet hell experience.

but with “_white” in its name. This alternative image is stored in `actorWhite`.

I created this “_white” image by opening the original PNG in GIMP, locking the alpha channel in Layers, and then filling it with the colour white.

The pseudocode of the class `Enemy` in `Listing20_Enemy.py` is:

```
import all modules related with enemies
class Enemy:
    def __init__(self, index, startY):
        initialize all members
    def draw(self):
        if not hit: draw the enemy
        else: draw the enemy as white
    def translate(self, posX, posY):
        translate the actor
    def update(self, bullets, spaceship):
        call pattern strategy, call translate
        call bullet strategy to shoot a bullet
    def hit(self, powerups):
        decrease the number of hits
        if the number of hits is 0:
            call powerup strategy
        return True to destroy the enemy
```

Now we can define the start of every enemy with the help of the program `Listing21_CreateStartEnemies.py`, with similar functions to the previous programs.

With every click on the mouse, a new enemy will follow its pattern, exactly as it will in the final game. After pressing SPACE, we must copy the text from the Python console and assign it to a new variable, `enemyStart`, stored in the file `Listing22_MapEnemyStart.py`. Every element of `enemyStart` contains the index of the enemy to add, the start Y position, and the global time to start.



▲ Do you have what it takes to reach the end of the level?

```
enemyStart = [[5,275,245], [5,261,315],
               [5,245,381],
```

`Listing23_AllEnemies.py` traverses the list `enemyStart` and, when the time arrives, a new enemy defined by the index is added to the list at a position Y. `nextEnemyIndex` contains the position in `enemyStart` of the next element to be added, and is incremented after a new enemy has been added.

```
class EnemyList:
    def __init__(self):
        initialize a list to store the enemies
    def draw(self):
        draw all objects in the list
    def update(self, globalTime, bullets,
               spaceship):
        update all elements of the list
        if currentFrame equal to frame of next
        enemy, add new enemy to list
        update polygons for collisions
    def destroyEnemy(self, index, explosions):
        add explosions and delete from list
    def laserHit(self, index, powerups,
                 explosions):
        call enemy.hit(). If destroyed: call
        destroyEnemy
```

THE PLAYER'S SHIP

The player's ship is defined in the file `Listing24_SpaceShip.py`. `shipName`, which contains the name of its image file, and `shipPolyCoords` contains the collision polygon.

```
shipName = 'spaceship'
shipPolyCoords = [(-48, -5), (4, -9), (10, -3),
                  (40, 1), (19, 5), (-1, 5), (-1, 18), (-37, 18)]
```

This collision polygon has been created with the program `Listing06_CreatePolygons.py`, but this time using the line instead of `ListingXX import`:

```
imageNames = ['spaceship']
```

The class `SpaceShip` defines the functionality of the player's spaceship: collision polygon, remaining lives, number of lasers to shoot, and

POWER-UPS

Please feel free to add new types of power-ups – you could add ship-following satellites like the ones in *Gradius*, as described in the Source Code section of Wireframe #16. You only need a new power-up image and a new behaviour when the power-up is collected.

some variables to control what happens when the ship's hit and how it respawns at the centre of the screen, with a time-limited invulnerability to allow the player to recover from the last hit and continue playing.

Lasers: The laser is the player's only weapon. **Listing25_Lasers.py** defines the class **Lasers** that contains a list with all the lasers currently on screen. All lasers are moved 20 pixels to the right every update, and when a laser exits the screen, or collides, it's deleted from the list.

Power-ups: Certain enemies release power-ups when destroyed. **Listing26_PowerUps.py** keeps a list of the current power-ups on screen.

Score: **Listing27_ShowScore.py** shows an icon of the player's ship and the number of lives. You could add a point score here.

Explosions: The code in **Listing28_Explosions.py** controls and shows the explosions that appear after the destruction of a game object: a laser, a bullet, an enemy, or the player's spaceship.

There are two types of explosions: small and large. Both are made using PNG images. The small explosions are named "impact_" and a number from 0 to 7, and are used when a bullet or a laser impacts on something. The large explosions are named "explosion_" and a number from 0 to 9, and are shown when an enemy or the player's spaceship is destroyed.

```
explosionNames = ["impact_", "explosion_"]
explosionFrames = [8, 10]
```

COLLISIONS

Listing29_Collisions.py manages all the possible collisions between game objects. The functions that check and manage the collisions with every pair of collidable objects are the following:

```
def collisionShipBullets(bullets, spaceShip,
explosions, showScore)
def collisionBulletsForeground(bullets,
foreground, explosions)
def collisionLaserForeground(lasers,
foreground, explosions)
def collisionLaserEnemies(lasers, enemyList,
explosions, powerups)
def collisionShipForeground(spaceShip,
```

```
foreground, explosions, showScore)
def collisionShipEnemies(spaceShip, enemyList,
explosions, showScore, powerups)
def collisionShipPowerUps(spaceShip, powerups)
```

Every function receives its parameters from related objects, which then check, delete, or if applicable, add other objects, such as power-ups or explosions.

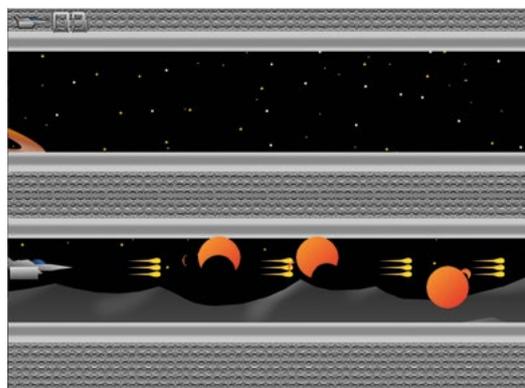
FULL GAME

Now we have all the elements, it's time to put it all together. The code that does this is **Listing30_FullGame.py**. It imports the rest of the modules, defines the variables, and calls the functions with the necessary connections.

The pseudocode of this last listing is:

```
import modules
define variables
def draw():
    draw all the objects from far to near
def update():
    update the positions of all objects
    check collisions between objects
    if something collides:
        destroy the colliding object
        update the objects and game status
```

We now have a single level which could form the basis of a much bigger shooter. To make it into a full game, you could add a title screen with a menu; extra levels, waves of enemies and additional boss battles; a score and high score table; a continue screen, music and sound effects, and lots more besides. Why not use your creativity to see how you can extend and improve it? 🗨



TOOLTIPS

The code in this article creates a level in a full shoot-'em-up game. To follow the advice in the Wireframe issue 48 article 'How not to code', it should be encapsulated in a class named **Level1**, with graphics and maps as creation parameters, so for every level in the game, we could reuse and call this same class. The top-level game loop code should be clear and short.

◀ When an enemy is destroyed, and a laser or a bullet collides, an explosion is created. Explosions are kept in **ExplosionList** at **Listing28_Explosions.py**.

Source Code

Code your own pinball video game



AUTHOR
MARK VANSTONE

Get flappers flapping and balls bouncing off bumpers

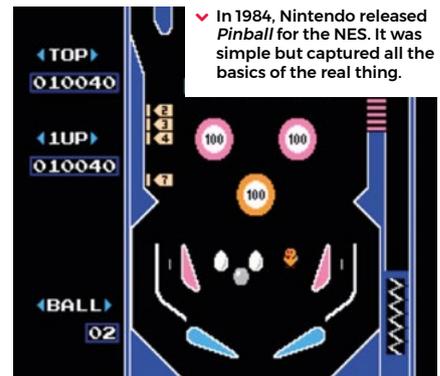
There are so many pinball video games that it's become a genre in its own right. For the few of you who haven't encountered pinball for some reason, it originated as an analogue arcade machine where a metal ball would be fired onto a sloping play area and bounce between obstacles. The player operates a pair of flippers by pressing buttons on each side of the machine, which will in turn ping the ball back up the play area to hit obstacles and earn points. The game ends when the ball falls through the exit at the bottom of the play area.

Video game developers soon started trying to recreate pinball, first with fairly rudimentary graphics and physics, but with increasingly greater realism over time – if you look at Nintendo's *Pinball* from 1984, then, say, *Devil's Crush* on the Sega Mega Drive in 1990, and then 1992's *Pinball Dreams* on PC, you can see how radically the genre evolved in just a few years. In this month's Source Code, we're going to put together a very simple rendition of pinball in Pygame Zero. We're not going to use any complicated maths or physics systems, just a little algebra and trigonometry.

Let's start with our background. We need an image which has barriers around the outside for the ball to bounce off, and a gap at the bottom for the ball to fall through. We also want some obstacles in the play area and an entrance at the side for the ball to enter when it's first fired. In this case, we're going to use our background as a collision map, too, so we need to design it so that all the areas that the ball can move in are black.

Next, we need some flippers. These are defined as Actors with a pivot anchor position set near the larger end, and are positioned near the bottom of the play area. We detect left and right key presses and rotate the angle of the flippers by 20 degrees within a range of -30 to +30 degrees. If no key is pressed, then the flipper drops back down. With these elements in place, we have our play area and an ability for the player to defend the exit.

All we need now is a ball to go bouncing around the obstacles we've made. Defining the ball as an Actor, we can add a direction and a speed parameter to it. With these values set, the ball can be moved using a bit of trigonometry. Our new x-coordinate will move by the sin of the ball direction



✓ In 1984, Nintendo released *Pinball* for the NES. It was simple but captured all the basics of the real thing.

multiplied by the speed, and the new y-coordinate will move by the cos of the ball direction multiplied by speed. We need to detect collisions with objects and obstacles, so we sample four pixels around the ball to see if it's hit anything solid. If it has, we need to make the ball bounce.

If you wanted more realistic physics, you'd calculate the reflection angle from the surface which has been hit, but in this case, we're going to use a shortcut which will produce a rough approximation. We work out what direction the ball is travelling in and then rotate either left or right by a quarter of a turn until the ball no longer collides with a wall. We could finesse this calculation further to create a more accurate effect, but we'll keep it simple for this sample. Finally, we need to add some gravity. As the play area is tilted downwards, we need to increase the ball speed as it travels down and decrease it as it travels up.

All of this should give you the bare bones of a pinball game. There's lots more you could add to increase the realism, but we'll leave you to discover the joys of normal vectors and dot products... 🌀



✓ Here it is: your own pinball game in less than 100 lines of code.

Play the silver ball

Here's Mark's code for a simple pinball video game. To get it running on your system, you'll first need to install Pygame Zero – full instructions can be found at wfmag.cc/pgzero.



```
# Pinball
import pgzrun
import math
import random
from pygame import image, Color

WIDTH = 600
HEIGHT = 800
collisionMap = image.load('images/background.png')
flipperLeft = Actor('flipperl', center=(210,660), anchor=(20, 20))
flipperLeft.angle = -30
flipperRight = Actor('flipperrr', center=(390,660), anchor=(112, 20))
flipperRight.angle = 30

def init():
    global gamestate, ball
    ball = Actor('ball', center=(560,310))
    ball.speed = 5 + random.randint(0, 7)
    ball.dir = 4 + ((random.randint(0, 10)/10)-0.5)
    gamestate = 0

def draw():
    screen.blit("background", (0, 0))
    flipperLeft.draw()
    flipperRight.draw()
    if gamestate == 0 or random.randint(0,1) == 1: ball.draw()

def update():
    if gamestate == 0:
        if keyboard.left:
            flipperLeft.angle = limit(flipperLeft.angle+20, -30, 30)
        else:
            flipperLeft.angle = limit(flipperLeft.angle-20, -30, 30)
        if keyboard.right:
            flipperRight.angle = limit(flipperRight.angle-20, -30, 30)
        else:
            flipperRight.angle = limit(flipperRight.angle+20, -30, 30)
    moveBall()
    checkBounce()
else:
    if keyboard.space: init()

def moveBall():
    global gamestate
    ball.x += ball.speed * math.sin(ball.dir)
    ball.y += ball.speed * math.cos(ball.dir)
    if ball.x > 570 or ball.y > 760: gamestate = 1
```

```
def checkBounce():
    global score
    d = math.degrees(ball.dir)%360
    inc = -1.5
    if d > 90 and d < 270:
        inc = 1.5
        ball.speed -= 0.03
        if ball.speed < 0: ball.dir = 0
    else:
        if ball.speed < 10: ball.speed += 0.04
    if flipperRight.collidepoint(ball.pos):
        ball.dir = 4 + (flipperRight.angle/50)
        if keyboard.right :
            ball.speed += 0.3
            moveBall()
        if inc == 1.5:
            ball.dir = 0
            moveBall()
    if flipperLeft.collidepoint(ball.pos):
        ball.dir = 3 + (flipperLeft.angle/50)
        if keyboard.left :
            ball.speed += 0.3
            moveBall()
        if inc == 1.5:
            ball.dir = 0
            moveBall()
    rgb = collisionCheck()
    while rgb != Color("black"):
        ball.dir += inc
        moveBall()
        rgb = collisionCheck()

def collisionCheck():
    r = 22
    cl = [(0,-r),(r,0),(0,r),(-r,0)]
    for t in range(4):
        rgb = collisionMap.get_at((int(ball.x)+cl[t][0],int(ball.y)+cl[t][1]))
        if rgb != Color("black"):
            return rgb
    return rgb

def limit(n, minn, maxx):
    return max(min(maxn, n), minn)

init()
pgzrun.go()
```

ScavLab, and building high-density multiplayer games

How can you make a game with thousands of players interacting at once? Improbable co-founder Rob Whitehead's team is working on the answer



AUTHOR
ROB WHITEHEAD

The co-founder of Improbable, Rob has worked on many areas of multiplayer technology, such as the SpatialOS networking engine and titles that use it. More recently, he's led the ScavLab tech team.

In ScavLab, the experimental mode within the Early Access multiplayer survival game *Scavengers*, it's possible to have over 10,000 characters (both zombies and players) on screen at the same time. While you may have seen this sort of scale in RTS games such as the *Total War* series, doing this in a third-person action game like *Scavengers* is especially hard.

This is because while most game engines can do a great job of rendering thousands of instances of static geometry – like foliage, rocks, debris, or buildings – the problem of rendering animated characters is a considerably more difficult task, as the geometry isn't static, but rather changing every frame as the character runs their animations. All of this animation work –

figuring out what precise pose the character is in – is usually performed on the CPU rather than the GPU, which creates performance bottlenecks.

There are special rendering techniques that can be used that push this animation work from the CPU to the GPU, which has far more horsepower to crunch the different poses of characters, but the results are sometimes of a lower quality than the typical way of doing it. So how do we keep the high-quality animations, but also have the scale?

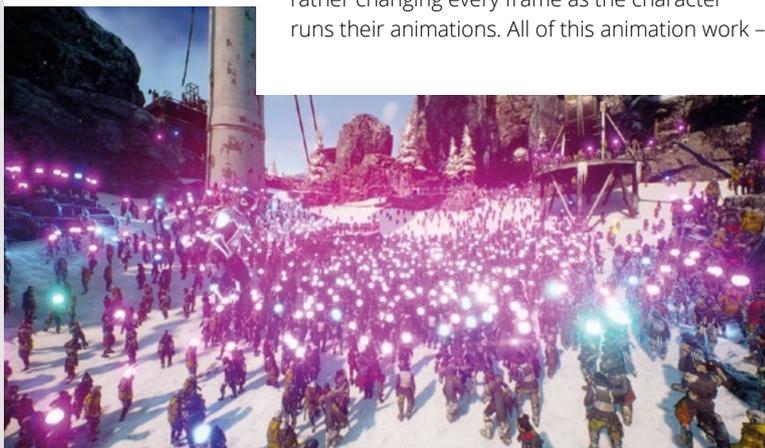
The trick is to use multiple techniques at the same time – traditional characters for your own player, and those near you, but switch to simpler and simpler representations further away. This happens automatically every frame; a prioritisation system understands what you're looking at, and ensures that the largest things on screen get the best visuals.

NETWORKING/BANDWIDTH

Networking is one of the craziest problems when trying to make a game session of this size and density, where every player can see every other player. This is due to how quickly the problem grows as you add more players.

If you think about a game of *Counter-Strike* or *Overwatch* – ten players in an arena, with data being sent 60 times per second (known as the tick rate) from the server to clients, that's a total of 6000 player updates per second (ten players

✓ Arming hundreds of players with grenades is a recipe for disaster. Multi-coloured glow sticks? An oddly soothing sight.





> **Figure 1: Real players would never stand in perfect formation like this, but it's still a great way of testing out the game.**

to send to ten players, where each player can see 60 updates per second). That's quite a bit, but modern cloud servers can easily handle sending millions of updates.

Let's scale that up to ScavLab scale: 10,000 players at our peak tests. The above calculation would come to six *billion* updates per second! That's a million times more than the first example. So what's going on? The problem grows quadratically – doubling the player count quadruples the amount of player information you need to send.

In practice, ScavLab sends a bit less – 250 million updates per second. This is due to the fact that as a player, much of what other players are doing isn't relevant at such high rates. You can receive only two updates per second from the characters furthest away from you, and it still looks fine. As you get closer to those characters, a prioritisation system increases the update rate. While we're using these sorts of techniques at a large scale, they're equally applicable to battle royale-scale games too.

PLAYTESTING WITH LEGIONS

Playtesting a game as often as possible is one of the most important things you can do as a developer. But what do you do when your game session size is 200 times larger than the team working on it? It's really hard to understand game balance, performance, or even how busy the game feels to be inside.

To help with this, the team heavily relies on what we call "simulated players" – thousands of instances of game clients that log into the game,

running in the cloud, and move around just like a player would, and exert the same amount of load on systems as they would, too.

We program simulated players with different patterns of behaviour for testing. In **Figure 1**, we can see them politely walking in formation,

which is a great way of testing that they're perceiving the world correctly. They also have the ability to replay recordings from previous

real-player runs, which is really useful when understanding how they look en masse, moving around an environment in a realistic way.

THE FUTURE

So what's in store for the future? To quote our creative director Bernd Diemer, for the ScavLab events, we wanted players to "feel the goosebumps you get when you experience something amazing in a crowd of people – that shared feeling of being able to react and participate in that moment". It's this feeling – enabled by our architecture for density – that ScavLab is creating and experimenting with in our live events.

We'll be continuing to explore new game mechanics at this scale to learn what works. Our dream is to be able to eventually get to hundreds of thousands of people being able to congregate in virtual spaces like a virtual Glastonbury Festival. We've got a way to go, but many of the above techniques will be critical to push scale further in the future. 🎮



It's your density

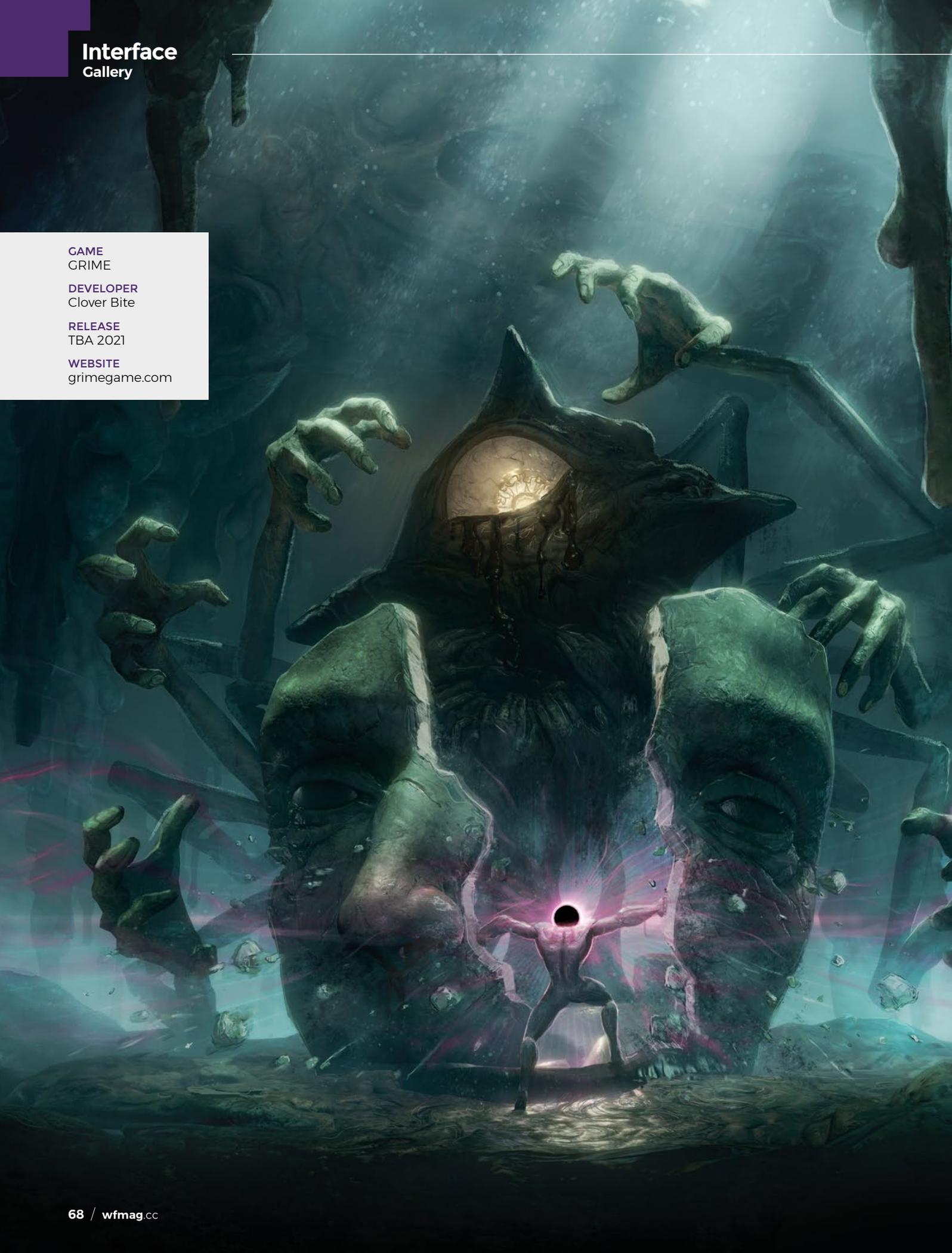
Creating dense game experiences is more than just a technical challenge – what gameplay makes sense in a place that can teeter between orchestration and absolute chaos? While you might be tempted to give everybody grenades, we learned quickly (the hard way) that this is a recipe for absolute carnage, and not a great time. We're starting with the fundamentals, and working our way from there – how it feels to move around with a huge crowd of people, creating ways to interact (such as waving neon glow sticks of different colours), and introducing massive in-world 'messengers' with real-time voice chat to direct the crowd, just like a performer would on a stage.

