

£3
ALL FORMATS

Wireframe

LIFTING THE LID ON VIDEO GAMES



Issue 37 £3
wfmag.cc



LIBERATED

UNMASKING ATOMIC WOLF'S DYSTOPIAN ACTION ADVENTURE

PRINCE OF PERSIA

Jordan Mechner on
crafting a classic

DUNGEON MAKER

Code a procedural
level generator

FALLOUT-LIKES

The return of the
isometric RPG

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 problem with PEGI

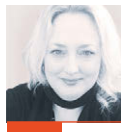
There are a lot of things you don't notice before you have kids. Things you took for granted. Low TV cabinets are one (why, why, WHY do we put our expensive electronics right in little people's sight-lines?). Daylight savings and weekends are another (no, kids don't care what time it is. Ever. Not even on holiday).

PEGI ratings? They fall into the 'didn't notice' column. Once listed alongside – and then the successor to – information from the British Board of Film Classification (BBFC) on video games, I was aware of them in the same way we all were when we were 13 and stuffing bras to pass as 15 at the local cinema. By the time they were enforced, I was old enough to buy grown-up titles, anyway. Before switching careers, I sincerely don't believe I ever thought about them. Not once.

This changes when you're a parent, though. My empirical experience intimates that parents that grew up gaming are the strictest (my 13-year-old son's pals might have been playing *GTA 5*, but I can guarantee you that my boy certainly wasn't... not in my house, anyway), but for those who grew up without video games, the Pan European Game Information board (PEGI) is all they have to help them make educated, informed choices on what titles are best for their children.

The issue isn't that PEGI's ratings aren't clear – they are, and my quick and dirty, non-scientific survey of friends and family clearly indicates that they are – but while the similar system employed by the BBFC takes account of both content and context when rating movies, as well as issues like divorce or bereavement, PEGI only rates on the former. But with one classification system commonly conflated with the other, some parents mistakenly believe PEGI's judgement implies the age at which a child can play and understand a game, not just its suitability.

Take *Animal Crossing: New Horizons*. Terrifying Bunny Day ambassador Zippy aside, the latest *Animal Crossing*



VIKKI BLAKE

It took 15 years of civil service monotony for Vikki to crack and quit. She's now an experienced reporter and critic working for the likes of Eurogamer, MCV/DEVELOP, MTV, and Variety.

instalment is a fresh, friendly island romp, forged on building and maintaining friendships and exchanging the local flora and fauna for cold, hard cash.

Animal Crossing: New Horizons is PEGI 3. This suggests at a glance that children from as young as three may watch, or play, the island adventure without fear or psychological scarring (I feel compelled to mention exception Zippy again here. Seriously: he's terrifying).

The issue, however, is that age ratings don't necessarily equate to difficulty or understanding. So while a four-year-old might be able to watch and enjoy the pretty colours and relaxing music of *New Horizons*, they certainly won't be able to comprehend the reams of text, the complex local economy (although to be fair, I don't think many of us understand Crook Nook's economics, either; why the hell am I paying for your bridges and inclines, Tom? Riddle me that, eh?), and the often subtle nuances of good neighbourly relations.

And yet a parent who grew up outside of gaming would be forgiven for flipping the game over in the supermarket, spotting the PEGI 3 rating, and presuming it's suitable for their six-year-old.

Of course, all children are different. Of course, parents should be encouraged to do due diligence and carefully research the suitability of games before they buy them, or – better yet – play alongside their kids. But this is so much harder in practice, especially if your toddler is throwing a wobbler in the middle of Tesco.

Presenting a judgement forged merely on suitability, not difficulty – however well explained on the PEGI website – endangers not just PEGI's reputation, but that of the entire industry. It jeopardises the necessary work done to convey the importance of protecting young people from frightening or mature content, and the importance of game ratings in the first place. PEGI needs to modernise its system and make its classifications meaningful, or parents and children alike may learn to distrust – or even ignore – them entirely. 🗣️

#37



Contents



Attract mode

Interface

06. Liberated

Fighting Big Brother in Atomic Wolf's dystopian action game

10. ScourgeBringer

Hands-on with a razor-sharp rogue-lite platformer

12. Lord Winklebottom

Upper crust animals star in a 1920s murder mystery adventure

16. Incoming

Football, fighting, and frightening futures

18. Prince of Persia

Jordan Mechner on the making of a pioneering classic

24. Headspun

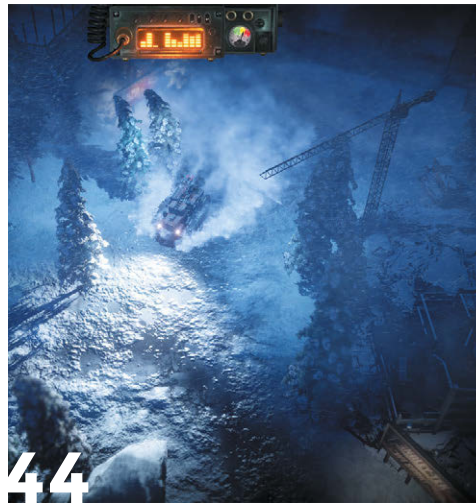
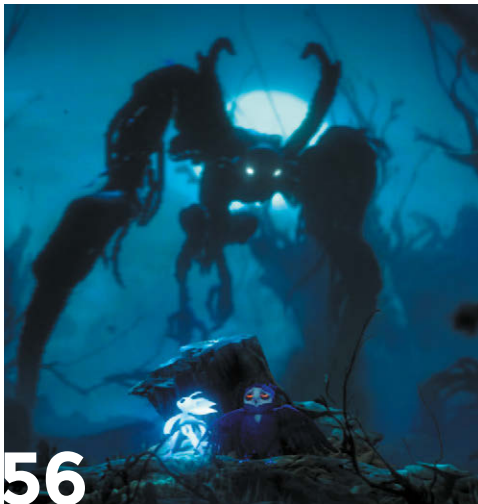
A solo developer's sci-fi trip inside the human mind

44. Fallout-likes

The ongoing brilliance of isometric, post-collapse RPGs

50. Sega Game Gear

Revisiting an under-appreciated nineties handheld



Toolbox

- 28. Design Principles**
Why game development needs a special type of programmer
- 30. CityCraft**
How urban planning can help with your game design
- 32. Dungeon Generator**
Make your own procedurally generated levels
- 40. Source Code**
Code a modern take on Atari's Lunar Lander coin-op

Rated

- 56. Will of the Wisps**
Moon Studios' Ori sequel doesn't disappoint
- 58. Pilgrims**
An adventure game for the mobile masses
- 59. Corruption 2029**
Turn-based tactics in a dark and doomy future
- 64. Backwards compatible**
The dusty world of retro games and tech

WELCOME

Tabloid papers have long had a hate-hate relationship with games. I can still remember TV presenter Anne Diamond appearing in a 2008 edition of the Daily Mail, posing for a photo while clutching copies of *Dead Rising* and *Halo 3*. "This game shouldn't be allowed to be sold, even to adults," she wrote disapprovingly of *Resident Evil 4*. "It wallows in violence for violence's sake." Her argument was that her kids shouldn't be playing such violent games, which is true – those games are clearly labelled with huge 15 and 18 certificate warnings for a reason. That she couldn't figure out how violent they were just by looking at the back cover seemed to escape her.

More recently, the same outlet – and others like it – seized on a new piece of 'research' (I use the term advisedly) that purports to show "what avid gamers could look like in 20 years." Spend too long playing games, the study suggests, and you'll end up like Michael – a ghoulish, computer-generated man with pale skin, arthritic and blistered hands from repeatedly prodding away at a controller, and dreadful posture from sitting in a chair all day.

The way I see it, there are at least two problems with this study. One: most people probably realise that doing the same indoors-y thing for hours each day (watching telly, knitting, and yes, playing games) isn't ideal for their long-term health if they don't also get out and exercise, so it's hardly a groundbreaking discovery. Two: the outfit behind this research? OnlineCasino.ca. Thankfully, nothing bad has ever come from excessive online gambling.

Ryan Lambie
Editor



FIGHT THE POWER

Reading comics and smashing the state in Atomic Wolf's tale of revolution and terrorism, *Liberated*

Maybe one day, *Liberated* will be looked back on as a portent of things to come: a surveillance state; government snooping emboldened by an always-online society; a slip, a tumble, a fall into authoritarianism. Maybe. Or maybe it'll just be an exceptionally pretty action-adventure game set in a comic book world (quite literally), telling multiple stories of multiple people in this near-future dystopia, from multiple perspectives – and all that entails.

See, *Liberated* isn't just about one righteous person's quest to bring about truth, justice, and the [insert country here] way – it's a tale meant to discuss and highlight, to prod and question, rather than ram a viewpoint down any particular throat. From a narrative perspective, *Liberated* brings to mind not the action comics of superheroes, but the darker, grittier world of the graphic novel. Helpfully, it looks the part too, with the game taking place inside the panels of a comic book, scrolling and moving around as you move from area to area. It is, in a word, 'stylish'. To add another word, 'cool'.

It's also a game in a more traditional sense; while leaning heavily on its narrative and offering the player chances to influence matters with the conversation

choices they opt for – at its core, this is a side-scrolling shooter/platformer. Think *My Friend Pedro*, but monochromatic, with fewer bananas, and with more conspiratorial aspects. It's a linear game, with players pootling along and getting into gunfights with their enemies (whoever they might be), but with elements like stealth, quick-time events, and puzzles thrown in to keep things fresh and mix it up a bit.

The whole thing is brought to life through Unity, using the Playable Graphic Novel (PGN) framework developed by

L.inc, giving *Liberated* its distinctive look. Behind the scenes, it's a tool that allows developers and artists to blend their talents more easily

and results in something that isn't just a game that looks like a comic book, or a comic book-alike that has bits of game in it. Everything is closer, more tied together, and it *feels* more of a whole as a result.

And that's a big help, because if it were more jarring than it is, *Liberated* might lose your attention – and attention is what it needs to keep that thick atmosphere suitably... well, thick. Whether you're battling against Themis – the once-benevolent social credit system turned abused tool of invasive observation – or approaching things from another perspective in one of *Liberated*'s other

"IT IS, IN A WORD,
'STYLISH'. IN ANOTHER
WORD, 'COOL'"

'issues' (in-game chapters being divided into different issues of the comic and all), there's a standout vibe to the whole thing.

We chatted with Robert Purzycki, producer on *Liberated* and CEO of L.inc, and Konrad Walkuski, community manager for *Liberated*, to find out more of the secrets behind the game: the learning, the support, and the clever puns sequestered as marketing terms.

I think it's safe to say this is Atomic Wolf's most ambitious project to date – how have you approached such a big undertaking?

Ha! It is, and by a mile! We've got smaller indie games out there between our team (like *NecroWorm*, *Mad Age & This Guy*) but those were basically tiny productions. Just in terms of development time and the number of people involved, it has been a huge adjustment. We've been at it for over two years now, and there are real highs and lows. And it is a learning process – we do our best to support each other and get better as we go.

How has the process been, making something so ambitious?

It's been a long, long process. It all started with just figuring out what a PGN should be. We've played *Comix Zone*, *Max Payne*, *XIII* – all games that were inspired by comics – but we wanted to create something more than a video game: a Digital Comic 2.0.

Our goal was to make you always feel like you're looking at a comic book. So that means we had to really get the art of graphic novels to even approach writing one. And then how do you transition that over to digital – we had to research how people read comics, [including] simulating teeny-tiny details like how your eye moves across the pages.

Making comics is hard. Making games is hard. We had to start over from scratch at least twice, but we've learned so much about the nuance of both art forms.

What makes *Liberated* a "revolutionary playable comic"?

I like this question, because that line works on two levels and we all thought it was just the wittiest thing. First of all – it is literally a story about revolution. What does it take to push people over the edge? We'll ask some tough questions about right and wrong, whether ends justify means, sacrifice for the greater good, and we'll show you both sides of the story. True to life, it's a complex issue with no simple answers.

And on the other hand, we want *Liberated* to be the start of something new in comics and to revolutionise the genre. Sure, we already have digitised comic books, and action and adventure games, but no-one's really put them all together until now – not like this, at least. *Liberated* is first and foremost a graphic novel that allows you to jump in and be part of the action. We hope that in the end, we'll have shown that comic book creators and game developers can start doing great new things together. ➔

What influences have you looked to outside of just gaming?

The world of *Liberated* is kind of like holding up a mirror – it's about where we are now and where we could easily be headed. So, of course, it's a fictional, dystopian story set in the near future, but the individual bits and pieces are pulled right from the world around us. The technology used to create this Big Brother state is modern, the reasoning and processes are things that we're seeing right now all over the world.

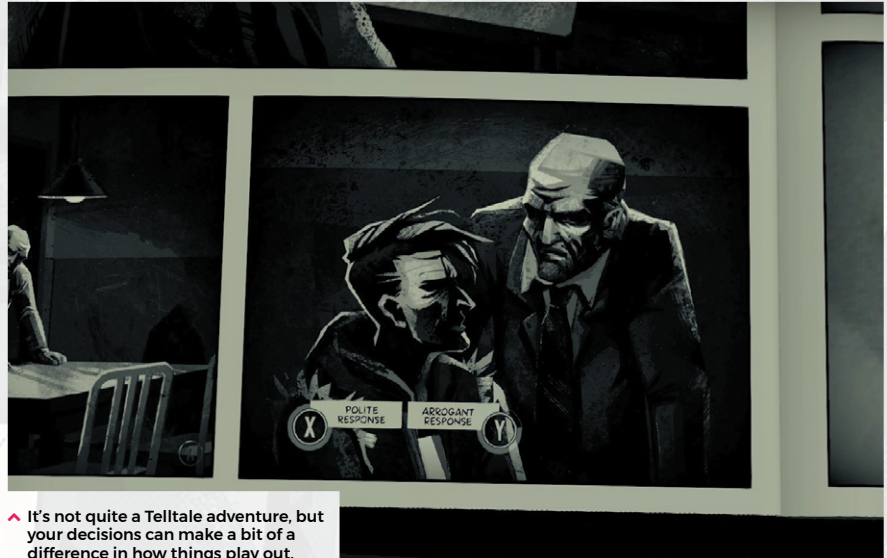
It's funny, because it really makes you see that we're basically already living in a cyberpunk world. Eighties cyberpunk writers had it right in so many ways... except for the flying cars and cybernetic upgrades, but we did get WiFi and Twitter instead.

Anyway, *Liberated* might be all too real, but photorealistic it is not. So the art style is just as important, and here we owe so much to classic noir graphic novels including *Sin City*, *V For Vendetta*, and even one-off experiments like *Batman Black and White*.

Altogether, we like to think about *Liberated* as too-close-for-comfort-cyberpunk-noir.

And, of course, what influences inside gaming have you looked to?

It's not by sheer accident that folks compare *Liberated* to *Comix Zone*, *INSIDE*,



▲ It's not quite a Telltale adventure, but your decisions can make a bit of a difference in how things play out.

and *Shadow Complex*. A few folks also compared it to *My Friend Pedro* recently, but I guess that one was by accident.

What is Liberated, the group in the game, fighting for? Or against?

In a nutshell, the Liberated are this revolutionary movement fighting against an authoritarian state. They believe in all that good stuff: transparent government and democracy. They also unquestioningly believe in a massive state conspiracy, and are carrying out a bloody fight at any cost to prove it. I won't spoil the story details, but they are, in no uncertain terms, extremists who believe their fight is just.

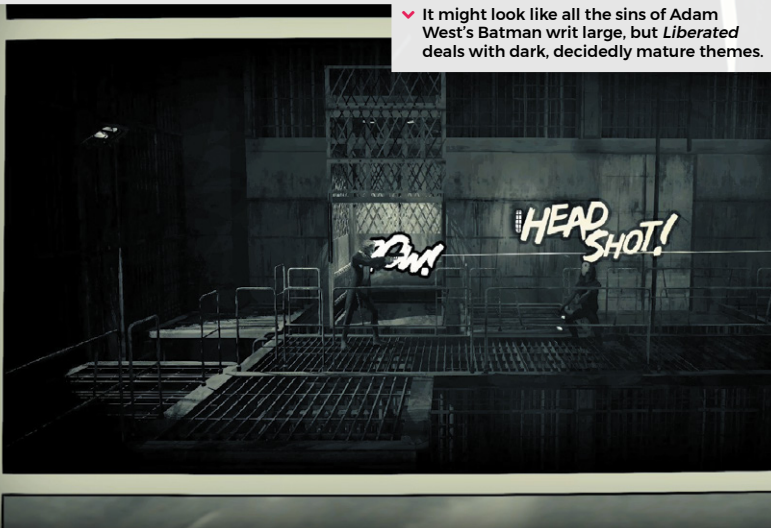
How much of the real world bleeds into this dystopian fantasy? Is it easier to write given the world is... as it is?

It feels like the longer we work on the game, the closer it hits to home. Years ago, when we were just starting to think about the world of *Liberated*, the extent of our game's Citizen Credit System was basically fiction. Well, not anymore!

What's the distinction between the different character chapters?

We're doing four chapters on release – each one will be an issue for your lovely comic book collection, and together they'll tell the whole story. We want to show you

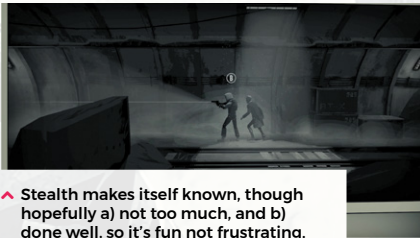
▼ It might look like all the sins of Adam West's Batman writ large, but *Liberated* deals with dark, decidedly mature themes.



▲ Where would we be without the odd puzzle thrown in? Nowhere, that's where.



^ Your choices and moves all play out in true comic book fashion and remain there on the page for all to see.



^ Stealth makes itself known, though hopefully a) not too much, and b) done well, so it's fun not frustrating.

it with the setting, themes, art, and story... together, it's what we hope makes *Liberated* something a bit unique.

What's the engine behind *Liberated*?

Liberated runs on Unity, but as we go we're developing our PGN framework. This is what drives the simulation of the comic book. It lets us insert gameplay into the comic, and brings it all together seamlessly. It was a really big part of our

different characters' points of view – so you'll join the revolution, and you'll take the fight right back to them. We're also playing with different ratios of story-to-gameplay between each issue, just figuring out the right balance, and we hope players/readers will let us know what works best.

Away from the story and setting, who are you aiming *Liberated* at, player-wise?

Action gameplay is a big part of *Liberated*, so we always make sure to say that it's an action-adventure game and we do want a rewarding challenge in terms of combat as well as puzzles. At the same time, we're developing a much more forgiving 'reader mode'. Basically, we want *Liberated* to be accessible and enjoyable for both core gamers and comic book fans who enjoy gaming from time to time.

How much does the story tie in with the game? Or are we looking at more two separate-but-equal elements?

Liberated is a story-driven game, the two are inseparable. Of course the core gameplay loop centres around action, gunplay, stealth, and puzzles – but you've seen all that before. It's when you combine

"IT FEELS LIKE THE LONGER WE WORK ON THE GAME, THE CLOSER IT HITS TO HOME"

development process, and I think we have only scratched the surface. At the end of the day, we want the PGN framework to be the foundation for future playable comics.

How big is your team? How has working with that many people been?

Oh gosh, altogether we're at... nearly 30 people? It's really a collaborative effort. Over half of us are with the Atomic Wolf team (the game's artists, programmers, designers, and story writers) and L.inc (where the whole project and PGN framework got started), then there's seven of us at Walkabout Games on the



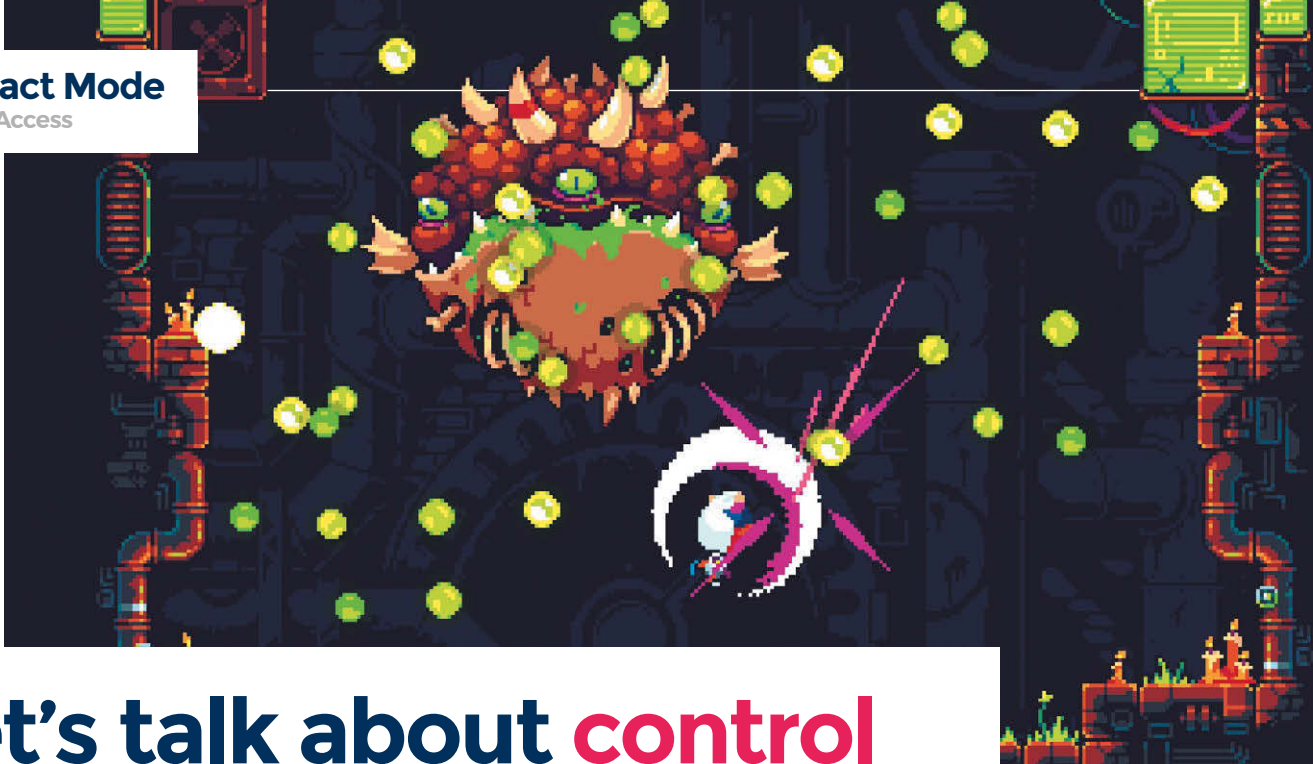
publishing and marketing side of things, and we've collaborated with two very talented writers in the process as well. It's all very new for us and can be a challenge to keep all the pieces running smoothly, but we're all a team so just doing our best to support each other.

Finally, what are your hopes for *Liberated* once it releases?

In a lot of ways, *Liberated* is our proof of concept for the future of comics. We hope that it does well enough to rock the boat a little. We want to do more *Liberated*, of course – there's so much potential here to tell meaningful stories, and bring on new perspectives and voices. It'd also be amazing to work on other comics that are already out there, and take those works to a whole new level.

We're doing our thing with *Liberated*, but just in terms of gameplay and interactivity, the sky's the limit with what's possible in Unity. More *Liberated* stories and other IPs, collaborating with comic book writers and artists from around the world. That's the dream! 🤖

Liberated releases May 2020 on PC and Switch



Let's talk about control

Every aspect is handled with the personal touch in ScourgeBringer

Info

GENRE
Rogue-lite platformer

FORMAT
PC / Switch

DEVELOPER
Flying Oak Games, E-Studio

PUBLISHER
Dear Villagers, Yooreka Studio

RELEASE
Out now (Early Access)

▼ Boss fights are, as you'd expect, a case of learning patterns and responding to them. And maybe getting lucky sometimes.

It's all about control. Not the Remedy game. The most important element in *ScourgeBringer* is making sure you, the player, are in full control. It's a part of the project that its developers have focused on with some intensity, and from early flirtations with the platformer both pre- and post-Early Access, it's clear that things are paying off. Your character runs when you tell them to, jumps when you press the button, attacks and dashes as soon as your thumb depresses: it works phenomenally well already, and it's not even done.

But what is *ScourgeBringer*? Why, it's a rogue-lite combat-platformer with a focus on always attacking, always moving, and not really defending. In short, it's a game that demands a tightness of control, and it's a game that provides just that. How? "A lot of playtesting,

and a lot of prototyping," explain Thomas Altenburger and Florian Hurtaut, the game's designers and, respectively, coder and pixel artist. "We prototype and test a lot of things. We spent a year figuring out the 'game feel'. The core discipline we have is to work on very small chunks at a time, and in an iterative way.

"For instance, we break everything down into atomic-level designs, like asking how it is to just jump around. We then design prototypes to test the jumping system, over and over until it feels satisfying. We do that for every system, and at some point we zoom out and start assembling the smaller parts, checking if they still work together, eventually discarding some of them or adapting them. And so on until we zoomed out enough that it feels like a game."

It's that focus on working and reworking elements until they get it just right that helps *ScourgeBringer* stand out in an increasingly crowded field. Roguelikes and rogue-lites aren't exactly slim pickings these days – and that's something that did play into the devs' minds. "We were wondering about that when starting the project," they explain, "and it was quite clear to us we were not going to stand out on innovation, so we had to push the quality level to the max. That's why we spent an entire year just tweaking the game feel. We had to get things right and compensate with a rock-solid feel."

While admissions throughout there are no illusions *ScourgeBringer* is a revolutionary game, Altenburger and Hurtaut maintain throughout our chat that they want to make something of high quality – "a sort of quintessential experience of the genre." Early reactions are





◀ Moments of respite are few and welcome, offering the chance to take stock and prepare to try again.

proving the approach is working, with plenty of positive user reviews following the game's Early Access release (and a massive early thumbs-up from this writer).

With *ScourgeBringer* changing form many times early on – from a pulp sci-fi romp through a platforming iteration starring a space luchador – there is always the risk things could be confused. But again, it's about control: Flying Oak settled on the game's final design after about a month of pre-production and has kept control over that vision since.

Building a custom engine and tools on top of MonoGame and FMOD was "the most comfortable and efficient" way to do things from there, and... really, it just shows. *ScourgeBringer* looks like it was made by a team comfortable in what it was doing, by one focused on making something as well-honed as it could possibly be within its sub-genre limits. It's not reinventing the wheel, it's just trying to make it as perfectly round as it can.

Part of the process involves learning from the past, and Flying Oak has a lot to take away from past project *NeuroVoider* – a well-received twin-stick shooter/RPG released in 2016. "We learned a lot from *NeuroVoider*," the devs tell us, "starting with what didn't work and shouldn't be reproduced." Trying to make everything procedurally generated was the first big tick (or cross) on that particular list – again, it's about maintaining control. If boss fights are randomly generated, as they were in the last game, they're never going to be that good, or that interesting. "Boss fights kind of all looked the same, because there are human limits as to what you can actually put into an algorithm," they explain.

"So we toned down a lot our usage of procedural generation," they continue. "We are

still using some, but it [has] become more like a tool within our tools to generate ideas; in the end, we're handcrafting everything now. There's only the [layout of the rooms] that is generated, otherwise, all rooms are handcrafted – and we made the bosses to be unique and memorable."

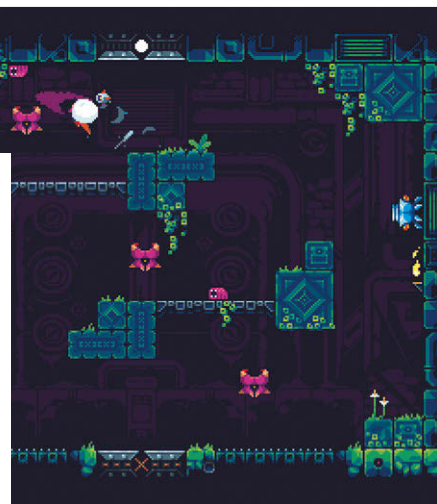
Beyond bosses and levels, there's the simple fact that using all procedural generation limits the shared experience from player to

player – and when it's a game so reliant on skill and ability to get through challenges, that *is* important, whether it's through

sharing strategies, or just for your classic git gud boasting points. Keeping control over what players play matters as much as the level of control players have over it.

With *ScourgeBringer* currently available on Early Access (wfmag.cc/Scourge) and the team operating an open development process via its Discord, this is actually a situation where players can be involved before the game is finished and – perhaps – influence its development in some way. Even so, what we have right now is well-honed, well-made, and already a hell of a lot of fun. It was back in Gamescom in August 2019, it still is now in April 2020, and it likely will be later in the year, whenever *ScourgeBringer* gets its final release. 🎮

"We spent an entire year just tweaking the game feel"



▲ It cannot be understated just how fluid and snappy *ScourgeBringer's* controls feel. It's a testament to Flying Oak's hard work, tweaking, and tinkering with timings.



