

£3

ALL FORMATS

Wireframe

LIFTING THE LID ON VIDEO GAMES

EVENT HORIZON

MYSTERY AND WONDER
IN SABLE

METRO EXODUS

POST-APOCALYPTIC TRAINSPOTTING

Issue 6 £3
wfmag.cc



GAMESALAD

INSIDE THE CODE-FREE
GAME CREATION TOOL

AFTER SHOCK

REINVENTING THE FPS WITH VOID BASTARDS



Subscribe today

12 weeks for £12*



Visit: wfmag.cc/12weeks to order

* UK Price. 6 issue introductory offer

Bungie's jump into the unknown

This month brought some news that surprised pretty much everyone. (No, not that thing about the game development executive who allegedly left a USB drive full of naughty internet videos at a meeting – from what I've heard, that surprised almost nobody.) I'm talking about the biggest divorce of 2019 so far, apart from the Bezoses: Bungie and Activision, the developer and publisher that have been partnered for eight years in order to bring the loot-chasing space opera *Destiny* to the world, are splitting up. This has given Bungie back the publishing rights to its own game, whose development Activision has spent hundreds of millions funding and marketing – more than half a billion, in fact.

This doesn't look like an acrimonious divorce, with both parties exchanging Twitter niceties and friendly statements, and no plans afoot to kick *Destiny* off the Battle.net platform on PC or change much about how the game operates. It's almost certainly good news for the *Destiny* series, which will presumably be free to evolve at its own pace now instead of being subject to an aggressive DLC and expansion release schedule.

Since 2014, *Destiny* and its sequel have proffered paid-for expansions about twice a year, and they have been wildly variable in quality: *The Taken King* dramatically changed the original *Destiny* for the better and the most recent *Forsaken* update for *Destiny 2* has brought it closer to the series' heyday, whereas *The Dark Below* and *Destiny 2's Curse of Osiris* expansions left players wondering what exactly they had paid for. Bungie hasn't shared its plans for *Destiny* going forward, but it's safe to deduce that working to a twice-yearly update schedule has put pressure on the developer. Gossip from inside Bungie suggests that plenty of people there were getting tired of dealing with Activision.

However, this new state of affairs also represents a lot of extra work for Bungie, which will now be



**KEZA
MACDONALD**

A games journalist for more than 13 years, Keza MacDonald is currently video games editor at *The Guardian*. She once dressed her baby up as Pikachu for Halloween.

publishing and marketing *Destiny* as well as building and maintaining it. The transition will undoubtedly come with a raft of new problems and new opportunities – even if the split has essentially made Bungie one of the world's biggest indie devs.

There are plenty of people who saw this news as a plucky developer finally breaking free of the chains of the money-men, having been constrained for years by a vast corporate entity. (Bungie certainly knows something about vast corporate entities; before this Activision partnership, it was owned by Microsoft from 2000 until 2007, and when that relationship came to an end, Bungie didn't get to walk away with *Halo* like it has with *Destiny*.) Bungie is in a position to do very well independently, with a game that's already up and running, very popular, and well-liked. But its celebrants ought to remember that there's no way it could have gotten there in the first place without Activision.

Publishers are often seen as the moneybags bad guys in the video games industry – and justifiably, as the absence or overpresence of money often has a corrupting effect on creativity. We've all heard stories about market-obsessed executives suggesting or forcing changes to a game's creative direction that would make anyone's toes curl. But game development is getting more expensive, and without the support of publishers an awful lot of games could never exist. Thriving as an independent developer is not just about trying hard enough to survive alone – ask anyone who's actually done it. Investment has to come from somewhere.

Instead of bidding Activision farewell with a one-finger salute, *Destiny* fans should try to find some appreciation for the company that helped make the game happen. MMO shooters were not a proven concept in 2010, when this deal was struck. *Destiny* was a risky and expensive bet that only looks like a safe one in retrospect. Good luck, Bungie – there's nothing quite like taking control of your own *Destiny*. 🙌

#06



Contents



6

Attract mode

Interface

06. Void Bastards

System Shock 2's Jon Chey on his new strategy shooter

10. Metro Exodus

Hands-on with 4A Games' latest post-apocalyptic horror

12. Sable

Inside a world of exploration and mystery

16. Incoming

Barren moons, skating birds and woolly dinosaurs

18. To hell and back

Thomas Brush reflects on the sometimes hellish making of Pinstripe

22. Automata

Mel Croucher takes us through the birth of the UK game industry

44. GameSalad

A closer look at an appetising codeless creation tool

50. Amy Hennig

One of modern gaming's leading lights profiled



12



Toolbox

- 28. CityCraft**
The pitfalls to avoid when designing a game city
- 32. Source Code**
How to recreate the whipping alien tail from R-Type
- 34. Localisation**
Prepare your game for a global audience
- 38. Level design**
Six tips to make your level designs shine

Rated

- 56. Travis Strikes Again**
Suda51's manic action hero makes a welcome return
- 58. Gris**
An artistically stunning exploration of grief
- 61. Monster Boy**
A delightful successor to the Wonder Boy series
- 63. My Time at Portia**
The devil makes fun farming work for idle hands

WELCOME

It was a truly shudder-inducing image. On 21 Jan, Twitter's VisionsofMatt posted a photo of his Nintendo Switch: punctuated by dozens of dog bites, the little console was chewed up beyond repair. The image has hundreds of responses from Twitter users sharing in Matt's pain: there are expressions of horror and sympathy; GIFs of guilty-looking canines, and more besides.

Those reactions say a great deal about the emotional connection we can have with our game hardware. Sure, the Switch isn't a cheap system, but the visceral response goes beyond the thought of having to buy another one. It's hard to imagine the same reaction to an image of, say, a dead fridge freezer, even though they're likely to cost more than a Switch.

It's also easy to join in the collective shudder if you've lost a console to a pet yourself. I once owned a cockatiel that would nibble through the cable of my Sega Game Gear's AC adapter; I'd save up for weeks to buy another, and then the feathered assassin would do the same thing again.

Then again, pets aren't always to blame. I once killed a rare console by accidentally using the wrong AC adapter. As soon as I caught the whiff of burning circuits, I realised my mistake. Never mind the pets: when it comes to expensive, much-loved gaming equipment, we can't even trust ourselves. Now if you'll excuse me, I'm off to lock my Switch in a vault.

Ryan Lambie
Editor



After Shock

BioShock and System Shock's **Jon Chey** aims to reclaim the systems-based FPS crown with *Void Bastards*

There's a sense of unfinished business to *Void Bastards*, the latest from Aussie indie studio Blue Manchu. See, the team is headed up by one of Irrational's co-founders, Jon Chey, a man who headed up the project that eventually became *The Bureau: XCOM Declassified*, a game that was not only a huge missed opportunity for the franchise, but was one Chey didn't see through to the end of its production, leaving 2K Marin as he did during the game's development. It released years later to a seriously muted reception, and lived up to very little of the initial promise it had shown early on in its life.

So is *Void Bastards* a righting of this particular wrong? A way in which Chey can bring us the game we were meant to have before the jam-coated hands of publisher interference smeared their sticky obfuscation all over XCOM's FPS attempt? Well... no, not really. But it's still a jumping off point for the project, and *Void Bastards* combines the hallmarks of the classic strategy-action of the alien defence simulator with a host of intensely creative, exciting elements, as well as touches from Chey's historical dabbling in the likes of *BioShock* and *System Shock 2*. If you're not excited by that mix, you may need to check you're still breathing.

It is, on the face of it all, a first-person shooter, but dig a tiny bit deeper and *Void Bastards* reveals itself as much more. The stylised visuals bring to life a

world in which the player has to navigate the titular illegitimate ones out of the Sargasso Nebula, choosing where to go, what missions to take on, what supplies to acquire, and what tactics to employ along the way. It's a deep, systems-based title with all the promise in the world – and a sense of humour to boot, with Douglas Adams a source of inspiration for *Void Bastards'* general outlook on (space-)life.

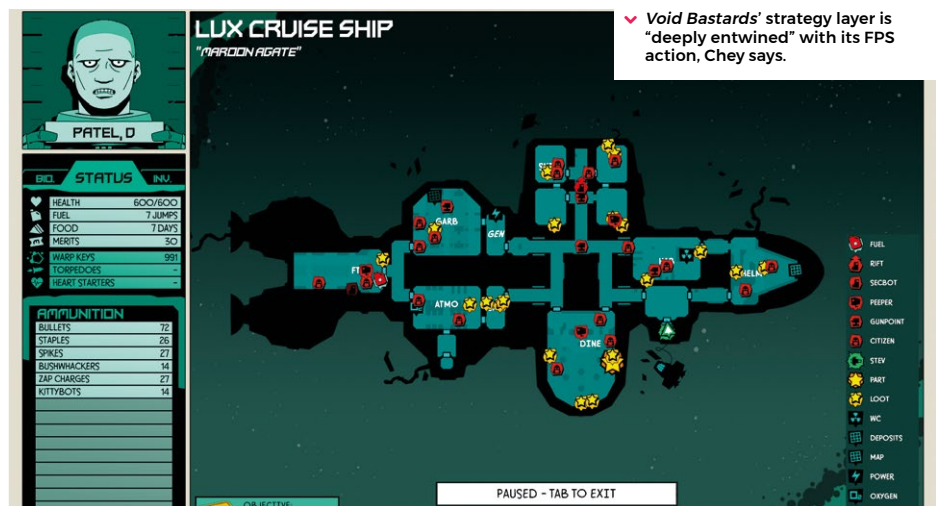
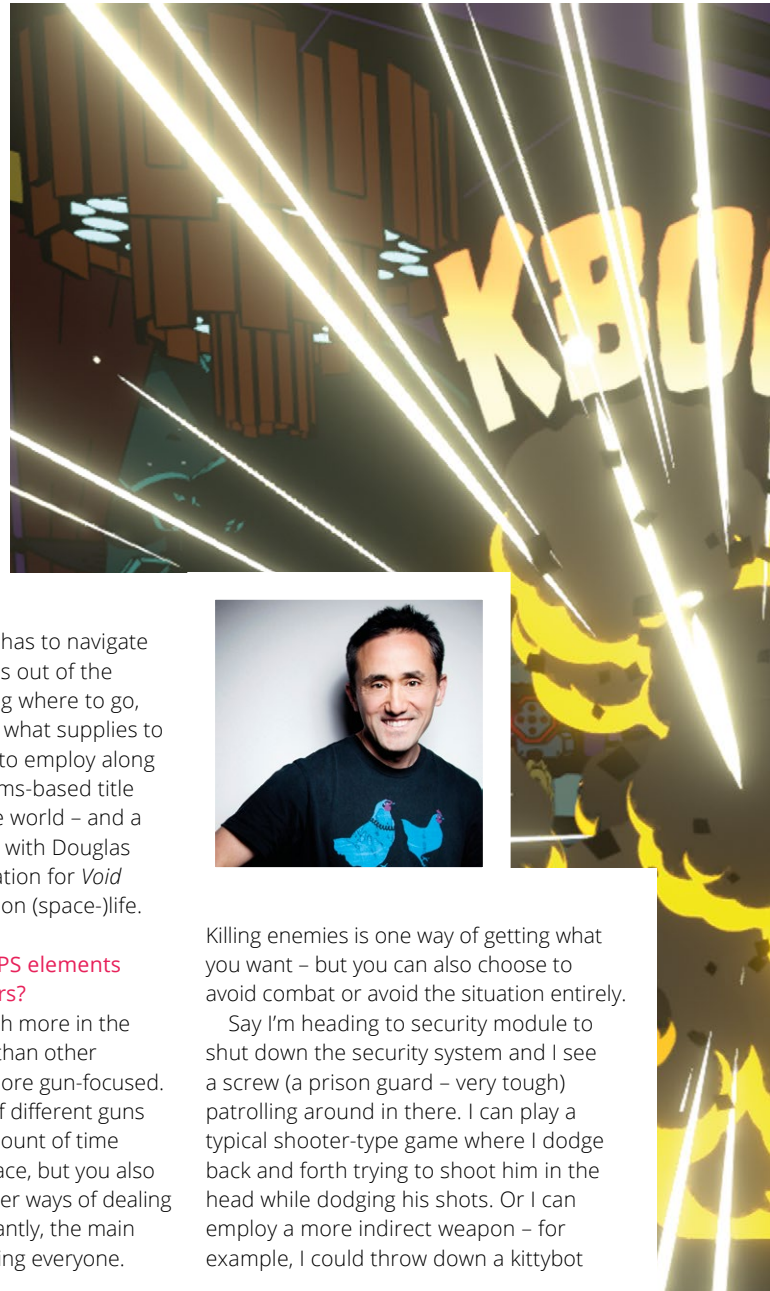
How will *Void Bastards'* FPS elements differ from other shooters?

Void Bastards is really much more in the lineage of *System Shock 2* than other shooters that are much more gun-focused. Of course, we have a lot of different guns and you spend a good amount of time shooting enemies in the face, but you also have access to a lot of other ways of dealing with hostiles. And, importantly, the main focus of the game isn't killing everyone.



Killing enemies is one way of getting what you want – but you can also choose to avoid combat or avoid the situation entirely.

Say I'm heading to security module to shut down the security system and I see a screw (a prison guard – very tough) patrolling around in there. I can play a typical shooter-type game where I dodge back and forth trying to shoot him in the head while dodging his shots. Or I can employ a more indirect weapon – for example, I could throw down a kittybot





^ *Void Bastards'* illustrated look saves on expensive hi-res textures, and also looks magnificent.

to distract him while I run in and use the security terminal. Or I could wait until his back is turned and sneak in. Or I could just say "Nah, I won't bother today." There's a lot more to think about than just pure optimisation of combat.

What's the balance between action and strategy, would you say?

In terms of time, it's probably 80-90% action and 10-20% strategy, but the action element also involves a lot of tactical planning. So the strategy is deeply entwined and doesn't just live in its own mode.

You clearly love card games and board games. Is the strategy element of *Void Bastards* also influenced by tabletop games?

It's probably more influenced by PC strategy games. It has a lot of classic PC strategy game elements: resource juggling, a limited but rich set of choices to make each turn, and a palette of information to support those choices that try to highlight and pull out the relevant factors.

What of *XCOM* has been brought to *Void Bastards*?

For me, *XCOM* was the defining game that created the notion of a strategy game where *you* are in charge of the execution of the strategy as well as the high level planning. Of course, now, there are other games like that, but not a lot of first-person shooters. So, our goal was really to translate this two-layer notion into the FPS genre.

The key part of this two-layer structure is that you navigate through the nebula and then choose which ship to go into - which you then play out in first-person, but we try to build a lot of other interactions between the two layers as well.

For example, you can find tools in-mission that you employ at the strategic layer (torpedoes, warp keys and so on) and vice versa. On some ships you can find a supply flare that sends out a distress beacon. When you come back to the strategy layer, a supply van will come to your location carrying the stuff you requested. Of course, you have to be careful not to bump into pirates or other →

hazards while trying to hook up with the supply van...

How deep does the Douglas Adams influence go? Can we expect more of the dry, surreal comedy we've already seen?

One of the big influences on the story was Adam's game from the eighties: *Bureaucracy*. It's a text adventure which tells a story about someone who has to deal with a whole series of problems stemming from the fact that he changed his address. He goes to the bank to let them know and they tell him to fill out a form that they've sent to his old address. It all goes downhill from there.

We wanted to capture that sense of frustration and anger that comes from having to deal not with real problems, but ones created by an inflexible system. The whole way through *Void Bastards* there's a sense that everything could be resolved if BACS, the admin computer, would just get on with things but it's determined that all the paperwork be completed satisfactorily, even if it's at the cost of several hundred lives.

What engine does *Void Bastards* run on, and how have you achieved that wonderful cel-shaded effect?

It's this incredible super-secret engine you've probably never heard of called Unity!

Seriously though, our approach is usually to try to do something smart with existing tools rather than write a lot of never-



Blue Manchu says the game's campaign will last around 12 to 15 hours.

before-seen tech. In this case, I think a lot of our unique look comes from a rigidly enforced art style (e.g. we try to hand-draw shadows and shading in a way that isn't 'realistic' but instead fits with an illustrated style). Another part is that we render lines explicitly rather than doing them via a textured polygon. That way we make sure that the line thickness is constant rather than varying with distance – exactly as it would be in a hand-drawn image.

Making a game reliant on systems and their interactions is something most of us mere mortals don't understand.

What's the most fun aspect of stitching all of this together?

For me, the most fun is when stuff emerges that you haven't thought of. For

example, we made a rifter gun that pulls enemies out of the world and then dumps them back where you want. The obvious use for that is to grab tough enemies and then dump them back somewhere where you can deal with them (either in a locked room or maybe just into an airlock). What I didn't think of is that it's also a great tool for grabbing a subverted turret and moving it around like a mobile killing machine.

The 'endless supply' of prisoners – your 'barrel of Bastards', if you will – to use on scavenging missions; are these procedurally generated?

Yes, each prisoner is generated and has a unique set of traits that vary their characteristics and what play style you'll need to use to get the best out of them.

THE THRILL OF THE HUNT

After leaving 2K and setting up Blue Manchu in 2011, Chey's first project was... well, exactly the reason he felt the need to leave 2K: a free-to-play, browser-based card battler which would effortlessly lampoon the world of Dungeons & Dragons while providing a deep, strategic, and rewarding well of fun. 2013's *Card Hunter* didn't set the world aflame, but it proved two things: one, Chey was still capable of running a studio that could produce great games, and two, it was possible to make games he wanted to make without the big, bad corporate overlords breathing down his neck.





For example, one might be capable of silent running, making them easier to use in a stealthy way, another might be allowed to take an extra weapon in their loadout, making them more tactically flexible.

What, if anything, has changed in your attitude to the industry since releasing *Card Hunter*? Did it open your eyes to anything unexpected?

Card Hunter's reception reinforced to me that there's value in really focusing on your niche and not trying to be distracted by mass market appeal. Of course, coming from triple-A development, the mass market mindset was something I had to try to shake off. In triple-A, you'd never try to do a retro RPG-themed card game/board game hybrid. But, as an indie, you can do that and it's probably best to really focus on nailing what you're trying to do rather than diluting for people who aren't going to be into it anyway. So, in *Void Bastards*, we really focus on our core strengths: tactical gameplay, strategic goals, planning and thinking, and less on ultra-smooth gunplay. The gunplay is more like the stuff you'd find in nineties sprite shooters and really can't be compared to modern triple-A stuff like *Destiny* or *Call of Duty*.

Does *Void Bastards* feel like a logical step forward, technically, from *Card Hunter*?

Not really, no! It's more of a weird sideways step. *Card Hunter* was an interesting project for us since my background is almost entirely in single-player first-person shooters (*System Shock 2*, *BioShock* etc.). OK, we did do some multiplayer (*Tribes: Vengeance*) but I wouldn't say that's where our real expertise was. And we did do a real-time strategy game (*Freedom Force*) that I was very fond of, but I had never made a card game or a turn-based strategy game. So *Card Hunter* was lots of new tech: database back end, web front end, Flash etc. And now we're kind of throwing that all away and starting again with a pure single-player shooter experience. Not logical, but it keeps things interesting.

▲ There'll be times when running – rather than shooting – is the better tactic.

***Void Bastards* is coming to Xbox One – what brought you back to console?**

Money. No, not really – we always thought of this as a console and PC game, not least because *BioShock* was our first big console/PC title and it's clear that lots of people who probably want to play these kinds of games want to do so on console. So we'd be kind of silly to not cater for them.

“The most fun is when stuff emerges that you haven't thought of”

***System Shock 2* cost around \$700,000 to make. What's the secret to making the most of a budget like that?**

Part of the secret to making *System Shock 2* was some pretty insane crunch. We don't do that anymore so we have to find other ways to make a limited budget work. There are probably two keys for us: one is that we have a revenue sharing system so everyone draws less salary but looks forward to making money from game sales. The other is that the team are in a position where they can handle a relatively long project cycle: people set their own hours and generally aren't full-time, so the whole thing proceeds at a relatively leisurely pace. This allows us to be very efficient with the actual hours that we do bill to the project.



▲ Ben Lee, the artist responsible for *Void Bastards'* comic book visuals.

What's the set-up at Blue Manchu like these days? In the beginning you were working remotely – is that still the case?

Right now, we have a small studio in Canberra where most of us work but we also have Ben Lee and Cara Ellison in the UK and Ryan Roth in Canada (though he's in Japan at the time of writing). So we're still somewhat virtual with the advantage of having at least some of the team in one place.

Is there anything you miss about working on big budget games, or is this really the gaming equivalent of moving to the countryside and having a much happier life?

Things I miss: the company paying for me to fly business class to the US from Australia.

Things I don't miss: reporting to a boss, working with Microsoft Project, endless meetings, marketing-driven decision making. I could go on, but I probably shouldn't. ☹

***Void Bastards* releases on PC and Xbox One later this year.**





The promised land

Metro Exodus sees the series come up for air. Aaron Potter goes hands-on with 4A Games' latest survival shooter

Info

GENRE
Survival shooter

FORMAT
XBO (tested)
/ PS4 / PC

DEVELOPER
4A Games

PUBLISHER
Deep Silver

RELEASE
15 February 2019

Cold, darkness, and claustrophobia. The the first two editions of 4A Games' post-apocalyptic shooter series were defined by these traits: a blend of survival horror and stealth, they added a Siberian twist that we don't often see in mainstream gaming. And yet, if my recent hands-on time with *Metro Exodus* is anything to go by, the new third entry is set to live up to its subtitle in every sense of the word, swapping the dank bitterness of Moscow's underground Metro for a multi-seasonal road trip across the continent.

Of course, it isn't unheard of for *Metro's* lowly protagonist Artyom to peek his head above ground. Both 2010's *Metro 2033* and its 2013 sequel, *Last Light*, gave us brief glimpses of what the post-nuclear fallout looked like on the surface, but in *Exodus*, it's given far more attention. The sequel picks up roughly two years after the events of the last game, with Artyom and his band of Spartans

aboard a locomotive, The Aurora, in search of a new home. Those worried that the series' sense of horror might vanish among *Exodus*' vistas can rest easy: the move into open-world territory comes with a whole host of nightmarish threats to contend with.

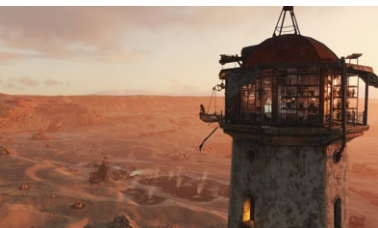
SURVIVING THE SANDBOX

An early section I played through set in the spring, The Volga, is a great example of this. It may be a more polished version of the gameplay demo revealed at E3, but it's a great demonstration of the multi-layered, more open-ended direction *Exodus* is heading. At first, it appears all too familiar: the remnants of

“The weather effects impress, and show a new side to Metro’s world”

last winter's frost are still in the air as Artyom is asked to search a nearby refuge by Miller, the Spartan Order's commander. Whereas previous games would funnel you to the objective via a linear route, though, *Exodus* takes off the restraints and lets you go wherever you please.

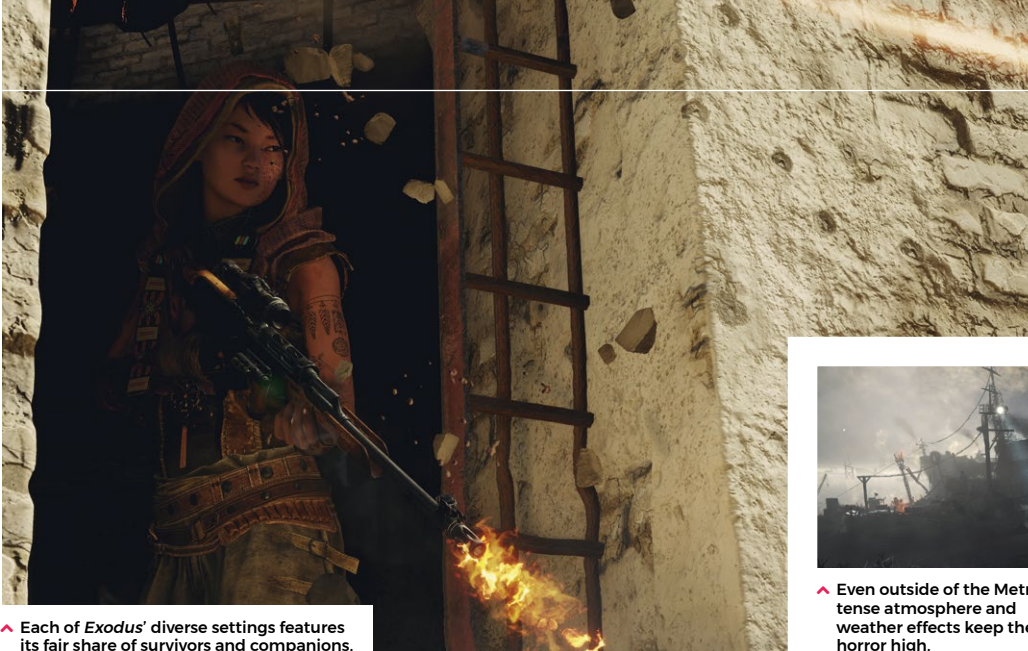
My path leads me to a rowing boat, which I use quietly so as to not scare the cult-like community responsible for signalling the Spartan caravan to a stop. Once aboard, I find a mother and daughter in need of safety, who do much to convince me that the bandits who've taken them hostage aren't worth reasoning with. From here, I enter into *Metro's* familiar brand of 'fight or flight' territory, whereby stealthing through heavily patrolled sections is encouraged, but often ends up in a firefight.



▲ The Caspian is one of many open-world hubs *Exodus* tasks you to survive in.



▶ Finding new parts for your weapons gives you another reason to kill enemies and loot their pockets.



^ Each of *Exodus'* diverse settings features its fair share of survivors and companions.



^ Even outside of the Metro, tense atmosphere and weather effects keep the horror high.

4A Games describes *Metro Exodus* as “a survival sandbox shooter”, yet while this theme ties into the narrative’s oppressive feel and emphasis on scavenging, you’re never forced to worry about hunger and thirst as you are in *Fallout 76*. *Exodus* instead keeps you on the back foot by keeping you mindful of whether your mask is on within a highly radiated area, or how many bullets you have left in a magazine clip. Sometimes, while on your way to a new objective, it’s better to simply sneak past a gang of amphibious mutants rather than waste precious ammunition.

A PLACE IN THE SUN

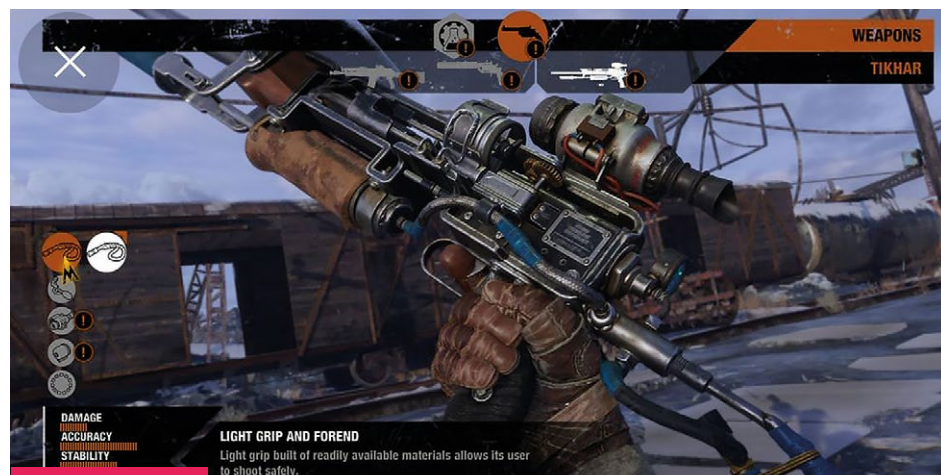
The bulk of my five or so hours’ playtime was spent in a new summer-set portion of the game, The Caspian, where the dried-out air and sun-soaked nature of the desert has driven the game’s characters to a terrible thirst. No spoilers here: just know that it takes place significantly later on in the campaign, where the caravan has moved on to a new part of Russia. The basic formula didn’t deviate too much from The Volga, but The Caspian is indicative of just how much variety each hub contains.