GAME
GRIME

DEVELOPER
Clover Bite

RELEASE
TBA 2021

WEBSITE
grimegame.com





GRIME

**READ
MORE**
turn to
page 18

Upcoming action-RPG Metroidvania *GRIME* draws on a range of games for its side-scrolling action, but the same's true for its darkly fantastical imagery, which takes notes from all kinds of sources both within and outside the realm of video games. "My main sources of artistic inspiration were Peter Mohrbacher's *Angelarium* and Wayne Barlowe's *Hell* series," explains director Yarden Weissbrot. "Though I ended up looking at so many artworks and getting inspired by so many artists, I couldn't possibly name all of them. The Rockheads [enemies], for example, were inspired by *BioShock Infinite's* concept art of human faces distorted by time, as well as *NieR:Automata's* robots. The weapons were clearly inspired by *Bloodborne*, and the player character itself was probably inspired to some extent by *League of Legends'* Dark Star skins... whether I fully realised it at the time or not. Eventually, *GRIME* ended up becoming its own thing, as I strayed farther from the religious designs of heaven and hell, and more into what universe would make sense for a humanoid black hole to exist in."

As for Weissbrot's wonderfully outlandish creature designs, these generally begin on a piece of paper. "I usually begin with a visual goal," he tells us. "Like, I know this creature should have a lot of hands in its design to tie in with its lore, and from there I just start scribbling out what I think would look the coolest. I then begin thinking about animations and what sort of attacks could work for them, and how would that differentiate them from other enemies. Before it begins being modelled, I usually consult our programmer and rigger [to see] if it will be feasible from their perspective. If not, which has happened, it's time to find an alternative solution."



The forgotten story of

Italy's early game studios

The Italian games industry struggled in the eighties and nineties, but it produced some popular games and a wealth of talent that went on to bigger things elsewhere. We talk to a range of developers about their first steps in a young industry

WRITTEN BY DAMIANO GERLI



- ✦ A pair of Simulmondo's comic book games: *Dylan Dog* and *Tex*. These were sponsored by football player Antonio Cabrini.
- ✓ Simulmondo from an interview in *Zzap!64* magazine in 1989. From left to right: Francesco Carlà, Ivan Venturi, and Federico Croci.



Bologna, resting roughly halfway between Venice and Florence, is an Italian city prized for such culinary delights as lasagne and tortellini.

Founded by the Etruscans, it was one of the most populous European cities throughout the middle ages, while over the last century, the city's growing industrial sector has given rise to several large food and electronics companies. It's little surprise, then, that some of Italy's earliest game studios emerged in Bologna.

In 1987, Francesco Carlà was living in Bologna and working as a journalist. But he also had a passion for video games, and insisted that the medium should be taken seriously, like an art form. Most of all, though, Carlà was haunted

“Carlà was haunted by a vision: a world populated by simulated beings”

by a vision: developing a world populated by simulated beings that would respond to the player's interactions. In *Neuromancer*, author William Gibson called it 'Virtual reality'; for Carlà, it was 'Simulmondo' (literally, a world of simulation). That vision seemed to edge closer to reality the day he met Ivan Venturi.

Venturi, then a teenager, had already been a programmer for several years, working away on his Commodore 64, teaching himself to design increasingly complex adventures. Carlà proved to be a strict mentor, encouraging Venturi to improve his skills. "He showed me *Cauldron II* by Palace Software and said: 'This is how games are meant to be designed, otherwise you're going nowhere,'" Venturi recalls.

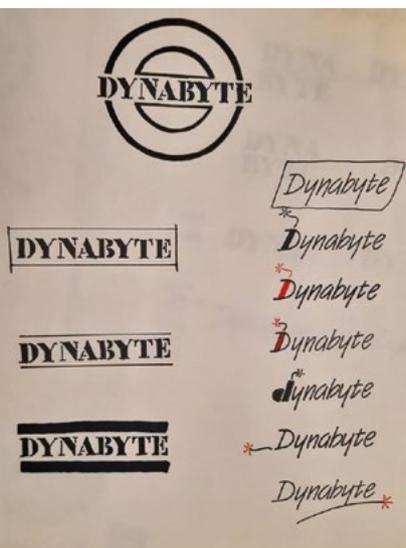
When Venturi showed some progress, Carlà decided to found a games company: Simulmondo. Carlà was fixated on the idea that Simulmondo should represent "Italian electronic entertainment for the world", and so Venturi's skills were honed on sports titles like *Bocce* on the C64 (later released for the Amiga as *Bowls*) and *Simulgolf*. Later, Venturi worked on conversions of *Italy '90 Soccer* and *F1 Manager*, and he describes the latter as his best work on the C64. "Adapting a complex 16-bit Amiga title for an 8-bit computer wasn't an easy task at all," he says. "I was definitely proud of the results."

By 1989, with Simulmondo established in an actual office, several new programmers ➤



✦ Simulmondo present their comic book games in 1993. Ivan Venturi is the first on the staircase on the left, Riccardo Cangini is on the lower right in the red shirt.

► The Big Red Adventure's original release also included a t-shirt, which is now very difficult to find.



had joined the fray. Among them was Riccardo Cangini, who remembers the first ideas given by Carlà, like *Mussolini Age*, a game that took place during the era of fascism. "I had a couple of vague ideas thrown my way, little more than cool-sounding titles," says Cangini. "The development didn't really progress that much since it wasn't clear what the overall design was supposed to be."

In 1990, along with programmer Mario Bruscella, Cangini started developing one of Simulmondo's most fondly remembered titles, *I Play: 3-D Soccer*. "Carlà pitched us an idea of a 3D, first-person soccer simulator with the player controlling a single player. A unique concept for sure, but we were the ones tasked with bringing that vision to life on a system with 64kB of memory!"

It wasn't the most playable of simulations, but *3-D Soccer* was widely noted for its ambition – not to mention a Draconian protection system developed by Bruscella. "I know people who are still trying to crack it but, as far as I know, to this day no one has succeeded," Cangini recalls.

As its continued growth saw it move to a two-storey office, Simulmondo blossomed into one of Italy's few professional software houses. It also managed to self-publish its games, and closed international publishing agreements with companies elsewhere in Europe. It was around this time that Carlà had an idea: with Bologna home to several comic book artists, he felt the medium could be brought to life through video games. Simulmondo's first licensing agreement was signed with author Tiziano Sclavi for his paranormal investigator, *Dylan Dog*; the

subsequent game was written and directed by Ivan Venturi. *Dylan Dog: Murderers*, released in 1992, developed for Amiga, C64, and DOS, was a graphic adventure with arcade sequences. It was one of Simulmondo's most recognised games, going on to enjoy international success (the UK's *Zzap!64* magazine awarded it 89%). Unfortunately, the game would also signal the beginning of the end for Simulmondo.

By the end of 1991, Carlà had seen potential for a whole series of tie-in games, and had sealed licensing agreements for properties like *Diabolik* (a hugely popular Italian comic) and *Spider-Man*. Sporting a simplified "interactive comic book" design, the adventures were intended to be sold at a budget price alongside comic books in newspaper kiosks. It's worth mentioning here that the 'comic book games' idea was also meant to bring the fight directly against piracy, which was a major problem in Italy at the time: illegal games were still being sold in newspaper kiosks all over the country. Italy had a 'professional' industry of pirated games that even 1993's anti-piracy law couldn't completely eradicate.

With Carlà also making up original characters to jump-start more comic book series, however, Simulmondo began spiralling out of control. By mid-1992, the company wasn't investing in research and development anymore – the workforce was instead fighting to meet strict deadlines for its tie-in games, released monthly.

Soon, recalls Venturi, the quality of the games started to deteriorate and the repetition of development began to affect the programmers'

"I'd been working under strict deadlines since I was 15... It was over"

morale. "It went from being a creative endeavour to working on an assembly line – we were getting burned out," says Venturi.

"The comic book business wasn't even bringing a significant income," adds Cangini. "It was a completely different market segment than what a software house was accustomed to."

Tensions mounted, and several developers left, with Venturi following suit in October 1993. The dream of "Italian electronic entertainment for the world", and a studio that could compete with the stronger game industries in France and the UK, was all but over. Venturi's departure hit the company particularly hard; without him, there was nobody to balance the programmers'

GREAT EXPECTATIONS

Simulmondo was always on the lookout for new talent, but Venturi also remembers the gallery of weird characters that would write letters or even show up at the office. "They'd read the success stories of programmers like Jeff Minter and would send us simple demos, expecting to be met with showers of money. It was like, 'Here's the floppy disk, where's my Testarossa?'"

sour moods and Carlà's demands. "I was deadily tired," Venturi recalls. "I'd been working under strict deadlines since I was 15. I just couldn't take it anymore. It was over."

By 1994, after losing most of its talent and saturating the newsstand market, Simulmondo was forced to work on smaller projects. In the following years, it mostly dabbled in multimedia products. The company disappeared from the public eye until 2000, when Carlà closed it down for good. "It was such a sad ending for a great company. It was supposed to shape talents for our future," sighs Venturi.

DYNABYTE

Not far from Bologna sits Genoa, one of the key ports in the Mediterranean Sea. It's often referred to as "La Superba", which means "the arrogant one". It was here that, in 1991, another Italian software house was born, having first appeared in the dreams of a bright-eyed young gamer.

Christian Cantamessa, a teenager in love with cinema, games, and pen-and-paper RPGs, spent his days dreaming of a cinematic point-and-click adventure starring a thief. Cantamessa had zero experience in programming, but as luck would have it, he met graphic designer Massimo Magnasciutti, who'd previously worked, along with programmer Paolo Costabel, on a project for English publisher, Mirrorsoft. That project was *Crimetown Depths*, a complex action-adventure game for the Amiga that was cancelled before its release.

With Cantamessa's father putting up money for his video game project, the three started working on a project then called *Steve Sailing on the Crime Wave*. Cantamessa remembers leaving the project as soon as the others found out he had no experience in game design; years later, he'd become the lead designer for Rockstar's

Manhunt and *Red Dead Redemption*. Humble beginnings indeed.

Still, with Cantamessa out of the picture, Magnasciutti and Costabel carried on developing their prototype, working under the studio name of Dynabyte. The game was eventually released as *Nippon Safes, Inc.* for the Amiga and later PC. It sold solidly in Italy, but fared less well in other countries around Europe. "As far as I'm aware, *Nippon Safes* was officially released only in Italy, so I have no clue what happened," says Costabel.

With their debut garnering favourable reviews, Costabel – along with Marco Caprelli and other programmers – started work on a sequel set in Russia, called *Operation Matrioska*. Intent on avoiding further mishaps, the studio scouted around for an international publisher. While showing a demo at a London trade show, they get the attention of Core Design, who agreed to publish it. Core did, however, insist on changing the title to *The Big Red Adventure*, since – according to Caprelli – they thought it sounded more European.

Thanks to the publishing deal, *The Big Red Adventure* would go on to become Dynabyte's biggest European success, and one of the more high-profile Italian adventure games. The ending hinted at a third episode, in the style of *The Arabian Nights*, that Costabel says was never developed. Caprelli recalls that Core wasn't interested in another project, either: ➔

TEQUILA

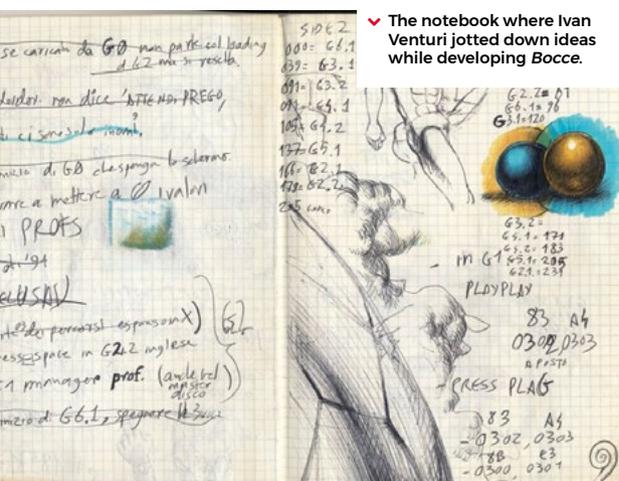
MOCKINGBIRD

Tequila & Boom-Boom was published by Sacis, a public company owned by the Italian television network (RAI), that seemed to have little to no experience in marketing video games. They also briefly distributed Ubisoft titles like *Rayman*.

▼ *Tequila & Boom-Boom's* character designer was Alessandro Barbucci – he'd later become a major Disney cartoonist.



Hello, Can I join in?
Why don't we see each other outside.
Who's winning?



▼ The notebook where Ivan Venturi jotted down ideas while developing *Bocce*.

a cartoon Wild West adventure in the style of Sergio Leone. "They deemed it too childish," he says. "By then, with the Playstation, a game had to be in 3D or it would never see the light of day".

Tequila & Boom-Boom, released in 1995, was Dynabyte's final release. Plagued by distribution problems, it sold poorly even in its native country. The following year, internal issues in the company led to a split, with Costabel going on to work on pioneering CG movie *Final Fantasy: The Spirits Within*, and Caprelli deciding to take his final project, *Blood & Lace*, to another publisher.

Designed as a gothic adventure with QTE, *Blood & Lace* shared the same engine as Dynabyte's previous games, but was redeveloped as a *Tomb Raider*-like 3D action-adventure. Released in 2001, it quickly vanished without trace. Costabel's now at Sony Santa Monica, while Caprelli went on to work as a brand ambassador for Ubisoft in Italy.

TRECISION

In the nineties, so much talent emerged from the Genoa area that it became jokingly referred to as "Basilicon Valley" – a wordplay on basilico (basil), one of the main ingredients of pesto. Rapallo, a quiet tourist town that lies not far from Genoa, was home to one of the most important and long-lived studios in Italy: Trecision.

Founded by Pietro Montelatici, Edoardo Gervino, and Gabriele Pompeo, Trecision's first title was *Profezia*, an adventure released in 1991 for the Amiga. Its second project was the puzzle



▲ *Pray for Death* features Lovecraftian deity Cthulhu in its range of fighters.

game *Extasy*, programmed in MS-DOS by Fabrizio Lagorio and published by Simulmondo. Both Lagorio and Trecision co-founder Montelatici recall they were later hired by Carlà to work on an adventure game based on the aforementioned Italian comic, *Diabolik*. "I had a feeling the deal wasn't going to lead to anything and, well, that's what came to pass," Lagorio says. "Carlà said our development was too slow and the deal was off. We never received any money."

Instead, Trecision decided to shape the project into something else, without the *Diabolik* licence, and sell it internationally as shareware. The studio was surprised when the resulting game, *In the Dead of Night*, was something of a hit. "We were being mailed all this international cash," recalls Lagorio. The studio realised that the market in Italy was stagnant, and therefore decided to scout for an international publisher for their follow-up project. "We got a reply from Stewart Bell at International Computer Entertainment [ICE]," recalls Lagorio. "They worked on the design, while we completed the graphical engine. It felt a bit like being their employees."

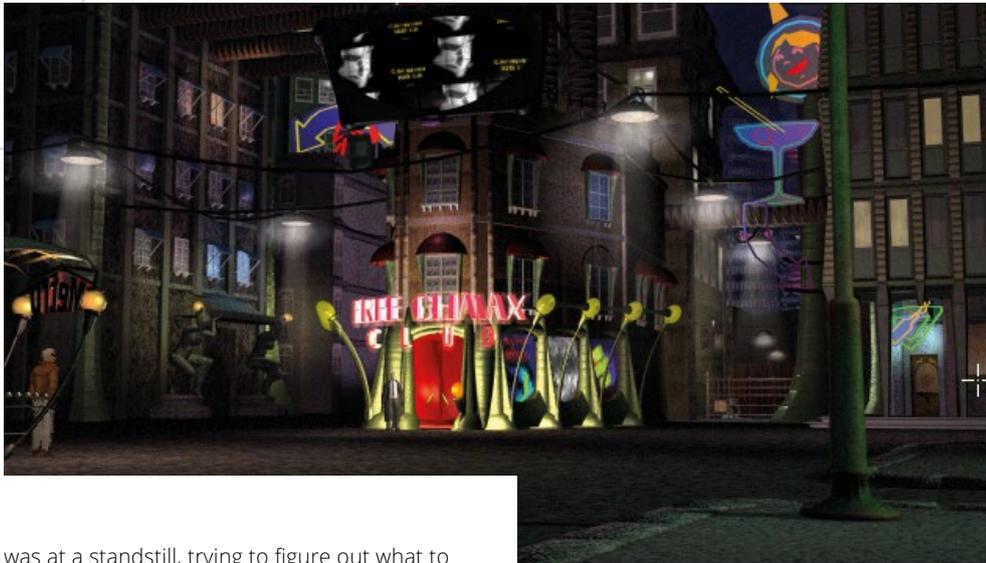
The project, a point-and-click adventure named *Alien Virus*, was released both in Europe and the US, but not without some unforeseen issues. "We were told it had been recalled from American stores because the cover was racist," says Lagorio. "But we hadn't even been informed of any changes to the European box art!"

The deal with ICE kept going through their second adventure, *Ark of Time*, but Trecision was getting tired of the issues brought on by its publisher, and decided to search for another deal for their next game. *Nightlong: Union City Conspiracy*, distributed by Team17 in 1998, is often mentioned as Trecision's best – and most technologically advanced – release. A cyberpunk adventure with fully animated 3D characters on 2D backgrounds, *Nightlong* ran up huge costs for the localisation and dubbing required to publish the game internationally.

With Team17 seemingly uninterested in working with Trecision again, the company

✓ You can see Trecision's old office in Rapallo here in the middle of the picture.





was at a standstill, trying to figure out what to do next. But then a joint venture left Trecision with some generous funding, and so it started thinking about becoming a hub for promising Italian developers. "By 2000, we'd become the biggest Italian software house – there was no way to go but up," recalls Montelatici.

Montelatici talks about the period with both excitement and a tinge of disappointment. "We welcomed to Trecision several talents, like former Westwood developer Rick Gush and *Kick Off's* Dino Dini. Unfortunately, many projects remained in limbo, since we couldn't find a publisher. We tried to sell Dini's soccer game to Microsoft – they even invited us over to their campus but, in the end, they still said no."

