

£3
ALL FORMATS

Wireframe

LIFTING THE LID ON VIDEO GAMES

ETERNAL CODERS

The veterans making games on mobile

REVLING RODENTS

Why rats continue to infest our video games

POINT BLANK

Code your own shooting gallery

EL HIJO

STEALTH MEETS SPAGHETTI WESTERN

Issue 20 £3
wfmag.cc





UPGRADE
TO LEGENDARY



AG273QCX
2560x1440

144 HZ

1 MS

Curved

QHD

FreeSync2

amazon

BOX

ebuyer.com

AOC
GAMING

f @aocgaming

t @aoc_gaming

i @aocgaming

aocgaming.com

F2P game makers have a responsibility to their players

Free-to-play has brought gaming to billions of new players across the globe: from the middle-aged Middle Americans discovering hidden object puzzles to the hundreds of millions of mobile gamers playing MOBAs in China. People who wouldn't (or couldn't) buy games now have an unfathomable choice, spanning military shooters to interior decorating sims.

F2P is solely responsible for this dramatic expansion of our art across the borders of age, race, culture, and continents. The joy of gaming is now shared by nearly the entire globe. Yet many inside and outside the games industry see F2P as a blight, where the model's success is down to its ability to trick unsuspecting players into spending.

Having spent nearly a decade making F2P games, I know that this isn't an accurate picture of the average spending player. The titles most prolific under the model are successful because they become meaningful to players' lives, with spending healthily repeated over months and often years. I've personally spent an average of £1500 per year on *Magic: The Gathering* over many years, well within the limits of my income, and have no regrets. The game has brought me a lot of joy and introduced me to some of my closest friends.

But it would be foolish to suggest that there aren't players who make purchases in F2P titles that they live to regret, especially if they do so with impaired reasoning, as is often the case with children and vulnerable adults. I appeared on the BBC News back in 2013 talking about children accidentally spending money in games, and now, six years later, the BBC's still running stories on the problem.

Some may say that the responsibility to protect lies with platform holders, parents, and carers. But this is a shirk. Game makers have a responsibility to protect their players, even if it's from their own actions, and I believe there is a simple way to do so: refunds.

It's possible to see the F2P industry adopt a simple policy, one which I know many studios already secretly



WILL LUTON

Will Luton is a veteran game designer and product manager who runs Department of Play, the games industry's first management consultancy. He is the author of *Free-to-Play: Make Money From Games You Give Away*, and has worked with Sega, Rovio, and Jagex. He is also an avid retro games and pinball player.

offer: any player can close their account and receive a full refund for any spending within the last 60 days. This time limit gives parents, guardians, carers, and remorseful spenders a full billing cycle to spot errant spending and another cycle to request their funds back. It's also a clear signal to our players and the gaming community that F2P is not predicated on one-time tricks but on building healthy, sustainable relationships with players.

How such a refund policy would be publicised and adopted is a little more tricky. But faced with mounting calls for censorship in the 1950s, the comic industry formed Comics Magazine Association of America to self-regulate. Today, the UK's UKIE and the United States' Entertainment Software Rating Board, along with Google, Apple, Valve, and other platform holders, could build a similar association.

The 1940s saw Fiorello LaGuardia, then mayor, take to the streets of New York with a sledgehammer to destroy pinball tables. Much of early gaming history is now lost, and the evolution of pinball was stunted until New York lifted the ban in the 1970s. Without self-regulation, F2P faces a metaphorical sledgehammer.

We've already seen regulation in F2P, with Japan outlawing a form of monetisation called 'kompu gacha' in 2012. Kompu gacha, meaning 'complete gacha', is a mechanic where players are rewarded for completing character sets dropped from gacha (also known as loot boxes). Since then, political pressure has been mounting across Europe and the world.

The F2P industry stands on the brink of regulation, which may cause endless headaches for developers. Governments have been proven not to understand video games, and their presence could tie developers in bureaucracy rather than focusing on what really matters: making great games. As game makers, we need to avoid censorship and regulation, push boundaries, but also accept responsibility for our players. And if we fail our players, then we need to make things right for them. 

#20



Contents



Attract mode

Interface



06. El Hijo

Honig Studios tell us about their unique stealth western

10. The Bradwell Conspiracy

More than just another day at the office

12. Chicken Police

We uncover a brooding, bonkers noir adventure

16. Incoming

Cavemen, Geralt, and Kojima's lucky ladders

18. Rodents everywhere

The meaning – and prevalence – of rats in video games

24. Revita

The lone dev behind a great-looking roguelite

44. The eternal coders

Meet the 1980s programmers still making games today

50. Shinji Mikami

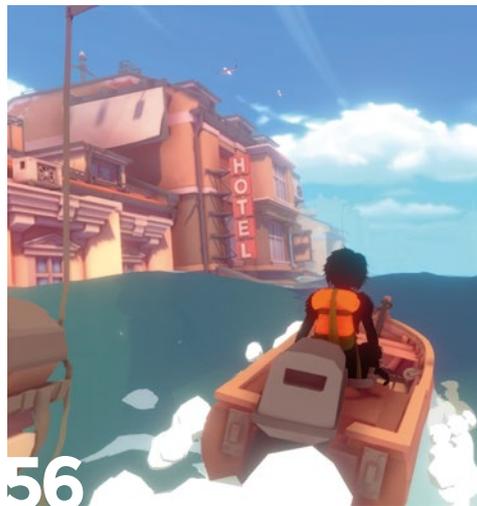
Inside the mind that gave us Resident Evil and lots more



12



18



56

Toolbox

- 30. New rules**
Tips for building better dev teams
- 32. Distributed development**
How even large studios can now work remotely
- 36. Live coding**
A guide to using shaders to entrancing effect
- 40. Source Code**
Make your own mini shooting gallery in Python

Rated

- 56. Sea of Solitude**
Set sail in a striking study of loneliness and turmoil
- 61. SolSeraph**
A nineties homage gets caught in the ActRaiser
- 62. They Are Billions**
An enthralling real-time war of attrition
- 64. Hamsterdam**
We hope the sequel's called Jason Stathamster

WELCOME

It's all gone a bit animal crackers this issue, what with chicken cops on page 12, an infestation of rodents on page 18, and our pick of some top-quality in-game creatures (a suspicious number of which are of the canine variety) on page 22. With this in mind, I dusted off my old copy of *Growl* (also known as *Runark*) – a Sega Mega Drive port of Taito's relatively obscure beat-'em-up from the early nineties. Now, by most yardsticks, *Growl* is a pretty bad game; its colours are drab, the music's forgettable, and it's possible to brawl your way through the entire experience by repeatedly jump-kicking everybody.

There is, however, one great thing about *Growl*: its premise. Cast in the role of an animal rescue ranger of some sort, it's your job to pummel your way through an army of evil, heavily armed poachers. As you variously punch, kick, and gun down said poachers, you can free animals from their cages, who'll return the favour by occasionally joining in the fray. This means that, just when the novelty of using rocket launchers and machine guns to fight men in flat caps begins to pall, a convocation of eagles will descend and peck a bunch of villains to death, or an elephant will enter stage left to headbutt a tank into oblivion. It's all brilliantly daft, and builds to a rousing finale that involves a mutating boss in a red top hat and a garage full of dancing wildlife.

Growl simply adds to our (highly unscientific) theory that adding animals to even the most mediocre game will make it ten percent better. Enjoy the new issue!

Ryan Lambie
Editor



THE GOOD, THE BAD, AND THE STEALTHY

We catch up with Berlin-based developer Honig Studios to find out more about their stealth western, *El Hijo*

The most immediately striking thing about *El Hijo* is how boldly its theme goes against the grain of a typical sneak-'em-up. Where most of the biggest stealth games tend to go for military themes (*Metal Gear*, *Splinter Cell*) or far-eastern settings (*Tenchu*, *Shadow Tactics*), *El Hijo* is instead inspired by the spaghetti westerns of director Sergio Leone. Its protagonist, meanwhile, is far from a hardened gun-slinger; following the destruction of the family homestead by vicious bandits, six-year-old El Hijo is left in the safety of a monastery by his mother. Determined to be reunited with her – and track down the villains who burned down his home – El Hijo decides to escape from the monks' care and traverse miles of desolate and often dangerous deserts and towns, with only his wits and a tiny catapult to protect him.

That plot is one of many allusions to the work of maverick Chilean director Alejandro Jodorowsky – specifically, his surreal 1970 western, *El Topo*, which even features a kid named El Hijo who is deposited at a neighbourhood monastery. But where *El Topo* was violent and often disturbing, *El Hijo* is playful and wonderfully delicate in its

presentation: the monastery, where the game begins, is all candlelight and long shadows, as monks quietly sweep floors or trudge up and down its gloomy hallways, seemingly lost in thought.

As *El Hijo* threads his way through the building, clinging to low walls or hanging back in the shadows, he – and by extension, the player – needs to find ever more ingenious places to hide and methods to avoid getting caught. When we got our hands on the game earlier this year, we constantly found ourselves smiling at the care developer Honig Studios have put into its design and animation: the way *El Hijo* nimbly

ducks behind curtains, jumps into tubs of water, or scuttles down the sides of a crumbling building to the next hiding place. Every frame of animation and background element feels as though it's been designed and rendered with the utmost care.

These early scenes also help forge a bond between the player and central character; unlike, say, Solid Snake, or *Splinter Cell*'s Sam Fisher, *El Hijo* can't exactly fight back if he's cornered – once spotted, the boy's only choices are to either run and find a shadowy place to hide, or simply stand and get caught. In the early areas we played through, the monks are a fairly benign bunch, and capture merely set us back a few paces

“That plot is one of many allusions to the work of maverick Chilean director Alejandro Jodorowsky”



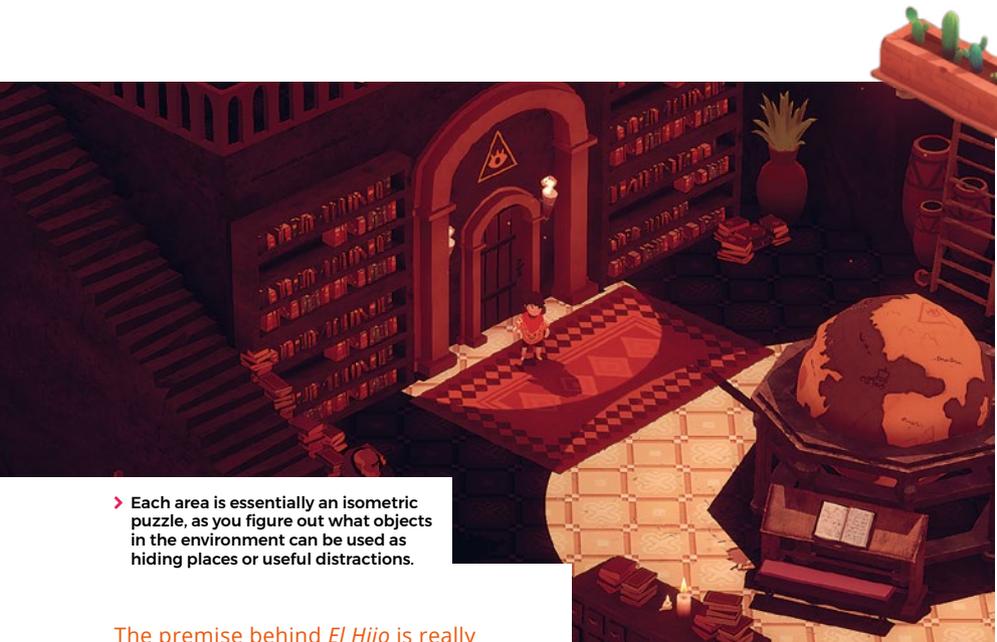
▲ *El Hijo*'s creative producer, Maria Grau-Stenzel.

in our hunt for freedom. Later, however, outdoor areas provide fewer places to duck, and marauding bandits only add to the escalating sense of threat – *El Hijo*'s catapult is handy for causing diversions and solving the odd puzzle here and there, but it's far too feeble to be considered a viable weapon.

El Hijo, then, is set to offer a refreshingly different spin on an established genre: in the place of stealthy kills or tension-lowering bursts of action, it's instead a game of hide-and-seek, with each area offering its own puzzle box of obstacles, refuges, and opportunities for mischief. With all this in mind, we tracked down *El Hijo*'s creative director Maria Grau-Stenzel to find out more about the game's development, and the thinking behind its likeably diminutive hero. ➔







➤ Each area is essentially an isometric puzzle, as you figure out what objects in the environment can be used as hiding places or useful distractions.

The premise behind *El Hijo* is really unusual, especially for a stealth game, which normally has military or far-eastern themes. So what came first, the western premise or the stealth mechanic?

We wanted to tell the story of a little boy's journey in the wild west. Stealth felt like the natural interaction kids would have if they found themselves in an intimidating and dangerous setting, relying on wit and smarts rather than strength. We also felt that using the unconventional western setting allows players to come into the game without the expectations of the usual stealth tropes.

Did you always plan *El Hijo* to be a larger, multiplatform title, or did you initially envisage it as a more contained mobile game, like *Impossible Bottles* (see boxout)?

We started developing *El Hijo* with the idea for mobile and PC. However, we always tried not to restrict ourselves. Keeping the bigger picture in mind, we wanted to leave as many doors open as possible.

I read that *El Hijo* was inspired by *El Topo*; there can't be many game developers inspired by Alejandro Jodorowsky, so what was it from the movie that first planted the seed for the game?

The strong visual language of the film was very inspiring. We wanted to try to translate this visual language into the game. Therefore, we opted to not have any verbal dialogue and mainly use visual

storytelling by creating a sense of story through settings, characters, animations, environment, development in gameplay, music, and so on.

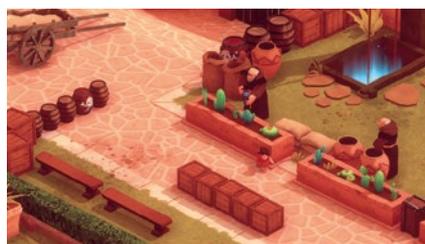
Does the game get as surreal as Jodorowsky's films so often did?

Initially, we had dream-like sequences. These were quite surreal, in which monks were throwing wine barrels resembling the ape in *Donkey Kong*. We also intended to implement a giant sand octopus in the desert that El Hijo had to outsmart, and so on. But as the game developed, we moved away from these scenes to focus on the main story, which was challenging enough.

El Hijo's also described quite specifically as a 'spaghetti western stealth game'; what is it about that particular sub-genre that appeals to you? Is it the cinematic style of, say, Sergio Leone's films?

The atmosphere is drawn [from] Sergio Leone's westerns and the

✓ Although you're largely defenceless, there are later stages where the boy's catapult can have an explosive effect if you aim it in the right place.



aesthetics from Saul Bass and German Expressionism. We were fascinated by the contrast that spaghetti westerns have: defenceless characters, trickery and deceit of opposing gangs, humour, and even surrealism.

The slice of the game I've played didn't use any dialogue, or didn't appear to. Is that something you've stuck to throughout the experience?

In *El Hijo*, we are translating the verbal dialogue into a visual language by allowing the environment, character animations, and their encounters, as well as the music, to tell the story without ever having the need to actually talk.

What's the process of developing a stealth game like? Is it a long, iterative process designing stealth levels that are challenging but entertaining at the same time?

We think that it's very similar to any other game development.

Initially there's a concept, there are some tests, maybe a prototype. Then there are new tests, adjustments/balancing and even more tests. New features are added, which require – guess what – tests again, before we actually start to visually design the levels, implement animations, music etc.

Overall, it is a very iterative process. It's very challenging at times, but also very rewarding when it works out.

Do you have ideas sessions, where you come up with the different NPCs and hazards for the player to encounter?

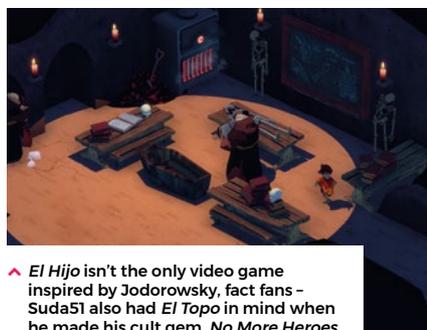
We started with a bunch of ideas for different NPCs and puzzles, however, as the game develops, ideas change. We tried to keep this as tangible as possible; giving space to explore and test. Giving us the chance to adjust or exchange ideas that make more sense while you develop, and that could not be predicted when [we started].

On that note, what's been the biggest challenge so far in *El Hijo*'s development?

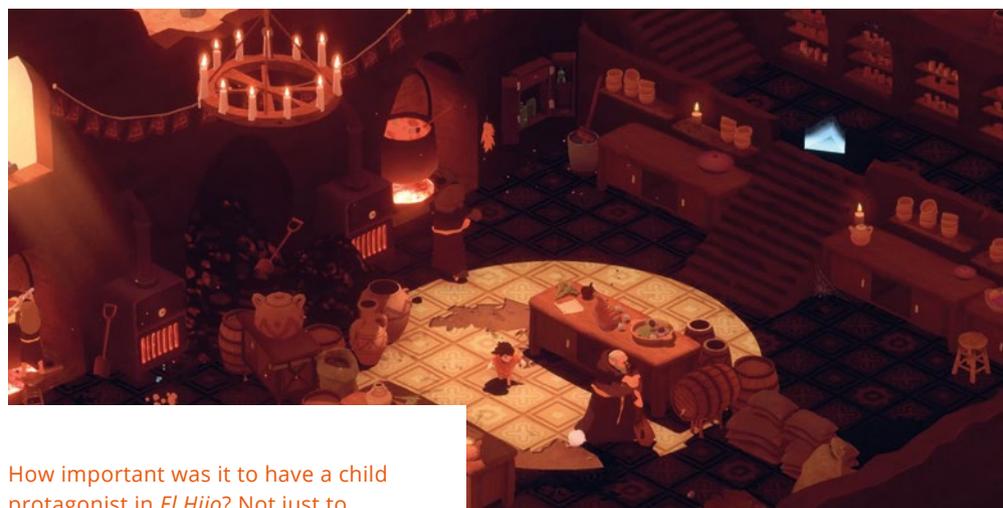




^ *El Hijo* uses subtle visual cues to guide the player along.



^ *El Hijo* isn't the only video game inspired by Jodorowsky, fact fans - Suda51 also had *El Topo* in mind when he made his cult gem, *No More Heroes*.



^ Honig Studios have described *El Hijo* as a coming-of-age tale, with the boy growing more heroic as he goes on his journey.

The biggest challenge was in the pre-production - finding the right tools that helped us translate our vision into a game.

Is the stealth genre a relatively untapped one for indie developers, do you think?

Stealth games are surprisingly sparse in the indie scene, although there are of course some brilliant examples. We definitely see a lot of potential and would love to see what other developers could come up with.

There's a strong sense of light and shade in *El Hijo's* visual style, which feels just right for a stealth game. Did it take a while to hit on the right style and colour palette for the game?

Oh yes, the first ten levels play mainly indoors in the monastery, which was fairly easy in regards to using light and shade. However, once we hit the desert, where shadows are rather rare, we had to come up with a new idea. Our logic switches though: once the player is out in the desert and shadows are scarce, he/she will need to find other ways to distract the foes to make his/her way through the environment.

Did you need to do a lot of real-world research as you were coming up with the game's locations?

We did a lot of research and collected a lot of different moods on possible locations, in order to find the right style and the right tools to get the exact look of the environment we wanted.

How important was it to have a child protagonist in *El Hijo*? Not just to establish a sense of vulnerability, but also to eliminate any possibility of violence. In many stealth games, you can at least fight back to a degree if you're caught; that doesn't appear to be an option in *El Hijo*.

Putting a child in the central role was pivotal in allowing us to naturally extract the violence that has become synonymous with the stealth genre. We feel that this allows us to place more importance on the environment and characters. Allowing the player to explore and go on an adventure at a human level rather than living out power fantasies.

Finally, what's your overall aim for *El Hijo*? Is there an emotion or theme that you want players to walk away with?

In our story, the protagonist El Hijo does not accept his fate - he sets off to find his mother even if this means a long arduous journey, because this is what he wants to pursue. Over the course of his journey, El Hijo will come across many children who have passively accepted their fate. He will inspire these children who will evolve throughout the different environments, becoming progressively more active and revolutionary in response to El Hijo's interaction with them.

We would like to pass a bit of El Hijo's attitude on to the player. If you believe in something, don't accept that it might seem impossible. Be proactive about it. Be [courageous], as your positive drive might be inspirational for others. 🙌

***El Hijo* is due for release later in 2019 for PC, Mac, Switch, PS4, and Xbox One.**



^ A handheld rhythm game, *Impossible Bottles* showcases Rafael Varona's animation.

MISSION: IMPOSSIBLE

El Hijo isn't the only game in the works at Honig Studios: one of its other works-in-progress is *Impossible Bottles* - a rhythm game that takes in electronic music, robots, and a curious storyline that involves a scientist's efforts to fuse machines and organic life to create a new kind of renewable energy. Like *El Hijo*, its visuals are immediately striking, with the entire game built around the surreal animated loops of Berlin-based artist Rafael Varona. Cut to an evolving score by Phivos-Angelos Kollias and due out this year for Nintendo Switch and mobile, *Impossible Bottles* could well become the most offbeat rhythm game since *PaRappa The Rapper*.



The Bradwell Conspiracy

A Brave Plan's debut focuses on narrative and engaging settings

Info

GENRE
Adventure, Puzzle

FORMAT
PC

DEVELOPER
A Brave Plan

PUBLISHER
Bossa Studios

RELEASE
Autumn 2019

▼ Despite the allusions to *Firewatch*, *A Brave Plan* promises the relationship between the player and Amber is "like no other."

Following the success of narrative-led exploration games like *Firewatch* and *Gone Home*, many doors were opened on the indie scene for developers who prioritised telling stories and creating interesting environments over designing complicated mechanics. This was the exact motivation London-based studio A Brave Plan needed when producing its mystery thriller, *The Bradwell Conspiracy*.

"We saw *Firewatch* get a warm reception, and we felt validated that we were on the right path," says *The Bradwell Conspiracy* director Georg Backer, whose previous work includes *Fable* and *The Movies* at Lionhead Studios. Backer calls the game "Spielberg-esque" as it doesn't lean too hard into sci-fi or horror, and that it remains accessible for everyone to enjoy.

You are an employee of the tech conglomerate Bradwell Electronics in the near-future year of 2026. At the launch of the company's latest innovation in humanitarian

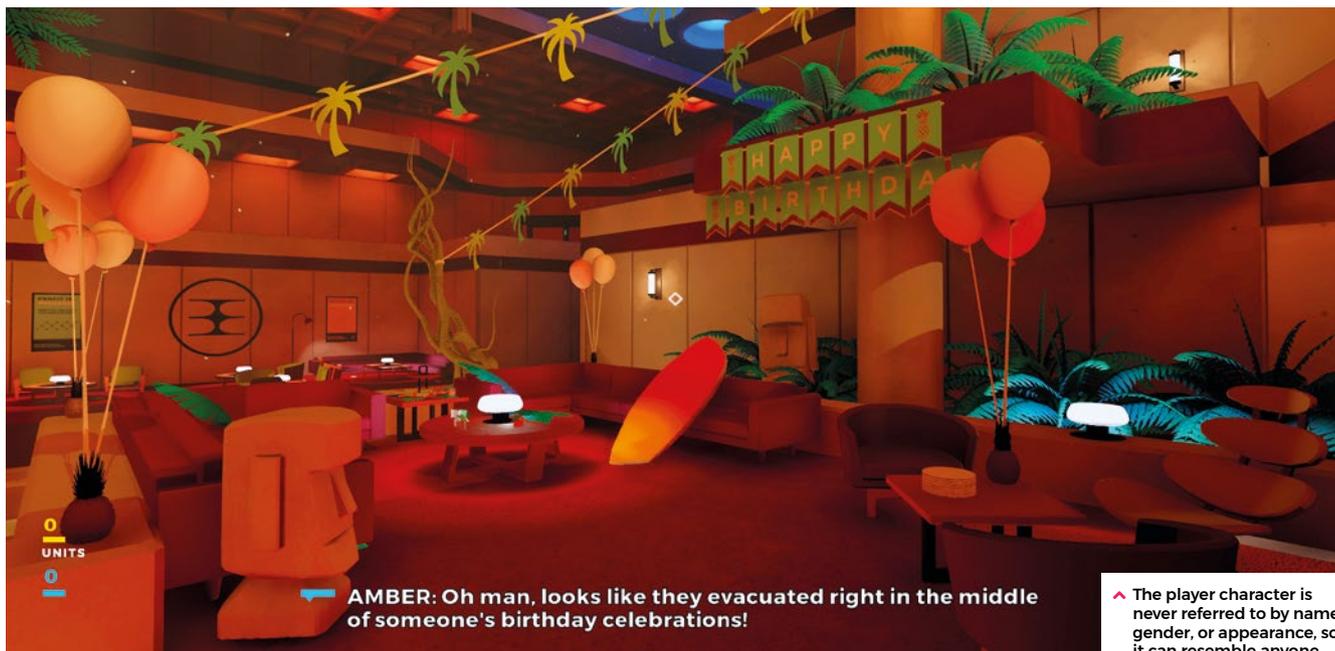
aid, an explosion occurs that puts the facility on lockdown. Trapped in an isolated environment, your only companion is Amber – a Bradwell worker stuck in another part of the building.

Your primary form of communication is from using a pair of augmented reality glasses, which lets you take photos you can send to her. This allows you both to work together to find a means of escape. You can photograph anything you come across, from doors to signs, to random objects within the facility, with each point of interest yielding unique dialogue opportunities with Amber.

"I wanted to design a unique way to communicate with another character that doesn't use dialogue trees," Backer says. "We have a system that tracks where Amber is within the story, so when you send photos to her, she might react differently depending on what point you're at. If you don't check in with her after a while, she may get in touch with you to see what's up. The story is all about you both having the same goal, so it feels natural for her to be concerned about the player."

This type of relationship with an AI character immediately calls back to *Firewatch*'s Delilah, and how you would communicate with an unseen individual via a walkie-talkie. While Amber – like the protagonist – is an employee of Bradwell, she doesn't know everything. As you explore the facility, there are areas you discover together. This is where the Induction Narrator, voiced by television personality Jonathan Ross, steps in and acts as a personal adviser for the player, like a museum tour guide, and fills in the blanks where Amber can't.





AMBER: Oh man, looks like they evacuated right in the middle of someone's birthday celebrations!

^ The player character is never referred to by name, gender, or appearance, so it can resemble anyone.

Puzzle segments also form a large component of the game, with many involving the use of a 3D printing mechanic to create objects that help you progress through an area. The setup of these puzzles as self-contained rooms is reminiscent of *Portal*'s test chambers, areas that acted almost like playgrounds for testing the game's limits while also providing narrative clues that added to the overall plot. *Bradwell*'s environments act similarly but have a much bigger focus on telling stories.

According to art director and narrative designer Holly Pickering, the team's main intention is to use these environments to create profound stories. "When we first started work, we looked up games like *Portal* and *Firewatch*.

We're also big fans of the immersive sim genre, and we love how deep those stories can go. We're a small team so we can't do a big, physics-based game, but we can build up smaller things to help you get a further understanding of the world.

"You draw lines and build the story up in the background as you see how this relates to your adventure with Amber. We want to reward players who take the time to reach into this information, and they'll come away with a deeper understanding by the end." Pickering had previously worked on *Ether One* while at White Paper Games, which took a similar approach of prioritising narrative over puzzles.

The Bradwell Conspiracy is A Brave Plan's first game, created as a collaborative effort between

▼ Puzzles may not be the focus, but solving them is still an integral part of the story.