The section begins with Artyom taking shelter in a nearby restaurant, saving himself from the sudden appearance of a vicious sandstorm. *Metro Exodus* was always going to look great, being the first entry developed for current-gen consoles (I played it in 4K on an Xbox One X), but the weather effects here still impress, and show a new side to the harshness of *Metro’s* devastated world. After a couple of run-ins with a mutant and bandit inside, my last kill yielded a key, which then gave me the chance to test out a vehicle – a first for the franchise. The camper van didn’t exactly handle well, but then I’ll take anything I can get in the apocalypse.

It isn’t long before Miller is back on my case, radioing in to ask me to check in on a local

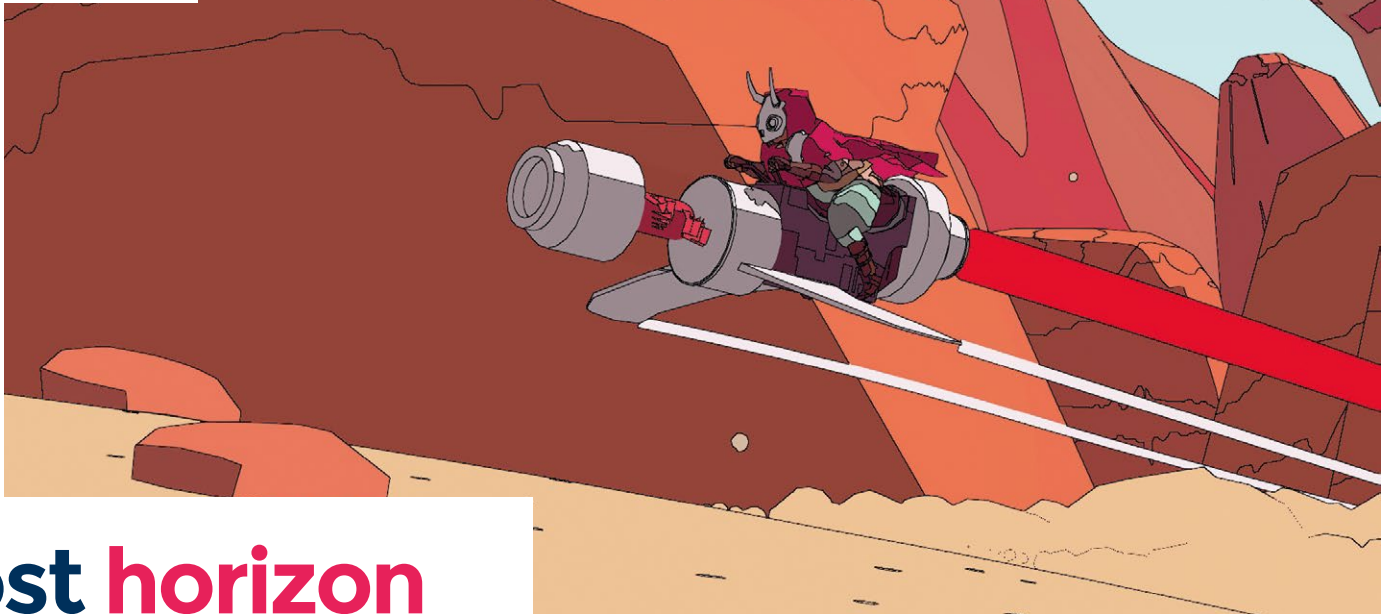
survivor holed up in a lighthouse and in need of assistance. Getting there involves driving across the Caspian map, zip-wiring into a dimly lit cave system before surfacing on the other side and taking down any enemies.

Trust *Metro* to take a section set in summer, and flip expectations by returning to the dark, close-quarters spaces of previous games. Best of all, *Exodus* also fits in a human story at the centre of all this devastation. It’s indicative of a series willing to explore new territory without losing sight of the horror overtones that made earlier entries such cult hits. *Metro Exodus* looks set to remain true to those roots, while expanding its horizons to take in more of its bleak – and now even more beautiful – post-apocalyptic world. 🗺️



LOCK AND LOAD

A full suite of makeshift contraptions – like the series’ infamous pneumatic gun – remain at your disposal in *Metro Exodus*, all boasting specific perks useful for dispatching a mutant or underwater beastie. Weapons are still upgraded via workstations, letting you craft the perfect killing tool with customisable sights and improved magazine capacities. You can also scavenge attachable gun pieces from fallen enemies. When is a shotgun not a shotgun? When it’s been modified with a revolver chamber.



Lost horizon

Shedworks developers Greg Kythreotis and Dan Fineberg explain how curiosity and wonder will guide the story in *Sable*

Info

GENRE
Open-world
adventure

FORMAT
PC / XBO

DEVELOPER
Shedworks

PUBLISHER
Raw Fury

RELEASE
2019

If you already know anything about Shedworks, the developer behind *Sable*, then it's likely to be that its two core team members while away the hours in a shed at the bottom of a garden in North London. This shed represents a rather obvious icon of indie gaming; just two guys making it work as they produce games for the sheer desire to do so. Neither Greg Kythreotis nor Dan Fineberg, the two friends that formed the company, had experience in game development before, and instead sought to make that experience for themselves. "The shed is a focus because it's the reason we've been viable as a business," says Greg, "because it kept our overheads down even when we weren't really earning much money, and it meant we could survive. Without it, we couldn't really survive in the same way."

But the shed isn't what makes Shedworks so interesting; rather, it's the duo's attitude towards the development of *Sable*. While the developer

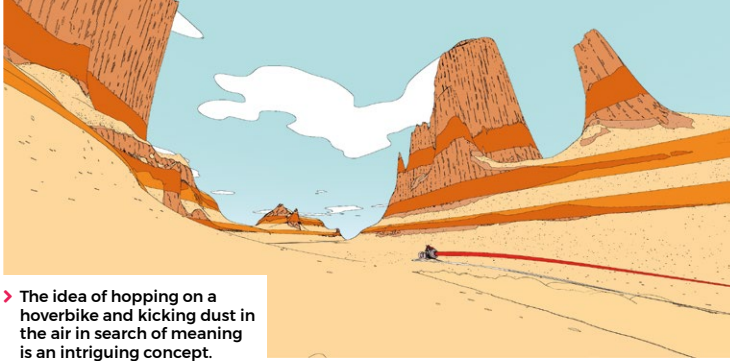
has released mobile games before, *Sable* is its first 'proper' project, for want of a better, less insulting word. It is a game that – even in screenshots, like the ones you see on this page – is able to captivate. It's visually unique and creates a sense of curiousness about the title that, as it turns out, is an integral part of its ideology. "The art style is something that we felt we needed in order to serve the experience of the game that we wanted," explains Greg. "If the art didn't work, the experience we wanted to create didn't work, so it was crucial to get correct. We didn't know exactly how it would end up looking once we got everything working, but the effect was extremely powerful; we also never thought it would have the level of appeal that it's ended up having."

This appeal was largely born from its first public showing at E3 last year, standing out on stage amid numerous other upcoming indie games. It's here that interest in *Sable* exploded, the initial Moebius-inspired art style piquing gamers, though Greg and Dan name-drop narrative-focused games like *80 Days*, *Firewatch* and *Submerged* as gameplay influences. "*Breath of the Wild* was the one that convinced us that the structure of our open world would work," adds Dan, "but even then that still had survival mechanics, kind of, and it still had combat."

And that's the thing: *Sable* has no combat whatsoever; the experience is built around exploration, some puzzle-solving and interacting with the world and its inhabitants. It's driven by a narrative, but one that is at the behest of the player and not a scripted, guided hand.

▼ *Sable's* line drawing effect is heavily drawn from French comic artist, Moebius.





› The idea of hopping on a hoverbike and kicking dust in the air in search of meaning is an intriguing concept.

ONE MAN AND HIS BIKE

“The core is just exploring a desert on a hoverbike,” suggests Greg, alluding to the Podracer-like vehicle that the player will use to traverse the open world. “The core experience is that of wonder and curiosity and trying to encapsulate that feeling as you look out at this vast, lonely landscape.” That’s where the game’s essence of curiosity comes in, since players are guided entirely by their own desire to explore. “The desert is obviously quite empty and quite lonely,” says Dan, “and we wanted there to be some strange-looking thing on the horizon and for the player to think ‘Huh, I wonder what’s over there’, and then for them to drive up to it, explore it and find out something about it.”

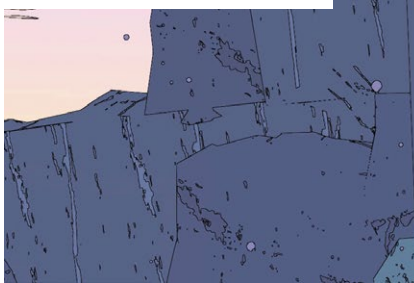
This alone doesn’t sound all that novel, admittedly, but it’s the freedom of exploration that stands out. The story is dealt out piecemeal, with ‘secrets and mysteries’ littered throughout the world, or disparate short stories shared by NPCs – a refreshing take on familiar game staples.

“Open-world games are a very well-established

genre with quite suffocating conventions of side quests and icons on the minimap,” explains Dan of designing curiosity into the game. “The sell of those games is that they’ve built this massive world and it’s packed full of stuff for you to do, but I find that quite stressful. There’s only so many fetch quests and side thingies that you actually have to do before you can see where it’s going.”

As its own answer to the frustrating tropes of open-world gaming, Shedworks is eschewing the typical clutter that these games are crammed with: no towers to climb to unlock the map, no resources to gather or bases to upgrade. *Sable* is an open-world adventure in the truest sense of the phrase. This will carry over to the storytelling, too, with certain parts locked off

› Exploration isn’t limited to driving either, though how much climbing we’ll do is still an unknown.



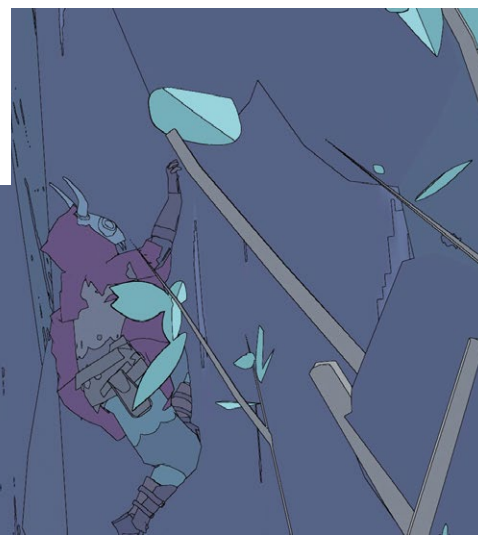
› Allusions to the much-beloved *Journey* are obvious, though *Sable* is open world instead.

by player choices, as Greg explains. “I’m quite happy to say to people: you’ve made a decision and now that’s meant that you block out a storyline for yourself; it’s a decision you’ve made and you just have to live with that. I like games that do that.”

It’s a brave attitude to take with a modern game, but then that seems to be the nature of Shedworks; the pair have made it clear that if they weren’t working on something they were passionate about, they likely wouldn’t be making games at all. “The dream is that each person will seek out the stories that they care about and they find interesting,” adds Dan. “You should be able to have a conversation with a friend and their

playthrough could be completely different.”

So far, the excitement around *Sable* is purely a visual one, its trailer setting the tone for what the game intends to be and little else. Shedworks is tight-lipped about details, and rightly so; without its mystery to discover, *Sable* perhaps wouldn’t be nearly as tantalising as it currently is. If these two inexperienced developers can maintain that sense of curiosity, *Sable* could well be a standout open-world experience. 🗨️



▲ Don’t expect acres of fetch quests and busywork in *Sable*; instead, it’s about exploration and uncovering secrets.

A LEARNING PROCESS

Neither of Shedworks’ two key developers have experience with game development: Dan studied English literature and Greg studied architecture. As such, there’s a lot of learning on the go with *Sable*’s development, with Greg suggesting that most of the time, the pair turn to Google for help. “Unity forums, GDC Vault talks, YouTube tutorials,” he says. “When we were starting out, Lynda tutorials were helpful, too.” The contract work the pair did prior to starting *Sable* was significant, too, since it meant they could “see how professionals with experience structured projects and how they made decisions and conducted themselves.”



^ Bossa's *Worlds Adrift* uses both Unity and SpatialOS.

Temporary disUnity

As a recent spat between Unity and Improbable shows, you should always read the terms of service



^ Unity: protecting its engine, but upsetting some devs.

It was over before it *really* began, but a change to Unity's terms of service late in 2018 flirted with disastrous consequences – both for those using UK firm Improbable's SpatialOS distributed server system, and potentially for those using any other third-party service alongside Unity. While everything was quickly sorted – more on that shortly – the simple fact remains this was a scare that highlights just how important it is for developers to pay attention to the TOS, to know what it is they're allowed to do, and to make sure their actions are officially supported.

This whole tale began back in December, when Unity pushed out an updated set of terms of service for use of its engine. The long and short of it being a clause stated Unity could not be run or simulated in the cloud/over a remote server without a separate licence or the authorisation of Unity. So use of SpatialOS in the backend of Unity-made titles like Bossa's *Worlds Adrift* and Automaton's upcoming *Mavericks* – allowing them, with SpatialOS's distributed server tech, to create massive, persistent worlds with thousands of players in them (as far as the marketing claims) – would no longer be allowed. That was, as many pointed out, not a good thing.

Improbable responded publicly with blog posts deriding this change, and pointing out the company's engine licences had been cancelled

by Unity. Developers of Unity/SpatialOS games were confused, with one such dev – Spilt Milk Studios – taking its game *Lazarus* offline until clarity was restored. Unity publicly responded, claiming Improbable had been in breach of its licence terms – and knew this fact – for around a year already, and stated repeatedly it did not intend to stop people working with third-party solutions to run and operate Unity. But those terms of service still stood.

For Unity's part, the move to disallow the use of systems such as SpatialOS was a pretty straightforward one: it protects the engine from being too closely associated with other software, engines, middleware, and whatever else might be used, when said third-party implementations aren't *officially* supported. It was a move borne out of a protective spirit, it can be argued, and



^ Unity's market share is huge, with the engine powering all manner of different flavours of game.



one that shows how much Unity does value its engine – and how much it pays attention to the potential of future successes *and* failures for those using it.

All the same, the move was not met with wild acclaim by Improbable or other devs – and that’s putting it lightly. The war of words continued between the two central actors, with Improbable posting more blogs to explain its position, claims countered by Unity, and counters countered themselves... really, the word ‘spat’ couldn’t be more appropriate. But then confusion and consternation began to flow in from developers working on Unity games supported not just by SpatialOS, but by other software too – even Epic’s Tim Sweeney stepped in to point out the revised TOS could see the likes of *Fortnite* (were it made using Unity) banned on the service, such was the wording of the new document.

Fast forward through the bickering and the result came a few days later, as Unity – avoiding a PR nightmare – publicly backed down, restored Improbable’s licences, hosted an AMA on Reddit to counter any concerns people might have, and generally set things right in the world again. And for Improbable’s part, the company didn’t gloat too much about its apparent victory. But what was just a week or so in real terms has to have felt like a lot longer for those temporarily caught up in this whole mess.

But the real thing worth taking away from all of this, beyond the drama and squabbles, beyond the pettiness and grandiose statements of taking games off Unity and porting them to Unreal (which Epic was offering cash incentives for, in partnership with Improbable), is that things can change in an instant. You can go from comfortably developing a passion project in your spare time and approaching something like release, to having the rug pulled out from under you thanks to the rewording of a single paragraph of text. It pays to be vigilant, to

“An incredibly positive thing came out of the events of early January”



^ **Mavericks’ 1000-player battle royales were in danger, for a short while.**

formulate contingency plans, and to not be *too* reliant on a series of free engines and other such tech supporting your project. They’re not yours, you don’t own them, and everything could change because of a decision you had absolutely no say in.

Naturally, the best option would be to develop your own engine, but that’s a ridiculously difficult undertaking for most people. As it is, the Unity/Improbable fallout – and subsequent making up – just highlights yet another pitfall of development, especially for those working alone or in a small team who can’t afford to be batted around in these corporate quarrels.

In a statement to Wireframe, an Improbable spokesperson said: “An incredibly positive thing came out of the events of [early January]. The three largest engine makers in the games industry have now confirmed that developers should be able to host engines wherever they want in the cloud. This is a key step, technologically, towards making the next generation of virtual worlds possible.”

And an even more positive thing could come out of this whole escapade if people are to pay closer attention to terms of service attached to software. If you’re a developer working with a third-party engine, especially a free one, know your rights. It hasn’t bitten anyone in the backside thanks to quick action on both sides, but in future folks might not be so lucky. ☹

INDIES TOGETHER

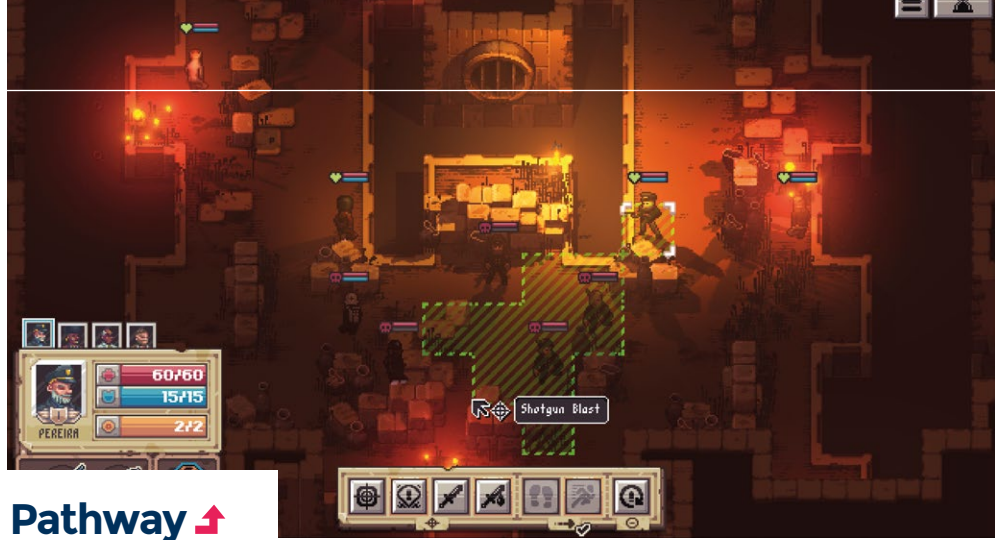
CI Games, the studio behind the likes of *Sniper: Ghost Warrior* and *Lords Of The Fallen*, has announced a new publishing group falling under its wing: United Label. Said group will specialise in helping independent studios publish their titles, utilising CI Games’ 16-plus years experience in the industry to good effect.



The new venture has four titles involved at its announcement, *Röki* from Cambridge-based Polygon Treehouse; *Eldest Souls* from Derby’s Fallen Flag Studios; Mancunian Odd Bug Studios’ *Tails Of Iron*; and *Horae* from Parisian team Lunaris Iris.

And as for the future? “United Label will pursue its expansion by gathering creative talents, offering a structure to success, and building long-term relationships with developers,” said Marek Tyminski, CEO of CI Games. There’s a new (indie publishing) horse in town, folks.

Wireframe



Pathway ↗

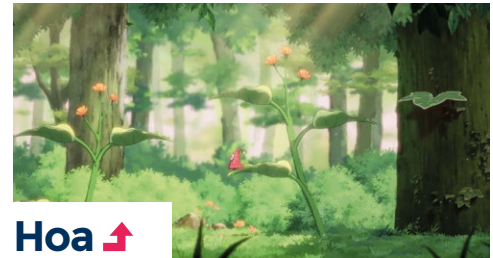
The pixel graphics look terrific, but we're really interested in this game's deeper aspects; it's an upbeat adventure-RPG set in the 1930s, and takes in turn-based combat, treasure-laden tombs, and a procedurally-generated map that changes each time you play. Developer Robotality previously made a similar tactical game, *Halfway*, and they've spent the past four years or so creating a bigger, better expansion on its ideas.

Release date: TBC 2019

SkateBIRD ↘

Could this be the hybrid of *Micro Machines* and *Tony Hawk's Pro Skater* we never knew we needed? Quite possibly. You control a tiny bird who, for reasons unexplained, wears headphones and loves skateboarding. The courses you navigate are made up from ordinary household objects, so expect to pull off sweet jumps over marker pens and grind along pencils rather than railings. There's also the promise of being able to set up your own park of sticky tape half-pipes, where you can invite other tiny birds along to compete. It all sounds thoroughly delightful.

Release date: TBC



Hoa ↗

Currently being developed by two former animation students from Singapore, *Hoa* takes inspiration from the movies of Studio Ghibli. A platform game set in a leafy fantasy world, *Hoa's* hand-painted artwork looks good enough to eat, even at this early stage. Look out for a more detailed preview in a future edition of Wireframe.

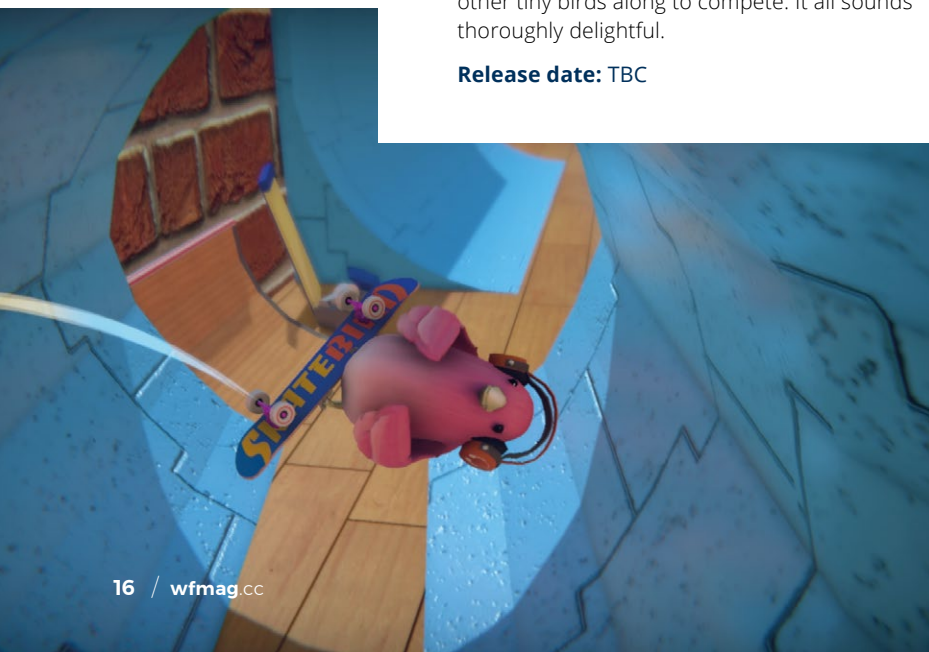
Release date: TBC 2020

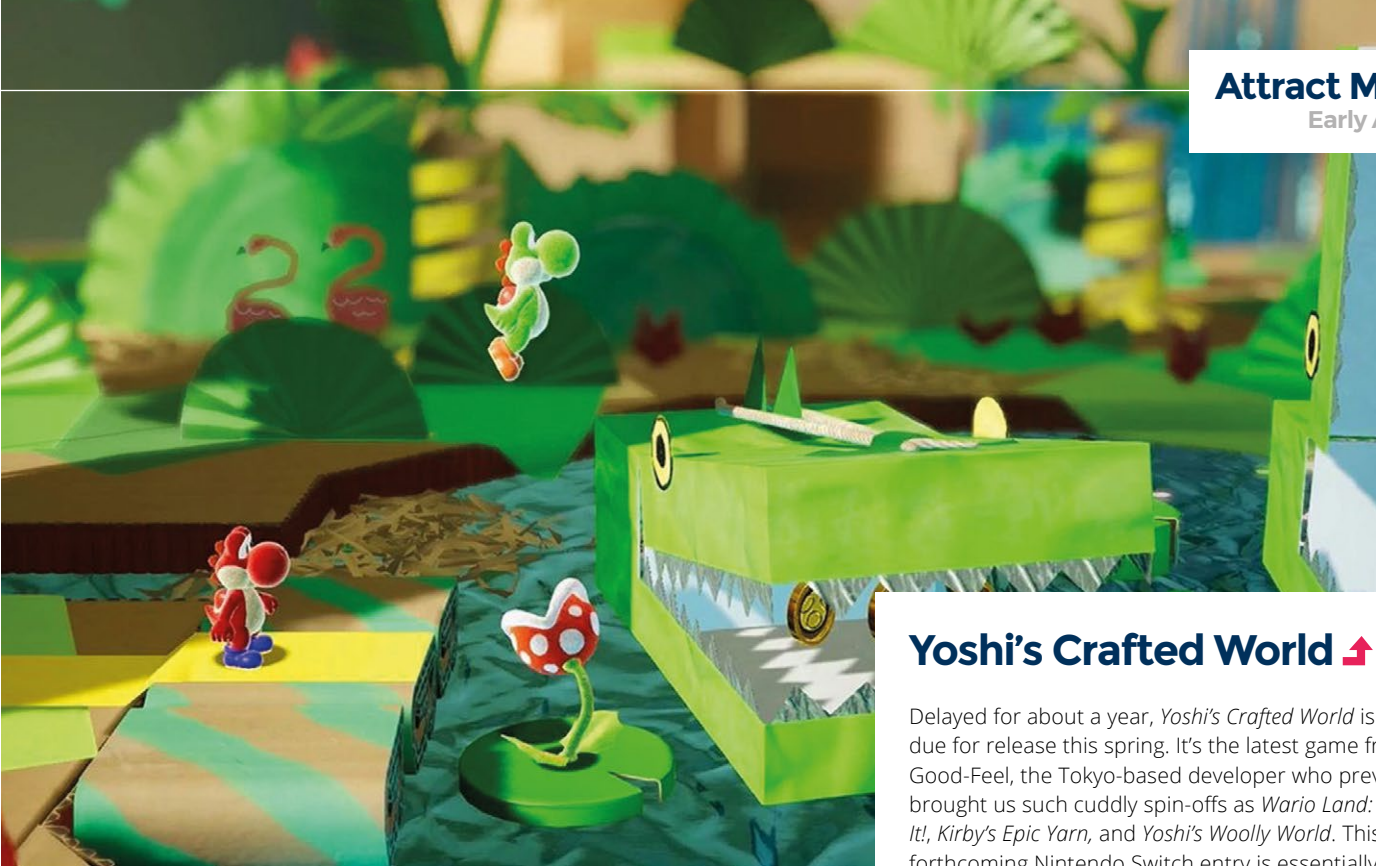


Industries Of Titan ↗

Turn one of Saturn's barren moons into a humming sci-fi metropolis in this forthcoming sim-strategy game; then, once your off-world colony's sufficiently built up, protect it from enemy intruders in fiery, real-time combat in the skies above. *Industry Of Titan's* depiction of a futuristic techno dystopia looks downright irresistible. If that doesn't convince you, consider this: developer Brace Yourself Games previously made the acclaimed *Crypt Of The NecroDancer*, a superb dungeon crawler-rhythm game hybrid.