"We tried to sell Dini's soccer game to Microsoft, but they still said no"

Trecision's last title, 2001's *The Watchmaker*, sold decently, but didn't garner wider appreciation like *Nightlong*. In the years after, Trecision tried to stay ahead of the market, working on mobile games and PlayStation titles, but, when a two-game agreement with French publisher Cryo fell through, Montelatici decided the only option was to close his studio down. "I don't regret the choice – I really didn't want our programmers to be left without money," Montelatici says of the 2003 closure. "Maybe we could've persevered by releasing another couple of games, but we mostly had nowhere to go."

Montelatici left the industry a few years later, while Lagorio still works in gaming with Perpetuum Media. Gervino is happily retired.

Beyond its games, Simulmondo had another legacy: the companies founded by several of its former employees. From the company's ashes, new teams materialised: the first was

Light Shock, founded in 1994 by Massimiliano Calamai and Francesco Iorio, which released just three games in its two-year lifespan: one-on-one brawlers *Pray for Death* and *Fighting Spirit*, and the vehicular action title *Black Viper*.

Meanwhile, Artematica, based in Chiavari, was started by Riccardo Cangini following his years at Simulmondo. His first project was 2001's *Druuna: Morbus Gravis*, an adventure based on the comic by Paolo Eleuteri Serpieri. Cangini continued

to favour graphic adventures over the years that followed, and several former Simulmondo

programmers, including

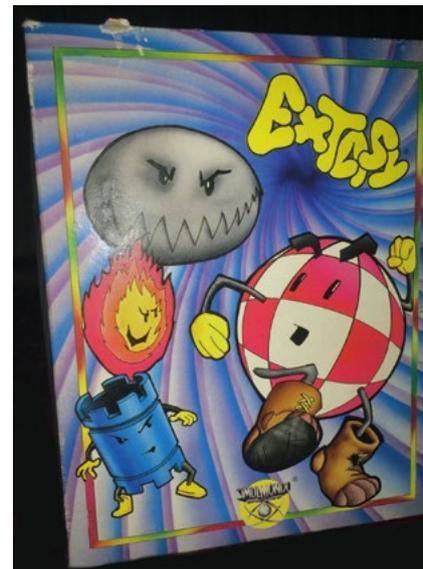
Mario Bruscella and Massimiliano Calamai, also moved over to Artematica. The studio's last adventure, *Julia: Innocent Eyes*, was published by Warner Bros. in 2010.

Together, these studios' histories tell a tale of a nascent market plagued by piracy and limited funding, which, in part, may explain why Italy long-struggled to establish a successful games industry. Despite it all, the country's still managed to produce talents like Cantamessa and Venturi, and in recent years, with better organised academic courses and government funding, things are finally changing for the better. Studios like Milestone and Stormind have produced such high-profile games as *MotoGP 21* and *Remothered: Broken Porcelain*, to name but two examples.

Italian studios may have struggled in the 1980s and 1990s, but today, the country's games industry undoubtedly has a far brighter future ahead of it. 🗨

◀ The *Blade Runner* inspiration in Trecision's *Nightlong* is immediately evident.

✓ *Extasy* was published by Simulmondo in 1992. It's unclear who did the art.



CROSSED

WIRES

While Team17 was showing *Nightlong* at the European Computer Trade Show, Lagorio remembers ICE walked in saying that it was actually their property. "They thought it was *Alien Virus 2*. I don't know what happened next, perhaps they reached an agreement, but surely it didn't win us any points with Team17."

Subscribe today

Wireframe

The MagPi

HackSpace
TECHNOLOGY IN YOUR HANDS

CUSTOM PC

3 ISSUES FOR £10

+

FREE BOOK



wfmag.cc/freebook



Subscriber benefits

- > **Free delivery**
Get it fast and for free
- > **Exclusive offers**
Take advantage of our offers and discounts
- > **Great savings**
Save money on the cover price compared to stores

Special offer:

3 issues for £10 plus a FREE book worth £10

- > **Low initial cost**
- > **Choose from one of five fantastic books**
- > **Free delivery**

3 for £10 offer only available in UK

Subscribe for 12 months

wfmag.cc/subscribe

£55 (UK) £90 (USA)

£80 (EU) £90 (RoW)

Offers and prices are subject to change at any time



Digital subscriptions from

£1.99



Visit wfmag.cc/subscribe or call **01293 312192** to order

Subscription queries: wireframe@subscriptionhelpline.co.uk



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

Interactive / *Crystal Story: Awakening*

Developer Fred Brown tells us how he made his top-down action adventure – with zero programming knowledge

Can one person, with almost no programming experience, craft their own top-down action RPG in the style of *A Link to the Past*, *EarthBound*, or *Chrono Trigger*? If you're Fred Brown from Georgia, USA, then the answer's yes. A graphic designer and illustrator by trade, Brown began work on *Crystal Story: Awakening* in late 2018. Originally a webcomic, it weaves a fantastical story about a little girl named Mina, who's charged with reuniting the four shards of a magic crystal capable of granting wishes.

From its opening cutscene, *Crystal Story's* influences are plain to see: everything from

its resolution to its colour palette to its sprite design screams Super Nintendo, and Brown's eager to point out that the game's influenced by the SNES titles he played (and watched being played) as a child. What's also striking about *Crystal Story*, though, is how solidly made it is; Mina's basic fireball attack is quickly joined by a sword, and the basic enemies you initially meet are stunned with a burst of flame and then finished off with a swish of your blade. It's a satisfying combo that's soon broken up by *Zelda*-like puzzling, exploration, and trickier combat challenges. "I spent a lot of time studying games like *Metroid* and *Zelda*," Brown explains. "I hooked up a SNES Classic next to me on my second monitor, and actually recorded footage/let's play videos and studied animations, like Link's sword swing, frame-by-frame, seeing how the masters made it work."

Although Brown's programming experience was fairly minimal, he'd dabbled in a bit of GameMaker as a youth, and it was this platform he turned to when he began thinking of turning *Crystal Story* from a webcomic to a game. "I decided to give [GameMaker] a shot, since I felt like I was pretty strong with pixel art, and the more I learned, the more motivated I became to stick with it," Brown recalls. "What happened was that I spent a whole year making *Crystal Story: Awakening* from the ground up, with basically





- ^ Many screens contain ingenious little *Zelda*-esque puzzles, such as this one with its cheery ghost and destructible crystals.
- < Turn-based battles are punctuated by neat minigames that have to be completed to dodge an attack or charge up your Action Points bar.

zero knowledge. It was fun, nerve-wracking, and challenging. When things weren't working, it was a huge pain in the butt, but when I fixed something, or got a new mechanic in, it was so rewarding. I remember being ecstatic when I made the initial Mina sprites, and was able to get her to light up some lanterns I drew beneath some arches as she walked underneath them!"

Before heading to GameMaker, though, Brown planned the game by carefully storyboarding everything from its cutscenes to its mechanics; in its early stages, Brown says, *Crystal Story* was fairly basic, with no enemies to fight. "I wanted to make it easy enough that this could be something I felt confident I could do," he tells us, before adding that, as his development skills grew, his ideas also began to increase in ambition.

"What ended up happening was that my speed of learning outpaced those initial storyboards, and I ended up raising the scope significantly. I'd say about 60% of the stuff in the game was me flying by the seat of my pants, coming up with new ideas to have fun with."

Indeed, *Crystal Story's* scope grew to such an extent that what began life as a demo eventually became the first chapter in a much bigger saga – hence the *Awakening* suffix in the game's title. *Awakening* was completed and published on itch.io in June 2020, and Brown's now working on the next chapter, *Crystal Story: Dawn of Dusk*. From the first chapter to the latest, Brown's had some vital support from other artists and musicians in the process of crafting his miniature fantasy epic – Skittlegirl Sound and

Tetrix are working on the soundtrack, while Brown's wife is providing character designs, illustrations, and a spot of playtesting.

With *Crystal Story's* foundations laid, though, an ever-present challenge remains: finding spare time. "Since I work full-time, I had to reframe my way of thinking a bit," Brown explains. "I tackled the development of *Crystal Story* like this: I work from eight until five. I come home, do what I need to do, and I can get started working on the game at around seven or eight-ish. Weekends are typically open for me year-round as well. That gives me about four or five hours (six if I'm feeling bold) to work each

night. Some nights I don't work on anything and just chill. That's cool too, and is extremely important to me! I try to take a week or a few days off after fixing

a huge bug, or finishing up and demoing a big, new mechanic or cutscene as a reward. I try to reward myself frequently after taking care of something tough or important to the game."

Understandably, the global pandemic disrupted the development of *Crystal Story: Dawn of Dusk* somewhat, but Brown's hoping he'll have the second chapter ready before the year's out. "I'm working hard on episode two this year, now that the craziness has passed us for the most part," Brown tells us. "I'm having a blast working on the game, and I can't wait for players to try out all the new goodies. I want this game to be a step-up from *Awakening* in every possible way, so I've been taking my time."

Crystal Story: Awakening is available to download now at wfmag.cc/CrystalStory. 🎮

MAGIC INGREDIENTS

Some of *Crystal Story's* influences are easy to spot, but some are far more left field; its turn-based boss battles, in particular, have a loose connection to a certain *Super Mario* villain. "Besides *Zelda*, there were a ton of influences, some of them not even SNES games," Brown tells us. "Mina acquiring [and using] new abilities were both influenced by *Metroid* and *Zelda*, the back-tracking in *Awakening* was inspired by *Metroid* again... the boss battle system was inspired by *EarthBound*, *Undertale*, and *WarioWare*. I liked how *Undertale* made you more engaged with its turn-based battle system, but I thought it'd be cool to take a new mechanic you've learned from a dungeon (like in *Zelda*), and use it in a fast-paced 'trial' inspired by *WarioWare* to fight a boss."

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

Interactive /

Super Funky Bowling and Station Z-52

We catch up with the winners of last month's Wireframe/FUZE retro game coding competition

If you read last month's edition, you may recall we were so overwhelmed by the quality of entries in our FUZE retro game coding competition, we decided to pick two winners. And while they both use the same platform – FUZE4 Switch – the games we picked couldn't be more different. The first is physics-based action game, *Super Funky Bowling*, created by Nathaniel 'Scrubz' Strub, and the second is a point-and-click adventure, *Station Z-52*, designed by Michael 'Dinocoder' Fuller. The former is a technically ingenious game in the vein of *Super Monkey Ball*, while the latter's a story-driven piece akin to the classic output of LucasArts in the eighties and nineties.



▲ *Station Z-52* designer Michael 'Dinocoder' Fuller from North Carolina.

▼ Although built primarily in FUZE, *Station Z-52*'s wealth of pixel art assets were drawn and animated in Aseprite.

Super Funky Bowling was created by a computer science major who's evidently a bit of a genius at programming, while *Station Z-52* is the work of a freelance artist with almost no coding experience at all.

For North Carolina-based Michael Fuller, *Station Z-52* was designed to make the most of his skillset. "I wanted to ensure I chose a genre that my current coding ability could handle, and it was a genre that I'd been wanting to make a game in – especially a space/sci-fi game," Fuller says.



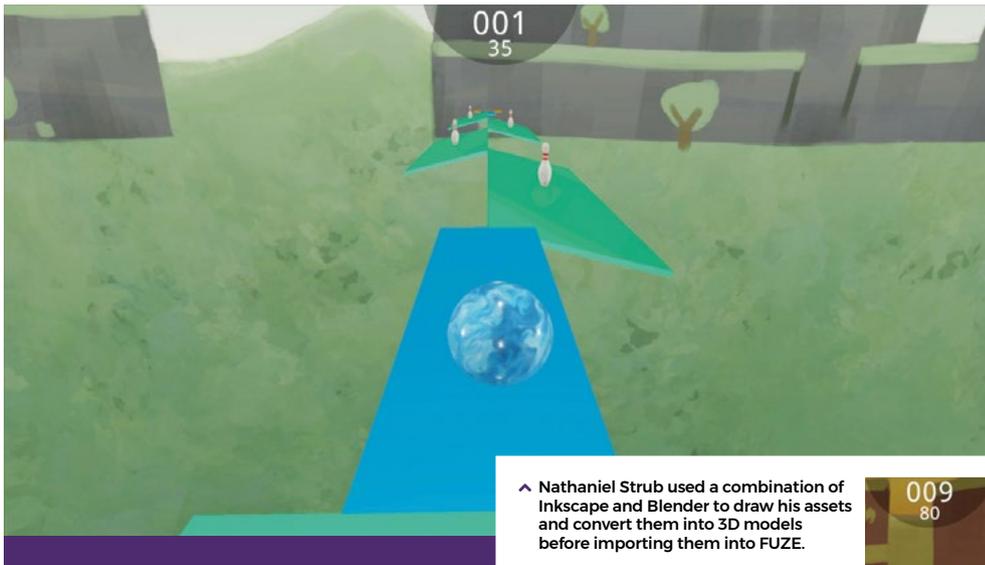
▲ *Station Z-52* opens with the kind of minigame that makes you go, 'Is this it?' before the proper adventure kicks in. It's a genius bait-and-switch.

The result is an absorbing opus that takes place aboard the titular orbiting station; an incident has left it severely damaged, and so it's up to you to try to rescue the situation by solving puzzles in time-honoured point-and-click style.

"Two point-and-click games that inspired me when making *Station Z-52* were *Indiana Jones and the Fate of Atlantis*, and *Tardy*," Fuller says. "The first-person maze using the robot was inspired by the mazes and dungeons in *Phantasy Star*."

Despite his lack of coding experience, Fuller soon made progress thanks to the wealth of tutorial videos online; even so, the sheer scale of the project he embarked on meant that it took time to design all the puzzles and draw the huge number of sprites the story required. "The development of the game took pretty much the entire length of the competition, and I put several hours into it most days," says Fuller. "The artwork took more than half of the time. If I'd had more time, I may have added the characters' quarters as rooms on the station, and I had an idea for





^ Nathaniel Strub used a combination of Inkscape and Blender to draw his assets and convert them into 3D models before importing them into FUZE.



^ Creator of *Super Funky Bowling*, it's Nathaniel 'Scrubz' Strub from East Texas, USA.

an outside area for fixing the station antennae before being able to call for help at the end. I also would've liked to have made a separate sprite for the main character wearing his spacesuit, and also improved his walking animation."

ROLLING, ROLLING, ROLLING

Super Funky Bowling's Nathaniel Strub, meanwhile, had made 3D games before, including a project with a similar ball-rolling premise to his winning entry, but never in FUZE. "Every time I've made a 3D game in the past, I've done all of the 3D rendering myself," Strub tells us. "Having FUZE do the heavy lifting was quite refreshing. I didn't have quite as much control over things as I normally would, but it worked well enough for me. Sometimes optimisation is a huge part of development, but this time I was able to get away with only minor improvements here and there."

While Strub didn't have too much trouble getting the game up and running in FUZE, the biggest challenge came from an unexpected source. "You might assume that the most challenging part of development was the custom 3D physics or assets, but no," he says. "You know how the ball rotates when it moves? Yeah – that was by far the hardest thing to create. I wasn't able to use FUZE's existing commands to rotate the ball how I wanted, so I created my own 3D rotation system from scratch. I could do all sorts of fancy rotation within my system and then convert it to FUZE's. That may sound simple, but it took about two full days of complicated math to figure it out."

"I created my own 3D rotation system from scratch"



One of *Super Funky Bowling's* standouts is its level design, which includes some ingenious, puzzle-like layouts, including switches that flip the rules of gravity. "I designed the game around speedrunning, so I needed to make the levels themselves fun to speedrun," Strub explains. "Virtually every level has some sort of shortcut or faster strategy to make speedrunning more fun. I also tried to keep every level unique and interesting, which was one of the most time-consuming parts of development. I came up with many of the levels myself, but occasionally took ideas from various ball-rolling games."

In their own individual way, *Super Funky Bowling* and *Station Z-52* are both excellent games, especially given the limited time and resources their creators had to make them. Ultimately, they're both testament to what can be done in a platform like FUZE – all you need is patience, a willingness to learn and problem-solve, and a rock-solid game concept. 🎮

GET PLAYING

As we mentioned last month, the winning games in our retro coding competition are all freely available to download on FUZE4 Switch now. From the main FUZE menu, go to 'Programs' and then press **Y** to enter the following codes:

Super Funky Bowling:
NXZZDNDNPNP

Station Z-52:
NXVKGMD9F

Once downloaded, the games will appear in the code editor. From here, you can see the code and run the program using the **+** button or **F5** on your keyboard.

To get the codes for all the other winning entries, just head back to issue 52, over at wfmag.cc/52.



GAME

Jack Move

DEVELOPER

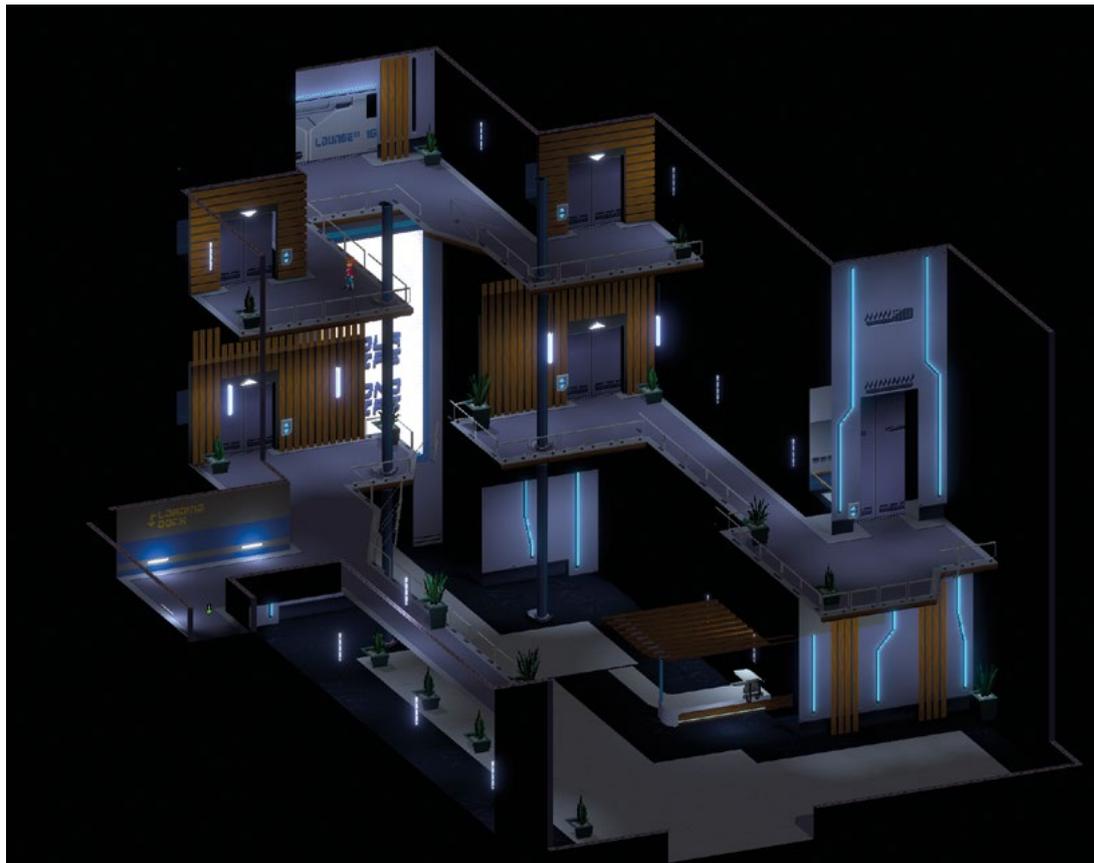
So Romantic

RELEASE

2021

WEBSITE

soromantic.co.uk

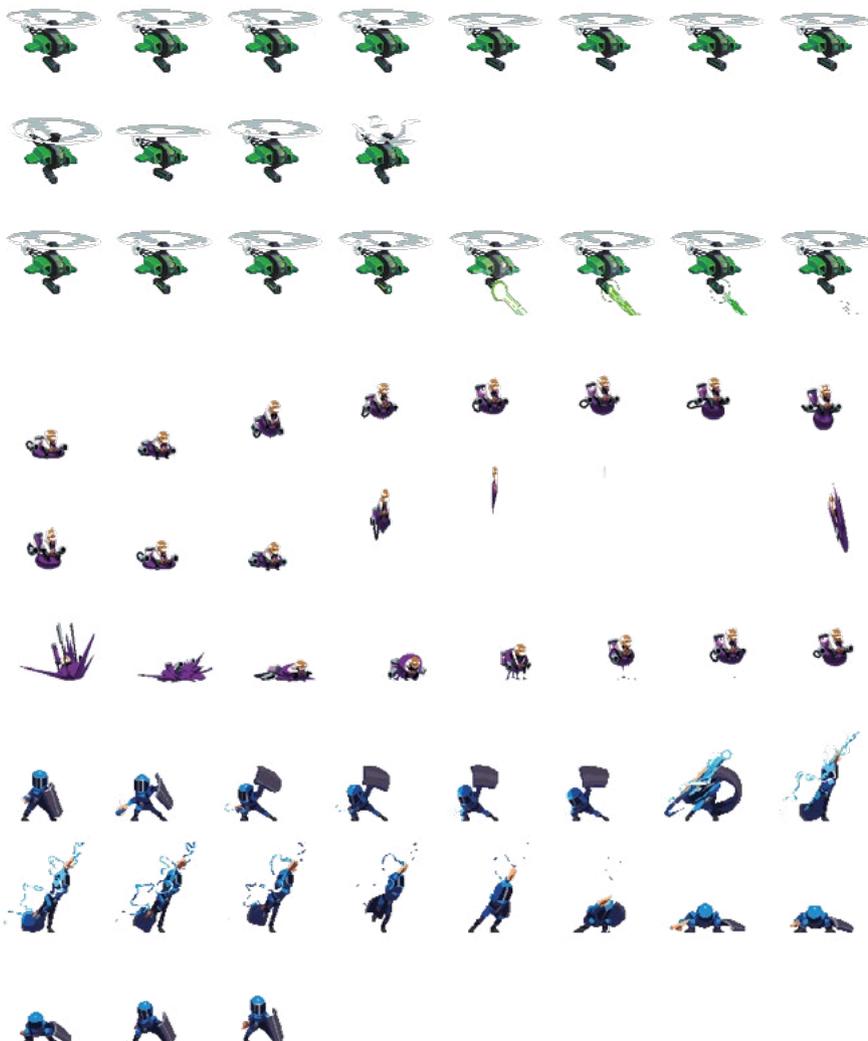




Jack Move

A game we previewed last month, *Jack Move* is a cyberpunk RPG brought to life with vibrant, Japanese-inspired sprites and backgrounds. The world *Jack Move* creates is a pleasing mix of William Gibson-esque dystopia and anime-like whimsy – an aesthetic largely created by its lead artist, Joe Williamson. “As soon as I saw Joe’s pixels, I knew I wanted to work with him,” says designer and programmer, Edd Parris. “He has such a lovely, fluid style.”