▼ Upgrades, of course, play a big part – you can succeed without them if you're good enough, but the right new weapon at the right time can be a blessing.





Lord Winklebottom Investigates

Developer Charlotte Sutherland chats to us about her quintessentially British whodunit

Info

GENRE
Sleuth-'em-up

FORMAT
PC / Mac

DEVELOPER
Cave Monsters

PUBLISHER
Cave Monsters

RELEASE
Summer 2020

It goes without saying that nature's red in tooth and claw, but if a recent crop of adventure games are anything to go by, then animals are even more devious when it comes to their acts of killing than we previously thought. Sheep, rabbits, cats, and mice are among the murder suspects in the forthcoming *Chicken Police*; the suspicious death of a feline businessman forms the basis of Goloso Games' *Inspector Waffles*; and now we have the slaying of a wealthy axolotl to solve in developer Cave Monsters' *Lord Winklebottom Investigates*.

Where *Chicken Police* and *Inspector Waffles* are steeped in the staples of American crime fiction – they're all scruffy cops, urban malaise, and corrupt one-percenters – *Lord Winklebottom Investigates* follows the grand tradition of British thriller writers like Agatha Christie and Arthur Conan Doyle, and the movies adapted from their most famous works. Lord Winklebottom is an

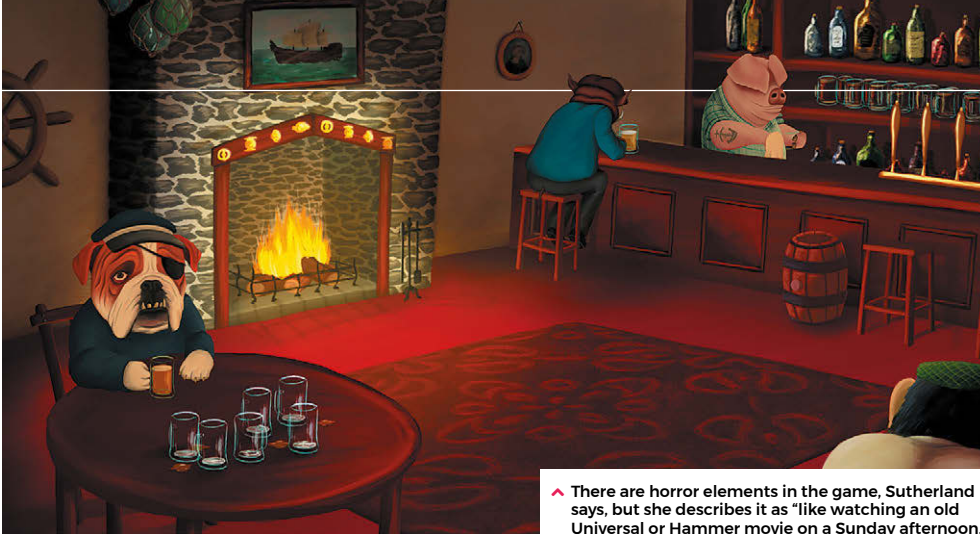
urbane sleuth who also happens to be a giraffe; his sidekick is an unassuming hippopotamus named Doctor Frumple, and their latest case is the murder of Admiral Gilfrey: axolotl, ex-sailor, explorer, and old friend of Lord Winklebottom's. Their point-and-click adventure is, to use an appropriately old-fashioned term, a proper ripping yarn.

"I've always been a big fan of old British murder mysteries," Cave Monsters developer Charlotte Sutherland tells us. "I grew up reading Agatha Christie novels and then moved onto the Sherlock Holmes stories. I really enjoy watching the old black and white movies, and loved the fact that even though they were thrillers, they usually tried to add a bit of humour into them in some way."

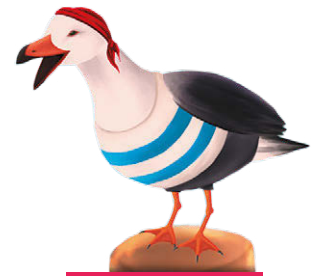
Winklebottom and Frumple's point-and-click investigation takes them to the late admiral's private island, and a bracing mystery that takes in a rogues' gallery of suspects, an array of puzzles, and some increasingly bizarre discoveries. But while the game has its roots in British murder mysteries, development on *Lord Winklebottom* didn't begin with the fevered writing of a complex thriller, but rather a series of illustrations, Sutherland explains. "Originally, I just painted the characters as standalone pieces, and wasn't really thinking about them being for a game," she says. "Winklebottom was the first character I drew, as I liked the idea of a posh, slightly snooty giraffe, especially after seeing how graceful they are in real life. After I'd created a few other characters in the same style, I started to think about the world they might



▲ Sutherland's art style, which depicts a 1920s world of decadence and interspecies romance, is undoubtedly a big draw.



▲ There are horror elements in the game, Sutherland says, but she describes it as “like watching an old Universal or Hammer movie on a Sunday afternoon.”



WEAPON OF CHOICE

Given how strong the art style is in *Lord Winklebottom*, we were keen to discover Sutherland’s process for creating it, and how she integrates it into the game. “For the artwork, I’ve been using Clip Studio Paint,” she says. “I find the brushes really mimic the feel of painting naturally. To save time with the animation, rather than using sprite sheets, I’m using Spine, which allows for skeleton-based rigging and weighting of 2D artwork. Spine easily integrates with Unity, so it allows for a very fluid workflow.”

Over in Unity, Sutherland then uses the Adventure Creator plug-in to handle scripting and in-game objects. “Since chatting to other developers of point-and-click adventures, I’ve found that Adventure Creator is quite widely used for creating narrative games. One great advantage of using an engine like Unity is that it’s reasonably easy to port to different platforms, so the game will be on consoles as well as PC/Mac.”

live in, and the idea for using them in a murder mystery-style adventure game came together.”

Lord Winklebottom Investigates’ characters are a clear standout, and there’s a good reason for that: Sutherland has over a decade of experience as an animator at such studios as Sumo Digital, Rare, and EA, and has games like *Little Big Planet 3* and *LEGO Batman* on her CV. Now based in Sheffield, she divides her time between lecturing in game development and working on her first solo project. “I’ve found it allows me to have a better work-life balance and enables me to spend the rest of my time being dedicated to developing the game,” Sutherland says of her part-time work at Sheffield Hallam University. “It’s definitely not something I think I’d have been able to do if I were still working full time in triple-A.”

After one earlier false start on Kickstarter, Sutherland got her first big boost in May 2019, when *Lord Winklebottom*’s second campaign successfully amassed £12,551 in pledges – funds that she’s been able to plough into the current phase of the game’s development. The difference the second time around, Sutherland tells us, is that she was able to build more of an audience through social media. “The second campaign benefited a lot from this, and got off to a much stronger start thanks to the people who’d found out about the game from the first Kickstarter. It’s very hard to build momentum if the campaign doesn’t get off to an immediate strong start.”

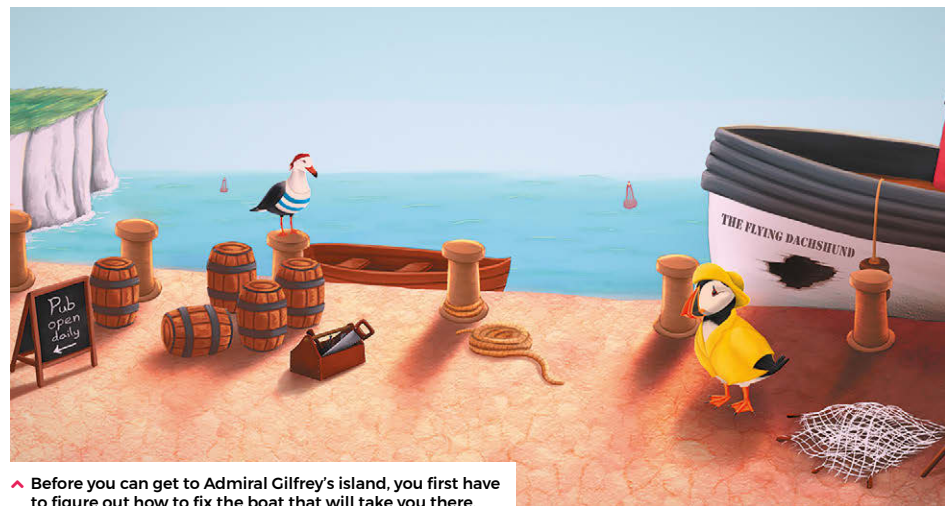
Since then, Sutherland has been working away on the game’s script and puzzles, since the two are so closely intertwined; once those are in place, she’ll then turn her attention to recording and editing the rest of the game’s quintessentially British voice acting. “I’ve not done the full voice recording yet,” Sutherland explains. “I want to leave it as late as possible to avoid having to do pick-ups later. I’d expect the VO to run into a few hours, though, so there will

“I’ve always been a big fan of old British murder mysteries”

be a lot of editing to do at some point. Picking the best takes, cutting out the individual lines and naming them correctly so they match up to the script in-game takes quite a long time, as I’ve found out while working on the demos. Sometimes lines change slightly when recorded too, so you have to go back and fix the game script to match.”

As you’ve probably gathered, there’s a rich seam of gentle humour running through *Lord Winklebottom*, from its plummy dialogue to its surreal visuals: there’s definitely something morbidly fascinating about the sight of a dead amphibian wearing a smoking jacket. And as for the murderer – well, Sutherland tells us

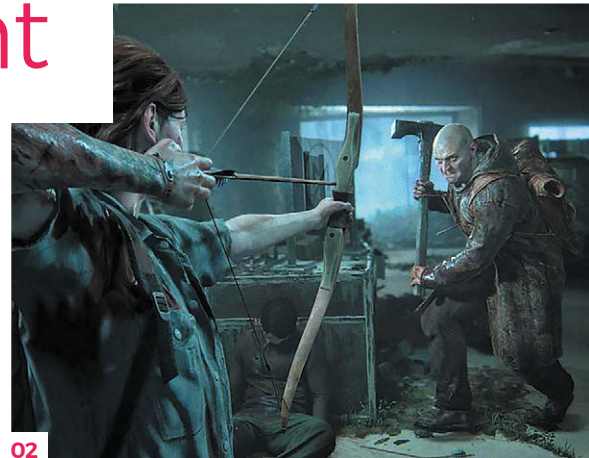
that we can expect the mystery to take a more macabre turn as the yarn unspools. “The game certainly gets darker as it goes on, and you’ll see a bit more of the nature of the animal characters come out from underneath their civilised veneer. By the end, I hope that people will feel that the animal characters made sense, and that this wasn’t a story that could have worked just as well with humans.”



▲ Before you can get to Admiral Gilfrey’s island, you first have to figure out how to fix the boat that will take you there.

Headlines

from the virtual front



01. Borderline bonuses

Gearbox's *Borderlands 3* devs will not be getting pay cheques as large as they might have expected, leading many to claim an exodus from the studio will follow soon enough. As Kotaku reports, employees at Gearbox will receive bonuses, but "nothing close to the tens of thousands or even hundreds of thousands that many had expected." The studio offers below-average games industry salaries for workers, with the promise of profit-sharing bonuses based on royalties. While this has been a positive in previous years, and while *Borderlands 3* has sold tremendously well, the increased costs around development (and a new studio in Quebec, Canada) means those bonuses have taken a massive hit.

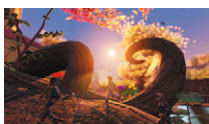
02. Definite indefinite delay

The Last of Us: Part II has been delayed 'indefinitely' by Sony as a result of the ongoing global pandemic. It's not quite as drastic as it sounds – the game hasn't been cancelled, it's just there isn't a new release date yet, so the 'indefinite' moniker has to stick. While the game itself is all but finished, Sony made the call to delay the Naughty Dog-developed title as a result of logistics issues. Basically, the publisher would be unable to get the physical game out there to enough places under current circumstances, so rather than half-arse it, the delay means it can be full-arsed at a later date. So logistics, sure, and the whole 'game about a pandemic wiping out humanity' thing, we'd guess.

03. EGX go-ahead

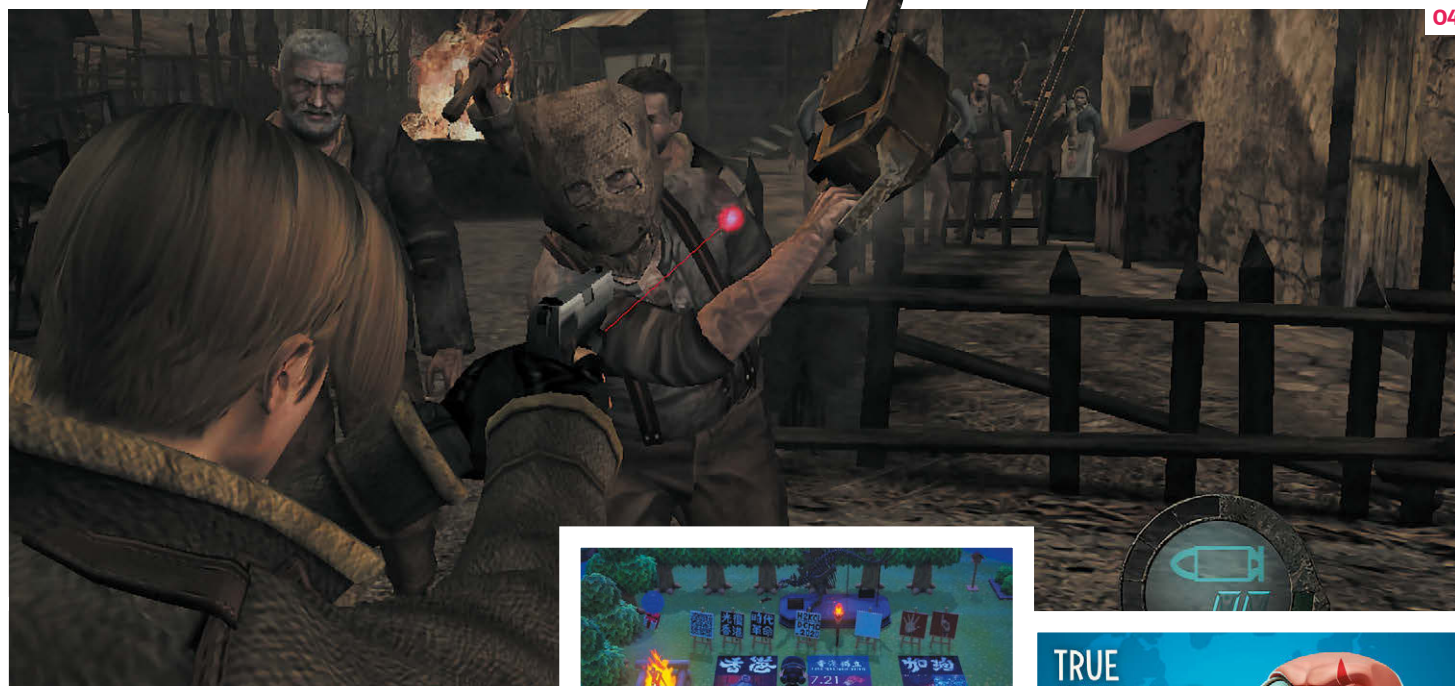
EGX, ReedPOP/Gamer Network's big event where all the games come together and you go there and play them and stuff (not its official description) will go ahead from 17-20 September, all things going to plan. The event will take place at the ExCeL in London, as originally planned, though few further details are available right now.

The main question raised is that of timing, with ExCeL currently being used as a giant field hospital to help London cope with the ongoing COVID-19 pandemic. If the hospital space is still needed by September, it'd be unlikely for a gaming event to go ahead. We'll keep an eye on it.



Obsidian's *Grounded* to include arachnophobia mode for spider-averse players

Project Zero return 'up to Nintendo', says series producer



04. Resident news-ville

Rumours in a *Resident Evil* style: *Resident Evil 4* is getting the remake treatment, will release in 2022, and is being handled by M-Two, a new-ish team made up of ex-Capcom and Platinum staffers. Shinji Mikami is also rumoured to be in on the project, which has to be a boon. This via Video Games Chronicle and leaker extraordinaire Dusk Golem (probably not their real name).

Elsewhere, we have some juicy rumour-details about the as-yet-unannounced *Resident Evil 8*, again from Dust Golem: it'll be called *Resident Evil: Village* (because 'VIII' sort of fits in the word, hmm), it features a witch who 'haunts' the player, and Chris Redfield will make himself very much known as a major focus of the sequel. Finally, it was originally set to be a new *Revelations* game, but was shunted to being a full-on *Resi 8*, and is slated for release in 2021. Take it all with a gargantuan pinch of salt.



05. Animal (not) Crossing (borders)

Sadly, not everyone with access to a Nintendo Switch is able to play the turnip markets in *Animal Crossing: New Horizons*, as sales of the game in China have halted. Is it because of this rural echo of Gordon Gekko filling players with an insatiable avarice for all things stock market-y and turnip-shaded? Nope, it's because of anti-government/pro-Hong Kong messages in the game. Note: not by Nintendo itself, but by players. At the time of writing, there had been no official government acknowledgement of the ban – possibly because the game was never *officially* available in the country to begin with – but online ports of entry for the title are no longer offering the title for delivery to the country.



06. Government Rebellion

UK developer, publisher, and comics company (and sometime movie studio) Rebellion has partnered with the Department for Digital, Culture, Media and Sport "to deliver crucial health messages through our games." What, and why? It's essentially a bit of that 'stay inside, save the NHS' messaging you've seen everywhere from the government, but patched into Rebellion's existing games (and used in some of its comics, too). Messages will appear in easily updateable 'live' areas of online games across all formats, and will be in the studio's titles such as *Zombie Army 4*, *Sniper Elite 4*, and *Strange Brigade*.

PC *Final Fantasy IX* patch temporarily removes... *Final Fantasy IX* on PC



Crysis Twitter account comes back to life after three-and-a-bit years

Wireframe



Hydroneer ↗

We're not going to pretend this is much more than a slightly more realistic-looking *Minecraft*-like, but that doesn't change the fact that *Hydroneer* has some appeal. And what is that appeal? Why, it's all in digging big holes, of course. Sometimes you just want to dig big holes, whether that's with a shovel or an earthmover, and this game lets you do just that.



Clodhoppers ↗

A stop-motion *Smash Bros.*, *Clodhoppers* lets you bring together up to eight people – local or online – to fight. That's about it, really. What makes it stand out is that visual style, though, with developer Claymatic Games opting for a clay model stop-motion animation technique for all its in-game artwork. Will it make *Clodhoppers* a better game? That's up for debate. But it definitely stands out in a dominated genre.

Rock of Ages 3: Make & Break ↘

It's almost ten years since the original came out, and still the *Rock of Ages* series continues. ACE Team returns with its third in the series, lacking the excellent subtitle of the second game (*Bigger & Boulder*), but with more of that fine mix of action and tower defence that's made the series a fun one over the years. Beyond the tweaks to setting and better presentation you'd expect, the sequel also brings the chance to share your created levels online – always a boon.

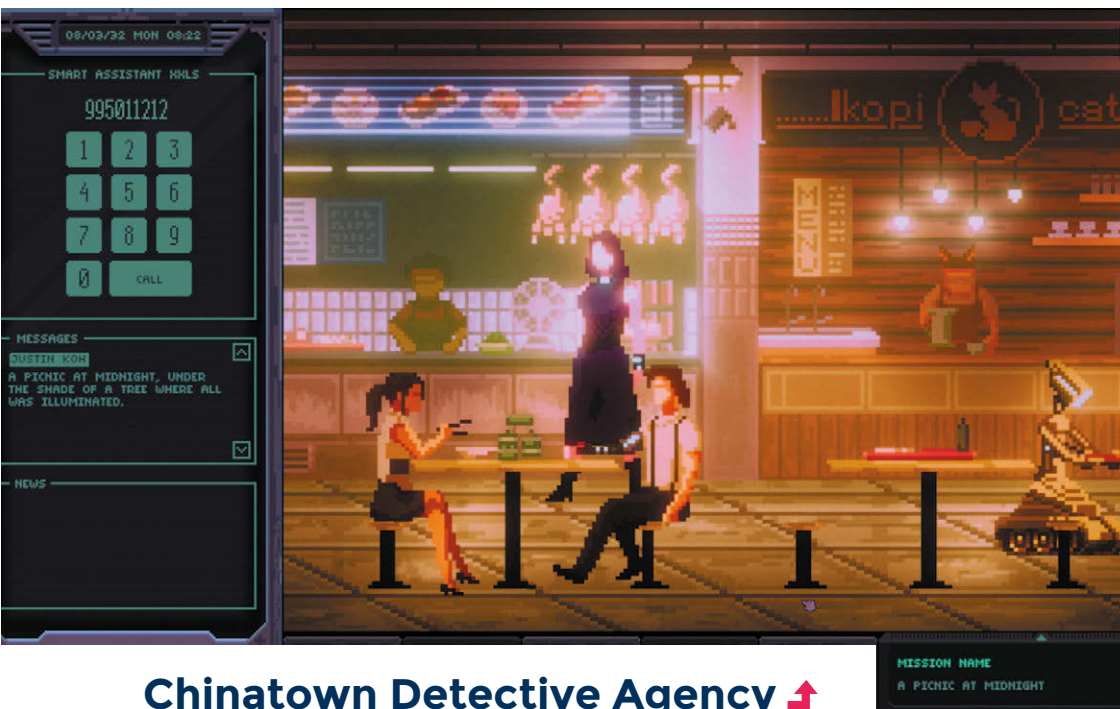


Witchbrook ↗

Chucklefish's third game was announced, went a bit quiet, and then was recently sort of re-announced with a new look and... a few details. In short, it now looks like *Final Fantasy Tactics*, which is brilliant, and it's been a bit more confirmed that it's a town life sim (*Stardew Valley*, *Harvest Moon*, *Animal Crossing*, et al) mixed with a magic school student-'em-up. We invented that genre. *Stardew Potter?* Maybe.

It's not much to go on, but it's enough to whet our appetite significantly. Look for more on *Witchbrook* in a future issue of *Wireframe*, as long as we're able to conjure up the right unholy demons for the ritual sacrifice! Wait, that went too dark.





Chinatown Detective Agency ↗

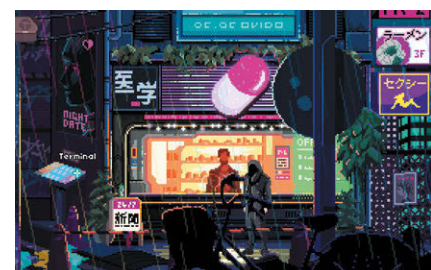
General Interactive Co., the studio made up of two developers, returns with this Kickstarted point-and-click adventure game set in a 'cyber-noir' world. In it, players are tasked with solving cases brought to their titular agency – but rather than the usual rubber-chicken-pulley solutions of old, *Chinatown Detective Agency* pushes players to look into the real world around them for the solutions.

It's all inspired by the *Carmen Sandiego* edutainment games you might remember from your younger years (you might not though, given they were markedly more popular in

places outside the UK). That is to say: it's lots of globetrotting, lots of mysteries to solve, and lots of overarching conspiracies to unravel – though *Chinatown* looks a lot less aimed at ten-year-olds than *Carmen* and co was. That real-world factor makes itself known by the player having to, say, search online for a quote to see where it came from in order to figure out a password. How that plays out will be key to how the game as a whole holds together, and hopefully, it won't just be a gimmick slapped on for the attention it (rightly) brings. You can play a demo now, right here: wfmag.cc/CDA.

VirtuaVerse ↘

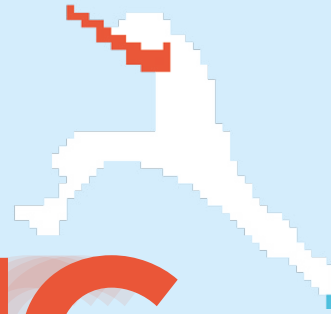
Cyberpunk, modern sprite artwork, and point-and-click are three elements that have been mixed together frequently in the past decade or so, as well as on these pages. *VirtuaVerse* leans into this reliable mix to produce a deep and atmospheric near future full of darkness, neon, and rain. It'll live or die by its story and puzzles, of course, but for now, we can uncritically enjoy how pretty it is.



Football Story ↘

It's the *Soccer Kid* follow-up we never knew we wanted. Alright, it's nothing to do with Krisalis' Amiga classic-of-a-sort, but *Football Story* does bring with it handfuls of childlike wonder surrounding the sport that's so utterly corrupted by the time you're old enough to understand where all its money comes from. A narrative-heavy, single-player campaign is the real draw here, and it'll be genuinely interesting to see what developer fructus temporum ends up producing.

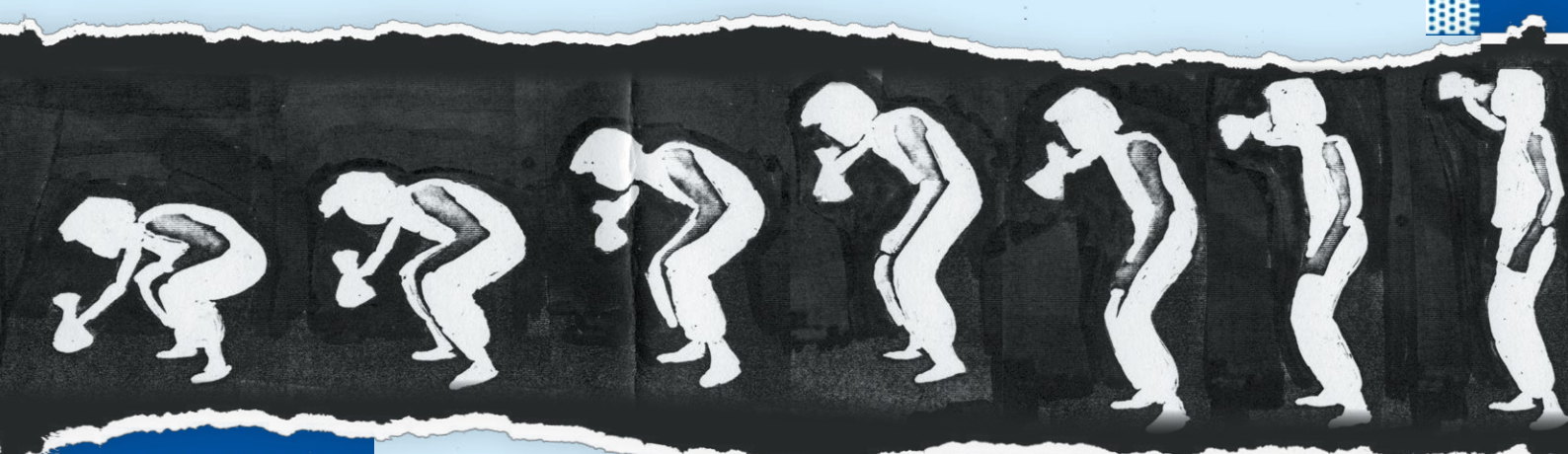




TURNING BACK TIME

Prince of Persia designer Jordan Mechner reflects on the pioneering techniques he used to make a classic

WRITTEN BY
DAVID CROOKES

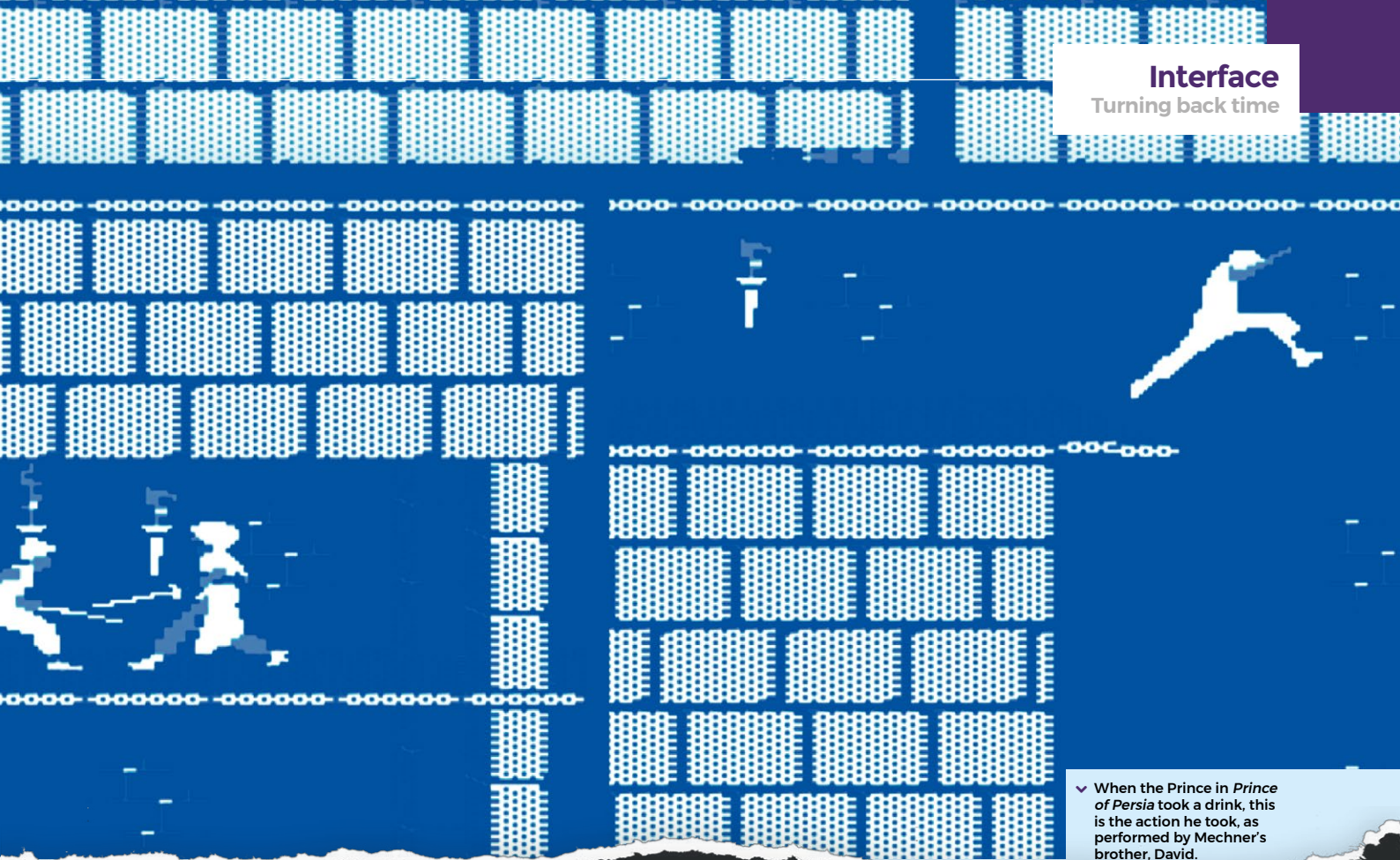


When Jordan Mechner was growing up in 1970s New York City, he, like many kids, enjoyed playing the arcade machines that were just emerging at the time. "I used to play the likes of *Space Invaders* and *Breakout*, and they became the kind of games that I aspired to make," he recalls. After saving up enough money from drawing cartoons to buy an Apple II computer, Mechner set about coding his own version of *Asteroids*. "But Atari was cracking down on companies selling unauthorised knock-offs of their titles on floppy disks," Mechner says, "so my game – called *Asteroid Blaster* – couldn't be published."