"award-winning BAFTA luminaries and triple-A veterans" with a team consisting of developers from games such as *Fable*, *Assassin's Creed*, and *Tomb Raider*. "It was all about picking the right people for the right project," Backer says. "Whoever we needed, we knocked on their doors and asked them kindly. Most of them said yes, which was great."

While keeping further details under wraps, A Brave Plan has since shown the game at public events. "Everyone's been really positive; it's great to see people are taking the time to let the worldbuilding sink in," Pickering says. "Normally when we get developers to playtest, they're so concerned with the mechanics and technical side, but the narrative and art is really hitting home with the public. At the end of the day, we're building a game for people to immerse themselves in when they play at home." 🐾

"We're a small team so we can't do a big, physics-based game"

FAMILIAR VOICES

There's a lot of high-profile talent behind *The Bradwell Conspiracy*. Joining TV's Jonathan Ross is Abubakar Salim, who voiced Bayek in *Assassin's Creed Origins*, and Broadway actress Rebecca LaChance. Performance director Kate Saxon, whose credits include *Alien: Isolation*, *The Witcher 3*, and the 2016 reboot of *Hitman*, brings these characters to life. Austin Wintory, composer for *Journey* and *Assassin's Creed Syndicate*, creates music for the game.



Chicken Police

Less Police Squad, more Chinatown... but with chickens

Info

- GENRE**
Noir adventure
(with chickens)
- FORMAT**
PC
- DEVELOPER**
The Wild Gentlemen
- PUBLISHER**
The Wild Gentlemen
- RELEASE**
TBC 2020



Not pictured:
Rocket Raccoon.

Characters are all animal heads and human bodies.

It might look like your traditional wacky internet nonsense, but *Chicken Police* – by developer The Wild Gentlemen’s admission – is best described as ‘Orwellian buddy cop animal noir’.

This might be a game featuring a couple of chickens as – surprise – police officers, but there’s also a dark, satirical story behind it all.

What do you do as the rooster cops? Detective work, of course, with point-and-click mechanics joining elements from visual novels – and a bit more sprinkled on top that the studio’s keeping close to its chest for now. But by far the most striking aspect of *Chicken Police* is its visual style, mixing classic forties film noir with a cut-out collage look straight from the pages of *Farm Animals Weekly* (not a real magazine).

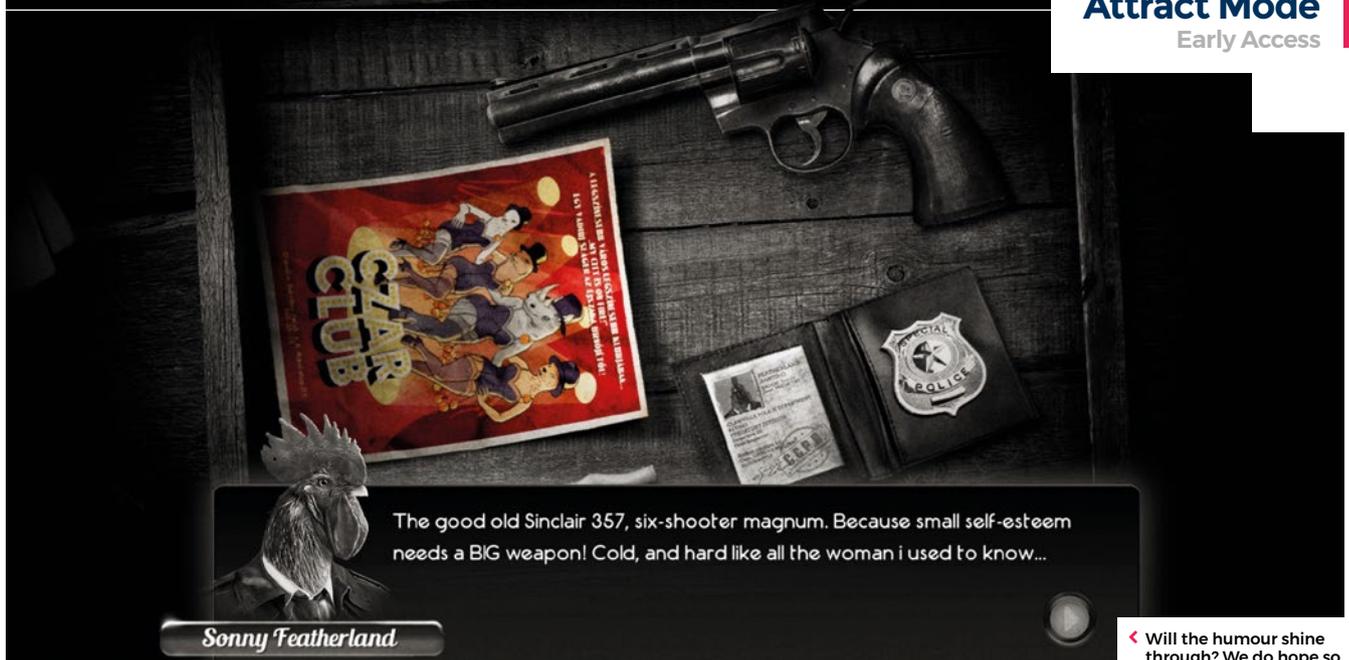
“We use the animal characteristics of the game largely the same way Orwell’s *Animal Farm* did,” explains Peter Nadas, co-founder of The Wild Gentlemen. “While not all our characters are purely there for satirical reasons, the world of the game, which we call The Wilderness, is.” Though Nadas isn’t forthcoming as to how

or what *Chicken Police* satirises, the general overview sees players looking after rooster cop partners, Sonny and Marty. The duo was once known all around Clawville – as, Nadas says, “celebrity cops, real local heroes” – but the story takes place a decade down the line, with each character radically different from the cop – and chicken – they were before. “*Chicken Police* intentionally confronts its hard-boiled story and design with its absurd humour to create the same ambivalence our heroes encountered during their careers,” Nadas says.

Roosters and cats and foxes might lead you down the path of thinking this is a game for kids, but those who’ve been paying attention to the words used previously like ‘gritty’ and so on won’t be surprised to hear it’s not the case. “While our story is intended for an adult audience, the absurdity of our characters creates a strong dissonance,” Nadas says. “That’s what makes the animal characters, the comical title, but even the intentionally limited use of colours, so important. The contrast/dissonance itself is a deliberate design decision which the game itself completes and will partially resolve. The ambivalent environment and this sharp contrast are both main strengths of the game, and the world we’ve built around it.”

This unique approach came about partly thanks to one of the studio’s founding members funding development to get the game to a saleable point. The team has been working on it as a game since “somewhere between 2014 and 2015” with the original idea actually popping up as an idea from *Chicken Police*’s writer, Balint Bank Varga, who envisioned the project as





◀ Will the humour shine through? We do hope so.

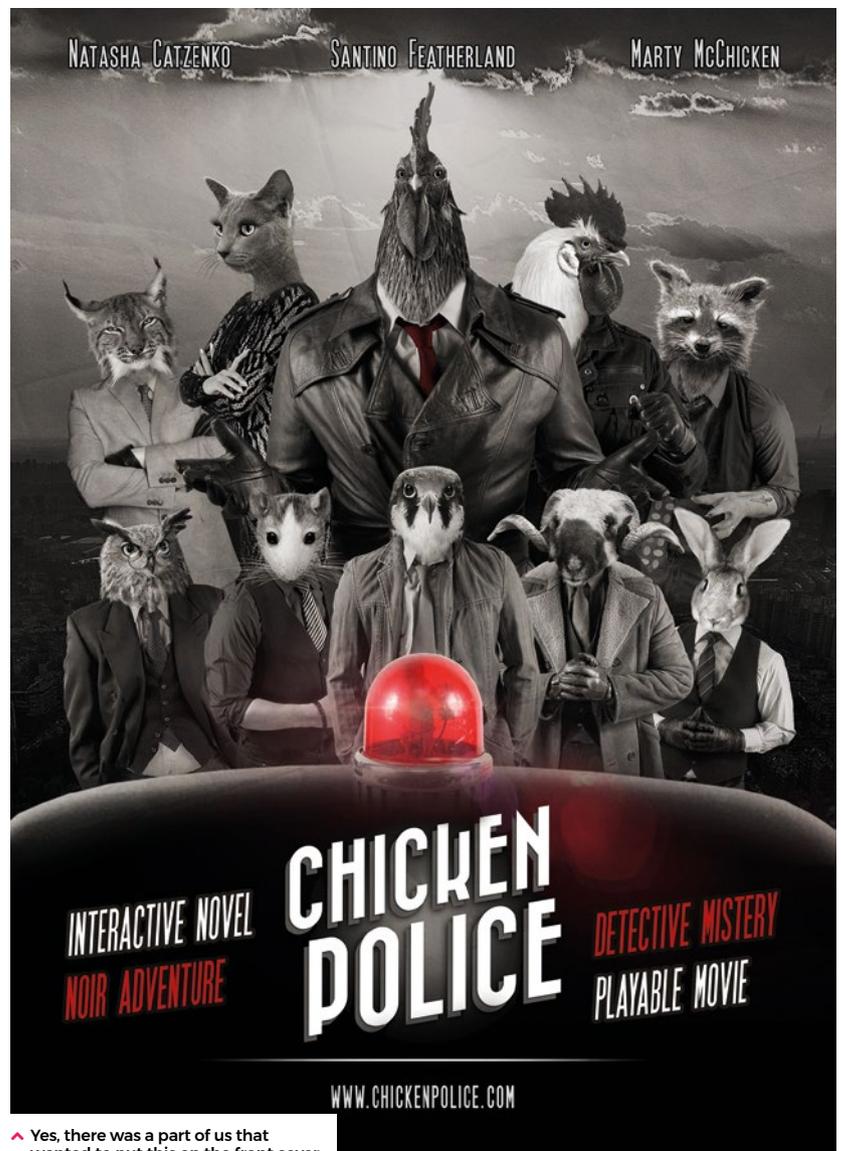
an animation. “The Wild Gentlemen – our studio – was only founded in 2018 when we finally had the chance to get more serious about the game,” Nadas says. “We threw away a lot of the original concept – though we kept the story, characters, and main concept – and started building the project from scratch. So while the roots are long and deep, the real development only started a year ago in Unity.”

It’s a game that needs to do the rounds to get people on board, and a fine little trailer –

“While the roots are long and deep, the real development only started a year ago”

complete with original lounge jazz-style noirish song – did a good job. Beyond that, Nadas and the team have been touring *Chicken Police* around events, and will be releasing a demo to the public towards the end of August. “The show demo was well-received, and we gained a lot of valuable feedback, too,” Nadas says. “We’ve been approached by several publishers, which has been a huge boost, and basically everyone who sat down and played the game got up with a smile on their faces, which was great feedback for us.”

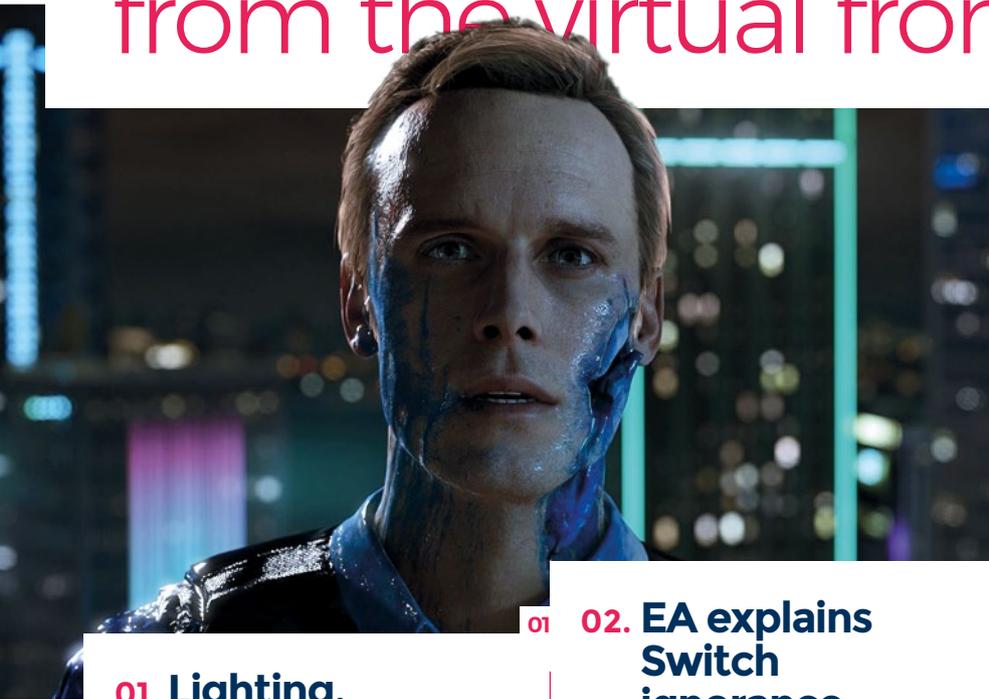
You’ll have a similar smile on first seeing the game, I’d assume, but the hope is there’s enough behind this first release from the Hungarian indie studio to break through the initial smirksome look of things. As long as the satire and humour holds up, there’ll at least be a couple of other things to discuss beyond the fact that it’s a game called *Chicken Police*, in which the main characters are chickens. Who are also police. 🐔



▲ Yes, there was a part of us that wanted to put this on the front cover.

Headlines

from the virtual front



01. Lighting, apparently

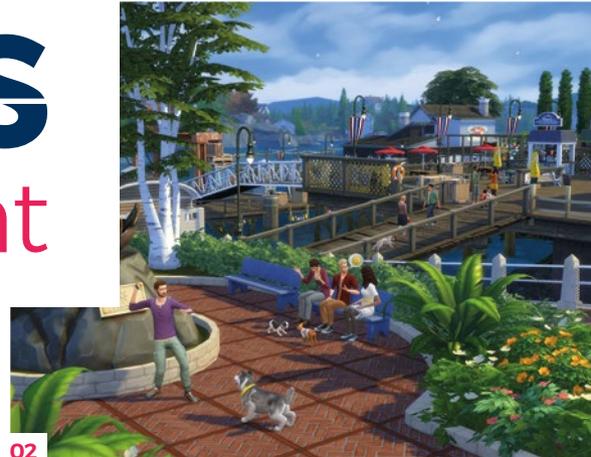
We tend to avoid much mention of David Cage, but the man did make a good point recently so let's break that duck for now: chatting in an interview with GameSpot, the Quantic Dream CEO and director of the likes of *Detroit: Become Human* and *Heavy Rain* explained where he sees the next technological arms race in gaming going. And it's not resolution.

While the hardware manufacturers might want to push things like 8K at us, Cage highlights elements like ray tracing as an actual key forward step in graphical tech. "I don't think it's going to be a war about resolution," he said. "I know that people talk about 8K these days and blah blah blah. I don't think this is the real next battle. I would rather put focus on lighting. Lighting, lighting, lighting." Lighting then, David?

02. EA explains Switch ignorance

We're running a bit low on EA games for the Switch, which is odd considering the two companies went to some length to show off the fact they were Super Best Friends at the console's launch. Instead, we've ended up in a situation similar to previous Nintendo consoles, where one of the world's biggest publishers largely ignores it.

We may have found out partly why that is thanks to an investor's call with EA chief Andrew Wilson – when asked why things like *The Sims 4* weren't on Switch, Wilson replied that the company evaluated two things on bringing a game to a platform: is it a good fit, and is it something Switch owners wouldn't just play somewhere else. Yep, EA doesn't bring games to Switch because Switch owners will just play them somewhere else. Brilliant.



02



03

03. IMPORTANT NEWS STORY

Is this worth an entire headline? We're going to go with yes. Hideo Kojima was stopped on his way into the United States, where he was travelling to for Comic-Con in San Diego, because he was carrying a physical re-creation of the baby in a pod from *Death Stranding*. The one you can see above these words. Border control was wondering just what a man would need a fake baby in a plastic container for, and we're sure the questioning surrounding it was... enlightening. Maybe the border guard now knows more about *Death Stranding* than we do, who knows?



Cyberpunk 2077 to get physical card game, *Afterlife*, next year

Ninja leaves Twitch, joins Mixer for reported \$50-100m. Million. Not pence



04

All these stories and more!
(we have more space on the internet)
can be found over on wfmag.cc



05
DONE BY
MICHAEL K GLASS

05. Egg hunt

The very first gaming Easter egg may well have been discovered, with *Spitfire* on the Fairchild Channel F revealed to have been hosting a hidden credit to its creator since its release 42 years ago.

Discovered by user RT-55J on the SelectButton forums, the credit is accessed by inputting a very long series of numerical inputs on the title's game select screen, resulting in a page flashing up saying 'Done By Michael K Glass.'

Adventure, the Atari 2600 game from 1979, was home to the former earliest-known Easter egg, before being pipped by the 2017 discovery of another hidden screen in Atari's 1977 release *Starship 1*. After a bit of digging – and a newspaper ad confirming things – forum-goers were able to clarify that *Spitfire* did indeed release some four months prior to *Starship 1*, in or around April 1977. Thus, as far as we know right now, it is indeed the earliest Easter egg ever seen in a video game.

You can read more – and you should, because there's plenty more to it – here: wfmag.cc/first-to-the-egg.



06

06. Mega delay

Sega's Mega Drive Mini has been delayed in Europe, the company has announced, owing to 'unavoidable logistical challenges'. Originally set to launch in September, the tiny chunk of nostalgic plastic will now hit UK retail on 4 October, still for the not-that-bad price of £69.99.

"We are committed to a successful launch for consumers in all territories, and this step will ensure we can meet demand and fulfil all the individual pre-orders and retail orders in Europe and the Middle East at the same time," said a statement from Sega.

There's been no mention of a delay in the US, nor has there been mention of Sega correcting the name of the console for the territory – there appears to be a misprint where the device is referred to as a 'Genesis', which is very strange.

04. Undefiant

Defiant Development, the studio behind *Hand of Fate*, has announced it will no longer be working on new video games. Its development team is currently looking for new roles as a result. What remains of the studio will continue in a caretaker role, taking... care of the studio's existing releases, dealing with low-level support, and that kind of thing.

A statement from the studio read: "The Defiant model has always focused on creating games nobody else would... That is a risky way to make games, and we knew that. When it succeeds, it delivers things you could never have considered possible. When it fails, it leaves you without a safety net. The games market has changed in ways both big and small in the nine years we've been in business. We have not been able to change quickly enough to continue with them."

Fortnite World Cup viewing figures peak at 2.3 million



The Outer Worlds is coming to Switch, which could be amazing

Preview



Jet Kave Adventure ↗

Ignore the promotional art and go straight to watching videos of *Jet Kave Adventure* – in doing so you're rewarded with the knowledge this looks like it'll actually play something like *Donkey Kong Country*. Thanks to living in the future, as we do, this caveman platformer is actually getting more of a look-in than it might otherwise have.



The Last Campfire ↗

Hello Games has been hard at work trying to meet impossible expectations elsewhere, but it hasn't forgotten about making new games. *The Last Campfire* is a smaller experience than *No Man's Sky* – it's described as a 'short' – and it certainly has less hype behind it. That shouldn't stop you from keeping your eyes on this characterful little adventure, though.



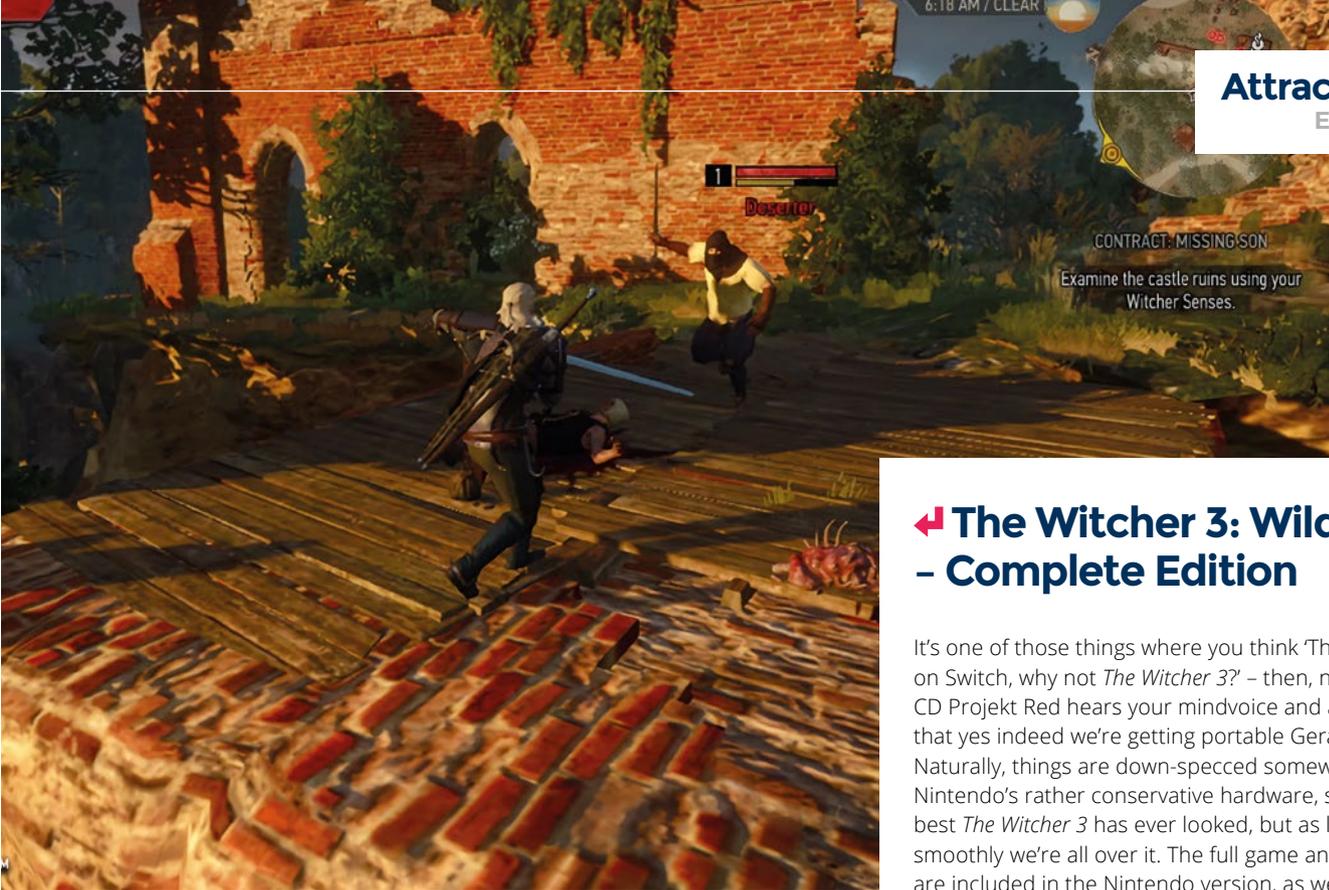
Neon City Riders ↘

Fighting, puzzling, exploring, and cyberpunk...ing, *Neon City Riders* is aiming to combine the best of the eighties – or at least the best of our vague memories of the eighties – with some fine indie game sensibilities. It's a simple case of working your way through four different gang-controlled turfs, each with its own distinct aesthetic, and working your way to the area's gang boss in order to take them on and out. It looks great, it sounds refreshing, and we're looking forward to this one releasing in 2020.

Alaloth – Champions of The Four Kingdoms ↗

No game is for everyone, so it's a redundant statement to make, but really, in this case... *Alaloth* won't be for everyone. Stubbornly old school in its visual style, this action-RPG from Gamera Interactive is actually a bit more interesting under the surface, describing itself as an 'isometric Soulslike'. Combat and dungeon-crawling is to be expected, along with character customisation and multiplayer of both the competitive and co-operative style. The cherry on top of this is the involvement of Chris Avellone, who has inked the story behind *Alaloth*. None of this guarantees it'll be a great finished package, but the individual pieces are looking good enough in their own right.





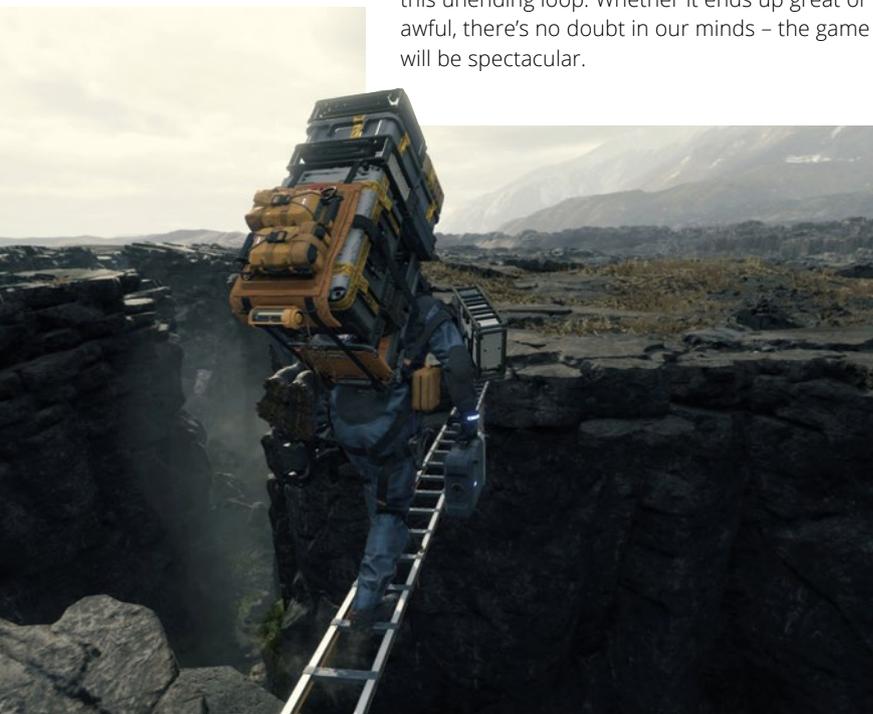
← The Witcher 3: Wild Hunt - Complete Edition

It's one of those things where you think 'They did *Skyrim* on Switch, why not *The Witcher 3*?' – then, naturally, CD Projekt Red hears your mindvoice and announces that yes indeed we're getting portable Geralt of Rivia. Naturally, things are down-specced somewhat to fit into Nintendo's rather conservative hardware, so this isn't the best *The Witcher 3* has ever looked, but as long as it runs smoothly we're all over it. The full game and all add-ons are included in the Nintendo version, as well as *that* scene with the unicorn – another step towards Nintendo completely abandoning its family-friendly image.

This isn't a case of the game featuring all-new newness as it arrives on the format, with no announcements of any significant additional content or the like, so usually, it would be the sort of thing we'd smile and nod at before moving on. But it feels more significant than that; it's another vote of confidence from the world of game makers, in a format still sometimes written off for its relative lack of power. And long may it continue.

Death Stranding ↘

The hype train is picking up pace as we approach the release of Kojima Productions' first independent title, and it's getting odder as we go along. As well as an ultra-ladder, extendible seemingly forever, *Death Stranding* has also introduced us to a man who lives in 21-minute bursts, and so has to plan his entire life around this unending loop. Whether it ends up great or awful, there's no doubt in our minds – the game will be spectacular.



Moving Out ↘

While it's true you can make a video game out of pretty much any experience, we were under the – correct, we assumed – assumption that making moving house fun is something even the world of interactive digital entertainment products couldn't handle. *Moving Out* begs to differ, bringing the ethos of controlled chaos from *Overcooked* into the world of moving boxes from one place to another. However this one turns out, there's guaranteed to be less backache.



