

£3

ALL FORMATS

Wireframe

LIFTING THE LID ON VIDEO GAMES

JOURNEY TO THE SAVAGE PLANET

A SLAPSTICK SPACE ADVENTURE FROM THE DIRECTOR OF FAR CRY 4

VITA FOREVER

HOW HOMEBREW HAS KEPT SONY'S HANDHELD ALIVE

FABLE

THE RPG SERIES' EPIC MAKING-OF STORY

EASTWARD

EXPLORING A GORGEOUS TOP-DOWN ZELDA-LIKE

Issue 27 £3

wfmag.cc



9 772631 672000

JOIN THE PRO SQUAD!



G-MASTER™
MONITORS 4 GAMERS
by **iiyama**

RED EAGLE™

GB2560HSU¹ | GB2760HSU¹ | GB2760QSU²



Panel	TN LED / 1920x1080 ¹ , TN LED / 2560x1440 ²
Response time	1 ms, 144Hz, FreeSync™
Features	OverDrive, Black Tuner, Blue Light Reducer, Predefined and Custom Gaming Modes
Inputs	DVI-D ² , HDMI, DisplayPort, USB
Audio	speakers and headphone connector
Height adjustment	13 cm
Design	edge-to-edge, height adjustable stand with PIVOT



gmaster.iiyama.com

Virtual reality dogfights, robo dancing, and real-world convenience

My plane cuts through a sky the colour of a television screen tuned to a dead channel. I glance over my right shoulder and see a bogie approaching. I yank the yoke and spin wildly, losing all sense of orientation. My goggles weigh heavy, and my brow drips sweat. Rat-a-tat-tat. I've been hit. Smoke gushes from my engine, and there's a grotesque smell. Not burning. What is that? It's a fart.

My first experience of virtual reality, a Kickstarted Oculus Rift DK1, was punctuated by the uncontrolled sphincter of an intern. His warm flatus left a distinct impression of our unique vulnerability inside VR. But today, over six years later, I'm playing a VR game again – and I'm dancing with a robot. I pull close my mechanical partner, then pop him back, and release the grip of my left hand. He spins away from me, his CRT eyes meet mine, and he smiles. My cheeks rub the edge of the headset as I start to speak excitedly.

"You need to see this," I call to my real-life partner upstairs. "It's a game-changer..."

I hear the creak of the landing. Did she hear me? Is she coming? Is she in the room? I swing the robot's hand back and forth as I drop low. He follows me. "Are you here?" I call out again. "You need to see this. It's a game-changer."

I feel like I'm on the last week of *Strictly*.

"What are you doing?" my partner says as she enters the room. "Why's the sofa pushed up against the wall?"

I lift the Oculus Quest from my head and strap her in. She complains that everything's blurry. We adjust the lens distance and shift the headset. Then she's dancing. Much better than me, but then, she is professionally trained. Her lips curl into a smile. She's loving it. "Wow! That's so cool." Then, mere minutes later, she says, "Can you help me with the sofa?"

The Oculus Quest, those in the know about VR tell me, is a game-changer. It's the first time a decent VR



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: Making Money From Games You Give Away*, and has worked with Sega, Rovio, and Jagex. He is also an avid retro games and pinball player.

experience has been truly wireless. Where frame rate and tech specs are such that the experience is smooth. And it is smooth – the experience is brilliant.

But those in the know about VR are those who took that early rush of venture capital cash, predicated off the salesmanship of a young, barefoot, alt-right inventor. They, like Mark Zuckerberg, see how undeniably cool this technology is. They *believe* in this technology and they want it to succeed.

But consumer interest has been lukewarm, with the PSVR now likely selling five, maybe, six million units at most. These numbers betray the complete lack of cultural impact PSVR, Oculus, and the also-rans have had. Headsets gather dust, with PSVRs cluttering eBay and CEX's glass shelves, slashed below £100. This future isn't the one Ernest Cline wrote about in *Ready Player One*. Nobody wants OASIS.

While it's easy to point at the technical limits of the Quest – the light leaking, the weight, jaggy rendering – the technology has surpassed the point where it's merely good enough. The problem with VR is not the technology or ergonomics, but a fundamental usage issue.

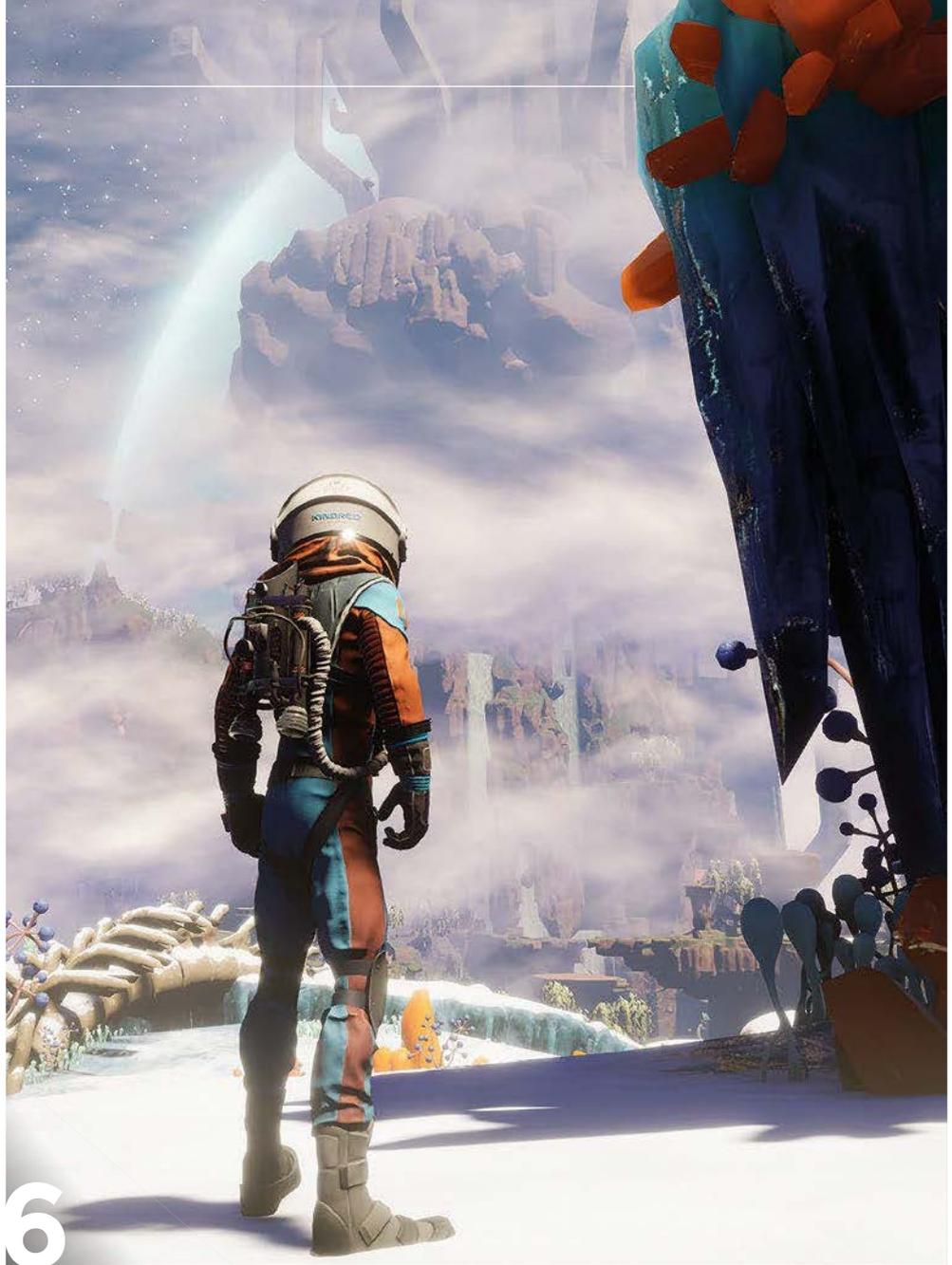
VR bucks a trend not just in technology but in society over the last century – VR is a technology that is less convenient than its incumbents. As devices become more immediate and physically shareable, VR is slow, cumbersome, and isolating. As we increasingly use more screens to tackle more tasks, VR is devoutly single-screen monotasking. Strapping on goggles to stand, duck, and wave our halo wands is immersive, but it's a much more arduous way of wizarding than sitting, twiddling our thumbs. Meanwhile, making a brew or tweeting our head-composed political hot take means completely unstrapping.

Is the VR experience worth the inconvenience? Is a twirl with Johnny 5 worth the isolation and vulnerability of our nostrils? My sofa remains four feet from the TV. 🐼

#27



Contents



6

Attract mode

Interface

06. Savage Planet

Typhoon Studios give us a tour of their antic sci-fi world

10. Haven

Loved up on an alien planet in The Game Bakers' latest

12. Eastward

A sumptuous pixel art Zelda-like from Shanghai

16. Incoming

Up to our knees in a Blizzard of game announcements

18. Leaving Albion

We uncover the epic story behind the Fable series

44. Vita forever

The homebrew community keeping the PS Vita alive

48. Lunark

A solo dev's homage to Prince of Persia and Flashback

50. Namco

A slice of arcade history, a selection of fine titles



44



60



18



50

Toolbox

- 28. Design principles**
Breaking the rules to make better video games
- 32. Blender add-ons**
Using Python to speed up your 3D asset creation
- 38. War on piracy**
How Steamworks fends off pirates – with angry fish
- 40. Source Code**
Recreate Konami's Frogger in just a few lines of code

Rated

- 56. John Wick Hex**
Bodies hit the floor in a methodical Reeves-'em-up
- 58. EarthNight**
A stylish roguelike runner has us chasing dragons
- 60. Valfaris**
Blasting alien scum has never felt so rock and roll
- 62. Blasphemous**
A Metroidvania with a side order of dismemberment

WELCOME

I had the article all worked out in my head: introduce a rare piece of development software for a half-forgotten handheld, explore how it all works in a bit more depth, and then relate my flailing attempts to make a game with it all.

The dev software in question is called the WonderWitch, released in the early 2000s for Bandai's Japan-only WonderSwan handheld. As you've probably gathered, the WonderWitch offered a combo of software and hardware that allowed you to connect the handheld to the PC and make homebrew games using C+. In fact, the WonderWitch was so comprehensive that a couple of commercial games were made using it, including a vertical shooter called *Judgement Silversword*.

The WonderWitch is pretty rare nowadays, but I recently managed to pick up a copy for a surprisingly low price from an eBay auction. A few days ago, the parcel came through the door; I opened it, and found a WonderWitch box, but no software or connective dongles inside it. Baffled, I went back to eBay and checked the auction listing again. It turns out I'd misread the description and paid the princely sum of £40 for an empty box.

My hopes of properly exploring a rare piece of game history are, for the time being, on hold. Enjoy the new issue, readers.

Ryan Lambie
Editor



FORBIDDEN PLANET

Typhoon Studios tell us about their slapstick sci-fi adventure, *Journey to the Savage Planet*, and its debt to *Starship Troopers*

The last time we spoke to Typhoon Studios' Alex Hutchinson, it was autumn 2018, and *Journey to the Savage Planet* was still

firmly under wraps. But here we are again a year later, with the Montreal-based studio on the cusp of releasing its debut title: an upbeat action-adventure that feels like a playful riposte to those tough, vast survival games designed to soak up hundreds of hours of your time.

Cast as an intrepid astronaut (one of a pair if you choose to play in co-op mode), it's up to you to explore an alien planet, scanning the flora and fauna, collecting materials to turn into upgrades, completing missions, and seeking out the truth behind a mysterious alien structure looming up on the horizon. You've been despatched by a dubious corporation, Kindred Aerospace ("voted the fourth-best interstellar exploration company by our peers") to size up the planet's suitability for human colonisation. What's surprising about AR-Y 26, as it's called, is how hospitable it first appears.

Where most video game worlds have been designed to be harsh, cruel places, the one in *Savage Planet* is lush and colourful, like a cross between Pandora in

James Cameron's *Avatar* and a psychedelic rock album cover. Most of the alien life forms you encounter in the game's early stages – bird-like critters called Pufferbirds – pose almost no threat at all, and while you'll soon encounter much more aggressive creatures as you progress through the *Metroid*-like game structure, there's an overwhelming sense that the most savage animal on the planet is the gun-wielding one you're controlling.

Then again, this is all part of Typhoon Studios' grand design: *Savage Planet* is intended as a focused, light-hearted game that evokes memories of pulp sci-fi movies and books, while its procession of organic upgrades send you bouncing and swinging around its exotic landscape.

To find out more about *Savage Planet*'s blend of first-person action and manic comedy, we caught up with Hutchinson, the game's creative director, head of art Erick Bilodeau, and executive producer Reid Schneider.

I'm a complete sci-fi nerd, so the whole premise of the game is catnip for me. How did you settle on that premise?

AH: When we started, it felt like a sub-genre that hadn't been explored in games at that moment. But as always



pressure off. If you're making a game where the business is saying they're the fourth best at everything, then players automatically know there's room for error.

It's an exploration game, but there's structure to it as well, isn't there?

AH: We're a small team – there's only 24 of us, so the idea of making something completely open was a bit beyond our reach. So in the end, it has a more *Metroid* lock-and-key structure – you explore an area, and find an obstacle you need to overcome. By scanning it, you'll add a quest, and that'll tell you what you'll need to gather to upgrade your gear so you can access that new area. It opens up a bit like an onion from there.

Is there any procedural generation going on, or is it all hand-built?

AH: In terms of the world, it's all hand-built. We didn't want to have any procedural generation.

EB: We asked ourselves that question at the beginning of the project: whether we wanted to do something procedurally generated, and go wider, but we specifically chose to not go [down] that route, and make everything crafted. We didn't want the player to enter an area where it looks like there was something to explore, but because it's computer-generated, there's no reward at the end.

AH: I think it's difficult to build meaning into discovery in a procedurally generated world – you're going to find something that a computer chose, whereas we wanted the player to find something that a person built. ➔

seems to be the case, by the time you finish it, there are other people around you with something similar – there's *The Outer Worlds* and *Outer Wilds*. They're in a similar tonal area. But we wanted to make something that was positive and upbeat and optimistic. When you go back to that period [of sci-fi], you didn't need to be saving the world; you're out there looking for the future of humanity.

Were you looking at specific books or films as background research?

AH: For me, it was a lifetime of reading science fiction and watching movies. *Starship Troopers* was a touchpoint in terms of a movie that can be serious as well as comedic, without losing its [meaning]. I was thinking of those early sci-fi films like *Forbidden Planet*, and then classic writers like [Roger] Zelazny and early Isaac Asimov.

Did you sit down right at the beginning and say, right, we want this to be a comedic game?

AH: For me, it was using humour to take some of the



^ Everything in *Savage Planet* is likeably gelatinous, from the organic upgrades to the showers of goo you get when shooting an alien.

It feels like a focused game, too. In between huge, 1000-hour titles that take over your life, it's kind of welcome to have games that you can enjoy in shorter sittings.

AH: Yeah. I think it's easier to convince someone to switch from whatever MMO or sports game they're playing. To say, 'Hey, take a break for ten or eleven hours to try our game' seems like something we could actually do.

RS: If you think of a triple-A game as a buffet...

AH: ...we're making a side salad.

RS: A triple-A game has to appeal to everyone. As an example, we knew we were never going to make human AI that was better than what you'd have in *Call of Duty* or *Assassin's Creed*. So we said, 'The only things you interact with are these crazy, weird creatures.' Because then we could let [our designers] have much more creativity...

AH: And we're trying to stand out. We're trying things that other games aren't doing, so we don't have any direct comparison.

RS: Exactly. We didn't want to do a *Call of Duty* clone.

The creatures you're meeting in the early part of the game look quite cute – they're small and almost defenceless.

EB: Yes, that's intentional. We wanted that contrast where you have something cute, and you're doing something horrible to it, because it actually drives emotion, right?

AH: Also, we didn't want it to *just* be a shooter; having creatures around that aren't attacking you on sight was important, so you can enjoy exploring the environment as opposed to just surviving.

EB: And it's also to make you feel a little bit like a dick. [Laughs]



^ Many of the planet's creatures are adorably defenceless. Some can be lured into the teeth of deadly traps to unlock barriers. Others simply run up and down, screaming.

“WHEN WE HAD THE OXYGEN MECHANIC, IT JUST ENDED UP DETRACTING FROM THE EXPERIENCE”

AH: You've arrived on this beautiful world, and you're ruining it!

So in theory, could you complete *Savage Planet* without killing anything?

AH: We had that as a goal early on, but no – there are bosses you have to shoot. We had the goal of having as little shooting as possible, but during playtesting, people were shooting a lot.

Was there a lot of experimentation with the survival element? Some survival games are brutal and unforgiving, but the tone of this isn't harsh, is it?

AH: Definitely not. Early on, we had an oxygen mechanic we were toying around with, but we simplified a lot of things to get it down to an adventure game about jumping and running and solving physical challenges.

RS: When we had the oxygen mechanic, it just ended up detracting from the experience. Players were always so

focused on running out, they ended up rushing through the world.

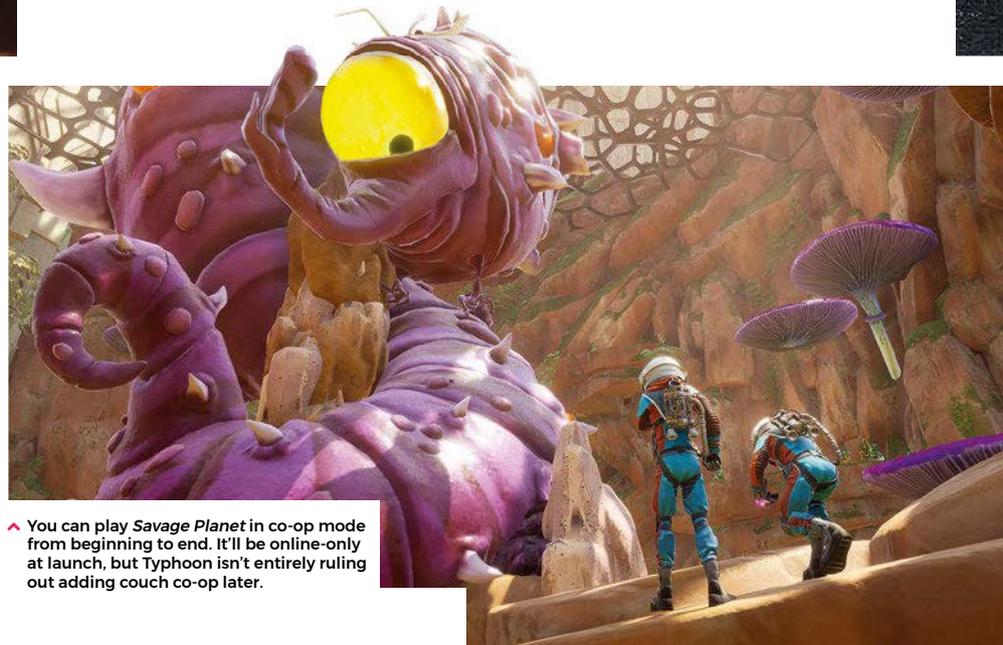
You have live-action adverts that fill in the game world. How did those come about?

AH: Pretty early, we realised it's cheaper to hire actors and build sets than it is to do CG. So it was a bit of a budget concern as much as a creative one. I thought I had this original idea where the TVs would be a window back to Earth, and you'd see human actors. But then someone said, 'That's like *WALL-E*' and I realised it wasn't an original idea at all! [Laughs]

Is it another nod to *Starship Troopers* or *RoboCop*? They used ads to fill in the wider world, too.

AH: In that sense, yes, definitely. *RoboCop* stuck in my head from when I was younger.

EB: Those eighties action movies are a real inspiration for us – [for] the tone of the game at least.



^ You can play *Savage Planet* in co-op mode from beginning to end. It'll be online-only at launch, but Typhoon isn't entirely ruling out adding couch co-op later.

RS: Also, the adverts [in our game] get weirder. As you get further in the game, they get even more whacked-out.

What's been the most challenging part of making the game?

AH: For me, it was [making] a new IP. That was inherently tricky, because everything's possible. Everyone can have an opinion; no decision's set in stone, so everything's a debate. Early on, it's like, 'Why would we have a gun in it at all?' Everything's open; anything can be changed.

EB: Before we settled on a tone, that was really difficult.

In playtesting, where you saw people using the gun more than you'd expected them to – was there a sense that players were telling you what the game was as much as you were creating it yourself?

AH: All game companies play with the idea of consumer testing and putting it in front of people. In reality, most game companies don't get the game finished until five minutes before they release it. Then they do all these tests, and it's too late to change anything. But with this one, because we didn't know how fast we could build it or how much game we could make,

we made a rough draft early, which means we've been able to play it for a long time. So seeing people reacting to things means we can then make decisions to push on something or cut something or change something. Playtesting has been a much bigger part of this game than a lot of the games I've worked on previously.

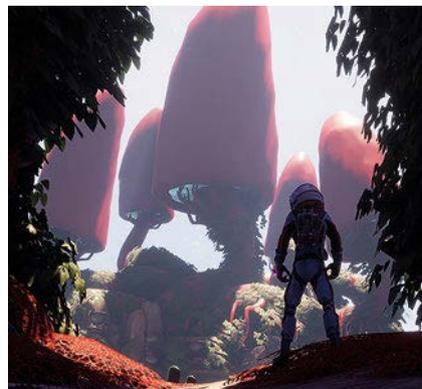
RS: We had a full, rough draft of the game at the end of last year. It was clunky and had lots of rough bits...

AH: It had rough bits, but you could get the context, right?

RS: ...and then we could start saying, 'OK, this is great for more investment of resources,' or 'This is crappy and let's just throw it away.' Generally, with games, you don't have that opportunity.

EB: [In other games] it's a vertical slice that you can play, but you never have the big picture.

^ A more authored experience than something like *No Man's Sky*, *Savage Planet* will have a campaign that lasts around a dozen hours.



RS: You never really know what you've got until beta.

AH: We were actually sitting with a couple of team members who'd done a walkthrough of the game and made all their notes on it. Which is quite unique – it's quite sad that it doesn't often happen, but we could take the game home, play the whole thing, and say, 'The ending sucks,' or 'This bit in the middle makes no sense.'

EB: I think that's one of the smartest things we did.

What do you think your plans are as a studio? Does *Savage Planet* set the tone of the things you'll make in future, or will the next game be wildly different?

AH: I think the answer's both. In my head, if we stay with these pulpy, upbeat games with a hint of satire, then we could make a satirical horror game...

EB: That's a thing we talked about: we can keep the tone, but we can tackle different [themes] with the same weird eighties humour.

RS: Because we've built up some confidence inside the studio, we definitely want to push more in that direction, and give players more opportunities to tell stories and have their own interesting experiences. I don't think we'll ever be the kind of studio that makes *Uncharted*. That's not what we do. 🐵

Journey to the Savage Planet releases for PC, PS4, and XBO on 28 January 2020.



Haven

The Game Bakers' last game was all boss fights. Their next is all about love

Info

GENRE
Adventure / RPG
FORMAT
PC / PS4 / Switch
DEVELOPER
The Game Bakers
PUBLISHER
The Game Bakers
RELEASE
Early 2020

Source's landscape is intentionally sparse, without complex vegetation or architecture to obscure the fluorescent view.



If you played The Game Bakers' ultra-difficult, all-boss fighter *Furi*, a tender romance is probably the last follow-up you'd expect. Then again, it might be exactly what you should expect from a French indie team who boldly claim to "make games like we cook food: with a lot of love," whose Montpellier office is located above a patisserie, and whose creative director Emeric Thoa confesses a passion for romantic comedies.

"*Haven* is a much riskier bet [than *Furi*]," Thoa says, "but the one thing that the two games have in common is the goal of creating something that stands out. A game that you've never experienced before. The story of a couple already in love and fighting for their freedom. An established relationship – I don't think that's ever been done in video games. What love looks like when you're past the early seduction phase. When you can be your true self with one another. Creating an RPG around that idea is pretty outstanding to me."

Haven's protagonists, Yu and Kay, share a comfortable domestic life on Source, a 'lost'

planet they escaped to so they can stay together. Playing as both characters, you'll glide over serene, grassy landscapes to explore the planet and unravel its mysteries, while also learning who Yu and Kay are and how they got here. "I really like when the characters know more than the player at the beginning of a game," Thoa says. "Spending time with them will tell you the backstory, but also make you fall in love with them as you 'make them live' their couple's life."

Their *Persona*-like daily routine will include meals and pillow talk, with dialogue choices shaping the story and sometimes having long-ranging consequences. And what about what happens under the sheets? While sex is an essential part of their love life and may come up in conversation, "there is nothing graphic other than seeing them in underwear," Thoa says. "They have sex, and they don't make a fuss about it, neither do we. Overall, sex in the game is presented in a very natural way."

Yu and Kay's utopia, which Thoa describes as "a fragmented planet of flying islets," is a stylish place of flowing grass, floating rocks, and tumbling waterfalls. This setting will be enhanced by an original soundtrack composed by electronic musician Danger, who also contributed to *Furi*. "He's created a wonderful range of tracks that will carry you from the positive energy of a sunny Sunday morning to the tension of a walk in the mountains [on] a moonless night," says Thoa.

Getting around on Source involves gliding with the help of 'flow boots', sometimes hand-in-hand with your beloved. "It's very relaxing to glide



▲ Thoa says gliding was a late addition; for the first year of development, the team worked on a version that lacked this navigation system.

through the landscape and [that's] one of the strong intentions of the game, making the player feel relaxed," Thoa says. "This art style really came from the game's priorities: simplicity and elegance. No trees or complex vegetation, very little architecture. The serenity of a mountain valley covered in tall grass."

In motion, *Haven's* navigation elicits comparisons to the casual sand surfing in *Journey*, but Thoa says it's more like a kart game. "You hold a button to glide and another one to take tight turns. It's very simple, but it feels a little bit more sporty than *Journey*. The intention was to make the feeling closer to skiing or ice skating. Their flow boots require a bit of skill and energy."

"The danger is not hunger, thirst, or cold – it's a bigger threat than that"

Idyllic as it all sounds, there's trouble in paradise. Thoa is tight-lipped about what trials the couple will face, but their goals will be to stay on Source and to stay together. "Although it's about survival, the gameplay doesn't ask you to really manage their survival needs," he says. "The danger is not hunger, thirst, or cold; it's a bigger threat than that. A threat to their freedom."

The first gameplay trailer hints at machinery breaking down, giant mythical creatures roaming about, and a grassy open world that Yu and Kay will glide through to initiate combat and sweep up the rust that blankets the landscape.

