

£3

ALL FORMATS

Wireframe

LIFTING THE LID ON VIDEO GAMES

20 BUILD-EM-UPS

The best construction games you can play today

LEARN UNITY

Make a kart-racing game with all free tools



KOMBAT KLONES

Inside the world of gory nineties brawlers

Issue 38 £3

wfmag.cc



MASTERPLAN

Create robots and just about anything else in Main Assembly

PHILIPS

Monitors

get
in the
moment

innovation  you



Momentum 32" 4K UHD, LCD display with Ambiglow
326M6VJRM

4K
Ultra HD



VESA CERTIFIED
DisplayHDR™ 400



UltraWideColor



Adaptive Sync

Available at:

amazon

ebuyer.com

SCAN[®]

The game journo's guide to gamifying girlfriends

“We want violence, people’s heads exploding, fast cars, big jets, and gouts of hot arterial blood splattered against cobblestones.

We want wars and vast armies of ourselves crushing other vast armies of people different to us into the dust.” According to the May 1999 issue of PC Zone magazine, it’s these violent delights that draw men to video games. The only problem? Women are more interested in stuff like “talking to people, reading books, watching films, living life.”

At least that’s the relationship conundrum posited by PC Zone’s ‘How to Get Your Girlfriend Into Games’. The article goes on to offer a twelve-step programme that guys can use to introduce their girlfriends to video games, including picking the right game (complete with talking points to sell her on selected titles), tidying up their gaming space and deleting any porn from their PC before getting started, and, in a *Clockwork Orange* twist, adjusting the room’s lighting so “she has nowhere to look if she gets bored.”

Amidst PC Zone’s attempts to mine the ‘nagging girlfriend’ stereotype for comedy, it’s a boxout exploring the alleged science of why women aren’t interested in games that presents its most laughable claims. Waving its hand vaguely in the direction of scientific studies, the piece suggests: “For some women, the 3D space and layout of an area in a game like *Quake* is not immediately obvious to them. Tunnels which lead off from a room, or even the entire architecture of the room itself, may be ‘invisible.’”

I’m not so sure explaining to your girlfriend that her lady brain might be incapable of perceiving complex 3D spaces is the best way to get her into video games.

PC Zone wasn’t alone in running dating tips alongside its more conventional game reviews and



JESS MORRISSETTE

Jess Morrisette is a professor of Political Science at Marshall University, where he studies games and the politics of popular culture.

strategy guides; similar pieces appeared in many gaming magazines at the turn of the millennium, likely driven by the then-burgeoning popularity of ‘lad mags’ like Maxim and FHM. For instance, Official U.S. PlayStation Magazine published ‘10 Games Your Girlfriend Will Play’ in its January 1999 issue, again presenting the girlfriend as an adversarial figure who must be persuaded – or even tricked – into giving games a whirl. Meanwhile, the February 2000 issue of PC Accelerator tweaked the formula with ‘A Game Geek’s Guide to Getting Girls’, abandoning the pretence that its readers might already be in a relationship. In a passage that reads today like a severely downvoted Reddit post, the article asks, “What if we’re so good at gaming, it somehow triggers an ‘I want the alpha male’ response in females?”

Make no mistake: these articles were intended, by and large, as satire. While the humour falls flat today – and likely didn’t bring the house down 20 years ago either – I doubt readers were meant to take this relationship advice too seriously. Nevertheless, these relics of the late 1990s and early 2000s have a lot to teach us about gaming culture, both then and now.

There’s, of course, the implicit assumption of a straight male readership that forms the basis of these articles. There’s also the reinforcement of a certain brand of dude masculinity, defined by a love of violence and retrograde attitudes toward women who are simultaneously presented as both objects of desire and obstacles to boys having their hard-earned fun. Perhaps most significantly, there’s an underlying judgment about who video games are for. Mainstream games, these articles imply, are made for men. Women – including girlfriends – are outliers, welcome to join the fun by invitation only. It’s a lesson certain pockets of gaming culture are still struggling to unlearn today. 🎮

#38



Contents



Attract mode

Interface



- 06. Main Assembly**
An early tour of Main Assembly's robot build-'em-up
- 10. Inkulinati**
It's turn-based strategy – with a dash of Monty Python
- 12. Occupy White Walls**
The gallery sim that asks: what if art was actually fun?
- 16. Incoming**
Fleeing foxes, killer penguins, and biopunk space stations

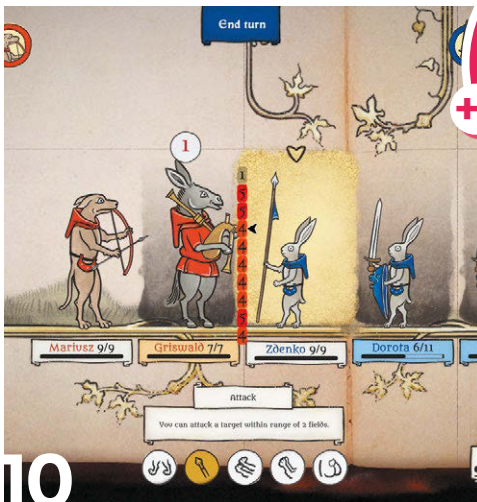
- 18. Construction games**
Our pick of 20 great building sims you can play today
- 22. Attack of the Klonos**
Inside the weird world of nineties Mortal Kombat-alikes
- 44. Unleash the hounds**
How game designers recreate the thrill of the hunt
- 50. Bluepoint**
The masters of ports and HD remasters: profiled

Important!



60

GET
3 ISSUES
FOR £10
+ FREE BOOK
see p54



10



30

Toolbox

- 28. CityCraft**
How small details can enliven a game city
- 30. Karting in Unity**
Follow our guide and make your very own mini racing game
- 36. Narrative design**
Analysing Firewatch's peerless introduction
- 40. Source Code**
Code Robotron: 2084's twin-stick shooting action

Rated

- 56. Nioh 2**
Team Ninja serve up a punishing action sequel
- 58. Resident Evil 3**
Nemesis returns in Capcom's 2020 horror remix
- 60. Animal Crossing**
Time to dabble in the stalk market again in New Horizons
- 64. Backwards Compatible**
We take the PC Engine CoreGrafx Mini out for a spin

WELCOME

I'll dispense with my usual 200 words of random thoughts and whimsy and get right to the important bit: as of issue 39, Wireframe magazine will be going monthly.

There'll be a brief gap between the edition you hold in your hands and the next one, to give us time to write all the words and generally make the transition, but Wireframe will be back even bolder, better – and bigger – than before. The new 116-page issue will be packed with more in-depth features, more previews and reviews, and more of the guides to game development that make the magazine what it is.

The change means we'll be able to bring you new subscription offers – you can read about those on page 54 – and generally make the magazine more sustainable in a challenging global climate. As for existing subscribers, we'll be emailing you all to let you know how your subscription is changing, and we'll have some special free issues on offer as a thank you for your support.

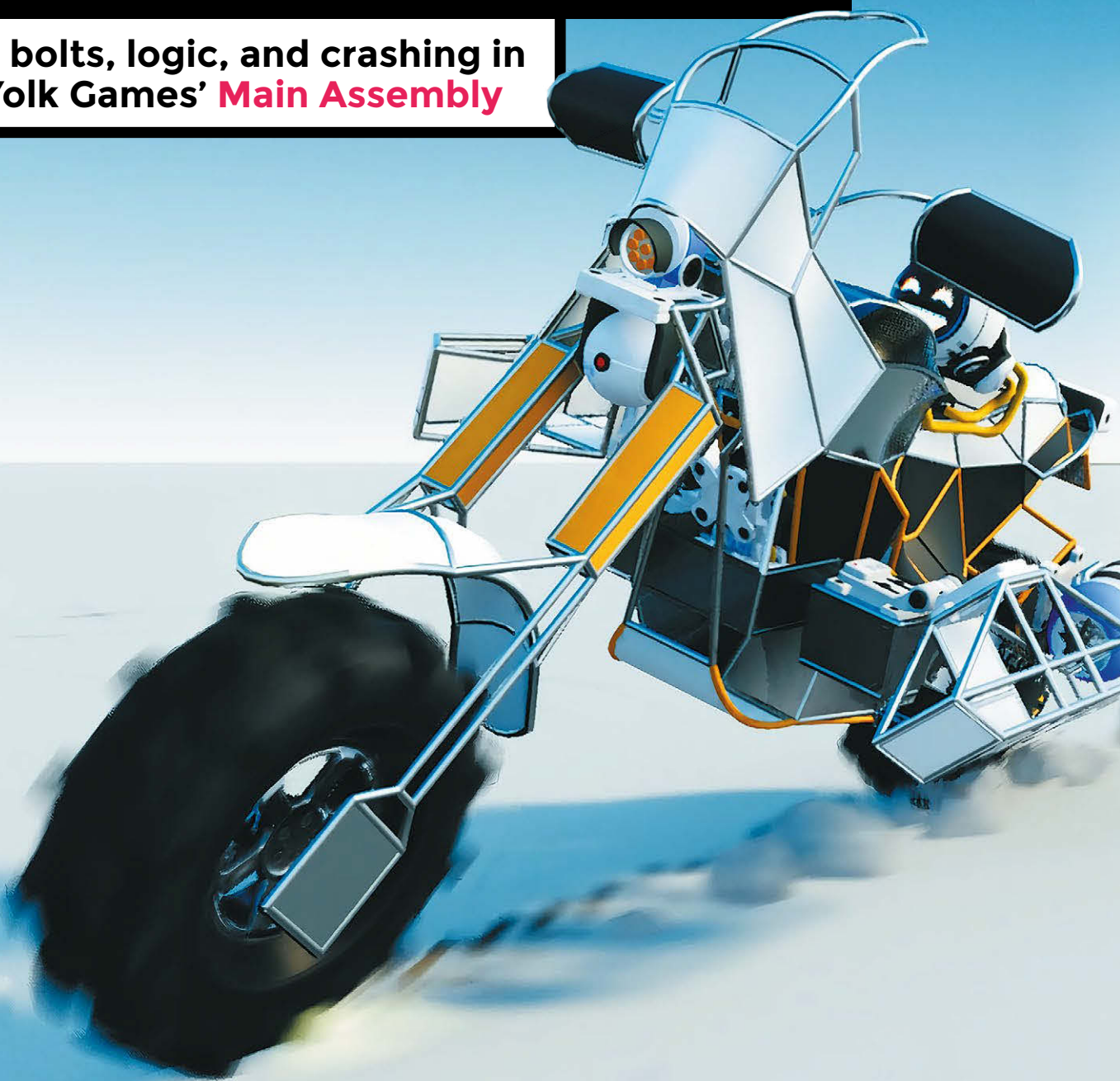
That first monthly issue will be out on 4 June. You'll be able to order a copy online, or you'll find it in selected supermarkets and newsagents if you're out shopping for essentials. It's been a pleasure to bring Wireframe to you, gentle reader, each fortnight. We hope you'll join us as the magazine moves into its next, exciting phase.

Ryan Lambie
Editor



IF YOU BUILD IT, THEY WILL COME

Nuts, bolts, logic, and crashing in
Bad York Games' **Main Assembly**





Crafting is big business in modern gaming, with all manner of titles offering the chance to go forth and create... *something*. *Main Assembly*, from Swedish studio Bad York Games, boils things down a mite; rather than looking at players creating homesteads and communities, or even worlds, it focuses instead on creating devices. Machinery. Vehicles. Robots. Lego with a brain. Modern Meccano. Half-conceived cars you can happily crash into walls. That kind of thing.

Of course there's the crafting, featuring modular parts and fully customisable looks (yes, everything can be yellow if you want) – that's the part where you snap everything together to give it form using *Main Assembly's* freeform creation tool. But once it's made, it's not left to its own dumb devices – you're able to do some light coding, using a visual programming interface to set up controls, logic, even automation if you're good enough. Yes, your half-conceived yellow cars can happily crash into walls *by themselves*. Failing that, you can just use a premade logic system, or look to the community for help. If you *must*.

We spoke with Joel Jonsson, CEO of Bad York Games, to find out what it was that drew this team of triple-A veterans (with credits on the likes of *Doom*, *Gears of War*, and *The Division*) to build a creative suite, what it's like to make a fresh start as a new studio, and just how much everyone at the developer embraces the culture of *mucking about*. We're not sure if it's everyone, but Jonsson's definitely a fan.

How did *Main Assembly* come about?

The idea itself was that we [Jonsson and Michael Paixao, co-founder] played a lot of other crafting and building games. We really enjoyed them, and we always had this fun

teamwork of me loving to do the complex things while he did all the pretty things, so we co-built a lot of things in *Scrap Mechanic*, and *Space Engineers*. And then it was just, 'Oh, I wish this game had this feature from the other game, and I love this thing, and what if they could do it this way instead?' Then we just wanted to do it ourselves.

You've said your team is having fun working on the game – why do you think that is?

I think it's all of us developers come from a triple-A background... It's so different from being a small cog in a machine or something like that, to now be able to work in a small group of developers where we use a pretty flat hierarchy, and we can all impact the game and come with input and ideas and just try to make a better game and get your

“BY BEING A SMALL STUDIO, YOU GET TO DO VERY DIFFERENT THINGS... IT'S VERY RARE THAT I FEEL BORED WORKING ON THIS GAME”

voice heard. And on top of that, by being a small studio, you get to do very different things. I mean, game development, even at a triple-A studio, can be vastly different from day to day. But at a small studio, it's so much more, there's always a new problem, and it's very rare that I feel bored working on this game.

When something's focused on individual creativity and a sandbox style, it can be difficult to focus on the game part. How have you addressed that element?

We know of that issue, and we've been trying to tackle it, but we have plans to make it more goal-oriented during Early Access. We want players to be able to create their own challenges and races, things like that.

Because [at the moment] you can only build your robot and everything inside of the world.

Right now, we try to tie [players] down a bit through the single-player challenges, where the player can learn from the game there and get more of a flow of how to play the game, more or less, and to sort of give them the creativity. Because in the beginning, if you just give them a sandbox, they don't know what to do.

What kind of player is it aimed at?

It feels almost like a constant battle of opinions, where we want it to be easy to get into, but we really want the depth of digging into and making things as optimised and amazing as possible. So let's say we have a part – the RSC thruster – [which] can help you stabilise a vehicle, or if you're trying to do a quadcopter, it can help you fly stably, but it's not super-strong so you can't do very big things, it's too underpowered. If you wanted it as good as possible, you'd have to do it in a more complicated way yourself. So you get this

sort of easier way out, but it's not the *best* way, usually.

So we try to do it somewhat in that direction, but there are so many newer things... We're really trying to make it easy for the player to figure out the control scheme; we have a pretty long tutorial, one of the longer tutorials in the game. It might scare people when I say that. But we try to make it more fun and tied into the storytelling a bit, having the humour and everything.

So, even though it's trying to guide you, it's not necessarily forcing you to do things because it tells you, 'You can do this thing this way'. The only difference is that you get a diploma for doing what they say, or you can just [say], 'Nah, I don't care', and drive into the goal and progress to the next map. ▶

So we try not to be punishing if people don't really care about those things.

There's a bit of a split between actually designing machines and then programming them. What if you're good at one and bad at the other?

There are two approaches we have to that. One is that we tried to auto-program for the player. So when you place your docking station, which is the driving seat for your character, it will start to connect the steering and driving and everything, and will connect thrusters to left shift, so it's sort of like a boost. And that is a way to not only remove the programming in a way for the players, but when you add something new, and you open the program in the menu, you will see how these things work. So it will sort of teach or make the player understand. [They'll see if they] do it the same way, they can make this piston go up and down, things like that.

Then we have the second approach, which is something we've been introducing more and more now. It's something we call kits, which is a pre-built thing of the chassis parts where you can just place it and have it do something. For example, we have a few default kits that players can use, like a forklift where it's pre-built with a piston and a forklift part and everything, and the programming to make it easier to use.

The only thing that you need to do is connect an input saying 'When I press E, the forklift should go up'. And we've done similar things right now for Early Access, for example, [where] we're going to have these walker legs. So if you want to create a spider or any other walker, then you just need to place that part, and inside it, you can just see the forward, backwards, left,



^ Your goals are fairly simple; it's how you go about approaching them where the real creativity comes into play.

right, spring, and you jump. To make that work, you only need to connect it. And if you don't understand all the programming behind it, you can open it in-depth and see it within [its] own CPU controller, how it was programmed, and try to reverse-engineer it if you so wish.

"I'VE NEVER HAD THIS MUCH FUN DRIVING INTO WALLS IN A GAME BEFORE"

We want these presets to help the players understand. And that can also help with, for example, Steam Workshop. You download something cool, and then you might wonder, 'How can I do this cool thing?' You can open the programming there as well.

Are there any aspirations for *Main Assembly* to be a kind of teaching tool?

We have explored that option a bit, and we're actually still looking into it, because it's a pretty good way of understanding rotations and inputs. If you work in a STEM field or are interested in it, *Main Assembly* can help you think [about that]... We have had teachers contact us saying they would love to have this in their school as a way of teaching programming, seeing it visually [and] immediately on-screen. It helps people to understand.

How does multiplayer factor into all of this?

One thing we learned pretty early on was that we really wanted a great 'undo' system, because if you do something incorrectly, it's so nice to just undo it. We really wanted this nice flow of undoing and redoing. If people are going to build on the same thing, that system would not work; it would be weird if you undo what they're doing, but you just wanted to undo your thing. So that was the issue there, and we prioritised the undo system for crafting.

But [for] multiplayer, we have a goal in mind that we haven't implemented yet – we want multiple people on the same creation. I mean, the easiest thing for that is a tank; one can drive, and one can turn the turret or something. Multiplayer in these types of games can be very difficult – because we have this freeform crafting system, there are a lot of rigid bodies. And then if you try to replicate that over the network and everything, it can give you a lot of issues.

It's physics calculations and multiplayer, the two things that love to go hand in hand in all games [laughs]. So we've had a lot of issues with that, and trying different ways of optimising, how players can interact with each other, how to handle latency in the best ways. We're still experimenting and trying to find what would be optimal, because it's fun to crash into each other or stand in the way of your friend when they're trying to do something.



^ Designs can be as simple, complex, or plain daft as you want them to be. They won't all work, though.



▲ Probably salvaged from an early design sketch at Tesla – we've no idea.

Can we still have fun just breaking things?

Yeah. I find with our destruction system, I've never had this much fun driving into walls in a game before. It's like you're doing something wrong or you do a jump, and you have a really rough landing, and then you'll see your wheel rolling past you. That feeling is, even though you failed, [just] so much fun. And when these things happen, we just laugh. Some of us at the office are constantly recording to save highlights, and we just share it [around the office].

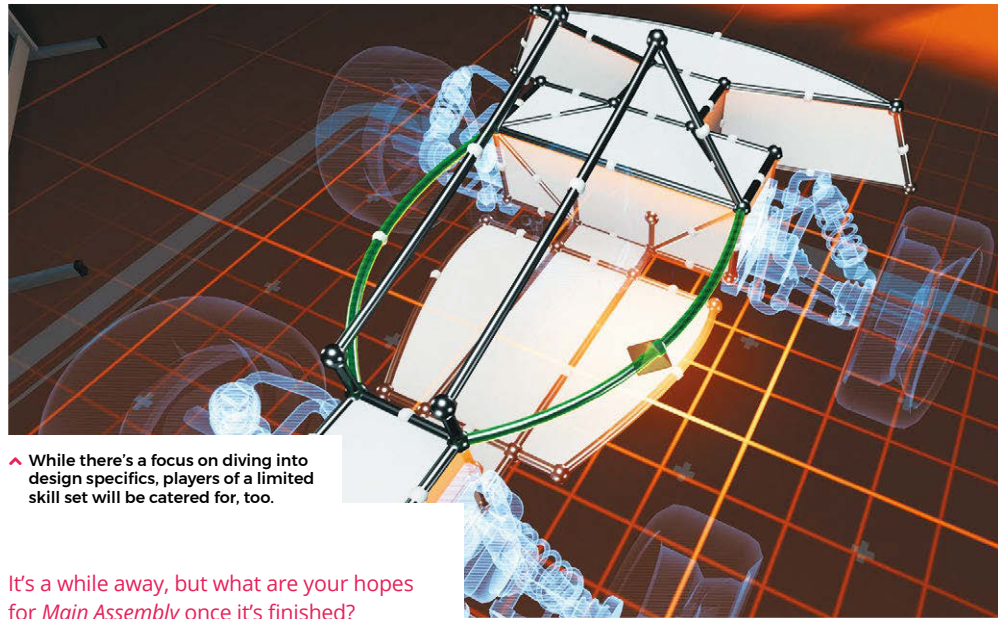
Balancing it is something that we work a lot with and try to make it always feel good. [In a recent beta] there were some optimisations in the destruction system that caused things to break way easier than they should, and we realised it just wasn't fun for everything to explode so easily.

Have you put together any modifications of your own to the Unreal Engine?

No. We ripped out PhysX and put in Bullet for our physics engine. We needed it to be more accurate. PhysX has had some nice updates now, so we're interested in seeing if that would be better, but that's nothing we have the time for now. But Bullet is pretty reliable compared to a lot of other physics engines. That's the biggest change to the engine.

Are you still able to do much of the programming yourself?

Yeah. I get to work. I've learned now how to work well between the business side and the development side. It took some time to get into it. I think it was mostly because I was trying to learn more of the business side, and that learning period was the thing taking more focus than actually doing the business stuff. But when you have learned enough stuff – I wouldn't say everything because that would be a lie – then it's easier to handle new things that pop up. And it's also easier to juggle tasks around... But I would say the first year especially was really tough to try to understand things.



▲ While there's a focus on diving into design specifics, players of a limited skill set will be catered for, too.

It's a while away, but what are your hopes for Main Assembly once it's finished?

For me, I want *Main Assembly* to be the best building game there is. We try to make everything feel as good as possible, and we want to [influence] other people to do similar things. We hope that this can help boost other people's creativity, maybe even in a similar way to how *Minecraft* has taught people and actually opened up creativity in a lot of players. I would love to see it as an inspiration to people, because I think we really need creativity right now. It's a thing that often goes missing in life. ☺

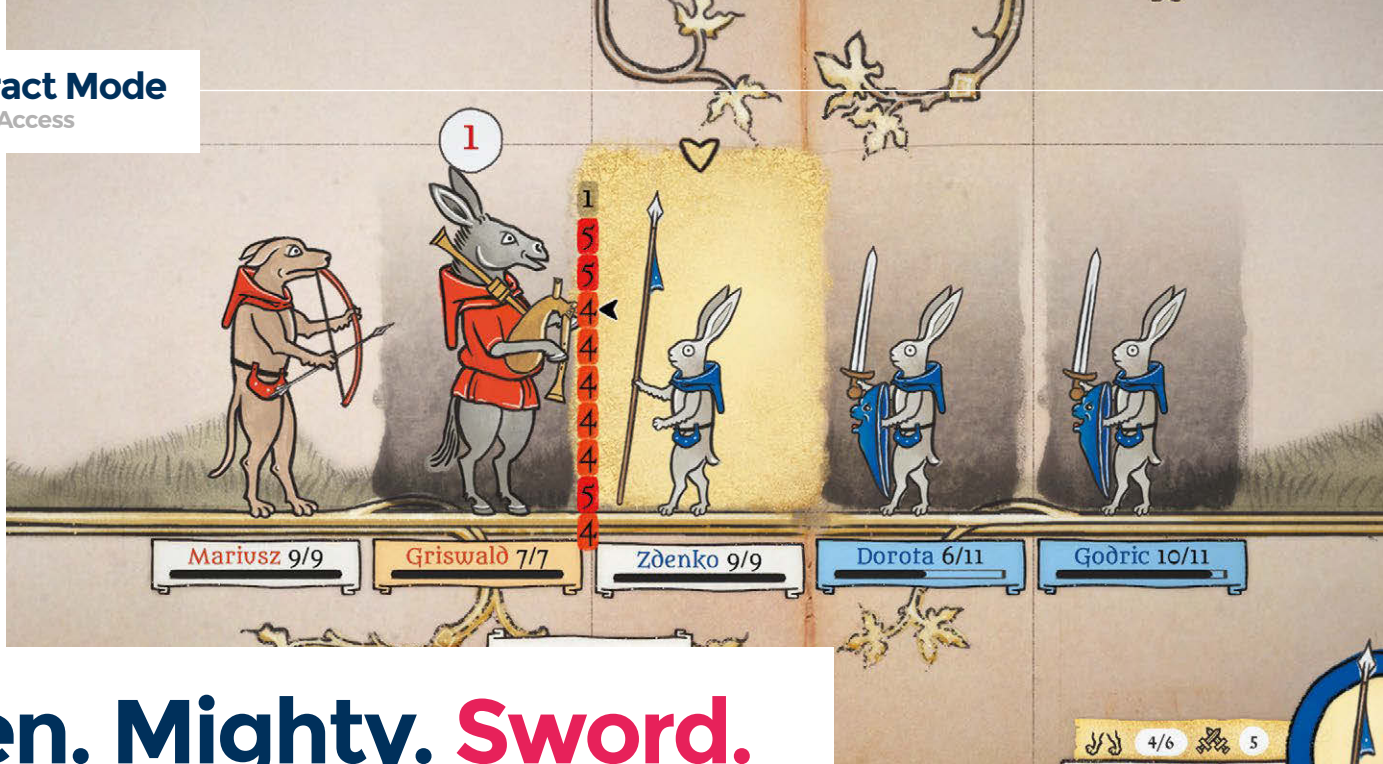
Main Assembly leaves Early Access and releases on 11 June.

▼ Assembly is the sense of humour behind everything – it's serious creation, but with a lighter heart to its presentation.



▼ A tank? A multifunction street cleaning machine? They're all possible in Main Assembly.





Pen. Mighty. Sword.

It's Monty Python's Turn-based Combat, aka Inkulinati

Info

GENRE
Turn-based strategy

FORMAT
PC

DEVELOPER
Yaza Games

PUBLISHER
Yaza Games

RELEASE
TBC 2020

A

concept that came about as an end of school project, *Inkulinati* – a portmanteau of ink, illuminated, and illuminati, fact fans – takes place entirely on a medieval manuscript, and sees comical battles between various animals take place in front of the player. Visually, it's unique. In motion, it's striking. And it all transpired because some students were amused by 700-year-old doodles.

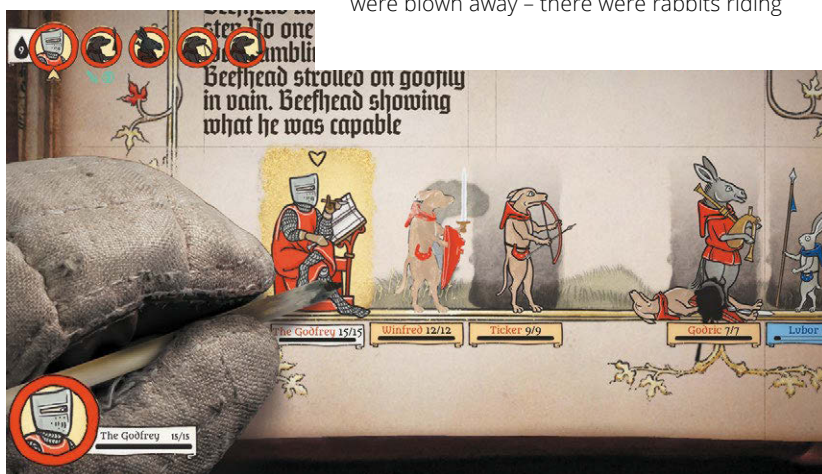
"Back in the Warsaw Game Design School days, we needed to prepare a prototype of a game," explains game designer, Wojtek Janas. "We had no idea what we wanted to do. Nothing gave us that 'Yes, let's do it!' feeling. And then one day, Dorota [Halicka], our art director, came in and showed us medieval marginalia – the crazy and bizarre doodles of medieval monks and scribes. When we saw these drawings, we were blown away – there were rabbits riding

on snails, donkeys playing harps, people with trumpets up their bottoms... It's like a whole new world opened up to us all."

As such, the theme and art style came first with *Inkulinati*, with the genre "falling naturally into place" soon after. While the main thrust of the action is turn-based strategy in a recognisable form – different units with different strengths and weaknesses, different approaches for players favouring alternative strategies and so on, this too ties in with the game's look. As it turns out, the action is being drawn on the fly by the medieval illuminators who inspired *Inkulinati* – and sometimes, just like when an animator wanted to rub out Bugs Bunny, the hand that holds the quill gets in on the action, too. Fists can smash the opposition, units can be moved around by hand, obstacles can be drawn to slow down progress – it's a neat touch, mixing mechanics with the overall presentation, and goes to show this is more than just a gimmick from the young team at Yaza Games.

From a storyline perspective, these manuscripts come alive thanks to 'the living ink', a substance that brings these animals to life on the pages, and one hoarded and used solely by... well, the secretive Inkulinati. Obviously. But it's not a game aimed at conspiracy theorists; instead, it aims to bring in strategy fans alongside those who enjoy medieval themes. "We hope we can reach others, too," Janas says. "People that enjoy a *Monty Python*-esque sense of humour, playing against their friends, and those that like to play something perhaps a little bit different. But deep down, I guess we are making a game that 700-year-old medieval

▼ The artist's hand coming into play harks back to the old *Looney Tunes* jokes of the same style.





scribes would want to play themselves. That's our target market!"

Jokes aside, Yaza Games is... putting the jokes aside. *Inkulinati* isn't a comedy game per se; while its presentation raises a fair few smiles and it's being marketed as 'Look at this kooky project!' (it does stand out, after all), the underlying game is focused, and the attention to detail requires the team maintain a level of accuracy in what it presents. "As fun as it might be to have a bazooka in our game, it just doesn't fit the overall vision," Janas says.