Undeterred, he spun the project into *Deathbounce* – "a game like *Asteroids*, but with

brightly coloured bouncing balls" – which he sent to the publisher, Brøderbund. Again, it was rejected, but he received a pivotal call from the publisher's founder, Doug Carlston. "He said my game was a little bit last year – kind of 1981," Mechner laughs. "He also said Brøderbund's big hit at the time was *Choplifter* and he encouraged me to check it out, sending a copy in the post and the joysticks to play it. That game just blew me away."

Choplifter was a side-scrolling action game developed by Dan Gorlin for the Apple II. Assuming the role of a helicopter pilot, players rescued prisoners of war while fending off the enemy. "What I loved was that it told a story, and it wasn't just racking up points to get a high score," says Mechner. "*Choplifter* was a



▼ When the Prince in *Prince of Persia* took a drink, this is the action he took, as performed by Mechner's brother, David.



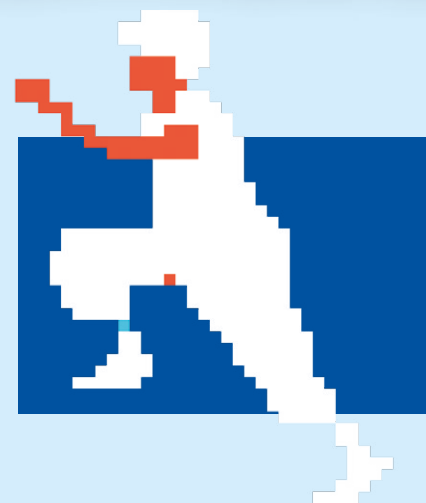
revelation, because I realised I'd been copying the coin-op format without realising that home computers didn't have a coin slot, and so games could be different and tell tales with a beginning, middle, and end."

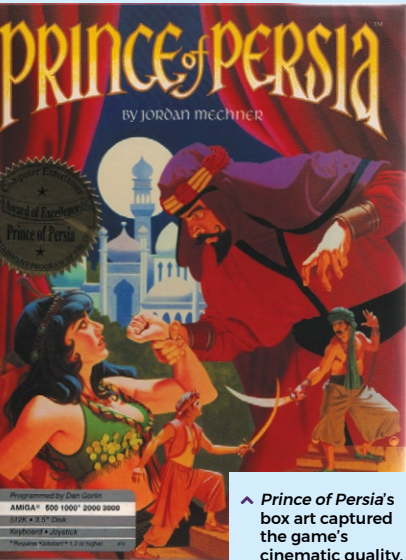
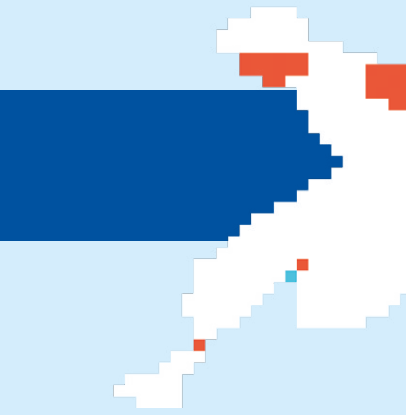
DEVELOPING A TECHNIQUE

Mechner's entire approach to video games changed in an instant, and he decided he wanted to develop titles with a cinematic quality. As a film student at Yale University, he was influenced by the movies of Japanese director Akira Kurosawa, in particular the epic 1954 drama, *Seven Samurai*. "I enjoyed the Samurai mystique of bravely going forward and accepting death as a possibility," he explains. "I was fascinated and inspired by the spirit of Bushido

which dictated the Samurai way of life, as well as the stylistic simplicity of Japanese landscape prints, such as [Katsushika] Hokusai's *Thirty-six Views of Mount Fuji*."

Mechner also enjoyed kung-fu movies, which led to the development of *Karateka* – a martial arts action game which earned itself a Guinness World Record for being the first video game to make use of motion-capture animation. The technique Mechner used was called rotoscoping, which involves filming real-life movement and tracing over the footage to copy it to a computer. "*Karateka* was the first time I'd used rotoscoping," Mechner says. "I used a Super 8 camera to film my karate teacher doing the punches and kicks that would be needed for the game, and I filmed my dad running and ▶





▲ *Prince of Persia's* box art captured the game's cinematic quality.



▲ Mechner taking a break from work on *Prince of Persia*, with the rotoscoping images he took while developing the game stuck to the walls.

climbing onto the hood of a car to simulate climbing up a cliff."

For Mechner, it was a way of exploring a new medium in a manner that hadn't been seen since the early days of film. "I was watching early silent films, and I realised how the cinematic language that we take for granted with the close-ups and camera movements had to have been invented," he says. "I thought computer games were this new audio-visual medium with a potential that hadn't really been explored yet, and my eyes were open to the possibilities."

FRAME BY FRAME

Rotoscoping dates back to 1915, when animator Max Fleischer first used it in the silent era series, *Out of the Inkwell*, which featured a highly realistic moving character, Koko the Clown, posed by Fleischer's younger brother, Dave. It was a technique Disney's animators would use to similarly groundbreaking effect in the studio's first feature film, *Snow White and the Seven Dwarfs*, in 1937.

"Those animators were tracing onto paper and then onto acetate cells, but the process of replicating the technique for *Karateka* on the Apple II was a little more complicated," Mechner explains. "I had to trace pixel by pixel, and it was very much a homemade process."

"One of the other difficulties was the worry about wasting film. With Super 8, you get just a couple of minutes on the roll, so you're kind of in a panic because if you wasted 20 seconds of film, you couldn't just rewind and do another take. You also had to get that film developed, which took time."

Made in just 48kB of memory, *Karateka* was a massive hit when it came out on the Commodore 64 in 1984. Buoyed by that success – and having made enough money to pay off his loans from college – Mechner began working on his second game. An even more ambitious project than *Karateka*, *Prince of Persia* took a long four years to develop. Coded in 6502 assembly on the Apple II once more, it too made use of the rotoscoping technique, this time swapping a Super 8 camera for VHS.

"It was a step up, but it was still a very slow, manual, step-by-step process," says Mechner. "The big advance between 1982 and 1985 was that a company in England called Compu-Tech had created a digitiser card, which you plugged a video camera into. I just needed to create sheets in which each animation frame was clearly silhouetted in white against a black background."

To create a sheet with eight to twelve frames of animation on it, Mechner would videotape a model wearing white clothes – in this case,

"I had to trace pixel by pixel, and it was very much a homemade process"

his brother Dave again, who would run, jump, and climb, creating the movement that would translate into the character of the Prince. "I would then put the tape into a VHS player that had a clean freeze-frame and put a 35-millimetre film camera on a tripod in front of the TV screen," Mechner explains. "The idea was to do a frame advance, take a picture, do another frame advance, take another picture and so on, using every third frame, and then take that roll of film to the photo lab to be developed. I'd get back a little stack of photographs, each of which was a frame of video. I'd then paint those photographs with a black marker pen and use white to create a silhouette before taping them all together onto a sheet."

▼ *Karateka* was released in 1984, initially for the Apple II computer. There was a remake released for the Xbox 360, Windows, PlayStation 3, and iOS in 2012.



READ TO WRITE

Jordan Mechner's love of a particular genre of cinema led him to create the game worlds for *Karateka* and *Prince of Persia*. "I was inspired by exotic fantasies set in historical periods and cultures that I'd never experienced directly – movies such as *The Thief of Bagdad*," he says, referring to the Arabian fantasy film released in 1940. When he revisited *Prince of Persia* for *The Sands of Time* game and movie, he began avidly reading. "I read the original *One Thousand and One Nights* as well as *The Book of Kings* [an epic poem written by the Persian poet, Ferdowsi]. I wanted to get closer to the source, and I found Persian culture to be a wonderful, deep treasure trove of powerful themes and timeless characters."



This sheet would then be digitised into the Apple II and cleaned up, pixel by pixel. The new process was faster. A VHS tape could be played back immediately to see what had been recorded; mistakes could be rewound and a new recording made. Sure, there were easier ways to create graphics, but Mechner wanted to raise the bar. "I appreciated good animation – Disney animation," he says. "I'd grown up on that since I was a kid, and the fluidity and illusion of life was wonderful to me."

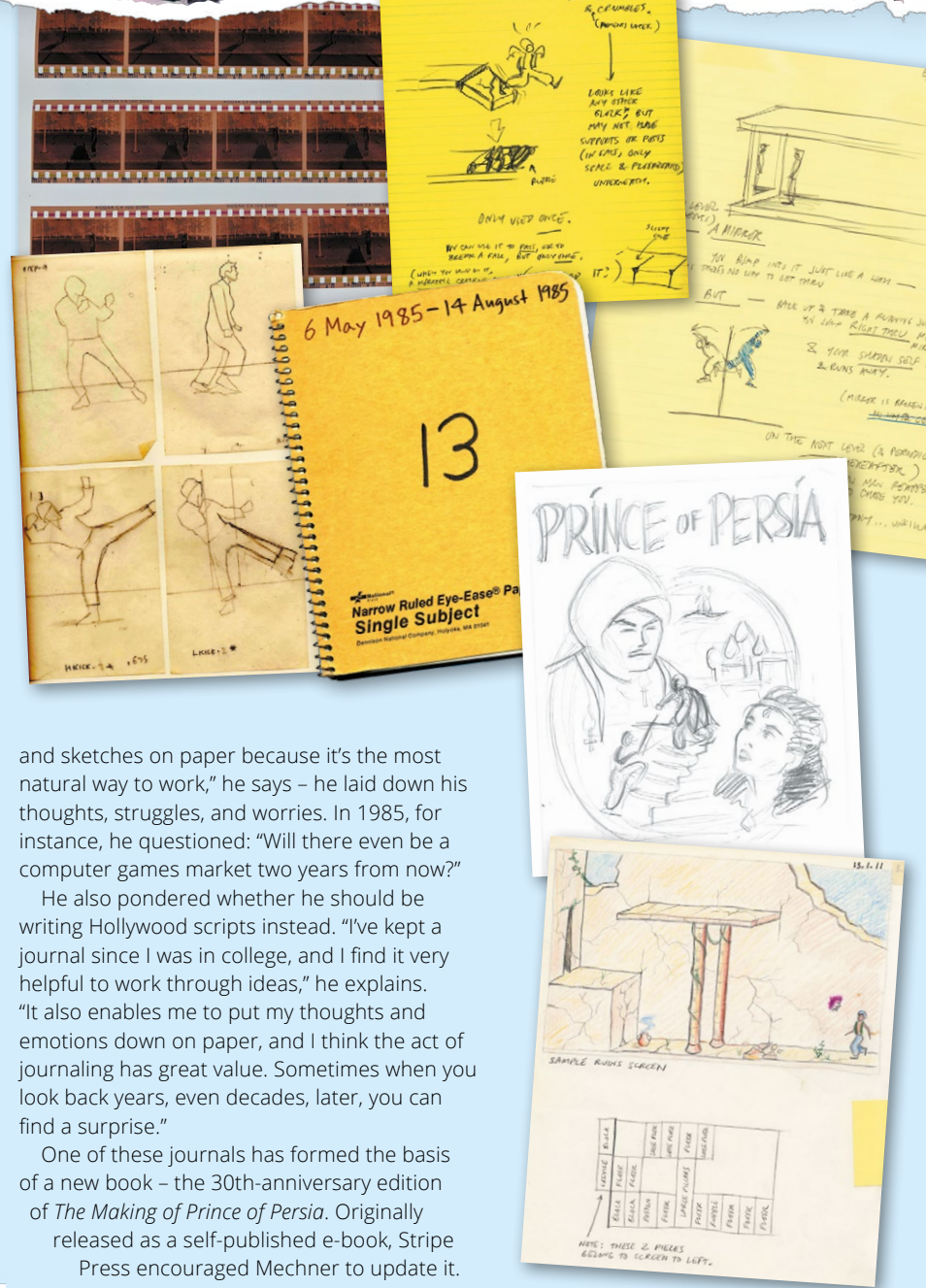
Mechner had tried to create animations without a filmed reference, but he felt the movement always looked artificial. "You could see that they had no weight and they never quite looked right," he says. "My vision was to have a character who felt fluid and lifelike so that players would feel suspense, knowing if they moved too far, they could fall into spikes and that it would really hurt. Rotoscoping was my answer, and I found out Disney animators had used the technique as well."

Prince of Persia was a masterpiece of game design. It gave players just 60 minutes to lead the Prince out of a network of dungeons to save his imprisoned lover from the evil clutches of the power-hungry Jaffar. Cutscenes and explanatory text progressed the narrative, while the game itself was high on tension. The Prince battled against enemy swordsmen, avoided spikes and deep pits, and used pressure pads to open time-sensitive gates. The fluidity of movement saw the Prince hang on to ledges, pull himself up, and leap across chasms with a human-like fidelity.

"I was taking the puzzle-solving, exploring aspects of a platform game and combining it with the visceral sense of danger of an *Indiana Jones* movie," Mechner says. Much of his inspiration also came from the 1983 game, *Lode Runner*, which he felt would work well in combination with the cinematic drama of *Karateka*. "Yet with *Lode Runner*, you didn't feel a sense of jeopardy; it was abstract, and you were only really trying to beat the level."

MIND OVER MATTER

During the development of these games, Mechner kept a journal. As well as using it to sketch his ideas – "I've always started with notes



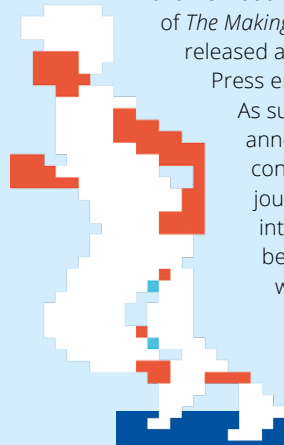
and sketches on paper because it's the most natural way to work," he says – he laid down his thoughts, struggles, and worries. In 1985, for instance, he questioned: "Will there even be a computer games market two years from now?"

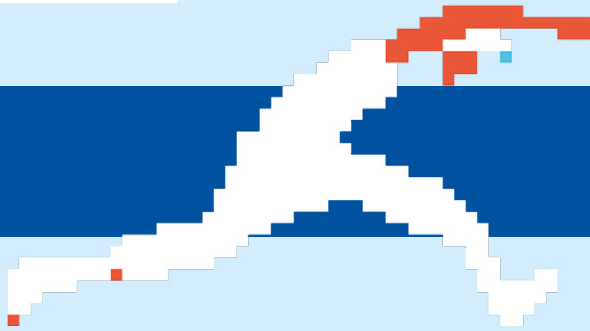
He also pondered whether he should be writing Hollywood scripts instead. "I've kept a journal since I was in college, and I find it very helpful to work through ideas," he explains. "It also enables me to put my thoughts and emotions down on paper, and I think the act of journaling has great value. Sometimes when you look back years, even decades, later, you can find a surprise."

One of these journals has formed the basis of a new book – the 30th-anniversary edition of *The Making of Prince of Persia*. Originally released as a self-published e-book, Stripe Press encouraged Mechner to update it.

As such, he's added illustrations and annotations explaining his design concepts. "There are parts of the journal that might be especially interesting to game developers because, as designers, we struggle with the same issues," Mechner says. "It's also interesting to see the ways in which the industry ➔

▲ "Journals can set your memory straight – you can't deny what happened in the past because you've got a primary source that can teach great lessons and perspectives," Mechner says.

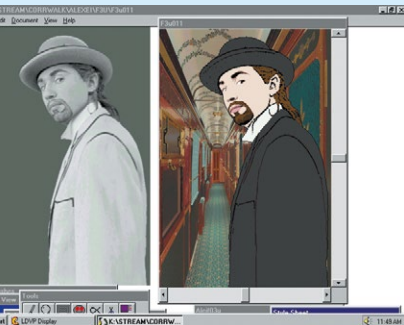




▲ The pioneer today: it's Mr Jordan Mechner.

UNIQUE EXPERIENCES

Rather than push players into a pre-determined experience, Jordan Mechner sought to create opportunities for personal moments in his games. "With *Prince of Persia*, I wanted players to experience an organic, cinematic thrill – a feeling that what just happened to their character in the game was something they'd brought about themselves. It was about allowing players to create moments that not everyone who played the game would encounter."



▲ Footage captured during *The Last Express*'s 22-day shoot was processed using a patented in-house tool, which removed all colour and created black-and-white line drawings that were then hand-coloured.



▲ Mechner (pictured right) had a professional studio set up for the rotoscoping technique used in *The Last Express*, with actors set against a blue screen.

has evolved and the ways in which it has stayed the same. We still have crunch, battles for QA time, and struggles with marketing. There's still an emotional rollercoaster of trying to do something that you're not sure is going to work."

Mechner has certainly experienced his share of highs and lows. He was delighted when *Prince of Persia* was widely acclaimed by critics, and saddened when sales were initially slow (the Apple II version sold just 7000 copies in its first eight months or so). But as word got out and the game was ported to other machines, those sales increased hugely. "It's always worth remembering that finished products don't just spring into being," he says. "As can be seen in my journal and in other people's memoirs, production can be dicey.

Creators can be a mess. They don't have all of the answers, and yet that can give others hope."

LEVELLING UP

As if to underline that struggle, Mechner turns our attention to *Prince of Persia*'s level design. At first, he intended to include his level editor on the floppy disk, just as the developers of *Lode Runner* had done with their game. That didn't happen, but the tool was so flexible that Mechner ended up using it to rapidly draft and rework *Prince of Persia*'s levels. Nevertheless, Mechner's pursuit of perfection took development to the wire. "Most of the final twelve levels that everybody knows were built or rebuilt in the last couple of months before the game shipped, and everything that I had done up to that point, kind of became the first draft," he recalls. "But that ability to iterate quickly to build new levels was really valuable. It allowed me to fine-tune the difficulty and sense of

adventure. We've lost that today with games that are so extensive."

Variety was important to Mechner, because he wanted to avoid repetition in his level design – a real danger, given that his dungeon was made from relatively few set-pieces. "I was looking at creative ways to use those pieces," he says. "I was asking myself how I could recombine falling floors and pressure plates in a way that was different from the way we'd already done it ten times before. So the level design was a combination of storytelling, pacing, and managing the difficulty of teaching the players skill. I wanted them to eventually master the challenges that were so tough at the beginning."

Just as crucially, Mechner also wanted the levels to play out in a film-like manner by keeping things tense. "In arcade games, the general rule was to have three lives," he begins. "But, in *Karateka*, I had a short enough story to just have one and, with *Prince of Persia*, where the levels were longer and greater in number, I had more thinking to do. I felt limiting lives would be too mean to the player, but that infinite lives would lose the tension. So I used a health indicator and placed potions around the levels that would top it up. Taken with the 60-minute time limit, I'd still allow the player to get sweaty palms as they tried hard not to die, but they'd also get a lot of gameplay under their belt before that point."

EVOLVING OVER TIME

Prince of Persia 2: The Shadow and the Flame continued the theme set in the first game, albeit with more combat and the further headache of having to fend off up to four attackers. There was a 75-minute time limit

"I felt limiting lives would be too mean to the player"

this time around, while Mechner took on a different role in the game's making. "We had a programmer, sound designer, and artists for that game and I was the creative director," he says. For this title, Mechner's sketches and script were used to guide the development team, and it continued to ratchet up the tension and suspense that you would find in an action movie.

"I always liked that feeling in *Prince of Persia* where you think you can make a jump and, when you do, you get this huge feeling of satisfaction," he says. "There's a sense that, if you miss, it's your fault, but that you can get it right next time and make it through. This created those little movie moments like pushing a guard off the ledge with your sword and seeing them fall on to spikes or backing them into a slicer. It's like when Indiana Jones backed a Nazi mechanic into the plane propeller in *Raiders of the Lost Ark*."

After the *Prince of Persia* sequel, Mechner largely moved away from the series, only briefly consulting on *Prince on Persia 3D*, published in 1999. He founded Smoking Car Productions and brought together a team to create a real-time, non-linear adventure called *The Last Express*, which took four years and a \$5m budget to make. Again, it used rotoscoping, this time with professional actors filmed against a blue screen.

"It was a real labour of love recreating the 1914 Orient Express from our studio in San Francisco," Mechner says. "It involved a pretty sophisticated, automated digital rotoscoping process, and it combined my filmmaking studies with everything I learned from making the *Prince of Persia* games."

Thousands of frames of animation were created making use of CD-ROM and computers with many times more memory than the Apple II. "In order to rotoscope that many frames, we had to automate it, so we built and patented a process to turn film footage into something resembling a kind of pen and ink drawing," says Mechner. "We were inspired by Art Nouveau, and we didn't want it to look like full-motion video, but more hand-drawn."

Banks of Macs were needed for the 3D rendered interior of the train, and the resulting game felt akin to an art film.

A lack of marketing meant that *The Last Express* didn't sadly, get the attention it deserved – a fate that thankfully wasn't



FRESH PRINCE

After designing *Prince of Persia: The Sands of Time*, Mechner created a trailer of the game's footage and pitched it to Hollywood in the hope of attracting the attention of movie moguls. It worked: producer Jerry Bruckheimer snapped up the film rights and Mechner wrote the first screenplay. The film was eventually directed by Mike Newell and starred Jake Gyllenhaal as Prince Dastan, grossing more than \$336 million – making it the fourth highest-grossing video game movie adaptation of all time.

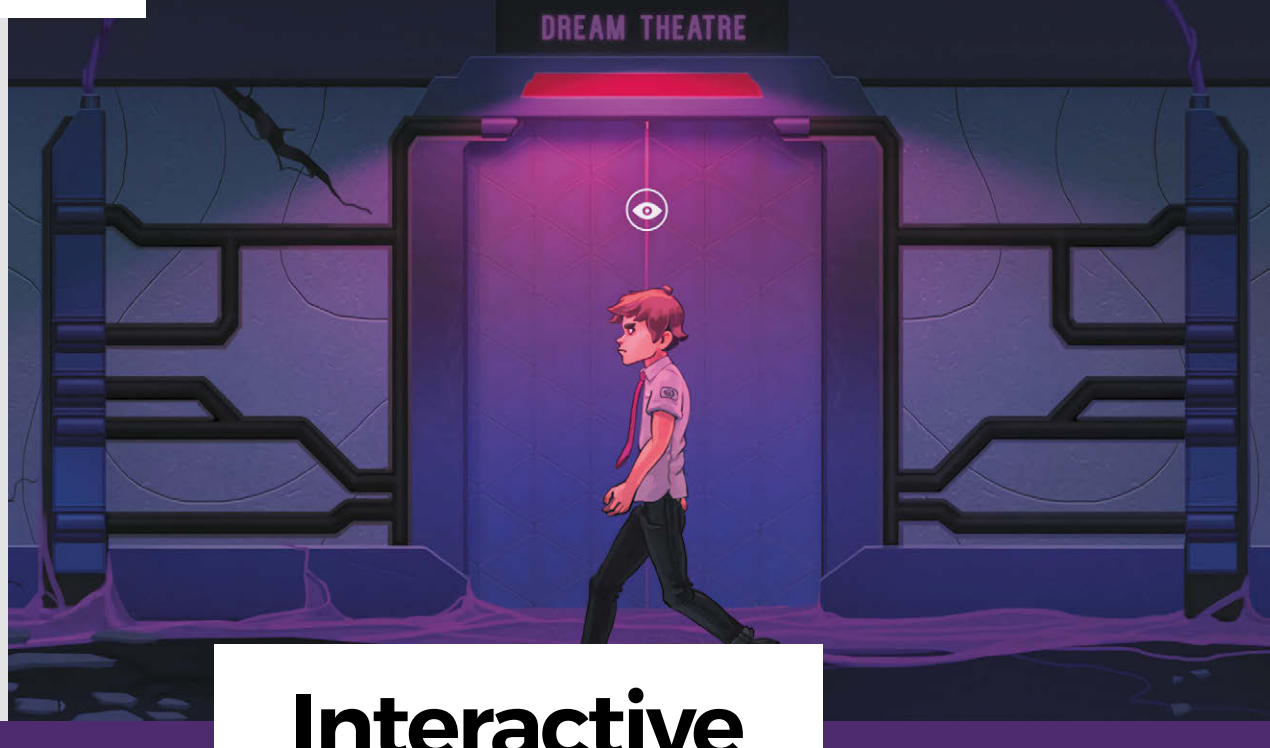
shared by his next game. In the early 2000s, Mechner was approached by Ubisoft, who planned to reboot *Prince of Persia*. "I took what I learned from *The Last Express* and made my first real-time 3D game," he says, referring to the 2003 release of *Prince of Persia: Sands of Time*, on which he was credited as game designer, writer, and creative consultant. The game revitalised the series, spawned several sequels, and gave rise to a Hollywood movie.

Mechner played an important role in those later iterations of *Prince of Persia*, but he still retains a fondness for the earlier games. "Part of the charm was that the technology was so limited, which gave them a kind of handmade quality," he muses. "But they still combined storytelling with exploration, which is what has always fascinated me about video games. Interactivity, and the player's freedom to explore, are in opposition to the control a storyteller has to shape an experience to a certain end. But when those two opposing principles work in harmony, it can be wonderful." 🗨️

***The Making of Prince of Persia* is available now from jordanmechner.com**



▲ *The Making of Prince of Persia* by Jordan Mechner is a beautiful recreation of his journals.



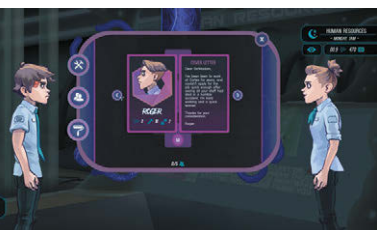
Interactive

Headspun

Developer Jamin Smith talks us through the trials of making his ambitious point-and-click hybrid

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

▼ Rebuilding Cortex - and by extension, Theo's mind - requires hiring and managing workers. Completing minigames will earn cash that you can spend on more workers and repairs.



As we've seen in previous editions of Wireframe, balance is one of the most difficult aspects of being a solo game developer: juggling family life with stolen hours in front of a computer; finding the energy to push your pet project forward after a long and exhausting day at work. This was something designer Jamin Smith found particularly challenging with his debut game, the point-and-click adventure and FMV hybrid, *Headspun*, initially released for PC in August 2019. What was originally intended as a diverting hobby quickly grew into a three-year project that had to be squeezed in around a demanding day job; towards the end, burnout set in, as Smith tells us the game "ballooned into something far too big and complex for one person to manage."

