£3
ALL FORMATS

Wireframe

LIFTING THE LID ON VIDEO GAMES

Degrees of Separation

Chris Avellone on narrative design

Livin' the Dreamcast

Inside its thriving indie scene

Material worlds

Making games with wool and clay

Issue 7 £3

wfmag.cc

9 772631 672000 07

PROJECT FEAR

THE DEVS BACK FROM THE BRINK WITH SOMEDAY YOU'LL RETURN

Subscribe today

12 weeks for £12*

Visit: **wfmag.cc/12weeks** to order

* UK Price. 6 issue introductory offer

2019: the year indie dies?

This is not an indiepocalypse column. We've had plenty of those, and they've all been wrong. This is, however, a cautionary piece. I'd argue that 2019 will be remembered as the year 'indie' died. Why isn't this an indiepocalypse? Well, let's dig into some terms.

Originally, 'indie' just meant independent, or made without the aid of a publisher. We're so far past that point that indies have themselves become publishers. Chucklefish and Stardew Valley come to mind here.

So when I say 2019 is the year indie died, I mean in the original sense. The death of self-funding, and the return to publishers as default.

The indie bubble was created in large part by Steam curation. It artificially forced a small number of tiny games in front of, effectively, the entirety of PC gaming. That created the initial wave of indie hits, and we've been coasting along on Steam's good graces ever since.

That was always going to end, and it did, with 2018's discovery algorithm shift. To be fair, Steam's hand was forced a bit by the flood of games (most of them good), but we're far enough past the algorithm change to know that indies impacted by it aren't rebounding, which sets the timeline.

Most studios try to set aside six months of buffer money. Those six months start burning in January, when the payout for Steam Christmas is much lower than expected. Those who can move into contracting will, but most will simply taper into non-existence by the autumn. A handful will attempt Kickstarters, but most of those will probably fail. (This isn't a knock on Kickstarters or indies, but an observation of which tier of indie all this hits hardest.)

Many will shift back to pitching games to publishers. Indies without six months of buffer will be hit hardest.

MEGAN FOX

Megan Fox runs Glass Bottom Games, a maker of absurdly cosy games about animals doing people things.

This was their first or second major release, and they were counting on Steam Christmas money that didn't come. Those same studios are too young to pull off Patreons or Kickstarters due to a lack of community, and had insufficient forewarning to build one.

They are, in part, who I'm writing this to. You just saw your sales report, and it punched you in the gut. Part of you is going, "We'll buckle down and hit Steam again." Don't. Stop, and recognise the game has changed. We're no longer in a market where selling a good game on Steam is likely to work. You need fallback plans. You need to work on establishing the core community that will buy your games no matter what, or find someone to hire you to work on their game. Next-gen consoles are also about to emerge, and you could pitch your next game directly to them.

I'm also writing this for gamers. We're seeing a market shift – and boy, will this one favour you. Player eyeballs have been gobbled up by huge stores, and well-funded newcomers – Game Pass, Twitch, Discord – are fighting for those eyeballs by offering free games. The money won't last forever, but while it does, you'll be drowning in games. It's the same force washing the lower end out, and pushing everyone else to strike deals (possibly with Game Pass or Discord) to survive the flood.

We game devs need to position ourselves so we can survive until the market recovers – which it will. After that, we'll see next-gen consoles begin to pull everyone back up. Consoles tend to remember indies are handy when struggling for exclusives, and that's often enough to power a rebirth. So we'll see.

Until then? It's a good time to climb into a bunker. Take a bunch of great games in there with you – goodness knows there's enough of them floating around for the taking lately.

#07

Contents

6

Attract mode

Interface

44

Toolbox

Rated

WELCOME

Games may be made up from clever, scientific things like polygons and thousands of lines of code, but they're also worryingly good at zoning in on our primal fear centres. Long before the survival horror genre was a thing, fear was a vital part of game design. Look again at *Space Invaders*, and the aliens' slow, zombie-like advance toward the hopelessly outnumbered player; in its own, rudimentary way, it's pretty nightmarish.

As Lottie Bevan points out in her column on page 26, technology is allowing developers to create ever more vivid scenes of terror and gore – Konami's intense *Resident Evil 2* remake is certainly a testament to that. But at the same time, it's often the games that generate suspense without gore or violence that disturb us the most. In *Alien Isolation*, it was the build-up to the appearance of the title monster (or those pesky androids) that sticks in my mind. Similarly, the eerie stillness of *Everybody's Gone to the Rapture* was oddly flesh-crawling, even though there wasn't a monster in sight.

It's why we're so intrigued by *Someday You'll Return*: it aims to create a psychological horror with no weapons or combat. Of course, we don't know if the team behind it will achieve their goal of making a great scary game without jump scares, but as *Gone Home* proved, it's often the quieter, more low-key games that provide the most suspense.

Ryan Lambie
Editor

Exploring the silent hills

Czech studio CBE Software's Jan Kavan and Lukáš Medek tell us about Someday You'll Return, and a horror that will linger

The story behind *Someday You'll Return*'s path to becoming a finished game could be framed as the triumph of the plucky underdog; a success snatched from the jaws of defeat. A few years ago, the game's developer CBE Software released *J.U.L.I.A.: Among the Stars* in partnership with UK publisher Lace Mamba. To say the relationship didn't work out would be an understatement – royalties were said to be unpaid, profits disappeared, Lace Mamba ended up closing, and CBE was left with a black eye.

Since reclaiming *J.U.L.I.A.* and re-releasing an updated version, CBE has dragged itself back up from the brink, and focused all its efforts on the first-person psychological horror of *Someday You'll Return*. The game centres on a father's missing daughter, his search taking him to locales modelled on real places in the Moravian region of modern Czechia and Slovakia. As well as looking gorgeous, CBE's two-man team promises the game's horror is something that will elevate the experience, helping it to live on in the minds of players for years to come.

Surviving the horror of a rogue publisher has to leave scars. *Someday You'll Return* could prove itself to be CBE's absolution; a triumph in an industry that once seemed hostile. We spoke with Jan Kavan and Lukáš Medek about *Someday You'll Return*'s road to release.

You're very clear about *Someday You'll Return*'s influences. What's the thinking behind this heart-on-your-sleeve approach?

Jan: The thing is, our game is a bit different. When the trailer comes out, there is immediately this 'walking simulator' connotation – people say it will be another of that genre. So one of the reasons we list these titles is, not only because we like them and are influenced by them, but to show we're not going to be a pure walking simulator, because we have influences from games that are definitely not walking simulators!

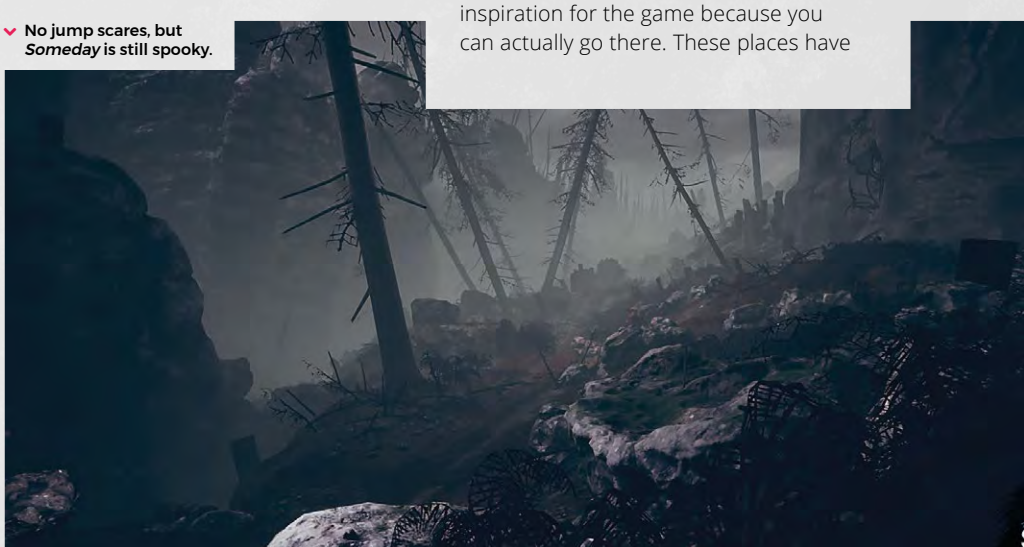
It's very hard to come out with a new IP and not provide a helping hand to the press, for example, in this gaming sphere. So it helps them categorise the game more, though the game does have some ideas which are very new in gaming per se.

Do you think it could harm the game at all, with people expecting a *Silent Hill 2* or *Resident Evil 7*?

Jan: If people are expecting something like *Silent Hill 2* they will find some stuff that will make them happy. But it's not like this is a clone of *Silent Hill 2* – it's more how we approach the symbolism, the meanings in the game and how we tell the story. There are some things that people who really liked *Silent Hill 2* – like me – will enjoy. Of course our game is different, it's not trying to be a copy of anything else. It could be interpreted that way, but that's always the risk!

You've opted for real-world locations for the game – why is that? What significance do they have to you, and how is that translated to the game?

Lukáš: We're putting into the game the places we grew up in – it's a great inspiration for the game because you can actually go there. These places have



No jump scares, but *Someday* is still spooky.

"It's not only about a missing child, or somebody running away – every father is guilty of things in the game"

a unique feeling for us because of the memories and experiences we had there, so it immediately brings new ideas for the story, for example. Also we can go there and get a whole picture of the place, its mood and atmosphere, so we can recreate it better in the game. That's not to say these regions are placed exactly like they are in real life – they're more an emotional link. They look different, but the mood is there, and it comes from our actual memories of the places.

How much of a 'Czech' game is this? Will your fellow Czechs feel an intimate bond with the game, for example?

Lukáš: We have strong feedback from locals who've seen the game, saying 'Oh I know it! It's like home!' And we think it could be great for foreigners, too – the Slavic environment is really popular from, say, the *Witcher* games. So we hope it could be something fresh for that... In the core of the story we're trying to implement as much as we can from local folklore and myths from these places. So the Bohemian mood from the old legends should be there.

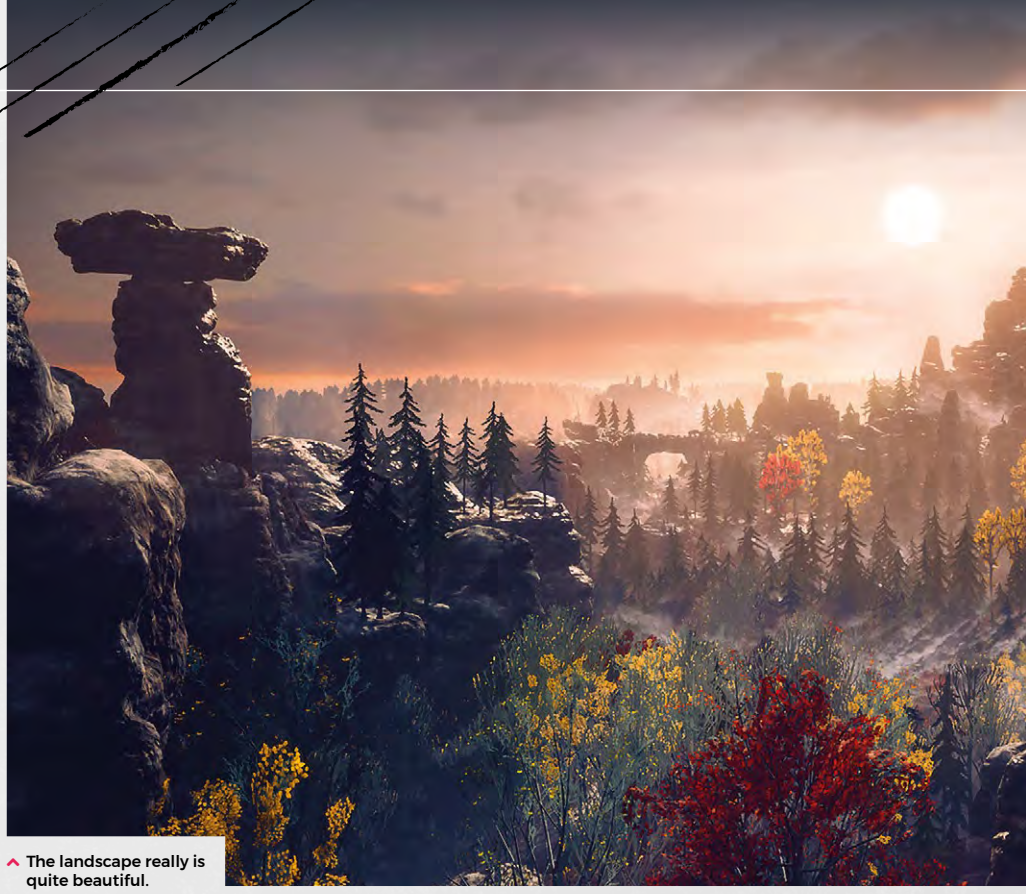
Jan: But still it's possible to digest the story even without being interested in Slavic folklore. It's really more to bring this local vibe into it all, but you're really looking for your daughter and it's a very universal feeling, losing your child. The same everywhere as in the Czech Republic.

How much does fatherhood inspire *Someday You'll Return*'s story and design – is it based on real fears?

Jan: I have four kids, so it's something that taps a lot into my deep fears. It's not only about a missing child, or somebody who's running away from you – it's more about... everybody, every father is guilty of the things which are in the game. It's something I want to present in the story – it's not only about the worst stuff that happens, there's all this mundane stuff which everybody does to a certain extent. The real horrors aren't just the strange, horrible things, but also the everyday, mundane things which we sometimes don't even acknowledge that we're doing.

Designing a game without combat is, oddly, the exception to the rule. What specific challenges are there, when you're making a game that does have a threat to the player who can't fight back?

Jan: One of the things that's nice about that – when you're fighting something, you always have the feeling there's a chance to beat it. You have a gun, and if you're skilful you can shoot it in the head or something. But if you don't have this option, then the fear is much stronger because you still want to survive, but you don't have a feeling where you can turn a former ➔



The landscape really is quite beautiful.

camping instructor into a marine who can go in with a gun and shoot a thousand monsters – which is total nonsense, because he doesn't have a skill-set for that. The other thing that's a big challenge for us is that you have to design many more elements when there's no combat – when you have combat, you can basically plug in a lot of enemy waves, you can plug in arenas where you just run around in cover and shoot, the default scheme of things these days. But if you don't have this, you have to come in with different ideas. We have a strong adventure game background, so we bring in stuff from there. There's crafting – we have a huge interactive system for that, where you have to actually build things – and we also have other elements like stealth, which help overcome this. But it is much harder, because when you can't just copy-paste a load of enemy waves you have to come up with original ways in which to overcome this situation. In this aspect it's much more difficult, without combat.

"We'll never take a game we invested millions into and give it to a company which may or may not care about it"

How did you decide on your funding approach this time around? It's been a rollercoaster of publishers and Indiegogo experiments over the years – how do you know this is the right way to do it?

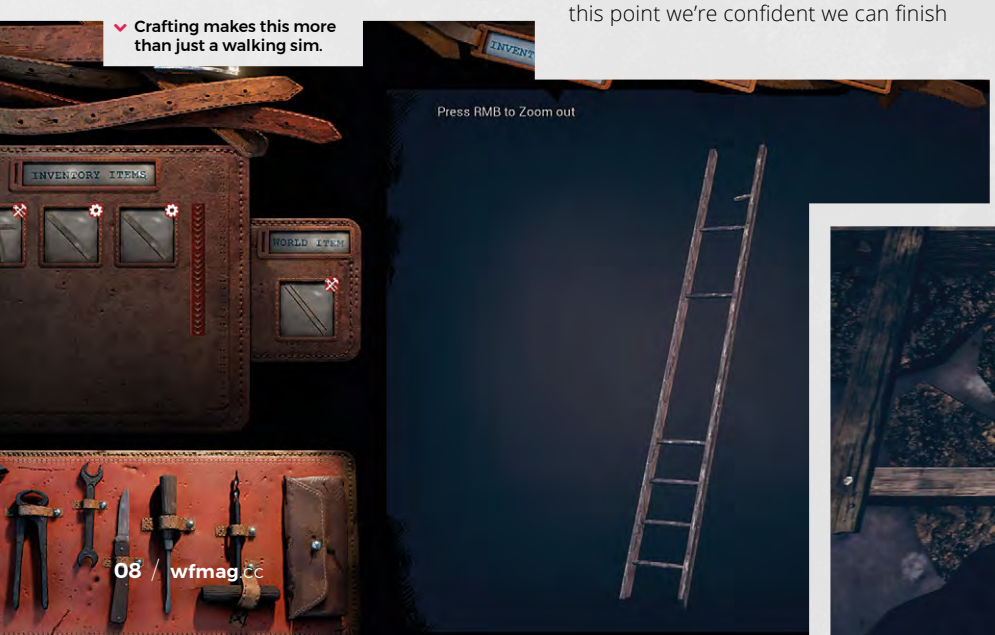
Jan: We have four main sources of income – our last game, *J.U.L.I.A.: Among the Stars*, was pretty successful, so the whole income from this game we poured into *Someday*. We were also very happy to be awarded a media grant by the European Union for development. There was also the Epic Games Development Grant for Unreal Engine games. And at this point there's also quite a big personal investment in this project by me, so that's another chunk of money we're putting on the table. At this point we're confident we can finish development and release that game with this funding.

I assume your experience with Lace Mamba and *J.U.L.I.A.: Among the Stars* factored into your decisions?

Jan: We've been in the industry a lot of years; we started CBE in about 2006, and we've been in the industry since 2002. And through all these years the story has just repeated, so we're confident that we know. If we're going through the publisher route, we're looking at local markets, but we'll never take a game we invested many millions into and just give it to some company which may or may not care about it. So that's the problem we had.

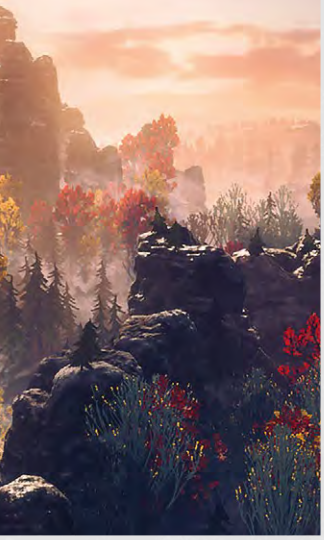
Visibility is going to be a challenge – how do you see the market for *Someday You'll Return*?

Jan: We are enthusiastic – the reception of our trailer was pretty huge, and we



Crafting makes this more than just a walking sim.

No, the hammer cannot be used as a weapon.



Can you send a text to get help? Probably not, no.

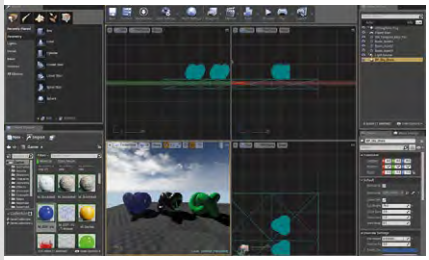
have the feeling there might be people interested in this game. We travelled to many of the conventions, we got some awards, but also the people there who played the game were enthusiastic. So we feel somehow we can tap into a much bigger playerbase than with *J.U.L.I.A.* – even though it was greatly received, with a great core of players and really positive reviews on Steam, still it's a niche if we compare it to *Someday You'll Return*. We think this will be much bigger.

What's your current team set-up like? What's the pipeline or workflow like on any given day?

Jan: Really we don't work in separate roles like you would in a corporate situation; we are much tighter, discussing any individual idea so we are clear from the beginning how it will work, if it will work, if it's not just a dream. Sometimes you have a great idea, but then when you figure out everything you need to make the idea, you have to carefully calculate if it's actually implementable for a reasonable amount of money! We have a Trello system where we have boards for our features and a task list; whenever we're done with a task we sit down and discuss stuff. We do a lot of iterations, so we can sit and look at the game, discuss how it plays, what works, what can be done, where it can be better, where it can be more streamlined, where it can be clearer for players. It all feeds back to design, so there might be changes to the lighting system, there can be changes in the visibility on levels, but there can also be many systematic changes in the code... so it's just basically like one huge circle going round and round to make sure everything works how we want.

This is the first time you've brought something to console – what's that been like?

Jan: We're afraid! But I've heard rumours the PS5 is coming, so if that's coming it can just be a straight port [laughs]. But at this point we're focusing on the PC build – when you have a PC build on Steam, for example, you can roll out a patch in a minute. Our goal at this point is to create the most bugless version possible, so when we port it to console we might fight with some console-specific issues, but not with the game itself. Patches on console need to be approved, there's a lot of workflow in that, so at this point we're working on finalising the Windows PC build.



UNREAL-LY

Someday You'll Return has been made using Unreal Engine – something Jan is convinced was almost inevitable: "I don't think we have an alternative," he says. "Unreal Engine is so far ahead of other engines at this point. You can meet a lot of challenges working with any engine, but with UE we are so comfortable now. We might have to constantly struggle in another engine, but here we can create something which is really unique, so I don't think we would be willing to change. This engine is really powerful, and has great support – not just for the engine, but for the community, too. It was the right choice for us."

Finally, I'm terrible with horror. Sell it to me – what am I going to get from *Someday You'll Return*?

Jan: You're not going to get jump scares or some kind of gore-fest. You'll get a horror that you will think about for a long time. While there are some very disturbing things in the game, it's not based around them – it's more like if you read a great book then you'll dwell on it for many months. That's what we're trying to recreate here, so it doesn't just end by killing a boss monster and calling it a day – it will really ask a lot of questions, and it'll be up to you to answer them. So it's a bit different. I'd be lying if I said it was just a walk in the forest, though – it gets pretty disturbing at times, but it doesn't focus entirely on these elements.

***Someday You'll Return* releases on PC in 2019; console versions will follow.**

Hoa's lush world is partly taken from its creators' environment, with the devs taking reference photos at Singapore's parks and botanical gardens.

Turning a new leaf

Hailing from Singapore, hand-drawn platformer Hoa is already a thing of beauty

It may still be in early development, but *Hoa* has one of those game worlds that make us want to crawl into the screen, breathe in the fresh air and maybe fall asleep under a leaf. A side-scrolling platformer created by two former art students in Singapore, *Hoa* was conceived as a soothing, non-violent experience right from the beginning.

"When I first came up with the story of *Hoa* about a year ago, I had no idea how the game would look," co-developer Son Tung tells us. "The only thing I knew at the time is that I wanted it to be so beautiful that you would want to play it just to be in it. I've always been attracted to beautiful platform games, so whenever I see an interesting visual, whether it's a photo, a painting or a movie scene, the first thing that comes to my mind is how can I create a side-scroller with this kind of look."

Hoa's luminous hand-painted style most obviously resembles the output of Japanese animation house, Studio Ghibli; the tiny heroine, who jumps from leaf to leaf in a verdant forest, recalls the firm's 2010 film, *Arrietty*. But while Son Tung readily acknowledges the influence of Studio Ghibli and its co-founder Hayao Miyazaki on *Hoa*, the game also draws its inspiration from a wealth of other sources.

"I had the privilege to attend classes of animation legends Hans Bacher and Ishu Patel during my time at ADM," Son Tung says, referring to Singapore's School of Art, Design and Media. "Their teaching had a lot of impacts on my works. I also closely follow artists like Aaron Blaise and Goro Fujita – their amazing work taught me a lot about the art of animation."

As well as traditional painting and animation techniques, *Hoa* also weaves 3D imagery generated in Maya to create a sense of movement and depth.

"We mainly use 3D for characters, creatures and elements that require a lot of animation," Son Tung says. "Rather than animating in a traditional way, working in 3D saves us a lot of time to create smooth animation. For the illusion of depth, there are two things we need to pay attention to. The first one is to arrange the elements in Z-axis to create convincing parallax effect, and the second one is to apply atmospheric perspective when painting the background."

While *Hoa* is still a nascent project, its creators hope to have a playable demo to show off by the middle of 2019; if the feedback's positive, then the aim is to have the finished game out by the following year. Rather than a free-roaming Metroidvania, *Hoa* will go for a more linear, story-driven design, Son Tung says. And while the imagery shown here is bathed in sunlight, he adds that *Hoa*'s narrative will also visit some darker places.

"There will be a lot of puzzles in the game," Son Tung says, "but probably no combat. Our vision from the start is for *Hoa* to be a game with minimum violence. The game starts on a bright note, but as players venture on, they will discover bleak corners of the game's world and the stories behind it."

Info

GENRE
Platformer

FORMAT
PC

DEVELOPER
Son Tung

PUBLISHER
Son Tung

RELEASE
TBC 2020

"There will be conflicts, but they will not be solved with Hoa launching fireballs towards the enemy," Son Tung says.

Taverns are areas where players will pick up information to complete questlines.

Answering the call

How Call of Saregnar plans to revive nineties-style RPG classics like Daggerfall and Betrayal at Krondor

Info

GENRE
RPG

FORMAT
PC

DEVELOPER
Rhuantavan

PUBLISHER
Rhuantavan

RELEASE
TBA

Looking at screenshots of *Call of Saregnar*, you might be forgiven for thinking it's a long-lost RPG from the 1990s. The characters are photographed actors, the textures are grainy, and the environments are made up of flat 2D objects. In reality, *Call of Saregnar*'s a brand-new game from Slovenian designer, Damjan Mozetič. Envisioned as a return to the western RPGs of old, it draws from titles like *Might & Magic VI* and *The Elder Scrolls II*. The biggest inspiration, though, comes from Mozetič's first love: 1993's *Betrayal at Krondor*.