"However, most of the work was done by the medieval illuminators – those guys had a wicked sense of humour."

That accuracy has been a bit of a negative – in the loosest sense – to the team, mind. "These artworks were created in a specific time, with specific meanings," Janas explains. "It would be disrespectful to just ignore the deeper historical context, and just use them willy-nilly, however much we like them. So we do a lot of research behind each character, and we work closely with a professional medievalist (the absolutely fantastic Łukasz [Kozak] from Discarding Images) to make sure that the time period is relevant, that the character's moves and skills are realistic, and to make sure that it all fits the larger context." Respect and honour is the approach, rather than just co-opting an artistic style and cramming in a game around it. Plus, in using such a unique look for a video game, Yaza ends up without much else to draw inspiration from, at least from a presentation angle.

"It's always handy to see what other people did with the style that you developed, but in our case, there aren't that many," Janas says.

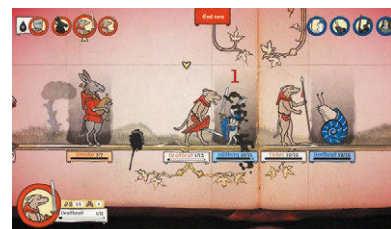
"*Marginalia Hero* is the only other game that comes to mind. So we have to 'invent' mechanics or UI ourselves and test it over and over again to see if they 'work'. But they're not really negatives per se – it's fun to see and work things out as you go along. You feel like an explorer a little bit, or a mad scientist."

With a small core team behind *Inkulinati*, progress has been slow but steady. Helped

out by folks outside the studio, like the aforementioned medievalist, an external musician, and with dev and QA support

from Pineapple Works, the project is coming together nicely. And just like the simple inspiration for the game, Janas has simple aims for *Inkulinati* once it releases: "Honestly, we hope that the game will be well-received and that we can sell enough copies of it so that we can make a second game. That would be fantastic." 🍷

"The art style came first, with the genre 'falling naturally into place' after"



Many dog owners will have seen this sight on a field at dusk, though the dog won't have been carrying a sword and the rabbit won't have had a shield. Hmm.

The presentation really is sublime, and holds you tightly in its thematic grip.





Occupy White Walls

Bringing browsing to a wider audience

Info

GENRE
Museum curation-
'em-up

FORMAT
PC

DEVELOPER
StikiPixels

PUBLISHER
StikiPixels

RELEASE
Out now (Early Access)

You can jump into *Occupy White Walls* with one goal in mind, if you like – that being to construct a museum to your exact specifications and fill it with the sorts of exhibits you want to see. You could leave it at that and have an attractive proposition – something a bit different, and something with a definite allure beyond the usual array of titles to let you shoot/jump/kick balls endlessly. It's in adding an MMO aspect, though, that developer StikiPixels really lets *Occupy White Walls* shine. With a level of collaboration underlying everything you do (should you want it to, at least), you're able to construct, curate, and even discover artistic works alongside others. As Alex Spyropoulos, creative director at StikiPixels, says: "It is a building MMO, with real-world art."

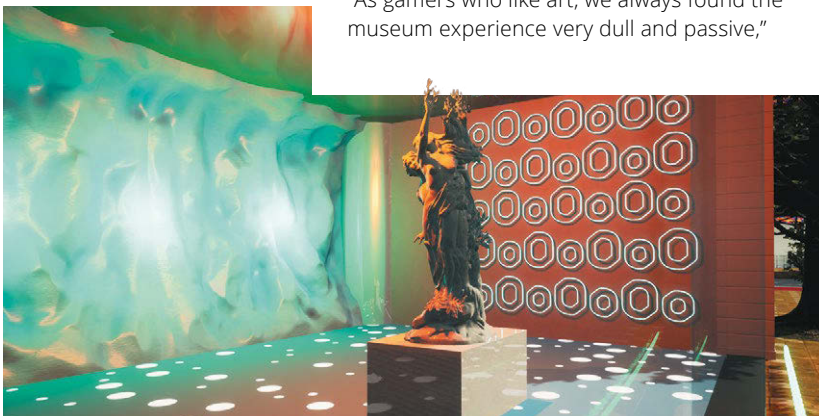
It's not your typical video game idea, so we asked what it was that inspired the decision to make this gallery/space/museum-'em-up. "As gamers who like art, we always found the museum experience very dull and passive,"

explains Yarden Yaroshevski, CEO of StikiPixels. "Visitors have no agency; no-one lets you redesign the Tate how you'd like it... it's like visiting someone else's taste in art. Games are the opposite – as the player, you have power, and the world revolves around you, so one day we asked ourselves 'What if art was *actually* fun?'. When we dug deeper, we discovered there were (almost) no games about art, so..."

As we all know – because we're smart and cultured and that's why we read (or work for) *Wireframe* – art is very much subjective, so making *Occupy White Walls* a game, or experience, that just presents the players with a predefined selection of *things* was never going to work. The community aspect was always necessary to make the title a success, but even with that in mind, the developers were taken by surprise at just how positive the response has already been. "[The community is] super-creative and engaged and has a ton of great ideas," Yaroshevski says. "On the other hand, sometimes we have ideas we think are awesome, but players don't like them, so we change them. We really can't stress enough how active the community is; they deserve at least half the credit for *Occupy White Walls*."

Another aspect of a community focus ties in with *Occupy White Walls'* approach to development, releasing first as an Early Access title, then making its way to Kickstarter to raise funding for further development. It has... not gone too well, falling short of the full £100,000 it asked for, but it hasn't seen the project cancelled or disappear – something you would likely see had this gone the more traditional

▼ At its simplest, *Occupy White Walls* is a virtual art gallery, with all that might entail.





route of 'fundraise first, make game second'. It just means there'll be less cash for the studio to pump into creating more walls, doors, staircases, umbrellas, origami, stones, and kitchen utensils for people to play around with.

Setbacks aside, the ambitions for *Occupy White Walls* remain the same as ever: it's about opening up the world of art, galleries, museums, and so on to a broader audience; one that might not think itself so interested in this particular world. "It would be easy to say [we aimed solely at] creative types, but that's not totally true," Spyropoulos says. "It is very important for us to open up and bridge the two worlds: art and games. So I am particularly pleased when we have players who play *Fortnite* or *Battlefield* – accountants, policemen, [and so on]." Yaroshevski offers a bit more clarity: "Our audience is creative gamers," he says. "We like to call them 'art curious' – people who are a bit interested yet underwhelmed by traditional museums and galleries, that are often patronising. Many of our reviews are along the lines of 'I'm not an artsy person, but I just spent three hours with Egon Schiele...! This is brilliant; we love it!'"

“One day we asked ourselves, ‘What if art was actually fun?’”

Occupy White Walls backs up its ambitious concept with a bespoke AI system called DAISY, a curation aid to help players discover new art they might not have even known they were interested in. "One thing we became increasingly aware of as we dug deeper and spoke to artists is that the current 'old-school' gallery system isn't working for most artists, in the way it's supposed to when it comes to connecting artists with people who like their particular work," Yaroshevski says. "This led to the realisation that any game/platform would need to have some very clever AI that can do that – successfully guess people's tastes in art and/or analyse artworks and find the right people who would appreciate them.

"This level of AI was, for a long time, very expensive, but with recent developments, it became more viable," he continues. "We then discovered it was much harder than we imagined, but we already started and eventually succeeded... Basically, the world caught up with our vision." By no means was this an easy achievement for the team of 15, working remotely with heavy reliance on both Unreal and

Slack, with AI dev described as "easy on paper but bloody hard in reality." But with a solid mix of persistence and intelligence (of the organic variety), the DAISY system was born; a tool to help art-adjacent types get more into something that might otherwise be a bit overwhelming to them. "Also, the players are super-creative and are eager to help," Yaroshevski adds. "So we often listen to their ideas and incorporate them."

With ambitions to bring art, architecture and design to a wider audience, *Occupy White Walls* still has a way to go. But it's getting there, and with the help of an incredibly dedicated community, the project is trundling along – even with the Kickstarter falling short. And the

ambition sounds like it will just continue to grow, as Spyropoulos goes into some detail: "We want to have a lot of in-game creative tools, to have various activities like creating puzzles, add scripting, music composition tools, even advance locomotion for acrobatics and flying," he says. "We want to become an important platform for artists, and become a social place to hang out. We want to create the best building system ever. OK, I have to stop now because I can go on for hours..."

▲ Apparently, we're not allowed to call the game 'hipster Minecraft', so we won't call it hipster Minecraft. Ahem.

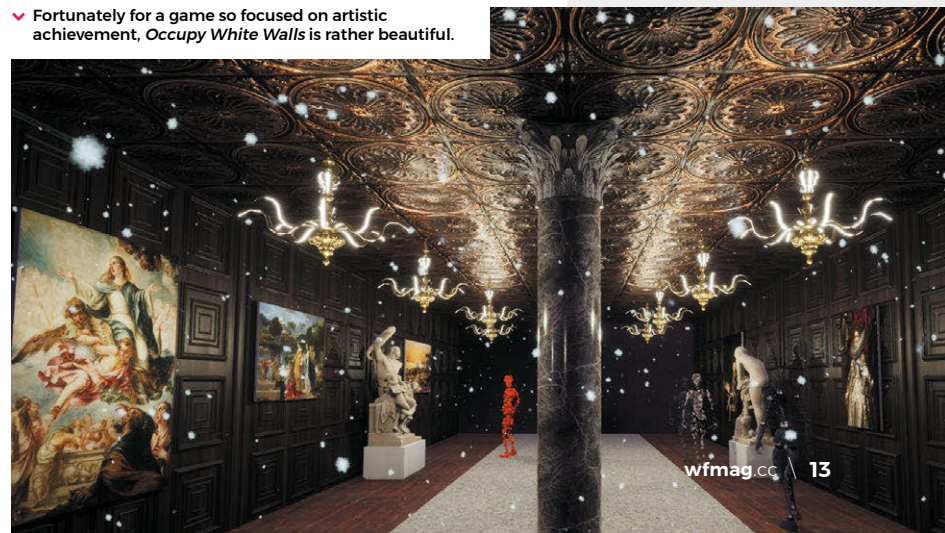
IS GAMES ART?

But is they? It is important question, must be ask. So we has: StikiPixels – is games art?

Spyropoulos: "Yes. I am going a step forward... are 'save games' art? I have seen amazing things in the game: exhibitions, installations, pixel art, text art, and beautiful architecture."

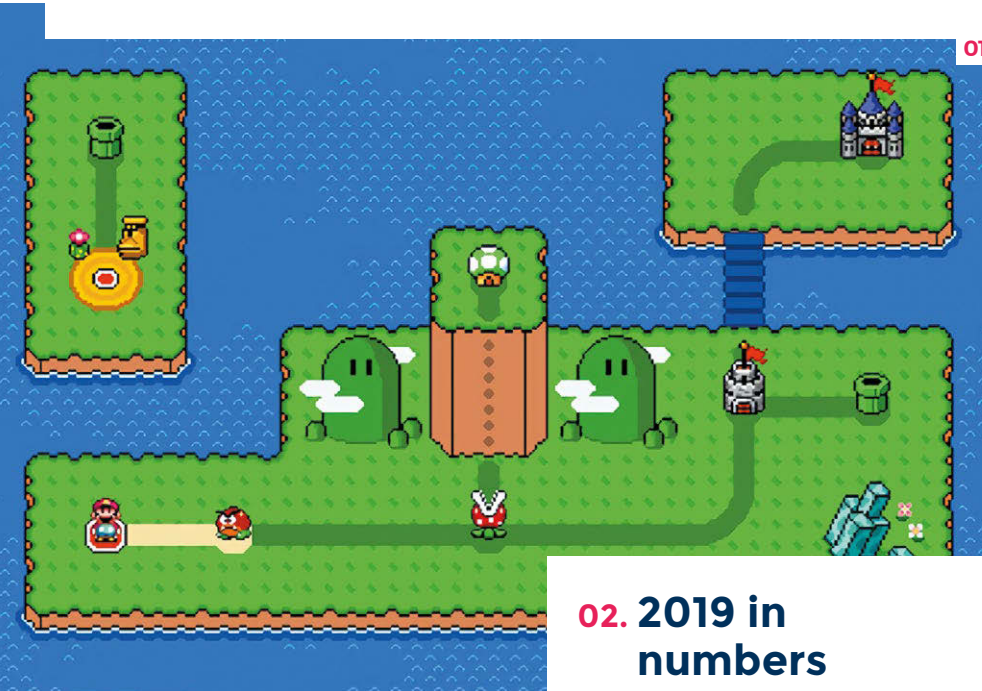
Yaroshevski: "Yes, if they are placed in a museum/gallery and labelled as such. That's what Marcel Duchamp would have said, anyway!"

▼ Fortunately for a game so focused on artistic achievement, *Occupy White Walls* is rather beautiful.

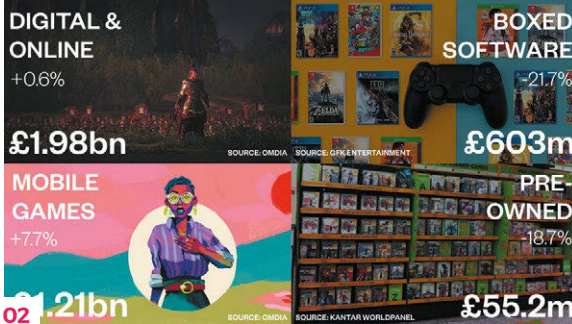


Headlines

from the virtual front



GAME SOFTWARE: £3.85bn



02. 2019 in numbers

The UK video game market saw a drop in value through 2019, down by 4.8% when compared to 2018. Priced up at £5.35bn, according to a new report from UKIE, it is still worth a pretty penny – but it seems there’s been an inevitable downturn attached to the upcoming new consoles. People just aren’t as keen when there’s something newer and shinier on the horizon, after all. Hardware sales dropped 14% to £1.35bn, somewhat inevitably, while physical media didn’t fare well – new game sales dropped 21.7% to £603m, and pre-owned fell 18.7% to £55.2m. But it wasn’t all down! Digital sales jumped up 0.6% to £1.98bn, while mobile game revenue leapt 7.7% to £1.2bn. This was 2019, though, and that was before the COVID-19 outbreak. Who knows what 2020’s figures will show us.

01. Overworlds

It’s not the world’s biggest news when a game gets an update, sure, but *Super Mario Maker 2* recently received its final patch – and if you haven’t seen what’s in it, you might want to either dust off your copy of the game, or pick up a copy online. It adds overworlds. *Overworlds*. Yes, those things we first saw back in *Super Mario Bros. 3* and probably made more famous through *Super Mario World* – you are now able to make your own, while making your own Mario game, in a game about making Mario games. By crikey, *Super Mario Maker 2* is a very good thing. Other power-ups are added in the update, along with those little irritants the Koopalings. It’s all good, but... overworlds!

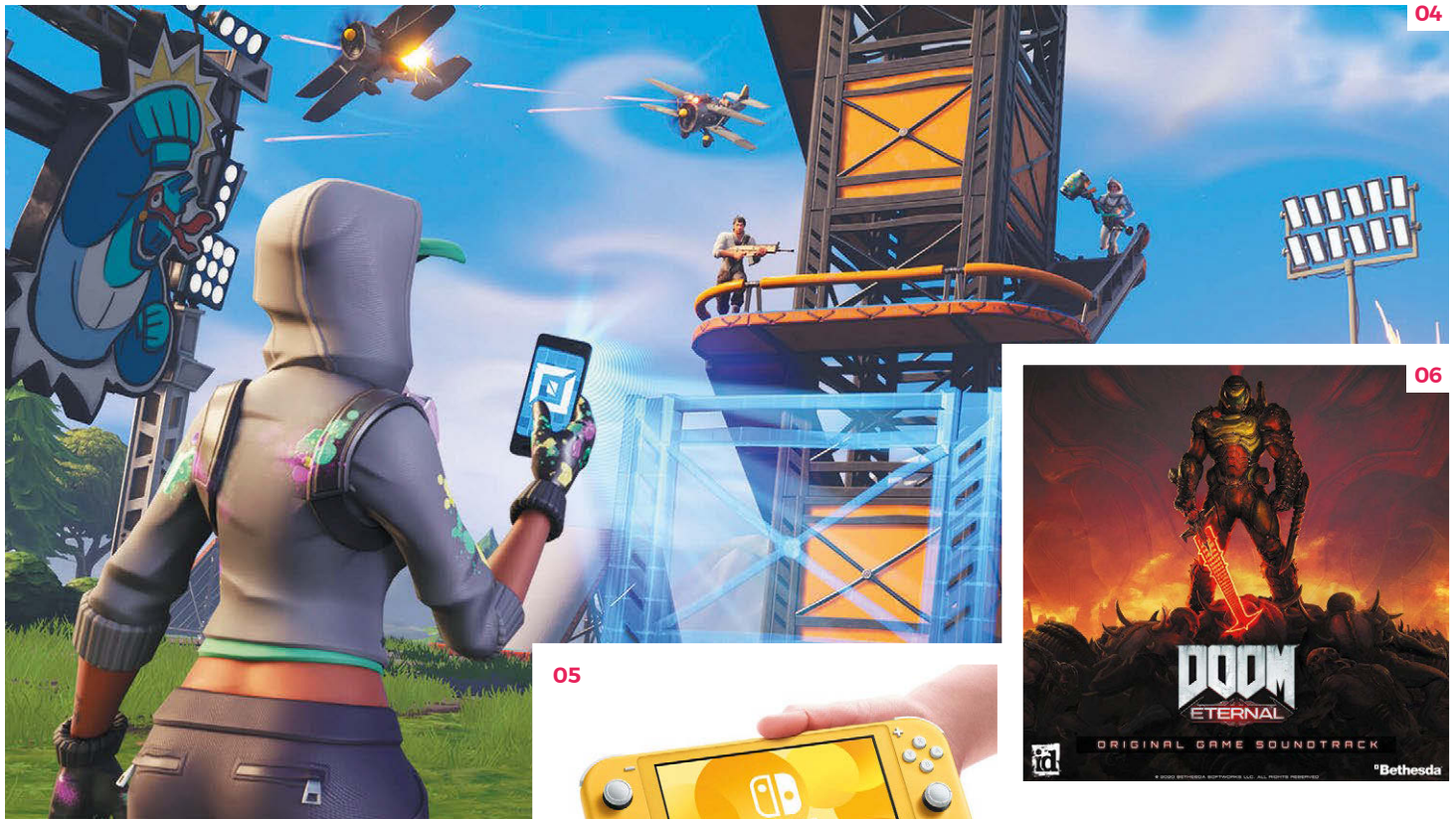
03. It’s an Atari casino

That’s about it, really: Atari is looking into launching a casino with the express intention of exploiting its long history of licenses for money-munching tie-ins. The aim is to use virtual, real, and even cryptocurrency in whatever machines and games come out of this, and if you’ll excuse me I need to go sit on the floor in the shower for a bit. Look, let’s just remind ourselves that Bally (linked to Midway Games) was part of the casino business, and both Bandai Namco and Konami are investing ever-more of their money and IP into the world of patrons losing money while their addictive personalities are taken advantage of. It’s not *just* Atari at fault here.



Death Stranding jacket, priced at £1500, sells out. It truly is the end times

Paul Haddad, original voice actor for *Resident Evil 2*'s Leon, passes away aged 56



04

06

04. Epic Fail redux (2?)

Epic has finally, after an 18-month struggle, given in: *Fortnite* is available directly through the Google Play store on Android devices. You may remember back in the mists of the past, the dev/publishing/engine giant decided it would circumvent Google's official store (and the high percentage cut the platform holder takes) and take advantage of Android's openness to having apps side-loaded, making *Fortnite* straddle the gap between proper, official *thing* and grey area naughty install. Turns out that approach isn't a great one, as Epic has now gone down the official route – though not without protest, citing how Google puts third-party installs at a 'disadvantage' and other related claims. Ah, drama.

05

05. Bad bots buying

The ongoing Nintendo Switch shortage is apparently being driven by an army of bots, at least according to Motherboard. The site revealed a community of, well, scalpers using open-source software to scan shopping sites for available stock of Ninty's console, snapping them up before many real humans can get in there to purchase. Bereft of one of gaming's best-ever machines, these poor folks turn to resellers and auction sites, where they end up paying through the nose for some light entertainment in troubling times. Thanks, bot-using resellers – you're definitely not scumbags!

06. Sounding off

Doom Eternal's soundtrack was made by, but not entirely mastered by, the guy who actually composed it, it turns out. An apparent reduction in dynamic range, as spotted by Ruff Audio founder Reace Niles when looking at the soundtrack's waveforms, prompted the check: was this the work of established audio magician/original composer Mick Gordon? Had something gone horribly wrong? Who could we *blame*? Gordon himself replied, pointing out he did not mix the tracks in question, and had actually only been involved in a few tracks on the finished soundtrack album. A purported message from Gordon via Instagram also seemed to show the composer did not think he would work with Bethesda on any future *Doom* projects. It's all bit... strange. Still, the game's good, and the music *does* sound fine, so... there's that.

Crysis remaster confirmed following Twitter account's rebirth, coming to PC, PS4, XBO, and Switch



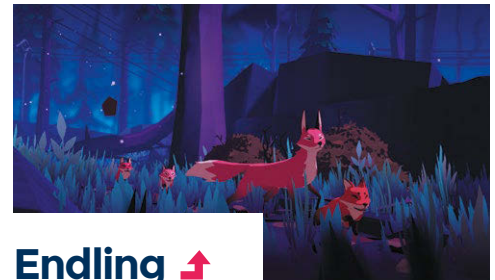
Hey look, it's an Xbox Series X logo. Fun!

Attract Mode



Penguin's Dogma ↑

We're not entirely sure whether this one will turn out to be worth playing, exactly (even its developers admit it's "silly" in its Steam listing), but *Penguin's Dogma* has such an engagingly bizarre premise that we had to give it at least a brief mention. It's a horror-action game where you run around some sort of industrial building, avoiding the attention of a giant, murderous cartoon penguin. According to developer MIYAKOPubl, "You can survive the penguin's wrath by locking it in the kitchen and setting it on fire." They further state that horror and humour are combined in "a precarious balance." How precarious are we talking, though? We look forward to finding out.

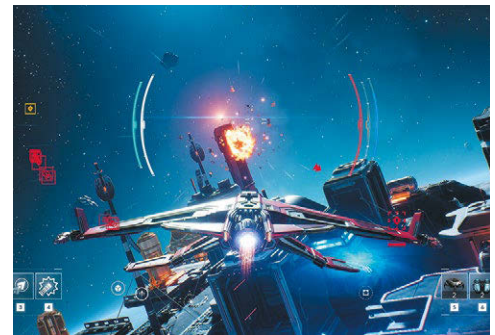


Endling ↑

Another survival game, this time with a more ecological undercurrent. Here, you play a fox battling to protect her litter of cubs in a world on the brink of collapse. You'll roam a side-scrolling landscape in search of food, and gradually make your way to a haven far away from those pesky humans. We genuinely hope nothing bad happens to any of these foxes in *Endling*. Our delicate souls couldn't take it.

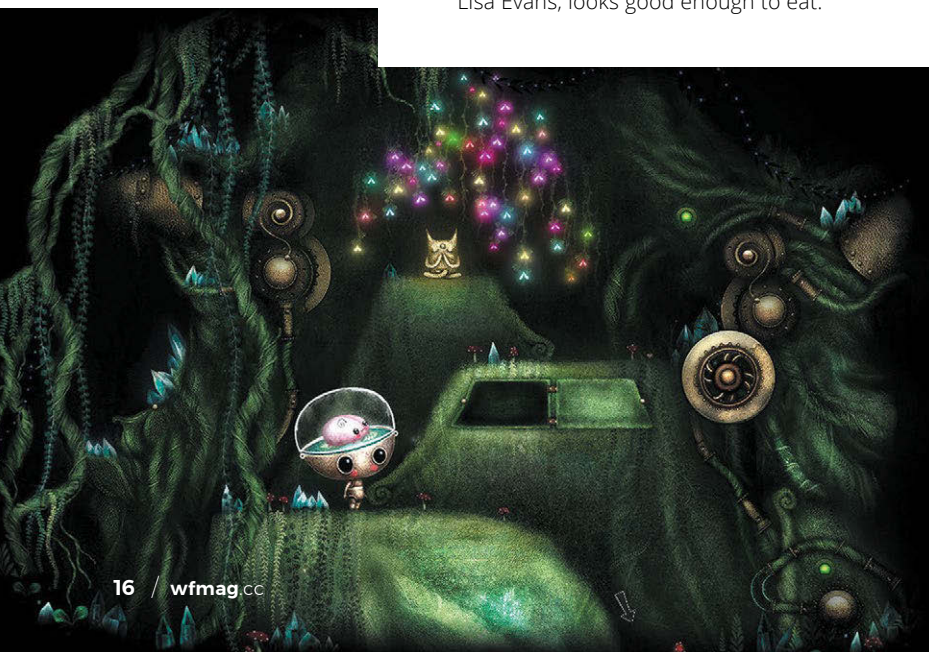
Growbot ↓

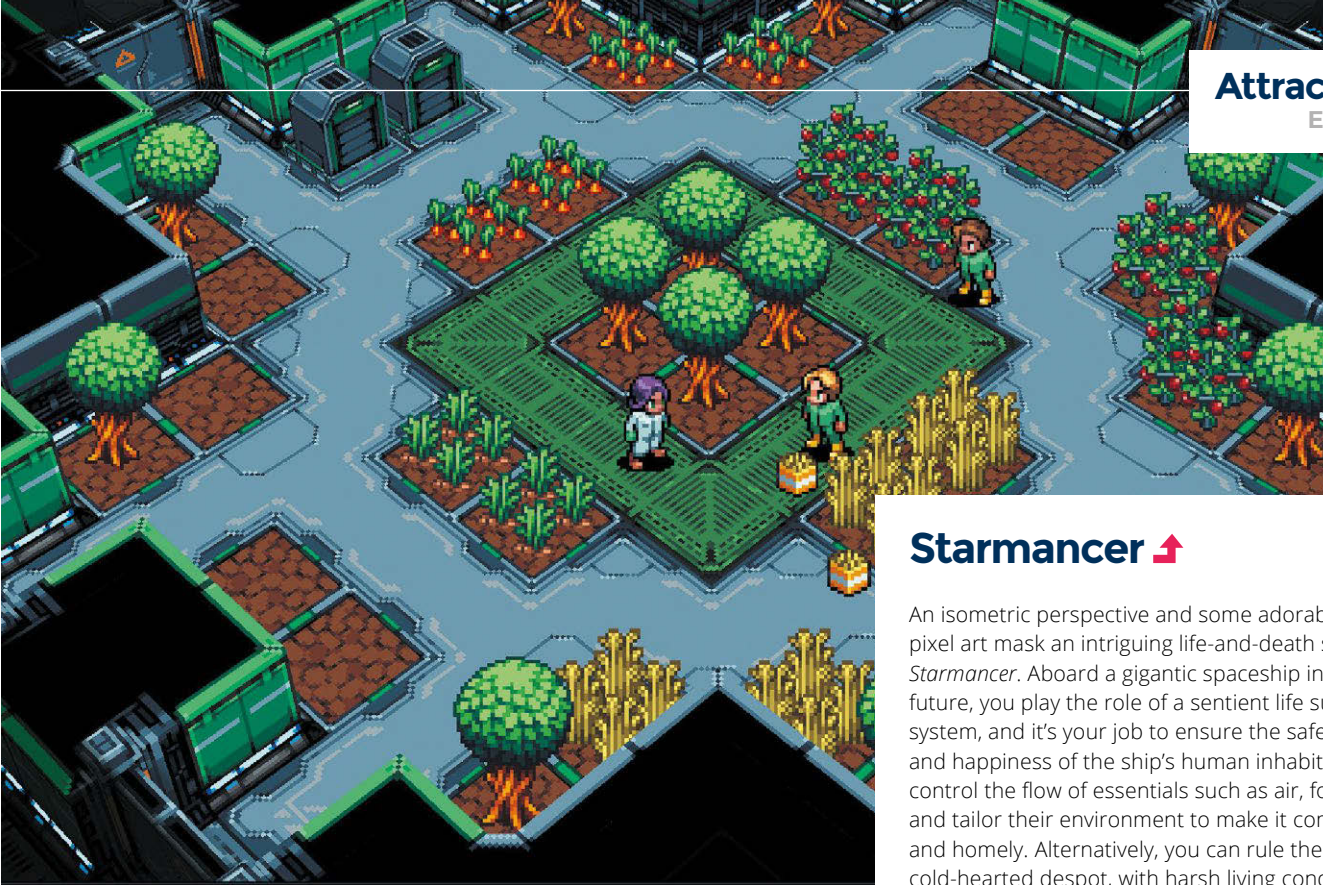
There's a touch of Amanita Design's eccentricity to this point-and-click adventure, and that's by no means a bad thing. The plot involves a robot attempting to save its home aboard what developer Wabisabi Play call a "biopunk space station", but the game's tone is more gentle and soothing than some of the other games covered here. The artwork, created by illustrator Lisa Evans, looks good enough to eat.



EVERSPACE 2 ↑

The original was a cracking space combat epic, so it's pleasing to see developer ROCKFISH Games return with a sequel. It won't be more of the same, either – the first title's roguelike trappings are out, and the exploration of a sprawling open galaxy is in. *EVERSPACE 2*'s Early Access launch was recently pushed back to December due to the ongoing coronavirus pandemic. We're sure it'll be worth the wait.