Reviling RODENTS

How games made rats the bad guys

WRITTEN BY SEAN MARTIN

From a staple enemy of fantasy RPG, to the visceral hordes of narrative-driven games, rats are frequently depicted as evil. But what does that evil amount to? Why are rats the perfect enemy? Or are they even enemies at all?

I remember my first encounter with video game rats – in 2001's *Max Payne* there's a subway level, where if you throw a grenade into a rathole, you will be granted an extra objective stating, 'I have declared war on the rats'. The rodents will arm themselves with Desert Eagles, and proceed to enact their revenge on the gurning detective. It's a weird Easter egg, but 'killer rats' is actually pretty on-point in reflecting the way we view rats in games. In the years that followed, I covered myself in rat blood in *Dragon Age*, farmed rats for humanity in *Dark Souls*, and committed single-handed genocide in *Vermintide* – all without ever really thinking about it. That's probably why rats are such great enemies – our cultural distaste for them runs so deep, that in our minds they easily occupy the position.

We tell stories about rats doing evil things – in *The Witcher 3*, it's a paralysed nobleman's daughter being eaten alive, in *Dishonored*, a city of rat bites and body bags. But who's to blame in those stories? What's far more interesting about video game rats is their ability to be both

an enemy, and somehow occupy a position of blamelessness. They only act according to their nature. But that begs the question: in human terms, is their nature evil?

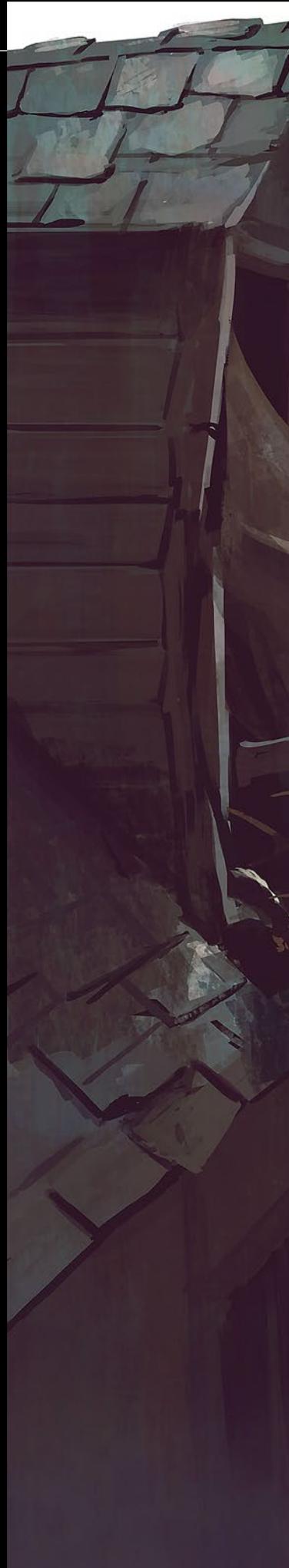
THE PLAGUE

One of our oldest negative associations of rats relates to the Black Death, the period of plague during Medieval Europe, which they were originally blamed for. *A Plague Tale: Innocence* is a game which seeks to exaggerate that story for the purpose of the in-game experience – creating a plague of rats more akin to folk-tale than a historically accurate depiction. As creative director on the game, David Dedeine says, "We have been warpers of history!" The result of this so-called warping is, however, pretty terrifying. To experience one of *A Plague Tale's* rat-swarms is to witness a nightmare. "It speaks directly to the reptilian brain, your body actually reacts to it," Dedeine jokes. "The brain can't describe it, that's what makes it scary." Staring into the writhing, red-eyed mass of rodent bodies, pouring from the walls, and through cracks in the floor, I'm inclined to agree with him. But part of what also makes them scary, is that they're always present. "It's like the shark in *Jaws*," Dedeine explains. "When you're on the surface of the sea, you don't know where the shark is. Rats are the same." We all know the urban myth, that in a city you're never more than two metres away from a rat – *A Plague Tale* effectively creates the mechanical version of this. As Dedeine says: "Every single hole in the wall can begin to tell a story."

Another reason *A Plague Tale's* rats are terrifying is because they cease to be ➔



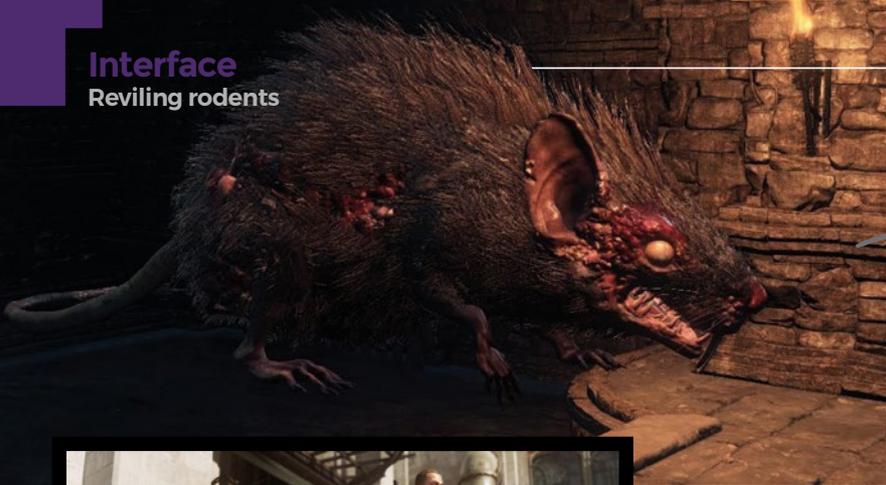
◀ The anthropomorphism of the Skaven is evident.





LIGHT AND SHADOW

"We realised that light was a really important thing at this stage in history – a candle was something for the rich. When you were poor you didn't have the money to even light your own house, or it was with a bad candle that polluted the room," Dedeine says. "Light is given here, where for them it was not. Then the prototype of the rat-swarm came. In the shadow, you are hidden from the inquisition, but are prey for the rats. In the light, you are safe from the rats, but revealed to the inquisition. It works both ways."



^ Visually, you're being pointed in a definite direction as to what to think of our ratty non-friends.

< The rats of *Dishonored* share an affinity with all those who have touched the void.



All you need do is look at their clans to realise this – Clan Pestilens, based on our association between rats and plague; Clan Eshin, on the idea that rats are sneaky and treacherous; Clan Skryre, twisting our idea of lab-rats; and Clan Moulder, playing on our disgust at rat survivability. The Skaven are essentially cartoon villains, caricatures, and their society, a hierarchical, self-cannibalising monster, that just won't stop growing – in many ways they're the perfect antagonist. The *Vermintide* games sought to take advantage of this – four heroes facing off against tides of ratmen, slashing and bludgeoning in a horde-killing frenzy. But the majority of the rats you kill in the game are Skavenslaves, the lowest strata of rat society – ignorant creatures, riled into a frenzy and let loose. You rarely kill the masters, the Warlords and Grey Seers pulling the strings. So it begs the question, are Skaven evil by nature? Or are they merely victims of their society?

Anders De Geer, game director on *Vermintide*, knows the answer. "A Skavenslave is just a Warlord that hasn't found a way to climb or back-stab his way to the top yet. They are egocentric, greedy, envious, unloving, cowardly, power-hungry, ruthless creatures." The anthropomorphic quality of the Skaven is one of their most interesting aspects – it's in the name: ratmen. But which part is rat? Which part is man? De Geer insists that what is worst in them comes from us. "Skaven are not supposed to be rats acting like humans, but rather humans acting like rats." Just as in *A Plague Tale*, we see how rats can act as a mirror for "mankind's worst characteristics", as De Geer phrases it. While it's absurd to compare Skaven to humanity, it does draw some uncomfortable parallels. As De Geer says: "I think the main reason Skaven commit their worst atrocities against themselves, is that they live among other Skaven."

THE VICTIM

Dishonored has always been a game series about victimisation – its protagonists stripped of their finery and cast into the gutter, alongside the rats and plague victims. It fosters an almost sympathetic alignment with rats, which – while dangerous – become circumstantial allies. Through its Chaos system, *Dishonored* shows an awareness that rats merely feed upon evil – the more bodies you leave in your wake, the

MISUNDERSTOOD CREATURES

"The Outsider was abused, cast out, and murdered in order to unify a community in the commonality of their hatred. The Outsider is perceived as the source of evil and as an object of blame in order to displace society's problems away from themselves," Monforton explains of *Dishonored's* backing tale. "As such, the Outsider chooses to bestow magic on people who have been wronged and scapegoated in a similar manner, and watches to see what they will do when given power over their oppressors. In this understanding, the Outsider shares an affinity with the rats for the same reasons: they are all hunted, murdered, misunderstood creatures whose suffering fuels the world."

individuals – in their uncountable swarms, they represent only a unity of hunger. But is hunger evil? Smash a guard's lantern with main character Amicia's sling, and they'll be eaten, but drop a ham into the swarm and the rats will devour it with the same glee. There is an equality to the rats' unscrupulous hunger, which makes it hard to blame them. Especially when throughout the game you'll witness such conscious evil acts – people being burned at the stake, battlefields littered with corpses, and more murders than you can count. Rats feed on evil – it's one of the main reasons our society hates them. They act as a mirror, their prevalence a directly linked reminder of our own evil acts. In *A Plague Tale: Innocence*, Amicia and her young brother Hugo are the innocent party traversing a brutal world; but I'm starting to think that perhaps rats too are innocent in their own way.

THE VILLAIN

In the world of Warhammer Fantasy, another interesting depiction can be found – the Skaven, a race of villainous sentient rats modelled almost directly upon our negative cultural associations.

^ Sometimes it's a combo, like the not-so-humble Mole Rat of *Fallout* fame.





^ The behaviour of the rats is exaggerated by what Dedeine refers to as "an alchemical imbalance".

more rats will appear, and the more the plague will spread. It's easy to see *Dishonored's* rats as evil at first, but in revealing that the Lord Regent displaced them and brought them to the city, we realise they're merely an unsuspecting aspect of his evil schemes.

This changing perspective becomes its most interesting in *Dishonored: Death of the Outsider*, a game which allows you, through dark magic, to hear rats speak. Hazel Monforton was a narrative designer on the game, and wrote many of these 'rat-whispers'. "We all deeply loved the idea of hearing the rats' inner thoughts for the first time in the series," Monforton explains. "They want to live, to be safe, to be fed, to have sex, to take care of their children. They have emotions, and relationships, and even an oral history." One of the most powerful parts of this animal perspective is that they speak in the voice of a young girl, Deirdre, protagonist Billie Lurk's childhood sweetheart. "The voice stresses that the rats are very small, very weak, and very childlike themselves. They come from a hostile environment that is stacked against them," Monforton says. "In particular, the blood-curdling scream when the rats are harmed or frightened is very affecting. I know some fans reloaded earlier saves after doing that because they felt so guilty." Billie Lurk also represents a unique perspective to empathise with the rats. "She is from the street and the city in a way that Corvo and Emily are not," Monforton explains. "She grew up in poverty, watched the rich destroy her life and loved ones, became a killer on behalf of others, and lived in the abandoned, filthy, and secret places of both Dunwall and Karnaca. The rats are a smaller reflection of Billie's same themes. Their lives are intimately intertwined."

"The voice says the rats are small, weak, and childlike themselves"

This familiarity and insight into the rats cast them in a completely new light – they become living inhabitants, victims of the city in much the same way as any other, exultant in pleasure, and despairing in pain. It is the counterpoint to *A Plague Tale's* depiction – a rat as a thinking, feeling, individual creature, defined by more than just hunger. It is also a far cry from *Vermintide's* anthropomorphic depiction of rats as a vehicle for the negative characteristics of humanity. "This is our baggage, not theirs," Monforton explains. "At most, rats are reactive to humans and how we choose to live. They follow us wherever we go, whether into cities or across oceans. They are associated with the death and decay that humans create, and so rats are burdened with our own guilt-ridden symbolism."

In more ways than one, they are our victims, and in terms of design, it's easy to see why they infest our games. Rats are enemies of convenience. As De Geer explains: "The Skaven gave us a great opportunity to push as many enemies as we could on screen at the same time." Dedeine echoes this same thought: "It was the easy part. You can find an algorithm on the internet, and very quickly, you can see lots of rats." Both *Vermintide* and *A Plague Tale: Innocence* use rats and their associations to create uniquely visceral experiences, filling players with horror and disgust. But sometimes it can also be good to look out beyond the revulsion, beyond the plague, the blood, and the hunger, to appreciate the reality of a living creature, which has inspired so much in gaming. 🐀



^ Hazel Monforton, having written a thesis on collective violence in 20th-century literature, truly understands the victims of *Dishonored*.



^ Anders De Geer talks with a profound love of Warhammer Fantasy, evident in *Vermintide's* lore-inspired game design.



^ David Dedeine speaks about game design like alchemy, reflecting a creative process which seems to have relied more on instinct than exact science.



Wireframe's favourite animals

Sometimes you just have to list your favourite in-game animals. Sometimes you get carried away with dogs

We're full of animals this issue, so what better time than to talk up some of our favourite in-game beasts? None. There is no better time. Some might say this is a hasty decision taken

to fill a couple of pages we needed to fill, while others will realise that's only half the truth and, in fact, a big reason for doing this is because we enjoy both lists and animals. Believe what you will. Regardless, here it is: animals, lists, and words.



Yoshi Mario/Yoshi series – 1990-on

The long-suffering dinosaur with a taste for fruits, Yoshi has been helping Mario for longer than the nowhere-near-an-actual-plumber can remember. Shigeru Miyamoto had the idea for a dinosaur companion before the original *Super Mario Bros.*, but it wasn't until *Super Mario World* that the green one made his first appearance. Since then, Yoshi has popped up in dozens of titles, and brought plenty more smiles. And been a bit annoying in *Super Mario Kart*, let's be honest. Still, Yoshi is fantastically designed, and has cemented his place in our hearts forever.

Tom Nook

Animal Crossing series – 2001-on

We love Tom Nook not because he's nice – he's little more than a loan shark with a cheery demeanour – but he had such an impact on those jumping into Nintendo's life sim, it's hard not to love him. Welcome to *Animal Crossing*, it's cutesy and laid-back and everyone's friendly and – hey – you can pay me back at this rate, and if you don't I'll set the Raccoon Goons on you. Ha ha, I'm only joking – or am I? You've got to respect the kingpin of an idyllic hamlet, even if you are in debt to him.



Amaterasu

Ōkami – 2006

Not only is Amaterasu an elegant and beautiful wolf, but she also has a tiny little travelling companion doing her speaking for her in the shape of the artist Issun, and she has some pretty special painting skills. Oh, also she's the Sun Goddess, so there's that too. Ammy's adventures in *Ōkami* put players in the paws of one of gaming's greatest animals, tackling a *Zelda*-like adventure as good as anything else Nintendo itself produced. Chibiterasu upped the cuteness ante with the whole 'is a puppy' thing for sequel *Ōkamiden*, but our heart is also with the goddess of the sun.

Dog Fable II/III – 2008, 2010

Has there been a better in-game representation of dogs before or since Lionhead's action-RPG? We would argue no. Your faithful companion across each of the *Fable* sequels isn't just a useful tool in navigating the world, digging up treasures, and fighting off enemies – it's a dog that fundamentally behaves like a dog. That is to say, it is an adorable idiot. With that behaviour comes the inevitable bonding, and with that bonding comes the inevitable heartbreak – the choice to save everyone or just your pooch should have been a hard one. It was not. He's just too good a boy.





Cuccos

The Legend of Zelda series – 1992–on

Those chickens you see all over the *Zelda* games are, in fact, cuccos – and they're something we love. If this sounds a bit stilted and as though it's written under duress, please believe that couldn't be further from the truth. We love and respect the cuccos, and are in no way terrified of or threatened by them. We certainly don't wake up in cold sweats after suffering repeated night terrors focusing on the definitely-not-chicken terrors. No siree, not us. We just love cuccos, and in no way is this a result of wondering what would happen if we were to smack a few of them around with our sword. Nope, they're great. Not terrifying at all.

Chocobos

Final Fantasy II–on – 1988–on

Where Yoshi had plenty of other appearances, highlighting his plentiful skills, *Final Fantasy's* chocobos have largely only ever been used as transport tools. OK, there was some dungeon exploring, and even a bit of kart racing, but the vast majority of the time these kweh-ing giga-chickens have been used as horse proxies. So why the love? Well, they're just adorable, aren't they? Useful, obviously – and we shouldn't overlook that – but the chocobo dance from *Final Fantasy VII* is something forever in our hearts. They are arguably a bit more angular – and, frankly, *terrifying* in *FFXV* – but it doesn't dilute the love we have any.



Agro Shadow of the Colossus – 2005

Fumito Ueda's PS2 classic presents the player with a sparsely populated world, bereft of much in the way of... anything. It's in this sense of isolation that Agro, your trustworthy steed, comes into her own – she's not just a transport tool; she's your *companion*. It also helps that some of the game's finest battles require her direct involvement to render any success at all – sprinting alongside the snake-like 13th colossus and leaping from horse to giant is a moment none who've played the game will forget. Just as none will forget the sacrifice she makes to save your life at the game's end. It's OK, dry your eyes, it's just a sprain.

Wolfdog Shadow Dancer – 1989

No ninja would ever ask for help in fighting dangerous terrorist organisations – we've all seen too many films to know this as fact. A ninja would, however, bring along his wolfdog to maul gun-toting terrorists if *Shadow Dancer* is to be treated as a truthful historical document. Which it should be. Far from a cheat companion, your ninja-dog – called in to attack enemies and allow you a free shot in the *Shinobi* sequel – could actually end up hurt and neutralised if you weren't quick enough on the shuriken-draw. It's bizarre how a game from the late 1980s could offer such a strong attachment to an in-game animal, simply by imbuing it with a level of vulnerability.



Pomeranian Tokyo Jungle – 2012

SIE Japan Studio's peculiar animal-focused survival title has largely been lost in the ether over the years, but we will never forget it – in a big part because of that Pomeranian. One of the game's two opening options to control, the newly ownerless toy pooch needs a lot of care, isn't capable of anything special, and proves a challenge for all but the best players when it comes to actually making progress. So really, it's a lot like an actual Pomeranian would be in a real-world anarchic wildland of humanless Tokyo. And that, dear readers, makes for quite the unembellished underdog story.



Dog Half-Life 2 – 2004

OK, this is one dog too many. No more dogs. He's not even an animal; he's a robot. A wonderful, caring, friendly, helpful robot with plenty of dog-like characteristics, sure. But still a robot. One dog too many.

Revita



Interactive

Revita

The bullets fly in solo developer Benjamin 'BenStar' Kiefer's roguelite action platformer

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

It's the little details that caught our eye: the way enemy creatures shudder and wobble as they're struck by bullets; the little casings that eject from the hero's gun as he lets off another salvo.

This is *Revita*, a twin-stick roguelite platformer currently being made by Germany's Benjamin Kiefer – better known online as BenStar. Like a lot of indie games, *Revita* came to our attention while we were browsing through Twitter one evening, and we were immediately struck by the fluid, chunky nature of its pixel art. In still images, Kiefer's character designs are bold and colourful; in animated GIFs, they positively fizz with movement.

For Kiefer, work first began about 18 months ago, as a series of older ideas began to coalesce into what would become *Revita*: a fast-paced action game where, in true roguelite fashion, you fight your way through enemy-infested levels that change each time you play. "Before I worked on *Revita*, I was working on fan games and the occasional game jam," Kiefer tells us. "When I was a lot younger, I always wanted to make my own video games, so at some point, I found

out about GameMaker, a tool that made game development relatively approachable, and I've stuck with it since."

In the years since YoYo Games' GameMaker first emerged in 1999, it's gained a reputation as a low-cost and easy-to-use platform for making 2D games; by using GameMaker Studio as its engine, *Revita*, therefore, joins such indie

darlings as *Hotline Miami*, *Undertale*, *Downwell*, and *Spelunky*. "GameMaker is extremely useful when it comes to making 2D games," Kiefer

enthuses, "and Aseprite might be the best pixel art software out there when it comes to pixel art and animation."

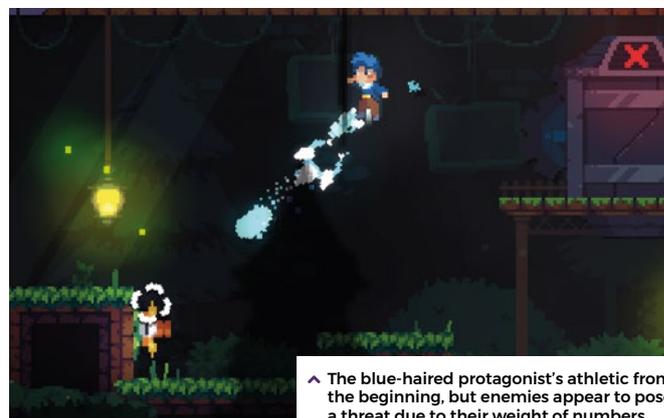
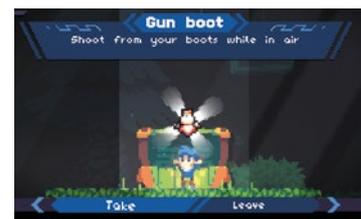
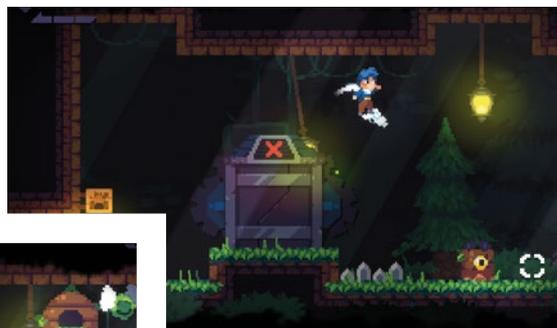
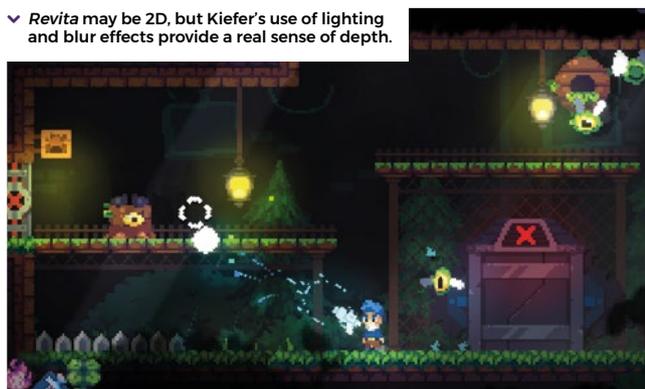
"GameMaker made game development relatively approachable, and I've stuck with it since"

MEMORY TEST

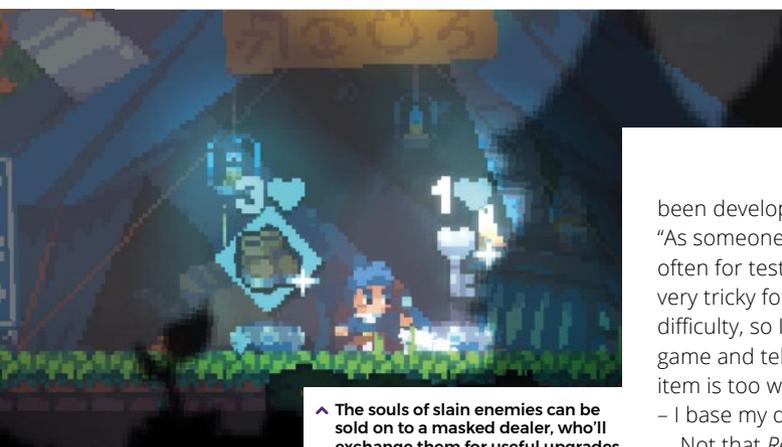
Kiefer won't be drawn on the specifics of *Revita*'s story, but then, that's because much of the game is presented to the player as an unfolding mystery. "I don't want to say too much on that note, since I think it should be up to players to discover," he says. "But what I can say is that you're playing a nameless child who has lost all their memories, with their only point of reference being a giant, mysterious clock tower.



▼ *Revita* may be 2D, but Kiefer's use of lighting and blur effects provide a real sense of depth.



▲ The blue-haired protagonist's athletic form from the beginning, but enemies appear to pose a threat due to their weight of numbers.



▲ The souls of slain enemies can be sold on to a masked dealer, who'll exchange them for useful upgrades.

So the goal is to climb the tower and figure out who, what, and where you are, and why you are in the situation you are in."

BALANCING ACT

What it all boils down to, though, is a hectic loop of blasting, collecting the souls from downed enemies, and using them to improve your weapons in order to progress higher up the clock tower; Kiefer cites the likes of *Flinthook* and *Enter the Gungeon* as inspiration for its roguelike mayhem. The process of actually creating *Revita*'s procedurally generated levels has, however, proved to be one of the aspects that Kiefer's worked hard to figure out. "In *Revita*, each combat room you enter is completely [procedurally] generated," he says, "which gets really tricky if you want to have fun and exciting rooms that look like they could have been designed by hand. So that must have been the trickiest challenge so far."

Balancing the game has also provided a challenge, Kiefer tells us – partly because he's

been developing *Revita* largely by himself. "As someone who's playing the game way too often for testing purposes," he says, "it gets very tricky for me to judge when it comes to the difficulty, so I usually have friends testing the game and telling me what they think. Maybe an item is too weak, maybe an enemy is too strong – I base my decisions on that [feedback]."

Not that *Revita*'s entirely the product of one person; while Kiefer is handling the design and pixel art side of things, he's also found a composer and sound designer to help him create the game's sonic landscape. "They're both fantastic, and *Revita* wouldn't be what it is today without their help," he tells us. "When it comes to finding the right people to work with, social media is a big help. There are lots and lots of artists and programmers on there that are looking for work. So you've just got to keep your eyes open and get in contact with people."

Fittingly enough, given that we first encountered *Revita* when someone tweeted one of its animated GIFs into our timeline, Kiefer's keen to point out just how effective Twitter can be for indie developers. "I probably wouldn't be where I am with the game today if it wasn't for Twitter and the countless connections I made there," Kiefer says. "I heavily recommend, especially new developers, to get their stuff out there. Social media is essentially free marketing when it comes to financial means. If you have the time to use social media for marketing purposes, do it. It's more or less the easiest and cheapest way to market yourself to an audience." 🐦

FUELLED UP

If you want a taste of what Kiefer's making right now, then head over to his itch.io page (benstar.itch.io), where you'll find a selection of his smaller games. Our pick of the bunch? Definitely *.FUEL*, a cracking little shooter tailor-made for playing with a mouse: it's like Atari's venerable *Asteroids*, except your ship loyally follows your mouse pointer as you roam the screen, blasting encroaching aliens. An ever-decreasing fuel bar (topped up by picking up icons from the husks of your enemies), a combo meter, and a huge scoreboard in the background makes for an addictive quick fix.



▲ Kiefer's *.FUEL* is an addictive top-down blast in the mould of Atari's *Asteroids*.

Become an internet billionaire!



STEVE MCNEIL

People watch Steve play video games on the internet. Sometimes they give him money. He's as surprised as you are.

“Many of you might be under the impression it’s just teenagers with haircuts screaming as they kill digital humans”

N

early two years ago, I found myself with some free time and decided to dip my toe into the world of streaming. I'll be honest, I didn't really understand what

it was, or why it was a thing. Based on how it's portrayed in the media, many of you might be under the impression it's just teenagers with haircuts screaming as they kill digital humans, but streaming's actually not that awful and is a very nice thing to do. Honestly.

While I'm by no means Mr Big Cheese on the streaming platform Twitch, I have learned a thing or two over the last couple of years, and I will now share those things with you, should you wish to have a dabble yourself...

