

NEW £3
ALL FORMATS

Wireframe

LIFTING THE LID ON VIDEO GAMES

Getting started with Unity

How to make a
first-person shooter

Doom to Dota 2

When modders
turn professional

Thrunt XL

An indie game made
without a line of code

SCARE TACTICS

X-COM'S JULIAN GOLLOP ON HIS
NEW GAME, PHOENIX POINT

Issue 3 £3 wfmag.cc



Subscribe today

12 weeks for £12*



Visit: wfmag.cc/12weeks to order

* UK Price. 6 issue introductory offer

Expanding our definition of criticism

Right now, there's someone talking their way through a game on Twitch. Millions of people, all day every day, are having conversations about video games on Twitter, on Discord and forums, at work and school, with their partners, friends, and absolute strangers. Shouting at one another across vast distances of geography cleared in the zip of electrons.

Is Samus Aran trans? Who do we believe in *Dark Souls* – Kaathe or Frampt? Does Dead Eye trivialise gunplay in *Red Dead Redemption 2*? Should there be black women warriors in a *Battlefield* game? At what point does a game justify its human cost?

We need to challenge the very definition of what games criticism is in the first place. Video games are, in a sense, a hybrid medium – so, of course they'd develop hybrid forms of criticism. And we need to acknowledge this.

From a piece with five reads on Medium to hyper-subscribed YouTubers and major game websites reviewing the latest triple-A blockbuster, games criticism is happening everywhere at all times. Here's a revelation for you – your parents have done it (“What the hell are you playing?” or “Why is Mario being mean to those mushrooms?” would be considered critiques, as reader-response theory hinges on audience interpretation).



DIA LACINA

Dia Lacina is a queer indigenous trans woman writer, photographer, and founding editor of CapsuleCrit.com, a monthly journal dedicated to microgenre work about games. She tweets too much at [@dialacina](https://twitter.com/dialacina)

Games are everywhere now, an almost inescapable part of the media landscape. And that means more people are responding to them, creating meaning, placing them into context for themselves and others. They'll hit publish, share, send tweet. This is criticism. Real criticism. We're beyond the Reviewer/Critic binary.

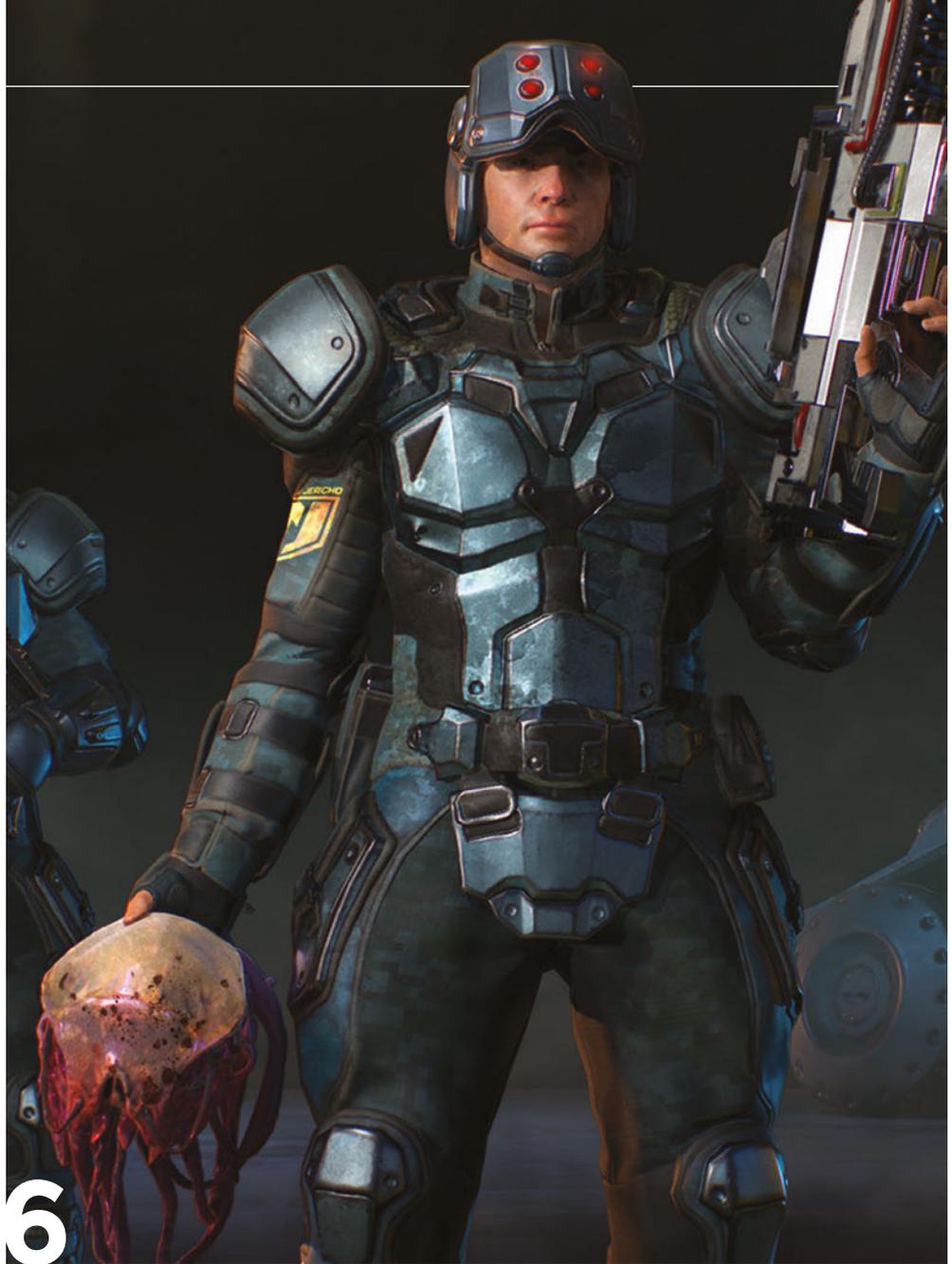
And maybe that's why it's so hard to see it. Because criticism isn't just formal or academic. Games criticism can operate within these frameworks, but it doesn't need to be confined to a narrow retrogress in an era marked by the pursuit of equity via the digital. We can have lyrical or narrative criticism, woven – as in articles on Kotaku, Polygon, Waypoint, or Giant Bomb – throughout a 'typical' review; even ephemeral, ekphrastic response, criticism-as-performance, curation, and chat discussion during streams.

Yes, a lot of it is bad. But so is Harold Bloom's *The Western Canon*. Bad art and criticism are part of our experience. An increasing number of voices are engaging in critical analysis. 'Real criticism' is a sentiment born out of gatekeeping and elitism. It's classism steeped in the racism and misogyny of decades past that gives us concepts like canon in the first place. It doesn't necessarily get us better criticism. And maybe we can shed that hegemonic albatross, along with the idea that something must be worthy of exaltation to be 'art', and let games and games criticism continue to evolve. 🐧

#03



Contents



6

Attract mode

Interface

06. Phoenix Point

X-COM legend Julian Gollop talks exclusively about his new tactics game

10. Dickey Dungeons 2

We catch up with Terry Cavanagh to chat about his forthcoming roguelike

12. Super Animal Royale

The fur flies in a top-down multiplayer shooter

16. Incoming

Dungeon crawling, cats, flying saucers, and devilish drinking contests

18. When modders turn pro

We meet the modders who've turned their hobby into a career

24. Interactive: Whipseey

How a platformer from El Salvador survived an attack of the clones

44. Thrunt XL

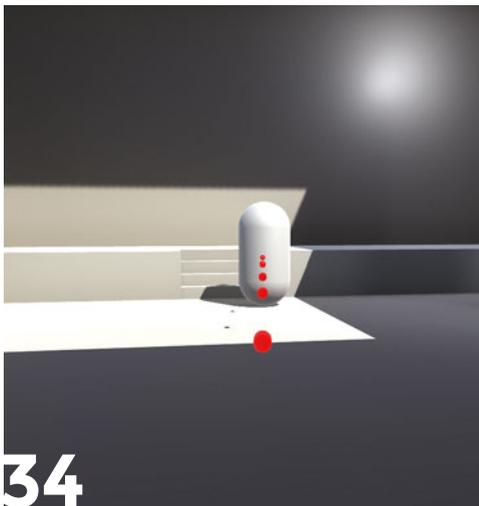
It's the fast and furious indie game made without writing a line of code

50. SNK at 40

A legend of development, publishing, and hardware profiled



44



Toolbox

- 28. Design principles**
What Yars' Revenge can teach us about design goals and planning
- 30. CityCraft**
The abstract models that will make your video game cities shine
- 32. Source Code**
How parallax scrolling gives 2D games the illusion of depth
- 34. Build a shooter in Unity**
Get to grips with Unity in our step-by-step guide to making an FPS

Rated

- 56. Hitman 2**
IO Interactive's latest sandbox murder sim is solidly executed
- 58. Grip: Combat Racing**
An homage to the Rollcage series provides a rush of adrenaline
- 62. Battlefield V**
The shooter sequel provides gritty realism and some rough edges
- 63. Black Bird**
Onion Games' 2D shooter is a darkly surreal delight

WELCOME

There's long been something magical about designing levels for an existing game. Back in Wireframe #2, level designer Steve Lee described the revelatory moment, aged eleven, when his brother first showed him a *Doom* map editor. For Lee, it was the first step on a path that would lead him to work on games like *BioShock Infinite* and *Dishonored 2*.

Even for those of us who didn't get into triple-A development, the discovery that games aren't just a predefined, unchangeable dollop of entertainment was a powerful one.

In 1988, the UK magazine Your Sinclair ran a type-in listing for something called Gauntlet Editor. It allowed users to create their own levels in the ZX Spectrum version of *Gauntlet* – an adaptation of Atari's dungeon crawler. The program itself was bare-bones, but the results were pretty jaw-dropping if you were about ten years old; here was a game you already loved, but now it was running a maze-like level that you'd designed yourself.

As we'll see on page 18, the ability to edit and modify games can lead to some pretty extraordinary fits of creativity. Without modders, we wouldn't have the likes of *Counter-Strike*, *Dota*, or *Fortnite*. Today, such games as *Far Cry 5* and *Super Mario Maker* give players the ability to make their own maps. Maybe they'll light a similar creative spark in today's generation of young gamers.

Ryan Lambie
Editor



Scare tactics

X-COM creator **Julian Gollop** talks to us about his new sci-fi tactics game, *Phoenix Point*

British game designer Julian Gollop has created some of the best strategy games ever made, from *Chaos* and *Rebelstar* in the 1980s to *X-COM* in the 1990s, to the hidden gem, *Ghost Recon: Shadow Wars* in 2011.

Now, Gollop's back with *Phoenix Point*, developer Snapshot Games' new turn-based tactics game in the sci-fi tradition of the classic *X-COM*. Like that 1994 hit, it sees Earth under threat from invaders – this time, an alien virus capable of twisting human and animal DNA into horrific new forms. Fans of *X-COM* (not to mention Firaxis's reboot, *XCOM*) will recognise the turn-based skirmishes and global-level strategising, though the use of procedural generation to create a shape-shifting enemy promises to give players a new challenge. To find out more about *Phoenix Point*, we caught up with Gollop on Skype, where we also talked about the making of its forebears, *Rebelstar* and *X-COM*, and his thoughts on what makes a believable (and entertaining) computer-controlled opponent.

▼ Although evidently a spiritual successor to *X-COM*, *Phoenix Point*'s mutating alien threat gives the game a ghoulish atmosphere of its own.

The procedurally generated aliens: could you talk about how that concept came about?

The idea was to find something that was alien but not quite so alien, so we generated the idea of a mutation virus. The idea was that we can combine elements of humans, creatures, and then something very alien. The mutation system is designed to give the aliens a sense of progression and change over the course of the game, and also give the player some interesting and fresh challenges. The basic idea behind it is that we'll have one archetype creature, which is the Crabman, and he has various types of body parts. He doesn't have very many at the moment, but he'll have a lot more when the game's released.

The most complex creatures are these giant monsters, which are almost like battleships in the sense that they

have multiple armaments and defence mechanisms. We have a variety of combinations that number in the hundreds – body parts, abilities. It's not easy to implement, because it's tough on programming – everything's built from components and they all have to work with each other. It's quite an ambitious thing.

A tactical game's all about balance, so is it difficult to maintain that if you have all these combinations to account for?

Sure, but for the player, there's always something you can counter the aliens with. And usually, when you capture an alien, you can study its mutation, and it'll give you some hints as to what you can do to counter it, or it'll give you a new piece of equipment that you can manufacture to help you.

▼ The armoured soldiers of *Phoenix Point* can be traced all the way back to Gollop's own *Rebelstar Raiders* from the 1980s.



“What I appreciate about strategy games is having really unusual situations that no other player’s encountered before”

Looking back through your games, there’s a clear through-line from *Rebelstar Raiders* to *Phoenix Point*. What’s interesting is that you’re constantly refining your ideas. So is that like a game in itself, for you, that refinement?

Yeah, even going back to the original *Rebelstar*, I had the idea that you had these characters – even though it’s a squad-based tactical game – these were characters in a universe and they had a story. There was something bigger and wider and more meaningful going on. Obviously, that was developed quite substantially with the first *X-COM* game, though with *Phoenix Point* it’s definitely developing it even further. You have soldiers who can develop uniquely in different ways and, as with *X-COM*, your more powerful soldiers are the ones that you really try to protect and make sure they don’t die. On the other hand, they’re really powerful, so you need to use them.

On the wider, strategic level, I’m quite obsessed with the idea of linking things at multiple levels in the game, so *Phoenix Point* has a strategic level which has different factions with their own leaders. Those leaders can have personalities, objectives, and agendas which direct their strategic level. You as a player are intervening in this world in ways you can choose, but the world will nonetheless continue without your intervention. *X-COM: Apocalypse* touched on this – it had multiple factions and alliances and hostilities between them, but it’s much more fleshed out in *Phoenix Point*.

Going back to *Rebelstar*, what was it like to write the AI for that? It must have been difficult to create an AI that’s convincing, and fun to interact with, in just a few kilobytes. ➔

What I appreciate about strategy games is having some really unusual situations that maybe no other player’s encountered before – they give you something unique to defeat or overcome in some way. I do like to build a lot of randomness in my games – though of course it’s not pure randomness, it’s sort of contained within these rational heuristics. I guess a lot of 4X games do this: they have random rule setups and lots of customisation and so on. To some extent, *Phoenix Point* also has some influence from that in a way, but is not a 4X game in the classic sense.

That randomness goes right back to *Chaos*, in a way. It was a single-screen strategy game, but you had a spell like Disbelieve that could completely throw someone’s strategy out of the window in one move.

Yeah. If you’re playing with human players, it brings that whole psychological element to it.

▲ Julian Gollop: creator of *X-COM*, *Chaos Reborn*, and now *Phoenix Point*.



Released for the ZX Spectrum way back in 1985, *Chaos* formed the basis for *Lords of Chaos* and 2015's *Chaos Reborn*, and remains highly playable today.

It was a real challenge. The first game in the series was *Rebelstar Raiders*, and that was a two-player game. The sequel was just called *Rebelstar*, and that was originally also a two-player game. When I took it to show Telecomsoft, the company who were publishing the Firebird and Rainbird games, they said they needed a single-player version. I thought, "Oh shit, how am I going to do this?" I actually holed myself up in my room, because I was at college at the time, and I spent two weeks implementing the AI. I managed to do it in that time frame, and I think all the data and code for the AI was contained in something like 3.5kB of memory. It was relatively simple, but that was how it came about. It worked – but only just!

✓ The ability to go into an over-the-shoulder view and pick vulnerable spots on your target makes *Phoenix Point*'s tactics feel close to the *Fallout* playbook.



“You want to give a sense to the player that the AI is also thinking on a longer-term strategic level. You can put random aspects into that, and it kind of makes the AI seem more intelligent”

The AI made you feel like you were playing against a proper opponent – one that actually had weaknesses you could exploit. So as a designer, what is the approach to making AI fun to interact with?

The trick is that you want the player to be able to recognise patterns and behaviours. And once you've understood how the enemy's working, you can devise a counter-strategy that works. Then the player feels like they've achieved something – they've outwitted the aliens that are more powerful. The other trick, with AI, is that it needs to respond in some way. It needs to throw up a new situation

for the player – it's usually done in many games either by throwing in new types of units, or in some way changing the behaviours of the aliens themselves.

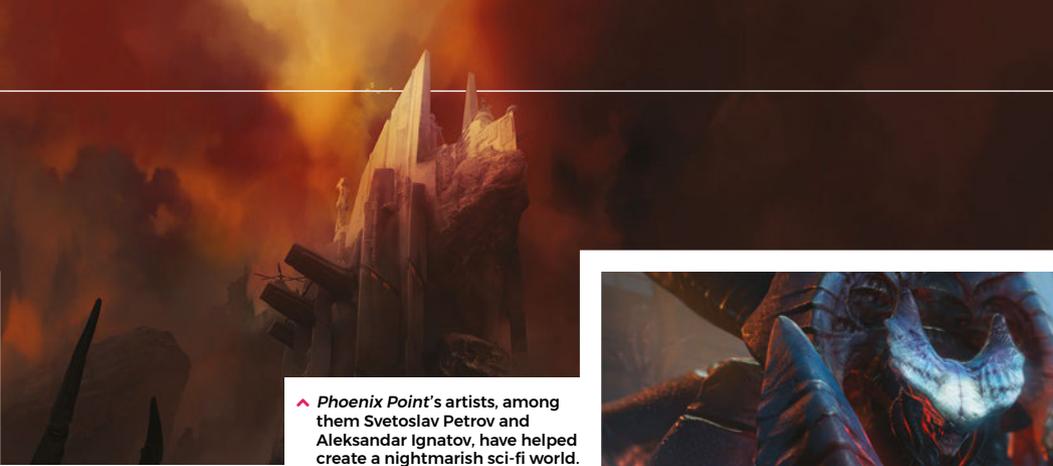
You want to give a sense to the player that the AI is also thinking on a longer-term strategic level. You can put random aspects into that, and it kind of makes the AI seem more intelligent because it's doing something different, even though it may have just randomly picked it. In terms of the player's internal narrative, the AI's responding. In fact, in games that involve prediction and bluff, some kind of element of randomness used in the right way is in fact often the most intelligent thing to

do, because you don't want to be totally predictable to the player. You need to keep players on their toes somehow.