Before we get too far ahead of ourselves, though, here's a bit of back story. Smith began work on *Headspun* after a bout of reading pop psychology books - among them Professor Steve Peters' *The Chimp Paradox*. "It basically outlines how we have two voices in our heads," Smith explains. "The rational, intelligent,

conscious voice, and the primal, irrational, subconscious voice." All of this planted the seed for *Headspun's* premise: an adventure game set in the human brain, where two halves of a character's personality vie for control. Like a darker take on Pixar's animated film, *Inside Out*, *Headspun* introduces Theo, the survivor of a car accident who's lying in hospital, and the two characters inside his head whose job is to rebuild the mental devastation wrought by the crash. *Headspun* imagines Theo's brain as a high-tech science facility, overseen by the businesslike Ted (who the player controls) and the more rebellious, laid-back Teddy.

While the meat of *Headspun* unfolds as a 2D adventure, with hand-drawn graphics in the vein of artist Jamie Hewlett's work, the story also progresses through live-action sequences, where we see the outside world through Theo's eyes. It's a design choice that neatly delineates the game's interior and exterior events, while also saving Smith the time-consuming and costly process of designing and animating these sequences by hand. "As a solo developer, I could never create the amount of content needed to service the story I wanted

"Smith quickly realised that telling the story he wanted with real actors and sets would prove costly"



▼ Like *Lord Winklebottom Investigates* (see page 12), *Headspun* used Spine to generate its 2D animation.



to tell," he says. "I could, however, film it, while also benefiting from a very clear 'real world' outside of the game. The idea of a game which was half live-action and half 2D adventure really grabbed me creatively."

Smith quickly realised, however, that telling the story he wanted to tell with real actors and sets would still prove costly: he'd originally planned to have the game's framing story take place at a house party, which would have required more locations, actors, and extras to wrangle. To sidestep this, Smith rewrote the story so that it took place after the party, where Theo deals with the fallout of that night's events from a hospital bed. "This meant I could shoot the game in one room over that key period of recovery, with a small cast of actors, and tell the story I wanted to tell with the limited budget I had."

THE CRUNCH

Around two-thirds of the way through its development, *Headspun* was picked up by indie publisher Wales Interactive – a deal which meant that the finished game would ultimately

be ported to the PS4, Xbox One, and Switch.

The publisher interest was undoubtedly a boost for Smith, but it also meant that what had begun as a casual side project was now, as he puts it, "a second full-time job" that took place in-between his daytime career as global brand manager at Square Enix.

"The most challenging aspect of development was the entirety of [that last] year," Smith tells us. "I'd work 10-6.30 at the day job, travel home and rush a bit of dinner, and then work flat out from 8ish until 3am on *Headspun*. I'd do this near-enough every day of the working week, with more time poured in over the weekend. It was tough going, and burnout hit pretty hard towards the end... It was a classic case of feature creep, and my complete disregard of any sort of schedule meant that new systems and mechanics kept appearing in the game. Ultimately, I developed a game that was too big to handle as a solo developer, which meant it suffered in certain areas."

Given that it's the product of just one developer, though – one handling live-action filming and scriptwriting as well as 2D art, animation, and programming – *Headspun* is an impressive achievement. And while Smith admits that he may have given himself too much to do for a debut project, he's now taking what he's learned on *Headspun* into his next title. "The scope is certainly smaller – and something I can handle as a side project far more efficiently – but the ambition behind it is genuinely much bigger," he says. "It still has narrative very much at its core, and I've been working with some really exciting technology to bring its story to life." 🗨

▲ "I loved the idea of a dramatic distinction between the world inside and outside of the brain," Smith says.

LAUNCH DAY

After a long development, having a game finally emerge in the real world might sound like a dream come true. But for Smith, *Headspun's* release was a little more complicated. "I found launch day quite traumatising. I hated it. The game launched with a fair few bugs – especially on the console versions – and it was all I could focus on. There was some good PR for the game in terms of exposure, with some big publications picking up the game for review due to the unique concept, but I couldn't bring myself to read a word of it. All I could think about was how I needed another month or two to make it better. I think the pressure solo developers can put on themselves mounts up pretty quickly in instances like these – it's all on you, there's no blame to shift – and it all comes to a head when the game is suddenly available publicly."

“Start where you are. Use what you have. Do what you can.”



LOTTIE BEVAN

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

“We talk about problems, but do we do that much about them?”

Arthur Ashe was a pretty cool guy. He turned a tragic personal setback – he contracted AIDS from a blood transfusion in the early eighties – into positives: safe-sex awareness campaigns, an AIDS foundation, and ethnically integrative programmes for people without health insurance 20 years before Obamacare. He grew up in the 1940s in segregated Virginia and went on to become the only black tennis player ever to win the Australian Open, the US Open, and Wimbledon. He also had a pretty cool motto. “Start where you are. Use what you have. Do what you can.” I've found it surprisingly useful in games.

Devs do a lot of talking. Over the last decade, I've heard increasingly frequent discussion of sexism and feminism, ethnic diversity, sexuality and gender, and whether you can play games on 'easy' and still call yourself a gamer. But for all the noise on social media and all the panels at events, I don't see the same level of activity in real life. We talk about problems, but do we do that much about them?

I think there are three reasons why we don't. Firstly, actions are harder than words. Secondly, it's easy to feel that sharing a post or talking about a problem is the same as actually fixing it. Thirdly, the industry issues we talk about are all big problems: it's not 'we've run out of milk', it's 'solve homophobia', or 'get more black schoolgirls to code and apply for programming jobs when they're older'. I can see why faced with a problem like that, it's hard to know where to start.

A new genre of games emerged over the last few years which reflect this sense that everything's too much. *#SelfCare* or *Kind Words* are good examples: they're designed to give players a dedicated space to do something small

and easy and feel good about it. More widely, escapist wholesomeness like *Animal Crossing* or *Ooblets* are more and more in demand, their popularity inversely proportionate to how stressed and unhappy we are in real life.

We all need an escape from reality, particularly in a global pandemic. And games about being kind to yourself are great. But doing your laundry in *#SelfCare* doesn't do your laundry in real life. Playing diverse games won't encourage more black women to code. To make our industry a better place, we need to do things in the real world. But you don't have to solve a problem definitively or find the ultimate cure-all. You can just help a bit. Spend an extra half hour finding unusual jobs boards to widen your likely pool of applicants. Update your studio's HR policy with more generous parental leave. Give £20 to a local charity when you have it to spare.

Retweets are marginally better than not doing anything at all, but real-world acts are better. If we see any flaws in our industry, any at all, it's our problem. Not someone richer, or someone more famous, or someone with probably more time on their hands. Our industry will be a better one if we take a more moderate, more achievable approach to social change: don't worry if your game doesn't tackle all possible diversity at once. It's OK if you haven't donated to every charity listed on UKIE's website. Start where you are. Use what you have. Do what you can. Even if it's small, it's something. 🐼

▼ *Animal Crossing: New Horizons* – the perfect, wholesome escape from an increasingly worrying reality.



Toolbox

The art, theory, and production of video games

28. Design Principles

Howard Scott Warshaw
on game programmers

30. CityCraft

Urban planning as a
game design road map

32. Dungeon maker

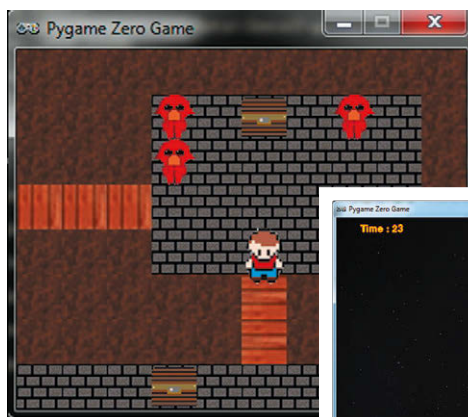
Use procedural generation
to instantly make levels

40. Source Code

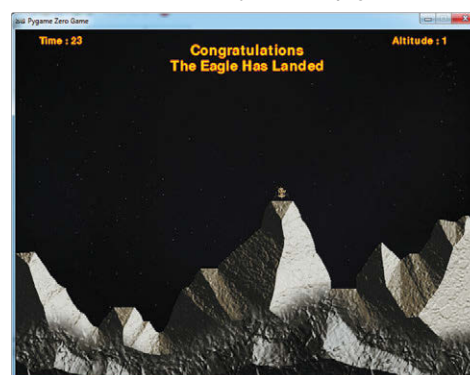
Code an update to
Atari's Lunar Lander

42. Directory

Free Unity tutorials,
and lots more besides



◀ Instant dungeons? It's easy with procedural generation. Mac shows you how on page 32.



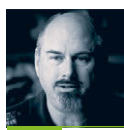
▶ The Eagle Has Landed? Only if you have a steady hand. Code your own Lunar Lander-like - see page 40.



▼ How can real-world urban planning help with game design? Find out on page 30.

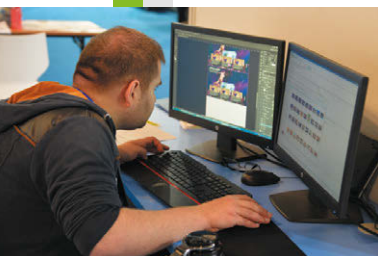
The principles of game design

What makes a game maker? And just what is the 80/20 model? Howard explains all



AUTHOR
HOWARD SCOTT WARSHAW

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



▲ Being a game developer means meeting technical requirements, sure, but also understanding what's actually entertaining to play.

There are two kinds of people in the world: those who divide the world into two kinds of people and those who don't. I frequently do.

One of my favourite kinds of these two kinds of people are Splitters and Groupers. Groups are very important to us as human beings. From an evolutionary standpoint, our very survival depends on this idea. A naked human alone in the wild is extremely vulnerable. But a group of humans, united and making tools, becomes the apex bio-competitor. Deep down in our DNA, we know that groups are an important key to our safety and survival in the world.

Groups are all about similarities and differences. Similarities help us separate members from non-members. Differences help us separate members from each other, thereby splitting the group into subgroups. I'm fascinated by patterns of similarities and differences. That's how I make sense of the world around me, by sorting out what's what and then choosing which to engage with and which to avoid.

Splitters and Groupers is an interesting way to see things, because every time we declare a group,

we create a new split, and every time we declare a split, we create new groups. Splitters and Groupers are truly the yin and yang of two kinds of people.

That's interesting, but what's the point? Well, this profoundly pedantic preliminary preponderance proves an apropos precursor to my proximate pending postulate. That being:

There are two kinds of programmers. Over time, I've developed a theory about how most programmers fit into one of two categories. I call it the 80/20 Model of Programmers. In my experience, approximately 80% of programmers prefer objective facts and figures and indisputable results. They love that computers do precisely what you tell them, behave consistently, and never lie or play politics. The other 20% are more subjectively focused and enjoy expressing themselves. They are always looking for an outlet and see the computer as a vast frontier, teeming with previously unknown creative possibilities. Infuse a nerd with ham, and you've got a 20%er.

An 80%er tends to see their computer as a refreshing alternative to dealing with people. If they weren't programmers, they might be accountants, fact-checkers, or actuaries. On the other hand, 20%ers tend to see their computer as an intricate and complex conduit, offering new ways to connect with people. If they weren't programmers, they would likely be artists or performers of one sort or another (and probably starving).

For more than a quarter century, I hopped all around the software industry. I've been an employee or contractor for more than two dozen companies doing all manner of software development. In all that time, this perception (and proportion) has remained solid. Everywhere



➤ Could you cope with an eleven-year-old kid telling you your game sucks? If so, you may be a game programmer.



^ Could you make a game capable of keeping this quintessentially 20th-century family entertained? If so, you may be a game programmer.

I worked there were mostly 80%ers and some 20%ers... with one glaring exception: the video game industry uniformly violates the 80/20 model. Game programmers are almost exclusively 20%ers, and Atari was populated by some of the most perverse 20%ers.

This isn't a coincidence. 80%ers have a hard time with games because they don't tolerate ambiguity well. Engineers in general (and 80%ers in particular) like things to be clearly defined and well thought out. Most software just needs to meet the technical specification, which is a pretty objective criterion. No-one says, "This word processor isn't nearly as exciting as my C++ compiler." If text box A pops up when I click button B, I know my program works, and I know I'm done. This is perfect for 80%ers.

PITTANCE AND PIZZA

A video game is different. A video game must still meet all its technical specifications, but it also has to be fun. This is a very subjective criterion. When you take people who spend their life chronically avoiding ambiguity and ask them to add "It's got to be fun" to their technical specification, well... What does fun mean? How do you define fun? How do you measure it? Can you describe in clear, unambiguous detail, a process which will guarantee fun? (Let me save you some time here, the answer is no).

You spend up to a year of your life pouring your heart and soul into the code, crafting and polishing each aspect and nuance from every angle you can imagine. All those days, nights, and weekends lead up to that one special moment when you go sit behind the two-way mirror and watch your work presented to the 'client'. And who sits in judgement of your work? Some eleven-year-old boy, getting paid a pittance and pizza to express his opinion.

"Risk is something most programmers can't abide"

After two minutes, he tosses the controller and issues his verdict: "This sucks!"

And you know what? He's right! How do you deal with that? Ego-less programming? I don't think so. 80%ers don't like that at all. These are issues that make it difficult for pure nerds to accept jobs creating video games. Though passionate about playing games, they'll leave the making to others.

Video game development is different from typical programming, and it requires a different approach. You've got to take responsibility. You have to put yourself on the line. Good games don't come from disengaged people; they come from neurotic oddballs who seek out this kind of emotional punishment. Why would anyone do this? Because 'the entertainment industry'. Because 20%ers need to have their work seen and appreciated. Because when

someone plays *your* game and loves it... that feels amazing! It's unbelievably gratifying!

But you can't get there without personal investment, which means taking some risk. And risk is something most programmers can't abide. Risk is the opposite of guaranteed reliability, which is what software is ultimately supposed to deliver. The big question, in the end, remains: can you reliably deliver a consistent functionality which in the final analysis generates a fun experience?

If you can honestly answer 'yes', then you are one of the kinds of people who might call themselves a video game programmer. ☺



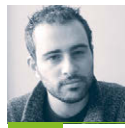
^ Howard Scott Warshaw at Atari in the early eighties. Possibly captured just before he jumped in a car to Reno.

Hard Drivin'

An interesting thing about classic game developers at Atari is that many of them really enjoy gambling. Betting on sports, games, horses, or themselves. When not in the lab, you can find us at horse tracks or in casinos. We used to have occasional scumbagathons. A scumbagathon occurs when a number of developers jump in a car (or cars) and head for Reno. Casinos were a mere three-and-a-half to five hours away (depending on who's driving). Every once in a while, some spontaneous stress relief was mandated, and when it was, we answered the call!

Land use planning for video game cities

Here's how urban planning can help
shape game city environments



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

A city is a place of intense, varied, and intertwined activities; activities that by definition, have to occur and be organised in the urban space. The space that cities reserve and assign to each of these activities is what urban planners and geographers commonly describe as 'land use', a concept at the very core of real-life urban planning. Land uses define where things can happen, and what sort of things occur at specific places, and are thus important when it comes to approaching imaginary and virtual cities, too. What's more, anyone who's ever played *SimCity*, and zoned an area as industrial, commercial, or residential, has already been exposed to land use planning.

Arranging land uses aims to achieve optimal financial, social, real estate, and environmental results. Most city plans, whether real or imaginary, intend to be as efficient and pleasant as possible, while land use planning,

in particular, is the process of designing and regulating space in an attempt to regulate activities, and prevent conflicts. Common goals of contemporary planning include productive efficiency, public health, environmental quality, minimisation of costs, a reduced exposure to pollutants, expressing ideology, and quality of life. However, what is considered optimal or even ideal isn't the same across all classes and social groups, and so fundamentally different planning logics can arise as a result.

USES OF LAND

Any attempt at urban planning for an imaginary city must involve land uses. We have to know where virtual people live, and where they work; where they shop, where they entertain themselves, where they bury their dead, and where their parks and places of assembly are. A typical land use plan will, therefore, have to include this sort of information, while employing colour to differentiate between uses.

Figure 1 is an example based on 19th century London where you can see a sketch for a small neighbourhood: red is for industry, yellow is residential, blue is for commercial- and leisure-related land, brown is for storage, sheds, and warehouses, and green is for gardens and open space.

The land use categories we tend to find in the modern city haven't changed much in the past century, are varied, and tend to differ between planning systems. Generally, though, we can speak of space reserved for residential purposes, for industry, for office space, and residence of all types, but also of



▲ **Figure 1: A land use planning example.**

Victorian Horror

Modern urban planning, and alongside it land use planning, were born during the 19th century to combat extreme and often deadly situations. In Victorian London for instance, there were no regulations separating heavily polluting industries from the residences of the poor, meaning that factories and homes were often located next to each other. Respiratory illnesses were thus commonplace, and this was a situation that didn't help much when it came to raising the average life expectancy of the working class.



parks, open spaces, transportation facilities, security services such as police and fire departments, hospitals, and social welfare. In a land use plan, we also delineate space for religion, warehouses, commerce (retailing and wholesale), light industry, education, farming, and even unused land. More specific uses such as slaughterhouses, greenbelts, monuments, archaeological sites, private gardens, or construction sites can also be required in some cases.

Districts across a city aren't commonly reserved for a single use, though homogeneous areas do exist. Suburbia, for example, is almost exclusively residential, and most industrial zones are predominantly productive, though most areas feature a mix of land uses.

Restaurants and shops are routinely found in residential areas. In general, the proximity of housing and polluting activities is avoided. These land uses are considered to be conflicting, and attempts to separate them – often via parks and greenbelts – have been numerous throughout recent history. It's also worth noting that certain uses (for example, a grocery store) have a local radius, whereas others (such as a courthouse) can have a metropolitan or even national radius. In other words, a grocery store only attracts users from its immediate environs, whereas a courthouse from a wider metropolitan area.

Many of the aforementioned land uses, residence being a prime example among them, can be found in cities and settlements across history. But as societies evolve, their needs, beliefs, and technologies also inevitably transform. Defence as a land use was pretty common for most city-states and smaller kingdoms before modernity, while residence and production routinely happened under one roof – for example, in a medieval guild house. Public voting places like the agora were mandatory in ancient Greek democracies,

necropolises defined the ancient Egyptian city, and I am certain that a fantasy or sci-fi settlement would probably require new or different land uses. Just as industry was eventually pushed out of cities, so could alchemists and their toxic magical fumes be pushed to the periphery – outside the walls – of a fantasy town.

VISUALISING LAND USE

A land use plan is used to describe how a place is organised and how it is to be constructed. A land use plan can thus be both descriptive and prescriptive. But like all plans, it can't fully describe the life that will take place on the

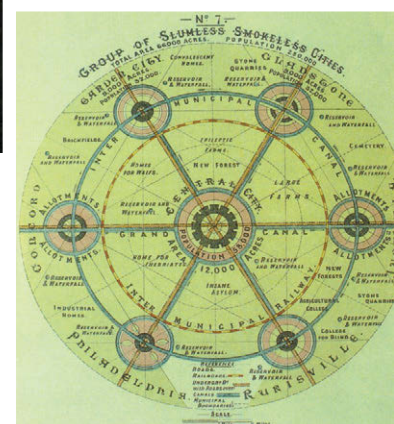
space it covers. It's one thing looking at a plan of American suburbia of the 1950s, and another to visit or live inside one of its houses. A plan can't show housewives without

access to the family automobile trapped inside these homes, or kids playing on the lawns, or recreate the smell of freshly cut grass.

Similarly, describing Hong Kong as a city of mainly mixed land use neighbourhoods doesn't prepare anyone for the sheer vibrancy of its packed streets, nor does it describe how colourful shops open out to the streets, how offices and bars climb up the floors, and how signs dominate the visual character of the city. Interestingly, Hong Kong's mix of land uses and activities also nourished the very unique, and very characteristic architectural form of the tong lau: the tenement building designed for both commercial and residential use.

Finally, a generalised land use description like, say, 'monument' or 'industry' can rarely conjure the history of the Colosseum or Chernobyl's nuclear power plant. A land use plan, then, creates a fundamental basis, which then has to be imbued with life and character. 🌐

“Land uses define where things can happen”



▲ There are many ways of planning land uses; here's Ebenezer Howard's classic 1902 plan for the 'Garden Cities of Tomorrow'.

▼ The tong lau of Hong Kong are an iconic example of mixed-use buildings. Photo by HenrySZTO.



Land Use and Transport

When designing transportation systems, and when trying to figure out why and how traffic patterns are generated, looking at (or, when appropriate, devising) a land use plan is always enlightening. Land use defines most aspects of both vehicular and pedestrian traffic. People travel from their homes to their jobs, creating rush hours; bars and restaurants draw in crowds at night; a mixed-use area is friendlier for walkers, and doesn't require the constant use of a car.

Make your own proc gen dungeons

Create dungeons filled with loot and monsters in seconds with procedural generation



AUTHOR
MAC BOWLEY

Mac Bowley is an educator, gamer, and programmer. A keen student of game development, he can often be found messing around with a new game engine or playing around with Unity and C#.



Follow this guide, and you'll have your own proc-gen dungeon up and running in no time.

If your game relies on large amounts of loot and hordes of enemies, then some randomness in the level design can keep your game feeling fresh and keep players on their toes. Games like *Diablo* thrive on their variety of environments and dungeons for players to explore in the search of better loot. *Diablo* makes great use of procedural generation to create its dungeons, which are formed each time the player reaches a new location.

I'm using Pygame and Python to create our dungeon generator, but I'm going to keep the explanations as general as possible so you can apply these principles to whichever development environments you enjoy working in. The point of this guide is to show you how you can implement procedural generation, and some of the techniques involved.

I'm going to create a tiled level stored in arrays that can be drawn by any script. I've made this choice because it allows me to use my procedural generation in whatever script I am running, so I can keep it separate from any game code I'm writing and just use it to populate the `Map` array my game uses.



```
# Importing modules
import secrets # a randomness library
import math # for the math
from collections import namedtuple
from pygame import Rect

# Map variables
TILE_SIZE = 32 # Dimensions of the tiles
TILE_ACROSS = 32
TILE_DOWN = 24

# Pygame variables
# Setting the width and height of the window
WIDTH = TILE_SIZE * TILE_ACROSS
HEIGHT = TILE_SIZE * TILE_DOWN

def create_map():
```



```
Map = []
for y in range(TILE_DOWN):
    row = []
    for x in range(TILE_ACROSS):
        row.append(0)
    Map.append(row)
return Map
```

This is the start of my project, and it establishes our first set of rules. Using variables as much as possible to control the constants in our system makes it easy to change them later. The tiles I'll be working with are 32×32 pixels, but if I change them and want 64×64 tiles, I only need to change one variable, and my whole map will adjust. The other variables are dependent on the tile size, but the **WIDTH** and **HEIGHT** variables will dictate how big the map is in pixels.

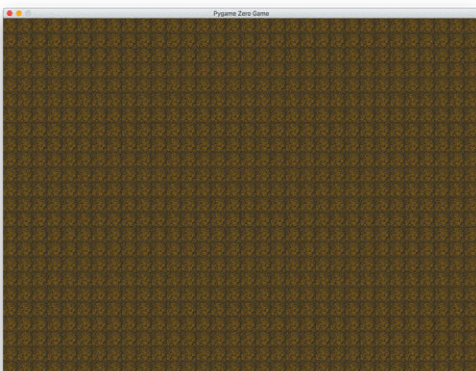
“Diablo makes great use of procedural generation to create its dungeons”

The map is a two-dimensional array, which matches the **TILE_ACROSS** and **TILE_DOWN** dimensions. To start off with, it will be populated with entirely zeros. When I draw my map, 0 will identify a dirt tile, which means my map currently looks like **Figure 1**. It looks a bit boring, so let's create some rooms next.

CREATE A ROOM

To create rooms (see **Figure 2**), we first need to designate some rules. When you're creating things through procedural generation, you still need to design, but you're creating a set of rules

▼ **Figure 1:** An empty map, entirely made of dirt tiles.



rather than getting tangled in the nitty-gritty of a level's layout. The idea is that your rules, if they're followed, should guide the randomness in a way that means the levels have the look and feel you're after.

The initial rules for my rooms are:

1. Choose a random width and height
2. Choose a random position

These are my starters, but they aren't finished yet, and there are some flaws that need to be addressed.

We don't want a room to have a width or height of 0, or even 1. A room one tile high or wide would be more of a corridor. I also don't want a room the same size as my world, or even half of it. This means I need to set maximum and minimum room sizes, which I can accomplish through variables and an equation for randomising the width and height.

I should also put some restrictions on the positions my rooms can appear in. I want there to be a border of at least a tile around the edge of my map.

The final rules for the creation of my rooms are:

1. Choose a random width and height but...
 - + it shouldn't be larger than 7
 - + nor less than 2
2. Choose a random position on the map.
3. The rooms' details should be stored so they can be used later on

```
MAX_ROOM_SIZE = 7
MIN_ROOM_SIZE = 3
ROOM_PADDING = 2

MAP_BORDER = 1

NUMBER_OF_ROOMS = 12

Room = namedtuple('Room', ['width', 'height',
                             'pos_x', 'pos_y'])

def generate_room():
    rm_width = MIN_ROOM_SIZE + secrets.
    randbelow(MAX_ROOM_SIZE - MIN_ROOM_SIZE + 1)
    rm_height = MIN_ROOM_SIZE + secrets.
    randbelow(MAX_ROOM_SIZE - MIN_ROOM_SIZE + 1)

    rm_pos_x = secrets.randbelow(TILE_ACROSS -
```

PYGAME ZERO

To follow along with our guide, you'll need to install Pygame Zero. You can find full instructions at wfmag.cc/pgzero.



^ **Figure 2:** A map with a single room generated.

```

rm_width)
    if rm_pos_x < MAP_BORDER:
        rm_pos_x = MAP_BORDER
    rm_pos_y = secrets.randbelow(TILE_DOWN -
rm_height)
    if rm_pos_y < MAP_BORDER:
        rm_pos_y = MAP_BORDER

    return Room(rm_width, rm_height, rm_pos_x,
rm_pos_y)

```

```

height*32))
    rm_2 = Rect((padded_pos_x*32, padded_
pos_y*32),
                (padded_width*32, padded_
height*32))
    if rm_1.colliderect(rm_2):
        intersect = True
    Rooms.append([rm, {"NORTH": None, "SOUTH":
None, "EAST": None, "WEST": None}])
    return Rooms

```

GENERATE MULTIPLE ROOMS

To fully populate the rooms on my map, I'll use another variable to decide how many there'll be. I've been working with ten or so rooms, but feel free to mess with these variables to suit your needs. The main challenge with this stage is making sure the rooms don't overlap.

Using a named tuple to store the information about the rooms I'm generating will be essential for the work I'll do later, but it's also useful to make sure rooms don't overlap. To do this, I've created another function called `create_rooms()`. This will run a loop and generate an array of rooms. It calls the `generate_rooms` function to create rooms and then check whether they overlap with other rooms already generated. If the room does overlap, it's rejected and a new one is created.

```

ROOM_PADDING = 2

NUMBER_OF_ROOMS = 12

def create_rooms():
    Rooms = []
    for i in range(NUMBER_OF_ROOMS):
        intersect = True
        while intersect:
            intersect = False
            rm = generate_room()
            for other in Rooms:
                room = other[0]
                padded_pos_x = rm.pos_x - 1
                padded_pos_y = rm.pos_y - 1
                padded_width = rm.width + 3
                padded_height = rm.height + 3
                rm_1 = Rect((room.pos_x*32, room.
pos_y*32),
                            (room.width*32, room.

```

I'm using the `Rect` objects in Pygame to test for collisions between rooms. I've set a `ROOM_PADDING` variable which makes sure there's at least one square either side of my rooms. I create two rectangles that represent the rooms, and ask Pygame to check if they collide. This is done recursively, and checks all the rooms that have already been created. I've also updated the `create_map()` function to draw the rooms into the `Map` array when they've been generated and tested.

```

def create_map():
    Map = []
    for y in range(TILE_DOWN):
        row = []
        for x in range(TILE_ACROSS):
            row.append(0)
        Map.append(row)
    Rooms = create_rooms()
    for rm in Rooms:
        for y in range(rm[0].height):
            for x in range(rm[0].width):
                Map[rm[0].pos_y+y][rm[0].pos_x+x] = 1
    return Map

```

With that, my map has been populated with rooms (see **Figure 3**).

CREATING CORRIDORS

Now that I have rooms in my level, let's connect them with corridors. To do that, I first have to think about some rules. A small caveat here: there are many ways to do procedural generation, and there are lots of ways of connecting your dungeon together. As I go through the next bit, I'll point out where I could have used another method to connect the rooms, and shed some light as to why I chose to do it the way I have.



^ **Figure 3:** Our map's now populated with several rooms.

I'm creating a tile-based game, and so my options for corridors are already limited. I can't have curved corridors (thank goodness – curve maths is tough), and to be honest, diagonal corridors don't tend to look that good. This means my corridors will be straight, which ties in well with my current map style. Next, I have to decide how I want the rooms to connect.