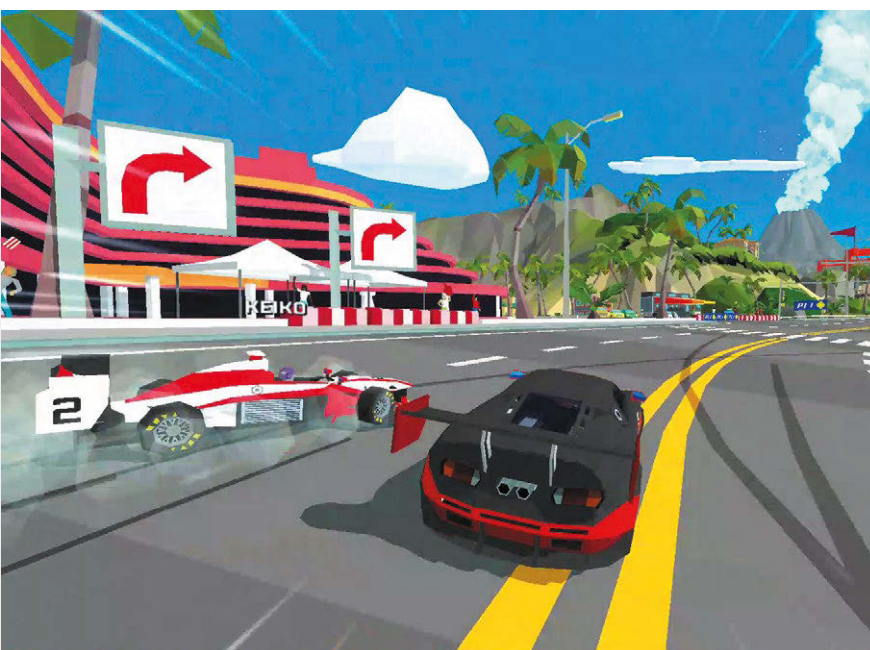
Starmancer ↗

An isometric perspective and some adorably chunky pixel art mask an intriguing life-and-death scenario in *Starmancer*. Aboard a gigantic spaceship in the distant future, you play the role of a sentient life support system, and it's your job to ensure the safety, health, and happiness of the ship's human inhabitants – you'll control the flow of essentials such as air, food, and water, and tailor their environment to make it comfortable and homely. Alternatively, you can rule the station like a cold-hearted despot, with harsh living conditions for one stratum of its society and plush, cosy spaces for another. Be careful, though: developer Ominux Games suggests the game's inhabitants are a clever bunch, and will plot to overthrow you if your approach is too harsh.

Outside the confines of the ship itself, there's a wider, randomly generated universe to explore and strip-mine, as well as the threat of pirates to contend with. There are some ambitious ideas bubbling away beneath the game's modest exterior, and we'll be fascinated to discover just how complex its simulation of a society in space will eventually become.

Hotshot Racing ↘

We're getting a distinct nineties vibe from *Hotshot Racing*, and not just because of the low-poly graphics. Its arcade-like drifts and outrageous turns of speed are in the vein of such games as *Daytona USA* and *Ridge Racer*; it's co-developed by Sumo Digital, which is no stranger to high-octane driving experiences like these. We'll be looking more closely at this one in a future issue.



Zhelter ↘

Hailing from South Korea, this top-down survival game has a well-thumbed concept – kill zombies, avoid zombies, try not to die – but the execution looks terrific. There are sprite designs that vaguely recall those in the *Metal Slug* series, and gameplay that nicely shifts between the relaxing (building, foraging, chopping down trees) and the borderline terrifying (firing shotguns into heaving tidal waves of the undead). *Zhelter* hits Early Access in late June, and we can't wait to try it.



20 corking

construction games

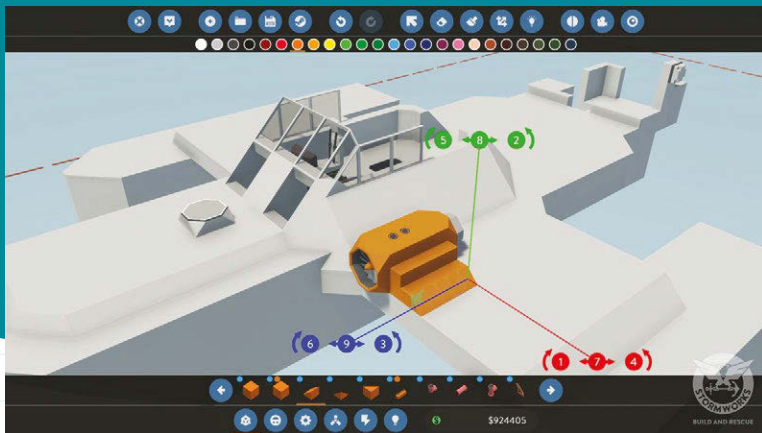
you can play *right now*

Want to scratch that creative itch? You may need some cream for that. And afterwards, try out a few of these great construction games

Stormworks: Build and Rescue

2018

From British developers Sunfire Software, *Stormworks: Build and Rescue* is that rarest of things in video gaming: a coastguard rescue simulation. What that breaks down to is a comprehensive and challenging marriage of block building and growing a company, along with conducting rescues and missions. The building side, in particular, gives the game a real distinction, getting you designing vehicles – often, er, quite unusual-looking ones – from minuscule blocks. It's a mechanic that's ignited a flourishing fan base, sharing their designs and work. A taxing, challenging game certainly, but an utterly engrossing one.



Kerbal Space Program

2011

And it all looks so easy in the movies. A game that was in public stages of development for four years before it officially launched, *Kerbal Space Program* charges you with setting up and running, well, a space programme. In this particular case on behalf of an alien race. Truth be told, they probably regret asking. It'd be fair to describe what happens next as both educational and just a little intense, a game in which success has to be earned. It's also absorbing, witty, and a huge time-gobbler, made all the more so by the active community of mods, and the emergence of a full sequel due next year. We can't bloomin' wait.



Dreams

2020

If ever there's a purchase worth digging out the old 'you get out of it what you put in' cliché for, surely *Dreams* is it. From *LittleBigPlanet* creators Media Molecule, it's not unreasonably described as a game universe, giving PlayStation 4 gamers an extraordinary collection of tools to fashion their own creations. The scope is far beyond making games (music, art and, well, stuff are well within its purview too), although there is a short, strong, and involving story mode at the heart of it, that itself travels through many genres. The 'Made In Dreams' badge is likely to crop up a lot over the coming years.



Banjo-Kazooie: Nuts & Bolts

2008

Ten years after Rare debuted *Banjo-Kazooie* as a Nintendo exclusive, it made the jump to the Xbox 360. It wasn't just a change of hardware for the series either, as the game itself took a little bit of a sidestep. The emphasis wasn't entirely on platforming, but also constructing your own vehicles to get you through the game. A generous collection of options there were too, with over 1000 components to toy with, married up to an interface that impressively never got in the way of you mashing things together. End result? An ambitious sequel, that got a slightly unfair mixed response on release.



LittleBigPlanet

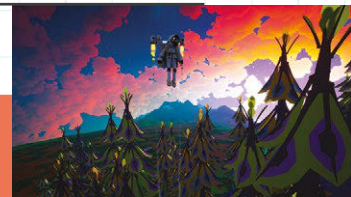
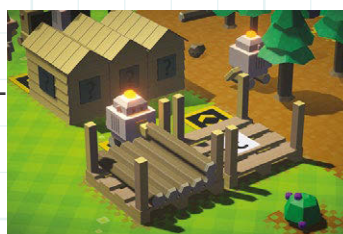
2008

On its release, the first *LittleBigPlanet* felt like something really quite different for the PlayStation 3. Not just because it had the beating heart of a traditional platform game, but also the customisable options were so dramatically different. The game – a delight to play, even without the tinkering – was sold heavily off the back of being able to create levels, change the central character, and basically play around with every facet of the game as provided. It was the debut game from Media Molecule, itself formed by former employees of Peter Molyneux's Lionhead Studios. Sequels have followed, and are just as delightful.

Autonauts

2019

A charming game that gives you a spaceship, worlds to explore, and planets to colonise. Under the surface, *Autonauts* is also a very well done introduction to visual programming techniques. Above the surface, you create items for crafting and then use bots to automate the process, making sure you get the instructions just right. As you colonise more and more worlds, your bots need more and more instructions, and the complexity increases. It's a deceptively huge game this, presented in an accessible, bright, and cartoony visual style. It aims towards the younger end of the market, but its charming simplicity is cover for a time-gobbling game.



Astroneer

2019

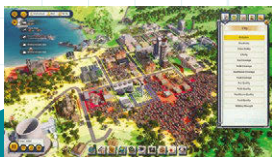
Set in the 25th century and again with a core of space exploration at its heart, *Astroneer* takes you to different planets, and encourages you – either alone or in multi-player – to pull together the bits and bobs for bases, gathering the required resources and surviving in increasingly tricky terrains (that bring with them differing, escalating demands). In spite of limited oxygen and the need to explore the unknown to survive, this is actually quite a relaxed game, without a narrative bedrock to fall back on. Instead, it's a slow burn, beautiful to look at, with a bit of a learning curve.



House Flipper

2018

Fine practice for being an adult, this *House Flipper* tasks you with buying a home, stripping out the junk, cleaning it up, planning and restructuring the interiors, and then selling the place (and dealing with demanding buyers) so that you can do it all again. Nice and relaxing after a hard day at work, right? And yet, it really is. Lots of small little jobs, a real sense of achievement, and the chance to smash up some plasterboard without having to explain yourself afterwards. The *Garden Flipper* DLC is well worth the extra investment, too.



Tropico 6

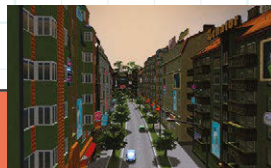
2019

It's impossible to imagine a starkly authoritarian leader emerging in a First World nation, so we need games to fill the gap. Behind the *Tropico* series' 'knockabout veneer' lies a complex marriage of government, construction, and hard strategy, as the player is asked to be 'El Presidente', the head of a fictional Caribbean island. The satirical edge gives it a slightly different flavour to its rivals, and this sixth instalment doesn't reinvent much, but cherry-picks the many highlights of the series to date.

Factorio

2016

The exhausting logistics of designing and running factories is made admirably approachable in *Factorio*. As with most games of this ilk, you spend a decent amount of time accruing resources, before you then put your production facilities together and gradually escalate your operation. There's an underlying warning of the perils of capitalism here, too, as you balance the pollution of your factories against the protests of the locals. Pure escapism.



Cities: Skylines

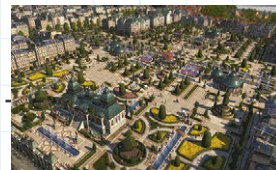
2015

The dominance of the *SimCity* series meant there was once some hesitancy to give what became *Cities: Skylines* the green light. The turning point was Maxis and EA's infamously disliked 2013 *SimCity* entry; afterwards, Paradox Interactive rolled the dice on *Cities: Skylines*. The result is the clear market leader in city planning strategy titles, with a dozen expansion packs and counting to date. You can read more about Ian's metropolis-building antics on page 63.

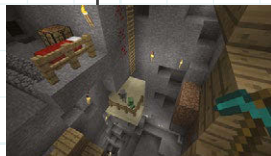
Anno 1800

2019

Regarded as a return to form for the hardly struggling *Anno* series, this entry attempted to answer hardened fans' complaints that recent games had reduced the complexity somewhat. *Anno 1800* took the series back to its historical roots, with an industrial revolution setting, and as always you have to build things up from scratch. You juggle production infrastructure, resources, and expansion, and things really open up as you build ships and go exploring. It's a 'book two weeks off' game.



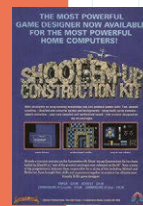
The obligatory Minecraft entry: Minecraft 2009



The game that's never going to stop being built. Rather than make a sequel, its developers keep expanding on *Minecraft*'s existing foundations. A decade on, and millions of people are still crafting and building on a daily basis. *Minecraft* is the most influential and popular building game of its generation – a genuine phenomenon whose tools have even been used to plan out the action of a feature film (and its own movie is on the way, too). It'll outlive us all.

Shoot-'Em-Up Construction Kit 1987

Commonly known as SEUCK, here was an early attempt to give non-programmers the tools to make their own arcade games. In truth, it was all a bit limited, and most of the resulting games that flooded the Amiga and Atari ST public domain were samey takes on the standard 2D shooter. But as a stepping stone into game design, SEUCK deserves its place in history alongside such contemporary packages as the Professional Adventure Writer and Graphic Adventure Creator.



Dragon Quest Builders 2 2018

This sandbox spin-off hides some impressive strategic and RPG mechanics beneath its cartoony visuals. In *Dragon Quest Builders 2*, the brilliance lies in the details: you can become obsessed by the tiniest of things as you put rooms together, and likely will. But then alongside that is a detailed RPG adventure to follow. It's a hybrid that comes together impressively, with plenty of fan service too for devotees of earlier *Dragon Quest* titles.

HONOURABLE MENTIONS

Dezaemon (1991)

A series that let console gamers create their own shoot-'em-ups.

3D Construction Kit (1991)

A tool for 8- and 16-bit machines that brought the pioneering Freespace engine to gamers.



Super Mario Maker 2

2019

Super Mario Maker and its sequel have made plain just how brilliant the level designs are in the main *Super Mario* games. As Nintendo's peerless construction kits for the Wii U and Switch have proved, it's relatively easy to design a difficult *Super Mario* level; crafting a stage that's challenging, clever, and fun is an entirely different proposition. Nevertheless, we've seen some spectacularly imaginative designs emerge from the *Maker* community over the years; the constant addition of new elements (such as the recently added overworlds) has only fanned those creative flames.



Planet Coaster

2016

Taking huge influence from *Theme Park* and *RollerCoaster Tycoon*, the hugely successful *Planet Coaster* is a detailed, scientific, and compelling theme park construction mix of strategy and simulation. Put together by Cambridge-based Frontier Developments, it doesn't stray too far from the idea behind its forerunners, but the number of options available and the amount of creativity it and its add-on packs afford is gigantic. One search of YouTube offers ample evidence of that. *Jurassic World Evolution* and the more recent *Planet Zoo* have subsequently been developed and released too, both in a similar vein.



RimWorld

2018

With its catalogue of influences including classic science fiction such as *Dune* and *Firefly*, *RimWorld* kicks off by leaving three people marooned on a far-off world. The challenge is to build from there. Two factors give this distinction. Firstly, the pared-down graphical style of the game (you won't need a beast of a machine to run it as a consequence). Secondly, an AI story generator that adds a relationship element to the action, and some degree of unpredictability to it. It doesn't always quite hang together, but it's ambitious, interesting, and wears its quirky heart and love of sci-fi very much on its metaphorical sleeve.

Two Point Hospital

2018

If *Planet Coaster* builds on the influences of Bullfrog's *Theme Park*, then *Two Point Hospital* is the obvious heir apparent to the same company's joyful *Theme Hospital*. Appreciating medical games may not be most people's idea of escapism in these taxing times, the game challenges you to build up a hospital, while researching a cure for the assortment of baffling and played-for-comedy extreme diseases. It certainly captures the spirit of its classic ancestor – and indeed veers very closely to it – unsurprising given that the some of the creators of *Theme Hospital* are behind the new game.



Train Valley

2015

A family-centric successor to the likes of *Railroad Tycoon* and *Transport Tycoon*, *Train Valley* and its sequel follow the same idea of getting you to put together a locomotive infrastructure, starting in the 19th century and heading off into the future. It's the cartoony approach that makes this instantly accessible, as well as the welcome addition of shorter, bite-size game modes to tackle. A strategy game that, for good reasons, doesn't threaten to eat up your evenings all in one go. There are games that do train management a lot deeper, but this is a nice way into the genre.

TerraTech (2018)

Build vehicles, find parts, and explore planets in this acclaimed title.

Block'hood (2017)

SimCity zoomed into neighbourhood level, where getting a network working together is key.

Prison Architect (2012)

Keep the lags in order as you build and run your own private prison.



BLOOD & BORE

The forgotten Mortal Kombat
klones of the 1990s

WRITTEN BY
STEVEN T. WRIGHT

Some of us might complain about the constant deluge of mediocre imitations of great games in today's market. But as these interviews reveal, it's far from a new trend

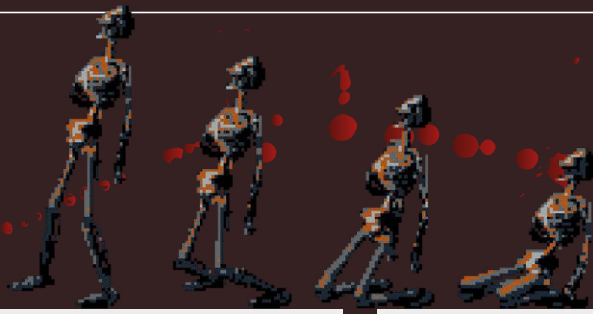
One of the enduring clichés of the video game industry is the compound adjective 'hit-driven'. Usually used to describe itself, it's

the idea that most commercial games follow an inextricable pattern: a hit game suddenly appears in the mind of the Genius Designer, emerges fully-formed from the sheer power of his brilliance – for it's always a man – and the money comes rolling in, hand over fist. As with most clichés, there's at least some dusting of truth on this age-old blueprint. Even today, in an industry that's larger and more diverse than ever before, we rarely read about modest successes – no, we want to read about the smash hits, the record-breakers. But over time, as games have become more complex, the titles that

truly capture our collective imagination are not the originators of a new idea, but the ones that perfect it – or, alternatively, in today's increasingly uncertain digital economy, concoct the right monetisation scheme. After all, *PLAYERUNKNOWN'S BATTLEGROUNDS* might've sold millions of copies, but it's the free, ever-evolving *Fortnite* that every live game is tripping over themselves to emulate.

In gaming's distant past, the rush to capitalise on a growing trend was even shorter and bumpier than today, but the stories rarely get told, especially if the process failed to produce another blockbuster hit. Consider, then, the crimson flood of violent fighting games that followed the overwhelming arcade success of the original *Mortal Kombat* in 1992. From the bizarre (and never-





released) *Tattoo Assassins* to Midway's own subpar 3D fighter *War Gods*, the next five or so years were marked by bloody B and C-tier fighters that hoped to cash in on the pile of gold that had leaked out of *Sub-Zero's* famous 'spine rip'.

These days, with the advent of lengthy Early Access periods and open betas that precede an official release, it's not uncommon to see what we might term 'clones' emerge on the market at the same time as their inspiration. (For example, Blizzard's watershed hero-shooter *Overwatch* came out in May 2016, and the similar free-to-play *Paladins* followed just a few months later.) For *Mortal Kombat*, the first imitator came within a month, with Incredible Technologies' weapon-based fighter *Time Killers*. According to David Thiel, the sound engineer who worked on the game, Incredible intended *Time Killers* as their first bold venture into the video game market, and the pressure on the team was very high.

MIXED INFLUENCES

"I think a lot of people think of *Time Killers* as a *Mortal Kombat*-kind of thing, which is definitely fair," Thiel says. "But I think it's important to remember that in those days, trends moved really fast. [*Mortal Kombat*] didn't exist when we started development, we were looking at all the money that Capcom was making with *Street Fighter 2* and we thought, 'Let's get some of that.' So we started putting something together, and I think the guys at Midway thought the same thing... I think what people sometimes forget [now] is that those were the days of the arcade, where you had all of two seconds for someone to look at your game and say, 'Oh, this is its thing.' You needed a hook. We thought blood was our hook, and clearly, others did too."

Today, Thiel is best known as one of the devs behind the arcade classic *Q*bert*, but he describes the year-long development of *Time Killers* as one of the highlights of his career in games. While the studio had notched several respectable successes

in the arcade space in the late eighties with sports fare like *Capcom Bowling* and *Winter Games*, Thiel describes *Time Killers* as their first attempt to compete with larger players in the space. Unlike *Mortal Kombat*, where the ultra-violent finishing moves could only take place at the end of a match, in *Time Killers* the fighters could chop off an opponent's limb at any time,

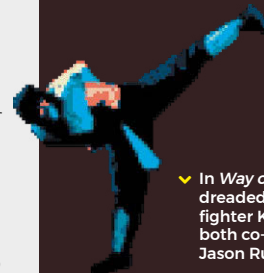
"For us, we always thought it was very silly, like a cartoon"

or even behead them with a lucky shot for an instant win.

"When some people saw [the violence], I guess they thought it was upsetting," Thiel says. "For us, we always thought it was very silly, like a cartoon. You can chop off somebody's arm and leg, and they'll keep coming at you. The inspiration for that was the scene in *Monty Python and the Holy Grail*, with the Black Knight that says, 'I can fight!' even though he doesn't have any limbs left. We wanted it to be funny like that." →



▲ The claymation magic used to make the hulking Goro a reality on-screen was a major influence on *Way of the Warrior*.



▼ In *Way of the Warrior*, dreaded Ninja (and fighter Konotori) are both co-developer Jason Rubin in a mask.



Other Clones

The nineties saw a far greater wave of *Kombat*-likes than we could possibly cover here. *Bio F.R.E.A.K.S.*, planned for arcades but made for home systems like the N64, cheerfully cribbed *MK's* deadly finishing moves. Others, like *Primal Rage* and *ClayFighter*, used similar digitising techniques to bring their characters to life. And then there was *Thrill Kill*, whose attempts to recreate *Mortal Kombat's* controversy (kontroversy?) spectacularly backfired: its violence and sheer kinkiness provoked such horror that the game was never officially released.





Killer tunes

David Thiel says he's particularly proud of a certain conceit he was able to program into *Time Killers*: each fighter has their own musical theme, which plays in their stage. However, when they take the lead in a match, the background music will change to that of the healthier player, and back and forth, as the match progresses. Each time it switches, however, the music will get slightly faster. "It took a lot of memory – we only had 64 kilobytes in those days – but I think it was a cool effect," he says.



▲ The violent final blows in *Time Killers* were considered extremely graphic in their day.



Crunch Bandicoot

While talking about the subject of crunch, Naughty Dog's Andy Gavin revealed a somewhat troubling fact: "With *Way of the Warrior*, the crunch wasn't so bad, to be honest... In the *Crash Bandicoot* era, from January 1997 to September 1997, I was in the office every single day, and I didn't go into a store during that period. I went into mini-marts and gas stations when I was driving, but I didn't step foot in a grocery store. That's just how it was then."

A CHANGING PARADIGM

For his part, Thiel emphasises that the game came out at a transitional era for coin-op companies like Incredible. While the arcade industry continued to burn for a few more years, once the likes of *Mortal Kombat* and *Street Fighter 2* launched on the consoles of the day, publishers began to realise that there was a lot more money in selling millions of copies of the home version on the Super Nintendo and Sega Mega Drive versus tens of thousands of arcade machines; as such, the resources shifted accordingly. Though Thiel admits that *Time Killers* came in a distant third compared to the hit fighting games of the day, he pushes back on its reputation as a failed also-ran, which he blames on the disastrous Mega Drive port that didn't emerge until the mid-nineties, which garnered embarrassing scores from the era's magazines.

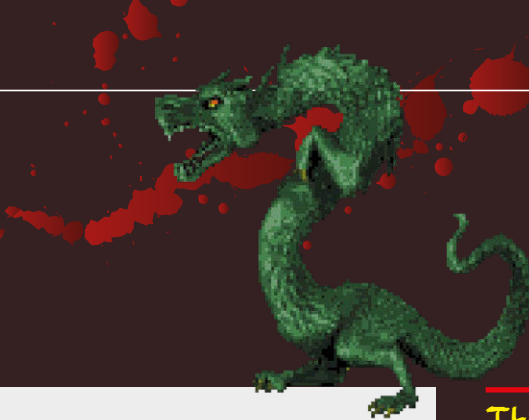
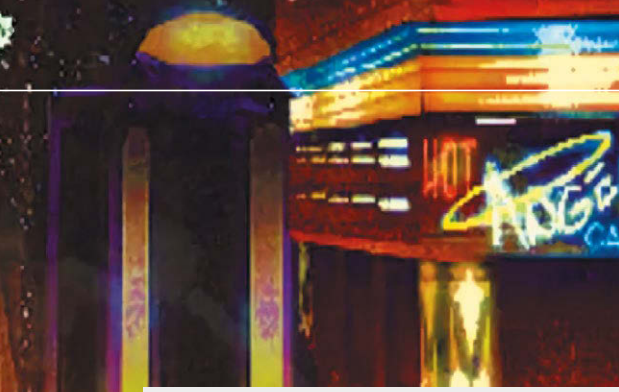
"It made us quite a bit of money," he says. "We were definitely far behind the big guys, but the pie was so big that even that tiny slice was enough to make a big profit. As [for] the Mega Drive port, I wasn't involved, but I knew it was going to be a hell of a job. The soundboard in the arcade cabinet had more processing power than the Mega Drive itself. I'd actually never seen the port until recently. I was doing some research on YouTube, and I saw footage of it – it was just brutal. It was like

they took my baby from me and just did horrible things to it. I couldn't watch much of it, to be honest."

LATE IN THE GAME

Getting in early on the blood and gore bandwagon might've made Incredible Technologies a tidy sum, but the *Mortal Kombat* clones continued well into the mid-nineties, from unlikely corners of the industry. Just ask Andy Gavin, co-founder of the much-acclaimed Naughty Dog studio. While you're probably familiar with many of their creations, including *Crash Bandicoot*, the game that first put the studio on the map, you most likely haven't heard of the one that preceded their breakthrough, a very strange fighting game that they called *Way of the Warrior*. Released in 1994 for the ill-fated 3DO – one of several systems that valiantly attempted to bridge the gap between nascent 3D tech and the stalwart 2D of the era – *Way of the Warrior* is perhaps the most brazen rip-off of *Mortal Kombat* ever made, complete with regrettable ethnic stereotypes, multiple claymation characters, and loads and loads of fatalities. When asked why Naughty Dog decided to experiment with a kung-fu fighter, Gavin just laughs.

"Our first games were radically original, I'd say," he says. "[Isometric RPG] *Rings of Power* definitely had some antecedents in *Ultima IV-VI*, but its interface was very unique, and it has a design vocabulary that was very unusual. However, because EA convinced us to use this weird save-game



hardware instead of the battery backup, a lot of retailers wouldn't take the game, so we ended up selling a fraction of what we would've – maybe 400,000. *Way of the Warrior* was the first game where we looked at the market, looked at what was popular, and tried to make that kind of game. We knew that we would make that kind of game for cheap, so even though the game ended up selling only around 100,000 copies, it still made money for [publisher] Universal, which is a big part of what enabled us to make *Crash*. It was a transitional game."

KEEPING IT CHEAP

To keep costs down, Gavin and co-founder Jason Rubin worked strictly out of what Gavin calls their "dude apartment", imitating the technique that *Mortal Kombat* famously used to digitise and capture its character models from real-life stand-


"It was like they took my baby from me and just did horrible things to it"


ins. However, since the duo didn't have the budget to rent a studio space, or to pay their actors, they ended up inviting their friends, acquaintances, and even their girlfriends to clumsily imitate kung-fu moves in front of a curtain in said apartment. (For example, the stick-wielding Australian Shaky Jake is played by Gavin's own brother.) They didn't even have room in the tiny apartment to film, however, so they had to open the front door and hang the curtain in the hallway, which drew strange looks from those who also lived in the building.

Their creative process was ramshackle, but organic: when they had a new idea for a character, they simply put Rubin in a ninja mask, so he wouldn't be recognised as another fighter in the cast. They added unconventional mechanics to the basic fighting system, including 'magic spells' that allow savvy players to heal mid-fight, as

well as a secret, unauthorised copy of the pioneering space combat game *Spacewar!*, hidden behind a code. According to Gavin, the two of them voluntarily worked brutal hours that were typical for the era.

"We would get up at four in the afternoon after having gone to bed at dawn, work on *Way of the Warrior* until dawn again, get takeout, or go to a Costco-type place to load up on candy, then do it all over again," Gavin says, laughing. "In many ways, that era of Naughty Dog, from five employees to 40, was our apartment on steroids. It was just, 'Live and breathe video games, never do anything else.' I wouldn't do it the same now, but it worked for us, and I think if you go back and ask anybody who worked during that era, they would say they had a good time doing it... We hardly fired anyone. People self-selected, they quit if it wasn't for them. By virtue of the people we knew, we were in our early-to-mid twenties, this similar age bracket, so it worked. I recently did some consulting for a small game, it's a little more PC these days, and they have more best-practices. But the basic feel of 'game people' is still there. Sitting in a room, getting it done."

All-in-all, it's clear that most aspects of games production have changed dramatically over the years, even when it comes to cranking out what some might call 'rip-offs' of bigger games, most of which are mostly forgotten as time passes. But even as the market has shifted, there'll always be room for imitators. And who knows? Every now and then, they might strike gold themselves. 

All-in-all, it's clear that most aspects of games production have changed dramatically over the years, even when it comes to cranking out what some might call 'rip-offs' of bigger games, most of which are mostly forgotten as time passes. But even as the market has shifted, there'll always be room for imitators. And who knows? Every now and then, they might strike gold themselves. 

This is the Way

While Andy Gavin fondly recalls the development of *Way of the Warrior*, it was also marred by tragedy. His brother Mitch's best friend, Tae Min Kim, volunteered to be put in the game as the shirtless Liu Kang-esque Dragon. Shortly after the game's final release, Kim was killed in an automobile accident in late 1995. "Somebody just ran a red light and hit his bike," Gavin says. "My brother was very sad about it at the time."



▲ *Way of the Warrior's* claymation characters were impressive in their day, but they've aged quite poorly.