In *Phoenix Point*, the aliens have a strategic level of AI. They will abandon mutations that no longer work very well against the player. The mutation system itself is quite random, so it will throw up some combinations to the player that aren't very strong, but it will also throw up some that are challenging. The idea is to give the player an occasional surprise, so they have to adjust their strategy and figure out what to do.

I saw in a recent interview, you said something like, “The original X-COM succeeded in spite of itself.”

Part of the problem was that bits of the game were only joined up together near the end of development. The problem was, how can I test it from start to finish? If I do test it from start to finish, and make various adjustments, how do I then test it



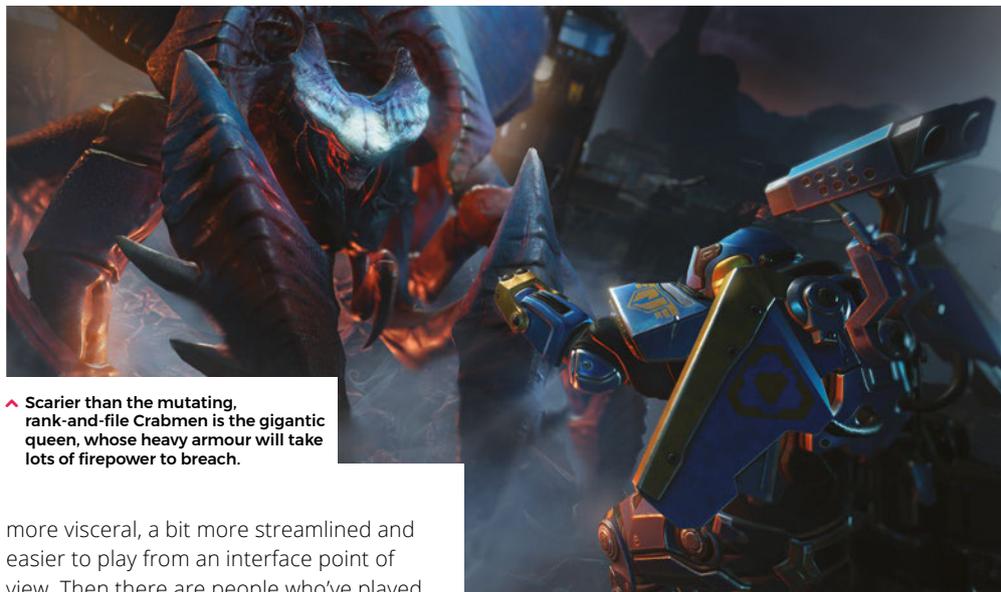
^ *Phoenix Point's* artists, among them Svetoslav Petrov and Aleksandar Ignatov, have helped create a nightmarish sci-fi world.

again? I could be here for years. This is the conundrum that we had. Even though we had a dedicated team of QA testers on the game for the last three months or so of production, the testing I was doing for the game balancing was kind of split. I would say, "I'll try and test the early game. I'll try and test the mid-game." Testing the late game proved more difficult – that's where the game falls apart a little bit, actually. It was very difficult to do, because it was so systemic – something that happens early in the game can have a knock-on effect later on, which can either make the game too easy or too difficult, or even impossible to finish. It's the risk you have with these long, complex strategy games.

With your recent titles, like *Chaos Reborn* and now *Phoenix Point*, you're creating games in an online environment where you can get feedback from players pretty much instantly. So how has that change been, given that in the 1980s you had no feedback until release?

Yeah, in the eighties I was the main audience for my games. If it didn't work for me, then I thought, "Well, that's enough." These days, that has got tremendously complicated. In the case of the *Phoenix Point* audience, a substantial number of them are players of the original *X-COM* games, for example. They really do want something very similar to that experience, right down to the mechanics. Then there's the audience who've only played the new *XCOM* games – they want something a bit

▼ **Regarded as one of the finest PC games ever made, *X-COM* spawned a franchise that's still going today.**



^ **Scariest than the mutating, rank-and-file Crabmen is the gigantic queen, whose heavy armour will take lots of firepower to breach.**

more visceral, a bit more streamlined and easier to play from an interface point of view. Then there are people who've played neither – they might be into these tactical RPG-style games, whether it's *Wasteland* or *Fallout* or whatever.

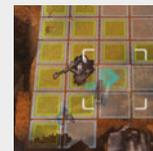
Managing audience expectations there is quite difficult. In *Phoenix Point*, we've tried to do something interesting: you have a class-based system, but you're allowed to take one soldier and give him a secondary task or a tertiary class if you want, and each class represents a set of abilities that are derived from training in that particular specialisation. So you can have characters which draw on abilities from different classes, or you can have one that's really specialised in one area. I quite like that flexibility.

So what's your roadmap for *Phoenix Point*? It's out in June 2019, but it's one of those games that could be quite open-ended.

Yeah, it's quite a modular system. But the main thing is to make a really good player experience first time around. Then we'll be planning several DLCs – we have tentative plans for a couple of those. They won't just extend the game by adding stuff, but really change some of the nature of how it works in a way, which could be really interesting. And we'll continue to support the game throughout the rest of next year through a combination of free updates and DLC, so all players will have something to look forward to. 🗨

***Phoenix Point* is scheduled to launch on Steam and GOG.com in June 2019.**

SHADOW WARS



Although he's probably better known for his computer games, Julian Gollop's also made some great

strategy titles for handhelds. One of the best was arguably *Ghost Recon: Shadow Wars*, released in 2011. Among the very best launch titles for the Nintendo 3DS, *Shadow Wars'* development was, according to Gollop, "Difficult" – partly because he and his team were working with an unfinished dev kit.

"We were saying, 'Why is the top screen bigger than the bottom screen, when it's the bottom screen that's the touchscreen? Surely they made a mistake?' By then we had touch input for the game working, we had all the action taking place on the bottom screen."

It was a matter of months before the console's launch that Nintendo finally revealed that the 3DS's top screen was stereoscopic – which meant that Gollop and his team at Ubisoft Sofia had to scramble to rework their game's screen layout and controls.

"We said, 'Oh shit, we've got to completely swap around which of our screens is doing what.' So we had to make the top screen the main screen, and then the bottom screen is just an informational screen. They really kept that secret until very, very close to the launch of the console."

Attract Mode
Early Access



Loaded Dice

VVVVVV creator Terry Cavanagh talks to us about his unconventional new roguelike, Dicey Dungeons

Info

GENRE
Roguelike

FORMAT
PC

DEVELOPER
Terry Cavanagh

PUBLISHER
Terry Cavanagh

PRICE
£5.90

RELEASE
2019

W

ith a history stretching back some four decades, the roguelike genre traditionally put its combat and procedurally generated

dungeons at the fore, with hidden dice rolls used to resolve skirmishes. *Dicey Dungeons*, from VVVVVV and *Super Hexagon* creator Terry Cavanagh, cheerfully puts these once-hidden rolls front and centre.

Dicey Dungeons first emerged during 2018's 7-Day Roguelike game jam and, according to Cavanagh, the inspiration behind it was rather straightforward: "It essentially started as a *Dream Quest* fan game," he tells us, "I was really interested in thinking about ways to take that deck-building roguelike concept in different directions, and this is my way of exploring that. More people should play *Dream Quest* – it's a much overlooked masterpiece!"

Recent builds include six character classes, along with a hidden Bear class, which requires certain actions to unlock ("The moment when players discover it is basically my favourite

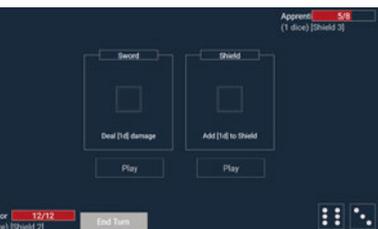
thing in the game right now," Cavanagh says). Each class has a difficulty rating and unique abilities; the Warrior is considered the easiest, with his special ability allowing dice to be re-rolled. The Inventor creates new gadgets after each turn by scrapping an item; the Thief can 'steal' an opponent's ability, while the Witch has a changing book of spells: "The Witch is my favourite class; it took a long time to get it right, though," Cavanagh reveals.

Dicey Dungeons' dice-rolling combat is easily picked up within a few minutes, but beneath the cuddly 8-bit graphics and chiptune music, there's some deceptively deep and absorbing systems at play. Players can toy with a broad variety of spells and items, while extra dice and bonus equipment are awarded as players level up. It's something that's being refined and expanded through subsequent builds, Cavanagh explains.

"The main thing I'm interested in doing with the game is taking this very simple set of core mechanics and, then, with different classes, exploring those systems in lots of cool ways," he says. "I'm currently working on a system which pushes that even further: a series of challenges for each character where the conditions, rules, and mechanics are slightly tweaked, requiring you to play differently and think about how you approach each character in a new way.

"I think these tweaks bring a surprising amount of variety and interesting challenges to the game. To me, this is really the central idea in *Dicey Dungeons* that makes it interesting."

Dicey Dungeons' distinctive look was created by Canadian artist Marlowe Dobbe, who worked



▲ The prototype *Dicey Dungeons* from this year's 7-Day Roguelike game jam.

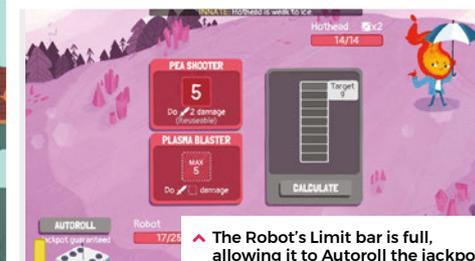
▼ The Thief has stolen 'Bump' from the Gardener, allowing him to increase the value of a dice.





▲ Procedural generation means *Dicey Dungeons* plays differently every time.

EQUIPMENT



▲ The Robot's Limit bar is full, allowing it to Autoroll the jackpot.

on *CARamari: Tentacle Four-Way* and *Dread Over Heels*. Her animation brings the enemies to life, and gives *Dicey Dungeons* a distinctive style all of its own.

"When I set out looking for an art director, the main thing I wanted to do was try and get away from the 'grimdark' aesthetic that so many roguelikes

have," Cavanagh says. "So I feel so lucky to have found Marlowe for this game – her artwork is amazing. It's so bright, colourful, funny, and full of life. I feel like there's nothing else out there that looks quite like this, and I love that."

Northern Irish composer Chipzel worked with Cavanagh before, on *Super Hexagon*, and is creating some excellent new chiptunes. "I think her work on the game has been incredible," Cavanagh says. "She has such a high standard for everything she creates, and she's so driven to make everything as good as it can be. She has this vision of what she wants the game to sound like, and it's been really inspiring."

Ahead of *Dicey Dungeons'* launch, Cavanagh is effectively running his own form of Early Access,

"I wanted to get away from the 'grimdark' aesthetic that many roguelikes have"

with purchasers of the current alpha version of the game offered any future upgrades for free.

"I don't think it works for every game, but I'm definitely planning on going with this approach again for future projects. With roguelikes in early access, it can be tempting to basically work on them forever,

but that's not something I'm interested in doing here."

The end is in sight, then, with *Dicey Dungeons* v1.0 out on Steam and itch.io in early 2019. With a healthy community building around the game already, Cavanagh has every reason to feel chipper about his latest indie project.

"I'm feeling very motivated to work as much as possible on *Dicey Dungeons* right now, which is a good sign," Cavanagh tells us, "This game has been a lot of fun to make." 🐼



▲ The formidable Dragon boss can 'burn' multiple dice.

A ROGUELIKE HISTORY

- 1975: *The Dungeon / pedit5*, a dungeon crawler with random monsters but fixed layouts.
- 1980: *Rogue* had players searching for the Amulet of Yendor, facing permadeath in ASCII-drawn dungeons.
- 1982: *Hack* added character classes, shops, and fortune cookies.
- 1983: *Moria* introduced different races, with defeating the Balrog the ultimate goal.
- 1987: *NetHack* added a variety of items (that could be blessed or cursed) and gained a dedicated dev team that keeps the game alive today.
- 1992: *Torneko no Daibōken: Fushigi no Dungeon*, the first in Chunsoft's *Mystery Dungeon* series (with later *Final Fantasy/Chocobo* and *Pokémon* spin-offs).
- 1994: *Ancient Domains of Mystery* takes place in Ancardia, with its unique Story Mode.
- 2005: 7DRL, the annual 7-Day Roguelike game jam, begins.
- 2008: The first International Roguelike Development Conference creates the Berlin Convention on what a roguelike should contain.
- 2012: Part of the roguelike renaissance, *FTL: Faster Than Light* releases to rave reviews.
- 2016: The first Roguelike Celebration is held, with key figures in the genre giving talks.



Super furry animals

Pixile Studios talks us through its violent but cute multiplayer shooter

Info

GENRE
Battle royale
FORMAT
PC / Mac
DEVELOPER
Pixile Studios
PUBLISHER
Pixile Studios
RELEASE
December 2018
(Steam Early Access)

It's the little details that make a game, sometimes, rather than the big, boundary-pushing concepts. Take the multiplayer shooter, *Super Animal Royale*: it's another gun-heavy free-for-all in the mould of *PUBG*, but it's the nuances that set it apart. First, *Super Animal Royale*'s viewed from an old-school, top-down perspective rather than in 3D; second, its horde of up to 64 combatants aren't buff humanoid soldiers, but cute, cartoon mammals. You'll see foxes running around with machine guns, bears throwing hand grenades, and cats pointing rifles – it's like a cross between a Saturday morning TV show and an issue of *Guns & Ammo*.

Aside from the adorable character design and animation, though, *Super Animal Royale*'s most eye-catching little detail is its use of line of sight. As your murderous critter runs around the forest map, its view of enemies and collectable items is frequently blocked by other objects. You can approach a tree, entirely unaware that another player's hiding behind it; you'll only know for sure if you're about to be attacked when either you or your opponent breaks cover. It's a design choice that adds a huge amount of impact to the game's suspense, as designer Michael Silverwood explains.

"It was really important that you could still have a level of stealth and surprise in close-quarters combat, so that it would feel dangerous as you're going around," Silverwood tells us. "Because if

you could see everyone everywhere, then a lot of the fights end up happening at the edges of the screen, because you're able to see each other all the time. They weren't exciting fights. So we brought [line of sight] in pretty early, and it's really important to the feel of the game."

Super Animal Royale's line of sight mechanic has a subtle visual impact on the game, too: the direction of light changes as you move around the map, which not only affords you a visual clue as to where enemies might be hiding, but also gives the 2D environment a greater sense of volume and depth.

Work began on *Super Animal Royale* in October 2017, with Silverwood and Pixile Studios co-founder Chris Clogg taking inspiration from other battle royale games they enjoyed, like *H1Z1* and *PUBG*. The idea of having scowling cartoon animals blasting away at each other came

about early on because, Silverwood says, he "really liked the juxtaposition" of 64-player violence and pastel-shaded whimsy.

"A lot of prototypes didn't see the light of

day, and then this one – we really liked how it was feeling," Silverwood says. "So we decided to leave our jobs and go all-in again and try to do another game."

Before *Super Animal Royale*, Pixile Studios' first game was an iOS title called *Stratosphere* – a 2013 tower defence game that Silverwood and Clogg made while they were still studying computer science in high school. For a few

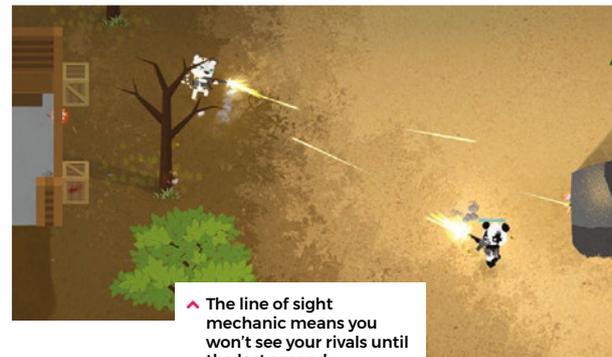
"It was really important that you could still have a level of surprise in close-quarters combat"

✓ The hamster ball is a charming way of getting around – plus you can use it to run over other animals.





^ Another squad of hapless animals drops into battle.



^ The line of sight mechanic means you won't see your rivals until the last second.