For Mozetič, *Betrayal at Krondor* was a formative experience. At age 14, he managed to get his hands on a pirated copy of the game – a practice not uncommon at the time, given the state of the Slovenian market after the break-up of Yugoslavia in 1991. The experience was like nothing he'd ever played before, and led to several late nights spent at a computer with a dictionary at hand to compensate for his basic understanding of English.

Mozetič chose to use real actors in order to give players a real person to relate to.

Thanks to a custom shader, *Call of Saregnar*'s trees now have shadows.

"I've wanted to make a game like *Betrayal at Krondor* since forever, because the sequel was never released," says Mozetič, referring to *Krondor*'s cancelled follow-up, *Thief of Dreams*. "I mean, there was *Return to Krondor*, but that was not the game that was originally supposed to be released."

SPIRITUAL SUCCESSOR

"*Krondor* had it all," Mozetič enthuses. "It had great characters. It had a great world, which was Midkemia from [the series' author] Raymond E. Feist's books. So it had this really nice background to put a story in, and Neal Hallford, the author of the story and the characters, did a great job."

Call of Saregnar is in many ways a spiritual successor to the original *Krondor*. Not only do both feature real actors in the role of non-playable characters, but the two games are also divided into chapters and include an open world you can explore almost from the beginning, populated with new missions depending on the chapter. They also feature turn-based combat and strategy elements, with the player able to launch sneak attacks on enemies or fall into ambushes in either game.

Making a game like this isn't easy: financing and workflow aside, *Saregnar* uses techniques that aren't common in contemporary game development. This often means educational resources are hard to come by, with his game's aesthetic being particularly difficult to get right.

"There's no info online whatsoever," Mozetič says of his approach to *Saregnar*'s graphics.

Players can set up a cold camp in dangerous territories to avoid attracting attention from bandits.

There are a number of towns to explore in the game, all complete with NPCs, including the villages of Romar and Ergendon.

"I had to figure out by myself how to do it. For example, the trees were interesting. I eventually got Charles Thomas, a freelance shader programmer from France, to help me with a shader. I wanted those trees to have a shadow on the ground, so he helped me with that. Because, as the sun moves around, the shadows [need to move]. The trees are still billboards, they're 2D, but there's a technique we use to project the same shape of the tree on the ground. It ended up looking really nice."

A STROKE OF LUCK

There's also a logistical problem of finding enough actors to populate the world's locations. When you think of RPGs, you tend to think of towns and villages filled with roaming NPCs and quest-givers. For Mozetič, this presented the issue of finding and casting a large group of performers to fill the roles.

Luckily, though, he soon crossed paths with the Principesca Contea di Gorizia, an Italian medieval re-enactment group (of which he's now a member) which comprised several fans of classic RPGs. They were more than willing to help out, and even provided some of their own costumes for the roles.

"The characters are the biggest challenge, because I had to do things that weren't traditional game development," says Mozetič. "It's not development, it's not programming. It's getting people together and organising the photo shoots and hoping that everything will work right – so it is a big relief when it does. I currently have about 20 characters, with some of the same actors wearing different clothes, but since I'm doing low-resolution, I can get away with having the same face repeating in other scenes. I also photographed the characters from different directions, so I can show them from the front or from the side or from the back."

"I've wanted to make a game like *Betrayal at Krondor* since forever"

Collaboration has been key to the development of *Call of Saregnar*. Though Mozetič wants to keep his team small in order to avoid moving away from his central vision – a move that has resulted in a somewhat protracted development – everyone who's come on board so far is equally passionate about the game. Composer Tony Manfredonia, for instance, has produced several medieval-style tracks inspired by *Krondor*'s score and other similar games from the era, while sound designer Benjamin Reichstein has worked hard to nail the warm, rough, retro sound that fans expect.

If you're wondering when we will see *Call of Saregnar* on digital storefronts, it's not anytime soon, according to Mozetič. The game is still relatively early in its development, with a Kickstarter and demo being the next target the team are working towards. After that, Mozetič hopes he'll be able to quit his nine-to-five job and work full-time on the game, as well as bring on some new faces in order to speed up development.

With *Call of Saregnar*, Mozetič is hoping to hark back to a simpler time for western role-playing games, where the focus was more on evoking a sense of atmosphere through adventuring, and telling a story rather than providing complicated combat systems or increasingly detailed environments.

"There are many action games around, but RPG games like mine, there are basically none," says Mozetič. "I really want to make a game that differs from [other role-playing games]. I'm trying to tap into that niche market, which I think people don't even know they're missing."

A TIMELINE OF INFLUENCES

1993: Betrayal at Krondor
One of the first truly open-world RPGs, *Krondor* tells a tale of magic and political intrigue.

1994: Realms of Arkania: Star Trail
The second entry in *The Northland Trilogy*, a series of games based on the German RPG, *The Dark Eye*.

1996: The Elder Scrolls II: Daggerfall
The second *Elder Scrolls* game boasts a map roughly half the size of the UK. It follows the *Hero of Daggerfall* and his attempt to solve the mystery surrounding the death of King Lysandus.

1998: Might and Magic VI
A party of four adventurers rise to defeat the invading Kreegans. This superior sequel included deep character customisation systems and plenty of side quests to uncover.

Headlines
from the virtual front

01

02

03

01. Epic flex

Announced in December, the Epic games store has since launched and is... well, another online store controlled by a developer/publisher. Needing to upset Steam's hegemony over the PC digital sales market requires aggression – first the 12% cut taken by Epic (Valve takes 30% for Steam), and now something to directly impact consumers.

Epic has bagged THQ Nordic's *Metro Exodus* as a timed exclusive on the store, meaning it will not be available on Steam – even though pre-orders (which will be honoured) have been open a while. Valve's reaction to this move has been to label it 'unfair', while THQ Nordic has put the decision solely at the feet of sister company, Koch Media. The waters are getting a bit choppy, it seems.

02. Starbreeze's blame game

Sad news towards the end of last year saw Starbreeze filing for reconstruction (basically bankruptcy protection) and its CEO Bo Andersson stepping down. A report from Eurogamer, though, shed some light on the last days of Andersson – and it turns out he blamed the studio's developers for its lack of success.

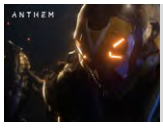
In his final email to staff, the outgoing CEO wrote: "Personally though I lost all my money, my family in divorce and my kids custody through the toil over the last 2–3 years, working 100-hour weeks for Starbreeze and keeping you devs paid and in the game. With less and less developers willing to put in the extra care in a product, it clearly limits the possible result of enough quality in time.

"This is a new era and I did not leave the old one and adapt in time – my fault. It's ok – it's new times."

03. Leather fallout

Bethesda appears set on testing consumers with as much *unique* merchandise as it can, with the announcement of a *Fallout* leather jacket. Following on from the dubious 'canvas' (initially polyester) bags for *Fallout 76*'s collector's edition and its bottles of (plastic fascia-ed) Nuka Rum for the same game, we're hardly surprised there was a reaction at the jacket's announcement. That is to say: people aren't impressed, and people are very wary of Bethesda's merchandise these days.

It costs £210, by the way.



Anthem demo suffers connection issues, unsurprisingly

Apple plans 'Netflix for games'; nobody surprised

04

05

06

04. Press A/F to Jason

Quantic Dream's exclusive affair with Sony is over, after the French studio received an undisclosed minority investment from Chinese firm, NetEase. With the money to hand, the team is to release future games on multiple formats, after having been PlayStation-exclusive since 2010's *Heavy Rain*.

CEO David Cage said: "Having NetEase by our side as a strategic partner will allow us to expand our creative vision and develop the company to its fullest potential," which is much less pretentious than what we're used to.

This isn't NetEase's first throw of the dice, with the company investing £76m into Bungie towards the tail end of 2018.

05. Mr Fighterman

Pop quiz: who would be your ideal addition to the roster of a *Mortal Kombat* game? If you said 'Shaggy from *Scooby-Doo*, obviously' then congratulations – you're aware of daft little jokes getting out of hand!

Yes, The Internet has decided it wants Shaggy in *Mortal Kombat 11*, and a petition has seen hundreds of thousands of signatures demanding the addition of the cowardly hippy to a game about ripping off arms and beating an opponent to death with the wet end.

Will it happen? Series co-creator Ed Boon is aware of the petition, so we're allowing ourselves to dream a bit here.

06. Valve time is 'really nice'

Valve Time – the somewhat pejorative term thrown at the studio when speaking of its glacial pace towards... well, anything – isn't a bad thing, according to Jane Ng, artist at Campo Santo (brought under Valve's wing in 2018).

"In reality," she wrote on Twitter, "it's really nice to walk around a mostly empty office at 5:30pm because I know people get to be parents to their kids, and/or just be living their lives outside of their jobs.

"How this should just be the norm for everyone in a wealthy country is a separate conversation. For now, I'm glad this is the culture Valve folks have collectively decided is the right course."

Sort of puts a different spin on things really, doesn't it?

Rami Ismail announces gamedev. world, running June 21–23

Original Wii Shop Channel closes for good; tears shed

Incoming

Atomic Heart

Imagine an alternate version of the Cold War-era Soviet Union, where futuristic things like robots and holograms nestle alongside cathode ray televisions and kitsch wallpaper. That's the setting for *Atomic Heart*, the upcoming shooter from Russian developer Mundfish. The game's already received plenty of attention thanks to the fidelity of its graphics, but what we're most excited about is how imaginative its environments look. Its abandoned science facility is filled with exotic-looking robots and surreal anomalies – one recent trailer shows a brood of chickens suspended in some sort of jelly, marauding water tentacles, and crazed mannequins with unnerving kung-fu skills. *Atomic Heart*'s creators cite Stanislaw Lem, the Strugatsky Brothers and *Black Mirror* as influences, so beyond all the technical brilliance on display, we're hoping to find a compelling piece of science fiction, too.

Release date: TBC 2019

Kine

Guide a trio of spindly robots around blocky obstacle courses in this forthcoming 3D puzzler, currently being developed by Gwen Frey. Each robot can be manipulated so that it can span chasms or flip over walls; the aim is to collect pages of sheet music and avoid falling to your doom.

Release date: TBC

Tales of the Neon Sea

It's *Blade Runner*, but rendered in a captivating early-nineties pixel style. *Tales of the Neon Sea* is a cyberpunk adventure about a crumpled detective navigating a dystopian cityscape of cyborgs, seedy bars, and general malaise. We don't yet know what mysteries we'll be asked to solve, but developer Palm Pioneer's murky world looks well worth a careful investigation.

Release date: TBC 2019

The Bradwell Conspiracy

If games have taught us nothing else, it's that working for mysterious tech companies is a terrible idea. In first-person adventure game *The Bradwell Conspiracy*, you wake up in a seemingly deserted museum following an unspecified disaster.

Armed only with a pair of augmented reality glasses, your task is to find out what happened, and what your employer, Bradwell Electronics, has to do with it all. Austin Wintory is composing the music, so we already know this one will sound spectacular.

Release date: July 2019

Family Man

A blocky world straight out of *Minecraft* gets a jolt of Coen brothers-esque weariness. You play an increasingly harried father who faces the choice of working a regular job that barely pays the bills or descending into a life of crime. Forget the high-octane gratification of *Grand Theft Auto*; *Family Man* faces you with mundane activities like flipping burgers and boat theft. And, at the end of it all, a gnawing sense of sadness and guilt. It sounds thoroughly miserable, and we can't wait to try it.

Release date: TBC 2019

Wira & Taksa

Hailing from Peru, here's a jolly-looking platform puzzler that combines gravity-flipping action (akin to Terry Kavanagh's cult hit, *VVVVV*) with a mechanic that allows you to switch between the two central characters. Taksa is fast and nimble, while Wira is slow and cumbersome, but wields a gigantic hammer that he can use to smash enemies. No prizes for originality, then, but the puzzles and presentation make *Wira & Taksa* worth your consideration.

Release date: July 2019

Road to Guangdong

We're seeing an increasing number of indie developers take a more measured approach to the driving genre of late, and *Road to Guangdong* is an upcoming example. Like *Jalopy* (that earlier game from publisher Excalibur), this is a kind of road-trip drama that's as much about the relationship between its two lead characters, a young woman and her aunt, as it is about cruising along an open road.

Release date: May 2019

PLASTICINE, NOT PIXELS

Real-world materials and video games

WRITTEN BY
FRANCESCA HARRALL

With digital artwork so easy to create, why do some devs favour using physical materials like paper, plasticine and wood in their games? Wireframe finds out

A fingerprint, a fresh crease in a piece of card, a smudge of dirt – while errors in digital game design can be erased and fixed, physical imperfections are encouraged in games made from real-world materials. In fact, that's the point of choosing tangible items to craft with; designers want the flaws and the irregular feel of a piece of fabric or a rusty nail. Without them, their games lack something physical that often can't be re-created in a 3D modelling package.

For decades, we've seen alternative art styles from studios that have harnessed the irregular feel of organic materials; they've reshaped and formed them into imaginative environments or new characters. Recent and upcoming examples include the otherworldly adventure game, *Sluggish Morss: Pattern Circus*, beautiful paper adventure, *Lumino City*, and the nautical *Harold Halibut*, due for release later this year. They're proof that the interest in games diverting from more traditional digital art styles is far from dying out. ➔

^ 3.5 tonnes of clay were used in the making of *The Neverhood*.

Ready-made designer

"Sometimes we don't even have requirements", says Luke Whittaker, co-founder of State of Play Games, when I asked him about the process of hiring people to work on physical medium games. "[One future employee] turned up with a miniature model of himself in a presentation box, marketing himself as a ready-made designer in waiting. We weren't looking to employ anyone, but took him on based on that. He was soon making models, designing pinball tables in *INKS*, and puzzles in *KAMI 2*, so his role changed on the job."

WHERE IT BEGAN

There's something about the organic feel of real-life fabrics, clays, and metals that some developers actively seek out. The games made from these materials can be puzzles, fighting games, platformers, RPGs, or something else altogether. They can be creepy, dreamlike, wholesome, or all these things at once. Handmade games don't even have to include 3D models or sets; they can begin as an etch on a piece of parchment or a blot of ink, and transform into something breathtakingly unique: Nomada Studio's *Gris* is one recent example.

Using physical materials in game design isn't an easy route to take; there are all kinds of practicalities to consider when you're manipulating wires or bending plastic. There's the cost of labour and supplies. Materials might not look or behave the way you expect them to: they can break, burn, and collapse, and there's no 'undo' key to rescue them. You also have to consider how to actually make things; how to translate the physical into the digital. Often, this involves painstaking efforts to photograph, digitise, and animate your once-blank pieces of paper and shapeless lumps of clay. Then there are the pressures of deadlines to consider. All of which raises an obvious question: why do developers even bother?

Let's start our journey back in the 1990s, with a generation of games that began to use different materials to bring their characters to life. Among these were the *ClayFighter* titles, the first released in 1993, which used stop-motion animation and dollops of plasticine to bring their combatants to life.

By the 16-bit era, home computers and consoles were advanced enough to display digitised frames of stop-motion animation; you couldn't necessarily see every thumb print or scratch due to the systems' low-resolution, but games like *ClayFighter* and *Claymates*, both created by developer Visual Concepts and publisher Interplay, certainly had a more organic feel than most other games available at the time.

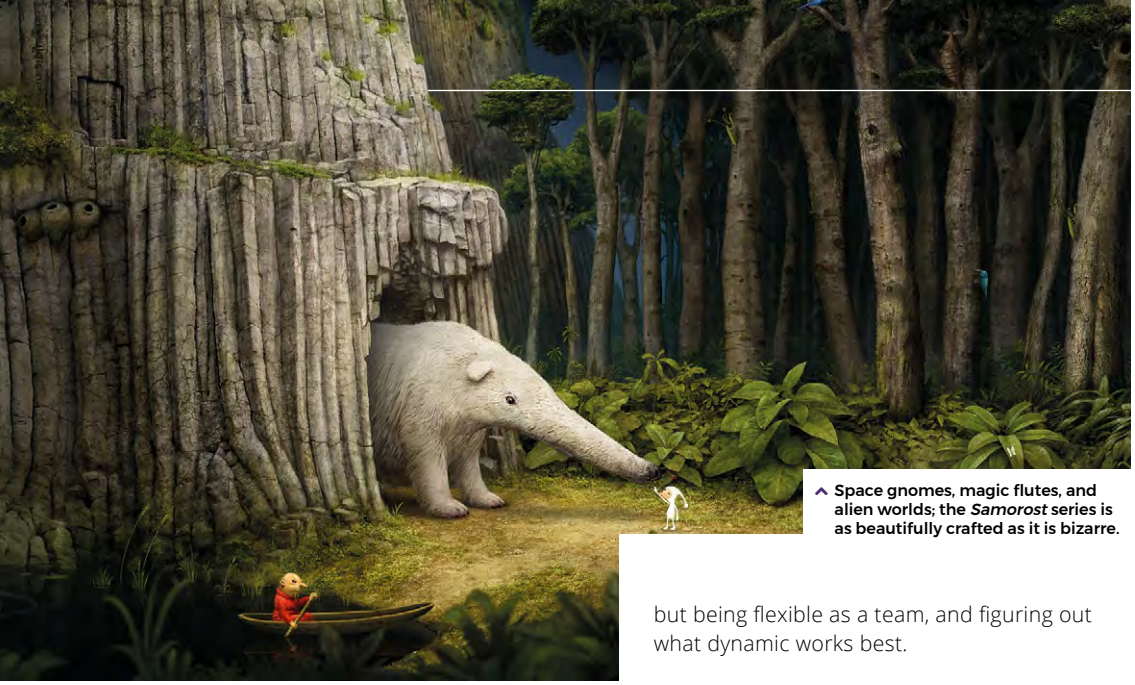
One of the most ambitious claymation games of the era was arguably *The Neverhood*, by The Neverhood, Inc – a studio founded by *Earthworm Jim* creator, Doug TenNapel. Released in 1996, it was a point-and-click adventure created by using over three tonnes of real, actual, physical clay. At the time, 3D polygon-based games, while relatively new, were very much in vogue, so *The Neverhood* – in all its quirky, uneven glory – was definitely an outlier.

Since *The Neverhood* opened those clay floodgates, there has been a steady flow of similarly creative games, made with a wide range of materials and styles. The Czech Republic's Amanita Design is just one independent studio with a knack for creating handmade games, each with its own art style. From hand-drawn adventures such as *Questionaut* (a short web game made for the BBC in 2008) and the gorgeous, award-winning illustrations of *Machinarium*, to the surreal *Samorost* series, Amanita's games have a style and atmosphere you can safely describe as 'completely unique'.

› Erik Zaring working at the IKEA desktop, where *The Dream Machine* was created in its entirety.

CRAFTING THE RIGHT TEAM

Building a harmonious dev team is no easy feat, especially when you have your sights set on designing your entire game from real-life materials. All of the usual technical aspects of game design still have to be worked around, like coding it to work the way you want it to or writing a believable story. But additional factors come into play – you have to consider building



Space gnomes, magic flutes, and alien worlds; the *Samorost* series is as beautifully crafted as it is bizarre.

real sets, figure out which materials will work and which won't, and potentially cope with something falling to bits, requiring extra time to correct it. This involves hours of physical labour and the willingness to bend your plans – and your artistic vision – to make unexpected changes fit, all without warping your end goal. "If an idea couldn't be built on a small IKEA desktop, with some dirt and two twigs, we just had to come up with a different idea," says co-founder of Cockroach Inc, Anders Gustafsson.

THE DREAM MACHINE

Similarly, Jack King-Spooner, creator of bizarre platform adventure game, *Sluggish Morss: Pattern Circus* says: "When making the models, I like to experiment. Some I simply take a bit of modelling clay and a bit of wire and play until something or other starts revealing itself. Some characters have a bit more planning behind them, with sketches and concept art."

"What is important is that people share the same love of creating things"

Experience as a developer *and* having worked with real materials in game design, or having made a handmade game, would make a very rare applicant for any studio. Luke Whittaker, co-founder of BAFTA award-winning studio State of Play, explains: "What is important is that people share the same love of creating things. There's a general outlook on what games are and could be, which we look for. Sometimes we'd look for skilled model makers to help with certain scenes, but the team has been multidisciplinary. Our developers have an artistic eye, and our artists understand the basis for how things will work."

So it seems that experimenting is at the heart of handmade games, not only in fusing together various materials to make a vision come to life, but being flexible as a team, and figuring out what dynamic works best.

TEAMING UP

Consider, though: what about when your aesthetic vision is progressing nicely, but the game mechanics or story are lacking something – do you scrap the idea? Not necessarily. It may be that it's time to bring in fresh minds or work with other developers to kick the game back into gear. Did you know that *Kirby's Epic Yarn*, for example, was never originally a Kirby game? Beginning its life as *Fluff's Epic Yarn* (who later became Prince Fluff, a secondary character), Good-Feel set about deciding the game's aesthetic by littering a desk full of cloth, felt, and yarn whilst figuring out prototypes. The design was cute, warm, and safe – but the game lacked a real hook to keep players invested. Nintendo suggested transforming the project into a Kirby game, which would add a level of structure and fun. Integrating Kirby, an existing character, gave the game the substance it needed – his presence turned a lacklustre game with an interesting aesthetic into something truly worth playing

HAL Laboratory, which got involved at a later stage in character creation, was glad to be taking a surprising direction with the design of the game, when other developers at the time were trying to get closer to realistic-looking digital graphics. The real-life materials – digitally photographed and 'pasted' over in-engine polygons – heightened the warmth and cosiness of a cute character, but in a new and unexpected way. It helped to show that bending original ideas and encouraging new ones is often a necessary part of turning something good into something great.

UNPREDICTABILITY

The very nature of designing games built from everyday objects is unpredictable. Whether

Handmade design doesn't have to just be clay or paper; *Machinarium* involves some pretty complex illustrations.

Stop-motion Kombat

The early 1990s saw *Mortal Kombat* famously digitise real actors into a gory brawler, but it was Visual Concepts that took things to a more tactile level. *ClayFighters*, released in 1993 on SNES (and later, Mega Drive) featured a claymation style of animation in a game parodying the likes of *MK* and *Street Fighter*. Its animation was achieved by making real clay (or plasticine) models of combatants, and laboriously digitising traditional stop-motion photography via that most reliable of workhorses: an Amiga.

Anders Gustafsson, one half of Swedish indie studio, Cockroach Inc.

Until you see Luke standing next to the set for scale, you don't realise just how vast the handmade *Lumino City* set really is.

A tiny portion of the intricate set of *Lumino City*, made from paper, card, and mini lights.

you're building dollhouse-esque sets in your bedroom like the Slow Bros. with *Harold Halibut*, crafting weird dreamlike environments from condoms and pork chops in a small basement, à la *The Dream Machine*, or using paper and wires in a huge handcrafted set (as in *Lumino City*), there are bound to be setbacks or surprises along the way. Whether that's because materials themselves can act differently to what you expect, or because they spontaneously decide to crumble halfway through building a set or character, designing games using real-life odds and ends can – and does – present unique challenges that all the planning in the world may not prepare a team for.

Consider how some everyday objects may look inside a game: using recognisable items like bottle tops and labels is going to have a profound effect on the surrounding characters. For example, they could make your world seem 'miniature' – think of how Yarny looks in the surroundings of *Unravel* compared to the life-size furniture, cups, and plants. If it's not an effect that fits your storyline, then it may need to be swapped for something less conspicuous. Your time frame always has to be kept in the back of the mind – figuring out which materials work and rebuilding sets that look wrong or fail in some way can take weeks or even months, and can put a big dent in deadlines.

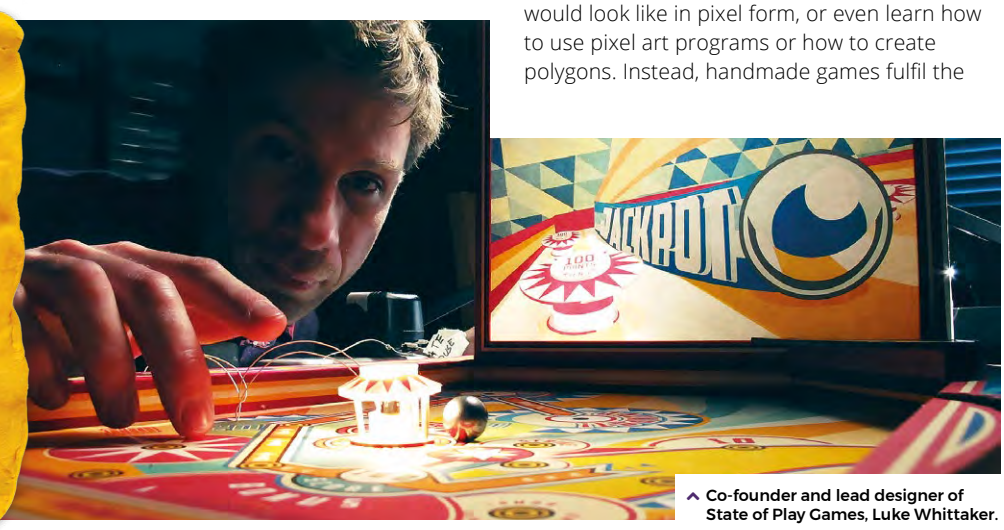
Learning about materials and how they connect to the characters and story is a fundamental part of getting better at creating games using real-world materials, but by spending too long on one area, it's fair to say others get neglected and the entire game could suffer. "We somehow spent three weeks trying to get the roof of the photographer's house right," says Whittaker. "It looks like the leather folded front of an old camera, but just using one taken off a camera wouldn't have looked right. So we spent ages folding paper, refolding and colouring it with different inks. It took too long, really, but it did help us work out that visual language."