Each room on my map could be connected to each of the others, but I suspect this would leave my map feeling cluttered. To keep it simple, I'm going to limit the connections to four possible connections: North, East, South, and West. These are potential connections, not a guarantee: I would like some degree of randomness in my connections.

The rules for my corridors will be:

1. Each room can have a maximum of four connections
2. They can go in one of these four directions: North, East, South, West
3. Before building the connection, there will be a chance to skip it

“Now that I have rooms in my level, let's connect them with corridors”

CHOOSING THE CONNECTIONS

This brings me to the algorithmic section of this task: how to choose which rooms to connect to. Saving the information about each room will help me here. Let's take a look at the following room and see how we can find connections for it.

With as many rooms as I have, there's never going to be a need for corners in the corridors. I should be able to find a connection in any of the directions using only a straight line.

The best way to do this is to test the other rooms to see if they meet two criteria:

1. It's in the given direction in relation to the current room (i.e., they have a lower y if it's a northern connection)
2. It overlaps with the current room on either the x- or y-axis (i.e., for a North connection they should overlap in the x-axis)

Let's take another look at the map from earlier (Figure 4). The rooms highlighted in orange all meet the first criteria for a North connection, but not the second. The range of x coordinates covered by the rooms doesn't overlap with the

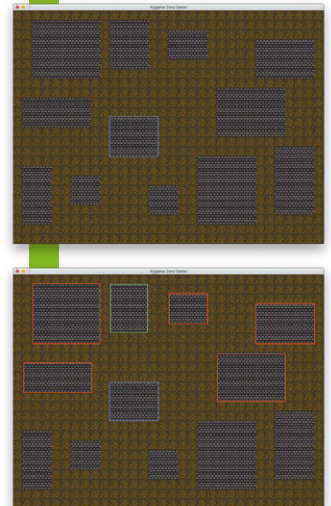
range of the room we want to add a connection to. The room highlighted in green however, does meet both criteria. I can test each room's x range for both North and South, but for East and West, the overlap will be in the y range covered by the room. Here's the code to choose the corridors:

```
def create_corridors(rm, Rooms):
    candidates = {"NORTH": None, "SOUTH": None,
                  "EAST": None, "WEST": None}
    for other in Rooms:
        if other[0] != rm[0]:
            current_room = rm[0]
            other_room = other[0]
            left_marker = max(current_room.pos_x,
                              other_room.pos_x)
            right_marker = min(current_room.pos_x +
                               current_room.width, other_room.pos_x + other_
                               room.width)
            horizontal_overlap = list(range(left_
            marker, right_marker))
            if len(horizontal_overlap) > 0:
                vertical_corridors(candidates, other,
                                   rm, horizontal_overlap)
            top_marker = max(current_room.pos_y,
                              other_room.pos_y)
            bottom_marker = min(current_room.pos_y +
                                 current_room.height, other_room.pos_y + other_
                                 room.height)
            vertical_overlap = list(range(top_
            marker, bottom_marker))
            if len(vertical_overlap) > 0:
                horizontal_corridors(candidates,
                                     other, rm, vertical_overlap)
    return candidates
```

If there are multiple rooms that overlap and meet the first criteria, then a decision has to be made. There are lots of ways to do this: shortest path, best angle, and so on. I'm going to go for the shortest path, so if there are two options that meet the criteria, I'll select the one closest to the direction we're currently working on. Here are the functions to create horizontal and vertical corridors:

```
def vertical_corridors(candidates, other, rm,
                      horizontal_overlap):
    current_room = rm[0]
    current_connections = rm[1]
    other_room = other[0]
```

▼ Figure 4: Another procedurally generated map, with a room highlighted in the centre.





DUNGEON SMASH

To demonstrate how the map generator could work in a finished game, we've made *Dungeon Smash* – a tiny adventure with loot to collect and monsters to fight. Like the rest of the files for this guide, you'll find it on our Github at wfmag.cc/wfmag37. Use the arrow keys to move your character around, and the A key to attack.

```

other_connections = other[1]
if current_room.pos_y > other_room.pos_y
and other_connections["SOUTH"] == None and
current_connections["NORTH"] != 0:
    connector = candidates["NORTH"]
    if connector == None:
        candidates["NORTH"] = (other,
horizontal_overlap)
        other_connections["SOUTH"] = 0
    else:
        if other_room.pos_y + other_room.height
> connector[0][0].pos_y + connector[0][0].
height:
            connector[0][1]["SOUTH"] = None
            candidates["NORTH"] = (other,
horizontal_overlap)
            other_connections["SOUTH"] = 0
        if current_room.pos_y < other_room.pos_y
and other_connections["NORTH"] == None and
current_connections["SOUTH"] != 0:
            connector = candidates["SOUTH"]
            if connector == None:
                candidates["SOUTH"] = (other,
horizontal_overlap)
                other_connections["NORTH"] = 0
            else:
                if other_room.pos_y < connector[0][0].
pos_y:
                    connector[0][1]["NORTH"] = None
                    candidates["SOUTH"] = (other,
horizontal_overlap)
                    other_connections["NORTH"] = 0

def horizontal_corridors(candidates, other,
rm, vertical_overlap):
    current_room = rm[0]
    current_connections = rm[1]
    other_room = other[0]
    other_connections = other[1]
    if current_room.pos_x > other_room.pos_x and
other_connections["EAST"] == None and current_
connections["WEST"] != 0:
        connector = candidates["WEST"]
        if connector == None:
            candidates["WEST"] = (other, vertical_
overlap)
            other_connections["EAST"] = 0
        else:
            if other_room.pos_x < connector[0][0].

```

```

pos_x:
        connector[0][1]["EAST"] = None
        candidates["WEST"] = (other, vertical_
overlap)
        other_connections["EAST"] = 0
    if current_room.pos_x < other_room.pos_x and
other_connections["WEST"] == None and current_
connections["EAST"] != 0:
        connector = candidates["EAST"]
        if connector == None:
            candidates["EAST"] = (other, vertical_
overlap)
            other_connections["WEST"] = 0
        else:
            if other_room.pos_x + other_room.width <
connector[0][0].pos_x + connector[0][0].width:
                connector[0][1]["WEST"] = None
                candidates["EAST"] = (other, vertical_
overlap)
            other_connections["WEST"] = 0

```

The final consideration is where to actually put my corridors to connect the rooms, and I already have the information I need to do this. When comparing the overlaps, I'm left with a range of overlapping values. You could randomly choose one of the values, which will correspond to a straight line in the x- or y-axis that connects both rooms. I decided to grab the middle value of the range and place the corridor there. I like randomness, but I found this method looked more dungeon-y to me. This is an important thing to remember about procedural generation: it's all subjective and about how you want your level to look and feel.

This will be done in the `create_map()` function, and at this point, it's just a path-drawing algorithm. There's a little fiddling around with directions on drawing and start and end positions for the corridors.

```

def create_map():
    Map = []
    for y in range(TILE_DOWN):
        row = []
        for x in range(TILE_ACROSS):
            row.append(0)
        Map.append(row)
    Rooms = create_rooms()
    for rm in Rooms:
        for y in range(rm[0].height):

```

```

for x in range(rm[0].width):
    Map[rm[0].pos_y+y][rm[0].pos_x+x] = 1
corridors = create_corridors(rm, Rooms)
#Create the corridors on the map array
for key, value in corridors.items():
    if len(corridors.items()) > 1:
        skip = secrets.randbelow(100) #
Chance to skip drawing this corridor
    else:
        skip = 100
    if value is not None and value is not 0
and skip > 10:
        dir = [0, 0]
        start_pos = [rm[0].pos_x, rm[0].pos_y]
        end_pos = [value[0][0].pos_x, value[0]
[0].pos_y]
        mid_overlap = value[1]
[0].pos_y]
        [len(value[1])//2]
        if key == "NORTH":
            dir[1] = -1
            start_pos[0] = mid_overlap
            start_pos[1] -= 1
            end_pos[0] = mid_overlap
            end_pos[1] += value[0][0].height
        elif key == "SOUTH":
            dir[1] = 1
            start_pos[0] = mid_overlap
            start_pos[1] += rm[0].height
            end_pos[0] = mid_overlap
            end_pos[1] -= 1
        elif key == "EAST":
            dir[0] = 1
            start_pos[0] += rm[0].width
            start_pos[1] = mid_overlap
            end_pos[0] -= 1
            end_pos[1] = mid_overlap
        elif key == "WEST":
            dir[0] = -1
            start_pos[0] -= 1
            start_pos[1] = mid_overlap
            end_pos[0] += value[0][0].width
            end_pos[1] = mid_overlap
        Map[start_pos[1]][start_pos[0]] = 2
        Map[end_pos[1]][end_pos[0]] = 2
        distance = (start_pos[0] - end_pos[0])
+ (start_pos[1] - end_pos[1])
        next_pos = start_pos
        for i in range(abs(distance)):
            next_pos = [next_pos[0] + dir[0],

```

```

next_pos[1] + dir[1]]
        Map[next_pos[1]][next_pos[0]] = 2
    return Map

```

With that, we have generated a random dungeon for our players to try and loot (see **Figure 5**)!

“I need to decide on some rules for placing these enemies and the boss”

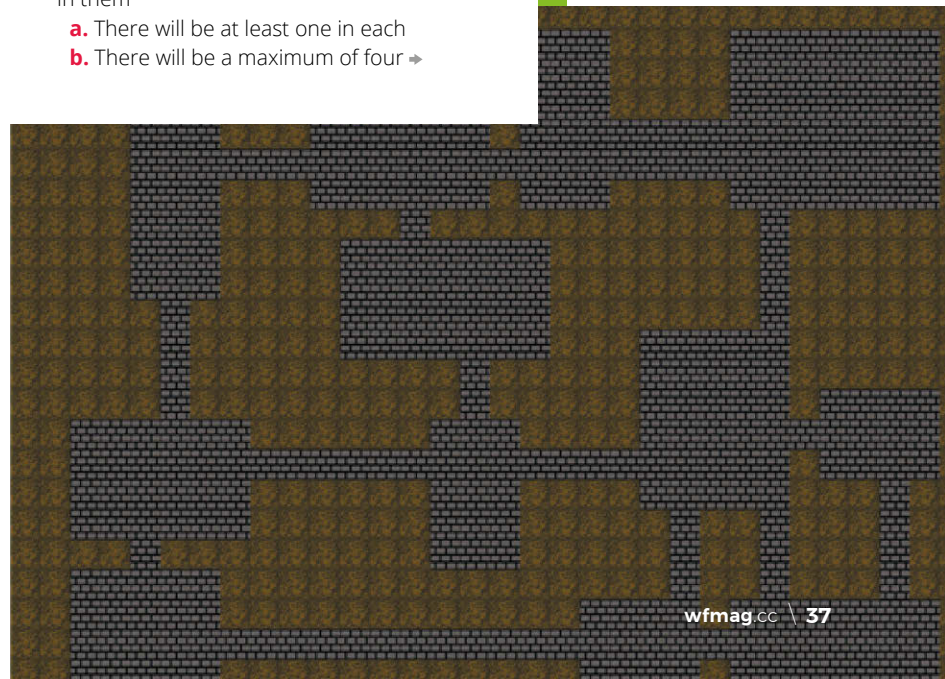
ADDING ENEMIES AND LOOT

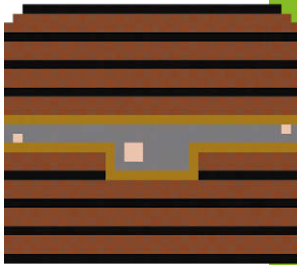
The dungeon layout is now complete, so it's a shame that there's nothing for our player to fight or loot – let's fix that. I'll start with enemies and then add some chests to contain loot. I'll create two types of enemies: henchmen and bosses. Henchmen will be my run-of-the-mill enemies, and there will be one boss per dungeon level. To start with, I need to decide on some rules for placing these enemies and the boss.

Rules for enemy placement:

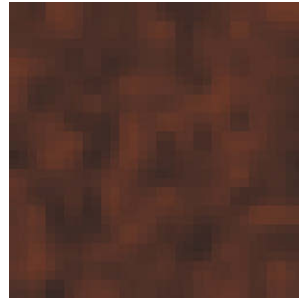
1. The boss will go in the largest room of the dungeon
 - a. The boss will go in the centre of the room
 - b. Henchmen will flank him
 - i. At least one
 - ii. No more than three
2. The smallest room will be used to spawn the player and have no enemies
3. The rest of the rooms will have henchmen in them
 - a. There will be at least one in each
 - b. There will be a maximum of four →

▼ **Figure 5:** Our dungeon is now constructed with corridors connecting each room.





^ I'll be using a mixture of tiles I've made myself using piskelapp.com and characters from opengameart.org.



These rules give me some structure while keeping the randomness in check. Now I have to implement it.

Throughout this project, I've used numbers to represent different tiles; I'll do the same thing for my enemies and loot. I'll use the following numbers to represent the different enemy types:

- Henchman** - 4
- Boss** - 5
- Chest** - 6
- Player** - 7

When I'm running a game, I can search for these numbers in my map and spawn enemy actors in those positions.

IMPLEMENTING THE HORDE

The enemies will need to be placed before the map is returned at the end of the `create_map()` function; and to avoid cluttering up that section of code, I'm going to use functions for each part of the process. First, I make a function to find an empty position in a given room on the map:

```
def find_position(Room, Map):
    x_pos = Room.pos_x + secrets.randbelow(Room.
width - 1)
    y_pos = Room.pos_y + secrets.randbelow(Room.
height - 1)
    while Map[x_pos][y_pos] != 1:
        x_pos = Room.pos_x + secrets.
randbelow(Room.width)
        y_pos = Room.pos_y + secrets.
randbelow(Room.height)
    return x_pos, y_pos
```

The `find_position` function takes a room and randomly selects a position in it; that position is then checked using the map. I only want to place something in an empty room tile (represented as a 1 on my map), so if the position is not a 1, the function regenerates the x and y position and checks again.

POPULATING THE ROOMS

My rooms will be populated using a function that will itself call functions for each of the stages and rules I laid out above:

```
def populateRooms(Rooms, Map):
    biggestArea = 0
    biggestRoom = ""
    smallestArea = math.inf
    smallestRoom = ""

    for rm in Rooms:
        area = rm[0].width * rm[0].height
        if area >= biggestArea:
            biggestArea = area
            biggestRoom = rm
        if area <= smallestArea:
            smallestArea = area
            smallestRoom = rm
    rooms_to_populate = Rooms
    rooms_to_populate.remove(biggestRoom)
    rooms_to_populate.remove(smallestRoom)

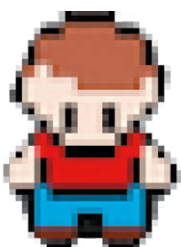
    Map = placeEnemies(rooms_to_populate, Map)

    boss_room = biggestRoom[0]
    Map = placeBoss(boss_room, Map)

    player_spawn = smallestRoom[0]
    Map = placePlayer(player_spawn, Map)

    return Map
```

The above code will first find the biggest and smallest rooms on my `Map`, and stores them in variables to be used later. Once they've been found, according to my rules, all the other rooms need to be populated. I made a new list of rooms – called `rooms_to_populate` – and placed all the rooms into it, before removing the largest and smallest rooms. Those rooms remaining in the list are passed to the `placeEnemies` function, which



loops through each of them, generates a random number of enemies, and places them.

The `placeEnemies` function looks like this:

```
def placeEnemies(Rooms, Map):
    for i in range(NUMBER_OF_ROOMS - LOOT_
ROOMS):
        room = secrets.choice(Rooms)
        amount_of_enemies = secrets.randbelow(4)
        for enemy in range(amount_of_enemies):
            x_pos, y_pos = find_position(room[0],
Map)
            Map[x_pos][y_pos] = 4
            chest_x_pos, chest_y_pos = find_
position(room[0], Map)
            Map[chest_x_pos][chest_y_pos] = 6
            Rooms.remove(room)
    return Map
```

The function also places a chest in each room for the player to loot. There's some room for adjustment if you want to use this technique; I wanted one enemy in each room, but you may want to leave some rooms empty to add suspense. It all depends on the feel and style of game you want to create.

The last two functions needed to populate the rooms are `placeBoss` and `placePlayer`, which are self-explanatory. The `placeBoss` function takes the largest room and the `Map` as parameters,

“I've generated a dungeon, fully equipped with baddies and loot for the player to fight and find”

generates a random number of henchmen (at least one and at most four), and places the boss in the centre of the room along with two chests. The other, `placePlayer`, takes the smallest room and places the player somewhere inside it.

```
def placeBoss(boss_room, Map):
    x_pos = boss_room.pos_x + (boss_room.
width//2)
    y_pos = boss_room.pos_y + (boss_room.
height//2)

    Map[x_pos][y_pos] = 5
```

```
boss_enemies = secrets.randbelow(4) + 1

for henchman in range(boss_enemies):
    hench_x, hench_y = find_position(boss_
room, Map)
    Map[hench_x][hench_y] = 4

for chest in range(2):
    chest_x_pos, chest_y_pos = find_
position(boss_room, Map)
    Map[chest_x_pos][chest_y_pos] = 6

return Map

def placePlayer(room, Map):
    player_x, player_y = find_position(room,
Map)
    Map[player_x][player_y] = 7

return Map
```

With those two functions done, only one change remains to finish my map generator: calling the function `populateRooms` in my `create_map()` function:

```
def create_map():
    ...
    Map = populateRooms(Rooms, Map)
    return Map
```

I've now generated a dungeon, fully equipped with baddies and loot for the player to fight and find! 🗝

✓ The finished dungeon, lightly sprinkled with loot and enemies, and with the boss lurking in the centre.





✓ Atari's cabinet featured a thrust control, two buttons for rotating, and an abort button in case it all went horribly wrong.



▲ Atari released the first *Lunar Lander* arcade game in 1979.

◀ The player must manoeuvre the lander to a landing site and carefully descend to the surface.

Source Code

Code a homage to Lunar Lander



AUTHOR
MARK VANSTONE

Shoot for the moon in Mark's version of an Atari hit

First released in 1979 by Atari, *Lunar Lander* was based on a concept created a decade earlier. The original 1969 game (actually called *Lunar*) was a text-based affair that involved controlling a landing module's thrust to guide it safely down to the lunar surface; a later iteration, *Moonlander*, created a more visual iteration of the same idea on the DEC VT50 graphics terminal.

Given that it appeared at the height of the late-seventies arcade boom, though, it was Atari's coin-op that became the most recognisable version of *Lunar Lander*, arriving just after the tenth anniversary of the Apollo 11 moon landing. Again, the aim of the game was to use rotation and thrust controls to guide your craft, and gently set it down on a suitably flat platform. The game required efficient control of the lander, and extra points were awarded for parking successfully on more challenging areas of the landscape.

The arcade cabinet was originally going to feature a normal joystick, but this was changed to a double stalked up-down lever

providing variable levels of thrust. The player had to land the craft against the clock with a finite amount of fuel with the Altitude, Horizontal Speed, and Vertical Speed readouts at the top of the screen as a guide. Four levels of difficulty were built into the game, with adjustments to landing controls and landing areas.

To write a game like *Lunar Lander* with Pygame Zero, we can replace the vector graphics with a nice pre-drawn static background and use that as a collision detection mechanism and altitude meter. If our background is just black where the Lander can fly and a different colour anywhere the landscape is, then we can test pixels using the Pygame function `image.get_at()` to see if the lander has landed. We can also test a line of pixels from the Lander down the Y-axis until we hit the landscape, which will give us the lander's altitude.

The rotation controls of the lander are quite simple, as we can capture the left and right arrow keys and increase or decrease the rotation of the lander; however, when thrust

is applied (by pressing the up arrow) things get a little more complicated. We need to remember which direction the thrust came from so that the craft will continue to move in that direction even if it is rotated, so we have a direction property attached to our lander object. A little gravity is applied to the position of the lander, and then we just need a little bit of trigonometry to work out the movement of the lander based on its speed and direction of travel.

To judge if the lander has been landed safely or rammed into the lunar surface, we look at the downward speed and angle of the craft as it reaches an altitude of 1. If the speed is sufficiently slow and the angle is near vertical, then we trigger the landed message, and the game ends. If the lander reaches zero altitude without these conditions met, then we register a crash. Other elements that can be added to this sample are things like a limited fuel gauge and variable difficulty levels. You might even try adding the sounds of the rocket booster noise featured on the original arcade game. 🎮

Download
the code
from GitHub:
[wfmag.cc/
wfmag37](https://wfmag.cc/wfmag37)

Lunar Lander in Python

Here's Mark's code for a simple, modern take on *Lunar Lander*. To get it running on your system, you'll need to install Pygame Zero – full instructions are available at wfmag.cc/pgzero.

```

import math
from pygame import image, Color
import time
start_time = time.time()
backgroundImage = image.load('images/background.png')
lander = Actor('lander', (50, 30))
lander.angle = lander.direction = -80
lander.thrust = 0.5
gravity = 0.8
lander.burn = speedDown = gameState = gameTime = 0

def draw():
    global gameTime
    screen.blit('background', (0, 0))
    screen.blit('space', (0, 0))
    r = lander.angle
    if(lander.burn > 0):
        lander.image = "landerburn"
    else:
        lander.image = "lander"
    lander.angle = r
    lander.draw()
    if gameState == 0:
        gameTime = int(time.time() - start_time)
        screen.draw.text("Altitude : "+ str(getAlt()),
            topleft=(650, 10), owidth=0.5, ocolor=(255, 0, 0),
            color=(255, 255, 0), fontsize=25)
        screen.draw.text("Time : "+ str(gameTime), topleft=(40,
            10), owidth=0.5, ocolor=(255, 0, 0), color=(255, 255, 0),
            fontsize=25)
        if gameState == 2:
            screen.draw.text("Congratulations \nThe Eagle Has
            Landed", center=(400, 50), owidth=0.5, ocolor=(255, 0, 0),
            color=(255, 255, 0), fontsize=35)
        if gameState == 1:
            screen.draw.text("Crashed", center=(400, 50),
            owidth=0.5, ocolor=(255, 0, 0), color=(255, 255, 0), fontsize=35)

def update():
    global gameState, speedDown
    if gameState == 0:
        if keyboard.up:
            lander.thrust = limit(lander.thrust+0.01, 0, 1)
            changeDirection()
            lander.burn = 1
        if keyboard.left: lander.angle += 1
        if keyboard.right: lander.angle -= 1
        oldPos = lander.center
        lander.y += gravity
        newPos = calcNewXY(lander.center, lander.thrust, math.
        radians(90-lander.direction))
        lander.center = newPos

```

```

        speedDown = newPos[1] - oldPos[1]
        lander.thrust = limit(lander.thrust-0.001, 0, 1)
        lander.burn = limit(lander.burn-0.05, 0, 1)
        if speedDown < 0.2 and getAlt() == 1 and lander.angle >
        -5 and lander.angle < 5:
            gameState = 2
        if getAlt() == 0:
            gameState = 1

def changeDirection():
    if lander.direction > lander.angle: lander.direction -= 1
    if lander.direction < lander.angle: lander.direction += 1

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

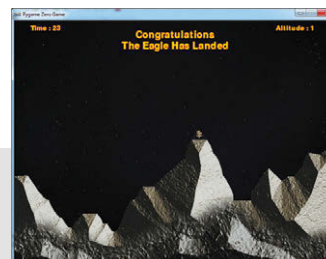
def calcNewXY(xy, speed, ang):
    newx = xy[0] - (speed*math.cos(ang))
    newy = xy[1] - (speed*math.sin(ang))
    return newx, newy

def getAlt():
    testY = lander.y+8
    height = 0;
    while testPixel((int(lander.x), int(testY))) ==
    Color('black') and height < 600:
        testY += 1
        height += 1
    return height

def testPixel(xy):
    if xy[0] >= 0 and xy[0] < 800 and xy[1] >= 0 and xy[1] <
    600:
        return
    backgroundImage.get_at(xy)
    else:
        return
    Color('black')

```

> Our homage to the classic *Lunar Lander*. Can you land without causing millions of dollars' worth of damage?



ENGAGE

The direction of thrust could be done in several ways. In this case, we've kept it simple, with one directional value which gradually moves in a new direction when an alternative thrust is applied. You may want to try making an X- and Y-axis direction calculation for thrust so that values are a combination of the two dimensions. You could also add joystick control to provide variable thrust input.

Learn Unity for free

Unity Learn Premium is free for a limited time: get on it

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

✓ You can use Unity to make games of all kinds, including, er, one about a blue block running away from a gang of green blocks.

It's not a great time for the world as a whole right now, so you have to throw a pile of thanks at the likes of Unity for offering up its Unity Learn Premium package completely free of charge for three months. The offer started back in March and is valid until 20 June, so there's still time to get in on it before it all shuts down back behind its really rather reasonable \$15 (£12) a month payroll.

What do you get with Learn Premium? Why, all the access to the free Unity Learn platform, of course, which includes a plethora of learning resources, tutorials, projects to work on, and more. Beyond that, you get access to *more* of all that good stuff, alongside three key elements:

- Learn Live: Interact with Unity Certified Instructors and an engaged peer community
- A growing library of beginner to advanced on-demand tutorials, hands-on projects, and in-depth courses updated for the latest Unity release
- Access to high-quality learning resources on partner platforms like Coursera, Udemy, Pluralsight, and Pathstream all in one subscription

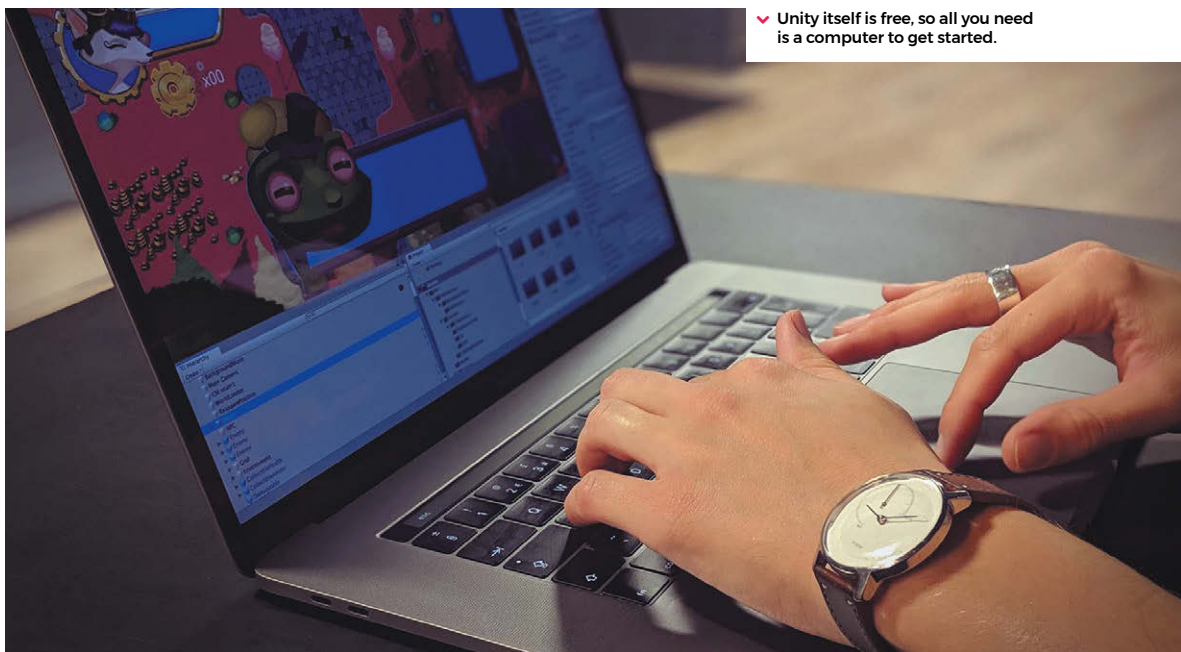
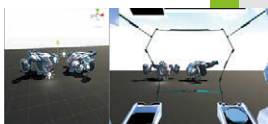
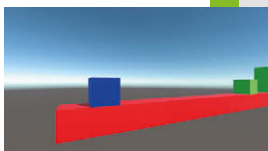
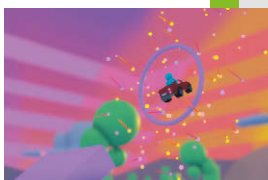
What this amounts to is a collection of fantastic learning resources, some projects to kickstart your creative ambitions, and more than enough

free freeness to get you going and keep you focused along the way, *backed up* by improved access to people who know what they're talking about and who can give you tips and pointers on your projects. And to hammer home the point, it's all for free, for the time being.

Reviews for Unity Learn Premium have been mixed when it comes to certain elements – namely the availability of live support/advice – so it's fair to point that out. That said, this is a free three-month trial for those who want to give it a go: more than enough time to make sure it's the sort of platform you're able to get something useful out of and, of course, enough time for Unity to convince you to keep on paying for it once the free period ends. Remember, if you don't want to carry it on, cancel your membership before it runs out. Nobody likes those accidental additional charges, and Unity doesn't offer refunds.