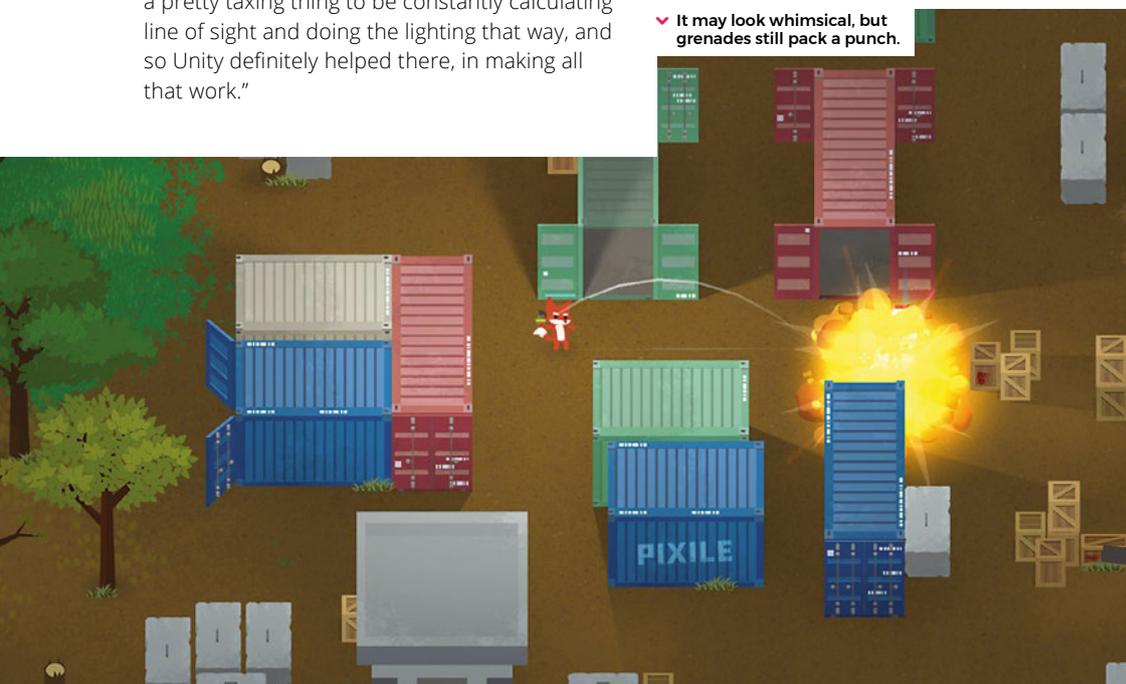
years, day jobs intervened, but the pair never stopped playing around with various game ideas and prototypes – even Silverwood's move from Vancouver to his current base of San Francisco didn't stop the two collaborating remotely. Indeed, *Super Animal Royale* is something of a global operation these days; character artist Amy Deng is based in Los Angeles, while animator Erika Signini is based in the Milan area of Italy. Together, the team have spent the past year or so working on *Super Animal Royale* across separate time zones.

Super Animal Royale was built in Unity, while the neat line-of-sight lighting effect mentioned above was achieved by repurposing an existing Unity plugin. "That was really helpful because of the performance," Silverwood tells us. "It can be a pretty taxing thing to be constantly calculating line of sight and doing the lighting that way, and so Unity definitely helped there, in making all that work."

Currently in its beta phase at the time of writing, *Super Animal Royale* has a single last-critter-standing mode to its name, but Pixile has plans for lots more online play options, including a co-op mode and squads. As it stands, though, the game's already an absorbing and tense little shooter; the range of weapons, from samurai swords to sniper rifles, are satisfying to use, while the game's sole vehicle – a hamster ball – allows you to zip around the map, gleefully running rival players over.

As in the games that inspired it, there's a certain thrill to dropping into the battle zone, hunting for weapons, and taking out your rivals in a hail of lead. That you can do this while in the guise of a panda wearing a top hat and monocle merely adds to the charm – plus, you can even do a little victory dance at the end if you want to. Once again, it's the little details that count. 🐼

^ It may look whimsical, but grenades still pack a punch.



MAPS TO THE STARS

Like a lot of developers, Michael Silverwood and Chris Clogg first got into game design by creating their own maps for games like *Team Fortress* when they were kids. "We were making maps for *Warcraft 3* and *Team Fortress*," Silverwood recalls. "So we were never professionally working on games, but it was a hobby for a really long time. We always wanted to do it."

As we'll explore in more detail on page 18, modding and making maps is a great way to learn the basics of game design. And while it's early days, Silverwood doesn't rule out the possibility of adding a map editor to *Super Animal Royale*.

"It's definitely something we'd like to do," Silverwood says. "We already have our own map-making tool that we use to build the map; because it's so big, we really needed a custom map-maker. Right now, Chris just uses it, but it would be nice, in the future, to polish it up, put a nice UI over everything, and let people make maps."





▲ An artist's impression of Rebellion's multimedia future. With golf carts.

Rebel filmmakers

British publisher Rebellion moves into the film and television industry with £78m studio plan



▲ *Rogue Trooper* is one of the 2000AD names that Rebellion plans to adapt.

Rebellion, the Oxford-based games developer and publisher behind the likes of *Sniper Elite*, *Battlezone*, and *Strange Brigade* – and owner of 2000AD comics – has purchased a \$100 million (£78m) facility to be used as a film and television production studio. The Didcot-based property brings with it over 20,000 square metres ready for conversion and, according to a release from the company, could create up to 500 jobs in the region.

This is Rebellion's second studio, with the firm also owning motion-capture specialist Audiomotion, known for its work on the likes of *World War Z*, *Harry Potter and the Deathly Hallows: Part Two*, and *Iron Man 2*.

Two big-name projects will be developed at the new Didcot facility, with the movie adaptation of 2000AD's *Rogue Trooper* to be directed by Moon and Source Code's Duncan Jones. *Mega City One*, the serialised spin-off of the Judge Dredd stories – focusing on characters aside from the stern-faced one – will also be filmed at the facility.

Away from Rebellion-owned properties, the new studio will be open to anyone who wants to (and can pay to) use it – some elements are open for business already, but the full roll-out of services will be completed by around spring of

2019. The company clearly hopes it can relieve some of the strain – and provide options – to a British film industry that's increasingly feeling the pinch.

KEEPING FLEXIBLE

It's still early days for the whole project though, and as Rebellion CEO Jason Kingsley told *Wireframe*, elements like who's being hired and for what are still being planned out: "There's no hard and fast answer to who we're going to hire and what we're going to hire them for because there's no hard and fast plan for which projects we're developing will actually go into production," he said. "So we have to remain flexible at this stage. But we're definitely putting together a list of people we can work with in the future."

Naturally, there's also potential for Rebellion's increasing push into the world of movies and ongoing series to incorporate the studio's gaming properties. It's not difficult to see *Sniper Elite* getting the green light, and a *Strange Brigade* serial riffing on Indiana Jones? That could work. Rebellion owns a wealth of gaming IP it didn't develop too, from *Evil Genius* to *Ground Control*, so there would be plenty to pick from.

Kingsley admitted to *Wireframe* that the studio would look to what it owns for inspiration:



"It's all part of our intellectual property," he said. "We have things like Roy of the Rovers, for example. We have a huge back catalogue of really quite sophisticated comic book products going back all the way to 1880. So yeah, there are an awful lot of things to look at and an awful lot of possibilities.

"There are lots of great stories to tell and video games are a part of it. So yes, if we can, *Sniper Elite* might be something we could do something with. Same with *Battlezone* – that would be an interesting one. Look at how brands have been made into live action before, and now with the state of visual effects being so sophisticated, who knows what we could do?"

Chris Kingsley, CTO at Rebellion, said in a statement: "The levels of growth in television content for streaming services, such as Netflix and Amazon, have been astronomical in recent years. The demand for ongoing original content is bigger than ever and we're seeing more big players wanting to get in on the action. This is very exciting for the domestic and global film industries, but it's also meant that our infrastructure is under increasing pressure.

"The new studios will help relieve some of that pressure whilst also adding to the growing figures of people employed in the UK film and TV industry, which has increased by 20 percent in the last five years. Our creative industries are appealing to other markets for our talent and generous tax reliefs – and we must ensure we have the studio space and infrastructure to keep furthering this ongoing growth."

Funding for the facility has been supported by HSBC UK, and praise for the expansion of the UK film and television industry has come from far and wide. "This is yet another vote of confidence

▲ The 2012 *Dredd* movie wasn't a huge hit, but it has a huge cult following. A *Mega City One* TV spin-off is coming soon.

"It's all screen entertainment, some of it is interactive and some of it is not"

in our world-leading creative industries," commented Margot James, minister for digital and the creative industries, "and the government is committed to stimulating creativity, broadening opportunities, and securing even more growth for this booming sector."

A more human reaction came, meanwhile, from Stuart Fenegan, producer on a number of Duncan Jones's films: "World-class UK crews and Tax Credit mean studio availability has been scarce in recent years," he said. "Jason, Chris and the Rebellion team establishing another major UK studio is huge. Thrilled for them – and at the prospect of being able to shoot *Rogue Trooper* on truly home turf."

Meanwhile, Rebellion will still operate its gaming division – there's no fear of a loss of focus there, and as Kingsley told *Wireframe*, developing the movie and TV side of things will have "absolutely zero" impact on the games side of the business from a strategic point of view, "although it's nice for people working in games to know that we are doing different things."

"Some people in games might want to help out with working in film and TV, and vice versa," he added. "It's all screen entertainment, some of it is interactive and some of it is not! So no, we are still hugely focused on making big video games. There are a number of announcements coming up that we are excited about. But this is an expansion rather than a shift in focus." 🗨

DANCE OFF

Fortnite's push to take over the world came with a lot of dance emotes in tow – but now it seems the people who actually came up with the dances to begin with are starting to get a mite annoyed. Rapper 2 Milly said he intends to take legal action against Epic for its purported use of his move, the Milly Rock, in the game. The derived dance, known as Swipe It, has since disappeared from sale in *Fortnite's* store.

Meanwhile *Scrubs* actor Donald Faison turned down a request to perform his famous dance from the show, which has since been copied – and popularised – on *Fortnite*. "I don't get no money," Faison said. "That's what y'all are thinking, right? Somebody got paid? No. No. I did not. Somebody stole that shit, and it's not mine any more."



Attract Mode



Eastward ↗

Shanghai-based Pixpil is behind this intriguing little number – a single-player, dual-character RPG mixing dungeon crawling and combat with a lot of trains. With its mix of dark subject matter – this is a post-apocalyptic ruin of a world – with a lighter dynamic brought about by the visual style, there looks to be a lot to look forward to with *Eastward*. And with *Stardew Valley* publisher Chucklefish on board, it's sure to get in front of a fair few eyes.

Release date: TBA 2019

Afterparty ↘

The next big thing from *Oxenfree's* Night School Studio, *Afterparty* sees you take control of a couple of friends involved in a drinking contest. Simple. It is fair to note, though, said friends are recently deceased, they're in Hell, and the drinking contest is against Satan himself. Your actions in *Afterparty* can have huge consequences for Hell and its denizens you meet with throughout. It's full of minigames and plenty to discover – but really, this is one about having a drinking contest with Satan.

Release date: TBA 2019



Mineko's Night Market ↗

Cat lovers rejoice! *Mineko's Night Market* combines elements of farming, mining, crafting, night market-ing, and – naturally – catting. 'Whimsical' seems an apt word for this twee little number making its way to PC and Switch from Meowza, the studio behind the likes of *Alphabear* and *Road Not Taken*. Whatever happens, it will be adorable.

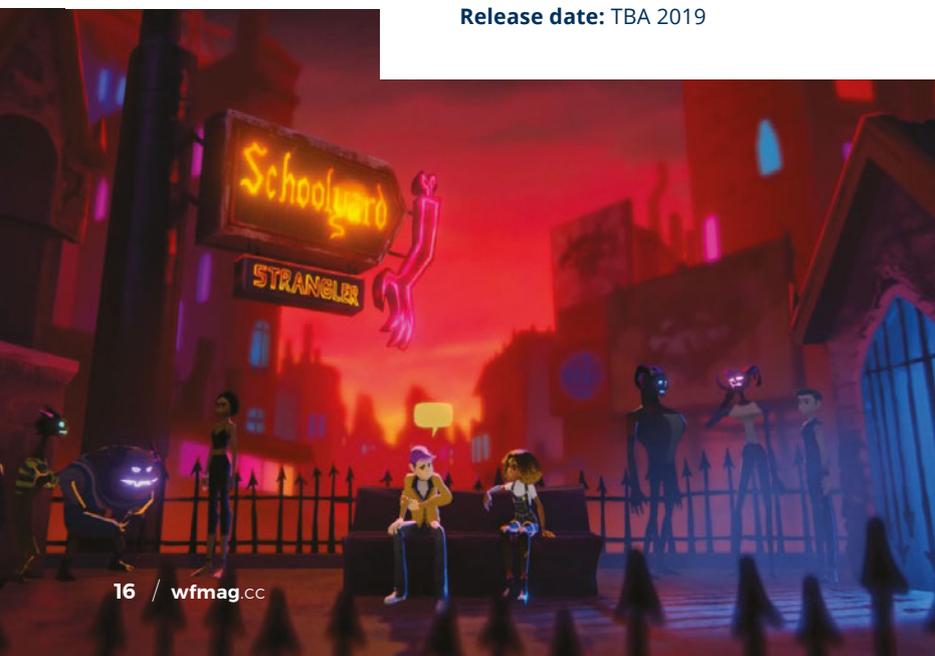
Release date: TBA 2019



Onimusha: Warlords ↗

Capcom has not one, but two remasters coming in January – *Resident Evil 2* is firmly in the camp of full remake, while *Onimusha: Warlords* is getting a lick of paint (i.e. a higher resolution), analogue controls, and a new soundtrack. It clearly isn't getting as much love as *Resi*, just like *Onimusha* didn't back in the day. Hopefully the game itself holds up.

Release date: 15 January 2019





Knights and Bikes ↗

On the quaint little British isle of Penfurzy (it's not real, don't look for it), friends Nessa and Demelza spend the late 1980s exploring their own private kingdom of play on the back of their bikes. As they make their way around on their trusty metal steeds, the pair will encounter a mystery to uncover, new friends to be made, and denizens of Penfurzy in need of rescue. As well as puddles to splash about in – they are kids, after all.

Rex Crowle and Moo Yu, who between them worked on *Tearaway*, *LittleBigPlanet*, and *Ratchet and Clank*, spun off into the world of indies a few years ago, running a successful Kickstarter for *Knights and Bikes*. The £126k raised for their *Goonies*-and-*EarthBound*-inspired romp looks to be being used well so far, and the hand-drawn 3D world already has us enraptured. Anyone who ever went on a rainy caravan holiday in their younger years is sure to feel the same, and we'll be taking a closer look at *Knights and Bikes* – and Demelza's pet goose – in an upcoming issue.

Release date: TBA 2019

UFO 50 ↘

Fifty games from the creators of *Spelunky*, *Downwell*, *Time Barons*, and more, *UFO 50* brings together all manner of styles, single- and multiplayer, and even manages to tie it all together with some vague storyline reasoning (each game is, in story, made by a development studio in the eighties). Emulating 8-bit classics while infusing them with modern elements – from all manner of stylistic backgrounds, as well as from indie devs with proven gumption for making things that balance simplicity with pure fun – could be a recipe for something special.

Release date: TBA 2019



Black Future '88 ↘

Set in a never-ending alternate 1988, *Black Future '88* sees players try to fight their way to the top of an ever-changing procedural tower, before their heart explodes. It is described as 'synth-punk', and said descriptor is incredibly fitting. Multiple characters and buffs along with local co-op have helped to plonk this roguelike shooter firmly on the radar, and its extended pre-release development should be coming to a close soon.

Release date: Spring 2019





The route less travelled

when modders turn pro

We talk to the modders who got into games for fun, and later made a career out of making games for fun – and money

^ There were few dissenting voices in the room when XCOM: The Long War's small team revealed it had been hired by Firaxis to produce DLC for XCOM 2.

If ever any element of the games industry has truly embraced a culture of minor rebellion, it's the modding scene. A world of amateur developers taking existing games and tinkering with them – ripping them apart, building them back up, standing on the shoulders of (professional) giants, it's all fair game. And it's all done without asking permission: simply grabbing what those established teams have worked on for years and making it their own.

While the boom of the indies has seen modding take a step back in the public eye, there are still plenty of folks out there working hard to craft something fun, gorgeous, ugly, boring, nude, or whatever it might be for an existing game. And some folks, sometimes, manage to take a step beyond and become professionals, all via the entry point of the modding scene. This is the story of a few of those people, their mods, and the games they ended up bringing to market.

Modding has existed in some form or another for as long as games have been around – just as with any aspect of tech, there's always going to

be someone who wants to tear it open, even if the 'it' in question is a stack of code, and root around inside. From there they'll see how it works and figure out ways of making it do different things – it's human nature to sate this curiosity, and its business nature to realise there's something to cash in on there. One of – if not the – earliest mods around set something of a standard for the micro-industry, by taking an existing property (*Castle Wolfenstein*) and adding in something of contemporary pop culture (*The Smurfs*). *Castle Smurfenstein* was born in 1983, and a whole new world of opportunity with it.

It only took until 1984 before the release of *Adventure Construction Set*: a tool, rather than a game, which allowed users to create their own adventure games in all manner of different flavours and – via the medium of copied floppy disks – share them around their mates. It was a limited commercial approach to the nascent hobby of modding, but it showed quite clearly where things were heading: publishers had an eye on what people were crafting in their homes, and they weren't afraid to get in on the action if they saw benefit in doing so.

▼ Aliens TC fundamentally altered the style of play from *Doom*, making for a more suspenseful, tense game than the bombastic original it was based on.



DOOM'S PUSH

The explosion in modding around the original *Doom* back in 1993 brought mods out of the niche and into the mainstream, with amateur teams and individuals continuing Castle Smurfenstein's approach of slapping well-known properties – this time the likes of *The Simpsons* and *Aliens* – into what was then the world's most popular game. There are distinct memories of this period, of walking into gaming stores and being able to play Justin Fisher's *Aliens TC* ('total conversion') mod, with staff acting near-enough as though this was an actual, proper game.

Such a push from *Doom* did finally tip modding over the edge, with results like various officially licensed mission packs releasing at retail – as well as the very official, very much part of the 'proper' *Doom* franchise, *Final Doom* featuring no less than two whole campaigns designed by mod teams. Something new had started: people were seeing that they could use modding as a route into actual development *as well as* using it as a creative/mucking about outlet.

Nick Pearce created a mod for *The Elder Scrolls V: Skyrim*, called *The Forgotten City*. It took 1,700 hours of his time to make, was critically lauded, and downloaded 1.8 million times. With a track record like that, it's of little surprise – though no less impressive – that the Australian former legal advisor took the jump into full-time game development, turning *The Forgotten City* into its own, complete, standalone title.

"I knew making another mod in my precious spare time wasn't sustainable," Pearce explains. "So I started thinking about how I was going to keep making game projects, without working myself into an early grave, or starving to death.