Though it's billed as an adventure/RPG, Thoa says *Haven's* gameplay falls more on the adventure side. "Both elements are important in the game, combat and adventure. But the game experience is more weighted toward story, exploration, and dialogue than combat.

Combat is fun, useful, sometimes required, but not at the core of the experience." When combat does arise, you'll hold down buttons to load actions for Yu and Kay to carry out when you let go, with tactical thinking and good timing

required to coordinate their attacks. It's neither real-time nor turn-based, but something in between, with loose similarities to *Final*

Fantasy's Active Time Battle (ATB) system and the early *Phantasy Star* games.

While it's not technically a co-op game, *Haven* does support local multiplayer, in that a second player can join in any time with a second controller. (On PC, keyboard and mouse will also be supported.) "You can fight together; you can explore together, choose dialogue answers together," Thoa explains. "It's very welcoming for, let's say, a couple who want to share a moment playing together."

Keeping in mind The Game Bakers' mission to make games like they cook food, *Haven's* recipe so far seems to be equal parts *Persona* and *Journey* with a pinch of *Romeo and Juliet* and a dash of *Danger*, but they've no doubt sprinkled their own secret ingredients into the mix.

After two and a half years of development, the gourmands plan to pull this sweet treat out of the oven in early 2020. 🍪



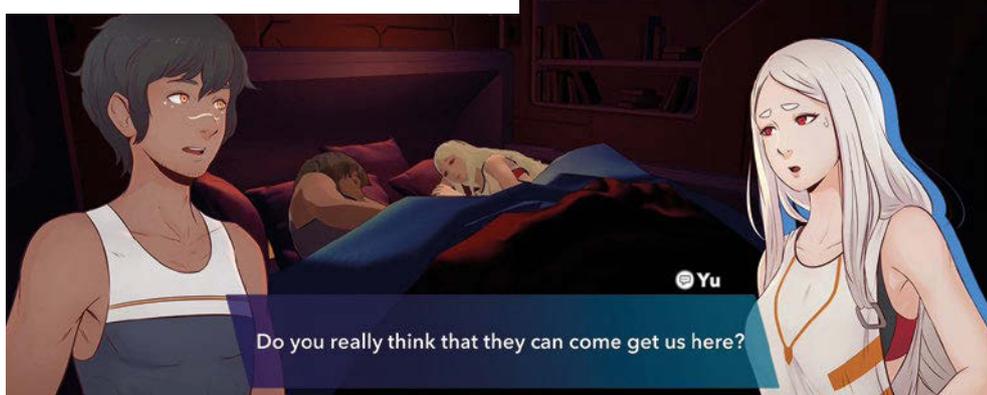
▲ Yu and Kay are a couple rarely seen in games, past the giddiness of first love and together for the long haul.

THE GAME BAKERS' FIRST COURSE

Before the team made *Furi*, they focused on mobile platforms with games like the tactical RPG *Squids*, the retro-inspired brawler *Combo Crew*, and a licensed beat-'em-up based on *Teenage Mutant Ninja Turtles*. "Mobile games feel like another life now," Thoa says. "Initially, we started with mobile because it was easy to produce for, and fun. We loved playing on mobile in 2010. We could make short and simple games. But the market became extremely competitive, and the dominant [free-to-play] business models didn't fit with either our desire or our skills as designers. We really belong to the premium PC/console space, so that move was for the best." If you need a snack to tide you over until *Haven's* launch, *Squids Odyssey* is available now for PC and Switch.



▼ Pierre Corbinais, developer of the Syrian refugee story game *Bury Me, My Love*, wrote much of *Haven's* dialogue.



Do you really think that they can come get us here?



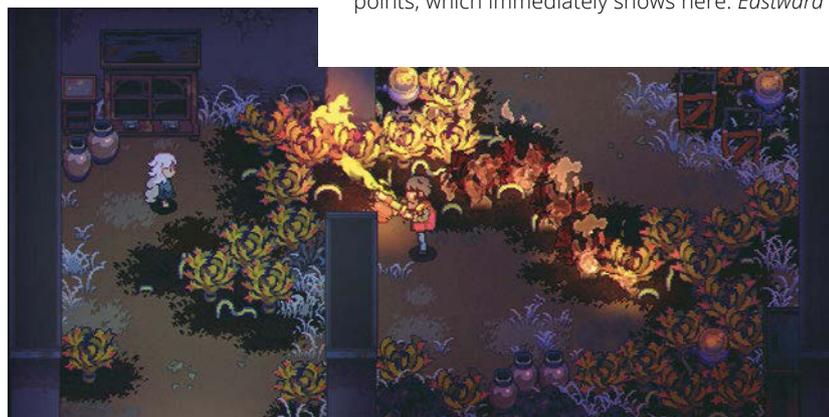
Eastern promise

The end of the world has never looked so good: it's Eastward

Info

GENRE
Zelda-like
FORMAT
PC / Switch
DEVELOPER
Pixpil
PUBLISHER
Chucklefish
RELEASE
2020

▼ A humble frying pan may be John's primary weapon, but we'll pick up other instruments of death - like this flamethrower - later in the game.



If the first thing that caught your eye as you opened these pages was the sumptuous pixel art, then you're not alone. Way back in 2015, programmer Tommo Zhou was so inspired by artist Moran Hong's glorious work that he decided they had to turn it into a video game. Back then, Shanghai-based developer Pixpil was still tiny, comprising Zhou, Hong, and designer Ye Feng. "Moran had a lot of different styles of pixel art," Feng tells us, "and that really inspired Tommo to start making an indie game. That was the very beginning."

Four years on, Pixpil is now a ten-strong team, and *Eastward* has grown into a sumptuous top-down action-adventure that, despite its clear debt to earlier entries in the *Legend of Zelda* series, possesses a style and atmosphere all its own. Feng cites *Mother* (or *EarthBound*, as it's known in the west) and the movies of Japanese animation masters Studio Ghibli as reference points, which immediately shows here: *Eastward*

offers up a lush, post-apocalyptic world full of round, squashy-looking enemies and leafy forests, where cars and lorries quietly rust into the ground. If you've seen Studio Ghibli's classic film *Laputa: Castle in the Sky* or *Nausicaä of the Valley of the Wind*, you'll feel immediately at home in Hong's pixel art impression of a technological world slowly reclaimed by nature.

Where *Eastward* departs from its *Zelda* underpinnings is in its twin character mechanic. You can switch between the two leads - a middle-aged-looking guy named John and a young girl called Sam - with the tap of a trigger. Sam has magic attacks which only work on certain enemies, and by holding X, she can fire a ranged attack that zaps across the screen like a lightning bolt. John's the more battle-hardened of the two, though his weapons are hardly conventional; he wields a gun, but his primary weapon is, curiously, a frying pan, which he can swing for quick attacks, and also charge for a slower but stronger blow.

The duo's weapons are used to solve some gently taxing environmental puzzles, where John's frying pan is used to propel rafts across rivers and streams, while Sam's lightning bolts are used to remove harmful barriers that drift into their path. "We started with the two character designs at the very beginning," Feng explains. "It's always been a debate over how much freedom you want to give to the player when they control the two characters - whether they always move around together, or whether they can [control them] whenever they wish. It's kind of tricky to make it work and smooth for a player who's just picked [the game] up."



▲ *Eastward* may look welcoming, but it offers a tough challenge; the build we played ended with a battle against a particularly aggressive, Terminator-like robot.



▲ Part America, part apocalyptic fantasy: *Eastward's* world building is captivating.

There are elements of the game that are still a work in progress in the build we tried at EGX in October; there are hints of a cooking mechanic, where you'll collect ingredients to make potions, that aren't finished yet. The city overworld we wandered around, and the leafy dungeon we battled our way through, will also be integrated a little differently in the finished game, according to Feng. But the areas we sampled offer a firm yet enjoyable challenge, whether it's using John and Sam's abilities to navigate around a tricky network of hazards, or switching between their frying pan and lightning attacks to take out some of the game's wilier enemies; there are some great level design ideas in here, borne from a lengthy process of iteration and refinement. "The level design has taken the longest time," Feng says. "With level design, you have to go back and forth – to be honest, we're still learning and making progress. It's not easy to make a classic, 2D *Zelda* level that's so smooth, people can [just] pick it up and play it."

"It's not easy to make a classic, 2D *Zelda* level"

Perhaps unsurprisingly, the process of creating all that pixel artwork runs the level design a close second in terms of time and effort: according to Feng, around 70,000 hand-drawn frames of animation have been created for *Eastward* so far.

To help speed up the development process, Zhou set up an asset pipeline that can import layered artwork directly from Aseprite – Hong's pixel art editor of choice – to Zhou's custom-built game engine. Nevertheless, it's easy to see how a comparatively small team have taken almost four years to get *Eastward* to this point: its developers have poured hours of thought

into everything from the movement of grass in the aftermath of an exploding bomb, to the particular way water ripples beneath a raft as John hops aboard.

There's much we don't yet know about *Eastward*; Feng won't be drawn on story specifics, except to say that, while there isn't "a concrete story at the beginning," we'll gradually discover "what's happening in this world" as we explore it. Feng can, however, clear up one small mystery

for us. Why, of all the blunt implements he could have picked up, did John choose a frying pan? Is it a nod to the ever-popular battle royale genre, perhaps? "I think [it's] because a lot of us love cooking!" Feng laughs. "It's true. This was before the battle royale had become popular – it was already a decision that we wanted a frying pan in the game. It's like a Jackie Chan movie – everything can be a weapon, so why not?"

Well, why not indeed. 🍳

▼ As well as all this lovely pixel art, expect a soundtrack by *The Unfinished Swan* composer, Joel Corelitz.



▼ Moran Hong's pixel art has bags of personality; his landscapes feel like the American South, viewed through a Far Eastern lens.



Headlines

from the virtual front



01. Deeper

Yep, we'd forgotten about it too. Announced all the way back in 2013, *Deep Down* was to be Capcom's answer to *Dark Souls* – at least at first glance – and got us all a bit hot under the collar in the early days of the PS4. Then? Years of silence. But it's maybe-possibly-not-quite cancelled yet.

Speaking to Eurogamer, Capcom bod and *Street Fighter* chap extraordinaire Yoshinori Ono said of the MIA project: "The original team is clearly no longer together at this point, but people might have noticed that we've kept the trademark registered, and it's not been completely given up on.

"Every year we examine titles we're doing in future, and we bring up projects to approve and move forward. There's not much I can say about it, but we haven't given up on the title completely."

Deep Down isn't quite dead yet, then. But if the game does appear, it probably won't be until at least the PS5 era...



US Congressman calls out Blizzard for allowing white supremacist group in *WOW*



02



03



03. Can we be terrified yet?

As we march ever forward toward the singularity, what one occurrence will we look back on as the tipping point? For some of us, it will be this, where Google's DeepMind-fuelled AlphaStar AI became a *StarCraft II* Grandmaster.

It wasn't as straightforward as you might expect – the AI had to learn, it didn't just turn up and win. As the DeepMind team wrote: "We chose to address the challenge of *StarCraft* using general-purpose learning methods that are in principle applicable to other complex domains: a multi-agent reinforcement learning algorithm that uses data from both human and agent games within a diverse league of continually adapting strategies and counter-strategies, each represented by deep neural networks.

"We evaluated our agent, AlphaStar, in the full game of *StarCraft II*, through a series of online games against human players. AlphaStar was rated at Grandmaster level for all three *StarCraft* races and above 99.8 percent of officially ranked human players." So... an AI is really good at a wargame then? Cool. Coolcoolcool.



04

04. Outer / Switch / Kerbal / 2

The *Outer Worlds* (review next issue – spoiler: it's good!) will be arriving on Nintendo Switch early in 2020, according to Take-Two CEO, Strauss Zelnick. "It's outperforming our expectations handily," said Zelnick. "We're very happy." We're happy too, Strauss, because it's better than *Fallout 76*.

The second take from the Take-Two news is that *Kerbal Space Program 2* has been pushed back a bit – or potentially more than a bit – with its release date shoved into the vague 'fiscal 2021'. That means it will launch somewhere between 1 April 2020, and 31 March 2021.

"*Kerbal Space Program 2*, the sequel to the beloved original space sim, is now planned for launch on PC, PlayStation 4, and Xbox One, in fiscal 2021, in order to allow more time to make the experience as terrific as possible," said Take-Two president Karl Slatoff. As *Kerbal* is one of the best things to ever happen, we will accept a delay to make the sequel better. Kindness runs through our veins.



05

05. No Rush

Supercell either will or has shut down its new title *Rush Wars*, depending on when you're reading this, with the plan being to close the title down 30 November. This comes just three months after the game launched a beta... and was revealed to be a big pile of garbage, basically. They can't all be *Clash of Clans*.

"When we started developing *Rush*, the team's goal was to make a different type of build and battle game that casual players could enjoy," a post on the Supercell blog read, going on to explain how changes and tweaks had been made after the game received negative feedback – but none of it was enough to fix a fundamentally flawed project.

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



06

06. Rest in peace, SingStar

Sad news for those of us who hosted a lot of parties around the 2010s: Sony has confirmed it will be shutting down the *SingStar* servers on 31 January 2020. Let us all croon in its honour. Of course, the songs you have will still work – just as they do on the PS3 and PS2 versions – it's not *fully* dead, but you won't be able to download new/re-download old songs, there won't be any more updates, the online services will be switched off, and social aspects are to be summarily culled.

A post from the *SingStar* team read: "After 15 incredible years, we have made the difficult decision to shut down the SingStore servers on 31 January 2020... We've loved watching the community grow, and have lots of fond memories from working on SingStar. Your support over the years has meant the world." *They did it theeeeerrrrr, waaaaaaay.*

Boss of Sony says Sony's console, the Sony PlayStation 5, is easiest to code for



IO Interactive: "Next *Hitman* game is well underway"

Wired



Xenonauts 2

It's a case of slowly, slowly, catchy alien-y with Goldhawk Interactive's follow-up to the excellent *Xenonauts*. With air combat implemented in the latest beta, the sequel is starting to resemble a more complete *XCOM*-like, and we're looking forward to the nips, tucks, and tweaks the studio will make to turn this into a more divergent follow-up than the early version hints at. It's already been a while; we can wait a bit longer.



Watch Dogs: Legion

Recently pushed back into the vague 'FY 2021' period, Ubisoft's London-set open-world sequel has seemingly fallen victim to two main factors: the poor performance of *Tom Clancy's Ghost Recon Breakpoint*, and the ambitious nature of *Legion's* 'play as anyone' feature. The former appears to have shaken Ubi somewhat, with the gossip saying the publisher is pushing all of its internal projects a bit harder on the whole 'being unique' thing – surely a net positive.

The latter factor is easier to understand but harder to call. *Legion* allows you to literally play as (almost) any NPC in the game, each with their own attributes and skills. Director Clint Hocking's ambition is clear, but with ambition comes increased complexity – and with that more technical issues to tackle. More time in the oven for this one sounds like it's absolutely the right course of action.

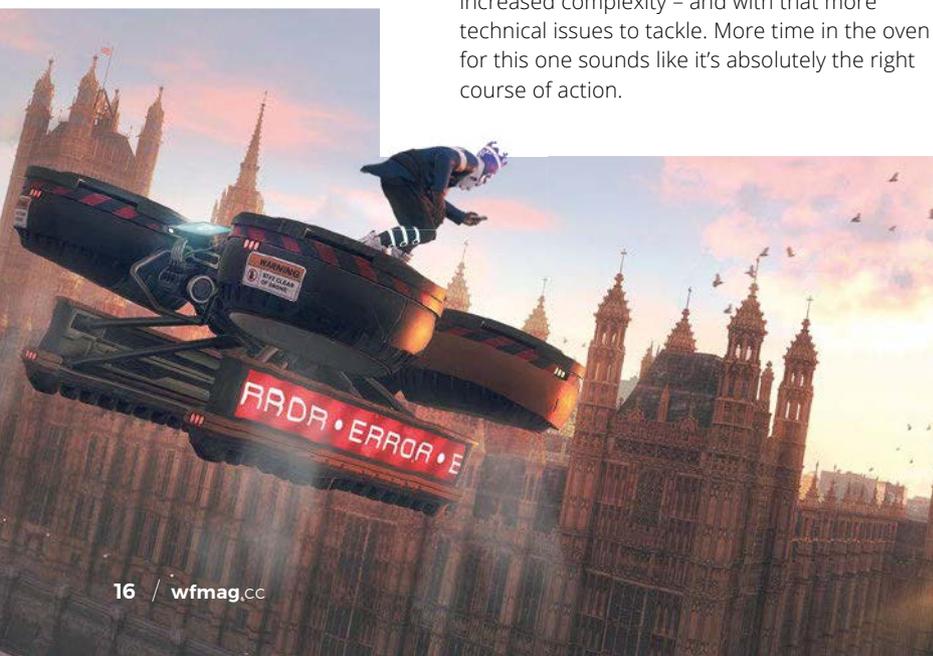
Get in the Car, Loser!

Sometimes our words don't cut it – and this is very much one of those times, for this RPG road trip-'em-up from the co-creator of *Ladykiller in a Bind*: "Will you succeed in your mission to save the world for another thousand years and hit every diner along the road on the way, or will you just be a useless gay disaster?" Well, will you?



Overwatch 2

On one hand, *Overwatch 2* is unexpected, with the original still performing well. On the other hand, it's absolutely expected because *money*. Oh, and technical upgrades, of course. *Overwatch 2* will push hard on the lore side of things, and the story backing the new PvE modes – think *Left 4 Dead* but with fewer zombies – will make a lot of fans of the original very happy.





Diablo IV ↗

A quick journey from rumoured to confirmed might have been quite the rollercoaster for the next in Blizzard's long-running action-RPG series, but there's going to be quite the wait before it actually releases – some guesses (and that's all they are) have *Diablo IV* down for a 2021 launch. It could be later still.

Regardless, this is the one fans wanted to hear – none of that mobile game silliness, a return to the darker atmosphere seen in *Diablo II*, and the ability to play alone or with friends. On that last point, though, you do always have to be online to play, regardless of whether you are on your tod or not. The reason for this is down to *Diablo IV*'s MMO-like style – the main hub world is shared by players, with the ability to engage in PvP combat and other features planned, while dungeons are instanced for individuals or parties. It doesn't sound like a particularly bold step for the series, but it's still one to keep a close eye on over the coming weeks, months, and years.

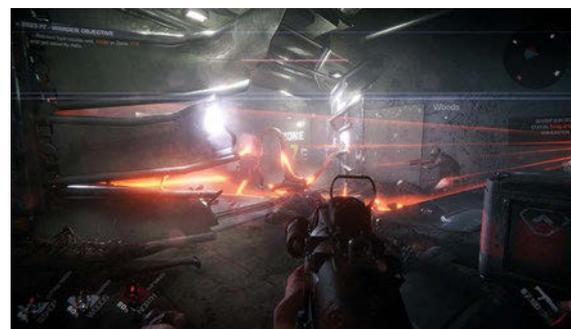
Dungeon Defenders: Awakened ↘

It's been a while since this was Kickstarted, and it's been even longer since the original released way back in 2010. Since then, the world has changed *a lot* – it'll be interesting to see if the mix of multiplayer and dungeon defence still holds up. One major element of note is the game's local four-player mode on Switch, which bolsters the whole effort significantly.



GTFO ↘

Hands held up here – we'd almost forgotten this game even existed until it popped up again around Halloween, so thanks for the reminder, 10 Chambers Collective. A four-player co-op FPS, *GTFO* sees you battling alien enemies, solving puzzles, and enjoying communal brown trousers moments as a team. We played it a while ago, and it was great fun then; with the significant amount of polish it's had since, hopes are high.



Leaving. Albion

The epic making of Fable

We speak to Fable co-creator Dene Carter and
ex-Lionhead staff to hear how Fable came to be,
and their thoughts on Fable IV



WRITTEN BY JACKYARWOOD



Fable's Albion was unlike any of the stereotypical fantasy worlds that dominate the genre. Gone were the armies of orcs and lengthy preambles about ancient dwarven kingdoms, replaced instead with tantalising tales of ghosts, balverines, and chicken chasers. It was a world that felt uniquely British: where its inhabitants spoke in regional dialects, and even the bravest heroes weren't safe from the odd bit of name-calling every now and then.

Years have passed since the last *Fable* entry, and reports are now circulating about a new game in development at Playground Games. What better time, then, to look back at the history of the *Fable* series and the impression that it left on those who made it.

THE BIRTH OF FABLE

The story of *Fable* begins, not at Lionhead, but at the Lionhead satellite studio Big Blue Box. Ian Lovett and the brothers Dene and Simon Carter had started the small studio after leaving Electronic Arts in 1998, and were working on a bold, new multiplayer game called *Wishworld*. But after pitching the game unsuccessfully to numerous publishers, the studio realised they needed a different approach. It was here that Lionhead Studios co-founder Peter Molyneux stepped in and suggested they make an RPG instead. Fed up with the " clichéd" and "pompous" RPGs popular at the time, the Carter brothers quickly agreed.

"We wanted the populace to be ignorant, and the world's past to be mysterious and magical," Dene Carter explains. "We wanted rumour and reckoning to win out over exhaustive histories learned peculiarly well by even the most ignorant peasants in the land."

To say that *Fable* was an ambitious undertaking is an understatement. Not only did the project contain RPG elements such as quests, loot, and

upgradeable abilities, but it also moonlighted as a social sim that let players buy their own home, marry villagers, and decide whether to be good or evil. This ambition resulted in some huge problems for the development team, and eventually the decision was reached to merge Big Blue Box with the rest of Lionhead, after the former company's acquisition. The problems didn't stop there, though.

There would be an unhealthy culture of crunch at Lionhead in order to get *Fable* finished and out the door. Tight deadlines became the norm, with studio members working punishingly long hours to hit important milestones. On top of this, the game went through many drastic redesigns, with features constantly being revised with little time to spare. The prevailing sense at the time was that the project was destined to fail, with none of the pieces clicking together as intended.

"There was a period near the end where I was convinced our game sucked," remembers Dene Carter. "The combat felt terrible. The cutscenes were janky, and the dialogue was spoken with robot voices we'd generated... I was convinced we were sliding into a pit of doom borne of our inexperience in making 3D action games, and that the game was going to be a failure at every level."

"Only a couple of months before the end, a couple of small things happened. Russell Shaw's music went in. The hundreds of lines of actors' dialogue went in. The particle effects went in. Over the course of about six weeks, the lurching, ungainly, vacuous 'thing' we'd been making started to transform into a charming fairytale world."

ACORNS AND ORPHANS

In the first *Fable*, players control the Hero of Oakvale, a young orphan whose village was

Production headaches

Fable's development was renowned for its lack of planning. "They'd come up with things on the spot, so there were a lot of technical problems that appeared," says Charlton Edwards. "I remember they were making these amazing art assets of 500 polygon daisies and then they'd suddenly realise that the Xbox didn't have as much memory to play with, so they had to chop up the open world into regions and alter the story and the gameplay to fit."



▲ Meat Loaf doesn't make games, but if he did, they'd look like this.



^ Despite Peter Molyneux's teases, the trees didn't grow over time in *Fable*.

attacked by raiders and a masked figure known as Jack of Blades. Adopted by the Heroes' Guild, the player must hone their skills and complete quests, before deciding whether to save Albion or seize Jack of Blades' power for themselves.

On *Fable's* release in 2004, response to the game was mostly positive.

Critics praised the game for its unique sense of humour and open-ended gameplay,

but criticised its short story and the absence of certain features that Molyneux had teased prior to launch: the ability to have children, and plant acorns and watch them grow over time.

"Peter was busy with *Black & White* at the time of Acorngate," explains Dene Carter. "All he knew was that we had this engine left over from the *Wishworld* project and that it was capable of morphing in real-time and generating nice forests. He saw no reason why we couldn't extend this morphing to every single piece of plant-life in the game. As far as Simon, Martin Bell [the engine coder], and I were concerned, tree growth was something that would require a real-time update of navigation data, would break our foliage technology, and – as far as we could see – would have no actual gameplay value whatsoever."

Carter refers to Molyneux's habit of announcing unplanned features as "testing ideas on the press," a tactic he had done in the past at his previous studio, Bullfrog. Whenever Molyneux would tease a new idea, panic would set in among the team at home, with everyone attempting to come up with a solution to make it work at short notice.

"Whenever Molyneux teased a new idea, panic set in"

After *Fable* released, the team took a small break and then set to work on an expanded version of the game called *Fable: The Lost Chapters*. This introduced new areas, new quests

to complete, and new customisation options for characters, helping to alleviate some of the frustrations that people had with the length of the original release.

500 YEARS LATER

The development of *Fable II* would be far less hectic than the previous game. Whereas *Fable* had been the culmination of wild experimentation, *Fable II* benefited from a well-planned production schedule and a much larger development team – a result of Lionhead's acquisition by Microsoft.

"On [*Fable*], we would come up with a sort of first draft and then the writers would amend it →

> *Fable* gave players the freedom to choose between becoming the saviour of Albion or a harbinger of doom.



Albion unlocked

Fable II featured more wide-open areas than the first game, with level designers Charlton Edwards and Andrzej Zamoyski making it their mission to move away from the "pretty corridors" of *Fable*. Some inspirations behind *Fable II's* locations were games such as *Shadow of the Colossus* and *The Legend of Zelda*, and the beautiful Surrey countryside that surrounded Lionhead's studio.



^ Acorngate aside, the *Fable* games looked reliably pretty.





^ Children are a wonderful thing. That is, until you have to pay for them with your hard-earned wood chopping money.

Games for change

Fable II was a step forward for queer representation in mainstream games. Players could get married to members of the same sex and the game also featured a quest called 'The Blind Date', about a son coming out to his father. These additions had a profound impact on some players. "A story that will always stick with me is how a quest in *Fable II* gave someone the confidence to tell their parents they were gay," says senior scripter Ted Timmins. "It was at an E3, and it was such a powerful moment, we were both in tears. I could recount hundreds more I've heard from players over the years, but it's proof that video games are powerful – in my mind the most powerful medium there is – and yet it feels like we are still only scratching the surface."

afterwards," says Neil Whitehead, a scripter on the three mainline *Fable* games. "[For *Fable II*], we had writers on-board from the start, and they'd produce briefs, and then we'd have a whole eight-page document about the exact dialogue that needed implementing. It all felt a bit less spontaneous. I think that's maybe what the bigger team brewed."