INVOLVING THE COMMUNITY

There's more than one way to enjoy the game worlds created for us by talented developers. When we're not playing the levels and have powered off our consoles and PCs, we can find comfort and enjoyment in the characters – and the worlds they live in – by creating our very own versions of them. With handmade games, this is even more rewarding, because what we craft out of crochet, papercraft, or plasticine actually looks just like those gracing the screen. We don't have to try and imagine what sketches or paintings would look like in pixel form, or even learn how to use pixel art programs or how to create polygons. Instead, handmade games fulfil the

Craft in work

"I've rediscovered a love of making things by hand," says State of Play's Luke Whittaker, "and learned that other people respond emotionally in a similar way to me to seeing craft in the work. It seems like a crazy pitch, to make a game by hand, and that's partly why we self-funded – I didn't want to have to justify it to publishers. I've learned that if it feels right, it's worth doing. Others will likely feel the same."



Co-founder and lead designer of State of Play Games, Luke Whittaker.

creative parts of those outside of gaming, too. Think of *Yoshi's Woolly World*, a game designed to look just like a cosy landscape formed from plush knitted characters, felt levels, and fabric scraps. Whilst most of this is computer-generated, the amiibos of *Yoshi* are made from real yarn, thus expertly transposing the faked look of the game to a very real-world version of things. A very real-world *adorable* version, that is.

The developers, the same as those for *Kirby's Epic Yarn*, unsurprisingly, initially decided on a fabric *Yoshi* world to bring the warmth and kindness of the character out in a whole new way. What better way for fans to bring him to life off-screen than by crocheting and knitting their own real-life characters, monsters, and levels?

It's not just Nintendo, either – capitalising on the popularity of the *Unravel* series, EA released an in-depth guide on how to make your own version out of yarn and wire, inspiring fans of the game to experiment with his design. Hundreds of fans have since shared their creations on social media, popping their own Yarnys into natural environments just like the Swedish countryside found inside the game itself. Real-world materials allow fans to carry on the adventure in real life, strengthening the bridge that developers build when imagining their games, using everyday items that fans can replicate.

THE FUTURE

Whether we're talking about worlds crafted purely from raw materials and everyday items, ones where designers have reimagined or replicated the feeling through digital artwork, or if it's somewhere in between, there's no doubt players have a hankering for an authentic experience. We're reaching a point in game development (particularly in the indie scene) where tools and platforms for creating game environments are becoming easier to access and cheaper than ever, so everyone can try their hand at bringing their imagination and hard work to life for us to play and enjoy.

Can the same be said for handmade games, when much of the work is physically crafting the landscapes, characters, monsters, and puzzles?

Certainly – if the game warrants a particular style, and it's in keeping with the game storyline and characters. Game designers will always yearn to create in the way that their visions guide them, but it's unlikely it'll ever replace 'traditional' digital designing. "At the end of the day, it's just a technique. It's just a way to execute an idea. If it doesn't suit the idea, or if the idea stinks, real-world materials won't save it," says Anders Gustafsson, and he has a valid point. Although we can all appreciate a game's impressive graphics, or when a studio tries something new, games are not one-sided and cannot be great, innovative, or enjoyable by relying solely on one of its attributes.

"If it doesn't suit the idea, or if the idea stinks, real-world materials won't save it"

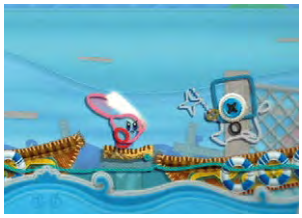
REAL VS FAKE

All parts of a game have to work in harmony for them to be fun and engaging. A beautifully designed game with hollow, shallow characters won't work, the same as an aesthetically gorgeous puzzle game with awful physics won't get away with being sub-par, simply because it's nice to look at.

In the future, the 'feeling' of handmade games could theoretically get easier to produce, with photorealistic imagery becoming less complicated to make; though whether you'd want to is another question.

The flaws we commonly see in handmade games would have to be reproduced in CG, which is an expensive process by itself (unless, of course, you have the financial backing of a company like Nintendo). The process could also be as time-consuming as building a real set, and introduce a whole new host of technical problems by itself.

Besides, it's those little flaws – the creases, the fingerprints, the peeling bits of tape – that make games made from real-world materials so engaging in the first place.

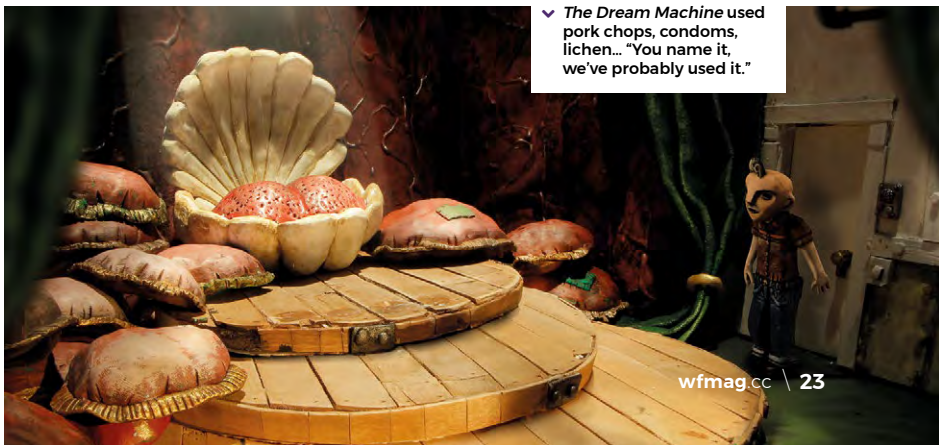


Kirby's Epic Yarn **inspired a re-release on the 3DS, coming in March 2019.**

Kirby's clay inspiration

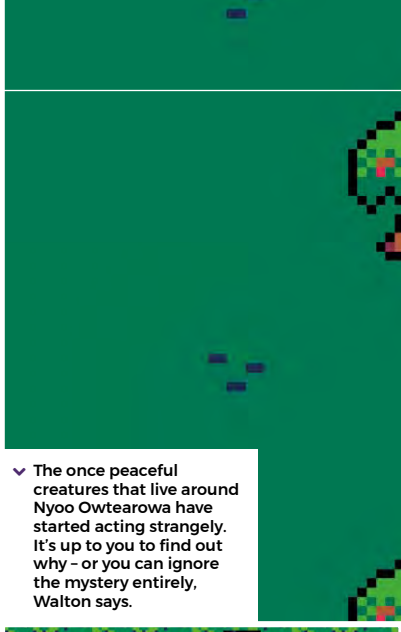
Following the successful plushie-inspired Kirby game, *Kirby's Epic Yarn*, Nintendo and HAL Laboratory crafted yet another game inspired by the real world, this time in faux-clay: *Kirby and the Rainbow Curse*. Released in 2015 on Wii U, this tribute to claymation is a gorgeous example of how well Kirby fits into a cosy, warm, handmade environment. Maybe one day HAL will move into the realm of actual, physical sets.

Breathtaking Swedish countryside has inspired fans of the ***Unravel*** **series to create their own Yarny models.**



The Dream Machine **used pork chops, condoms, lichen... "You name it, we've probably used it."**

› "Other than creating and implementing all the battle moves, which took quite some time," Walton tells us, he'd designed and named all his monsters "in two days."



› The once peaceful creatures that live around Nyoo Owtearowa have started acting strangely. It's up to you to find out why – or you can ignore the mystery entirely, Walton says.

Interactive

Notemon, the tiny monster-catching RPG made in PICO-8

Meet Nick Walton, the solo dev who's making a Pokémon homage with a New Zealand flavour

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

In September 2018, solo developer Nick Walton headed to YouTube to make a confession. "Making *Notemon* was a mistake," he wrote in his video's description, "but I'm committed to it."

Nine months earlier, Walton had just begun work on his monster-catching RPG: a blocky, colourful, free-roaming game developed in the virtual console, PICO-8. But while Walton had made plenty of games before, *Notemon* quickly proved to be a more ambitious proposition: as well as monster management and exploration, it also contains such elements as combat, fishing, and quests. In other words, *Notemon* isn't just one game, but several games, each woven carefully into the other. As work progressed, Walton gradually realised how much work all these elements would require.

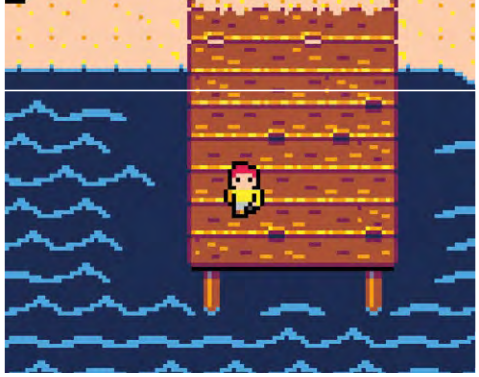
"With my previous games, I basically picked one small thing to make the game about," Walton tells us. "I think it would have been better if I'd broken *Notemon* down into pieces and made a small game about each part – a turn-based battling arena game, a game where you walk around talking with NPCs, a game where you collect items for quests. Then after I had that experience I'd be able to put it all together to create one great RPG."

"I think it would have been better if I'd broken *Notemon* down into pieces"

Despite the steep learning curve, Walton's stuck with his *Pokémon*-like monster-catcher, which has evolved from a straight fan-game to something far more personal. Before he moved to Houston, Texas in 2015, Walton grew up in a small town in New Zealand, and *Notemon* is heavily informed by his home country's culture. The action is largely set in a peaceful fantasy world called Nyoo Owtearowa, and its inhabitants' dialogue is all written in a phonetic approximation of the New Zealand accent.

"*Notemon* doesn't have any voice acting, but I still wanted a way to portray a New Zealand accent, as according to my headcanon that's how the characters would speak," Walton says. "The idea for writing phonetically just came to me by thinking a little about that problem, and after having some people read it, they all

A shot from *Notemon* when it was still in its early stages back in January 2018. It's changed quite a bit over the past year.



"For me, PICO-8 is a cosy crucible, charming in its presentation yet forcing me to become a better programmer," Walton says.

Walton used PICO-8's built-in sprite and map editors to create his characters and gentle fantasy world.

understood the words and pronounced them the way a New Zealander would, so I went with it!"

DOWN ON THE FARM

Notemon also departs from *Pokémon* in other ways besides its dialect and setting. For one thing, players won't have to fight monsters in order to capture them; instead, Walton says, "you build a barn and give them a home so they'll join your team." There are other changes, too, including stats that restore after a battle ("You don't have to walk back to a recovery location halfway through a cave!") and moves that can be changed as you level up, rather than being overwritten.

Notemon might look diminutive compared to the latest entries in the *Pokémon* series, but Walton remains intent on pushing the virtual console's limitations as far as he can. There are 20 unique monsters to catch, an expansive game world with multiple locations spread over nine maps, and a non-linear story that the players can, Walton says, engage with or ignore as they see fit.

"If a player just decides to do the main objectives in the game, they'll get a lot of gameplay out of it," Walton says. "It's up to the players how much they get out of *Notemon* – if all they want is to build the barn and completely ignore what's going on in the world they can do that, too."

Fitting all those monsters and locations into PICO-8 has, Walton tells us, been one of the biggest challenges of making *Notemon* – though he also adds that fighting against the system's limitations has been a valuable learning experience.

"I have a love-hate relationship with PICO-8," says Walton, "but ultimately I think it has been very educational to me as a game designer and programmer. PICO-8 has a cartridge size of 32kB, and I chose to make *Notemon* with it so I'd be forced to decide what's really important to the game and cut the rest. Within a very short time, I found myself hitting the code token limit without even having all the core features done, so I figured out a way to spread the game across [currently] 12 cartridges which all automatically load between each other when necessary."

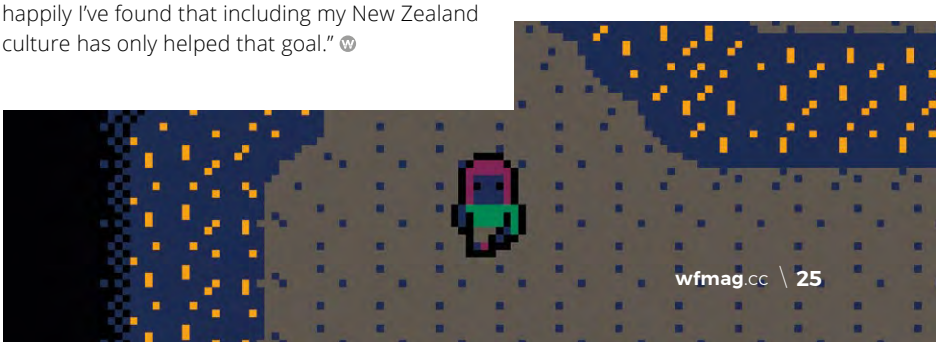
In summary, then, Walton no longer feels as though he's made a mistake by embarking on his deceptively complex top-down RPG project. "I'd say I'm over the hump," Walton says, before listing some of the things he has planned for its release later in 2019: it'll be available on Steam, Game Jolt and itch.io for Linux, Windows, and Mac as well as PICO-8, and there'll be a free demo for would-be monster-catchers to try out, too.

Aside from the appeal of catching and housing tiny, tiny monsters, it's those personal touches mentioned above that really endear *Notemon* to us: the quirky NPCs that say things like, "Aw gidday" or, "My old bahn, shee's prity bustid up aye!" For Walton, it's these little cultural flourishes that can make an indie game truly unique.

"I think the more cultural variation we can have in games, the better," Walton says. "My aim with *Notemon* was to make a great game, and happily I've found that including my New Zealand culture has only helped that goal."

TUTORIAL MODE

For Nick Walton, *Notemon* is the latest step in a hobby that's been developing since he was a kid growing up in the quiet town of Feilding, New Zealand. Early experiments with GameMaker as a 10-year-old led to a lasting interest in development. "My friends all dropped off game dev, but I kept on with it as a hobby, learning C++ and other languages, trying some game engines and whatnot," Walton says. Along the way, he's found online tutorials an invaluable – and often free – means of learning more about making games. Says Walton, "I learned everything I know through experimentation and free online tutorials, which is why I've just started making my own tutorials to pass on some knowledge." You can find Walton's own YouTube channel at **wfmag.cc/VmwSRi**



Violent delights have violent ends

LOTTIE BEVAN

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

I'm a cowardly, escapist completionist but even I fell prey to *Resident Evil 4*. The grimy art direction, the wet squelch of gore, the prolonged stress and terror, the weirdly forgivable sexism, and (if I'm being honest) Leon S. Kennedy's fringe all added up to a surprisingly fun romp through the Costa del Zomb.

Resis 5 and *6* passed me by, but I was ready for 7. I'd heard it tried something new: more immersive than ever. More violent than ever. More frightening than ever. Fewer hokey Jill sandwiches, more maggoty, actually-human-flesh sandwiches. Due to my aforementioned wimpiness, I looped in a friend with an *Umbrella Corp* tattoo. In exchange for a few stir-fries, when the game launched she played it at my flat rather than hers so I could sort of play it, too (occasionally cowering behind a wok). Many nauseating hours later, we looked palely at one another and put the controller down. It was too much. The first-person perspective, the high production values, the basic human horror at being kidnapped, mutilated and often killed by a bunch of semi-mutant cannibal maniacs. Horror is fun when it's a swirl of fear, safety, tension and release. It's explicitly not fun if it actually happens to you. I paid £40 for the entertainment experience of role-playing a horror victim. Ethan Winters would pay a whole lot more to never experience that again.

The much-hyped *Resident Evil 2* remake is, at the time of writing, about to launch. It looks fantastic, and not just because Handsome McFringeboy is back. But I'm sadly unlikely to play it. Once again, it takes horror to its logical next step, but it isn't necessarily a step closer to fun. For example, it reintroduces Mr. X, a basically indestructible Dr. Manhattan lookalike who's been inexplicably kitted out with a trilby you can shoot off his big scary head. He sets a frenetic pace: no careful supply sweeps, no room-by-room zombie culls, no time to catch your breath. Mr. X and his tiny hat force you onwards into the fray, dodging fights, missing pick-ups, and making you do dangerous, stupid, exciting things. It's much more like actually being Claire Redfield on the worst day of her life, and I can see why the designers don't want people literally playing it safe. But Claire doesn't seem to be enjoying herself, and as my experience now closely mirrors hers, neither am I.

I respect *Resi 2*'s decisions. It's my fault I'm a coward, not the game's. But I think this will keep happening. Increasingly realistic blood and gore will necessitate increasingly nasty art, animation and events to inspire the same level of revulsion. Increasingly normalised violence leads to increasingly unshockable players, which may force horror game-makers to nastier and nastier places. These violent delights have violent ends, and the more intense they get, the larger the cohort of the left-behinds. The *BioShock* players. The Dead Spacers. The Alien: Isolationists.

'Dweeb Island' isn't a cool address, but the real tragedy is its inhabitants missing out on otherwise excellent games. There will undoubtedly be a rise in alternative horror to fill the growing demand, and at some point – surely – even video game nasties will reach their grim apex. For now though, I'll just have to get the vegetables in. Anyone fancy dinner round mine?

"It's explicitly not fun if it actually happens to you. I paid £40 for the entertainment experience of role-playing a horror victim"

Broad shoulders, tiny hat: *Resident Evil 2*'s terrifying T-00, also known as Mr. X, or Trenchy to his friends.

Toolbox

The art, theory and production of video games

Perfect a smooth Super Mario-like jumping action in Source Code on page 32.

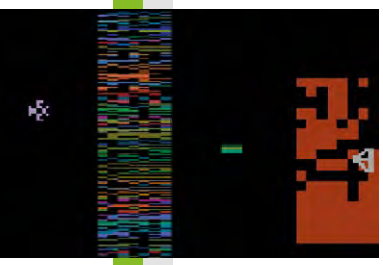
To make a great city, you also need to consider its surroundings – see page 30.

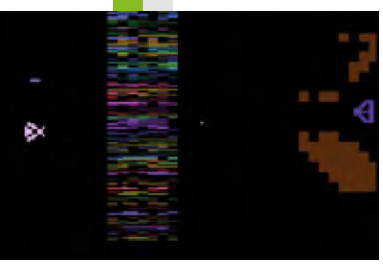
The principles of game design

Great game design requires a mixture of economy and theatre, Howard writes

AUTHOR
HOWARD SCOTT WARSHAW

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





Tricks like cycling colours and altering sizes created a lively-looking game in just a few kilobytes of memory.

I came to Atari with a bachelor's degree in mathematics, economics and theatre, as well as a master's in computer engineering. Most people think the maths and computer engineering were the keys to success at Atari, but I believe it was the economy and theatre. After all, making games on the 2600 is about finding the most economical approach to maximising entertainment value. When you only have 4K of ROM for code and graphics, and a precious 128 bytes of RAM (including the stack), well... to paraphrase Monty Python: every byte is sacred, every byte is great. The big question, then, is how best to use them?

Keep in mind that my latest game wasn't yet called *Yars' Revenge* during development – it was an unnamed project originally assigned as a conversion of the arcade shooter, *Star Castle*. My working title for the game was *Time Freeze*, but there was no story concept yet. This means there was nothing to suggest which way to go next – but then again, there was nothing restricting my vision, either. Unfettered by storyline, I only had to make everything on the screen more visually stimulating. Naturally, the first place I looked was the low hanging fruit. What was easy to do on this machine?

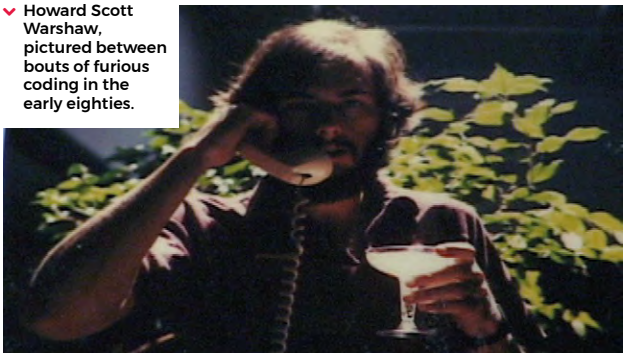
For the player's Zorlon Cannon, I animated it very simply by alternating its horizontal size: this gives the weapon a pulsating impression which adds motion to the screen even before it's fired. Another thing that's easy to animate on the 2600 is colour: it's a simple way to add another visual dynamic. With the Qotile (the game's antagonist), I cycled its colour slowly through the rainbow, which gave it the feeling of morphing; this foreshadowed its transition to a dangerous super-weapon.

Both techniques added to the visual impression of the game without any additional bytes for graphics. The ultimate example of this kind of economy came when I decided the centre-left of the screen was too stark, and needed something for visual balance. What could I stick in there that would look cool? The idea of using the game's code itself for both graphics and colour (as detailed in the sidebar of my last article) was a creative and cost-effective solution, and it's just so shiny and twinkly. As a special bonus attraction, it happened to automatically animate the colour of the Zorlon Cannon as well.

INNOVATION

No one had ever seen anything like this on the 2600. When people walked by the lab, they would stop to exclaim, 'What is that?' It had no game purpose at that point, and it didn't matter. It was catching eyeballs. It was working!

When it came time to create my big payoff sequence, the elaborate animation I'd originally planned seemed like too much work.



Howard Scott Warshaw, pictured between bouts of furious coding in the early eighties.

The original release came with a comic book which helped fill out the sci-fi backstory to *Yars' Revenge*.

Instead, I simply riffed on this Ion Zone technique to create the first full-screen explosion in video game history. Why? Because it was dazzling and easy to do. Also, each of these enhancements was associated with a simple but unique mini-sound algorithm.

Animations were key as well. The Yar's flying animation was originally done for visual interest, but I added the more elaborate death sequence for punctuation. And the second level shield had rotating bricks which weren't apparent until the first time it was breached; then the increased motion becomes an interesting new element. When it came time to structure the completed game experience, that's when my resistance to convention really paid off.

I liked the clean feel of the unframed black background; it gave great contrast for the glittering colours on the screen. When played at night with the lights off, the game melted into the darkness of the room and made it feel bigger. I liked that effect. Why mess it up with a score? Besides, doing a separate screen for scoring gave me several advantages.

First, I didn't have to mess with my original code to add anything to the main screen. It was all working, so why change it? A separate scoring screen was trivial to program. But what's huge is turning control over to the player. Now *they* could decide when to restart the game with a joystick button push. This created a significant innovation: *Yars' Revenge* was the first video game with pause mode. Most video game thinking at the time was derived from coin-ops, where a pause mode was unthinkable. Not so for home games. I also added the ability to reset and launch a new game from the joystick after a loss. So, next time you pause or restart a game with a simple button push, remember to say, "Thank you, Howard."

A GAME OF FIRSTS

Yars' Revenge broke a lot of new ground. On top of pause mode, reset from joystick, full screen explosions, and elaborate colour and sound design, *Yars' Revenge* was the first game to display its code on the screen. *Yars'* didn't have the first Easter egg in video games (that was *Adventure*), but it was the first time the marketing department approved, and got behind, the Easter egg concept. It was also the most tested game in Atari history. Ultimately, it was play-tested against the 2600 version of *Missile Command*, and *Yars'* won.

It was nearly the first VCS game to become a coin-op, too: Dave Theurer (*Tempest*, *Missile Command*) was actually considering making an arcade version of *Yars*. (Spoiler Alert: he didn't.) It was the first Atari game to credit its programmer. Atari's perennial refusal to do this was a sore point, which led in part to the formation of Activision. And there's one more industry first for *Yars*, but this one will require more space: next time, I'll talk about the naming of *Yars' Revenge*, and creating the first video game backstory.

"Making games on the 2600 is about finding the most economical approach"

Test match

No game was tested more than *Yars' Revenge*. Each time it seemed the game was going out, some concern was raised and another test was ordered. The test would go well, release was scheduled and then another concern arose. I had a bad case of Releasus Interruptus. It was killing me. The real drug at Atari was seeing your game in stores and on TV – I wanted that high so bad I could taste it. And when it finally happened… it was so worth the wait.

Heavily advertised in the nascent games press, *Yars' Revenge* was a hit in 1982.

CityCraft: designing a city's surroundings

What surrounds a city is just as important as the setting itself, as Konstantinos explains

AUTHOR
KONSTANTINOS DIMOPOULOS

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

Styles of Empire

Empires, nations and great collectives spanning star systems often tend to exhibit a more or less unifying visual style. From a commonly seen flag to a repeated building typology or prevalence of a colour, some sort of aesthetic cohesiveness is usually displayed. This isn't mandatory, of course, but repeated architectural and planning rhythms can make an imperial space instantly recognisable as such. An orthogonal town with an amphitheatre, columned temples, and a rigid grid-based road system would, for example, be instantly identifiable as part of the Roman Empire.

Fig 1: By-rivers link into rivers flowing away from linear mountain ranges and towards the sea, providing humans with settlements options.