Creating *Jack Move*’s colourful, often eccentric cast of characters is a collaborative process, Parris tells us. “I’ll grab a bunch of references and write some descriptions – which are often very loose – and just let the character designer (usually Joe) just run with it,” he says. “This usually skips the pen and pencil part and goes straight to pixels. We have a huge file with tons of different sketches for NPCs and enemies. We’ll then have a chat about what the enemy is all about, what kind of elemental attacks they use, the area they’re situated in, and so on. From there, we spitball ideas of what kind of attacks they could have and iterate on the design a bit more. Then the animator will hop straight in and start roughing out animations. This is a really fun way to work. My original ideas are usually blown out the water by what the animators suggest and come up with.”



Developer Profile

Hudson Soft

Bomberman. The PC Engine. A fond look back at one of Japan's earliest and most important video game companies



At the height of its powers in the late 1980s, Hudson Soft was a force to be reckoned with: its PC Engine console, created in partnership with Japanese hardware giant NEC, was threatening to eclipse Nintendo's Famicom in terms of sales. At the same time, the Sapporo-based firm was developing highly successful games for other platforms, while a separate publishing division in North America ensured its games were being

pushed in that territory. Hudson wasn't a household name like its rivals Sega or Nintendo, admittedly, but for more than a decade, it was still one of the biggest – and historically significant – video game companies in Japan. Not bad going for a business that almost started out as a coffee shop.

Hudson Soft was founded in 1973 as CQ Hudson, an electronics store mostly specialising in amateur radio equipment. Brothers Yuji and Hiroshi Kudo were, it seems, intent on simply starting a business, and weren't particularly bothered about what they were going to sell; Hudson (named after their favourite train) might have wound up as a café had its founders not realised there was already a similar establishment in the same building.

Within a couple of years, Hudson pivoted to selling early computers, such as NEC's TK-80, Sharp's MZ-80K, and Commodore's PET. The home computer market was still in its infancy in the mid-1970s, and software was in short supply. To counter this, Hudson began writing small programs to give away to their customers. "The games we made



▲ *Bomberman*: Hudson's most recognisable and long-lasting contribution to gaming.





were tiny," recalled one of the company's earliest employees, Takashi Takebe, in Read-Only Memory's book, *Japansoft: An Oral History*. "We would include five to ten of them as a bonus with the purchase of a new computer."

Seeing the potential in software, Hudson began to sell its games and other programs on audio cassettes: it would put out adverts in magazines, and sell them via mail order all over Japan. The sheer number of competing computers available at the time meant there was a ready market for games,

and so Hudson had to grow to meet the demand: a second office was opened in Tokyo, and extra staff were hired to keep up with the process of porting software to different systems. The early Japanese home computer scene was, in short, quite similar to the UK's – it was a time when a developer could make a game within a few weeks or even days, get it on an audiotape, and sell it to a community of computer users hungry for a new diversion. The quality of Hudson's early output varied wildly as a result, with the young company putting out dozens of titles per month in the early 1980s. Still, there were some ingenious ideas in the mix – not least the 1983 game, *Bomberman*, released in the UK under

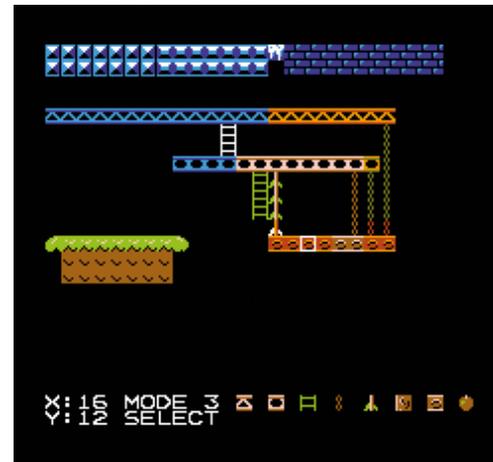
the strange moniker, *Eric and the Floaters*, the explosive maze game would prove to be one of Hudson's most enduring contributions to the medium, appearing on just about every platform imaginable, and still going strong in 2021 with *Super Bomberman R*.

In fact, 1983 marked a major turning point for Hudson as a video game company. Not only was it the year that *Bomberman* was first born – thanks to programmer Yuji Tanaka – but it was also the year when Hudson signed on as the first third-party developer for Nintendo's then-new Famicom console.

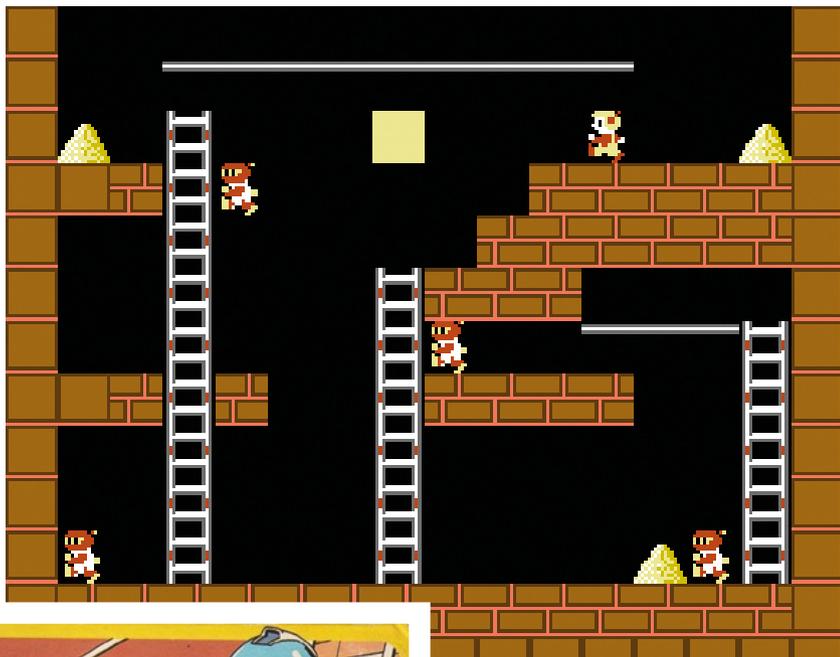
For Hudson, this was a risky move – manufacturing cartridges was expensive, and Nintendo was still a new player in the console market; in 1983, the Famicom's success was far from preordained.

Ultimately, though, Hudson and Nintendo would quickly forge a strong partnership, with Hudson first porting several of its computer games to the system – among them the unforgettably named platform puzzler, *Nuts & Milk* – before getting the go-ahead to translate some of Nintendo's biggest names to other machines. Thanks to Hudson, such games as *Balloon Fight* and *Mario Bros.* appeared on Japanese systems like the PC-88 and Sharp X1. Hudson also ➔

"The quality of Hudson's early output varied wildly"



▲ It wasn't a huge seller, but Family BASIC put programming and simple game dev into the hands of a generation of young Famicom owners.



▲ Hudson's origins as a computer game maker meant it loved giving players the tools to make their own levels. *Lode Runner* on the Famicom had its own editor, for example.

▶ Many of Hudson's shooters featured a Caravan mode - a two- or five-minute high-score challenge that formed the backbone of its travelling tournaments.

developed the software side of Family BASIC - a combination of keyboard and development platform designed to turn the Famicom into a fully fledged home computer.

By the mid-eighties, Hudson Soft had flourished into a major industry player: its Famicom ports of *Bomberman* and puzzle-platformer *Lode Runner* had sold over a million copies apiece, and its annual series of Caravan events (see box) helped spread Hudson's brand all over Japan. Meanwhile, a marketing executive named Toshiyuki Takahashi became the public face of the company as Takahashi Meijin: appearing on comics and television, he became famous for his rapid-fire prowess at 2D shooting titles, and even starred in his own series of video games.

Hudson had made its mark as a developer of video games and other software by the mid-eighties, while its technical prowess was such that it had begun to design its own custom graphics chip, which it unsuccessfully tried to sell to Nintendo. At the same time, NEC was looking to enter the console market, and partnered up with Hudson Soft to develop the venture.



▲ Hudson Soft's distinctive bee logo, sitting proudly atop the company's HQ. It was finally removed in 2017.

The resulting system, 1987's PC Engine, was arguably the most forward-thinking games machine of its era: it was compact, flexible - a CD-ROM add-on, the first of its kind for a console, came out the following year - and far ahead, technically, of Nintendo's Famicom. As well as co-developing the PC Engine, Hudson Soft released a hefty library of games for the system over its lifespan - such titles as *Bonk's Revenge* (PC-Genjin in Japan), *Bomberman '94*, and *Soldier Blade* were among the best ever made for the console. NEC's attempt to push the PC Engine in North America - where it was rebranded as the TurboGrafx-16 - met with only limited success, but still, the system endured in its home country until well into the 1990s.

What's worth noting is that, even at the height of its success, Hudson Soft never lost its geeky, playful edge. Its research and development laboratory on the leafy outskirts of Sapporo, for example, had a miniature railway track that staff could sit on and ride around



the offices – the track even snaked around the outside of the building, like a bantam theme park ride. Even Hudson Soft’s logo – a cartoon bee – had a hint of nerdiness to it: in Japanese, the number eight, hachi, also means bee. The radio call sign for the Hokkaido region is JA8, and so a bee was chosen as Hudson’s mascot – a nod to its amateur radio origins.

Sadly, Hudson’s glory days couldn’t last. The 1994 successor to the PC Engine, the PC-FX, was a major commercial misfire; by the time it was scrapped in 1998, it had sold just 400,000 units – to put the disaster into perspective, the PC Engine had reportedly sold half a million units after its first week on sale. An even bigger blow came from far outside Hudson itself: the firm’s bank, Takushoku, went bust in the late 1990s, severely damaging Hudson’s financial standing in the process. In order to raise much-needed capital, Hudson went public in 2000, a move which allowed it to reinvest money in such new ventures as the growing mobile gaming market, as well as a publishing deal with Infogrames in France. At the same time, however, floating Hudson on the stock market left the company open to a takeover – which is precisely what happened when Konami brought some 5.6 million Hudson Soft shares in 2001. This was, in essence, the beginning of the end: with

“A bee was chosen as a nod to Hudson’s amateur radio origins”

Hudson now controlled by its majority shareholder, it was gradually absorbed into Konami’s corporate bulk, and by March 2012, had vanished altogether. Many of Hudson Soft’s properties still survive – *Super Bomberman R* successfully revived the series in 2017, while Konami capitalised on the lingering nostalgia for retro consoles with the PC Engine Mini device, released in 2020.

For a company with such a long and important history as Hudson Soft, though, the cold nature of its demise seemed all the more cruel. Today, its old R&D office and miniature railway stands empty. Over at its former headquarters,

Hudson’s distinctive bee logo stood proudly over the entrance until 2017, when a pair of builders arrived and quietly took the signage away. What remains, though, is its huge library of games and the equally sizeable impact it made on the medium’s history – thankfully, that’s one legacy that can’t be so easily erased. 🐝

Caravan of Courage

For a period of approximately 15 years, Hudson Soft held a series of competitive tournaments, which it took all over the country under the banner of Hudson All-Japan Caravan Festival. Players competed in two- to five-minute bouts on a specific game chosen each year: 1985’s game was *Star Force*, a top-down shoot-em-up that set the pace for future tournaments. Between 1985 and 1992, all but one of the games taken around Japan were shooters – the glaring exception being 1988, when the featured game was a baseball title called *Power League*. When the shooter began to ebb in popularity in the early nineties, Hudson switched to *Bomberman* games: *Hi-Ten Bomberman*, a title developed exclusively for the Caravan Festival in 1993, ran on an early HD television and supported up to ten players.



◀ Takahashi Meijin became an unlikely celebrity in 1980s Japan – a Hudson ambassador famed for his rapid-fire joystick technique.



▲ From 1985 onwards, these touring ‘Caravan’ tournaments were a fixture in Hudson Soft’s calendar.

10 unmissable Hudson Soft classics

Bomberman, yes, and lots more besides



01 Stop the Express

ZX Spectrum – 1983

One of a handful of Hudson titles released for the ZX Spectrum in the West, this was also one of the 8-bit micro's best action games. Later reworked as one section of the Famicom title, *Challenger*, *Stop the Express* also featured one of the great final messages in gaming: "Congraturation! You Success!"



02 Lode Runner

Famicom / NES – 1984

Hudson didn't create this fast-paced puzzle platformer – full credit to Doug Smith – but its rendition of *Lode Runner* was a key release for the developer. It was its first game for the Famicom, and its first console hit. *Lode Runner* also marked the first appearance of what would later become the classic *Bomberman* sprite.



03 Star Soldier

Famicom/NES – 1986

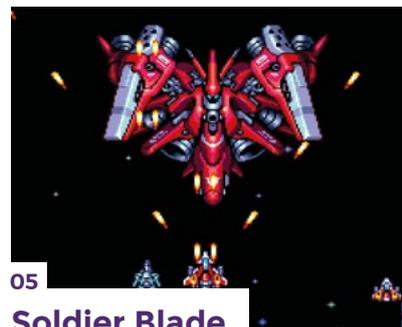
An influential shooter with plenty of zip, *Star Soldier* was Hudson's refinement of Tecmo's markedly similar *Star Force*, which Hudson ported a year earlier. *Star Soldier* was far superior, though, and sparked a whole string of increasingly feisty sequels, many of them appearing in Hudson's Caravan competitions.



04 Bonk's Adventure

PC Engine – 1989

Of the strangely prevalent cycle of caveman-themed games (among them *Chuck Rock* and *Joe & Mac*), *Bonk* is arguably the best. An anarchic platformer with lots of surreal set-pieces, *Bonk* (aka *PC-Genjin*) was the PC Engine's refreshing alternative to the wholesome *Super Mario*.



05 Soldier Blade

PC Engine – 1992

The last *Star Soldier* game for the PC Engine, this sequel took the vertical shooting template and sharpened it to a gleaming point. Fast and eminently replayable, *Soldier Blade* was one of Hudson's greatest ever action games. It also features one of Hudson's very best 'Caravan' score attack modes. An essential shooter.



06

Saturn Bomberman

Sega Saturn – 1996

Opinions differ over which of the numerous *Bomberman* games is the best, but we'd choose this 32-bit outing. Not only is its multiplayer mode the most raucously enjoyable, with a wealth of power-ups to pick up and tinker with, but its single-player campaign is also the most colourful and imaginative in the series to date.

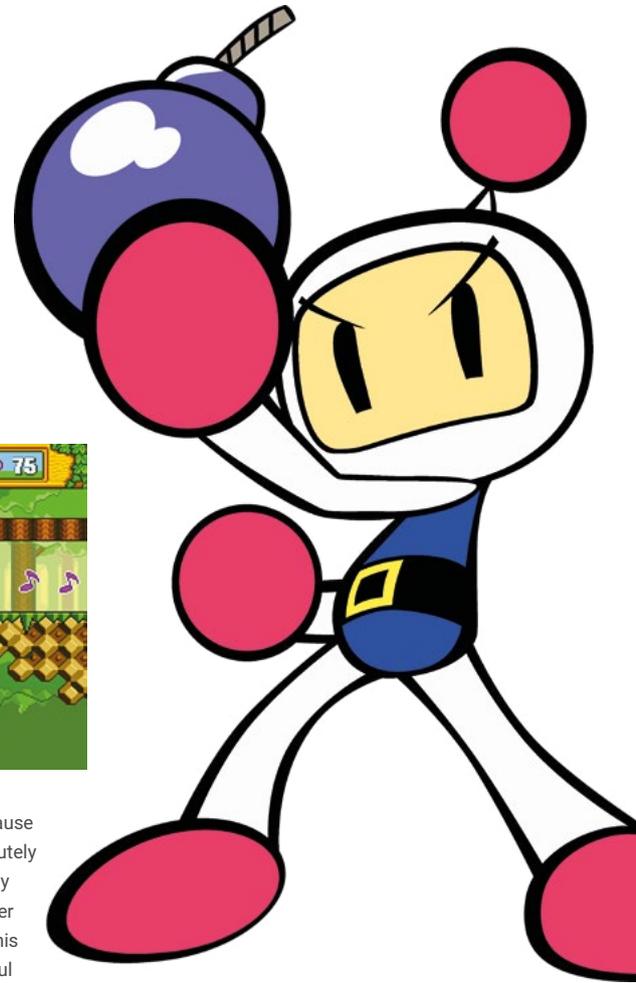


07

DoReMi Fantasy

Super Famicom – 1986

Not all rare games are collectible because they're worth playing, but this is absolutely the case with this charming Japan-only platformer. A sequel to the much darker Famicom title, *Milon's Secret Castle*, this one offers plenty of charming, colourful run-and-jump action.



08

Mario Party 3

Nintendo 64 – 2000

Hudson's relationship with Nintendo continued well into the new millennium as the developer put out a whole series of these digital boardgame/minigame collections. This third main entry for the Nintendo 64 was perhaps the best of a simple yet consistently entertaining bunch.



09

Ninja Five-O

Game Boy Advance – 2003

Another one of Hudson's more obscure releases (and known in some regions as *Ninja Cop*), this run-and-slash action fest plays like a belated answer to the original *Ninja Gaiden* games. Tough yet consistently engrossing, *Ninja Five-O* is also one of the Game Boy Advance's most scarce and sought-after titles.



10

Lost in Shadow

Wii – 2010

Hudson's best days were long behind it by this point, but *Lost in Shadow* displayed a late glimmer of creativity. A vaguely *Ico*-like platformer where you control a silhouette, it was a cracking third-party Wii game that was sadly buried under the avalanche of forgettable shovelware crowding the system at the time.

DLC: Delicious Little Cards



STEVE MCNEIL

Steve is so obsessed with these games we're beginning to think he only writes for us because our pages are also rectangles.

“DLC often has a bad name, and for good reason”

As regular readers of this column will no doubt be aware, I'm somewhat partial to* deck-builders. From *SteamWorld Quest* to *Slay the Spire*, from *Griftlands* to *Gloomhaven*, I can't get enough of oblongs.