Release date: TBC 2019





Yoshi's Crafted World ↗

Delayed for about a year, *Yoshi's Crafted World* is finally due for release this spring. It's the latest game from Good-Feel, the Tokyo-based developer who previously brought us such cuddly spin-offs as *Wario Land: Shake It!*, *Kirby's Epic Yarn*, and *Yoshi's Woolly World*. This forthcoming Nintendo Switch entry is essentially more of the same: a 2D platformer set in a world where everything is seemingly made from cardboard and bits of string. The targeted egg attack from early *Yoshi* games is back, but *Crafted World's* main gimmick is the ability to flip stages through 180 degrees to reveal hidden items and pathways. So while we aren't expecting anything earth-shatteringly innovative with *Crafted World*, its developers have a solid track record of creating games that are as beguiling as they are simple. In this regard, Good-Feel really lives up to its studio name.

Release date: 29 March 2019

Baba Is You ↘

Created by Finnish developer Arvi Teikari, *Baba Is You* looks at first glance like a throwback to old block-pushing games like *Sokoban* or *Eggerland*. The twist here is that, by pushing blocks, you can also fundamentally alter the game's rules. Push one block, and you might turn off the collision detection that makes a wall impassable; push another, and you could turn an inanimate object into a controllable character. Beginning life as an itch.io game, *Baba Is You's* award-winning reception at 2017's Nordic Game Jam means it's now being developed for PC and Nintendo Switch.

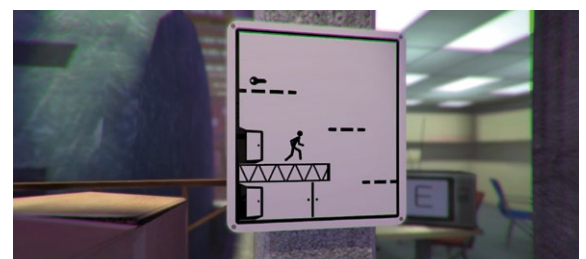
Release date: TBC 2019



The Pedestrian ↘

Here's a platform puzzler that mixes 2D and 3D spaces to ingenious effect. You control the simple stick figure commonly seen on road signs, and it's your task to move him from the confines of one sign to the next. Doing this requires the usual jumping and flipping switches, but there are also neat sequences where you have to draw connective pathways between respective areas.

Release date: Q1 2019





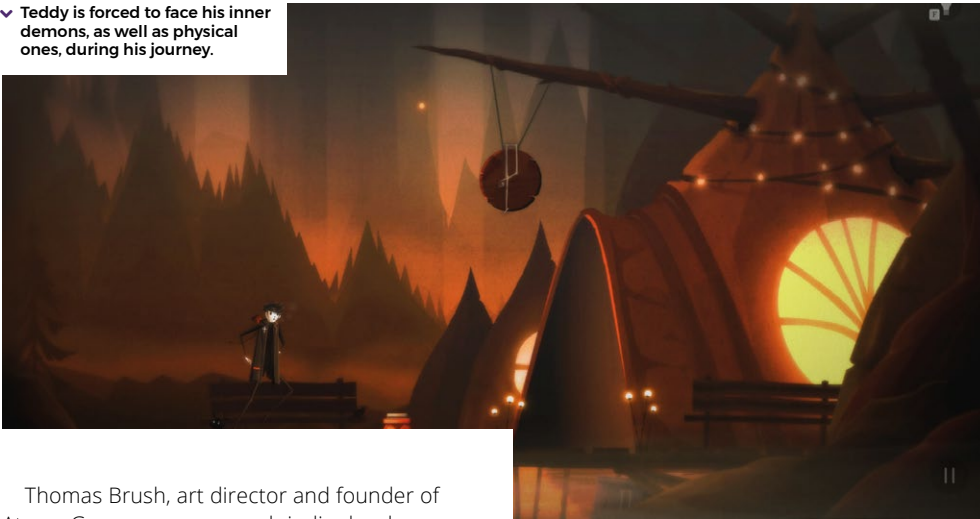
TO HELL — AND BACK AGAIN —

Pinstripe developer Thomas Brush discusses the trials of making a game alone, the inner workings of hell, and the occasional similarities between the two

Likely up there with scaling Mount Kilimanjaro or replacing a spark plug, creating a video game solo seems like an insurmountable thing – not least due to the many skill sets required.

Writing, art, music, code, programming: all these disciplines go into the adventures and worlds that make up our favourite pastime. Knowing this makes it even more mind-numbing to think of such weight resting on just one person's shoulders. And yet, thanks to the increased simplification of development tools and game engines today, there are those out there doing it.

▼ Teddy is forced to face his inner demons, as well as physical ones, during his journey.



Thomas Brush, art director and founder of Atmos Games, was one such indie developer, pouring every ounce of time and passion he had into his first commercial game. *Pinstripe* is a darkly delightful 2D puzzle-adventure that originally saw release on Steam in April 2017. But as with most solo projects, the road to launch would turn out to be a long one. "I started making games when I was in high school," says Brush, regarding his early beginnings as a one-man studio. "I made Flash games and released them online for free. This led to my first Flash game, *Coma*, which went viral and now it's at almost eight million plays. After that, I then knew that I wanted to make games."

With a hit on his hands, Brush's ambitions for his next title suddenly became loftier. But if you think that meant having access to all the necessary funding, think again, as *Pinstripe's* development began in the humblest of places, "in the basement of my university's library and my dormitory," Brush reveals. "It then started following me through to my marriage, and I would work in the evenings, at lunch, and in the mornings on that game while I was working my full-time job." After three years toiling away in the free spaces between his professional life, Brush went on to find success on Kickstarter.

COMMUNITY SUPPORT

"I've been a little scared to launch a Kickstarter," read a post on *Pinstripe's* crowdfunding page. "I'm here to ask the community for support to continue and complete *Pinstripe*. It's not only exciting to potentially get funding from you guys, it's also exciting to get some much-needed community support."

Like so many solo indie devs struggling to gain the attention of major game publishers, Brush opted to circumvent the traditional route and

"THE AFTERLIFE IS REALLY FASCINATING, AND I THINK IT'S A DISCUSSION THAT MORE AMERICANS REALLY NEED TO HAVE"

▼ Thomas Brush is the founder of Atmos Games, an indie studio dedicated to making games with a dark edge.



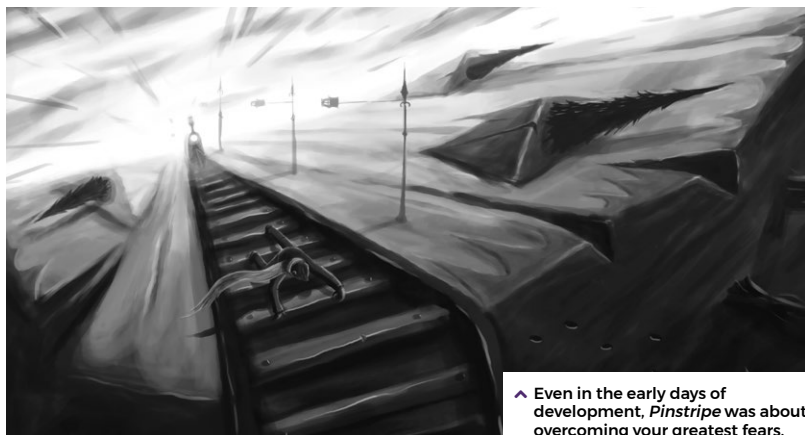
asked players themselves to believe in his. Tim Schafer's *Double Fine* famously did so in order to gauge interest in old-school adventure games – a move that led to the release of 2014's *Broken Age*. Why couldn't Brush do the same?

Pinstripe's crowdfunding campaign went on to amass nearly three times its original goal, all thanks to the backing of nearly 4,000 players who couldn't wait to venture through the game's unique depiction of Hell. Centring on a former minister, Teddy, who is tasked with venturing through the frozen depths of the underworld in search of his kidnapped daughter, *Pinstripe* is, according to Brush, "an emotionally charged story about fatherhood." With this core motivation, players are required to solve puzzles, defeat hellish creatures, and appease the needs of an eclectic cast of characters – all without giving in to the taunts of *Pinstripe's* eponymous antagonist.

When asked about his original inspiration for *Pinstripe*, the thinking behind such dark and potentially serious subject matter becomes clear. As Brush explains: "I really like movies like *What Dreams May Come* and *Constantine*, along with stories like *The Great Divorce* [by C.S. Lewis]. The afterlife is really fascinating, and I think it's a discussion that more Americans really need to have," he reveals. "I think it might be a little bit presumptuous to assume that there's no afterlife, regardless of your religious beliefs – to think that there's nothing behind the curtain of this world. It's an interesting discussion and I wanted to put that into a game."

GIVE THEM HELL

A far cry from the typical red and flame-filled depictions of Hell we're so used to seeing in ➔

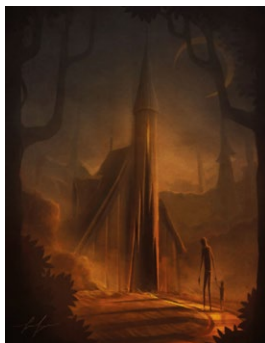


▲ Even in the early days of development, *Pinstripe* was about overcoming your greatest fears.

games like *Doom* and the *Diablo* series, *Pinstripe* instead chooses to take a more ethereal approach. For Brush, making this version of Satan's sanctum unique was crucial to *Pinstripe* having key appeal and helping it stand out in an increasingly crowded market. That, along with the art's notably Tim Burton-ish, fairytale-esque style. "For one thing, this version of Hell is frozen over, which is an interesting thought," says Brush. "It also plays into the popular 'Hell frozen-over' phrase. It's also unique in that it's more about an inner hell or the hell inside an individual and in their heart – as opposed to some entity punishing you in a lake of fire. It's more of a personal story than one you have no control over."

Taking in the eerily creepy sights, poking around in every nook and crevasse is key to helping Ted retrieve his daughter from the grasp of the evil Mr Pinstripe in time. Doing so will see players traverse through the increasingly strange levels of Hell, while constantly questioning the meaning and link between the (seemingly) separate-looking scenarios occurring around them. In terms of what Brush hopes players will take away from *Pinstripe*, he says: "I really want players to feel poignant at the end of the story. I want them to feel like Teddy may have made a lot of mistakes, but was able to find redemption in the midst of everything around him, condemning him."

"The game is about a guy who is looking for his daughter in Hell, and my wife gave birth to a baby girl earlier this year. I guess the understanding of Teddy's relationship with his daughter, Bo, has become manifest in me,



▲ *Pinstripe* sees the eponymous villain steal your daughter into his domain.

"OFTENTIMES,
YOU'RE
PUNISHING
YOURSELF"



more so than when I was actually making the game," continues Brush. "So, the story means more to me now than it did when I was actually making it, which kind of feels serendipitous."

Sticking with his close personal alignment to the game and the themes at play in *Pinstripe*'s short but profound two-hour adventure, it isn't lost on Brush how hellish developing a game solo can feel at times. "It's lonely. Oftentimes, you're punishing yourself, feeling critical about the work you're doing, and feeling unworthy of such a daunting task," he says. "If that's not a small bit of hell, I don't know what is."

LESSONS LEARNED

In addition to developing games, now as part of a slightly larger team forming Atmos, Brush finds solace in providing help to others thinking about doing the same – whether via his creator-focused podcast, *PlayJob*, or his studio's own YouTube channel. Using his own experiences developing *Coma* and *Pinstripe*, Brush continues to share tips on how to avoid the typical trappings of indie development, along with the lessons he learned.

Regarding the one-man studio approach, "I'd encourage people to go solo on at least one project," he says. "It doesn't have to be a big project, but just one project. Not because it's a profitable thing to do financially, but mainly because you learn so much. If you're going to be an indie game developer it's really important for you to learn the ins and outs of the industry – artwork, music, design, and development."

Undertaking side projects like this is also a healthy approach some developers might gain value from, Brush finds. Rather than stay encased within an isolated setting with potentially you as your own worst enemy, it's important to stay engaged with both creators, as well as those you're making the game for. "I'm really enjoying livestreaming right now," Brush tells us. "So, I livestream on my YouTube channel and it's really fun to have people hang out with me while I work and ask questions about how I do things. It also motivates me to work hard and not be distracted by watching YouTube videos, reading articles or something."

If that wasn't enough to keep Brush and his team busy, Atmos Games has also been hard at work increasing scope and getting *Pinstripe* into the hands of as many people as possible. Following initial release on Steam in 2017, the



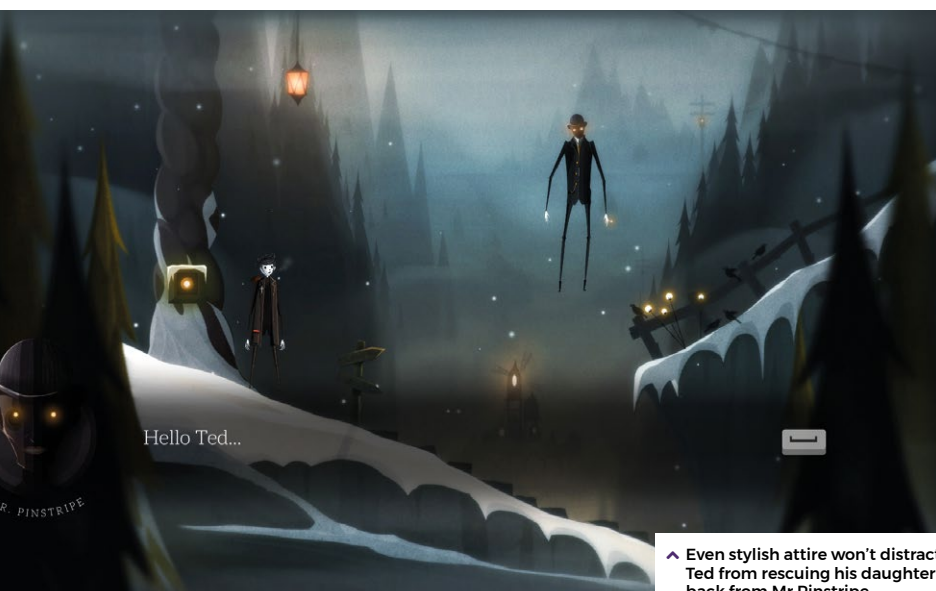
^ As it turns out, surviving Hell is made all the easier with the good aim of a slingshot.

journey of Brush's first, fully-fledged indie title at last made it to the shores of Switch, PS4 and Xbox last October – this time with a little help from specialist publisher, Serenity Forge. It's the end of a chapter for Brush, as he highlights another potential pitfall solo developers should avoid becoming too obsessed by: negative reviews.

"It was really difficult for me to find joy in the positive reviews by the time *Pinstripe* had released, especially when we launched on Steam, because we had one very bad review, and I kept mulling over that as opposed to enjoying the positive ones," continues Brush. "In the midst of all the positive reviews, that one negative review sort of inspired other blogs to use the same kind of language and coring about

A SEQUEL 10 YEARS IN THE MAKING

Following the successful release of *Pinstripe* on Steam, Brush recruited a team of junior developers to help him work on *Once Upon a Coma*, the sequel to his original Flash game that went viral. Like its predecessor, the sequel merges charming character design and puzzles with a distinctly dark edge and sense of melancholy. Originally due to release in 2018, the game will now hit digital store shelves some time this year.



^ Even stylish attire won't distract Ted from rescuing his daughter back from Mr Pinstripe.




^ *Pinstripe* features various stages of Hell, from snow tops to dankly dark mines.

the game. And so, it brought our score down and I was really frustrated with that."

This insistence on listening to the vocal minority rather than the various others championing your piece of art proves to be a challenge all developers – indie or not – are forced to overcome. Especially when such criticisms are so easily proven to be unfounded. For Brush, this proved to be a valuable lesson. "I've learned to let that go," he says. "Now that I've released on Xbox, PlayStation and Nintendo Switch, I've found that reviews don't really mean much, mainly because we're getting great reviews on Switch and it's basically the same game."

FULL CIRCLE

Speaking to the future, and development to Brush's original viral hit, *Coma*, is in full swing – this time with a team of others, as well as the online community, surrounding him in support. "Currently, my studio is working on *Once Upon a Coma*, which is a game about a little boy who wakes from a coma only to find his parents are missing and all of the adults are nowhere to be found." Very much cut from the same cloth as *Pinstripe*, it's a "creepy side-scroller adventure, but it focuses on childhood, innocence, and the perversion of that innocence as we begin to grow older."

All this is to say that, despite how impossible the mountain of developing a game might appear, there's great value in the experience. Even when going through Hell and back, there's light at the end of the tunnel – something Brush knows all too well. "Me and my wife are in transition, because we're building a house and I usually work from a studio space in our home," he says. "I'm actually working in my parents' basement as it's the only place available. It's funny, because I started making *Coma*, my first game, in my parents' house, and now I'm working on its sequel a decade later in the exact same house." 

Automata for the people

Video game pioneer Mel Croucher provides a personal insight into the birth of the UK games industry

WRITTEN BY
MEL CROUCHER



▲ An intoxicating mix of music, narrative and weird imagery. *Deus Ex Machina* was a fascinating interactive experiment.

▶ Released in 1977, the Commodore PET 2001 felt as futuristic as its name.

There is a species of insect that only hatches once every couple of decades, when the world is ready and the conditions are perfect. I am such an insect. According to the history books, I founded the UK video games industry at such a moment, and then crawled back underground to wait for the world to turn again. Here's what happened.

I programmed my first games machine in 1955. It was a primitive metal sequencer called a Sooty Xylophone, and the program was a little row of coloured dots. The idea was to bash a colour-coded key in the order each dot appeared, but those preprogrammed sequences were boring. So I reordered them into more interesting combinations, and wrote my own sequences. The result sounded like a tin xylophone hit with a stick at random, which is exactly what it was. But hey, I was only seven years old.

A few years later, I inherited a pianola. A pianola was a sort of robot jukebox for Victorians, and the software that called the tunes was great. It was stored as holes punched into rolls of paper, and programming was simple. I got a roll of wallpaper and drew up a grid of eighty-eight squares times infinity, one square for each note on the piano keyboard and infinity representing time. If I wanted a note to play, then I punched a hole in the right place, which triggered a tiny hammer onto the associated piano string and played a pitch-perfect note. My system was made possible by a computer program invented in 1801 by Joseph Jacquard, to create geometric designs in rolls of cloth and starve French weavers to death.

It was at the end of the 1960s when I entered a room with No Entry on the door and met The

Beast. I belonged to nobody, but The Beast belonged to the Ministry of Defence. The Beast was big enough to walk around inside and feel the heat coming off ranks of glass valves. Programs were written in a language no more complex than my xylophone or my pianola, and the computer code was stored as little holes on punch-cards. I knew I could make The Beast perform lewd multimedia acts, and there was no question in my mind that the true purpose of electronic brains was to entertain us. Sure enough, after six months of programming, I got the giant computer that was entrusted with the nation's defence to beep *Twinkle Twinkle Little Star* in sync with a row of flashing light bulbs. And then I forgot all about computing for years.

The Commodore PET was launched in January 1977. It looked just like the sci-fi machines we were all familiar with from *Star Trek*, even though the casing was made of the same tin as my Sooty Xylophone. It was blessed with 4 kilobytes, which was as much memory as The Beast, but instead of occupying an entire room, it could sit on my desk. The coding language





▲ Mel Croucher (seated, holding a candlestick telephone) with the original Automata team from 1977.

was PET BASIC. You couldn't write a simple instruction like, "make the blob go up the screen until it hits something." You had to mask your ideas with arcane expressions which sounded worryingly suggestive: spreadsheet, RAM, nibble, byte, peek, poke. After washing my hands and growing a moustache, I set about revealing the true purpose of computers to an innocent public. And that purpose was to have fun.

RADIO GA GA

I founded my multimedia company Automata in November 1977. Everyone was talking about computers, but nobody actually owned one. Ignoring this fact was a handicap, but I never once questioned my belief. I knew for certain that computers were for people to play games on. All I had to do was find those people. There were no computer magazines, or enthusiast clubs, or ready-made software, but we did have radio, and the boss of my local commercial radio station, Radio Victory, was someone I went to school with. Using the fact that we once played mother and daughter in the school play, I went to him with an idea. My idea was that sizeable numbers of computer owners across the South of England would receive my video game signals through their radio sets and get so excited by the concept of computer entertainment they would want to contact me. Then I could try and fleece them for money.

My first on-air video game was broadcast on the 257FM waveband late at night on 15 December 1977, and I topped and tailed the coded signals with a crummy prize competition

to try and stop my audience switching off. After the first broadcast I got three responses. But by the end of the season the number of radio listeners with access to a computer was beginning to grow, and I was given a mainstream evening slot every Thursday night. And that's how I found myself midwife to a new branch of the entertainment business.

By the time the British home computing boom exploded, my little company had become the best-selling video games outfit in the land and the electronic world was our oyster, mainly because we had almost no competition. Between 19 November 1977 and when I quit on All Fool's Day 1985, we produced around six dozen titles. We were absolutely fearless because there was nothing to fear, and we

never obeyed any of the rules because there weren't any rules to obey.

For the next 30 years, as the

multinationals took over and video games grew into a \$120 billion industry, I found the creativity and excitement of the early days completely stifled by derivative, repetitive pap. Technically brilliant, visually stunning, emotionally bereft. Until now. As in all revolutions, the wheel must turn full circle, and video games are no exception. The direct distribution I helped create is back, thanks to global downloading. The stranglehold of the corporates I scorned is breaking down, thanks to crowdfunding and in-game sales. Like those little insects in their underground holes, all we have had to do is wait patiently for the world to turn. And prepare to fly again. 🐛

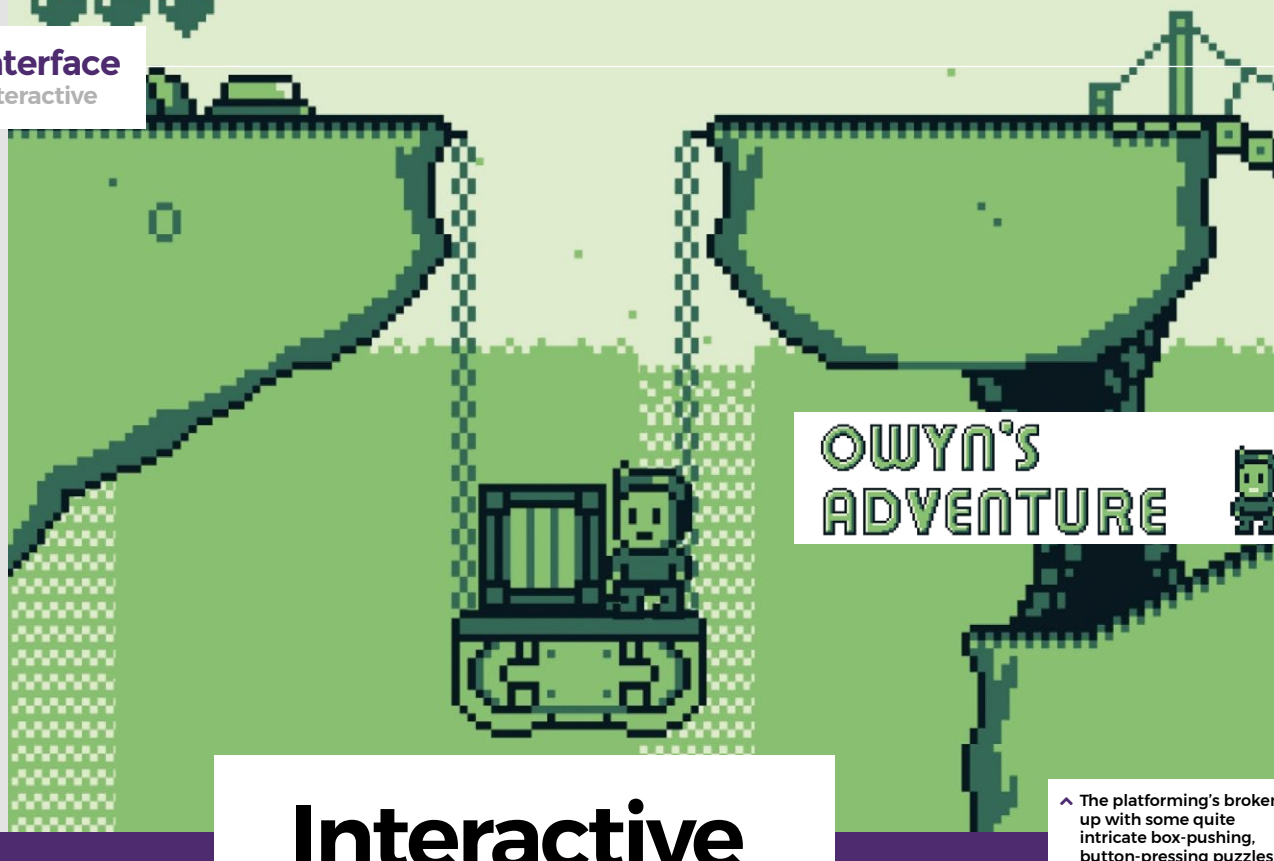
- ▼ *Pimania*, released in 1982, was an adventure game with an eccentric range of objects to interact with, including cuddly toys, musical instruments and pork pies.



LIFE OF PI(MANIA)

Founded in the seventies, Automata UK leapt to fame in the following decade with a string of ambitious and often surreally funny games. *Pimania*, released in 1982, was an adventure game with a real-world prize as a hook: if they successfully solved all the clues, the winner would discover the location of a golden sundial worth thousands. Automata's most famous title, though, was 1984's *Deus Ex Machina*, a kind of experimental, interactive art installation that mixed animation, music and celebrity voices (hello Jon Pertwee). Although it wasn't a hit, *Deus Ex Machina* is still remembered today as one of the most inventive games to emerge from the UK's early games industry. As for the golden sundial: it was finally located one rainy day in Sussex – a full three years after *Pimania*'s release.





▲ The platforming's broken up with some quite intricate box-pushing, button-pressing puzzles.

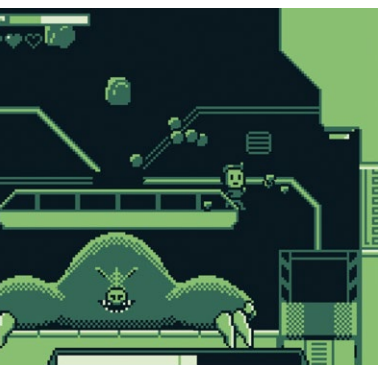
Interactive

Owyn's Adventure, the Game Boy homage created in lunch breaks

Developer Gaz Thomas explains how he made his retro platformer after his lunchtime sandwich

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

▼ The story begins in a fairly typical garden, but heads into far weirder territory as the game progresses.



It goes without saying that making games is incredibly time-consuming. Spare a thought for solo developer Gaz Thomas, then, who somehow managed to make his 8-bit platformer *Owyn's Adventure* in the little time he had left between his day job and looking after his two kids. "From start to finish, it was a year and three months, give or take a few days," Thomas says of his action-puzzler's development period. "With two small kids at home – Owyn nearly three, and Betty ten months – I only really get my lunch breaks to do game development. After I'd finished off my sandwich, that left roughly 30 to 40 minutes a day!"