As a strategy, sieges are based on the assumption that the vast majority of cities could never grow their own food. It's true that surrounding areas (or hinterlands) most commonly sustain urban centres, and that vast fleets of freighter spaceships keep fully urbanised planets like *Star Wars*' Coruscant and *Warhammer 40,000*'s Terra surviving. The politically dominant hubs of vast transportation, production and financial networks were the first to grow to metropolitan sizes.

Cities, you see, cannot exist in a vacuum. They need to be connected to other settlements, function as parts of wider political organisations, and be in a constant dialogue with their wider environments. Of course, we don't need to be reminded of such realities when recreating New York City, or when setting a game in Istanbul, where real-world geography is plain to see. Any attempt at imaginary city building, though, must take wider surroundings into account; both those sculpted by nature, and by human hands.

NATURAL GEOGRAPHIES

Coming up with an environment for your city involves decisions regarding topography, geology, climate, flora and fauna: where, and in what sort of world, is your city located? In many cases, such answers involve the setting's universe (is it ruled by a demented pantheon of prankster gods?), planets (can urbanism survive only in protective domes?), continents (do they all share similar species?), and an abundance of details for the specific locations of specific urban centres. Getting inspired by real topographic, geological and other thematic maps worked wonderfully for *A Song of Fire and Ice*, and can easily work for your world, too.

Then again, following Tolkien's more imaginative example, one could craft everything from the beginning, keeping in mind that space is by definition cohesive.

Also, certain basic rules have to be followed to maintain consistency and ensure a sense of believability. The processes defining the formation of mountains and rivers are, for example, some of the more fundamental ones. So, mountain ranges, being the result of tectonic pulls or pushes, are almost exclusively linear, and never intersect with each other at 90 degree angles. Mountains also tend to push rivers away from their mass, and that's why rivers flowing in parallel to them often feel wrong.

Rivers, unless thoroughly magical, tend to flow from higher to lower altitudes until they reach the sea or a large body of water. What's more, rivers only join into bigger ones, and don't split up or branch until they reach their deltas. Admittedly, with nature (imagined or real) being infinitely varied, such rules needn't apply to all situations, and could be redefined by strange materials, alien technologies and magical forces. Mordor's unnatural fortress-shaped mountain range was the result of Morgoth's powerful

Networks of cities and the importance of geographic connections are evident in games such as *Civilization VI*. Urban centres cannot exist in isolation.

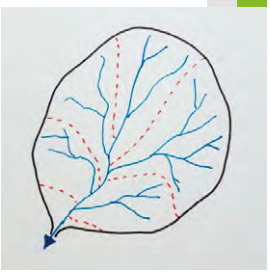


Fig 2: A drainage basin is a region in which all water is gathered and drained to a common outlet. It is divided in sub-basins.

sorcery, and in distinct contrast to the more sensible, Earth-like environment depicted in **Figure 1**.

Most intriguingly though, (human) societies don't treat their environment as something static. They change, adapt and add to it. They assign value to land as their needs and capabilities change, and choose where to locate according to their priorities. A fortified position in a valley means nothing on a peaceful planet; a natural harbour is worthless when you cannot sail.

"Societies don't treat their environment as something static"

HUMAN GEOGRAPHIES

'Pure' nature that hasn't – at least to a tiny degree – been influenced by humanity doesn't actually exist on contemporary Earth. Deserts have been made to bloom, mountains have been excavated, islands constructed, rainforests irrigated, and traces of our emissions have affected Antarctica, and the ocean floor. In addition to this, we've divided land into nations, countries, states and kingdoms, each with its borders, beliefs and economies for our settlements to function in. Nature and society both birthed the geographies we build on, and there's no reason why any imaginary construct shouldn't be inspired by our rich geographic histories.

Technologies dictate how cities are built, what activities they need or care for, and allow for elaborate fortifications. Nations and borders often lead to wars, and thus geopolitics are a powerful factor in shaping (or levelling) urban centres. During the ancient Egyptian New Kingdom's apex, the Pharaoh's peace resulted in cities without protective walls, whereas many northern towns of the declining Byzantine Empire were little more than heavily fortified outposts.

Human geography also includes connections and networks. Often, an intersection of two major trade roads led to the founding of transit towns that with time evolved into financial powerhouses. Transportation, along with cultural and trade exchanges, on the other hand, can be disrupted by non-natural barriers such as China's Great Wall, or reach new levels as telecommunications instantaneously overcome distances. Hard and soft borders, as well as state or national ideologies, can further influence architectural trends in cohesive parts of the world, civic lifestyles, or the kinds of food people can buy in a marketplace.

BELIEVABLE REGIONS

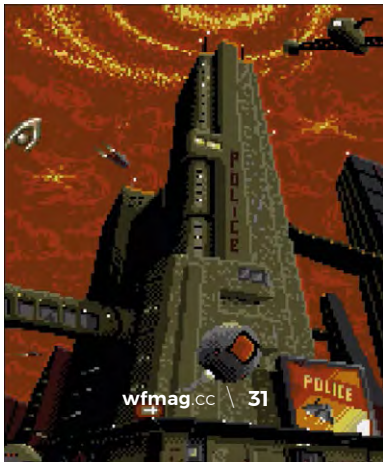
In short, location matters, and it's the combined effects of imaginary societal and natural geographies that can make a region believable, and fill in its details. *Morrowind*'s tamed Silt Striders are one such example, while Terry Pratchett's Clacks telecommunications network across the *Discworld* is another.

Believable regions have appropriate beasts of burden, economies adapted to their dynamic environments, and logical, hierarchical patterns of settlements ranging from villages and towns to cities and metropolises. Specialised settlements (port cities, holy cities, industrial cities) might make sense, too – though the ways these urban spaces are formed will need a whole CityCraft instalment to themselves.

Water and flooding

If the world you've constructed has water and gravity, then that water will flow downwards. A city's lowest points will therefore be the first to flood, and will remain flooded only if something keeps the water from flowing further away. Following this logic, drainage basins would also have to exist (see **Figure 2**), which in turn lead to rivers and lakes. This might sound obvious, but you'd be surprised at how many otherwise masterful world-builders forget about flow and flooding.

Ubisoft's Terrapolis (1989) survived harsh planetary conditions under a dome, and required temporal bridges to keep in touch with the rest of the universe.



Source Code

Super Mario-style jumping physics

Learn how to create your own Super Mario-style running and jumping action in Python

AUTHOR
RIK CROSS

Before writing any code, it's best to decide on the rules of your world. Will you allow your player to double-jump, or change direction in mid-air? There are no right or wrong answers to these questions, but it pays to plan ahead. We'll be using Python and Pygame Zero to code the game world, but the ideas are transferable to other languages.

The first thing we need to do is create a player and some platforms to jump on! As Pygame Zero has built-in support for collision detection between game Actors and rectangles, we've stored the platforms as a list of rectangles with varying dimensions.

As vertical and horizontal motion are perpendicular to each other, they can be considered independently. Horizontal motion will involve moving the player to the left or right by updating the player's x-coordinate if the arrow keys are pressed (and the player is within the screen bounds). This can be improved later, but we'll keep this simple initially to concentrate on the vertical motion. Before writing the code for vertical motion, let's look at the physics:

- Acceleration is the rate of change of velocity. Vertical acceleration is due to gravity, and is a positive value (i.e. acting downwards). Gravity will be stored as a global constant, as it acts on all game objects.
- Velocity is the rate of change of position. Initially the player's vertical velocity will be 0, as the player is at rest. When the player jumps, the velocity will be set to a negative value (i.e. acting upwards).

Both of these values change with respect to time, which for the sake of simplicity can be thought of as increasing with each frame. There's nothing special about the values chosen for gravity and jump velocity – these can be tinkered with to suit. You can also adjust the height and gaps between the platforms to increase your game's challenge.

In each frame, the following algorithm is used to update the player's vertical position:

- Add the acceleration value to the velocity value
- Add the velocity value to the position value

COLLISION DETECTION

The next thing to fix is that the player's velocity (and therefore position) isn't yet affected by colliding with a platform. One way to do this is to calculate the player's new position, but only move the player to the new position if they don't hit a platform. If there's a collision, then the player isn't moved, and its velocity is set to 0. No collision means the player is free to move to the new position. Making the player jump is a matter of setting the player's vertical velocity to the predefined jump velocity. However, the player should only be allowed to jump if there's a collision. This means that the player is touching a platform.

Jumping physics in Python

Here's a code snippet that illustrates Rik's platform-jumping physics in Python. To get it running on your system, you'll first need to install Pygame Zero – you can find full instructions at **wfmag.cc/XVIIeD**

Download the code from GitHub: **wfmag.cc/wfmag7**

```
# define screen size
WIDTH = 800
HEIGHT = 800
# define a colour
MAROON = 128,0,0
# vertical acceleration
GRAVITY = 0.2

# a list of platforms, each a rectangle in the form ((x,y)
(w,h))
platforms = [
    Rect((0,780),(800,20)),
    Rect((200,700),(100,100)),
    Rect((400,650),(100,20)),
    Rect((600,600),(100,20))
]

# create a player and define initial vertical velocity
player = Actor('player',(50,450), anchor=('left','top'))
player.w = 20
player.h = 20
# define initial and jump velocities
player.y_velocity = 0
player.jump_velocity = -7

def update():

    # horizontal movement
    # calculate new horizontal position if arrow keys are
pressed
    if keyboard.left and player.x > 0:
        player.x -= 2
    if keyboard.right and player.x < 780:
        player.x += 2

    # vertical movement
    # temporary variable to store new y position
    newy = player.y

    # acceleration is rate of change of velocity
    player.y_velocity += GRAVITY
    # velocity is rate of change of position
    newy += player.y_velocity

    # create a rectangle for the new y position
    newplayerpositiony =
Rect((player.x,newy),(player.w,player.h))

    # check whether the new player position collides with a
platform
    y_collision = False
    for p in platforms:
        y_collision = newplayerpositiony.colliderect(p) or
y_collision

    # player no longer has vertical velocity if colliding with
platform
    if y_collision:
        player.y_velocity = 0
    # only allow the player to move if it doesn't collide with
any platforms
    else:
        player.y = newy

    # pressing space sets negative vertical velocity only if
player is on ground
    if keyboard.space and y_collision:
        player.y_velocity = player.jump_velocity

def draw():
    screen.clear()

    # draw platforms
    for p in platforms:
        screen.draw.filled_rect(p,MAROON)

    # draw player
    player.draw()
```

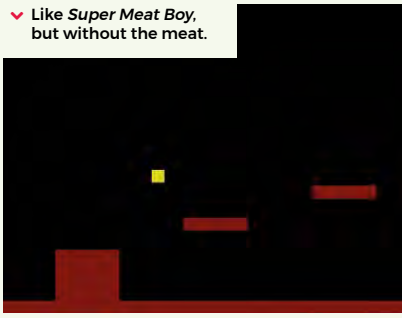
MAKING IMPROVEMENTS

We've fixed one potential bug, by only allowing the player to jump if they're touching a platform. If you run the code on the left, though, you'll notice a few other bugs:

- The player can jump if they're touching any platform, even if they're underneath. This can be fixed by only counting collisions below the player, by comparing the player and platform's y-coordinates.
- There's no horizontal collision detection, so the player can walk through platforms. This can be fixed with horizontal collision detection.
- If the player collides with a platform at high velocity, they'll stop just short of the platform and then drop slowly to the ground. One way of fixing this is to calculate the distance between the player and this platform, and move the player so they're on top of the platform they would have collided with.

These bugs have been fixed in a second version of the code, `jump_physics_improved.py`, also available in the GitHub repository link above.

Like *Super Meat Boy*, but without the meat.



Finalising your Unity first-person shooter

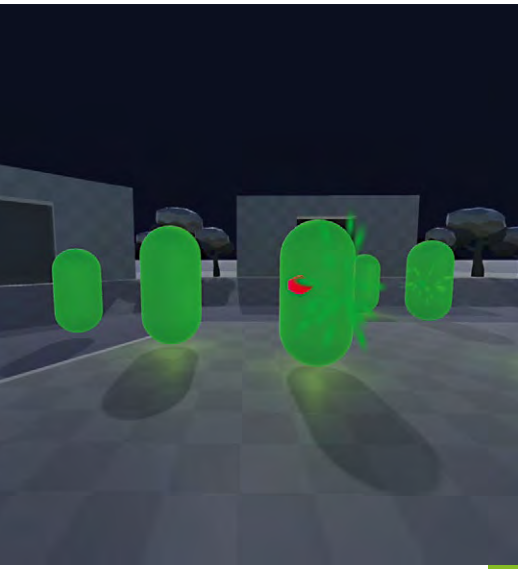
In this third and final part, we'll finish up our Unity shooter with visual effects, menus and more

AUTHOR
STUART FRASER

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

In our earlier tutorials, we looked at building a first-person shooter, and worked through adding gameplay elements and mechanics, such as firing a projectile and creating a simple enemy type. We also added a wave-based survival mode, which we'll now explore further by adding an enemy spawner and rounds. We'll also polish the game, adding a menu system and visual effects to our projectile hits.

With the addition of rounds, menus and effects, our project is really starting to feel like a proper game.

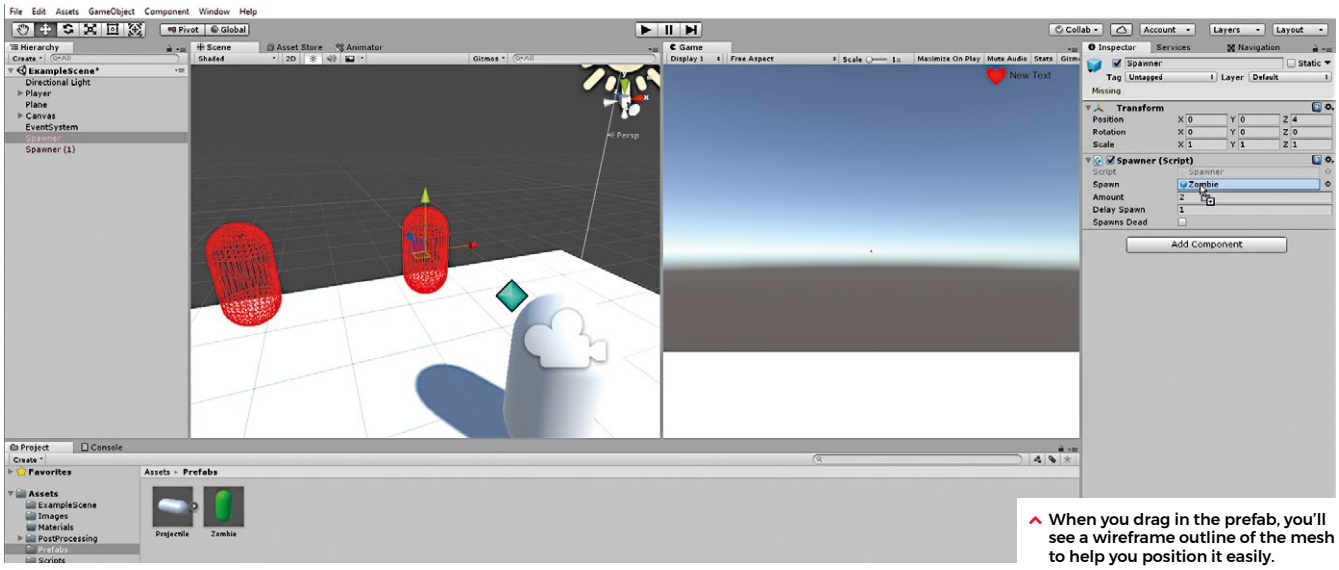


CREATING AN OBJECT SPAWNER

First, let's create a spawner that we can use to spawn multiple enemies. This is going to be very simple, and we've already used some of this logic to spawn our bullet object. We'll also expose some variables or values so that we can expand the idea of a wave-based survival mode.

To do this, we need to create a game object to be our spawner object. You can simply right-click in the Hierarchy and select Create Empty. In the Inspector window for this object, rename it to Spawner. We'll add a script that allows us to choose the object to spawn, the number of spawned objects, and a delay between spawns. In Inspector, select Add Component and then select New Script and set the name to Spawner. We can then double-click the script to open the script editor of our choice, and then replace the script with our code. Remember to save and go back to Unity when you are done.

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

public class Spawner : MonoBehaviour {
    public GameObject spawn;
    public int amount = 1;
    public float delaySpawn = 1;

    private int getAmount;
    private float timer;
    private int spawned;

    private void Start()
    {
        ResetRound();
    }

    private void ResetRound()
    {
        getAmount = amount;
    }
```



When you drag in the prefab, you'll see a wireframe outline of the mesh to help you position it easily.

```
    void Update () {
        timer += Time.deltaTime;
        if (delaySpawn < timer)
        {
            if (spawned< getAmount)
            {
                //Reset our timer.
                timer = 0;
                spawned++;
GameObject instance = Instantiate(spawn,
transform);
                instance.transform.parent
= null;
            }
        }
    }

    private void OnDrawGizmos()
    {
        //Draw the wireframe mesh of what
we intend to spawn in our editor.
        Gizmos.color = Color.red;
        if (spawn != null)
        {
            Gizmos.DrawWireMesh(spawn.
GetComponent<MeshFilter>().
sharedMesh,transform.position, spawn.
transform.rotation, Vector3.one);
        }
    }
}
```

We made our Zombie enemy a prefab in our Project in the last tutorial, so we should be able to select this, and drag it onto the slot named Spawn that is shown for the Spawner object in the Inspector window. You can then delete any other Zombie objects that are in the Hierarchy, as we can now spawn them via this spawner object.

"Let's add some particle effects to the action"

Now test the spawner by pressing Play to preview the game. The first thing you may notice is that the enemy is spawned, but it won't move, and we have an error in the Unity log. The reason for the issue is that while we have the Player in our scene, the Zombie prefab is in the Project so the engine doesn't understand this exists. This is not a huge issue, but we need to change how we're going to set the goal for our AI script.

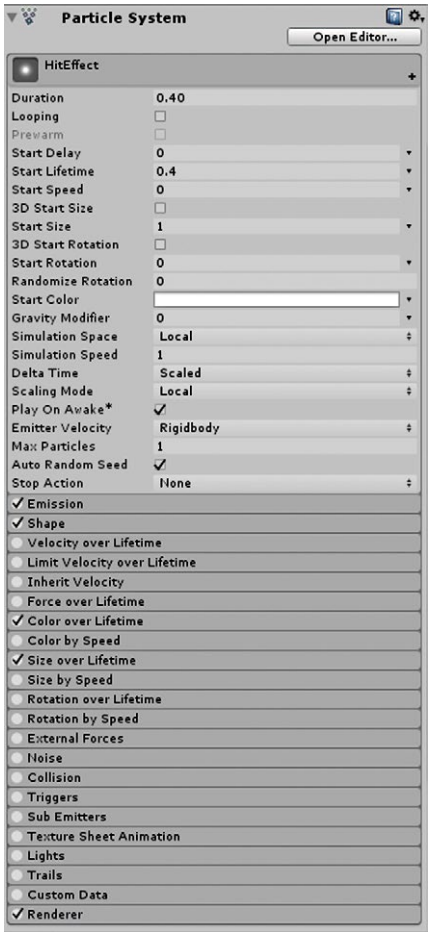
We need to stop the game playing, and then we need to select the MoveToPosition script. The fix is going to use the tags that we looked at in the second tutorial. We'll tell the script to set the goal to the object with the tag Player as soon as it spawns. So, open the script and you can replace the existing code with the changes below.

```
using UnityEngine;
using UnityEngine.AI;

public class MoveToPosition :
MonoBehaviour {

    private Transform goal;
    private NavMeshAgent agent;

    void Start()
    {
        goal = GameObject
FindGameObjectWithTag("Player").
transform;
        agent =
GetComponent<NavMeshAgent>();
    }

    void Update()
    {
        agent.SetDestination(goal.
position);
    }
}
```

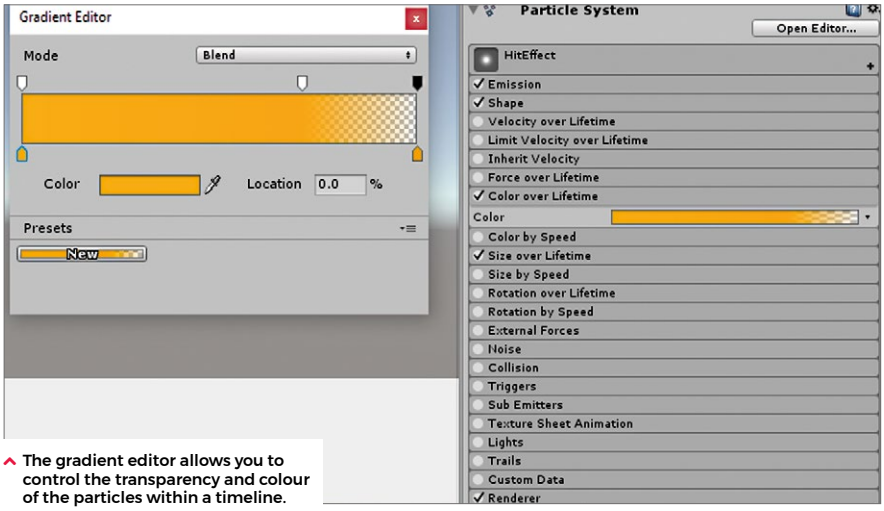
Save this, run the game again, and the error should be gone, and the AI will ➧

CATCHING UP

If you missed the first two parts of this guide, you can download free PDFs of Wireframe #3 and #5 from **wfmag.cc/3** and **wfmag.cc/5**. You can also download the complete unity project from **wfmag.cc/UnityFPS**



The Particle System contains a lot of useful modules that allow you to control your particle effect.



The gradient editor allows you to control the transparency and colour of the particles within a timeline.

work as before. You can then exit the play mode and we will look at improving the look of our projectiles.

ADDING WEAPON PARTICLE EFFECTS

Let's add some particle effects to the action – this will make the game look more engaging. First, some setup: we're going to add another camera. We do this so the particle hit effects render before the rest of the scene – otherwise, the particle effect will clip with whatever it hits. To achieve this, select the Player in the Hierarchy and expand it until you find the Main Camera, then right-click and select Camera. We'll get a warning if we have two or more Audio Listeners on cameras, so select Audio Listener in the Inspector and then right-click to remove the component.

Check the Depth on the camera component is 0 rather than the default of -1. This is telling Unity to render this camera before the main camera. We'll also set the drop-down under Clear Flags to Depth only. Next, we select the Layer drop-down to the top-right of the Inspector window and select Add Layer. We'll now see the Tags & Layers tab we used on the last tutorial. We need to expand the Layers option and then in an active empty layer type WeaponFX.

Now we need to select our Main Camera in the Hierarchy, and then in the Inspector select the Culling Mask drop-down, and then untick our new WeaponFX layer. You'll notice that the drop-down will now say Mixed – this is the correct behaviour. Now, select your new camera and select the Culling Mask drop-down and then Nothing from the options. Now reselect the Culling Mask and tick just the WeaponFX layer.

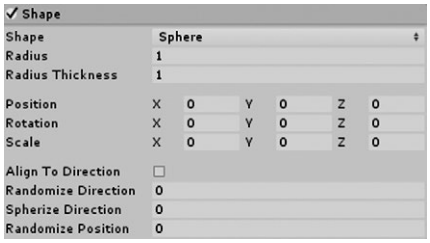
"We'll add a spark effect on top to make it more dramatic"

We're now ready to create a particle effect and make it a prefab. To make this easier, we'll create a new level – from the Taskbar, select File > New Scene. Save the previous scene if you're prompted. In the Hierarchy, right-click and select Effects > Particle System. In the Inspector for the Particle System, reset the position to 0,0,0 and then expand the Particle System component. We need to uncheck Looping and change the parameters of Duration and Start Lifetime to 0.4, the Start Speed to 0, and Max Particles to 1. Now expand the Shape module and then change the Shape drop-down to Sphere and set the Scale for the shape to 0,0,0.

If you select Restart from the Particle Effect window that appears in the Scene viewport, you'll see a single particle appear at a fixed position, then disappear. We'll now enable Size over Lifetime module and the Color over Lifetime module. Open the Color over Lifetime module and click the box to the right of the word Color. You'll see the Gradient editor. This has several sets of arrows that control the transparency or the colour of the particle over time.

Let's add a new arrow along the top by clicking in the same approximate area as the other down arrows. We select this arrow and drag it to be about 3/4 along the top. Now select the down-arrow to the top-right and you'll see an Alpha slider; change the value from 255 to 0. This should make a nice fade out when your particle is about to die off. Next, select the up-arrows that are along the bottom. This will let you set colours of your choice. I've selected the same orange colour for both the left and right arrow, but this is up to your own artistic choice. Finally, close the Gradient editor and try replaying your effect by restarting the playback.

We'll add an additional spark effect on top of this to make it more dramatic. With our Particle System still selected in the Hierarchy, right-click and select Effects > Particle System. Select the new Particle System and then in the Particle System



The shape module allows you to set the shape of the volume that the effect will be emitted from.

component, unselect Looping and change the Start Lifetime and Start Size to 0.2, and the Duration and Start Speed to 2. Select and open the Emission module and change Rate over Time to 0 and then under the Bursts parameter select the + to the bottom-right; the defaults are fine here.

Next, select the Shape module and change the Shape drop-down to Sphere. Select the Color over Lifetime module, and again, open the Gradient editor and set-up the Alpha and Colour settings to mirror the ones for the first particle. In the Renderer module, select the Render Mode drop-down and select Stretch Billboard and then change the Speed Scale to 0.2 and the Length Scale to 1.

You can then preview the effects together; this should be quite satisfying, but feel free to tweak the settings to your preference. For ease of identification, select the first Particle System we made and in the Inspector name it HitEffect. One last change is to change the Layer drop-down to WeaponFX and select Yes, change children from the prompt.

As an addition, we'll add a script to destroy the particle effect so it won't clutter our inventory. In the Inspector, select Add Component, select New Script, and name this DestroyEffect, then open the script and replace with the code below.