▼ If a competitor lost both their arms, a *Time Killers* match would end early.



Is it time?



STEVE MCNEIL

Steve likes playing games, but also has a child. How can that possibly work?

“I always knew in my heart I was essentially an adult playing a kid’s game, but now I’m meant to be, because I have a kid”

One of the trickiest parts about being a parent that is also a gamer is how and when to introduce your child to your hobby.

I’m lucky enough to have a Picade (a tabletop arcade cabinet with a Raspberry Pi in it – yes, you’re right, that does sound awesome, you should get one), so I’ve been tentatively dazzling my daughter with retro hedgehogs and plumbers almost since birth but, while she had an immediate interest in the flashing colours and shapes, she never really actively played or stuck with it for more than a few minutes at a time.

After some time, she started to recognise characters (her ability to recognise more games than my wife is a source of great, eternal pride to me) but, until recently, I’d never really pushed to get her fully engaged in playing.

And then lockdown happened. And, regardless of your parenting style, when in lockdown, the correct answer to when you should introduce your child to video games, regardless of your child’s age, is *immediately*. Have you seen how long days are, when they’ve just got crayons, Play-Doh, and Igglepiggle in them? It’s awful.*

My favourite game of all time is, undoubtedly, *Minecraft*, and thankfully my daughter now loves nothing more than telling me what to do as we build a humble farmstead in the game.

For my part, I love to subtly coerce her into asking for things I want to do in the game, so that I can play it for ages under the guise of parenting and nurturing her spatial awareness or whatever.

Animal Crossing, too, has been a revelation. Sure, I always knew in my heart I was essentially an adult playing a kid’s game, but now I’m meant to be, because I have a kid. Thank you, sperm!

The only problem is, as a Twitch streamer who has been broadcasting the game, I have to manage her usage of the game to prevent her

from affecting the stream. What this essentially boils down to is having to sneakily get on the game in the morning before she does, so I can grab all the valuable fruit and fossils, in order to progress my own game at the expense of hers, while simultaneously convincing her that Nook Miles exist solely to purchase endless street lamps, because I’ve decided our island’s paths would look nicer with them.

So, while on the one hand, I’m letting her enjoy video games on a daily basis, I’m also making sure that ultimately she’s enjoying them in the way that most benefits me. At this point, I honestly don’t know if I’m the best dad ever or the worst one. 🙄

* *To be fair, I love Play-Doh, and I don’t mind the Pontipines either, the little scamps.*



▲ Steve McNeil’s Picade. Stuffed full of retro goodness.

Toolbox

The art, theory, and production of video games

28. CityCraft

Adding texture and fine detail to your video game town or city

30. Karting in Unity

Hone your development skills and beat your lap times in our arcade racer

36. Narrative Design

Examining Firewatch's assured storytelling, and how it introduces its lead character

38. Kingmaking

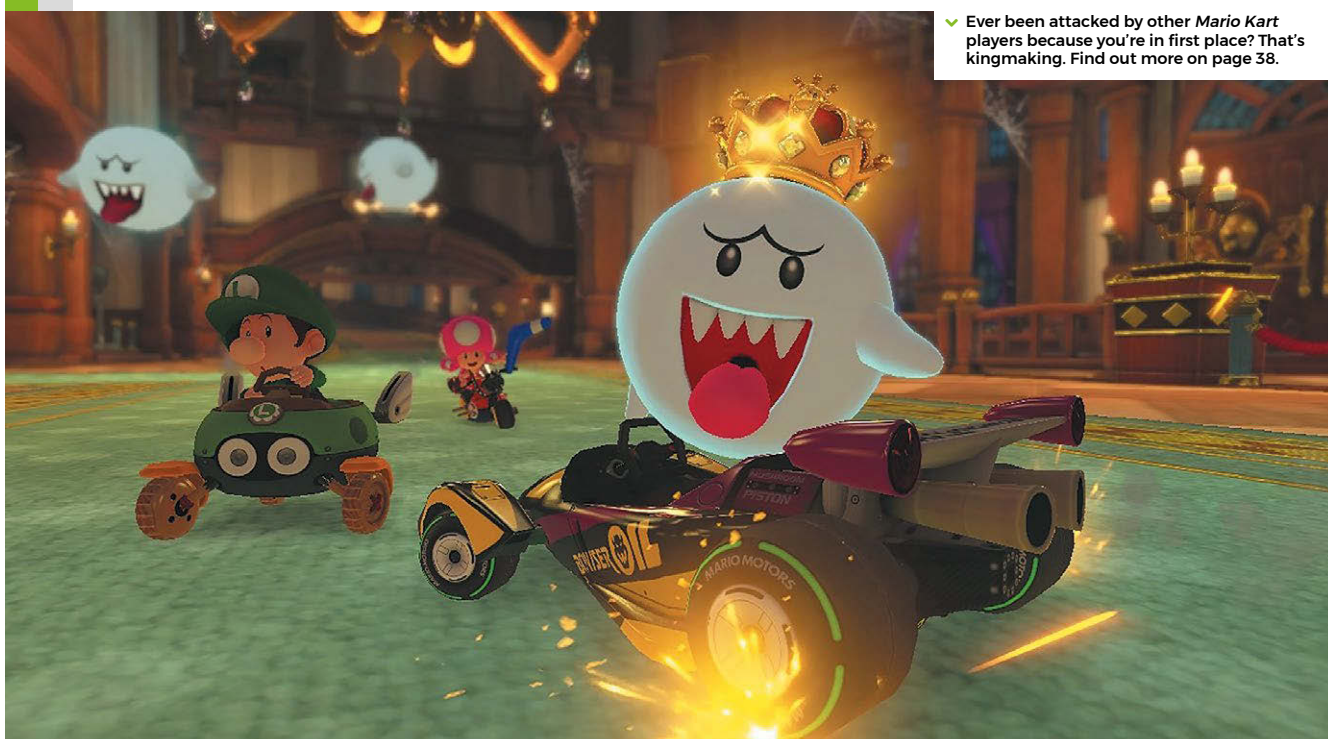
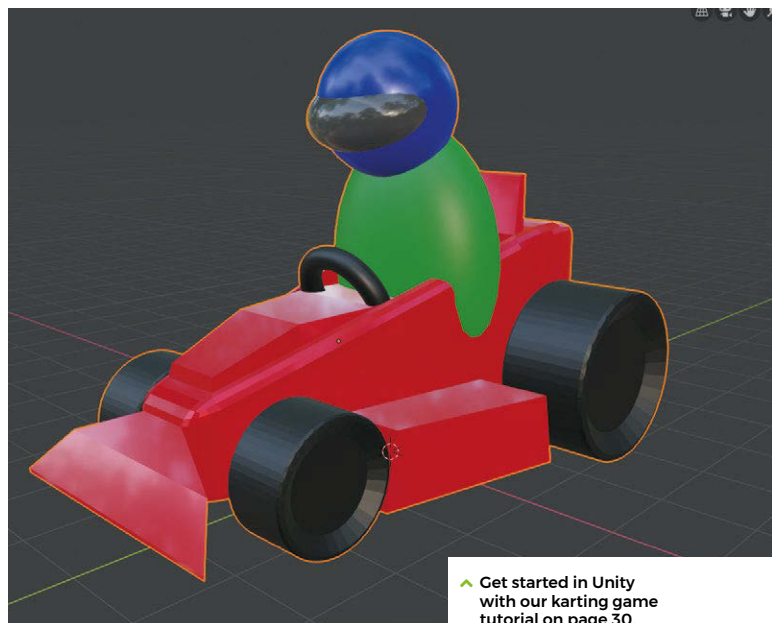
Why players do it – and why it isn't cheating

40. Source Code

Code Robotron: 2084's twin-stick shooting

42. Directory

Charities that could use your help during the current lockdown



Texturing a virtual city

How applying layers of detail, and carefully selecting characteristic touches, can help a virtual city pop out



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

W

hen planners and architects discuss urban textures, they tend to focus on specific, rigidly defined metrics of the built environment. When

world-builders and game designers approach the texture of an imaginary place, however, they aren't usually looking to describe such spatial structures, patterns, and arrangements – or at least I'm not. Instead, I look to express how a place is perceived by all our senses, how it can tickle our brain, and make us feel. Texture, in this particular context, goes beyond measurable cityscape characteristics, topologies, interrelations of land use, and street layout.

Virtual urban texture is all about the ways in which non-physical places feel tangible and real – the ways they can indirectly trigger our senses. We want to craft unique cities, able to conjure up smells, temperature, and even vibrations. To put it another way, adding texture means making game cities seem as populated with interesting details as the average adventure game scene; cities ready to be 'examined' or 'looked at', and capable of providing all sorts of information and trivia.

PALPABLE LAYERS

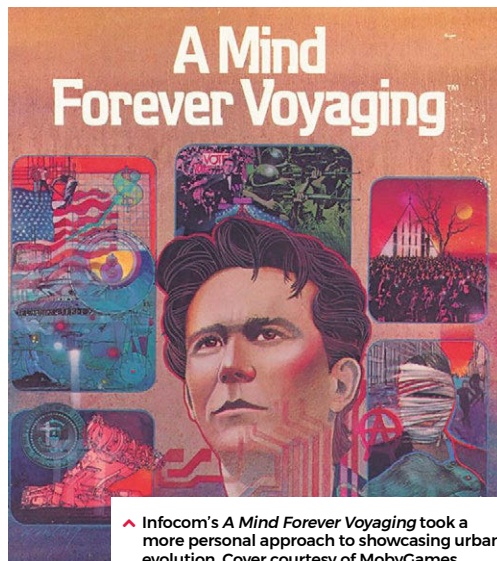
A textured virtual city is one filled with elements and scenes that make us take pause and look. Its spaces are meaningful and often interactive, and its vistas are spectacular. Precisely defining this type of texture, and prescribing the exact methods towards achieving it, is essentially impossible since it combines countless things. The way the gravel on the road sounds underfoot, local accents, music coming from basement taverns, and smoke rising from dozens of food stalls are only part of the wider tapestry that constitutes a place's texture.

One way of approaching texture, then, is to make certain that all five human senses are catered for – the recipes in *The Legend of Zelda: Breath of the Wild* spring to mind – and that urban environments are carefully designed to make sense, be memorable, and function as living, breathing, social, and evolving entities.

Think of how every element of a city sounds, looks, feels, smells, and – when applicable – tastes. Consider the acoustic and reflective qualities of construction materials. Reference familiar tastes and smells such as coffee or hot bread to help

Points Of View

The perception of a city is always personal and thus partially subjective, and that's exactly why experiencing a city's space and evolution via a relatable, interesting character and their immediate circle can be effective, easier to characterise, and, more often than not, truly empathetic. The futuristic town of Rockvil, the star in Infocom's classic interactive fiction, *A Mind Forever Voyaging*, let players experience its collapse first into an authoritarian dystopia and then into violent urban chaos not as an objective historical event, but via the troubles of a specific family in a specific part of town.



▲ Infocom's *A Mind Forever Voyaging* took a more personal approach to showcasing urban evolution. Cover courtesy of MobyGames.

paint a more legible picture, and don't be afraid to add emphasis by having characters discuss what's supposed to be important.

Approaching texture in layers often works well, too. Personally, I tend to start with the planning and architecture of a game space, and see how to spice up things there. Working on an already structured plan with defined districts, roads, important nodes, and land uses, I imagine the types of landmarks present – their style and scale as well as their relation to their surroundings. I also play with the balance of open and built-up spaces, and the city's local scales (from the intimate to the monumental), before considering localised street forms, lines of sight, density variations, design contrasts, and whether houses open up to the street or fight for space with productive and commercial uses.

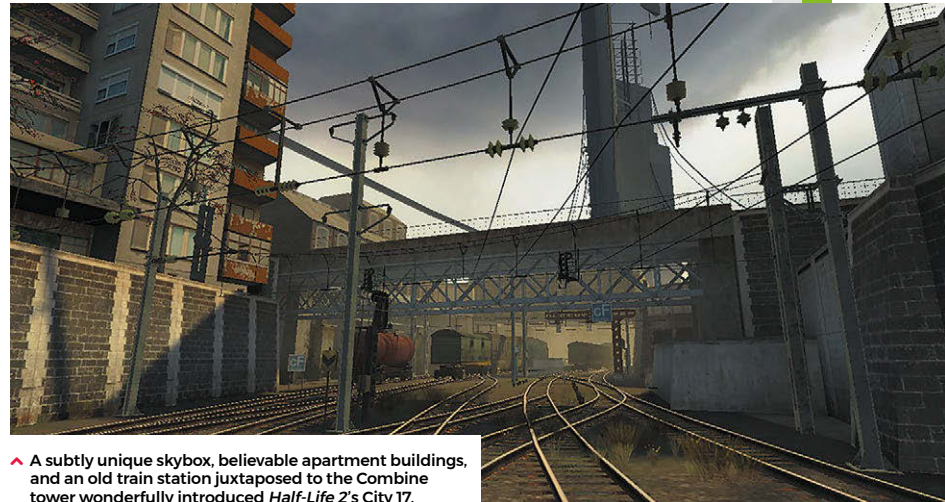
Then I try to imagine the activities of the inhabitants in as vibrant detail as possible. Do people dress in a particular way, depending on their job? Are there colourful idioms, sayings, or customs worth showing off? Do I want the parks of the aristocracy to feel serene and cut off, or would I rather spend resources on amplifying the sense of urban enclosure by adding large edifices in my city's backgrounds?

Other layers to consider are local art and entertainment, history, and, unavoidably, politics and the economy. These will influence each other as they interact, and finding pithy ways of expressing such interactions can be especially effective in constructing texture. Oh, and there's always the very game-specific layer of game, level, and narrative design that aims to reward exploration with surprises, support or inspire the gameplay, and establish themes.

DETAILS OF INTRIGUE

Details, preferably unexpected ones, can create the illusion of a real world. A switch that illuminates a furnished room, a seemingly unimportant character graced with a full dialogue tree, a dead tree in the park, the way light falls on a poodle, or a hidden shop packed with exotic wares can all imply that everything is possible.

On the other hand, the construction of all those details can be both expensive and time-consuming, and that is why recognising, focusing on, and properly framing the most effective ones is crucial. As is, of course, taking as much



▲ A subtly unique skybox, believable apartment buildings, and an old train station juxtaposed to the Combine tower wonderfully introduced *Half-Life 2's* City 17.



▲ A shop packed with characteristic (in this case, voodoo-related) merchandise is a fine way of adding texture to a game.

advantage of game-wide systems – such as day/night cycles and the weather – as possible.

Ultimately, we want to intrigue and fascinate the visitors of our cities. We want to surprise them and help sustain their suspension of disbelief, while inspiring them to interact with spaces in the ways we've already foreseen or designed for. We want to guide them, subtly draw their attention towards our details, weave those in readable patterns, and grace them with meaningful connections.

Although we can lead both the eye and the cursor, we sometimes need to discover what people want to fiddle with. We can never anticipate everything, and this is where playtesting can be helpful.

American game designer Al Lowe, when working on the first *Leisure Suit Larry* adventure, extensively tested the game to see what players typed in and attempted to do. He then simply addressed the most popular of these inputs by having the game react to them in some manner, and this is exactly what made its locations feel so real and textured. 🌐



▲ Urban spaces in open-world games don't have to be generic. Novigrad, from *The Witcher 3: Wild Hunt*, is a fine example of a thoroughly textured virtual city.

Procedural Details

Touches that add depth and detail to a city can be approached in a systemic or procedural way. The way trash accumulates on streets, the flying patterns of bird flocks, weather effects, tides raising fishing boats, and flaking paint on building façades are all effective, and relatively easy to simulate.

Make a karting game in Unity

It's easier than you think to go from zero experience in Unity to a functioning karting game. Mark shows you how



AUTHOR
MARK VANSTONE

Mark Vanstone is the technical director of TechnoVisual, author of the nineties educational game series, *ArcVenture*, and after all this time, still can't resist game coding. education.technovisual.co.uk

There's something hugely satisfying about racing around a track and beating your own lap times, so with this in mind, let's build a karting game where you can do just that.

We'll build the game using Unity: by the end of this tutorial, you'll have a fully playable racer – and most importantly, you should be able to understand how everything works.

If you haven't used Unity before, you'll need to head to unity.com and download the latest version. They provide an install manager called Unity Hub that allows multiple versions of the system to be used on the same machine, as there can be significant differences between one version and another. It's usually best to choose a version (perhaps consider the latest one) for a project and then stick with it.

USING BLENDER

If you already use a 3D modelling program, the chances are that Unity will be able to use the models that it creates, but if you haven't done much 3D modelling before, it's worth using Blender (blender.org) to create your assets for Unity. It's free, has plenty of tutorials, and also a friendly community ready to help you out. At the time of writing, the current version is 2.82, but earlier versions also work with Unity. We won't go into the details of how to use Blender here, but

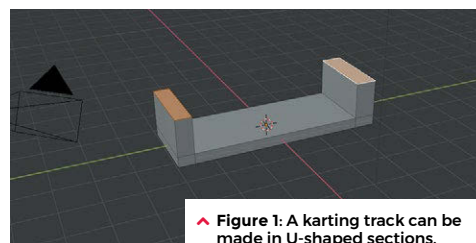
there are plenty of good guides and videos online. This tutorial will assume we're creating our models with Blender, but there's no reason why you can't use a 3D package you're more familiar with.

SETTING UP A TRACK

First, we need a track to race on. We can start in Blender with the default cube, and flatten it by scaling it on the z-axis and make it into a longer flat rectangle by scaling on either the x- or y-axis.

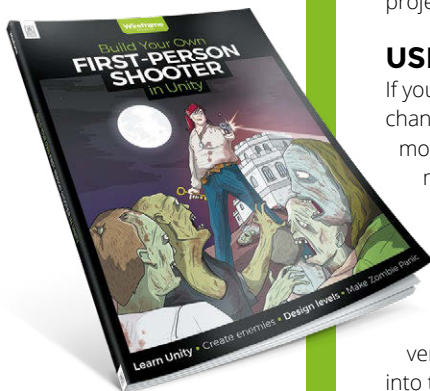
Now, if we go into edit mode, we can either extrude a small amount on the sides, or we could use the loop cut tool to create the foundation faces of the sides of the track (see **Figure 1**). We want to keep the polygon count low, as it's going to be used for collision detection later on, and the simpler the geometry, the fewer calculations are needed for collisions.

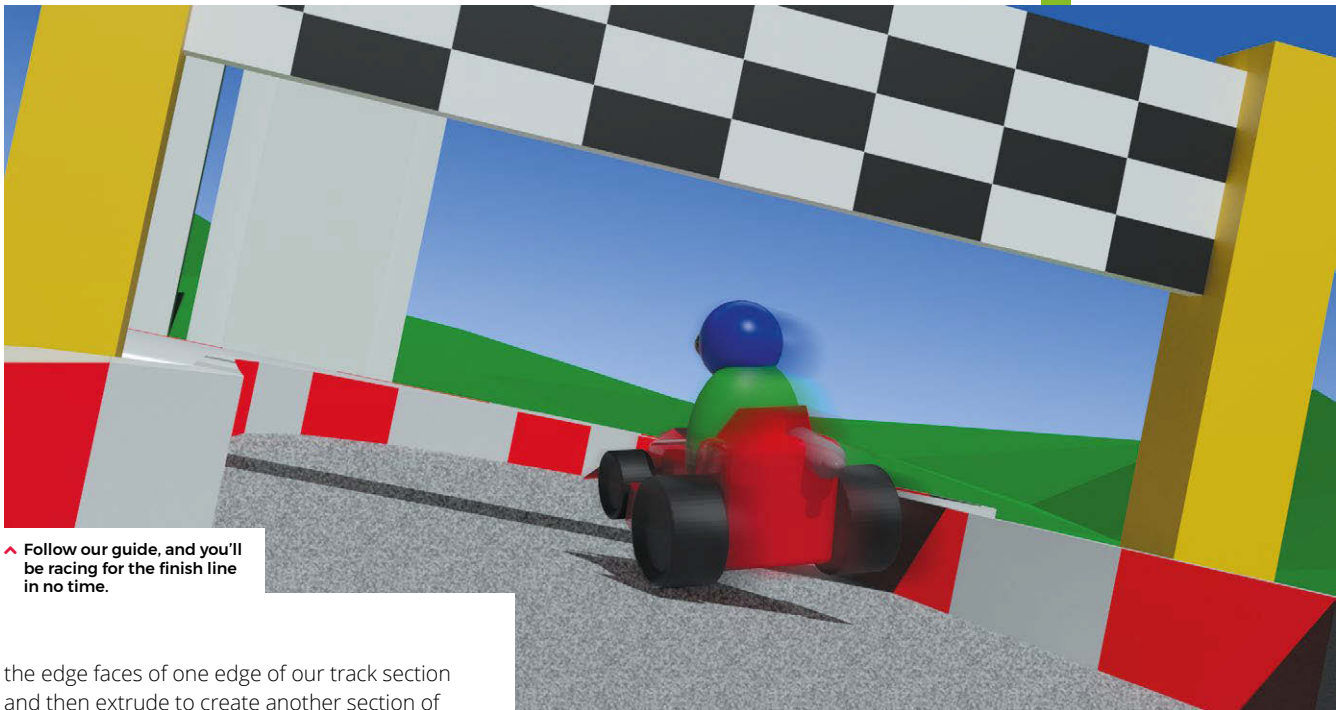
Next, if we extrude the side faces up, we end up with a 'U' shape which will make up one section of the track. Still in edit mode, we need to select just



▲ **Figure 1:** A karting track can be made in U-shaped sections.

♥ If you enjoyed this tutorial, then why not check out Wireframe's *Build Your Own First-Person Shooter In Unity*? Buy a copy now at wfmag.cc/fps.





^ Follow our guide, and you'll be racing for the finish line in no time.

the edge faces of one edge of our track section and then extrude to create another section of track. While the new section edge faces are still selected, you may want to rotate them around the z-axis to start creating bends in the track.

This track layout part of the modelling is probably best done from an overhead view, so that rotations only happen around the z-axis. Now keep extruding more sections and rotating to create a track that snakes around and then joins up with itself (see **Figure 2**). It's probably a good idea to make all the sections roughly the same size for consistency. (NB: If you prefer, you can use our pre-built track model, available to download at wfmag.cc/wfmag38).

For our game, the track model will be fine at this point, but you may want to add some scenery around the track itself.

Make sure that any other parts of the track file are separate objects and not part of the track object (you'll see why later).

“The simpler the geometry, the fewer calculations are needed for collisions”

We could create some rolling hills around the track and perhaps some buildings. It would also be good to have some sort of structure over the start and finish line. You could also apply some materials to the objects, such as a grey colour for the track and perhaps white and red stripes on the sides which will enhance the visual effect of speed as we race around the track (see **Figure 3** for our example). When you're done, make sure you save your work.

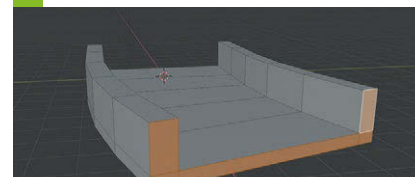
Now let's head to Unity. First, you need to create a new project in the Unity Hub

(**Figure 4**) – once that's loaded, you should see a nice empty project.

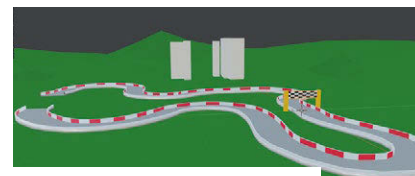
To import your track model into Unity, all you need to do is drag and drop your BLEND file into the Project Assets window. The file will be loaded into the project, and you can expand the new asset to see its internal parts (**Figure 5**). You may see that the thumbnail image of your model is tilted on its side. This is because Unity uses a different order of axes than Blender, but don't worry – it will work it out when we put the model into the scene.

We can now add the track to our scene either by dragging it into the Scene window or dragging it into the Hierarchy window. Either way is fine. When it's there, you'll see the track model added as a GameObject in the Hierarchy window and the model displayed in the Scene window. Make sure you save your work.

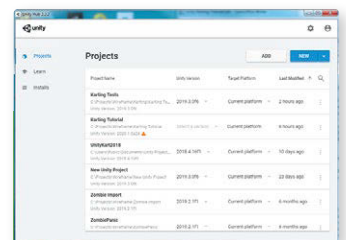
In Unity, it's best to save the scene and also the project (they're separate in the File menu). ➔



^ **Figure 2:** Keep extruding and rotating the sections until the track loops back on itself.



^ **Figure 3:** You can get creative with added scenery around the track.

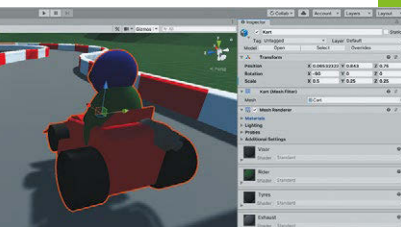
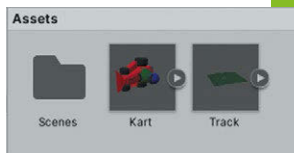


^ **Figure 4:** You can choose which version of Unity to use when you create a new project.



^ **Figure 5:** Expand an asset to see all its internal parts.

▼ **Figure 6:** Unity doesn't always show Blender models the right way around.



▲ **Figure 7:** Resize the kart model so that it's a good size for the track.

► **Figure 8:** Position the camera so it's facing the back of the kart.

ADDING A KART MODEL

With our track in place, we now need a vehicle. You can make a simple kart model in Blender, or simply just use our pre-built model, which you can download at wfmag.cc/wfmag38. Be sure to save your kart model as a BLEND file, and as with the track model, drag and drop the file into your Project Assets window in Unity. Again, don't worry too much about the orientation of the thumbnail (see **Figure 6**).

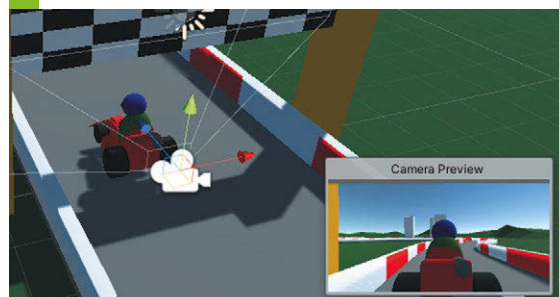
Now drag the kart asset into the Hierarchy or the Scene. You'll need to adjust the size and perhaps the rotation of the kart to line it up with the track. Don't make the kart too big or it may be difficult to get around the track – about a quarter of the width of the track is about right. You can alter the scale and rotation in the Transform panel in the Inspector window to the right of the scene (**Figure 7**). You may want to leave the kart a little way above the ground at the moment – you'll see why shortly.

If you now press the play button or go to the Game tab, you'll see a side view of the track with the kart just above it. We need to get the camera to look at the back of the kart to give us the player's view.

ATTACHING THE CAMERA

First, let's get the camera in the right place by selecting it in the Hierarchy and then moving it to a position behind the kart, either by using the Transform tools in the Inspector window or just dragging it around in the Scene window. Once it's in the right place, use the Rotation tools to change the y-axis until the camera's pointing at the back of the kart (see **Figure 8**). While the camera's selected, you should be able to see a Camera Preview window which will make it easier to get the view right.

Pressing the play button will show us the same view in the Game tab as we saw in the Camera

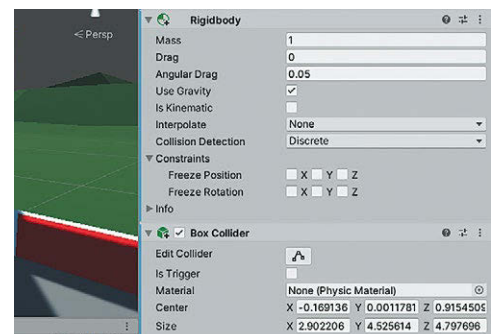


Preview window. If the kart were to move off down the track, however, the camera would be left where it is. To fix this, make the camera a child of the kart by dragging the camera object in the Hierarchy and dropping it onto the kart object. It will appear as a sub-object. Now, if you move the kart in the scene, you'll see the camera moving with it. Of course, we need to make the kart move in the game itself, but before that, we need to deal with something fundamental: physics!

Unity has a built-in system for dealing with how objects react to each other, and what we need to set up is for our kart to sit on the track while driving along it and bouncing off (or over) the side barriers if our kart hits them. To do this, we'll make our kart into a 'Rigidbody' and our track into a collider. So first, select the kart, then from the bottom of the Inspector window select 'Add Component'. Then, in the pop-up menu select 'Physics' and then 'Rigidbody'. This component adds weight to the object, so if left to its own devices in the game, it will fall downwards until it hits something (see **Figure 9**). We also need to define how collisions will be detected, so also add a 'Box Collider' component from the Add Component menu.

The other part of the collision equation is the track. We don't need to make this a Rigidbody as it does not need to have weight, but it needs to collide with our kart, so after selecting the track (and only the track part of the model), add a 'Mesh Collider' component. We need the detail of the Mesh Collider because the track can't be represented as a box – it has twists and turns and is a complex shape. Once the kart and track are set up with their physics components, you should be able to play the game, and you'll see the kart falling towards the track and stopping as the colliders take control.

▼ **Figure 9:** The kart needs some physics components – Rigidbody and Box Collider.



CONTROLLING THE KART

We can't have a karting game without being able to move the kart around the track, so we need to get some input from the player. So far, we haven't done any scripting, but C# scripts can easily be created to control Unity objects. To make a new script, right-click in the Assets window to pop up the menu and select 'C# Script' from the Create menu. The new script will appear along with your other assets, which you may want to rename to something like 'KartMovement'. You'll see in the Inspector window the script has already been filled in with some code.