As you may have gathered, *Owyn's Adventure* is named after – and stars – Thomas' young son. It follows the boy's adventures as he leaves the safety of his home to find his grandfather, only to encounter a faintly surreal garden landscape of platforms, deadly wasps and snails, and assorted traps and puzzles. It's a pleasingly bouncy, light-hearted game, and marked out by its monochrome look: like Christophe Galati's *Save Me Mr Tako*, *Owyn's Adventure* is inspired by the greens and yellows of the Nintendo Game Boy's dot-matrix display. The simple two-button

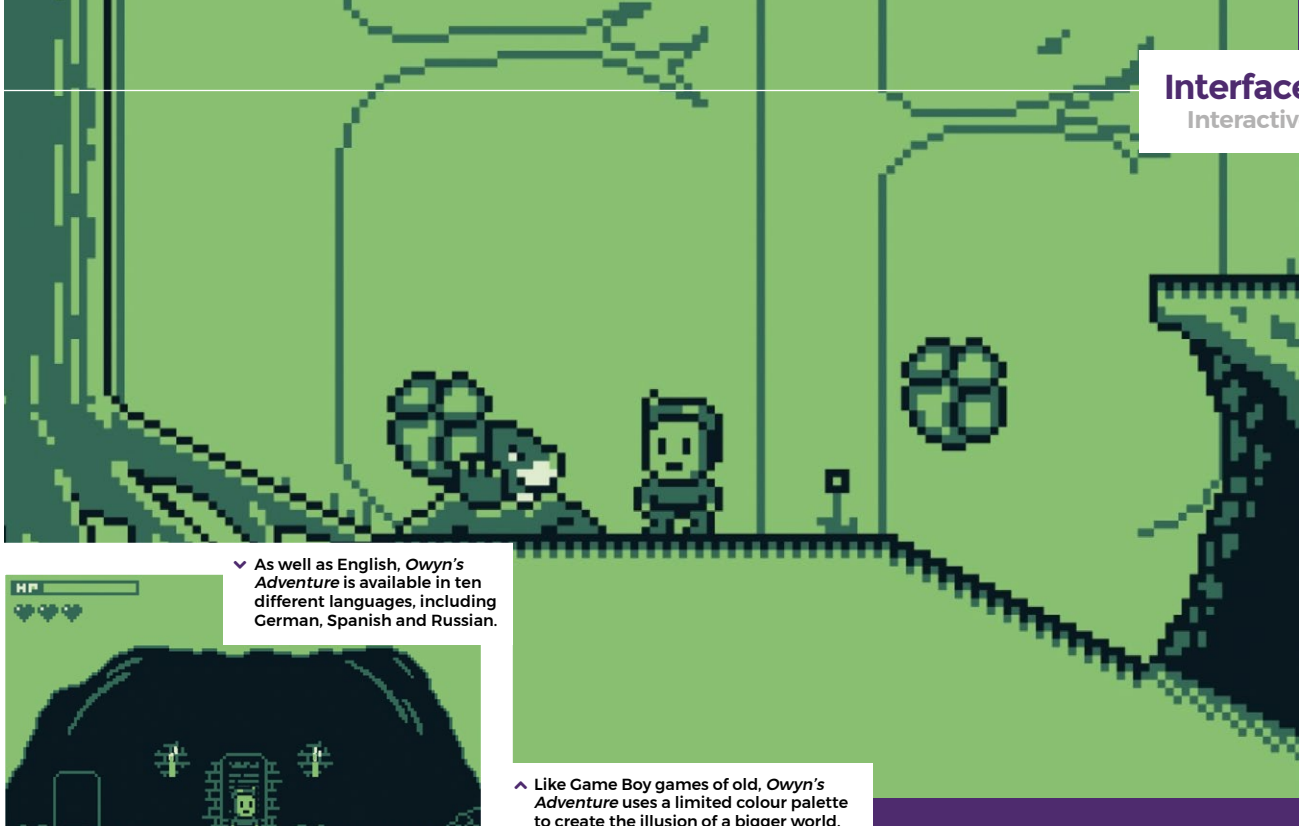
controls and upbeat chiptune soundtrack also fit the retro theme, and like so many original Game Boy games of yore, the action's punctuated by lever-pulling and block-pushing puzzles.

"For me, a good puzzle is one where you've already taught the player the tools they need," Thomas tells us. "You show them what they need to do – take X to Y – but then there's a twist, an unexpected catch to the puzzle that needs to be overcome. When a player gets that penny drop or 'ahhh' moment, that's what I want to see."

UP AND RUNNING

There's a good reason why *Owyn's Adventure's* pixel action is so polished: Thomas spent eleven years in the games industry – eight of them at London's Curve Digital (formerly Curve Studios) – before he left to work at a consultancy. "I had the time of my life, starting in game art then technical art, level design, and then game design," Thomas says. "I really miss working in games, so now it's my hobby."

Owyn's Adventure was built using Unity, with most of the coding handled by using the visual scripting tool, PlayMaker, a tool called Dialogue System for in-game text, and Tiled for designing levels. As for animation, Thomas took a



▼ As well as English, *Owyn's Adventure* is available in ten different languages, including German, Spanish and Russian.

▲ Like Game Boy games of old, *Owyn's Adventure* uses a limited colour palette to create the illusion of a bigger world.



GOING GLOBAL

One unusual aspect of *Owyn's Adventure* is just how many languages it has been translated into. As well as English, it's also ten additional tongues, including French, Spanish, Swedish, Portuguese and Russian. For Thomas, the additional time spent organising all those translations – handled by a mixture of friends, fellow developers and his wife – was well-spent, since it means the game is now available to a truly global audience.

"I'd made an iPhone game a long time ago with a friend, and he got his girlfriend to translate it," Thomas explains. "I think of the £100 we made, half of it was from one of the languages she'd

done! When working at Curve, I remember the production team talking about EFIGS [English, French, Italian, German, Spanish] a lot, so it was

on my mind while developing the game. I used the asset Dialogue System for strings and had a Google doc with all the languages mapped out."

Owyn's Adventure made its debut on itch.io in December last year, bringing to a close a project that first began at a game jam back in 2017. And while fitting its making around daily life wasn't always easy, Thomas still has plans for the game's future, including ports to mobile and Steam, as well as ideas for a bigger, possibly brighter sequel. "I'm thinking about a game set in the same world as *Owyn's Adventure*, but featuring Betty, his sister," Thomas concludes. "And with colour this time – I need a break from pure green..." 🎮

somewhat unusual approach: "I'd first draw each frame as a stick figure with different coloured limbs, copy that frame and then draw the next, get all the frames, and test in Unity. When it felt right, I'd draw the character on top."

Understandably, though, it was simply fitting *Owyn's Adventure* around everything else that proved to be the biggest challenge for Thomas. "The limited time was a blessing and a curse, in that I could only ever get so much done in a day, so I would then be frustrated and desperate to work more the next,"

Thomas says. "It was particularly hard when facing a harder or longer task, as there was the danger that I might not remember where I was the day before. Keeping going was difficult because you're working alone and you start to question yourself."

What spurred Thomas on, though, was showing off the game to other people. "I found putting it in front of others super-motivating. For difficulty, I knew I wanted it to be fun rather than challenging, so I tested it on people with different levels of gaming experience. Over our last family holiday, I got everyone to play it individually, from my gaming brother to my eight-year-old niece. A few non-gamers at work, too."

"With two small kids at home, I only get my lunch breaks to do game development"

8-BIT INSPIRATIONS

"I was aiming to make a platformer that gave the player the feeling of adventuring and wonder, like *Zelda: Link's Awakening* gave me on long car journeys as a kid," Thomas says of his influences when making *Owyn's Adventure*. As for more modern inspirations, Thomas also cites the likes of *Fluidity*, *Limbo* and *Pikmin*, particularly when it comes to his game's leafy setting and puzzles. As for the Game Boy look and feel, Thomas says there's more to the design choice than just nostalgia. "I think there may be something in the limited number of buttons on the Game Boy that gave games a certain clarity or cleanliness in user experience," Thomas tells us. "It's easier for the player to focus on what's on screen than have to keep a mental note of what to press."



► Having sold over 100 million units, it's little surprise that Nintendo's handheld has influenced a generation of new developers.

Online > Television



STEVE MCNEIL

Steve is an award-winning comedian and partnered Twitch streamer, best known for the video gaming TV show he created and appeared in, *Go 8 Bit*.

“Not everyone in TV ‘gets’ games, which can lead to compromises”

Just over five-and-a-half years ago, my show *Go 8 Bit* began its life as a drunken shambles in a student union cupboard at the Edinburgh Fringe Festival, borne from a belief that if funny people competed on games like *Mario Kart* or *Street Fighter*, the inherent entertainment value of video games combined with the performers’ quick wit would be enough to entertain an audience. Less than three years later, *Go 8 Bit* was a shiny-floor TV panel show on the UK channel, Dave. Lest this comes across as self-aggrandising, the show has since been denied a fourth series, so that’s me knackered.

Making a video gaming show for TV comes with a number of complexities which we’d either not considered, or underestimated. For example, as a show which existed solely for entertainment (despite my protestations, Susan Calman dressed as an extra from *Baywatch* was deemed to have zero academic value), we were unable to use any games without first securing approval from their creators. When it comes to 30-year-old Spectrum games, it can be virtually impossible to even work out who owns them and, from a legal standpoint, “it’ll probably be fine” is apparently not a plan. More problematic was the unavoidable truth that not everyone in TV ‘gets’ games, which can lead to some unfortunate compromises to the detriment of the overall product. I, for example, actually felt the screen we played the games on shouldn’t have lag. I’m a maverick.

Alongside *Go 8 Bit*, I found myself increasingly immersed in the world of online gaming, and eventually began to dabble in Twitch streaming from the comfort of my own home. Over the last 18 months, I’ve grown to love the platform more than TV, as it gives me complete creative control over what I make, when I make it, and a direct connection with the people that choose to engage with it. I’m a bit of an anomaly on Twitch, being neither young, nor particularly good at the games I choose to play, but I’ve somehow found enough kindred spirits with similar interests for



^ Funny people competing: that’s pretty much all you need for pure entertainment.

it to be a viable source of income. In fact, last year, whilst we didn’t film any more episodes of *Go 8 Bit*, a combination of streaming and live shows allowed me to earn more income than I’d done the previous year when the TV show was made. Whilst TV has huge budgets, it also has huge costs, and my modest audience online and around the country was able to directly fund my delusion without any intermediaries creaming their slice off the top.

Of course, it’d be nice to be able to bring the production values of TV shows like *Go 8 Bit* to an online audience in a form other than eSports, but with that scale comes complexity, compromise and cost. Until someone works out how to do it successfully, I’m happy just playing *Mario* badly in my grandad’s old chair. 🎮



^ Things certainly weren’t lacking in effort when it came to *Go 8 Bit*’s stage dressing.

Toolbox

The art, theory and production of video games

28. CityCraft

Pitfalls to avoid when making a city

30. Make a text adventure

Hit goblin, go north

32. Source Code

R-Type's use of modular sprites

34. Localisation

Prep your game for a global reach

38. Improve your levels

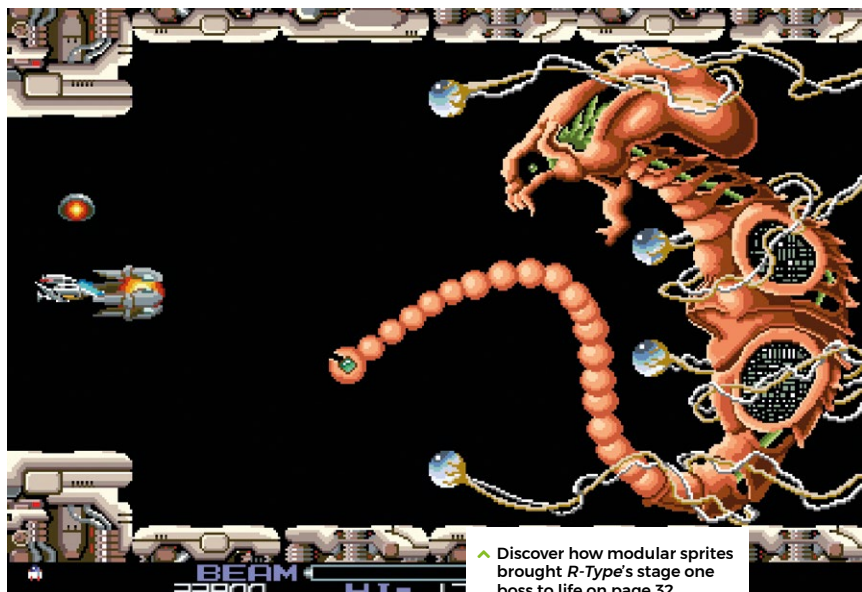
Six tips to make your designs shine

40. Making Anew

An indie dev shares his advice

42. Directory

Time-saving tools for indie devs



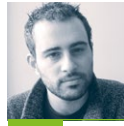
^ Discover how modular sprites brought *R-Type's* stage one boss to life on page 32.

✓ Greyboxing will help you test level designs more quickly. Find out more on page 38.



CityCraft: How not to get cities wrong

Common mistakes in imaginary city building, and how to avoid them



AUTHOR
KONSTANTINOS DIMOPOULOS

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

Fantasy Wells, Medieval Waters

Providing people with drinking water has defined the locations and infrastructure of cities for millennia, and yet the vast majority of fantasy towns (often inspired by medieval urbanism) seem to ignore this fundamental need. There are no wells or aqueducts (not even springs nor close-by rivers) to be found, even if these are crucial elements for a settlement's survival, and also important places for gossip, conversation, and trading. If wells feel banal, how about water merchant caravans, magical humidity catchers, geysers surrounded by temples, and tanks, cisterns, and pools to preserve rainwater in?

There are fundamental, methodological mistakes one can make when attempting to create a game city, and then there are other elements that could go wrong or simply remain ignored that won't necessarily break your world building on their own. Allow these to pile up, though, and both your players' suspension of disbelief, and your setting's desired effect, can suffer.

Assuming you won't ignore such crucial concepts as urban functions or fundamental realism, here's a (far from exhaustive) list of the problems, mistakes, and omissions a designer of video game cities should keep in mind.

COMMON MISTAKES

Game cities often get vehicular – and human – traffic wrong, either by forgetting it completely or approaching it inadequately. Transportation

networks and vehicles are usually an afterthought, despite strongly influencing the image, and function, of any real-world city. Traffic in game cities is commonly too scarce, overly simplified, and cities often lack ranked road networks: a modern town should always have a hierarchy of roads from the local to the interurban feeding traffic into each other. All road networks should have sensible, functional shapes, that can also help with modelling the movement of crowds.

Then there are the obvious mistakes of topology, and spatial organisation: of how districts and sub-areas are placed in relation to each other. An airport shouldn't be located downtown, a city's core is always denser than its suburbia, and polluting factories in a rich neighbourhood or a prison next to a palace will feel out of place.

Trying to simulate city life via eternally re-enacted, animatronics-style interactions rarely works, and neither does having a handful of NPCs randomly moving around or just standing stock still. On the other hand, urban vignettes breathe life into a city, and people do sometimes just stand there. To do this right, keep track of your pool of little vignettes, and make sure each is seen once (or at least not too often). Failing to notice an instance of a little girl buying a newspaper is much better than seeing her do this every time you turn a corner.

Handling traffic, on the other hand, depends on settlement scale. A village's population



▲ *Ghost in the Shell*, showing just how beautiful the mundane can look. Do not ignore it.



could be given individual itineraries, whereas civic masses should move according to general rules, which are often group-specific. Workers, for example, commute towards factories in the morning, and back home in the afternoon, while medieval guards patrol walls at regular intervals, and cooks buy ingredients before breakfast.

On a related subject, scale is also easy to get wrong. City size, functions, and variety are all correlated. A city with only one butcher's shop would feel plain wrong, even though it would seem perfectly logical in a village. A mistake like this can be fixed with a few words: don't direct the player to the butcher's, but to John's – the best place to grab meat in all the city, and one of many. Or direct them to the only place open at, say, six in the morning. Similarly, it's OK for the tiny town of Thimbleweed Park to feature a single factory, but it would be preposterous for the city in *GTA V* to contain just one bar.

“Cities are the battlegrounds of classes and social groups”

THE USUAL OMISSIONS

The lack of interesting topographies, and natural geographies, can also hurt virtual cities. A featureless landscape is boring, and, besides, dramatic topography makes orientation easier. Topography has also been crucial in historically determining urban shapes, structures, and location, and can thus act as a strong world-building tool, as differences in elevation can be associated with class, ridges can shape defences, and picturesque hills can add charm.

While flat cities do exist, homogeneous ones do not. All urban centres are divided (whether porously or with clearly defined borders) in a multitude of ways according to class, land use or prevalent activity, architecture, proximity to sub-centres or centres, construction age, and even – if applicable – relative position to the walls. Obviously, districts can combine activities: industry can appear in poorer residential zones, exclusive marketplaces in posh areas, and guild halls next to Papal palaces.



^ Cities take decades or even centuries to build, yet games almost never show us medieval or fantasy buildings under construction.

One of the crucial elements imaginary cities commonly lack is a hinterland: a wider natural or man-made geography. Cities, you see, do not exist in a vacuum. They are parts of kingdoms, empires, states, and federations.

They exist in particular climates, are connected to other settlements, and influenced by geopolitics. What's more, their tissue always rests

on a functioning infrastructure (glimpses of which make them more believable), and they should include public spaces, street furniture such as benches, dumpsters, or lights, and of course dominant constructions like cathedrals, palaces, or corporate headquarters to express local ideology.

A WORK IN PROGRESS

Cities are always dynamic works in progress. Created by complicated histories, they are the battlegrounds of classes and social groups, and the places of countless human interactions, and this dynamism has to be reflected in the built environment. So, mix old and new buildings of varying styles, have your central temple be under construction, and remember that the average life-expectancy of a New York City building is less than 25 years. Think of gentrification, of how lofts changed their function in a couple of decades, or of how radically a square might reinvent itself for a major festival. Have cranes, and images of construction infuse your cities with a sense of history in the making, and always keep in mind that all settlements have histories; histories that have to be researched, possibly made up, and carefully layered. 🗺

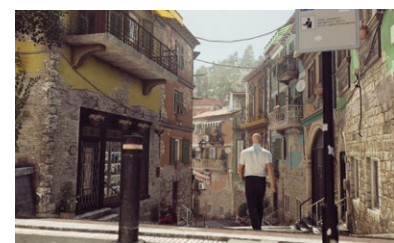
✓ L-shaped, right-angled roads are not only ugly; they are also very rare in real life, dangerous to drive on, and not particularly practical.



Naming Gone Bad

Just as the names City 17, Rapture, Oneiropolis, and Raccoon City instantly set a tone and evoke an initial image, naming your city 'The City' will probably put people off, or at least fail to intrigue them. Such naming feels rushed, dull, and is sadly overdone; it is though still better (and safer) than using something obviously Latin to describe a Chinese town. Oh, and please don't overuse the cyberpunk trope of Neo New York, Neo Tokyo and New London. It has been done to death, and civic naming never worked like that. York, for example, did not eventually become New York.

✓ The dramatic topography of Sapienza adds both character, and murder opportunities, to *Hitman's* urban environment.



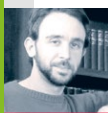
```

>
You are in the woods. There are lots of trees.
north: The lake
> north
You are by the lake. It is very watery.
south: The woods
> south
You are in the woods. There are lots of trees.
north: The lake
> █

```

^ You could try expanding the example code with additional locations and commands.

In the game: making a text adventure



AUTHOR
ANDREW GILLETT

Andrew explains how to make a simple text adventure in Python – and what pitfalls to avoid

The first game I ever wrote was named *Pooh*. It had nothing to do with the bear. In September 1982, I was four years old, and the ZX Spectrum home computer had just been released. It was incredible enough that the Spectrum let you play games on the TV, but like most home computers of the time, it also came with a built-in language called BASIC, and a manual which explained how to program it. In my first game, *Pooh* (the title was a misspelling), the player controlled a baby, represented by a pound sign, and had to guide it to a potty, represented by the letter O. There were no obstacles, no enemies, and if you tried to walk off the screen, the program would stop with an error message. I didn't have any idea how to create a graphical game more complex than *Pooh*. I didn't even know how to display a sprite on the screen.

Instead, I focused on writing text adventures, where the game describes scenes to the player ("You are in a

comfortable, tunnel-like hall. You can see a door," from 1982's *The Hobbit*) and the player enters commands such as "Go through door" or "Kill goblin with sword." Although this type of game is comparatively easy to write, I implemented it in the worst way possible. The code was essentially a huge list of IF statements. Each room had its own set of code, which would print out a description of the room and then check to see what the

"The correct way would have been to separate my code and data"

player typed. This 'hard-coding' led to the code being much longer than necessary, and more difficult to maintain.

The correct way would have been to separate my code and data. Each room would have had several pieces of data associated with it, such as an ID number, the description of the room ("You are in a small cave"), an array of objects which can be found in the room, and an array of room

numbers indicating where the player should end up if they try to move in a particular direction – for example, the first number could indicate which room to go to if the player enters 'NORTH'. You'd then have the main game code which keeps track of the room the player is currently in, and looks up the data for that room. With that data, it can then take the appropriate action based on the command the player typed.

GETTING IT RIGHT

The code to the right shows how to implement the beginnings of a text adventure game in Python. Instead of numeric IDs and arrays, the code uses string IDs and dictionary data structures, where each piece of data is associated with an ID or 'key'. This is a more convenient option which wasn't available in Spectrum BASIC. We first create a list of directions in which the player can potentially move. We then create the class `Location` which specifies each location's properties. We'll store a name, description and a dictionary data structure which stores



^ *The Hobbit*, released in 1982, was widely praised for its intuitive parser.

the other locations that the current location is linked to. For example, if you go north from the woods, you'll reach the lake. The class includes a method named `addLink` which adds entries to the `linkedLocations` dictionary, after checking that the specified direction and destination exist.

Following the class definition, we then create a dictionary named `locations`. This has two entries, with the keys being `woods` and `lake`, and the values being instances of the `Location` class. Next, we call the `addLink` method on each of the locations we've just created, so that the player will be able to walk between them. The final step of the setup phase is to create the variable `currentLocation`, specifying where the player will start the game.

We then reach the main game loop, which will repeat indefinitely. We first display the description of the current location, along with the available directions in which the player can move. Then we wait for the player to input a command. In this version of the code, the only valid commands are directions: for example, type 'north' at the starting location to go to the lake. When a direction is entered, we check to make sure it's a valid direction from the current location, then update `currentLocation` to the new location. When the main loop restarts, the description of the new location is displayed.

I moved on from the ZX Spectrum eight years after my dad first unpacked it. Despite the poor design of my code, I'd learned the essentials of programming. Ten years later, I was a game developer. ☺

Further reading

If you're keen to learn more about making a text adventure in Python, you could check out Phillip Johnson's guide to the subject, *Make Your Own Python Text Adventure*. The same author has also written a condensed version of the same guide, which you can find online at wfmag.cc/TEXT.

A text adventure in Python

Here's a simple text adventure that you could modify with your own locations and more. To run it on your device, you'll need to install Python. Find out how at wfmag.cc/AWeU

```
directions = ['north', 'south', 'east', 'west']

# Data structure to store details of each location in the game
class Location:
    # Constructor - set up
    def __init__(self, name, description):
        self.name = name
        self.description = description
        self.linkedLocations = {} # Empty dictionary - will store which locations
are linked to which other locations
    def addLink(self, direction, destination):
        # Add link to linkedLocations dictionary (if the specified direction and
destination are valid)
        if direction not in directions:
            raise ValueError('Invalid direction')
        elif destination not in locations:
            raise ValueError('Invalid destination')
        else:
            self.linkedLocations[direction] = destination

# Dictionary with location ID strings as keys and Location objects as the values
locations = { 'woods':Location('The woods', 'You are in the woods. There are lots
of trees.'),
              'lake':Location('The lake', 'You are by the lake. It is very watery.') }

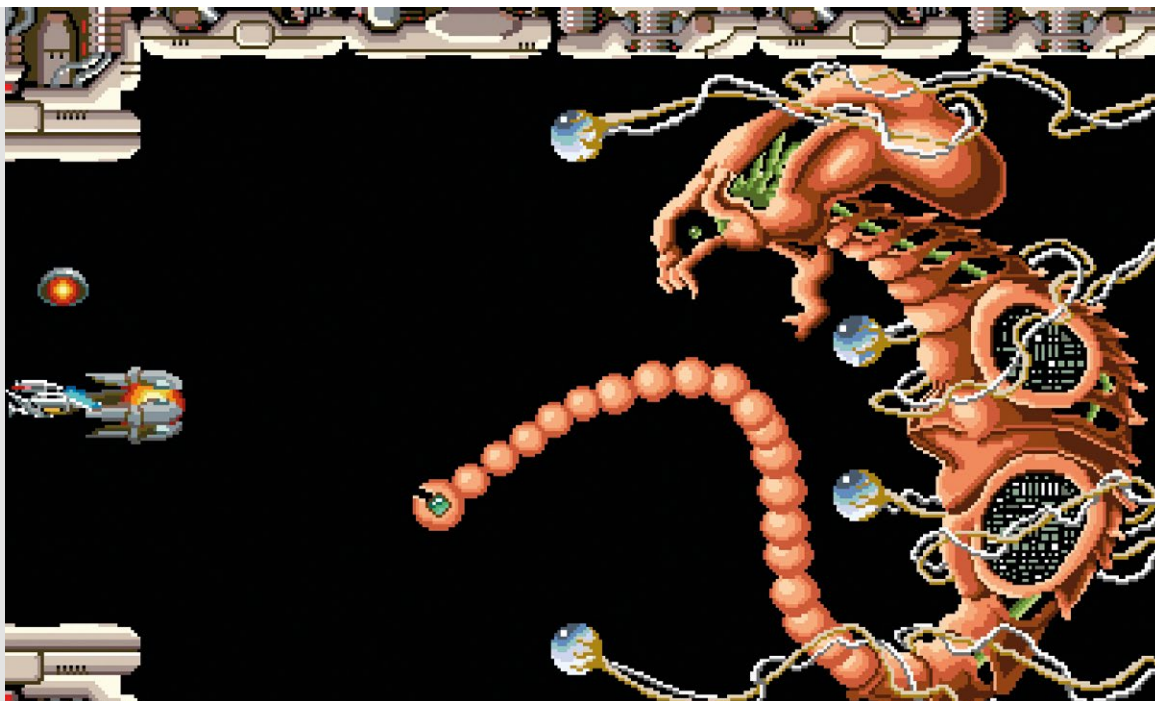
# Join the two locations together
locations['woods'].addLink('north', 'lake')
locations['lake'].addLink('south', 'woods')
# Player will start in the woods
currentLocation = locations['woods']

# Main game loop
while True:
    # Display description of current location
    print(currentLocation.description)

    # Display neighbouring locations
    for linkDirection, linkedLocation in currentLocation.linkedLocations.items():
        print(linkDirection + ': ' + locations[linkedLocation].name)

    # Read player input
    command = input('>').lower()
    if command in directions:
        if command not in currentLocation.linkedLocations:
            print('You cannot go that way')
        else:
            newLocationID = currentLocation.linkedLocations[command]
            currentLocation = locations[newLocationID]
    else:
        print('Try one of: ' + ', '.join(directions)) # Show list of directions,
separated by commas
```





Source Code

▲ *Dobkeratops*: stunning in 1987, still a cool design decades later.

R-Type's spectacular use of modular sprites



AUTHOR
RYAN LAMBIE

The 1987 shooter *R-Type* used multiple sprites to create the illusion of a much larger end-of-level boss. Here's how to recreate the effect

Still images don't quite do it justice. The screen faded to an inky black; an ominous new tune began; then the monster scrolled into view. The tail appeared first, lashing up and down; then in came the rest of the abomination, with its skeletal frame, snapping jaws and floating eyeballs. It was a startling sight, particularly back in 1987, when *R-Type* was still a newcomer to the world's amusement arcades. Japanese developer Irem evidently knew that it had something special on its hands with this huge, screen-filling monster, which they called Dobkeratops: they placed it right at the end of stage one, maximising the chance that players would get to it; much of the game's promo artwork also featured the creature's hideous likeness.