The most recent game to catch my eye is *Roguebook*, a roguelike PC title from Richard Garfield, creator of *Magic: The Gathering*, the latter being widely accepted as the first modern CCG (collectable card game). When its demo was released earlier this year, it got a lot of positive attention. However, when it released in full in June, it suffered an immediate backlash due to day one DLC.

By contrast, another game that's consumed countless hours** of my time, which had a far more positively received DLC offering, is *Monster Train: The Last Divinity*, released ten months after the game's successful launch, was everything good DLC should be. It offered substantial additional content, including a new game mechanic that breathed fresh life into everything that had come before (Pact Shards, if you're asking) plus a new end boss to extend every playthrough, for those brave enough to take it on.

Prior to this, several substantial free updates had been added to the game. The first added

a pile of new 'mutators' to modify gameplay as well as expert challenges for those who had conquered what the game had to offer. This was quickly followed by a second free *Friends and Foes* update which added new champions, bosses, cards, and artefacts as well as some nice additions to gameplay. It also implemented various quality of life improvements which recognised popular fan requests. This was then also followed soon after with official mod support, allowing the community to grow their passion even further.

In short, developer Shiny Shoe nailed it. Their commitment to ongoing development and their community meant that, when the paid DLC launched, it would have been hard to argue it wasn't worth the investment if you had enjoyed everything the game had offered to date.

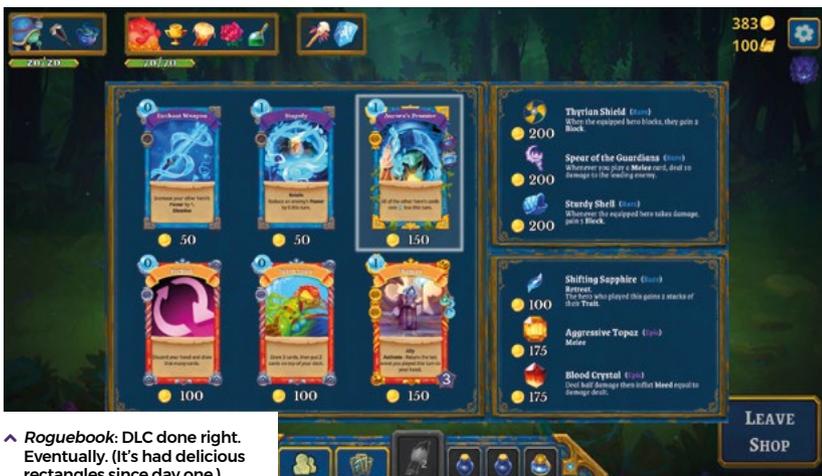
The good news is, *Roguebook* quickly set things right, posting a blog entitled *We messed up* just four days after launch. Originally, they had envisaged their Digital Deluxe Edition to focus on cosmetics, but felt they wanted to offer more to those showing additional support for them, and so added exclusive content. However, they recognised this was a mistake, and worked with Steam to lift restrictions on anyone who wished to seek a refund due to this initial error.

Two very different situations, but both demonstrating a willingness to listen to their fans, and an essential commitment to the community which will hopefully allow the game to develop and thrive for years to come. DLC often has a bad name, and for good reason, so it's great to see two examples of companies giving the issue proper consideration when deciding on their release strategies.

And it means I get more pretty rectangles. Win-win. 🎉

*obsessed with

** actually entirely countable on my Xbox and Steam profiles, but I'd rather not say precisely how many hours I've put into it. Suffice to say, it's somewhat more than one might consider healthy



^ *Roguebook*: DLC done right. Eventually. (It's had delicious rectangles since day one.)

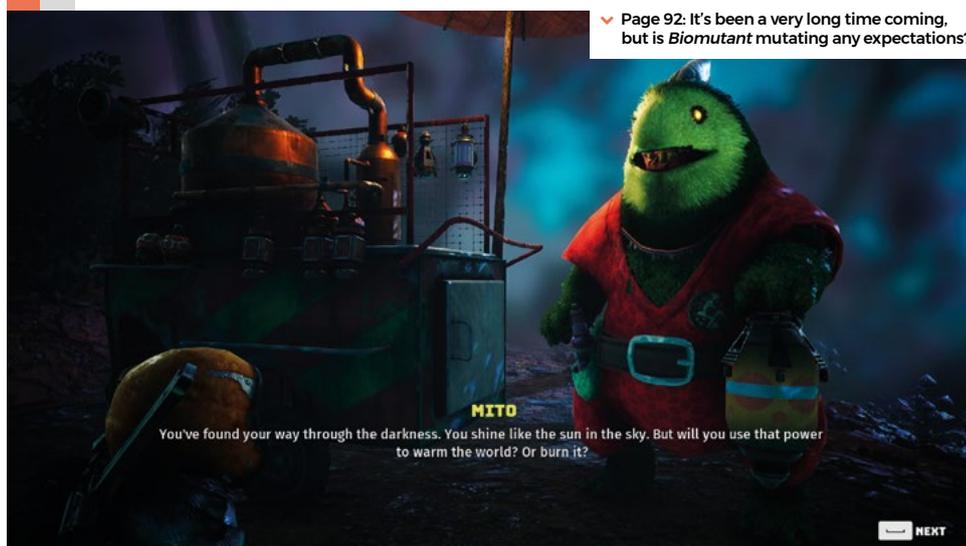
Backend Contents

Reviews, retro games, and lots more besides

- 92. Biomutant** PC, PS4, XBO
- 94. Wonder Boy – Asha in Monster World** PC, PS4, Switch
- 95. Knockout City** PC, PS5, PS4, XB S/X, XBO, Switch
- 96. Operation Tango** PC, PS4, XBO, XB S/X, PS5
- 97. Stonefly** PC, PS5, XB S/X, PS4, XBO, Switch
- 98. Slipways** PC, MAC
- 99. The Magnificent Trufflepigs** PC, Switch
- 100. Necromunda: Hired Gun** PC, PS5, XB S/X, PS4, XBO

OUR SCORES

- 1-9**
Trash. Unplayable; a broken mess.
- 10-19**
A truly bad game, though not necessarily utterly broken.
- 20-29**
Still awful, but at a push could be fun for two minutes.
- 30-39**
Might have a redeeming feature, but otherwise very poor.
- 40-49**
Adds in more redeeming features, but still not worth your time.
- 50-59**
Average. Decent at best. 'Just about OK'.
- 60-69**
Held back by glitches, bugs, or a lack of originality, but can be good fun.
- 70-79**
A very good game, but one lacking spit and polish or uniqueness.
- 80-89**
Brilliant. Fabulous fun. Everyone should at least try it.
- 90-99**
Cutting edge, original, unique, and/or pushes the medium forward.
- 100**
Never say never, eh?



Page 92: It's been a very long time coming, but is *Biomutant* mutating any expectations?

PLUS

- 101. Stream of Consciousness**
Demystifying the world of streaming by meeting some of its top names
- 102. The HOTLIST**
The PC's very best games, according to us here humble folk at Wireframe
- 104. Backwards compatible**
Celebrating the N64's quarter century and more retro stuff to make you feel old
- 108. Now playing**
The games that have been taking up at least some of our time this month



Page 104: Looking back at some of Nintendo's most surprisingly violent games and more.



Review

Biomutant

Info

GENRE

Action adventure

FORMAT

PS4 / XBO /
PC (tested)

DEVELOPER

Experiment 101

PUBLISHER

THQ Nordic

PRICE

£54.99

RELEASE

Out now

REVIEWED BY
Nic Reuben

Bio-Mute-it

I can't hate *Biomutant*. It's too lovely to look at, too earnest to be truly cynical about. Still, I don't think I've ever played a game that would benefit so strongly from having all spoken and written dialogue completely removed from it. For a game so ostensibly concerned with the environment that it barely lets you turn a corner without spouting platitudes at you, it seems utterly averse to letting its world breathe.

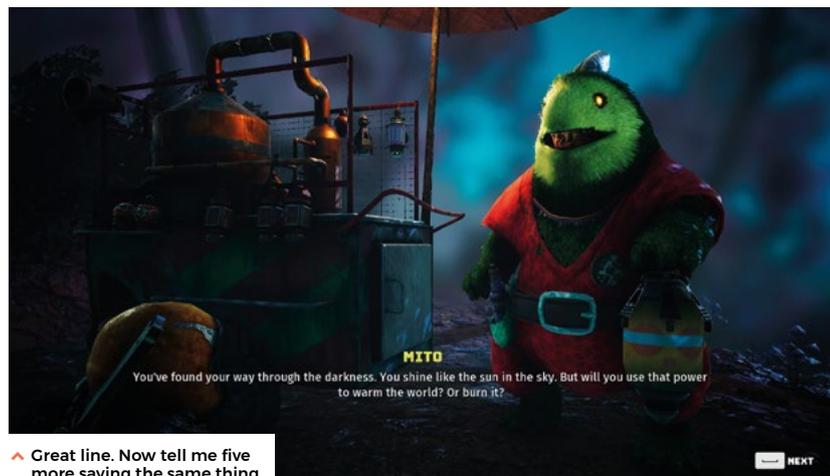
It's too gorgeous a world to drown out, too. Even the creatures trying to kill you are so charming that it's hard to criticise the game at all for fear of seeing a look of disappointment on

their stupid, adorable faces. Life is full of lessons, though. So kudos to *Biomutant* for teaching me that I could learn to hate a dachshund puppy if it explained the difference between right and wrong to me every time I went to feed it.

Lots of great games have bad writing, and I think that's something most of us have made our peace with. That *Biomutant*'s identity hinges so much on the strength of its storytelling is the fault of combat and exploration that could be enjoyable in service of a greater whole, but is too inconsistent to carry a whole game. Think *Arkham*, sans tightness, and *Devil May Cry*, sans freedom and fluidity.

It's passable, and sometimes fun. An oversized metal rocket-glove called the Klunkfist makes one helluva satisfying clank when it connects, and a fighting staff lets you twirl and lunge like a bin-scavenging ballerina. Animations are often exciting, even if largely limp sound design robs them of weight. But with so many weapons and different enemy types, along with inconsistent parry and block tells, it's a dice throw as to whether it feels good.

The art design and colour palette do a damn-near heroic job of making exploration worthwhile for its own sake, but even this is soured by a refusal to let you engage with the world on your own terms. "There's so much more to see now," says the narrator, ambiently, as I explore. But I know that, old mate. I can see it myself. The world is bright and beautiful enough, the music

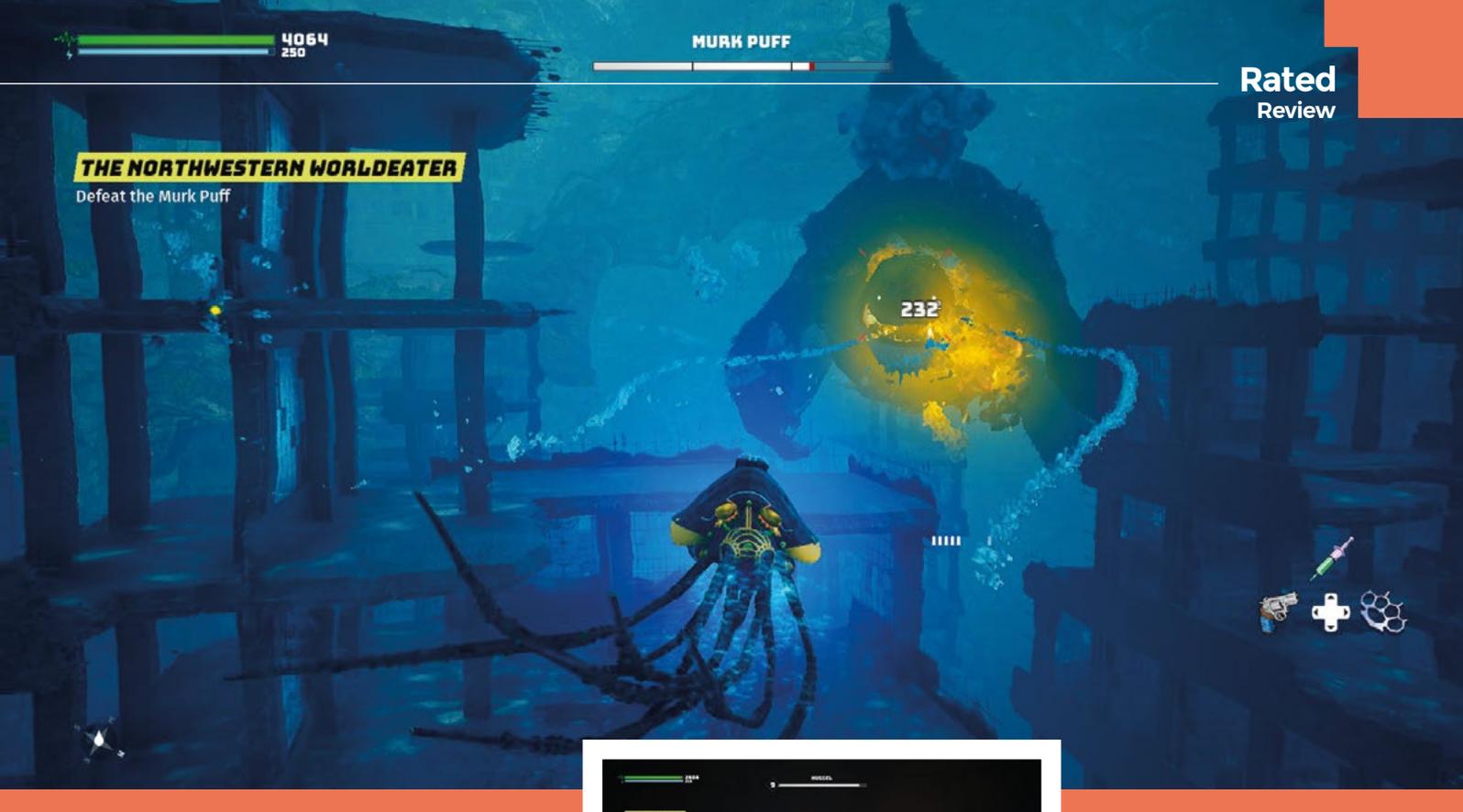


Great line. Now tell me five more saying the same thing.

NEXT

THE NORTHWESTERN WORLDEATER

Defeat the Murk Puff



beckoning and promising enough, that I can feel these things without being told to.

During the writing of this review, a patch was released that lets you turn off the narrator entirely. Previously, conversations lasted twice as long as they needed to. You'd get spoken to in charming gibberish, then the narrator would translate. Now, you can just opt to have gibberish and subtitles, or just the narrator. It's a welcome change, but it doesn't fix the fact that up to five sentences are regularly used to say what could have been accomplished in just one. I don't believe brevity should always be a hard and fast rule, but so few speakers in *Biomutant*'s world are interesting enough as characters to justify these monologues.

Most egregious of all is a morality system that Peter Molyneux would find quaint, with consequences so limited they make *Fable* look like *Planescape: Torment*. Colour-coded sprites appear and argue with each other about dark and light every time you make a moral decision. It's excruciating. I've heard the argument that this is a consequence of *Biomutant* being designed for kids, but being patronised sounds the same whether you're five or 50. Dr. Seuss, Roald Dahl... heck, I was ten when I played *Abe's Oddysee*. Morality tales don't have to be patronising. Please let your world speak for itself. I have never felt the tired writing mantra 'Show, don't tell' as viscerally as I did playing this.

I don't think *Biomutant* is a bad game, though – just an ill-considered one. It's still satisfying to absent-mindedly run about the world, solving

simple puzzles and collecting loot. Even the sprint animation made me grin, making your furry little weirdo bound across the plains on all fours. You can ride on a clockwork hand that shoots rockets from its fingers, a mech that shoots squirrels, and you unlock fast travel points by peeing on them. One mission had me stealing a giant toothbrush from a billboard so I could scrub clean a giant cat-ogre-mutant thing that had got itself trapped in an oil spill. I loved this stuff. It kept me pushing forward, hoping to find more of it, more charming little moments that understood the difference between child-like and childish. Occasionally, I did. Mostly, I just looted chests, hit things, and got talked at.

There's likely a few weekends of fun here, for the right person, with the right podcast on in the background. Something to shore up the gaps and lulls, the empty hills, the dull tunnels, the same enemy camps we've all been clearing for a decade now. Parents, too, might want to take a closer look, especially now the narration issues have been patched. Otherwise, I'm left despairing at the industry trends that made the developers feel they had to turn such a strange and wonderful premise into such an unremarkable game to make it profitable. 

▲ Visually impressive boss fights are let down by dull mechanics.

◀ It's a huggel, obviously.

HIGHLIGHT

The first encounter with each new creature feels like turning a new page in a wonderfully odd coffee table art book, and each new gadget and gizmo is a real delight to discover.

VERDICT

Charming visual design and undeniable heart makes *Biomutant* hard to hate, but inconsistent quality and baffling writing choices make it hard to recommend, either.

60%

Rated
Review

Estaf Plateau

9 / 10

9 / 10

3 / 13

1 / 33

HIGHLIGHT

Asha is unable to perform certain basic movements – like gliding, floating, and double jumping – without her trusty Pepelogoo. It's a cool twist to the platforming format that could have resulted in some unique puzzle design, yet it's never fully capitalised on and often hinders your traversal more than it helps.



Review

Wonder Boy: Asha in Monster World

Not quite a wonderful world

Info

GENRE

2D action-platformer

FORMAT

Switch (tested) / PC / PS4

DEVELOPER

Monkey Craft

PUBLISHER

STUDIOARTDINK

PRICE

£31.49

RELEASE

Out now

REVIEWED BY

Aaron Potter

VERDICT

A staunch remake almost to a fault, cutesy visuals and animations only distract from old-school design issues for so long.

46%

Few protagonists get as excited over finding a simple item chest as this green-haired heroine. Always saddling up to them with swagger, rocking her hips side to side in a jig... all this over what's sure to be a few coins (again). I wish I could share in the joy. Because as charming as these touches are, they're also indicative of the simplicity rampant in this ground-up remake of 1994's *Monster World IV*. This action-platformer is harmless fun if you can get past the lack of innovation, sure, but rarely does Asha's revitalised adventure live up to contemporary standards.

The only real attempt to modernise is in the graphical department. Where most other recent *Wonder Boy* remakes remain intent on the flat look, *Asha in Monster World* at least does something different by opting for a cell-shaded 2.5D style. In most indie titles there's a risk this design choice could look cheap, but here it

does well to blend the past and present together, lending the four dungeons you explore an added dynamism. Colourful locations like ice temples, sky-high mountains, and volcanoes always feel alive.

Sadly, it's never long until the illusion is broken, let down by stiff character animations, boring combat, and an extremely rote narrative. It's in all these areas that *Monster World* constantly reminds you it's a remake first and foremost, and not in a good way – refusing to move on from the

original game's blueprint and update important features. Clearly the intention here was for developer Monkey Craft to stay as slavish as possible outside of aesthetics, leaving anyone who isn't a die-hard fan of the series underwhelmed.

Asha's goal is to rescue four sprites scattered throughout the kingdom in the hopes of preventing the oncoming darkness. To do this, she's joined by her cute flying friend Pepelogoo, who can be called upon to melt ice blocks and prevent arrows from hitting you, as well as gifting you with other traversal manoeuvres. Most of these abilities are only ever necessary during specific instances within stages, though, and having to call him over just to double jump is annoying. It's a shame his inclusion isn't handled better because Pepelogoo is one of the main differentiators that separates *Monster World* from other games of its type.

Every new setting you visit offers a feast for the eyes, but that does little to help *Asha in Monster World* from feeling like anything more than an old-school relic with lots of missed potential. It's not an outright travesty or broken as far as remakes go. However, it does little to win over new players craving a challenge or genuine creativity from their side-scrolling platformers. Jig all you like, Asha – you're left without a dance partner on this occasion. 🗨

“It's harmless fun if you can get past the lack of innovation”

Rated
Review

HIGHLIGHT

With cross-platform play and a Friend Pass system, pulling a pal in to play *Operation: Tango* is as easy as sending an invite. There's certainly scope to play this through with multiple friends, too, especially if you insist on swapping between Agent and Hacker, depending on what each player has previously experienced.

Review

Operation: Tango

It takes two to tangle with tricky spy missions

Info

GENRE

Co-op

FORMAT

PC / XBO / PS4 (tested)

DEVELOPER

Clever Plays

PUBLISHER

Clever Plays

PRICE

£13.99

RELEASE

Out now

REVIEWED BY

Jordan Oloman

VERDICT

Operation: Tango's chaotic co-op is well worth your time, even if it's a little rough around the edges.