"I considered my options. Modding is no way to make a living – for now, at least"

I considered my options. Modding is no way to make a living – for now, at least. Getting a job at an existing studio wasn't an option because in Australia, where I'm committed to living, the industry is not well developed and there weren't any suitable job opportunities.

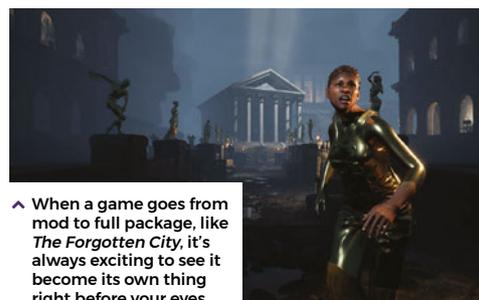
"My last option was making an indie game, which is a hugely risky move, of course,

particularly for me as it meant taking a break from the legal career I'd worked for ten years to establish – and I might never

be able to go back. But it was also the only way I could pursue my passion for making games." Pearce, of course, took the risk: with the track record he'd built up through the original mod's release and reception, he was able to get support from Film Victoria, an arts funding group in Australia, and the project is in full development – it even has its page on Steam up and running.

IN TRAINING

Some don't take the route from modding to indie development, of course, instead moving from ▶



▲ When a game goes from mod to full package, like *The Forgotten City*, it's always exciting to see it become its own thing right before your eyes.

▼ Nick Pearce, creator of *The Forgotten City*.



CHEX OUT THIS MOD

It's definitely an odd one, but in 1996 a mod appeared bundled with packets of cereal in the US. *Chex Quest* was a modification of *The Ultimate Doom*; a total conversion with the violence removed and the product placement through the roof. While not an amateur production – it cost about \$500,000 to make – *Chex Quest* was a defining moment for the development of mods, showing the money men knew the power present in modifying existing properties. The game saw two full sequels, though we've yet to hear if licence holder General Mills has been bitten by the *Chex Quest* battle royale bug.

THE SPECIALS

It's no guarantee of success, but these four mods turned into something special for their creators:

Counter-Strike

At times more popular than *Half-Life* – the game it was a mod of – *Counter-Strike* led to its creators being hired by Valve.

XCOM: The Long War

A similar story, the Long War mod was so good the team behind it was brought on by Firaxis for *XCOM 2* to create bespoke DLC.

Defense of the Ancients

Not one, but two games came from the *Warcraft III* mod – *Dota 2* from Valve, and *League of Legends* from Riot. And they're two of the world's biggest games.

Black Mesa

A passion project revamping the original *Half-Life*, *Black Mesa* was given Valve's blessing to be released as a paid product on Steam.

▼ Matt Peddlesden of *Train Sim World*.



the mod scene to that of working as part of a larger, established team. The thing is, some people don't actually intend to do things that way – it just sort of... happens. Matt Peddlesden, senior producer on *Train Sim World* at Dovetail Games, was a software engineer working on telecoms, banking software – that kind of thing – by day. By night he was a passionate train modder extraordinaire, building his portfolio to the point that the pros came calling, with the likes of the *Microsoft Train Simulator* team tapping him up for designs and other content.

"On a number of occasions I was fortunate to be invited to work with the original developers of *Train Simulator* to consult on upcoming projects they were working on," Peddlesden says, "and then later with the team that eventually became Dovetail Games. Knowing the team well, at a point in their growth they asked if I could come on board full-time to help bring other modders and third-party developers on board, as well as help with some of the production of internal projects. Suddenly the hobbyist modder was a professional game developer."

▼ With such a focus on the amateur modders at home, it's of little surprise that members of the *Train Simulator* team come from a modding background.



That leap from amateur, doing things in your spare time for the love of it, to professional – doing it because it's your job and if you don't you risk losing your house or starving – is a tough adjustment. "You can spend any amount of time on a project," Peddlesden explains. "[Whereas] commercially you need to balance the cost of creation against the return you're likely to receive (otherwise nobody's getting paid!), as a free modder you can spend three years making something that ten people will hugely enjoy because it's simply something that you're passionate about."

Pearce paints a similar picture: "When you make a mod, you're cherry-picking all the fun stuff from game development," he says. "Creative writing, level design, scripting, casting actors, and making trailers. [Whereas] when you make an indie game, you sign up for a huge amount of serious, boring work on top of that, like starting and running a company, securing funding, budgeting, tax returns, recruiting collaborators, contracts, licensing, version management software, protecting intellectual property, marketing and PR, and so on."

It's not all bad, of course – the chance to work with a professional team, to be able to profit from your labour, to see a project become more than just a few bits and pieces layered on top of an existing project, that's all brilliant for the modders-turned-professionals. It's just, as with anything, a sudden shift from one way of working to another can be jarring, and it won't be an approach that sits well with everyone.

TRADITIONAL PATHS

Even modders don't have to take the path of mods-to-pro, of course, and there are plenty of other 'traditional' routes into development: QA, studying game design at university, placements and internships, and so on. But there's a special kudos that comes with going from modder – hero of the people, crafting the projects the big publishers don't want to touch – to taking that same project into the big leagues and making something that combines the initial, great idea with the financial clout of a proper publisher. It just... isn't an approach for everyone.

"The indie marketplace is so terrifyingly overcrowded right now that you need something big to distinguish yourself – an X factor – to have any chance at breaking through all the noise," Pearce says. "Having a popular, award-winning



^ Arguably the most legendary of the mods-done-good, *Counter-Strike* changed the landscape in so many ways.

mod under your belt worked for me; it meant that PC Gamer invited me to reveal *The Forgotten City* standalone game on stage at their E3 show, here. If I didn't have that advantage, there's no way I'd even consider making an indie game. It would just be a waste of time."

Peddlesden has words of advice for those definitely aiming for a game development position, meanwhile: "Start really paying attention to what the developer of your favourite game is doing: what kinds of products they release and why they're doing that. Start thinking about how you can make exceptionally great mods that put your name above the crowd, but think about how you can do that in a way that looks commercially reasonable. How can you make mods that work well with the product set that the company is putting out. Is there a particular area such as modelling, audio, domain knowledge, etc. that you excel at and could put out some packs that enhance that aspect of the game?"

Branding is a key aspect, Peddlesden says, and features like Steam Workshop allow modders to build up a community around their projects in a simpler, more centralised fashion than ever before – so it's key to chat with your fans, he advises.

"If you're so inclined, it wouldn't hurt to start building YouTube and/or Twitch communities either," Peddlesden continues. "Stream some content creation tutorials, for example. Later, when a game developer hires you, they'll be making a big splash announcement of the fantastic new personality they've introduced to their team!"

MOBA MODS

Modding embedded itself in gaming culture so fiercely that it has directly led to three of the most popular genres/titles in all of gaming today, with each initial mod setting their creator/s on a path to professional, paid work.

Back in 1998, a modder known as Aeon64 created a custom map for Blizzard's RTS phenomenon, *StarCraft*. Aeon of Strife, as it was called, set the ball rolling into what would become a *Warcraft III* mod, *Defense of the Ancients* – a fan-made tweak to the core formula, which was then itself iterated on by other players, adding more features, fixing issues, and generally acting as enthusiast QA support.

The original *Dota* mod ended up – in its various guises – an incredibly popular add-on for *Warcraft III*, so it was of little surprise that its creators would spin off to create something

using the formula – only this time for actual monetary reward. *League of Legends*, from Riot Games, was the first full, professionally produced game from the original *Dota*'s creative team. The other was a direct sequel, developed by many of those who worked on the original mod, who were scooped up by Valve to produce a standalone version. *Dota 2* arrived in 2013, and has been trading blows with *League of Legends* for supremacy ever since. Regardless of which you prefer, the simple fact is: a fan-made map for the original *StarCraft* led directly to two games which see eSports prize pools in the tens of millions of pounds each and every year.

The second way mods have directly led to some of the most popular games and genres comes via the more direct route thusly: *Half-Life* released. *Half-Life* saw a mod brought out for it called *Counter-Strike*. *Counter-Strike* was very popular indeed. Valve (as with *Dota*) came a-knocking, and asked *CS*'s creators if they →

"Think about how you can make exceptionally great mods that put your name above the crowd... and are commercially reasonable"

ARE YOU READY?

Nick Pearce has five questions for those looking at taking a mod pro – if you answer yes to all of these, "it's still risky, but you'd have a fighting chance."

- Do you understand what's involved in making games, and do you have a burning desire to do it anyway?
- Do you have a way to survive while making your game, like savings, contract work, and/or a supportive partner?
- Are you resilient and able to deal with challenges, setbacks, and tedious business activities? (Securing finance, setting up a company, bookkeeping, contract management, protecting your IP, etc.)
- Are you a strong networker? Are you good at forming genuine connections with peers, journalists, and influencers?
- Are you a fast learner with a strong aptitude for technology?
- If you're converting your mod into a game, have you obtained legal advice and secured the necessary intellectual property rights?

✓ Basing a game on a *Warcraft III* mod, itself based on a *StarCraft* map, might seem an odd way to make one of the world's most popular games – but that's what *Dota 2* went and did.



▼ The original *Natural Selection* was a mod of *Half-Life*, but success was such that the team – for the commercial release of the sequel – developed its own, in-house engine, *Spark*.



▼ Someone decided to make *The Hunger Games* in *Minecraft*, and suddenly we had a genre – battle royale – that had absolutely taken over.

wanted to come and make things official. Today, *Counter-Strike* is still played by millions, and pretty much verges on being a religion to some.

Third, there's the little story of something called *Fortnite*. The game itself isn't a mod – it's an original production from a monolithic publisher, the suitably named Epic Games. It isn't a story of triumphing over the fat cats or a few developers making something teams of 300 couldn't manage; it's just a great game made by a huge studio. But the *idea* for something like *Fortnite* came from two modding sources, and without them we likely wouldn't have the world's current most popular game.

HUNGRY

One mod was for the ever-popular *Minecraft*, originally entitled *Hunger Games* – renamed *Survival Games* because litigation is a thing – it saw up to 48 *Minecraft* players fighting to be the last person standing in an arena. Then there were the various mods to a game that was already a mod itself, *DayZ*, which pared back the action to a more frenetic, seat-of-your-pants survival situation – again, with the last person standing declared the winner. These game

modes struck a chord with plenty of players, and it surprised literally no one when plenty more commercial products hit the market with this new-fangled 'battle royale' mechanic at their core. *Fortnite*, *PUBG*... even *Call of Duty* has added a BR mode in its latest iteration – and it all comes from a few players sat at home, making something they thought might be fun in their spare time.

It's stories like these that can motivate, and it's definitely something that's close to both Pearce and Peddlesden. "I really liked the story of the Chinese Room," Pearce says, "which became famous for making their experimental narrative mod, *Dear Esther*, into a standalone game. It was exciting because it was the first time I'd ever seen a really artistic, poetic, and abstract game become that popular, and it seemed to herald a new age for video games as a storytelling medium." While he admits he wasn't a fan of the game per se, the ideas on show offered something new – potential Pearce hadn't really associated with gaming before.

Peddlesden's motivational tale, meanwhile, comes from closer to home: "[I] was working with a publisher creating the first retail add-on pack for *Microsoft Train Simulator*. I'd built some models of a British locomotive called a Deltic and even though they weren't the best, they were unique in that there weren't too many custom locos around at the time. They'd asked if they could put them in the pack, which I was happy to

GECK A LOAD OF THIS

Modding these days isn't any easier than it used to be, but it is simpler to get your foot in the door. *Skyrim* and *Fallout 4* have a full set of tools – Creation Kits – available for download by anybody who owns a copy of either, and *StarCraft II* maintains a robust set of mod tools for players to make their own content. There's the likes of *Trackmania* and older games like *Neverwinter Nights* have much-praised modding tools, and even more recent titles – and their console versions – like *Far Cry 5* and *Doom* feature solid tools for making your own maps, modes, and minigames. There are plenty of opportunities, basically.



▲ If it weren't for mods, we likely wouldn't have current world's most popular game *Fortnite*. Fact.

agree to; as a freeware developer I just wanted my content in as many hands as possible! They showed me some mock-ups of the retail box which featured an American locomotive (not entirely surprising as it was a US company compiling the pack) and promised to send me some copies once it was all completed.

"Upon its release, I travelled to my nearest Game store and wandered over to the shelf to see if it was there, and staring back at me on the front cover of quite a large cardboard box was my Deltic locomotive as the front image on the pack. I was completely shocked and it took me a few moments to recover.

From that point, I became keen to create more content that found its way onto retail shelves."

THE HORROR

There are horror stories, of course. The tale of Trauma Studios – purchased by DICE for its work on *Desert Combat*, a *Battlefield 1942* mod, before being closed down just a year later – often springs to mind. And there are more failed projects out there than it's possible to count – or list on a page with a limited word count. It's never

▼ Tripwire Interactive continues to go from strength to strength, emerging as a mod team and – most recently – putting out *Killing Floor 2* to good reviews.



a safe option to make mods, and it's certainly not a safe option to go pro, even with the backing of a big publisher's millions. But that risk, for the sake of pure creative drive, is often worth it.

Mods have been near-enough ever present in gaming – the desire to tinker is just too strong. From them we've had some wonderful, throwaway experiences, we've had games fixed or made to perform better than their actual developers could make them, and we've had some genuinely brilliant, creative, unique takes on what games can do and be. The move towards digital distribution and the rise of the indies over the past decade has made it more fruitful than ever for the modders to turn pro, even if some of the modding community does see that as sacrilege.

But with the engines out there for free, the mod kits created by the developers themselves, marketplaces like Bethesda's offering modders cash for their creations, and a greater push in

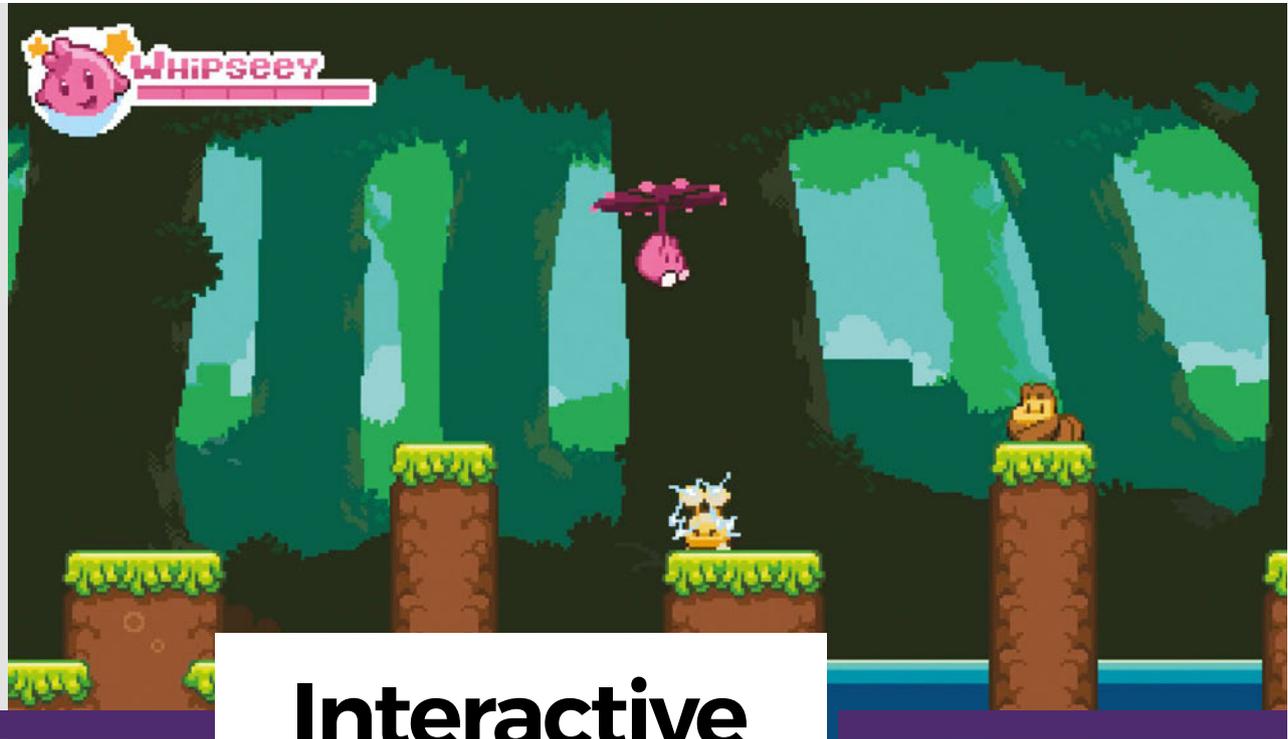
the crowdfunding world – *Grand Theft Auto V* modders are able to make money on Patreon – the future is looking bright

for all aspects of modding, not just those who want to turn pro. But as we've seen numerous times in recent history, a single mod – with professional backing, the right development nous, and a bit of luck – can change the wider, non-modded gaming world as we know it. 🗨

"It's never a safe option to go pro... But that risk, for pure creativity, is often worth it"

SOURCING MODS

If you're looking to pick up mods, there are plenty of official repositories these days – games like *Skyrim* and *Far Cry 5* have in-game menus leading you to the best of their unofficial downloads. Steam, meanwhile, hosts the Workshop for many games on its platform, meaning you can find all manner of tweaks and alterations through the storefront. There are also plenty of mods still available through unofficial capacities, with the ever-brilliant ModDB proving an invaluable resource for your user-made content for games past and present. In fact, ModDB tends to be the only reliable site for plenty of older mods, especially those from the *Doom / Quake* era.



Interactive

Whipseey, the console platformer homage from El Salvador

Developer Daniel Ramirez talks us through the highs and lows of making his 8-bit-style platformer, Whipseey

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

A

lthough making games is a fun hobby, it can also be immensely challenging – sometimes, even all-consuming. Take *Whipseey And The Lost Atlas*, a 2D platform game

that takes its cue from the era of games like *Kirby's Adventure* on the Nintendo Entertainment System. Creator Daniel Ramirez began work on the game in 2017 while he was still in university in El Salvador; he initially thought the project would only take a few months. One year later, and development is still ongoing.