Aside from the striking graphic design – which owes a debt to *Alien* artist Hans Ruedi Giger – *R-Type*'s level one boss showcased

Irem's technical ingenuity. Characters the size of Dobkeratops were vanishingly rare in eighties games, largely because hardware and memory restraints made putting them on the screen so difficult. What Irem did, though, was use several smaller sprites to create the illusion of a single, animated monster.

“Characters the size of Dobkeratops were rare in eighties games”

Look again at Dobkeratops in action, and you can see that only three elements move: its jaw, which moves up and down; a green, parasite-like organism that emerges from its stomach (another nod to the 1979 movie, *Alien*), and most eye-catchingly, the long, whipping tail. The rest of the beast is essentially a static image, stored in the

hardware's memory in sections. The jaw and stomach-bursting parasite are dealt with in just twelve frames of animation altogether. Similarly, the tail actually consists of 18 relatively tiny sprites, which are programmed to whip and curl in a smoothly organic fashion, like a horrifying string of beads. Unless it was pointed out to the average player grappling with *R-Type* back in the eighties, they probably wouldn't have noticed Irem's sleight of hand.

ORGANIC ORIGINS

Ingenious though it was, *R-Type* built its success on the foundations laid by other games. In 1985, Konami launched the seminal *Gradius* – another scrolling shooter that, like *R-Type*, saw a lone pilot fly through a succession of hostile environments, blasting aliens. While that game was in development, its director, Hiroyasu Machiguchi, said that he wanted to create a stage with an organic



^ Chained together and animated across a sinusoidal curve, these little sprites create the effect of a whipping, alien tail.

How it all works

To illustrate how Irem created that tail, we asked Daniel Pope to come up with a piece of code in Pygame Zero – you can find this on the right. It uses the same principle that Irem came up with: multiple sprites are drawn on the screen and then animated using the sine function to create a whipping motion. “So that the components of the chain appear to stay connected, we keep the pieces at a constant distance (`SEGMENT_SIZE`) from each other, and only control the angle between each piece,” Daniel explains. “Given an angle, we use the trigonometric functions `sin()` and `cos()` to work out the Cartesian (x and y) position of the segment, and each segment is positioned relative to the previous one.”

feel, since most of the other stages featured mechanical enemies. While wondering how to make something move organically on eighties hardware, Machiguchi reportedly came up with an idea: he told his designers to draw a sprite “that looks like a pachinko ball”, and use multiple instances of them to create a moving tentacle.

In a 2006 interview (translated by the good people at shmuplations.com), designer Kengo Nakamura recalls that it took two days of experimentation with this concept, but the results were immediately striking. “After about two days, I believe, we came up with the creeping movement of the tentacles,” Nakamura said. “We made each little pachinko ball in his arm move individually, and everyone was amazed at how, in a short time, the design had become so realistic, disturbing and gross.”

Gradius's tentacled monsters were tucked away on stage five, and therefore less easy to reach than *R-Type*'s Dobkeratops. *Gradius* almost certainly laid the groundwork for *R-Type*, however; at least one of *R-Type*'s developers was a self-avowed fan of the game. If anything, though, *R-Type* uses the concept of organic movement through modular sprites to even greater effect; the expert sprite design, combined with that aggressively lashing tail, resulted in one of the greatest area bosses ever conceived. 🗿

An R-Type tail in Python

Here's a code snippet that shows an *R-Type*-esque modular tail working in Python. The code requires a pair of sprites to work: `tail_piece.png` and `tail_hook.png`, which you'll find (along with the code itself) at the GitHub link on the right. The code also requires you to install Pygame Zero – you can find full instructions at wfmag.cc/XVIIeD



```
from math import sin, cos

# Constants that control the wobble effect
SEGMENT_SIZE = 50 # pixels from one segment to the next
ANGLE = 2.5 # Base direction for the tail (radians)
PHASE_STEP = 0.3 # How much the phase differs in each tail piece (radians)
WOBBLE_AMOUNT = 0.5 # How much of a wobble there is (radians)
SPEED = 4.0 # How fast the wobble moves (radians per second)

# Dimensions of the screen (pixels)
WIDTH = 800
HEIGHT = 800

# The sprites we'll use.
# 10 tail pieces
tail = [Actor('tail_piece') for _ in range(10)]
# Plus a hook piece at the end
tail += [Actor('tail_hook')]

# Keep track of time
t = 0 # seconds

def draw():
    screen.clear()
    # First draw the even tail pieces
    for a in tail[::2]:
        a.draw()
    # Now draw the odd tail pieces
    for a in tail[1::2]:
        a.draw()

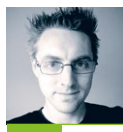
def update(dt):
    global t
    t += dt
    # Start at the bottom right
    x = WIDTH - SEGMENT_SIZE // 2
    y = HEIGHT - SEGMENT_SIZE // 2
    for seg, a in enumerate(tail):
        a.pos = x, y

    # Calculate an angle to the next piece which wobbles sinusoidally
    angle = ANGLE + WOBBLE_AMOUNT * sin(seg * PHASE_STEP + t * SPEED)

    # Get the position of the next piece using trigonometry
    x += SEGMENT_SIZE * cos(angle)
    y -= SEGMENT_SIZE * sin(angle)
```

The dark art of localisation

Here's how to get your game ready to sell to the whole world



AUTHOR
CHRIS PAYNE

Chris Payne has coded games for over 20 years, mostly at TT Games. At Quantum Soup, he develops original titles and helps out with client projects such as Weather Factory's *Cultist Simulator*. quantum-soup.com

Localisation is a useful word that encompasses not just translating text, but also making the myriad other adjustments required to cater to players in other cultures.

It's true that localisation requires additional work for developers, and it's not as exciting as, say, martial arts. But you can sell more games if you make them easier for non-English speakers to play – a lot easier. China recently overtook the US as the largest market for mobile games; India makes up ten percent of the world's gamers. It's not all about money, though. Localisation also gives you an insight into other cultures and makes those players feel appreciated. In this article, I'll tell you all about it, and how you can get your game ready for a global audience.

GOOD STRONG WORDS

Text is surprisingly complicated, so bear with me during this technical deep dive. The text in a game is often called 'strings' after their technical structure: a row of bytes in memory representing letters, numbers, and symbols. In fact, a byte is often called a 'char' in coding, because it's so frequently used to store a text character.

A byte can, however, only store 256 different values. Sounds like plenty, right? That's room for 26 letters in the alphabet, ten numerals, a scattering of punctuation, and so on. Unfortunately, when you start looking at other languages, it's clearly not enough. Cyrillic

languages like Greek and Russian use a whole extra alphabet, and as for the ideograms of Chinese, Japanese, and Korean – well, there are thousands even in the 'simplified' set. The Unicode Consortium currently defines 137,000 code points, including Egyptian Hieroglyphs and Elder Futhark runes (yes, that's a real thing). How do you handle so many possible characters?

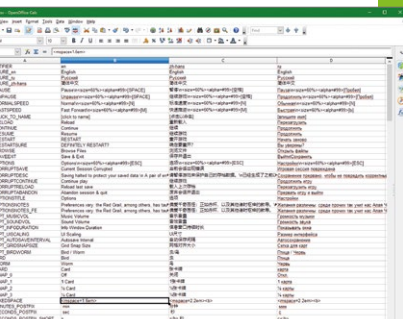
There have been a bunch of different solutions, but the clear winner is Unicode Transformation Format 8-bit, or UTF-8 to its friends. This is a clever encoding where each character uses only as many bytes as it needs, so English will still be compact, but Chinese can use three or four bytes to define glyph number 20,144 (京). Most importantly, you can mix and match them in the same file.

DANCING ON TABLES

In order to support a variety of different languages, you need to put all your text into an independent file so that you can change *all* the strings just by updating that file – avoid having any human-readable text in your code or entered into the editor. You can save any text file with UTF-8 encoding, but I like to use a simple spreadsheet format called 'comma-separated values' or CSV, which looks like this:

```
IDENTIFIER,en,zh-hans,ru
UI_RELOAD,Reload,重新载入,Перезагрузить
UI_CONTINUE,Continue,继续,Продолжить
```

World's most exciting screenshot. The CSV format is much easier to read when edited as a spreadsheet.



It looks a bit messy in plain text, but it can be opened as a spreadsheet in Excel or OpenOffice Calc, so that it's nicely tabulated with each language in its own column. If you want to include commas within strings, you'll need quotes around the entire string, and make sure your CSV parsing code treats everything inside the quotes as a single string.

Anywhere you need to display a string, you should use the string ID to look it up from this table – even for English. This ensures that all text is treated equally and any technical issues will show up in English, too.

It's useful to keep all the text in one file like this. If all languages are side by side, you can easily see which strings have been translated, and your translators can see if you've added new ones that they need to translate. Your native and localised string data can never get out of sync if it's all together. In addition, as a text file, it will play nicely with source control systems, so you'll be able to examine the file history and see who changed what and when.

So, we've got a UTF-8 CSV file containing all our translated text. Now, let's get it on the screen...

SILENTLY JUDGING YOUR FONT CHOICE

A distinctive font adds flavour and style to a game – just be careful to keep it legible. Really fancy fonts might work well for titles and headings, but for description text you need something easy to read, above all. So you might need several fonts just for English, as well as fonts that support Cyrillic or CJK (Chinese, Japanese, Korean) glyphs.

Once you've chosen a suitable font – most likely a TrueType font – you need to import it into the game. TTF is a vector format, which means it can scale smoothly, but it's very expensive to render.

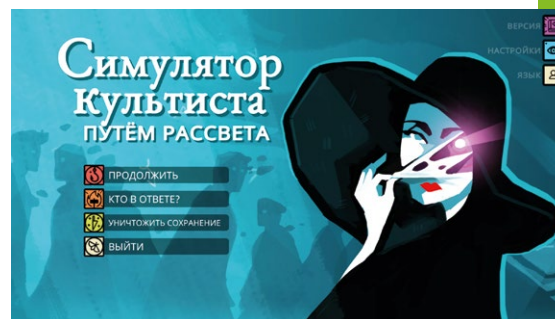
Most game engines will render off an example of each unique character into a font atlas texture, and then slice and dice that to render strings as fast as possible. That works well for Western languages, but how do you cope with many thousands of Chinese glyphs?

Well, all those glyphs have very specific meanings compared to the Lego bricks of the Western alphabet. So your game probably only

“You'll sell more games if you make them easier for non-English speakers to play”



◀ The stylish title screen uses one in-game font and a logo texture. Two more fonts are used in gameplay.



◀ The long Russian text on the third button has been scaled to fit.



◀ The subtitle does use an in-game font but was duplicated per language so each could be positioned to match the logo.

needs a small subset of glyphs. The solution to this is to extract a list of all the characters that are actually used in your CSV table, and tell the game engine to include only those characters in the texture.

Different languages can use wildly different length sentences to communicate a particular message. German, with its fondness for portmanteau words, often ends up a lot longer than the equivalent in English. This means it's essential to design your user interface with enough space to handle variable length strings, and to scale the text down if it still won't fit.

NUMBERS RULE THE UNIVERSE

Obviously, text varies across cultures, but numbers are numbers, aren't they? Not quite ➔.

ASCII ORIGINS

For a while in the 1970s, we got by with 128 characters defined as the ASCII character set: the American Standard Code for Information Interchange. This actually originated long before microcomputers, and was first used in telegraph communications. You've probably heard the acronym used to describe ASCII art – pictures made of that core set of letters and symbols that we had to make do with before computers could handle complex things like graphics or colours. When I was a lad, all this was green screen terminals.

Toolbox

The dark art of localisation



▲ This special background image is one huge texture, hand-customised per language.

While the numerals themselves may not change, different cultures favour different separators for currency, time, or even just very large figures. Some put currency symbols before the number, some after. Finally, of course, they don't all use 'm' and 's' for minutes and seconds.

This is mostly handled by storing a 'price' or 'timer' template in the CSV file so that it can be customised per language.

“Travel broadens the mind, but localisation is almost as good”

I JUST LOOK AT PICTURES

Why would graphics need localising? Well, sometimes you have text in images: graffiti, road signs, messages scrawled in blood... in *Crash Bandicoot: The Wrath Of Cortex*, we had a lot of TNT crates – an acronym that is familiar to Western audiences even if they're not English speakers, but meaningless to Japanese players. For the Japanese release, we replaced the TNT texture with a bomb icon so players knew what behaviour to expect.

Cultist Simulator also has a handful of images with text in them, which need to be swapped. Most are loaded on demand so it's easy to tweak the file name to load the image from a different folder. If no file exists in that folder, it reverts to the native data, so we don't need to duplicate images unless they actually contain text. One of the tarot card titles has a devious Russian translation, using careful kerning (spacing between letters) and some gunk in the frame to make one letter ambiguous... this creates a sinister double meaning that only works in Russian.

AUDIO

Since audio is usually loaded on demand, it's easy to divert the code to use a different set of

voice samples with matching names. As with text, though, the code needs to measure the sample duration to know how long to display subtitles for.

In some cases, you might need to play different lip-sync animations to match the localised audio. CD Projekt's *Witcher 3* has a very smart cutscene system which analyses the length of all the voice samples in a cutscene and adjusts the animation speed to match.

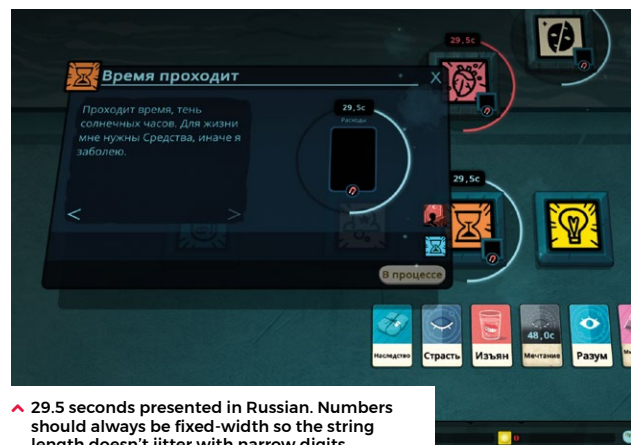
Subtitles for audio are surprisingly complex. For a start, make sure they're big enough to read from across the room. Different colours for different characters can be useful if the speaker is off-screen or sharing the screen with other characters. Subtitles in VR are even harder, as

it's most natural to place the subtitles in the 3D world – which is hard to do automatically in case they clip through scenery. *What Remains*

of Edith Finch, meanwhile, executes in-world subtitles beautifully, aligning them artfully to furniture and walls to guide the player through the environment.

TAKE BACK CONTROL

Even with identical hardware, control schemes can vary across cultures. In the West, we're used to using the bottommost face button to confirm, and the rightmost face button to cancel. In Japan this preference is reversed. This is best hard-coded into an input layer between the game and the hardware, so that you don't have to make the same change in every single menu screen.

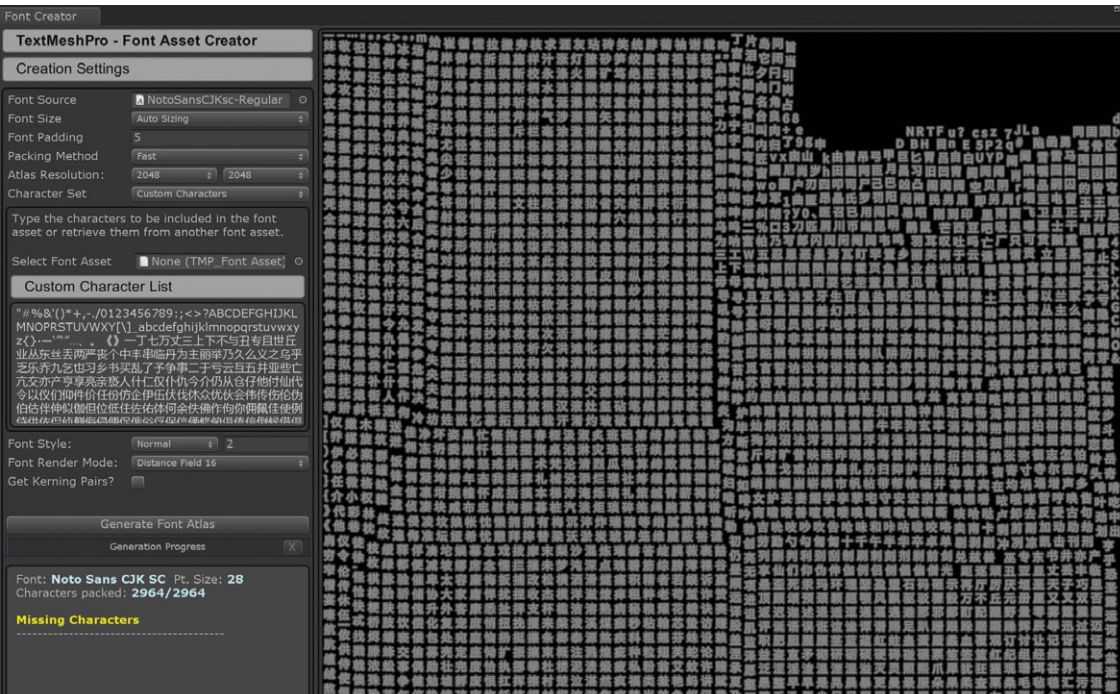


▲ 29.5 seconds presented in Russian. Numbers should always be fixed-width so the string length doesn't jitter with narrow digits.

NO TOFU

Be careful choosing fonts – even fonts that look fine in English might not include all the necessary accents for European languages, resulting in little 'tofu' squares replacing some characters. You must also check the licensing terms; some fonts are free for personal use or even fully open source, but many require purchase and crediting if you plan to include them in a commercial game.

A good starting point is Google's family of Noto fonts (short for No Tofu). These are completely free and support every character set you can imagine, including the aforementioned Elder Futhark runes.



◀ The Chinese font atlas for *Cultist Simulator*, a game of rich descriptions requiring nearly 3000 glyphs – still a fraction of the full Simplified Chinese set.

DIFFERENT VISIONS OF LIFE

There are a host of additional small differences to keep in mind, too. Red is a warning colour in the West, but a lucky colour in China. China has a strict censor which forbids content that criticises the nation, or promotes drugs, gambling, cults, or superstitions (it's going to be interesting to see what they make of *Cultist Simulator*).

In Japan, cartoon characters with only four fingers are offensive because missing fingers are strongly associated with abuse of the working classes. Until 2018, swastikas were banned in German video game releases, even in the correct historical context. Research the market you're aiming for, and be prepared to make adjustments to cater to that audience.

LANGUAGE IS WINE UPON THE LIPS

Travel broadens the mind, but localisation is almost as good. Localising a game forces you to empathise with speakers of different languages, and teaches you all sorts of minor linguistic and cultural quirks. It's a great reminder that no matter what language we speak or where we live, we're all human beings, and enjoy the same things once you get past those superficial differences – and that is surely worth celebrating. 🍷

JOINED UP

The trickiest major language to localise well is Arabic. It uses a unique alphabet, written right-to-left (except for numbers or foreign words). Not only that, but it's a cursive script – meaning it's joined-up so letters change depending on what letters they're adjacent to. There are many ways to present Arabic incorrectly. I'd recommend hiring a programmer who already knows the language so that they can test their own work, or be prepared to get extremely friendly with your Arabic-speaking QA team.

CRASH COURSE

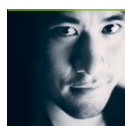
Crash Bandicoot also presents a rare localisation problem: all the letters are 3D models rather than sections of a 2D texture. This wasn't a big deal for the European languages, but for the Korean version, the font scene became very large indeed – even for the minimal text required in a platform game. This took a long time for the artist to create (based on the subset of ideograms actually used in the game) and required some additional optimisation to fit it all into memory.



◀ *Crash Bandicoot*'s 3D text and acronyms painted into textures presented localisation challenges.

6 tips for improving your level designs

Just starting out in level design? Then here are six tips to get you started



AUTHOR
STEVE LEE

Steve Lee is a former designer at studios including Arkane and Irrational, and is now freelancing and consulting. doublefunction.co.uk

In my previous column, I talked about getting into level design the way I did – using the level editors for games you can get for free – and how I still recommend people get into it today using something like Hammer, the editor that comes with *Half-Life 2* (and *Portal*, *Team Fortress 2* and *Left 4 Dead*).

This month, I wanted to follow that up with a series of simple but important tips on how to approach the level design process, for any of you who are giving it a go. Sound good? Good.

1. Know your goals

This might sound obvious, but before you start doing things, it really helps to understand what you're trying to do. Yes, you're trying to make a level – but what kind of level? Which game is it for? What is it about? Is the focus on exciting, cinematic action gameplay? Mind-bending puzzles? Subtle, engaging interactive storytelling? What kind of experience do you want players to have? How is this level similar to what has come before, and how is it unique? Are you making the level to convince companies to hire you as a level designer, and if so, which ones?

These are questions that it's worth having answers to. Always have sensible and specific goals, and consciously design your project or level to achieve them.

2. Keep your projects small

Speaking of sensible goals, try not to fall into the classic trap of being too ambitious with your project – especially if this is your first one.

A common piece of advice in game development is to take your estimate of how long you think a project will take, and double it (at least).

Game dev and level design is hard, and it can take a long time to try ideas out, see which ones work, and which ones don't. By keeping your projects small, everything becomes faster, you increase the chances that you'll finish them, and you give yourself more time to not only make things, but make them really good.

3. Greybox first, get playable quickly

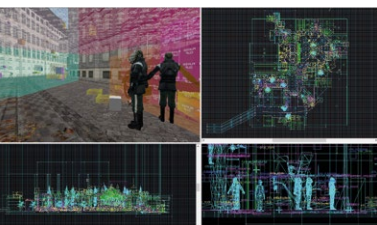
Another classic trap in game development is going overboard and too wrapped up in making stuff, or polishing little things that don't really matter, forever – losing track of the bigger picture and how all things fit together (if they do at all). To avoid this, it's important to work broad strokes to begin with, blocking things out and prototyping in a quick and functional way early on, trying things out and seeing what works (and just as importantly, what doesn't).

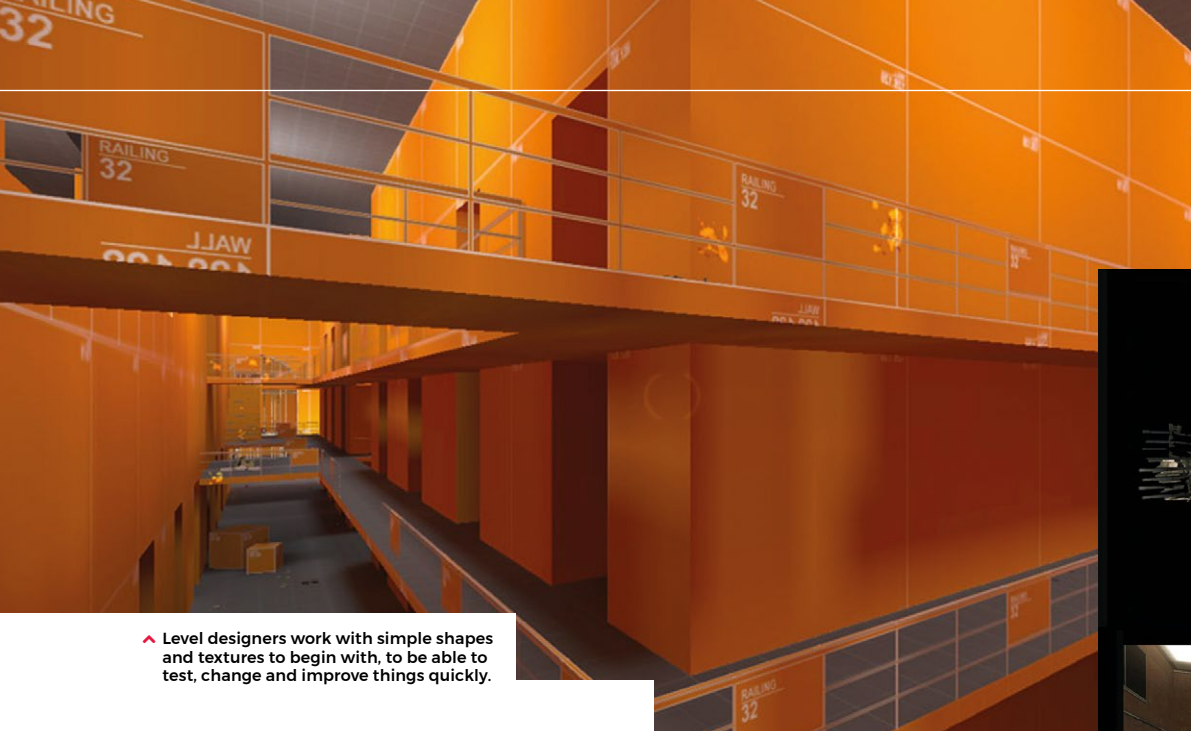
With 3D level design, this blockout phase is often done literally with big, simple boxes, covered in flat-colour development textures –

✓ Levels for triple-A games can take several people months, even years, to make. Start simple and small!



✓ *Half-Life 2* levels look like this, when viewed in Hammer, the Source Engine's level editor.





▲ Level designers work with simple shapes and textures to begin with, to be able to test, change and improve things quickly.

hence the term 'greyboxing' (or 'whiteboxing', 'orangeboxing' etc). It's important to work like this so that shapes and layouts can be tested early, and also iterated on quickly, when testing reveals problems and changes that need to be made.

4. Focus on the player experience

When you're making levels, it can be tempting to get swept up in things like nice graphics, or deep, complex fiction that could fill a book, or cinematic presentation and bombastic events. All of these things can be good, of course, but the most important thing is always the player experience. It can be very easy to make something that seems superficially cool or impressive, but really, is kind of boring to play.

As level designers, we don't just care about what happens or what the players see and do – but what they're thinking, and how they feel about it. Are they really thinking about and understanding the things you want them to? Are they really having the experience you intended?

5. Playtest your map with others

It's important to remember that we're not designing levels just to play ourselves – they're for other people (and hopefully lots of them). Every player is different, and is going to experience a level in their own way. While you need to test your own work constantly while you're making it, the only real proof for whether a level is really working is when you see other people play it, and they give you honest opinions about their experience.



Everyone is too close to their own work to be able to see it from every angle. So if you want it to be good enough for lots of people to play, you have to let other people show you how it looks through their eyes, played by their hands.

6. Quality only comes from iteration

To paraphrase the novelist Ernest Hemingway: "The first draft of anything is poo." He's on the money, and this applies to level design just as much as it does to writing novels. If you really want to make something good, you need to give yourself time to make something first, and then spend even more time repeatedly improving it.

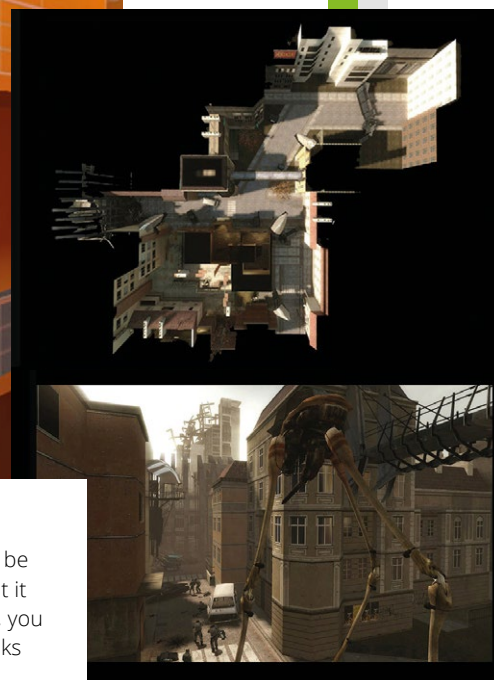
This obviously ties in with point number one, about keeping projects small. The first level you make probably won't be your life's masterpiece – but if it's small you'll be able to start iterating much earlier, and that much faster. 🗣️

"If you really want to make something good, you need to give yourself time to make something first"

Level design or environment art?