74%

A co-op game lives and dies by how engaging it is. You want it to force you to communicate, create tension, and usher in the occasional emotional outburst as you work together towards a greater goal.

Operation: Tango is a spy thriller that ticks all of these boxes, but not always for the right reasons. Over seven missions, you and your co-op partner can choose to play as an agent or a hacker. The hacker is typically trapped in cyberspace but has plenty of agency, altering and opening up the real world where the agent roams, ticking off mission objectives in physical space.

One great mission sees the agent scoping a busy subway train. The hacker has access to its passengers' apps and IP addresses and must use this information to direct the agent to a specific commuter who's been planted with essential data.

Throw timers, simultaneous hacking puzzles, looming threats, and door controls into the mix, and you've got pure co-operative chaos. By the end of the mission, we were defusing an explosive and barking orders as the timer ticked down to single digits. It was exhilarating and made stronger by the fact that the game is so challenging. It's not afraid to let you fail, unlike many other co-op experiences.

Missions tend to grow in complexity and intrigue as you progress, leading to virtual reality rhythm games and *Mission: Impossible*-style elevator shaft descents. You'd think cutting the

correct wire would be played out by now, but *Operation: Tango* loves to twist these precarious hero moments by limiting a player's senses and constantly warping the gameplay in imaginative ways, *WarioWare* style.

The 3D segments add character, but it's clear the developers are more confident in 2D space, with the sharp-lined UI and vibrant cartoon art outshining simple 3D models and environments.

Even the 3D mechanics, like a clunky EMP grenade and a dodgy remote-controlled Roomba, feel hamstrung. The 2D subterfuge is consistently intuitive, but there's one frustrating mechanic called the Password Checker. You have a four-digit code to figure out, made of ones and zeros. While one person guesses, the other person relays what digits were wrong and out of position.

It's trial and error that forces players to repeat themselves over voice chatter until they crack it – and it drove us both mad.

Regardless, it's easy to look past some of the game's poorly thought-out spy mechanics, as they're short dips within varied and fun missions. The bugs were trickier to overlook. The worst one soured the final act – a visual issue where we didn't see the same puzzles on our screens. This appeared multiple times and made it impossible for us to collaborate, forcing us to restart long missions.

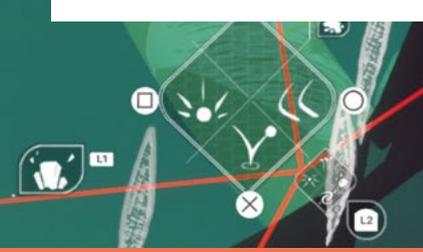
Disruption aside, I still left *Operation: Tango* thrilled and excited for more after its campaign ended. Ultimately, there are few games like it, and it's easily one of the best of its kind. 



Rated
Review

HIGHLIGHT

The biggest resource hauls come from monstrous alpha aphids that sporadically burst from underground, caked in gems. Of course, the precious goods attract all the bugs to the yard, so like a Black Friday sale, it's a hectic dash to grab as much as you can, without getting barged aside.



Review

Stonefly

Bug-ridden

Stonefly is further evidence that Flight School Studio isn't short of original ideas. Like 2019's *Creature in the Well*, what looks routine at a glance is actually enticingly different, with a distinct aesthetic to boot. Yet beyond its premise, *Creature in the Well* was ultimately rather skeletal, flattering to deceive, and *Stonefly* doesn't exactly break the pattern. To muddle the metaphor, while it does have more meat on its bones, it's still a little undercooked.

Stonefly's originality springs from a world where humans are tiny (or everything else is massive – same difference), making shelter on rocks and branches, and zipping about in gliding machines modelled on the giant bugs around them. As Annika, a young mechanic, you're hunting the thief of your father's prized rig, piloting an abandoned old vehicle that you'll fix up along the way. But upgrades require minerals, and minerals attract insects, so you'll have to fight for your share.

It really is a good idea, and there's simple joy in weaving your metal bug between twisting brambles or launching from leaf to leaf in the forest canopy. A sense of serenity oozes through pastel crayon shapes and mellow beats into the undulating hop and glide of your rig, a blend of cranefly knees and ladybird wings, light and laconic enough to coast with the will of the breeze.

Defence, meanwhile, involves toppling a litany of pushy beetles onto their backs with ground-rattling bombs then generating a gust of wind to waft them off the edge of the leafy battlefield. It's a method that's tactile in its familiarity – like



Return to camp between forays to speak to friends, shop for parts, fix up your rig, or take a quick nap.

humane scooping up a bedroom spider on a coaster then blowing it out the upstairs window. It'll be fine. Probably. Here, there's a colonial guilt attached, kicking the locals off their patch to mine minerals, but that's quickly repressed among the positional jockeying of bombing and gusting, luring fatter opponents close to the edges, while trying not to get biffed by a charging stag.

The intensity is sporadic, however, as, like Annika's craft, *Stonefly* has a habit of drifting vaguely. The autumnal scenery is striking but light on meaningful composition – between skirmishes the main challenges are obscured camera views and distinguishing solid landing points from ethereal set decoration – while featureless and flat arenas ensure the insect turf wars eventually lose their lustre. Complete Annika's story and there's a neat pay-off about what we value and, appropriately under the circumstances, not seeing the wood for the trees. But much of the lead-up feels loosely sewn together, only occasionally condensing into a captivating nugget. Then the second half wastes too much time on loops of resource grinding to pass arbitrary toll-gates.

Such repetitive labours in a game of dreamy exploration finally confirm the sense of an idea that doesn't quite know what to do with itself. Still, it says something that Flight School have again not quite found a winning formula, yet leave us keen to see what's next. 🐛

Info

GENRE

Action adventure

FORMAT

PS4/5 (tested) / PC
/ Switch / XBO /
XB S/X

DEVELOPER

Flight School Studio

PUBLISHER

MWM Interactive

PRICE

£17.99

RELEASE

Out now

REVIEWED BY

Jon Bailes

VERDICT

Inventive and visually lush, *Stonefly* lacks the depth to avoid wearying repetition.

63%

HIGHLIGHT

It's the pulsating probe mini-game. When you're aiming where to send probes, their scanning range delicately pulses, meaning if you time it right, you can discover one or two extra planets. It adds a subtle little skill check into things – and if you can't be bothered? Just turn it off in the options.

Review

Slipways

In space, no one can hear you screen (for new trade routes)

Info

GENRE
Probe-'em-up

FORMAT
PC, Mac

DEVELOPER
Beetlewing

PUBLISHER
Beetlewing

PRICE
£13.99

RELEASE
Out now

SOCIAL
@SlipwaysGame

REVIEWED BY
Ian Dransfield

VERDICT

A focused, delightful puzzle game masquerading as space empire-based grand strategy.

90%

W

e went over it in our preview in issue 49, but it bears repeating: *Slipways* is not a 4X game of galaxy-spanning strategy.

No, *Slipways* is a puzzle game.

It tells you you're creating a vast intergalactic empire generating supply and meeting demand, and it has the trappings of alien races, planetary systems to probe, and new technologies to research. But the game boils down to one key aspect: connecting your planets to one another.

You do this using slipways: space tubes, if you will, comprising a straight line between two colonised planets. Goods can travel one or both directions depending on the connected planets' needs or what they produce. Planet A produces food but needs people to make it; Planet B produces people but needs food to make them. One slipway later, problem solved. And from there it builds. Needs change and secondary factors come into play – more supply means more demand – and soon your simple, small-scale galactic empire is forced to expand to meet these new needs. Would you believe it, that's where the real game begins.

Your initial, furtive expansion sees planets popping into existence as you probe probable signals – a minigame in itself – with expanded needs on the original home planets being met by the provisions of the new. But then the new planets have needs of their own, and so on, until you have a civilisation spanning hundreds of planets and just as many spacefaring

thoroughfares. Bar some late-game tech you can unlock, routes can only be in a straight line, as noted, and can't cross over one another. Suddenly sounds challenging, doesn't it?

Slipways is superb: the core concept carried off with efficient, smart design, a tutorial system to ease in new players, and the shorter nature of each playthrough meaning it's a title that, while it engages your brain to work out the best routes possible, doesn't eat up all of your free time. Except... well, it sort of does, because once its teeth are sunk in, there's no escaping its lure.

Galaxies are randomly rolled when you start, and while the majority of the time you get a chunk of space that's balanced and something you can work within (and around) to get your routes set up perfectly, *sometimes* you're just sold a pup. In a game where runs lasted hours – even days – this would be unforgivable.

In *Slipways*, where a run is usually about half an hour? Get frustrated, start a new game, forget about the last one, have fun again. That's the only concern of note I could think of here.

Slipways absolutely nails what it's aiming for. It's a tight, focused puzzle game that manages to both challenge the grey matter and also offer a much-needed respite from the stress of everyday existence. It's only going to improve as developer Jakub Wasilewski builds on what he's been working on for a long time, but even here in the moment after *Slipways*' initial release, we've already got a *great* game on our hands. 🐛

“Once your teeth are sunk in, there's no escaping its lure”



HIGHLIGHT

The absence of animated character models doesn't undercut the drama as much as it might. This game sometimes feels like a radio drama, soundly performed by Luci Fish as Beth and Arthur Darvill as Adam.

Review

The Magnificent Trufflepigs

Playing the field

This quiet metal-detecting game set in pastoral England starts with a phone call. Stanning Farm has just been sold. Before the new owners arrive, Beth can't resist the chance to drop the busywork weighing her down and venture back onto the fields where she once discovered some precious jewellery. First, she needs a hand, and old friend Adam picks up. The questions first posed in Beth's call fuel the narrative engine of *The Magnificent Trufflepigs*. What's so special about this jewellery? Where's Beth's fiancé? And what's Beth looking for, really?

It's a small engine, admittedly. The story of this self-described "evening-sized" game unfolds over walkie-talkies and text messages while we, playing Adam, patrol various fields. We have a slim toolset: a spade, a trowel, and a metal detector. Standing in our way is grass. Simply sweep the detector with its beeping and flashing indicators to locate finds, and the game prompts us to dig, rotate, and photograph. What we'll dig up won't be turning up on dodgy eBay listings. Instead, they're personal, each one prompting a gobblet of

conversation. A locket will get Beth talking about her shirking partner. A tent peg will remind her of a childhood spent outdoors. Gradually, they'll needle Beth's understanding of herself.

These are the game's two gears: detecting and listening. It doesn't shift from this pattern. This is OK. Good, in fact, because the person who planted this debris is Andrew Crawshaw, lead designer of *Everybody's Gone to the Rapture*, with which *Trufflepigs* shares an enchanting sense of place. The first time the game lets us walk around, it's tempting to just stand there, listening to the music and the birds and the rustling grass.

When we do grasp the detector, there isn't much in the way of deduction. Minimally interactive, it lacks skill-based elements that might make it more satisfying. More than anything, *The Magnificent Trufflepigs* is a sympathetic vignette of Beth's disaffection with rural life. We see her negotiating the inheritance of the family business, her mangled hopes with her fiancé, and her confused relationship with Adam. The development here is all Beth's, whose story beats take place out of frame. Conspicuously missing from *Trufflepigs'* story is Adam, through whose eyes and ears we experience the whole thing. We're left feeling a bit redundant, with the meaning of their relationship largely unexcavated.

A narrative game where you walk around with a metal detector? I'm still attached to the premise, although in practice, it wears thin. Yet for these grumbles, *The Magnificent Trufflepigs* also welcomes us into a fold of slightly imperfect British landscape, happy for us to exist there for a while. That might be what we've come looking for. 🐷

Info

- GENRE**
First-person detector
- FORMAT**
PC (tested) / Switch
- DEVELOPER**
Thunkd
- PUBLISHER**
AMC Games
- PRICE**
£9.29
- RELEASE**
Out now

REVIEWED BY
Kyle Hoekstra

VERDICT

The short story inside this metal-detecting game needs fresh batteries.

65%



Rated
Review

4x Ammobox

52.22 FPS
19.91 FPS

HIGHLIGHT

A literal highlight comes from summoning your beefy mastiff with the squeeze of a squeaky dog toy, as his powerful nose outlines enemies in red so you can see what's going on. He's useful in combat, too, and can be kitted out with cybernetics to clamp steel jaws around your opponents. Good boy.

Review

[L. Shift] ② ↑
[Q] 5 | ①
97 | 500

Necromunda: Hired Gun

A rusted development

A little past the halfway point of *Necromunda: Hired Gun*, something's on our mind. We're facing a boss, both of us armed with a chain gun, and both of us forgo cover to stand in the open, hammering hundreds of bullets into each other's torsos, until he finally drops dead. Anticipating the outcome, we take a moment to reflect: "Is this really how this game is meant to be played?" Honestly, for all the RPG upgrades and intricate level design in *Hired Gun*, it seems it really is.

The levels really are intricate, though, at times almost spectacularly so. Stream On Studio's latest excursion into *Warhammer 40k* territory truly respects the aesthetic and lore of its source material. The industrial gothic squalor of its Underhive is lovingly crafted from rust and feats of metallic engineering – a living mass of ancient mining operations, heavy manufacturing, giant railways, and pillars crowned with iron skulls that loom in judgement. The nests of pipelines and girders aren't mere background detail either, with their rocky verticality and hidden nooks where local outlaws stash their valuables. A veritable adventure playground for your tough bounty-hunter, fully equipped with cybernetic implants and a no-nonsense Yorkshire accent.

Add in slick control and weapons that come with a healthy kick, and the stage is set for something as appealing as a mid-budget *DOOM*. But it rarely quite works like that. The enemy are one reason – the distinct traits of factions

muted by genre limitations and production-line spawning, as hundreds of 3D-printed clones, camouflaged by their own insignificance, spray you with whatever until you plant your sight on them and squeeze. There are a few exceptions – teleporting witches, bulky Ambots – but mostly you don't know who you're shooting and it barely seems to matter.

The other reason is that deploying skill or smarts in fights is less efficient than simply charging towards opponents, blasting. The incessant rain of fire from goons dovetails with a hyperactive gain back mechanic that refills your health whenever you do damage, undercutting much need for things like evasive manoeuvres, target selection, or special powers. The only thing better than simply shooting, in fact, is your melee attack – there's no downside to getting amongst a mob and despatching each in turn, pressing a button to trigger a (needlessly long) kill animation, during which you're invincible.

There are high points. A couple of the levels slow or spread out enough to make it worth altering your approach, and the scale and shape of the place remains impressively varied. Yet most grand locales wind up merely hosting a drunken rumble, where as long as you can recharge your health slightly quicker than they drain it, you'll be just fine. Add to that some minor technical messiness – the joys of accidentally restarting a mission because the wrong menu item is highlighted – and the result is the mindless chaos of chain gunning a bullet-sponge boss to paste. 🗨

Info

GENRE
FPS

FORMAT
PC (tested) /
PS4 / PS5 /
XBO / XB S/X

DEVELOPER
Stream On Studio

PUBLISHER
Focus Home
Interactive

PRICE
£34.99

RELEASE
Out now

REVIEWED BY
Jon Bailes

VERDICT

For all its grand designs, this hired gun too often shoots itself in the foot.

55%

▶ STREAM OF CONSCIOUSNESS

This month it's **Trista Bytes** in the hot seat

What's your favourite game?

I always find that question so hard! If we were going on hours played, it would be *Quake III Arena* or *Phantasy Star III*. The game with the biggest impact though was *Sonic the Hedgehog*.

And why is that?

Sonic was a game-changer. I was in love with the Master System and very much a Sega kid, but I played *Sonic* on a Mega Drive and felt like I was seeing the future of gaming – Sonic was so fast, bright, and exhilarating compared to a certain plumber I played with my friends on their NES (Sorry, Mario fans).

What game got you into gaming?

It wasn't one game but the entire concept of gaming itself. I remember discovering computer games at friends' houses; the idea you could load a cassette up then make the pixels on-screen move yourself was like magic to me. I wanted to play anything, and I could from that day on. I got to enjoy a huge range of games on Speccy, C64, Atari, and Amiga at other people's houses before we got a Master System.

Have you ever been put off gaming?

I've never gotten into online gaming with random people. I always played people I knew, even online. I saw *WOW* launch and knew immediately to ban myself if I wanted to get my degree! I don't play *Fortnite* or any of the big online games nowadays, either. Originally this was due to them eating time away from work; nowadays, it's as much about the toxicity against women in gaming and the industry. I haven't been put off gaming, but I certainly choose my areas where I'm attacked less. I mostly play solo games with my audience backseating rather than going online and near any PvP games. It's a real shame that it's still such a big issue. I remember being a kid in the eighties and nobody thought gaming was for boys – yet

nowadays, I see people acting as if women and girls gaming is new. I can only hope it's something that changes for the better.

Why play games for an audience?

I came to streaming late as I didn't think people would want to hang out with me and chat about stuff with all my over-enthusiasm, tendency to wander onto tangents... but it turns out that what used to be my flaws can be a positive in streaming. I love entertaining people and sharing knowledge and making them smile. Things grew from there; the silly character voices I put on became improv comedy characters with running gags. Because I often get enthusiastic about a topic when a viewer reminds me of something, I also do entire streams looking through old collectables I have, or reliving memories in old Argos catalogues. For me, it's all about sharing passions and having a break from the real world. Making a career out of making people laugh and sharing my love of gaming would be a dream come true if I succeeded. 🐾



“I remember being a kid in the eighties and nobody thought gaming was for boys”

Trista Bytes streams full-time, five days a week, over on Twitch: [wfmag.cc/Trista](https://www.twitch.tv/wfmag/cc/Trista)



The Wireframe **HOTLIST**

The best PC games, according to Wireframe, catering for whatever your mood might be

The games for... **BIG ADVENTURES**

Assassin's Creed Odyssey / *Ubisoft* / **93%** (Issue 1)

Yakuza: Like a Dragon / *Ryu Ga Gotoku Studio* / **90%** (Issue 45)

Amnesia: Rebirth / *Frictional Games* / **87%** (Issue 46)

The Last Campfire / *Hello Games* / **86%** (Issue 47)

Resident Evil 2 / *Capcom* / **86%** (Issue 7)

Far Cry New Dawn / *Ubisoft* / **85%** (Issue 9)

Journey to the Savage Planet / *Typhoon Studios* / **84%** (Issue 33)

The Outer Worlds / *Obsidian Entertainment* / **84%** (Issue 28)

Monster Boy and the Cursed Kingdom

/ *Game Atelier* / **84%** (Issue 6)

Nioh 2 / *Koei Tecmo Games* / **80%** (Issue 38)



The games for... **REPEATED PLAY**

Hades / *Supergiant Games* / **94%** (Issue 44)

They Are Billions / *Numantian Games* / **88%** (Issue 20)

Sekiro: Shadows Die Twice / *FromSoftware* / **87%** (Issue 11)

Streets of Rage 4 / *DotEmu/Lizardcube/Guard Crush* / **86%** (Issue 40)

Trials of Fire / *Whatboy Games* / **84%** (Issue 50)

Katamari Damacy REROLL / *Monkeycraft* / **84%** (Issue 4)

Spelunky 2 / *Mossmouth* / **83%** (Issue 44)

Hitman 2 / *IO Interactive* / **82%** (Issue 3)

Alba: A Wildlife Adventure / *ustwo Games* / **82%** (Issue 46)

Slay the Spire / *Mega Crit Games* / **81%** (Issue 45)



The games for... **SOLID STORY TIMES**

Disco Elysium / *ZA/UM* / **94%** (Issue 28)

Mutazione / *Die Gute Fabrik* / **86%** (Issue 26)

Whispers of a Machine

/ *Clifftop Games/Faravid Interactive* / **85%** (Issue 14)

Mythic Ocean / *Paralune* / **84%** (Issue 36)

Sunless Skies / *Failbetter Games* / **83%** (Issue 7)

Arise: A Simple Story / *Piccolo Studio* / **82%** (Issue 31)

Assemble with Care / *ustwo Games* / **81%** (Issue 27)

The Walking Dead: The Final Season

/ *Telltale Games/Skybound Games* / **81%** (Issue 11)

The Procession to Calvary / *Joe Richardson* / **80%** (Issue 40)

Outer Wilds / *Mobius Digital* / **80%** (Issue 17)

The games for... **FIRING UP BRAIN CELLS**

Telling Lies / *Sam Barlow* / **92%** (Issue 24)

Kentucky Route Zero / *Cardboard Computer* / **90%** (Issue 33)

Heaven's Vault / *inkle* / **89%** (Issue 12)

The Pedestrian / *Skookum Arts* / **84%** (Issue 35)

The Legend of Bum-Bo / *Edmund McMillen* / **83%** (Issue 31)

A Monster's Expedition / *Draknek & Friends* / **82%** (Issue 47)

Total War: Three Kingdoms

/ *Creative Assembly/Feral Interactive* / **82%** (Issue 16)

It Takes Two / *Hazelight Studios* / **81%** (Issue 51)

Wanna Survive / *PINIX* / **80%** (Issue 42)

Superliminal / *Pillow Castle* / **80%** (Issue 34)

The games for... **HIGH-INTENSITY PLAY**



Tetris Effect / *Monstars Inc./Resonair* / **90%** (Issue 4)

Sayonara Wild Hearts / *Simogo* / **89%** (Issue 25)

Star Wars: Squadrons / *EA* / **86%** (Issue 45)

Devil May Cry 5 / *Capcom* / **84%** (Issue 10)

Black Bird / *Onion Games* / **84%** (Issue 3)

BPM: Bullets Per Minute / *Awe Interactive* / **83%** (Issue 45)

Resident Evil Village / *Capcom* / **82%** (Issue 52)

Catastronauts / *Inertia Game Studios* / **82%** (Issue 1)

Olija / *Skeleton Crew Studio/Thomas Olsson* / **81%** (Issue 48)

DUSK / *David Szymanski* / **81%** (Issue 7)

The games for... **CURING THE INDIE ITCH**



If Found... / *DREAMFEEL* / **92%** (Issue 44)

Can Androids Pray
/ *Natalie Clayton/Priscilla Snow/Xavier Nelson Jr.* / **90%** (Issue 21)

Tales From Off-Peak City Vol. 1 / *Cosmo D* / **89%** (Issue 39)

Baba Is You / *Hempuli Oy* / **88%** (Issue 10)

Afterparty / *Night School Studio* / **86%** (Issue 33)

Witcheye / *Moon Kid* / **86%** (Issue 30)

Hypnospace Outlaw
/ *Tendershoot/Michael Lasch/ThatWhichIs Media* / **86%** (Issue 11)

Haunted PS1 Demo Disc / *The Haunted* / **85%** (Issue 39)

Xeno Crisis / *Bitmap Bureau* / **81%** (Issue 33)

Art of Rally / *Funselektor Labs* / **80%** (Issue 45)

PC Top 10



1 Disco Elysium / **94%** (Issue 28)

Smarter and deeper than anything else; truly an RPG in a class completely of its own.