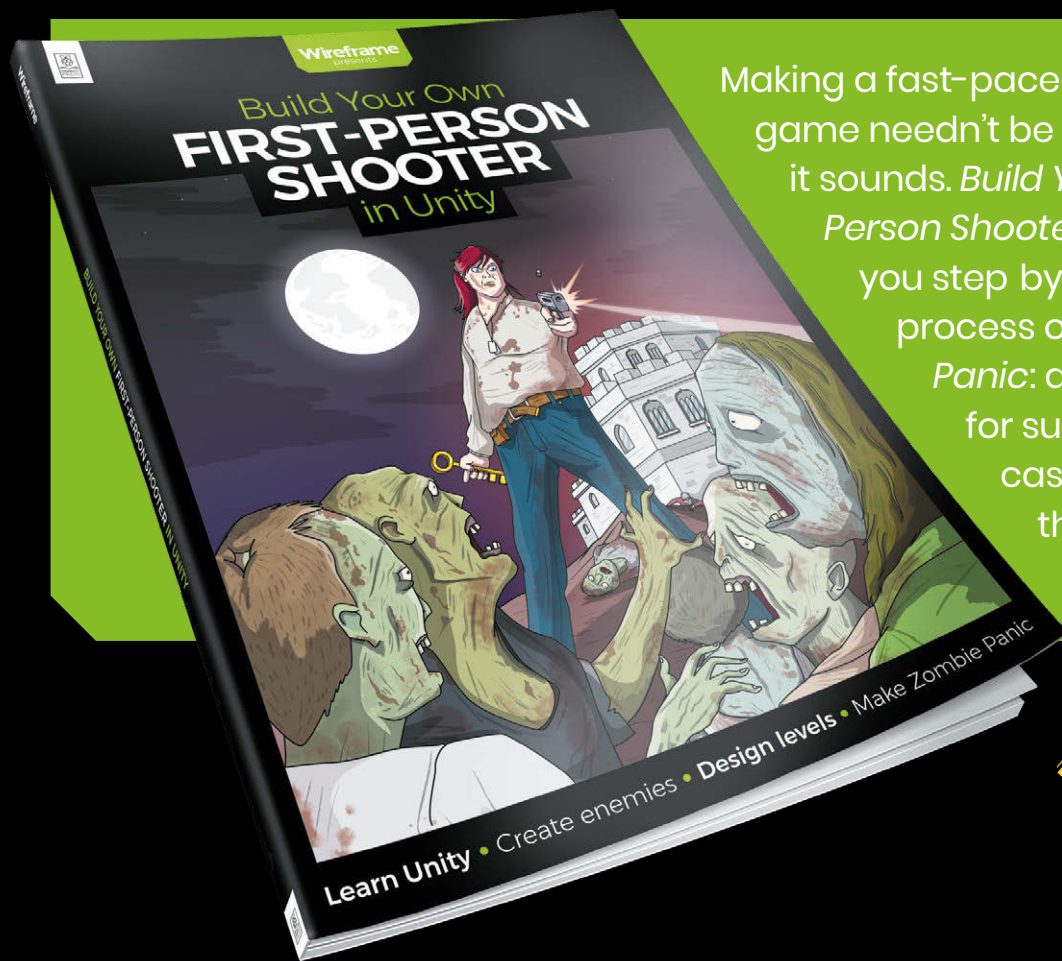
You can find Unity's offer in full over here: wfmag.cc/UniFree, with a lot more information, suggestions, and other such delights ready to take you on your game-making odyssey. Or your 'making a rough platform game in an hour or so' short trip, however you want to go about it. All you need is a Unity ID – or to sign up with one – to get things off the ground.

✓ Unity itself is free, so all you need is a computer to get started.



Build Your Own FIRST-PERSON SHOOTER in Unity

Making a fast-paced 3D action game needn't be as daunting as it sounds. *Build Your Own First-Person Shooter in Unity* will take you step by step through the process of making *Zombie Panic*: a frenetic battle for survival inside a castle heaving with the undead.



IN THE PROCESS, YOU'LL DISCOVER HOW TO:

Set up and use the free software you'll need

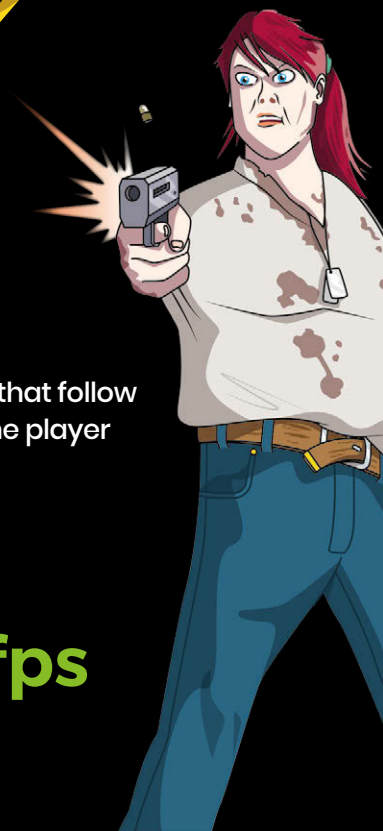
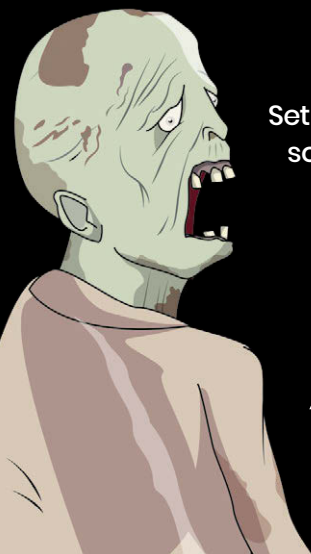
Create and texture 3D character models

Make enemies that follow and attack the player

Design a level with locked doors and keys

Extend your game further, with tips from experts

Available now: wfmag.cc/fps



BACK TO THE WASTELAND

THE POST-APOCALYPTIC, ISOMETRIC RPG RETURNS

We speak to a group of developers about the influence of *Fallout* and the re-emergence of the post-apocalyptic RPG

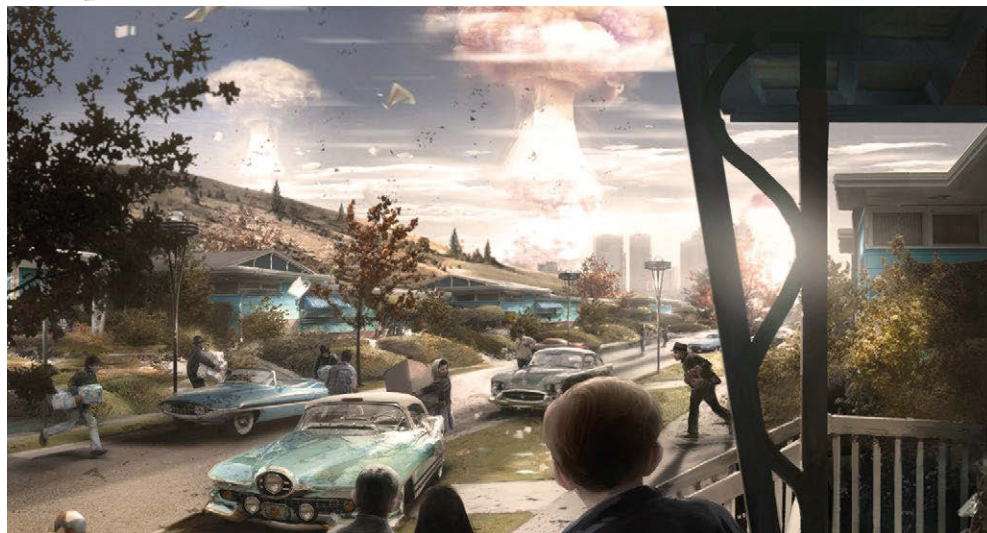
Released in 1997, post-nuclear RPG *Fallout* stands as one of the most influential games ever made. Taking place in the aftermath of a nuclear apocalypse, the game casts players as a survivor inhabiting a vault – an underground fallout shelter developed before the bombs fell. On discovering that your water purification system is broken, you're forced to leave your home and explore the perilous New Californian desert. Together with its 1988 predecessor, *Wasteland*, *Fallout* set the template for post-apocalyptic fiction in games, and has since led to a number of sequels and spin-offs, including a controversial foray into the live service genre with *Fallout 76*.

Today, *Fallout's* influence can be felt on several independent projects, where smaller teams are putting their own spin on the classic isometric role-playing formula. Here's what the creators of a new crop of modern *Fallout*-likes have to say about the genre and its renewed popularity.

GOING UNDERGROUND

In the words of its lead developer Dejan Radišić, better known as Styg, *Underrail* is a game about a "post-post-apocalypse." Hundreds of years have passed since a mysterious cataclysm drove the remnants of society underground where they now live in protected station-states. As a member of the independent South Gate Station, players must

WRITTEN BY
JACK YARWOOD



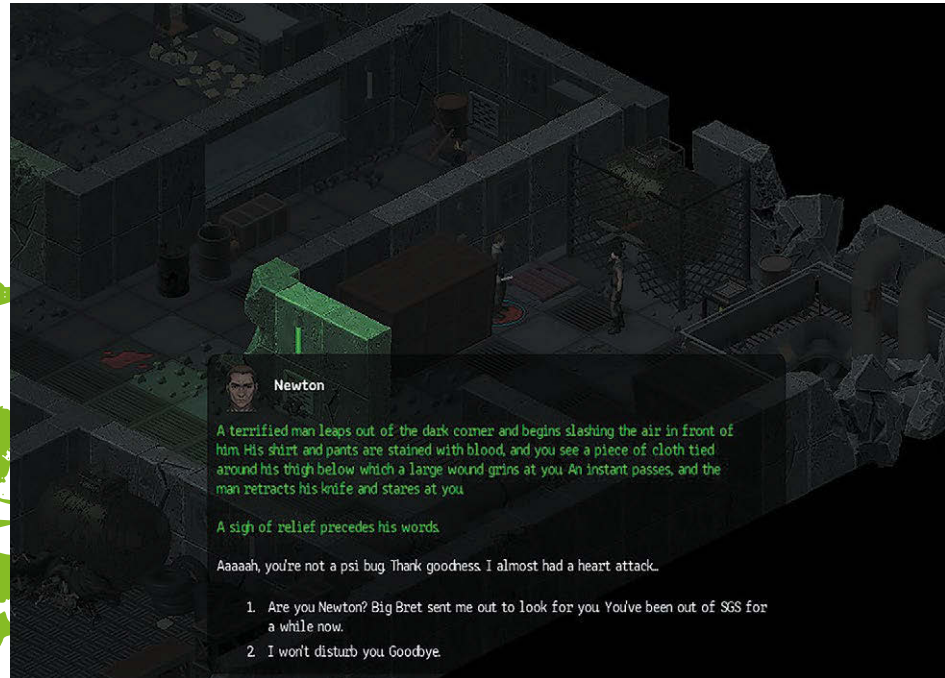
A WASTELAND ADVENTURE

Interplay's *Wasteland* was released in 1988, at the tail end of the Cold War. The project was born out of Brian Fargo's fascination with all things post-apocalyptic. "It was a setting that I was always drawn to," recalls Fargo. "Comics like *Kamandi: The Last Boy on Earth*, films such as *A Boy and His Dog* and *The Road Warrior* [aka *Mad Max 2*], and books like *The Stand* and *A Canticle for Leibowitz*. I also wanted to present a more contemporary setting that differed from the magical fantasy worlds I had been working in."

explore these vast networks of tunnels, battling psi beetles, mole crickets, and hostile humans.

For Radišić, the decision to make an isometric CRPG in the style of *Fallout* was a risky one – companies had already moved away from making them when production began on *Underrail* in late 2008; Radišić, however, had grown up appreciating the look and the mood of games like *Fallout* and *Baldur's Gate*, and wanted to make something in a similar vein. "I always wanted to make a single-character game, where that character would go through this hostile world and have a persistent feeling of isolation and danger," he explains. "So the game wouldn't have expansive urban areas – you mostly just crawl through tunnels like ventilation shafts, blow up holes to make passages, and encounter all sorts of monstrous creatures and races of people."

Though the game borrows a lot of its aspects from classic RPGs, there were many areas that Radišić wanted to improve on from those games. For instance, he criticises the combat in the original *Fallout*, and wanted to expand on it in *Underrail* to give the player more options on how fights can play out. "It's not so restrictive," he says. "If you have a first-person shooter, like *Fallout 3*,



your game ends up being kind of shooter-y even if you can pause the game. In *Underrail*, you can go around planting traps or throwing gas grenades.

There's a lot more stuff you can do in these isometric games. Maybe because the perspective is more conducive to a tactical approach."

"There's a lot more stuff you can do in these isometric games"

Underrail released in 2015, after three years in Early Access, but the game has continued to receive updates and downloadable content as recently as 2019,

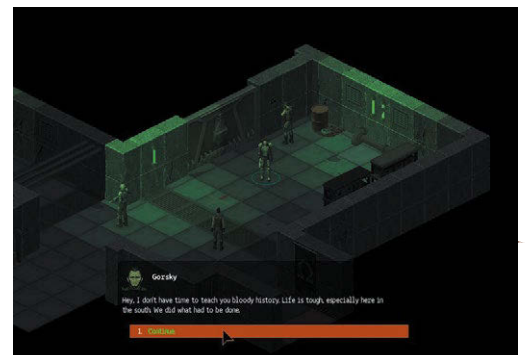
with Radišić and his team having built up a loyal following over the years among fans of the genre.

SPLITTING THE ATOM

Compared to *Underrail*, Atom Team's *ATOM RPG* is somewhat closer to a typical *Fallout* experience, since it allows you to explore above ground in the ruins of a post-nuclear wasteland. But *ATOM RPG* nevertheless provides a unique perspective, since it takes place on the other side of the Iron Curtain in the mid-eighties. Small towns are full of tired-looking inhabitants, and the streets are lined with the once-proud relics of the Communist regime. →

▲ Players in *Underrail* will start out doing grunt work for the inhabitants of the South Gate Station. Quests like rescuing Newton and trapping some Cave Hoppers for the folks in the lab.

▼ In *Underrail*, everyone will be forthcoming with their answers. You'll have to explore and find out some things yourself.



Interface

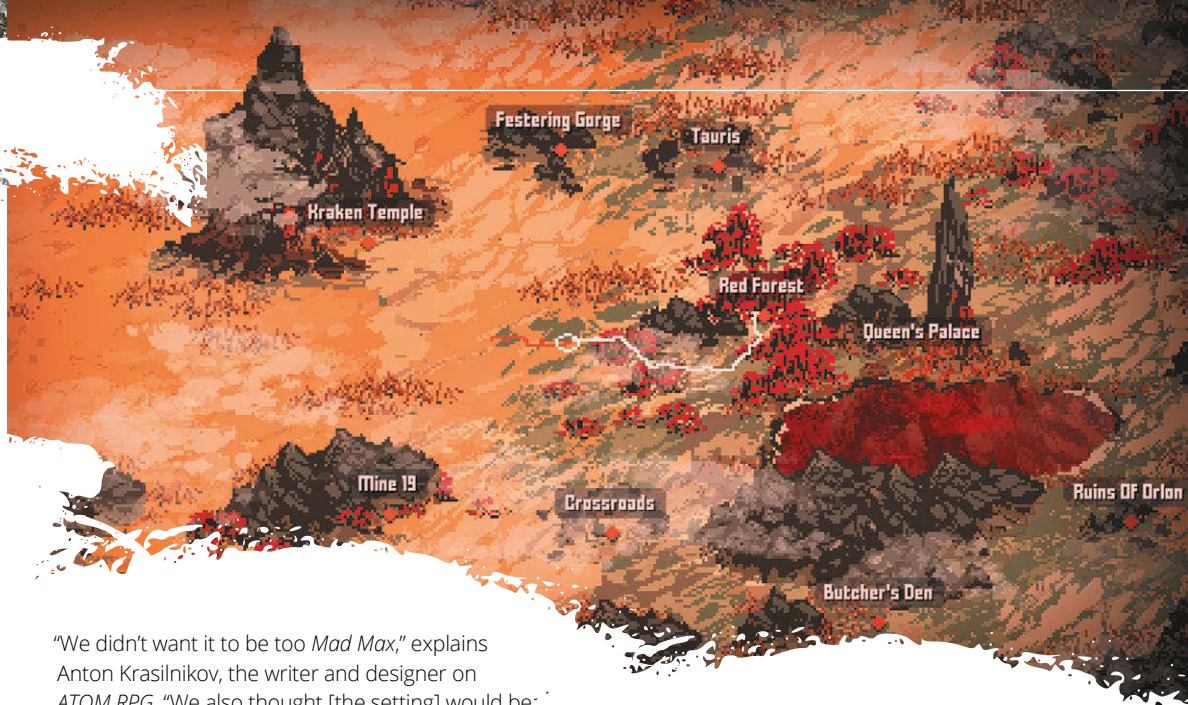
Back to the Wasteland

► *Death Trash* has a number of areas you can explore. In particular, we have our heart set on visiting the Puke Bar. Sounds welcoming.

IN THE FALLOUT

The development of *Fallout* began six years after *Wasteland*, in early 1994, when then-Interplay employee Tim Cain started developing a prototype using some pre-existing assets from other Interplay games. In many ways, the project was pitched as a spiritual successor to Fargo's *Wasteland*, with the game sharing a lot of the same post-apocalyptic influences as well as the same laissez-faire approach to player morality. Players could be as evil as they wanted, or decide to help out other survivors fix their individual problems.

▼ *Death Trash* has a beautiful pixel art aesthetic – and not a shade of green in sight as far as we can tell.



"We didn't want it to be too *Mad Max*," explains Anton Krasilnikov, the writer and designer on *ATOM RPG*. "We also thought [the setting] would be interesting. We all live in post-Soviet countries, so it's much easier in a way to show."

ATOM RPG began life as a high school project made by lead programmer Dmitriy Martynenko and 3D modeller Ivan Semenov. Inspired by their love of the original *Fallout*, the pair wanted to create a game in the same style, but ended up putting the project on hold when they realised the amount of time and resources it would require. In 2015, the two returned to the idea with a renewed passion, and over the next year, recruited a number of other personnel to help them, off the back of a successful crowdfunding campaign.

"The idea from the start was actually to make it super-hardcore," Krasilnikov states, referencing, among other things, the game's skill checks. Similar to the original *Fallout* and other classic CRPGs, players in *ATOM RPG* are able to customise their characters prior to starting a new game by spending points on skills and characteristics. They can also select two additional distinctions

– these are character traits that will alter stats either permanently or during specific gameplay scenarios. "Most of the dialogue has a list of skill checks," explains Krasilnikov. "For example, charisma governs how people will react to you. Will they answer your questions? Will they give you some additional information? Your intellect

also gives you additional options in dialogue. But some of those options are actually hidden. That's part of the appeal of classic RPGs, I think: your stats and your skills contribute to your character."

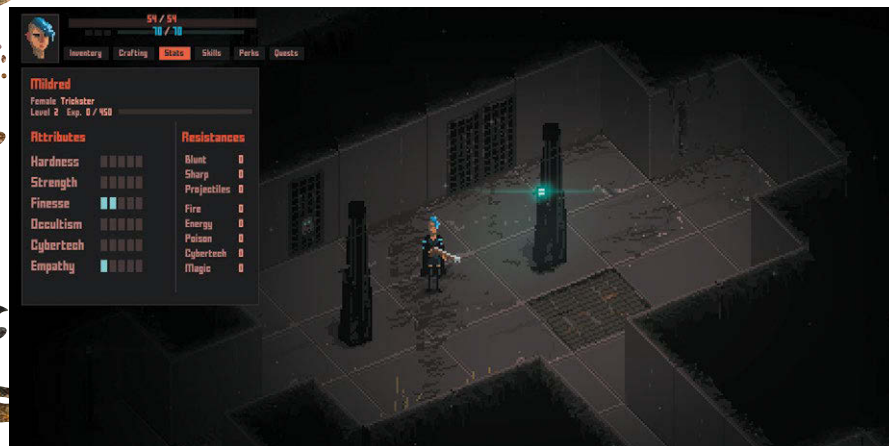
"That's part of the appeal of classic RPGs, I think: your stats and your skills contribute to your character"

ATOM RPG gives players room to approach different objectives how they like, but sometimes this has the potential to go a little too far. To illustrate the point, Krasilnikov tells me about the unintended result of putting cigarettes in the game. "We created a system where you could give certain items to characters," he recalls. "We were thinking you could give medbags to your followers or to certain other characters. But we didn't [consider] that we'd added cigarettes into the game. Cigarettes give you a bonus in crafting, but they will lower your health minus one. What we didn't expect was players would give cigarettes to NPCs to kill them."

Moments like these capture what's special about classic CRPGs. *ATOM RPG* players can have a totally different experience based on their decisions and how they use their tools. Uncovering these approaches is half the fun.

ROAD WARRIOR

ATOM RPG isn't the only game that creates a *Fallout*-style experience in a different country. The upcoming RPG



Broken Roads drops players in the Australian outback, where they'll encounter roaming gangs, mutated wildlife, and makeshift towns. "The game starts in Western Australia, which, when you leave Perth and head inland, starts to get very low population density," explains Craig Ritchie, the game's director. "I'm down in Victoria, and I'm close to Little River where the first *Mad Max* was filmed. We haven't really seen a lot of Australian culture explored in games, but it has this reputation of being a dangerous place where a lot of the animals are out to kill you. You can have a lot of fun playing with that."

The studio behind the project, Drop Bear Bytes, is located in Torquay, Australia, and is staffed by ex-CCP, Riot, and Ubisoft employees. For Ritchie, it's a passion project, having spent much of his life playing classic CRPGs and documenting their history in his long career as a freelance journalist. "I'm privileged to have been able to talk to the people who made these games I was passionate about growing up," says Ritchie. "Asking them about the mistakes they made, what they learned, and being able to take some of that on board. So it's a combination of some pragmatic considerations, but also passion. If there was one thing I was going to make, it would be the style of game I've probably sunk the most hours in through my lifetime of games."

Unsurprisingly, *Fallout* has had a massive impact on *Broken Roads*' development, but it wasn't the only influence; other inspirations include *Wasteland*, *Roadwar 2000*, and Joe Dever's gamebook, *Highway Holocaust*. This, in addition to Ritchie's childhood in England, surrounded

by eighties Cold War hysteria. Among the major deviations from the *Fallout* formula in *Broken Roads* is the introduction of the Moral Compass. This is another layer of characterisation that exists on top of conventional skill checks, which is divided into four categories: Machiavellian, nihilist, utilitarian, and existentialist.

"On top of skills, we also have the attitude of your character, and we have a range of options that relate to your character's attitude," explains Ritchie. "So we could have a skill check, where if you have +10 strength and you're sufficiently nihilistic, then you could then do this [option]. We want your character's attitude [to reflect] how they see themselves, and the kinds of things they're willing to do."

TRASHING THE FORMULA

Like *Broken Roads*, *Fallout* also influenced Stephan Hövelbrinks' isometric RPG, *Death Trash*. Hövelbrinks was a long-time fan of classic CRPGs like *Planescape: Torment* and *Fallout*, and dreamed of making his own game. However, his first attempts were unsuccessful. "I learned programming, then tried to code my own 3D engine," Hövelbrinks recalls. "I failed horribly. So I gave up for a while, got stuck in a normal job, and got very frustrated there. So much so that, a few years ago, I decided to take my savings, move to Berlin for the networking, and learn game making for real."

Hövelbrinks released a few smaller games during this period, before teasing his idea for ▶

INNOVATION

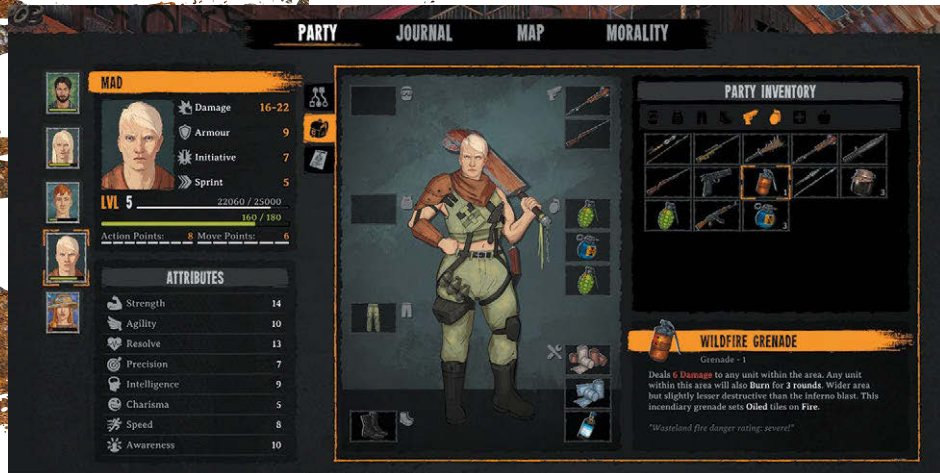
One question that hangs over the *Fallout*-like revival is how much room there is for new ideas in a genre so tied up in nostalgia. Brian Fargo, at least, has a positive outlook. "One of the things I love about RPGs is that there are always ways to innovate, and we strive to find ways to do that with each new one we create," says Fargo. "There are obvious ways we looked to improve *Wasteland 3*, such as visuals, voice acting, and UI design, but there are RPG-specific improvements that we're leaning into. We make deep and reactive games, so we're spending more time on surfacing the effects of player choice – we don't want the player to have to play the game twice to understand the depth."

▼ Nothing to see here. Just trying to hold a conversation with whatever this thing is supposed to be in *Death Trash*.

1 That's a lot of flesh.

2 What ugly beast are you?

3 So, what are you doing all day?



▲ In *Broken Roads*, you'll be accompanied by a party of companions.

Death Trash on Twitter in the summer of 2015. *Fallout 4* had just released, Hövelbrinks felt like making his post-apocalyptic game, and so he posted some of his pixel art online. Buoyed by the positive response, he immediately set to work. "Since then, the idea got a bit broader," says Hövelbrinks. "The world of *Death Trash* is from my own ideas and lots of different sources for inspiration. The general gameplay/genre target I would define as halfway between old-school RPGs and modern indie action games."

Death Trash takes place in a future where humanity has relocated to the planet Nexus, where cosmic terrors lurk in the shadows, and killer robots have risen against their makers. In this chaotic world, players assume the role of a raider living among the ruins, and will have to navigate these obstacles with care.

"Where *Death Trash* differs most [from *Fallout*] I think is in its world, which is more colourful, more alien, and more dreamlike," says Hövelbrinks. "The setting is a wild mix of things, ranging from horrifying and grotesque to occasional light humour, and our goal is to merge all of that into an organic and atmospheric experience."

Another innovation, besides the setting, is its roguelike trappings. *Death Trash* features a real-time combat system not typically associated

with the genre, resulting in more action-packed enemy encounters. A second player can also jump in at any time and occupy the same space, and also interact with NPCs. They'll share quest progress, but won't be able to travel to other locations independently from the main player. Despite these deviations, Hövelbrinks believes *Death Trash* will still appeal to fans of the classic *Fallout* games. "I'd define *Fallout* as a balanced mix of interesting narrative, deep

gameplay that's still accessible, lots of player freedom, and a grim world offset occasionally by humour. And these are traits *Death Trash* strives for, while still doing its own thing."

A FRESH PERSPECTIVE

Among the developers we interviewed, there's a consensus about the role that Kickstarter and crowdfunding played in reviving interest in *Fallout*-like RPGs. There's perhaps no better example of this than the return of *Wasteland*, the series that started it all.

In early 2012, Brian Fargo, the director of 1988's *Wasteland* and the studio head at inXile Entertainment, announced a Kickstarter for a sequel, *Wasteland 2*. This was to be a new isometric take on the series, and was received with open arms by fans.