```
using UnityEngine;

public class DestroyEffect :
MonoBehaviour
{
    public float maxTime = 1;
    private float timer;

    // Update is called once per frame
    void Update()
    {
        timer += Time.deltaTime;
        if (timer > maxTime)
        {
            Destroy(gameObject);
        }
    }
}
```

Save the script and then return to the Unity editor. We'll now drag this object into our Project window to make a Prefab. Next, we load our original scene from the Project window – you don't need to save the current scene, as we have our Prefab.

We now need to make some updates to our existing scripts. This will allow us to spawn a particle effect on the exact point our bullet hits the collider and add a knockback force to the Zombie enemy. Let's first find the BulletHit script in the Project and double-click to open it. We then replace the existing script with the modified code below.

```
using UnityEngine;
using System.Collections;

public class BulletHit : MonoBehaviour {
    public GameObject particle;

    //When we touch the collider we
disable this object.
    void OnCollisionEnter(Collision
other)
    {
        //Find the contact point on the
object we collided with.
        ContactPoint contact = other.
contacts[0];
        //Set the exact position and
rotation we hit the collider at.
        Quaternion rot = Quaternion.
FromToRotation(Vector3.up, contact.
normal);
        Vector3 pos = contact.point;
        //Spawn our particle using the
above parameters.
        Instantiate(particle, pos, rot);
        gameObject.SetActive(false);
    }
}
```

THE JOY OF LAYERS

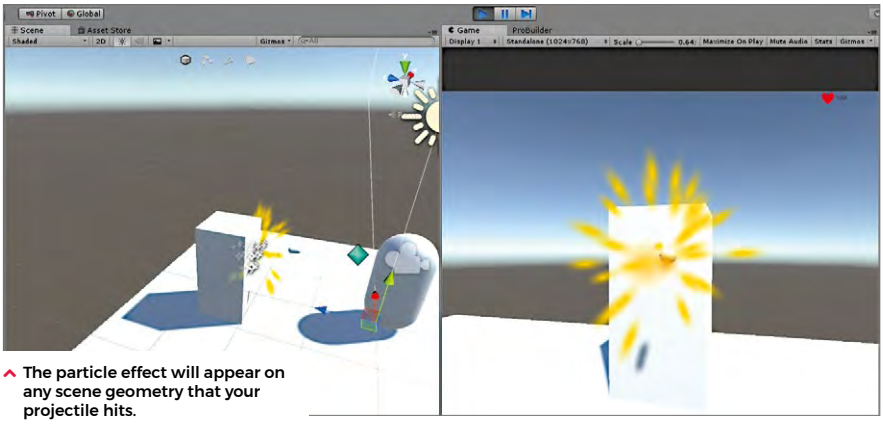
Layers are extremely useful and can be applied to more than camera rendering. We can use them to specify which lights would cast on an object, or which objects can interact with each other. You can find out more about layers from the Unity documentation: **wfmag.cc/sWAQFT**

Save this and then we want to open and replace our MoveToPosition script in a similar fashion.

```
using UnityEngine;
using UnityEngine.AI;

public class MoveToPosition :
MonoBehaviour
{
    public float knockbackTime = 1;
    public float kick = 1.8f;
    private Transform goal;
    private NavMeshAgent agent;
    private bool hit;
    private ContactPoint contact;
    private float timer;

    void Start()
    {
        goal = GameObject
FindGameObjectWithTag("Player").
transform;
        agent =
GetComponent<NavMeshAgent>();
        //Set timer to the same a
knockback in first instance.
        timer = knockbackTime;
    }

    void Update()
    {
        if (hit)
        {
            //Allow physics to be
applied.
            gameObject.
GetComponent<Rigidbody>().isKinematic =
false;
```

```
            //Stop our AI navigation.
            gameObject.
GetComponent<NavMeshAgent>().
isStopped=true;
            //Push back our enemy with an
impulse force set via the kick value.
            gameObject.
GetComponent<Rigidbody>().
AddForceAtPosition(Camera.main.transform.
forward * kick, contact.point, ForceMode.
Impulse);
            hit = false;
            timer = 0;
        }
        else
        {
            timer += Time.deltaTime;
            //After being knocked back,
restart movement after X seconds.
            if (knockbackTime < timer)
            {
                gameObject.
GetComponent<Rigidbody>().isKinematic =
true;
                gameObject.
GetComponent<NavMeshAgent>().isStopped =
false;
                agent.
SetDestination(goal.position);
            }
        }
    }

    void OnCollisionEnter(Collision
other)
    {
        //We compare the tag in the other
object to the tag name we set earlier.
        if (other.transform.
CompareTag("bullet"))
        {
            contact = other.contacts[0];
            hit = true;
        }
    }
}
```

Save this script and move back to Unity editor, select the Projectile in the Project window and expand it by clicking the right-arrow and select the Bullet mesh. In the Inspector for the mesh, you should see our BulletHit script. Select the slot labelled Particle, click the small circle next to it, and then select our HitEffect particle Prefab.



The particle effect will appear on any scene geometry that your projectile hits.

DEVELOPING OUR ROUNDS SYSTEM

We want to add a rounds system. For this, we'll make a new Game Object. Go to the Hierarchy, right-click in an empty space, and select Create Empty. In the Inspector, rename this object to GameManager. We then select Add Component and then New Script and call the script GameManager, then we will open this and add the code below.

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

public class Spawners
{
    public GameObject go;
    public bool active;
    public Spawners(GameObject newGo,
bool newBool)
    {
        go = newGo;
        active = newBool;
    }
}

public class GameManager : MonoBehaviour
{
    public GameObject panel;
    public delegate void RestartRounds();
    public static event RestartRounds
RoundComplete;

    private int health;
    private int roundsSurived;
    private int currentRound;
    private PlayerDamage playerDamage;
    private Text panelText;

    public List<Spawners> spawner = new
List<Spawners>();

    void Start () {
        Time.timeScale = 1;
        panel.SetActive(false);
        playerDamage = GameObject.
FindGameObjectWithTag("Player").
GetComponent<PlayerDamage>();
        panelText = panel.
GetComponentInChildren<Text>();
        foreach (GameObject
go in GameObject.
FindObjectsOfType(typeof(GameObject)))
        {
            if (go.name.
Contains("Spawner"))
            {
                spawner.Add(new
Spawners(go, true));
            }
        }
    }

    void Update () {
        int total = 0;
```

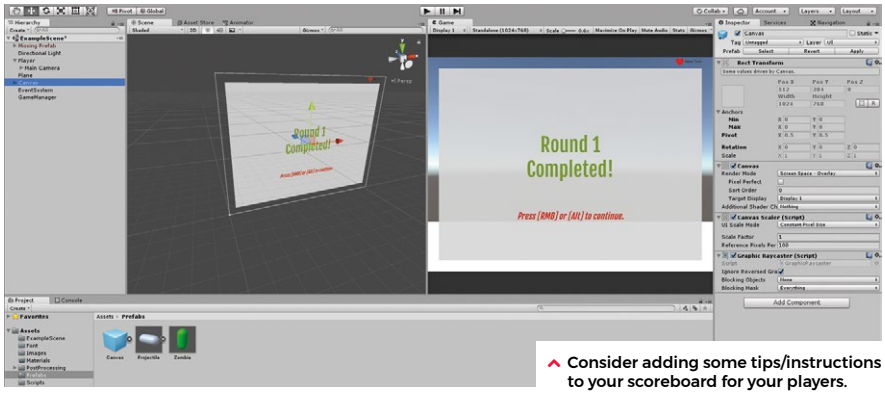
```
        health = playerDamage.health;
        if (health > 0)
        {
            for (int i = spawner.Count -
1; i >= 0; i--)
            {
                if (spawner[i].
go.GetComponent<Spawner>().spawnsDead)
                {
                    total++;
                }
            }

            if (total == spawner.Count &&
roundsSurived == currentRound)
            {
                roundsSurived++;
                panelText.text =
string.Format("Round {0} Completed!",
roundsSurived);
                panel.SetActive(true);
            }

            if (roundsSurived !=
currentRound && Input.GetButton("Fire2"))
            {
                currentRound =
roundsSurived;
                RoundComplete();
                panel.SetActive(false);
            }
        }
        else
        {
            if (Input.GetButton("Fire2"))
            {
                Scene current =
SceneManager.GetActiveScene();
                SceneManager.
LoadScene(current.name);
            }
            else
            {
                panel.SetActive(true);
                panelText.text =
string.Format("Survived {0} Rounds",
roundsSurived);
          Time.timeScale = 0;
  }  }  }  }
```

We now need to update the Spawner script. This is because we want to be able to trigger the spawners to restart when we've completed a round. The manager will look at when all the spawners are marked as depleted and restart spawning when a new round initialises. So we need to open our Spawner script and replace it with our updates below.

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

public class Enemy
{
    public GameObject go;
    public bool active;
    public Enemy (GameObject newGo, bool
newBool)
    {
        go = newGo;
        active = newBool;
    }
}

public class Spawner : MonoBehaviour
{
    public GameObject spawn;
    public int amount = 1;
    public float delaySpawn = 1;
    public bool spawnsDead;

    private int getAmount;
    private float timer;
    private int spawned;
    private int enemyDead;

    public List<Enemy> enemies = new
List<Enemy>();

    public void Start()
    {
        GameManager.RoundComplete +=
ResetRound;
        ResetRound();
        while (spawned < getAmount)
        {
            //Increment the amount
spawned count.
            spawned++;
            //Create the prefab as an
instance.
            GameObject instance =
Instantiate(spawn, transform);
            enemies.Add(new
Enemy(instance, false));
            //Removes the spawned object
from the spawner object.
            instance.transform.parent =
null;
      instance.SetActive(false);
        }
        ResetRound();
    }
    public void ResetRound()
    {
        spawnsDead = false;
        getAmount = amount;
        spawned = 0;
        timer = 0;
        enemyDead = 0;
    }

    void Update()
    {
        //Increase timer per frame.
        timer += Time.deltaTime;
        //Do the spawn if our timer is
larger than the delay spawn we set.
        if (delaySpawn < timer)
        {
            //And we haven't reached the
spawn amount.
            if (spawned < getAmount)
```

NOISES OFF

You can easily add sound effects to game objects by using the Audio Source component. By default, these will play the audio as soon as the object is active in the scene. You can import standard audio formats: MP3, WAV, and OGG. A great addition is to add a weapon firing audio effect to your bullet prefab; each shot will then play the effect on spawning.



Consider adding some tips/instructions to your scoreboard for your players.

```
            {
                //Reset our timer.
                timer = 0;
                //Set our bool to track the state of the enemy.
                enemies[spawned].active = true;
                //Set the enemy to be active.
                enemies[spawned].go.SetActive(true);
                //Get ready to set isKinematic.

StartCoroutine(SetKinematic(spawned));
                //Increment the amount spawned count.
                spawned++;
            }

            for (int i = enemies.Count - 1; i >= 0; i--)
            {
                //If another script disabled the object but we set them active above.
                if (enemies[i].go.activeSelf == false && enemies[i].active == true)
                {
                    //Reset the spawn position and set our tracking bool that they are not active.
                    enemies[i].go.transform.position = transform.position;
                    enemies[i].active = false;
                    enemyDead++;
                }
            }

            if (enemyDead == enemies.Count)
            {
                spawnsDead = true;
            }
        }
    }

    IEnumerator SetKinematic(int id)
    {
        //We set isKinematic at the start of the next frame to avoid confusion with other commands.
        yield return null;
        enemies[id].go.GetComponent<Rigidbody>().isKinematic = true;
    }

    private void OnDrawGizmos()
    {
        //Draw the wireframe mesh of what we intend to spawn in our editor.
        Gizmos.color = Color.red;
        if (spawn != null)
        {
            Gizmos.DrawWireMesh(spawn.GetComponent<MeshFilter>().sharedMesh, transform.position, spawn.transform.rotation, Vector3.one);
        }
    }
}
```

We need to make one very small change to an existing script. Open the EnemyDamage script and replace this with:

```
using UnityEngine;

public class EnemyDamage : MonoBehaviour
{
    private int hitNumber;

    private void OnEnable()
    {
        hitNumber = 0;
    }

    void OnCollisionEnter(Collision other)
    {
        if (other.transform.CompareTag("bullet"))
        {
            //If the comparison is true, we increase the hit number.
            hitNumber++;
        }
        if (hitNumber == 3)
        {
            gameObject.SetActive(false);
        }
    }
}
```

DISPLAYING OUR ROUNDS SCOREBOARD

We'll also set up a new Panel in our canvas, this will let us display a round scoreboard and a message when you run out of health. In the Hierarchy, right-click the Canvas and select UI > Panel. Select the new Panel and in the Inspector change the name to

AMAZING SCENES

You can create a game object with a script that is persistent in your scene (or set of scenes) and will not get disabled or deleted. Essentially, they take in input from other scripts, can control elements in other game objects, and help manage other elements of gameplay. I tend to refer to these game objects as managers, but you may see these called a slightly different name elsewhere.

Just using the UI elements we are familiar with, we can create our title screen.

ScorePanel. You might want to change some of the other parameters in the Image component, such as the opacity and colour of the panel image.

We now right-click in the Hierarchy with the new panel still highlighted and choose UI > Text. You can try changing the Alignment, font size and colour of the text component in the Inspector. Once you're happy with the look of the panel, select the GameManager object and drag the ScorePanel into the slot called Panel on the Inspector window.

We now have a complete experience in terms of our game loop; we'll know when we've completed a round, and when we get down to no-health we'll see the total rounds complete and have a chance to restart the game. To progress to the next round or to restart, we can use the binding for Fire2 which equates to the right-mouse button or left **ALT** on the keyboard.

IMPLEMENTING OUR GAME FRONT-END

It would be quite cool to add a front-end menu and be able to play through the game as a standalone executable like any other PC game. Let's start with the main menu first, and create a new scene by selecting File > New Scene and saving our current work.

In the new scene, go into the Hierarchy and right-click and select UI > Canvas. Next, right-click and select UI > Panel. As with the score panel, try changing the defaults to give this a look and feel that suits your game. Again, keep the panel selected, then right-click and select UI > Text.

I'd place this at the top of the canvas by selecting the Anchor Presets from the Rect Transform in the Inspector. Remember last time, where we held the **SHIFT** key to change the behaviour of the anchor? We'll do this again and then select the icon with the blue dot at the top-middle of the inner square. You'll then need to type in the value of 0 to Pos Y.

You should also change the values for the width and height of the Rect Transform to 300 by 200. Choose a font size of about 70 and set your alignments to the centre for the Text component. You can then come up with a title for your game; I chose Zombie Panic for mine.

We'll make two buttons; one to start a new game and the other to exit. First, select the Panel from the Hierarchy, then right-click and select UI > Button. We then repeat the process to add our second button. The two buttons will be overlaid, so with the second button still selected, we can use the move tool. Select the green arrow that's pointing up in the Scene window, then drag it downwards when highlighted.

"We should have a complete experience in terms of our game loop"

Next, we'll expand the Button object by selecting the right-arrow next to it in the Hierarchy. You should see another Text object attached. Select this and then in the Inspector change the text from Button to Exit. We expand the first button we created and repeat the process; however, we want to replace text with Start.

We need to make a script to start or exit the game. We can then link this to the buttons with an OnClick event. First, select the Canvas in the Hierarchy and then in the Inspector select Add Component. We then select New Script and name this MenuScript, then open it ready to replace it with the code below.

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class MenuScript : MonoBehaviour {

    public void StartGame()
    {
        SceneManager.LoadScene(1);
    }

    public void ExitGame()
    {
        Application.Quit();
    }
}
```

Save the script and return to Unity editor, then we can reselect the first Button we created. In the Inspector, look for the OnClick option. To the bottom-right is a +, so click this and a new entry will appear. Select the Canvas and drop this into the slot that displays None (Object). Now select the drop-down that says No Function and then choose MenuScript > StartGame. Repeat the step with our exit button, but this time choose MenuScript > ExitGame.

We now save this scene by selecting File > Save Scene as... from the taskbar, then go back to the taskbar and select Build Settings. In the new window, select the Add Open Scenes button. Close the window and now load our game scene. Next, select Build Settings and again Add Open Scenes. We're now in the position to make an executable. All we need to do is select Build and Run, select a suitable folder and file name for the game, and Save.

We'll now be able to try our menu system and play through the entire experience from start to finish. There are still many more improvements we can make, but you can see how we're building up the layers of a complete game experience.

Having the spawns separated and in their own 'caves' increases the challenge.

Directory

Representing data: a free online course

Want to know how to manipulate images, video and sound with computers? Then check out the Raspberry Pi Foundation's new online course

What's the thinking behind the course?

We wanted to present some of the more theoretical parts of the subject in a fun, practical and engaging way. Data is everywhere – it's such an important topic nowadays, with real world impact, so we've made sure it's useful for anyone who wants to learn about data through the lens of creative media.

Who is it for?

Anyone who wants to understand how computers convert data into digital media such as images, sound, text and video. This might be someone who is thinking about a career using digital technologies and wants to build a solid basis of understanding. It's also great for secondary school-aged students who are learning about this topic as part of their studies.

What will I get?

By the end of the course, you'll be able to:

- Describe how computers represent things in binary
- Produce your own emoji in bitmap and vector forms
- Understand lossy and lossless compression
- Understand common text encoding (ASCII and UTF-8)
- Investigate the physics of sound, and how sampling allows computers to represent sounds

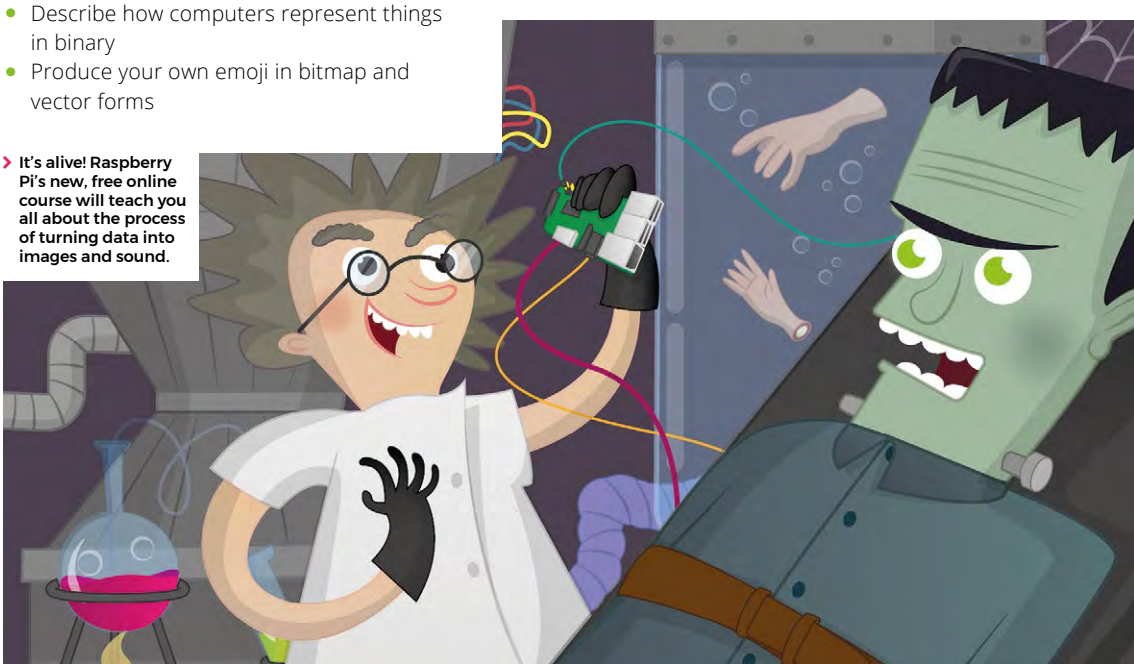
The take-away will be an understanding of how computers present to you all the media you view on your phone, screens etc, and you'll gain some new skills to manipulate and change what you see and hear through computers.

How much do you need to know before you start?

A basic understanding of Python. Taking our Programming 101 course on FutureLearn (also free) would be sufficient.

Where do I sign up?

Visit our course page: **rpf.io/datawf**



It's alive! Raspberry Pi's new, free online course will teach you all about the process of turning data into images and sound.

GET INVOLVED

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

Wireframe

LIFTING THE LID ON VIDEO GAMES

Download the app

Out now for smartphones & tablets

Save 45% with an annual subscription

£1.99 rolling subscription

or

£34.99 subscribe for a year

Available on the App Store

GET IT ON Google Play

DREAM ON

Inside the Dreamcast's homebrew scene

WRITTEN BY
ROBIN WILDE

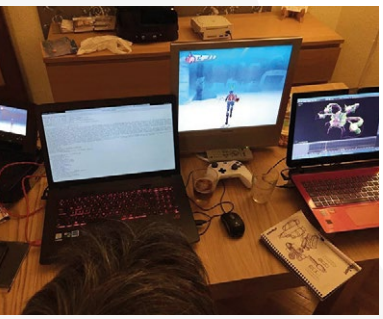
Despite its apparent death 17 years ago, the Sega Dreamcast still has a hardcore group of developers behind it

In 1998, the release of the Dreamcast gave Sega an opportunity to turn around its fortunes in the home console market. The firm's earlier system, the Saturn, though host to some beloved titles, was running a distant third in sales behind the Nintendo 64 and PlayStation. The Dreamcast, by contrast, saw a successful launch and quickly became the go-to system for arcade-quality ports of fighting games, among other groundbreaking titles like *Seaman* and *Crazy Taxi*.

Unfortunately for fans, it wasn't to last. The Dreamcast struggled to compete against the PlayStation 2, which launched in 2000, and at the end of March 2001, in the face of the imminent launch of the Nintendo GameCube and Microsoft's new Xbox, Dreamcast left the stage, and Sega abandoned the console market altogether.

None of this stopped a vibrant homebrew development scene springing up around the console in Sega's place, and even years later, the Dreamcast remains a thriving venue for indie developers. Roel van Mastbergen codes for Senile Team, the developers of *Intrepid Izzy*, a puzzle platformer coming soon to the PC, PS4 and Dreamcast. Of the port to Sega's ageing console, van Mastbergen tells us, "I started this project with only the PC in mind. I'm more used to developing for older hardware, though, so I tend to write code with low CPU and RAM requirements by force of habit. At some point I decided to see if I could get it running on the Dreamcast, and I was happy to find that it ran almost perfectly on the first try."

One of the pluses of the Dreamcast, van Mastbergen points out, is how easy it is to develop for. "There are free tools and sufficient documentation available, and you can run your



Retro Sumus test out updated builds using the Dreamcast Broadband Adapter, which saves on blank CDs.

JoshProd's games are a mix of new homebrew and ports of classic games – with the blessing of the original developers where they can get it.

own code on a standard Dreamcast without any hardware modifications or hacks."

Games burned to CD will play in most models of unmodified Dreamcast, usually with no extra software required. While this doesn't result in a huge market – the customer base for new Dreamcast games is difficult to measure but certainly small – it makes development for original hardware far more viable than the need for often expensive and difficult to install modchips needed for other systems.

Many of the games now being developed for the system are available as digital downloads, but the state of Dreamcast emulation lags behind that of its competitors, with no equivalent to the popular Dolphin and PCSX2 emulators for GameCube and PS2. All this makes boxed games on discs more viable than on other systems – and, in many cases, physical games can also become prized collectors' items.

KICKSTARTING DREAMS

By now, you might be asking yourself what the point is of developing for these old systems – especially when creating games for PC is a much easier and potentially more profitable route to take. When it comes to crowdfunding, though, catering to a niche but dedicated audience can pay dividends.

Belgian developer Alice Team, creators of *Alice Dreams Tournament*, asked for €8,000 in funding to complete its Dreamcast exclusive, which began development in 2006. It eventually raised €28,000 – more than treble its goal.

Intrepid Izzy didn't quite reach such dizzying heights, only just meeting its €35,000 target, but van Mastbergen is clear it wouldn't have been funded at all without the dedicated Dreamcast base. "The project has been under-funded since the beginning, which is slightly problematic," van Mastbergen tells us. "Even so, it is true that the Dreamcast community is responsible for the lion's share of the funding, which is a testament to how well-loved this system still is."