2 Hades / **94%** (Issue 44)

Proving 'roguelike' isn't a dirty word, learning-and-dying is a joy from start to finish.

3 Assassin's Creed Odyssey
/ **93%** (Issue 1)

The point where Ubisoft realised over-the-top adventures were the right direction.

4 Telling Lies / **92%** (Issue 24)

This FMV mystery asks more of the player than most, with rewards to match.

5 If Found / **92%** (Issue 44)

A compelling and beautifully illustrated narrative, as moving as it is memorable.

6 Yakuza: Like a Dragon / **90%** (Issue 45)

A bold, brash, and joyous rebirth for the long-running gangster series.

7 Tetris Effect / **90%** (Issue 4)

The question is 'how do you better *Tetris*?' The answer is: like this. This is how.

8 Kentucky Route Zero / **90%** (Issue 33)

Abstract style meets concrete commitments in this fantastic magical realist adventure.

9 Can Androids Pray / **90%** (Issue 21)

A healthy dose of existential anxiety in a minimalist, bite-sized package.

10 Tales From Off-Peak City Vol. 1
/ **89%** (Issue 39)

A walking sim/adventure; a work of remarkable imagination and humanity.

Backwards compatible

WRITTEN BY IAN 'CABLE GUY' DRANSFIELD

Nintendo 25

Nintendo's Nintendo 64 officially turned 25 a couple of months ago, celebrating the anniversary of its Japanese launch back in June. We went over the history of the machine and its impact on the wider sphere of gaming in Wireframe #47, but the N64's anniversary actually pushed me into doing something I've honestly not done a lot in my life: play some N64 games that aren't just *WWF No Mercy*. In fact, I opted for something rather wild, given Nintendo's reputation for family-friendly output – I homed in on some of the platform's most violent games.

Starting out with a whimper, because who wants to enter with a bang, I opted for *Bio F.R.E.A.K.S.* – Midway's 3D one-on-one fighting game that slots so effortlessly into that period of mid-to-late-nineties schlocky violent fighting games that it should officially be called the *Bio F.R.E.A.K.S.* Period.

Let's not waste space here: there's a reason it's not a game that's remembered fondly, or at all. While it doesn't look too bad and has a certain...



charm to proceedings, the action is banal, the fighting near-nonsensical, and while it's funny to see your limbs get shot off halfway through a fight, it's an unbalanced mess of a game. Next!

Because I hate myself, apparently, I moved on to *Carmageddon 64* (right). Curiosity often overwhelms common sense, meaning I went from a mediocre-to-bad fighting game straight into one of the worst games the N64 has to offer. Clumsy to control, a choppy frame rate, an empty atmosphere, and none of the kinetic thrill of the original series of idiotically fun racer/smashers. This one was turned off before I'd even finished a full race.

The nosedive in full effect, it was time to opt for something a bit different: *Resident Evil 2*. A personal favourite. I'd never actually played it on N64 at any point – my time with Capcom's classic restricted to either the PSone or GameCube versions. So it was with no small amount of shock that I saw just how *incredibly* well done a port *Resi 2* is on N64. This is a game heavy on cutscenes and recorded dialogue, playing on a console that uses cartridges and so doesn't have much storage when compared to CD. You can practically *feel* the compression coming out of this one, with that telltale tinniness to dialogue ringing through the ears... but really, it's not an issue. This is a spectacular technical achievement and a fine port of a wonderful survival horror game. Safe to say, I had a good time here.

And then, dear reader, I moved on to what I'd intended to play from the very beginning: *Body Harvest* (left). DMA Design's proto-*GTA III* offers an open



world to potter about in, vehicles to nab and drive, people to interact with and get missions from, a brutal and spooky alien invasion to avert, and some surprisingly *Zelda*-like aspects here and there. OK, so it's not violent in the way that there's viscera splattering all over the screen, but it has an unnerving violence about it – an omnipresent threat that normal people *cannot* avoid without your help; an oppression that oozes through the game's very being. It is implicit in its violence, rather than *Bio F.R.E.A.K.S.*-like. But that's beside the point: the point is, *Body Harvest* is something I never expected to be as much fun as it is, and I'm going to be heading back here to play through as much of the game as I can be bothered to. At least until it gets too hard, which I assume it will soon enough because it's a game from the nineties and that sort of thing happened all the time back then. Maybe *that's* the real violence... Happy birthday, Nintendo's Nintendo 64!

'Member me

Elsewhere in doomed projects I undertook recently, a thought crossed my mind: can you get improved Nintendo 64 Controller Paks? Well, yes, you can. But that's not where I ended up with the Datels and Electronics Boutiques and whatever other old brands made third-party peripherals – instead, I ended up back at the soldering station, original Nintendo-branded Controller Pak in hand and an array of delicious (electronic) chips to my side.

See, it turns out you're able to replace the RAM module inside the Controller Paks for the N64 – the original 256kB storage was battery backed up, meaning after the cell inside the housing expired,

you would lose the data. This module could be replaced with ferroelectric RAM (FRAM) – basically, the same size of chip, the same physical construction, the same capacity, but non-volatile, meaning it doesn't even need a battery to begin with.

It's a straight swap, like-for-like, and a fun little soldering project that doesn't need much expertise or an incredibly steady hand to upgrade some old kit. As such, it's definitely worth a go if you're keen on original hardware and want a bit of a mini-project to spend some time on. And I only broke three memory cards in the process, so at least I've got that going for me.



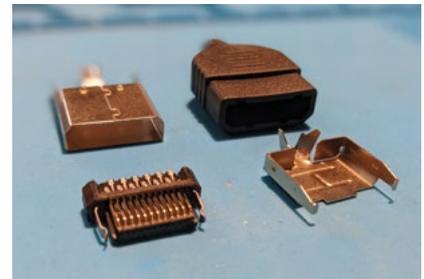
Component components

Someone tried playing their original Xbox recently, and that same someone was shocked to remember just how much electrical interference the console picks up if you use poorly shielded component cables with it. My original solution – a YPbPr-wired SCART lead – was no longer an option (I sold my OSSC since the cables play well with *nothing else*), so it had to be a component lead. So it was I fell down the new rabbit hole of getting an official Xbox

360 component cable, snipping the connector off it, and Frankensteining an original Xbox connector onto it. It's a bit more complex than that, but honestly – not that much more complex.

A fiddly task ensued, with much twisting of wires and cursing very loudly at the stupid ground wires which were doing a good job of emulating A Material You Cannot Successfully Solder. Progress was slow, and quite frankly *bad*. Inevitably, a catastrophic failure occurred thanks to impatience and heavy-handedness – so it's a good job past-lan had planned for this and picked up multiple cables and multiple original Xbox connectors; with the former coming in at £3 each and the latter £1.50 a pop, it proved an inexpensive series of attempts and failures.

Long story short, as much as I tried, I just couldn't do it. It's not difficult; it's just more fiddly than my ever-shaking hands can cope with. And so it was I did



the next best thing: I bought a pre-made cable from eBay for £25. It arrived, it was very well made, and it works absolutely spectacularly – I swear this original Xbox now outputs a better signal than it ever has before. Bright, colourful and – importantly – lacking in electronic interference, it's set me up to play through some gems from Microsoft's original big burly box. For some reason, the first game I tried was *Enter the Matrix*. Apparently I never learn to stop torturing myself. Cables!



Backwards compatible

WRITTEN BY RYAN 'BURNED FINGERS' LAMBIE

Tickled Pink, Part 2

Last month, I embarked on a mission to build the ultimate Game Boy Advance. I installed a brand new LCD screen, replaced the buttons, shell, and lens with the understated Kirby-themed parts you can see on these pages, and then added a nifty CleanJuice Battery Pack which allows me to charge the handheld up with a USB-C cable. But! I'm not finished yet. This month, I decided to start work on the sound; courtesy of RetroSix (retrosix.co.uk). I have a Wire Free Dehum Dehiss kit, and a CleanAmp Pro mod, both of which should offer a mightily improved audio quality over the stock GBA's hardware.

Installing the Dehiss kit is simplicity itself: it's essentially a single flexboard with a pair of chunky capacitors mounted on top, and sits neatly on the GBA's mainboard. Once the mod's correctly positioned – the pads need to be lined up so they touch the GBA's battery terminal and a handful of the handheld's capacitors – it's a case of carefully securing it to the relevant components with some neat blobs of solder. There are no extra wires to worry about, and the entire job can be done in a few minutes, assuming you're competent with a soldering iron and have a steady hand.

Next: the CleanAmp Pro mod. This is pretty straightforward, too: first, I needed to desolder and remove the GBA's original speaker – again, a process that takes a few seconds, since it's only held in place by a couple of thin wires. Once that's out of the way, the mod's installed in a similar way to the Dehiss board: I pre-tinned the pads with a bit of solder, gently positioned it in its intended location (just beneath the A and B buttons), and then used my soldering iron to connect the pads to the various pins and components as required. The new speaker, also provided in the kit, is then soldered directly onto the CleanAmp Pro flexboard – to make things even easier, it doesn't even matter which way around you solder the two speaker wires. Whichever way they're soldered to the mod, the whole thing will still work.

The final modification I wanted to make for my GBA was an LED kit, again from RetroSix. It's a purely cosmetic add-on that makes the handheld's shoulder buttons light up when pressed. To match my Kirby theme, I've got some oh-stylish pink LEDs. (It also goes without saying that you need clear shoulder buttons for this mod, otherwise nobody will see your glorious lights in action.)

The kit comes with the two LEDs, a pair of resistors, and a length of wire. The idea is to solder a wire from one pad on the GBA's shoulder button switch to

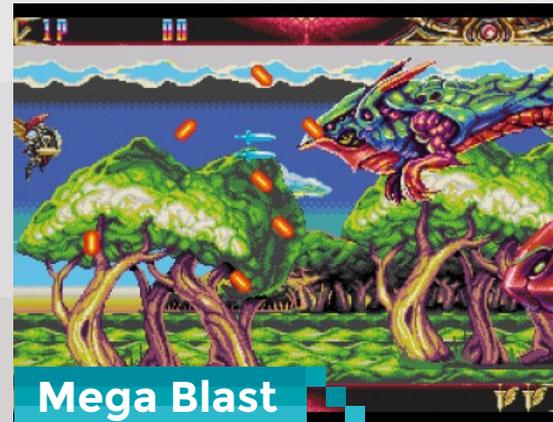




the short leg of the LED, a second wire from the LED's longer leg to a resistor, and then a third wire from the resistor to a pin on the cartridge connector. The process is then repeated for the other shoulder button. And it's here I that I ran out of talent: soldering onto boards is one thing, but soldering wires to tiny components, without anything useful like a Helping Hands station, made the process an absolute chore. After about an hour of swearing, though, I got my entire daisy chain of LEDs, wires, and resistors all installed. I secured my LEDs in place with a couple of self-adhesive pads, and used Kapton tape to strengthen the joints on my wires and prevent any electrical shorts. It wasn't a tidy job, though, and when I put the

console back together to test it – guess what – it didn't work.

Still, the sound mods do function nicely – the result is a much clearer and significantly louder audio than a stock GBA. I even dug out an unmodified Advance so I could compare the two, and there's a clear improvement. As for the pink LEDs – well, it's back to the drawing board on that front. I'm going to order a Helping Hands thingy, prise my GBA open again, and see if I fix what I'm certain is a dodgy bit of soldering on my part. Sure, the light-up shoulder pads are a bit of a novelty item, but I refuse to be beaten. Tune in next month to find out whether I finally get them working or I manage to burn my office down in the attempt...



Mega Blast

Xeno Crisis and *Tanglewood* gave the 33-year-old Sega Mega Drive some much-deserved new titles, and there's another promising game on the horizon for the 16-bit console: the anonymously titled *ZPF*. Developed by a three-strong indie team, it's a side-scrolling blaster in the mould of *Gate of Thunder* or the Mega Drive's *Gynoug*: you pilot a winged warrior in a pitched battle against an armada of grotesque monsters large and small. The action's time-honoured stuff, but the visuals look stunning so far: the assorted demonic heads and vaguely aquatic beasts are on a par with anything that emerged for the Mega Drive in the eighties and nineties. You can follow *ZPF*'s development at wfmag.cc/ZPF. If this one's getting a physical release on cartridge, I'll definitely be in line to snag a copy.





Havin' a brew

Break out the Dreamcast: Ian wants to play Xenocider

We looked at the bizarrely active Dreamcast homebrew scene a couple of years ago in the pages of Wireframe (issue 7, to be precise) – one title that stood out from the crowd was *Xenocider*. A fully 3D game made by amateur team Retro Sumus, riffing on the classic *Space Harrier* and – eventually – releasing in 2021, it's something a lot of people have been keen on for a long time. Well, 'a lot' in so far as the Dreamcast homebrew scene doesn't exactly have millions of card-carrying members, but you know what I mean.

Duly whet of appetite – after reading our feature of so long ago, naturally – I coolly and calmly set up my Dreamcast. It had been in a drawer for about two weeks, so it wasn't that hard to remember where everything went.

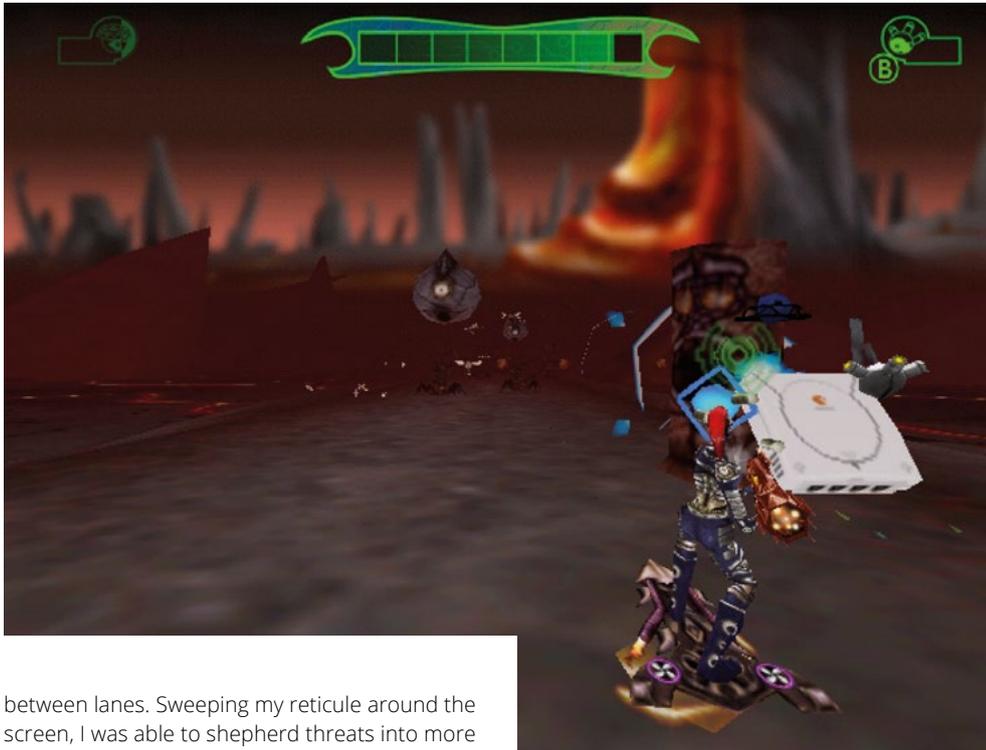


And to playing I did, indeed, get. It wasn't too long before I'd had my first game over, because *Xenocider* is a game that just doesn't want to play nice now, *does it?*

Expectations adjusted, my second play went much better, but there was still something off. The game is an on-rails shooter – like *Space Harrier*, of course, or *Rez* for a more recent example – in which you use the analogue stick to control your targeting reticule and try to blast the dozens of enemies, and final level boss, in each stage. Movement, at least that you have control over, is limited to switching left and right between five different 'tracks' – but that's more than enough to be able to avoid obstacles and incoming projectiles.

It sounds simple – you even set auto-fire on so you only have to worry about targeting, rather than targeting *and* shooting – but in practice, it needed a bit of time to punch any plasticity into my brain and allow it to get itself around the concepts. There's a tutorial covering the basics, but the fact is *Xenocider* is old-school in its approach: it relies on you playing and playing and playing to just get better at things, rather than giving you the world at your fingertips and all the help you could want.

That is to say: it's refreshing. Jarring initially, but refreshing. Soon enough I had the hang of it, especially after switching to my Retro Fighters Dreamcast controller with its (optional) digital shoulder buttons, which are used to switch



between lanes. Sweeping my reticule around the screen, I was able to shepherd threats into more manageable situations and bring some level of control to the chaos that unfolds in every planet-based level.

There was still the feeling, every now and then, that I was being cheated – projectiles heading on a (relatively) horizontal plane when you're sat in one of the outside lanes are, as far as I could tell, impossible to avoid. And some of the boss fights are frustrating as the big bads have the habit of recharging health along the way to prolong things. But the overall feeling with *Xenocider* was a faintly nostalgic joy.

There are impressive elements to it outside the main game – actual achievements are in there. Obviously not at a system level, it's just in the game, but that's a really neat touch. There are unlockables too, like an into-the-screen *Space Invaders*-style mode, and – brilliantly – a much more direct take on *Space Harrier* which sees *Xenocider* look, control, and, generally, *feel* pretty much like a modern (well, a 1998-ish) 3D version of *Space Harrier*.

As far as I'm concerned, *Xenocider* is up there with the absolute gold standard when it comes to homebrew. It's taken a while to come out, and that does happen a lot with these amateur productions, but you can see where a lot of that time went. If this had released around the 1999 mark, I don't think anybody would have been

“Xenocider is a game that just doesn't want to play nice now, does it?”

surprised. It might not have got the attention and plaudits it's been getting in 2021, mind you, but it would have fitted right in with the official, erm... *officebrew?* games of the Dreamcast's roster.