"It changed a lot," says Dene Carter. "We didn't crunch the team for 18 months at a time. We had a good production team right from the start, rather than having to bring on the utterly transformative Louise Murray [as producer] for the last year or so. We had a lot more confidence that we could actually make this style of game. It was – in a word or two – more fun. We didn't drive anyone mad, and felt like we were actually professionals making a professional thing rather than a bunch of loonies scampering around putting out an increasingly large succession of fires."

In *Fable II*, the land of Albion had changed considerably. Five-hundred years had passed since the events of the first game, and the Heroes' Guild had been reduced to ruins. Dene Carter attributes this jump in time to the influence of the British TV show *Blackadder* and an attempt to prevent over-familiarity. In place of the medieval influence present in the first game, *Fable II* drew inspiration from 17th-century Britain. The player could wield flintlock pistols as opposed to crossbows, and would encounter highwaymen while travelling between locations.

The story also went in a different direction, with a new villain, Lord Lucien Fairfax, taking over

from Jack of Blades. In the game, players would become the Hero of Bowerstone, an orphan whose sibling Lord Fairfax had murdered at the beginning of the game. With the help of the seer Theresa, players needed to track down the remaining heroes in Albion and get revenge on Lord Fairfax.

In typical *Fable* fashion, though, you could also ignore the main quest and take part in other, more trivial activities. Marriage returned in *Fable II*, for example, with players able to buy a house and have children. You could also take on additional jobs for gold: chopping wood, making weapons, and hunting down criminals.

CONCRETE AND DOGS

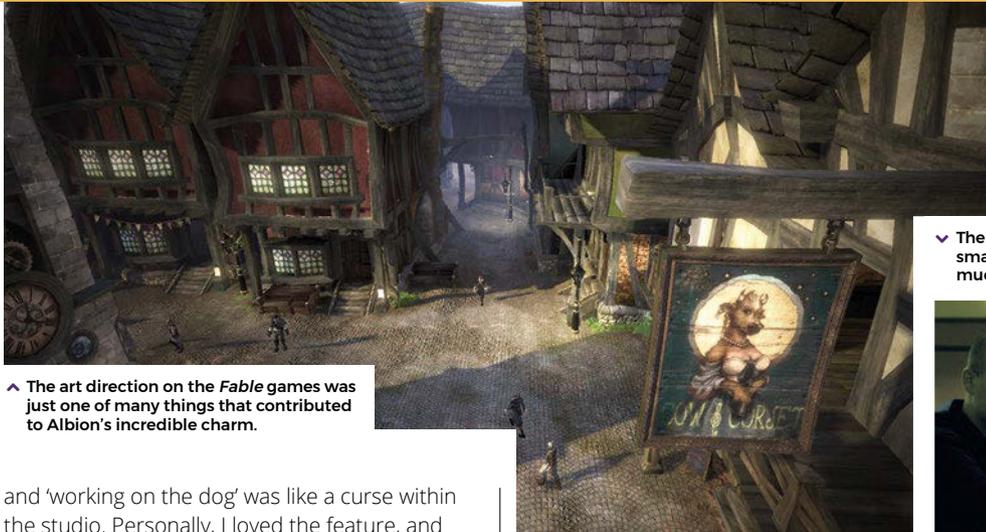
Another huge change was the addition of an AI companion, a dog, that could follow the hero around and bark whenever treasure was nearby. Peter Molyneux personally requested this feature be included, though the reasons for this seem to change depending on who you ask. Some claim it was a result of the death of his real-life dog, that had been shot by a farmer sometime prior to *Fable II*'s development, while others point to a somewhat less tragic influence: a text adventure called *NetHack* that Molyneux was a fan of.

Regardless of its origins, the dog proved to be a nightmare for the programming team, taking a considerable amount of effort to get working correctly.

"We had more confidence that we could actually make this style of game. It was – in a word or two – more fun"

"It went from programmer to programmer to programmer," says Charles Griffiths, a member of the scripting team on *Fable II*. "It became kind of cursed, because some of the programmers that were responsible for it left [or were reassigned]. That poor dog kept getting handed around from owner to owner, with everyone struggling to get it to behave."

"It definitely proved one of the biggest challenges for the programming and animation team," adds Andrzej Zamoyski, a *Fable II* level designer. "AI companions are always a nightmare,



^ The art direction on the *Fable* games was just one of many things that contributed to Albion's incredible charm.

and 'working on the dog' was like a curse within the studio. Personally, I loved the feature, and thought it fitted the world of *Fable* perfectly."

Another issue that people cite with the production was Concrete, the new, all-encompassing engine that Lionhead had built to be used on all their projects at the time. The engine suffered from extremely poor load times and pretty shaky stability, which made some simple tasks a chore. Losing days of work wasn't uncommon, as well as having to wait up to half an hour for groups of assets to delete.

Fable II released in 2008 to critical acclaim. Members of the press praised the game for retaining the charm of the original, while introducing innovative new features, such as the breadcrumb trail. At a launch party in a London nightclub, Molyneux would reveal the review scores to the rest of the team.

"One thing I remember before everything turns to a bit of a blur was Peter stood on a makeshift stage," Ted Timmins, a senior scripter

on *Fable II* and *III*, remembers. "When you've worked so hard on something, it can be hard to see the wood through the trees... so when Peter got to the Edge score, everyone went quiet, and he read out '9/10!'. The whole place erupted, booze went everywhere, the cigars came out – my first-ever cigar – and the party really started."

THERE AND BACK AGAIN

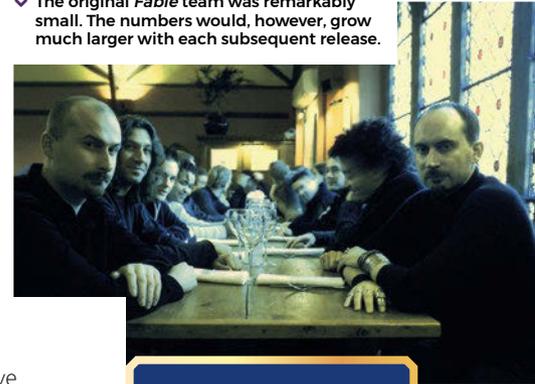
Production on *Fable III* began almost immediately after *Fable II*, with a strict deadline set for only two years in the future. Because of this, and likely the business imperative to produce more than one game with the same engine, a lot of the tools had to be reused on *Fable III*. While some sources say those tools did eventually improve, change was slow to come.

"I poured my soul into *Fable II* and the DLC packs, and when it was over, like much of the team, I was completely burned out," Zamoyski says. "When work began on *Fable III*, and it became clear the tools were not going to get fixed, I had to get out. While continuing to make new levels, I fought against management who refused to let me switch projects. Eventually, I spoke with Peter and with his help managed to escape with my sanity – only to return to *Fable III* one project later."

Despite the looming deadline, *Fable III* introduced a number of significant changes to the *Fable* formula.

The action moved forward another 50 years, with players becoming the son of the hero from the last game. Skill points were axed in favour of a new progression system called the Road to Rule, and the inventory was replaced with the Sanctuary, a three-dimensional area ➔

v The original *Fable* team was remarkably small. The numbers would, however, grow much larger with each subsequent release.

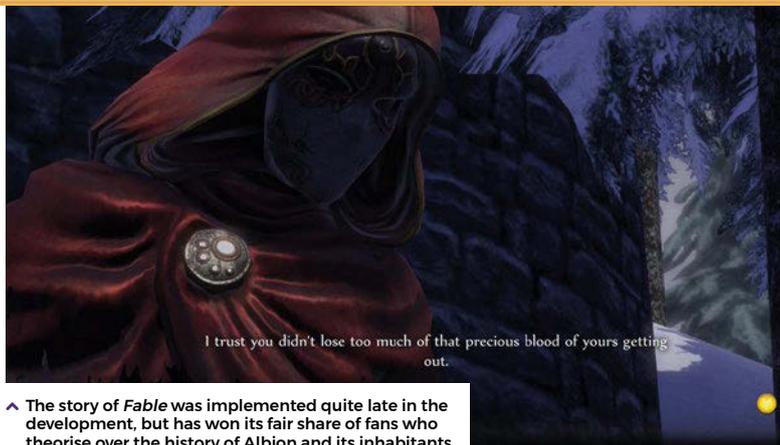


A team effort

It's become common for conversations around *Fable* to be dominated by talk of Lionhead co-founder Peter Molyneux. But discussions with ex-Lionhead staff reveal the huge role that the rest of the team played. "Fable was Dene Carter's baby, and together with Peter, they shaped the core experience," says Andrzej Zamoyski. "The optional quests were the result of other designers expressing themselves within that framework. Some games succeed by reflecting the consistent vision of a single director, whereas in *Fable* you can feel the love and personality that many individuals put into their work. It's what gives the games their signature quirkiness and character."

v Unfortunately, you cannot pet the dog in *Fable II*. You can in *Fable III*, though.





^ The story of *Fable* was implemented quite late in the development, but has won its fair share of fans who theorise over the history of Albion and its inhabitants.



^ I don't understand why he hates me, I'm only a landlord - oh.

The breadcrumb trail

The breadcrumb trail in *Fable II* was one of the game's most celebrated features, but it wasn't a popular mechanic with the team, as Charles Griffiths recalls. "I remember thinking, 'Really? You're going to have a GPS that takes you directly to the point? Where's the exploration?' The team didn't really love it. They were like, 'That's not what people want from an RPG'. But fair is fair, Peter was right. Now that's just industry standard."

where players could change their costumes and equip items.

Says Neil Whitehead: "Where previously there was more scope to experiment and toy around with things, with this one, there was more of a hard date of 'We need to finish this in two years or so.' But despite wanting to finish it in two years, it was 'Here's all the things we're going to change and do differently.' Which yeah, made it a bit of a difficult development."

The Road to Rule, in particular, was a last-minute addition that Molyneux requested, but that production had said would only be possible with another six months. Timmins was the person who stepped forward to implement the mechanic in only three weeks, doing so in return for going to E3 with the rest of the team.

"We made some bad decisions about the importance of RPG [elements] during development, and then suddenly we were just two months out from delivering a gold master and Road to Rule didn't exist," says Timmins. "Quite honestly, we had a *Fable* game with no sense of RPG progression. Cue a 'deal with the devil', a last-minute trip to E3, and a lot of Domino's pizza, and we somehow salvaged something."

“Major aspects of the game kept changing up until the last minute”

In *Fable III*, the main goal was to gain support from allies and overthrow your brother, the tyrannical King Logan. However, the decision was made to have the game continue after Logan had been defeated, with a new threat, the Darkness, introduced so that players would have to raise funds as the king or queen in order to defeat it. This section of the game was originally conceived to be equal in length to what had come before it, with players being able to make important decisions as ruler. But it would be simplified considerably in the finished version due to the time that was available.

"If I recall correctly, you could alter laws and the world would change," says Edwards, who became a level designer for *Fable II* and *III*. "There were more little sliders. Then it became more big alterations. So, I think that's partly how it changed. Then they did that thing where the

more money you make to save the world, the better the type [of ending you'd get], which I thought was terrible, personally. *Fable III* just seemed to make

things a little bit simpler for people to understand rather than this complicated or more reactionary-type game."

"Major aspects of the game kept changing up until the last minute, which meant there was a lot of rework," says Andrea Roberts, an editor at Microsoft, who worked on *Fable III*. "We did over 100 revisions on the script for the intro

^ Goodbye orcs, hello hollow men!



^ The level design team at Lionhead would constantly guess their Edge score. None of them expected to get a nine from the publication.



scene alone. There was so much voiceover to record that we had three sessions running simultaneously every day for six weeks, and we were working late in the night, every night, to have everything ready to record the next day. It was the most intense crunch I've ever done."

Fable III released to positive reviews, but there were some who criticised the game for its clunky inventory system and its departure from the series' role-playing roots. Regardless, it's hard not to look at the project as an impressive feat, especially considering the odds the team were up against.

AN INDELIBLE MARK

"If you asked a lot of people who worked on *Fable III*, I feel confident it would be unanimous that we really needed another six months' dev time," says Timmins. "*Fable* took six-plus years, *Fable II* was four years, and then for *Fable III*, we were suddenly calculating the project in months! There's a great game in there just waiting to be finished, but I remember it being about an 18-month dev cycle which, for a game as ambitious as we were trying to make, didn't turn out that way."

Following *Fable III*, another mainline sequel failed to materialise. First, was there was an on-rails spin-off, *Fable: The Journey* for Kinect, released in 2012. Then there was *Fable Anniversary*, a remake of the first game, in 2014. Lastly, there was *Fable Legends*, a four-vs-one multiplayer game that would eventually be cancelled when Lionhead closed its doors in 2016. Everything went quiet until 2018, when Eurogamer reported a new game was entering development over at British developer, Playground Games.

The response to this news among ex-Lionhead staff is mixed. Some are critical of whether the tone of *Fable* can ever be replicated without the same personnel. Others are excited about the prospect and the attention it may bring to their work. "It's going to look amazing, I'm sure, because the tech is going to be amazing," says Edwards. "But the only thing is, will they try too hard to be *Fable-y* or will they do their own thing and not be *Fable-y*? It's capturing that spirit. The ability to kick a chicken and a farting animation isn't enough. I hope they don't think, 'Oh, that's what *Fable* is; let's stick that in.' As long as it's not a sterile, high-fantasy RPG... I'm excited to see what they come up with."



▲ The creature designs were a *Fable* series highlight.

"*Fable*, to me, is not just a brand or one or two key directors," says Griffiths. "It's also a lot of the senior coders, scripters, artists, etc. Sure, teams can change and staff come and go. But when it's a complete, wholesale replacement of one team with another... Well, at that point it's really just a new game. I wish Playground – or whoever it is making the new game – all the best, but to me, if it's not Lionhead or Big Blue Box, then it's not really *Fable*."

Lionhead left an indelible mark on the British games industry, one that is still felt today in every scrappy indie and creative studio that rose from its ashes. *Fable* was a huge part of that legacy, changing the lives and fortunes of many of those who made it. "*Fable* taught me I was a creative," says Dene Carter. "Creating Albion was a lesson in active dreaming, in taking mood, tone, and atmosphere and crafting it into something the audience can tell is filled with love. *Fable* is more than its mechanics and systems. It is Albion; filled with quirks, mysteries, and a curious, bucolic, silly charm. When reviews mentioned these aspects, it made me feel the struggle had been worthwhile. Making *Fable* transformed me." 🗨️



▲ The love of villagers was hard-earned. Well, you often danced a bit for it.

From the ashes

Deadbeat Heroes: Developed by ex-Lionhead developers Adam Langridge and Imkan Hayati, with a story from *Fable* writer James Leach, *Deadbeat Heroes* is a tribute to the Dreamcast game *Power Stone*, as well as classic British superheroes.

Kynseed: The most *Fable-y* of the games on this list, *Kynseed* is a life sim currently in early access, from ex-*Fable* devs PixelCount Studios. Players can run their own business, go adventuring in the mystical land of Quill, and develop deep relationships with NPCs. The story also revolves around a mystical acorn, a humorous nod to the first *Fable* game.

The Sexy Brutale: A time-loop game set against the backdrop of a masked ball, *The Sexy Brutale* tasks players with reliving the same day and unlocking the mysteries of the party's strange guests. It is out now on Steam and was developed by ex-Lionhead alumni at Cavalier Studios.

Gloomhaven: A digital adaptation of the popular board game of the same name, *Gloomhaven* is a game combining dungeon crawling with tactical role-play. Right now, the game is in Early Access, with plans to implement new mercenaries, enemies, and a longer campaign as part of its road map. Flaming Fowl Studios, a group of Lionhead veterans, are developing the game.

The Law of Boob



LOTTIE BEVAN

Lottie's a producer and co-founder of award-winning narrative microstudio Weather Factory, best known for *Cultist Simulator*. She's one of the youngest female founders in the industry, a current BAFTA Breakthrough Brit, and founder of Coven Club, a women in games support network. She produces, markets, bizzes and arts, and previously worked on *Fallen London*, *Sunless Sea*, *Zubmariner*, and *Sunless Skies* as producer at Failbetter Games.

"You're a rat in a spaceship shooting clothes off megalithic cat-women. I think"

Every week, I receive ICO Partners' Steam newsletter. It's a simple round-up of all the week's latest Steam games, alongside their number of reviews and overall review score. As well as keeping me in the loop about new games I might be interested in, it's also taught me a valuable lesson about nicheness, audience fit, and what can be reliably termed the Law of Boob.

The single most consistent thing about new releases on Steam is that if you make a game with breasts in them, you're looking at a baseline of 80%+ positive reviews. This seems to happen with an uncanny consistency that makes me wonder if I and my heterosexual feminist principles are a blocker for good business decisions. Every week I see games called things like *Hentai Asmodeus* outrank games that are palpably higher quality, and while they might not make the megabucks of lower-rated but higher-grossing offerings, they show an audience resonance that many more successful titles fail to match.

This weekly trend is borne out across Steam. *Monster Hunter: World* is a cool 75%. *Sid Meier's Civilization VI* is 71%. Once phenomenally popular, *PUBG* sits now at 51%. All are apparently less good than *Cultist Simulator*, the game my studio made in our pyjamas for £142k, which is niche and flawed and doesn't have a tutorial and sits at 78%. This doesn't mean triple-A PC games are doomed to middling review scores, of course. *The Witcher 3: Wild Hunt* is 97%, and famously excellent triple-A offerings like *Fallout: New Vegas*, *Batman: Arkham Asylum* and *Divinity: Original Sin 2* are all happily 95%. But it tells me two things: one, that trying to please a wide audience tends to lower your overall review score, and two, we should all be putting colons in our games' titles.

You know what's better than *The Witcher 3*? *Melty's Quest*, sitting pretty at 98% positive, which as far as I can tell is about a queen's journey to find a bra. CD Projekt Red is also put to shame by *DEEP SPACE WAIFU: NEKOMIMI*, another 98% positive bastion of gaming

brilliance where you're a rat in a spaceship shooting clothes off megalithic cat-women. I think. Three other offerings equal *The Witcher 3* and its 800 awards: *NEKOPARA Vol. 3*, 'a heartwarming cat-comedy' set in a sexy patisserie; *LOVE³ -Love Cube-*, which appears to have very little to do with cubes, and *The Ditzzy Demons Are in Love With Me*, which surely needs no explanation. All of these are 97% positive because the people who like anime boobs really like anime boobs, and they're all wonderful examples of product-market fit.

I'm not suggesting that games are at their best when they're thinly-veiled pornography. But I am saying that we indies who don't make boob games should consider what our version of boobs is. What's the one thing about our games that will really resonate with potential players? What will excite people into loving our products, rather than filling the generic game-shaped spaces in people's lives? If you don't have an answer to this, you might be in trouble. Every developer wants their game to be universally loved. Triple-A studios have the money to sometimes – sometimes – make that happen. Indies don't. So don't aim for everyone and end up making Meh Soup. Aim for one particular audience and really knock it out of the park. Or, y'know. Consider hentai. 🍑

▼ Saucy visual novel *The Ditzzy Demons Are in Love With Me*: better than *The Witcher 3*, apparently.



Toolbox

The art, theory, and production of video games

28. Design Principles

Why breaking the rules can make a better game

30. CityCraft

A tour of modern video game metropolises

32. Custom scripts in Blender

Using Python to speed up your workflow

38. A war on piracy

How fish can convert pirates to customers

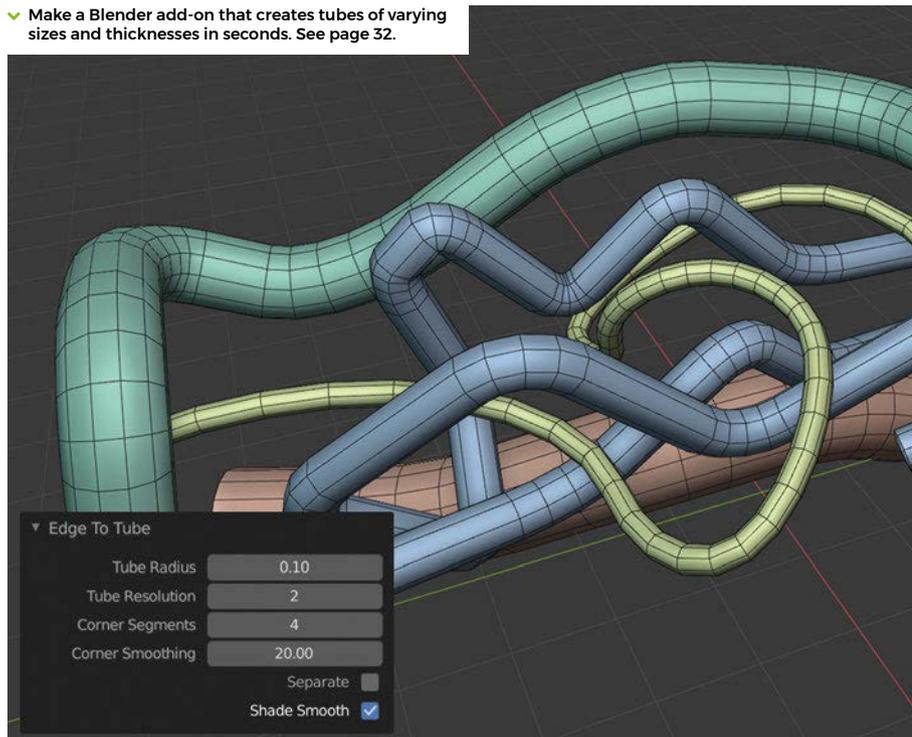
40. Source Code

Code a modern take on Konami's Frogger

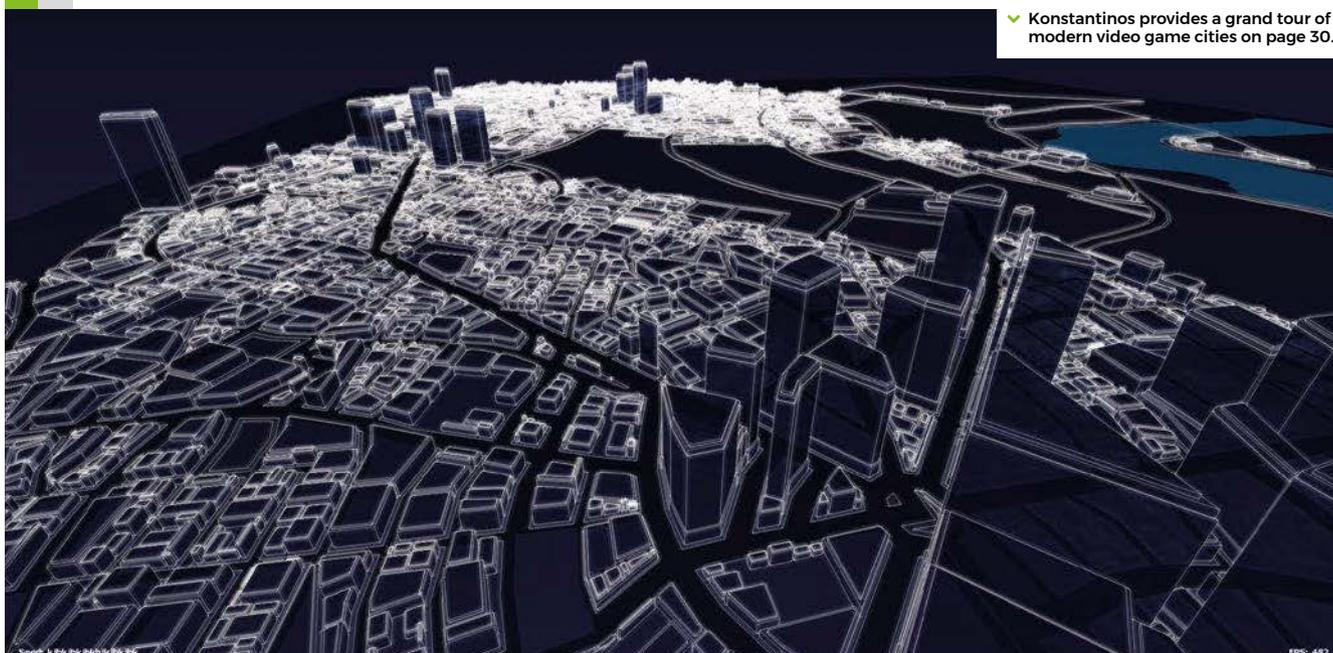
42. Directory

Plug-ins and resources for making game cities

✓ Make a Blender add-on that creates tubes of varying sizes and thicknesses in seconds. See page 32.



✓ Konstantinos provides a grand tour of modern video game cities on page 30.



The principles of game design

Breaking from established rules can often result in better game design, Howard writes



AUTHOR
HOWARD SCOTT WARSHAW

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

✓ The original *Star Castle* coin-op, released in 1980, was a vector shooter in the mould of *Asteroids*, right down to its rotate-and-thrust controls.



When creating a video game, our design choices come from all kinds of places. Some are clear thinking from our heads, some are wish lists from our hearts, and some are pulled randomly from our butts. But every choice we make has an intention, and that intention springs from some expectation about the play experience we seek. There's an old saying in therapy circles: an expectation is a down payment on resentment.

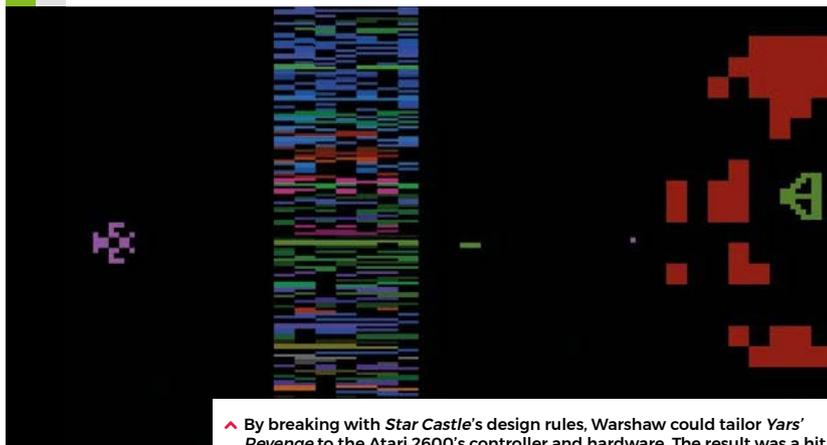
But this isn't therapy, this is video game design. What does resentment mean in this context? It might mean anger, frustration, or bewilderment once we find ourselves stuck with the aftermath of a design choice pursued which didn't pay off. Once we realise our latest design avenue has become a dead-end, do we abandon the entire implementation or just certain parts?