1. Make it look not rubbish!

If you're streaming in 240p from a dark room with a webcam you bought in 2003 and the headphone/mic combo you got free with your last phone, people will be turned off. If you're serious about trying to stream, treat yourself to a decent mic and webcam, and a PC that can handle streaming. You don't need to spend a fortune on any of these, but you do need something fit for purpose. OBS (and its Streamlabs variation) offer free software that can have things looking professional with just a little bit of effort.

2. Don't be unreliable!

Set a streaming schedule that fits around your other commitments and stick to it. If people never know when you're going to be live, they will not find you. Obviously.

3. Talk to the lovely people!

People have tuned in to watch you, so talk to them. If you run out of steam, ask the viewers questions to get the ball rolling. If someone writes a message in chat, reply to it! If someone new turns up, make them feel welcome. If you become an affiliate and are able to accept donations and subscriptions, express gratitude.

Don't do what one guy I'm aware of did, and complain that a donation was excessive, unless you want everyone to think you're a dick.

4. Community is everything!

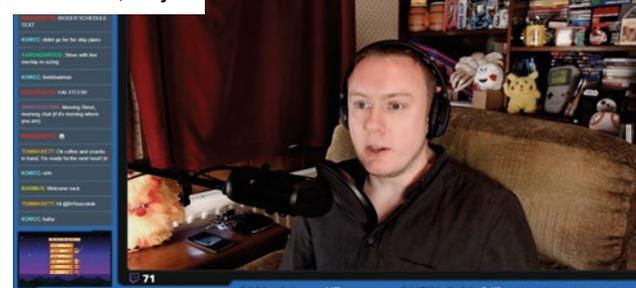
The main thing I didn't understand before I began streaming is that for the people who engage with what you do, the community aspect is huge. So, don't just interact with them during streams, interact with them outside of streams too. Use social media to engage with viewers, consider setting up a Discord server which gives people somewhere to hang out when you're not live, and engage in other communities (but never be pushy). Host other streamers on your channel when you're off-air, and send your viewers over to watch other streamers using Twitch's 'raid' system to further increase your reach. Of course, if contact with other humans makes you feel angry and/or sick, streaming might not be for you.

5. Make sure you don't hate doing it!

Fundamentally, you're playing video games. That's meant to be fun. Whilst it's helpful to become known for certain games, and you may want to choose something that isn't saturated by hundreds of rival streams, fundamentally you should pick games you enjoy – your pleasure is infectious, and no one wants to watch you be sad. Or, if they do, you probably don't want those guys to become your regulars, because they're nasty.

Follow all these tips and you will be able to retire from your vast internet wealth by Christmas. T&Cs apply. 🙄

▼ Streaming: not that awful at all, really.



Toolbox

The art, theory, and production of video games

28. CityCraft

When designing a game city, abstraction's your friend

30. Team-building

Five tips for forming a top-notch dev team

32. Remote working

Why distributed development's the future of making games

36. Live coding

Using algorithms to create mesmerising visuals

40. Source Code

Code a Duck Hunt-style shooting gallery

42. Directory

Go GUI over a free user interface course



▲ Discover the world of live coding, and how you can use it to create your own trippy imagery, on page 36.



▼ Development teams don't all have to be under one roof to make great games. Find out why on page 32.

CityCraft: abstracting virtual cities

It's impossible to recreate all the detail of a real city in a game, so here's a brief guide to abstracting your urban environment



AUTHOR
KONSTANTINOS DIMOPOULOS

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

Even gaming's greatest cities pale in size, complexity, and density of detail when compared to a real-world city. Victorian London, the first modern metropolis, a city that in its time surpassed every historical civic limit, was represented by less than 300 city blocks in *Assassin's Creed Syndicate*; driving around *GTA V*'s supposedly vast Los Santos barely takes more than 30 minutes. Still, these open worlds feel highly convincing. Symbolic representations, clever design solutions, obstructed views, and, above all, the careful abstraction and stylisation of urban space, give players the illusion of much larger cities.

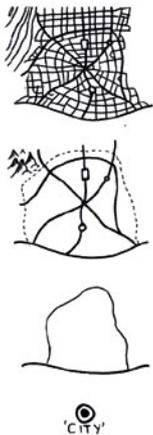
It's the ways we choose to present our game cities, their fidelity, and their aesthetic style, that define how much abstraction we'll need to apply. In most cases, though, the logic that cartographers use when drawing maps will provide a good initial approach. In **Figure 1**, I've sketched an example of how a city can be represented as we zoom out and reduce the scale of our map. The increasing degrees of abstraction lead from a relatively detailed map, where all major roads and contour lines are visible, to an outline with a few highways and stylised mountains, to a shape of the city's

outline, and finally to a simple point with a label. To see such a process in real time, simply visit Google Maps and zoom in and out.

THE ART OF ABSTRACTION

Cartographers don't carelessly remove spatial information when they're abstracting a real location, and similarly, designers of virtual cities need to be careful with what they choose to show or ignore. We have to make sure that the crucial, defining elements of our cities survive, and thus preserve their intended shape, structure, and character. I would, for example, always make sure to feature the 'L' curve of the River Thames in any version of London, and include at least a simple iteration of Central Park for a virtual New York.

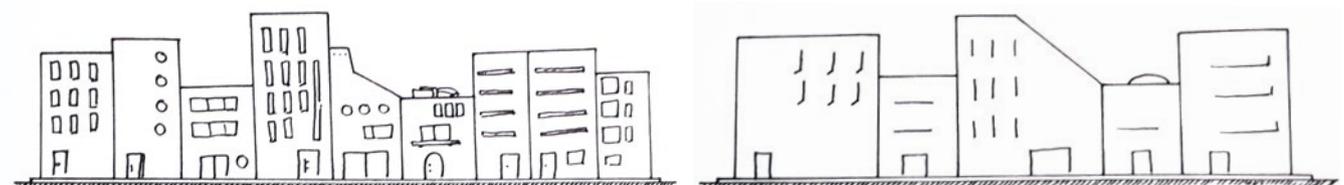
Besides the removal of information, civic generalisation must also show off what remains. It's vital to subtly reorganise, and provide the virtual space with clarity in order to frame its characteristics and hide its omissions as best as you can. Spatial elements may have to be moved around or even enlarged. Roads can be curved to camouflage their short length. Intricate shapes will have to be simplified; smaller city blocks, or linear features such as parallel train lines, may have to be aggregated



▲ **Figure 1:** The cartographic generalisation of a simple city in four stages.



▼ **Figure 2:** Sometimes abstraction can preserve silhouettes and volumes while altering the architectural characteristics of an area.





▲ Even impressive, large, and detailed virtual places like the bustling New Bordeaux from *Mafia III* are heavily abstracted versions of urbanism.

into areas; whole sub-areas may be represented with emphatic points and landmarks, whereas repeated elements can be reduced in number while retaining their character. A grid of 15×15 blocks could, in most cases, be represented by an 11×11 grid, and three adjacent parks could easily be fused into one. The city itself can also be simplified – for example, by reducing its 20 or so types of road to three or four, and assigning all its citizens to one of three classes.

In **Figure 2**, I applied this generalising approach to the façades of a city block. Initially, the number of buildings was brought down from nine to five, and the block itself was slightly reduced in size. Then each building was itself simplified, and enlarged to preserve the overall volumes of the built environment, and to roughly mimic the original silhouette. If architectural details were important to my game's setting, I could have instead chosen to focus more on preserving the block's character and not its size while still saving on assets.

In this case, I could simply pick five of the original buildings, not tamper much with their appearance, and thus present players with smaller but more detailed city blocks. I could have also been slightly more subtle when enlarging each building, and maybe only added a barely perceivable 10 to 15 percent to the length of each frontage, but the truth remains that such decisions are always an exercise in balancing contradicting aesthetic and functional needs. Capturing the essence of a place and deciding what to preserve and what to abstract will always sit at the core of this process, and will always require a thorough understanding of each place.

Abstracting, merging, and simplifying, especially when combined with well-crafted modular assets or with the procedural generation of environmental aspects, can allow

for the creation of impressive yet relatively low-cost virtual cities. Obviously, any stylisation will have to be applied to the entire city map, and not just one block or a road. Not all areas have to be equally abstracted, however; the *Assassin's Creed* series tends to recreate important landmarks and their surrounding areas more faithfully, and in greater detail, than less spectacular districts, which it often treats as spatial filler.

URBAN SCALE

Figure 3 offers an example of how abstraction could work on the scale of a neighbourhood. In this particular case, blocks and parks were merged, and shapes were simplified, while still keeping the area's defining spatial aspects in place. The major avenues, the north-western parks, and the overall shapes

of blocks and roads have all been retained, even as the area's complexity was drastically reduced.

Simplifying a whole town can follow this exact logic, though the process shouldn't just involve the reduction of each district's complexity, but also the number of districts. A modern metropolis might have tens of entertainment hubs, countless suburbs, and several industrial districts, but selecting only the right ones to show can be enough to convey the richness of its fabric, and suggest its size.

Entire locations can also be condensed into a single edifice or landmark. *Skyrim's* Whiterun, for example, features three distinct districts, but the upper one consists of a single building: a huge keep that symbolises the world's high-born society. 🌐

Degrees of Fidelity

To sustain its players' suspension of disbelief, a photorealistic setting requires a greater density of detail than a cel-shaded one. Even though both approaches could contain an equal number of buildings, the level of simplification of each individual building will be necessarily different. The cel-shaded world could feature low-poly buildings with simple shapes and stylised windows, whereas in a photorealistic world, even the doorbells need to look right.

▼ **Figure 3:** Generalising a neighbourhood by fusing blocks and parks. A simple yet usually effective solution.



Five rules for team-building

Building games is hard; building great games is even harder. Here are Reid's tips for hiring and forming a dev team



AUTHOR
REID SCHNEIDER

Reid is the producer of *Splinter Cell*, *Battlefield Vietnam*, *Army of Two*, *Batman: Arkham Origins*, and *Batman: Arkham Knight*. Follow him on Twitter: [@rws360](#)

■ was recently sharing an elevator with an old friend and colleague, Patrice Désilets – creator of *Assassin's Creed* and Ubisoft's *Prince of Persia*. Like us at Typhoon Studios, he split off from the corporate world, and started a company called Panache Digital Games. They're currently building *Ancestors* (see issue 18). As we talked in the elevator, Patrice said something which has stuck with me: "Finishing any game is always a minor miracle."

At first I kind of laughed, but then as I thought more about it, the more I agreed with him. It takes a great team to finish a game, but great teams don't just happen overnight; they happen through people working together, over long periods, to achieve a common goal. I've made some great hires over the years, and also made some disastrous ones. Hopefully, the tips below will help you in your team-building journey.

1. How does this person treat others?

This one is highly personal to me as I have an allergic reaction to arrogance. In an interview

situation, people will, for the most part, be on their best behaviour, and try to tell you what you want to hear. I like to refer to this as 'meeting their representative', since it's not necessarily who the person really is. I've found that the best way to really get to know someone is to see them in a more social situation. For example, if you go to lunch or a bar, how do they treat the staff? Are they the same person, or are they a jerk?

About a year ago at Typhoon, we were looking for some support in a very specific domain. At the time, my wife Mandy (who's been a rock star in supporting the company and still teaches yoga classes here), would sometimes sit at the front desk and greet visitors. Anyway, our potential employee walks in, and he happened to know one of our other co-founders quite well. To say that he was rude to my wife – who he assumed was a receptionist – would be an understatement. A few moments later, I walked over while he was talking with other staff, and I said, "I see you met my wife, Mandy." A look of horror came over his face, and with that, let's just say he didn't get the job.

2. Beware the 'law-enforcers'

Early on in the evolution of WB Games Montréal, we needed to hire a producer for a new game we were developing – *Batman: Arkham City* on the Nintendo Wii U. We had received a recommendation from another employee who said he knew someone he thought was good, but warned he could be a bit "militaristic" on occasions. We interviewed him, thought he could be valuable, and the team liked him. Something felt a bit off, but we couldn't directly identify it.



▲ Based in Montréal, the team at Typhoon Studios has grown rapidly since its founding in 2017.



Ultimately, we hired him, and gave him a shot. It's fair to say the appointment didn't go well. After he started, the new producer tried to turn the team into a "combat platoon", with a chain of command that could never be questioned. This ran counter to the values of the studio, and the team quickly began to resent him. He didn't last, and we had to let him go. In short, trust your gut, and beware of anyone more interested in war re-enactments than team solidarity and development.

3. Hire for diversity

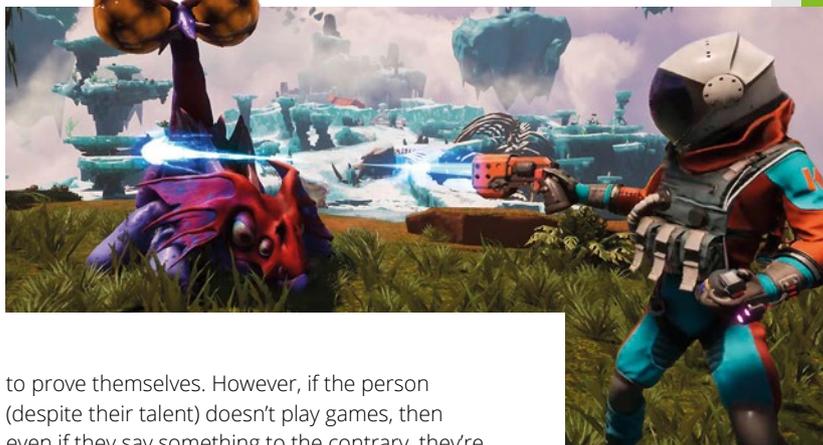
Having a diverse team is critical for game development. If you're lucky, your game will be played by thousands, and potentially millions of people. And guaranteed, these people *will not* think like you. Having a diverse team isn't simply a checkbox to tick. It's a benefit for your company, as it will bring different perspectives to your project, and help you identify 'blind spots' which you might otherwise miss.

Diversity makes good business sense. Diversity will help you create successful projects and sell more games. In full transparency, we've been OK at this at Typhoon, but we still have a long way to go. The good news is that more and more women are entering game development. This is a win for everyone.

4. Hire people who enjoy what you make

This one seemingly goes without saying, but it's worth mentioning. Pretty much every interview I do with a candidate, I ask them what they're currently playing. They don't need to be the most hardcore player, but they do need to have an interest in games. This is particularly important in cities like Montréal, Vancouver, Los Angeles, and London, where there are high concentrations of both game developers and movie VFX companies. With consoles and PCs becoming more powerful, and films featuring more and more CG, the crossover between these two workforces is increasing.

When you're interviewing, you may come across an artist, animator, or programmer who has a lot of experience in film but not games. If the candidate is genuinely interested in your project and games in general, then take the risk and make the hire; they will be motivated



▲ The team at Typhoon are currently working on their debut outing, *Journey to the Savage Planet*.

to prove themselves. However, if the person (despite their talent) doesn't play games, then even if they say something to the contrary, they're probably just looking for a bridge between jobs.

5. Be wary of demotions

In Typhoon's early days, we were fortunate to have people who were excited (or crazy) enough to want to join us. Our founding team and first employee consisted of a creative director, technical director, producer, and head of art.

We had two individuals who were really excited about joining us, who in their previous lives had been creative director and art director respectively. Both were really good at their old jobs, but made the decision to 'step down' to the roles of design director and principal artist.

Initially, it was great; everyone was getting along well and doing their part. Unfortunately, game development is rarely all rainbows and sunshine. The challenge we had was that the pair were very used to getting their way in terms of creative choices, and they found it difficult when this didn't happen. Now, I've seen cases where this situation does work, so this isn't a hard and fast rule, but rather something to keep in mind: what it really comes down to is a team member's personality.

Like game development, team-building is hard, and it's an iterative process. Where you start out is almost certainly not where you finish. I was listening to a podcast a few months ago, and there was a start-up founder talking about his approach to hiring. The main takeaway was to always hire 'missionaries' rather than 'mercenaries'. What this boils down to is, hire people who are intrinsically motivated by what your team is building, and for whom the pay cheque (while important) isn't the number one driver. We try to live by this rule every day at Typhoon, and while challenging sometimes, it's done well for us. ☺

“Diversity will help you create successful projects and sell more games”

▼ The Nintendo Wii U port of *Batman: Arkham City* proved to be a solid one, despite a few producer problems behind the scenes.



Distributed game development: the future?

Forget about games made under one roof: it's now possible for teams to work remotely, Kieran writes



AUTHOR
KIERAN D'ARCHAMBAUD

Kieran has held positions including lead programmer and technical director for studios including Rare and Evolution. He now contracts remotely for clients such as Ultimatum Games. koding.co.uk

The games industry is still comparatively young. Looking at how development has changed over the last 30 years or so, we've constantly changed the techniques and approaches we take, whether we're looking for a competitive advantage over rival developers, or hoping to create a unique hook in our games.

One thing that has changed massively is the number of people that collaborate to make a game. In the industry's early years, games

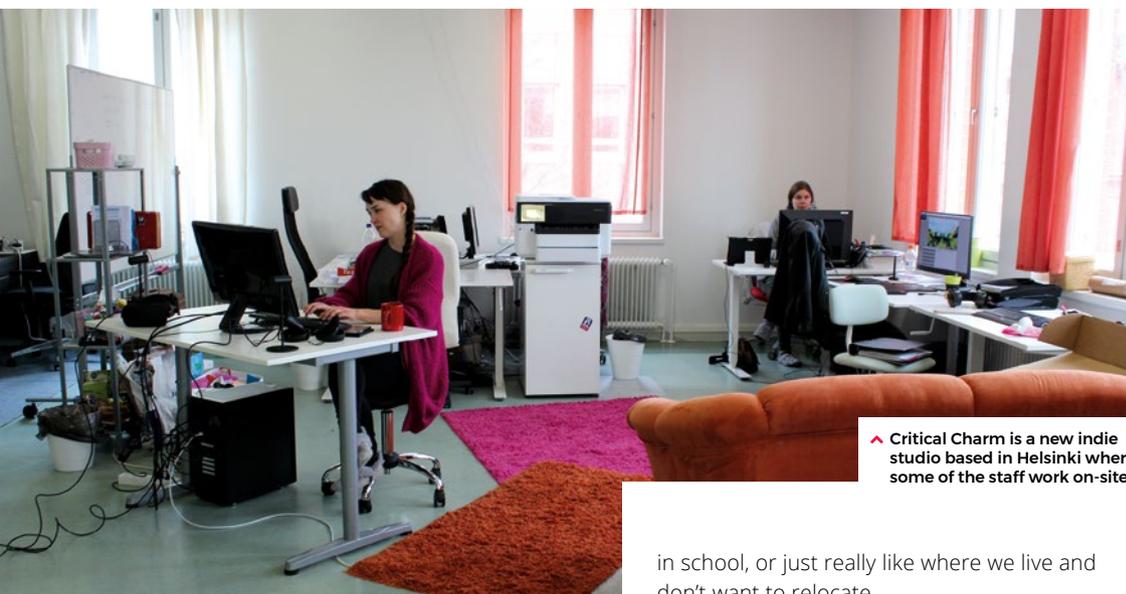
were commonly made by individuals who wrote the code, created the graphics, sound, and everything else themselves. Then people started working in small teams, breaking down the work between different disciplines. As time marched on and the scope of games expanded, so did the teams. We started making games with tens of people, then hundreds. Now, in some cases, over a thousand people can be involved in the development of a single triple-A title.

While development teams have expanded, in most cases, those teams operate under a single roof. The reason for this is simple: ease of communication. In any creative industry, one of the hardest things to maintain as team sizes increase is communication, and in industries where the work of one person can impact a large percentage of the rest of the workforce, this can be difficult to manage just to ensure that people can do their work rather than spending most of their time sending and reading emails and sitting in endless meetings. Project managers are unsung heroes in this regard, because it's their job to ensure that communication happens in a way that means games actually get made.

In some cases, games are made collaboratively between two or more studios,

✓ Shane McCafferty at ten years old, doing solo game development from home on his Spectrum +3 in 1986.





▲ Critical Charm is a new indie studio based in Helsinki where some of the staff work on-site.

and as such, communication can suffer between these different locations. For this reason, the work done at each studio is normally broken down to reduce the communication required. One studio may work independently on a single-player mode, for instance, so the people collaborating on that part of the game are still in the same building.

This doesn't solve the problem, but it reduces the negative impact substantially. It can, however, have a negative impact on the finished product, where there are noticeable differences in aspects of the game that were worked on in different locations.

FOLLOWING THE WORK

While the games industry has grown at a great rate, it's still relatively small. Employees often have to move, sometimes to other countries, to follow the work they want to pursue. Also, as an artefact of the industry being fairly new and having grown so fast, companies are still figuring out how to create games while still ensuring long-term stability for their workers. It's not uncommon to hear of companies having to let some of their workforce go once a project's complete. The industry's generally good at making new opportunities available for those who do lose their jobs, but again, this often requires relocation for workers.

For some, this is exciting, and they use it as an opportunity to travel around the world, meet new people, and face new challenges. This is, however, a small percentage of the workers in the industry; many of us have families, with kids

in school, or just really like where we live and don't want to relocate.

So what are the options if you don't want to move? You could leave the industry – something that lots of people do at some point in their careers. That's a real shame, and we lose some great talent when people make that choice. You could set up a small indie studio. You may just want to write code or draw pictures. What other options are there?

BETTER COMMUNICATION

Suppose that we could fix, or at least greatly improve, the issues we face with communication. Could we allow workers to work wherever they want, and still get the product made to the same quality, in the same time, with the same budget? I think we can. On a small scale at Ultimatum Games, we've built systems and procedures which allow us to work collaboratively as a distributed team. As a result, I feel just as productive (if not more so) than at any other time in my career.

As previously discussed, communication is the biggest barrier to distributed game development. The first step in solving this is to identify when communication needs to happen, and then what method should be used for that communication.

There will be cases where a worker requires information so that they can complete their current task, and they choose to ask someone else to get the answer. When this happens, the person who replies will be interrupted from their work, causing them to switch context, which often has a much greater cost than the time to have the conversation. You have to add the time it takes for that person to get back into what they were previously doing. ➔

FURTHER READING

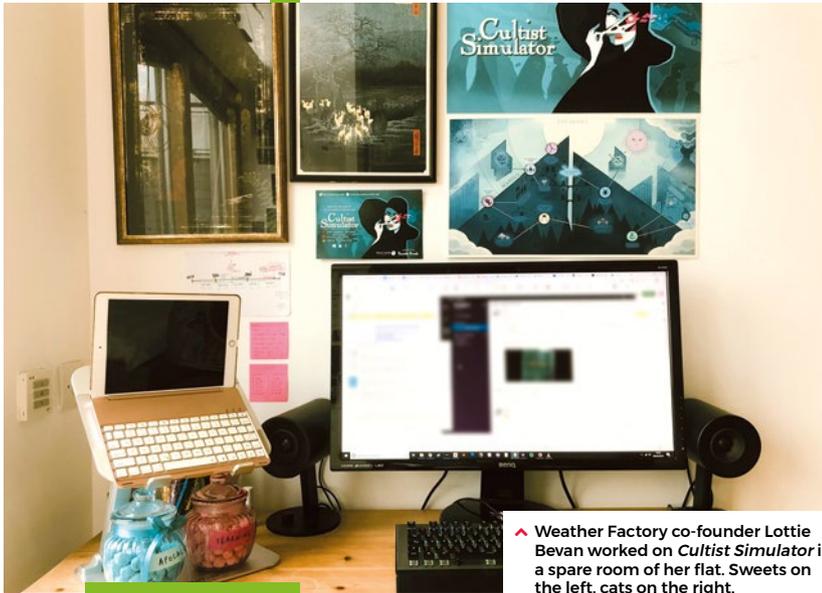
The book *REMOTE: Office Not Required*, written by the creators of Basecamp, is an essential read if you're considering remote working, both for the worker and also for managers who want to understand how you should manage a distributed workforce. Many industries are already embracing remote working and the benefits that it offers, but the games industry is currently lagging behind. This book acts as a great primer to allow you to catch up with the benefits that others are already enjoying.



▲ A practical guide to remote working, written by the very successful distributed company 37signals (now Basecamp).

Toolbox

Distributed game development: the future?



▲ Weather Factory co-founder Lottie Bevan worked on *Cultist Simulator* in a spare room of her flat. Sweets on the left, cats on the right.

OUR TOOLS

Most game developers will use a continuous integration server such as Jenkins, TeamCity, CruiseControl, and so on. The role of that server is to monitor for additions or changes to the project and, when detected, build it for the various platforms being targeted. This is done automatically without human intervention. This means that a short time after submitting a change to the game, there will be a playable build of it provided automatically for all the platforms you want to release it on. Some expand on this so once the game has been built, tests are run automatically to verify correct functioning of the game. If there are any issues with building or testing the game, the developer will be informed so that the issue detected can be resolved quickly.

A further extension we've added at Ultimatum Games is a bot, which is a small piece of software we've written to watch for requests or questions in our asynchronous communication system. It's able to respond back to developers, providing information and answers to questions. It's also able to communicate with our continuous integration server where it can fetch the results of work previously done – for example, the last built version of the game for a specific platform. It can also request work to be done by the continuous integration server.

An example of this is that a user can ask the bot to request a light bake of a specific level in our game. The bot will take this request and ask our continuous integration server to start that work. The bot will then communicate with the requesting user to inform them that their requested job is in progress. When the job has finished, the bot will then inform the user that the work is done. In this example, the user may have made a change to a level in the game, requested it to be light baked, then went on to do some different work. Some time later (light bakes can take a long time), the work will have been done for them and there will now be a beautifully lit new level in the game. While this has been going on, the developer has been able to get on with other work without their own personal computer being tied up doing the light bake itself.

REMOTE CONTROL

"I've been working remotely for 15 months now," D'Archambaud says. "I spend some days on-site at companies, but the majority are spent in my attic. I set the hours I work, and these vary from day to day. Often, I'll choose to do extra hours early in the week so that I start my weekend earlier. Some days I'll end my day abruptly if the sun's shining and I want to head outside. I work when I'll be most productive."

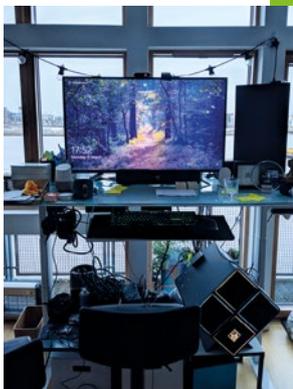
Don't underestimate how long this can take! In some cases, it could take a while to get back into the same mental position so that person is able to continue with their work.