"One of the pluses of the Dreamcast is how easy it is to develop for"

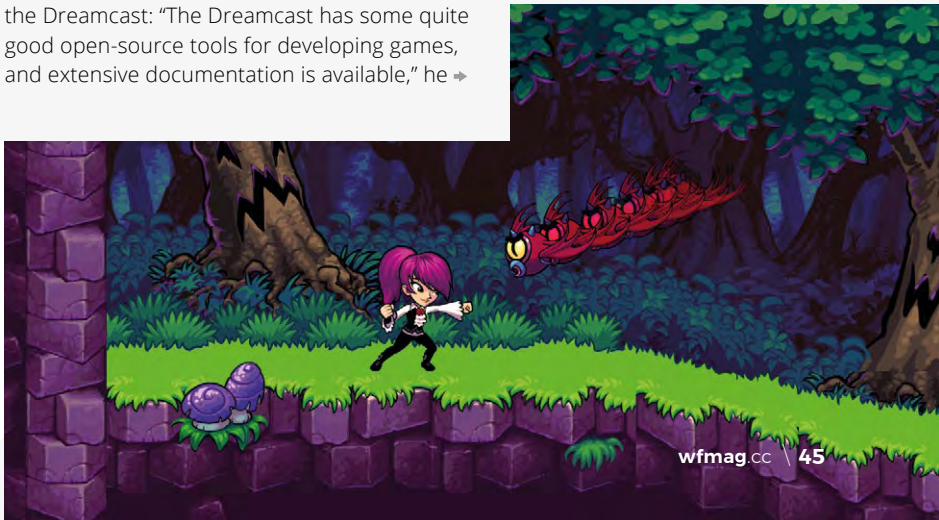
TRICKS AND TOOLS

Carlos Oliveros, whose team Retro Sumus are developing *Xenocider*, the console's first fully-3D game in 17 years, agrees that there's still life in the Dreamcast: "The Dreamcast has some quite good open-source tools for developing games, and extensive documentation is available," he

INTO THE FUTURE

With no more physical hardware being produced, is there a risk that eventually development will dry up? Roel van Mastbergen doesn't think so: "New hardware has been invented to replace worn-out parts, and the number of games released annually has actually increased in recent years. Also, considering that people are still making new games for hardware from the 1970s to this day, I wouldn't be at all surprised if new Dreamcast games will be made another 20 years from now."

Intrepid Izzy is developed with a custom code library that works across multiple systems; it's simple to downscale PC assets and export a Dreamcast binary.



› ***Xenocider*, in development for Dreamcast since 2015, combines old hardware with new tools, including 3D modelling work done in Blender.**

LIVING THE DREAM

While the USA and Europe saw their last official Dreamcast games in 2002, Sega continued to license new releases up until 2007 in Japan – long after the release of next-generation consoles. *Triggerheart Exelica* and *Karous* were the last two titles to receive the manufacturer's blessing, and were both vertically scrolling shoot-'em-ups ported from arcades. The shoot-'em-up genre has continued to be a staple of Dreamcast development well into the homebrew era.

Duranik's 2013 shoot-'em-up, *Sturmwind*, was the only homebrew Dreamcast game that year – since then, over 25 new titles have seen release.

Physical releases for Dreamcast can be a big event – *Sturmwind* launched with a special edition including Krakor, one of the game's bosses.

says. "Sure, there is nothing remotely as versatile as Unity, but you can create a little something here and there in a reasonable amount of time... and it's not nearly as frustrating as developing for the Saturn!"

Xenocider has been in development for four years, and is largely built on all-new technology. It's a 3D shooter inspired by Sega's arcade classic, *Space Harrier*, with, Oliveros says, "a heavy influence from Treasure's *Sin & Punishment*, which we realised later when we were creating the first assets for the game."

"The Dreamcast is special because it was the last arcade-inspired console in history"

The game's engine, Dreamer, was built from the ground up by Daniel Lancha, who goes by the handle Chui. He's a big name in the community, and was responsible for dreaming up a variety of tools, including ports of the popular SDL and OpenGL libraries, as well as a range of emulators.

"The Dreamcast is special because, in my opinion, it was the last arcade-based, or arcade-inspired, console in history. Besides, it is pretty accessible from a developer's perspective" he says, before admitting the console has no real advantages over more modern systems. "Just three words for you," Lancha adds. "16MB of RAM. Talk about insignificant. That really is the biggest challenge!"

Sturmwind made news on release in 2013 as one of the first of a new generation of boxed Dreamcast games. German studio Duranik developed the side-scrolling shooter, starting in 2006. Johannes Graf, who developed the game alongside his brother, says that on starting out, developing for the Dreamcast was an easy choice. "Modern consoles were not an option, as we developed the game only in our free time and there was no easy access to devkits for modern consoles back then," Graf tells us. "PC would have been an option but we are not so much PC gamers. Also, we always loved 'old' machines like the Dreamcast, Mega Drive, SNES, Lynx and Jaguar."

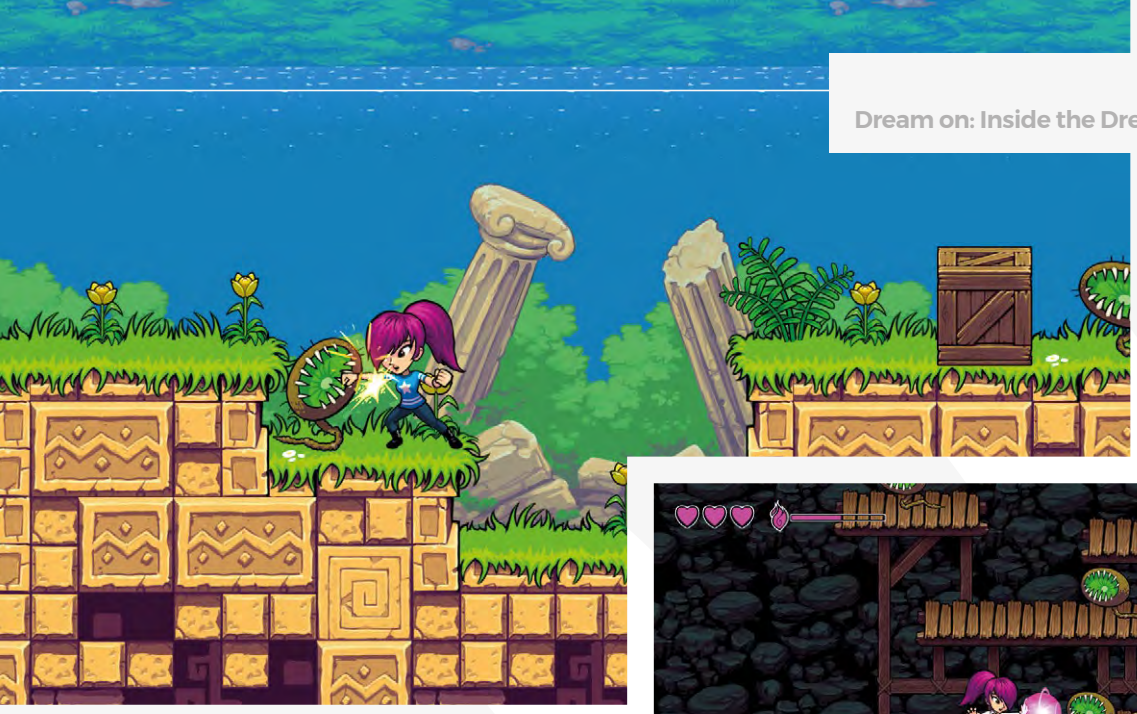
Developing for the Dreamcast also has another advantage, Graf notes: it's a much smaller market to enter than, say, Steam.

"Most of these new marketplaces are completely crowded; there are maybe a couple of thousand new apps released for the mobile platforms every day," says Graf. "How do you want to get at least a bit of visibility there? With the retro platforms, it's completely different – if there are 10 to 20 new releases a year, it's a lot."

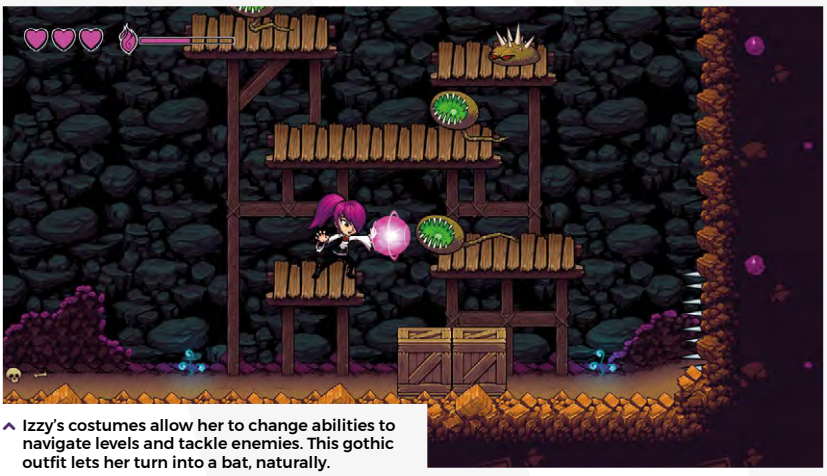
With crowdfunding able to bring these dreams to life, it might be surprising that publishers, of a kind, still operate in the Dreamcast homebrew community. But JoshProd, relatively new on the scene, does just that, producing high-quality boxed releases for new games. These have even included mainstream titles, with the last couple of years seeing the release of Delphine Software's nineties classic action-adventures, *Another World* and *Flashback*.

PUBLISHING AND DISTRIBUTION

Like Daniel Lancha, Philippe Nguyen of JoshProd says the main limitation is the Dreamcast's 16MB of RAM, which bottlenecks the scope for new ideas. "We had to cancel lots of good games like *Gigantic Army* or *Infinos Gaiden*," he says. Nguyen's justification for working around these



It runs at a lower resolution than on PC, but *Intrepid Izzy* still maintains a smooth 60fps on Dreamcast.



Izzy's costumes allow her to change abilities to navigate levels and tackle enemies. This gothic outfit lets her turn into a bat, naturally.

difficulties, however, echoes other developers we spoke to. "The reply is simple: passion and nostalgia," Nguyen tells us. "You don't make money with Dreamcast, you make a part of game history".

Boxed releases slot well into the existing infrastructure for console mods and homebrew games. Michael Mrozek runs DragonBox, a German-based online store selling everything from replacement Nintendo 64 analogue sticks to independent releases for a range of consoles, including the Game Boy Advance, Mega Drive and, naturally, the Dreamcast.

"Dreamcast games sell really well, to be honest – better than cartridge-based games," Mrozek says. "I can imagine that one of the reasons is the price: homebrew titles usually tend to be a lot simpler than commercial titles, with some exceptions like *Sturmwind*, *Wind and Water: Puzzle Battles* or the upcoming *Intrepid Izzy*, or *Tanglewood* for the Mega Drive. While high-quality games like *Sturmwind* sell well for a normal retail price, a lot of customers can't justify having to pay a high price for more simple titles. On the Dreamcast, that's not that big a problem, as producing the discs is very cheap, so you can sell simple games for a small and reasonable price."

While custom cartridges for the Mega Drive are available, they sell on DragonBox for €11 apiece, without counting the additional cost of the hardware to write the ROMs to the carts. With such a high markup for production, Dreamcast games typically cost less than half as much as Mega Drive titles.

Around 60 homebrew games have now been released for the Dreamcast in total – a significant percentage of the console's entire library – and, to date, there's no sign of the scene slowing down. While developers may be pitching for a limited audience, it's one that is still captivated by the possibility of a console that many feel didn't fulfil its potential in its own time. With more games being released year on year – and with ambitious 3D games like *Xenocider* on the way – it's clear that this small community is still dreaming big.

REGION FREE

The Dreamcast development scene is surprisingly geographically concentrated, with most current developers based in mainland Europe – particularly Germany, Spain and France. Definitive sales figures for the console are hard to find, but those available indicate sales of less than two million in Europe. "Dreamcast games sell about the same here as international, which is pretty good," DragonBox's Michael Mrozek says, but can't offer any explanation for the high number of developers in the region. "Because no one can do it as good as we can," he suggests.

The VMU has a homebrew scene of its own, with apps and ports of *Pac-Man* and *Tetris* available.

Hot and cold

Veteran writer Chris Avellone tells us about the beguiling world of Degrees of Separation

Info

GENRE
Puzzle platformer

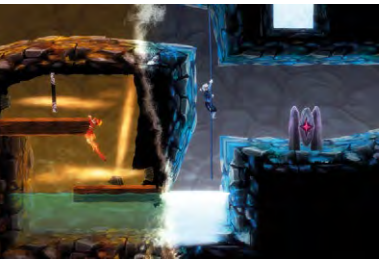
FORMAT
PC / PS4

DEVELOPER
Moondrop

PUBLISHER
Modus Games

RELEASE
Out now

Two worlds, one love, lots of platforms to overcome.



Degrees of Separation grabs you by the eyeballs. With its marionette-style animation, reminiscent of one *Earnest Evans* on the Mega Drive, it's immediately different. The visuals on this page pop; they're gorgeous. It's alluring – beguiling, even. Norwegian studio Moondrop has done a great job of making something that does the hard work of standing out in a crowded market, and that's a talent that hasn't come on suddenly.

See, Moondrop has been making steady progress over the past decade, releasing a couple of pretty-looking puzzle titles, each to a positive reception. *Kesper's Keep* was a browser-based puzzle-platformer released in 2011, involving smart use of light and colour in its brainteasers. *Amphora* saw the studio make its way into the big leagues of Steam, and the physics-based puzzler – the *gorgeous* physics-based puzzler – went down well with the limited audience it was exposed to.

This time, though, *Degrees of Separation* is gunning for the attention of far more – it wants a bigger audience, and it has already snagged a name to add credence (and quality prose) to the game's story: Chris Avellone. The freelance narrative designer/writer made his name on the likes of *Fallout*, *Planescape: Torment* and other classic story-heavy CRPGs. As such, the move to a less narratively-focused title was a surprising one – but it hasn't impacted Avellone's ability to put together a compelling tale.

"From the perspective of constructing a game narrative," he explains, "rather than adopting a linear structure, sometimes presenting story elements that reflect the player's own pace and approach to each challenge, and also reflecting what world/area/region the player selects to explore, is more in keeping with what makes a game a game – in these instances, you're not dictating the narrative, you're supporting the player's journey, and *Degrees* is very much that type of game."

Degrees of Separation tells the tale of two characters, Ember and Rime, whose love for each other compels the two of them to overcome an enforced separation by harnessing each soul's powers of heat and cold.

"I hadn't tried my hand at writing romance and relationships in games before – or at least not to this extent," Avellone explains. "I wanted to try it because I felt it would be a challenge – but as it turns out, *Degrees* provided something even more: it not only had a romantic component but a romantic component that was being presented in a mature fashion.

"It wasn't simply infatuation and elation and happily-ever-after and all the upsides of a relationship. Moondrop went further and

The game's animation is an almost marionette style, limbs articulating with abandon.

presented a relationship arc that included many of the challenges we all face in relationships – conflict, uncertainty, hesitation, doubt – and Moondrop's desire to include those elements in both the narrative and gameplay arcs felt very honest to me."

Of course, this being a puzzle-platformer, it is something more focused on the actual game people play rather than the tale being told, but that story pushes things along and frames it in an emotional context most might not have otherwise expected.

When you're solving puzzles because they're there, that's one thing; but when you solve them to drive forward a narrative you're honestly invested in, it becomes a whole other... well, *story*. Just take a look at *Braid*, for example.

"When there's a clear correlation between the narrative and the game mechanics, that's something I think makes any game storyline stronger versus being divorced from the mechanics," agrees Avellone. "In *Degrees*, the game mechanics mirror the emotional challenges Rime and Ember are facing, and both the game and the story are better for it."

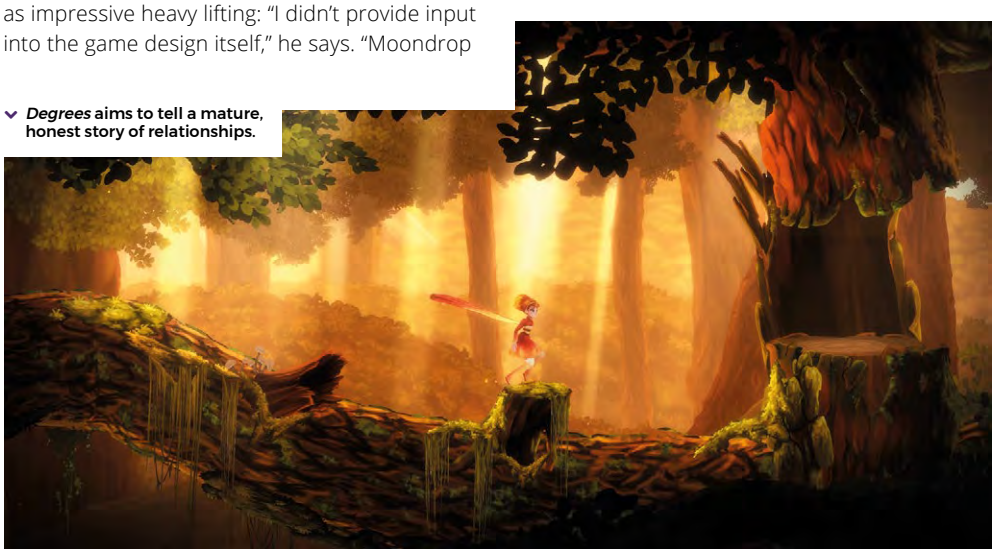
"Correlation between narrative and mechanics makes any story stronger"

The writer's involvement in the creative process amounted to – you may be surprised to hear – writing. However, that's not because Avellone is a man lacking in opinions or drive to help make a project better if he can – it's just because the developers did what Avellone sees as impressive heavy lifting: "I didn't provide input into the game design itself," he says. "Moondrop did the system and level design and I never felt the need to interject as the design of the title was clear from the outset.

"I don't think I ever had a single 'gameplay mechanic' comment on the title. And seeing the systems and levels laid out and being able to play through them allowed me to focus on a narrative structure that reinforced the game's mechanics and level design."

It may be a move that surprised those who have followed his career over the years, with an arguably darker, certainly more 'traditional' background to Avellone's titles. But he doesn't think the bright colours and bright fantasy of *Degrees* in any way holds it back: "*Degrees* is not a fairy tale," he says.

"It asks deeper questions about relationships and the spectrum of emotions relationships can go through, and not all of them are positive, which feels honest. And in being honest... that's a positive thing."



Degrees **aims to tell a mature, honest story of relationships.**

WORK/LIFE BALANCE

Since leaving Obsidian Entertainment, a company he co-founded, Avellone has found plenty of work on the freelance circuit. You might be annoyed with him popping up on the credits of so many games, but his decades-long career shows the man has serious stylistic chops with the written word. Away from the Obsidian fold, Avellone has seen credits on the likes of *Torment: Tides of Numenera*, *Prey*, *Into the Breach*, *Omensight*, the *System Shock* remake, and *Dying Light 2. Degrees of Separation* sits in some fine company, and should benefit from the substantial experience Avellone affords.

Developer Profile

Tokuro Fujiwara

The quietly 'scary' pioneer behind the survival horror genre, and too many Capcom classics to name

Konami's loss was, it soon turned out, Capcom's gain. Then a 21-year-old graduate, Tokuro Fujiwara joined Konami in 1982, where he designed two moderately successful arcade games: *Pooyan*, a curious shooting game with cartoon pigs, and *Roc'n Rope*, an action-platformer with an unusual grappling hook mechanic. After just one year, however, Fujiwara was poached from under Konami's noses by Capcom – a rival firm that had decided to branch out into the video game development market in 1983.

Although far from a household name, Fujiwara's 13-year run at Capcom would result in some of the company's most celebrated video games. In 1985, he directed the hit arcade game, *Commando* – an up-the-screen blaster that, with its military theme and relentless action, sparked a wave of imitators, most notably SNK's cheekily similar *Ikari Warriors*, released less than one year later. While *Commando* was being developed, Fujiwara was also working on an equally bold arcade title –

One of Fujiwara's last arcade games, ***Ghouls 'n Ghosts***, was something of a masterpiece.

a side-scrolling action platformer with an unusual Gothic horror theme. The result was *Ghosts 'n Goblins*, released just a few months after *Commando*, and another major success for Capcom.

Had Fujiwara only worked on arcade games for the rest of his career, he still would've deserved a place in the medium's history for his early titles alone: 1987's *Bionic Commando* was an ingenious upgrade on the concept he first dreamed up in *Roc'n Rope* for Konami; 1988 sequel, *Ghouls 'n Ghosts*, improved on its *Ghosts 'n Goblins* predecessor in just about every way.

PATIENT ZERO

It was when Fujiwara – reluctantly at first – moved over to Capcom's console game development team that his most innovative work emerged. He produced the NES sequel, *Mega Man 2,* in 1988, which became a 1.5 million-selling smash, and continued to work on the franchise until the mid-nineties. In 1989, meanwhile, Fujiwara directed a relatively low-key game called *Sweet Home*. Based on a Japanese horror film of the same name, it was a change of pace from the relentless action of Fujiwara's earlier games. A top-down RPG set in a haunted mansion, *Sweet Home* was tense, mysterious, and was – despite the limitations of its host platform, the Japanese NES – surprisingly tense.

Although never released in the West, due to Nintendo of America's aversion to gore and violence, *Sweet Home* was a patient zero for the survival horror genre. In retrospect, it's easy to see how *Sweet Home*'s ideas informed the later, far more successful *Resident Evil*, released in 1996: both share the same mansion setting, and both use things like limited inventories and resources to generate suspense. Fujiwara himself certainly had *Sweet Home* in mind when he began work on what would become *Resident Evil*; he later recalled in an interview that, with the improved hardware of Sony's PlayStation now at his disposal, he wanted to create an original horror game that contained all the things he was forced to leave out of *Sweet Home*.

"Fujiwara's work is varied, but the best of it is laced with horror"

Directed by Shinji Mikami, *Resident Evil* was nevertheless heavily influenced by Fujiwara, who served as co-producer. Fujiwara once claimed that Mikami hated horror movies, and was reluctant to embark on making a game with a horror theme; Fujiwara eventually coaxed him into taking the project on, reasoning that someone who experienced fear so acutely would be far more adept at making a scary game than a director who didn't.

Looking back over his career to date, Fujiwara's work is varied, but the best of it is often laced with horror. Lest we forget, the NES version of *Bionic Commando* concluded with its antagonist's head exploding in a crimson shower – a foreshadowing, perhaps, not only of the splashes of claret we'd see in *Sweet Home* and *Resident Evil*, but also the blackly comical fountains of blood on display in PlatinumGames' *MadWorld*, the Wii brawler he co-designed in 2009.

Of Fujiwara's appetite for the violent, the exciting, the gory and the outlandish, maybe Shinji Mikami put it best. "He is a scary master for me," he said in an oft-quoted interview from 2001. "He's not big or macho, and he doesn't raise his voice either, but he is really scary."

Occult horror

After 13 years at Capcom, Fujiwara grew weary with the firm's focus on making sequels to its biggest franchises, which at the time included *Mega Man* and *Street Fighter*. He therefore left in late 1995, shortly before the release of *Resident Evil*, to pursue his own projects. Of these, two were horror-themed: the chilly *Extermination*, released for the PS2 in 2001, and which trod similar ground to *Resident Evil*. *Hungry Ghosts*, created for the same platform and launched two years later, offered a mesmerising first-person descent into the afterlife. It was a unique experience that, frustratingly, never saw a release outside Japan.

Fujiwara's horror games, *Sweet Home* and *Hungry Ghosts*, were never released outside Japan. Both deserve to be more widely played.

Designer and producer Tokuro Fujiwara. Photo courtesy of Alex Aniel.

Blood and thunder

10 of Fujiwara's finest

Fujiwara's cracking coin-ops and creepy console classics

01 Ghosts 'n Goblins

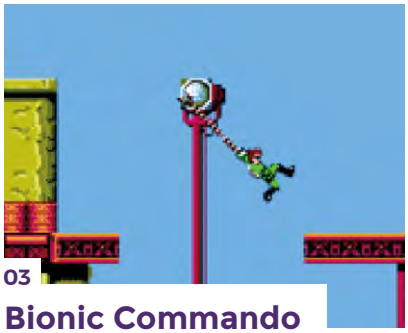
Arcade / various – 1985

Fujiwara's horror comes laced with comedy: the player's knight famously loses his armour when touched by zombies and other monsters, revealing a pair of boxer shorts beneath. The humour helped disguise how brutally hard *Ghosts 'n Goblins* is; Fujiwara reportedly adjusted the game several times before release to make it even harder to beat.

02 Commando

Arcade / various – 1985

Action movies starring lone soldiers were all the rage in the mid-eighties, and Fujiwara's relentless up-the-screen blaster *Commando* (whether it meant to or not) tapped into the jingoistic zeitgeist. Even on the battlefield, there are comic flourishes: note how enemies wave their arms and legs when they're hit. Copycat shooters from rival developers soon followed.



03 Bionic Commando

Arcade / NES – 1987

The arcade version was OK, but it was the NES port that really brought *Bionic Commando* to the masses. An innovative platformer, with the player swinging and climbing with a grappling arm rather than jumping, the game's design ideas are still being widely borrowed today. Capcom's own 2009 reboot was, sadly, a flawed sales disappointment.

04 Ghouls 'n Ghosts

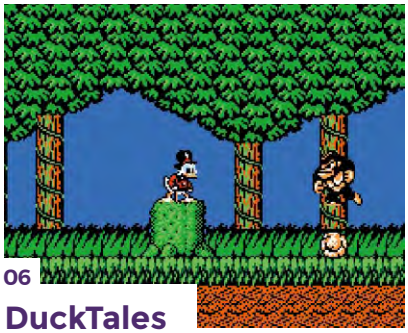
Arcade / various – 1988

A refinement of the original game's premise, *Ghouls 'n Ghosts* is nevertheless a superior follow-up. Still fiercely hard, and still bursting with energy and ideas – the pixel art and animation is easily among the best of its era. Fujiwara would later return to the series he created with *Ultimate Ghosts 'n Goblins* on the PSP in 2006 – to date, his last game as director.