To edit the script, you can double-click on it. Depending on how your system's set up, you should see the script open in a program editor. If you don't have a program editor installed, you can download the Microsoft Visual Studio Code editor free from code.visualstudio.com. To configure the code editor that Unity uses, go to Preferences in the Edit menu and select 'External Tools'. Once you're set up with an editor, the first thing to do is rename the class to KartMovement. In the script file, you'll see several lines above the class, which include the libraries the script will need in order to run. Inside the class's curly brackets - { } - you'll see two functions, one called **Start()** and the other called **Update()**. **Start()** will run when your game begins, and **Update()** will be run each time the scene refreshes, which will be many times a second. Have a look at the code below to see what we need to add to this script.

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

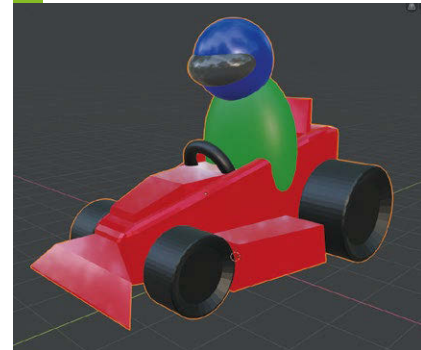
public class KartMovement : MonoBehaviour
{
    private float speed;
    private float drag = 3;
    // Start is called before the first frame update
    void Start()
    {
        speed = 0;
    }
    private void FixedUpdate()
    {
        transform.Translate(speed * Time.
deltaTime, 0, 0, Space.Self);
```

```
}
// Update is called once per frame
void Update()
{
    float rot = Input.GetAxis("Horizontal")
* 10;
    gameObject.transform.Rotate(0.0f, 0.0f,
rot* speed * Time.deltaTime, Space.Self);
    float acc = Input.GetAxis("Vertical")
* 10;
    speed += acc * Time.deltaTime;
    if(speed > 18){
        speed = 18;
    }
    if(speed > 0){
        speed -= drag * Time.deltaTime;
    }
    if(speed < 0){
        speed += drag * Time.deltaTime;
    }
}
```

You'll notice that inside our KartMovement class, we declare two variables; one to keep track of the kart speed and one to provide a value for the drag effect of the kart on the track. In the **Start()** function, we can set the speed to zero, which will mean the kart is stationary at the beginning of the game. We have added a new function called **FixedUpdate()**. This is a Unity function used to update aspects of Rigidbody objects that are connected to the physics engine. In this case, we will update the position of the kart based on the **speed** variable.

In the **Update()** function, we need to capture the input from the player. Unity has an object called 'Input' to handle this. If we ask it to **GetAxis("Horizontal")**, it will return us a value from a joystick or gamepad, the left and right cursor keys, and the keys **A** and **D** by default. Similarly, if we **GetAxis("Vertical")**, a value is returned corresponding to up and down motions, cursor keys, and the **W** and **S** keys. If we detect horizontal input, we rotate the kart, and if we detect vertical input, we increase or decrease the **speed** variable. We need to limit the speed to a reasonable top value, so we have a check for that, and then we can apply our drag value so the kart will slow down when the player stops accelerating.

We can now attach our KartMovement script to the kart. First, select the kart in the Hierarchy or ➔



▲ Our kart's just a basic example to get you started. You could create a more complex model if you're handy in Blender.



▲ Our UI means we now have a lap timer to race against.

the Scene, and then either drag and drop the C# script from the Assets window into the Inspector, or use the Add Component button and select the KartMovement script from the list of scripts. If everything's gone well, when you press the play button and use the cursor keys in the game, the kart should start moving along the track. If your colliders are set right, you should be able to guide the kart around the track, and if you go too fast, you'll hit the sides and probably spin into the air. Now, this is starting to look like a playable game!

DISPLAYING TIMINGS

Next, let's make a timer in the top left of the screen. To do this, we'll create a 2D overlay on the screen: a User Interface, or UI for short. First, go to GameObject in the main menu, then UI, and select Text. Unity will then create a new Canvas with a Text object inside and an EventSystem object in the Hierarchy.

Select the Text object, then select the 2D button at the top of the Scene window. This switches you into 2D mode so you can see the canvas and the Text object in 2D. You may need to zoom out a bit to see the canvas with a white outline and the Text object in the bottom left of the canvas. Change the Pos X and Pos Y values in the Inspector to move the Text object somewhere near the top left of the canvas. Now when you press play, the Text object should be displayed on the screen. You can also change the text size and colour in the Inspector.

To make the text display a time, another script is required. Create a new script as you did with the movement one, but this time call it 'Timer' – see code below to see what we need to put in this script. We need to add a new 'using' entry at the top so that the script will use the UI library, and then inside the Timer class, we can define a Text object and a value to keep track of the time since the game began. In the **Update()** function, the timer value is increased and then split into minutes and seconds, formatted into a nice string which is then passed to the Text object. Make sure this script is saved, and then attach it to the track object in the scene.

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

public class Timer : MonoBehaviour
```

```
{
    public Text timeCounter;
    static float timer;
    // Start is called before the first frame
update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        timer += Time.deltaTime;
        int minutes = Mathf.FloorToInt(timer /
60F);
        int seconds = Mathf.FloorToInt(timer -
minutes * 60);
        string niceTime = string.
Format("{0:0}:{1:00}", minutes, seconds);
        timeCounter.text = niceTime;
    }
}
```

When the track's selected, you'll see the Timer script in the Inspector, and a property called Time Counter, which is set to None (Text). This means we need to drag a Text object into that property, so drag and drop the Text object we made earlier from the Hierarchy into that property box. Now when you press play, you should see the Text object updating the minutes and seconds since the game began.

CRASH AND FINISH LINE DETECTION

Now we can check when the kart crosses the finish line and save the lap time so the player can try to beat it. All we need to do is add another Text object to the Canvas and put it in the top right of the screen in the same way we positioned the timer Text object. We can then expand our Timer script to incorporate the new Text object. But wait – how do we know when the kart crosses the finish line? We use an invisible cube. From the GameObject 3D section in the main menu, create a new cube, and position it where you want the finish line to be. You might want to stretch it out across the track to cover the whole width (see **Figure 10**).

With the cube selected, go to the Inspector and inside the Box Collider component, select 'Is Trigger'. This means our kart will drive straight

through it, but we'll receive a signal we can catch in a script. We don't want to see the cube in the game, so uncheck the tick on the Mesh Renderer component and it won't be drawn.

We can incorporate this trigger into our Timer script, but first, delete the script component from the track and attach it to our new cube. This means we can use the function **void OnTriggerExit(Collider other)** in our script, and in this function, set the last lap time Text object to whatever the current timer says and zero the current timer.

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

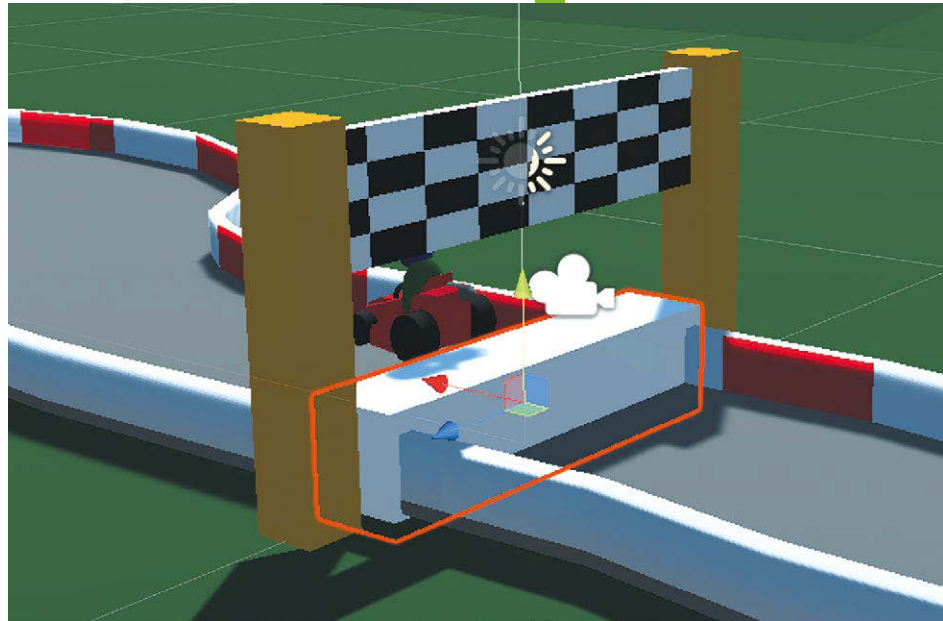
public class Timer : MonoBehaviour
{
    public Text timeCounter;
    static float timer;
    public Text LastLapText;
    public GameObject RestartButton;

    void Start(){
        RestartButton.SetActive(false);
    }

    public void restartGame(){
        RestartButton.SetActive(false);
        SceneManager.LoadScene("KartingScene");
    }

    // Update is called once per frame
    void Update()
    {
        timer += Time.deltaTime;
        int minutes = Mathf.FloorToInt(timer / 60F);
        int seconds = Mathf.FloorToInt(timer - minutes * 60);
        string niceTime = string.Format("{0:0}:{1:00}", minutes, seconds);
        timeCounter.text = niceTime;

        if(GameObject.Find("Kart").transform.position.y < 0){
            RestartButton.SetActive(true);
        }
    }
}
```



^ Figure 10: The finish line trigger is a cube that stretches across the track.

```
        if(Input.GetKeyDown("joystick button 9")){
            restartGame();
        }
    }

    void OnTriggerExit(Collider other){
        LastLapText.text = "Last Lap: "+timeCounter.text;
        timer = 0.0f;
    }
}
```

The only other thing we may want to add is a reset switch for when the kart goes flying off the track, which we can add as a UI button. Have a look at the revised Timer.cs script in the code above to see how we join all this together. When you have this script attached to the finish line cube, make sure you have set the Time Counter and Last Lap Text objects and set up a restart button to drag into the Restart Button field of the script. If all is set correctly, you should now be able to play the full game with the finish line triggering the timer and last lap time displays.

There's plenty more that can be done with this setup, but this should give you a good start to creating your own karting game. Try changing some of the parameters in the scripts to see the effects and get a good understanding of how it all works. All the assets for this project are available to download from wfmag.cc/wfmag38.

Firewatch's Henry: tutorialising narrative

How Firewatch's immaculate intro teaches the player to think and behave like a very particular character



AUTHOR
ANTONY DE FAULT

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

Internal Comms

The Twine introduction in **Figure 1** was originally made as an internal replacement for unwieldy character docs, as *Firewatch* developer Sean Vanaman felt it would be better to give the team something to play in order to understand who Henry was. Later, they were dealing with problems where the player didn't feel they knew how to respond as Henry, and at some point, the idea to make the Twine story an intro to the main game was born.

Firewatch is a game about overcoming the detailed, personal baggage of a man named Henry. Mechanically, the main game will have you simultaneously controlling interactive conversations over a radio while navigating 3D environments, and so for their intro sequence, developers Campo Santo strip that gameplay back to its two pillars: interactive fiction, and first-person exploration. They deliver a masterclass in efficient, impactful narrative design, so in order to illustrate these techniques, I've mapped the entire sequence on the opposite page, to which I'll be referring frequently. Let's dive in!

CHOOSE-YOUR-OWN

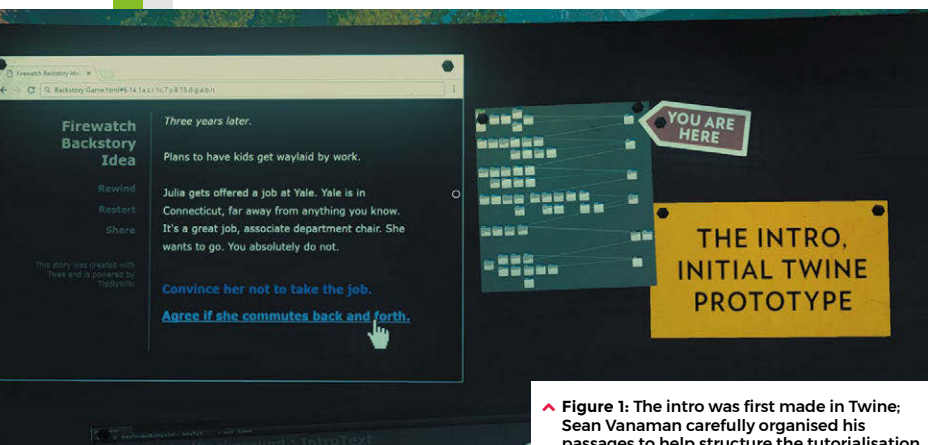
Firewatch opens with nothing more than text on the screen, which you advance through by

selecting what Henry does next. At first, you only have a single option, but soon you encounter choices between two possible actions which 'branch' down different pathways. For example, in Choice 1, the player must decide between two ways to break the ice with Julia, protagonist Henry's future wife, in a bar. They can either open with a compliment, or they can ask about her degree. It's a small choice, and each will give a slightly different passage of text on the next screen. This teaches the player that a) you must choose, and b) your choices will alter Henry's path. This is interactive fiction reduced to its most simple.

The next four decision points introduce slightly more consequential choices. Now they sometimes stick around (like choosing a dog) or are emotionally heavy (like choosing whether to yell at or ignore Julia). The branches are very short, rarely lasting longer than a passage before the narrative paths converge. One such initially innocent dilemma is Choice 5, in which Julia asks you to strike a pose for her to draw: flex like He-Man, or frolic like a Victoria's Secret model. We'll come back to that.

RAMPING UP

As the 3D Interludes teach movement and make us wonder where Julia is, Choices 6–8 become weighty, dealing with violence, major life decisions, and serious illness. Through these choices, the player is learning not only how interactive narrative works, but also how to role-play Henry. Every branch point is perfectly poised to have the player express not their own, but Henry's personal way of responding to



▲ **Figure 1:** The intro was first made in Twine; Sean Vanaman carefully organised his passages to help structure the tutorialisation.

She's not quite drunk but she's clearly been having a fun time. You fight when she gets between the sheets.

You get mad.

You ignore her.

▲ The presentation is stunningly simple: beautiful, readable, large text, on an impressionistic coloured background.

▼ This is about half the size of the raw map, which details every passage. You can see more at wfmag.cc/firemap.

the situation: "You're playing as a specific guy, with a specific life," says Jake Rodkin, one of the game's writers. In Choice 7, when Julia is offered a fantastic job across the country, your only choices are to a) convince her to stay or b) make her commute. You absorb that sometimes Henry is selfish, and so must you be if you're going to play him. "You don't have to learn it through his dialogue," says Rodkin. "You made the choices that brought him where he is today."

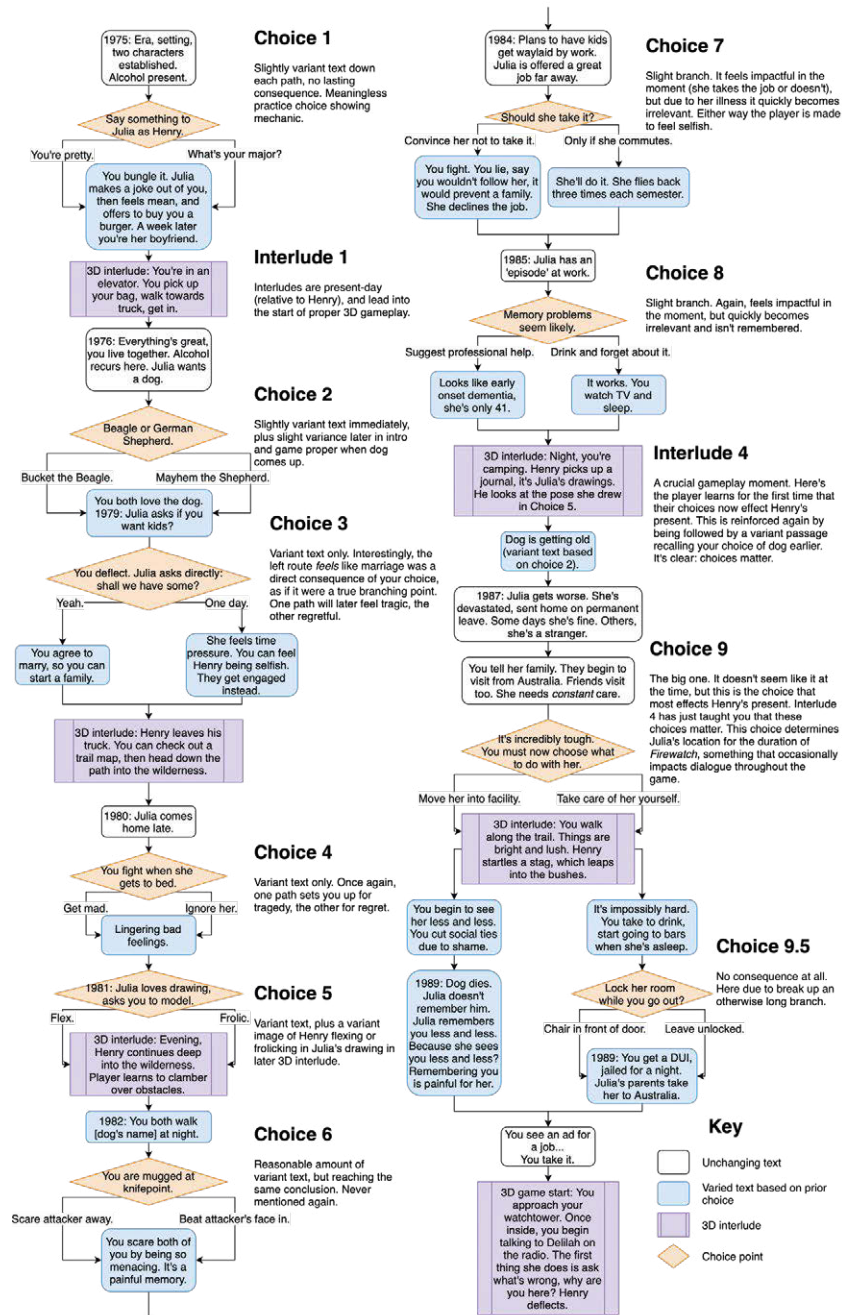
NO GOOD OPTIONS

Then comes the intro's climax. In Interlude 4, Henry picks up a journal and opens it to Julia's drawing of him. Depending on your earlier choice, the drawing can be either burly or burlesque. It hits the player: your narrative choices will impact Henry's present, permanently. It's not just differently flavoured text passages, and although the teaching example here is a silly drawing, what follows is anything but. Choice 9 lays on you the most traumatic dilemma yet, and the only 'true'

"Sometimes Henry is selfish, and so must you be if you're going to play as him"

branching point in the whole intro. Julia's dementia is unbearable: do you put her in a home, or continue her care yourself? As if it wasn't a tough choice already, we've just been taught that this could well have lasting impact. Later, the repercussions will affect both Julia's present-day location and Henry's motivation for taking the fire-watching job.

So what do we learn from *Firewatch*'s intro? Well, if you ask me, it succeeded not by allowing the player to express themselves, but by making players express personalised variances of a *very specific*, bitter-sweet character. It gets you to engage seriously with that character by taking its tutorialisation seriously, building to and earning its big, tragic branching point methodically, as laid bare opposite. 🍷



The subtleties of kingmaking

Kingmaking can be controversial in games. Let's look at what it is, why players do it, how to stop it – or why you might even encourage it



AUTHOR
STUART MAINE

Stuart Maine has been a designer for 22 years across PC, console, and mobile. He helped set up Well Played Games, and is currently working on Warhammer Combat Cards.

POWER PLAY

As the name suggests, kingmaking isn't limited to gaming, with people using it in ancient politics (when it was literally about making kings), to the nuclear arms race. In 2011, Davide Fiammenghi came up with 'the security curve'. This shows that as someone gains power, their immunity from attack goes up (as no-one wants to be punished), but if they continue to gain power it starts to drop (as others are forced to gang up on them). However, if they survive this phase, then further power gains effectively make them immune (with everyone turning on each other for second place).

Let's say you're in the final stages of a multiplayer game and you're winning. Your objectives are now simple – increase your lead and attack whichever player is threatening you the most. But what's your objective if you aren't winning? Most players would probably say 'to attack the winning player'. But now let's say the game is nearly over and there's no possible way you can catch the leader. Is your goal the same or do those factors change it?

Some players say that your goal must stay the same – to attack the player in first place, even if doing so lets someone else win, and you come last. Other players think that if you can't affect

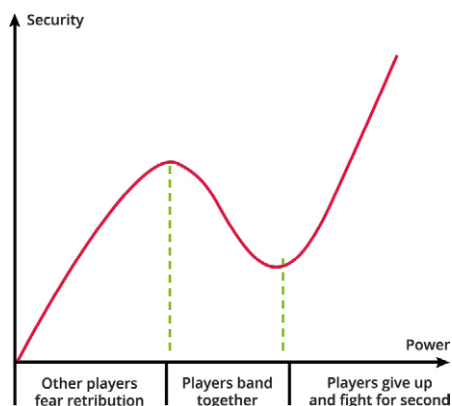
who wins, you should do whatever you can to come second, attacking players other than the leader to secure the best possible position you can.

If you go for the first option – attacking the game's leader to weaken their position and help someone else win – then you're kingmaking (or queenmaking; it's a historical term, but the point still stands). As you can imagine, kingmaking can be controversial, with players using it to help their friends win or even just drag down someone who wronged them earlier ('I'm not winning, but neither are you!').

SOCIAL CONTRACTS

It's important to note that kingmaking isn't cheating; no-one's breaking the game's rules to do it. They are, however, potentially breaking an implied 'social contract'. Briefly, a social contract is an unwritten code of rules that players expect each other to abide by. Things like 'Don't quit out of a game you're losing', or 'Don't get in the way of your teammates'.

The problem is, different players have different social contract 'rules' in their heads and get annoyed when others don't follow the behaviour they were expecting. Some players think the rule is, 'Always do what's needed to help your team win', while others say it's 'Do what you can to get the top score, even if it costs your team the match'.



> Taken from MIT Press' International Security journal, the 'security curve' argues that increasing power can actually make you a target.



▲ **Mario Kart** features several catch-up mechanics to ensure races are exciting, but the blue shell is by far the most controversial.

To bring this back to kingmaking, whether players attack the leader (even if it doesn't help their personal position), or attack someone else because it helps them, even if doing so pushes the leader further ahead, *someone* is probably going to get annoyed.

KINGMAKING IN ACTION

For a great example of kingmaking, check out Rock Paper Shotgun's series of diaries by players running through the turn-based strategy game, *Solium Infernum* (wfmag.cc/solium). With its complicated alliance and victory rules, this game strongly encourages politics and backstabbing, and it's fascinating to watch as players make secret deals and discuss who they're going to betray.

If you're thinking that kingmaking solely features in strategy games, consider *Mario Kart*. Amusingly, this game manages to annoy both players who enjoy kingmaking and those who don't, thanks to the much-argued-about blue shell. If I'm lagging in seventh place and the game gives me a Bullet Bill power-up, firing that is likely to take out the players in, say, fifth and sixth place, helping me towards my goal of winning the race. But the blue shell power-up skips over every player in front until it hits the race leader, helping the player in second place without improving my race position at all. That's kingmaking right there.

ENCOURAGE OR DISCOURAGE

So, what is it about some games that encourage kingmaking? Basically, the game must have more than two players, and players must be able to

recognise when they aren't going to win. Let's explore both of those, as they also provide methods for avoiding kingmaking.

First, if players can't tell that they've definitely lost, then they're more likely to keep fighting to win, rather than giving up and kingmaking. You can do this by hiding the scores until the end, by adding post-match bonuses, or by ramping up score values as the match gets close to the end.

Alternatively, you can make it difficult to know exactly when the game's going to end, such as by including multiple victory conditions (see *Civilization's* scientific, cultural, dominion, religious, and diplomatic victories). You can also link games together, so even if I can't win game one, it's still worth me pushing for points as that may help me win overall.

KNOW YOUR AUDIENCE

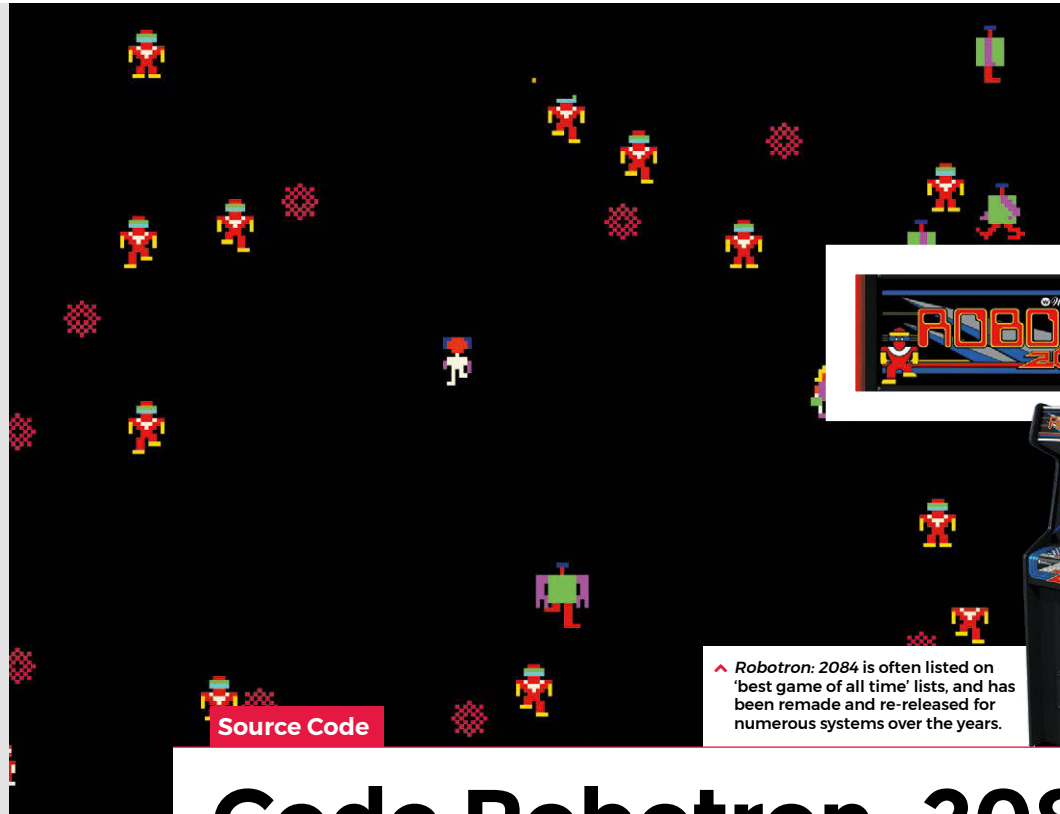
Despite those suggestions to prevent it, kingmaking isn't inherently bad for games; indeed, some players love this sort of backstabbing and politicking (see CCP's *EVE Online* for systems designed to encourage it). So rather than saying 'this is bad', I wanted to raise the issue so you can decide whether you want to encourage or discourage it, and how kingmaking affects the type of players you want your game to appeal to. ☺



▲ Games such as *Armello* demonstrate that kingmaking isn't inherently good or bad, with some players complaining that the game is rife with it, and others enjoying that.

KINGMAKING CONDITIONS

Kingmaking occurs in games for more than two players because you can't kingmake if it's just you versus an opponent. This can also apply if multiple players are grouped into two teams, either at the start of the game or by becoming allies. Why and when players become allies is a topic for a separate article, but whether a group of allies can win together (like the board game, *Cosmic Encounter*) or are forced to back-stab each other (like the *Game of Thrones* board game) will have a big impact on kingmaking.



^ *Robotron: 2084* is often listed on 'best game of all time' lists, and has been remade and re-released for numerous systems over the years.

Source Code

Code Robotron: 2084's twin-stick action



AUTHOR
MAC BOWLEY

Move in one direction and fire in another with Mac's re-creation of an arcade classic

Released back in 1982, *Robotron: 2084* popularised the concept of the twin-stick shooter. It gave players two joysticks which allowed them to move in one direction while also shooting at enemies in another. Here, I'll show you how to recreate those controls using Python and Pygame. We don't have access to any sticks, only a keyboard, so we'll be using the arrow keys for movement and **WASD** to control the direction of fire.

The movement controls use a **global** variable, a few **if** statements, and two built-in Pygame functions: **on_key_down** and **on_key_up**. The **on_key_down** function is called when a key on the keyboard is pressed, so when the player presses the right arrow key, for example, I set the x direction of the player to be a positive 1. Instead of setting the movement to 1, instead, I'll add 1 to the direction. The **on_key_down** function is called when a button's released. A key being

released means the player doesn't want to travel in that direction anymore and so we should do the opposite of what we did earlier – we take away the 1 or -1 we applied in the **on_key_up** function.

We repeat this process for each arrow key. Moving the player in the **update()** function is the last part of my movement; I apply a move speed and then use a **playArea** rect to clamp the player's position.

Now for the aiming and rotating. When my player aims, I want them to set the direction the bullets will fire, which functions like the movement. The difference this time is that when a player hits an aiming key, I set the direction directly rather than adjusting the values. If my player aims up, and then releases that key, the shooting will stop. Our next challenge is changing this direction into a rotation for the turret. Actors in Pygame can be rotated in degrees, so I have to find a way of turning a pair of x and y directions into a rotation. To do this, I use

the math module's **atan2** function to find the arc tangent of two points. The function returns a result in radians, so it needs to be converted. (You'll also notice I had to adjust mine by 90 degrees. If you want to avoid having to do this, create a sprite that faces right by default.)

To fire bullets, I'm using a flag called 'shooting' which, when set to **True**, causes my turret to turn and fire. My bullets are dictionaries; I could have used a class, but the only thing I need to keep track of is an actor and the bullet's direction.

You can look at the **update** function and see how I've implemented a fire rate for the turret as well. You can edit the **update** function to take a single parameter, **dt**, which stores the time since the last frame. By adding these up, you can trigger a bullet at precise intervals and then reset the timer.

This code is just a start – you could add enemies and maybe other player weapons to make a complete shooting experience. 🎮

Twin-stick shooting in Python

Here's Mac's code snippet. To get it running on your system, you'll need to install Pygame Zero – you can find full instructions at wfmag.cc/pgzero.