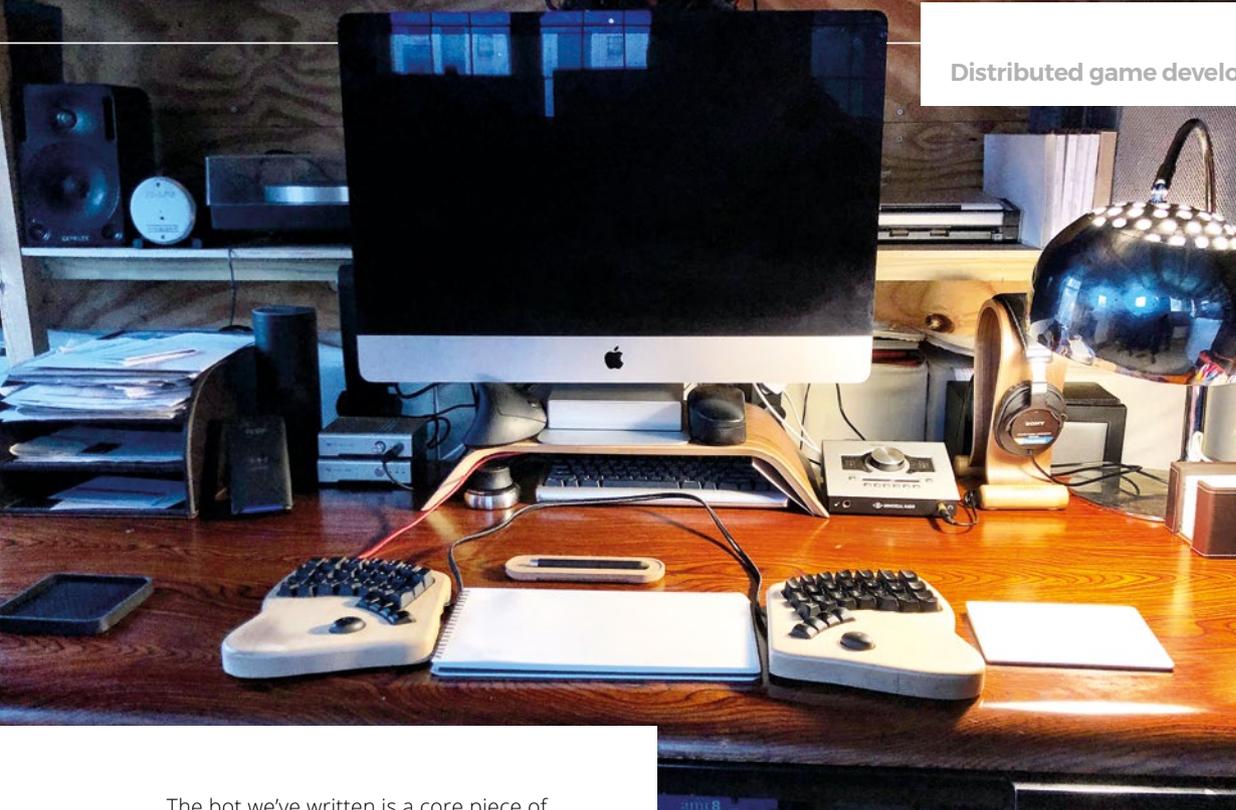
Whenever someone interrupts someone else, the first question should be, 'Why?'. Could you make it so that in the future, someone with a similar question can get the answer they need without interrupting someone else? In some cases, the answer is no, but interrupting someone else should be the last resort. This is where asynchronous communication – using Slack, Microsoft Teams, and so on – comes in handy, since the person responding can do so at a time that's good for them. Try not to use direct messages; open up the question to the team. You may get a quicker response. You may also get a better response if the question sparks a discussion among team members. The answer may help someone else who reads it from needing to ask the same question in the future. Finally, ensure that work is planned, so that if someone finds that they need help, they won't be blocked while they wait for an answer.

When a question is asked, ask yourself whether that question should require a person to be at the other end to answer it. Could they have searched for the answer – perhaps someone else has previously asked the same question in the communication tool of choice. Could a piece of software give them that information instead?

"A user can ask the bot to request a light bake of a specific level in our game"

▼ Weather Factory is a three-person studio working from home. Two work with this stunning view in the background.





◀ **Shahid Kamal Ahmad** is the MD of **Ultimatum Games**, which is a completely distributed games developer. He commands the company from his shed.

The bot we've written is a core piece of technology, and we are constantly giving it new abilities. For instance, when we want to push a build of the game to a store front, we just ask the bot to do it. Importantly, the bot uses exactly the same communication mechanism as we use for everything else, and like the developers, it will communicate asynchronously which allows the developer to have 'conversations' with it at a time which fits in with them. This tool is one of the key places that we turn the question of 'Why did that person need to interrupt someone?' into a solution, which ensures that it doesn't happen again.

Following these rules – identifying 'bad' communication and providing processes and tools to cover those situations – has allowed us to become progressively more efficient. We don't have long meetings. We do have regular video calls, but they aren't drawn out. We generally all know where we are up to, given our open communication on Slack and by using our suite of tools. Those times are more for 'face time'. It's important to know who you're working with, to build relationships, to talk.

REMOTE WORKING BENEFITS

What benefits does remote working bring? It's easy to spot the ones for the workers. You get to work from where you want, and assuming your asynchronous communication procedures are correctly implemented, this can also be when you want. You can be located anywhere in the world – potentially, a different place every day.

What about for the business? You can get the very best people regardless of whether they want to relocate. They're choosing the time and place where they can do their very best work,

somewhere where they won't get interrupted. You will most likely find that you get increased productivity from them.

Admittedly, remote working isn't for everyone. Some people want to have regular social contact with the people they work with. You can take these procedures and implement them in combination with the classic single or multi-site office, however. Give your workers the choice. If they want to be in an office, let them. If you can only get the person you want on your team by allowing them to work from a remote location, let them. If you have someone who needs to relocate for personal reasons and can no longer commute into the office, give them the option of working remotely.

All these options can work, but all of them will fall apart if you don't have your communication procedures correctly set up – more so if some of your team are working in the same location. They need to use the same communication systems, otherwise, there will become a 'them and us' situation, with meetings being held on-site with remote parties excluded.

SCALING UP

The remaining question is, can this scale to much larger teams? Could a thousand people work on a project completely distributed with the same efficiency as they do in current triple-A game development? That we cannot say for sure right now. But by following the process to identify and fix bad communication, I think it could work, and would allow the industry to scale to new heights. ☺

TOOLS OF THE TRADE

"The software that I use for remote collaboration is as follows," D'Archambaud says. "Slack for asynchronous communication with both people and software bots that we write. Slack video calling for face-to-face meetings and discussions (previously used Discord). Bespoke Slack bot that communicates with our bespoke Jenkins server farm. Google Drive File Stream for Jenkins jobs to easily push data to Google Drive for the team to access."

▼ **Shane McCafferty** at 42 years old, doing distributed game development from the comfort of his sofa.



Live coding and ray marching

Coding shaders can also be a performance. Char shows you how to write a basic ray marching renderer in a live environment



AUTHOR
CHAR STILES

Char Stiles is a researcher and computational-artist-in-residence at the Frank-Ratchye STUDIO for Creative Inquiry at Carnegie Mellon University. CharStiles.com

What's at the intersection of raving and low-level computing? Algoraves! At an algorave, the music and visuals are all algorithmically generated and often live-coded, which means during a performance, a programmer will write the code that produces the music or visuals as well as projecting their code for the audience to enjoy. In contrast to static programming, live coding has the ephemeral, collaborative, and reactive nature of live performance. Because there's instantaneous compilation, it becomes

“At an algorave, the music and visuals are all algorithmically generated and often live-coded”

a tool that you can use intuitively – it can lead to surprising results, because the development environment is fluid and spontaneous itself.

This article will walk you through a technique of programming interesting visuals that lend themselves to live coding shaders. The rendering method is called ray marching. This is very close to ray tracing; the difference is that ray marching uses a contained and mathematical way to describe a 3D scene, while ray tracing is a more general technique that lots of rendering platforms use, like Nvidia's RTX. They're both physically based, in that you follow a ray the same way photons of light bounce around in the real world.

All code in this walkthrough is written in GLSL (OpenGL Shading Language), which is a C-syntax language executed on the GPU. Specifically, we're coding a fragment shader. You can apply these concepts to any language that can spit out an array of colours. For example, you could write it in Python with no renderer, and then hand a 2D array of numbers corresponding to colours to your kid cousin and tell them to colour in a grid, with each number

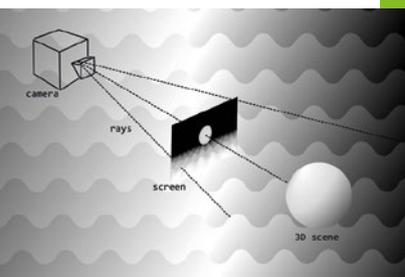
corresponding to the numeric values. If you do that, you can tell your kid cousin they're a rendering machine, and congratulate them for their efforts.

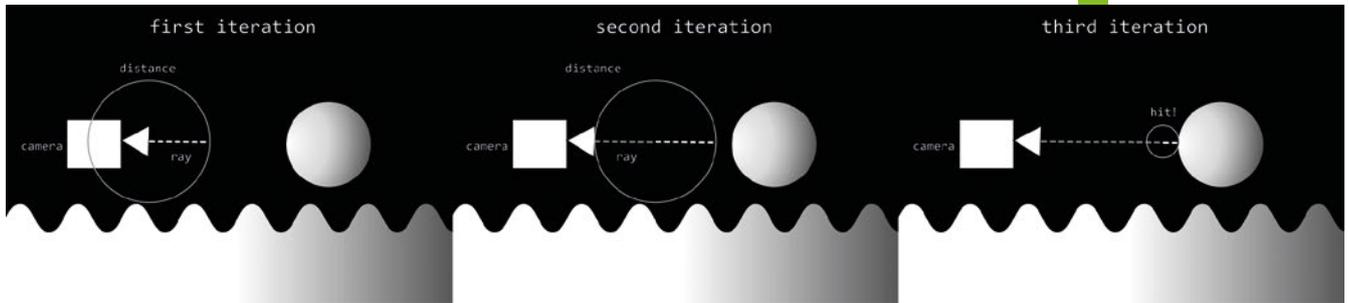
Of all the different ways to render a ray-marched image, I prefer live coding, and the best live coding tool I've come across is KodeLife by Hexler. Whenever I teach live coding GLSL workshops, the participants are advised to download KodeLife, because there's no setup for desktop, it just works out of the box on Mac, Windows, Linux, and Raspberry Pi.

As **Figure 1** illustrates, ray marching consists of:

- Creating a 3D scene out of signed distance functions (I'll go over what those are in a moment)

▼ **Figure 1:** In ray marching, a ray is cast from every pixel, and pixels are coloured according to what they hit. The camera is there to determine the angle at which the rays are cast.





▲ Figure 2: Sphere tracing is how the ray travels using SDF functions. In the first two iterations, the floor is the closest object, but it doesn't hit the floor - it hits the sphere on the third iteration.

- Casting a ray from every pixel on your screen into that 3D scene
- Colouring that pixel the colour of whatever the ray hits (or doesn't hit)

Now, how we do this is where it gets interesting. We're going to be using a specific version of ray marching called sphere tracing.

We describe the 3D scene using SDF functions - SDF stands for signed distance function, and for these purposes, they're just distance functions. This is cool, because if you describe an object by not defining its position in space but rather its distance to a point, you can use mathematical equations to describe shapes.

For example, this is a sphere:

```
float s = length(position)-radius;
```

This is a wavy floor:

```
float f = position.y + sin(position.x) + cos(position.z);
```

So we can tell the maximum distance a ray can travel without hitting any object, but since SDFs are directionless, the maximum distance is a sphere, as you can see in **Figure 2**. This is why it's called sphere tracing. Let's roll our sleeves up and think about how this is going to be implemented. First, some things to know about the limitations and the features of the shader.

A fragment shader is executed simultaneously per pixel, which in this case is the whole screen. This is useful, because it's how this type of program uses the GPU's parallelism; what's less helpful is that you can't easily look at the pixel's neighbours or have mutable variables across all pixels per frame. The input we assume we have is the UV passed in from the vertex shader. A UV is simply two floating-point numbers that describe a pixel's position on the screen, so the bottom left-hand corner is (0,0) the top right is (1,1), and the middle is (0.5,0.5). With UV and

screen resolution as your inputs, your job as the programmer is to output the colour of the pixel at that UV position.

Let's take a look at this idea implemented in GLSL. All of the following code was written with OpenGL Legacy context, and was tested in OpenGL ES 2.0 context as well.

```
#ifdef GL_ES
precision highp float;
#endif

// These are defined by KodeLife, or whatever
// environment you are using.
uniform float time;
```

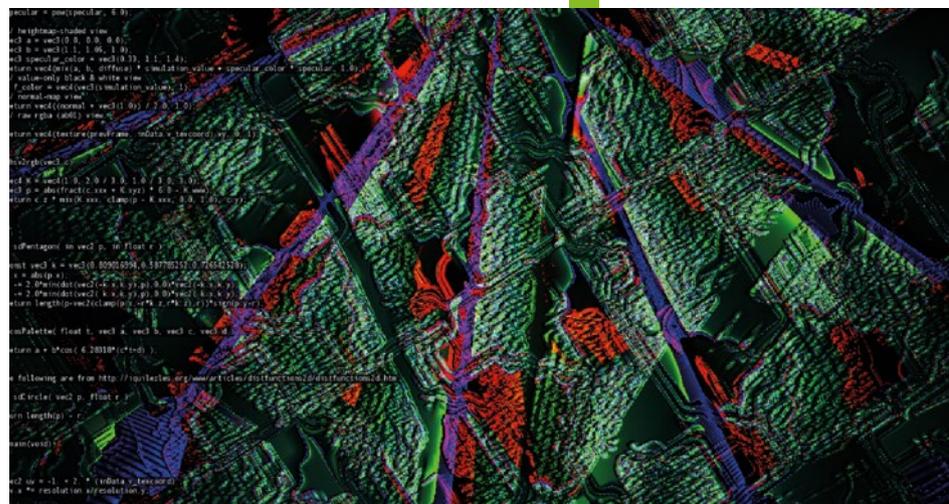


FUN FACT

UV doesn't stand for anything: it's named such simply because XY was already taken for the position of an object in a 3D scene.

◀ Here's an example of a ray-marched blob sketch. I was messing around with refraction, giving life to squishy diamonds.

▼ Here's an example of a live-coded shader made on the spur of the moment.



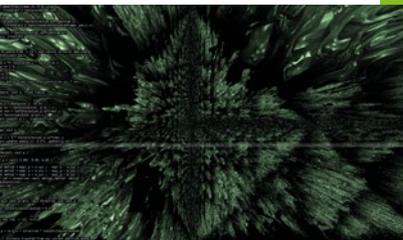


▶ A distorted torus shape is given a fleshy look by ray marching to create translucency.

DEMOSCENE

Live coding visuals originated in the demoscene, a community of people who hold conventions and competitions to code the most impressive graphics scenes (or demos) using the smallest amount of code. The demoscene stems from people coding the graphics version of a graffiti tag called 'intros' into the beginnings of pirated video games. Ray marching is a popular technique in writing demos.

✔ The visual part of my AV set with Pittsburgh musician Danielle Rager - or (arsonist), as she's known on stage. I'm using a simple Phong function to shade these ray-marched blobs.



```
uniform vec2 resolution;

varying vec3 v_normal;
varying vec2 v_texcoord;

// Define some constants

const int steps = 128;
// The maximum amount a ray can march.

const float smallNumber = 0.001;

const float maxDist = 10.;
// The maximum distance a ray can travel.

float scene(vec3 position){

    // This is different from the sphere
    // equation above in that I'm splitting the
    // position into its three different
    // positions and adding a 10th of a cos wave
    // to the x position so it oscillates left
    // to right and a (positive) sin wave to the
    // z position so it will go back and forth.

    float sphere = length(
        vec3(
            position.x + cos(time)/10.,
            position.y,
            position.z+ sin(time) +1.)
        )-0.5;

    // This is different from the ground
    // equation because the UV is only
```

```
// between -1 and 1 we want more than 1/2pi
// of a wave per length of the screen so we
// multiply the position by a factor of 10
// inside the trig functions. Since sin and
// cos oscillate between -1 and 1, that
// would be the entire height of the screen
// so we divide by a factor of 10.
```

```
float ground = position.y + sin(position.x *
10.) / 10.
        + cos(position.z * 10.) / 10. + 1.;
```

```
// We want to return whichever one is
// closest to the ray, so we return the
// minimum distance.
```

```
return min(sphere,ground);
}
```

```
vec4 trace (vec3 origin, vec3 direction){

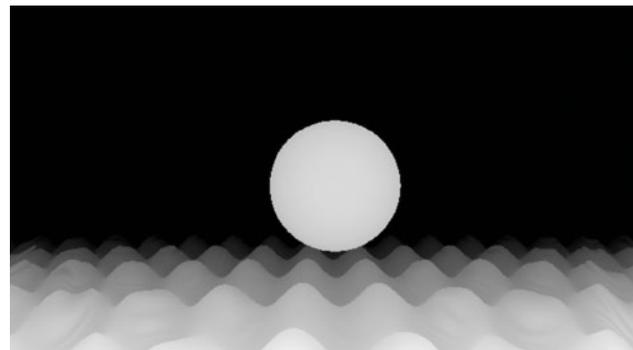
    float dist = 0.;
    float totalDistance = 0.;
    vec3 positionOnRay = origin;

    for(int i = 0 ; i < steps; i++){

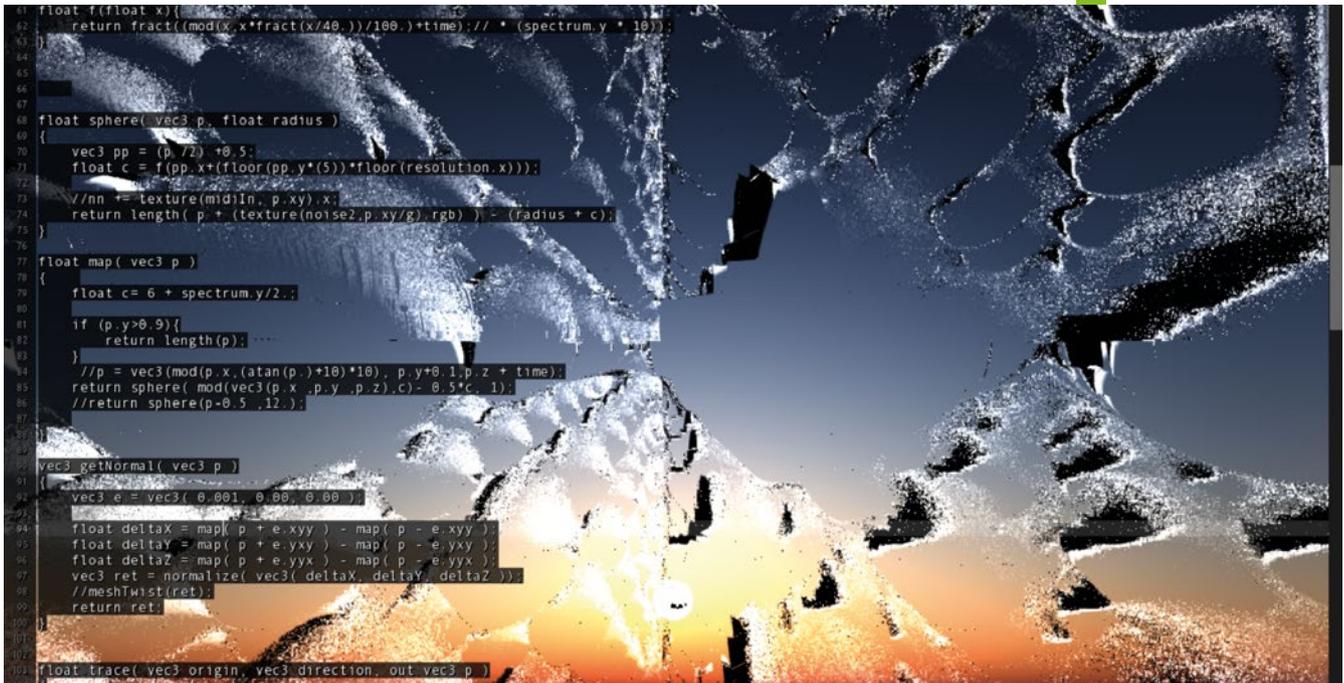
        dist = scene(positionOnRay);

        // Advance along the ray trajectory the
        // amount that we know the ray can travel
        // without going through an object.

        positionOnRay += dist * direction;
```



▲ Figure 3: This is what the code renders to. The grey tone corresponds to how far the ray needs to travel before hitting an object.



▲ The visual part of an AV set with {arsonist}.

```

// Total distance is keeping track of how
// much the ray has traveled
// thus far.
totalDistance += dist;

// If we hit an object or are close enough
// to an object,
if (dist < smallNumber){
    // return the distance the ray had to
    // travel normalized so be white
    // at the front and black in the back.
    return 1. - (vec4(totalDistance) /
maxDist);
}

if (totalDistance > maxDist){
    return vec4(0.); // Background color.
}
}

return vec4(0.); // Background color.
}

// main is a reserved function that is going
// to be called first
void main(void)

```

```

{
    // We are redefining the UV coordinates
    // (aka texcoords) to be 0,0 in the
    // middle of the screen this is because its
    // easier to work with the camera at
    // (0,0) instead of (0.5,0.5) for the SDFs
    vec2 uv = -1. + 2. * v_texcoord;

    // Unfortunately our screens are not square
    // so we must account for that.
    uv.x *= (resolution.x / resolution.y);

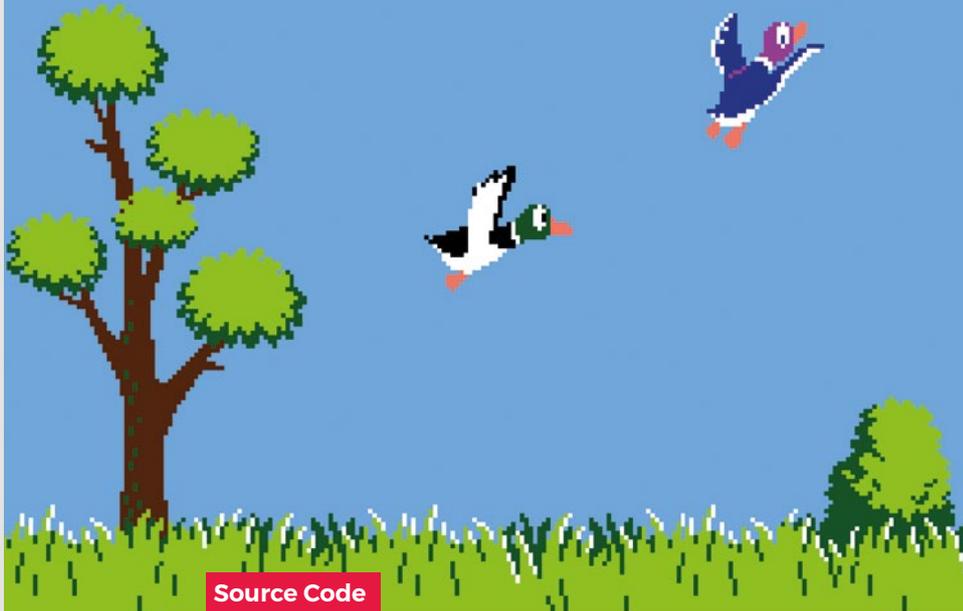
    vec3 rayOrigin = vec3(uv, 0.);
    vec3 camOrigin = vec3(0., 0., -1.);
    vec3 direction = camOrigin + rayOrigin;

    // This reserved variable is what we must
    // set the final color to
    gl_FragColor = trace(rayOrigin, direction);
}

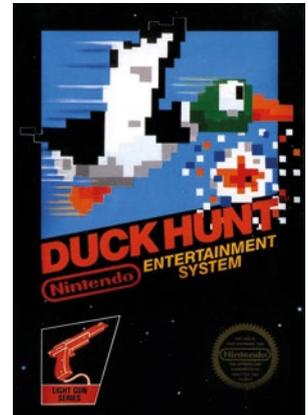
```

If everything works out, you should see something like **Figure 3**. It will vary a bit across screens with different resolutions.

I hope that you find this tutorial useful in learning about this new direction of rendering. You can find this code and the next steps you can take to learn more about live coding, shading, and ray marching at charstiles.com/raymarching. 



◀ *Duck Hunt* made effective use of the NES Zapper, and made a star of its sniggering dog, who'd pop up to heckle you between stages.



Create your own 2D shooting gallery



AUTHOR
RIK CROSS

Rik shows you how to hit enemies with your mouse pointer as they move around the screen

Shooting galleries have always been a part of gaming, from the *Seeburg Ray-O-Lite* in the 1930s to the light gun video games of the past 40 years. Nintendo's *Duck Hunt* – played with the NES Zapper – was a popular console shooting game in the early eighties, while titles such as *Time Crisis* and *The House of the Dead* kept the genre alive in the nineties and 2000s.

Here, I'll show you how to use a mouse to fire bullets at moving targets. Code written to instead make use of a light gun and a CRT TV (as with *Duck Hunt*) would look very different. In these games, pressing the light gun's trigger would cause the entire screen to go black and an enemy sprite to become bright white. A light sensor at the end of the gun would then check whether the gun is pointed at the white sprite, and if so would register a hit. If more than one enemy was on the screen when the trigger was pressed, each enemy would flash white for one frame in turn, so that the gun would know which enemy had been hit.

I've used two Pygame Zero event hooks for dealing with mouse input. Firstly, the `on_mouse_move()` function updates the position of the crosshair sprite whenever the mouse is moved. The `on_mouse_down()` function reacts to mouse button presses, with the left button being pressed to fire a bullet (if `numberofbullets > 0`) and the right button to reload (setting `numberofbullets` to `MAXBULLETS`).

“Pressing the light gun's trigger would cause the entire screen to go black”

Each time a bullet is fired, a check is made to see whether any enemy sprites are colliding with the crosshair – a collision means that an enemy has been hit. Luckily, Pygame Zero has a `collidect()` function to tell us whether the rectangular boundary around two sprites intersects. If this helper function wasn't available, we'd instead need to use sprite `x` and `y` coordinates, along with

width and height data (`w` and `h` below) to check whether the two sprites intersect both horizontally and vertically. This is achieved by coding the following algorithm:

- Is the left-hand edge of sprite 1 further left than the right-hand edge of sprite 2 (`x1 < x2+w2`)?
- Is the right-hand edge of sprite 1 further right than the left-hand edge of sprite 2 (`x1+w1 > x2`)?
- Is the top edge of sprite 1 higher up than the bottom edge of sprite 2 (`y1 < y2+h2`)?
- Is the bottom edge of sprite 1 lower down than the top edge of sprite 2 (`y1+h1 > y2`)?

If the answer to the four questions above is **True**, then the two sprites intersect (see **Figure 1**). To give visual feedback, hit enemies briefly remain on the screen (in this case, 50 frames). This is achieved by setting a `hit` variable to **True**, and then decrementing a timer once this variable has been set. The enemy's deleted when the timer reaches 0.



A simple shooting gallery in Python

You'll need to install Pygame Zero to get Rik's code running. You can find instructions at wfmag.cc/pgzero

```

WIDTH = 800
HEIGHT = 800

crosshair = Actor('crosshair')

# creating a new enemy
def newEnemy(pos):
    e = Actor('enemy', pos=pos)
    e.hit = False
    e.timer = 50
    e.hits = []
    return e

# creating a bullet that has hit an enemy
def newHit(pos):
    h = Actor('bullet', pos=pos)
    return h

# create 3 enemies at various positions
enemies = []
for p in [(0,200),(-200,400),(-400,600)]:
    enemies.append(newEnemy(p))

numberofbullets = 8
MAXBULLETS = 8

def on_mouse_move(pos, rel, buttons):
    crosshair.pos = pos

def on_mouse_down(pos,button):
    global numberofbullets
    # left to fire
    if button == mouse.LEFT and numberofbullets > 0:
        # check whether an enemy has been hit
        for e in enemies:
            if crosshair.colliderect(e):
                # if hit, add position to 'hits' list
                e.hits.append(newHit(pos))

```



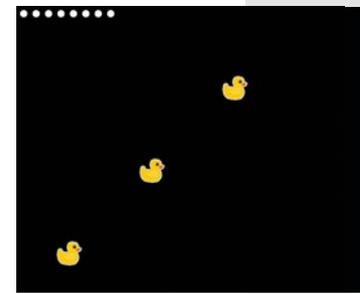
```

        e.hit = True
        break
        numberofbullets = max(0, numberofbullets -1)
    # right to reload
    if button == mouse.RIGHT:
        numberofbullets = MAXBULLETS

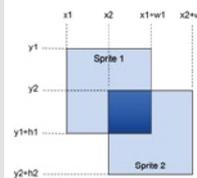
def update():
    for e in enemies:
        # hit enemies continue to display
        # until timer reaches 0
        if e.hit:
            e.timer -= 1
            if e.timer <= 0:
                enemies.remove(e)
        # move enemies if not hit
        else:
            e.x = min (e.x+2, WIDTH)

def draw():
    screen.clear()
    # draw enemies
    for e in enemies:
        e.draw()
        # draw enemy hits
        for h in e.hits:
            h.draw()
    crosshair.draw()
    # draw remaining bullets
    for n in range(numberofbullets):
        screen.blit('bullet',(10+(n*30),10))

```



^ Our simple shooting gallery in Python. You could try adding randomly spawning ducks, a scoreboard, and more.