"When I started the project, I naively thought that it would take around six months," Ramirez

tells us. "It's been a year and we're still working on it, mostly because we have expanded the scope of the game. It's close to being finished, but I'd be lying if I said that a year of development hasn't taken a toll on my mental health in some way."

Whipseey began life in the summer of 2017 as *Whip Master* – a title that describes its squashy hero's ability to swing across gaps and knock

out enemies with a well-timed lash of his scourge. Development progressed smoothly, with Ramirez posting regular updates and colourful GIFs of the game in action on Twitter – a tactic that helped garner some much-needed attention, and also attract a trio of useful collaborators.

"I've been very lucky to find very talented people to work with me," notes Ramirez. "Roy Nathan de Groot takes care of all the art for the game, Benji Inniger is working on music composition

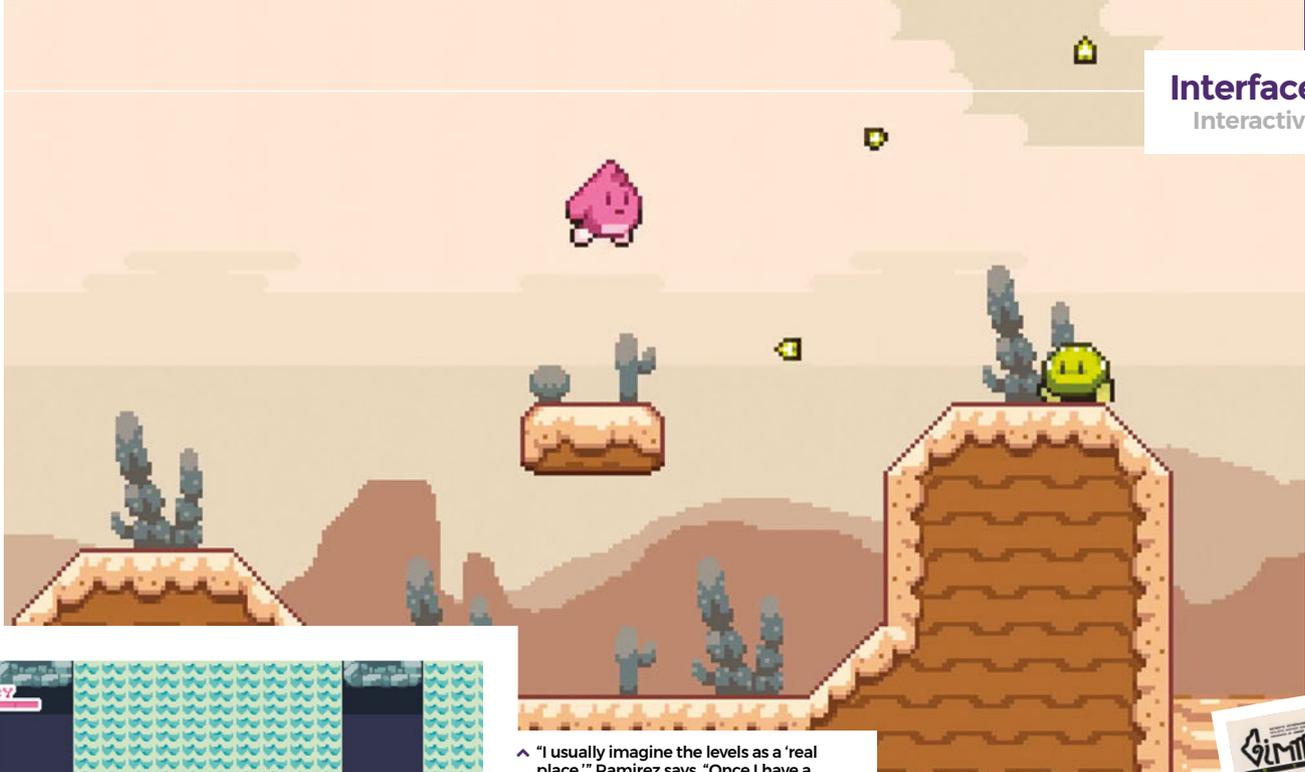
and sound design, and Bruno Rodrigues Pablo has recently joined our team to help with programming."

Social media has had a downside, however. In

April this year, a rather cruel individual essentially stole Ramirez's concept – right down to its title and early sprite designs – and put the clone on the Google Play store.

"Some people contacted me asking if the game had been released or if it was a spin-off, and that's how I became aware of the situation," Ramirez says. "I contacted the developer and he agreed to take the clone down."

"I wish I could give some advice to stop people from cloning your game... it's such a big issue in the industry"



^ "I usually imagine the levels as a 'real place,'" Ramirez says. "Once I have a general idea, I start designing the enemies, how they should behave, and how the player will overcome that challenge."

^ According to Ramirez, the whip mechanic was adjusted "multiple times" until its handling felt just right.



A rare inspiration

As well as *Kirby*, *Whipseey* is also influenced by a much more obscure hero from the NES era: *Mr. Gimmick*. Made by Sunsoft towards the end of the NES's life, *Mr. Gimmick* was another colourful – and difficult – platformer, and produced in such small quantities that it's keenly sought after by collectors today. For Ramirez, Sunsoft's ability to make such an intoxicating world on the NES's limited hardware remains a major inspiration.

"*Mr. Gimmick* is probably my favourite game ever," Ramirez enthuses. "I fell in love with it the first time I played it. I don't think those games continue to be alive only for the nostalgia factor, I think that great game design is the reason why those games stand the test of time. The 8-bit era allowed game developers to create game worlds that felt as if they actually existed inside your TV."

^ Sunsoft's indescribably rare *Mr. Gimmick* on the NES: a key inspiration for *Whipseey*.



With the crisis averted, Ramirez decided to change the name to *Whipseey And The Lost Atlas* – a title better suited for search engine optimisation, and also unique enough to protect from would-be copycats.

WHIPPED INTO SHAPE

"Sadly, it's the type of thing that makes indie developers afraid of sharing their projects online," Ramirez says. "I wish I could give some advice to stop people from cloning your game, but this is such a big issue in the industry that I feel like I have no good answers. It's very difficult for an indie to fight this type of stuff. Having a publisher that has your back when it comes to legal issues could help a lot; also, having a strong community that follows the development of your game can help when it comes to reporting game clones to stores."

Despite all this, work has continued apace on *Whipseey*, with Ramirez quitting his part-time job as a computer repairman to give the game his full attention. *Whipseey* is a self-described love-letter to

the 8-bit Nintendo era, from the vibrant character designs to the up-tempo music – and the difficulty level, Ramirez says, will be similarly old-school.

"I ask my team to help with level testing and to assess the difficulty. I play a lot of hardcore NES titles like *Ghosts 'n Goblins*, *Mega Man*, and *Ninja Gaiden*, so what I would consider difficult might be a bit too hard for most people. I always try to keep that in mind."

Making *Whipseey* hasn't always been easy, then, but Ramirez says the end is now in sight – the finished game is currently scheduled for release in the second quarter of 2019. And, despite the whole cloning incident, Ramirez argues that social media has still played a hugely positive role in getting *Whipseey* noticed.

"Social media has been vital," Ramirez reveals. "I met everyone on my team through Twitter. Everything that has happened to the project – good and bad – has had some connection to social media." @



> Although it looks like an 8-bit title, *Whipseey* was created in Game Maker Studio, while early versions of the sprites were made in GraphicsGale.

Put on your game face: it's triple-A time



LOTTIE BEVAN

Lottie Bevan is a producer and co-founder of the experimental narrative microstudio Weather Factory, known for its Lovecraftian card game *Cultist Simulator*. Before founding Weather Factory, she was producer at Failbetter Games where she worked on *Fallen London*, *Sunless Sea*, *Zubmariner*, and *Sunless Skies*.

“Triple-A’s weakness is that it’s hard to humanise spectacle”

In November, Riot Games set the world on fire. The spark was K/DA’s *Pop/Stars*, a K-pop song which debuted at *League of Legends*’ 2018 World Championship. It came complete with Korean pop idols, augmented reality, a Justin Bieber protégé and a badass music video, which was watched 38 million times in the first week alone. Its coolest moment is an extended neon rap section, which is probably responsible for a 38 million percent increase in graffiti, too.

You might think all this spectacle was announcing a new multimillion dollar IP. Or to reveal something nuts, like Riot working with Valve to finally make *Half-Life 3*. It, er, wasn’t. Its immediate purpose was selling four new cosmetic *League of Legends* skins. It was part of larger marketing and visibility for Riot too, of course, but it remains the point where I, an indie dev, throw myself into the sea.

Don’t get me wrong: everything about this is epic. It’s a promotional triumph. It’s the catchiest song I’ve heard in while. It’s a PR stroke of genius, somehow managing to balance four consumable, sexualised women made by an allegedly sexist studio and dancing for a majority male audience with companionable Spice Girls-esque feminism. Who do you like the best? Meanie Spice? Painty Spice? Foxy Spice? Or the one who isn’t very memorable and gets the least screen time, N/A Spice?

But this sort of spectacle is impossible for indies. The greatest indie success stories – *Stardew Valley*, *Papers Please*, *Spelunky*, even *Minecraft* – don’t have gigantic real-world annual events with K-pop supergroups selling in-game stuff. In Roger Zelazny’s *Lord of Light* – a Buddho-apocalypse sci-fi novel you must read – he accidentally sums up the difference between triple-A and indie: ‘An army, great in space, may offer opposition in a brief span of time. One man, brief in space, must spread his opposition across a period of many years’. Riot Games, with its galactic user base, its top Twitch rankings, its live world championships, and above all, its



▲ We don’t know what’s going on here, but it has something to do with *League of Legends* and makes us feel very, very old.

whackloads of cash, can afford to nitroglycerin the internet with one huge event. Indies poolle along for years, picking up followers and building hype each day, generally trying not to die.

However, there’s yin to this yang. Triple-A’s weakness is that it’s hard to humanise spectacle. It’s shiny and awesome and memorable, but it’s not going to play in soft-focus on Christmas Eve when you’re curled up with your loved ones thinking about *what really matters*. Rather, humans have evolved to respond to people. Even God is portrayed as a benevolent old man because he’s more impactful that way. And this is where indies shine. We can say ‘Hi, I’m Lottie, and here’s my game’ and generate a different sort of interest than big shiny Riot can. If you’re watching the indie space, this is why there’s increasing interest in open development, showing your work, live coding, being all over the Twitters with a personal account... We can’t go head-to-head with the big dogs. But we can be Frodo, the likeable schmuck sneakin’ round the back while triple-A’s Uruk-hai stomp off someplace else.

I love *Pop/Stars*. I love its size, its pageantry, its chutzpah. But Riot has to computer-generate humanity, selling human skins to human people because they’re too big to connect with anyone directly. I’ll never be able to pull off a *Pop/Stars*, and my face isn’t half as symmetrical as K/DA’s. But it is, at least, a real one. 🐼

Toolbox

The art, theory, and production of video games

28. Design Principle

Yars' Revenge and the importance of design goals

30. CityCraft

The abstract models that will give your virtual cities structure

32. Source Code

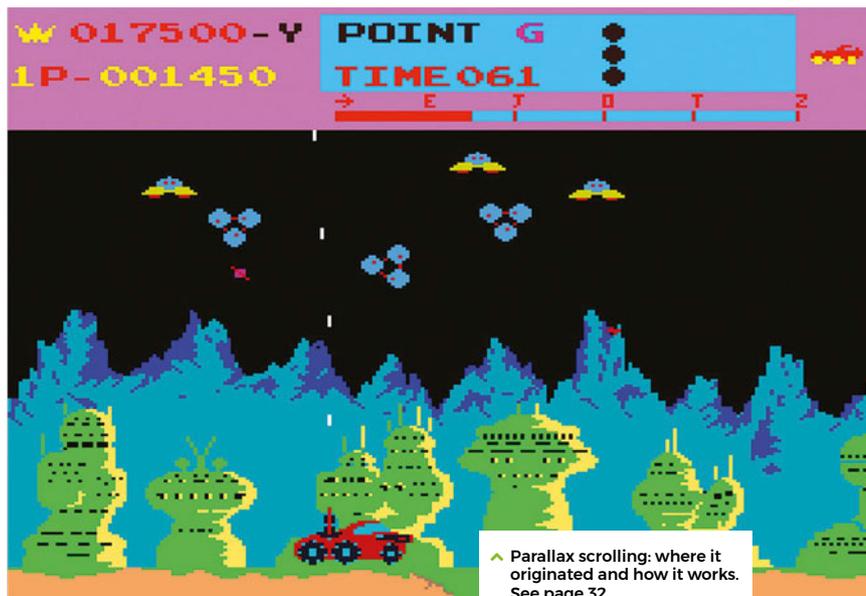
How parallax scrolling adds the illusion of depth to a 2D game

34. Build a first-person shooter in Unity

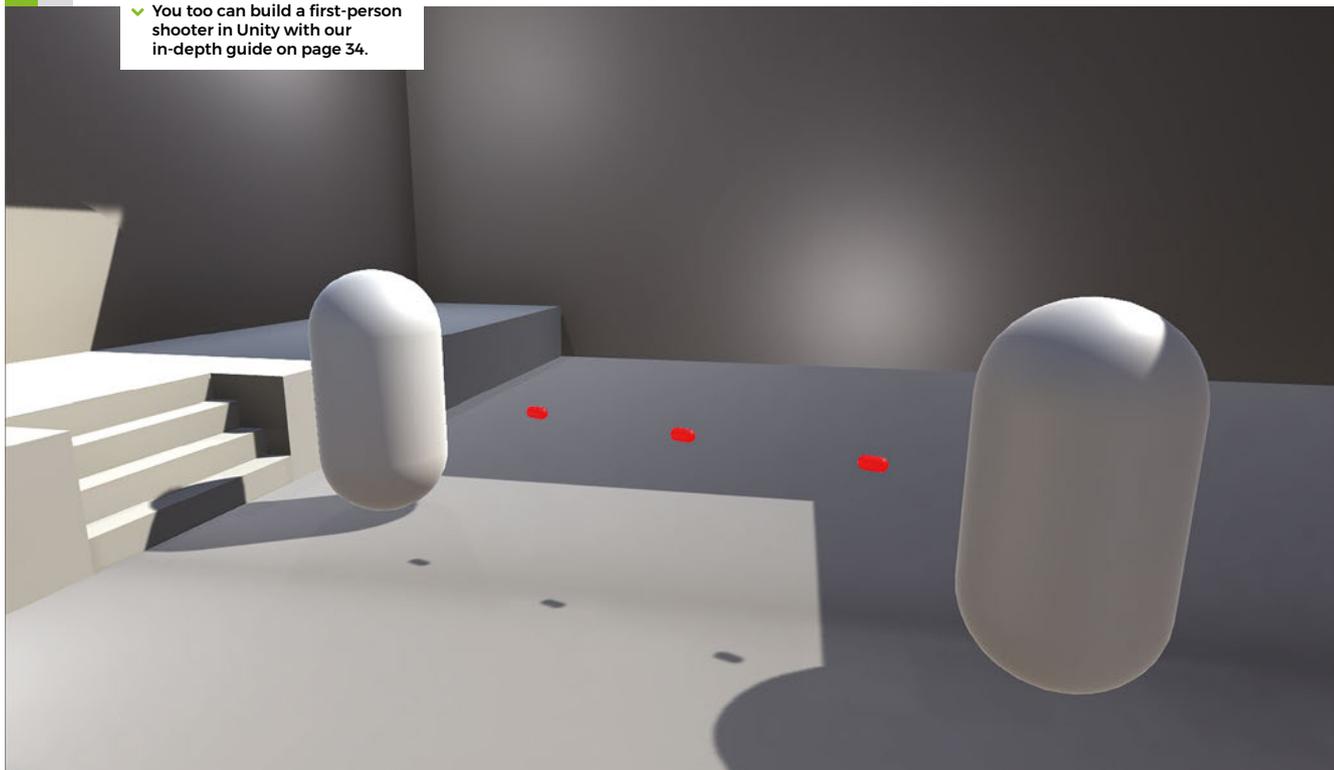
A step-by-step guide to making your own simple FPS

42. Directory

Find training courses and collaborators



✓ You too can build a first-person shooter in Unity with our in-depth guide on page 34.



The principles of game design

What the making of the Atari 2600 classic *Yars' Revenge* can teach us about the importance of design goals



AUTHOR
HOWARD SCOTT WARSHAW

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

My first game for Atari ultimately became what is now known as *Yars' Revenge* – and the story of how it started is a good way of illustrating some fundamental design principles.

Originally, *Yars' Revenge* was assigned as a coin-op conversion of a Cinematronics arcade game called *Star Castle* – a 1980 shooter that involved shooting an enemy cannon protected by three rotating, circular shields. With its sharp vector graphics and fast-moving enemies, however, *Star Castle* wasn't exactly the ideal fit for the Atari 2600's chunky sprites and meagre CPU. Clearly, a simple port of the arcade game wouldn't be possible, so I had to think carefully about how I could adapt some of its core mechanics.

The key to design is knowing what you're trying to do. With *Star Castle*, my design goals were:

- 1 To make a splash:** I want my debut to be a contribution, establishing my reputation as a game maker.
- 2 To create a sensory experience:** I want it to be distinctive – an eye- and ear-catching extravaganza that cannot be ignored.
- 3 To break new ground:** I don't want to iterate on existing material, I want to create something fresh and innovative.

Was I asking a lot? Sure, but why aim low? This is what I'll strive for and the results will fall where they may. Now, let's look at *Star Castle*

as a candidate and see how it fares with these three goals.

IMPLEMENTATION

1 Splash? *Star Castle* is a decent game with some interesting mechanics. Vector graphics are easy on powerful coin-op hardware, but miserable to recreate on the 2600. I could see the particulars of this game would be a nightmare. It was clear to me from the outset: this game would suck on the 2600. Would it be a contribution? More likely a charity case.