05 Mega Man 2

NES – 1988

With the rising popularity of consoles, Fujiwara was convinced to move to that side of Capcom's business in the late eighties. His first task in his new role was to oversee the production of *Mega Man 2* – a double-or-quits sequel that even he wasn't sure about making at first. All the unpaid overtime soon proved worthwhile: unlike its predecessor, this would go on to be a smash hit.



06

DuckTales

NES – 1989

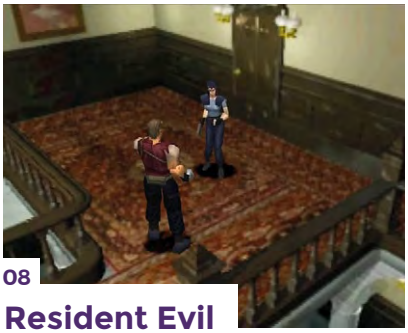
The fluffy Disney licence aside, *DuckTales* offers more pixel-precise platforming action from the *Mega Man* team, and a neat mechanic that allows Uncle Scrooge to bounce on his cane like a pogo stick. Still regarded as a classic, *DuckTales* would receive a 21st century remake, and an homage, of sorts, in Yacht Club's similarly bouncy platformer, *Shovel Knight*.

07

Sweet Home

NES (Japan-only) **– 1989**

Had this horror RPG been released in the West, its brilliance may have been appreciated earlier. Instead, it took several years, and a rock-solid fan translation, before *Sweet Home*'s contribution to the survival horror genre was finally understood. More than a dry-run for *Resident Evil*, this is a classic game in its own right.



08

Resident Evil

PlayStation / PC / Sega Saturn – 1996

With its overbearing sense of scarcity and claustrophobia, *Resident Evil*'s impact is still being felt over 20 years later. Taking the basic premise from *Sweet Home* – characters trapped in a horror-filled mansion – it took full advantage of the then-new PlayStation's hardware with its atmospheric lighting and cinematic camera angles.

09

Hungry Ghosts

PlayStation 2 (Japan-only) **– 2003**

Offering nothing less than a journey across the hereafter, *Hungry Ghosts* offered a genuinely unsettling first-person horror experience. Obscure even at the time of its launch, this is one Fujiwara joint that sorely deserves a belated worldwide release – we could even imagine *Hungry Ghosts* working as a skin-crawling virtual reality title.

10

MadWorld

Wii – 2009

A ferocious brawler that allowed players to bash in heads and chainsaw off limbs with a waggle of the Wii remote, *MadWorld* sparked controversy, but failed to make a splash in terms of sales. Like many late Fujiwara games, though, *MadWorld* soon found a cult audience. Spiritual sequel *Anarchy Reigns*, made without Fujiwara's input, was far less satisfying.

Subscribe today

SAVE 49%

wfmag.cc/**subscribe**

Try 12 weeks **for only £12**

Subscriber benefits

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

Introductory offer

wfmag.cc/12weeks

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

Subscribe for 12 months

wfmag.cc/12months

£40 (UK) **£75** (USA)

£65 (EU) **£75** (RoW)

Offers and prices are subject to change at any time

Digital subscriptions from

£1.99

Available on the App Store

GET IT ON Google Play

wfmag.cc/subscribe

Review

Resident Evil 2

When there's no more room in Hell, the dead will be remade in 4K

Info

GENRE
Survival Horror

FORMAT
PS4 (tested) / XBO / PC

DEVELOPER
Capcom

PUBLISHER
Capcom

PRICE
£44.99

RELEASE
Out now

REVIEWED BY
Ian Dransfield

Inventory management, staring at map screens, and paying careful attention to how many bullets you have left (it's usually 'not enough'): *Resident Evil 2*'s 2019 remake can be boiled down to elements like these without being too facetious. On one hand, it's a fine nod back to the 1998 original, of which this is a complete remake – not a remaster, not a mere update. On the other hand, it's the kind of thing – when you describe it as a space-managing, map-staring, bullet-fretting exercise in *paying attention* – that makes *Resident Evil 2* sound... shall we say, 'uninspiring'. And yet, *Resident Evil 2* is the best since the series began.

That's all down to the simple fact, as simple as something like this could ever be, that Capcom has entirely captured the spirit of the original. The oppressive atmosphere of working your way through a claustrophobic police station, knowing full well the city around you has gone to hell, is present and accounted for. The giddy terror of being jump-scared out of your boots by a licker dropping in from above is just as pulse-wrecking as it ever was. And, in so very many ways, the modern *Resident Evil 2* ups the ante – not least of which through its very smart, very *frightening* use of one particular enemy: the Tyrant.

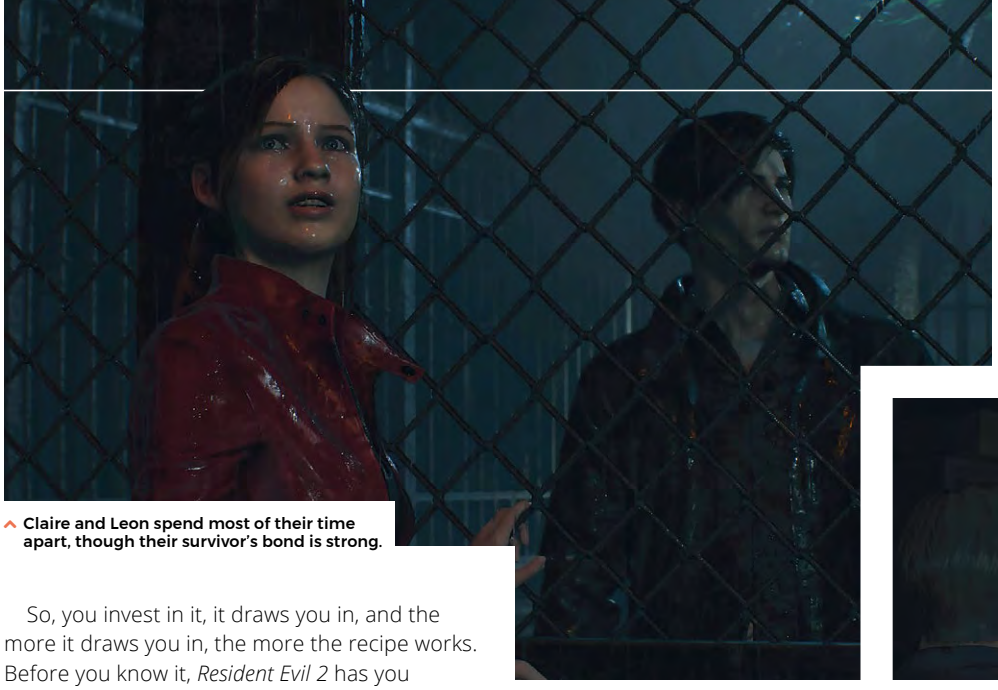
You might fight off most zombies in relative comfort, choosing to inflict a bit of damage before skirting around the edges of a room and moving on to save ammo. You might sneak past those lickers, or throw a few bullet-bones for the undead pooches littering the place. But, at certain points in the game, there's an ever present stomp-stomp-stomp, reminding you and your blood pressure that there's a more serious threat on its way. And he is always on his way. The Tyrant, to borrow a fitting phrase, will not stop, ever, until you are dead. That's quite a lot of pressure to have to deal with.

But there is respite to be had, and that's what pushes you on: the desire to be safe. To get this over with and escape. *Resident Evil 2* manages to make you feel as desperate and flustered as its protagonists, Claire Redfield and Leon Kennedy – no mean feat.

HIGHLIGHT

It was always going to be the Tyrant, wasn't it? Mr. X himself is nothing less than panic-inducing, relentlessly pursuing the player and leaving very few areas actually safe – even some you would think have to be. Come your third or fourth playthrough, this stomping monstrosity still manages to wedge your heart at the very top of your throat.

You'll soon learn that while it's in short supply, fire is a good friend.



Claire and Leon spend most of their time apart, though their survivor's bond is strong.

So, you invest in it, it draws you in, and the more it draws you in, the more the recipe works. Before you know it, *Resident Evil 2* has you perched on the edge of your seat, lights down low, headphones firmly in place, begging it not to scare you again. You just need to make it to the next area; to find that key to move on and get one step closer to safety. *Stomp-stomp-stomp*.

Problems rear their heads, and no playthrough is free from feelings of slight irritation every now and then. The backtracking might be accurate to *Resident Evil*'s past – and that of survival horror in general – but it really does feel like a waste of time.

It's also striking just how linear your movement is, with extraordinarily few occasions in which you're able to even mantle, and even fewer where you can climb things (at least without a ladder).

Enemies can sometimes feel unfair with regards to how often, how quickly and how accurately they can grab/attack... puzzles are often illogical (though obvious) at best... Look, we're clutching at straws – nothing is ruinous, and most of it does just hark back to what *Resident Evil 2* was to begin with.

It's not a free running simulator, it isn't a series of logical physics-based puzzles, and the threat of the zombie (et al) hordes is *meant* to scare and frustrate. So it's a flimsy argument to say this brings the experience down in any real way, as my irritation at any of these factors quickly drifted away.

One aspect I'm not so forgiving of is the way in which the first and second scenarios intersect and impact one another – or, I should say, how they don't.

In the original, the A and B scenario for each character saw plenty of moments replayed from a different angle, fundamentally different routes taken and – importantly – decisions made in A directly impacting the situation in B.

While there is bisection between the first and second stories in this remake, generally it's just another playthrough of the main game with the odd different route thrown in. Puzzles are largely the same, items required are the same, and beyond a few notes from the other character, there's little to show you're working your way through the story parallel to them. And the lack of impact between the two runs is jarring, coming from a *Resi 2* vet: having to weigh up whether or not to grab the machine gun in your playthrough of A – knowing it would mean it wouldn't be there for the character in the more difficult B – was a genuinely tough choice.

Again though, it's not ruinous. It just couldn't be. *Resident Evil 2* manages to expertly straddle the line between homage and entirely new game, and in both respects it's a huge success. It feels old, but plays new; it carries the imagery of the original, but looks spectacular by 2019's standards. *Resident Evil 2* makes a fantastic case for the importance of revisiting – and honouring – gaming's past, while at the same time not overshadowing or forgetting about the original. It is, in short, bloody brilliant.

"*Resident Evil 2* is the best since the series began"

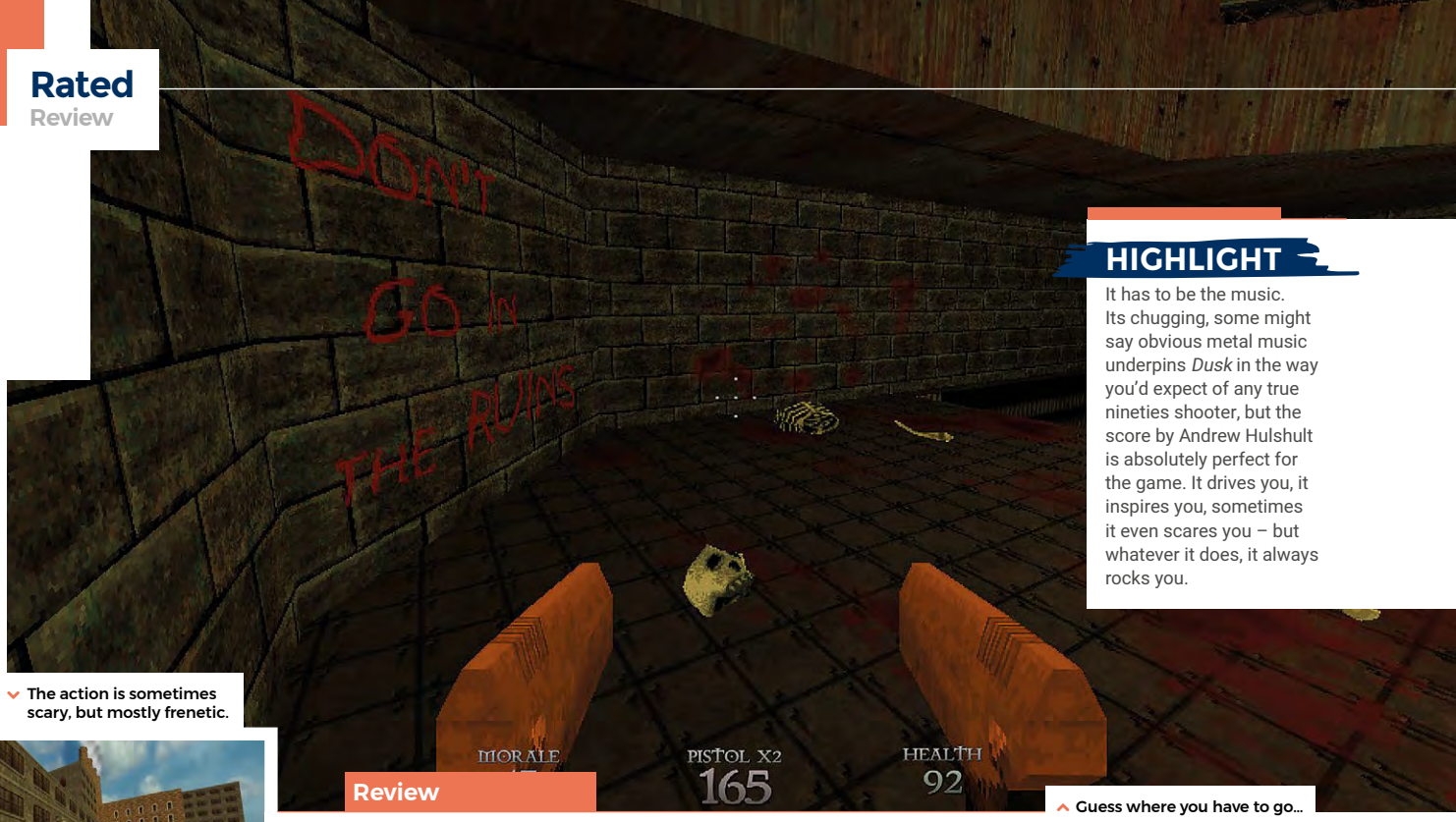
It's not really the sort of welcome you expect in a new job...

VERDICT

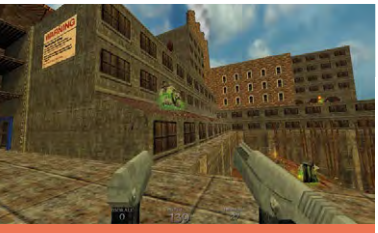
As a remake, it's glorious. As its own thing, it's brilliant. A stellar effort by Capcom.

86%

Famous scenes from the original are given a new lease of life, often with spectacular results.



The action is sometimes scary, but mostly frenetic.



Guess where you have to go...

HIGHLIGHT

It has to be the music. Its chugging, some might say obvious metal music underpins *Dusk* in the way you'd expect of any true nineties shooter, but the score by Andrew Hulshult is absolutely perfect for the game. It drives you, it inspires you, sometimes it even scares you – but whatever it does, it always rocks you.

DUSK

Yes, these are actual PC game screenshots

Info

GENRE
First-person shooter

FORMAT
PC (tested)

DEVELOPER
David Szymanski

PUBLISHER
New Blood Interactive

PRICE
£15.00

RELEASE
Out now

REVIEWED BY
Ian Dransfield

VERDICT

One of the best shooters of the nineties arrived in 2018, for some reason.

81%

It's so much more than a shallow emulation of nostalgia. *Dusk* – somehow, some way – manages to be one of the best shooters the nineties ever produced, but it was made 20 years later using Unity. My brain is frazzled from the pace, the unforgiving nature, the sheer thrill of it all, and I am so completely happy that this FPS anachronism exists.

The obvious influence is *Quake* – I mean, just look at it – but there are nods to plenty of other shooters from the nineties and beyond: *Doom*, *Duke Nukem 3D*, *Redneck Rampage*, *S.T.A.L.K.E.R.*, *Blood*... you get the point. From a pure aesthetic standpoint, *Dusk* nails it – it looks and sounds the part, with a chugging, intense metal soundtrack backing up all the quick-paced, blasty action. From the perspective of the tributes it offers to those that came before, *Dusk* is respectful, somewhat fawning, and – again – absolutely nails it.

But all of that really becomes window dressing once you get stuck into the game proper: *Dusk* is brilliant fun. You're not looking at recharging health and waist-high cover, carrying two weapons (no more than that) and travelling from A to B in order to trigger a cutscene. No, in *Dusk* you're – very quickly – scooting around smartly designed levels, hunting a variety of coloured keys, blasting an ever-increasing array of enemies, and just trying to get to the exit. It doesn't sound like much on paper, admittedly, but there's the *feel* of the thing that just nudges it, pushes it, and bundles it into the realms of something genuinely great.

Dusk doesn't faff about; it respects you as a player, it respects your time, it respects your ability. It's hard at times, sure, but it's never unfair. If you're good enough, you can do it. And it's open enough to let you tackle things your own way, allowing levels to be selected with any loadout you so require. This isn't about Szymanski forcing you to endure his vision: it's about you using the tools the developer provides to have a good time. And sometimes be a bit scared, especially when you're playing with headphones on.

"*Dusk* doesn't faff about; it respects you as a player, it respects your time, it respects your ability"

Dusk isn't a tribute, an ode, a nineties-style facade over a paper-thin set of mechanics. It is a legitimately good first-person shooter: consistent in the fun it offers; the challenge it throws at the player; and the creativity running through to its very core. I came in expecting a neat little tribute to the games I loved 20 years ago (and still do love, admittedly), but I came away with a modern favourite. *Dusk* is fantastic.

To keep this potentially infinite concept simple, puzzles are limited to just interacting with three layers: yourself, your larger self and your smaller self.

HIGHLIGHT

As demonstrated by *Moss* last year, a story told with a diorama is a comfortable fit for VR, except *A Fisherman's Tale* essentially has you in a diorama, within a diorama. You also get an old-fashioned storytelling vibe as the narrator doubles as the voices of the other characters.

Review

A Fisherman's Tale

A mind-bending story inside your VR headset

When it comes to games, the player is the puppet master of every protagonist you assume the role of. It's an idea Innerspace's immersive VR adventure, *A Fisherman's Tale*, doesn't shy away from, since it casts you in the role of a puppet.

Specifically, you're a puppet resembling your fisherman creator, who frankly isn't much of a fisherman, as the narrator tells you about how he spends his days in his cabin building models, including an exact replica of the house where you both live. The house also has a cabin model with a fisherman puppet inside, who in turn has his own model cabin. Or wait, is your master also just a puppet himself who might have another creator? It really is a mind-bending Matryoshka doll of a paradox.

To keep things relatively grounded, the game prioritises your puppet's perspective, so you're the sentient one whose actions will be mimicked by the other versions of you. Hunch over your fisherman's cabin diorama in the centre of your room, then look up, and you'll see your giant master also peering from above.

This is where the fun comes in, as you're required to solve a series of puzzles that cleverly play with scale, such as how dropping a fun-sized object into your model cabin results in a larger version of the same object dropped into your cabin by your giant self, or vice versa.

But for all *A Fisherman's Tale*'s storybook wonder, the experience is often hampered by the controls. In PSVR, locomotion is done by teleporting, but instead of blinking to set points, you aim your Move controller at a trajectory, which is fine when you want to move from one side of the room to the other, but awkward when you're just trying to reposition yourself about a foot to the left. Another mechanic extending your puppet arms should solve the problem of interacting with items just out of reach – an issue I've had with *SUPERHOT VR* – but because it's limited to one setting, it often results in your arms then becoming too far out, so I'm having to stretch my arm backwards so my hand can come back within an item's grasp. There's also some clumsy glitches where things get stuck through a wall, in which case you'll have to wait around for an item to respawn in its original location.

Control issues aside, *A Fisherman's Tale* is also yet another very short VR-based narrative. That's not to say this land-dwelling fisherman's quest needs to be dragged out, but when its mechanics are this inventive, I feel that it could have easily been used to sustain more inventive puzzles around a game not limited to its short narrative. To paraphrase a certain character, I could do with more – and bigger – fish.

This giant fish is one of the few bizarre characters you'll meet.

Info

GENRE
Adventure / Puzzle

FORMAT
PSVR (tested) / Vive / Rift / Windows MR

DEVELOPER
Innerspace VR

PUBLISHER
Vertigo Games

PRICE
£11.99 (PSVR)
£13.99 (Steam)

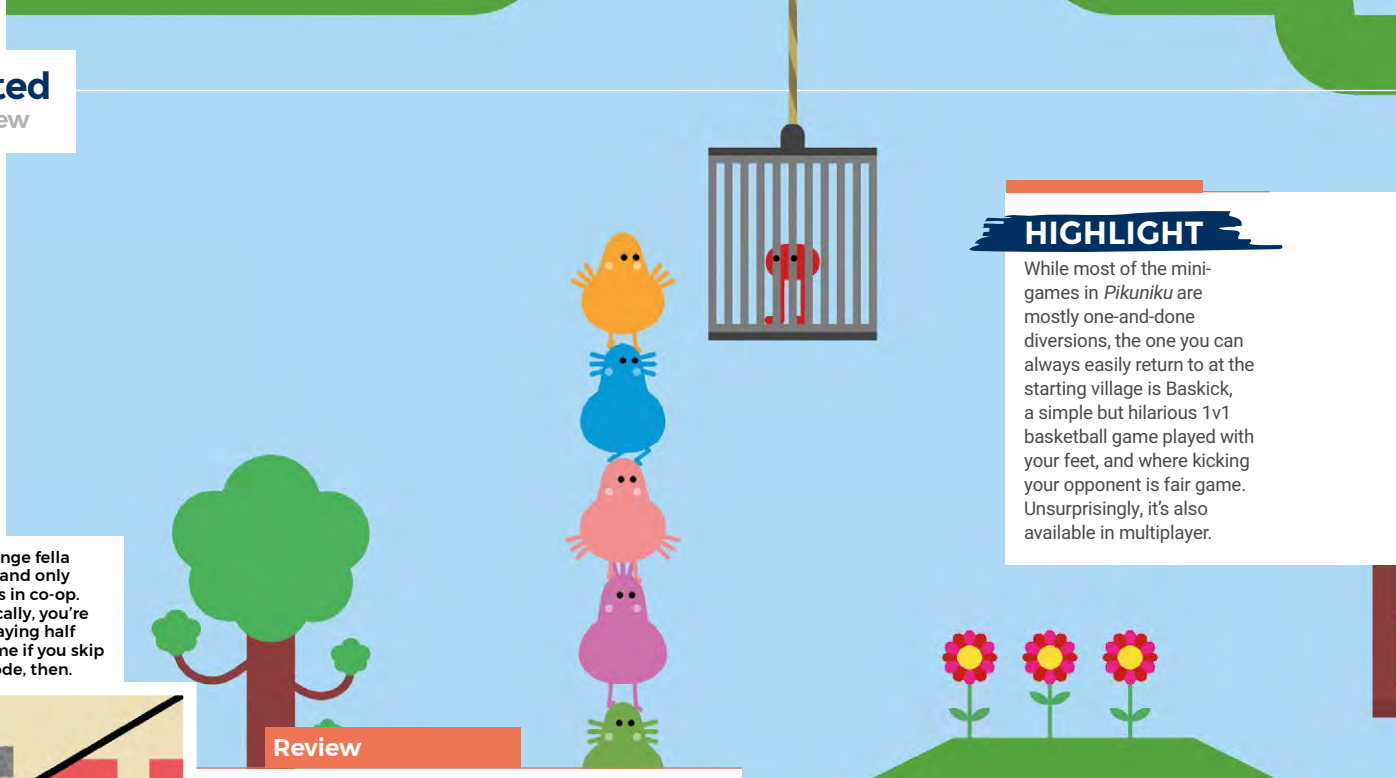
RELEASE
Out now

REVIEWED BY
Alan Wen

VERDICT

An original but short-lived VR narrative, while awkward controls interfere with its clever ideas.

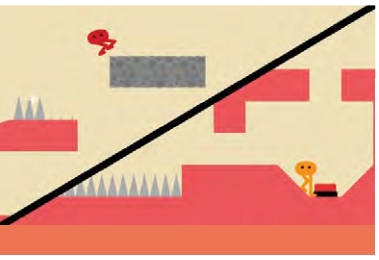
65%



The orange fella is Niku and only appears in co-op. Technically, you're only playing half the game if you skip this mode, then.

HIGHLIGHT

While most of the mini-games in *Pikuniku* are mostly one-and-done diversions, the one you can always easily return to at the starting village is Baskick, a simple but hilarious 1v1 basketball game played with your feet, and where kicking your opponent is fair game. Unsurprisingly, it's also available in multiplayer.



Review

Pikuniku

Joyfully sticking, rolling and kicking it to the man

Short at only nine levels long, but still a delight.

Info

GENRE
Puzzle adventure

FORMAT
Switch (tested) / PC

DEVELOPER
Sectordub

PUBLISHER
Devolver Digital

PRICE
£11.69 (Switch)
£10.29 (Steam)

RELEASE
Out now

Maybe I'm just getting old, but it seems almost impossible to look at a seemingly innocent-looking children's cartoon aesthetic and not wonder if there's a more sinister or profound message going on underneath. So when *Pikuniku*, a seemingly innocent-looking game with characters that look as simplistically drawn as the Mr. Men, is marketed by publisher Devolver Digital as a dystopia, it's hard to resist an eye-roll.