< Figure 1: A visual representation of a collision algorithm, which checks whether two sprites intersect.

As well as showing an enemy for a short time after being hit, successful shots are also shown. A problem that needs to be overcome is how to modify an enemy sprite to show bullet holes. A **hits** list for each enemy stores bullet sprites, which are then drawn over enemy sprites.

Storing hits against an enemy allows us to easily stop drawing these hits once the enemy is removed. In the example code, an enemy stops moving once it has been hit.

If you don't want this behaviour, then you'll also need to update the position of the bullets in an enemy's **hits** list to match the enemy movement pattern.

When decrementing the number of bullets, the **max()** function is used to ensure that the bullet count never falls below 0. The **max()** function returns the highest of the numbers passed to it, and as the maximum of 0 and any negative number is 0, the number of bullets always stays within range.

There are a couple of ways in which the example code could be improved. Currently, a hit is registered when the crosshair intersects with an enemy – even if they are barely touching. This means that often part of the bullet is drawn outside of the enemy sprite boundary. This could be solved by creating a clipping mask around an enemy before drawing a bullet. More visual feedback could also be given by drawing missed shots, stored in a separate list. ^W

Programming with GUIs

Learn how to write your own Graphical User Interface-based applications with this free Raspberry Pi course

GET INVOLVED

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

What is it?

Discover how to build your own Graphical User Interface (GUI) with guizero. On this course supported by Google, you'll learn how to incorporate more interactivity in your programs, experiment with different types of widgets, and build the confidence to design more complex GUI-based apps in the future.

You'll have the opportunity to put what you've learnt into practice using the guizero library. Once you've completed the course, you'll be able to produce applications with interactive elements such as buttons and text boxes.

Who's the course for?

This course is designed for teachers delivering computer science lessons at GCSE level or equivalent. It's aimed at teachers who have basic experience with Python.

What will I learn?

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

- Experiment with different widgets that can be used to build a GUI
- Develop a GUI that responds to the user through the use of events
- Explain how event-driven programming works, and the differences between an event-driven program and a procedural program
- Produce a countdown timer using guizero's built-in methods
- Design several different GUIs, including a text editor and a picture-matching game.

When's it on?

Every run of a course has a set start date, but you can join it and work through it after it's started. To find out more, and sign up, simply visit the course page at wfmag.cc/gui-course.

> Learn how to build your own sleek user interface with Raspberry Pi's free GUI course.





The UK's
**BRAND NEW
FILM MAGAZINE**

TRY
**THREE ISSUES
FOR £9.99**
with the code 'RASPBERRY'

webscribe.co.uk/magazine/filmstories

From 8-bit to mobile: **THE ETERNAL CODERS**

Veteran 8-bit devs tell us about the pluses and minuses of making games – both back then and today

WRITTEN BY
GRAEME MASON

Few things mark the passing of time as poignantly as technology and, as a corollary, video games. In 40 years, we've progressed from *Pong* and *Space Invaders* to *Red*

Dead Redemption 2 and *Sekiro*. Yet while the worlds inside video games may have advanced, one thing has remained the same: as long as there have been games, there have been programmers working diligently to create them.

"Back in the late seventies, I bought a Sinclair programmable calculator," reveals Jas Austin, author of many notable 8-bit games, especially on the ZX Spectrum. "It was a small white LED machine, and as well as being a scientific

calculator, it also had the ability to enter a short program using a basic set of instructions. I found it fascinating that you could effectively change what it could do."

Unbelievably, Austin's first game was created on this primitive computer. "I didn't write any code of my own," he says, "but had a fun time entering and running the programs that came with it in a series of books. One of them was a very basic number-based game of *Lunar Lander*."

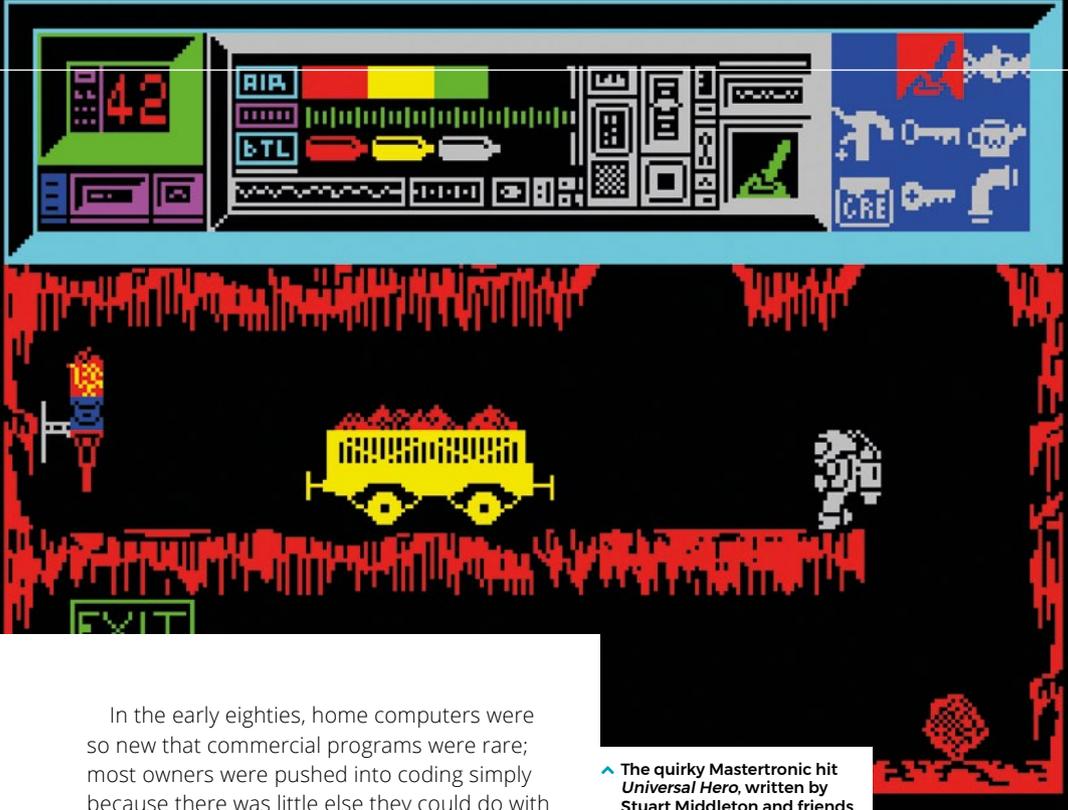
These type-in listings – pages of code in books or magazines – were a key factor in encouraging the curious into coding.

A NEW WORLD

Sinclair computers were a staple of the early eighties game scene, at least in the UK. Stuart Middleton, the bedroom coder behind the Mastertronic budget classic *Universal Hero*, among others, recalls receiving a Sinclair ZX81 for Christmas in 1981. "I didn't really know what it was or what you could do with it, I just knew you could program it," he enthuses. After convincing his parents to let him open the package before Christmas Day, Middleton read the manual from cover to cover. "I remember just printing my name on the screen and thinking of all the things I could do with it. I was hooked from that moment."



► Stuart Middleton's *How to be a Hero* was another decent game for the UK's ubiquitous publishers of budget titles, Mastertronic.



▲ The quirky Mastertronic hit *Universal Hero*, written by Stuart Middleton and friends.

In the early eighties, home computers were so new that commercial programs were rare; most owners were pushed into coding simply because there was little else they could do with their machines. Video games, thanks to *Space Invaders* and its arcade peers, had already captured the imagination of youngsters all over the world. Lancashire local Anthony Ball was one such pixel-eyed kid. "Next to the bus station in Accrington, there used to be an arcade, so I'd go in there while waiting for the bus and play video games. Then, in Blackburn, I'd go to WH Smith and read magazines like *Byte*, *Compute!*, and *Computer and Video Games*."

Interest piqued, Ball acquired the BASIC programming cartridge for his family Atari 2600. "It was rubbish," he recalls.

"But at school, they had a Research Machines 380Z which they kept locked in a store cupboard because they were afraid of what might happen if they turned it on."

At weekends, Ball would journey into Preston to experiment on the computers lined up in the electrical store, Laskys. "The staff were impressed and let me tinker around all day. I guess it helped sell the computers."

John Passfield, meanwhile, gives a perspective from Australia, where the Sinclair brand was nowhere to be seen. "I encountered my first computer, a Commodore PET, loaded with the game *Colossal Cave Adventure*," he remembers. "My mind was blown. I had to have my own computer. I got a VZ-200 that Christmas and began learning how to code from the manual and computer magazines."

FIRST EFFORTS

Passfield continues: "My very first attempt at a game was pretty dismal. It was called *Attack of the*

Invisible Werewolf. You had to move a dot around the screen until a random amount of time passed and the message, 'You were killed by an invisible werewolf' appeared on the screen."

Austin, having graduated from the scientific calculator to the ZX Spectrum, via the VIC-20 and ZX81, realised machine code was the way forward. "My first published game was *Bunny*

for Automata," Austin says. "It was written 99.9% in BASIC, but I wrote a tiny piece of machine code that flashed the border

and made a terrible noise. While unimpressive, it made me realise how much more powerful code could be, so my next game, *Pi-Balled*, was totally machine code. It was amazing how much more you could get out of the Spectrum [with machine code] compared to using BASIC."

Austin got his break early in coding, thanks to living so close to a software publisher. For Passfield, opportunities weren't quite so convenient. "I grew up in a country town in Australia, which was about as far away from the computer scenes in England and the US as you could get. In my town, I was the only kid making games, and when my first game was published in 1984, I kept it to myself because it was such a nerdy thing to do!"

THE COMMERCIAL DAWN

"I remember it was all very exciting," says Middleton when asked about the video games scene in the early-to-mid eighties. "All my friends were playing games, and we got together ▶



CODER PROFILE

STUART MIDDLETON

FIRST PROGRAMMED ON
ZX81

SELECTED EXPERIENCE
Elite Systems,
Mind's Eye, *Rebellion*,
FreeStyle Games

GAMES THEN
Universal Hero
ZX Spectrum

How to be a Hero
ZX Spectrum

Tank Attack
ZX Spectrum

RoboCop 3
Game Gear

GAMES NOW
Isle of Tune
iOS

Guitar Hero Live
iOS



CODER PROFILE

JAS AUSTIN

FIRST PROGRAMMED ON
Sinclair Programmable
Calculator

SELECTED EXPERIENCE
Creative Reality,
Bits Studios, Origin8,
FunFair Technologies

GAMES THEN

Pi-In-Ere

ZX Spectrum

Pi-Balled

ZX Spectrum

Nemesis the Warlock

ZX Spectrum

Rex

ZX Spectrum

The Planets

ZX Spectrum

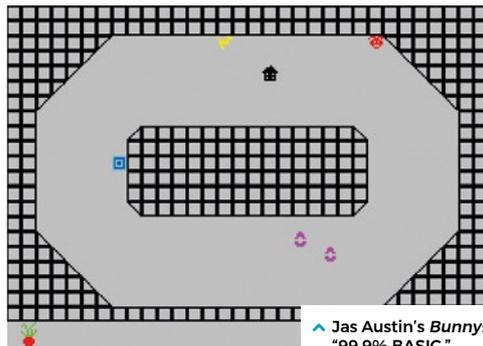
GAMES NOW

Transport Tycoon

iOS, Android

RollerCoaster Tycoon

iOS, Android



▲ Jas Austin's *Bunny*:
"99.9% BASIC."



▲ Jas Austin's *Transport Tycoon* (iOS).

all the time to watch and take turns playing them. They were excited about the work I was doing and pushed me to do more – some people played sports, or were good at art; I wrote games."

Yet video gaming wasn't the huge industry it is today, and was often regarded as nothing more than a passing fad, or a new hobby for kids. "I'd love to say I felt like I was forging a new path," nods Austin, "but I was quite young, and it honestly just felt like a cool hobby that I might get some extra pocket money from." Careers advice at school reflected the prevailing attitude towards games programming. "Whenever it was mentioned, the careers advice was dismissively 'Go work at IBM', which made me even more determined to continue and focus fully on game development." Meanwhile, Ball, having cut his teeth on an Atari 800 and SynAssembler, opted for the Atari ST, skipping BASIC and teaching himself 68000 Assembly.

After starting a *Gradius*-style shooter called *Bizarre* (created alongside future *Traveller's Tales*

"I was also constantly optimising, finding tricks like unrolling loops to increase speed"

co-founder Andy Ingram), Ball began work as a freelancer. At the same time, the way games were being made was already rapidly changing. "Coding at home was very much coding on the machine that played the game – if it crashed, you had to reload everything," he recalls. "Then, working for Zippo Games, they introduced me to the PDS [Programmers Development System],

so you could code on an IBM PC and send to a device such as the NES."

Working on Atari ST and Amiga games, Ball was soon teamed up with a graphics artist.

The days of the bedroom coder, working alone in their parents' house, were numbered.

THE BIG ISSUES

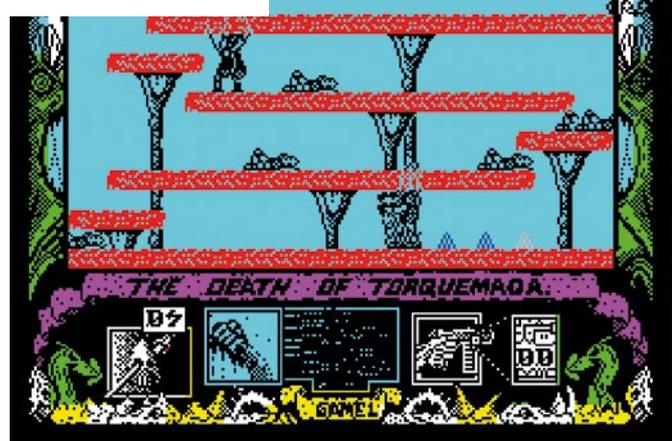
"Looking back, it always amazes me what we managed to do with those early machines," notes Middleton. "But at the time they were the norm, so any issues I faced only seem painful in retrospect."

Youthful enthusiasm, and the feeling they were working on the cusp of a breakout industry went a long way, as did the ingenuity to work around the problems that commonly arose. Remembers Austin: "Working on the Spectrum, memory was always a big hindrance, along with getting games to run at a reasonable speed."



▲ Regarded by many as one of the best ZX Spectrum games: *Rex*, by Jas Austin and Dave Dew.

▼ 2000 AD-licensed *Nemesis the Warlock* became a cult hit for Jas Austin.





The solution? Optimise, and wage war against anything that increased memory usage. "In some of my games, I 'cheated' and improved speed by having the gameplay area smaller than the screen size, which meant less memory to update," Austin says. "I was also constantly optimising, finding tricks like unrolling loops to increase speed here and there. But this normally increased memory usage – it was a constant battle."

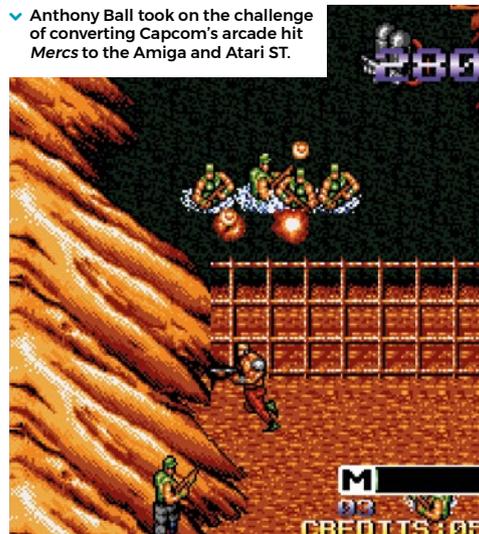
Even in small teams, coding was an isolated profession when compared to today, with little support and no internet to fall back on, as Passfield recalls. "One of the biggest issues was a lack of information; information about how to make games and how to get them to market. I learned a lot from the magazine listings – how wonderful it would have been to have had Google back then!"

PUBLICATION AND DISTRIBUTION

Games development in the eighties, as today, was just one facet of the industry. With all software distributed via cassette, cartridge, or disc, distribution chains and publishers flourished. Having made a game, your only real choice was to pick a bunch of software houses and post it off. "It was the wild west," exclaims Passfield. "I sent a game off to a local publisher, Honeysoft, and they accepted it. There was no contract, just me replying by mail to say, 'That sounds great!'"

Middleton, having worked for publisher Elite Systems, gained valuable experience in production and publishing. When he subsequently wrote *Universal Hero*, he called several publishers to see who would bite. "Mastertronic were nice enough to wine and

^ Anthony Ball took on the challenge of converting Capcom's arcade hit *Mercs* to the Amiga and Atari ST.



dine us, and give me an advance," he recalls, "so they got the game."

Buoyed by the success, Middleton used his experience to investigate self-publishing for his next game. The main pitfall was inevitable. "I called WH Smith etc for distribution, spoke to magazines about advertising, and contacted a duplication plant for a quote," he explains. "Then it all stopped when my local bank wouldn't give me a loan to pay for it all."

It's not hard to imagine a middle-aged bank manager with zero knowledge of video games staring confusedly over their horn-rimmed glasses at the young programmer's request for funds. But there was a hungry market for games, and the talented would get published, seen, and sold. The market was changing quickly, but good games were always in demand by publishers constantly living on the edge as technology evolved. ▶



^ Anthony Ball's *Gauntlet*-inspired *Dungeons* (iOS/Android).



CODER PROFILE

ANTHONY BALL

FIRST PROGRAMMED ON
Atari 2600

SELECTED EXPERIENCE
Zippo Games, Tiertex,
SinisterSoft

GAMES THEN

Cabal
NES
Mercs
Amiga, Atari ST

GAMES NOW

Dungeons
iOS, Android
Bacteria
iOS, Android
RetroStar
iOS, Android

Interface

The eternal coders



CODER PROFILE

JOHN
PASSFIELD

FIRST PROGRAMMED ON
Commodore PET

SELECTED EXPERIENCE
Gee Whiz! Entertainment,
Krome Studios,
Pandemic Studios,
Red Sprite Studios

GAMES THEN
Chilly Willy

Microbee

Halloween Harry

Microbee

Alien Carnage

PC

*Flight of
the Amazon Queen*

PC & Amiga

GAMES NOW
Save Our Village

iOS, Android

Beetastic

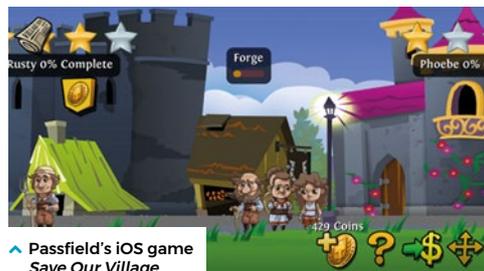
iOS, Android

Brainiversity

iOS

Snappy Word

iOS, Apple Watch



▲ Passfield's iOS game *Save Our Village*.

"With the fast turnaround back then, we had enough trouble just handling the development side," says Austin, who by 1987 had formed Creative Reality with two friends, David Dew and Neil Dodwell. "I was always happy to leave the publishing side of things to the companies that hired us."

FROM 8-BIT TO MOBILE

Initially, at least, the mobile gaming industry was as exciting and free as those early days of 8-bit development. "On mobile, to me, it's very much like the eighties," says Ball, who's created several retro-style games for iOS and Android. "With one or two-person teams creating something that's original and not based on a licence." Middleton, who worked on just about every platform going, recalls: "Working with mobile used to be great back when there was only one iPhone, and no-one had ever heard of Android."

Twelve years later, the constant stream of models and versions of each mobile phone bring their own quirks and problems. "It'd be easy if you only had to write for the state-of-the-art £1249 iPhone XS Max," continues Middleton, "but you must also consider the £15 Vodafone Smart first 7."

Austin echoes Middleton's lament,

describing an "ever-changing platform" where a game developer's builds can be instantly broken by the latest update – and then there's "the sheer amount of different devices out there," Austin adds, "all with varying operating systems, screen sizes, and capabilities."

Alternatively, both Ball and Passfield voice a differing concern with modern mobile games. Says Ball: "The main issue is marketing – with so many new releases, it's virtually impossible to get noticed. I'm surprised anyone makes any money at all." Passfield agrees. "It's a very crowded market, so you have to do something special to stand out."

"It's been an honour to have been part of an industry pretty much from its birth"



▲ The first of Passfield's two games for the Microbee computer – *Chilly Willy*.

IT'S NOT ALL BAD NEWS

Games can be marketed directly by the developer on mobile, which sounds like an improvement on the fixed distribution chains of old – but then there's the sheer number of competing games to contend with. For fans of old-school text adventures, this will strike a chord; by the late eighties, thanks to adventure creation utilities such as the Professional Adventure Writer and the Graphic Adventure Creator, the genre was saturated by efforts from amateur adventurers. But these utilities were simply the enablers, and that's partly what drives the saturation today.

"The tools for developing games these days are amazing," says Passfield. "Unity, Xcode, Flutter, Corona, Unreal, Buildbox, and Defold are just a few of the many options to make games, and most are free." And, unlike the past, there's lots of help at hand. "The amount of resources in terms of tutorials, videos, books, and blogs is also incredible. It's definitely easier to make a game these days."

Then there are the debugging features available with today's tools, as Middleton notes. "Back then, there was only limited debugging and no error handling at all – if your game did something stupid, it crashed. These days we have the wonders of structured exception handling to catch you when you're stupid, as well as high-level languages that take away a lot of the painful base code, and libraries that you used to have to write before you could do anything at all." But of course, there's even a drawback to this.

TO THE METAL

"I still love assembly and hitting the hardware directly," says Ball. "Finding new hardware tricks,

making the machine do the impossible – that's part of the thrill that's missing these days."

Adds Austin: "Back in the eighties, games were written in machine code, whereas now high-level languages like C++ and game engines make development easier." A result of this disconnect is coders such as Austin looking to develop on new machines such as the Spectrum Next, a computer that gives them the opportunity to revisit a method that is otherwise unavailable to them today.

It's not an attitude that everyone shares. "Has improved technology taken away the joy of programming? No way!" exclaims Passfield. "In fact, it's taken away a lot of the boring stuff and let me focus on making games. I was never one to make my own game engine; I just want to get in and get my game on screen as fast as possible. Shaping a game is what brings me joy."

FINAL ANALYSIS

"The parallels between indie mobile projects and writing Spectrum games in the eighties are undeniable," says Middleton. "My first mobile game was written entirely in my spare bedroom, and so reminiscent of coding as a teenager in my bedroom. In some ways, it's even easier now to write games from your bedroom."

There's a key difference, however: back in the eighties, if you knew what you were doing, your game was almost guaranteed to sell enough to make you a living. "Now you can spend a

year on a game and only make enough for a cup of coffee," Middleton reflects.

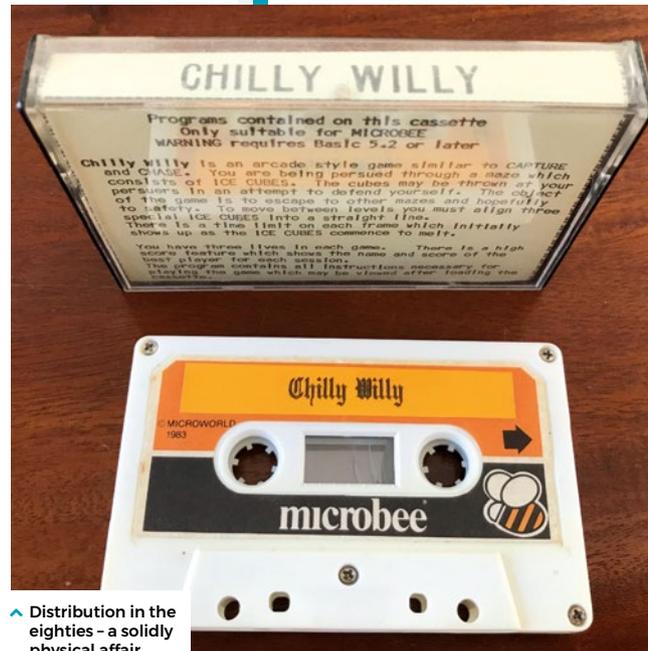
Austin, who's developed mobile games both independently and with a studio, also observes the similarities. "The indie industry today feels like it has a similar vibrancy to back in the day, where a single person or small team can retain 100% creative control over a game. But everything now feels far more organised and refined, and it's a crowded marketplace."

It's impossible for a comparison between two disparate eras to match in any significant way, but despite the relentless march of technology, there are still common strands.

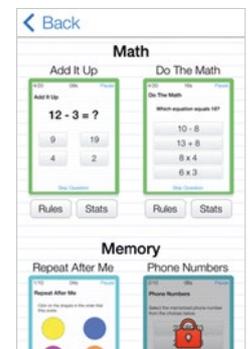
"For niche machines, people are self-publishing boxed copies of their games again, which is great," says Ball. It's this retro movement that has inspired him to code again, chiefly modern interpretations and mash-ups of classic games such as *Star Raiders*, *Boulder Dash*, and *Gauntlet*. Other trends are not as welcome, as Middleton notes. "One thing that doesn't seem to change is how often software houses go out of business. I've not been at one place for more than around four years, not through choice, but because most of them closed."

With even large, seemingly stable studios shutting down satellite offices with worrying regularity, this is a connection everyone would like to see less of. But having been there right from the start, how do our coders feel about their journey? "It's been an honour to have been part of an industry pretty much from its birth, and to see it grow into the ever-changing scene it is now," says Austin, who's presently working on a Spectrum Next version of perhaps his finest 8-bit game, *Rex*.

"You still find programmers who so obviously love their job," concludes Middleton. "And they live for it. They aren't everywhere, and there are a lot who just show up to do a job. But there are some, and they remind me of everyone back in the eighties. It wasn't something we did. It was something we were." 🐝

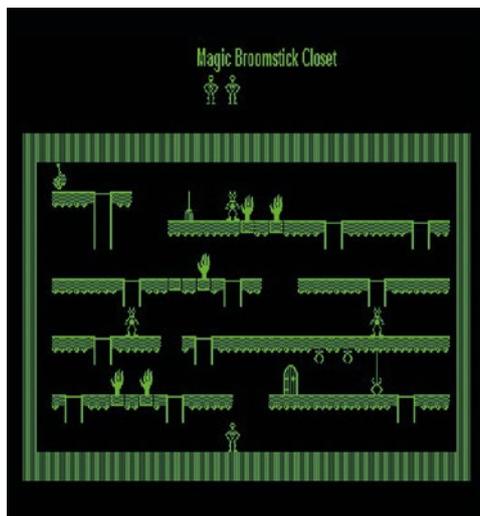


▲ Distribution in the eighties – a solidly physical affair.



▲ As a big fan of brain training games, John Passfield created his own version for iOS called *Brainiversity*.