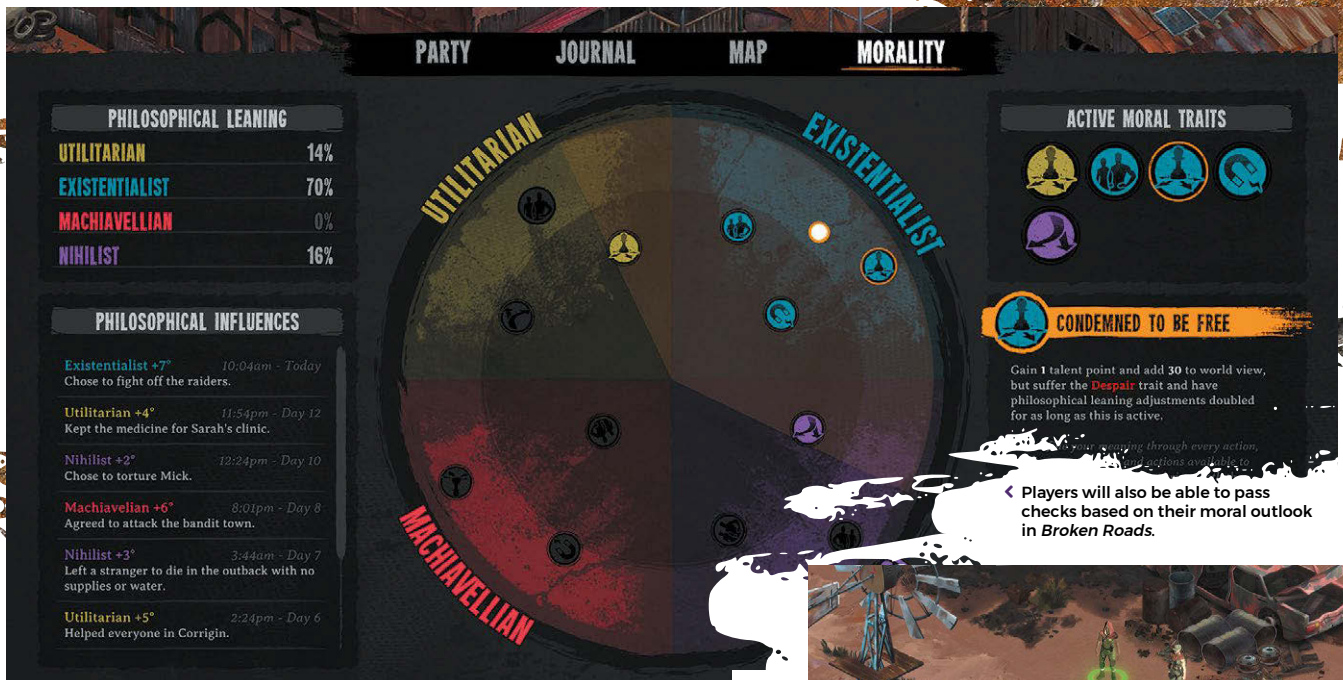
"It was clear that the RPG audience wanted to return to the gameplay of the isometric games from the nineties," says Fargo. "It was a special moment in time where we had gamers who wanted them, and developers like ourselves who wanted to make them. Previously, we were kept apart due to retailers not wanting to support

▼ It's all getting a little *Lord of the Flies* in *ATOM RPG*.



AN OLD-SCHOOL ITCH

Isometric RPGs are back in vogue, partly due to fan demand. "There's still clearly a market," argues Craig Ritchie. "People are still playing these games. I know, because I buy almost everything... There's just so much replayability, and the content is evergreen. And you can just explore some things in a way that I don't think you capture in first person. Like *Skyrim*, the new *Fallout* games, they do amazing things, and they're some great first-person RPGs, but there's something about that pulled-back view of the area you're in, walking around exploring. It's why *Pillars of Eternity* got four million dollars in funding, and *Wasteland 2* got funded so quickly."



the category, or lack of publisher funding, but Kickstarter and digital sales allowed us to break down those walls and deliver a new *Wasteland*. An isometric approach also allowed us to create a tremendous amount of content within a reasonable budget, and we had a big story to tell."

Two new *Wasteland* games are currently in development at the time of writing: a remake of the original *Wasteland*, and a sequel, *Wasteland 3*. In the latter, the Desert Rangers have moved to snowy Colorado, where they face new enemies and environmental threats. "We thought it would be more interesting to get the Desert Rangers out of their element in both weather and notoriety," explains Fargo. "They're strangers in a strange land who need to create new alliances and deal with enemies they're unfamiliar with. There's a power structure in place, and the Rangers are brought in to help keep order (or maybe you choose not to)."

Having recently been acquired by Microsoft, inXile Entertainment finds itself in a unique position with *Wasteland 3*, since it now has the resources to add extra polish to the game. Says Fargo: "The most important ingredient for game production is time, and of course that equals money, which also lets us add resources which have improved the game across the board. Our graphic fidelity made quite a leap, our UI and

character creation were greatly enhanced, the whole game is fully voiced now, there's more music and sound effects, and we're using the extra time for iteration.

"It's important to feel the totality of the experience so you can focus on nuance and

communication," he continues. "Did we telegraph this player choice properly? Does this gun sound seem satisfying? Are we paying off the skill choices adequately enough? What is the right

mix of ambient sound and music? *Wasteland 3* is a huge game, so these iteration passes are very time-consuming. I'm super-happy with the tone of the game, and I look forward to getting it in front of RPG fans."

The games covered here demonstrate the demand for isometric, post-apocalyptic RPGs, and the passion of the developers who fought to keep the genre alive during its quieter years. With greater access to development tools and the shift in the market to digital storefronts, it's finally become more viable for studios to pursue these types of projects. *Fallout* may have long since shifted perspectives, but right now, players looking for an alternative are truly spoiled for choice. 🎮



A NEW EXPEDITION

In July 2019, Stygian Software released a new paid DLC for *Underrail*, called *Expedition*. This added a ton of new waterways connecting existing areas, as well as water-based vehicles and a new area called The Black Sea. A new storyline was also added, pitting players against menacing pirates, corrupted robots, and a bunch of unique creatures.

System Profile

Sega Game Gear

It ate batteries and wasn't massively portable, but Sega's answer to the Game Boy still had its moments

Read enough material about the Game Gear, Sega's nineties handheld console, and you'll find the same sentiments tend to crop up: its battery life was dismally short. It had a colour screen, but it was low-contrast and blurred a lot when things like bullets or blue hedgehogs moved around too quickly. And compared to its rival, Nintendo's Game Boy, the Game Gear was generally regarded as a bit of a damp squib – a chunkier, costlier device that lacked the charisma, extensive library, and portability of its competitor.

Look beyond the accepted wisdom, though, and there's a bit more going on with Sega's device. For one thing, its launch was far from a disaster. Following its Japanese release in October 1990, the Game Gear was officially supported by Sega until spring 1997 – hardly the sign of a commercial flop. (The Dreamcast, for the sake of comparison, had a lifespan of just three years.) Reliable sales figures

are difficult to track down almost 30 years on, but Sega's then-president Hideki Sato regarded the Game Gear as at least a moderate success; interviewed in 1998, he claimed that the handheld sold somewhere in the region of 14 million systems worldwide, which he described as "a respectable chunk of market share."

"The Game Gear soon built up a respectable library of games, ranging from ports of arcade hits to system exclusives"

The Game Gear's hardware also makes more sense when viewed in its early nineties context. Nintendo had essentially owned the handheld market for a decade by 1990, thanks to the success of its *Game & Watch* series and its first proper console, the Game

Boy, released in 1989. As a response to the monochrome, compact, and affordable Game Boy, Sega attempted to make a system capable of replicating some of the colour and attitude of its arcade games.

Components such as a 3.5MHz Z80 processor (identical to Sega's Master System) and backlit colour screen meant that it could run decent renditions of games like *OutRun*, *Streets of Rage*, and *Sonic the Hedgehog*. That processing power came at a cost: the Game Gear could easily drain its six AA batteries in three hours or less. Devices like



Rare beasts

The Game Gear was dark grey as standard, but it survived long enough to get a surprisingly wide range of colour variants and special editions. Some of these – including a blue 'Sega Sports' variant – came out in North America, but most of the more interesting ones came out exclusively in Japan. A branded red edition, bundled with a copy of the RPG *Magic Knight Rayearth*, for example, came out in 1994. Later in its life, Sega renamed the handheld the Kid's Gear, and launched a particularly jazzy *Virtua Fighter Mini*-themed edition. Featuring a graphic of series regular Akira Yuki splashed across the case, it was arguably the liveliest Game Gear ever released – and today, one of the rarest on the second-hand market.





A/C adaptors, battery packs, and car chargers sidestepped this issue, but still made the device less portable than the svelte Game Boy. (In some respects, though, the Game Gear's heft worked in its favour: the device felt comfortable in the hands, and the placement of

the D-pad and buttons on either side of the screen felt natural, even after extended periods.)

In the first couple of years after its launch, the Game Gear soon built up a respectable library of games, ranging from ports of those arcade hits mentioned above, to a decent selection of system exclusives, some of which you'll find listed overleaf. The Game Gear's hardware commonalities with the Master System also meant that it could run that system's games through a cartridge converter.

Despite some aggressive marketing, however, the Game Gear spent the nineties eclipsed by bigger events, whether it was the storming success of the Game Boy, or the headline-grabbing war between Sega's Mega Drive and its arch-rival, the Super Nintendo. Out of all the other pretenders to the handheld crown, though – whether it was Atari's even bulkier Lynx, released in 1989,

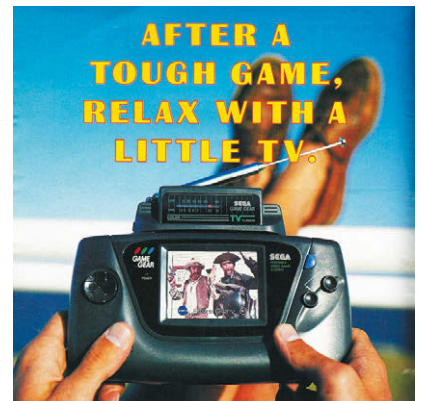


^ Sega stopped making the Game Gear in 1997, but Majesco Entertainment kept the console going for another three years.

Play Today

Ageing, faulty capacitors mean it's getting increasingly difficult to find an original Game Gear in full working order, but if you want to play the system's games in 2020, then it's possible to get an existing handheld fitted with new capacitors and a modern LCD screen. McWill make an excellent replacement kit that both massively improves on the original system's blurry display and also reduces its battery consumption. If you're confident with a soldering iron, it's a modification you can do yourself – kits can be purchased for about £100 from online stores like dragonbox.de – though you'll also find refurbished Game Gears available on eBay for around £150–£200.

v The Game Gear amassed a wealth of peripherals over its lifetime, including an expensive but novel TV Tuner.



or the tawdry Gamate, which dribbled out of Taiwan the year afterwards – the Game Gear at least made a dent in the market. The Game Gear was always something of an underdog, but thanks to its curvy nineties styling and an underrated roster of games, it undoubtedly remains a likeable one. w



If you were colorblind

Why bother seeing only one color? Why that color? Some questions we just can't answer. We make Game Gear™. A multi-color portable video game system with a monitor so good you can turn it into a mini-television. All you've got to do is pop in the TV Tuner when it's not necessary. But the best thing about it is the games. And by now there are 75 of them available. Like Sonic™, The Hedgehog™, 21™, Mario™, Streets of Rage™, Streets of Rage™ 2™ and

and had an I.Q. less than twelve

Remember those old days when you were going to bed there. After we blew a few million to catch our breath, it's back to the drawing board, designing new games, and making sure Game Gear remains the number one color portable, which should be no problem considering the competition.

then you wouldn't care which portable you had.

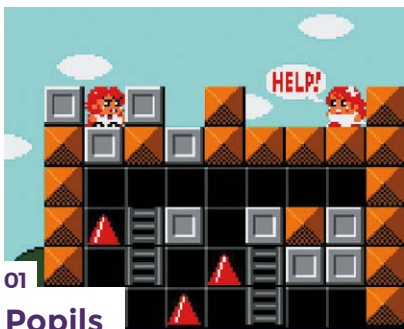
(Of course you wouldn't care if you drank from the toilet either.)

^ Some of Sega's 'edgy' advertising from the nineties – like this one, which takes aim at the Game Boy – hasn't aged well.

Top Gear

10 great battery munchers

Some exclusive Sega Game Gear corkers still worth playing today



01
Popils

1991

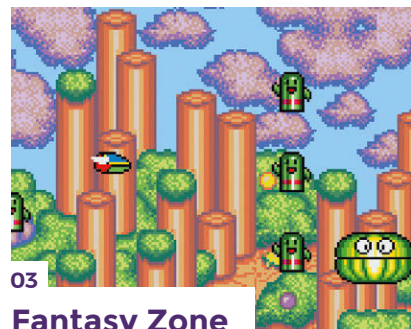
This fixed-screen puzzler, which involves shifting blocks to guide the male hero to the trapped heroine (well, it was the nineties) is perfect handheld fodder, and a pleasurable way of spending a few hours. There's a poignant story behind it too: this was the last game from the eminent Fukio Mitsuji, designer of the classic *Bubble Bobble*.



02
Ganbare Gorby!

1991

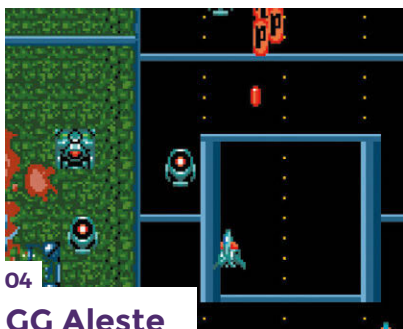
All games are a product of their time, but this action-puzzler could only have come from the late Cold War era. It stars Soviet leader Mikhail Gorbachev, and his job is to hit switches and ensure that his factory's conveyor belts keep churning out food and supplies to a hungry populace. Deeply strange, but also fun. Emerged in the west (sans Gorby) as *Factory Panic*.



03
Fantasy Zone

1991

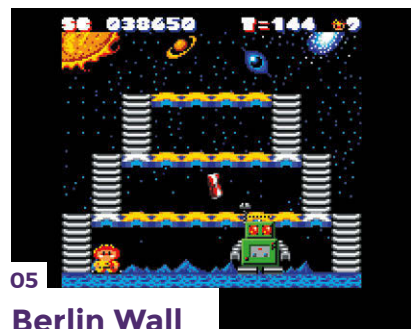
Contemporary gripes that the vibrant colours made bullets hard to spot were valid, but this handheld adaptation of Sega's shooter series looks superb on a modded Game Gear. Now free from unsightly motion blur, its psychedelic backdrops no longer mask the various projectiles and whimsical enemies zipping around the screen.



04
GG Aleste

1991

Arguably the best shooter on the Game Gear, Compile's miniaturised spin-off from its main series is frenetic and packed with cracking tunes. A low production run makes this a rare item; its sequel was actually released in Europe as *Power Strike II* – a little cheaper to buy, and also a corking handheld blaster. If you pick up either one, you're in for a good time.



05
Berlin Wall

Another product of the Cold War era, this obscure *Space Panic* clone has little to do with late 20th century Germany. Instead, it's a jolly little platformer about a Japanese school kid (with a natty yellow hat) who wreaks havoc in a fantasy world by digging holes, and trapping enemies in said holes, before beating them to death with a hammer. One of a handful of Game Gear titles that has a two-player co-op mode.



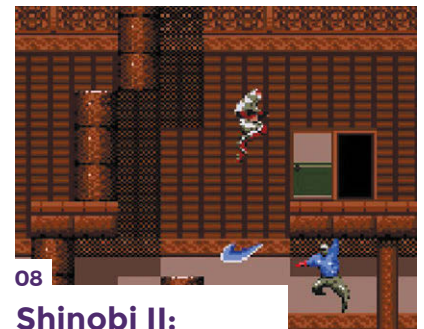
06
Halley Wars

1991
This Taito shooter – adapted from its own obscure coin-op – is pretty austere compared to *GG Alete*, but it has one cool idea: the more invading enemies you let past you, the more damage the Earth will receive. It gives the player an actual in-world reason to kill everything they see, beyond simply racking up a high score, and manages to make you feel a bit bad for failing.



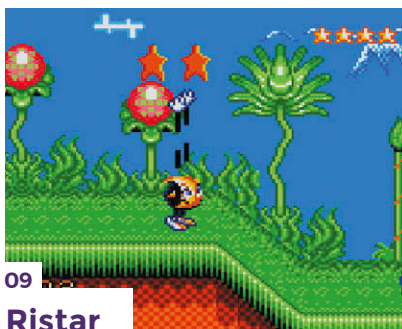
07
Ninja Gaiden

1991
Its name might imply that it's a straight port, but this edition of *Ninja Gaiden* is different from its popular NES counterpart. The platforming action and blue-clad ninja protagonist are immediately recognisable, but the level designs and tunes are completely new. It's short and far easier than the NES version, but it's still a fascinating, overlooked entry in the series.



08
**Shinobi II:
The Silent Fury**

1992
Far from cut-down versions of their big console brethren, the *Shinobis* on GG introduced new ideas, including a *Mega Man*-esque concept of 'rescuing' ninjas from each level, which you can then switch between to take advantage of their powers of grappling, levitation, and so forth. A truly underrated pair of platformers.



09
Ristar

1995
Very different from its better-known Sega Mega Drive counterpart, this mascot platformer is actually a lot better than the *Sonic the Hedgehog* adaptations and spin-offs that appeared on the Game Gear. Sadly, despite some decent reviews, this sprightly platformer arrived near the end of the console's life, and barely got a ripple of attention as a result.



10
Sylvan Tale

1995
Very much a *Zelda* clone, but the Japan-only *Sylvan Tale* is a superbly made one, and a fine addition to the Game Gear's library, which isn't exactly full of titles in this mould. The staples of exploring overworlds and puzzling your way through monster/trap-filled dungeons are here, but it's all done with such charm and polish that speedrunners are still grappling with it today.



Subscribe today



SAVE
49%

wfmag.cc/subscribe

13 issues for just £20

Subscriber benefits

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

Introductory offer

Rolling monthly sub

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

Subscribe for 12 months

Receive all 26 issues

£40 (UK) £75 (USA)

£65 (EU) £75 (RoW)

Offers and prices are subject to change at any time



Digital subscriptions from

£1.99



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

Subscription queries: wireframe@subscriptionhelpline.co.uk

Info

GENRE
Platformer

FORMAT
Xbox One (tested)
/ PC

DEVELOPER
Moon Studios

PUBLISHER
Microsoft

PRICE
£24.99

RELEASE
Out now

Ori and the Will of the Wisps

Just don't call it a Hollow-like

W

hile there are familiar themes to his previous output, I can't imagine Hayao Miyazaki would approve of *Ori and the Will of the Wisps*' methodology

compared to its predecessor. *Ori and the Blind Forest* may not have been styled like an anime, but its themes certainly evoked the films of Studio Ghibli like *Princess Mononoke* and *My Neighbor Totoro*. This sequel transports us to a new setting, not so much a forest but rather a whole region in a state of decay.

As you'll discover early on, when Ori picks up a torch, it's not just for lighting the way in the dark but also for attacking enemies. Before long, you ditch this for the more durable and powerful Spirit Edge, an ethereal lightsaber with a surprisingly wide arc, all the better to swing at any threats. And that's just for starters, as Ori's arsenal gradually expands to an enchanted hammer, a bow, even spirit bombs. While in the first game, Ori was a borderline pacifist, combat designated to your fairy companion doubling as a drone, you've graduated to becoming a Swiss Army killing machine. It's an evolution Moon Studios seems to relish seeing as your combat prowess is regularly tested by rooms gated until

you eradicate a wave of enemies. The truth of the matter is that it all feels incredibly satisfying as well.

In fairness, these new weapon skills aren't all in the name of violence. Taking inspiration from *The Legend of Zelda*, they're just as important for solving puzzles and reaching previously inaccessible areas. Ultimately, *Wisps* still excels when it's challenging you to combine your traversal skills to leap to previously unimaginable heights as you unlock new tricks to keep Ori airborne. It's why, even though there are some incredible set-piece boss fights, grand chase sequences are still part of its DNA.

There's a dizzying amount of new skills to learn, not all of them mandatory for progression – indeed, as the latter half lets you tackle areas in a non-linear fashion, you'll find the necessary upgrades in the same place. It would have been nice if Ori's older skills were available from the

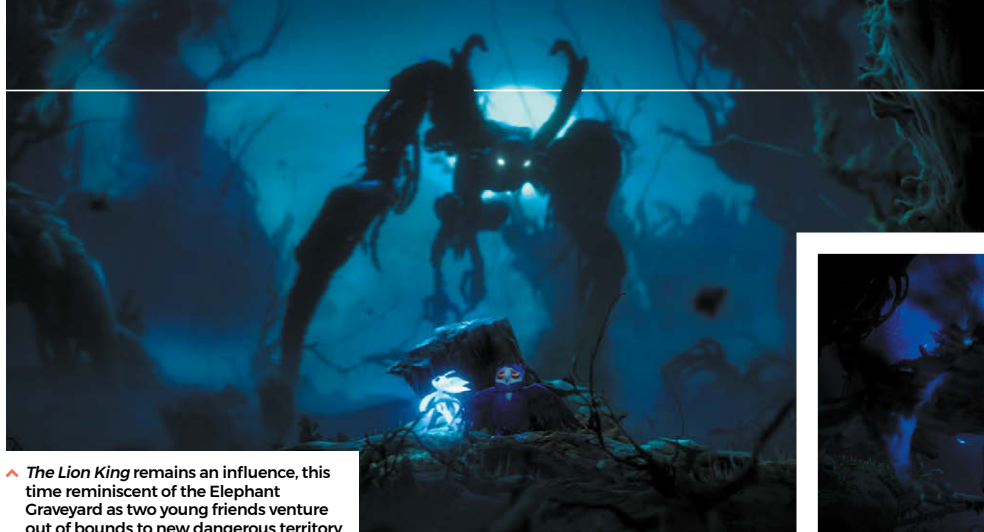
▼ **With more dark threatening areas than before, you have to appreciate the gorgeous environments and lighting where you can.**



REVIEWED BY
Alan Wen

HIGHLIGHT

While the first *Ori* was a fairly lonely affair, you'll encounter more characters throughout your time in *Wisps*. Most of them you'll find gathered in Wellspring Glades, where you can also unlock fast travel. More interestingly, the majority of the quests in this hub involve little more than helping fix the place up to become more homely for its inhabitants, which is surely the best kind of reward.



▲ *The Lion King* remains an influence, this time reminiscent of the *Elephant Graveyard* as two young friends venture out of bounds to new dangerous territory.



▲ Dealing with gargantuan threats is now balanced between both fight and flight.

beginning though, instead of pulling a *Metroid* and having you start from scratch (but without a plot excuse).

Ah yes, you were probably expecting *that* portmanteau to come up sooner or later. However, if anything, *Wisps* seems less keen on being bundled in as a *Metroidvania*, and more in taking notes from recent fellow indie darling *Hollow Knight*. A *Hollow*-like, if you will, much to creative director Thomas Mahler's chagrin. But even if Team Cherry's game actually released three years after the first *Ori* title, there really are quite a lot of stark similarities. There's the NPC who sells you maps of the area you're lost in; collectable spirit shards that add

modifiers to your skills; you even have a skill that lets you use your energy points to heal yourself.

By all means, the above are great ideas to pilfer, and *Wisps* wisely leaves the more *Souls-y* elements well alone. Some players will also be relieved that its predecessor's manual save creation system has been ditched in favour of auto-save checkpoints, while you can also pick between three difficulty levels at the start (though this will be locked in for the duration of the game). Whichever you choose, the platforming remains resolutely hardcore. It's all the better to showcase the fantastic traversal abilities, though you're also asked to combine a mixture of skills with all manner of twitch reflexes, with one mistake often leading to either instant death or else dropping you back to the bottom of the towering obstacle course.

Most pleasingly, *Ori's* gorgeous visuals and evocative soundtrack remain as strong as ever, a grand antithesis to the usual photorealistic grimness of Microsoft's other IP. Even if the story won't have you reaching for your tissues as immediately as the first, it's nonetheless a sweeping tale that's both mournful in how its world is presented while your antagonists have tragic back-stories worth your sympathy.

It's only a shame that at launch, some glaring technical issues have prevented *Wisps* from

singing at the top of its range – and that's even after installing a day-one patch. These include frame rate drops, sometimes amounting to a complete freeze, though the most woeful has been a bug that seemingly ignored my check-pointing progression, despite the icons and messages indicating the game had been saved, at worst setting my progress back by an hour. Moon is aware of the issues and is working on them at the time of writing, but this is a review of what was in front of me at the time

– and honestly, a save bug really does undermine the improvements made to the game's progression system.

It's to Moon Studios' credit then that in spite of these

hitches, *Wisps* is still a confident and mesmerising sequel, improving and expanding on its world and systems in many richly satisfying ways. For those not in a rush to reach the endgame, there's also myriad side quests and secrets to discover, optional challenge trials in both combat and racing form also available from the title screen's quick access if you're into topping the leaderboards. It's a game that's big on content then, but just as big on heart – and don't let the death-dealing melee action make you think otherwise. 🗨

“You've graduated to a Swiss Army killing machine”

▼ *Ori's* bash ability once again reminds you of its unique omnidirectional traversal. New upgrades take it to even greater depths.



VERDICT

A uniquely challenging and beautiful platformer that's everything a great sequel should be.

78%

✓ Tap on the interactive objects in the hand-drawn interiors to make them wiggle and you may be rewarded with a new card in your deck.



HIGHLIGHT

The menu design in *Pilgrims* deserves to be highlighted. By tapping the star, you're taken to a deck of dusty cards which emerge in animated fashion from their water-damaged green box, providing different tappable tabs. Each option from Achievements to Credits is also accented by a suit icon. It oozes charm!

Review

Pilgrims

Manifold Destinies

▲ The character designs and environments of *Pilgrims* ooze Amanita: they're eerie and unsettling with folk-tale appeal.

Info

GENRE
Adventure

FORMAT
iOS (tested)

DEVELOPER
Amanita Design

PUBLISHER
Amanita Design

PRICE
Apple Arcade (£4.99/month)

RELEASE
Out now

REVIEWED BY
Jordan Oloman

VERDICT

A reimagined adventure game for the mobile masses, *Pilgrims* is a short jaunt full of slapstick silliness and smart ideas.

75%

Pilgrims starts out as any good game should. A bunch of hard-up, moustached fellows are playing cards with the devil and imagineering ridiculous trades.

Suddenly, you need to find a tropical bird in exchange for a rowboat ride. From then on, the world Amanita Design has crafted is your scruffy oyster.

Pilgrims is the kind of game that inspires your brain to think differently. It coaxes ideas out of the daftest crannies of your grey matter and rewards you for committing to them. It's all about solving puzzles in the most curious way possible through a card-collecting mechanic.

You'll approach each set-piece with an ever-growing hand that you can play to fix people's problems.

Yet, the beauty of *Pilgrims* is that you'll never know if you're doing the right thing or not. A character might not ask for one of the items in your set but will perform a special animation and start babbling away if you offer it, opening up a new solution in the process and unlocking one of the game's 45 achievements.

The sound effects are adorable, and I will never tire of the protagonist's reactions to a variety of situations. The way he pats at his knees and murmurs in Czech after crawling into a tent delights me every single playthrough. The world and its characters are *Salad Fingers* surreal, from hairy growths trapping swords and fishing hooks to giggling frog-toed plutocrats who could easily

haunt your dreams. What I'm trying to say is that this is Amanita Design's artists at their eerie best. This is brought together by a masterful score full of unusual instruments. Tracks range from chipper to unsettling, employed to great effect depending on the situation in hand.

Pilgrims' overworld is hand-drawn and looks wonderful on the iPhone's screen, with rushing water and twinkling fires popping up betwixt the puzzle points as I hobble my way through the landscape. It succeeds in making you feel small and adds to the spirit of this pocket adventure.

Unfortunately, it's over far too soon, and the execution of its clever new genre-bending ideas

is contained in a short, delightful burst. With those aforementioned achievements, there's plenty to do once you wrap it up, but I imagine for many people, it'll be one and done. Hopefully, Amanita

will explore this idea into a more fleshed-out game of its own, as with a few more systems and a wider map it could certainly become a pocket powerhouse, a charming commute-killer.

The ever-dwindling number of folks proclaiming adventure games are dead have Amanita's latest to thank for that particular opinion not holding any water. The genre is very much alive and kicking in the world of *Pilgrims*; it's simply just had a change of clothes. The indie boffins at Amanita Design have worked their scraggly magic to create a bewitching modern take that thrives in your pocket. 🐾

"The way he pats at his knees and murmurs in Czech after crawling into a tent delights me every playthrough"



Rated
Review

Current Location
Fort Stafford

LOCATE and ACCESS the SAMPLE COMPUTER.

Location: Fort Stafford

ELIMINATE all hostiles in the area.

Location: Checkpoint Yellow

Review

Corruption 2029

Muted Year Zero

For all its muted colours and unbloomed sci-fi seeding, I found something understated, rather than desolate, in the urban ruins of turn-based tactics/stealth hybrid,

Corruption 2029. The harsh reds of my foe's insignia, and the uniform black of the military buildings. Protest graffiti. A book burning. Busted black tarmac scattered with autumnal leaves, something both ancient and transient to reflect the changing tides of revolution. Reminders that the LED signs and supersoldiers, science fiction now, will be ancient history before long.

The setting, too, imports a sense of unease that never really fades, even when the actual plot stays in the background. The snatches of woodland your squad ghosts through before being interrupted by the crumbling, harsh punctuation mark of urban decay. The way your disposable clone soldiers are referred to as "units", suddenly making you reconsider every faceless, reproducible entity you've sent to their deaths in *Command & Conquer*. The newspaper trimmings hinting at a world outside the waist-high wall laden, tiled battlegrounds. It never feels quite deliberate

enough to be called subtle, but vacancy itself can be powerful. With lighting, models, and effects rendered this flawlessly, it's hard not to give it all the benefit of the doubt.