2 Sensory experience? *Star Castle's* on-screen motion is interesting, but the focal point is always the centre of the screen. Black and white vector graphics are not known for stunning visuals. In fact, all the colour in the game comes from a plastic overlay. Not the eye candy I'm looking for. The sounds feel a bit limited and somewhat monotonous. *Star Castle* is kinetic but not dynamic. I want to do better.

3 New ground? Let's face it, when you're doing a knock-off, innovation is not the thrust of the work. It didn't take me long to realise that coin-op conversions are the opposite of what I wanted to do.

My primary implementation goal was to create an action game I would enjoy playing, and *Star*

▼ *Yars' Revenge* is regarded as a classic, partly because it was carefully designed around the Atari 2600's limitations.



▼ Although *Yars' Revenge* looks entirely different, many of *Star Castle's* shooter concepts still remain.





The bounds of originality

Yars' Revenge came to be known as Atari's biggest-selling original game. Was it original? Clearly, it was assigned as a coin-op conversion, which is the antithesis of original. This begs the question: at what point is any game (or any creative endeavour) original? If you look to copyrights and legal perspectives, it's not so much the ideas or concepts, it's all about the execution. Two versions of the same game can yield very different play experiences. So, what are we evaluating? Is it the game or the player's experience? These are all useful things to think about when you're devising a game of your own.

▼ *Star Castle*, the 1980 shooter that formed the basis for *Yars' Revenge*.



▲ *Yars' Revenge's* working title was originally *Time Freeze* – a reference to a game idea that ultimately went unused.

Castle wasn't going to do it. Therefore, my first task was changing my assignment, creating room for me to do something different. As a veteran of a few days, I went to my manager Dennis and told him straight up that this game would suck on the 2600. However, I felt I could take some of the basic mechanics and tweak them into something that would play much better on our console. Fortunately, Dennis was receptive to the idea and said, "Go for it!" (this would never have happened two years later in the Age of Marketing, but that's another column entirely).

ADAPTATION

So, if not a direct translation, what would I do? Well, as we covered in my last column, talent borrows, genius steals.

Where does *Star Castle* shine? The basic mechanic of having to break through obstacles to expose the target is nice, and the fact the target fires back with a major weapon is also good. Constant danger creating the need for constant motion is an essential action game criterion in my eyes.

On the other hand, line drawing is nightmarish on the 2600, and the central visual focus gets tiring. How shall I change it? I'll push the target to one side of the screen to create room for play, and change the shield from lines to bricks which are easier to display.

I'll add a high-power player weapon and put it on the opposite side for visual balance. And maybe put something shiny in the middle.

It was amusing to me that *Star Castle* was made by Cinematronics because I approached this development like movie-making. Being a huge movie buff, my design thinking for games is cinematic. I think of screen composition and how game action drives eye movement. If I can keep

your attention continually moving up, down, right, left, this will increase the intensity of the experience. With the drone forcing the player into vertical motion, and the player's weapon moving left-to-right and the target's weapon moving right-to-left, a compelling dynamic could be achieved. Next, I'll animate both graphics and colour, resulting in a sensually intense visual ballet! At least, that's the hope.

I also use movie economics. Sound is cheaper than visuals, especially when the sound effects

are cheap. I don't want the sounds to be just about action, I want sound to dictate mood. Most games use sound simply as feedback for

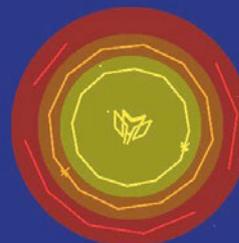
action, but I want to create a soundscape that incites mood and increases tension. I'll start with a background theme (fifties sci-fi lab electrodes and a faint buzzing), then build a layered hierarchical sound schema on top of that to accent game events and to foreshadow danger. You have only two sound channels on the 2600; I want to keep them both busy all the time.

I wasn't sure exactly what my game was going to be, but it wasn't going to be *Star Castle*. I knew I'd have to figure out a title and setting at some point, but there was plenty of time for that.

One thing was absolutely clear to me from the beginning: I needed this game to be great!

This was the mindset from which I began, and it was to evolve substantially. Next time, I'll share with you how this unfolded, as my plans and reality collided in the development system. ☹

"The key to design is knowing what you're trying to do"



CityCraft: structuring your video game city

A guide to using simple geographical models to give your cities structure



AUTHOR
KONSTANTINOS DIMOPOULOS

Konstantinos Dimopoulos is a game urbanist and author of the forthcoming *Virtual Cities* atlas.

High fantasy structures

You might think that real-world models wouldn't apply to a city set in a fantastical universe – these can feature truly weird, magical functions, and exotic, sometimes surreal land uses. But even here, the shapes, topologies, and general layouts of the three Ecology School models can still be useful. The city's core could house demons, while the surrounding rings could contain dragon pens or fortifications. Also, a dangerous magical area could easily be as undesirable to live near as, say, a power station or coal refinery.

W

hat we experience when walking through New York, visiting Beijing, exploring Rome, or living in Athens is not the result of a single planner's all-encompassing vision. Despite all of history's grand designs, despite all the Haussmanns and Le Corbusiers, urban spaces are the combined product of complex historical geographical processes, revolutions, competing urbanistic ideas, shifting ideologies, architectural trends, and a variety of evolving needs.

Rulers and architects have shaped important aspects of our cities, but they could never create or regulate the urban experience as a whole.

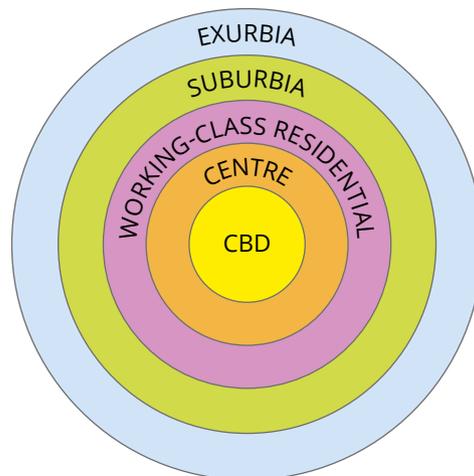
And yet game designers, with their comparatively tiny teams, are trying to achieve exactly this: simulate the joint effects of centuries of history and millions of people in order to create the illusion of believable cities.

THE URBAN PROCESS

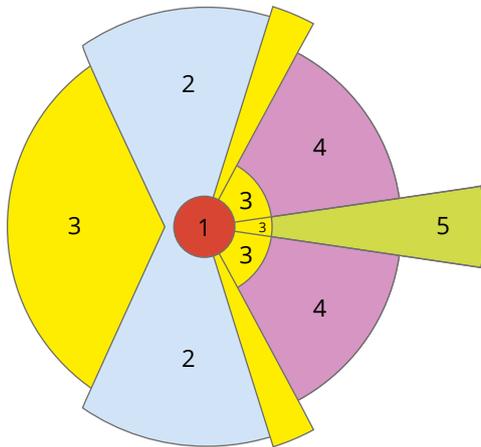
When starting to design your own virtual city, keeping things as simplified and abstract as possible is key. I suggest beginning by asking where, when, and how big your city will be. Placing one's urban setting in time and space is as important as determining its size, and will influence everything from the dominant architecture and core urban functions, to the layout and climate.

Unfortunately, that setting, important as it might be, will not help much when sketching the layout of your city. Even if you do manage to write a couple of paragraphs describing the city, its economy, and its people (which you should), or commission some evocative concept art, actually mapping it out can still be daunting.

Handily, though, all urban formations come with structure, and structure can be worked out with relative ease. Every contemporary city allocates space for housing, services, and production (and more besides); this is essentially its structure, and this can be described via a theoretical model. These models, created by scholars like Canadian urban sociologist Ernest Watson Burgess, provide a more easily understood, visual diagram of a city's layout and usage. Keep in mind, though, that structural models don't dictate all aspects of



> Figure 1: The concentric zone model is probably the most famous description of the modern city.



^ **Figure 3: The sectoral model was proposed by Hoyt back in 1939, and can still be very handy.**

city form; they do not, for instance, include data on street patterns, transportation networks, or architecture. All the same, these models provide elegant blueprints that can easily be modified and expanded, and can apply to any modern city, from the mid-19th century all the way to a cyberpunk future. So with that in mind, let's take a look.

THREE CLASSIC MODELS

Three widely taught Ecology School models can serve as fine starting points for the creation of game cities. The first one is Burgess's famous model of concentric zones, only slightly updated to fit modern cities better (see **Figure 1**). In this case, the central business district (CBD) sits in the middle, surrounded by the wider centre: the downtown. The ring around it includes working-class residences, themselves surrounded by the wider suburbia, and exurbia where most manufacturing and farming activities are located. The concentric model maximises the core's influence, while its density decreases from the centre outwards (see **Figure 2**). Oh, and do keep in mind that only when deeply abstracted can each individual zone only contain a single land use.

Hoyt's sectoral model (**Figure 3**) is usually both closer to geographic reality, and more adaptable to it. Area 1, in the centre, is the combined CBD and downtown, area 2 contains wholesale and light manufacturing, area 3 is low-class residential, area 4 middle-class residential, and area 5 dominated by the homes of the elite.

The more complex, more adaptable multiple nuclei model (see **Figure 4**) is a better fit for larger, metropolitan-scale cities, and acknowledges the existence and importance of

secondary or specialised centres. As for the labels' numbers, they stand for: 1. CBD, 2. Wholesale and Light Manufacturing, 3. Low-class Residential, 4. Medium-class Residential, 5. High-class Residential, 6. Heavy Manufacturing, 7. Outlying Business District, 8. Residential Suburb, and 9. Industrial Suburb.

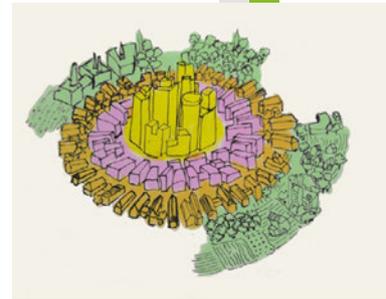
ENRICHING YOUR CITY

Picking one of these three models, and enriching it with such things as street patterns, specialised economies, and historic influences, should provide you with a more interesting, convincing civic foundation: something to guide your research, and provide you with a starting layout.

You should never hesitate to force mutations on models, or to introduce anomalies according to particular histories, eras, and natural geographies. For example, a town in the Roman Republic would be organised around a forum, a dark cathedral would be a suitable focal point

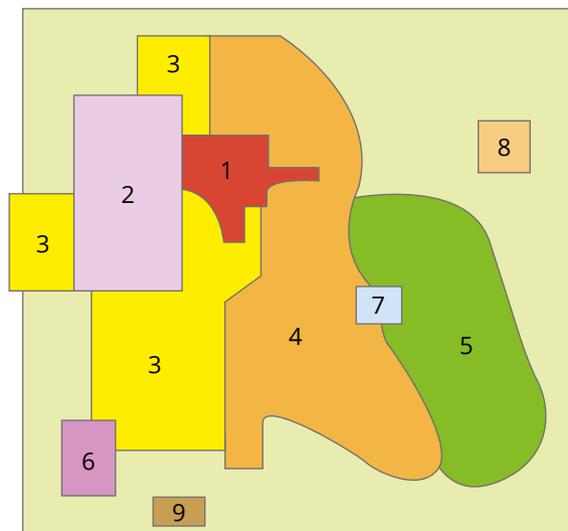
for a city inhabited by the undead, and a densely populated space station might go for a vertically differentiated structure.

A city by the sea, on the other hand, could adapt a semicircular concentric zone model; palaces are central in medieval urbanism; whereas a new landmark, canal, or great fire should inform your model accordingly. Considering all these things will make your city more interesting to explore and play in. 🌐

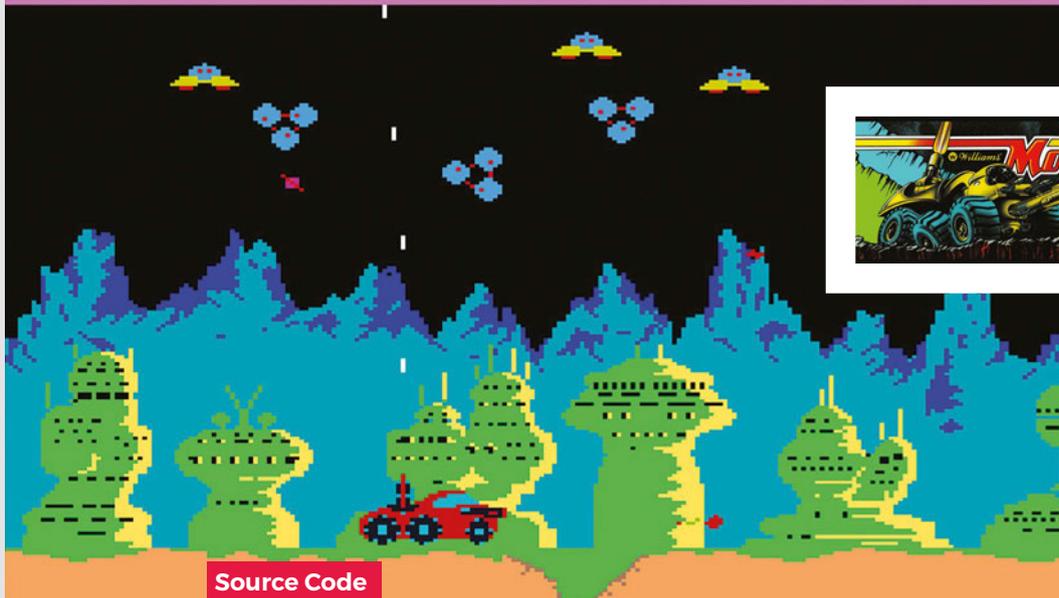


^ **Figure 2: Adding density data to land use zones can provide us with a glimpse of urban form and volume.**

"I suggest beginning by asking where, when, and how big your city will be"



< **Figure 4: The multiple nuclei model comes close to approaching actual metropolitan structures.**



Moon Patrol's illusion of depth

Classic games like Moon Patrol used parallax scrolling to create a sense of depth. Here's how you can recreate the effect for yourself



AUTHOR
RIK CROSS

Parallax scrolling is an effect that can be used to give the illusion of depth in a two-dimensional world.

Often in games, player motion is simulated by keeping the player in a fixed position on the screen, with platforms, backgrounds, enemies, and other objects moving across the screen, relative to the stationary player. This gives a 'window' on the game world, with the player taking centre stage.

A parallax scrolling effect is achieved by moving an object at a speed dependent on its distance from the player. Background objects (for example, clouds or distant mountains) move across the screen more slowly than objects in the foreground (such as platforms and trees). The quicker an object moves across the screen, the closer to the 'window' the object is perceived to be.

You can see this effect for yourself if you look out of a window and move your head horizontally from side to side (you'll just have to make sure that no one is watching you!). Nearby objects will move across your field of vision more quickly than objects further away.

"The idea predates computer games, and is borrowed from the 'multiplane camera', an invention used by Disney"

The effect was first used in computer games in the early 1980s, in titles such as *Moon Patrol* and *Jungle Hunt*, and in many games since. You'll notice the effect in classics such as *Super Mario World* and *Sonic the Hedgehog*, and it was used to create a creepy atmosphere in the more modern *Limbo*. It's not just used in platformers, either: nearby stars and distant galaxies

have been used to create the same parallax effect in countless space shooters.

In fact, the idea predates computer games, and is borrowed from the 'multiplane camera', an invention used by Disney and others to film cartoons in the 1930s. This achieved the effect by moving multiple layers of artwork past the camera at different speeds and distances.

To create the parallax scrolling effect for yourself, you'll first need to decide how to break up your background into a number of separate image layers that will move independently from each other.

In my example, I'll use three layers for far, mid, and near mountain ranges, but the technique will work for any number of layers – how many of them you'll need will depend on the level of detail you're aiming for.



PARALLAX SCROLLING in PYTHON

Here's a parallax scrolling effect that uses three layers to create a moving background. To get it running on your system, you'll first need to install Pygame Zero – you can find full instructions at wfmag.cc/XVzieD

```
# set screen width and height
WIDTH = 800
HEIGHT = 400

# create the back layer
layer_back = Actor('image_back')
layer_back.topleft = 0, 0
layer_back.speed = 1

#create the middle layer
layer_middle = Actor('image_middle')
layer_middle.topleft = 0, 0
layer_middle.speed = 3

#create the front layer
layer_front = Actor('image_front')
layer_front.topleft = 0, 0
layer_front.speed = 5
```

```
#add layers to list
layers = [layer_back, layer_middle, layer_front]

def update():
    for l in layers:
        # move each layer to the left
        l.left -= l.speed
        # if the layer has moved far enough to the left
        # then reset the layers position
        if l.right <= WIDTH:
            l.left = 0

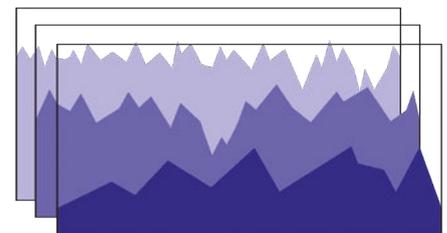
def draw():
    screen.clear()
    # draw all images in the image list
    for l in layers:
        l.draw()
```

Each layer should consist of a duplicated image, giving a layer that is exactly twice the width of the screen. The images should contain some transparency, so that the movement of other layers behind can be seen. To avoid a 'seam' in my mountain layers, I've also made sure that the mountain height is the same at the left and right side of the image. If you find creating the layer images difficult, you can either use the images I've created for my example, or go to an open-source media repository like Open Game Art (opengameart.org) and search for 'parallax' – you'll find lots of great ready-made layer sets to use.

Each layer is given its own speed, which will be used when updating its position. You can tinker with the numbers; all that's

important is that the speed of a layer is higher than the layers behind it. Layers start with their left edge at the left of the screen, and every frame, layers are moved to the left according to their speed. Once the right hand edge of a layer's image has reached the right edge of the screen, it is reset to its original starting position.