▼ Passfield has even ventured into Apple Watch games.



▲ John Passfield followed up *Chilly Willy* with the superbly named *Halloween Harry* on the Microbee.



Developer Profile

Shinji Mikami

King of survival horror: the man who hates being scared

Born in the Yamaguchi prefecture of Japan in 1965, Shinji Mikami didn't start playing video games in any real sense until he was 20. He'd dabbled, sure – who hadn't? – but it wasn't the man's calling. No, young Mikami had initially dreamed of being a Formula One driver and, once that particular fantasy had been sidelined, worked on a degree in development of commercial science at Doshisha University. Was his major software development? Theory of horror? Not quite. It was the study of merchandise.

From 1985, though, Mikami began on the path he still wanders to this day – a lover of games, a player of games, and,

from 1990, a maker of games. Hired by Capcom about halfway through 1990 after being told a week or two prior he wasn't getting a job there, the 25-year-old set to work on... well, it was a quiz game on the Game Boy. An inauspicious start? Maybe. But everyone begins somewhere, and there was enough going on in this tiny, largely forgotten title that the brass moved him on to better things.

A stint on Disney licensed properties followed, with Mikami bringing his influence to the likes of *Who Framed Roger Rabbit* and the vastly overlooked SNES near-classic (though that might be an exaggeration), *Goof Troop*. The TV cartoon tie-in mixed inventive puzzles and combat with a serious *Legend of Zelda* vibe, and proved an early example that it was possible to make something worthwhile from an existing license. Disney's *Aladdin* followed not long after, this one with Mikami in position as planner and showing us a friendlier, swordless version of the street rat's adventures in Agrabah when compared to the US-developed Mega Drive version.

1993 proved the turning point for Mikami, though, as he was called in to discuss working on a project based on a 1985 Capcom NES title, *Sweet Home*. This new game was to bring horror gaming to the mainstream, using systems like the item management of the NES game, and Mikami was asked to head up the project. Naturally, he... was unenthusiastic, as he had zero experience making anything related to horror, and admitted





^ Mikami: doesn't like being scared; does like hats.



^ Goof Troop was far better than expected.



^ Early Resi concept art shows two dropped characters.

he didn't even like being scared by entertainment products. Nevertheless, Mikami agreed to head up the project, and the rest is industry-changing history.

'MAKE IT HARDER'

'Don't you dare put Capcom's name to shit like this' was the reaction of Capcom higher up and Mikami's project boss on *Biohazard* – retitled *Resident Evil* in the West. Yoshiki Okamoto was not happy with an early version of the game sent his way, complaining in large part about the steep difficulty of the game, and worrying that players around the world would be put off by elements like disconnected item boxes, requiring massive backtracking just to get a thing you needed at any given moment.

Despite the request to reduce the game's difficulty, which Mikami agreed was necessary and implemented, when *Resident Evil* made its way to the West, the request came in to make it harder again. "The US staff asked us to create this more difficult version, so that the game could be rented in the US," Mikami explained in a 1998 WildStorm interview. "If the game could be completed in a few days, fewer units of the game would sell." We can't say for sure if this worry stood on shaky ground or not, really, but the fact is *Resident Evil's* 1996 release saw millions of copies sold and a new paradigm for cinematic gaming as a whole. Yes, the "Jill sandwich" line was rubbish, but there's no denying the impact of Barry Burton and co.

From there, it was a mix of moonshots and middle ground, with Mikami able to pick and choose the projects he



^ The Evil Within: run away from the thing with the chainsaw.

worked on over the years. At Capcom, he was afforded the chance to head up projects like the Capcom Five and Clover Studio, but life as a producer never quite suited the man. "I can't help thinking that if *Resident Evil* hadn't sold so well, maybe I could have spent that period of my 30s doing creative work

"At Capcom, he headed up projects, but life as a producer never quite suited the man"

as a studio director instead of working as a producer," Mikami told GameSpot in 2016. "Although I still feel like a very lucky man to be able to have those kinds of worries."

The desire to return to a purely creative role proved too strong, though, and Mikami has spent the past decade-plus moving through a few studios – most recently Tango Gameworks, which

he founded in 2010 – and tinkering with unique projects along the way. While he has returned to the well a couple of times, notably with the *Resident Evil* remake and its fourth iteration back at Capcom, Mikami has generally kept his focus straight ahead, looking for the new newness and making sure not to retread too much old ground. When you've got a man who will say things like: "The *Colin McRae* games on the PlayStation... got very close to the survival horror structure," as Mikami did to *The Guardian* in 2014, you know you've got a special mind at work. Long may it continue. 🎮

^ The upcoming *GhostWire: Tokyo* looks spooky, but not scary.



Planning to direct

10 titles Mikami planned or directed

‘Planner’ is just another name for ‘director’, right?

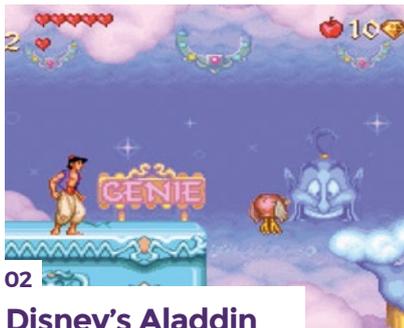


01

Capcom Quiz: Hatena? no Daibouken

Game Boy – 1990

Mikami took three months to make his first game at Capcom, the result being a Game Boy trivia title based on existing Capcom franchises, with the likes of Firebrand and Dr Wily trying to stop the player from... answering questions correctly. Look, it's never been translated to English, and quiz games in another language are hard to get a handle on.



02

Disney's Aladdin

SNES – 1993

Cited as one of the SNES' most overlooked games, *Aladdin* is a surprisingly complex take on platforming action. Less fondly remembered than Virgin Interactive's Mega Drive version, the SNES game has built up a decent following in the years since its release. It's not his best work, but generally speaking, *Aladdin* showed there was real potential in the young Mikami's ideas.



03

Resident Evil

PS1 / Saturn / PC / DS – 1996

It 'didn't matter' if the original *Resi* sold many copies – 150,000 each on PlayStation and Saturn was enough for Capcom. The lack of commercial pressure on Mikami and his team of 60 meant the creative juices flowed, a new genre was established, and *Resident Evil* went on to sell millions and create a franchise fast approaching its 10th main entry. 300,000 would have been enough.



04

Dino Crisis

PS1 / Dreamcast / PC – 1999

Sidestepping the zombie behemoth, Mikami shifted focus to another form of survival horror at the millennium's end. *Dino Crisis* riffed on *Jurassic Park*, tasking players with uncovering the secrets of a research facility overrun with unfossilised dinosaurs. Real-time 3D backgrounds – originally intended for *Resident Evil* – made an appearance and *Dino Crisis* went on to sell 2.4 million copies.



05

Resident Evil

GameCube / multi – 2002

It's not a misprint. Mikami returned to the franchise he had created to head up development of a near-complete remake of the original game. Shifting to the significantly more powerful GameCube, 'REmake' was a chance for the director to recreate the original game in a form closer to his original vision. It has since gone down as one of the best remakes in gaming history, so... that's a bonus.



06

P.N.03

GameCube – 2003

The first of the Capcom Five – five titles released exclusively (initially) on GameCube in a project headed by Mikami – *P.N.03* was... well, it got better when it was effectively repurposed into *Vanquish*. A mishmash of ideas, the third-person shooter was seen by some as clunky, others as intentionally challenging. All things considered, it was a bold experiment from Mikami, but ultimately lacking.



07

Resident Evil 4

GameCube / PS2 / multi – 2005

The fourth release in the Capcom Five, meanwhile, reinvigorated a flagging franchise and single-handedly raised the bar for pretty much all action games to follow. It's easy to overlook just how impactful *Resident Evil 4* was, but back in 2005, it was a revelation. It looked great, it changed what we thought we knew about games, and it kickstarted a revolution in third-person action titles.



08

God Hand

PS2 – 2006

God Hand was Clover Studios' final release, and on reflection would appear to be every single idea Mikami had in the few years he was working at the studio, all screwed up into a big ball and with an intensely difficult fighting game wrapped around it. *God Hand* is the stuff of actual legend – true 'love it or hate it' stuff, and for our money one of the best things gaming has ever produced. IGN gave it 3/10.

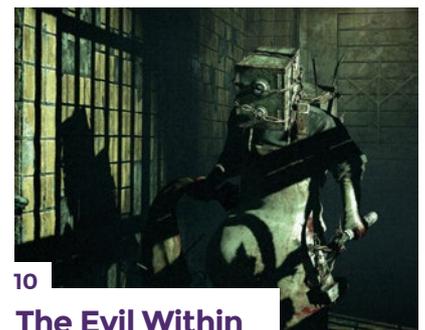


09

Vanquish

PS3 / X360 / PC – 2010

Not a follow-up to *P.N.03*, but clearly everything that non-classic should have been – *Vanquish* was and is one of the best pure action shooters we've ever played. Mixing tight third-person mechanics with an almost shoot-'em-up-like focus on all-out attack, combos, and beating your previous scores, this would be PlatinumGames' one true masterpiece, were it not for the existence of *Bayonetta*.



10

The Evil Within

PS4 / XB1 / PC / PS3 / X360 – 2014

Mikami's Tango Gameworks produced *The Evil Within* as its first release, and it marked the first time the director worked with a non-Japanese publisher. The result was what we expected, with Bethesda generally staying out of the creative process and allowing Mikami to do what he does best: scare people. Oh, and have them chased by an unstoppable thing with a chainsaw.

Subscribe today



SAVE
49%

wfmag.cc/subscribe

13 issues for just £20

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

Rolling monthly sub

- > **Low initial cost** (from £4)
- > **Cancel at any time**
- > **Free delivery to your door**
- > **Available worldwide**

Subscribe for 12 months

Receive all 26 issues

£40 (UK) **£75** (USA)
£65 (EU) **£75** (RoW)

Offers and prices are subject to change at any time



Digital subscriptions from

£1.99



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

Subscription queries: wireframe@subscriptionhelpline.co.uk

Info

GENRE
Narrative
adventure

FORMAT
PS4 (tested) / PC
/ XBO

DEVELOPER
Jo-Mei Games

PUBLISHER
EA

PRICE
£15.99

RELEASE
Out now

REVIEWED BY
Jon Bailes

HIGHLIGHT

In one sequence, Kay is forced to climb the inside of a crumbling tower, the light pulsing on and off as air vents flare up with the breath of an angry monster. It's *Sea of Solitude's* most traditionally game-like section, but the simple idea of being repelled and plunged into darkness as you struggle to ascend makes it the most evocative.

Sea of Solitude

You're never alone when you've got your inner demons

Before you start *Sea of Solitude*, it tells you its aims in no uncertain terms. A message from creator Cornelia Geppert tells you that this is "a personal project about loneliness" and "what it means to be human and live with all of life's ups and downs." It's an apt introduction to a game that's committed to confronting sensitive topics, but spells them out a little too bluntly.

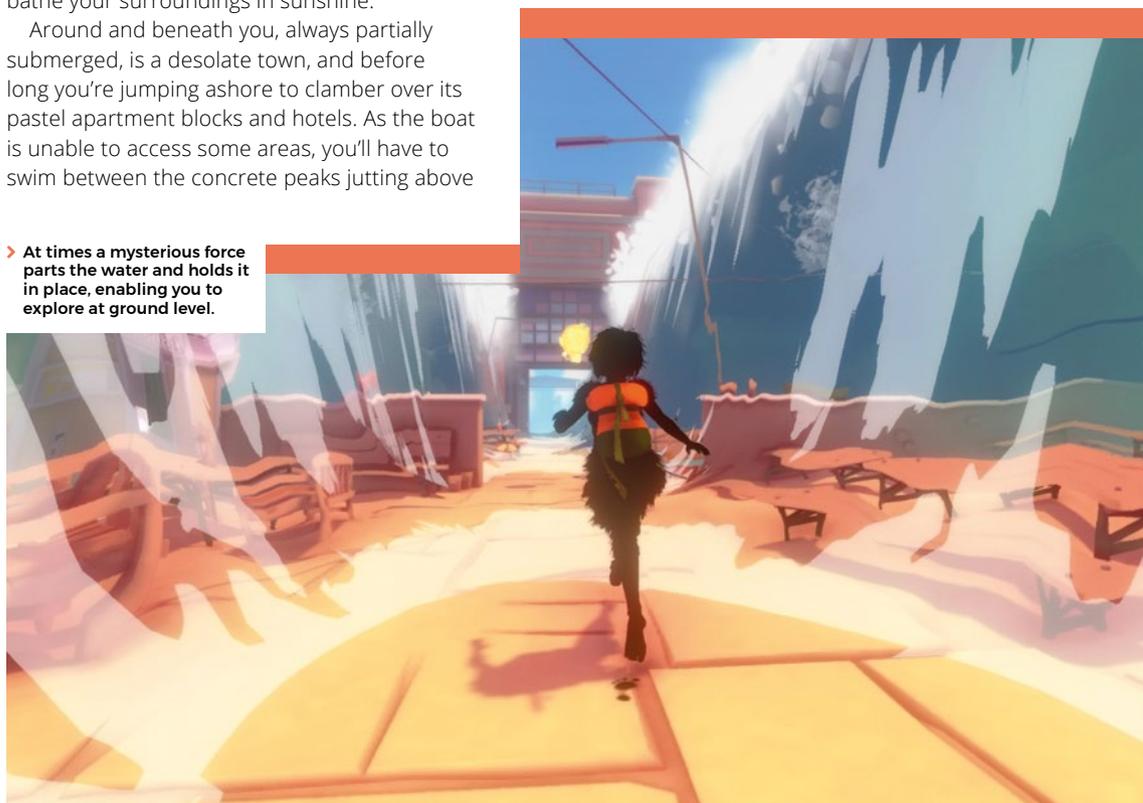
The first thing you hear as the prologue begins is the main character, a young woman named Kay, lamenting her sense of isolation, despite the friends and family around her. Then she wakes up in a tiny boat, alone in a vast expanse of water beneath a black sky, the rain battering down. Progress through this reflection of Kay's mental state is all about light. You soon gain the ability to shoot flares to pinpoint the way forward, and clear areas of 'corruption' to bathe your surroundings in sunshine.

Around and beneath you, always partially submerged, is a desolate town, and before long you're jumping ashore to clamber over its pastel apartment blocks and hotels. As the boat is unable to access some areas, you'll have to swim between the concrete peaks jutting above

the surface. With the dark comes a huge sea monster, circling like a hungry shark and turning each dip into a panicked dash. In the light, you take your time to look around, enjoy the warmth, and seek out collectables.

The crux is how quickly conditions can change. At any moment, day can return to night, or the sea can rise or sink to reconstruct the landscape, often bringing forth a new monster. With these shifts, the town has an almost dreamlike ability to rework itself and reveal new locations in disorientating loops. You travel far away into a structure that you hadn't noticed before, only for it to be swallowed again and lost forever as you exit, somehow depositing you back at a central landmark. It's impossible to gauge the scale of the place, or whether the structures themselves are moving under the waves.

› At times a mysterious force parts the water and holds it in place, enabling you to explore at ground level.





▲ A clean visual style with blocks of colour helps create the game's dramatic contrasts in atmosphere.

For a game about life's ups and downs, it's an exquisite metaphor, and a great base to explore themes such as bullying, depression, and, of course, loneliness. Kay feels adrift in her life, lacking a singular sense of direction, tossed around by unstable relationships and self-doubt. As the water surges, she feels flooded, out of her depth; as it falls, she feels stranded but also on solid ground. That huge body of liquid is a temperamental character, threatening to engulf you one moment, only to recede in the next and uncover a new path.

There's complexity here too. Different weather conditions symbolise different emotional states, such as the grey fog that turns water to suffocating mud. And even solitude itself isn't always expressed as painful loneliness. When you're wandering sunlit streets or floating under blue skies towards a distant building, there's a wonderful serenity.

You also learn that the problem relationships in Kay's life are as likely to require more space as more intimacy. There's no universal answer.

All Kay can do is keep trying and learn to accept her mistakes.

It's a shame that *Sea of Solitude* has a habit of smothering its metaphors. Kay talks a lot, and the chatter between her and the monsters is often needlessly literal. "You only think about yourself"; "You never do anything," shouts a shelled fiend that embodies her inner fear. Another sad monster, representing her distant brother, keeps moving away as she draws near. "Why won't he let me get close?" Kay asks, in case you'd missed the obvious. It doesn't help that the voiceovers are stilted and unconvincingly emotive, or even out of sync with the action.

As the game delves deeper into Kay's relationships, it relies more on chunks of overlaid dialogue. These repressed memories and past revelations depict very real experiences, and may well touch a nerve with anyone who has gone through similar problems, but they start to invade the game space.

"A heartfelt exploration of difficult themes"

Whereas some sections cleverly represent themes through environment design and forms of interaction, others fill time with generic tasks while the monsters work through their issues.

It would miss the point to want greater depth or challenge, but certain areas need something more meaningful to do. It's symptomatic of a rawness in *Sea of Solitude* that conveys its passion, but could still do with some refinement. An ability to interact with objects, for example, is completely forgotten for long stretches of the game, when it might have been more gainfully employed. And there's some clumsy objective design, like when you're left free to explore an area before you've activated the trigger that fills it with stuff to discover, which drains the game's fluency.

What you'll hopefully remember most is the town, the sea, and the monsters.

The combination of the changing geography and the distinctive forms of its shadowy inhabitants gives each part of the game its own emotional note. In its forms of communication, *Sea of Solitude* can't match the quiet understatement of *Rime* or the dense craft of *Celeste*, but there are times when it comes close, and it's a heartfelt exploration of difficult themes. 🗣️



▲ Kay's family members appear in forms that match their personalities and relationship to her.

VERDICT

A striking study of loneliness that could have done with a lighter touch.

62%



▶ There are few actual dangers in the game, so the presence of this toothy creature is especially alarming.

Rated
Review

HIGHLIGHT

Vacation Simulator introduces 'Memories' – short tasks you must complete for the bots who inhabit the island to unlock more areas and activities to enjoy during your stay. It's a clever, relaxed progression system with tons of varied objectives, like snapping pictures or delivering food.

Review

Vacation Simulator

Steam up your goggles with Owlchemy's sizzling summer sim

Info

GENRE
Virtual reality

FORMAT
PSVR (tested)
/ Other VR
platforms

DEVELOPER
Owlchemy
Labs

PUBLISHER
Owlchemy
Labs

PRICE
£26.99

RELEASE
Out now

REVIEWED BY
Jordan Oloman

VERDICT

Vacation Simulator is a delightfully silly summer fling hampered by some ancient peripherals.

77%

Given the effort it takes to pull on a plastic headset and wire yourself in to enjoy virtual reality, it's appreciated that Owlchemy Labs opted to simulate a vacation for... well, *Vacation Simulator*. A follow-up to the runaway success story of *Job Simulator*, a slice of absurd mundanity centred on office-based tasks, the studio's last game was polished YouTube bait – but after completing the limited set of objectives, it did start to feel quite hollow.

This time around, Owlchemy offers a more versatile virtual destination. You can recline in three charming low-poly holiday biomes – a forest, a snowy mountain, and a grill-tastic beach resort – each populated by floating CRT monitors with both sharp corners and tongues. The freedom to teleport around locations has been grafted in, which is liberating compared to its static predecessor.

Of course, Owlchemy couldn't settle for recreating the modern vacation experience without the addition of selfies, which use the in-game camera. These arrive in tandem with a character customiser; a fine addition, given one of the zones is a hotel room where you can brush your virtual fangs and gear up for another day of paradise. The dev team also proves itself adept in the comedy department too, with the game providing some big belly laughs. One poster I caught while riding the elevator down from my suite read 'Vacation Island – It sure is a place', which earned a good chuckle.

There are some surprisingly brilliant adventure game mechanics in *Vacation Simulator*. Items from

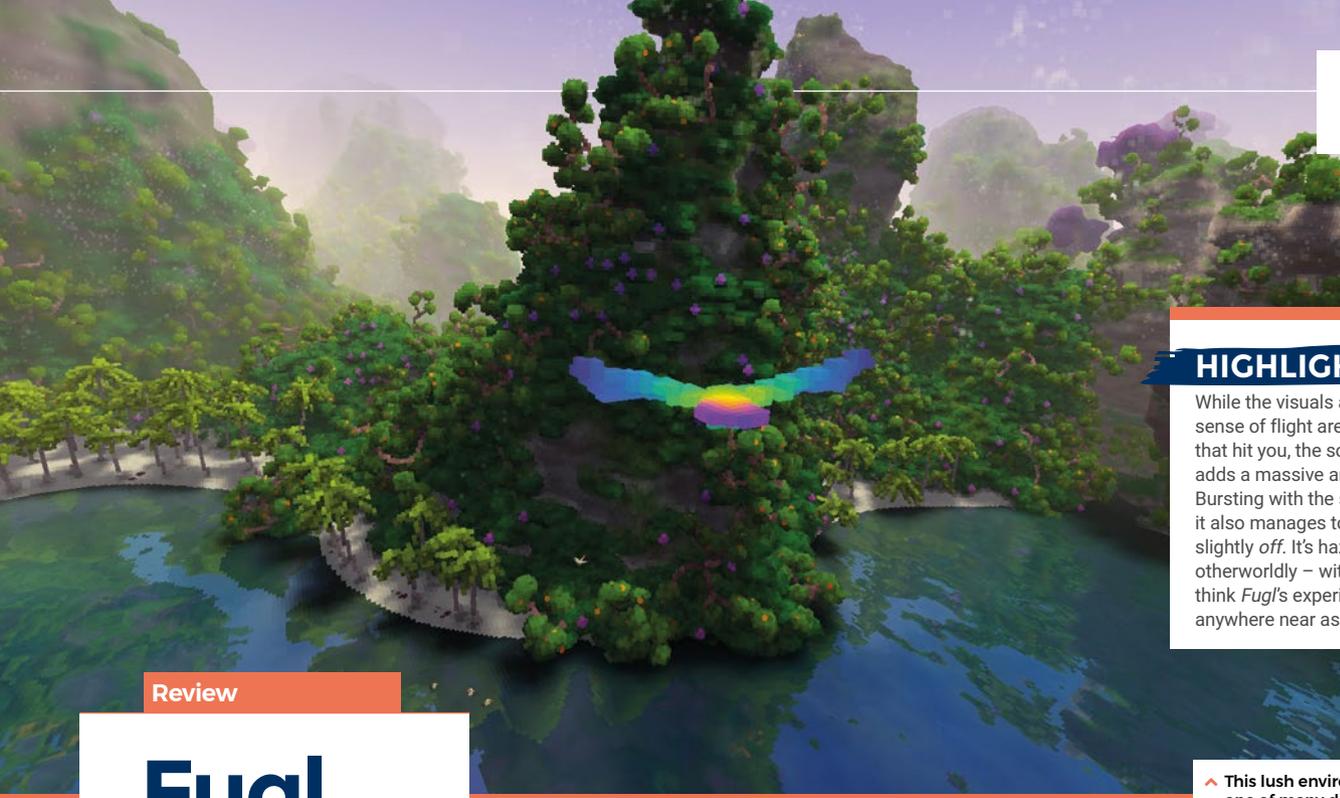
one location are often best used in others, and you can collect them in your backpack for ease of access. Even outside of puzzles, I loved packing my bag with snacks for a day out in the forest. Owlchemy truly understands the strange joy of boring tasks – *Vacation Simulator* is a tour de force of endearing banality.

So many headset-bound titles feel like theme park rides that want to drag you from location to location doling out short, often violent vignettes that have been done to death. You can take your time to chill in *Vacation Simulator*, and the game won't put the pressure on – a wholly underrated feature in VR. From volleyball to rock climbing and plenty of grillside maintenance, the activities are varied, interesting, and consistently toy with your expectations.

Unfortunately, the bane of PlayStation VR rears its head as soon as you strap on your Move controllers: the hardware's practically neolithic compared to its competition. Tracking problems soon mount up, and drag you kicking and screaming from the chilled ambience, while a painting minigame with Bob Ross vibes is derailed by ruinous glitches. It's definitely worth it if you don't have any other headset, but *Vacation Simulator* using PSVR is the least intuitive version of a fantastic sandbox. 🐼

✓ **Bullying your fellow island inhabitants is half the fun, enabled by the sheer volume of items you can chuck in *Vacation Simulator*.**





HIGHLIGHT

While the visuals and dizzying sense of flight are the first things that hit you, the sound design adds a massive amount to *Fugl*. Bursting with the sounds of nature, it also manages to feel ever so slightly off. It's hazy, dreamlike, and otherworldly – without it, I don't think *Fugl*'s experience would feel anywhere near as cohesive.

^ This lush environment is just one of many different biomes, including polar ice and canyons.

Review

Fugl

Fly free, as free as a crab!

Philosophy lesson time: Plato had this idea called the Theory of Forms. It's impossible to describe something, such as a cat or a chair, with properties that are both exclusive and universal to 'a chair'. Four legs? Nope, three-legged barstools exist, so that isn't universal. A platform for sitting? So are tables, beds, and picnic blankets, so that isn't exclusive. Therefore, Plato argued, there must be a higher 'form' of chairhood that we cannot attain, and all our attempts at creating one are just lesser shadows of that ideal. Replace chairs with birds, and that's, more or less, what *Fugl* is about.

In *Fugl*, you don't play as a single character. Instead, you're the concept of flight itself, and take on myriad appearances as you explore procedurally generated voxel landscapes. Where you fly, and which animals you interact with, will change you. Flit into trees, and you might turn into a flying monkey-bird-nightmare; heading into a cave could turn you into a bat.

Dive into the water, and you could turn into a fish. The closest thing *Fugl* has to a goal is to find and collect all the different forms throughout the map. Each new appearance is unlocked for future use and populates the main menu backgrounds, but there isn't really any sort of story, guidance, or justification for doing this. You just do it because it's there and can be done. Instead, *Fugl* is a highly meditative game, and it's an interesting one, because it straddles the line between two main types of game meditation. First, there's the obvious soothing aspect of flying through beautiful environments and taking in the world's sights and sounds – akin to games like *Flower*, *Journey*, and *Proteus*. This is doubly engrossing in the optional VR mode.

But there's also a thick layer of mechanical meditation, more commonly associated with stuff like *Tetris* and *Puyo Puyo*. Flying isn't easy or gentle; it's fast, sometimes fiddly, and unspoken challenges can occur spontaneously. Whizzing through a cave network at high speed, or avoiding all the trees in a forest, or skimming along the edge of the water without falling in, or pulling anything off without crashing headfirst into a cliff face, is surprisingly tricky – and absorbing.

While the lack of a distinct goal may prompt a sneer and the accusation of this being a 'walking sim', *Fugl*'s been on my mind ever since I played it. It's a mishmash of ideas that somehow come together to create a beautiful, challenging, allegorical, and downright bizarre game that's well worth checking out. Even if it's just to witness the eldritch abomination that is a bird-shaped monkey. 🐼

Info

- GENRE**
Casual?
- FORMAT**
PC (tested) / Oculus Rift (tested) / Mac / iOS / Android
- DEVELOPER**
Team Fugl
- PUBLISHER**
Kotori Studios
- PRICE**
£6.99
- RELEASE**
Out now

REVIEWED BY
Joe Parlock

VERDICT

It's a small game that you may be done with in less than an hour, but that experience is odd, vibrant, and absolutely worthwhile.

65%



^ Flying crabs. Eat your heart out, Attenborough.

Rated
Review

HIGHLIGHT

Not to be thoroughly shallow, but visually, *Blazing Chrome* is an absolute delight. It's closer to *Metal Slug* than *Contra* in motion, but either way that fabulous 1990s-style pixelated look does wonders for the whole experience – it offers a real sense of place, and you genuinely can, sometimes, forget you're playing something released in 2019.