One question is: do we see the difference? Sometimes when I start to throw away the code from a forsaken path, I leave some of it behind. Not intentionally, but simply by oversight or lack of recognition. But that remainder I leave behind can end up limiting my thinking about new directions. An example might help illuminate the darker corners of my thinking here.

On *Yars' Revenge*, much of the basic thinking in the initial implementation was simply employing the specifics of the *Star Castle* coin-op game. I knew from the start I wasn't really following *Star Castle* faithfully since it wouldn't port well to the Atari 2600, but that didn't stop it from shaping my approach. One of the biggest places this showed up was the control scheme. *Star Castle* has the classic *Asteroids* four-button set up: rotate right and left, thrust, and fire. So, I simply did that.

The Atari 2600's four-switch joystick isn't a horrible host for this scheme – right and left are obvious, as is thrust and using the button to fire, of course. This leaves stick-back for another function – in my case, activating the Zorlon cannon. It had everything I needed. So all was well, right? Well... no. Unfortunately, the feel of this control scheme totally sucked. It may have felt fine in *Asteroids*, but it didn't facilitate the kinds of manoeuvring *Yars' Revenge* demanded.

There I sat, thinking: 'The game isn't working and I can't let go of the control scheme.' If I go to a direct motion system – so, pressing up, down, left, or right moves the player's ship in those directions – I lose the ability to activate my main weapon, and then I can't win the game. I felt stuck in bad design. The first version of the *Yars'*



▲ By breaking with *Star Castle*'s design rules, Warshaw could tailor *Yars' Revenge* to the Atari 2600's controller and hardware. The result was a hit.

Revenge controls (a *Star Castle* legacy) wasn't so much a scheme as a strait-jacket.

I was stuck, but only until I released my limited thinking. I had to come back to a fundamental rule of game design: don't be overcommitted to a design choice. That may sound simple, but once you let one aspect go, can you then release the others? Or do they take root and seem to stand on their own, limiting other choices for no good reason? I'd dropped the idea of copying *Star Castle's* gameplay, but not its control scheme – at least, not yet. Perhaps it was time to let go of everything and rethink the game at a broader level. It would mean creating more work, but committing to a game that sucks is design suicide. After a while, self-preservation took over.

CONTROL YOURSELF

I opened my mind to consider new ways to activate the Zorlon cannon. Ultimately, I worked it into the gameplay, and happened on another fundamental rule of game design: whenever you take a button off the controller and work it into the play, the game becomes deeper.

This switch – going to a direct motion stick and using play action to activate the cannon – moved *Yars' Revenge* from a graphically interesting loser to one of the best 2600 games ever. Sometimes the 'rules' that limit us come from inside our heads (as with *Yars' Revenge*), and sometimes they come as well-established rules that everyone blindly accepts, even if they don't really apply to the current circumstances. Case in point: Nolan Bushnell's classic design goal for video games, 'easy to learn and tough to master.'

This is a great rule of thumb – for coin-op games. And that's what led to *Yars' Revenge*. After all, a coin-op game must be simple enough to understand so players will drop the first coin, and challenging enough to keep them dropping more. But what about console games?

I violated this rule. My games were frequently tough to learn and tough to master. Why did I do that? Because when you're allowed to demand more of a player up front, it affords the opportunity to deliver a deeper experience for the player down the road. But it's also about the economics of the purchase decision. A coin-op game purchase decision is made over and

“The feel of this control scheme totally sucked”



▲ *Star Castle* was designed by Tim Skelly, who went on to work as art director on *Sonic the Hedgehog 2*.

over again, one coin at a time. As soon as the player is dissatisfied, they simply stop playing (and paying). A home console game is different. The player makes a one-time purchase for a lot of coins, then goes home to reap the benefits of their investment. Since they're already invested, they're more motivated to do whatever's necessary to get the most value out of their purchase. This includes tolerating a steeper learning curve for entry. I relied on this fact in my games.

I sought to design games such that, when players invest more of their time into the game, they get more out of the experience. I'm not saying it always worked, but I refused to be limited by overly simple play when trying to maximise total player satisfaction. Guidelines are helpful, and they usually come from good places. However, recognising when rules do and don't apply to any particular situation can make a huge difference to the game you deliver. 🗣️

Hyper Sports

Are games an active or passive medium? Obviously, playing a game is an active experience, but recently, players are spending more and more time watching games being played by others. YouTube abounds with replays of dramatic successes and outrageous fails. Does this mean games are also a passive medium? I think it actually puts games more in line with live sporting events. After all, hanging in the pub watching a match is hardly a passive experience. What it means is video games have achieved a very significant sports milestone: they have both players and spectators.

A tour of modern video game cities

In his third and final instalment, Konstantinos's brief history of virtual cities reaches the urban sprawls of the present day



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

Developers are still far from able to recreate a truly realistic city in a video game, but the technical and artistic breakthroughs we've seen in the past couple of decades are undeniable. The complexity and scale of our virtual cities is far beyond anything possible in the 1980s and 1990s, and fresh ground is constantly being broken by modern designers.

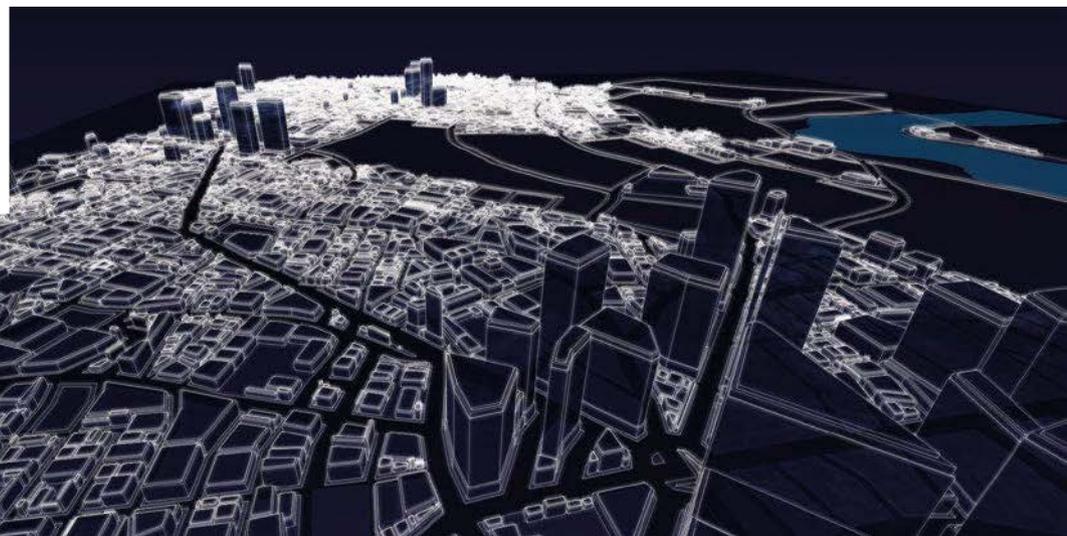
For better or worse, urbanism in contemporary games was defined by *Grand Theft Auto*. While open-world cityscapes existed before *GTA* – the innovative *Driver* is a prime example – 2001's *Grand Theft Auto III* was the smash that brought 3D cities to a mainstream audience. Following the series' top-down beginnings, *GTA III* introduced a complex 3D environment filled with activities, missions, secrets, and cartoony violence. *Vice City*, released in 2002, added an eighties vibe to the formula, before *GTA IV* and *V* fully embraced the air of cynicism and slick

production values that shaped mainstream ideas of what a game city should look like.

The *Mafia* games, developed by 2K Czech and later Hangar 13, offered an alternative take on *GTA*'s shoot-and-drive formula, with reinterpretations of existing cities tied to stories of old-fashioned gangsters. *Mafia III*'s New Bordeaux is one of the finest modern game cities – a rare instance of a virtual city that has been designed to fit in with the surrounding natural environment.

Away from the roads and traffic of *GTA* and *Mafia*, the towns of the vast *Elder Scrolls* games employ horses, carts, and exotic beasts to keep themselves on the move. *Skyrim*'s Whiterun was a convincing and evocative architectural creation that successfully blended all kinds of real-world inspirations: everything from its building materials and history to its social structure were all carefully considered. Similarly, *The Witcher 3*'s Novigrad is one of gaming's largest and most

✓ Now well over 20 years old, the *Grand Theft Auto* series' impact on video games, and how their cities look and behave, is still being felt.



➤ Introversion developed a promising tool for creating procedural cities while developing the now-cancelled *Subversion*.

thoroughly fleshed-out fantasy cities. Its level of detail is astonishing, and feels simultaneously authentic and exotic. Developer CD Projekt Red clearly spent countless hours researching Central European myths, medieval architecture, and Polish traditions to create a fascinating, densely populated island city full of grime and grinding poverty.

Insomniac's *Spider-Man* in 2018 was less eclectic in its influences. It took place in an impressively sprawling rendition of New York with more alleys than the real Manhattan – perfect for swinging around at dizzying speeds. Taking an even looser approach was *Fallout: New Vegas*, which imagined Sin City as a post-apocalyptic nightmare of mob bosses, casinos, mutants, and a retro-futurist architecture.

Ubisoft's *Assassin's Creed* games, meanwhile, have recreated important historical cities since 2007, with locations including Jerusalem, Athens, Paris, and London. The latter – seen in *Assassin's Creed Syndicate* – was a particularly ambitious virtual city, closely following the real capital's geography and shape, and offering a theme park version of Victorian

urbanism. Its faithfully recreated landmarks were correctly placed in relation to each other, while the areas in between were subtly altered to strike a balance between accuracy, budget, and gameplay needs. A crowd system filled beautiful pubs and busy streets with people and activity, as gangs, policemen, urchins, families, and workers walked, played – and shouted – at one another.

THE MODERN METROPOLIS

Convincing virtual cities aren't just restricted to open-world games, either. *Super Mario Odyssey*'s New Donk City was a playful marriage of colour, urbanism, architecture, and game mechanics. It offered the first (somewhat) realistic setting in the series' long history, full of over-the-top moments and imaginative design ideas.

In fact, New Donk City could easily become an urban blueprint for the platformer genre, just as City 17, the star of *Half-Life 2*, set the standard for cities in first-person shooters. It took elements from cities like Belgrade and Sofia, and perfectly balanced them to create an arena for tense firefights. City 17's imposing Citadel Tower set a tone for both the city and the game



▶ Contemporary game cities aren't always 3D. Here's a pitched battle taking place on the beautifully drawn streets of 2015's *Shadowrun: Hong Kong*.

as a whole, while keeping players oriented and focused on a specific goal.

The immersive sim, a genre that flourished in urban environments, was essentially defined by 1998's *Thief: The Dark Project*. Its location, simply called the City, had a sense of physicality that demanded to be experienced first hand.

There was a real weight to the City's spaces, its architecture memorably combining steampunk machines, Gothic architecture, film noir, and

high fantasy. *Deus Ex*, another seminal immersive sim, used the iconic cityscape of an alternate future New York to introduce players to a unique thriller that sat somewhere between *The X-Files* and *The Da Vinci Code*.

Then we come to one of the finest of all modern game cities. *Dishonored's* Dunwall somehow felt like a real historical place, capable of functioning and reshaping itself independently of the player's actions. Designed by Arkane Studios, it was a stunning illusion that tied together the game's action and stealth. Importantly, Dunwall didn't exist in a vacuum, but as part of an empire in a much wider imagined world, supported by its visual style which created a sense of storybook unreality, while avoiding the technical demands of photorealism.

Dunwall's distinct economy, which revolved around the production of whale oil, was pivotal to its design, informing its architecture, planning, politics, technology, and the daily life of its citizens. Dunwall may lack the scale and freedom of a *Grand Theft Auto* sequel, but its imagination and atmosphere remain unparalleled. 🗺️



▶ The harsh, masterfully designed beauty of *Dishonored's* Dunwall – a densely packed but not entirely open-world city.

Procedural City Building

Real cities are vast, packed with surprises, and infinitely detailed. Handcrafting every one of those details in a video game – especially on such a scale – would be impossible, even for large, well-funded teams, which is why procedural generation can be so useful to game developers. Important advances have been made in the field over the past few years, with the appearance of increasingly sophisticated tools that allow for the creation of roads and buildings. To cite one example, check out the freeware Medieval Fantasy City Generator. It randomly generates settlements that can then be warped and adjusted using keyboard shortcuts. You can check it out at wfmag.cc/city-gen.

Making custom Python scripts in Blender

You can use Python to speed up the process of creating assets in Blender 2.80. Andrew shows you how



AUTHOR
ANDREW PALMER

Starting out as a hobbyist level designer in the nineties, Andrew has contributed to a long list of published titles in design, art, and technical art roles. He currently works for boutique indie game developer 17-BIT, in Kyoto, Japan.



Download
the code
from GitHub:
[wfmag.cc/
wfmag27](https://wfmag.cc/wfmag27)

Blender has long been the go-to 3D modelling and animation program for artists on a tight budget, due to its free and open-source license.

More recently – especially with the release of Blender 2.80 – it’s becoming known not only for its power but also its price tag.

A powerful package in itself, Blender also has a rich Python 3 API, allowing users to create custom add-ons to tweak its behaviour or even add entirely new features. With just a modicum of Python knowledge, you can start writing your own scripts and bending Blender to the needs of your project and workflow. Here, I’ll show you how easy it is to get started.

PLAYING WITH THE CONSOLE

One of the easiest ways to get started is to use Blender’s built-in Python console to quickly test commands. Blender 2.80 has a default scripting workspace, where the Python console, text editor, and info panel are all set up for you to begin playing around with.

First off, let’s open Blender 2.80 and switch to the scripting workspace by clicking on the tab labelled ‘Scripting’ at the top of the window. For this exercise, you will only really need the 3D view, Python console, info window, and outliner, which are shown in **Figure 1**.

The 3D view should contain three objects: the default cube, a camera, and a light. If your scene doesn’t contain these, either reset the default settings from the menu (File > Defaults > Load Factory Settings) or add the objects manually.

Now let’s try using the Python console to interact with the 3D view. Unlike most other applications, the active area is set by the mouse cursor being inside it, so before doing anything else, make sure the cursor is inside the console, or nothing will happen when you type.

There are two main ways to access objects. First, by using the context to get objects that are active or selected, and second, by using `bpy.data` to access objects by name or index. Most of the time, you’ll want to operate on something that’s selected, so let’s enter our first commands to investigate the selection.



```
# Objects can be accessed directly by name
# and referenced by local variables
>>> obj = bpy.data.objects['Cube']
# Select and deselect objects using select_set
>>> obj.select_set(True)
# Selection state is accessed with select_get
>>> obj.select_get()
```

› **Figure 1:** Once you’ve moved over to Blender’s scripting workspace, you’ll be able to see all the panels you need in one place.

```
True
# All selected objects are stored in a list
>>> bpy.context.selected_objects
[bpy.data.objects['Cube']]
# The active object can be accessed directly
>>> bpy.context.object
bpy.data.objects['Cube']
# bpy.context.object is read-only, so setting
# it requires setting a different variable.
>>> bpy.context.view_layer.objects.active = obj
```

The active object is usually the last object that was selected by the user, and will be highlighted in bright orange. When there are multiple objects selected, selected objects will have a dark orange outline, so the active object should stand out. Many operations in Blender will operate on the active object, while others will simply operate on what's selected.

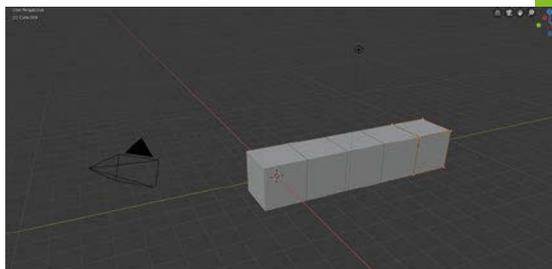
```
# Let's get some properties of this object.
>>> obj.name
'Cube'
# The object type tells us what's in obj.data.
# In this case, obj.data is a mesh
>>> obj.type
'MESH'
>>> obj.location
Vector((0, 0, 0))
# set cube position by modifying location
>>> obj.location.z = 3
>>> obj.location.z = 0
```

As you can see, it's easy to experiment with commands in the console. It's also great for discovering the properties of different object types, thanks to Blender's Autocomplete function. Simply start typing and press 'CTRL+SPACE' (or hit the 'Autocomplete' button at the top of the console) to see a list of possible completions. Try entering 'obj.' and running autocomplete to see the cube's properties.

Now let's try one last example in the console before we move to the script editor.

```
# Create four copies of cube if it's selected
>>> for i in range(1,5):
...     bpy.ops.object.duplicate()
...     bpy.ops.transform.translate(value=(0, 2, 0))
```

That was easy! Now there should be five cubes in the scene (see **Figure 2**). Both of



◀ **Figure 2:** Once you've typed the commands into the Python console, you should see five cubes appear in your scene, as shown here.

these commands operate on the selection, so first, the cube is duplicated and then moved. The selection then changes to the new cube, so each iteration of the loop creates a new cube and moves it two metres on the y-axis. We're going to use these cubes in the next section, so don't delete them!

WRITING A SCRIPT

The Python console is good for quickly testing commands and playing around, but once you begin to have an idea of what it is you want to do, it's best to start working in Blender's built-in text editor. This is a proper text editor, allowing you to modify files in much the same way you would in any other editor, but thanks to its integration, scripts can be conveniently tested at the click of a button.

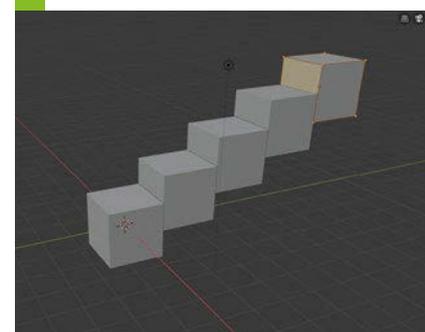
Move over to the text editor and click on the 'New' button at the top. This will create an empty file for you to enter your code. While you're up at the top, make sure that line numbers and syntax highlighting are turned on, as it will make editing your code a more pleasant experience. Now let's write some code and test our first script.

```
import bpy

# A generator gets all selected mesh objects
selected_objects = [o for o in bpy.context.selected_objects if o.type == 'MESH']
# We can also get all mesh objects in scene
all_objects = [o for o in bpy.context.scene.objects if o.type == 'MESH']

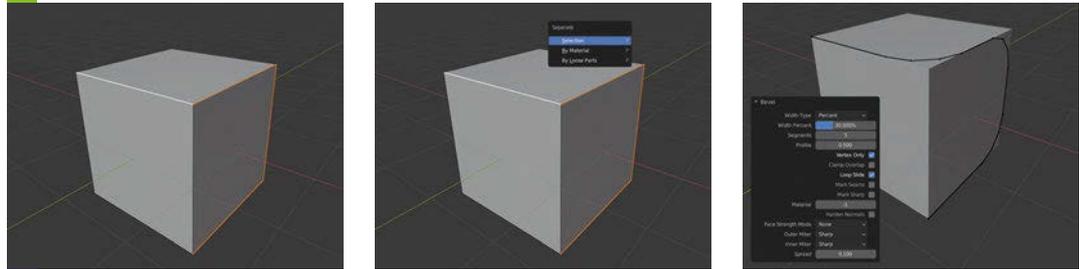
for i, obj in enumerate(all_objects):
    obj.name = 'Box_' + str(i+1)
    obj.location.z = i
```

This script can be run by pressing the button marked 'Run Script' in the top right of the text editor, or by pressing 'ALT+P'. If you entered the code correctly, the cubes created earlier should now be renamed Box_1 to Box_5 and be arranged like a staircase (see **Figure 3**). →



▲ **Figure 3:** After running the script, your boxes should now look something like this.

> **Figure 4:** We've created a simple cylinder that follows the edges of our cube. We'll soon create a script that makes this process much quicker.



Although this isn't exactly the most useful script ever written, it should give you an idea of how simple it is to batch operations in a script so they can be run on multiple objects.

A MORE USEFUL SCRIPT

Now we've covered the basics, let's try writing something a bit more useful. How about a modelling tool that makes it easy to create pipes and wires? It's already fairly easy to create pipes and wires in Blender, but the workflow for doing so is not as smooth as it could be, so how about making a tube tool that creates a tube along a set of selected edges?

Now this may sound like a huge jump in difficulty over the previous example, and perhaps it is if we write all our own algorithms, but as I mentioned, Blender already has some tools to make pipes and wires, so we can just use the Python commands these tools use behind the scenes to make our own tool.

As you've been working through the examples, you might have noticed that the info window has text printing in it each time you do something. This text is actually the Python commands to perform the last action, and we can copy it out of the info window to use in our own scripts. This means that if we manually perform the steps to create a tube along selected edges, most of the code we need to automate the process should be given to us in the info window. Let's give it a try.

PROTOTYPING THE SCRIPT

First, start a new scene so that you just have the default cube, camera, and light. Now follow the steps shown below and take note of the commands that appear in the info window:

1. With the cube selected, enter edit mode and select some consecutive edges.
2. Separate the selected edges into a new object.
3. Select the new object and enter edit mode. Select all the edges.
4. Use the bevel operator with 'Vertex Only' to smooth the edge corners.

5. In object mode, use 'Convert to' to convert the object from mesh to a curve.
6. In the object's curve data tab, set the Bevel Depth parameter to create a tube.
7. Convert the curve back into a mesh.
8. Join the tube back to the original object.

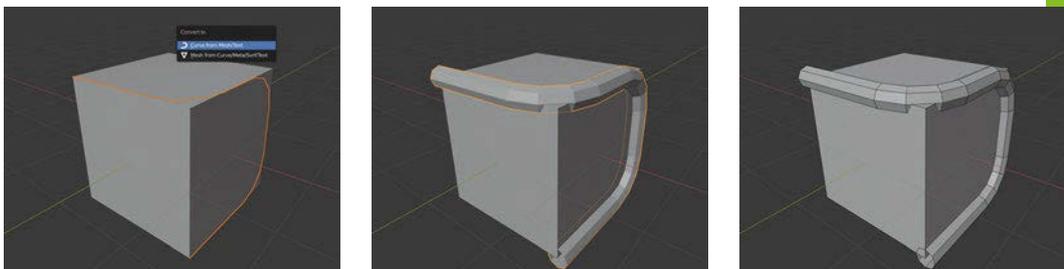
If you managed to correctly follow the steps above, you should end up with something like the image shown in **Figure 4**, with a tube following the selected edges. More importantly, most of the commands needed to accomplish this will be shown in the info window.

One important thing to note is that some of the commands we need to call will only run in a specific editing context, which for this example will be either object or edit mode. If a command is called whilst in the wrong mode, Blender will report an error, and our script will fail, so we will need to manually change the mode before calling the command.

```
# For object mode, use the following
bpy.ops.object.mode_set(mode='OBJECT')
# For edit mode, use this
bpy.ops.object.mode_set(mode='EDIT')
```

Mode changing often in scripts is not ideal, as it makes the code harder to understand and also has a performance impact, which will become clear when working with many objects at once. This method of prototyping is valuable, though, as it's an easy method of getting something up and running quickly.

Now all the commands needed to make our prototype are known, let's write the script. First, try writing the script yourself by copying and tweaking commands from the info window, paying attention to when you need to change mode or change the selection for the command to work correctly. If you run into problems, check your code against the finished script below. It's also a good idea to search the Blender Python API Reference if you need to know more information about an object type or function.



```
import bpy

def make_tube():
    obj_original = bpy.context.object
    # Separate the selected mesh elements
    bpy.ops.mesh.separate(type='SELECTED')
    bpy.ops.object.mode_set(mode='OBJECT')
    # Get object created from selected parts
    obj_tube = bpy.context.selected_objects[1]
    bpy.ops.object.select_all(action='DESELECT')
    # Set this to the active object
    obj_tube.select_set(True)
    bpy.context.view_layer.objects.active =
obj_tube
    # Enter edit mode and run bevel
    bpy.ops.object.mode_set(mode='EDIT')
    bpy.ops.mesh.select_all(action='SELECT')
    bpy.ops.mesh.bevel(offset_type='PERCENT',
offset_pct=30.0,
    segments=4, vertex_only=True)
    bpy.ops.object.mode_set(mode='OBJECT')
    # Convert to a curve and tubify it!
    bpy.ops.object.convert(target='CURVE')
    bpy.context.object.data.bevel_depth = 0.1
    bpy.context.object.data.bevel_resolution = 2
    bpy.ops.object.shade_smooth()
    bpy.ops.object.convert(target='MESH')
    # Join back to the original object
    obj_original.select_set(True)
    bpy.context.view_layer.objects.active =
obj_original
    bpy.object.join()
    bpy.ops.object.mode_set(mode='EDIT')

make_tube()
```

One quick note about this script: after separating the selection from the original object, there should be exactly two objects selected – the original object and the separated object. The active object will still be the original selection, so the `selected_objects` list is used in order to get the separated object.

Now we have a script that, when called, will turn selected edges into a smooth tube. This is a great start, but wouldn't it be nice if we could add this to Blender as if it were a built-in command and assign it to a shortcut or Quick Favourite? Even better, how about the ability to tweak the settings of the tube interactively? Well, we can, and that's what we'll do next.

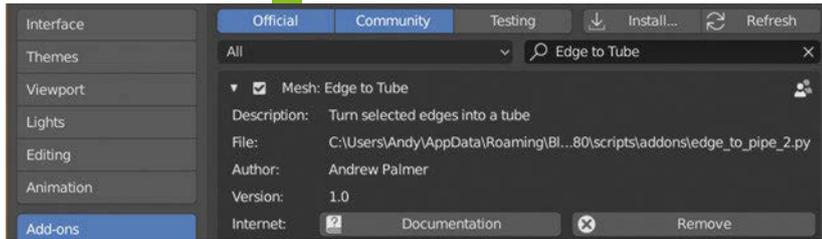
TURNING THE SCRIPT INTO AN ADD-ON

In Blender, an add-on is a type of script that can be installed through the preferences menu and will remain available to the user even after Blender has restarted. An add-on usually consists of one or more classes called operators, which are functions registered with Blender and exposed to the user in the same way as Blender's built-in tools. Although an add-on does require a little more code than a basic script, it's not that much, and Blender even includes a template add-on script that can be used as a starting point. We're going to write the extra code ourselves, though.