Play is split into three phases. First is a base menu where you prepare weapon and ability loadouts for your squad of three. Second is free-form stealth movement around gorgeous environments, attempting to create distractions and pick off stragglers, collect useful equipment, maybe hack a gun turret for later use, and ultimately prepare the perfect strike for phase three. Once you've mined stealth for its every advantage, you can purposefully engage in the same two-action, overwatch, reload and fire turn-based combat that you're likely familiar with by now.

This feeling of minimalism being uplifted by sheer quality persists throughout everything *Corruption 2029* does. Unlockable weapon types, like spread-fire shotguns, and perks like the ability to freeze enemies in place for a turn, slowly complicate and flesh out combat, but just trading shots from behind cover feels weighty and satisfying.

The lower price point, and comparatively toned down presentation, makes it feel like a companion piece to the developer's previous title. A space to test new design ideas, albeit an exceptionally polished one. I missed the comic book presentation, but I think I got more out of *Corruption's* sleek, tight, replayable encounters than I did out of *Mutant Year Zero's* run-on campaign. It's an understated, rather than lacking, play on *Mutant Year Zero*, and one that delivers tactical combat just as engaging.

HIGHLIGHT

Distracting a guard with a car radio, grabbing a turret code, then hacking a turret to unleash a lethal volley of fire on to a group of three elite foes – all before turn-based combat had even started. Patient exploration can often turn the power balance to your advantage.

Each mission gives you a chance to use stealth before entering combat, evening the odds in your favour.

Info

GENRE

Turn-based tactics

FORMAT

PC (tested)

DEVELOPER

The Bearded Ladies

PUBLISHER

The Bearded Ladies

PRICE

£15.99

RELEASE

Out now

REVIEWED BY

Nic Reuben

VERDICT

While it trades out *Mutant Year Zero's* charm and storytelling for self-consciously systemic, replayable challenges, *Corruption 2029* ultimately offers something equally captivating.

71%

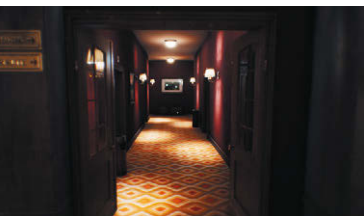
Storytelling is minimal, but effective.



HIGHLIGHT

Though the game hints at possible paranormal goings-on, it keeps signs of this very small and subtle, meaning that you're in doubt until near the end of the game as to whether anything you see or hear is really evidence of something spooky.

▼ Every trip down a corridor makes you dread that there might be something around the corner or someone watching from behind.



Review

▲ The game is broken up into days, often beginning in Nicole's old bedroom with a reason for you to have to leave it.

The Suicide of Rachel Foster

Atmospherically brilliant, narratively lacking

Info

GENRE
Adventure

FORMAT
PC (tested)

DEVELOPER
ONE-O-ONE
GAMES

PUBLISHER
Daedalic
Entertainment

PRICE
£14.99

RELEASE
Out now

REVIEWED BY
Paul Walker-Emig

VERDICT

An effectively haunting setting marred by dreadful storytelling.

68%

It helps to think of *The Suicide of Rachel Foster* as a game with two components. When it comes to what it does with atmosphere and the responses it wants to elicit via that atmosphere, *Rachel Foster* is a roaring success. When it comes to what it wants to do with its story, it is a catastrophic failure.

The game opens with protagonist Nicole arriving at a long-closed and partially derelict hotel that she has inherited following her father's death. Nicole is supposed to perform a quick inventory with the family's lawyer and sanction a sale. However, an unexpected blizzard leaves you trapped alone, with only a phone connected to a FEMA agent to keep you company.

Even early on, there is a sense of narrative events being inelegantly twisted around a concept that's yet to reveal itself. This, it turns out, is a warning of what's to come. The sense that it is contrived is easily forgivable early on due to the effectiveness of the setting the game places you in. Long empty corridors build dread and anticipation as you trudge them; dusty and dark spaces envelop you in paranoia about whether you are truly alone here; and forays away from the safe space of Nicole's old childhood bedroom, into the guts of the hotel's infrastructure, are expeditions of constant background terror. The game uses quiet creaks and shuffles to chilling effect, straddling a boundary where you can believe these are just the

kind of sounds you'll hear in any old building, or just maybe something far more unsettling. The oppressive loneliness of this space is broken up via calls to a FEMA agent who offers support and help to Nicole as she begins digging into the history of her family and the hotel. This is an idea clearly influenced by *Firewatch*, but it isn't as well-written, nor the shifts in the characters' relationship as well-earned. The way the game holds you in a state of constant tension means you will still be glad of the relief a call from him offers – in that sense, the device has its intended effect – but again, issues here are indicative of problems that grow as the game's narrative stumbles, unravels, and lands flat on its face in an awful conclusion.

Indicative of the way the game mishandles some heavy themes is its depiction of the relationship between Rachel Foster and Nicole's father. This amounts to a story about a middle-aged man grooming a 16-year old, but the game does not present it that way. It often feels like an exercise in justifying what it frames as a genuine love story, and is reckless in failing to confront issues of abuse that inevitably require at least some thought when depicting a relationship like this.

The Suicide of Rachel Foster is derivative, contrived, and poorly plotted. However, I can't deny that I still found its atmosphere terrifyingly effective. This place and the unease it provokes mean the game is, in spite of itself, at least a partial success. 🗿

HIGHLIGHT

This one is a literal highlight: the lighting in *Rima* really is quite special, merging with some beautiful art to create an experience that makes the most out of the newer iPhone's swanky screens. This is definitely a game to watch rather than play.

Review

Rima: The Story Begins

Let's hope the story ends pretty soon

▲ Objects that can hurt you are surrounded by this red glow. It's more a concession to the poor design than an aesthetic idea though.

There are points where *Rima: The Story Begins* takes your breath away. It's a gorgeous-looking game, and its densely layered aesthetic flows out in waves of pastel colours that catch the eye and drag you deeper into the experience.

Or at least they would, were it not for one glaring problem – the controls here are sloppy to the point of being almost useless, and they diminish any enjoyment you might have been able to eke out of the experience.

Rima sees you playing a charming little creature, bouncing your way through a series of sumptuous levels and challenges. There are a bunch of buttons on the screen that change their function depending on where you're standing, letting you push, climb, shoot, and clamber through the obstacles dropped in front of you. There are glowing red spikes to avoid, dinosaurs that charge at you, spiders that spit green goo in your direction, and plenty of other characters and bits of scenery

that are trying to kill you. You have an energy bar that depletes as you get hit, and you're going to be watching it going down a lot, thanks mainly to the joystick that you move around the game with.

Often it doesn't pick up your presses, or picks them up too slowly, leaving you stomping into an early death. Button presses are sluggish as well, and it all combines to turn what could have been a charming and engaging experience into a frustrating wander around some very nice looking backgrounds. *Rima* gets the visual side of things spot on – everything from its lighting through to its enemy designs is great – but that

“Just walking from A to B can be an unbelievable chore”

simply doesn't make up for the weakness inherent in its actual play. Jumps that should be easy become horrifying wars of attrition, bounce pads send you soaring into the air

only for your direction changes not to register, and sometimes just walking from A to B can be an unbelievable chore. Those problems expose even more drawbacks waiting underneath – while the game gives you plenty to do, none of it could be described as revolutionary.

We've been jumping over spikes and dodging acidic drips since the 8-bit era, and while the ones on display in *Rima* might be some of the prettiest, that doesn't change the fact that the game is rehashing ideas that weren't new 20 years ago.

Rima looks the part, but those good looks can only get it so far. Scratch below the surface and you'll find a game that's old-fashioned, poorly put together, and lacking the design nous to set it apart from the rest of the pack. 🙄

Info

- GENRE**
Platform
- FORMAT**
iOS (tested)
- DEVELOPER**
UP Entertainment
- PUBLISHER**
UP Entertainment
- PRICE**
£6.99
- RELEASE**
Out now

REVIEWED BY
Harry Slater

VERDICT

Poor controls and uninspired design make this a game that's more disappointment than anything else.

46%

▼ For some reason, a lot of the enemies you fight in the early stages are dinosaurs. Killing them often feels like trial and error, though.



Rated
Review



Detective Cross

This is real life, not some two-bit TV show.
Good police work takes time!

Review

HIGHLIGHT

Murder by Numbers' soundtrack enlists the talents of *Phoenix Wright* composer Masakazu Sugimori, whose jazz-pop earworms perfectly fit your crime scene investigations. The upbeat title song sets a cheery tone, but it's the lounge muzak that loops around behind the Picross puzzles that's most effective, as you tap along to its addictive melodies.

^ There's not much in the way of animation, but the character designs are generally excellent.

Murder by Numbers

Not quite a killer app, but well worth a shot

Info

GENRE
Visual novel/
puzzle

FORMAT
Switch (tested)
/ PC

DEVELOPER
Mediatonic

PUBLISHER
The Irregular
Corporation

PRICE
£11.99

RELEASE
Out now

REVIEWED BY
Jon Bailes

VERDICT

An awkward blend held together by strong writing and the evergreen power of Picross.

70%

W

hat do you get if you cross *Phoenix Wright* and Picross? A puzzling crime? A Pi-cross examination? Well, no. You get a visual novel about solving

crimes that's interspersed with number puzzles. And it works. Sort of.

As a visual novel, *Murder by Numbers* has an easy, cheesy appeal. Protagonist Honor is fired from her acting role on a 1990s network detective show, only to get drawn into real murder investigations with her robot buddy, SCOUT. It's a premise that smartly establishes the glossy irreverence of budget Hollywood then extends it into the game. Honor's cases recreate cheap TV tropes of shifty suspects, contrived plot twists, mild peril, and corny jokes, soaked in period styling and LA sunshine.


At first, it relies too heavily on stereotypes for its characters – dumb blonde starlet, camp make-up artist, dumb security guard – but many of them later reveal hidden depths. In fact, it's surprising where *Murder by Numbers* goes with its themes of manipulative ex-husbands and queer identities. It's rarely subtle, but it works to make diversity feel refreshingly ordinary.

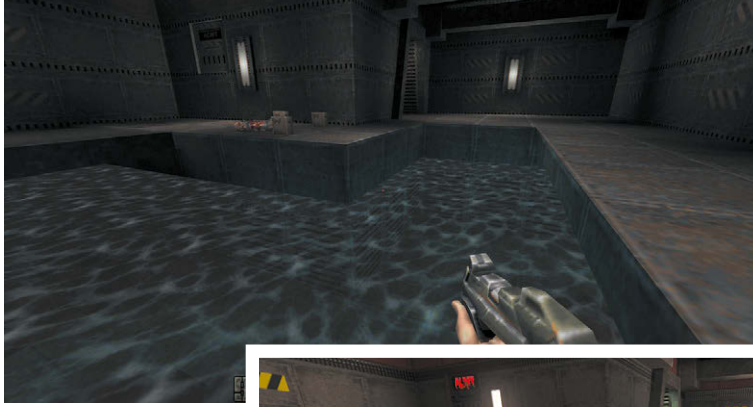
It's especially welcome since the actual detective work is fairly thin. During a case, you can converse with persons of interest and scan scenes for evidence, which you then reveal to suspects to try and catch them in contradictions. In practice, it's a case of methodically exhausting the small number of

variables in play at any moment. Scan until there's nothing left to find, trigger all dialogue options, and show new evidence to each character until something changes.

Except, whenever SCOUT scans a clue, you have to solve a Picross puzzle before he can identify it. As always with Picross, it's simple to pick up, as you work out which squares on a grid must be filled in based on numerical clues. But larger grids, up to 15×15 squares, mount a solid challenge. Occasionally, a hacking minigame has you complete smaller puzzles to a time limit, but otherwise, it's Picross as usual. Learn a few techniques, and it settles into a taxing but relaxing rhythm.

The question is whether the two game styles gel – they do and they don't. On one hand, the process of elimination in Picross creates a parallel with detective work, giving you some deductive reasoning to do with regular tangible results. But when larger grids can take 15 minutes or more to finish, it breaks the plot's momentum, and you might resent the frequent interruptions as you reach the climax of a case. It's like reading a crime novel where every other page is a crossword.

Still, because the plots and characters are bold and clear, you can pick up the threads again quickly enough, and while there are too many big puzzles, each sparks an engaging challenge. Yes, the pace stutters and the two main elements clash, but *Murder by Numbers* has enough heart and style that it's hard to raise an objection. 



Ex-gen/next-gen

Ian's cooing at the shinies in Quake II RTX

Helpfully, *Quake II* is still a fantastic game. Alright, so the purists might balk at how it veered away from the medieval-ish brownness of the first game, and the multiplayer crowd will chuckle at how quaint *Quake II* was online compared to the multiplayer-focused third game. But still: id's second entry to the series that *wasn't DOOM* was a great one in 1997, and it's still a great one now.

Why does this matter? Well, keen-eyed readers will have noticed I'm talking about *Quake II RTX* here – the ray tracing-enabled version of id's classic, released in the middle of 2019 as a freebie to show off those fancy new lighting effects (alongside many HD asset upgrades modded in). Picking a 20-odd-year-old game as a showcase of the next big thing in gaming visuals was something that initially struck me as utterly bizarre. Until, dear watcher of words, I played it. Then I got it.

This isn't something with limited impact like the endless drive to higher resolutions and more pixels per inch. Sure, that's fancy and nice and good, but the benefit is limited as to who it can impact. Admittedly, ray tracing is limited in the same way right now, but after a few years we'll see it as the norm – and as long as you have a display, you can reap its benefits. Well, as long as your hardware can pump out that RTX goodness (according to the Nvidia branding). Shiny puddles, then? Yep.

But so much more. Turning everything up to its top-notch level is nigh-on revelatory, with rooms lit with an air of subtlety we haven't previously seen in non-ray traced beams of light. Auras and reflections, glass refracting things, and, of course, *the puddles*. It's special. Graphics don't make a game, but if you've implemented a technology that makes something look as real as real life, you're stepping up a level – and it does have an impact.

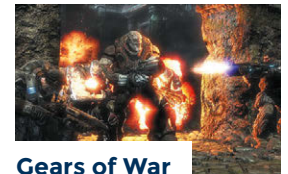
Do note ray tracing doesn't start and stop with realistic reflections. Any graphical tech like this can be used to venture further into the uncanny valley, sure, but it can also be exploited for other, more fantastical means. I am something of a novice when it comes to artistic mindsets, so this might be obvious to the galaxy brains out there, but I never before realised just *how* much

“Turning everything up to its top-notch level is revelatory”

difference brilliant lighting can make to a level, a character, a texture. Am I impressed? Yes, I am. Does this mean I've bought into Nvidia's marketing gumph? Yes, it does. Do I feel bad about this? Not at all.

The absolute best bit of *Quake II RTX*, though, is going into the graphical options and shifting the renderer from RTX to OpenGL. There's a time when your brain might tell you 'Oh, this doesn't look much better than the original *Quake II*'. Your brain is wrong. This one change proves it nicely. How far we've come, oh humble graphics. Next: *Half-Life 2 RTX*, please. 🙄

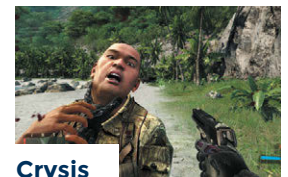
Wireframe Recommends



Gears of War

XBOX 360

Ohhh, the bump mapping. It wasn't the first, but the original *Gears* being tied to the rise of HD gaming lands it an eternal spot in my 'Wow-games-look-well-good-like' list.



Crysis

PC

The fact this is still used as a benchmark for plenty of PC hardware shows how far ahead of the curve *Crysis* was. It's not 2007 anymore, but this still looks real to me, damn it.



Super Mario 64

N64

Remember the *Mario* game before this was *Super Mario World*. The jump to *Super Mario 64* was mind-blowing, and Nintendo knew this, hence letting you poke Mario in the face.

Backwards compatible

WRITTEN BY IAN 'DYNAMO' DRANSFIELD

Skynet: the early years

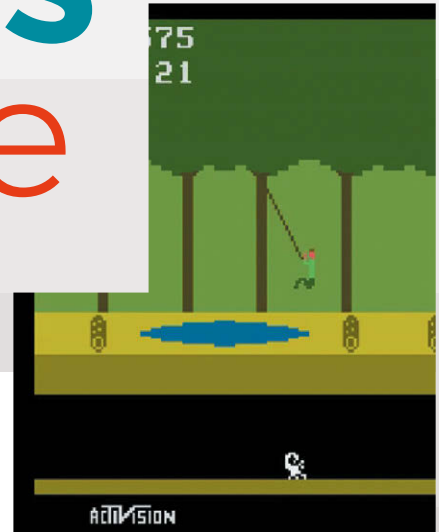
Remember that scene in *Terminator 2* where John Connor fails to beat *Missile Command*? Seems the movie was onto something: machines don't lose. Alphabet – Google's parent company – has revealed a part of its DeepMind AI has 'beaten' 57 Atari 2600 games in a feat we're calling 'actually pretty impressive'.

Proposed back in 2012, 57 Atari 2600 games (later dubbed Atari57) were proposed as a benchmark set of tasks and tests for artificial intelligence systems to try and overcome. The DeepMind team went to work creating Agent57 specifically for the task (after a bunch of try-outs with other AI systems), and just recently posted all the results of Agent57's performances on the Atari57. In short: we can't compare.

See, the benchmarks set by Atari57 factored in human performance at each of the games, obtaining average/mean scores or other performance metrics that could measure how well a player

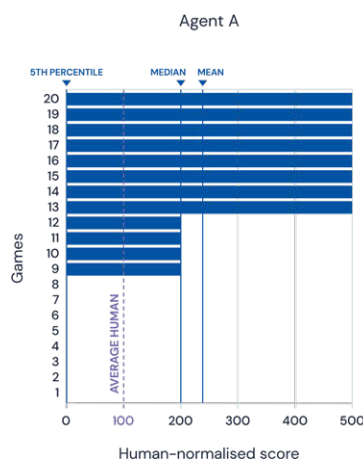
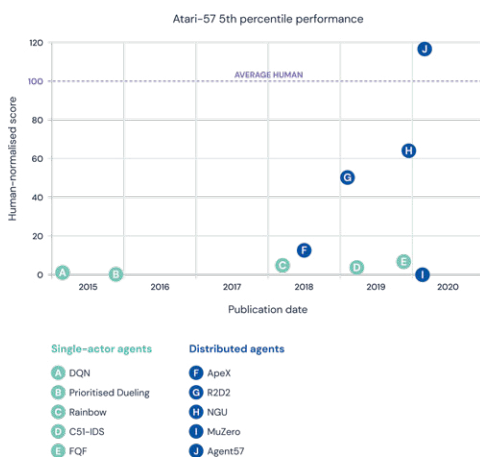
was playing at particular games. This data could then be normalised and used as a barometer for an AI's performance. Agent57, as you can see from the graphs on this page, wiped the floor with the puny meatbags of the Earth. Where previous attempts saw AIs perform well at some games but not all, Agent57 managed significant gains over any human performance to the point that we might as well all give up right now, to be honest. A score of 524,965 on the 2600 version of *Alien* that only stops there *because the video ends* is a bit much to contend with, honestly.

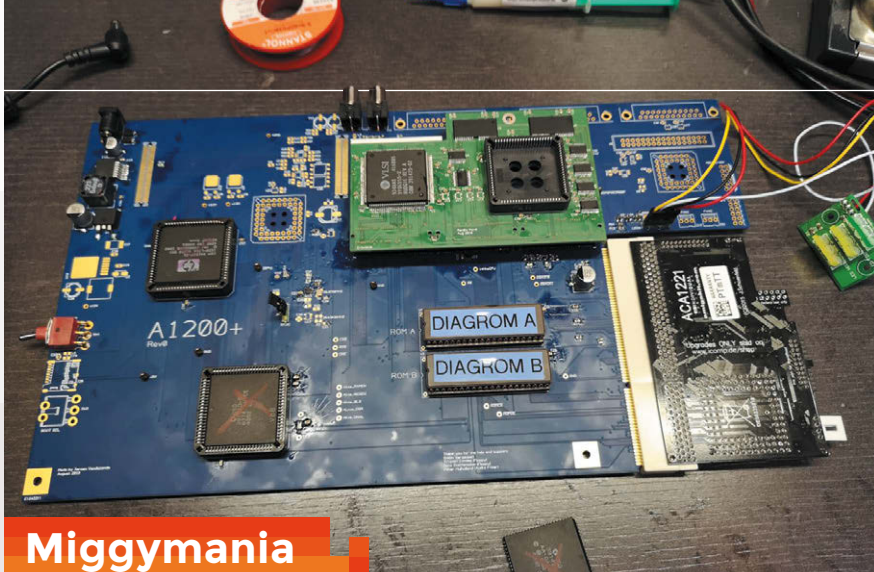
The system relies on a form of machine learning referred to as 'deep reinforcement' – essentially the AI learns from its mistakes. A paper released by the DeepMind team explained why video games are a good choice for training our eventual overlords: "Games are an excellent testing ground for building adaptive algorithms," it read.



"They provide a rich suite of tasks which players must develop sophisticated behavioral strategies to master, but they also provide an easy progress metric – game score – to optimize against. The ultimate goal is not to develop systems that excel at games, but rather to use games as a stepping stone for developing systems that learn to excel at a broad set of challenges."

If you want a bit of terror in your life at the thought of the machines not just destroying all of humanity, but also making us look like chumps in all video games ever, take a look at the DeepMind playlist. It's honestly mesmerising: wfmag.cc/Agent57.





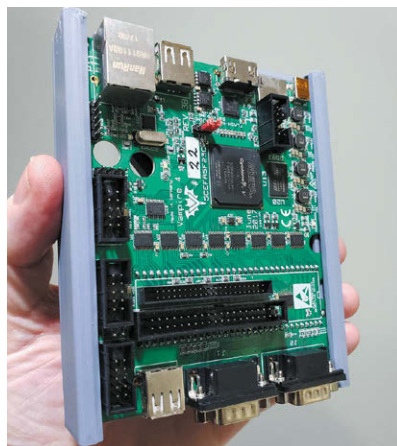
Miggymania

Something you might have missed in the incredibly niche world of 'people who love the Amiga and have a lot of spare money to throw at things': you can pick up Apollo Accelerators' Vampire 1200 V2 or Apollo Core's Vampire V4 Standalone for £394 and £508, respectively (sans postage). But... *what?* Fair question. Both are improvements over the original Amiga 1200, the former an expansion card for the original hardware, offering features like an FPGA-based Apollo core running a code-compatible 68080 processor and with 128MB RAM (among many others), the latter a standalone bit of kit mixing that FPGA 68080 with 512MB RAM and a lot of other ridiculously overpowered features for an Amiga.

Both let you display your Amiga at 720p as well as adding in elements like SD card readers and so much more. No, they're not cheap. Yes, they are amazing. Visit: wfmag.cc/Vamp.

Those who fit into that other niche group, 'people who love the Amiga but don't have a lot of spare money to throw at things, but *do* have the time and

ability to tinker', should look instead to the work of Jeroen Vandezande, whose A1200+ has been getting my heart all aflutter. It's a reimplementaion of the Amiga 1200 from its original schematics that has resulted in an open-source project for anyone to try and build themselves. You'll need a fair few bits and pieces – as well as original Amiga chips – but it's still an exciting project, and cheaper than the Vampire: wfmag.cc/A1200+.



Wonder force

Just re-Amiga-ing for a second here to point out progress on this brand new shoot-'em-up for Commodore's greatest machine (sorry, C64 fans). Titled *Boss Machine*, it's currently fairly early on in production but already runs quicker and smoother than any other side-scrolling shooter I can think of on the format. It's almost as though I didn't need to be jealous of my mate who had *Thunder Force IV* on the Mega Drive back in the day. *Almost*. Anyway, do an interweb search for '*Boss Machine*' by Daniel Stephens, Kevin Saunders, and Lee Smith if you want to know more.

Timely patches

Decades later, we've got patches for *Super Mario 64* and the original *Super Mario Bros*. The former fixes transparency effects in the smoke from Mario's bottom, making it look like it was supposed to, while the latter fixes (though there are arguments about this) the behaviour of our favourite spiny-turtle-throwing cloud dweller, Lakitu. Instead of lobbing the spiky death balls almost straight up every time, post-patch he lobbs them with some horizontal heft. It's a striking difference. The *Super Mario 64* patch can be found here: wfmag.cc/SM64, while Lakitu's behaviour can be modified through here: wfmag.cc/Lakitu.



SUPER MARIO BROS.

UNUSED SPINY EGG BEHAVIOR



BEFORE



AFTER



Prince of Persia

In Jordan Mechner's groundbreaking game, the constant death and gore resulted in an endearingly frail hero, Ryan writes

BRØDERBUND / 1989 / APPLE II/JUST ABOUT EVERY SYSTEM KNOWN TO HUMANITY

Designer Jordan Mechner's pioneering use of rotoscoping (see page 18) was about more than just smooth animation. Attention-grabbing though the running, jumping, sword-fighting Prince was, his movement also served a more fundamental purpose: it immersed the player more deeply in the game's austere world of pain and death.

The basics of *Prince of Persia's* action were already familiar by 1989: its acrobatic, chasm-hopping antics could be traced back to David Crane's *Pitfall!* on the Atari 2600. But *Prince of Persia* took the mechanics of the traditional platformer and added a sense of peril that had never been seen in the genre before; when the Prince threw himself across a gap and grabbed onto a ledge, it really felt as though he was clinging by his fingertips. When he hauled himself up, you could sense his muscles straining as he dragged himself to his feet.

Mistime a jump, meanwhile, and the resulting fall – complete with a strangled shriek and a bone-crunching thud – provided a grotesque reminder of the Prince's frailty. The further you delved into *Prince of Persia's* catacombs, the more horrifying the deaths would become. Spikes would leave the Prince's body skewered and spattered with gore; lose a sword fight, and the Prince would crumple in a pool of blood. Here, even the crockery was out to get you; while some jars would restore your health, others would poison you, causing the Prince to let out a gasp of agony.

“Mistime a jump and the resulting fall provided a grotesque reminder of the Prince's frailty”

Most eye-watering of all were the door-sized steel blades that Mechner gleefully sprinkled around his levels; these gnashing teeth of doom had to be negotiated with another carefully timed jump. Get it wrong, and you'd be treated to the sight of the Prince sliced in two, 8-bit blood left smeared across the gleaming metal. *Prince of Persia* may have been inspired by fantastical adventure movies like *The Thief of Bagdad*, but these bloodier moments were closer in tone to Steven Spielberg's *Indiana Jones and the Temple of Doom* – a sequel so dark and gruesome that it hastened the arrival of the PG-13 certificate in the US.

Once again, though, there was more to these lovingly crafted scenes of bloodletting and injury than shock value. They helped endear the

player to the Prince's plight; he could wield a sword like Errol Flynn (indeed, some of the sword-fight animation was traced directly from 1938's *The Adventures of Robin Hood*) but his human fragility was closer to Harrison Ford's portrayal of the aforementioned Indiana Jones, whose assorted fights and brushes with death left him bruised and exhausted.

Although steeped in Hollywood history, *Prince of Persia* pointed forward in time to gaming's future: the Prince's leaps, plunges, and horrible deaths provided a template for *Tomb Raider's* Lara Croft, and the eternal suffering of *Uncharted's* Nathan Drake. Simply put, *Prince of Persia's* killer feature was, appropriately enough, the imaginatively depicted spectre of death itself. 🗿

Next Issue

ON SALE 7 MAY

MAIN ASSEMBLY

Build robots and just about anything else in Bad York Games' sandbox

Don't miss out!
Subscribe
PG 54



Also

- The finest games about building and crafting things
- How developers capture the thrill of hunting
- The strange world of Mortal Kombat clones
- Make a karting game in Unity with our in-depth guide

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

Designer

Harriet Knight, Ty Logan

Contributors

Jon Bailes, Lottie Bevan, Vikki Blake, Mac Bowley, David Crookes, Konstantinos Dimopoulos, Paul Walker-Emig, Jordan Oloman, Nic Reuben, Harry Slater, Mark Vanstone, Howard Scott Warshaw, Alan Wen, Jack Yarwood

Publishing

Publishing Director

Russell Barnes
Email russell@raspberrypi.com

Director of Communications

Liz Upton

CEO

Eben Upton

Advertising

Commercial Manager

Charlie Milligan
Email charlotte.milligan@raspberrypi.com
Tel +44 (0)7725 368887

Distribution

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

Subscriptions

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

To subscribe

Call 01293 312192 or visit wfmag.cc/subscribe

Subscription queries

wireframe@subscriptionhelpline.co.uk



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

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





CQ32G1
2560X1440

144 HZ	FreeSync	1 ^{ms}
	Curved	QHD

BEND YOUR REALITY

Available at:



ebuyer.com



aocgaming.com



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



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



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