```
import pygame as pg
import math

WIDTH = 860
HEIGHT = 540

bg = pg.image.load("images/arena.png").
convert()
play_Area = Rect((150, 75), (560, 390))

player = Actor("treads.png",
center=(WIDTH//2, HEIGHT//2),
anchor=('center', 'center'))
turret = Actor("turret.png",
center=(player.x, player.y),
anchor=('center', 'center'))
pl_movement = [0, 0]
pl_move_speed = 5

pl_rotation = [0, 0]
turn_speed = 5
shooting = False
bullets = []
bullet_speed = 150
fire_rate = 0.15
fire_timer = 0

def on_key_down(key, unicode):
    global shooting

    # Movement
    if key == keys.RIGHT:
        pl_movement[0] += 1
    if key == keys.LEFT:
        pl_movement[0] += -1
    if key == keys.UP:
        pl_movement[1] += -1
    if key == keys.DOWN:
        pl_movement[1] += 1
    if key == keys.D:
        pl_rotation[0] = 1
    if key == keys.A:
        pl_rotation[0] = -1
    if key == keys.W:
        pl_rotation[1] = -1
    if key == keys.S:
        pl_rotation[1] = 1
    print(pl_rotation)

def on_key_up(key):
    global shooting
```

```
# Movement
if key == keys.RIGHT:
    pl_movement[0] = 0
if key == keys.LEFT:
    pl_movement[0] = 0
if key == keys.UP:
    pl_movement[1] = 0
if key == keys.DOWN:
    pl_movement[1] = 0
if key == keys.D:
    pl_rotation[0] = 0
if key == keys.A:
    pl_rotation[0] = 0
if key == keys.W:
    pl_rotation[1] = 0
if key == keys.S:
    pl_rotation[1] = 0

def update(dt):
    global shooting, bullets, fire_timer

    # Movement every frame
    player.x += pl_movement[0] * pl_
move_speed
    player.y += pl_movement[1] * pl_
move_speed

    # Clamp the position
    if player.y - 16 < play_Area.top:
        player.y = play_Area.top + 16
    elif player.y + 16 > play_Area.
bottom:
        player.y = play_Area.bottom - 16
    if player.x - 16 < play_Area.left:
        player.x = play_Area.left + 16
    elif player.x + 16 > play_Area.
right:
        player.x = play_Area.right - 16

    turret.pos = player.pos

    if any([keyboard[keys.W],
keyboard[keys.A], keyboard[keys.S],
keyboard[keys.D]]):
        shooting = True
    else:
        shooting = False
        fire_timer = fire_rate

    if shooting == True:

        # Rotate the turret
        desired_angle = (math.atan2(-
```

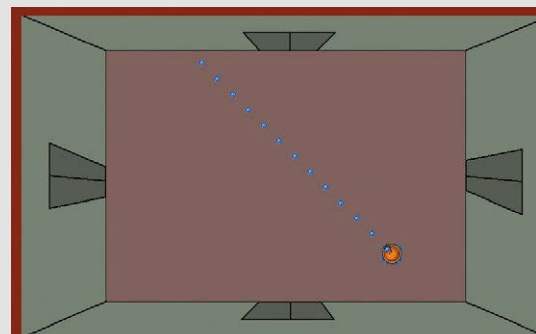
```
pl_rotation[1], pl_rotation[0]) / (math.
pi/180)) - 90
        turret.angle = desired_angle
        fire_timer += dt
        if fire_timer > fire_rate:
            bullet = {}
            bullet["actor"] =
Actor("bullet.png", center=player.pos,
anchor=('center', 'center'))
            bullet["direction"] = pl_
rotation.copy()
            bullet["actor"].x += pl_
rotation[0] * 4
            bullet["actor"].y += pl_
rotation[1] * 4
            bullets.append(bullet)
            fire_timer = 0

        bullets_to_remove = []
        for b in bullets:
            b["actor"].x += b["direction"]
[0] * bullet_speed * dt
            b["actor"].y += b["direction"]
[1] * bullet_speed * dt
            if not b["actor"].
collidect(play_Area):
                bullets_to_remove.append(b)

        for b in bullets_to_remove:
            bullets.remove(b)