There are essentially three parts to making our first add-on work. First, we must let Blender know some metadata about the add-on, such as its name, category, author, and version; second, the simple function we wrote must be converted into an operator; finally, the operator we add must be registered so Blender knows it exists.

```
bl_info = {
    "name": "Edge to Tube",
    "description": "Turn selected edges into a
tube",
    "category": "Mesh",
    "author": "Your Name",
    "version": (1, 0),
    "blender": (2, 80, 0),
    "wiki_url": "https://github.com/andyp123/
blender_addon_tutorial"
}
```

The `bl_info` dictionary is displayed in the add-on preferences once the add-on has been →



^ **Figure 5:** Our Edge to Tube add-on is now installed in Blender. Remember to check the box next to its name to enable it.

installed. Make sure to correctly set the 'blender' property, as add-ons made for previous versions of Blender aren't compatible and won't run if it's not set to (2, 80, 0). The 'version' property is less important, but can be used to differentiate between versions of your add-on if you decide to share it with other Blender users.

```
import bpy

# Naming operator like this avoids warnings
class Mesh_OT_EdgeToTube(bpy.types.Operator):
    """Edge to Tube"""
    # Shown as hover tooltips
    bl_idname = "mesh.edge_to_tube"
    # Internal name in Blender
    bl_label = "Edge to Tube"
    # Name shown in search etc.
    bl_options = {'REGISTER', 'UNDO'}

    def execute(self, context):
        # make_tube code goes in here
        return {'FINISHED'}

    @classmethod
    def poll(cls, context):
        if len(context.selected_objects) == 1:
            return (context.object.mode == 'EDIT' and
                    context.object.type == 'MESH')
        return False
```

This is the skeleton for the operator class we're going to make. At the top are some parameters that identify the operator to Blender and allow the operator to use undo and redo. The `execute` function is where the edge to tube code we wrote in the previous section will go. The `poll` function prevents the operator from being called if certain conditions are not met, which in this case are that there must be a single selected object in edit mode.

```
classes = {
    Mesh_OT_EdgeToPipe,
}

def register():
    for c in classes:
        bpy.utils.register_class(c)

def unregister():
    for c in classes:
        bpy.utils.unregister_class(c)

# Lets add-on run from Blender's text editor
if __name__ == "__main__":
    register()
```

Before a custom operator can be used, it must first be registered with Blender. The name of the operator class must be added to the `classes` list. The `register` and `unregister` functions use this list and are called when Blender loads, when add-ons are installed, removed, enabled, or disabled, and also when scripts are reloaded.

Now copy and paste the function body of the `make_tube` script into the `execute` function and run the script to register the 'Edge to Tube' operator in Blender. Test that it works by selecting some edges of a cube and running the operator from the operator search menu (F3, then type 'edge to tube'). Hopefully, everything works, but if there were errors, check the error, fix any problems in your code, and try again.

Once everything is working, save your script to a file named `edge_to_tube.py`. Now the script is saved, it can be installed in Blender through the Add-ons section in Blender Preferences (Edit > Preferences) by clicking the 'Install...' button and selecting the file you saved (see **Figure 5**). Once installed, you'll need to enable it by checking the box to the left of the add-on name. Now you can use your operator even if you restart blender!

The add-on isn't finished yet, and we're going to continue developing it. Having the add-on installed can cause conflicts, however, so hit the 'Remove' button to uninstall it for the time being.

TWEAKING TUBES

Although we've made a working add-on, it would be far more useful if we could tweak the results. This is where operator properties come in. Operator properties are `class` variables that are

exposed in the redo panel that appears in the bottom left corner of the 3D view after calling an operator (see **Figure 6**), and we will use them to make our tubes tweakable.

The first thing to do is identify which aspects of the tube we'd like to be able to adjust. Tube thickness is the most obvious setting, followed by the size of the bevelled corners, and detail or smoothness of the tube. By looking at the commands used in our **Mesh_OT_EdgeToTube** operator's **execute** function, we can see that changing these things is quite easy.

```
import bpy
from bpy.props import *
```

In order to use operator properties, we first need to import the property types. There are many different types of property, but we will need only **FloatProperty** and **IntProperty**.

```
bl_options = {'REGISTER', 'UNDO'}

tube_radius: FloatProperty(
    name="Tube Radius",
    description="Radius of the tube",
    default=0.1
)

corner_bevel_segments: IntProperty(
    name="Corner Segments",
    description="How many segments to add at
tube corners",
    default=4,
    min=0,
    max=32
)

corner_bevel_percentage: FloatProperty(
    name="Corner Smoothing",
    description="How much to smooth the tube
corners",
    default=20.0,
    min=0.0,
    max=50.0
)

def execute(self, context):
```

The properties defined above belong to the **Mesh_OT_EdgeToTube** class, and can be accessed inside **member** functions by preceding them with

the **'self'** object reference. This snippet shows how they can be used in the **execute** function.

```
# Enter edit mode and run bevel
if self.corner_bevel_segments > 0:
    bpy.ops.object.mode_set(mode='EDIT')
    bpy.ops.mesh.select_all(action='SELECT')
    bpy.ops.mesh.bevel(offset_type='PERCENT',
        offset_pct=self.corner_bevel_percentage,
        segments=self.corner_bevel_segments,
        vertex_only=True)
    bpy.ops.object.mode_set(mode='OBJECT')
# Convert to a curve
bpy.ops.object.convert(target='CURVE')
bpy.context.object.data.bevel_depth = self.
tube_radius
bpy.context.object.data.bevel_resolution = 2
```

Now add all the properties you like to your code, hooking them up in the **execute** function. Try to do a couple more than just what's shown above. When you run the Edge to Tube operator, you should see something like the redo panel shown in **Figure 7**. Tweak the settings and watch the resulting tube change shape!

JUST THE BEGINNING

Although we've made one relatively simple Blender add-on, hopefully you can now see how easy it is to prototype your ideas. It's possible to make simple operators and also entire libraries of functionality, buttons, widgets, and all manner of useful tools that can be bundled in add-ons and shared with other Blender users. We've barely scratched the surface of what's possible with Python in Blender, but there are plenty of online resources to make going deeper a less daunting experience. 

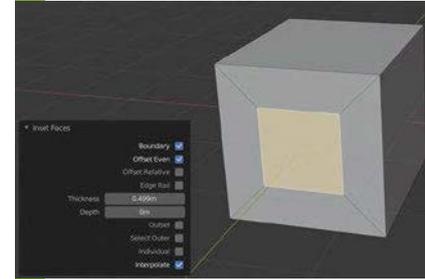
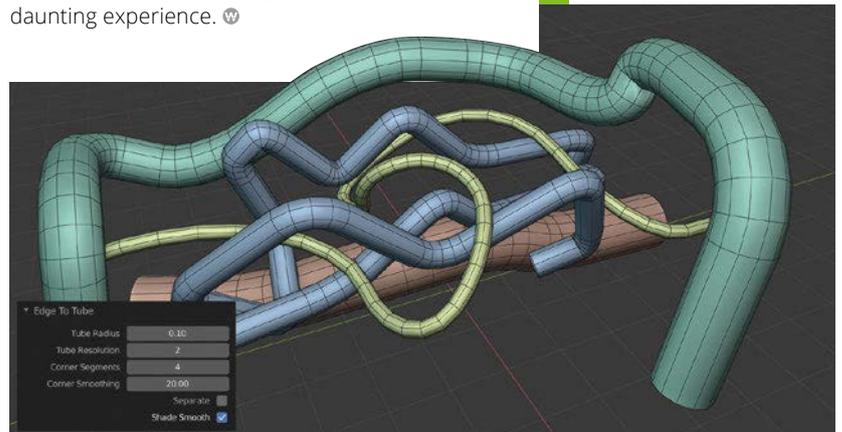


Figure 6: The redo panel for the Inset Faces operator.

Figure 7: That's it – our add-on is complete. You can now create tubes of varying shapes and thicknesses within seconds.

Declaring war on piracy

Piracy is a complex problem for developers. In Stormworks, Dan used the power of a thousand fish to compel pirates to become paying customers



AUTHOR
DAN WALTERS

Dan is a game maker with a passion for systems-based games. He's the lead developer on search-and-rescue sandbox game, *Stormworks*. mcro.org

W

ith the invention of video games came the invention of video game piracy. In 1988, *Carrier Command* asked you to enter a specific word from

a random position in its 70-page manual, when first starting the game. This wasn't the first attempt at copy protection, though, and today, if a game doesn't have some essential online component, it has probably been pirated.

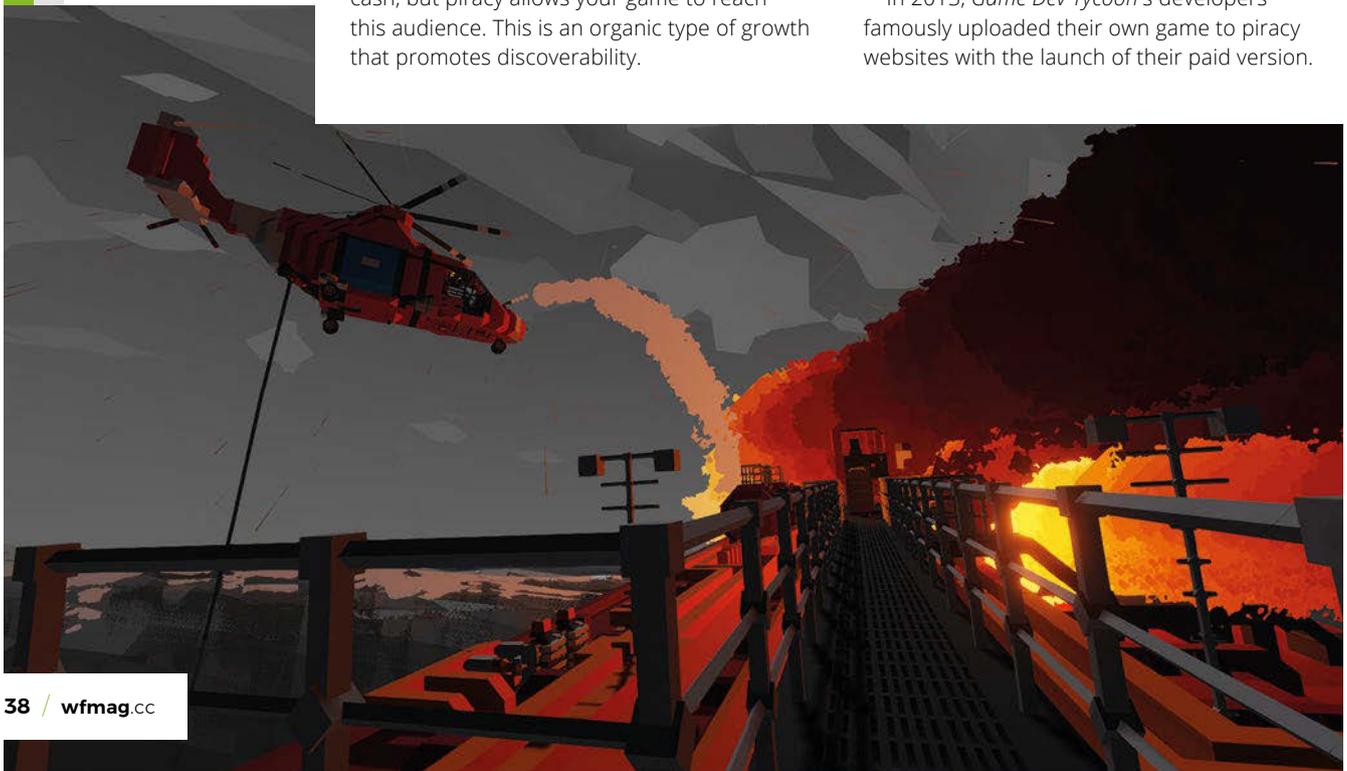
Many argue that piracy isn't even a bad thing for game makers. Some people are never going to part with their hard-earned cash, but piracy allows your game to reach this audience. This is an organic type of growth that promotes discoverability.

Other content creators just feel robbed. The movie and music industries have copyright lobbying bodies, campaigning to change the law, blocking content, and creating advertising campaigns designed to intimidate consumers.

And then there's the group I'm in. We accept piracy as part of the industry, but we think we can convert pirates into paying customers. If someone has taken the trouble to find an illegal download of our game, risked infection from a virus, then proceeded to invest their time into playing this game, they probably want the game very much.

In 2013, *Game Dev Tycoon's* developers famously uploaded their own game to piracy websites with the launch of their paid version.

▼ Launched in 2018, *Stormworks* offers a physics-based sandbox of vehicle-building and rescue missions.





▲ **Pirate *Stormworks*, and sooner or later those pesky fish will show up to sink your ship.**

However, this edition had one extra feature: piracy. When players released a game (within the game), sales would be tragic, and they were told why. When the less-bright pirates took to the games forum, complaining that piracy was ruining the game, the irony was not lost on Redditors and others, creating viral promotion of the game's launch.

Our approach, in *Stormworks*, is to detect if the game has been pirated, and treat the player's session as a demo. The game will play as normal for somewhere between 20 minutes to an hour, as players invest their time in creating vehicles, or building up cash and stock. When you author a vehicle design yourself, there's a real sense of ownership and belonging that bonds you to your creation, your world.

ENTER THE FISH

When our algorithm decides it's time for the demo session to end, a shoal of flying fish will begin chasing the player, and if they hit, they will stick to a vehicle, weighing it down while flapping their little tails. Players can run, and for some time can stay ahead, but eventually, the player will need to stop, and as long as piracy exists, the fish will just keep on attacking.

Years ago, when working for a free-to-play studio, I learned that it was far more effective to monetise a user by giving them something and then taking it away again, than if you had simply offered the item as a purchase in the first place. The psychology of loss is one of the most powerful motivators we have to convince players to spend. The technique works, and a lot of pirates are deciding it is worth buying *Stormworks*.

And then there's the joy of defeating a pirate. It feels a lot like seeing cheaters in a first-person

shooter getting banned. You play by the rules, but they don't, to your expense. Seeing the rules being enforced has a certain gratification.

We know this is not a war we can win. There is no detection algorithm we can make that cannot be defeated, but that's not a reason to surrender, and right now, we're winning the battles.

Our system is built from two components, the first being piracy detection. Most piracy involves modifying the game or running it in an altered

environment. With Steam games, many generic solutions exist that can plug into any game with no technical skill. Within moments of examining

a pirated version of your game, you should be able to see what component has changed. All that remains is simple code that looks for features of the hacked environment, or an absence of a legitimate one. We currently have eight checks, and occasionally add new ones as new cracks show up.

The second component is a delayed response. A delay has the advantage of allowing the player to demo your game, but also hides your detection from any person creating an illegal version. It is incredibly frustrating to make a change to software, only to have to wait an unknown amount of time to know if your change was successful. It makes the process very high effort and much less viable. And don't forget to have some fun with the way you terminate the player's session at the end of the delay. 🗑️

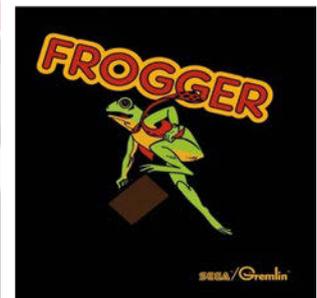
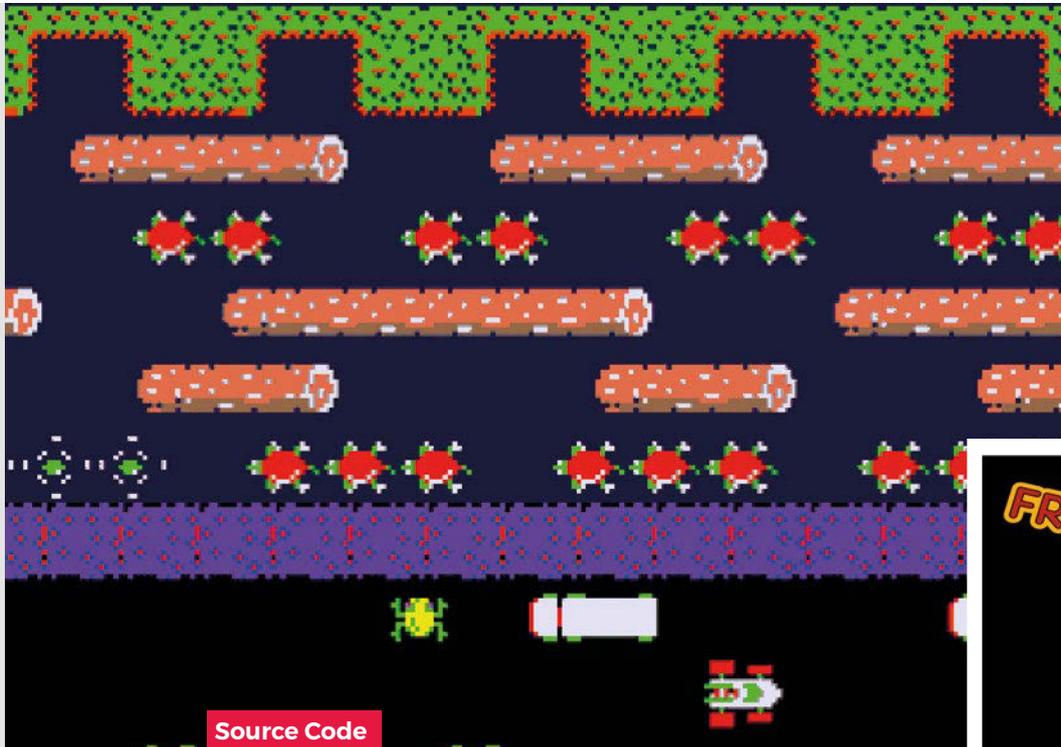
“As long as piracy exists, the fish will just keep on attacking”

An emerging market?

Piracy is an old problem, but our understanding of the underlying mechanics behind why people choose piracy, and the expanding field of study around converting free players into paying customers, has armed game makers with new tools to engage with piracy as an emerging consumer market.

▼ **There's no escape from the fish, even in frozen polar regions.**





Code a Frogger-style road-crossing game

Save the frog from busy roads and rivers with a simple remake of Konami's classic arcade game



AUTHOR
RIK CROSS

W

hy did the frog cross the road? Because *Frogger* would be a boring game if it didn't. Released in 1981 by Konami, the game appeared in assorted bars, sports halls, and arcades across the world, and became an instant hit. The concept was simple: players used the joystick to move a succession of frogs from the bottom of the screen to the top, avoiding a variety

of hazards – cars, lorries, and later, the occasional crocodile. Each frog had to be safely manoeuvred to one of five alcoves within a time limit, while extra points were awarded for eating flies along the way.

Before *Frogger*, Konami mainly focused on churning out

clones of other hit arcade games like *Space Invaders* and *Breakout*; *Frogger* was one of its earliest original ideas, and the simplicity of its concept saw it ported to just about every home system available at the time. (Ironically, Konami's game would fall victim

“We can recreate *Frogger* in just a few lines of Pygame Zero code”

to repeated cloning by other developers.) Decades later, developers still take inspiration from it; Hipster Whale's *Crossy Road* turned *Frogger* into an endless running game; earlier this year, Konami returned to the creative well with *Frogger in Toy Town*, released on Apple Arcade.

We can recreate much of *Frogger*'s gameplay in just a few lines of Pygame Zero code. The key elements are the frog's movement, which use the arrow keys, vehicles that move across the screen, and

floating objects – logs and turtles – moving in opposite directions. Our background graphic will provide the road, river, and grass for our frog to move over. The frog's movement will be triggered from an `on_key_down()` function, and as the frog moves, we switch to a second frame with legs outstretched, reverting back to a sitting position after a short delay. We can use the inbuilt Actor properties to change the image and set the angle of rotation.

For all the other moving elements, we can also use Pygame Zero Actors; we just need to make an array for our cars with different graphics for the various rows, and an array for our floating objects in the same way.

In our `update()` function, we need to move each Actor according to which row it's in, and when an Actor disappears off the screen, set the x coordinate so that it reappears on the opposite side. Handling the logic of the frog moving across the road is quite easy; we just check for collision with each of the





Download
the code
from GitHub:
[wfmag.cc/
wfmag27](https://wfmag.cc/wfmag27)

Frogger in Python

Here's Mark's code snippet, which recreates *Frogger* in Python. To get it running on your system, you'll first need to install Pygame Zero -- you can find full instructions at wfmag.cc/pgzero

```
frog = Actor('frog1', center=(400, 580))
frog.direction = frog.delay = 0
frog.onBoard = -1
cars = []
floats = []
gameState = count = 0
for r in range(0, 6):
    for c in range(0, 4):
        cars.append(Actor('car'+str(r+1),
            center=((r*20)+(c*(240-(r*10))), 540-(r*40)))
        if r < 5: floats.append(Actor('float'+str(r+1),
            center=((r*20)+(c*(240-(r*10))), 260-(r*40)))

def draw():
    global count
    screen.blit("background", (0, 0))
    for c in range(0, 20):
        floats[c].draw()
    if gameState == 0 or (gameState == 1 and count%2 == 0):
        frog.draw()
    for c in range(0, 24):
        cars[c].draw()
    count += 1

def update():
    global gameState
    if gameState == 0:
        frog.onBoard = -1
        for r in range(0, 6):
            s = -1
            if r%2 == 0: s = 1
            for c in range(0, 4):
                i = (r*4)+c
```

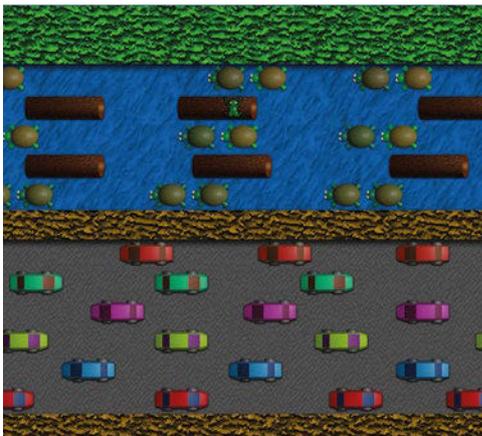
```
cars[i].x += s
if cars[i].x > 840: cars[i].x = -40
if cars[i].x < -40: cars[i].x = 840
if cars[i].colliderect(frog): gameState = 1
if r < 5:
    floats[i].x -= s
    if floats[i].x > 880: floats[i].x = -80
    if floats[i].x < -80: floats[i].x = 880
    if floats[i].colliderect(frog):
        frog.onBoard = i
        frog.x -= s
if frog.delay > 0:
    frog.delay += 1
    if frog.delay > 10:
        frog.image = "frog1"
        frog.angle = frog.direction
    if frog.y > 60 and frog.y < 270 and frog.onBoard ==
-1: gameState = 1

def on_key_down(key):
    if gameState == 0:
        if key.name == "UP": frogMove(0,-40,0)
        if key.name == "DOWN": frogMove(0,40,180)
        if key.name == "LEFT": frogMove(-40,0,90)
        if key.name == "RIGHT": frogMove(40,0,270)

def frogMove(x,y,d):
    if 800 > frog.x+x > 0: frog.x += x
    if 600 > frog.y+y > 0: frog.y += y
    frog.image = "frog2"
    frog.delay = 1
    frog.angle = frog.direction = d
```

cars, and if the frog hits a car, then we have a squashed frog. The river crossing is a little more complicated. Each time the frog moves on the river, we need to make sure that it's on a floating Actor. We therefore check to make sure that the frog is in collision with one of the floating elements, otherwise it's game over.

There are lots of other elements you could add to the example shown here: the original arcade game provided several frogs to guide to their alcoves on the other side of the river, while crocodiles also popped up from time to time to add a bit more danger. Pygame Zero has all the tools you need to make a fully functional version of Konami's hit. 🐸



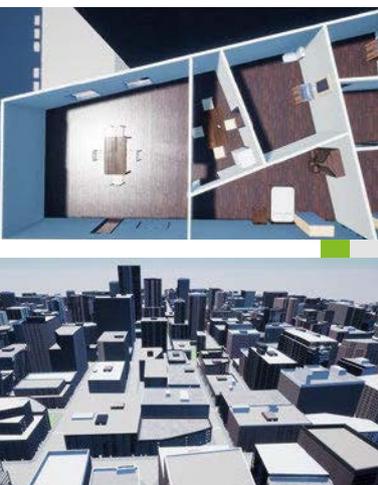
Amphibious machines

The *Frogger* arcade machine was groundbreaking in that it used two CPUs: a pair of Z80 processors, one to run the main game and one to handle the sound. Konami used the same Z80 CPU in many of its other arcade hits in the early eighties, including Konami's *Ping Pong*, *Time Pilot*, and the surreal pigs-versus-wolves action game, *Pooyan*. Along with *Scramble*, also released in 1981, *Frogger* was by far Konami's most influential game of the period, though, with its ports selling an estimated 20 million units across all systems.

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

✓ Tobias Elinder's plug-in generates entire cities, both inside and out. Jaw-dropping.



Directory

Handy city-building resources and plug-ins

If you want to create a sprawling city for your video game, here are a few useful websites and plug-ins to check out

➤ Medieval Fantasy City Generator

Create your very own Novigrad – or at least, the beginnings of one – with this simple yet effective map generator. The results can then be exported as a PNG or SVG for use elsewhere.

wfmag.cc/medieval-city

➤ Citygen

First developed in 2007, this city generator's a good few years old now, but still worth a look. Importing the results to 3ds Max or Maya requires the installation of the COLLADA plug-in.

citygen.net

➤ SceneCity

A city-building add-on for Blender; it's not necessarily cheap at \$97, but the flexibility of the software – and quality of the results – speak for themselves.

cgchan.com

➤ Procedural Cities for Unreal Engine

Incredibly, Tobias Elinder's city generator not only creates detailed networks of roads, parks, and skyscrapers, but also manages to model the interiors with stairs, offices, and desks. Utter witchcraft.

wfmag.cc/proc-cities

➤ City Generator for Unity

Laying down the princely sum of €22.33 will net you Wired Developments' plug-in, which takes your pre-existing assets (houses, tower blocks, trees, and so forth) and generates a city in a grid layout.

wfmag.cc/unity-city

➤ Procedural City Generation in Python

An open-source application that randomly draws roads and populates the edges with buildings. The results can then be exported to Blender.

wfmag.cc/python-city

➤ Building Generator

Here's a neat plug-in for 3ds Max 9 or above, which can be used to procedurally generate detailed buildings in a variety of shapes and sizes – complete with textures.

wfmag.cc/building-gen



BLACK FRIDAY

29 NOVEMBER



www.OKdo.com



Vita Forever

THE HOMEBREW DEVS BREATHING LIFE INTO SONY'S HANDHELD

Discontinued in 2019, the PS Vita is still alive thanks to a community of homebrew developers

WRITTEN BY LEWIS WHITE

The PlayStation Vita wasn't supposed to be hacked. Sony's diamond in the rough released in 2011, with the company having learned its lessons from the PSP's mistakes. Released in late 2004, the PlayStation Portable helped usher in a new generation of handheld gaming. But it was cracked early: the PSP's initial firmware, 1.00, easily allowed hackers to run unsigned code. Sony would spend years trying to block hackers, but their attempts were futile; the system was insecure, and insecurities are a hacker's favourite toy.