Level design and environment art sometimes get mixed up because of how much the two disciplines can overlap. Put simply, the latter is generally about making the environments look great, whereas level design is all about how it plays, and the overall player experience. Both involve a lot of thinking about presentation, composition and layout, and ideally complement each other. On the other hand, they can be in tension with each other and cause problems if they're not working together very well.

Creatively, most people lean more towards one of these disciplines than the other. And either is fine, of course. But if you consider yourself a level designer, you have to be focused on the experience people have when other people play your game, and how everything (including the environment art and the visuals) serve the goal of helping the player understand and interact with it.



▲ In 2007, Steve made his own two-part level for *Half-Life 2*, called *The Terminal*, which helped him get his first level design job in the industry.



AUTHOR
JEFF SPOONHOWER

Jeff Spoonhower is a developer, artist, and college professor who's been working on triple-A and indie games for over 17 years.
anewthegame.com

Indie reflections: making Anew

In the first of a new series, developer Jeff Spoonhower shares the challenges and rewards of making Anew

Talk to anyone who's been making games for some time, and they'll probably tell you: making games is hard. Game development is a fusion of art, music, storytelling, technology, maths and design. Learning and mastering the skills of game-making takes years, and the daily grind is not for the faint of heart.

Since 2002, I've worked in the industry as an art director, animator, video editor and sound designer on games in the *BioShock*, *Borderlands*, *Uncharted* and *Saints Row* series. In 2014, I co-founded the indie studio Resonator with my dev partner, Steve Copeland, and since then we've been working together on a game titled *Anew: The Distant Light*. *Anew* has been the most challenging and rewarding project I've ever worked on, and I'm happy to have the opportunity to share my development journey with you here. I'll reflect on the making of *Anew*

in an effort to educate and inspire those of you that are currently making, or planning to make, an independent game of your own. There's much to share and learn, so let's get started!

USEFUL QUESTIONS

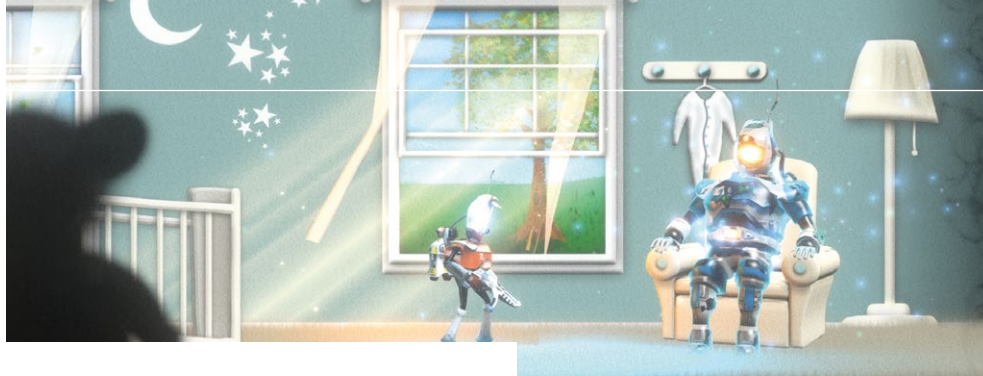
While it may sound a bit dramatic, making an indie game really is like embarking on an epic journey. You need to research your route, plan meticulously, and prepare your mind and body for the trek. Consider the following questions, and answer each of them before you dive in headfirst:

1 Where are you going?

Your answer to this should be "I want to make a game about..." or, "I want to make ... type of game." Do you want to make an FPS, a walking simulator, or a puzzle game? Steve and I love the action-exploration genre of games, oftentimes referred to as *Metroidvanias*, so we moved forward into production with that style of game for *Anew* after discussing several options. You will spend countless hours working on your project, so make sure you choose a style of game that you would enjoy playing as well. Be in love with it – the destination – even if it is a dream or a figment of your imagination at this point.

✓ The mega-mech is just one of several powerful vehicles the player can pilot in *Anew*.





2 How will you get there?

Review the map, check if anyone else has taken this route before you, and then dive into a period of research. If you're making an action-platformer game, for example, buy and play through as many similar existing games as possible. Take detailed notes on gameplay mechanics, storytelling techniques, art direction, and more. What did you enjoy? What elements do you think you could improve? Pick time-tested, effective genre conventions and incorporate some of these into your game's design. Equally as important, consider the ways in which you can innovate. In order to stand out in the frighteningly oversaturated indie games marketplace, you'll need to do at least one standout, unique thing with your game. Innovation is risky and time-consuming, so be selective and thoughtful on this front. Additionally, spend time gathering artwork, photographs, music – any materials that are relevant to your idea – and pore over them. These materials should inform and inspire your design. Don't worry too much about scheduling, or a budget, just yet. We'll discuss these topics in a later article.

For *Anew*, we decided to innovate on several fronts. Steve and I felt strongly about creating a player character that was triple-A in its responsiveness, agility and visual quality (animations, variety of movements). We spent well over a year, full-time, just working on these player-centric elements. Since most action-exploration games have the player on foot for the majority of play time, we wanted to add a variety of fun, overpowered, pilotable vehicles to *Anew*. We created a dune buggy, a giant mech, a flying saucer and a rocket ship to keep gameplay fresh and varied. In a genre that is sometimes light on narrative, we wanted to tell a mysterious, but specific story that is communicated solely through visuals, sound and music.

3 What tools will you need?

In order to make an indie game, you'll need to learn many different aspects of design and development. Unlike when I was a young man in 19XX, in the year 2019, there are a vast array of learning opportunities for the aspiring game developer. Of course, you could take a degree course to learn how to make games. Colleges are excellent learning environments: you're surrounded by experienced teachers

▲ Telling our story visually, without text or dialogue, was a risk we were excited to take.

and classmates who will support you along the way. College educations are also expensive, so if the university route is out of your budget, I'd recommend checking out online training sites such as lynda.com, udemy.com, and pluralsight.com. These are paid services that offer top-notch, self-directed learning content specific to game development. Download Unity, or Unreal, and get cracking. I know many successful game designers who were self-taught, started as QA testers at studios, or came from completely unrelated fields. Ultimately, your ability to learn the tools of the trade will hinge on a strong work ethic and self-discipline.

“Making games is hard, and gets harder as you progress”

4 Who will accompany you on your expedition?

While it's possible to make an indie game as a solo developer, you'll more than likely need to assemble a team. Size yourself up. Take an honest look at your own strengths and decide if you'll need help from someone with complementary skills. In the case of *Anew*, Steve is the technical lead (programmer) and I'm the creative lead (artist). Without each other, our game could not exist. There's very little overlap in our skills. Steve can't draw a stickman, and the last programming class I took was COBOL in 1998, so we both contribute to the game in essential ways. We intentionally kept our team small in order to challenge ourselves, create a game that's personal in tone and design, and minimise overhead and inefficiencies in communication. I suggest you do the same as you plan your team structure. Stay as lean and efficient as possible, and bring on colleagues who fill in gaps in your own strengths.

BE NOT AFRAID

This may seem like a lot of information to process, and you haven't even started designing your first level yet. Planning out the production of a game takes a significant amount of time and energy, and it can be frustrating and overwhelming. Remember, *making games is hard*, and it gets harder as you progress through development. By taking the time to thoughtfully answer the questions above, you'll set yourself up for a much more enjoyable ride, and in the end, a better finished game. ☺



▲ Indie developers are explorers. Here we see the dev team leaving their families to begin production.

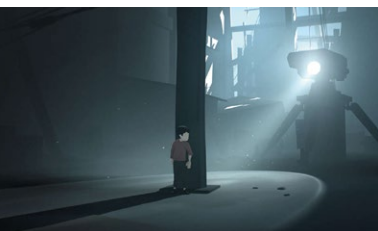
▼ You'll battle a wide variety of creatures in *Anew*.



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

▼ Acclaimed indie games including *Poi* and *Playdead's Inside* (pictured) have used the visual scripting tool PlayMaker during development.



Directory

Useful tools for Unity

As recommended by Owyn's Adventure developer Gaz Thomas (see page 24), here's a selection of handy Unity tools

> PlayMaker

Working with both indie and pro versions of Unity, PlayMaker is a visual scripting tool that lets you test out ideas or even create complete games without typing a line of code.

wfmag.cc/MubXHS

> Dialogue System

As its name implies, this tool allows you to add dialogue and quests to your games with ease.

wfmag.cc/zDOsRk

> Tiled

Whether you're making a 2D platformer or an isometric adventure game, this tool is perfect for efficiently creating and editing tiled stage layouts.

wfmag.cc/VblPuZ

> SuperTiled2Unity

A custom-built tool that means you can drop your Tiled assets directly into Unity.

wfmag.cc/kBDLEB

> Curvy

This powerful spline editing tool gives users the ability to create a range of curves, and extrude or move objects along them.

wfmag.cc/tzjQNQ

> Easy Save 2

You can efficiently add a save/load function to your game with this plugin, which includes encryption and PlayMaker support.

wfmag.cc/XXwwST

> Quick Toggle

Here's a freely downloadable tool that lets you toggle the visibility of objects directly from the scene hierarchy. Anyone who's used layers in Photoshop will appreciate how handy this is.

wfmag.cc/JhhMuv

> Pixel Perfect Camera

Avoid shimmering and other unwanted artefacts in your 2D pixel game with this script, which adjusts the camera size to ensure pixels match a device's screen resolution.

wfmag.cc/DDzDCd

Wireframe

LIFTING THE LID ON VIDEO GAMES

Download the app

Out now for smartphones & tablets



Save
45%
with an annual
subscription

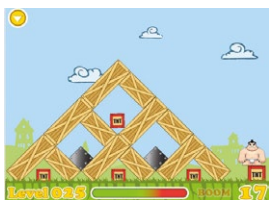
£1.99 or **£34.99**
rolling subscription subscribe for a year





Appetite for CONSTRUCTION

Codeless creation tool GameSalad could prove an appetising prospect for budding game devs. Wireframe finds out more



▲ Giles Laundry - a once prolific GameSalad user who went by the name of TShirtBooth - created *Tumble Blocks HD* for the iPad.

WRITTEN BY
DAVID CROOKES

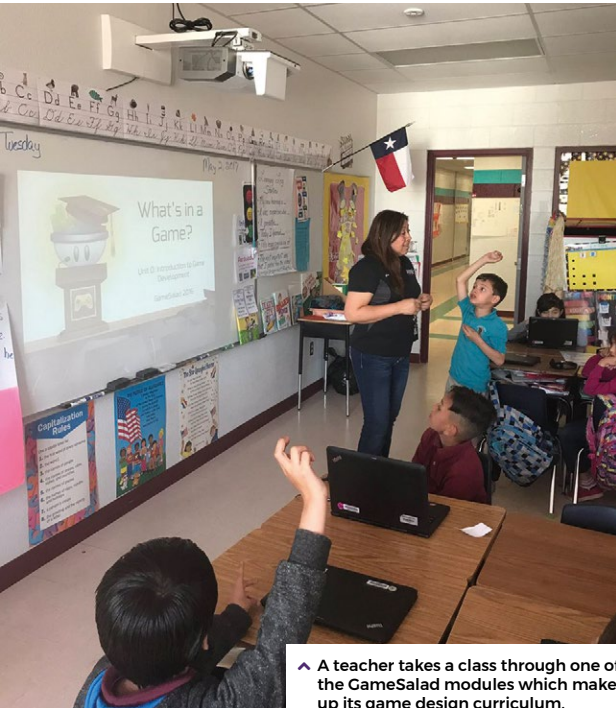
It was just a dream at the beginning: a dream that seemed to be giving way to frustration as something so simple proved far out of reach. Braydon Coyer was trying over and over to produce a game where the player had to keep a balloon up in the air. It was, he admits, a simple mechanic and yet, after six hours working with the cross-platform 2D game engine Corona, he finally lost patience and promptly gave up.

For many people, this is a familiar feeling: Corona uses the Lua scripting language and it relies on prospective developers getting their hands dirty with code. In that respect, it's far from unique - there have been many programming languages over the years, from BASIC and Java to Python and C#, and their complexities can prove to be a barrier for some. As a result, it's likely a whole bunch of killer ideas have never seen the light of day and yet, since his struggles in 2011, Coyer has gone on to create hundreds of games.

"Shortly after, I saw an intuitive tool called GameSalad, which allows you to make any 2D game mechanic that pops into your mind," Coyer says of a visual, block-based tool which launched as an open beta ten years ago. "I particularly like to use GameSalad to create physics and word games, and I've found it extremely easy to use."

Instead of requiring code, GameSalad allows creators to build games by dragging and dropping various elements and assigning behaviours to them. With a bit of thought and some well-produced graphics and sound, it's possible to create some polished and varied games, with concepts turning into working prototypes within a matter of hours - a godsend for anyone who's spent time trying to perfect a game's mechanics.

Coyer certainly found it was useful, since it enabled him to make a number of games on behalf of clients, building a flourishing business around a model of mass production. He also developed four original games: *Penguin Pounce*,



▲ A teacher takes a class through one of the GameSalad modules which make up its game design curriculum.

Penguin Pop, Frozen Finder, and Knight's K'west, and released them across major app stores.

KNOCKING THEM OUT

This was made possible because GameSalad's publishing tools allows games to be released on iOS, Android, Kindle Fire, macOS, Tizen and the HTML5 platforms. As a result – and enticed by the promise developers can retain full rights to their games and make money from them – as many as 65,000 titles have been created, with some going on to achieve great success.

To date, GameSalad has spawned 80 top-100 hits in the US App Store, with many number ones among them. Critically-praised titles have included *Roll Turtle*, *The Secret of Grisly Manor* and *ePig Dash*, the latter a big seller in Chile, and made by a magician-turned-developer. There have also been sparks of originality in games such as *Booger Boing*, alongside a plethora of zombie and retro-inspired offerings.

Even so, it's fair to say that the system has not always been without controversy. Some critics have accused GameSalad of encouraging shovelware and, when you take into account that it was originally Mac-only and created to allow non-coders to capitalise on the goldmine that was the Apple App Store, it's easy to see why. There was a low point in 2014 when a \$9.99 GameSalad template called *Red Ball*

became a best-selling game called *Red Bouncing Ball Spikes* – even though it was only ever so slightly tweaked and originally created as an instructional demo. This doesn't mean a codeless games creation system is devoid of merit, however; the important thing is that it's providing opportunities for people who might otherwise balk at developing their own game.

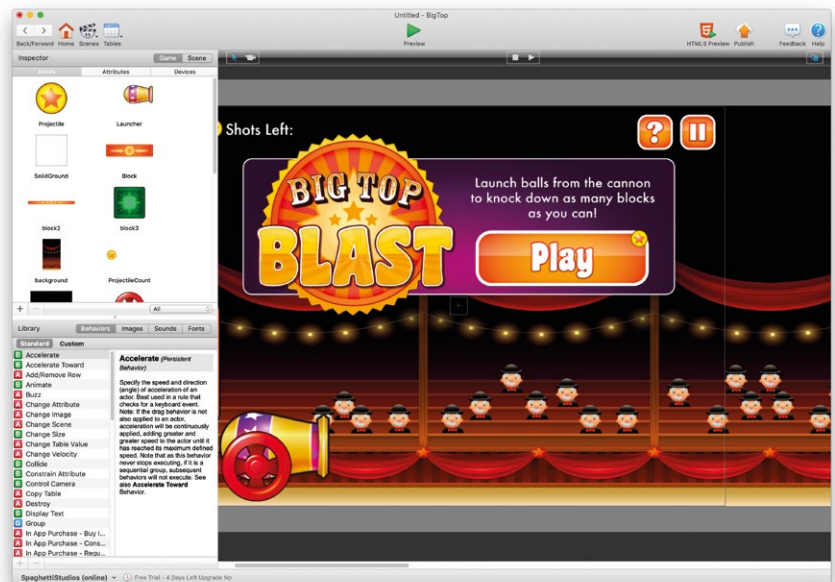
Certainly, Coyer says his own game, *Penguin Pounce*, was downloaded more than 100,000 times and that, for a time, *Knight's K'west* outsold *Angry Birds* on the Amazon App Store.

“To date, GameSalad has spawned 80 top-100 hits in the US App Store”

More importantly, the tool encouraged Coyer to study Computer Science, which ultimately led to employment as a web designer. Often, one thing leads to another.

“GameSalad introduces the concepts of computer sciences and programming in a better way than a lot of other engines out there,” Coyer says, pointing to a supportive community, countless downloadable demos, and video tutorials which are available to help users along the way. “I also think a lot of people find it more interesting and fun to create their own games instead of writing a ‘Hello World’ program.” ▶

▼ The desktop version of GameSalad is available for Windows and Mac, and there are viewers available for previewing projects live on iOS and Android.



Interface

Appetite for construction



▲ *The Secret of Grisly Manor* was one of the first titles created using GameSalad, and it proved a huge success.



ENTERING THE CLASSROOM

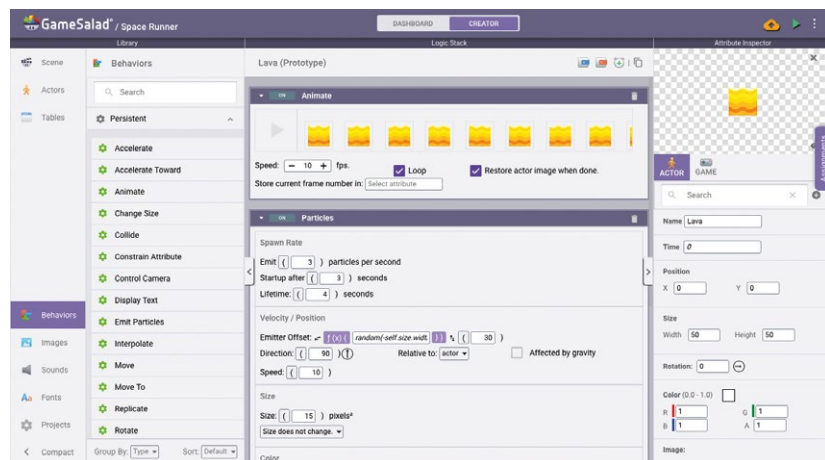
The makers of GameSalad reckon so, too, and haven't been slow to note the huge teaching potential for its tool. While the company has made mistakes in the past (it briefly stopped users self-publishing games in 2010), it has since pivoted towards education in a big way.

In the early days, for instance, a US company called iD Tech Camps ran summer courses for 10- to 15-year-olds, designed to introduce GameSalad as a learning tool. This has since been taken much further, with GameSalad now being taught in schools thanks to the development of a curriculum and a comprehensive set of classroom resources.

TESTING TIMES

GameSalad's CEO, Brent Dusing, was once a games developer, and he sees the production of games as a craft. "It's part science, too, but you don't know what you have until other people have tried it," he says. That's why he thinks GameSalad's ability to quickly compile games is so critical. "Being able to play immediately, especially on a device, is great because, as all developers know, there's a big difference between testing on a device and inside of a dev environment.

"Students can also develop a game, open up their phone, take a picture of the QR code and start playing," Dusing adds. "It's certainly possible to develop a game in less time and with less effort than in, say, Unity."



Key to this change was the 2016 appointment of current CEO, Brent Dusing, who joined following a spell as the chief of Lightside Games. Dusing couldn't have arrived at a better time: more and more schools across the world are teaching computer science, and Dusing believes his company – which has raised about \$8 million in investment capital since its launch – is well-placed to instil the necessary skills in its pupils.

"There are three major components to learning computer science: architecture, logical patterns and syntax," Dusing tells us. "We find that when we take students aged between 10 and 17 years old, which is typically our sweet spot for age ranges, that they are able to learn architecture rather quickly. They understand how programs come together and how different components work.

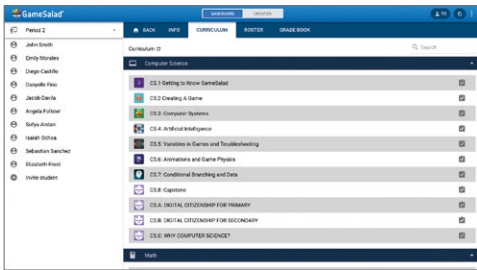
"But then things become trickier. Logical statements – things like the 'if then else' statements or booleans – are not necessarily intuitive: just like calculus, you could teach logic but people don't generally come up with it on their own. Similarly, with syntax. If you miss one word or put a punctuation mark in the wrong place, then suddenly the program crashes. We work on the premise that we need to build people up by giving them the building blocks for the principles they'll learn later down the line. In education, we call that scaffolding."

OFFERING SOLUTIONS

Such an approach is aimed at building confidence. By avoiding the need to code in the early stages of a child's introduction to games development and computer science, Dusing says they're less prone to being hamstrung by typos and syntax errors, and less likely to give up as a result.

It means the emphasis is on momentum and instant results, and it's no accident that GameSalad has a preview feature built into its interface that allows developers to instantly see how their game is progressing. In that regard, it provides an instant reward. Being able to then go ahead and put those games on to a mobile phone then proves doubly motivating.

"When a program crashes because of a mistake, you often get an emotional reaction of 'I'm not good at this' and that can cause people to give up," Dusing continues. "But we want to avoid that: you wouldn't give a twelve-year-old calculus and say, after they get an F



▲ A teacher dashboard allows educators to keep on top of the curriculum and the progress of their students.

on their calculus test, that they suck at math. They don't. It just means they were too young to take calculus and that we needed to give them algebra, geometry and algebra two and precalculus first. From my point of view, GameSalad gives students a capacity to learn architecture and logic without overburdening them with syntax too early."

Dusing says a no-code approach benefits teachers too and, indeed, the education section of the GameSalad website talks of the application being a "complete solution" – one which doesn't require a Computer Science degree to teach. Instead, educators

are able to attend a training course, while the available packages include a ready-made, project-based curriculum that has step-by-step tutorials, activities and assessments.

"Schools are not able to hire teachers who have computer science degrees because most of them will go straight to a big tech company," Dusing explains. "But what we can do is go into school districts and say we can successfully teach computer science to students without anyone needing to be hired or fired. It's about training the teachers we have and giving them the confidence and the skills. We concentrate on addressing the question of 'how do we teach this subject?'"

HANDS ON APPROACH

Such an approach doesn't come particularly cheap. Costs start at \$1,699 for a classroom of up to 30 students, making it a higher-priced inroad to the basics of programming than some of its rivals. Certainly, GameSalad can be compared to Scratch, the massively popular, easy-to-use visual coding language that is also capable of impressive results. Scratch is free, but GameSalad reckons it can be the tool to use once Scratch has been "mastered". "Scratch is

great for younger students while GameSalad is perfect for those of middle-grade age," Dusing claims.

As if to underline that, Dusing says GameSalad can touch on many concepts, from gaming structure to artificial intelligence ("children build a tank game where there's a bad guy you have to program coming after you"). This can be seen in action at Monroe Junior High School in Ohio, where GameSalad is used to teach the importance of design documents (allowing pupils to look at wireframes, one-page designs and storyboards) as well as programming, iteration and playtesting.

"During each lesson, we also focus on specific game elements like triangularity, debugging, game balance, scaling and difficulty, and scoring and reward mechanisms, to name a few," says Monroe's technology teacher and tech integration specialist, Andrea Carovillano. "The

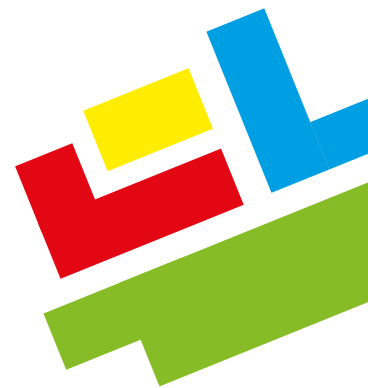
end result is a game that showcases principles of game design and creativity."

In each case, games are built one bit at

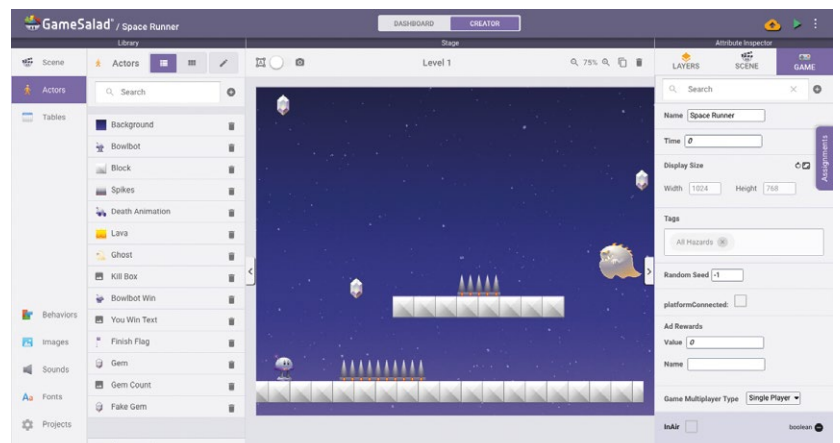
a time, and students are generally given a month to create something from scratch. The original concept often evolves during that time frame, depending on the approach to teaching. "GameSalad provides complete control over their games, giving them an endless ability to create games that feature what those students are interested in," says Carovillano. "What may have started as a simple shoot-the-aliens game like *Space Invaders* ends up being a bowling ➔



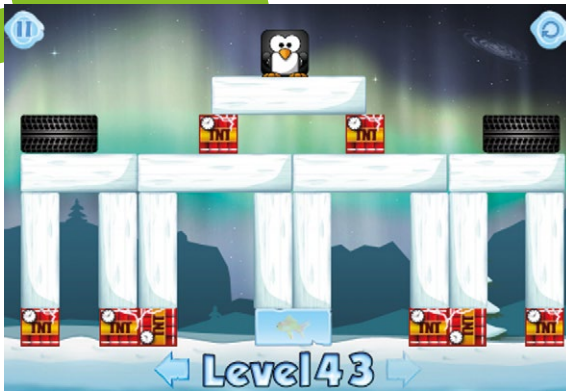
▲ The fast-paced puzzler *Stan Lee's The Traveller in Split Second Mayhem* was created for iOS – the format GameSalad was originally developed to produce games for.



▼ Graphics for the games are manually placed within the Scene Editor window. You can see the associated graphics to the left.



▼ Braydon Coyer, who produced *Penguin Pounce*, says GameSalad only has a few ad-monetisation options.



▲ These students are learning the basics of game design using the downloadable desktop version of GameSalad.

game, but the drag-and-drop format means changes can happen almost instantly.”

For the most part, Carovillano’s students have a foundational mastery of GameSalad within five lessons and, since GameSalad has moved over to a web interface, homeworking has also been made easier. This has also been felt at other schools. “Some students really get into their games and work on some of the projects from home,” says Edward Mitchell, a media arts teacher at Los Angeles Unified School District.

CREATING A FUTURE

Having already taught digital imagining, 3D art, animation and modelling, Mitchell believed GameSalad would be perfect for its ability to teach game design concepts and workflow. “Once you understand the interface, it’s very simple to access the instructional materials and video tutorials, and begin creating games,” Mitchell says. “As the teacher, you just assign the projects and have students begin working on designing the games. Ultimately, I was very excited and encouraged by how quickly students were able to acclimate to the game design

interface, and how successful they were in creating games.”