def draw():
    screen.blit(bg, (0, 0))
    player.draw()
    turret.draw()
    for b in bullets:
        b["actor"].draw()
```

✓ The arena background and tank sprites were created in Piskel (piskelapp.com). Separate sprites for the tank allow the turret to rotate separately from the tracks.



Directory

Worthy causes for your consideration

A few charities and organisations that could really use your help in difficult times

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.

As you can't have failed to notice, these are tough times for just about everybody at the moment – there are few people in the UK who haven't been affected in some way by the coronavirus. But we wanted to highlight a few video game-related charities and organisations that deserve a helping hand right now.

SpecialEffect

This unique charity helps to make games accessible to people with physical disabilities, and freely supplies specially-designed gaming equipment for those who need it. Although the current situation means the charity can't do face-to-face visits at present, it's continuing to support users and provide equipment online. If you'd like to support SpecialEffect, there are multiple ways to do so – you'll find details on the charity's website: specialeffect.org.uk.

The Centre for Computing History, Cambridge

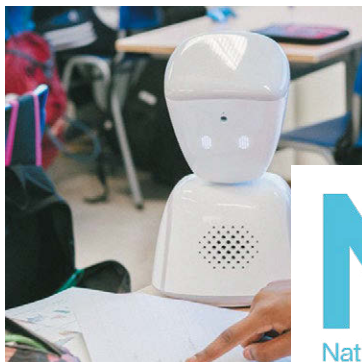
With over 36,000 exhibits, the Centre for Computing History has provided an invaluable

resource for visitors since August 2013. The coronavirus has meant that the museum has had to close to the public, which means it's lost valuable income over the past couple of months. If you want to help, you could donate to the centre's JustGiving page (wfmag.cc/CHgive), buy a twelve-month membership card for you and your family, or buy some items from its online gift shop (wfmag.cc/CHgifts) – the Games Consoles Trump Cards (housed in a cassette case for added retro appeal) really caught our eye.

The National Videogame Museum, Sheffield

Since 2015, the National Videogame Museum has provided an insight into video game culture, with dozens of exhibits, school workshops, and other events. Like the Centre for Computing History, the NVM relies on visitors to keep running, and has thus found itself greatly affected by the lockdown. If you'd like to make a donation, it also has a JustGiving page (wfmag.cc/saveNVM), and you can also become a member over at Patreon (wfmag.cc/nvm).

▼ SpecialEffect's projects include the wonderful BubbleBusters – a desktop robot that allows poorly children to communicate with friends and teachers when they're away from school.



▲ Just some of the systems on display at the Centre for Computing History, Cambridge.

◀ Sheffield's National Videogame Museum preserves and displays a wealth of classic titles and systems.

BIGGER. BOLDER. MONTHLIER.

The new **116-page, monthly edition** of your favourite games magazine: bigger, better, more feature-packed than ever. Out 4 June.



ON THE HUNT

Whether for money, upgrades, or simply survival, we hunt for a variety of reasons in the games we play. But what lies beyond these motivations? What is the player experience of the hunt, and how is it conjured?

WRITTEN BY
SEAN MARTIN

In recent years, hunting games have proven extremely popular – be it *The Witcher's* silver swords and alchemy, *Bloodborne's* trick weapons and dark city streets, or *Monster Hunter: World's* big-game pioneer fantasy. But why do we love hunting so much? In-game, those reasons are deceptively simple – harvesting currency and materials affords us the power to upgrade ourselves, making us more effective at hunting, and in doing so, gathering ever-more currency and materials. Either that, or harvesting prey simply allows us to survive and maintain ourselves. But as motivations, these cycles don't go very far to explaining what the player experience of the hunt actually is, or why we keep returning to it.

We've been hunting for thousands of years, and as time has passed, motivations have shifted, reflecting wider changes in our societies. But hunting, whether for sport or necessity, still retains that cultural history, which developers can channel when creating games. The hunt also offers a recognisable structure around which to craft mechanics. By exploring the fascinating ways in which developers are playing with these elements, we can begin to examine the true allure of the video game hunt.

HELL IS OTHER HUNTERS

Hunt: Showdown is one of the most original shooters to have emerged in recent years, in part due to its interpretation of an important aspect of the hunting experience: tension. In the backwaters of an 1890s Louisiana bayou, players face off against each other, tracking their supernatural quarry across the swamps in

▼ *Horizon Zero Dawn's* hunters craft their gear from the materials they harvest from machines.



hopes of claiming a bounty. But *Hunt's* secret? You don't know who else is out there. "Very early in the project, we realised the opportunity to create tension through deprivation of information," says Dennis Schwarz, lead designer on *Hunt*. "Keeping the player guessing about who else might still be in the game helped us instil a healthy level of paranoia. You can never be sure that you are truly alone on the map, so as a consequence, you have to expect an ambush around every corner." One of the main methods of achieving this tension is *Hunt's* masterful soundscape. While enemies represent a very real danger in their own right, the bayou is also littered with sound obstacles – broken glass, animals, chains, water, reeds, even footsteps must be considered. "All the sounds you hear [come] from a real source and attenuate realistically over distance," explains Florian Füsslin, *Hunt's* audio director. "If you hear crows fly off or a barrel exploding, someone must have caused that. In addition, we have a high dynamic



“With tension, rather than horror, you can keep the player on edge for longer”

▲ From shotgun-axes to explosive spears, *Hunt* reimagines the Victorian arsenal.

mix to have things stand out and give the player a realistic sense of how loud they are. That’s why hearing nothing can create more tension in an intensive firefight – it is too quiet to be true!”

Through this use of the unknown, *Hunt* creates an experience which places intense pressure on player approach. Through what Schwarz refers to as “the mental map of things happening around them”, players have to choose the right moment to act. “A patient hunter is a successful hunter,” Schwarz continues. “It’s important for players to read the flow of the game to understand when it makes sense to rush in and go for the kill, or when it’s better to wait for another team that made themselves known by triggering some crows, to cross that open field you have covered with your sniper rifle.” These second-by-second decisions, based on limited information, are what really tests the mettle of *Hunt*’s players, and creates such a unique hybrid experience. “We are very happy at having found a nice blend between the different aspects of a hunting experience

and the tension of a forsaken place overrun by terrifying horrors,” laughs Schwarz. “The key for us was realising that horror alone is not powerful enough. However, with tension, rather than horror, you can keep the player on edge for a much longer time. You could argue that the real tension in *Hunt: Showdown* doesn’t come from the monsters, as terrifying as they might be, but from the other players with their unpredictable movements at the most inconvenient times.”

THE THRILL OF THE CHASE

Two years ago, the fascinating robot-hunting epic, *Horizon Zero Dawn*, graced our screens for the first time. Mixing a science fiction concept with ancient tribal tradition, *Horizon* creates a form of future myth, a world where dwindling pockets of humanity must hunt machines to survive. “The world of *Horizon Zero Dawn* had to be a dangerous place in order to provide friction and challenge to the play experience, and the machines inhabiting it were a great ▶



▲ Joar Jakobsson grew up imagining fantasy animals, which prepared him well as lead designer on *Rain World*.

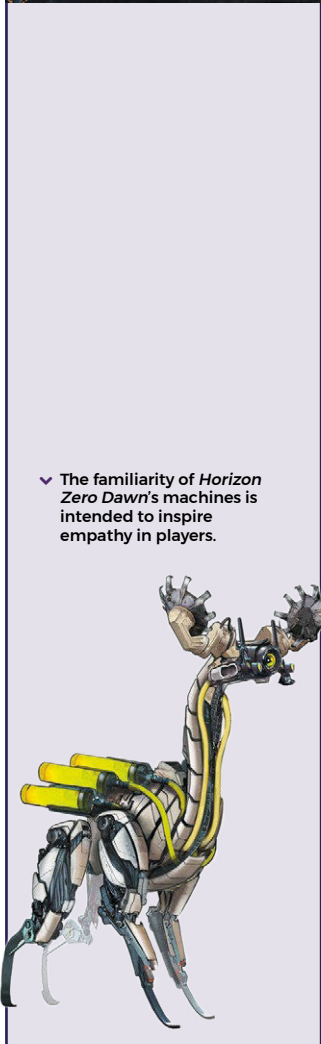
▼ Mathijs de Jonge was *Horizon Zero Dawn*’s game director.



asset for achieving that," says game director, Mathijs de Jonge. "They had to be dangerous and unrestricted by size or current technology. Conversely, human beings are limited in size and strength. It is basically a David vs Goliath setup."

As techno-tribal huntress, Aloy, players are almost always punching above their weight, but the experience of tackling these machines ties into another recognisable aspect of the hunt: the chase. It's an idea that implies momentum, which *Horizon* expresses not only through the incredible dynamism of Aloy's movements in traversing the environment, but in using your prey's own momentum against them. "For players to be able to play tactically and feel clever while hunting, Aloy had to be agile and equipped with a variety of tools and weapons," explains de Jonge. "The idea of using trip-wires to set off traps goes back very far in human history, so it made a lot of sense for us to use these as part of a more primitive toolset." Using a prey's momentum against them flies surprisingly close to real ancient hunting methods, such as the Native American buffalo jump, where animals were driven en masse over a cliff edge. Aloy also uses the ropecaster, a tool to ensnare even the biggest machines, and a method which can't help but conjure images of our ancestors' speculated tendency of preying on woolly mammoths trapped in swamps.

Every new machine encounter in *Horizon* is also a cautious experience – witnessing a creature which, though a recognisable facsimile, harbours alien capabilities. "The first fights are often about studying a machine to figure out how to best defeat it, but a large part is seeing all the cool things it can do," says de Jonge. "The fun of the fight – seeing a machine's abilities and behaviours, and adapting your tactics accordingly – has also been very important to us. It goes beyond merely hunting the machines for resources." *Horizon's* true uniqueness as a hunting experience lies not only in this David vs Goliath approach towards tackling prey, but by creating a complex relationship toward the very idea of 'prey' itself. By casting animals in wires and metal, *Horizon* raises a complex subtext of purpose, creation, and exploitation, channelling Descartes's idea of the 'animal machine', while also tapping into the rich parallel of machine exploitation in science fiction – all in order to question our relationship towards animals. As de Jonge so aptly states: "In many ways, the story



▼ The familiarity of *Horizon Zero Dawn's* machines is intended to inspire empathy in players.

of *Horizon Zero Dawn* is about the mistakes of mankind, and despite the beauty of the world it presents, there's a lot of tragedy at its core."

PREDATOR OR PREY?

Rain World was, without doubt, one of the most original games to emerge from 2017 – a unique experiment in both ecosystem and animal behaviour. You play as the humble slugcat, a creature fighting to survive in the sewers and smoke-stacks of what can only be referred to as a dystopian ecosystem, because *Rain World* is packed with terrifying predators. Whether they're masked vultures, neon lizards, or huge glowing spiders, they are all bigger, meaner, and highly adapted to eat you in their own particular fashion. Slugcat's only true advantage? Brains.

"Our idea about where the fun of the game came from was in outwitting creatures more dangerous than yourself," explains lead designer, Joar Jakobsson. "Slugcat's role as a nimble generalist and tool user grew from that. It felt natural that it would be more fun to take on the role of the clever little creature that can do anything to get by, rather than the hulking tank with a specialised and singular behaviour."

Through this approach, slugcat comes to reflect a key element of the hunting experience: skill. Through both tool-use and ingenuity, the slugcat becomes a rogue element within the ecosystem, able to define its own place within the food chain – while also paralleling the fluidity of the human role in the ecosystem.



▲ *Monster Hunter: World* creates an online group-hunting experience.

"The slugcat initially is very clumsy and weak, but with a skilled player controlling, it can be the deadliest creature in the environment," says Jakobsson. "Starting out weak and becoming powerful is almost omnipresent in games, but we decided to do it a little differently. Instead of gaining powers that make the character stronger and quicker, the idea was that the player's accumulated knowledge alone should support this progression."

Just as with hunting, slugcat's survival and success are also defined by luck – a significant part of which is expressed in the unpredictable behaviour of *Rain World's* creatures. "If you imagine each species arranged vertically and horizontally, every intersecting cell in that diagram would have something like 'hunts' or 'is afraid of' or 'territorial squabble' in it," explains Jakobsson. "The behaviours themselves, such as fleeing or hunting, were quite generalised, so I didn't have to handcraft every interaction. The idea here was of course that the game should exhibit emergent behaviour – basically 'do things that make sense but were not specifically added by the programmer'. Occasionally, the game does manage to surprise experienced players and even me, and that's when *Rain World* is at its best!"

"Our idea about where the fun of the game came from was in outwitting creatures more dangerous than yourself"

▶ The bayou is filled with abandoned docks, lumber mills, slaughter-houses, and even a Civil War-era fort.



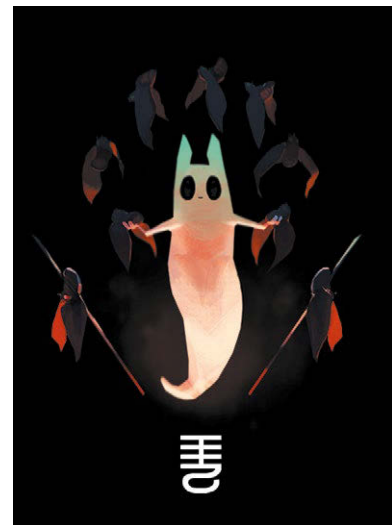
THE SOUND OF HUNT

"The soundscape of *Hunt* is built on three pillars: feedback, immersion, and emotion," explains Florian Füsslin. "Feedback is the audio language the player can read: what is it? Where is it? Is it dangerous? We want players to constantly make decisions based on sounds you hear around you. Immersion is most noticeable in the narrative and the game world. *Hunt* is based in the late 1890s, and the game takes place in the Louisiana bayou. Therefore the ambience has to reflect the hot and muggy weather, and the equipment has to have that gritty, haptic, heavy mechanical feeling of the arsenal back then. While grounded in reality, all sounds are designed to add emotion. The guns, for example, sound beefier to grant the player a satisfying feeling of power – or even fear if you are on the wrong side of the barrel."

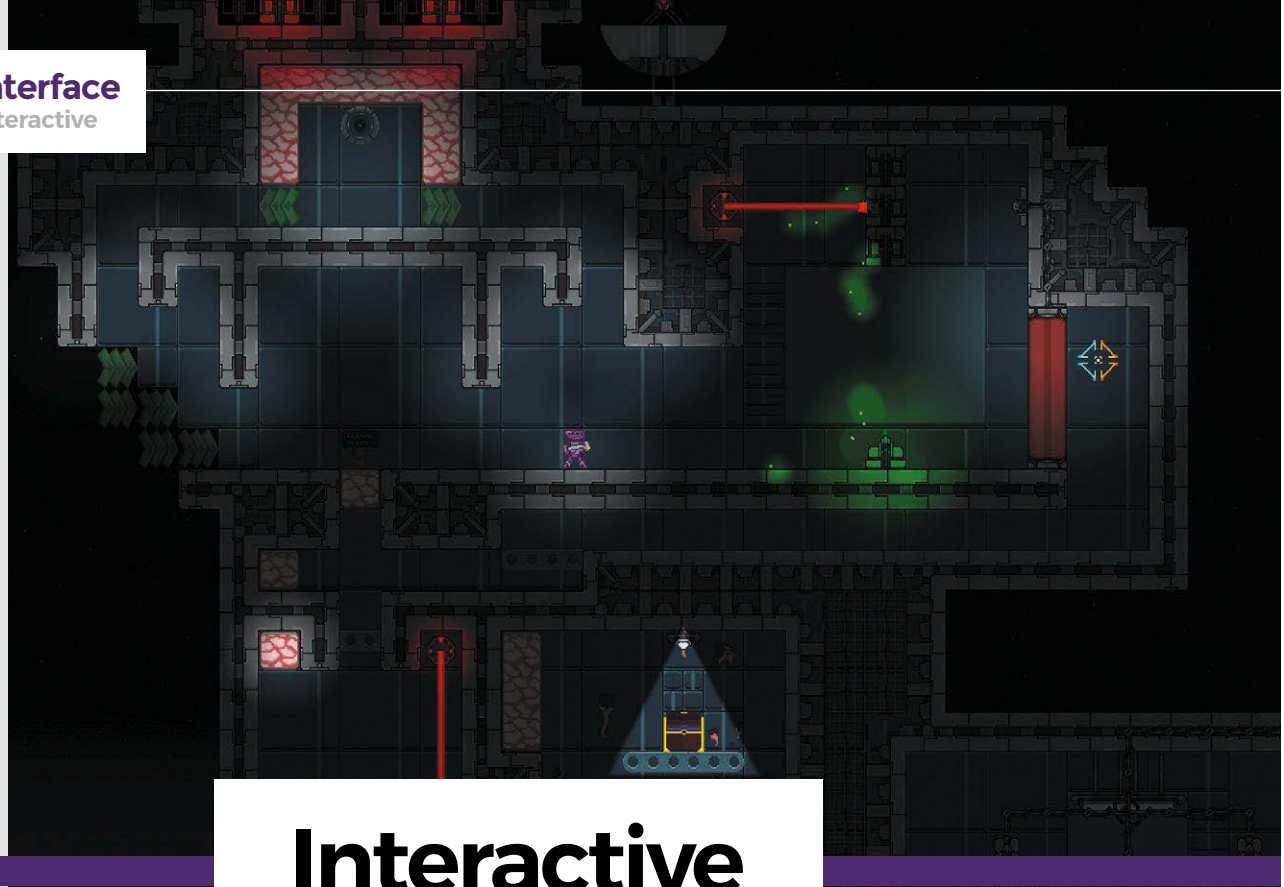
Rain World's hunting experience mirrors the origins of our own – tools and intellect allowing the hunting of larger, more dangerous prey. But it also forces players to recognise and respect the ecosystem. You have to respect something which can kill you nine times out of ten. At any moment in *Rain World*, the predator can become the victim, a perspective synonymous with most hunting games. "I think that as humans, we're almost hardwired to find it satisfying to overcome

difficult obstacles through wits and knowledge rather than raw power or single-minded specialisation," Jakobsson says. "Our

stories are full of trickster Davids, and in them, the brawny Goliaths are obstacles to overcome." This terminology of David and Goliath, earlier referenced by de Jonge, sums up a significant appeal of hunting games: overcoming the odds. Just as it was for our ancestors, the video game hunt is a struggle between opposing forces, differing in faculty and application, but both dangerous. What remains special about hunting games is the way a timeless struggle is consistently reimaged. It could be a bounty hunter stalking another in the demon-haunted swamps of Louisiana, a tribeswoman chasing a mechanised colossus in the far future, or simply a slugcat looking to turn the table on a hungry lizard. 🐸



▶ In *Rain World*, players can gain end-game achievements for hunting.



Interactive

Portal Mortal

It's the unholy offspring of Portal and Super Meat Boy. Introducing Portal Mortal...

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

Here's one of our favourite origin stories in video game history. Way back in the mid-eighties, designer Will Wright created a level editor to help with the development of his top-down shoot-'em-up, *Raid on Bungeling Bay*. Discovering that the process of constructing level layouts was more absorbing than the shooter itself, he kept refining his level editor until it became a standalone game – the seminal *SimCity*. We mention this not just because there's a distinct construction theme running through this edition of Wireframe, but because platform puzzler *Portal Mortal* had a similar genesis.

In 2012, Finnish indie developer IsmoLaitela (not his real name) was working on a completely different game called *Circuit Confusion*, which involved rotating circuits to control the flow of electricity. Like Will Wright years earlier, IsmoLaitela created a level editor to speed up the construction of puzzles, only for it to gradually build into an entirely new game with a life of its own. "It soon became so much larger that I just scrapped the original project," IsmoLaitela tells us.

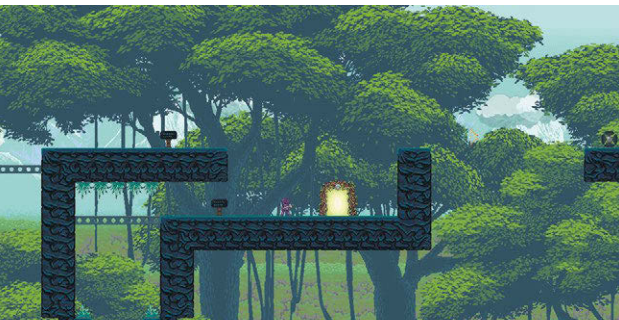
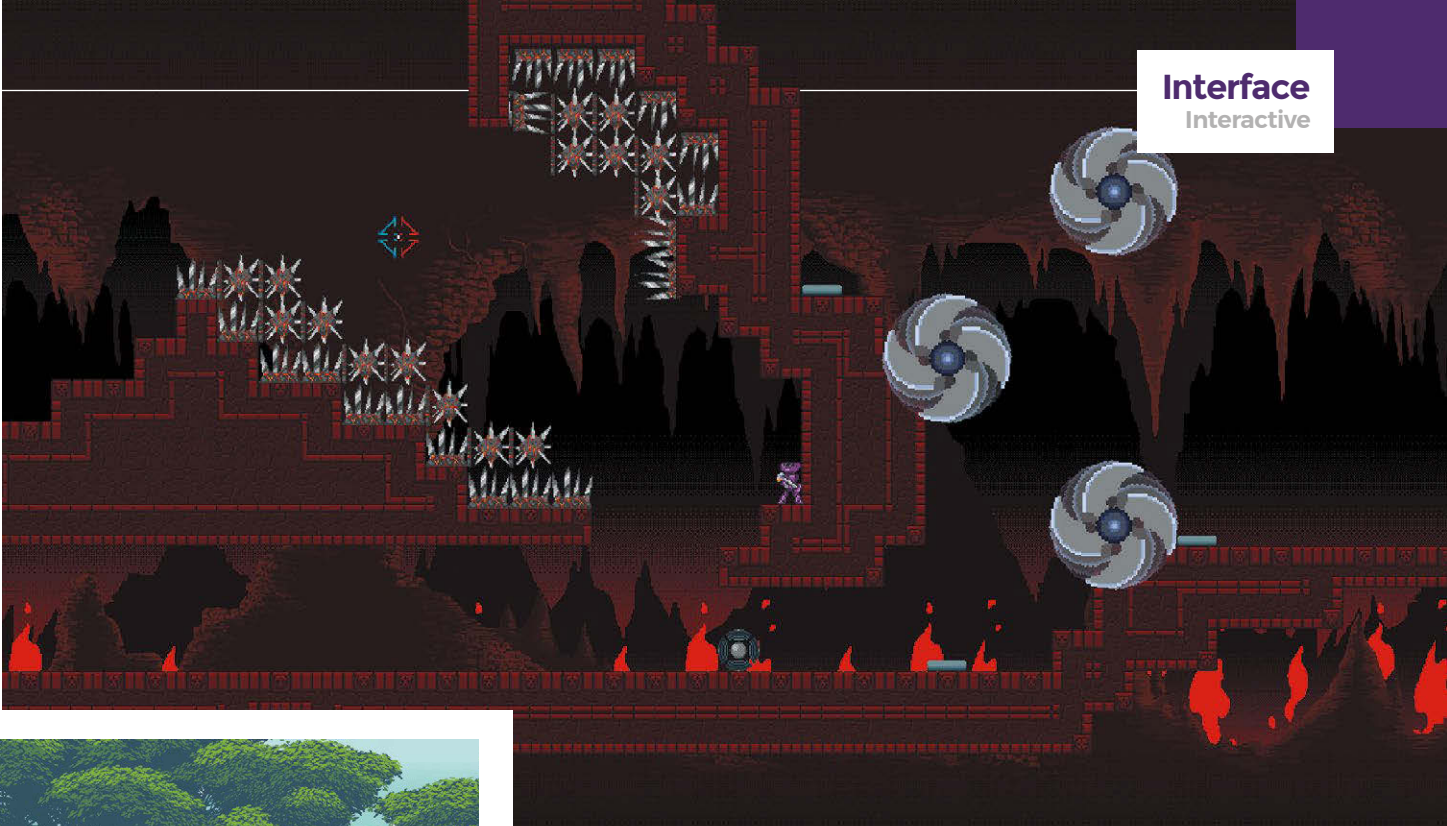
"Everything seems to work rather nicely, even though I think there's still at least one bad underlying bug"

Portal Mortal's title provides a clue as to what it's all about: think of a 2D take on Valve's *Portal*, but with the snappy speed and repeated gory deaths of *Super Meat Boy*, and you're getting close to the essence of its platforming action. Like *Portal's* Chell, your alien-looking character is armed with a gun capable of zapping them to different locations – firing at one wall with the left mouse button will create an entrance; firing at another wall with a click of the right button will forge an exit. Canny use of this ability will

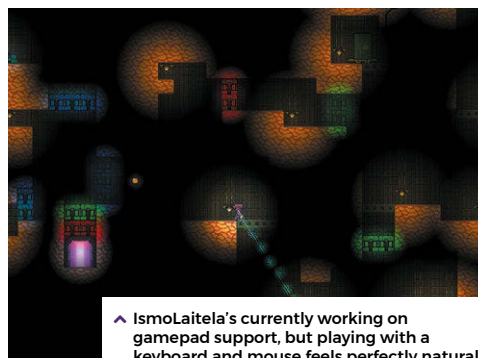
allow you to traverse the assorted barriers and traps IsmoLaitela has set out for you, and while the game's flat, side-on view might leave you thinking it's

easier to get to grips with than *Portal's* mind-boggling spatial puzzles, the sheer number of spikes, deadly lasers, and spinning blades mean that death is seldom too far away. Even the earlier stages require some pretty deft uses of tricks like flinging, and it isn't uncommon for backgrounds to become covered in blood as you repeatedly mess up your attempts to reach the exit.

A game made in IsmoLaitela's spare time over the past seven-or-so years, *Portal Mortal*



^ "I've already seen some fan-made content, like completely modded sidebars, backgrounds, and blocks," IsmoLaitela says of the game's editable nature.



^ IsmoLaitela's currently working on gamepad support, but playing with a keyboard and mouse feels perfectly natural.

^ Even straightforward-looking levels can contain devious traps and challenges.


was built in GameMaker Studio 2, and getting something as important as its physics and handling just right has, he admits, taken quite a while. "I'm still not sure if I'm completely happy about them," he says, candidly. "The player has full control over their character: they stop almost instantly after letting go of the 'up' key, and have full air control – unless they're being flung from a portal. There have been plenty of issues with this, and at one point, fixing one [mechanic] seemed to break another one. But nowadays everything seems to work rather nicely, even though I think there's still at least one bad underlying bug. I'm not sure if I'll ever manage to get it fully fixed."

SUPER PORTAL MAKER

Fittingly for a game that began as a level editor, *Portal Mortal* itself is open and hugely editable: players can not only create their own stage layouts, but also add their own sprites, audio, and sound effects. There's even a multiplayer mode – another inclusion that's provided IsmoLaitela with a design challenge, particularly as he's had to fit development on the game around his other work. "I do code as my day job,

mostly back-end stuff, but that's in a different field," he tells us. "I think this may actually be a good thing, since writing back-end stuff for *Portal Mortal* feels like working and has been not-so-motivating. Because I work whenever I want, I sometimes just open GameMaker and do nothing, and some days I spend a couple of hours writing code for a new feature or whatnot. So far I've managed to fit it nicely in-between sleeping, working, gaming, and marriage."

Although *Portal Mortal* is currently free to download, IsmoLaitela also has plans for a commercial release on Steam, with ports to Linux and Mac. Until then, he's still refining things like menus and gamepad support – fiddly jobs that, he admits, he's had to muster the energy to finish up. "It's not that they were too complex, but rather the lack of motivation I feel towards them. Luckily, my artist keeps on pumping out new graphics, so it gives me a bit of motivation to push forwards with these things!"

You can download *Portal Mortal* for free at portalmortal.net. 

MORTAL DANGER

"I've never used paper to scribble things," IsmoLaitela says when we ask him what his process is for designing levels. "I just boot up the game, throw in some blocks, and see what happens from thereon. The process of creating levels is so easy, and as I can test my ideas right away, I don't see the need for an extra step." As for balancing difficulty, IsmoLaitela says he doesn't think there are any "overly complex or over-the-top impossible puzzles" in *Portal Mortal* – apart from one. Called Trauma, the level was essentially designed by several players in one online session. "Builders could see us in edit mode trying to solve their puzzle, so they were able to adjust changes instantly on the go, and ask us to go over specific parts again until they were completely satisfied." The resulting design, IsmoLaitela suggests, was swear-inducingly tough. "It was a fun thing to do," he says, "but the level itself can go [have fun with] itself due to its difficulty curve."

Developer Profile

Bluepoint Games

The remastering maestros... remaestros

**W**

hen forming a game development studio, the common desire is to build something of your own; to unleash

your creative desires on a world that, to this point, likely hasn't been open to hearing the message you want to share with it. It's a move that says 'Here we are, this is what we're doing, and this is a reflection of us through our art'. You don't really get into game development – in the most part – so you can pick up games that have already been made by someone else, and tinker with their code and assets so they can function on newer (or just different) devices and look better (or, again, just different). Basically, you don't get into games with porting and remasters in mind.

Bluepoint Games started out with an original release in 2006 – *Blast Factor*, a downloadable PS3 title released in

November 2006, when digital-only console releases were in their relative infancy. It was the studio's first year of existence, a game on the more basic side of things, and didn't really show anyone much of anything, to be perfectly honest. Had the team continued down this road, it would likely have ended up as just another studio you forgot released a twin-stick shooter on the PlayStation Network. Because seriously, there were *so many* twin-stick shooters on PSN.

But that wasn't the path Bluepoint opted for, instead pivoting to the not-in-the-slightest new arena of remastering games. Sony wanted to bring back a bunch of its PS2 classics, wrapping them in a bit of high-definition sheen and using them as none-too-stealthy marketing for the bigger releases on PS3. The *God of War Collection* put the first and second entries to Kratos' action-adventure (and violence) romps into one package, slapped a 720p coat of paint on them, and threw them out to the masses as a taster for the then-upcoming *God of War III* (in the US, at least: the *Collection* released post-*GOW III* in Europe). It was well-received critically and commercially, and was the moment Bluepoint realised what its calling truly was: to make the old new again.

Bluepoint's second remaster followed a similar path: a bundle of two PS2 games, given a lick of HD paint, and re-released on the PS3 as a bundle. *The Ico & Shadow of the Colossus Collection*, though, did one extra-special thing that made players sit up and take notice: it fixed technical hitches from the second game in the package. This wasn't just a lazy hack



^ Sony has entrusted its most-loved franchises to Bluepoint.



^ Bluepoint is rumoured to be working on a Demon's Souls remaster for PS5.

job thrown out for easy money; it was something the team behind it clearly cared about – at least to some extent – resulting in players finally being able to play *Shadow of the Colossus* without crushing frame rate drops as experienced in the PS2 original. There was no stopping Bluepoint now.

AND NOT JUST SONY

Metal Gear Solid saw the remastered collection treatment (and Bluepoint's first move away from a Sony format, with the bundle also coming to Xbox 360), while a few titles – *PlayStation All-Stars Battle Royale*, *Flower*, and *Titanfall* – were straight-up ports from one system to another, rather than remasters. But all were of a type: games that had already been made, being redone for another format. Bluepoint became synonymous with 'this will be good', so when *Gravity Rush* and *Uncharted* were also fired out of the remaster cannon – the former

from Vita to PS4, the latter PS3 to PS4 – it was of little surprise they were of a type. Said type being 'good'.

But it was, fittingly, Bluepoint's return to a game it had already remastered

“The future of Bluepoint seems healthy, with the studio already working on a PS5 project”

once before that showed how strong the heart of the studio really was. *Shadow of the Colossus*, brought back once more in 2018, this time for PS4, saw more than the bit of work bringing it up to a smooth, playable standard, as seen in the PS3 do-over. The project came about because of Bluepoint's good relationship with Sony, but also because employees loved the game and wanted to make a 'definitive' version of it. This meant not just another

coat of paint, but breaking everything down and doing it from scratch – all assets were remade and structured around the skeleton of the PS2 original's codebase. Controls were updated, and graphical effects offered choices for players on the different tiers of PS4 systems – it was... well, definitive. The best version of one of the PlayStation's best games, which has been released three times but was only made by its creator once. What a strange little situation.

The future of Bluepoint seems to be looking healthy, with the studio, now around 90-staff strong and still sitting pretty in Austin, Texas, already working away behind closed doors on a project for the PS5. Rumours are being thrown at the studio every day, with a remaster of *Demon's Souls* seemingly the top of that particular pinch-of-salt list. Or maybe *Syphon Filter*. *Resistance?* *Silent Hill* could be a goer. Maybe the team will have a third pop at *Shadow of the Colossus*. Who knows? Whatever the case, it's a fair assumption the next game from Bluepoint will be a remaster (or port) in some form – and there's even less doubt it'll be handled incredibly well, and turn out arguably better than the original did. Unless it's *Blast Factor 2*, then all bets are off. 🤖



^ Three versions of one game, though its creator only had a hand in one version. Hmm.

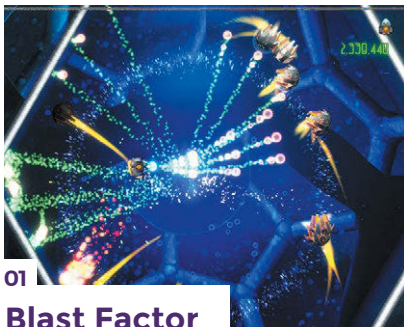
^ The team, based in Austin, Texas, is now around 90-people strong.



Bit o' blue

10 perfectly polished ports

Well, nine ports and an original game, at least

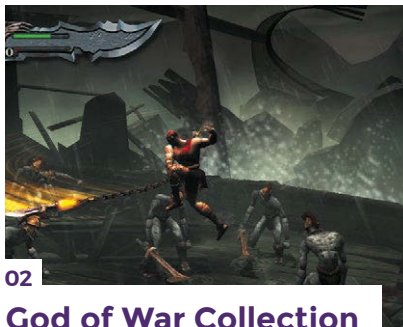


01

Blast Factor

PS3 – 2006

A straightforward twin-stick shooter, Bluepoint's only original release to date was half-decent, but not the sort of thing that stood out particularly in a mid-noughties field. If you were around back in the day, you probably played it – but really only because there were so few PSN games available at the time. Basically, *Geometry Wars* exists, so *Blast Factor* has been forgotten.



02

God of War Collection

PS3 – 2009

Sony's first HD remaster collection on PS3 was handed to Bluepoint, and it shouldn't be understated how important this project was to the publisher, with *God of War III* on the horizon in the US. Bluepoint delivered a well-presented collection of two games that offered a fine visual step up from the PS2 originals. Not much *changed*, but not much needed to.

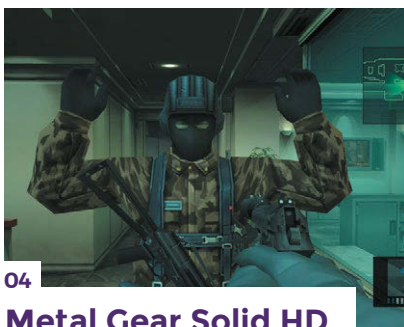


03

The Ico & Shadow of the Colossus Collection

PS3 – 2011

Two commercial failures were given another chance thanks to this Bluepoint do-over. Sony had settled into a rhythm with its HD remaster series, and opted for Team Ico's classic titles to be given a makeover. The end result was the games presented with a stability that just wasn't possible on PS2. A fine collection.

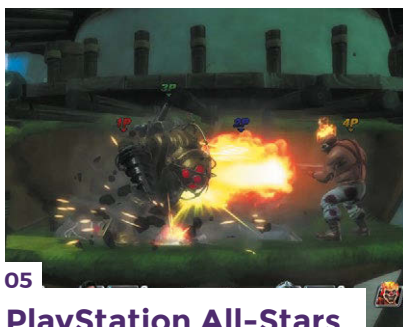


04

Metal Gear Solid HD Collection

PS3 / Vita / X360 – 2011

Konami contracted Bluepoint for this one, with the studio riding high on a remaster reputation thanks to stellar work so far. The *MGS HD Collection* was another feather in the cap, with a selection of well-crafted, considerate do-overs for some great games. It also marked the team's first foray into non-Sony consoles.



05

PlayStation All-Stars Battle Royale

Vita – 2012

Away from remasters, Bluepoint also saw some porting work come its way. *PlayStation All-Stars Battle Royale* might not be particularly well-remembered now, but the team's work on getting the PS3 original crammed into the Vita was magnificent, and arguably made it the definitive version – a brawling PaRappa on the go!



06

Flower

PS4 / Vita – 2013

Sony, needing an easy win for the PS4 launch, went once again to the reliable hands at Bluepoint. 'Make *Flower*, that amazing art game starring Petals, look better and work on PS4', the publisher said. Bluepoint had a fantastic game to work with already and, we'd assume, not a huge amount of difficulty in getting it up and running on the new console. So it did. The end.



07
Titanfall

X360 – 2014

There was no Sony at all in this one, just a straight port of the new-gen Xbox title to a past-gen Xbox console. *Titanfall* on Xbox 360 ended up being a fantastic port – it was feature complete, with nothing missing from the creative FPS's mechanics, and while it did see some obvious graphical shortcomings, it still ended up as a brilliant way to play the game.



08
Uncharted: The Nathan Drake Collection

PS4 – 2015

Another day, another fantastic remaster collection from Bluepoint, for Sony. When a publisher lets you handle its biggest franchises repeatedly, you're onto something. And would you believe it, the *Nathan Drake Collection* was a fantastic remaster – even if the underlying mechanics have aged somewhat.



09
Gravity Rush Remastered

PS4 – 2015

It's fair to say this was a less obvious one, with Sony handing over a cult classic to Bluepoint to help it jump from its original home of PS Vita to the less gimmicked-up PS4. It looked better, ran smoother, and generally ended up being the *better* version of the handheld original – as well as a good advert for the then-upcoming sequel.



10
Shadow of the Colossus

PS4 – 2018

All of which led to this: Bluepoint being able to request a chance to do *Shadow of the Colossus* one more time. The passion that went into this remaster is palpable, with the finished item being undeniably the original PS2 game, but with more than just an HD lick of paint on top of it. It is Bluepoint's masterpiece, built on top of Team Ico's.



Wireframe

GOES MONTHLY

The new **116-page edition** of your favourite games magazine: bigger, better, more feature-packed than ever



More features • **More game design guides**
More reviews and previews • Plus even more gaming goodness



Wireframe

The MagPi

HackSpace
TECHNOLOGY IN YOUR HANDS

CUSTOMPC

3 ISSUES FOR £10

+

FREE BOOK



Visit wfmag.cc/freebook
or call 01293 312192

Subscribe to Wireframe, The MagPi, HackSpace magazine, or Custom PC. Your first three issues for £10, then our great value rolling subscription afterwards. Includes a free voucher for one of five fantastic books at store.rpiexpress.cc/collections/latest-bookazines. UK only. Free delivery on everything.

Info

GENRE
Masocho
action-RPG

FORMAT
PS4 (tested)

DEVELOPER
Team NINJA

PUBLISHER
Sony

PRICE
£49.99

RELEASE
Out now

REVIEWED BY
Alan Wen

HIGHLIGHT

With former protagonist William not yet in the picture, *Nioh 2* lets you create an avatar that's as diverse and detailed as you like. The level of customisation provided is frankly ridiculous, which does pay off during cutscenes. For the fashion-conscious, if you're especially attached to an outfit with poor stats, you can even change the appearance of higher level gear to your preference.

Nioh 2

A second bout for the strongest Souls-like contender

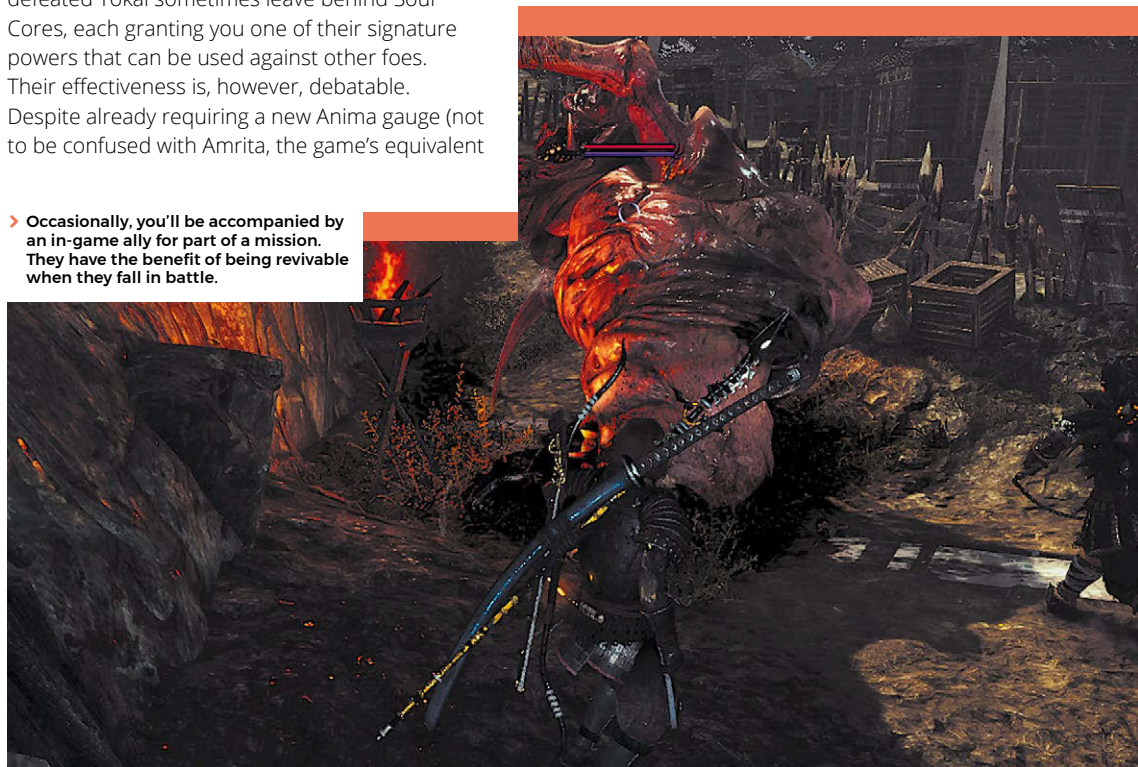
Nioh 2 only really iterates on what came before, rather than offering anything bold and new as the first game did. A key mechanic that made the original stand out as a genuine *Souls* contender and not just another pretender was – and honestly, the pun is unintentional – Ki. A delicious tactical twist on stamina management, you could either quickly recharge this gauge for another attack or leave yourself disastrously winded. Bringing down enemies' Ki was crucial to opening them up to more devastating damage, an idea that seems to have been an influence on *Sekiro's* posture and death-blow mechanics.

This sequel doesn't have anything quite as game-changing, though it does its utmost with a few hooks. First, *Nioh 2's* hero is a half-Yōkai – a supernatural entity originating from Japanese folklore. There's something of a gotta-catch-'em-all mentality to the action, as defeated Yōkai sometimes leave behind Soul Cores, each granting you one of their signature powers that can be used against other foes. Their effectiveness is, however, debatable. Despite already requiring a new Anima gauge (not to be confused with Amrita, the game's equivalent

of the souls in *Dark Souls*), these abilities run on lengthy cooldowns so can't be used liberally. The same applies for the new Yōkai Shift ability that's akin to *Devil May Cry's* Devil Trigger, which takes even longer to charge up, and never deals out substantial damage to make or break encounters where it matters most. Evidently, Team NINJA had no plans on giving players an easy out – the combat remains as hellishly brutal as its predecessor, arguably more so. But then, masochist that I am, I'd have it no other way.

There's no shortage of threats, whether human or Yōkai. The latter are naturally the bigger concern, though don't underestimate being pounced on by a group of guards who don't politely wait their turn. The ape-like Enki and one-eyed Oni are obvious brutes, but it's amazing how those at the bottom of the pecking order, like the Gollum-esque Gaki, still have the capacity to ruin your day in the late game.

Occasionally, you'll be accompanied by an in-game ally for part of a mission. They have the benefit of being revivable when they fall in battle.