With the console code disassembled within months of its US release, the PSP played host to a growing homebrew scene in the mid-2000s – key titles included *Nazi Zombies*, *Lamecraft*, *Wagic*, and *Kurok*. "The PSP defined my childhood," says prominent hacker Andy 'TheFlow' Nguyen. After seeing his father modify his console with

a Pandora battery, Nguyen's eyes were opened to both hacking and homebrew. Just twelve years old and a self-professed bad student, the PSP ignited his interest. "I was so amazed; I immediately started programming," he says. "I was driven by my dream of writing my own custom firmware."

Nguyen learned fast. After months of releasing plugins and learning through research, he released his own custom firmware, 6.20TN, the first public jailbreak for the PSPgo. A few years later, he made the move to Sony's successor, creating some of the most important homebrew utilities on the Vita – and bringing a whole new lease of life to a handheld that suffered poor treatment from its creator. The PlayStation Vita wasn't supposed to be hacked, but it's those dedicated hackers and coders who are still keeping the dream of a device with huge amounts of unfulfilled potential alive.



^ *Fallen Knights* is in early development, but it's already starting to look a bit like its inspiration, FromSoftware's *Souls* series.

Between 2012 and 2016, Nguyen and other members of the hacker group Team Molecule cracked the Vita wide open. "We only hack things we love," says fellow Team Molecule member, Yifan Lu. "We really care about the device; we banded together; we shared the same passion... we [even] designed our own development tools to rival Sony's. We spent most weekends reverse-engineering Vita code."

In 2016, Team Molecule released HENkaku, a web-based exploit that allowed Vita owners to run homebrew. Thanks to the team's reverse engineering and Nguyen's tools, a dedicated community had formed around a device that had been overlooked by mainstream gamers.

THINKING UPSIDE DOWN

One of the developers who joined the Vita's growing homebrew community was Vagelis 'LazyOx' Oganosov, who's currently making *My Bakku Pakku* – a mash-up of *Super Mario* and *Katamari*. "I picked up a PSP because of the homebrew but, well, it's old, so I decided to pick up a Vita," Oganosov says. Initially purchasing the system for emulation purposes, he changed focus after discovering a few homebrew titles made using his engine of choice: Unity. "I've always been a creative person, and when I discovered Unity was working on Vita, it just clicked. Unity equals game. Make the game, put it on Vita."

Before *My Bakku Pakku*, Oganosov started out with a smaller project called *Frying Master* – a re-creation of *Mario Party*'s Sizzling Stakes minigame, in which the player has to fry all six sides of a meat cube before the time runs out.

With this first game under his belt, Oganosov decided to move on to a more advanced project. "I'm primarily a mobile developer," he tells us. "I'm used to the process of optimising my games, but when [I started on *My Bakku Pakku*], it was running at 15 frames per second."

Having previously worked on *SouzaSim*, a mobile racing game, Oganosov brought experience of optimising for handheld formats to the new project. All the same, the early stages of *My Bakku*

"When I discovered Unity was working on Vita, it just clicked"

Pakku's development proved challenging: even with a visual style designed around the Vita's GPU, slowdown persisted in early builds. "I know if you have real shadows, not baked shadows, [lighting] effects, and a high-resolution image, it'll start lagging," Oganosov says. "But on Vita, I couldn't find why."

Theoretically, a simple 3D platform game should have worked perfectly well on the Vita, but the addition of *Katamari*-style physics – where the player pushes a sticky ball around a map, collecting objects – proved troublesome for the system. "After I designed the original graphics build, when I was originally testing →

▼ Created by FantaHG, *Speedrun* is a fast-paced 3D platformer.



PSVITA ENHANCED

Plugins can do a lot of things on the PlayStation Vita. From CPU and GPU overlocks to better quality screenshot functionality, there are a lot of cool things that can be done. However, one of the coolest may be Andy Nguyen's PSP patches. Using a homebrew application called Adrenaline – a project that unlocks access to the full suite of PSP features within his Vita – Nguyen is making PSP games *better* on Vita, with tweaks like dual analogue support for *Kingdom Hearts*, *Peace Walker*, *GTA*, and more welcome additions.



^ Crazy Traffic Jam 3D is an infinite runner with basic power-ups. Released in 2017, it shows how far the system's homebrew games have come.



^ Consisting of little more than a 3D room and a frying pan, *Frying Master* is a simple yet entertaining minigame.

PRINCIPLES ON PIRACY

Hackers seldom break a system's security for reasons of piracy, but it's nevertheless an inevitable outcome. "I do feel personally responsible," says Team Molecule's Yifan Lu. "Some people have this belief that those who develop the tools are responsible for what people do with it - I disagree. We tried our best to make our tools difficult to repurpose for piracy. I personally received a lot of hate. [Our jailbreak] is what allowed the influx of pirated games to happen. I regret being a part of it."

the game, I didn't have any objects," Oganosov recalls. "It was running fine, flawless. I started putting in objects, and it would just slow right down. I thought it was bug-related! I was constantly, like, 'Why?' The graphics are so simple: there's no lighting, there's no nothing! I'm literally hitting my head against the wall."

Since having a lot of objects on screen was a key focus of *My Bakku Pakku*'s design, Oganosov focused his energies on optimising the game for the Vita's hardware. "In the game, you pick up objects, they disappear, and they fill your backpack," he explains. "The problem was CPU usage... but it plays well now."

It's still a work-in-progress title, but Oganosov has hopes for *My Bakku Pakku* beyond a curio on a hacked system - the ambition is to bring the game to other mobile formats should it work out well on Vita. It's proof that, in a realm known for its piracy and other nefarious dealings, there's also something wholesome and positive emerging from the handheld hacking space.

PUT A SPELL ON YOU

Another mobile developer, 29-year-old Sakis (better known as VitaHEX), has also spent the past couple of years trying to get the most out of the Vita. "I have a great amount of experience from my work on mobile games," says Sakis. "Technically, Vita is just that: a mobile system, but it's not giving you its full power."

Since 2017, Sakis has been making impressive games that push the Vita to its limits. Starting with *Crazy Traffic Jam 3D*, one of the first 3D homebrew titles made for the Vita, VitaHEX's

projects have evolved into some of the most technically accomplished games available on the system. "*Crazy Traffic Jam 3D* was built in Lua Player Plus, a simple programming tool built for simple functions," Sakis says. "It couldn't handle 3D rendering at all."

Like so many other homebrew developers, Sakis has since moved to Unity. This has helped him create some incredible projects, like a remake of Hideo Kojima and Guillermo del Toro's horror masterpiece, *P.T.* Sakis has worked on remaking the AWOL *Silent Hills* demo for the Vita; titled *The Hallway*, this portable *P.T.* is still in development, with about ten percent of the original game faithfully recreated.

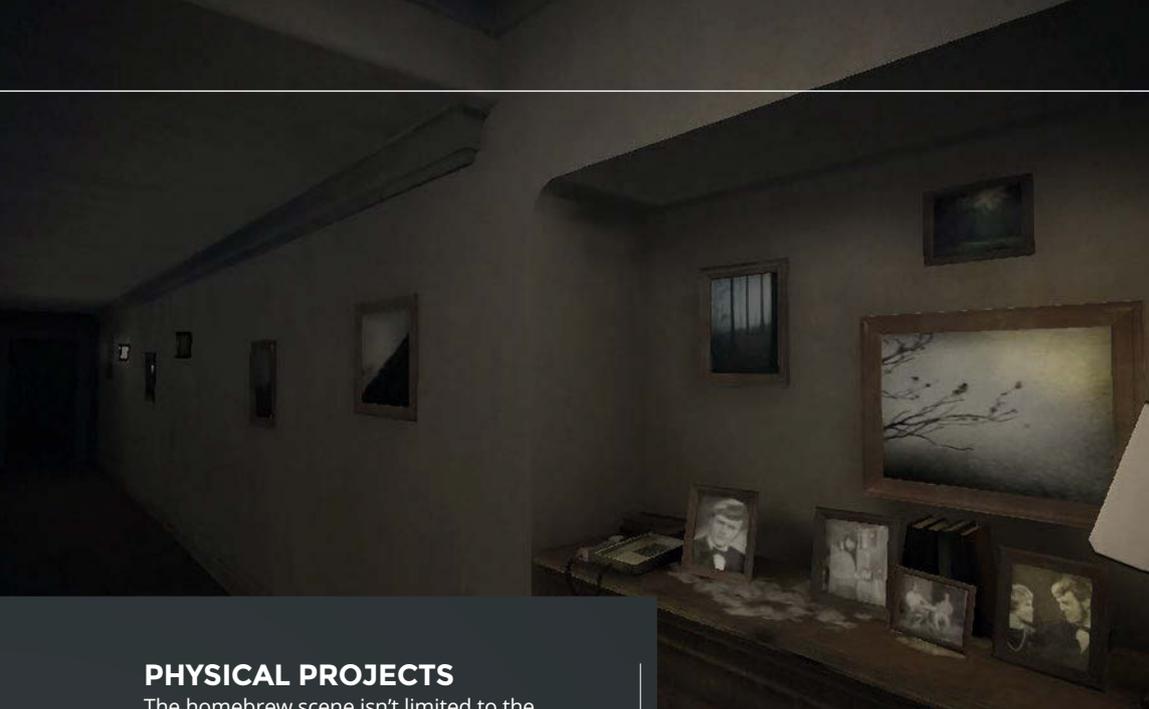
"I worked on most of the assets from scratch," Sakis says. "I studied the [original] game - I worked on every 3D model based on screenshots and gameplay videos." Despite being on a far less powerful system than its home on PS4, the downscaled Vita version manages to recapture the original game's eerily dilapidated location.

Creating such a detailed environment required optimisation: with so many objects around and atmospheric lighting aiding the mood, *The Hallway* required a lot of development streamlining. Its level of detail had to be strictly controlled; aside from a handful of key scenes, most of the game's lighting was pre-baked, and more mobile-focused shaders were also used.

"Real-time lighting was a big hit for the performance," Sakis says. "The Vita has a powerful GPU, but Unity doesn't let you take full advantage of it. I decided early on to bake all the game's lighting, only using the extra GPU budget to keep important lights working in real-time. The game has improved a lot since its first release."

As work on *The Hallway* continues, the VitaHEX brand has also expanded, with numerous projects being worked on in tandem. Most recently, *Fallen Knights*, Sakis' take on the *Souls* franchise, has been the talk of the homebrew scene. While still early in development, Sakis hopes he can create a decent *Souls*-like experience on Sony's handheld. "It's a long road ahead, and my free time is very limited, but I hope to get a good *Dark Souls* experience [on Vita]," he says.

"The Vita has a powerful GPU, but Unity doesn't let you take full advantage of it"



◀ Despite the Vita's limitations, VitaHEX's *The Hallway*, a portable rendition of *P.T.*, gets the look and feel just right.

PHYSICAL PROJECTS

The homebrew scene isn't limited to the digital sphere, though. While a high cost limits production of physical games and hardware in this semi-professional world, recent years have seen the release of more items we can actually hold in our hands. As we saw in *Wireframe #7*, the Sega Dreamcast is a goldmine of physical homebrew games released well after the format's death – and even fan-made hardware, like DCHDMI, designed to enhance the system's playback on modern TVs. But those projects aren't common: they require a dedicated fanbase, a lot of time and, most importantly, funding.

One such project making waves in the Vita homebrew community is the PSVITA Dock, a physical project inspired by the Nintendo Switch and aiming to turn Sony's handheld into a home console. It's like the PlayStation TV, but with higher compatibility for games. Like those homebrew games mentioned earlier, the PSVITA Dock is being made by dedicated fans and tinkerers: Croatian developer xCorra, along with a couple of other collaborators, managed to raise over £13,000 for the hardware on Indiegogo in early 2019.

"In the community, I'm just called The Dock Guy," xCorra laughs. "I talk to other developers – they like my work, they go, 'Oh, you're The Dock Guy!' At the start, I was an unknown who just burst onto the scene, showed off an idea, and asked for money. I had to prove myself: I made my own subreddit, YouTube videos – I updated the Indiegogo 15 times."

As he continued work on the Vita dock, improved the internals, and polished the software, the community took a liking to the project. "The biggest change [from the start] is doubt," xCorra says. "Doubt is amazing: at the beginning, all I'd get were floods of questions. 'What's the latency?' 'How durable is the plastic?'"

Trust is hard to gain, especially when you're asking for money. Many were confused by the

premise: PSVITA Dock is essentially a plastic shell housing a Raspberry Pi. Using a specialised version of a USB Streaming plugin coded by another hacker, Xerpi, the hardware interfaces with the RPi running VLC Media Player, and allows players to slap their handheld into an attractive slab of plastic and play their Vita on a TV. Simple.

"It's improved a lot over time," says xCorra. "At the start, we could only get 30 frames per second. [Vita] essentially uses the USB 2.0 data limit so we couldn't have 60fps at all. We all worked together, changed the code a bit, and made an optimised custom plugin for the dock." For 60fps, certain sacrifices had to be made; PSVITA Dock plays back 60fps games at 864x488 resolution instead of the system's native 960x544, while games that run at 30fps or below are displayed in their native resolution. As xCorra adds: "Most 60fps games don't run at native resolution anyway; I don't think [80% of the resolution] is too far for double the frame rate."

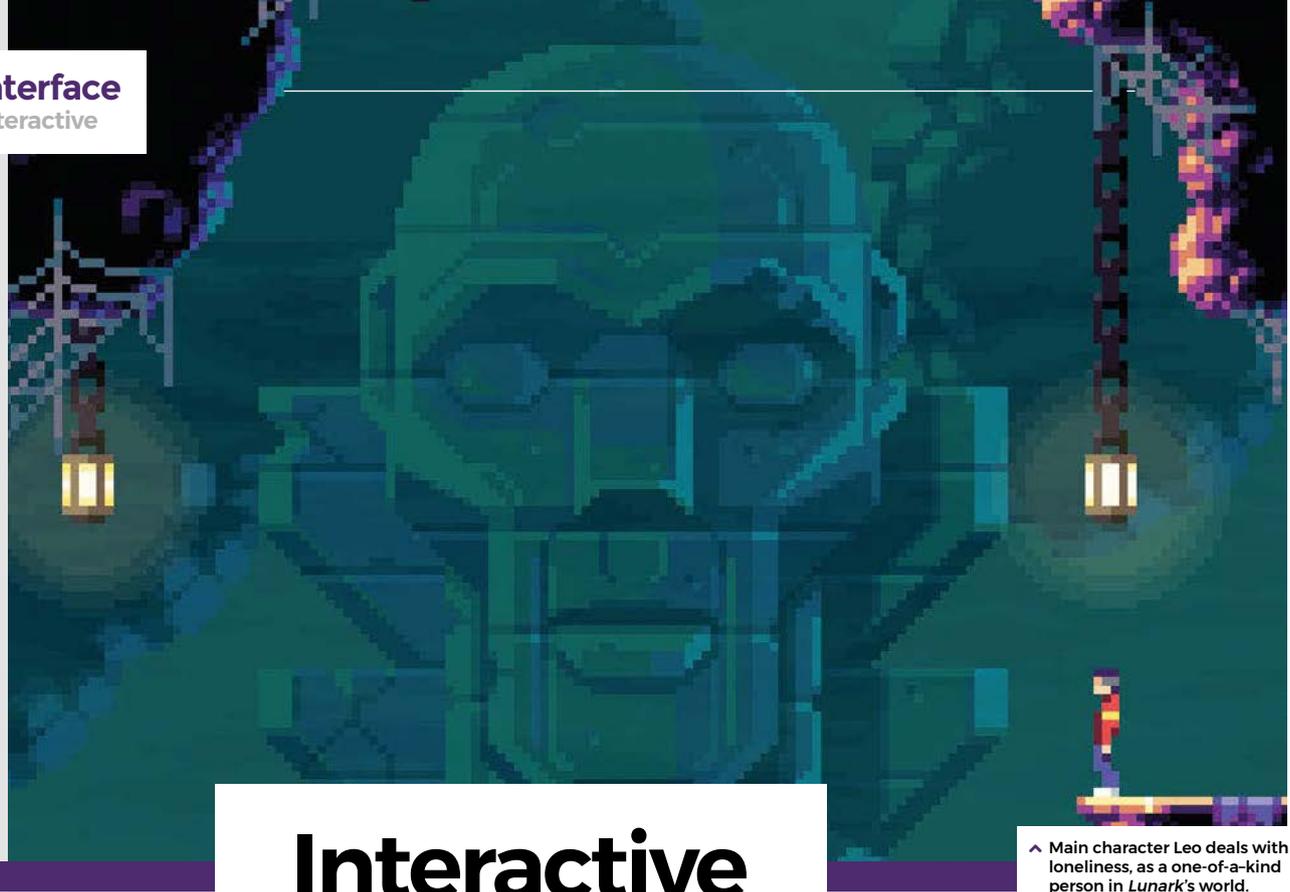
VITA NOVA

Vita homebrew is still going through growing pains. Controversial game jams like VitaJAM2019 and rampant piracy has hurt the image of a blossoming scene, but it's clear the PlayStation Vita is living up to the potential the hardware had inside it all along – even if Sony itself gave up on the device long ago. It's not too harsh to say the Vita was abandoned by its creator, but thanks to a dedicated homebrew scene, new life is being breathed into it. 🌱



✓ The PSVITA Dock houses a Raspberry Pi which helps connect the Vita to a television.





▲ Main character Leo deals with loneliness, as a one-of-a-kind person in *Lunark's* world.

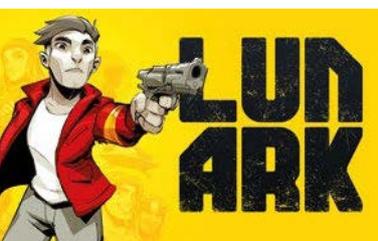
Interactive

Lunark

Catching up on this Flashback to the future

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

▼ Vinet began his coding life on the Atari ST, and at one point tried (unsuccessfully) to code a version of *Bomberman* for his graphic calculator.



With each passing year, the legend of *Flashback* strengthens; in part because it's just a fantastic game and has stood the test of time admirably (arguably better than stablemates like *Another World*), but also because there just haven't been many other games to even try and assume the mantle. With that introduction, it's of little surprise that – yep! – *Lunark*, from Canari Games, riffs heavily on the *Flashback* style. And it's looking great.

Johan Vinet was born in France, but made his way to Montreal, Canada to do the whole 'life' thing. After working in web/

graphic design, he moved into video games – a love of pixel art and animation earning credits on the likes of *Shovel Knight Showdown*, *Adventure Time*, and *Mercenary Kings: Reloaded Edition*. It wasn't until the end of 2018 that Vinet struck out on his own, forming Canari to commit to his first solo title (ignoring an earlier aborted attempt), taking inspiration from the likes of *Prince of Persia* and, yep, *Flashback*.

The decision was made to put *Lunark* on Kickstarter, with the project exceeding its initial goal and bringing in around £48,000.

Initially, Vinet planned to use Kickstarter funds to hire in additional talent to help bring the game to fruition, but in the end, it didn't raise quite enough, so this remains a solo project. "There's so much to say about a Kickstarter campaign so I'll try to keep it short," Vinet explains. "Each campaign is unique. I studied about 60 campaigns for four months, trying to determine the ingredients that contributed to their success, or failure. Genre, goal, price point, size of the company, video duration, rewards,

"I found it important and rewarding to remain enthusiastic and positive, even in uncertain times"

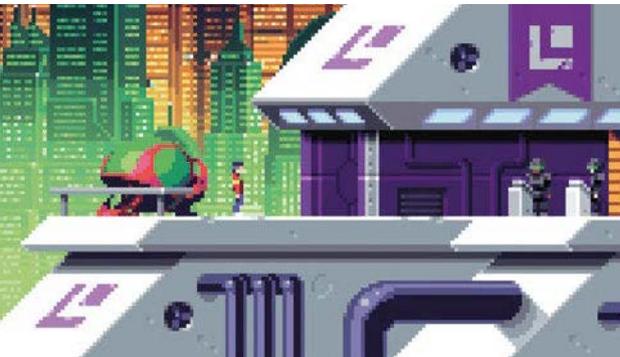
average engagement of the backers, etc... This allowed me to have a more precise idea of the ingredients that should compose mine."

After deciding to try and raise what he felt *Lunark could* achieve on Kickstarter, rather than what it *did* need to get made, Vinet finished up the campaign by trying to be as transparent as possible with the whole process; honesty and passion hopefully shining through: "I found it important and rewarding to remain enthusiastic and positive, even in uncertain times," he says.

Since the – successful – project's close, Vinet estimates his progress at around the 35% mark: slower than expected. This means the game will miss its originally intended release date, but the



^ Vinet is keen to make *Lunark* accessible, and is shooting for around two or three hours' time to complete the game.



^ There's a lot to be said for the little touches rotoscoping brings to animation.



developer remains confident. "One chapter out of five is completely done at the time of writing," he explains. "At the same time, many of the fundamental elements of the game have already been coded... Knowing that the backers won't get their game on the initial time frame adds a lot of pressure and makes some of my nights a little agitated, but I still prefer to deal with their potential impatience and deliver them a product I'm really proud of."

FLASH FORWARD

That product, as we say, is already looking great, with some (fittingly, given the inspirations) striking animation apparent at first glance.

We ask Vinet how he put together the smooth, enticing look of the game in motion: "It's a bit of a mixed bag of techniques," he says, "Most cutscenes are produced using rotoscoping: I film myself (or create 3D animations) and use this footage as a reference for the creation of the actual visual. For the sprites (the main character, the enemies, etc), most animations are made on the fly without necessarily using a model."

As for the tools used, Vinet admits it's hard to use anything other than Photoshop for animations like the rotoscoping, but others do factor in. "For the realisation of tilesets, I may also use Pyxel Edit as an extra tool," he says, "And I develop the game using GameMaker

Studio 2. It's a very complete tool that meets my needs perfectly."

All of this is done solo, of course, and it's something Vinet is cognisant of while working on *Lunark*, missing things like working closely alongside colleagues and hearing their ideas on a daily basis. "Also," he says, "When another artist provides me with their sprites, I can only focus on giving my best to animate them. Being alone at every step, it is very tempting to be too [much of a] perfectionist with each one of them. I often have to convince myself to accept that an asset is 'good enough' to be able to move on.

"On the other hand," he continues, "being alone on such a very personal project allows me to save a lot of time on other aspects of development. There is no time spent on passing information to others, giving examples in design docs, arguing or justifying solutions that had to be found to solve technical problems, etc. Besides, if there are any problems, I can only blame myself!"

We've mentioned *Flashback* more than enough times here, but it's not the only source of lightbulb moments for Vinet, as he explains: "*Akira* is a great inspiration for me when it comes to imagining a futuristic city... Many games and films from my youth – *Total Recall*, *Indiana Jones*, and so on – will have a tribute [whether or not it's deliberate or subtle] in *Lunark*." 🐾

NEWSFLASH

"My first idea was to mock-up a *Flashback*-like game with the limitations of PICO-8 or a Game Boy-style handheld," Vinet says of *Lunark*'s early stages. "My specs were set. After seeing the enthusiasm following the publication of my first tests, I stuck with that. Every day I wondered if there was an audience for a game with such a low resolution; I take the success of the Kickstarter as a 'yes'. If it's indeed faster to produce visual assets at this size, it is more difficult to get fluidity and details in the animations."

Developer Profile

Namco

Before Bandai, basically

There's a divide in game-players the world over – those who talk about Bandai Namco, and those who talk about *Namco*.

The former speaks of a studio that brings some great titles to the market, both from its own in-house studios and via publishing deals, tickling our fancies with *Tekken 7s* and *The Witcher 3s* as we wonder aloud when the company's next name change will occur. It's a solid company – good, even – but it doesn't exactly instill a sense of pure gaming joystalgia.

No, that's reserved for the original Namco – the developer and publisher that, when mentioned, sends folks of a certain age into misty-eyed reverie. This, the original Namco, was the home

of *Pac-Man* and *Dig Dug*, *Point Blank*, and *Ridge Racer*. While it's largely seen as just another games company in its modern incarnation, those who knew the pre-Bandai company remember it as a legend; a Namco logo on a game meant you were very likely to have a good time.

It's not a surprise, really – Namco had been running in one form or another (previously as Nakamura Manufacturing) for 23 years before releasing its first self-developed arcade video game, first under founder Masaya Nakamura's lone watch as he installed and maintained mechanical children's rides in Yokohama and Tokyo. These performed so well that by 1966, the company was contracted to develop rides based on Disney characters. But it was the creation of *Periscope* alongside Sega in 1965 that set the company on the path to its true calling.

Nakamura claimed *Periscope* – a shooting gallery arcade game of the pre-video variety – as his own design, the success of which led to more mechanical arcade games like 1972's *Formula-X*. A dalliance with Atari followed (see boxout), resulting in strong growth for what was now officially Namco, and by the late 1970s, the company was a bona fide arcade video game developer and manufacturer. Its strong financial position, helped in no small part by those early successes and the (by then dissolved) Atari partnership, put Namco in an enviable position compared to its competitors.

Fortunately for everyone – a rising tide lifts all ships, and all that – Namco's output erred on the right side of top notch. *Pac-Man*, of course, but also



^ *Dead to Rights* came from Namco's US wing. It... well, it had a dog at least.



^ *Galaga* got a boost of popularity when Matthew Broderick played it in *WarGames*.



^ *RIDGE RACER* - ahem.

Galaxian, *Rally-X*, and *Pole Position* – and many more – contributed hugely to the development of arcade gaming (and video gaming) as a whole.

Namco also managed, unlike many other studios of the era, to successfully transfer from the 2D 1980s to the 3D 1990s, innovating brand new series like *Ridge Racer* and *Tekken*, while staying true to its roots with callbacks to titles of old and the odd sequel here and there, like *Xevious 3D/G*.

Amazingly, Namco even managed to weather another huge shift in the gaming landscape, successfully transitioning from a company focused

largely on the arcades, to one focused mainly on home consoles. A big help there came through Namco's use of the System 11 arcade board, itself based on the original PlayStation's hardware. This meant titles made for the arcade system could be easily ported to the

“Namco transitioned from a company focused largely on arcades, to one focused mainly on home consoles”

home, with only minimal changes or downgrades (and, in fact, plenty of upgrades) during the transition.