Such praise will be music to GameSalad’s ears. Today, it continues to visit school districts across the US to invite more teachers on board, while ever-growing numbers of students enjoy seeing

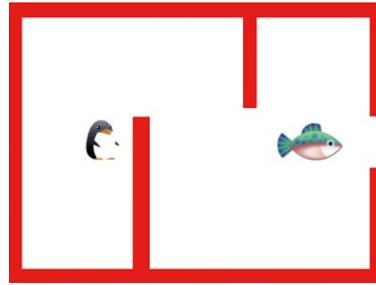
the fruits of their labour on mobile platforms. The company still hopes that individual would-be developers enrolled on its limited

■ **“I was encouraged by how quickly students were able to acclimate to the interface”** ■

free trials will upgrade to the paid-for Basic and Pro accounts (costing \$17 and \$25 per month respectively), but education is where most of GameSalad’s future revenue is likely to come from.

The eighties era of bedroom coders resulted in a thriving games industry, with the likes of Philip and Andrew Oliver, Richard and David Darling, and a host of other teenagers and young adults going on to make games that remain much-loved today.

Could GameSalad have a similar effect, and encourage children to level-up to coding proper? Whatever happens, GameSalad provides a clear and approachable means of getting into development – which, let’s face it, can only be a good thing. 🍷

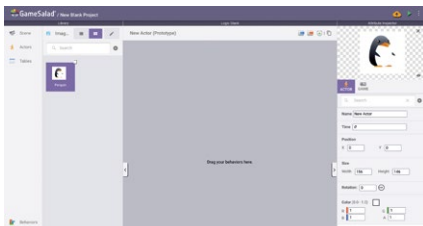


Create the basic elements of a game

GameSalad allows you to build games using a drag-and-drop interface based around attributes and behaviours. You can test the progress of your creation at any time by clicking Preview

1 Generate a lead character

First of all, launch GameSalad (visit creator.gamesalad.com) and create a New Blank Project. Once done, you'll find yourself looking at a window marked Initial Scene but, to start with, we'll create a main character. Ensure you're in the Actors tab in the left-hand menu and select '+' to generate a New Actor. Click the actor, then select Images in the bottom-left corner. Drag over an image from your hard drive, then place it in the top-right area. We're now ready to set some behaviours.



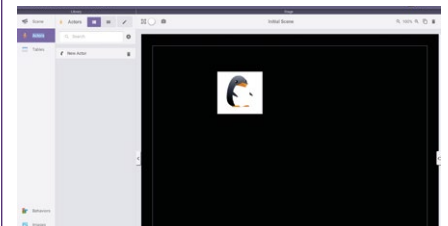
2 Set up the movement

Click Behaviours in the left-hand menu. Select the Add Rule icon from the top of the middle window and rename the box Direction. We're going to use some logic to ensure that pressing the right arrow key moves the character right. After the words 'Actor receives event' select 'keyboard key' from the drop-down menu. Then, in the text entry field, press the 'right' arrow on your keyboard. Now, from the list of behaviours, drag 'Move' into the space that says 'Drag Your Behaviours Here'. Set the direction to zero.



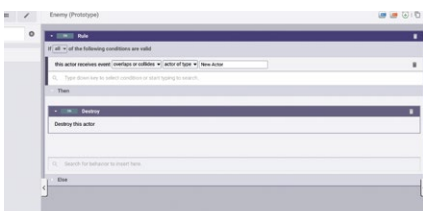
3 Place the actor

Setting the direction to zero means the character will move right. Setting it to 90 moves it up, 270 moves it down and 180 moves it left – and these should be set to the up, down and left keys, respectively. Now click Scene (top-left) again, choose Initial Scene, then select the Actors tab once more. Drag New Actor into the main gaming window. We'll now add a second actor: an enemy. Under the Actors tab, click '+', add a graphic as before, and double-click.



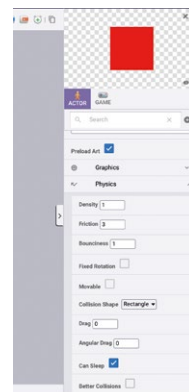
4 Create the enemy

You can change the name of actors by double-clicking New Actor. We're going to call ours Enemy. We're merely going to have the main character collide with the enemy and force the latter to disappear. To do this, we need a rule that causes an action when both come into contact, so click Add Rule and ensure 'Actor receives event' is set to 'overlaps or collides'. Enter the name of the actor it will collide with, then select Behaviours and drag over Destroy.

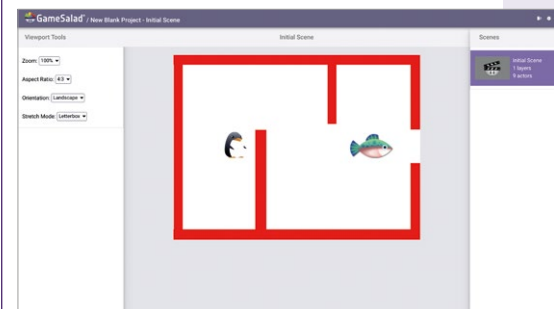


5 Add some obstacles

When you go back to Scene > Initial Scenes, place Enemy in the game window (assign it a graphic, too). You'll now want to create a challenge, so let's add some obstacles. Add another actor and double-click, perhaps selecting 'colour to red' to denote a brick wall. Under Attribute Inspector, look for Physics and untick Moveable. Go back to the scene and lay the obstacles in the game window. Now double-click the main actor and select Behaviours. Drag Collide and select the obstacle actor. When Actor 1 hits Actor 4, it will encounter an impenetrable wall.



This 'game' can be previewed by hitting the play button. It's very rudimentary, essentially allowing you to move a character around the screen, collect items, and come up against obstacles. From here, you could add a high score, experiment with jumps, create keys which unlock doors, and so much more. Never be afraid to play around!



Developer Profile

Amy Hennig

If at first you don't succeed, keep trying until a game doesn't get cancelled: the Hennig story

The early days of Amy Hennig's career weren't what one would honestly refer to as 'auspicious'. While these days the three-decade veteran of the industry is rightly hailed as one of the most influential women in all of games, she actually started out studying for the film industry, dropped into games almost by accident, and the first two projects she worked on were cancelled. So it's not like Hennig hit the ground running. Or even crawling, for that matter.

Still, a start is a start. The first cancellation came with an Atari 7800 remake of Lynx title, *Electrocop*, with Hennig providing art for the game while still attending film studies classes. Despite the setback, a switch had flicked – gaming, something Hennig had loved in her childhood but drifted away from, was a viable career path. She put together her portfolio and applied for a job at Electronic Arts, impressing enough that the film studies path came to an abrupt end – and a new era began for the young Hennig.

The Bard's Tale IV was cancelled, though, so the new era actually started out very similarly to the first one. The run of successes (and games that actually released) began in earnest as Hennig moved onto *Desert Strike*, putting together the art for a few loading screens and the like. A shunt into design came after an EA staffer left the company, leading Hennig to push



^ Even overlooked gem *Battlefield Hardline* saw the Hennig touch.

for the newly vacant role; the resulting title, *Michael Jordan: Chaos in the Windy City*, marked a frankly bizarre start for a nascent game design career.

It was the mid-nineties when things really began to crystallise for Hennig, (and no, we will never apologise for that wonderful pun): she joined Crystal Dynamics and, after some more work as an artist on *3D Baseball*, moved into the role of design manager on *Blood Omen: Legacy Of Kain*. This taste of a more narrative-focused, darker, more adult title proved intoxicating to Hennig, and the universe of Kain – and, more importantly, one of the vampire antihero's then-unnamed underlings – soon became her life for years to come.

SOULFUL LEGACY

Legacy Of Kain: Soul Reaver saw Hennig move into the role of director for the first time. She had only been at Crystal Dynamics around a year, but her gumption had obviously impressed the brass, and the finished product outperformed even the most optimistic expectations. *Soul Reaver* was a bar-raiser for gaming as a whole, bringing a (limited) open world to explore, deep puzzle and combat mechanics, genuinely intelligent design and – importantly – a

story worth paying attention to and caring about. Hennig was, of course, heavily involved in the writing of Razi's adventures through soul-sucking and time.

Soul Reaver 2 followed a couple of years later, and *Legacy Of Kain: Defiance* brought about some closure for the long-running series – and story – with

“Then came *Uncharted*, and Hennig was suddenly a household name”

Hennig still at the helm. That was it for Crystal Dynamics, though, and another new era was to begin as she made a move to a studio known for cartoony platformers and wacky action, Naughty Dog. Her first project, *Jak 3*, showed where Hennig was aiming: it darkened – matured, you could say – the tone of a series traditionally aimed at younger players. Then came *Uncharted*, and Hennig was suddenly a household name.

Recent years saw Hennig moving back to her first full-time employer, EA,



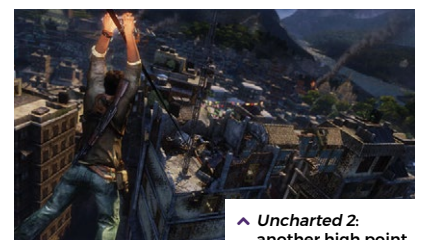
^ Hennig has spent three decades – and two stints at EA – in the industry.

with the high-profile acquisition arriving hand-in-hand with the news she would be heading up the direction for a new *Star Wars* game. Expectations firmly in place and development cracking on, the game released to huge amounts of praise and marked the first time a video game designer was awarded an Academy Award for writing, with... wait, sorry, no, there's an issue with these notes. What actually happened is the same as with Hennig's first EA game: the *Star Wars* project was cancelled, Volition, the studio developing it, was dissolved, and Hennig – eventually – took her leave from EA again, in the summer of 2018.

With a new venture firmly in her sights, Hennig's contributions to the industry are far from over. Freed from the corporate shackles of EA – again – this could prove a fruitful new era in the veteran's long career. Smaller teams and smaller budgets are in vogue, after all, and focused creativity often trumps the perceived wisdom that comes with design by committee. The best could very well be yet to come. Or maybe she'll just go back to EA for a third time, who knows? 🤔

Uncharted gets its own boxout

It was, of course, the *Uncharted* series that made Amy Hennig a household name, at least in households where people play games and pay attention to who makes them. She was a voice of consummate calm and dignified enthusiasm when hyping up the adventures of Nathan Drake, bringing a genuine air of wisdom to a series that could so easily have been presented to the baying public as a mindless killathon. In fact, Hennig's involvement in the storyline of the *Uncharted* series, making Nathan Drake a delightfully cheeky loveable rogue, made it easy to forget that the actual games did involve the wholesale slaughter of thousands. But we digress. The first three games in the series might have varied in quality, and time hasn't been so kind to the original, but there's zero doubt in anyone's mind just how influential they were; a big part of that has to come back to Hennig's direction and narrative focus in the games.

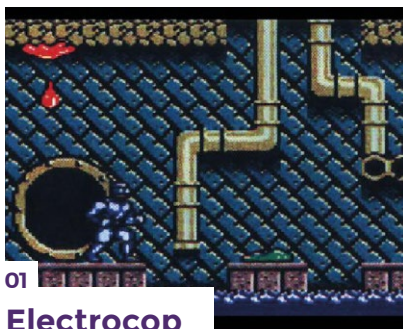


^ *Uncharted 2*: another high point.

Citizen Kain

Ten of Hennig's best projects

Working her way up through classics... and Michael Jordan?



01

ElectroCOP

Atari 7800 – Cancelled

Described by Hennig as 'a horrible *RoboCop* rip-off', this remake of an Atari Lynx action-platformer was around 95% completed before being cancelled. Why? It's not entirely clear, but the safe money would be on the fact that nobody cared about the Atari 7800, and it would have been a waste of time for everyone involved.



02

Desert Strike: Return to the Gulf

Mega Drive / SNES / Amiga / more – 1992

While Hennig's involvement in the project was on the smaller side of things, this did mark the first occasion in which a game she had worked on while under a full-time contract was released. Handily, *Desert Strike* was – and is – a stone-cold classic, albeit a politically insensitive one.



03

Michael Jordan: Chaos in the Windy City

SNES – 1994

Moving into the world of design proper, nobody would expect their first project to be taking the reigns on a frankly ridiculous platformer starring the then-most popular basketball player in the world. But there you go. *Chaos In The Windy City* was mediocre at best, but we all have to start somewhere, right?

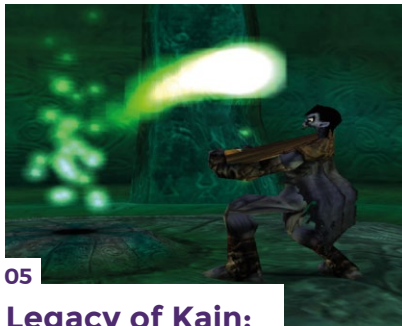


04

Blood Omen: Legacy of Kain

PlayStation / PC – 1996

The tide turned under the employ of Crystal Dynamics, and Hennig's work as design director on the *Zelda*-but-with-vampires title *Blood Omen* pushed her career in an exciting new direction. No longer languishing in ignored roles, *Kain's* antihero adventure marked Hennig's first steps into a bigger world.



05

Legacy of Kain: Soul Reaver

PlayStation / Dreamcast / PC – 1999

Inventive, original, something of a trailblazer, Razel's first adventure took the *Kain* mythos in an unexpected, exciting direction – as well as marking Hennig's directorial debut. The original *Soul Reaver* was a fine piece of work, and the sequel – while rushed out and with an underwhelming ending – was another great.



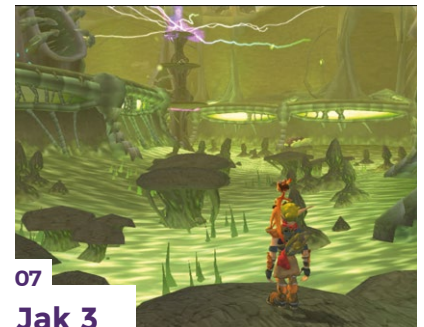
06

Legacy of Kain: Defiance

PS2 / Xbox / PC – 2003

Polishing off the series before making the jump elsewhere, Hennig's work tying up all the loose ends in the *Kain* series deserves plaudits.

Defiance doesn't quite receive the amount of praise heaped on *Soul Reaver*, with its action being a mite on the derivative side. But fans of the *Kain* series weren't let down by *Defiance's* – and the series' – dramatic finish.

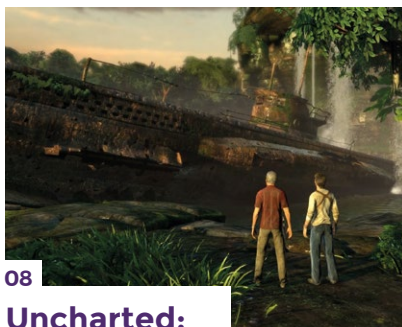


07

Jak 3

PS2 – 2004

Jumping across to Naughty Dog, Hennig co-directed the third of the series that began life as the more younger player-focused *Jak and Daxter*. A notably darker tone to *Jak 3* (building on what was established in the second game) marked the arrival of the journeywoman of gaming's writer's rooms – and the jumping-off point for a journey through *Uncharted* waters. Ahem.

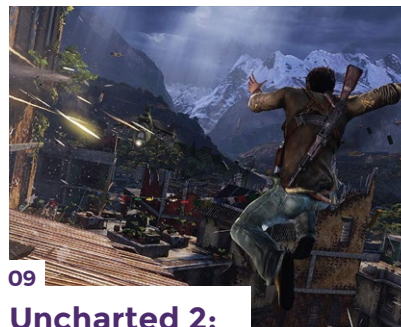


08

Uncharted: Drake's Fortune

PS3 – 2007

It's aged and doesn't quite hold up as well as others in the series, but Nathan Drake's first adventure on PS3 was a watershed moment for gaming. Mixing tight action with sympathetic and well-written characters, *Uncharted* went on to influence an entire generation of mass murdering action-adventure titles.



09

Uncharted 2: Among Thieves

PS3 – 2009

The second game, though, was where Hennig hit her peak. Drake was established, *Uncharted's* mechanics were refined, the crowd was already there waiting for more, and it was all knocked out of the park. A genuine classic of the cinematic action genre, *Uncharted 2* has still not been bettered by the series.



10

Untitled Star Wars game

PS4 / XBO / PC – Cancelled

The mix looked fabulous on paper – Amy Hennig as creative director and Visceral Games plonking all the pieces into place for a *Star Wars* tie-in with EA's money behind it. Sadly, it wasn't to be and the project was cancelled. At least this lets us point out the symmetry of Hennig's career, with both her first and last games at EA ending up on the cutting room floor.

Subscribe today



SAVE
49%

wfmag.cc/subscribe

Try 12 weeks for only £12

Subscriber benefits

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

Introductory offer

wfmag.cc/12weeks

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

Subscribe for 12 months

wfmag.cc/12months

£40 (UK) £75 (USA)

£65 (EU) £75 (RoW)

Offers and prices are subject to change at any time



Digital subscriptions from

£1.99



wfmag.cc/subscribe



Review

Travis Strikes Again: No More Heroes

Suda51's former No. 1 Assassin gets down with the indie kids

Info

GENRE
Action

FORMAT
Switch (tested)

DEVELOPER
Grasshopper
Manufacture

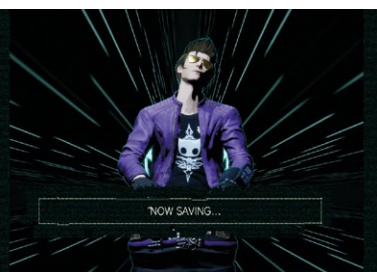
PUBLISHER
Grasshopper
Manufacture /
Nintendo [retail]

PRICE
£24.99 [eShop],
£34.99 [retail, inc.
Season Pass]

RELEASE
Out now

REVIEWED BY
Alan Wen

▼ Toilet breaks are actually the best opportunity to admire the many indie game T-shirt designs you can wear.



Over a decade ago, *No More Heroes* gave Nintendo's casual, family-friendly Wii a hardcore kick up the backside with the punk spirit of assassin anti-hero, Travis Touchdown. Vulgar, a bit basic at times, and obscenely ultra-violent, there was also nothing else quite like it. Here was, after all, a game that took the suggestiveness of the console's motion controls and made it so that recharging your weapon's power required a non-too-subtle waggle of the Wii Remote.

What was excitingly brash back then can, however, feel dated and juvenile today. But after a nine-year hiatus, Travis' return is fortunately not like seeing an old, edgy stand-up's shtick failing to land on a new generation whose tastes have moved on. This self-acknowledged spin-off is more like a tribute act to newer indie stars, all the more so given how they're thriving on the Switch eShop. Nothing says that clearer than the many T-shirts you can buy and unlock for Travis, sporting the logos and characters of many a modern indie gem, from *Hyper Light Drifter* to *Hollow Knight*, and even games yet to be released like *Disco Elysium*.

Not that you necessarily get many chances to see Travis sporting this merch. For one, save for the opening cinematic where you're introduced to Bad Man (father of Bad Girl, who Travis killed

in the first game) on his quest for revenge, in-game cutscenes are quickly forgotten along with much of that plotline. Secondly, *Travis Strikes Again* changes the camera so that the action is primarily top-down, which doesn't make it easy to make out such details.

The change of camera makes sense for the overall shift in how you control Travis, too. Instead of motion controls, *Travis Strikes Again* has a faster arcade-oriented flow akin to *Smash TV*, as you run from room to room beset by waves of enemies from all sides to cut down by mashing away your beam katana. It's not even really button-mashing, as now you can simply hold Y for Travis' light slash attack to continue as long as the beam katana has energy.

There's naturally a loss of the hi-octane slashing and less of the visceral bloodlust, since the action takes place inside a video game world and your enemies are glitchy bugs given form that at most burst into a shower of pixels. But there's no doubt this makes it more comfortable for pick-up-and-play sessions on the go, especially if a friend fancies dropping in for local co-op. None of this means Travis is getting soft in his old age, either. Combat might start off simplistic, but the action quickly escalates, as the cannon fodder give way to faster enemies, tankier types, ones that can annoyingly stunlock you, or bulbous ticking time-

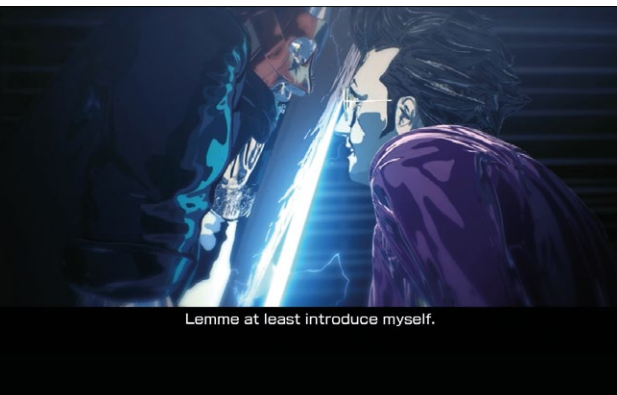


bombs with a huge blast radius. Suffice to say, it can get overwhelming.

Fortunately, you gradually acquire different cooldown-based ability chips, which you can equip up to four at any time, giving you some freedom to customise your playstyle, whether it's being able to stun nearby enemies, buff your attacks, or drop a healing pool. Your power level also increases the more you attack and stays that way provided you don't take damage, so there's an incentive for staying on the offensive while making use of the dodge-roll. Then there's your super that deals a lot of damage by charging into enemies up to three times. Seeing that Travis is a man of fewer words this time round, it's particularly satisfying hearing him swear loudly on the third charge.

It's good, scrappy fun then, but it's also the bulk of what you'll be doing over and over again. Although the conceit is that you're playing through seven different games made for phantom console the Death Drive Mark II (more like six, as one of them is 'unfinished'), they should be just considered levels. You're occasionally treated to a mini-

▼ The opening cutscene is cool if misleading. From here, story and dialogue is mostly related through text.



Lemme at least introduce myself.

▲ While drop-in co-op is supported, it comes with a caveat. Travis and Bad Man level up separately but your XP and chips are shared.

“This may feel little more than a stop-gap before the true sequel – a sentiment the game doesn't shy away from”

game to break up the action; a couple of highlights include a tribute to *Asteroids* with vector graphics, and a *Tron*-inspired take on drag-racing. I actually found the latter's gear-shifting mechanic quite fascinating, except winning subsequent races still requires you to go back to running through linear corridors then facing down a wave of enemies in an enclosed arena in order to secure the necessary upgrades.

What may prove divisive is that the other half of the game, aptly titled *Travis Strikes Back*, takes the form of an old-school monochrome visual novel – along with the litany of text elsewhere, it's an obvious tell that this spin-off is meant to be a lower-budget affair. But it does make a nice change of pace, cutting down on the open-world bloat of the mainline instalments. Paired with the action, the result is a leaner, tighter game that ultimately lends itself to its platform.

This may feel little more than a stop-gap before the true sequel, a sentiment the game doesn't shy away from with its share of nods and winks, and if that's something Pokémon can get away with, then why not Travis Touchdown – the obvious answer being that one of them makes a hell of a lot of bank, whatever it does. The good news is that the signature Suda51 style still shines through. Whether people will still be eager to see Travis headline again after dropping to the level of a tribute act is perhaps another matter. 🗨



HIGHLIGHT

If *Travis Strikes Again* is a lower-budget game, at least resources went into ensuring quality opening sequences for each of the *Death Drive Mark II*'s games, including the console's distinct psychedelic load-up. There are some choice references, from dodgy live-action FMV to PS1-era low-poly CGI cutscenes that used to blow your mind.

VERDICT

While it advertises indies that demonstrate more bite and innovation, a change of pace and perspective still makes *Travis Strikes Again* a lean and mean Suda51 joint.

75 %

Gris

Some stages of grief have a boss battle at the end

Info

GENRE
Puzzle
Platformer

FORMAT
PC (tested) / Mac
/ Switch

DEVELOPER
Nomada Studio

PUBLISHER
Devolver Digital

PRICE
£14.49

RELEASE
Out now

REVIEWED BY
Joe Parlock

HIGHLIGHT

The visuals are by far and away the main pull for *Gris*. Every single frame is stunning, and for such a short game I definitely shouldn't have as many screenshots saved as I do. This is art director Conrad Roset's first game, and what a debut it is.

W

hile some may at once write off an artistic platformer that focuses on themes and aesthetics over challenge and mechanics as 'arty-farty', or even 'pretentious', doing so with Nomada Studios' *Gris* would be massively misguided. While it does sometimes get caught up in its own hype, there are still some sparks of utter genius beneath the stunningly beautiful exterior that even the most art-resistant players will find joy in.

Gris is about grief. The opening seconds of the game see our heroine's entire world literally fall out from underneath her. Her love is gone, and her voice, the buildings, scenery, colour and even physics itself have disappeared with it. Weak, slow and distracted, it's only with the help of allies found throughout the world, and her own personal growth, that she can come to terms with her new existence. It's a complicated subject matter, and one that *Gris* manages to tackle remarkably healthily by never dwelling on the misery of grief for too long, nor by hand-waving away negative emotions to force a Disney-esque, happy 'it gets better' sort of ending.

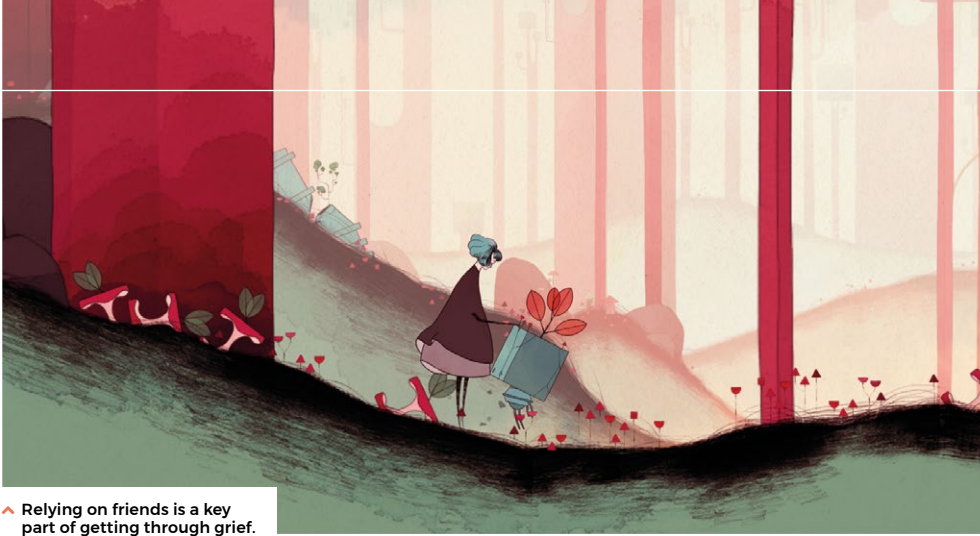
> It'd be easy to caption all of these pictures with 'phwoar'. Phwoooaar.

There is a discussion to be had about whether, in its attempt to be universally identifiable and aesthetically pleasing, the image of grief that *Gris* presents is as powerful as it could have been. The game abstracts grief to the point where it sometimes doesn't feel genuine; it's a short game with a tidy ending – anybody who's experienced grief knows it's a life-long process with multiple setbacks and stumbles.

Grief is a universal facet of life that Nomada Studios has to have experience of, but *Gris* doesn't feel like it's really their story so much as it does an audiovisual exercise that uses grief as its justification for existing. Would a more grounded – but less universal – take have served the subject matter better, such as *Papo & Yo*'s very Brazilian take on child abuse and alcoholism? It is difficult to say for sure.