^ Virtually all the major characters are introduced with their guardian spirit animal, giving this something of a flavour of *His Dark Materials* in feudal Japan.

Then there's the boss fights: not content with being fiendish bastards with more health than one bar ought to represent, they can also transport you to the Dark Realm. This new trick is an extension of the old Yōkai Realm pools which would cripple your Ki recovery, only this time around it envelops the entire room, and you can't dispel it like you previously would with a Ki pulse. That bosses tend to trigger this after you nearly have them in their death throes does sometimes feel like a step too far, difficulty-wise.

Fortunately, the enemy's other nasty surprise, a much more powerful attack distinguished by a red glowing aura, can be nullified by the new burst counter. It's one of the most compelling additions and utterly satisfying to pull off.

There are variations, depending on the guardian spirit you have equipped, but feral spirits prove to be the most useful since they let you dash into the enemy, all the better for the counter to connect. It takes a bit of time to get used to activating with R2 and circle, being one of numerous inputs you already have to juggle, from stance-switching to item shortcuts to ranged weapons. But players who felt too restricted by *Sekiro's* heavily singular approach will embrace the burst counter, as well as the bevy of other options available.

Should things get overwhelming, there's always the option of summoning up to two more players to help, though for friends the rule remains that the summoned must have already beaten said mission. If no-one's around, there's also the welcome addition of blue graves allowing you to summon benevolent AI-controlled players instead. The latter aren't the sharpest, though, so when it comes to boss fights, they're at best regarded as a kind of shield that will buy you a little time.

Mastering the punishing combat ultimately makes *Nioh 2* every bit as great and gruelling as its predecessor. It's only once you take your eyes off the combat that you'll become acutely aware of where it falls short. Despite some impressive

in-engine cutscenes bookending missions, the story is something of a shallow sojourn through Sengoku-era Japan, as embellished historical figures like Oda Nobunaga come and go with little fanfare. The level design is also rather samey – once you've seen one burned-down, ransacked hut, you've seen them all – and while you do have shortcuts, it gets predictable, especially when you reach a checkpoint then find that the next approachable gate is of course locked from the other side; Miyazaki-style world-building this is not.

Still, as long as you're prepared to put in the hours, those downsides will feel rather trivial.

With the *Souls* series laid to rest and *Sekiro* signalling how far FromSoftware has moved on, *Nioh 2's* presence is almost like eating nostalgic comfort food, if you crave that

oppressive flavour. But with its dizzying array of Yōkai to slay, a colourful abundance of loot drops, and all manner of weapons to master – and if you can play with friends – it's not just the next best thing to old-fashioned jolly co-operation, but strangely akin to playing *Monster Hunter*. 🐉



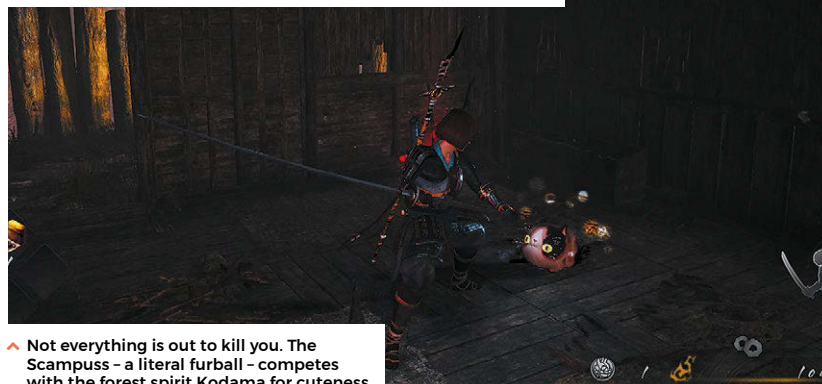
^ Bosses pulling you into the Dark Realm hardly seems like a fair fight, but it becomes more adaptable, while upgrades can help mitigate the effects.

“Masochist that I am, I'd have it no other way”

VERDICT

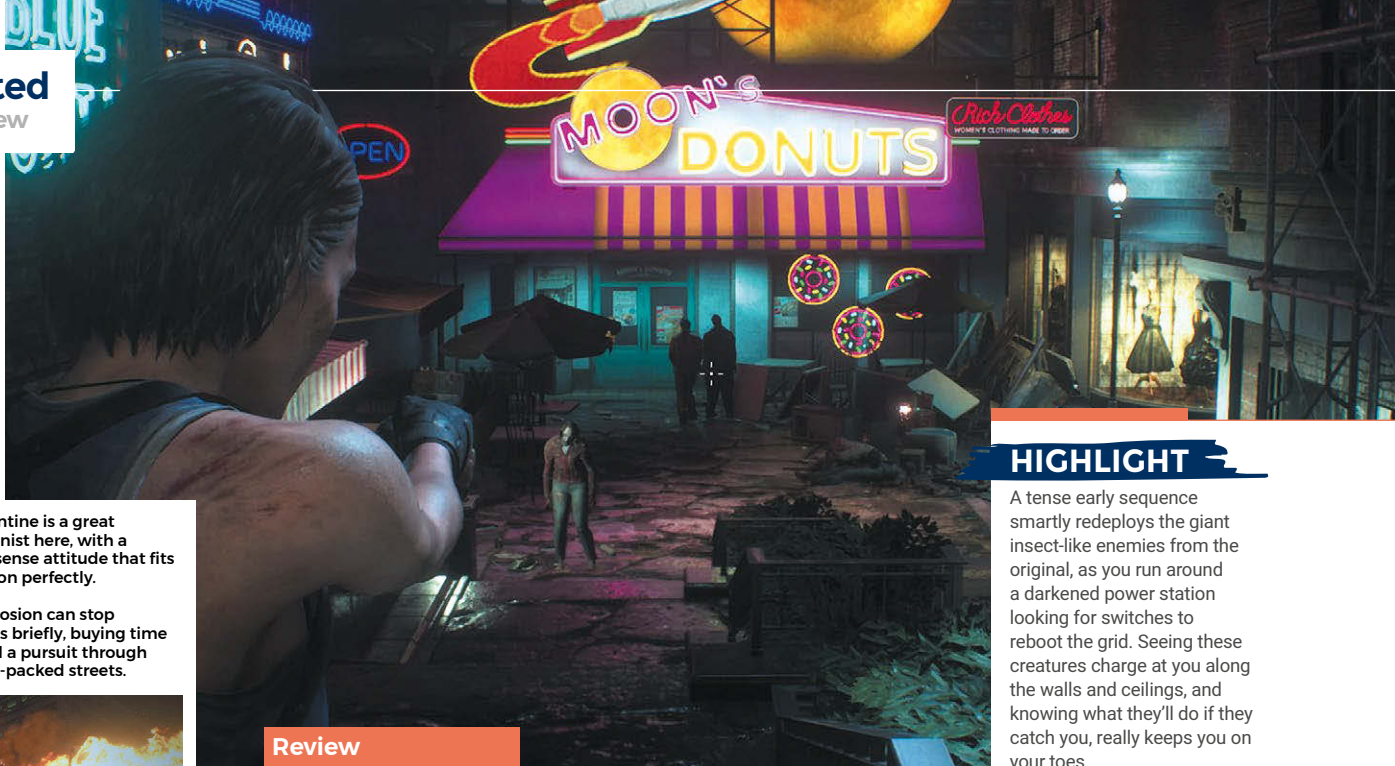
Nioh 2 may only iterate on its formula, but when it's a brutally brilliant one, you couldn't ask for more.

80%



^ Not everything is out to kill you. The Scampuss – a literal furball – competes with the forest spirit Kodama for cuteness.

Rated
Review



- > Jill Valentine is a great protagonist here, with a no-nonsense attitude that fits the action perfectly.
- > An explosion can stop Nemesis briefly, buying time to avoid a pursuit through zombie-packed streets.



Review

HIGHLIGHT

A tense early sequence smartly redeploys the giant insect-like enemies from the original, as you run around a darkened power station looking for switches to reboot the grid. Seeing these creatures charge at you along the walls and ceilings, and knowing what they'll do if they catch you, really keeps you on your toes.

Resident Evil 3

STARS in his eyes

Info

GENRE
Survival horror

FORMAT
PS4 (tested) /
PC / XBO

DEVELOPER
Capcom

PUBLISHER
Capcom

PRICE
£49.99

RELEASE
Out now

REVIEWED BY
Jon Bailes

VERDICT

A faster-paced survival horror that starts explosively but runs out of ideas.

69%

Resident Evil 3's prime antagonist, Nemesis, takes murderous stalking to new heights. He's a far more dynamic and versatile indestructible beefcake than *Resident Evil 2*'s Mr. X, and his ferocious intensity distinguishes this latest remake from its still-warm predecessor. But such energy comes at a cost. Nemesis's performance flares brightly, then quickly fades against the steady focus of last year's model.

On the surface, not much has changed. Capcom's bespoke engine again flaunts its style with impressive character details (not least Carlos's hair), chaotic scenes, and rotting inhabitants. Weapons are still satisfyingly feisty, and control still strikes a smart balance between agility, accuracy, and the artificial speed restrictions that survival horror demands. As a remake, it again plays fast and loose with the past, remixing, expanding, or redesigning the original, but always with a trail of callbacks – a chalk drawing on the floor, a zombie bursting from a cop car – to trigger déjà vu. This is remaking as respectful betrayal, which knows that surprise is as crucial to the source material as any specific event. It's also simply an expression of modern design. Downtown Raccoon City can be made to feel bigger and busier now, while cheesy B-movie monsters are benched to maintain the gritty, post-*Walking Dead* fiction.

What carries over from the old game is its momentum. Since you're not stuck in a single

location for any long stretch, there's less focus on items and puzzles, and more on running, dodging, and shooting barrels. Nemesis especially forces you to act fast, as he sprints behind you, reels you in with a tentacle, or leaps over your head to block your escape, then rocks up with a flamethrower just as you think you've got him sussed. His frequent appearances work to evoke a constant, Terminator-like presence.

But this change of pace relies on a rapid fire of setpieces and locations, and after its first act, *Resident Evil 3* runs low on ammo. Just when the city feels like it should open up, it contracts and forces you indoors. Within the police station, hospital, and obligatory secret research lab, you're funnelled through rooms and corridors that better suited last year's cautious, exploratory mode. Meanwhile, Nemesis has mutated into something too big to fit into these narrow spaces, restricting him to a handful of scripted encounters – a terrifying figure imprisoned in chunks of contrived game logic.

Resident Evil games rarely end as well as they start, but here especially, despite a bombastic finale, the second half lacks substance and lets the excitement flag. The inclusion of Resistance, a separate asynchronous multiplayer game, may be some compensation, and it's a decent distraction. But it's no substitute for a fuller core experience, with more of the tight design and choreographed scares that the series so often delivers. 🗡️

“This is remaking as respectful betrayal”



Review

Space Channel 5 VR: Kinda Funky News Flash!

Dance thrall

Tetsuya Mizuguchi must be rolling in his synaesthesia suit right now. One of his most visually striking Dreamcast classics – the dance adventure *Space Channel 5* – has made its way back to the mainstream via virtual reality. A recipe for success, you might suspect, what with the excellent handling of *Rez Infinite* and the audio-visual revolution that is *Tetris Effect*.

Well, he's not involved. But Mizuguchi, despite his talents, is not the lone auteur capable of making good rhythm games, and many of his *Space Channel 5* colleagues have led the development of this *Kinda Funky News Flash!*, brought to the plasticky portal of PSVR by Shibuya's Grounding Inc.

I hate to inform our readers that the game demands Sony's primordial PS Move controllers. If you're rightfully worried about relying on the woeful wands to get your *Space Channel* fix, don't worry – you won't be using them for long. The game's run time is a measly 35 minutes, and even then, a decent chunk of what you could refer to as 'the narrative' is made up of expository cutscenes, not in-game action. They relay that you're a rookie reporter following Ulala's lead to rescue humans from an alien scourge with the naturally journalistic power of dance. When the game eventually gets down to funky business, you speak truth to power by waving your arms in an omnidirectional fashion, depending on what orders Ulala barks your way.

The lack of content here is jarring. You might expect a big library of remixed classic tracks or a robust endless mode, but neither of these surface

once you wrap up the campaign. You can play missions again to get a high score or take on the 100-dance gauntlet, which, despite the name, is deceptively simple and clocks in at around ten minutes. *Space Channel's* challenge has been dumbed down for VR, to the point where you don't actually have to thrust your hands as part of a dance move – you can usually just hold them there, and the game will register your input as a success. This laughable approach to difficulty completely overlooks the point of the original and similar games like *Rez* – they're designed to enable the loss of self amidst the trance of dance, but this is always

an earned privilege. You would think that a VR port would help rather than hinder this issue.

Unfortunately, the open goal of a visual upgrade doesn't deliver either, the modern

style hacking away at the early 3D aesthetic of the original, where every stage looked like the backdrop to an ambitious nineties pop video, complete with pre-rendered backgrounds and glossy Y2K charm. Even the samples and low-resolution sound effects grafted from the original can't muster any meaningful nostalgia.

This all wouldn't be so contemptuous if the game didn't cost £36.99, a stunning price point for such a limited experience. At best, it feels like a lacklustre demo for the real deal. *Space Channel 5 VR* is at the very least honest in its intentions – this is a flagrant cash grab, preying on the nostalgia of a far better experience. You'd be better off dusting off the Dreamcast, because this isn't the grand return you might have been hoping for. 🗿

“You speak truth to power by waving your arms”

HIGHLIGHT

The diamond in this game's overwhelming rough are the character designs, the only 'kinda funky' thing about this soulless reboot. An optional dressing room lets you switch through a series of outfits for Ulala, and the upscaled evil Morolians appear cuddly – if only they weren't trying to enslave the populace!



▲ Ever wanted to lead a dance class with a giant purple alien? *Space Channel 5 VR* can help you realise those dreams.

Info

GENRE
Rhythm

FORMAT
PSVR (tested)

DEVELOPER
Grounding Inc.

PUBLISHER
SEGA

PRICE
£36.99

RELEASE
Out now

REVIEWED BY
Jordan Oloman

VERDICT

A nostalgia-baiting cash grab worthy of your contempt, *Space Channel 5 VR* sullies the name of a fun franchise.

25%

Info

GENRE
Life sim
FORMAT
Switch (tested)
DEVELOPER
Nintendo
PUBLISHER
Nintendo
PRICE
£49.99
RELEASE
Out now

Animal Crossing: New Horizons

Lord of the Flies, except nobody dies

REVIEWED BY
Andrew King

HIGHLIGHT

Blathers' museum is awesomely, and appropriately, cavernous – a much larger space than any other building in the game. Filling its echoey halls with realistically modelled dinosaur skeletons, its tanks with schools of anchovies, or its gardens with fluttering butterflies is a beautiful enterprise.

✓ The new pole vault makes moving around the island easier than in past entries.



In the midst of a crisis made exponentially worse by the failings of capitalism, it turns out Capitalism: Tropical Flavour is precisely what the doctor ordered.

In *Animal Crossing: New Horizons*, the latest entry in Nintendo's long-running life/indentured servitude sim, Tom Nook – the moneybags raccoon who has inspired nearly as many internet debates as *Dark Souls'* difficulty or ludonarrative dissonance – has expanded his real estate empire to an uninhabited, but resource-rich, island. As the game begins, you have joined Nook, and a few other adventurous animals in this brave new world.

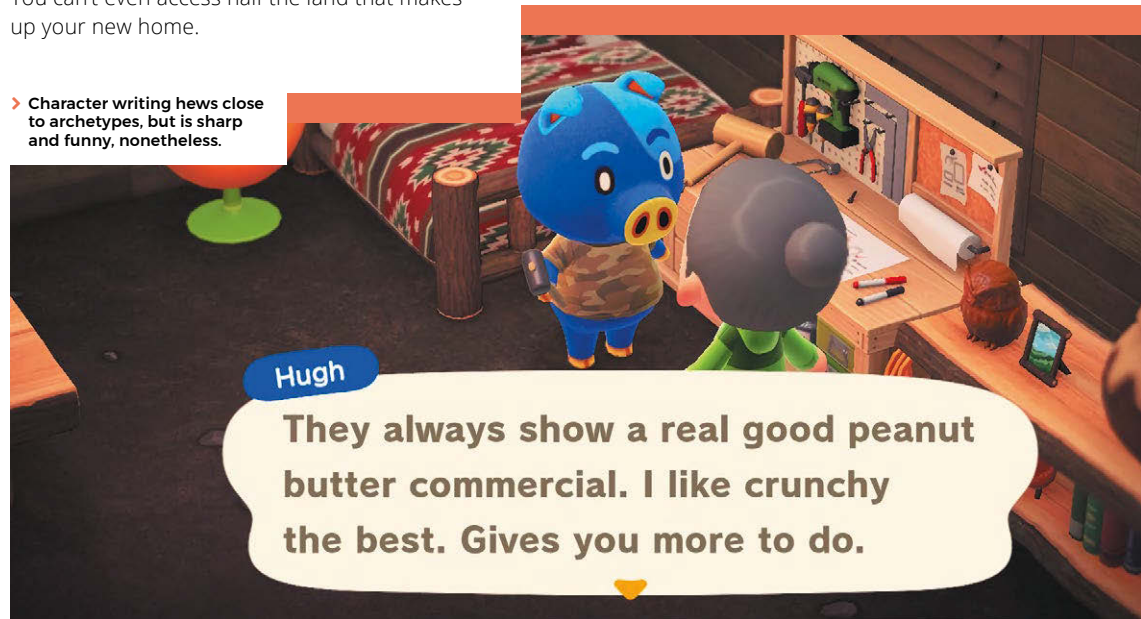
These humble beginnings mean that *New Horizons* is a much slower burn than past *Animal Crossing* titles. In the game's opening moments, rather than the customary one-room starter home, Nook offers you a tent. The rest of the island is, likewise, a bit of a fixer-upper. The general store and the visitor centre are both contained under a tarp, and the only building on the island is the airport where you arrived. You can't even access half the land that makes up your new home.

The first few days are substantially slower than anything that follows. You can hunt for bugs, catch fish, chop trees, sell stuff, work toward upgrading your home, and talk to the two other villagers that Nook brought along for the ride. But I imagine many folks will work through these sparse early hits and think, "Is this it?"

Thankfully not. *Animal Crossing's* rhythms have always been slower than those of other life sims, though, and that's still the case here. In-game time is tied to the tick-tocks of your Switch system clock, so if Nook tells you a catalogue order will take a day to arrive, it will truly take an IRL day. Working your way up from a glorified camp-ground to a bustling town will take time. Eventually, the museum, the tailor, and several new villagers will all arrive. But if you don't fiddle with the Switch clock to 'time-travel' ahead, this growth is spread out over weeks.

Fortunately, the pay-off is worth it. While each villager hews pretty close to a personality archetype, their dialogue is sharp and funny enough that I've enjoyed talking to them

› Character writing hews close to archetypes, but is sharp and funny, nonetheless.





each day. Once you earn enough credits, you can unlock tools that allow you to shape the island's terrain. The game's multiplayer functionality also means your friends can often provide inspiration if you don't know what to do next.

There's plenty to keep you entertained on the journey, too. *New Horizons* introduces a new currency, Nook Miles, which you can earn by completing tasks around the island. You'll keep track of these tasks – like watering eight plants or catching a specific fish – on your Nook-issued Nook Phone. These errands often yield materials, like wood or stones, which can then be used to craft DIY furniture for your home and, more crucially, breakable tools which you'll use to accomplish further tasks.

It's as self-conscious a treadmill as *Animal Crossing* has yet produced (aside from, maybe, the free-to-play *Pocket Camp*) and the thematic framing that Nintendo uses to contextualise the grind can take some weirdly dark turns. There are sections, for example, where you fly to (mostly) empty islands and strip away all the resources you need – sometimes while their sole inhabitant watches. Mostly, though, *New Horizons*' embrace of the endless checklist is a boon. I have some gripes about the way that Nook Miles turns every activity, even talking to your neighbours, into a transaction. But the trade-off is that as you work toward your long-term goals, you will almost always have plenty to do on the path to getting there.

There are areas where *New Horizons*' slower pace began to wear on me, though. Since release, the game has frequently been touted a path to relaxation in the midst of a global crisis. The biggest obstacle for me, in this regard, has been the irritatingly long amount of time it can take to accomplish basic things. To the end of making the island feel like a real community, many essential actions are accomplished through various agents. Want to travel to

a desert island to strip-mine its resources? First, you'll need to go to an ATM, buy a Nook Miles Ticket, take it to Orville (the dodo that runs the island's airline), click rapidly through several screens of text, select the correct option, sit through a loading screen as you wait for your plane to ferry you to your destination, then click through more text when you arrive.

The good (of which there is a lot) and the bad (of which there is a little) all stems from *Animal Crossing: New Horizons*' slow pace. Is it meditative? Is it sluggish? It probably depends on your mood at the time.

As I write this, it's been exactly one month since release. In that month, *New Horizons* has done numbers that would make Tom Nook blush, breaking Switch sales records in the UK and taking over the social media feeds of isolated players.

Since we bought the game at launch, my non-gamer wife has played every day, catching bugs, reeling in fish, and expanding her house to roughly three times the size of my hovel a few feet up our island's coast. *Animal Crossing: New Horizons* isn't perfect, but it's hard to deny that it's the perfect game for this moment. 🐼

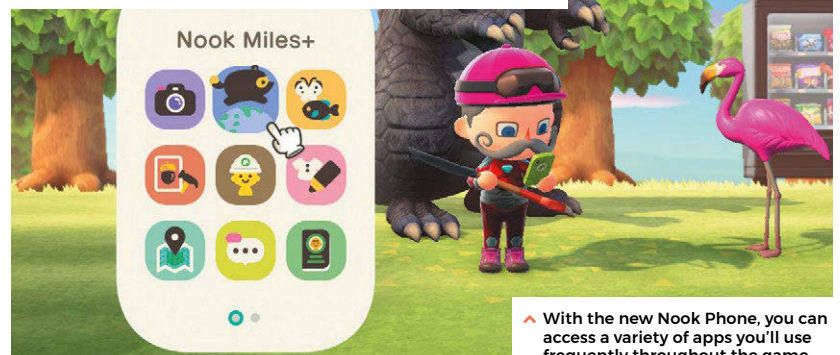
“My non-gamer wife has played every day”

^ Whenever you open a new facility, you'll have the option to hold a ceremony/photo op with your villagers to celebrate.

VERDICT

New Horizons' politics are suspect, but the consistent satisfaction of its loop, the ease with which it facilitates creativity, and the strength of its character writing will help take your mind off the strip mining.

85%



^ With the new Nook Phone, you can access a variety of apps you'll use frequently throughout the game.

Rated
Review

OM

HIGHLIGHT

It's hard to pick anything out of the mess that is *Ria* to really highlight. The character models are at least interesting, but when you're mashing a screen, trying and failing to fight an army of goblins and giant spiders, it's not really going to matter.

Review

Ria

Oh dear

Some games are all show and no substance, and it's sad to say that *Ria* fits comfortably into that category. Don't let the screenshots or the price fool you – this is a decidedly unfinished experience that needs serious work before you should even consider downloading it.

The game is a chunky action RPG that sees you playing a burly warrior with a bunch of weapons, a pendant for murder, and the ability to throw magical balls of energy around. You control everything with a clutch of buttons that clutter up the bottom of the screen – poke around a joystick to move, mash some icons to slice up anything that comes close to you. You can also swipe to move the camera – and you're going to need to because there's no lock-on feature when the fights get going. Sometimes you'll be looking in completely the wrong direction, trying to see what's hitting you, all the while incapable of doing anything because the digit you use to look is also the digit you use to attack.

There's a dodge button as well, but it never seems to take you far enough away from a scrap to actually make any difference. Every kill gives you souls, which you can spend on new equipment and spells, but they're very expensive, and it's going to take you a spectacular amount of time to get any of them. There's nothing wrong with grind when it's done right, but here it's just painful. The fights aren't entertaining; often you'll be overwhelmed within a couple of minutes of starting, and once you know you're doomed there's almost nothing you can do.

Ria looks the part, sometimes, but there's plenty in the game that feels like a placeholder asset – there are orbs floating around that act as gateways and signs, but they look like they've been lifted from a sci-fi game, not the dark fantasy world *Ria* is going for. Captions are so small they're difficult to read, and when you do manage they're often littered with spelling and grammar mistakes.

There's definitely potential in *Ria*, but right now it's a broken and clunky mess. When it needs to draw you deeper, it pushes you away; when it needs to be fun, it's just frustrating; when it needs to make you feel like a hero in the midst of an epic adventure, it makes you feel like a confused drunk stumbling between brawls you can't hope to win. It's obvious that *Ria* wants to be mentioned in the same breath as the *Dark Souls* games, but right now it'd be impossible to do that without laughing. 🗨



^ Red dots mark enemies on your HUD. They're not very smart though, and often you can escape a horde by walking around the corner.

Info

GENRE
Action / RPG

FORMAT
iOS

DEVELOPER
CreateLex

PUBLISHER
CreateLex

PRICE
£5.99

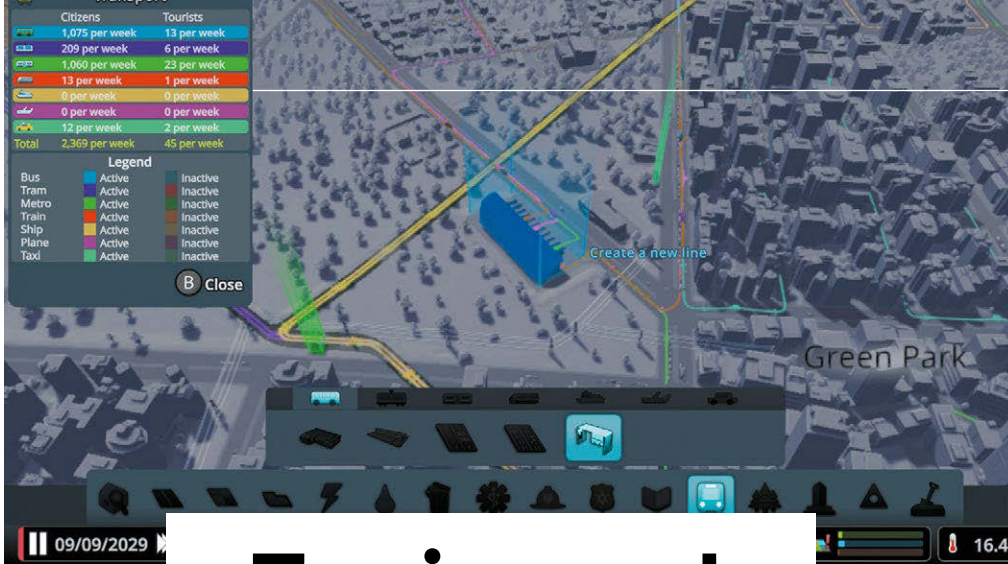
RELEASE
Out now

REVIEWED BY
Harry Slater

VERDICT

A clunky and clumsy tech demo that sometimes looks quite nice, but doesn't do much else.

27%



Zoning out

Cities: Skylines offers Ian solace through plotting bus routes

Why yes, I do have *Call of Duty: Warzone's* 100-player Battle Royale... thing... downloaded on my PS4; I have some challenges ready to be mopped up in *Doom Eternal*; my *PES 2020* Master League team needs some work to become the world-beater I know it can be. There's plenty I can be True Gaming with, playing the *pure* and *right* games that will make people know I am serious about the very serious business of playing computer games in my ever-more-fleeting free time. I've spent most of the weekend (at the time of writing, at least) planning bus routes. And train routes, and subway routes, and trying to re-route traffic so it didn't snarl up when people came into my metropolis off the freeway.

Oh, and accidentally flooding another part of that same freeway when building a hydropower dam in a place it *really shouldn't have gone*.

Yep, it's *Cities: Skylines* – Colossal Order's fantastic *SimCity*-alike that grabbed the baton from Maxis' once-great series and ran like hell with it. This is another one of those situations

where, were I to don my reviewer's beret, I would be sneering cynically at elements of the game. For one, I'm playing it on Switch, which regularly chugs worse than that aforementioned freeway choke point. But it's bearable, just about.

For another, you're just sort of *playing* it, without any real goals beyond expansion in mind. It's fun to adopt the mentality of the British Empire at times, sure, but with no real breakdown of goals beyond that endless stretching, it does feel a bit aimless after a while. But the beret of True Opinion is in its protective case; this is just me, playing a game, because the world's gone a bit wonky and I want to exercise some control over things.

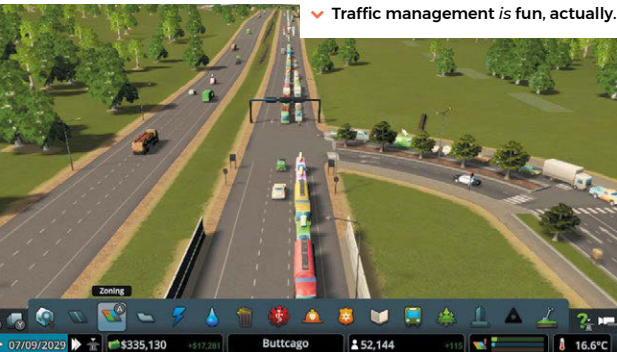
I still didn't expect that need for control to manifest itself in the form of bus routes, though.

The complexity of *Skylines'* simulation isn't so much that it's hard to play in any way, but there is depth to things beyond what you might initially expect – more so if you really want to run an efficient ship. City. Whatever. Public transport is required to get your people around the mini world you've created, and bus (and train, tram, etc) routes have to be plotted out to get commuters from home to work and back again. It's not hard to do. It's hard to get right. I still haven't got it right. But the relaxation felt through the simple act of floating through a city of my own creation – one that's organically grown into districts before my very eyes, one that I almost ruined by turning the entire downtown financial sector into a theatre district – and plotting out a route for number 34 is just *lovely*.

Admittedly, it's backed up by some serious frustration about how bad I am at planning roads, intersections, and placement of freeway off-ramps. But what can you do? 🤖

“A city I almost ruined by turning the downtown financial sector into a theatre district”

▼ Traffic management is fun, actually.



Wireframe Recommends



TheoTown

MOBILE, PC

A city builder that was so good on mobile it bagged itself a port to PC. It's no *Cities: Skylines*, but it's great fun in its own right – and well-suited to mobile.



RimWorld

PC

More survival than city builder, *RimWorld* nonetheless bags a place in this list of recommended-alikes in one part because it's brilliant, and another part to remind me to give it a proper entry on these pages.



SimCity 2000

PC, SNES, OTHERS

Yeah well, a *SimCity* has to be included, so why not the one that isn't necessarily the best, but has the fondest memories attached to it? It's limited by modern standards, but still great fun.

Backwards compatible

WRITTEN BY RYAN LAMBIE

Back to the 1980s: it's the PC Engine Mini

The PC Engine Mini may just be the most unusual nostalgia box yet released. Most of the systems released so far – including 2016's NES Classic Edition, which kicked off a whole wave of miniature consoles – capitalise on our affection for games we played decades ago. For a generation of gamers in the west, however, the PC Engine, or TurboGrafx-16, was a largely unknown quantity: it was never officially released in the UK, and only glimmered briefly in the US before Sega and Nintendo's rival consoles knocked it out of contention in the nineties.

What Konami's unexpected but welcome PC Engine Mini does, though, is put one of Japan's most fascinating consoles – and its great library of games – in front of a new audience. And judged by the standards of other mini plug-and-play devices, it does this quite nicely.

Like the NES Classic and Mega Drive Mini before it, the PC Engine Mini (or to use its full name in Europe, the PC Engine CoreGrafx Mini) connects to a modern TV through the HDMI socket, is powered via USB, and offers a selection of games that illustrate the best the console had to offer at the height of its popularity in eighties and early nineties Japan.

Although only slightly smaller than a vintage PC Engine (which was a tiny console to begin with), the Mini has some charming details, right down to the green power switch and its tiny plastic tab that, on the original console, prevented the user from removing a cartridge with the system turned on. Tactility is a big part of the appeal of these systems, and the PC Engine Mini certainly has this: the controller also looks and feels close to

the original, though it does feel a little lighter by comparison.

Another part of the system's appeal is its user interface. The PC Engine Mini's UI is by M2, the developer that handled the superb Sega Mega Drive Mini; if anything, this one's even better, with the selection of games divided up into separate libraries. You'll find US releases like *JJ. & Jeff* and *Splatterhouse* in the TurboGrafx-16 list, while such games as *Star Parodier* and *Fantasy Zone* are housed under the PC Engine roster, each with their own cover art, backgrounds, and theme tunes. Fire up one of the games originally released on CD, and you'll be treated to a charming animation of the disc spinning in the drive and an accompanying sound effect. It's touches like these that really set the Mini apart from, say, playing ROMs through a typical emulator.



▲ Japan and the US get their own incarnations of the PC Engine, while Europe gets a replica of the grey CoreGrafx, originally released in 1989.





◀ Taito's *Parasol Stars* was one of the best games on the PC Engine, so it's pleasing to see it on the Mini.

What about the games?

The PC Engine is expensive to collect for, so it's pleasing to see rare titles like *Sapphire*, *Parasol Stars*, and *Castlevania: Rondo of Blood* among the line-up of 50-or-so games. There's a huge array of shooters in the PC Engine's library, and that's reflected in the Mini with *R-Type*, *Super Star Soldier*, *Gradius*, *Galaga '88*, and more, but other genres are well-represented, too. You'll find the ace *Bonk* platformers; *Ys Book I & II* are sterling

RPGs; *Ninja Spirit* and *Ninja Gaiden* offer bracing side-scrolling action; the multiplayer anarchy of *Bomberman '93* and '94 make getting hold of some extra USB controllers a must. PC Engine fans will point at gaps in the Mini's selection, but it's difficult to argue with most of the choices here. My big gripe? Including Hideo Kojima's cult cyberpunk adventure *Snatcher*, but only in its Japanese form, feels like a big missed opportunity. Bah.



◀ The notoriously expensive *Ginga Fukei Densetsu Sapphire* is one of several rare games revived for the Mini.

Availability

Here's where things get just a little bit complicated. The PC Engine Mini is being sold exclusively through Amazon, and while the Japanese arm of the firm's fulfilled pre-orders in the Far East, the ongoing coronavirus pandemic has led to "indefinite" delays elsewhere. We were lent a PC Engine Mini unit to review by a PR company, but to date, there hasn't been an update as to when customers who've placed pre-orders will actually receive their device. So far, the most we've managed to glean is this statement from Konami: "Amazon is still accepting new pre-orders for the consoles, and we're working hard to resume normal delivery schedules as soon as possible. We thank everyone for their patience during this uncertain time."



▼ Including *Snatcher*, but without an English translation, is one of the Mini's few disappointing oversights.

Worth a purchase, then?

There are many ways to play PC Engine games in 2020. Purists might favour sticking to original hardware, but actually getting your hands on a proper console and a decent selection of games will cost a considerable amount these days – and then there's the task of marrying it all up to a modern television. Running a PC Engine core through an FPGA board and MiSTer, meanwhile, will provide more accurate results than an emulation-

based device like the PC Engine Mini, but even this will set you back more than the Mini's asking price of £99.99. The Mini, then, is by far the easiest and most affordable route, and while there are some flaws – a bit of sound and input lag here and there, plus the lack of display options that is strangely common to all these devices – this reincarnation of the PC Engine represents a package that's hard to resist.



F-Zero

More than seven games used Mode 7, and it was always mind-blowing

NINTENDO EAD / 1990 / SNES

You need to launch a console with something that makes jaws drop and chins wag – few have done a better job of that than *F-Zero*, which launched with the SNES in Japan in 1990 (and the US in 1991, and Europe in 1992). How?

Why, through the miracle of pseudo-3D effects carried out by rotating and scaling background layers, using one of the console's built-in hardware graphics settings, of course. Or Mode 7, if you want to shorten it a bit.

Yes, it was a marketing term thrown about to get us all excited about this so-called 'Super' Nintendo machine, but the fact is it was a graphics mode in the SNES architecture, numbered seven. So there was truth to the marketing gumph. Additionally, it was a) obvious when Mode 7 was being used, so easy to show off, and b) really quite brilliant at the time. It's impossible to say it was purely the utilisation of Mode 7 that made *F-Zero* so special and, in turn, was the sole factor behind the revitalisation of the racing genre – but at the same time it's not a stretch to make the claim.

By rotating and scaling an otherwise static, non-interactive background image, *F-Zero* gave the impression of vast, three-dimensional courses big enough to fit these vehicles travelling at hundreds of miles an hour. The sense of speed was palpable; the trick *worked*. Nobody with a technical mind truly believed what they were seeing was 3D, but it genuinely didn't matter. And when we fell to the ever-growing earth in

Pilotwings, or skidded around a *Mario Kart* track, the belief in such a natty feature just grew.

Over 60 games in the SNES library ended up using Mode 7 – the Game Boy Advance even introduced its own version of the effect when it launched – so this was by no means just a promotional tool, or an overhyped tech spec. Mode 7 made a lot of SNES games *look* like SNES games, and so had a huge hand in defining the visual character of an entire console.

The Mega Drive never had Mode 7-alike features natively, though the odd title did pop up here and there that managed an impersonation of the rotate-and-scale backgrounds. *Dick Vitale's "Awesome Baby!" College Hoops* – yes, that is what it's called – is a good example to take a look at. But as all these solutions had to be

entirely written in software, it was a laborious, difficult, and ultimately pointless task. Better to spend time elsewhere.

And that, looping back handily, is what helped to make Mode 7 on the SNES stand out so very much. It was easy to implement, being a standard hardware feature, meaning developers could either use it in a standard fashion quickly and easily, or spend a bit more time and mess about to get some more special effects. A particular favourite has to be *Super Castlevania IV's* rotating level, which not only took advantage of the visual aspect of the tech, but made it an integral part of that particular stage's mechanics. No, it wasn't really 3D, but Mode 7 was a brilliant, memorable, and striking feature all of its own. ☺

“F-Zero gave the impression of vast, three-dimensional courses”

Next Issue

ON SALE 4 JUNE

Wireframe GOES MONTHLY

The new 116-page edition of your favourite games magazine: bigger, better, more feature-packed than ever

Don't miss out!
Subscribe
pg 54

Also

- ▶ Sumo Digital on the arcade spirit of Hotshot Racing
- ▶ The strange history of movie-to-game adaptations
- ▶ An inside look at the Italian video games industry
- ▶ Make your own multiplayer space shooter in Unity

Editorial

Editor

Ryan Lambie

Email ryan.lambie@raspberrypi.com

Features Editor

Ian Dransfield

Email ian.dransfield@raspberrypi.com

Sub-Editors

David Higgs & Vel Ilic

Design

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Harriet Knight, Sam Ribbits

Contributors

Jon Bailes, Mac Bowley, Simon Brew, Anthony De Fault, Konstantinos Dimopoulous, Andrew King, Stuart Maine, Sean Martin, Steve McNeil, Jess Morrisette, Jordan Oloman, Harry Slater, Mark Vanstone, Alan Wen, Steven T Wright

Publishing

Publishing Director

Russell Barnes

Email russell@raspberrypi.com

Director of Communications

Liz Upton

CEO

Eben Upton

Advertising

Commercial Manager

Charlie Milligan

Email charlotte.milligan@raspberrypi.com

Tel +44 (0)7725 368887

Distribution

Seymour Distribution Ltd

2 East Poultry Ave, London EC1A 9PT

Tel +44 (0)207 429 4000

Subscriptions

Unit 6, The Enterprise Centre, Kelvin Lane,

Manor Royal, Crawley, West Sussex, RH10 9PE

To subscribe

Call 01293 312192 or visit wfmag.cc/subscribe

Subscription queries

wireframe@subscriptionhelpline.co.uk



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

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





CQ32G1
2560X1440

144HZ

FreeSync

1
MS

Curved

QHD

BEND YOUR REALITY

Available at:



ebuyer.com

amazon

aocgaming.com



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



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



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