Fortunately, those fears are unfounded. Sure, there's a greedy pink-faced capitalist calling himself Mr. Sunshine handing out 'free money' to the villagers of an island in exchange for plundering its natural resources, but with this out of the way, Sectordub doesn't pause for any deep state commentary. It just wants you to have a jolly good time.

It's a sensation you get from the moment you wake up from a cave as Piku, cuteness incarnate with minimalist effort. For just a red oval with a pair of lanky legs and a pair of dots for eyes, he's surprisingly expressive as you bounce him around this childlike 2D world, as he's capable of rolling around and through tight spaces, or using his long legs to kick switches, boxes or other hapless denizens, whether that's to solve a puzzle or just for, um, kicks. Better still, his leg also doubles as a lasso to swing on hooks or race up zip wires. Jumping may be a little too floaty, but for a game that's more adventure than platformer, traversal feels wonderful.

From being first mistaken as a 'ghastly beast' to joining an underground resistance against Mr. Sunshine, it's a brief and breezy journey, carried by a whimsical score from Calum Bowen. While nothing is exactly designed to challenge you, it's more about giving you a variety of things to play around with, from a rhythm-action dance-off to boss battles against big robots.

It's also worth splitting the Joy-Con with a friend for a standalone local co-op mode. While it only consists of nine levels, it's almost like playing a platforming version of *Snipperclips*, with some fun variation, bringing up split-screen when required, missions that see you both tethered together, or an impromptu competitive race.

It's charmingly compact then, perfectly content with dropping a bit of joy into your life without overstaying its welcome. But it's also the stuff that veers off-path that makes *Pikuniku* a delight. From coins to collect, weird junk to spend it on, even physical trophies that are wrapped up like presents – a fine consolation for the Switch's lack of an in-built system – has a dystopia ever felt this good?

There's a dystopian theme, but it's subtle.

REVIEWED BY
Alan Wen

VERDICT

No chin-stroking here, *Pikuniku* is simply a joyous and varied little adventure.

79%

Review

Pang Adventures

An arcade classic pops up again, but has its bubble burst?

This latest iteration of cult arcade franchise *Pang* arrives on the Switch, having been out for a couple of years on other platforms. Vintage gamers reading this will probably already be familiar with *Pang* due to its excellent home conversions, appearing on everything from the humble ZX Spectrum to the Amiga to Amstrad's doomed GX4000.

For the newcomers among you, *Pang* involves popping a screen full of bouncing bubbles with your harpoon gun, progressively splitting each one into two smaller bubbles until each screen is cleared. This time, though, there are aliens involved, as well as a selection of new power-ups, including shotguns and ninja stars. Basically, though, *Pang Adventures*' mechanics – essentially *Asteroids* crossed with *Space Invaders* – remain the same as they were back in the late 1980s.

"It's addictive, particularly in two-player co-op"

Pang's appeal lies not just in quick-fire arcade reactions, but in how each stage presents its own challenge. Obstacles like platforms and ladders alter the trajectory of a bouncing sphere, and the way power-ups alter your abilities mean stages require a modicum of thought – as well as accurate blasting – to complete.

Pastagames have done a fantastic job in capturing the look and feel of the original arcade game, even if *Pang Adventures* doesn't quite capture the charm and personality of, say, *Monster Boy* – another recent game based on a classic franchise.

Some major difficulty spikes are another minus, especially during boss battles. There's no obvious indicator of where or when to shoot your harpoon into the alien slimeball that stands between you and the next level, which feels like a bit of an oversight at best.

Despite this, *Pang* will still provide hours of retro fun – it's a curiously addictive game, particularly in its two-player co-op mode, which has always been the series' strongest point. There's something about the chaos of dodging bubbles and vying for vital power-ups that never quite gets old.

Alongside the main Tour mode, there are also two other modes to unlock after finishing the main game, which really put your bubble-blasting skills to the test.

Pastagames haven't revolutionised *Pang* as we know it here, but all the same, you could do a lot worse than popping this on your Switch.

Pang's standard harpoons are joined by a range of new weapons.

HIGHLIGHT

Pang Adventures has a neat scoring system baked in. There's points gained in the level and for time left, of course, but also bonus points for not missing a single shot. Each stage has its own target score, which is something to aim for. It's a good way of keeping you playing, trying to top your own high score and attain perfection.

Bosses look OK, but aren't the game's high point.

Info

GENRE
Arcade puzzle

FORMAT
Switch (tested) / PC / PS4 / Xbox One / iOS / Android

DEVELOPER
Pastagames

PUBLISHER
DotEmu

PRICE
£7.99

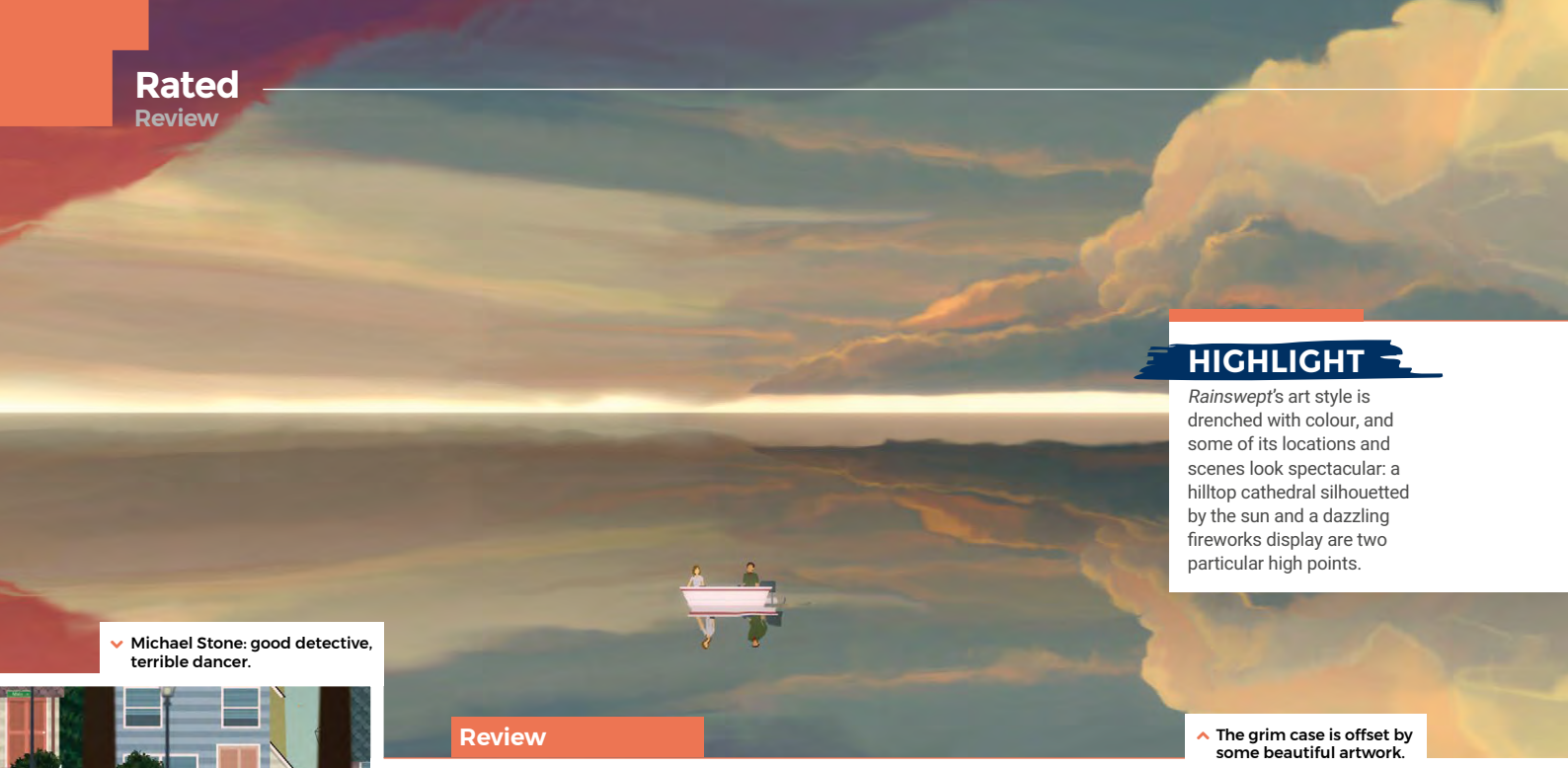
RELEASE
Out now

REVIEWED BY
Jake Laverde

VERDICT

Pastagame's revision feels most at home on the Switch. Perfect for killing time during your lunch hour, commute or bout of insomnia.

69%



HIGHLIGHT

Rainswept's art style is drenched with colour, and some of its locations and scenes look spectacular: a hilltop cathedral silhouetted by the sun and a dazzling fireworks display are two particular high points.

Michael Stone: good detective, terrible dancer.

Review

The grim case is offset by some beautiful artwork.

Rainswept

Play detective in a short yet compelling point-and-click mystery

A silhouetted figure, the sound of a single gunshot, and a crime-thriller soundtrack set the noirish tone for Armaan Sandhu's point-and-click adventure, *Rainswept*. It's a two-dimensional murder-mystery, with jaded detectives, philosophical discussions about tea, and, with its small-town location fringed by pine trees, obvious parallels to the classic television series, *Twin Peaks*.

Although there are locations to explore and puzzles to solve, *Rainswept* is driven primarily by its story, which is a design choice that has its advantages and pitfalls: it doesn't stray far from the template set by earlier point-and-click adventures, and the player's path through its narrative can feel somewhat narrow. Stick with it, though, and you'll find a gripping case with plenty of twists, all wrapped up in a captivating, minimalist art style.

The town's cast of residents are surprisingly well-developed, and the story's lighter character beats help leaven the bleak mood that surrounds your murder investigation – Detective Michael Stone is a likeable protagonist, and his awkward dance moves following a few drinks at a bar are a particular highlight, while his forthright manner and strong sense of obligation make him an endearing presence.

Other characters display some fascinatingly erratic behaviour, leading to plenty of discoveries as the case continues to unravel. There's the terse-yet-empathetic Officer Blunt; Grandpa, eccentric owner of the local bakery and prone to angry outbursts; and shy, young, aspiring photographer, Johnny – as well as plenty more. It's a well-rounded and engaging cast a lot of the time, and serves *Rainswept* well.

Some characters have their own secrets and hidden agendas, too, capable of diverting Stone down alternate paths. Away from the main cast (and side characters), there are the overheard conversations referencing video games and well-worn tropes. It paints a fuller picture of this world, and it just works.

Sandhu's game design, art direction, and story are perfectly complemented by micAmic's varied soundtrack; reminiscent of a late eighties' detective show, it ranges from slow piano melodies to smooth jazz playing in the local cafe.

The interface is engaging, too: flicking through the detective's notepad, which contains character observations, sketches, and other notes, I couldn't help thinking of that old family favourite, Cluedo. There are still a few bugs, however, such as action wheels not responding to mouse clicks and autosaving issues, but these are likely to be rectified before release.

Rainswept can be completed in a relatively brisk six hours or so, and despite the branching dialogue options, there's still the feeling that you're being funnelled to its conclusion. When the story's as rich as this one, however – it runs the gamut from poignant to gently amusing to downright unsettling – *Rainswept* is a mystery well worth solving.

Info

GENRE
Point-and-click adventure

FORMAT
PC (tested) / Mac / Linux

DEVELOPER
Frostwood Interactive

PUBLISHER
Frostwood Interactive

PRICE
£9.29

RELEASE
Out now

REVIEWED BY
George Sturgeon

VERDICT

With an intriguing story and superb visuals, *Rainswept* is an absorbing little detective thriller.

72%

HIGHLIGHT

Limping back to the nearest safe port from an expedition gone wrong, escaping from impossible monsters while your crew turns on you, your train spews acrid smoke, and your resources run low. Life in the Skies is a constant balance of risk and reward, making every run unique and exciting.

This is the British Empire, but not as you know it.

Every choice you make can have massive consequences.

Review

Sunless Skies

Failbetter's latest fantasy adventure title really shines

Nobody does game worlds like Failbetter. *Fallen London* and *Sunless Sea* are masterclasses in making settings that are dense, detailed and unique without ever needlessly falling into cliché. The atmospheric, gorgeously realised Gothic Victorian adventures never let up, cementing the studio as a leader in game narrative. It should be no surprise that *Sunless Skies* is yet another example of sublime writing in a fascinating setting, but it's also a streamlined, more approachable title, too.

Following London's escape from deep underground, Queen Victoria has defeated the sun and achieved immortality, spreading her empire throughout the stars. Picking your own backstory, ranging from academic to street urchin, you are the conductor of a flying train in the vast expanse of the skies. Through careful resource management, you're tasked with exploring this bizarre world of clockwork suns and time fissures, and the more human issues of war and labour rights.

It's in these many stories where *Sunless Skies* truly shines. It's more enjoyable when the combat difficulty and resource usage are turned all the way down to their easiest settings to let you get out into the Skies unimpeded. One moment you're on an expedition into the abandoned tourist centre in the middle of an artificial star, the next you're picking up a Devil desperate for escape from the nearest port. It's like Jules Verne distilled into a game – a celebration of adventure in all its weird, wonderful and horrific forms.

Storytelling aside, the details of commanding your train are slightly simplified from *Sunless Sea*. The world is split into multiple regions now, rather than the singular, massive Unterzee of *Sunless Sea*, making managing your resources easier. Permadeath still plays a big role, too, but it's more forgiving as some money, items and the map you painstakingly developed over your previous runs are passed down from one ill-fated captain to the next. This helps the game break free from its starting area a lot quicker, while also making a new run feel like a continuation of a storied history and less like just rolling a new game entirely.

"Yet another example of sublime writing in a fascinating setting"

Sunless Skies makes one stumble in its combat. It's not diabolical, but the 2D dogfighting is simple and shallow compared to everything else on offer. Circle around an enemy, let off a few shots, occasionally dodge a volley of fire, rinse and repeat. Gunfights are plodding and, compared to the constant sights and sounds of the Skies, terribly mundane.

Regardless, you'd be missing out if you skip *Sunless Skies*. Its endless enthusiasm in its own world is infectious, constantly pushing you forward to explore the unknown reaches of said world. This is how writing in games should be.

Info

GENRE
Adventure / RPG

FORMAT
Windows (tested)

DEVELOPER
Failbetter Games

PUBLISHER
Failbetter Games

PRICE
£18.99

RELEASE
Out now

REVIEWED BY
Joe Parlock

VERDICT

Sunless Skies **will pull you into its world and never let you go. This modern take on the old *Choose Your Own Adventure*-style books is a lesson in narrative perfection.**

83%

HIGHLIGHT

The character designs are abstract and just plain bizarre. A main character whose head has been swapped with a glowing diamond, enemies that look like overgrown viruses, and walls bursting into flocks of birds, all drenched in neon, are some of the oddities that give *Octahedron* its distinctive visual style.

Deciphering each level's rules and quirks is vital.

Review

Octahedron: Transfixed Edition

A mesmerising platform-puzzler hits the Switch

Info

GENRE
Platformer

FORMAT
Switch (tested) / PC / PS4 / XBO

DEVELOPER
Demimonde

PUBLISHER
Square Enix

PRICE
£9.99

RELEASE
Out now

REVIEWED BY
Joe Parlock

VERDICT

Octahedron **feels like a relic from a simpler time, but in a good way. The platforming's slick, the levels are tough, and the soundtrack sounds like a 2003 rave.**

70%

More games are being released right now than ever before. While that is, on the whole, a good thing, it also means plenty of perfectly fine and fun indie titles are getting buried because they prioritise nailing the fundamentals instead of trying to be the next big genre-defining *Minecraft* or *Kerbal Space Program*. That's why a game like *Octahedron*, which feels like something straight from the indie boom of the early 2010s, is so refreshing. It's not going to redefine the world, but it harkens back to a much simpler time when the likes of *VVVVV* and *Braid* were all the rage.

Octahedron's stages are highly vertical, the obstacles are unforgiving, and being sent all the way back to the beginning of a stage happens alarmingly frequently. To help you, you're given the ability to generate a limited number of temporary platforms that move along with you. This may sound like it makes a game all about jumping to and from platforms a tad easy, but the ingenious level design and use of multiple types of platform, each with their own properties to master, helps turn it into a devilishly challenging experience.

Literally everything glows like it's straight out of *Tron*.

Nothing in *Octahedron* should be taken at face value. Obstacles react to your presence and actions, requiring you to use your platforms as switches, lifts, shields and more. Vital ledges may fade out of existence as you move on the X axis, or disappear should you spawn your own platform in a different place. It's this requirement to observe and learn the rules of each level as you go that gives depth to what would otherwise be a very by-the-numbers platformer.

Of course, with a core conceit this uncomplicated, the presentation also needs to be stellar, and *Octahedron* has style to spare. The thumping trance soundtrack and neon visuals not only make the game an immensely appealing game to experience, it helps lull you into a state of flow that's halfway between *Tetris*-style relaxation and bullet hell hyperfocus. Every setback and failure blurs into a rush of colours and sound, with only the transitions between stages letting you come back up to the surface for a moment.

Octahedron isn't a game-changer, but neither is it trying to be. It isn't bringing anything new to the platforming genre, nor is it offering a deep and emotive story, and that's completely okay. What it *is* offering is fiendishly tricky level design, a thorough understanding of platforming as a genre, lush presentation, and a real 'indie spirit' that's rarely been seen since the early days of Humble Bundle. Calling it 'essential' would be overkill, but there's a charm to *Octahedron* that is absolutely worth checking out.

Saluting the Wii's weirdest game

Farewell to the Wii Shop, and its most bizarre title

Now that the Wii Shop's gone, it's easy to forget about how slow and awkward it could be, and start thinking nostalgically about its soothing loading sounds (which we heard a lot of) and perky muzak. Nintendo's decision to shut the Wii's digital store down on 31 January 2019 was an inevitable one, given the console's 13-year vintage. The question the closure raises, though, is what happens to those online-only games that were solely available for the system. The titles we've selected in the panel on the right are just three examples of Wii exclusives that, to date, have never reappeared anywhere but the less salubrious corners of the internet; unless you already have these games installed on your Wii, there's currently no legal way to download and play them.

This is doubly true for an obscure little gem called *Pole's Big Adventure*, a platformer released exclusively on the Japanese Wii Shop in February 2009. Emerging at a time when developers were beginning to look again at eighties-era game design (Capcom, for example, had recently put out the retro-themed *Mega Man 9*), *Pole's Big Adventure* reads as a demented parody of Sega and Nintendo's 8-bit output.

With its pixel graphics, chiptune soundtrack and bare-bones running and jumping action, *Pole's Big Adventure* could've come from some low-ranking Japanese developer from the days of the NES and Master System; what's immediately apparent, though, is that it's designed to baffle and startle the player at every turn. The staples of *Super Mario Bros.* and *Alex Kidd* are subverted from the first screen: collect a mushroom, and you'll grow to a vast size and immediately die. Try to head down a *Mario*-like drainpipe, and you'll resurface covered in muck. Collectible fruit will rot or emerge on the screen half-eaten. Meanwhile, a narrator frantically comments on the unfolding chaos. If Terry Gilliam made games, they might look something like this.

"If Terry Gilliam made games, they might look something like this"

What's most curious about *Pole's Big Adventure* is its origin: this wasn't the product of a lone bedroom coder's jaundiced mind, but *Phantasy Star Universe* producer, Takao Miyoshi; it was even published by Sega, who allowed its distinctive eighties speech sample to play as the game begins. Despite its pedigree, it's still easy to see why nobody could be bothered to localise *Pole's Big Adventure*: the humour's too odd, too risqué, and too text-heavy to be commercially viable. And now, with the global closure of the Wii Shop, this largely unknown title runs the risk of slipping even further into obscurity.

Nintendo and other companies may jealously guard against the online distribution of their lucrative back catalogues, but currently, console games like *Pole's Big Adventure* tend to fall through the cracks. They're old, so they cease to be available, and they're too niche for publishers to revive. Barring a legitimate and concerted effort to preserve these online-only games, ageing curios like *Pole's Big Adventure* run the risk of vanishing for good.

Wireframe Recommends

Alien Crush Returns

WIIWARE

Naxat Soft made some terrific pinball games in the 1980s and 1990s – among them *Alien Crush*, *Devil's Crush* and *Jaki Crush* – and this belated entry in the series made decent use of the Wii's motion controllers.

Star Soldier R

WIIWARE

Essentially a score attack mode, vertical blaster *Star Soldier R* feels like a proving ground for a more ambitious sequel – regrettably, Hudson Soft never got around to making one.

Gradius ReBirth

WIIWARE

More a remix than a rebirth, really, but this late addition in Konami's 2D shooter franchise captures the earlier games' arcade glory. To date, *ReBirth* marks the last proper entry in the series.

Red Faction: Guerrilla

Give a person a fish, and they'll eat for the day.
Give them a hammer, and they'll destroy half of Mars

VOLITION, INC / 2009, 2018 / PS3, XBOX 360, PC, PS4, XBOX ONE

There's always a first time, and even though *Red Faction: Guerrilla* was the third game in that particular series, and plenty of games tackled destruction in one way or another before Volition's Mars shooters did, I'm going to stick with *Guerrilla* as the first time I *truly* loved destruction.

The game itself was decent enough – your standard open-world fare, only this open world was a terraformed Mars in the grip of a proletariat uprising, presented with some none-too-subtle Soviet-inspired iconography. It would have been liked, not loved, but for one aspect: the destruction.

OK, so the actual surface of Mars was impervious, as were some smaller elements, but generally speaking, in *Red Faction: Guerrilla*, if it was made by people, it could be destroyed by people. Naturally these destructive tendencies were harnessed for what you'd expect – mission objectives and side mission/challenges.

But what was harder to predict was how much players would take to just smashing stuff up... oh, who am I kidding? Smashing stuff up is why we *live*. *Guerrilla* was an exercise in catharsis for dummies; the sort of brain-off action that satisfies a primal part of the brain. Hit a wall, it crumbles. Crumble enough wall, the building comes down.

From there, you'd just get creative. Hitting weak structural points to see what the minimum amount of damage you could inflict on a structure before its implosion, maybe.

Or – a personal favourite – riddling a vehicle with stick-on remote charges and driving it towards an enemy base, before diving out at the last second and turning the barrelling Mars-car into a mobile bomb. The speed at which I took to actually behaving like a guerrilla fighter still bewilders me to this day.

Another thing *Guerrilla*'s destruction did was acknowledge hammers in their rightful place. That being: the height of good comedy. Players completing the game with all sorts of jiggery-pokery would be rewarded with a series of special thwacking tools, culminating in the legendary Ostrich Hammer. Which was exactly what you'd think.

It was this joyful approach to smashing stuff that helped to make *Guerrilla*'s destructive tendencies so enamouring. It was silly, it encouraged you to muck about, and it was always fun to see how quickly you could take out a major base of operations with a pack of satchel charges.

"A series of special thwacking tools, culminating in the legendary Ostrich Hammer"

But behind all of the fun was a seriously smart bit of design, in the shape of the Geo-Mod 2 engine (built off the back of John Slagel's original Geo-Mod). It's testament to those physics-and-destruction routines that even on *Guerrilla*'s remaster ('Re-Mars-tered', as it was officially called) in 2018, the entire system is still genuinely impressive.

In the decade since *Guerrilla*'s first release (and even longer since the original *Red Faction*'s launch) there have been few attempts at introducing such levels of delightful destruction. *Crackdown 3*'s multiplayer does apparently up the ante with just how much can be destroyed – I wouldn't know, I haven't played it at the time of writing – and that can only be a good thing. A 'blast', you could say. But the first time (that wasn't actually the first time) is always the most special, and *Red Faction: Guerrilla*'s destruction really was smashing.

Next Issue

ON SALE 28 FEB

Don't miss out! Subscribe PG 54

Mineko's Night Market

Discover cats and crafting in Meowza Games' adventure-sim

Also

- Inside Sumo Digital, the UK's biggest indie studio
- How music made the C64's Wizball a classic
- Playing cyberpunk detective in Tales of the Neon Sea
- Squeezing a score attack game onto the BBC Micro

Editorial

Editor
Ryan Lambie
Email ryan.lambie@raspberrypi.org

Features Editor
Ian Dransfield
Email ian.dransfield@raspberrypi.org

Sub Editors
David Higgs and Vel Ilic

Design

criticalmedia.co.uk

Head of Design
Lee Allen

Designer
Harriet Knight

Contributors

Lottie Bevan, Rik Cross, Konstantinos Dimopoulos, Megan Fox, Stuart Fraser, Francesca Harrell, Jake Laverde, Joe Parlock, George Sturgeon, Howard Scott Warshaw, Robin Wilde, Alan Wen, Jack Yarwood

Publishing

Publishing Director
Russell Barnes
Email russell@raspberrypi.org
Tel +44 (0)7904 766523

Director of Communications
Liz Upton

CEO
Eben Upton

Distribution

Seymour Distribution Ltd
2 East Poultry Ave, London EC1A 9PT
Tel +44 (0)207 429 4000

Subscriptions

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

To subscribe
wfmag.cc/subscribe

To get help:
rpipresshelp@raspberrypi.org

recycle

BY NC SA

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.

Raspberry Pi PRESS

ISSN: 2631-6722 (print), 2631-6730 (online).

wfmag.cc