The most positive thing I have to say about *Xenocider* – aside from it making me laugh on finishing the first level and being congratulated for destroying 'one whole civilisation' – is that I wanted to play it more. I'd been intrigued simply because of the novelty of the subject: a manageable, but still ambitious project for the Dreamcast, 20 years after the console itself

had died. What's not to be curious about there?

But the reality is this is a solid game – solid both in design terms, and in difficulty terms – and even though I was playing it specifically to

write about my experiences with it here, I decided to go back to playing it some more once I'd done writing the previous few paragraphs (then secretly adding these in by the power of *magic*). It's not particularly deep or technical or clever, and it has reduced me to more than one bout of furious, dog-scaring swearing as that *arse* hermit-crab boss fluked a win against me for the fourth time. But for what it is, *Xenocider* is a great experience, and heartily recommended to all seven of you out there who still own a Dreamcast. 🐉

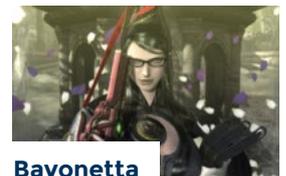
Wireframe Recommends



Space Harrier

MULTI

Doesn't hurt to go back to the OG, and *Space Harrier* is still as playable today as it's ever been. Fast, super-scaler-smooth, and offering a tough challenge, you can see why someone might want to emulate its style.



Bayonetta

PC, PS4, XBO, SWITCH, PS3, X360

Foolish? Of course not – *Bayonetta* features interstitial levels that riff on the likes of *After Burner* and, yep, *Space Harrier*. And they're so good they make one of the world's best games *even better*.



The Last of Us Part II

PS4

I haven't forgotten my promise to 'do' this one, and it will happen before the heat death of the universe. But for now, we have to live in the knowledge that I still haven't finished *The Last of Us Part II*, so it remains un-Now Playing-ed.



You've got to be kidding

Ryan braces himself for some old-school punishment in *Tanuki Justice* and *Alex Kidd in Miracle World DX*

The vast majority of video games prey on a peculiarly human quirk: we relish a challenge. This means that, in our boundless quest to defeat bosses, gain high scores, or simply reach a final cutscene, we spend a considerable amount of our spare time in a state that lies somewhere between intense concentration and outright frustration. Or at least, that's how I've spent most of my time this month, as I've found myself drawn into the febrile, deceptively hostile world of *Tanuki Justice*.

A game that came to my attention thanks to HappyConsoleGamer on YouTube, *Tanuki Justice* is a run-and-gun platformer from Wonderboy Bobi, who previously brought us the charming *Wonder Boy*-alike, *Aggelos*. *Tanuki Justice* is, even beyond its 8-bit-style sprites and chiptune soundtrack, retro to its very core: it offers up a nightmare landscape of one-hit kills, aggressively fast enemies, and levels with no checkpoints. Sure, you start from where you left off when you die, but lose all your lives, and it's back to the start of the stage with you. And while each stage is linear and short, you'll still find yourself bouncing off the same sections again and again if your skills aren't up to scratch – which, evidently, mine aren't.

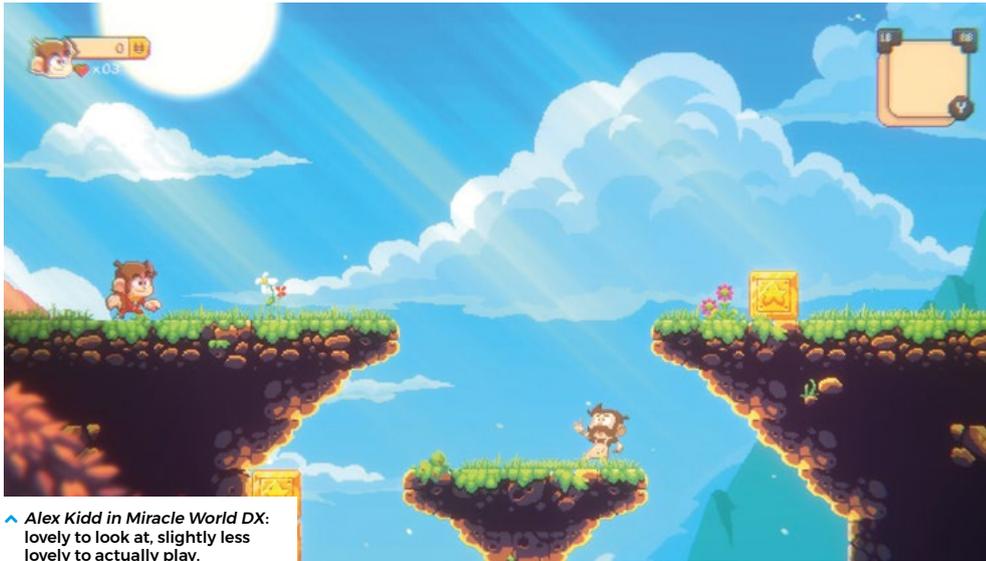
Tanuki Justice introduces an alternate feudal Japan of furry, shuriken-throwing avengers and feline samurai, and it's instantly disarming. As you fend off squirrels in bandanas and kamikaze koi carp, it's fun to spot all the references to other games: Tecmo's early *Ninja Gaiden* titles are an obvious touchstone, but background details also provide nerdy allusions to deep retro cuts like Irem's *Ninja Spirit*, Taito's *The Legend of Kage*, and (possibly) Jaleco's *Ninja-kun*. Stage two's leafy setting even appears to pay homage to *Ninja Gaiden* on the Sega Master System, a series entry likely played by all of a dozen people worldwide.

With those old-school ninja references comes the bit that every classic ninja game got exactly

▲ Low-ranking enemies can be murdered in order to fill up your charge meter. Circle of life!

▼ These enemies are less 'koi' than they first appear (sorry).





▲ *Alex Kidd in Miracle World DX*: lovely to look at, slightly less lovely to actually play.

right: tight, pixel-perfect controls. *Tanuki Justice* is brutally tough, but the agility of its raccoon hero and the responsiveness of his movement means it seldom feels unfair. Get into the flow, and there's a real thrill to narrowly dodging a barrage of projectiles, killing a few rank-and-file enemies to fill your charge meter, and then unleashing a huge super-shuriken to finish off a boss.

The same applies to the platforming: level three has you ascending a waterfall by jumping on narrow, single-character wide logs – a possible nod to a similar scenario in *The Revenge of Shinobi* on the Sega Mega Drive. When you get the timing right, it's a pleasing challenge to surmount, even if the process of getting to that level of proficiency can be controller-gnawingly frustrating.

At the time of writing, I haven't come close to completing *Tanuki Justice*, which is humbling given that a skilled player can finish the thing in less than 20 minutes. But far from being off-putting, the game's difficulty level is spurring me on: the further I journey into its fluffy world of death, the more attuned I become to its pace and timing. What seemed like ridiculously difficult sections can now be completed with ease, and even the meanest-looking bosses are starting to feel more friendly as I've begun to memorise their attack patterns.

Tanuki Justice is, in other words, a retro-style game done right. It's a contrast, in fact, to *Alex Kidd in Miracle World DX*, Merge Games and Jankenteam's remake of Sega's Master System platformer. At first glance, it's a charmer: the made-over sprites and backgrounds are

gorgeous, and there are a few quality-of-life improvements thrown in. On the original hardware, you had to press pause to get to the item menu; the problem was, Sega put this function on the console rather than on the joystick, which meant that you had to drag yourself off the sofa every time you wanted to activate, say, a power bracelet. In *Alex Kidd DX*, items are thankfully mapped to a face button.

In other areas, though, *Alex Kidd DX* feels retro in precisely the wrong way: Kidd's hitbox feels absolutely huge. Couple this with the

short range of his punch, and you're left with a lot of deaths simply because you edged closer to an enemy than intended.

Alex Kidd DX's most bemusing design decision, meanwhile, is to leave the original game's rock-paper-scissors battles pretty much intact. These were a frustrating element of the game in 1986, and they feel even more irksome in 2021; you've fought your way to the end of a stage, and whether you progress further or lose a life is down to sheer luck. Admittedly, *DX* doesn't knock you too far back in the stage if you're beaten, and it's still the case that an item you find later in the game will allow you to 'see' your opponent's move ahead of time, but it's still a curious thing to leave untouched when other elements of *Alex Kidd* have been so affectionately reworked.

Alex Kidd DX isn't an awful game by any means, and there are stretches where the series' charm shines through. When compared to *Tanuki Justice*, though, *Alex Kidd DX* feels like it's taken plenty of good bits from the medium's history, but also some of the bad bits, too. 🗨

“An alternate feudal Japan of furry avengers”

Wireframe Recommends



Aggelos

PC/VARIOUS

Tanuki Justice developer Wonderboy Bobi's first game, this action-adventure pales somewhat in comparison to the likes of *Monster Boy* and *Wonder Boy: The Dragon's Trap*, but it's still a sprightly, affectionately-made little jaunt.



Cyber Shadow

PC/VARIOUS

Like *Tanuki Justice*, *Cyber Shadow* is an aggressively tough throwback to an age of rock-hard ninja platformers. And again, some satisfyingly responsive controls and slick audio-visual design help leaven the frustration.



Save me Mr Tako

SWITCH/VARIOUS

This Game Boy-looking platformer was fun yet flawed on its release in 2018, but thanks to a new deal with Limited Run Games, it's back in an improved *Definitive Edition*. If you missed it first time around, it's well worth checking out.



No Man's Sky

The joy of not having to actually bother doing anything

HELLO GAMES / 2016 / PC, PS5, XB S/X, PS4, XBO

One big thing was drowned out in the Internet Rage that followed 2016's release of *No Man's Sky*; an element those who actually bothered to play the game instead of complaining that you couldn't fly under an alien Diplodocus' legs noticed and were too busy enjoying to get caught up in the, like, aggression, man. That being: the ability to just not have to bother if you didn't want to.

There are countless games out there facilitating the need to do nothing – the walking sims, the life sims, the sandbox physics creation tools, social MMOs, and more. It's by no means exclusive to *No Man's Sky*. But it's in Hello Games' ever-expanding epic that the true allure of reflection, contemplation, and general existentialism all came together behind the disguise of a heavily promoted game about spaceships, robots, and lasers.

Sure, you could explore with goals in mind – hunting down the trail of Atlas, maybe – you could play it like a sci-fi *Pokémon Snap*, attempting to catalogue every species you came across along the way. You could channel your inner *Elite* piloting skills and hunt down pirates for the sweet glory of victory in combat. It wasn't quite as... *refined* an experience as it is today, but *No Man's Sky* did offer players a lot of things to do from day one.

And one of those things was to eschew all goals, whether personal or set by the game. Obviously the argument can rightly be made there that aiming to eschew all goals is in fact

a goal in itself, but that's the sort of roundabout thinking that can bog down a piece like this and get us nowhere. It's also not in keeping with the fundamentals underlying these words: relax. Take a moment. Breathe. Just potter about, take it in, have an experience instead of demanding satisfaction.

Mind you, it's never been a given in *No Man's Sky* that you'll be able to relax and do nothing, even if you've been royally committed to that approach in the five years since the game came out. The randomised nature of the universe you explore means it's not unheard of to discover half a dozen planets in a row that seem to *hate* your existence and actually work on

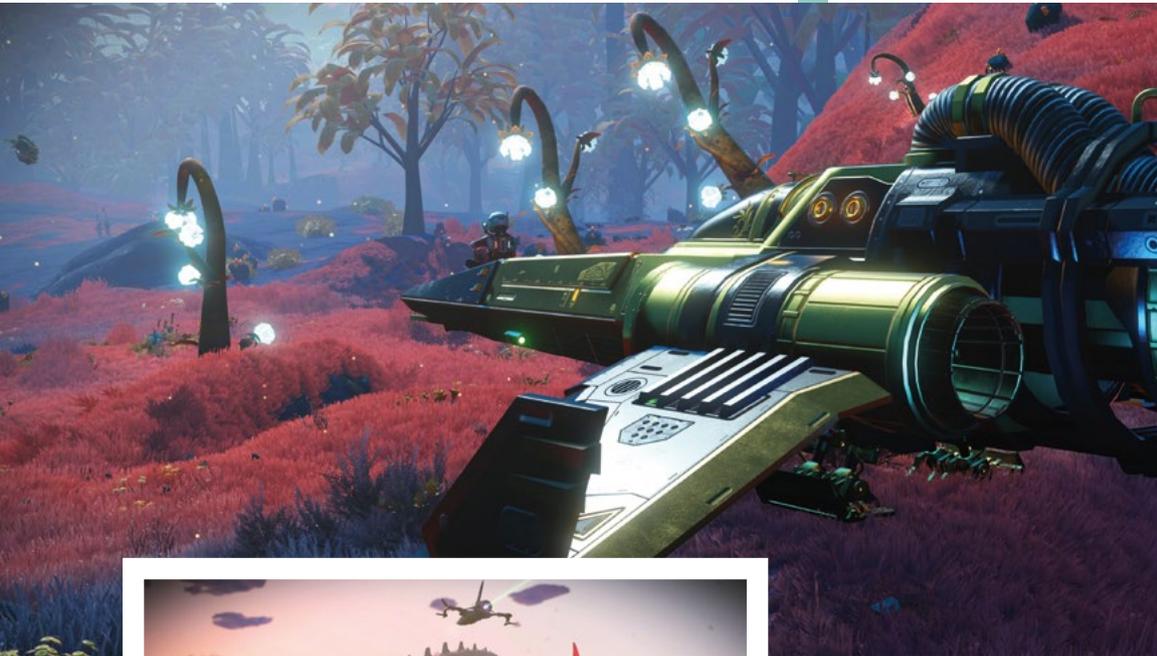
ending it. Be it extreme storms, radiation levels higher than living creatures can stand for more than a minute, or fauna with a penchant for murderous intent. Sometimes the cosmos does just work against you.

But that just makes those moments of Zen all the more satisfying. When, after

trying and failing for an hour or two in the real world, you suddenly find yourself on a lush, green planet full of placid life and a temperature that wouldn't be out of the question on a sunny day in Margate – when you realise you've found yourself somewhere you can take a load off and just *exist* – that has impact. Would you believe it, sometimes it is about the journey.

As mentioned, *No Man's Sky* has expanded significantly since those – on reflection rather scant – early days. Hello Games has quietly undertaken one of the great makeovers and

“When you realise you’ve found a place you can take a load off and exist, that has impact”



Where are we?

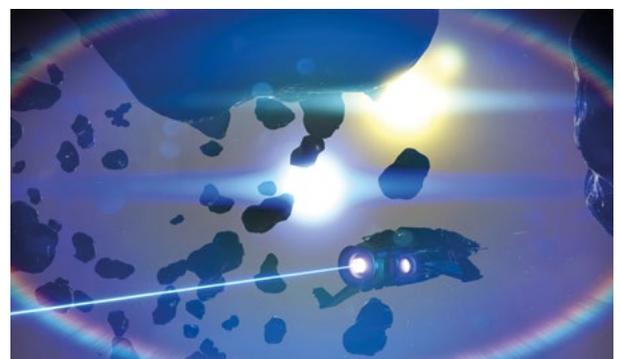
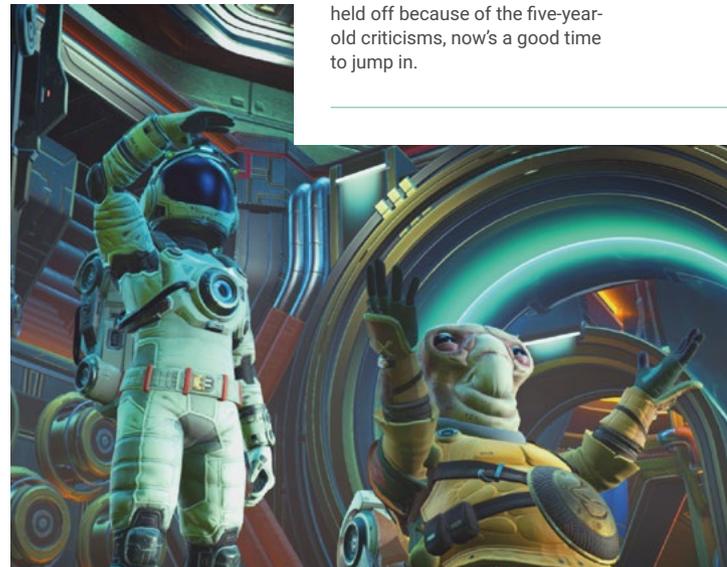
At the time of writing, *No Man's Sky* is five years old and in the midst of its 17th named update since launch, *Prisms*. It's impossible to list everything that's been added or improved on in the limited space here, but we've seen base building, vehicles, multiplayer (including cross-platform play), new-gen versions, countless graphical and procedural generation improvements, more varied worlds and wildlife, a more coherent narrative to follow, plenty of side quests, increased prettification (technical term), and – among so many other things – a photo mode. If you've held off because of the five-year-old criticisms, now's a good time to jump in.



refurbishments of the modern gaming era, bringing its passion project out of the doldrums and raising its reputation and standing in the gaming community to the point that it's now only really a punchline for the terminally uninformed. The redemption arc of the studio itself shouldn't be overlooked either, with a name that was poison in over-influential – but still influential – online circles now something of a paragon of virtue.

It only took five years of dedicated hard work and a Herculean ability to ignore the majority of abuse thrown the way of the studio – often directly at Hello Games chief Sean Murray. *Only*. We can merely guess blindly at the ridiculous toll the whole affair must have had on Murray, the rest of the Hello Games team, and their families and friends.

And all of this so someone can, in a magazine, write a few hundred words about how great *No Man's Sky* is to just relax and exist in. How tempting it must have been for Hello Games to lose itself in the universe of its own making; free of stresses, of the input of others – free of worry and having to care about anything going on anywhere else in existence. Instead we end up where we are now: a game still being tinkered with and improved, the result of thousands of collective hours of work, crafted with genuine love and affection – and *effort* – all so other people can use *No Man's Sky* to get away from everything. 🌌



Next Issue

ON SALE 2 SEP

FAR

CHANGING TIDES

Okomotive set sail for a bigger, deeper adventure sequel

Don't miss out!
Subscribe
PG76

ALSO

- > The unexpected rise of comedy golf games
- > Sports games for people who hate sport
- > The ingredients that go into making a James Bond game
- > How to create a computer emulator in Python

Editorial

Editor

Ryan Lambie

Email ryan.lambie@raspberrypi.com

Features Editor

Ian Dransfield

Email ian.dransfield@raspberrypi.com

Sub-Editors

David Higgs & Vel Illic

Design

criticalmedia.co.uk

Head of Design

Lee Allen

Designer

Sam Ribbits

Contributors

Jon Bailes, Dan Cooper, Antony de Fault, Konstantinos Dimopoulos, Damiano Gerli, Craig Grannell, Kyle Hoekstra, Kim Justice, Steve McNeil, Jordan Oloman, Aaron Potter, Rhett Thompson, Mark Vanstone, Howard Scott Warshaw, Rob Whitehead

Publishing

Publishing Director

Russell Barnes

Email russell@raspberrypi.com

Director of Communications

Liz Upton

CEO

Eben Upton

Advertising

Commercial Manager

Charlie Milligan

Email charlotte.milligan@raspberrypi.com

Tel +44 (0)7725 368887

Distribution

Seymour Distribution Ltd

2 East Poultry Ave, London EC1A 9PT

Tel +44 (0)207 429 4000

Subscriptions

Unit 6, The Enterprise Centre, Kelvin Lane, Manor Royal, Crawley, West Sussex, RH10 9PE

To subscribe

Call 01293 312192 or visit wfmag.cc/subscribe

Subscription queries

wireframe@subscriptionhelpline.co.uk



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, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. 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).

PHILIPS

Monitors

get
in the
moment

innovation  you



Momentum 32" 4K UHD, LCD display with Ambiglow
326M6VJRM

4K
Ultra HD



VESA CERTIFIED
DisplayHDR™ 400



UltraWideColor



Adaptive Sync

Available at:

amazon

ebuyer.com

SCAN[®]



ULTIMATE IMMERSION

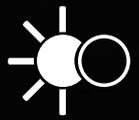
with the
AG49UCX

DQHD
5120x1440

120 Hz

1 ms

AMD
FreeSync
Premium Pro


HDR 400

Available at:

amazon

OVERCLOCKERS UK

AOC

f @aocgaming

t @aoc_gaming

ig @aocgaming

aocgaming.com