Regardless, it's impossible to deny that, whatever it is *Gris* is trying to do, it does it





▲ Relying on friends is a key part of getting through grief.

with an artistic flair that's only seen in games once in a while. Its limited colour schemes and hand-drawn animations are combined to make one of the most eye-catching games in recent memory, and it's difficult to not stop every few seconds and just take in the visual splendour. It directly ties in with the protagonist's healing, too, as shapes and colours layer on top of each other as the game progresses. Starting out in sparse deserts of reds and pinks, more beauty is brought into the world over time, before ending in sublime palaces of all the primary colours.

Extra credit needs to be given for the animations, too. Crafted by hand, it looks more like a particularly esoteric Don Bluth film than games we've seen before, with everything practically flowing through the environments. The boss encounters, few and far between as they are, are a particular instance of this, with the physical manifestations of grief quickly becoming beautiful, terrifyingly violent splashes of jet black ink on the serene backdrops. Games have done animation like this before – *Sundered*, *Hollow Knight* and *Battle Chef Brigade* being particularly good recent examples – but very few do it as effortlessly as *Gris*.

At times, though, that luxurious art style can feel like it's a bit too much. On larger displays, the sparse colours, detailed environments and minimalistic character design all come together to make something that is frustratingly easy to lose track of yourself in. Hunting for the small and difficult-to-see details that suggest where you go next can turn certain areas of the game into frustrating bouts of aimlessly running around, trying to interact with something – anything – for a way forward. Annoyingly, these are often moments that should have been dominated by taking in the artistic prowess.

Underneath all the metaphor and fancy aesthetics lies *Gris*' biggest problem: the puzzles aren't interesting. Much of the game is spent running in a single direction or using the few abilities you gain along the way in the same

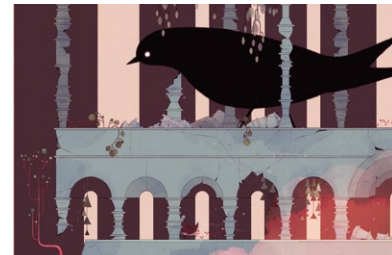


▶ If you can spot the protagonist in this shot, you have hawk vision or something.

couple of ways. Ideas that go beyond the most basic of platforming are then repeated ad nauseum, and no amount of lovely animations or cleverly assembled colour palettes can distract from that. It's a shame, because *Gris* could've omitted any semblance of traditional challenge and felt like a more confident and cohesive piece because of it; it's as though Nomada Studios felt it needed some sort of 'gaminess' in there somewhere simply because it's the done thing with indie platformers, and not because their presence adds to the game in a meaningful way.

Sometimes *Gris* gets its more gamified elements tangled up in its aesthetics and poetics, giving the entire thing the slight tang of an identity crisis. Does it want to be a piece of digital performance art (which in and of itself is a totally viable genre these days), or does it want to stand alongside the likes of *Thomas Was Alone* and *Braid* as a puzzle platformer with a pleasing aesthetic? Either way, clear a section of humdrum platforming and stumble into an abstract palace teeming with lights and sounds, and almost all of *Gris*' sins are immediately forgiven.

Whether it's connecting with a story about healthily moving forwards from grief, or simply taking in the vivid sights and sounds that make the game into such an artistic triumph, one thing is certain – *Gris* is an experience very much worth having. 🌍



▲ The sad thing about print is we can't show you this bird's excellent animation.

“One of the most eye-catching games in recent memory”

VERDICT

Mechanically lacking but artistically stellar, *Gris* feels stuck between the pillars of performance art and puzzle platformer, and both elements suffer as a consequence.

60%

HIGHLIGHT

The way the sandbox maps open up really incentivises getting right into the thick of it. A hidden bog or a surprise river can make you entirely rethink your approach, but finally struggling your way to a watchtower and revealing a new chunk of map feels like a big achievement.

Review

▲ By *Spintires* standards, this is a small puddle.

Spintires: MudRunner American Wilds Edition

Slow and steady most definitely wins the race

Info

GENRE
Driving /
Simulation

FORMAT
Switch (tested)
/ PC (tested) /
PS4 / XBO

DEVELOPER
Saber
Interactive

PUBLISHER
Focus Home
Interactive

PRICE
£34.99

RELEASE
Out now

REVIEWED BY
Joe Parlock

VERDICT

A simulator that makes the smallest trek feel like the biggest adventure. Anyone wanting a different flavour of driving game should look no further than this.

79%

M

oving five metres in 15 minutes shouldn't sound like a good time in any driving game – the genre's usually about zipping along in expensive supercars. And yet

Spintires: MudRunner manages to make the slow and arduous enjoyable in a way no other driving game has ever managed to.

Spintires: MudRunner launched on PC and consoles in 2017, but it was more recently re-released (and appeared on Switch for the first time) with a new expansion, *American Wilds*. It includes new maps, challenges and vehicles, including cranes to make picking your lorries out of a bog a little bit easier, but otherwise it's mostly more of the same as the base game at its core. This is by no means a bad thing when that core is so good.

The goal in *Spintires* is simple: deliver logs across an open map. However, between you and the logs are miles of mud, rivers and untamed forests, making navigation a challenge. The environment is the only enemy, and overcoming it means selecting the correct gear, wheels, a trusty winch and a lot of perseverance.

It's the relationship fostered with the environment that makes *MudRunner* such a compelling driving sim. It's one part observing and planning routes, and three parts Bear Grylls. You'll have to be a bit clever when the

plan inevitably falls apart and you find yourself submerged in a river, with your cargo violently being swept away. Small excursions feel like massive accomplishments in a way other sims often don't.

While shifting logs around is ostensibly the goal, the true appeal of *Spintires* is in the exploration. Much of the map is hidden in darkness until you go there, and so the real aim is facing unexpected obstacles and learning the environment to fill in the uncharted areas. For

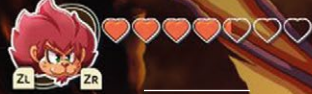
a game about lorries stuck in mud, there is a real sense of adventure here.

That isn't to say it's inaccessible, though. The challenge mode serves as a tutorial, priming you on the intricacies of fuelling, repairing and manoeuvring.

Instead of holding your hand, though, these challenges often push you to try out things you've not encountered before, such as crossing a river or keeping track of your vehicle's damage. It's an excellent system that primes you for the main event, the sandbox mode.

Spintires MudRunner is as niche as you can get. It doesn't have sexy, sleek vehicles, and it isn't about going fast. The camera's also a bit finicky and it isn't without the odd bug. *MudRunner* is, however, an intensely satisfying orienteering sim that just so happens to take place in the cab of a rapidly sinking truck. 🐾

“It's one part observing and planning routes, and three parts Bear Grylls”



805

Rated Review

HIGHLIGHT

The music is delightful, bringing back memories of the original *Wonder Boy* series without straight-up remaking those older tunes. Special mention goes to the theme, a corny but heartfelt song about adventure and saving the world, which is possibly the catchiest video game song in recent memory.

^ Bosses often test your forms' abilities more than your reflexes and platforming skills.

Review

Monster Boy and the Cursed Kingdom

A platformer that puts the wonder back in Wonder Boy

Some spiritual successors are more spiritual than others. Sometimes you get games that go for the same vibe as their classic predecessor, but otherwise have no involvement from their original creators (*Freedom Planet* and *Sonic*, for example). Then you get games like *Monster Boy and the Cursed Kingdom*, a 'spiritual successor' to *Wonder Boy 3: The Dragon's Trap* that has the involvement of original developer, Ryuichi Nishizawa. *Monster Boy's* brilliance makes it blindingly obvious that the name change is down to legal issues, and not because it doesn't measure up to *Wonder Boy's* glowing legacy.

Much like *The Dragon's Trap*, *Monster Boy's* conceit is the ability to transform between five creatures, such as a pig who can use his weight to his advantage, or a frog that can use his tongue to grab and carry pieces of the environment.

To save the kingdom (which has, shockingly, been cursed), using the forms' abilities to traverse a huge, open-ended 2D world, find better loot, and solve puzzles is essential.

And what an incredible world it is, too. It eschews the (also stunning) watercolour style of 2017's *Dragon's Trap* remake, and instead embraces a clean and vibrant Saturday morning cartoon aesthetic. Whether it depicts the heart of a volcano or takes players high above the clouds, every screen is gorgeous and lovingly animated, making *Monster Boy* easily one of the

most pleasant-looking games in the Metroidvania genre since *Hollow Knight*.

There's a decent balance between progression and challenge at first, with the game handing out new abilities and items to constantly push you forward just a little more. It encourages revisiting areas you've already blasted through, as each new skill or form can change your relationship to the environment; that wall you ignored before can now be barged through with your new lion form, or an undiscovered branch of the map can be uncovered with the dragon's flying abilities.

But then, about two-thirds in, things become a lot slower. The challenge suddenly becomes more focused on avoiding instant-kill sections, precise jumps and a ludicrous number of enemies, rather than playing with the forms and tools picked up. A specific example of this is the fire section, which lasts what feels like decades and consists of the same obstacles ad nauseam. It never devolves into making you want to quit, but it can push even the most ardent *Wonder Boy* fan's patience to the limit.

Mainly, though, *Monster Boy and the Cursed Kingdom* is essential – not just for *Wonder Boy* fans, but for anyone who enjoys Metroidvanias or platformers. Colourful and creative, even the iffy pace past the halfway point shouldn't be enough to put you off. Also, if you're not singing along with the theme tune by the end, then you officially have no soul. 🐉

"One of the most pleasant-looking games in the genre since *Hollow Knight*"

Info

- GENRE**
Platformer
- FORMAT**
Switch (tested) / PS4 / XBO / PC
- DEVELOPER**
Game Atelier, FDG Entertainment
- PUBLISHER**
FDG Entertainment
- PRICE**
£34.99
- RELEASE**
Out now

REVIEWED BY
Joe Parlock

VERDICT

The pacing sometimes lets down an otherwise clever, expansive, and vivid renewal of a classic and beloved series.

84%

Rated
Review

244 Damage!
2 Hits

Azure Wolf Strike



▶ Your party quickly exceeds the maximum four used in battle. Fortunately, everyone gains XP regardless, so roster management is purely based on preference.

✓ This definitive edition includes bonus outfits that are refreshingly free of the usual titillating fanservice.

HIGHLIGHT

Optional dialogue skits are a staple of the *Tales* games, great for fleshing out characters' relationships or purely for banter. Sometimes there's perhaps too much of it – *Vesperia's* talking head boxes felt cheap even in 2008 – but the chatter does help break up the flow of exploration and combat.

Review

Tales of Vesperia: Definitive Edition

The beloved JRPG series lives to tell another tale

Info

GENRE
JRPG

FORMAT
Switch (tested)
/ PS4 / XBO
/ PC

DEVELOPER
Bandai Namco

PUBLISHER
Bandai Namco

PRICE
£39.99 (Switch),
£34.99 (others)

RELEASE
Out now

REVIEWED BY
Alan Wen

VERDICT

Thanks to its memorable characters and timely themes, this is a tale worth retelling.

77%

Remember when Microsoft invested in JRPGs? One of its earliest achievements was striking a deal for a one-time exclusive entry from the long-running *Tales* series, which is now a fan favourite. Even then, old habits die hard, with Namco Tales Studio sneakily producing a superior port with more content for PS3 that never saw release outside of Japan. A decade later, this director's cut is finally available worldwide on all platforms.

What will make certain JRPG fans happy from the get-go is the choice to play in either English or Japanese. That's not to discredit the localisation and voice work which *Vesperia* has been well-regarded for – protagonist Yuri Lowell is after all voiced by Troy 'The Last of Us' Baker – but whichever you opt for, there's substantially more voice work here than in the original 360 version.

Your merry band of misfits also has a newcomer in the form of precocious, pint-sized pirate lass, Patty Fleur, while Yuri's childhood friend and chivalric knight Flynn becomes a permanent party member, at least when you're not crossing swords with him. They're worthwhile additions to the roster without feeling tacked on for the sake of providing more content; Patty's different stances giving her an unpredictable edge to the signature real-time battles.

It's easy to dismiss *Vesperia's* combat as mindless button-mashing, but there's skill and

strategy required, especially if you want to earn as high a grade as possible. You're also steadily introduced to different abilities to help with this, from Over-Limits that briefly let you string combos non-stop to insta-kill Fatal Strikes. It's not quite as advanced as a fighting game, though you can juggle enemies, but there is something akin to *Super Smash Bros.* in how your Artes are assigned to similar stick and button inputs. What's not aged as well is its awkward default 2D movement in a 3D environment, which naturally gets more confusing should you take advantage of the drop-in local multiplayer.

Outside of combat, *Vesperia's* story unfolds like many a JRPG, the beats and twists laid out as predictably as how you'll traverse Terca Lumireis' overworld first by foot, then by sea, then air. Familiarity, however, needn't breed contempt. Even today, Yuri is a refreshing leading man who'd rather downplay his role compared to white knight, Flynn, or wide-eyed upstart, Karol. The rest of the cast are an equally likeable bunch, even as their personalities and ideologies clash in dramatic and comical situations.

The story's thinly-veiled environmentalist allegory may not be a first in a JRPG, but this, combined with the theme of putting differences aside for a united cause, makes *Vesperia's* anime fantasy fable more resonant for our times than the tone-deaf portrayal of reality in the latest triple-A blockbusters. 🗨

HIGHLIGHT

The fact your efforts in the game have a demonstrable impact on the world itself is a positive that should be repeated in other life sims. You personally build the town's short-haul public transport system, for example, and having a direct hand in something like this does feel genuinely rewarding.



Oaks

Papa Bear told me not to drink from the river while the green goo's floating on there. He didn't need to tell me, the smell's terrible!

Review

Space

^ This is Oaks. His adopted father is a bear, meaning his wearing of a bearskin outfit is morbid at the very least.

My Time At Portia

Also known as Yard Work Simulator: 2019 Edition

W

hat it says about us as humans that plenty of us are happy to put countless hours into games which quite literally simulate rather

mundane aspects of real life, I do not know. That's another analysis for another time. What I do know is *My Time At Portia* is another one of those games that can very easily take over your life by offering a (fairly) streamlined, idealised version of the virtual life we all secretly want. Even if the farming isn't as in-depth as initially hoped.

To get it out of the way, let's tick off that list of influences – or at least an abridged version. *Portia* brings to mind *Stardew Valley*, *Harvest Moon*, *Animal Crossing* and *Minecraft*; even a hint of *Dragon Quest* in its RPG-lite elements and enemy design. High praise? Well, no. It riffs on this fine array of series, but *My Time At Portia* doesn't quite reach the lofty heights of its stablemates.

That's not to say it isn't fun, and those with a certain style of brain will certainly find a lot to love here. There's a distinct pang of joy when you manage to scrounge together all the elements required to build your first motorcab, and you're never shy of things to do – so long as you're willing to look around and search a bit. Safe to say, if farming, mining and building aren't your idea of fun, and if the endless busywork of *Animal Crossing* brought you out in boredom-hives, *Portia* won't change your mind.

Portia certainly offers yet another option for a virtual life to lead, and plenty of people will get their kicks from its frankly ludicrous amounts of Things To Do throughout. But it's difficult to really

heap praise on what is an obviously unfinished game being pushed out of Early Access. At the time of writing, bugs are still popping up all over the place, and the game's introductory screens point out work is still ongoing. So why, if it's very much a work-in-progress, has it seen a full release? The mind boggles.

As such, it's one of those 'bear that in mind' kinds of verdicts here – *My Time At Portia* has the potential to be very good, and plenty of folks are already enjoying the clunky fun it's been showering on them since not too long after its successful Kickstarter. But right now, what we have in front of us is an exercise in busywork that itself still needs a fair bit of work – optimisation and tweaks, bugfixes and additions. They'll surely come, and Pathea – Chinese devs behind the game – will continue to refine and improve *Portia* over the coming weeks and months (and years). Right now though, it's another fun-but-lacking option for your idealised virtual life. 🐻

^ The twee factor is definitely up there, and – unless you dislike the art style – *Portia* can be a pretty game.



Info

- GENRE**
Life Sim
- FORMAT**
PC (tested) / PS4 / Switch / XBO / Mac
- DEVELOPER**
Pathea Games
- PUBLISHER**
Team17 Digital Ltd
- PRICE**
£24.99
- RELEASE**
Out now

REVIEWED BY
Ian Dransfield

VERDICT

A million ways to pass the time; a million ways to waste your time.

68%



▼ Taking a bird's-eye view will help you isolate your next destination, so keep your eyes peeled for golden glints in the sun...

HIGHLIGHT

Few games compel us to slow things down and intentionally stall just to take in more of the soundtrack, but that's exactly what *Vane* manages to achieve with its moody, atmospheric score. It's the perfect accompaniment to this otherworldly adventure, especially as you creep toward *Vane's* baffling conclusion.



Review

▲ To progress, you must flip between being both a boy and a bird to solve puzzles, release captured birds, and move onwards.

Vane

You're so Vane (you probably think this game is about you)

Info

GENRE
Adventure
FORMAT
PS4 (tested)
DEVELOPER
Friend & Foe
PUBLISHER
Friend & Foe
PRICE
£17.99
RELEASE
Out now

REVIEWED BY
Vikki Blake

VERDICT

Glamorous but glitchy, *Vane's* all style with little substance.

50%

You can see what *Vane* wants to be. It's stuffed into every second of this moody, atmospheric game, sewn throughout its magnificent world, glittering in your iridescent feathers as you sweep across the sun-bleached sky.

Then you attempt to land, and your talons miss the perch. You try again. This time you overshoot the thing entirely, prompting an unforced error that takes you miles out of your way in order to course-correct. You begin to fear that the thirty seconds of silent, effortless gliding you just experienced is the only effortless part of this game you'll encounter. Unfortunately, that's not far from the truth. Eventually, you'll realise that everything – from the clunky, uncooperative camera to the clipping assets to the progress-breaking bugs – seems at painful, astonishing odds to the carefully crafted animation and stunning soundscapes.

Sadly, I can't even tell you that the story is worth the pain of grappling with the control scheme, either, because the truth is, I don't know. Much like *Journey* and others of its ilk, it's a tale told through mood and emotion rather than conventional storytelling, but there's not enough of anything tangible to decipher it. You're left with characters you cannot identify with, armed with motivations you cannot begin to understand, and not even *Vane's* gorgeous environments offer up enough clues to help you unravel its plot. That's not to say it impedes your progress – to its credit,

developer Friend & Foe does an outstanding job of teaching us what to do, and how to do it, with no explicit instructions – but somehow it manages to impede its own story.

Which makes it even more surprising when, halfway through, *Vane* turns your experience entirely on its head. Where you'd once been soaring through the skies in search of abandoned weather socks, you're now scrambling across a subterranean underworld. Where you were once a bird you're now – somewhat inexplicably – a young boy, and having to undertake the rest of your journey (mostly) on foot. The rest of the story sees you working in silent collaboration with your new pals to roll a giant, glowing orb through the cavernous environments. The skills and experience you'd built up as a bird essentially become useless. If the change is deliberate, it's a bold choice... and not necessarily an understandable one, either.

But any joy of experimentation and exploration is hampered entirely by those ugly controls and that obnoxious camera at every turn, with both the bird and the boy as difficult to direct as they are to understand. All and any emotion comes from the melancholic score and rarely the action on screen, making it almost impossible to connect in any meaningful way to our peculiar protagonists. The care so painstakingly applied to painting and orchestrating *Vane's* world vanishes when it comes to playing in it... and it's just not pretty enough to cover up the game's dazzling lack of direction and polish. 🗣️



No Man's Trek: Andromeda

Ignore the toys and there's fun to be had with Starlink

Toys-to-life isn't anything I have interest in – there's far too much clutter in my home to begin with. Plus, seeing as an addictive personality is something I'm very much afflicted with, I feel it better to avoid the temptation of purchasing something that gently pushes you in the direction of then purchasing many more things. *Starlink: Battle for Atlas* might have been hyped up on a wave of toys attached to your Switch – or other console – but it does have a digital edition, which is 100% playable without the pointless chunks of space-filling plastic.

And you know what? It's alright. It's very clearly a game made for younger players – though apparently with that aim widened as development progressed – so it's bright and gaudy, light and lacking in any real challenge beyond the odd timing-or-element-based bit of combat. And yet, it's something I'm finding taking up far more time than I ever expected it to. *Starlink* is, not to put too fine a point on it, fun.

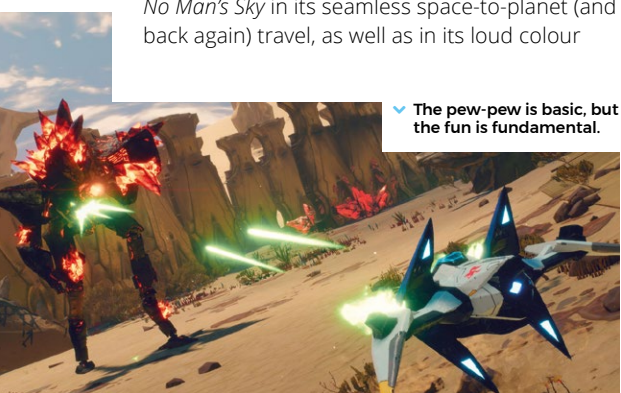
A big reason for this has to be because of its myriad influences – there's *Star Trek* in its motley crew of human and aliens working together exploring uncharted regions of the galaxy, there's *No Man's Sky* in its seamless space-to-planet (and back again) travel, as well as in its loud colour

palette. There's *Mass Effect* (and yes, even *ME: Andromeda*) in it with its story pushing you from planet to planet, taking in the sights as you build up to fighting the big bad. Oh, and there's *Star Fox*, if you're playing on Switch, which – even though you're part of a crew including aliens and can play as an alien yourself – is pretty strange to see. Maybe it's because I can't get the image of those *Star Fox* muppets from *E3 2015* out of my head.

"It's something I'm finding taking up far more time than I ever expected it to"

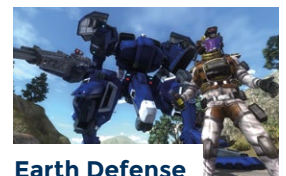
Basically *Starlink* is a bunch of fantastic sci-fi influences with a tremendously pointless toys-to-life aspect bolted on to an unspectacular, but enjoyable third-person shooter. Not to put a value judgement on something – what's a pittance to some is next week's grocery bill to another, after all – but it's definitely a game that's worth the £20 or less you can usually find it for these days. Though somewhat ironically that is the physical version, which does come with the aforementioned pointless toys. What a rollercoaster this is turning out to be.

But for those of us who found the joy in *No Man's Sky* (clue: you just had to look past the almost entirely manufactured internet hate brigade); for those of us who thought *Mass Effect: Andromeda* was alright (clue: you just had to look past the almost entirely manufactured internet hate brigade); and for those of us who still dream of muppets introducing keynote speeches in what you did assume the morning after was a fever dream, *Starlink* is a solid way to spend a bit of time. Of course, you whack your Switch into the dock and boot up split-screen co-op and you're into a whole new world of planet-exploring, alien-blasting fun. But that's a tale for another time. 🐾



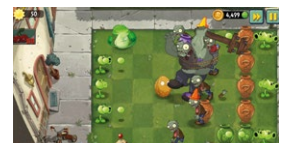
✓ The pew-pew is basic, but the fun is fundamental.

Wireframe Recommends



Earth Defense Force 5 PS4

Wilfully stupid or genuinely stupid? I don't know, and I don't care: *EDF 5* is brilliant. Sure, play a few levels in a row and endlessly mowing down giant ants is boring, but in short bursts it's phenomenal.



Plants vs. Zombies 2 IOS / ANDROID

No, really. You have to work hard to ignore the endless adverts, but *Plants vs. Zombies 2* – once stripped of its many freemium 'features' – is a worthy sequel to the original home-defence classic.



They Are Billions PC

I'm pining for mid-nineties style RTSes, but *They Are Billions* still manages to scratch an itch I never knew I had. Turns out frantically scrambling to fight off zombie hordes is... brilliant.



Rolling Thunder

For a few blissful seconds, Rolling Thunder's machine gun made you feel invincible

NAMCO / 1986 / ARCADE/VARIOUS

In the noisy, boisterous world of the eighties arcade, it was incumbent on game designers to make their experiences fun, but not too long-lasting. Fleet-fingered experts like Billy Mitchell may have been able to play *Pac-Man* or *Donkey Kong* for hours, but arcade owners were banking on most of their customers surviving for only a few minutes – all the better to keep those precious coins flowing into their machines. What was important, though, was to give players an occasional break from the bombardment of bullets, barrels, and alien invaders; a kind of *Rambo* moment where the tables turn and the player felt all-powerful. It's why we got those power pellets in *Pac-Man*, and the barrel-smashing mallet in *Donkey Kong*.

Namco's *Rolling Thunder*, an action-platformer with an espionage theme, came up with a bullet-heavy twist on this design philosophy. As the lanky, athletic Agent Albatross, the player strode through an enemy base, gunning down bad guys with his handy revolver and leaping across platforms. It was a tough game, partly because of the speed and relative intelligence of the bad guys, who could fire back at Albatross, lob grenades, or kill him with a single touch if he strayed too close, and partly because of the hero's rate of fire. The agent's revolver could only let off a couple of shots at a time, so if the player missed an enemy,

they'd have to wait an agonising fraction of a second before they could fire another.

All of this, coupled with a dwindling supply of ammunition and level designs that had enemies popping unpredictably out of doors in the background, made *Rolling Thunder* a relentless challenge right from its opening screen. On occasion, though, Namco's designers switched things up: behind certain doors, the player could find a machine gun. Unlike the blap-blap of bullets from the revolver, the machine gun let off an almost continuous volley of hot lead that left enemies melting helplessly in its wake. It was a thrilling moment, made all the more delicious by its brevity: that magazine of 100 bullets would evaporate in seconds unless carefully rationed. And once the bullets were gone, it was back to Albatross' standard-issue revolver.

Rolling Thunder's machine gun gave the action a thrilling sense of light and shade. For much of his mission, Albatross was assailed from all sides; he was a walking magnet for the hooded goons and other weird denizens of the main villain's lair. When *Rolling Thunder's* Killer Feature kicked in, by contrast, Albatross seemed almost invincible. In its own way, the machine gun was as ingenious an invention as *Pac-Man's* power pellet; certainly, its fleeting sense of power was something that even later, technically superior action-platformers couldn't always replicate. 🗨

**“A thrilling moment,
made all the more
delicious by its brevity”**

Next Issue

ON SALE 14 FEB

Don't miss out!
Subscribe
PG54

SOME DAY YOU'LL RETURN

Counting down to a terrifying
psychological horror

Also

- ▶ Chris Avellone on video game narratives
- ▶ The Sega Dreamcast's enduring homebrew scene
- ▶ Making an FPS in Unity: the final chapter
- ▶ Plasticine, not pixels: games that use real-world materials

Editorial

Editor

Ryan Lambie

Email ryan.lambie@raspberrypi.org

Features Editor

Ian Dransfield

Email ian.dransfield@raspberrypi.org

Sub Editors

David Higgs and Vel Ilic

Design

criticalmedia.co.uk

Head of Design

Lee Allen

Designer

Harriet Knight

Contributors

Adam Barnes, Vikki Blake, David Crookes, Mel Croucher, Konstantinos Dimopoulos, Andrew Gillett, Steve Lee, Keza MacDonald, Steve McNeil, Joe Parlock, Chris Payne, Dan Pope, Aaron Potter, Jeff Spoonhower, Alan Wen

Publishing

Publishing Director

Russell Barnes

Email russell@raspberrypi.org

Tel +44 (0)7904 766523

Director of Communications

Liz Upton

CEO

Eben Upton

Distribution

Seymour Distribution Ltd

2 East Poultry Ave, London EC1A 9PT

Tel +44 (0)207 429 4000

Subscriptions

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

To subscribe

wfmag.cc/subscribe

To get help:

rpiresshelp@raspberrypi.org



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).



wfmag.cc