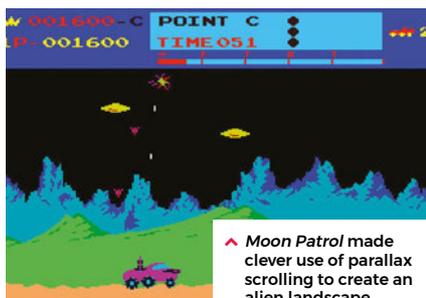
Layers are then drawn to the screen in order, from back to front. And that's all there is to it! Once you've achieved this basic effect, you can play with the code and experiment with different images to see what you can create. 🤖



^ The background is broken up into a number of separate layers that move independently from each other.



> Each double-width layer moves across the screen from right to left.



^ Moon Patrol made clever use of parallax scrolling to create an alien landscape.

Build a simple first-person shooter in Unity

Making a simple first-person shooter is easier than you think. From setting up Unity to creating a moving, firing character, here's how



AUTHOR
STUART FRASER

Stuart is a former designer and developer of high-profile games such as *RollerCoaster Tycoon 3*, and has also worked as a lecturer of games development.

Tools such as Unity and Unreal Engine have opened the way for just about anyone to make high-quality video games. In this guide, we're going to look at Unity, and how we can develop a basic first-person shooter. The great thing about the Unity engine is that it works well on multiple platforms, and the documentation is really clear, with a suite of easy-to-follow tutorials available for beginners and also experts.

GETTING HOLD OF UNITY

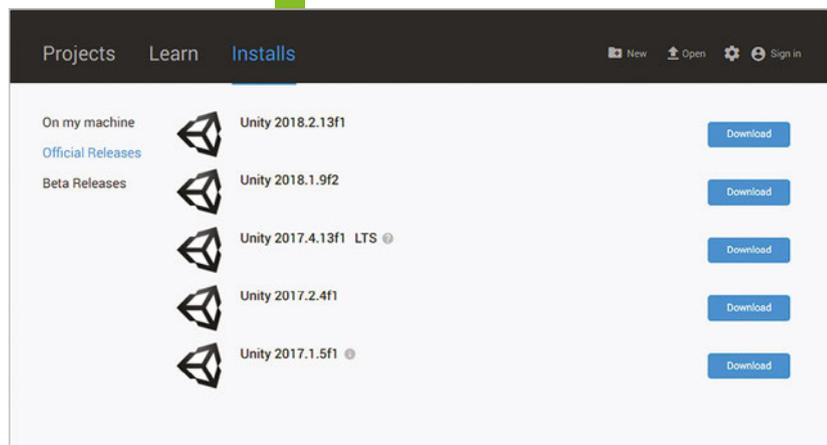
First, then, we need to get our hands on the Unity software itself and get it installed on your PC or Mac. While you can grab the latest version of Unity, I advise using Unity 2017.4.13, as this is the long-term stable branch. While we could use the latest version, we can't always guarantee that the latest features will work without issue. In this case, we have a long-term version that we know is tested thoroughly and is reasonably free of issues.

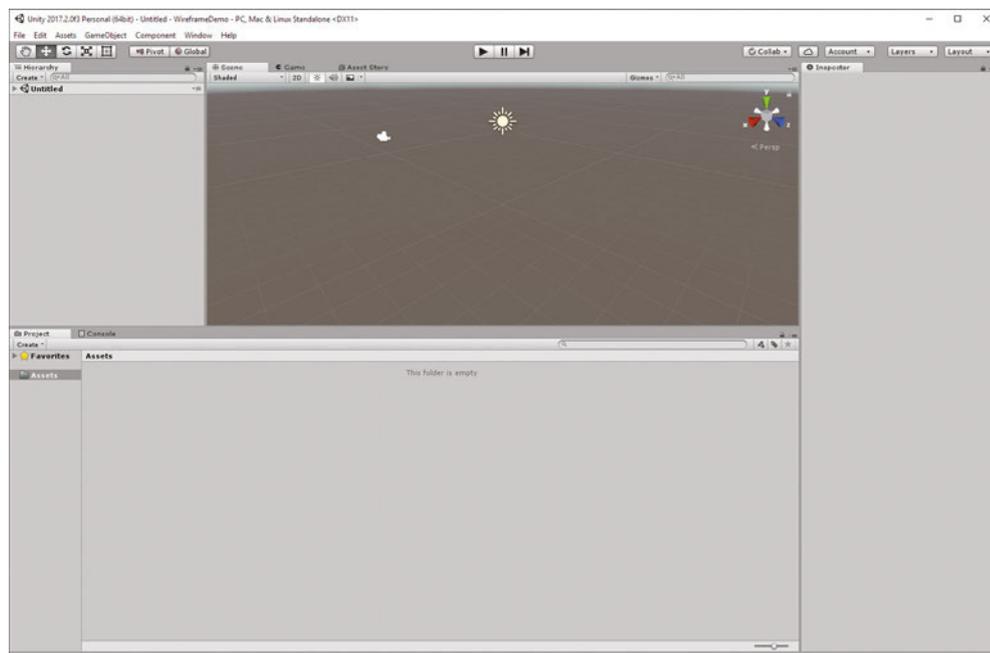
The maker has made it easy to get hold of any supported version of Unity by using a tool called Unity Hub. This is essentially a program launcher and still in beta, but it's simple, reliable, and will give you fast access to what you need. First, open up a web browser and navigate to the downloads page: wfmag.cc/fgqqSZ. Then you need to select Download Unity Hub, run the UnityHubSetup to continue, and select a suitable install location.

INSTALLING UNITY USING THE HUB

Once you open the Hub, you'll be presented with some choices in the launcher. They're pretty self-explanatory, with the headings Projects, Learn, and Installs. We'll touch on Projects

Under the official releases window we can see the various available versions of Unity. We may touch on some of the beta versions in the future.





◀ The default view from the Unity editor. We can easily customise the layout to suit your needs as a developer.

later, but this is where your games will live. As I mentioned earlier, the Learn section has some great resources. Finally, we'll choose the heading Installs, and on the left-hand side, click on Official Releases. Next, choose the Download option next to the version Unity 2017.4 LTS. Once it's complete, you'll be able to launch Unity and your projects from the Hub.

TAKE CONTROL OF THE EDITOR

Now you have the Hub, it's a simple process of selecting New from the top-right set of icons, and then giving your project a name and setting a location for it to live on your drive; by default, the version of Unity we downloaded is selected. We'll leave the Template drop-down set to 3D, and then complete the process by selecting Create Project. When you start up into the Unity Editor, you'll see a bunch of windows – this can be daunting to someone who hasn't used a games editor before. Again, Unity has some brilliant starter guides at wfmag.cc/zpaeSH, but

I'll take you through the process anyway.

The first thing we need to do is think about a typical first-person shooter: you generally can't see the character you're playing, but you have a viewport onto the world via a camera.

The first viewport is in the Scene tab, in the centre of the default layout. This is where we see the entire game world and build our

levels. Next to the Scene tab in the same window is the Game tab; click this and preview what our players will see when they start our game. Unity has given us a starting point of a camera and a light in its default startup scene. This provides us with the building blocks to get started – and if you hit the play button, you'll start the game running. You might be unimpressed with the result, though: there's no ability to move, and pressing the keyboard will have no effect. Press play again to exit out of the preview mode, and let's make the game do something more interesting.

SETTING UP OUR FIRST-PERSON CHARACTER

We want to add the ability to move our camera around and have it behave like a first-person character. The first step is to allow the camera to be controlled by the player. First, we need to do some setup in the Unity editor, then we'll be doing some very basic game

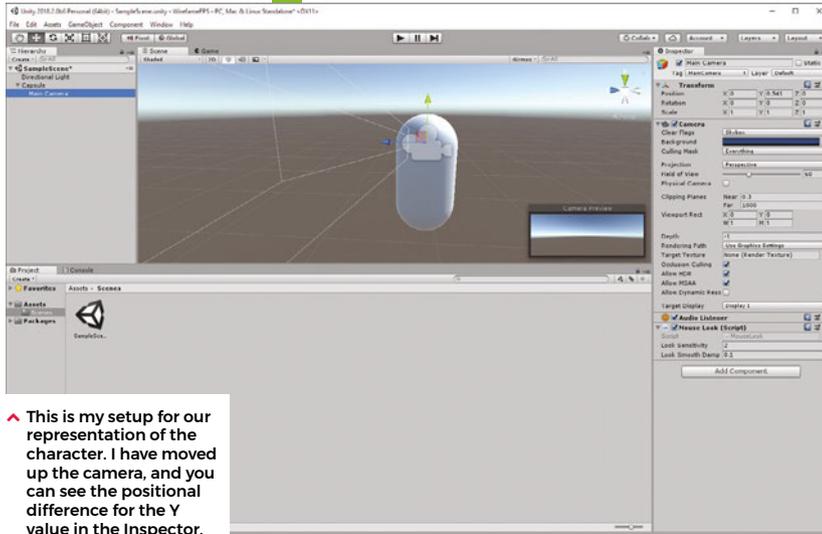
programming. This is sometimes referred to as game scripting, and allows rapid development of the gameplay mechanics in a programming language that is easy to understand. So let's get going and make our first-person character.

We'll use a capsule object to represent the player. From the toolbar, select GameObject > 3D Object > Capsule. While we won't see ➔

“The great thing about the Unity engine is that it works well on multiple platforms”

Toolbox

Build a simple first-person shooter in Unity



▲ This is my setup for our representation of the character. I have moved up the camera, and you can see the positional difference for the Y value in the Inspector.

TIP:

It's best practice to set your game world and game objects to the world origin – position 0,0,0 – when you import them into Unity or any other game engine. It makes things easier when implementing script logic or building your levels, as you aren't applying additional transforms to your objects.

this in the game, it gives us a reference point in our editor, and it will allow our player to collide with objects in the game world. Reset this capsule back to the origin, i.e. (0,0,0) – this can be achieved simply by using the Inspector, which is the panel on the right-hand side.

Look for the panel labelled Transform. If you don't see the values (0,0,0) for X,Y,Z, then select the cog icon to the right of the pane and select Reset Position. Now, select the Main Camera in the Hierarchy tab and set this back to the origin (0,0,0) using the above method. Finally, use the Hierarchy window and drag the Main Camera onto the Capsule game object. You should see that the camera is now parented to the capsule. In other words, the camera is attached to and placed below the capsule in the hierarchy.

We're going to do some more setup to the viewing position of our character. Select the camera that we parented and you'll see a camera preview in the lower right of the Scene viewport. You should also see a transform gizmo with three arrows (red, green, blue) extending out of the camera object. Select the arrow that's pointing up and coloured green. We need to use the left-mouse button and drag it up slightly so it's in the upper area of the capsule. Think about the capsule being your character: you want the camera to be its eyes. If you want to adjust your viewing angle within the Scene viewport, you can move the view angle by clicking and dragging the right mouse button within the viewport, and you can use **WASD** or arrow keys to move around.

ADDING THE BASE MOVEMENT

We're now going to add our script to move the character. Unity is mostly object driven, so we

essentially attach our scripts to the game objects that we want to affect. In this case, we need to select our capsule object in the editor. Go to your Inspector window on the right and click the Add Component button at the bottom. In the new pop-up window, scroll to the very bottom and select New Script. You need to set a name for your script – I suggest CharacterMovement – and then click the Create and Add button. This method is great, as it attaches the script to the object and saves it to your game project automatically.

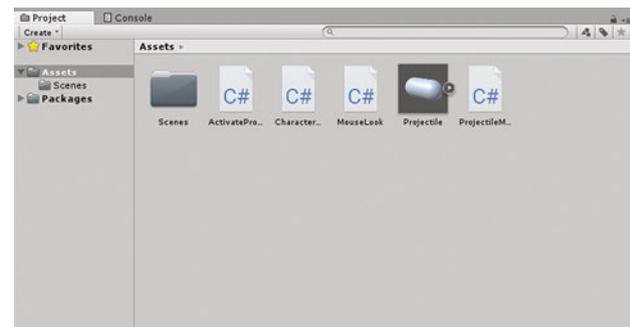
The only thing we have to do now is open the script. This can be opened from the bottom Project window by double-clicking. You'll be provided with some sort of scripting environment – this will be either MonoDevelop or Visual Studio. Now all we need to do is replace our code with the template script that Unity provides. Don't forget to save.

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

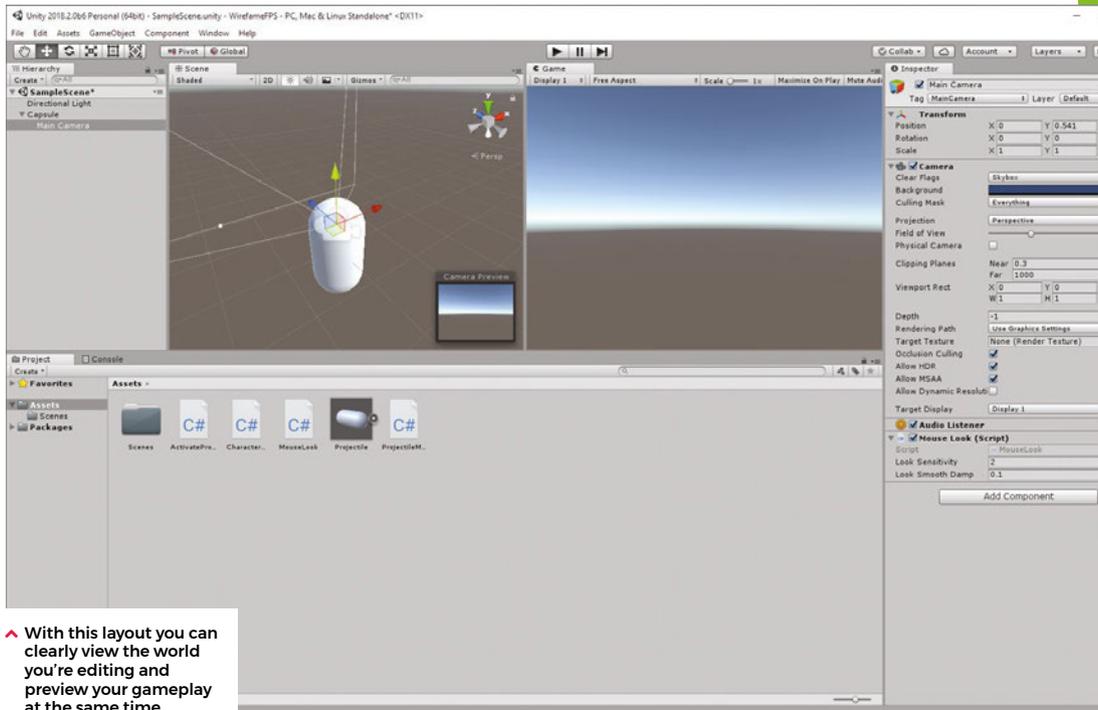
public class CharacterMovement : MonoBehaviour
{
    public float speed = 5;

    // Use this for initialization
    void Start()
    {
        Cursor.lockState = CursorLockMode.Locked;
    }

    // Update is called once per frame
```



▲ The project window will show all the assets you are using for your game and can be placed in your levels. This could be scripts, audio, textures, models, and much more.



^ With this layout you can clearly view the world you're editing and preview your gameplay at the same time.

```

void Update()
{
    float Horizontal = Input.GetAxis("Horizontal") * speed;
    float Vertical = Input.GetAxis("Vertical") * speed;
    Horizontal *= Time.deltaTime;
    Vertical *= Time.deltaTime;
    transform.Translate(Horizontal, 0, Vertical);

    if (Input.GetKeyDown("escape"))
        Cursor.lockState = CursorLockMode.None;
}

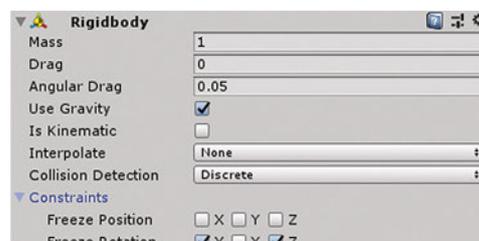
```

We should have our movement script ready to go. I'd suggest changing your viewport setting before pressing play so you can easily see the effect in your Scene view. Select the Game tab with your mouse, then drag the window, and it will unlock. Drag it towards the Inspector and it will expand, then release. You should now have two easy-to-access viewports for the Scene and the Game. Press play and use **WASD**, the arrow keys, or a controller to apply movement. While it's difficult to see in the Game viewport, it's easy to see the applied motion in the Scene viewport. Again, when you finish, you need to press the play button to exit this game preview.

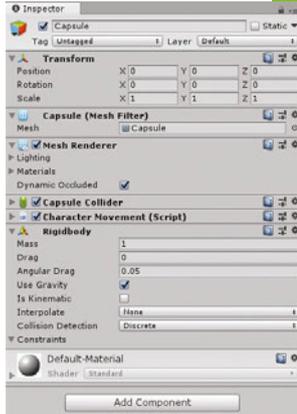
ADDING GRAVITY AND JUMP ABILITY

What we have is pretty limited, so we want to expand the range of motion and apply gravity limitations to our character. We're going to expand the code we created earlier, but we first need to expand our character so it understands that it has physics rules. First, we're going to add another component called a rigidbody – this is from a physics term that describes any solid object, and is used in the mathematical understanding of applying forces like velocity and acceleration. We'll select our capsule and then, in the Inspector, select Add Component. We can then use the search box at the very top of this window to search for Rigidbody.

I suggest adding a floor object to the scene, otherwise your character will fall forever. It also helps you to have a reference point when moving around your level. From the toolbar, select **GameObject > 3D Object > Plane**. ➔



< Components often have properties that you can override to change how they behave. In this case, we're stopping the object from rotating in an undesirable direction.



▲ You can view what components are on a game object in the Inspector. You can add multiple components to change the behaviour of these objects.

You can use the transform gizmo to move the object around; the arrows dictate the direction you'll move the object in. Move this under the player capsule. We're going to expand the code we have already, so we need to open up the same code editor we had before. Select the **CharacterMovement.cs** file and then simply replace the original example with the following code:

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

public class CharacterMovement : MonoBehaviour
{
    public float speed = 5;
    public float jumpPower = 4;
    Rigidbody rb;
    CapsuleCollider col;

    // Use this for initialization
    void Start()
    {
        Cursor.lockState = CursorLockMode.Locked;
        rb = GetComponent<Rigidbody>();
        col = GetComponent<CapsuleCollider>();
    }

    // Update is called once per frame
    void Update()
    {
        //Get the input value from the controllers
        float Horizontal = Input.GetAxis("Horizontal") * speed;
        float Vertical = Input.GetAxis("Vertical") * speed;
        Horizontal *= Time.deltaTime;
        Vertical *= Time.deltaTime;
        //Translate our character via our inputs.
        transform.Translate(Horizontal, 0, Vertical);

        if (isGrounded() && Input.GetButtonDown("Jump"))
        {
            //Add upward force to the rigid body when we press jump.
            rb.AddForce(Vector3.up * jumpPower, ForceMode.Impulse);
        }
    }
}
```

```
}

if (Input.GetKeyDown("escape"))
    Cursor.lockState = CursorLockMode.None;
}

private bool isGrounded()
{
    //Test that we are grounded by drawing an invisible line (raycast)
    //If this hits a solid object e.g. floor then we are grounded.
    return Physics.Raycast(transform.position, Vector3.down, col.bounds.extents.y + 0.1f);
}
```

Again, select to play your game and use the **SPACE** bar or buttons on the controller to try to jump as your character. You may notice some odd behaviour, as the character may topple over. We can fix this by selecting the capsule object, and in the options for the RigidBody, you'll see the word Constraints and a small down arrow. If you click this, it will expand and show Freeze Position and Freeze Rotation. Activate the checkboxes for X and Z for the Freeze Rotation only. If you play the game again, it should be impossible to make the character topple.