Review

Blazing Chrome

Like visiting a whole new *Contra* (“country”)

▲ Bosses veer ‘large’, and definitely stick with ‘challenging’.

Info

GENRE

Arcade shooter

FORMAT

Switch (tested)
/ PS4 / XBO
/ PC

DEVELOPER

JoyMasher

PUBLISHER

The Arcade Crew

PRICE

£15.29

RELEASE

Out now

REVIEWED BY
Ian Dransfield

VERDICT

A new *Contra* in all but name – great fun, though admittedly *rock hard*.

79%

Back in *my* day, we didn't have fancy pants things like unlimited continues – no, we just grit our teeth and got through things one failure at a time.

None of this snowflake nonsense like *Blazing Chrome* throws at you, no siree. Well, unless you were playing the Japanese version of *Contra: Hard Corps* – but who of us were actually playing that all those years ago? Only the rich kids, and they can do one.

I mean, this is balderdash; *Blazing Chrome* does offer unlimited continues (and a limited save function, allowing you to pick up where you left off, though only from the very beginning of a level), but in no way does that make it a breeze or a Friend To The Millennial. By the way, if you were born around 1982-ish to around 1999–2000, you're a millennial, but that's by the by. *Blazing Chrome*. What is it? It's a side-scrolling shooter that apes *Contra* so much it would be contra-rian to mention anything else, really.

There's some *Metal Slug* in there, sure, and probably the odd bit of a thousand other arcade shooters from the 1980s and 1990s, before everything Went Bad – but the main influence slathered all over this delectable little morsel is *Contra*. It's *Hard Corps* without the wolf man.

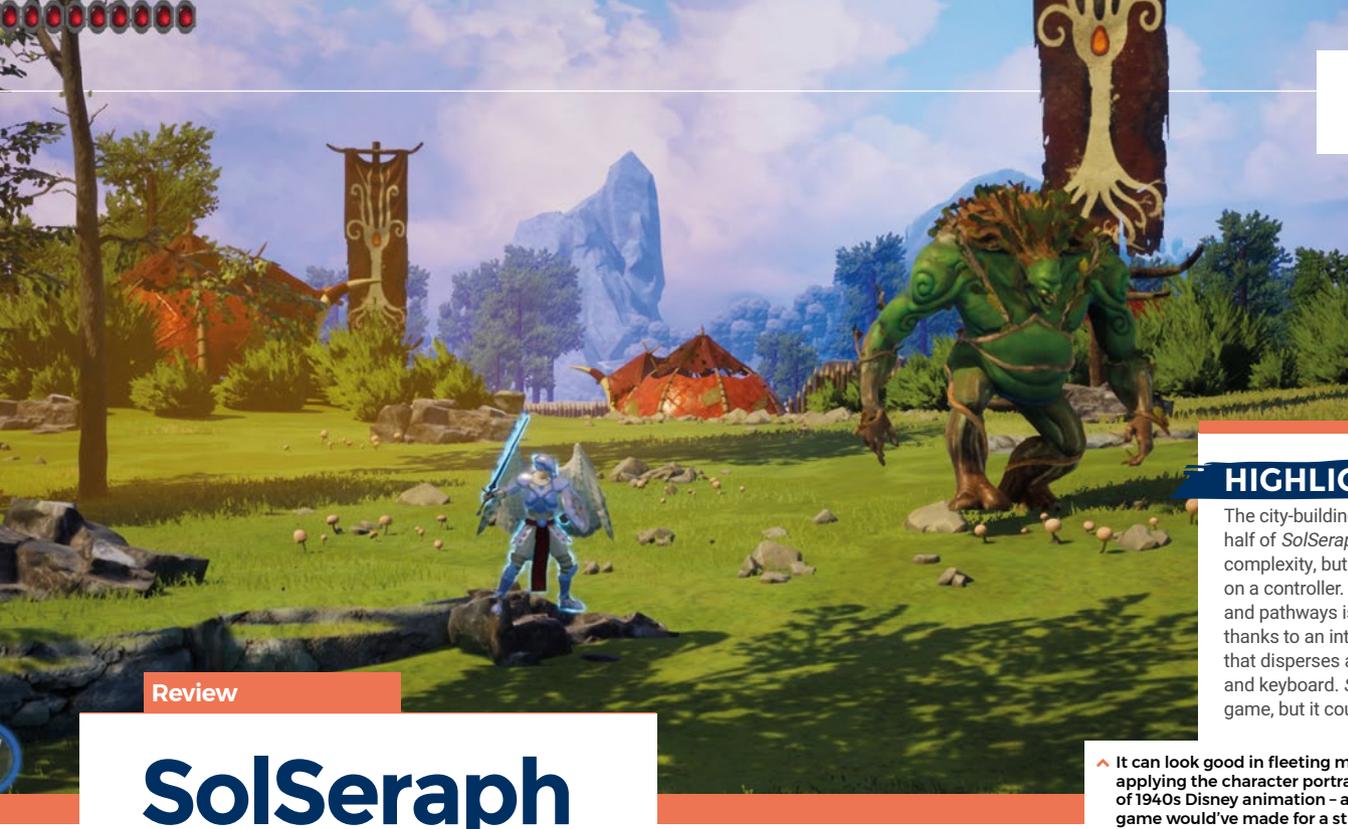
Alien Wars without those dogs eating from bins. The original *Contra* without the significantly better home conversion on NES. If we weren't getting a new *Contra* soon from Nobuya Nakazato – he of *Contra 3* and *Hard Corps* fame – I'd be hailing this as the best and only chance of playing a good *Contra* game we'll likely see for a long time.

And to think I promised myself I wouldn't mention *Contra* too much. So, *Blazing Chrome*: it's a handful of levels, each one being significantly different from the other and presenting the sort of lengthy 'Oh crap, I thought that was the end of it' challenge you genuinely don't see much of these days. Even the git gud crowd tends to know when it's approaching an end-point, and I have to say that obfuscation – is this the final boss? – does wonders for the tension of each playthrough. Tension? In a mindless shooter? Well, of course – it's hard. Very hard. One-hit-kill hard, and lose all your lives you're back to the beginning of the entire level hard. The weapon you have equipped (apart from the default potato gun) is lost when you die hard. *Hard-hard*.

Sometimes it feels a bit much, and more than once the Switch risked being flung through the nearest window. There are times when balance feels a bit skewed; as though enemies shouldn't be *able* to shoot as quickly as they do, or fire at *that* particular angle, but really it's just frustration borne of another failure. *Blazing Chrome* is great fun. A really surprisingly fantastic riff on *Contra* and arcade shooters of days gone. Its length is inflated artificially because of the difficulty so your mileage may vary, but for me, it's a solid recommendation. 🍷

▼ Weapon selection is limited, but varied, and includes this... thing.





Review

SolSeraph

A genre-mixing spiritual successor without a sol

A *ctRaiser* was a peculiar game, namely due to its unorthodox mixture of action platforming and city-building strategy. It's also nearly 30 years old, which is pertinent when you consider how much better it is than *SolSeraph* in almost every aspect of its design. Developer ACE Team's latest is, of course, a spiritual successor to the 1990 SNES game, yet it somehow feels even more dated than the original. While *ActRaiser* was innovative and experimental, *SolSeraph* is a poor mimicry that fails to muster the same creative spirit.

Structurally, both games are nigh-on identical, with isometric city-building sections interspersed with 2D action platforming. Neither of *SolSeraph*'s two halves is any good, but the side-scrolling levels are the more egregious of the two by a clear margin. There are basic sword attacks, you can shoot arrows, and there's a block that can also morph into a backdash. It's simple stuff, which isn't inherently bad on its own, but the controls are unresponsive and everything wrapped around these levels is uniformly dull. There's really not much to say about the level design other than you run from left to right and occasionally need to jump. Enemy designs are uninspired too, with bats, blue guys, skeletons, and little deviation from these familiar tropes. Some of the bosses are visually interesting, but rarely fun to fight.

A surprising amount of your time is actually spent waiting for enemies to slowly enter

from either the background or foreground. This encompasses the general flow of each level, as you wait for enemies, hit them a few times, and then repeat. Some enemies enjoy peppering you with arrows from off-screen, while others do everything in their power to stay away from you.

There's nothing enjoyable or satisfying about fighting any of them, and they mostly feel like they were designed just to irritate you. All you need to do is look up the word 'bland' in a thesaurus, and you'll have myriad ways to describe *SolSeraph*'s action platforming.

The city building doesn't fare much better either, although it avoids a lot of the previous frustration by virtue of its simplicity. Each level begins with the same rigmarole as you build houses, farms, and lumber mills to accumulate resources and people. Then you use these assets to construct barracks, archer towers, and other structures that will help to defend the town from an encroaching band of monsters. At this point, it becomes little more than a rudimentary tower defence game, where the easiest way to win is by simply building more stuff.

There's no nuance or strategy to be found here, which is indicative of *SolSeraph* as a whole. It does nothing to evolve or improve on its 30-year-old foundation, instead opting for an insipid replica that lacks any of the same charm or compelling ideas of its inspiration. *ActRaiser* pushed boundaries when it was originally released, but its memories were better off left in the past. ☹️

"It somehow feels even more dated than the original"

▲ It can look good in fleeting moments, but applying the character portraits - reminiscent of 1940s Disney animation - across the whole game would've made for a striking art style.

HIGHLIGHT

The city-building/tower defence half of *SolSeraph* may be lacking in complexity, but at least it performs well on a controller. Constructing buildings and pathways is done with ease, thanks to an intuitive control scheme that disperses any longing for a mouse and keyboard. *SolSeraph* isn't a good game, but it could've been worse.

Info

GENRE
Action, strategy

FORMAT
PS4 (tested) / Switch / PC

DEVELOPER
ACE Team

PUBLISHER
Sega

PRICE
£11.99

RELEASE
Out now

REVIEWED BY
Richard Wakeling

VERDICT

An uninspired and sleep-inducingly bland replica of a much better game.

20%

Rated
Review



▼ You'll see this screen. A lot.

HIGHLIGHT

I'm a sucker for steampunk, and *They Are Billions* manages to do it in a way that doesn't feel over the top or forced. There's a nice mix of the antique, the modern, and the fantastical, from arrows and ballistae to steam-powered mech suits and Tesla coils. Steampunk is so rarely done well in games, but *TAB* is a great example of it.

Review

▲ Might not quite be 'billions', but it's enough of them to be a problem.



They Are Billions

Billions as in your loss tally, not how many zombies there are

Info

GENRE
RTS
FORMAT
PC (tested) /
XBO / PS4
DEVELOPER
Numantian
Games
PUBLISHER
Numantian
Games,
BlitWorks
PRICE
£24.99
RELEASE
Out now

REVIEWED BY
Joe Parlock

VERDICT

Being overrun by walking corpses hungry for your spinal fluid has never been this much fun.

88%

Almost 80 hours into *They Are Billions*, I haven't managed to win once. Whether it's a single tile of undefended perimeter, or an overwhelming horde ripping through my colony's defences, my villagers are wiped out every single time. But that's fantastic, because it gives me that tiny bit of an excuse I need to spend another night fawning over Numantian Games' imposing RTS.

Set in a steampunk future that has seen the Empire brought down by a zombie plague, it's up to you to create new outposts and fight back the hordes of undead to reclaim the world. Starting from humble archers and wooden fences, before long your huts have expanded into a sprawling city of mechanical defences, augmented soldiers, and... oh. The zombies found a gap in your fence. Game over.

The zombies are more of a natural disaster than a usual RTS enemy. The game is split into days, with ever-increasing numbers of zombies flooding in from the edges of the map every few days in an avalanche of putrid flesh. Starting off as maybe a dozen at most, by the end every available pixel will be taken up by the hordes in battles that are up there with *Total War* when it comes to sheer scale. The undead are evil, annoying, and can bring a seemingly perfect colony to its knees in the blink of an eye, but

the sheer spectacle of watching a turret mow down thousands of zombies before eventually, inevitably being infected with the plague is amazing.

It's the loss that keeps *They Are Billions* going, though. The basics of running a colony are simple enough, but identifying the flaws in your design and patching them up for the next run takes a much greater understanding of the game's quirks than you'd expect. For example, archers may seem like weak starter units at first, but eventually, you realise they're quieter than their stronger Gunner counterparts. You may notice

this on your 19th run, and on the 20th you'll have a large strike team of archers clearing the map of any environmental zombies before the waves come crashing at your barriers. It's that sense

of experimentation and discovery that turns what could've been a repetitive slog into something that burrows deep into your brain and refuses to go away, long after the game's been closed.

They Are Billions is, without a doubt, the best RTS released this generation (which admittedly is a very small pool of competitors, but still). The aesthetic is great, the campaign is long and challenging, the survival mode is ludicrously compelling, and thanks to a lengthy early access period, it's all done with little in the way of bugs or instabilities, even with a billion undead hands clawing at your defences. 🗿

"They Are Billions is, without a doubt, the best RTS released this generation"



Rated
Review

Break Ultron's First

HIGHLIGHT

Look around certain levels hard enough and you'll come across what are known as Infinity Rifts – better explained as optional side challenges with specific completion requirements. Beating them is the only way to unlock four new playable characters, as well as unlockable costumes for the entire Marvel roster.

Marvel Ultimate Alliance 3: The Black Order

Keep on mashing the A button? I can do this all day

Marvel *Ultimate Alliance 3* is the latest title hoping to keep the comic (and movie) giant's output on top, following last year's *Spider-Man* for PS4 and the continuing successes of the *Lego* tie-ins. Opting for the satisfyingly simple (yet addictive) template of classic beat-'em-ups, *MUA3* doesn't throw out the rule book of Raven Software's 2006 original. Rather, new developer Team Ninja does well to replicate the button-mashing gratification of prior entries – and it still sees you guide a four-man superhero squad through a gauntlet of colourful stages and bosses.

The difference this time is a much greater emphasis on upgrading. Rather than just unlocking new moves as your favourite members of the *X-Men*, *Inhumans*, and *Avengers*, you're also constantly locking in buffs used to boost these various attributes further. From collectable compounds called ISO-8s that make characters stronger to the Lab which enhances the performance of your whole team, it's micromanagement to a ridiculous degree – yet worth it for when you feel its effects during battle.

Speaking of battles, *MUA3* features the same mix of switching between heroes, using their unique powers and having them combine as before. Now, however, doing the latter is highly encouraged thanks to the new Synergy Attack system. Ever wanted to know how Miles Morales' venom strike would work in tandem with Cap's shield throw? Then engage a combo to perform

a synergy attack and break down an enemy's stagger meter.

In terms of levels, it'll please comic book aficionados to know that almost every corner of the Marvel universe has been catered for. After all, the basic story sees you on the hunt for the infinity stones that have been scattered around, giving you good cause to venture through such beloved locations as Asgard, Avengers Tower, and even New York's Shadowland.

All this you do as your ideal line-up from a roster of 36 Marvel superheroes is unlocked at a regular pace throughout the campaign. Of course, sporadically dishing out team members means many will stay weaker than those in constant use, but replaying levels works fine to better power up any early additions. This is the type of game where grinding is just as thrilling as reaching the next cutscene.

Unfortunately, a couple of small niggles haven't been done away with in the franchise's decade away, the biggest culprit being the camera. From getting caught in the environment to not giving me sight of my squad, so many times was I constantly battling it alongside the enemies. Then there's the case of the block button being the same used to dodge, but it's less of a pest.

Marvel Ultimate Alliance 3 is a rambunctious return for the cult classic series, and takes an "If it ain't broke, don't fix it" approach to the formula in some areas while deepening it in others. Despite some flaws, this is a welcome celebration of the Marvel universe worth wasting hours in. 🗨

Info

GENRE
Action RPG
FORMAT
Switch (tested)
DEVELOPER
Team Ninja
PUBLISHER
Nintendo
PRICE
£49.99
RELEASE
Out now

REVIEWED BY
Aaron Potter

VERDICT

Keeping it simple has always been satisfying, and it's still true in this long-awaited sequel.

68%

HIGHLIGHT

Hamsterdam's boss battles break up the repetition of combat with *WarioWare*-style scraps that change on every occasion. They feel well-paced throughout and use different control schemes, ensuring you get a much-needed breather from the usual frenetic flailing.

✓ Hamsterdam's map screen is detailed but evokes a mobile style that sticks out on the Switch.



Review

Hamsterdam

Crouching vermin, hidden hammy

▲ You can activate rush attacks by pulling on both triggers and dragging the icon to dispatch difficult vermin.

Info

GENRE
Arcade

FORMAT
Switch (tested)
PC / Android
/ iOS

DEVELOPER
Muse Games

PUBLISHER
Muse Games

PRICE
£7.99

RELEASE
Out now

REVIEWED BY
Jordan Oloman

VERDICT

Hamsterdam's fun factor makes it a steal, considering the low price tag – and despite the mobile architecture undermining it.

74%

Plenty of games get by on the virtue of how they feel to play, especially when the unique manner of engagement evades gimmick-territory. *Hamsterdam's* haptic motion-controlled combat is special in this sense – it reminds me of *Samba de Amigo's* magnificent maraca-shaking or some of the more physically involved games on the Nintendo Wii.

You play as a hard-as-nails hammy called Pimm in this wonderfully weird homage to *The Wire*, which follows the plucky nibbler through the mean streets of Hamsterdam as they unravel a vermin-led criminal organisation with a flurry of chaotic kung-fu. It's hard not to root for the smirking little guy as he wallops bruiser bunnies twice his size. It's a vermin management puzzle at best, and an RSI speedrun at its worst.

Jolting your hands to punch, with perfect rhythmic timing necessary, causes you to land combos on the legions of pests surrounding our protagonist. Facing off against The Rodent Squad, meanwhile, requires counters straight out of the page of the *Arkham* playbook, another horizontal gesture that's also fun to pull off. It's all aided by some fluid animations which make the combat feel like a dreamy dance of death. And, as you progress through the missions, you're given seeds that you can trade-in for oven mitts, hipster glasses, and other cute cosmetics that offer upgrades and protections against enemies.

Bonus objectives provide some much-needed replay value, but the criteria is brutal. It's hard to

tell when some vermin are bracing to attack as the telltale glint blends into the fur. It draws a thin line between being unfair and exciting if you're trying to tick every box, but you're never actually in danger of failing missions as long as you're OK with flailing around like a fool in public. Given the Switch is yet to take advantage of its motion controls, *Hamsterdam* is a breath of fresh air in that respect.

The story is told by wordless slide show storyboards; a waste, given that the chibi-HBO premise is so unique. Even just a smattering of vocal talent would have elevated *Hamsterdam's* messaging – an exciting narrative to explain, but not a fun one to actually play.

The main thing that drags *Hamsterdam* down is the mobile game architecture surrounding it. It feels overbearing, tacky, and out of place on Switch – menus are sluggish and unintuitive, load times are slow, and missions are short and only just give enough of a taste to keep you going. There's a *Candy Crush*-style map and star rating at the end of each scrap too, with boring endless runner minigames breaking up the repetition. It's unnecessary, especially when Muse Games doesn't seem to be peddling any in-game currency.

Still, at such a low price point this is a worthwhile pick-up for your indie collection. The game oozes passion, and while it might have worked better if it sidestepped some of the trappings of modern mobile games, it shines as one of the best examples of the capabilities of the Switch's motion controls, which have so far been sorely under-used. 🐹



Settled habits

Demanding more supplies from The Settlers II

If this page of the mag was called 'extra review', or 'stuff we forgot to talk about but probably should have', you wouldn't be seeing something like this here.

Blue Byte's classic game of strategy and zero haste saw a 10th-anniversary remake – which is the version I'm playing – released back in 2006. That's 13 years ago, time fans. And yet here I am, still prodding away at my little proto-city in *The Settlers II* and trying to understand why my iron smelters refuse to smelt any iron even though they have everything they need to *smelt iron*.

What is it that keeps the experience so captivating, even 20-odd years after the original release? It's hard to outright state the indirect method of control as a positive, especially when it comes to combat, but it really is that hands-off management of things that makes it so engaging. Rather than just selecting your units and telling them to grab those berries, join in construction of that hut, and act as cannon fodder on that flank – akin to something like *Age of Empires* – in *The Settlers II* you give broad, village-planning instructions.

Build a woodcutter, build a forester so there are more trees once the cutter has cut them down, build a sawmill to process the felled logs into usable planks. A quarry for rocks. A farm for wheat (and a mill for flour, a bakery for bread). It's straightforward supply and demand, and in this respect shows *The Settlers II* – and the series as a whole – for what it is: an exercise in crafted

automation. The same can be said of other city-builders, of course – still, *The Settlers II* and its ilk have more in common with something like *Factorio* or *SpaceChem*.

Your work isn't focused on the end game; you aren't looking to tank rush an unseen enemy, nor does your actions per minute measurement matter in the slightest. You're focused on the initial setup – getting your settlement into some kind of workable shape, making sure the basics are provided and your people aren't overstretched, and making furtive plans towards expansion. Once your feet are steady and

planted on this new land, it becomes about that expansion – and with it, it becomes about efficiency. Anyone can build myriad paths winding about the place with all raw materials

technically ticked off for the supply line, but only those paying attention – those really focused – will be able to make A-to-B-to-C run like tiny people-powered clockwork.

There's genuine pleasure to be had in just making stuff work, and there's even more to be had in making it work *properly*. *The Settlers II* handled it so well back in 1996, it's easy to see why it got a 10th-anniversary version, and another re-release (of the original version) in 2018. Some games let you live power fantasies, some make you question the nature of the form, while others offer up basic blockbuster treats. *The Settlers II*, meanwhile, lets you show off your competence. Bliss. 🍷

“There's pleasure to be had in just making stuff work”

Wireframe Recommends



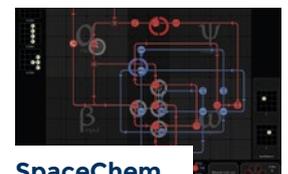
Cities: Skylines

PC / Mac / Linux / Consoles
Stealing the crown from *Sim City* as the best game focused on making a metropolis that doesn't immediately burn down, *Skylines* is a modern masterpiece.



Factorio

PC / Mac / Linux
This long-gestating early access title has been brilliant fun to play for years now, so don't let the unfinished tag put you off. Automating factories *should not* be this captivating.



SpaceChem

PC / Mac / Linux / Mobile
Less open-ended than other titles, *SpaceChem* instead presents you with a series of smaller, contained puzzles that need to be solved using automation. It's another great of the efficiency genre.



Tekken

It never gets old, being told you're great

NAMCO / 1994-ON / ARCADE, PLAYSTATION, MULTI

From day one, *Tekken* has tried to make sure you feel good about winning. Not just battering your opponent into unconsciousness – wow, games really do desensitise you – but by acknowledging out loud when you only just win a fight. I'm not claiming Namco's classic series is the only one to have done it – I've fond memories of *Street Fighter* acknowledging a cheesy victory with a hunk of fromage – but *Tekken* has a special place in my heart because of one word: *great*.

You're there, one-on-one with your opponent, thwacking each other in the face for the chance to reign victorious in the Iron Fist Tournament. It's not going so well, with said opponent doing more thwacking than you. You're on the cusp of losing, one more superpowered kick to the chin and you're out forever – you have to fight back. Block, reverse, throw, dodge, jab-jab-jab – slowly you turn it around before hitting King's massively overpowered (in the early games) Frankenstein for the victory. And there it is, the announcer barking for all gathered to hear: "*Great*."

It's not triumphant, it's not bathed with exclamatory passion or squawked like a YouTuber saying hello. It's just stated. A fact to be observed, entered for the records. What you just did then was great. You nearly lost, but you didn't, and that is great. You performed great, and the faceless voice just can't keep himself confined to calling round numbers and when someone has been knocked out – what you did was so good he had to tell you about it. *Great*.

In the world of fighting games – and I profess to be no expert here – there's a focus on the tightness of competition. Close bouts, epic comebacks, unexpected victories – they all weave a tapestry and tell a story in microcosm. Snatching victory from the jaws of defeat – and vice versa – is a celebrated outcome, one that highlights an individual's skill just as much as it shows the shortcomings of a felled opponent. I doubt that much thought went into *Tekken's* design, beyond a thought process of 'we should add something in to acknowledge close-fought bouts', but it still applies. It's not just the announcer saying it – we're all thinking it: when a victory comes with just a sliver of health remaining, when all hope seems lost, it really is *great*.

For some, the announcer's finest moment in any *Tekken* game was his similarly-intoned call of '*Chicken*' when picking up a roasted carcass to supplement your health in the third game's Tekken Force Mode. These people wouldn't be wrong per se, but the call of the clucker for me will never carry the weight – the importance – of the round-ending, battle-winning *great*.

In victory, it's a small boost to your self-esteem; your efforts recognised, as well as an audio highlight of the fact that you had so little health left. In defeat, it's a slap in the face, as though our announcer is stating the fact of what you resolutely are not. *Great*. What a word. What a call. What a way to end a round. 🗣️

"What you did was so good he had to tell you about it. *Great*"

Next Issue

ON SALE 29 AUG

Don't miss out!
Subscribe
PG54

R-TYPE FINAL 2

A classic shoot-'em-up
makes an explosive return

Also

- > The FMV genre's surprising revival
- > Pixel art, the Sensible Software way
- > A trip to the future with Neo Cab
- > Turning personal experiences into games

Editorial

Editor

Ryan Lambie

Email ryan.lambie@raspberrypi.org

Features Editor

Ian Dransfield

Email ian.dransfield@raspberrypi.org

Sub Editors

David Higgs & Vel Ilıc

Design

criticalmedia.co.uk

Head of Design

Lee Allen

Designer

Harriet Knight

Contributors

Jon Bailes, Rik Cross, Kieran D'Archambaud, Konstantinos Dimopoulos, Will Luton, Sean Martin, Graeme Mason, Steve McNeil, Jordan Oloman, Joe Parlock, Aaron Potter, Reid Schneider, Olly Smith, Char Stiles, Richard Wakeling

Publishing

Publishing Director

Russell Barnes

Email russell@raspberrypi.org

Director of Communications

Liz Upton

CEO

Eben Upton

Advertising

Commercial Manager

Charlie Milligan

Email charlotte.milligan@raspberrypi.org

Tel +44 (0)7725 36887

Distribution

Seymour Distribution Ltd

2 East Poultry Ave, London EC1A 9PT

Tel +44 (0)207 429 4000

Subscriptions

Unit 6, The Enterprise Centre, Kelvin Lane, Manor Royal, Crawley, West Sussex, RH10 9PE

To subscribe

Call 01293 312192 or visit wfmag.cc/subscribe

Subscription queries

wireframe@subscriptionhelpline.co.uk



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

Wireframe magazine is published by Raspberry Pi (Trading) Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS. The publisher, editor, and contributors accept no responsibility in respect of any omissions or errors relating to goods, products or services referred to or advertised in the magazine. Except where otherwise noted, content in this magazine is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0) license. ISSN: 2631-6722 (print), 2631-6730 (online).



UNLOCK YOUR GAME



JOIN THE PRO SQUAD

1ms

144Hz

Free Sync



RED EAGLE™

GB260HSU | GB270HSU | GB2760QSU



GET IN THE GAME

1ms
75Hz



Free Sync



BLACK HAWK™

G2288HS | G2530HSU | G2730HSU | GB2530HSU | GB2730HSU



IMMERSE YOURSELF IN THE GAME

4K



Free Sync



GOLD PHOENIX™

GB2888UHSU



ENTER A NEW DIMENSION

2560
x
1440

1ms
75Hz

Free Sync



SILVER CROW™

GB2783QSU