Success on success followed, and the Namco name was one very firmly established in the gaming lexicon. By 2006, though, the company as we knew it ceased to be, as the 2005 merger with Bandai saw the company renamed Namco Bandai (and later Bandai Namco). The company had been changing for a time, but it was the merger that put the nail in the coffin.

This isn't an 'Oh, how the mighty have fallen' requiem for the modern studio; Bandai Namco still pops out bangers on a relatively frequent basis. But it could never compete with what came before: not only is the Namco of the past home to a better logo and some of the greatest arcade and PlayStation titles ever made, it's also firmly nestled within the unassailable walls of Fort Nostalgia.

Maybe one day the modern company will reaffirm itself and become what we remember it once was. In the meantime, we can all just play arcade greats on the original PlayStation. Well, except for *Time Crisis*, unless you have a CRT knocking about. ☹



^ Not just arcade games: Namco's RPG roots run deep, too.

Breaking out

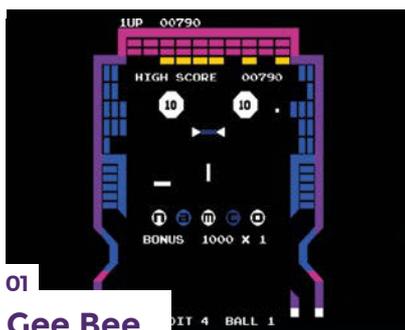
Nakamura Manufacturing purchased Atari Japan in 1974, with the subsidiary floundering by itself. By 1976, *Breakout* had launched and was a huge success, and with it came the chance to distribute – but not manufacture – the arcade machine around Japan. Soon enough, the proto-Namco ran out of cabinets to place around the country, and Nakamura found himself on the back foot as bootleg machines had been springing up thanks to Yakuza operations capitalising on the game's popularity. To make matters worse, a misunderstanding between Atari founder Nolan Bushnell and Nakamura resulted in the latter believing manufacturing rights had been obtained for *Breakout*, with Japanese-made machines following soon after. The resultant lawsuit, after an Atari rep saw more machines in Japan than the company had allocated, showed that was not the case. Even so, Namco and Atari maintained a relationship for many years to come, with the former purchasing a stake in the latter in the 1980s.



Enter the Funscape

10 fine Namco memories

Narrowing it down to just 10 is nigh-on impossible



01 Gee Bee

Arcade – 1978

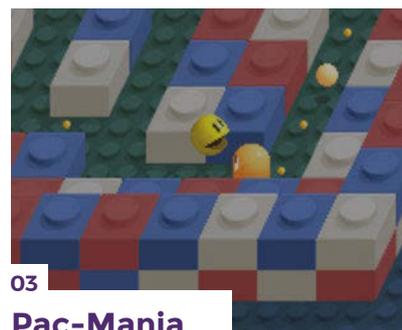
Active in gaming for a few years by this point, *Gee Bee* marked the first time Namco developed a game internally without any partner studios. The brainchild of *Pac-Man* creator Toru Iwatani, *Gee Bee* was not a tribute to high-pitched disco but instead a creative mix of *Breakout* and pinball. Handy, given *Pac-Man*'s dad had wanted to design pinball machines all along.



02 Xevious

Arcade / multi – 1983

It's no *Galaxian*. Sorry, but it's true. Still, *Xevious* made a mark by being one of the earliest titles to have a boss fight, as well as by using pre-rendered graphics. Its popularity helped kickstart the shoot-'em-up genre as we still know and love it today, and its performance in Japan has made it the stuff of legend. *Galaxian* is better, though.



03 Pac-Mania

Arcade / multi – 1987

When *Pac-Man* (and *Ms.*) has been done to death, where do you go? Well, Namco took things into the pseudo-third dimension with this arcade sequel, as well as adding blasphemy to the core mechanics: a jump button. You can't beat pure *Pac-Man*, but *Pac-Mania* deserves a shout-out for the changes – and risks – it brought to the fore.



04 Tekken

Arcade / PS1 – 1994

Some of *Virtua Fighter*'s team, poached by Namco, put together the first major challenger to Sega's 3D fighter dominance. *Tekken*'s fate was sealed when it was ported to the original PlayStation, thanks to the shared hardware specs of Sony's machine and Namco's System 11 arcade boards. It was popular in arcades – it was rather more than popular in the home.



05 Time Crisis

Arcade / PS1 – 1995

Light gun games needed a bit more dynamism, it seems. We didn't know this until *Time Crisis* blew up in everyone's faces in arcades (and again, homes) in the mid-1990s. The addition of a pedal to move in and out of cover, as well as a (tight) time limit to get through levels, was a one-two punch of genius moves. A bona fide classic, no less.





06

R4: Ridge Racer Type 4
 PS1 – 1998

The first game put the PlayStation on the map, and Namco stuck with Sony's machine for a long time. Eventually, the fourth home version appeared and it was... well, brilliant. Building on the arcade chops of the original and *Revolution* while enhancing the added depth of *Rage Racer*, *R4* was the perfect blend of styles for an arcade-with-depth game at home.

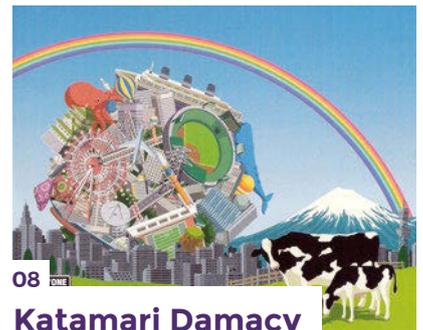


07

Soulcalibur

Arcade / DC / multi – 1998

Soul Edge hit earlier in the decade, but it was through *Soulcalibur* that Namco hit its stride on this divergence from the path of *Tekken*. Weapon-based combat could so easily have been a gimmick in a 3D brawler, but it was handled with such careful design – and panache – that it was impossible to do anything but love this Tale of Souls.



08

Katamari Damacy

PS2 – 2004

There was an element of *Pac-Man* to Keita Takahashi's classic *Katamari Damacy*, which saw players roll an ever-increasing sized ball to collect (or gobble) things up. But where *Pac* was limited to small mazes, the Prince would – eventually – stick the entire world to his cosmic sphere of wonder. A fantastic little game, and one absolutely riddled with character.



09

Tales of Symphonia
 GC / PS2 / PS3 / PC – 2003

It would be remiss to ignore all the good Namco did with its work in the RPG realm, and the *Tales* series – especially *Tales of Symphonia* – is a fine example of the studio at its best. OK, so it was pretty standard JRPG fare from a storyline perspective, but the game's mix of exploration and quick-paced combat made it captivating, even across dozens of hours.

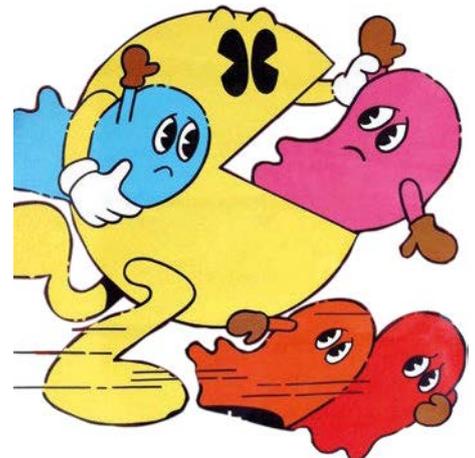


10

Ace Combat 5: The Unsung War

PS2 – 2004

Another arcade series that found its true calling on the PlayStation, *Ace Combat 5* is an overlooked gem. Simple arcade mechanics were backed up by an actual decent storyline. Hokey at times, sure, and the game itself could be shallow – but it's so exciting, so full of twists and turns, that you really don't care.



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

John Wick Hex

Tick tock, Mr Wick

Info

GENRE
Strategy

FORMAT
PC

DEVELOPER
Bithell Games

PUBLISHER
Good Shepherd
Entertainment

PRICE
£15.99

RELEASE
Out now

REVIEWED BY
James O'Connor

HIGHLIGHT

John Wick Hex's contemplative pace means the aftermath of violence is often more exciting than the violence itself. When you've gotten through a tense situation, and the bodies have piled up around you, it's good to take a breath and pat yourself on the back for pulling it off.

▼ Throwing your gun is faster than shooting someone, which doesn't seem right.



Part of the *John Wick* films' joy is the inevitability of John's success. In the first film, when crime boss Viggo realises that John is coming, he knows straightaway that they can't win against the 'Baba Yaga'. By comparison, the John of *John Wick Hex* – a prequel to the movies – often feels like he's coming off a bad night's sleep and can't quite work the stiffness out of his joints. As you control John, what he can't do is as glaring as what he can. He can't melee from a crouching position or stop firing a machine pistol after his target is already dead, for instance. He certainly can't quickly fire off a point-blank headshot on a downed target the way he does in the films.

John's limitations are deliberate, a way of evening the odds and making the character vulnerable – effective invincibility is less

interesting in a game than it is in a movie. Rather than a straight-up power fantasy, *John Wick Hex* is a strategy game that makes you perform constant cost-benefit analyses of your potential options for dealing with each level's goons. Combat is boiled down to a series of rules, percentages, and time codes, with a rigidity that can often be stifling. While there are moments where everything clicks into place, and your decisions lead to a satisfying sequence of actions and events, it rarely feels like *John Wick*.

Every action you can perform is mapped out across two interconnected spaces – the setting John is within, and the 'timeline' that runs across the top of the screen. You move through the node-based levels by clicking on dots on the ground and selecting actions, but over time you'll come to realise how important it is that you keep an eye on that timeline, which tracks exactly how long every action made by yourself and your enemies takes. Time stops when you're not moving, giving you time to plan out your next action and account for enemies, and assess whether your actions will supersede your opponents'.

If someone already has their gun trained on you, the timeline might show that if you start

▼ The blood in *John Wick Hex* is all neon purple, for some reason or another.





HEX - Edgar was loyal to the end, he told Wick nothing. A stronger man than many would be in his situation.

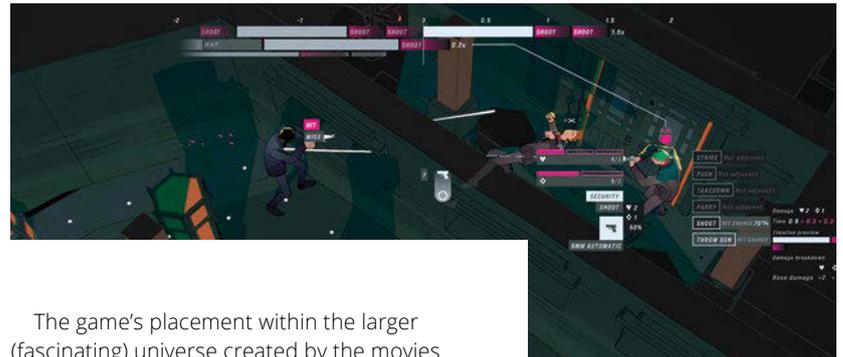
◀ The cutscenes and story are a solid reminder of just how vital Keanu is to making Wick work.

firing, they're going to get their shot off first and hit you – but you have enough time to crouch, meaning you can duck out of the line of fire. It might show that the goon to your immediate right isn't ready to throw a punch yet, giving you plenty of time to pull off a takedown, or push them up against a wall, simultaneously moving away from the gun-toting baddie you've also spotted walking through a door at your six o'clock.

When everything goes to plan, and you manage an efficient run through a level or you scrape through a tense situation, *John Wick Hex* can be extremely satisfying. The methodical planning feels closer to those Guy Ritchie *Sherlock Holmes* films than *John Wick*, though, and while this style of gameplay is meant to capture John's superior instincts and efficiency, the system often feels rickety. Enemies rarely go down from a single gunshot, and there's an associated cost with everything you do. Most actions require 'focus' (which you can recharge when you're not being attacked), healing requires bandages (which are in extremely short supply), and firing your gun requires bullets (which, weirdly, cannot be salvaged from dropped weapons – you need to pick up a new gun when you run out). And, of course, everything takes time. It's difficult to do anything 'cool' in *John Wick Hex*, and the stiff animations often make John feel more like an action figure with limited articulation than an efficient assassin.

Resources carry over across sets of levels, so if you use up all your bandages in the first section of a six-level area, you really can't afford to take further hits. If you find yourself towards the end of a section with low health and no bandages left, you'll need to either complete the rest of the missions perfectly or simply restart the whole set. It's satisfying when you're on top of things and feeling confident by the midway mission, but semi-random enemy placement and the game's punishing systems make a 'perfect' run brutally difficult.

"Punishing systems make a 'perfect' run brutally difficult"



▲ The visual style makes for great-looking levels and ugly character designs.

The game's placement within the larger (fascinating) universe created by the movies could not be duller, either. It's a prequel, with plenty of nods to the High Table and the rules dictating this universe, plus the voice chops of Lance Reddick and Ian McShane in a handful of cutscenes and narration interludes. But John himself is just a cypher that kind-of looks like Keanu Reeves. It all feels like window dressing; some of the terminology and aesthetics from the series are carried over, but the world isn't built on further in any meaningful way.

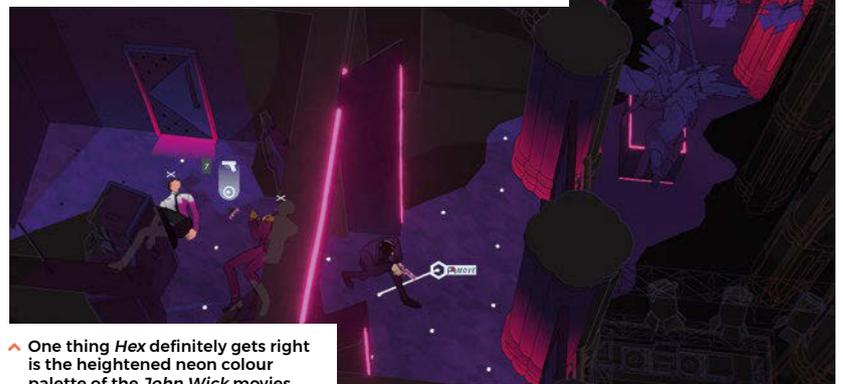
There's a 'replay' feature that lets you watch all your actions play out in real-time after a level's completed.

It's a fun idea, but it reduces even the game's coolest moments to an arthritic display as John very slowly walks through environments, rolling awkwardly, and taking forever to fire off a shot. *John Wick Hex* has its moments, but it feels disconnected from the series that spawned it. 🗨

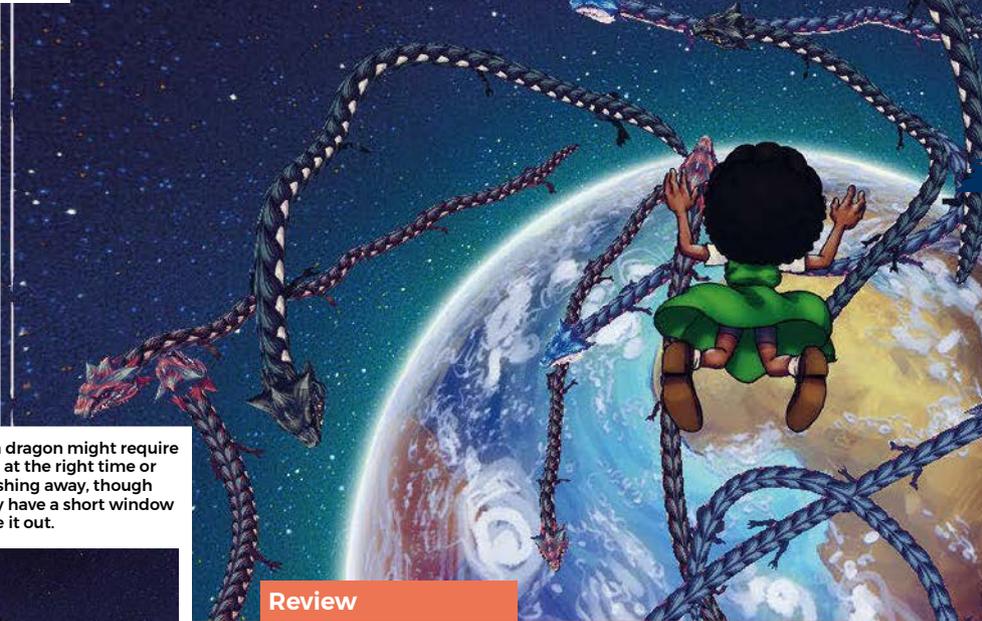
VERDICT

There are some worthwhile thrills among *John Wick Hex*'s frustrating moments, but it's not a great spin-off.

61%



▲ One thing *Hex* definitely gets right is the heightened neon colour palette of the *John Wick* movies.



HIGHLIGHT

Roguelikes may be a dime a dozen, as are auto-runners, so it's to *EarthNight's* credit that it manages to stand out with its striking hand-painted art, from Sydney's character to the many colourful dragons you encounter. The visuals are also complemented by a catchy pounding soundtrack composed by chiptune artist, Chipocrite.

✓ Killing a dragon might require tapping at the right time or just mashing away, though you only have a short window to figure it out.



Review

EarthNight

Chasing the dragons

▲ There's a hell of a lot of dragons, though it's actually possible to skip them during your descent.

Info

GENRE
Roguelike runner

FORMAT
Apple Arcade (tested) / PC / Switch / PS4

DEVELOPER
Cleaversoft

PUBLISHER
Cleaversoft

PRICE
Apple Arcade subscription

RELEASE
Out now [Apple Arcade]

REVIEWED BY
Alan Wen

VERDICT

Stylish design, but a little lacking in substance and replay value.

69%

Forget zombies or nuclear wastelands; an invasion from Asian-influenced dragons is my kind of apocalypse. These serpentine beasts haven't just taken over the world, they're ruling the skies over multiple layers of the Earth's atmosphere. It's why the heroes of *EarthNight* take refuge on board a spaceship until it's time to plunge back down to the surface and slay some dragons along the way.

As either young girl Sydney or older, balder Stanley, you find yourself dropping onto a dragon then auto-running along like a platformer, picking up treasures and avoiding or stomping on enemies that litter its back until you reach the beast's head. There's then a brief window to slay it before you skydive further down through the atmosphere before landing on another dragon to repeat the process.

The threats naturally intensify, and being a roguelike, things get more brutal the deeper you go. You also have more control over your movement than you'd expect for an auto-runner: you can change running speed or, after a jump, hold down to fall faster. This makes it easier to aim for platforms or enemies directly beneath you – the latter able to bounce you up to new heights and secrets, while a bouncing combo of at least five enemies recovers your health. If all else fails, at least your spaceship is always on

hand to whisk you back to safety after the fatal blow, where treasure and gains are tallied up that can unlock upgrades to (hopefully) give you a better shot on the next run.

But while a few mistakes are enough to make a run short-lived, the descent to Earth's surface is also fairly short, especially once it's apparent you can dive past most of the flying dragons; even then there's only a handful of different areas that layer up the atmosphere. Even though you can unlock previously greyed out items in each level, they're mostly the sort that give you a little more

luck, an extra jump here, a temporary shield there, rather than significantly transforming a new run.

As an auto-runner, you might also assume *EarthNight* would make a good fit for

mobile, although Cleaversoft have actually been working on it for many years, eyeing other platforms (including a Switch announcement), only to make the surprise launch for Apple Arcade first. Yet its controls also contain more nuance than a touchscreen can afford, especially for Sydney, who, after a jump, can perform another jump, dive-glide, or make a sharper 45-degree dive – options that are arguably easier to differentiate with individual face buttons. Fortunately, Apple devices now also allow you to pair PS4 and Xbox controllers the optimal way to experience this colourful and stylish (if short-lived) dragon-slaying ride. 🐉

“Being a roguelike, things get more brutal the deeper you go”

HIGHLIGHT

If nothing else, *Rogue Corps* offers a wealth of modes to dig into, including online multiplayer, couch co-op, and workshops where you can upgrade your character's abilities and develop new weapon parts. If only the game underpinning it all was actually fun.

Review

Contra: Rogue Corps

Alien war of attrition

The *Contra* series has always had a B-movie feel to it: the off-brand, muscle-bound heroes clearly modelled on Arnold Schwarzenegger and Sly Stallone; the massive guns; the extra-terrestrial cannon fodder cheerfully cribbed from the *Alien* franchise. *Rogue Corps*, Konami's belated return to the *Contra* series, carries on that tradition – it's loud, it's foul-mouthed, it's violent. Unfortunately, it's also cheap and tacky in all the wrong ways.

Rogue Corps has pedigree, if nothing else: it's directed by Nobuya Nakazato, who previously worked on, among other things, *Contra: Shattered Soldier* and the full-on classic, *Contra III: The Alien Wars*. Stepping away from the side-scrolling run-and-gun platforming of the earlier games, *Rogue Corps* instead emerges as a top-down twin-stick shooter: you roam an alien-infested city, gunning down endless waves of fiends that jump out from behind burned-out cars and buildings.

From the moment you let off your first burst of machine-gun fire, there's a sense that something's not quite right with *Rogue Corps*. The gun noises are tinny and distinctly lacking in oomph; more

frustratingly, your character's movements actually slow down while you're shooting, making this less a case of run-and-gun and more gun-and-plod.

Rogue Corps adds a few mechanics to spice the action up a bit, such as weapons that constantly overheat, forcing you to switch between your two main weapons: the brawny Kaiser, for example, can fire low-powered machine gun shots and explosives but slow three-way missiles. You can also eliminate certain enemies with a finishing move – a feature that might feel more satisfying if it didn't play precisely the same canned animation every time it's activated. There's more variety to be found in the four characters you can choose from (an alien/ninja hybrid, a panda, and an amorphous alien blob named Gentleman), each with their own weapon combos, and rankings to entice you to improve your scores.

The trouble is, none of this adds up to a hill of beans when the action itself feels so rote and lifeless. The best *Contra* games were marked out by their pace and the meatiness of their gunplay. *Rogue Corps*, meanwhile, is distinguished by a camera that's constantly either zoomed out too far from the action, or positioned so that your view's obscured by a piece of scenery in the foreground.

Couple all of the above with some unimaginative level design that simply throws waves of bullet sponge enemies at you, and some of the brownest graphics I've seen in a decade, and you have a game that really doesn't live up to the *Contra* name. This year's seen some great games inspired by the *Contra* series – such as the terrific *Blazing Chrome*, one of our favourite action titles of the year – which only makes *Rogue Corps* look all the more tawdry. 🗑️

✔ Of the four characters we spent time with, Kaiser feels the most balanced. Or at least, we died less when we tackled missions with him.

Info

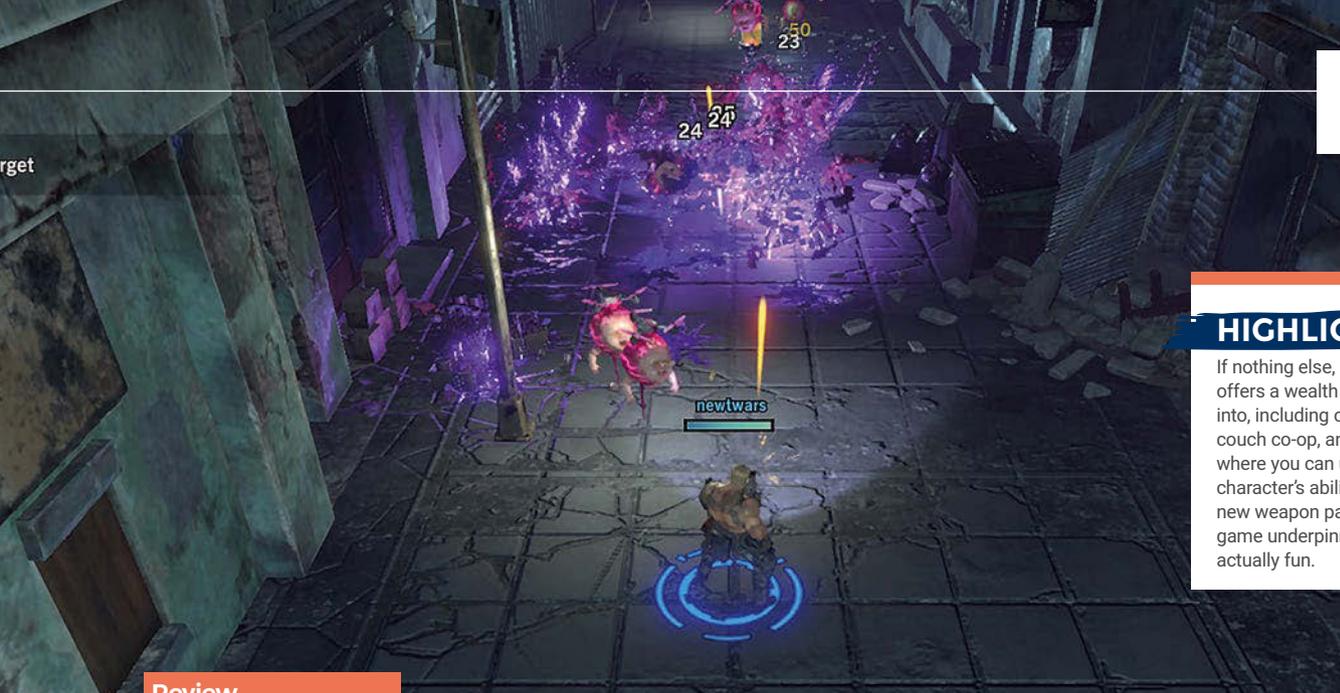
GENRE
Run-n-gun
FORMAT
PS4 (tested) / Switch / PC / XBO
DEVELOPER
Konami
PUBLISHER
Konami
PRICE
£36.99
RELEASE
Out now

REVIEWED BY
Ryan Lambie

VERDICT

This is one B-movie best left on the video store's bottom shelf.

28%



Rated
Review

HIGHLIGHT

From meaty shotguns to missile launchers, laser swords, arachnid limbs, and beams of electricity, there's a huge array of inventive and powerful weapons to use. You're unlikely to upgrade them all in one playthrough, but experimenting with different combinations is the game's greatest joy.

Review

Valfaris

Mega death awaits

▲ Walking tank levels provide a stomping change of pace as you destroy everything in sight.

Info

GENRE
Shooter

FORMAT
Switch (tested)
/ PC

DEVELOPER
Steel Mantis

PUBLISHER
Big Sugar

PRICE
£22.49

RELEASE
Out now

REVIEWED BY
Ed Nightingale

VERDICT

Devil horns at the ready: you'll need skill to rock this retro quest.

76%

This is one game you'll want to wear headphones for. Blasting alien scum has never felt so good, accompanied by the persistent thud of a kick drum, the wail of a melody, the chug-chug-chug of guitar chords.

Valfaris is, in short, *Contra* meets *Metroid* meets Iron Maiden. A screaming metal soundtrack amps up the thrills as you explore the apocalyptic wastelands, alien hives, and catacombs of the titular planet infested by swarms of creatures. The forgettable plot and characters have faint Norse overtones, but they're just an excuse for hard and heavy gameplay filled with rock references – from the hellish wolf and skull-ridden enemies, to protagonist Therion headbanging with glee on discovering a new weapon.

There's a pleasingly retro feel to it all, with its PSone-era side-scrolling graphics and linear level design. There's even a CRT mode that adds scan lines to the visuals. Gameplay is surprisingly elegant and nuanced for such a bombastic game: pistol shots are weak but infinite, while secondary weapons and your shield use up energy that's replenished by melee attacks. And so battles become a dance between attack, defence, and melee against hordes of parasitic aliens and robotic soldiers.

Collecting 'blood metal' allows you to level up your weapons, while resurrection idols have the dual purpose of increasing your health and activating checkpoints, adding a risk-reward wrinkle of more health or less repetition. The levels themselves are also filled with creativity:

there's not only running and gunning, but also platform challenges and on-rail sections. Amongst all this, the metal theme never feels contrived or overdone.

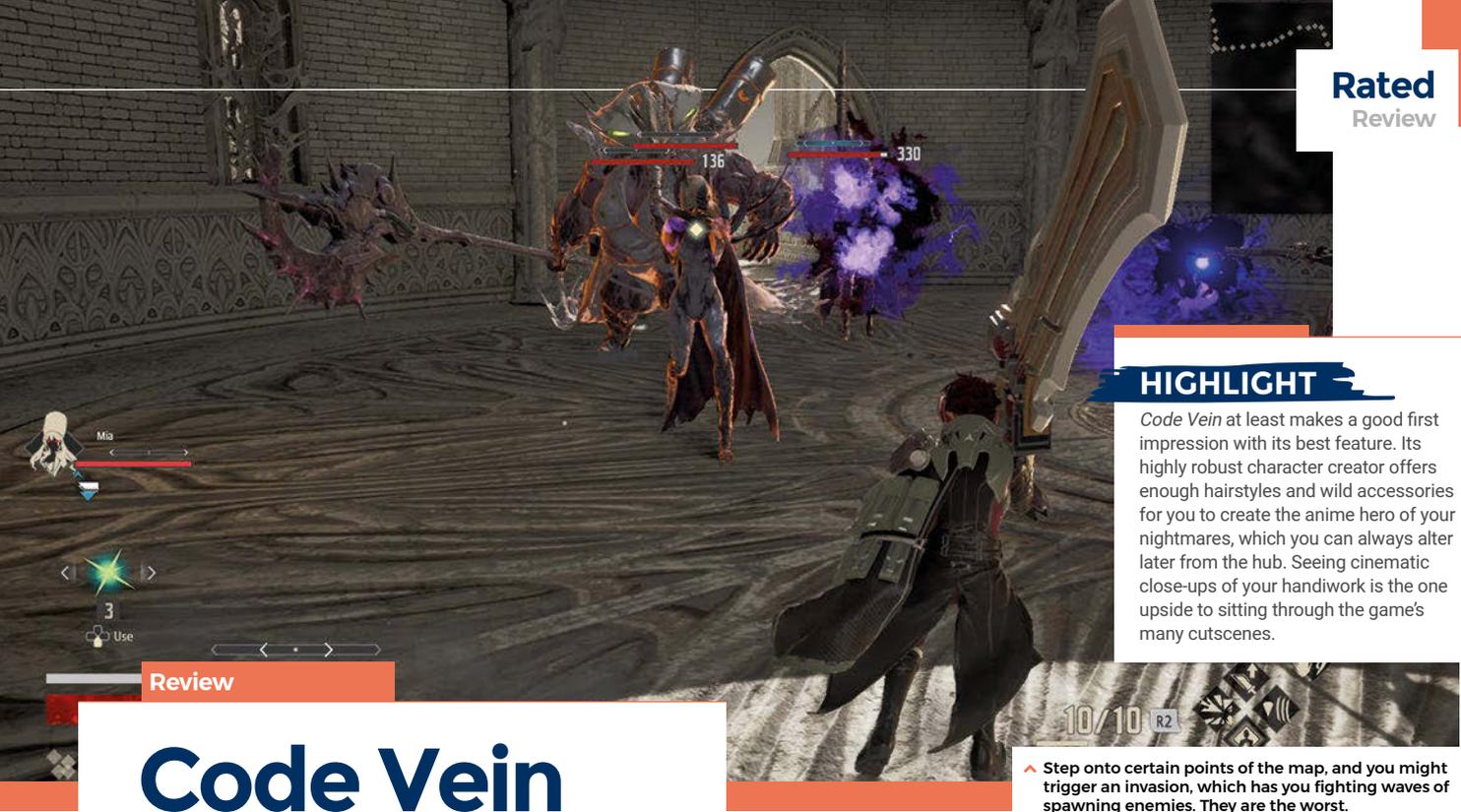
Therion is a little sluggish to control (a dodge manoeuvre would've been welcome), but he makes up for it in firepower. There are tons of weapons to collect, from the obvious pistols and shotguns to more inventive forms of alien torture, yet each feels satisfyingly weighty to wield, and they're given appropriate names like Hellhammer and The Bastard. Your arsenal is given plenty of feedback with enemies expiring in showers of goo, and there's a whole host of ingenious bosses to defeat – learning their patterns and weaknesses is a blast.

Death is frequent in *Valfaris*, but swift restarts numb the pain. Later levels do feel a little cheap, though, by simply ramping up the number of enemies in frustrating placements, instant death situations, and bosses that have an annoying tendency to kill you just after you think you've beaten them.

Even so, *Valfaris* is a slick, six-hour adventure with old-school bite that's more satisfying than smashing up a guitar. 🎸

▼ The pace of the game is relentless, but it's just challenging enough to keep you addicted.





HIGHLIGHT

Code Vein at least makes a good first impression with its best feature. Its highly robust character creator offers enough hairstyles and wild accessories for you to create the anime hero of your nightmares, which you can always alter later from the hub. Seeing cinematic close-ups of your handiwork is the one upside to sitting through the game's many cutscenes.

▲ Step onto certain points of the map, and you might trigger an invasion, which has you fighting waves of spawning enemies. They are the worst.

Review

Code Vein

Sucking the life out of the Souls-like

When you're the publisher of the most influential game series of this generation, it's easy to imagine why you wouldn't want to give it up so soon.

There have of course been other *Souls*-likes over the years, but when it's one from Bandai Namco, the similarities become all the more glaring. It's hard not to see *Code Vein* as little more than a cynical and underwhelming imitator, albeit with a Gothic anime aesthetic. At times, it feels brazenly close, from the familiar canned animation of opening doors to a boss duo that's a straight-up rip-off of Ornstein and Smough. Note, too, that the game's original tagline was 'Prepare to dine'.

This isn't to say *Code Vein* doesn't try to distinguish itself. As a revenant – a human resurrected as a vampiric being – how you play is bound by your blood. Blood codes function like equippable classes, as you start out with the fighter, ranger, and caster codes, while more unlock throughout the campaign. Besides transforming your stats and determining which weapons you

can wield, blood codes also unlock gifts, abilities in the form of special attacks, heals, or buffs, which are limited by another blood material known as 'ichor', which can be refilled by draining or defeating enemies. It's all a bit overwhelming at first, though ultimately these just wind up as a few neat ideas propping up the fundamentally un compelling combat at its core.

Despite a vastly different button layout, there are the *Dark Souls* staples, from stamina management to dodge rolls and parries. What it fails to replicate, though, is the weight of combat that makes melee encounters feel satisfying; attacking enemies is

more of a slog than a methodical challenge to master, not helped by some drab and uninspiring post-apocalyptic environments. You explore the world with a band of revenants, so there's

“Code Vein is a poor person's Dark Souls”

an AI companion with you by default, and their behaviour can be erratic: either they rush head first into an enemy you were hoping to sneak up on, or they execute an outrageous attack on a difficult boss before you've even had a chance to figure it out. You can, of course, opt to solo it, but when the game's designed to be played co-operatively, you're also unfairly handicapping yourself.

In the same way that *God Eater* (created by co-developer, Shift) felt like a poor man's *Monster Hunter*, *Code Vein* is a poor person's *Dark Souls*, dining on the genre's leftovers without bringing anything refreshing to the table. It only feels more dated when compared to the brutal brilliance of *Sekiro*. *Code Vein*'s revenants may be immortal, but they've already aged poorly. ☹️

✓ Your party of revenants increase over time, each with their own tragic backstory. Dialogue and cut-scenes do, however, drag on interminably.



Info

GENRE
Action RPG

FORMAT
PS4 (tested) / XBO / PC

DEVELOPER
Bandai Namco Studios / Shift

PUBLISHER
Bandai Namco

PRICE
£49.99 (PS4/XBO),
£39.99 (Steam)

RELEASE
Out now

REVIEWED BY
Alan Wen

VERDICT

A few interesting ideas can't save *Code Vein* from being an uninspired *Souls*-like that lacks weight and... soul.

48%

HIGHLIGHT

Executions aren't as frequent as the trailers made them seem, and depend on how much damage you've done to an enemy before you parry with a fatal attack. They're incredibly well animated, though, with every crimson detail shown off in extreme close-up.

Review

Blasphemous

Another Metroidvania, for God's sake

▲ The Penitent One wears a 'capirote' - a pointy hood worn during Holy Week in Spain.

Info

GENRE

Action platformer

FORMAT

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

DEVELOPER

The Game Kitchen

PUBLISHER

Team17

PRICE

£19.99

RELEASE

Out now

REVIEWED BY
Diego Arguello

VERDICT

Blasphemous doesn't reinvent the Metroidvania genre, but rightfully earns its place among the best.

70%

At first, *Blasphemous* looks like the kind of game that only cares about gore, with its dozens of gruesome ways to kill a pixel art monster. But The Game Kitchen's action platformer quickly stands out in a crowded genre thanks to its tight combat, contained but well-executed level design, and background story rooted in Spain's religious traditions.

Last year's Metroidvania *Dandara* was similarly loaded with Brazilian inspirations, and starred a woman fighting against slavery during the colonial period. In *Blasphemous*, the focus isn't on gaining freedom; it's about penitence, suffering, and fighting against unholy desires brought to life. You take control of a character named The Penitent One, who roams the plague-stricken land of Cvstodia, where tormented souls provide aid in return for favours, while those who've succumbed to blasphemy have become hideous and deadly beasts.

It's in the slicing and dicing that *Blasphemous* begins to shine, with responsive movements, pixel-perfect attacks and dodges, and parries which leave your enemy exposed to a fatal blow – counter-attacks grant you bonus currency to spend on upgrades. A skill tree lets you increase your moveset with additional attacks; it's nothing new, and a bit fiddly to interact with, but it works. Finely-tuned hitboxes and some grimly inventive sound effects make the combat feel weighty and satisfying, even if it can get a bit repetitive when you're asked to backtrack through familiar areas.

Enemies are terrifying: forget the skeletons and zombies commonly seen in horror games; here,

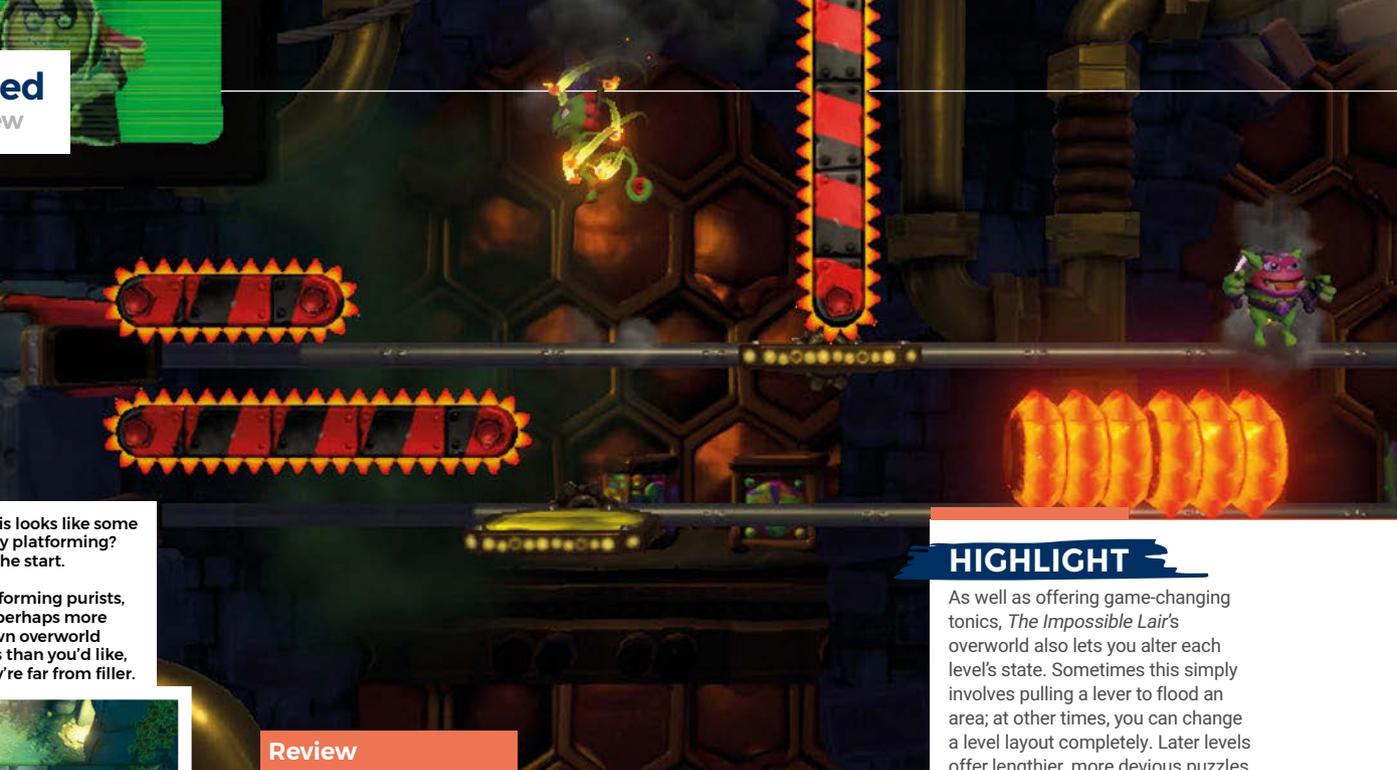
there are disembodied heads, giant babies, a woman who jumps and throws a deadly jar like a handball player, and an enemy that hides inside a broken bell, using it as a cover as they stampede in your direction.

Cvstodia is a horrifying place worth exploring; even if most of its details are buried in environmental storytelling or vague NPC dialogue, it's exciting to walk into a new setting. By now, I've encountered a buried, upturned bell, whose ringing sounds can be heard in distant lands. I've seen sinners turned into statues. I've been attacked by flying spears, angel statues, and whips soaked in blood. Even though *Blasphemous'* mechanics stick rigidly to the Metroidvania template, everything bursts with personality and wit. Like *Dandara*, *Blasphemous* creates a fictional universe by drawing from the tales and history of a distant culture. By the same token, this goriest of action-platformers also fails to bring much that's new to a crowded genre, but it's bloody good fun all the same. 🗨

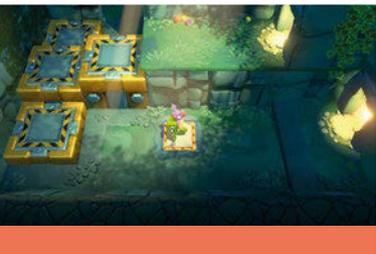


▲ Cutscenes are rare, but they carry a distinct visual style that stays in your mind.

Rated
Review



- › Think this looks like some dastardly platforming? It's just the start.
- ✓ For platforming purists, there's perhaps more top-down overworld sections than you'd like, but they're far from filler.



Review

HIGHLIGHT

As well as offering game-changing tonics, *The Impossible Lair's* overworld also lets you alter each level's state. Sometimes this simply involves pulling a lever to flood an area; at other times, you can change a level layout completely. Later levels offer lengthier, more devious puzzles.

Yooka-Laylee and the Impossible Lair

From Rare-rival to retro recurrence

Info

GENRE
2D platformer

FORMAT
Switch (tested)
/ PC / PS4 / XBO

DEVELOPER
Playtonic Games

PUBLISHER
Team17

PRICE
£24.99

RELEASE
Out now

REVIEWED BY
Alan Wen

VERDICT

Going 2D is a considerable improvement for *Yooka-Laylee*, even if it doesn't match the heights of its influences.

68%

Far from hyperbole, the titular lair is, in all seriousness, impossible. A long gauntlet of traps with multiple bosses and zero checkpoints, it's a challenge to test even the most hardcore of platforming fans. Naturally, I fail at the very first hurdle, prompting my lizard-and-bat duo to be booted back to the game's overworld.

That's when you realise that you're not supposed to waltz right in (though I'm sure it won't deter speedrunning wunderkinds). Instead, the aim of *Yooka-Laylee and The Impossible Lair* is to increase your odds by playing the rest of the game. For each level you beat, you rescue a Royal Bee Guard, each serving as an additional hit point to raise your chances of survival in the lair. Of course, if you think your platforming skills are up to snuff, you can always re-attempt at any time without even seeing the majority of the game, which makes it sort of like Hyrule Castle in *Breath of the Wild*, albeit where the endgame is conveniently located on your doorstep.

It's not the only influence, as *The Impossible Lair* takes a *Shovel Knight* approach to nostalgia by cherry-picking ideas across a variety of games. So, while the 2D platforming recalls *Donkey Kong Country*, the top-down overworld is classic *The*

Legend of Zelda, complete with its own puzzles for progressing or unlocking new levels. On the other hand, a riff on *Yoshi's Island* is less welcome, where getting hit causes you to lose Laylee, who flutters around unpredictably while Yooka's moveset is reduced. It's a woeful state to be in, with just a short window to reunite with the bat before the lizard's toast on the next hit.

"It takes a *Shovel Knight* approach to nostalgia by cherry-picking ideas across a variety of games"

It's far from the only irritation. While *The Impossible Lair* shares a degree of difficulty with its immediate rival *Tropical Freeze*, it rarely reaches the same platforming heights, owing to other frustrations that interrupt your flow. The auto-scrolling of a chase sequence sometimes fails to keep up with your momentum, while a checkpoint placement can often cruelly deprive you of an item you needed to reach an upcoming T.W.I.T coin – coins which aren't just optional collectables, but necessary to unlock paywalls in an overworld that gets ridiculously extortionate over time.

Yet in spite of these flaws, *The Impossible Lair* is nonetheless an improvement over its 3D predecessor, which suffered the misfortune of releasing the same year as *Super Mario Odyssey*. More than a mere spin-off, *The Impossible Lair* allows Playtonic's IP to carve its own identity, even if not all its ideas land successfully. 🎮



Outpost haste

Breaking vegan in Far Cry New Dawn

It's a weird one – to me, the core *Far Cry* series has felt a lot like busywork since the breakthrough game-changer that was the third game. It's consistently gone back to the well, serving up the same action in a slightly prettier skin each time. And while it's not *not* fun, it has worn very much thin for some of us. Oddly, the regular spin-offs have always felt like a bit of a palate-cleanser *even though they're the same thing again, again*. *Far Cry New Dawn* is exactly that. Again.

I think it's that the whole package is tighter; it doesn't promise 50-plus hours of aimless first-person wandering, instead saying 'Nah, much shorter, 20 hours or so' before dumping you into a recycled map from *Far Cry 5* and making you coo at the bright pink spray-paint and aurora in the sky. That almost feels manageable, see. Plus there's a huge focus on taking over outposts, and let's not beat around any post-apocalyptic bushes here: taking over outposts is the best thing about any *Far Cry* game of the past decade.

It's a hint of planning, a slab of scouting, a hearty pinch of improvisation, and often a lot of shooting when it all goes wrong and some idiot you didn't see bonks your noggin with a spiked bat.

When it goes well, it's fantastically satisfying. When it goes poorly, it's fantastically satisfying. I could honestly get by on a series of outposts and no open-world element for the next *Far Cry*. Take notes, Ubi.

“Taking over outposts is the best thing about any Far Cry game of the past decade”

New Dawn is still backed up by the same mechanics from the core titles, so you're still rooting endlessly through boxes and stripping corpses of supplies. But one thing *Far Cry* has done well, in general, since the third game is made what would be secondary pursuits far more tempting. It's not just doing things for a title or a scoreboard place – if you engage in secondary pursuits, you're rewarded with high-value items, things you can't get elsewhere, *stuff* that makes your time with the game demonstrably more fun. I'm talking, of course,

about hunting. It's not a hunting sim – more a case of running until you find where an animal lives, spotting it, then running after it while raining down arrows on its backside.

There's certainly no real skill involved, at least not in the way I play it. And, thankfully, the skinning animation has been entirely removed; while I can stomach murdering hundreds of humans in gory detail, showing me the skin being sliced off a recently killed animal was always a bit *too* gruesome for this vegetable-munching, sandal-wearing Guardian reader. It's just... fun.

It's pretty simple why I'm now playing *Far Cry New Dawn*: it cost about a tenner, it's brain-off fun, and its systems might be older than the oldest of hats (in relative video game terms, at least) but they fly true. I'll take (virtually) reneging on my (real-world) moralising for a monstrous buffalo skin that can be traded for high-value crafting items. Ah, video games. 🐾

Now playing
Outpost haste

Wireframe Recommends



Metal Gear Solid V: The Phantom Pain

PS4, XBO, PC, MULTI
Speaking of outposts, *MGSV* is another one with a superb take on the whole 'attacking a fixed position' thing. Endlessly responsive, it allows you to be as creative as you could ever want to be.



Red Faction: Guerrilla

PS3, X360, PC, PS4, XBO, SWITCH
I'm a broken record recommending this one, I know, but we're talking outpost assaults here, and what better way to do it than with the ability to also smash up all the buildings *in* a base? What better way indeed.



Mercenaries: Playground of Destruction

PS2, XBOX
Actually, thinking about it, Pandemic's explosion-'em-up featured outposts to assault when tracking down its 'playing card' targets – and you could blow the hell out of the buildings there too. Outposts are great.

Killer Feature

Metal Gear Solid: Peace Walker / V



Metal Gear Solid: Peace Walker / V

The Fulton mechanic offered a full ton of fun

KOJIMA PRODUCTIONS / 2010, 2015 / PSP, PS4, XBO, PC, MULTI

It's one of those features that's so embedded in the core structure of the series now, that it's hard to go back to earlier titles and function normally. Extricating prisoners of war, incapacitated enemy soldiers, and other such human, animal, or vehicle targets using the Fulton surface-to-air recovery system changed the very dynamic of *Metal Gear Solid: Peace Walker* – and absolutely for the better.

As with many things in the *MGS* series, the Fulton system was based on real technology – something little used (at least that we know of) in operations around the world, but viable nonetheless. A balloon would inflate and raise a lift line up to the height where a passing aircraft could hook on to it, dragging whatever was attached to up into the air, where it could then be reeled in. How very CIA/*Metal Gear*.

What made it such a wonderful addition to the series was how it opened things up to experimentation, allowing players to factor in a less-than-lethal approach that would actually offer a genuine benefit to them, beyond just being able to screech “I killed nobody!” from the rooftops. It was part real, part silly, and all fun.

See, the concept of a non-violent approach had been flirted with in earlier *Metal Gear* titles, and the ability to kidnap enemy soldiers and ‘convince’ them to switch sides had popped up

on the PSP with the earlier *Metal Gear Solid: Portable Ops*. The Fulton system itself had even appeared a couple of times in titles prior to *Peace Walker*. But it was all of this coming together that made for such a fine mechanic.

In offering the player a genuine boost to their home base abilities – and so allowing them to get more resources and other such trinkets from missions, as well as developing better equipment – the Fulton system wasn't just ‘a thing you could do’. It was an actual motivational factor in reducing the amount of violence you inflicted on your enemies; you would aim to incapacitate rather than kill, before extricating them from the battlefield and (so long as they weren't terrible at everything) bringing them into your private military fraternity.

There were modifications and updates to the system in *Metal Gear Solid V*, but it remained functionally the same – which speaks volumes about how killer a feature the Fulton surface-to-air recovery system was. At least in the games. We'd been nudged towards the anti-murder sentiment of series creator Hideo Kojima for decades, but it turned out we needed a bit more than just that to actually accommodate the play style he was nudging to. Instead, we needed to be rewarded for trying to be the good guys – extracting friend and foe alike with a balloon, a rope, and a plane turned out to be an elegant encouragement to noble ends. 🐄

“The system changed the very dynamic of *Metal Gear Solid: Peace Walker* – and absolutely for the better”

Next Issue

ON SALE 5 DEC

RÖKI

Into the woods with Polygon Treehouse's dark fairy tale

Don't miss out!
Subscribe
PG54

Also

- ▶ The developers evolving the battle royale genre
- ▶ Making licensed games: an indie dev's guide
- ▶ Smash-and-grab in the physics-driven Teardown
- ▶ Redwall: the novels' curious impact on games

Editorial

Editor

Ryan Lambie

Email ryan.lambie@raspberrypi.org

Features Editor

Ian Dransfield

Email ian.dransfield@raspberrypi.org

Sub-Editors

David Higgs & Vel Ilic

Design

criticalmedia.co.uk

Head of Design

Lee Allen

Designer

Harriet Knight

Contributors

Diego Arguello, Lottie Bevan, Konstantinos Dimopoulos, Will Luton, Emily Morganti, Ed Nightingale, James O'Connor, Jordan Oloman, Andrew Palmer, Mark Vanstone, Dan Walters, Howard Scott Warshaw, Alan Wen, Lewis White, Jack Yarwood

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 368887

Distribution

Seymour Distribution Ltd

2 East Poultry Ave, London EC1A 9PT

Tel +44 (0)207 429 4000

Subscriptions

Unit 6, The Enterprise Centre, Kelvin Lane,

Manor Royal, Crawley, West Sussex, RH10 9PE

To subscribe

Call 01293 312192 or visit wfmag.cc/subscribe

Subscription queries

wireframe@subscriptionhelpline.co.uk



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

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





CQ32G1
2560X1440

144HZ	FreeSync	1ms
	Curved	QHD

BEND YOUR REALITY

Available at:



ebuyer.com



aocgaming.com



[@aoc_gaming](https://twitter.com/aoc_gaming)



[@aocgaming](https://www.instagram.com/aocgaming)



[@aocgaming](https://www.facebook.com/aocgaming)