LOOKING AROUND THE ENVIRONMENT

One element of a first-person shooter is that you can look around the environment or level by using your mouse or the controller sticks to effectively move the head of the character. We are going to add another script to our player, but this time to our camera and not the capsule. First, we need to select the camera, and repeat the process of selecting Add Component and then making a new script. I would name the script **MouseLook**, and then you can open the code editor and add the code provided.

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

public class MouseLook : MonoBehaviour
{
    private GameObject player;
```

TIP:

Make sure you are careful with your script file name and the name of the public class in your C# script. These have to be identical to each other, else the script will fail to compile.

```

private float minClamp = -45;
private float maxClamp = 45;
[HideInInspector]
public Vector2 rotation;
private Vector2 currentLookRot;
private Vector2 rotationV = new Vector2(0,
0);
public float lookSensitivity = 2;
public float lookSmoothDamp = 0.1f;

void Start()
{
    //Access the player GameObject.
    player = transform.parent.gameObject;
}

// Update is called once per frame
void Update()
{
    //Player input from the mouse
    rotation.y += Input.GetAxis("Mouse Y")
* lookSensitivity;
    //Limit ability look up and down.
    rotation.y = Mathf.Clamp(rotation.y,
minClamp, maxClamp);
    //Rotate the character around based on
the mouse X position.
    player.transform.

```

```

RotateAround(transform.position, Vector3.up,
Input.GetAxis("Mouse X") * lookSensitivity);
    //Smooth the current Y rotation for
looking up and down.
    currentLookRot.y = Mathf.
SmoothDamp(currentLookRot.y, rotation.y, ref
rotationV.y, lookSmoothDamp);
    //Update the camera X rotation based
on the values generated.
    transform.localEulerAngles = new
Vector3(-currentLookRot.y, 0, 0);
}

```

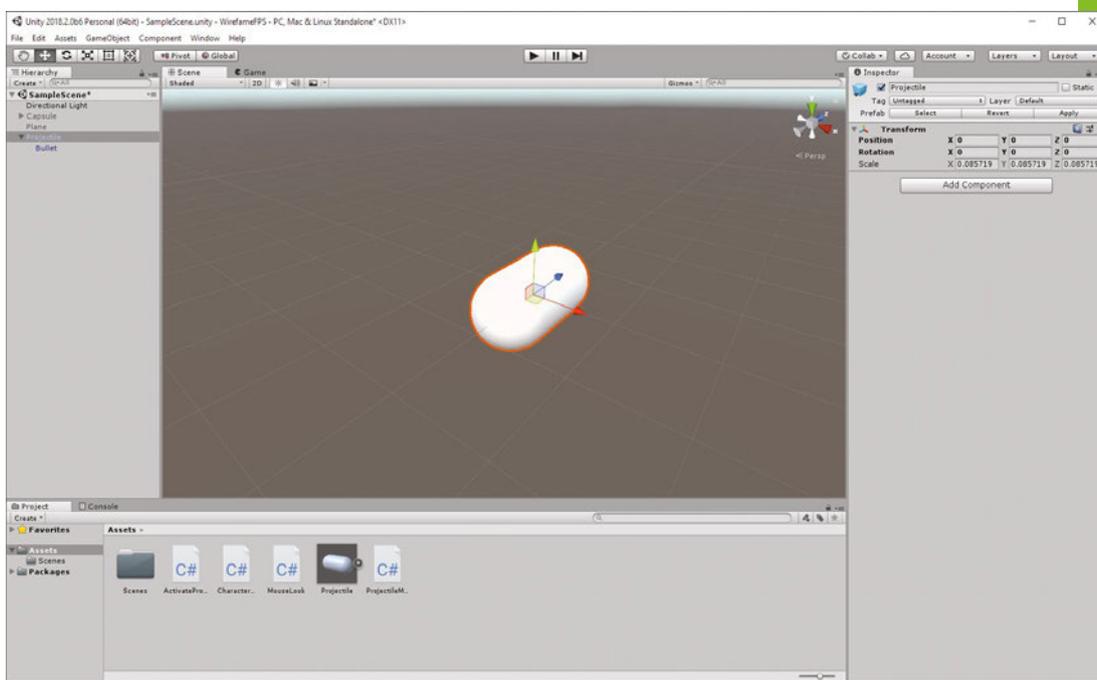
You may notice that this works as a mouse look only, and that it doesn't work on a controller. Unfortunately, creating controls like these are beyond the scope for this tutorial, but you can always try to implement this yourself if it's something you want to add.

MAKING A PROJECTILE WEAPON

Of course, this wouldn't be much of a first-person shooter without being able to shoot. There are many ways to implement a weapon mechanic, and a game like *Quake Champions* has several ways to handle weapon types, from ➔

TIP:

You can build out your level by using the basic 3D objects that are available in Unity, or you can explore the Unity Store. The Unity Store has both free and paid items that can help expand the quality of your projects, such as meshes, effects, sounds, and even scripts. You can access the store at any time by selecting Window > General > Asset Store within the taskbar or by using the key combination **CTRL+9**.



◀ I have parented my second capsule object, which has been rotated to an empty game object. You'll notice that it has a default rotation and no other components attached.

projectile to beam weapons. We're going to add ours in the most basic way and have a projectile that we can fire in the direction we face.

First, we need to select the capsule. This time, we're going to use a shortcut to parent a new object to it. We now need to right-click on Capsule in the hierarchy and select Create Empty. We're going to use this to spawn our projectile on. This is a quite common practice in games, where you have an empty or dummy object for positioning or spawning. In the Scene viewport, select the gizmo and highlight the blue arrow and move

it forward of your capsule object. I would also advise renaming the game object by typing in the text box at the top of the Inspector with this object highlighted. I would suggest calling this 'Weapon' or something meaningful.

We also need an object to use as a projectile, so we'll quickly create this from two objects: another dummy and another capsule. On the taskbar, select GameObject > Create Empty. We're going to use this to help us make sure the bullet moves in the correct direction even when we rotate the capsule object. I'd rename this object to 'Projectile' so we can find it later on. Now we need to select the new object in the hierarchy, then right-click and choose 3D Object > Capsule.

I'd rename this 'Bullet' in the Inspector window. Now we want to make some size and rotation adjustments. Just underneath the left-hand side of the upper-left taskbar, you have six icons. You'll notice that one is highlighted (a cross shape of arrows); this means you're using the position tool. Select the icon to the right of this; a tooltip will tell you it's the rotate tool. You'll notice the gizmo is now a sphere with overlapped circles. Select the red circle, and then drag until the capsule faces forward or the Inspector shows approximately 90 degrees in the rotation box labelled X.

Finally, we need to rescale the dummy object. First, you need to select this dummy object in the hierarchy. To the right of the rotation icon is the scale icon; select this, and the gizmo will change again. This time the gizmo will be similar to transform, but instead of arrows, we have cubes. We're going to select the bigger white cube at the centre of the gizmo; this will let us rescale all directions at the same time. If you

drag downwards you'll see that the capsule object will shrink. We need to make this about one-tenth the scale of the character.

The final step for the projectile is to centre it to the world. In the Inspector you can select the cog on the Transform for your dummy object, and then select Reset Position.

FIRING THE WEAPON

We are going to add two new scripts. The good news is that these are really simple and comprise only a few lines of code. We need one to make the projectile spawn, and one to make the projectile move in a constant direction. We'll deal with the movement first, as we should still be highlighting the projectile object. As with the other scripts, we add them via Add Component and then giving the script a name. I recommend the name of ProjectileMovement for your script.

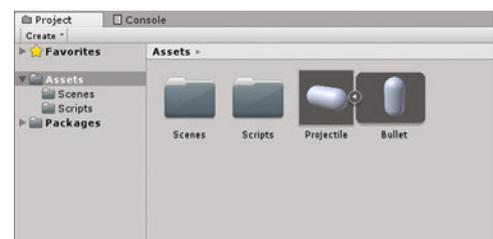
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ProjectileMovement :
MonoBehaviour {
    public float speed = 10f;

    // Update is called once per frame
    void Update () {
        transform.Translate(Vector3.forward*
speed * Time.deltaTime);
    }
}
```

Before we work on the next script, we're actually going to remove this initial projectile game object from the scene. However, we still

✓ Our prefab in our project window is ready to go. Prefabs are extremely useful as they can be used multiple times in the project and will always have the same behaviours we set.



TIP:

You can better visualise an empty object by adding an icon to it. In the top-left of the Inspector is a small cube with a down arrow. Select this and you can set one of several icon looks. This won't show in the game, but it's great as a guideline when moving objects in the viewport.



^ OK, so it isn't exactly *Crysis*, but it's still a solid basis for your own shooter in Unity.

want to use it in the project, so we're going to make a special type of object called a prefab. We can make this a prefab by selecting it in the Hierarchy and dragging it into the Project window at the bottom of the screen. You then need to make sure you've now selected the version in the Hierarchy and hit delete.

Now for the second half of this scripting adventure: select the Weapon game object and then, as before, you can select Add Component and follow the flow. Name your script `ActivateProjectile` and then open the script to add the final part of the code:

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

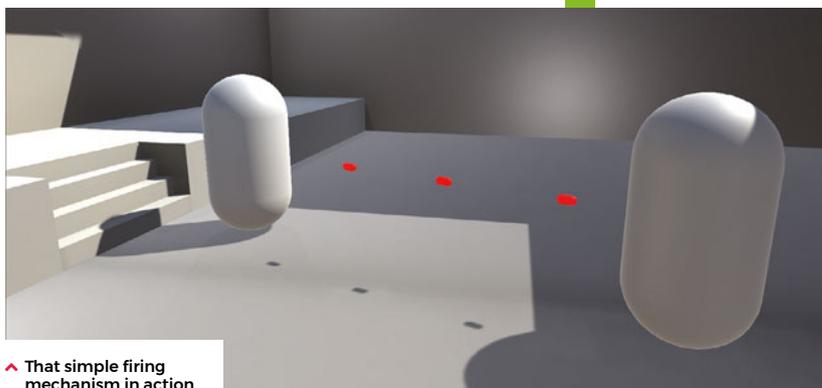
public class ActivateProjectile :
MonoBehaviour {
    public GameObject projectile;

    // Update is called once per frame
    void Update () {
        if (Input.GetButtonDown("Fire1"))
        {
            var clone = Instantiate(projectile,
gameObject.transform.position, gameObject.
transform.rotation);
            //Destroy after 2 seconds to stop
clutter.
            Destroy(clone, 5.0f);
        }
    }
}
```

We have one last thing to do. We have our prefab, but our spawn script doesn't know to use it yet. In our Weapon object we can see the script we added, called `ActivateProjectile`. You'll also see the words `Projectile` and a slot that says 'empty'. We need to add our prefab here by selecting the circle to the right of the slot. A window will appear, which has `Assets` and `Scene` tabs. Select the `Assets` tab and you'll see our `Projectile` prefab.

Make sure you've selected it, then press play to see the result. You should be able to fire using the left mouse button, and see multiple projectiles when you click. You may find you're getting pushed back by the bullet; the easiest way to fix this is simply to move the Weapon object further forward.

Congratulations: you've created the mechanical basis for your very own first-person shooter. 🎮



^ That simple firing mechanism in action.

WOULD LIKE TO MEET...

Are you a programmer looking for a level designer? Do you have an amazing game but no music to go with it? Then let Wireframe help you find a collaborator. Send details of your project and who you're looking for to wfmag.cc/hello

▼ Get your characters running and gunning in ex-LucasArts animator Chad Robert Morgan's Unreal 4 course.



DISCLAIMER:

We advise readers to create single-use email addresses when searching for collaborators via the Wireframe Directory. Raspberry Pi Trading Ltd accepts no responsibility for the interactions that might arise from the use of the directory.

Directory

Graphics & animation: online courses

From 3D modelling to animation and pixel art, here are eight online courses to sharpen your art skills

▶ Pixel art and animation

From making an animated character run and jump, to creating background objects, this course is designed to enhance your pixel art skills.

wfmag.cc/UwcfgV

▶ The art of modelling and animation in 3DS Max

Weighing in at just over 18 hours, this in-depth course takes you from naming and manipulating objects to lighting, texturing, and rendering.

wfmag.cc/RSaHIL

▶ Blender 3D modelling for beginners

The free-to-use Blender is an accessible (and most importantly, affordable) way for budding game designers to get into 3D modelling. This course provides a handy primer.

wfmag.cc/SdGMdy

▶ Rigging and animating in Blender

If you're already familiar with the basics of Blender, how about a guide to getting your characters rigged and animated?

wfmag.cc/xKLBmq

▶ Character animation in Photoshop

This course takes you through a range of projects to get your creativity flowing: moving typography, character animation principles, and even a brief guide to rotoscoping.

wfmag.cc/onxZGQ

▶ Animating characters in Unreal Engine 4

LucasArts veteran Chad Robert Morgan shows you how to import character models into UE4 and, you guessed it, how to animate them.

wfmag.cc/ojEzLd

▶ Maya: 12 principles of animation for games

Give your characters more life and personality by following these animation principles, introduced by the character animator from *Ghostbusters: The Game*.

wfmag.cc/gninXa

▶ Working with pixel art tilesets

Whether you're working on a retro platformer or a top-down RPG, you'll need to create pixel tilesets that can fit seamlessly together. This five-hour tutorial takes you through the basics.

wfmag.cc/OONDmH

Wireframe

LIFTING THE LID ON VIDEO GAMES

Download the app

Out now for smartphones & tablets



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



Fast and furious

How Unreal Engine allowed Barry Skellern to create his speedy indie game Thrunt XL – without writing a line of code

At the recent Play Expo gaming festival in Blackpool, a young boy was playing the indie game *Thrunt XL*. He was enjoying himself and doing quite well: guiding his agile lander around enemy bullets, and swerving to avoid colliding with deadly walls. Then, the kid suddenly burst into tears. Was the game too hard? Did it contain an unwelcome surprise? Concern began to spread across the face of the game's creator, Barry Skellern, and he crouched down to see what was wrong with the poor boy. It turned out that he'd become so engrossed that he'd lost sight of his family.

"I assure you, my game doesn't typically make children cry," Skellern eventually tells us, breathing a sigh of relief.

For Skellern, the primary emotions *Thrunt XL* tends to stir up are determination and frustration. "It's a difficult game by design," he explains, "but no one gave up on it, even the kids, and they all got a feeling that they wanted to finish the trickiest sections, persisting far longer than I thought they would."

Featuring a thrusting, rotating spaceship that needs to be carefully piloted from A to B while remaining in



AUTHOR
DAVID CROOKES

"This time around, he isn't getting his hands dirty with actual code, preferring instead to use a visual scripting system"

one piece, *Thrunt XL* is directly inspired by retro gems such as Superior Software's *Thrust* for the BBC Micro – itself heavily influenced by Atari's *Gravitar* from 1982.

"I call it a super-tricky thrust-'em-up," Skellern says, "and the control system is immediately recognisable to anyone who's ever played *Thrust* or, my particular favourite, *Grav II*, which was a shareware game for the Atari ST. The big difference with my game – other than the use of the 3D – is that there's no combat. It's the *Super Meat Boy* of *Thrust* games, with that just-one-

more-go appeal. You're only ever punished for the mistakes you make, which drives you on to do better next time."

Skellern, who works under the company name

of Phantom Aspect, can make bold decisions because his approach to development is similarly old-school. He creates his games entirely alone, in much the same way scores of creators did when the gaming industry was in its formative years; he harks back to a childhood of tinkering with BASIC and creating simple 2D line drawings on the ZX Spectrum. Only this time around, he isn't getting his hands dirty with actual code, preferring instead to use a visual scripting system. ➔





"I resisted the likes of Unity and Unreal Engine for a long time, because I've always been interested in programming from an empty screen," he says. "Using an engine felt like a cheat or a shortcut – it didn't feel right to download an engine, throw things in, and see it work. But in the end, I bit the bullet and gave it a go when I had a couple of days off. I've found they allowed me to approach games holistically rather than being forced to concentrate on the details of one discipline."

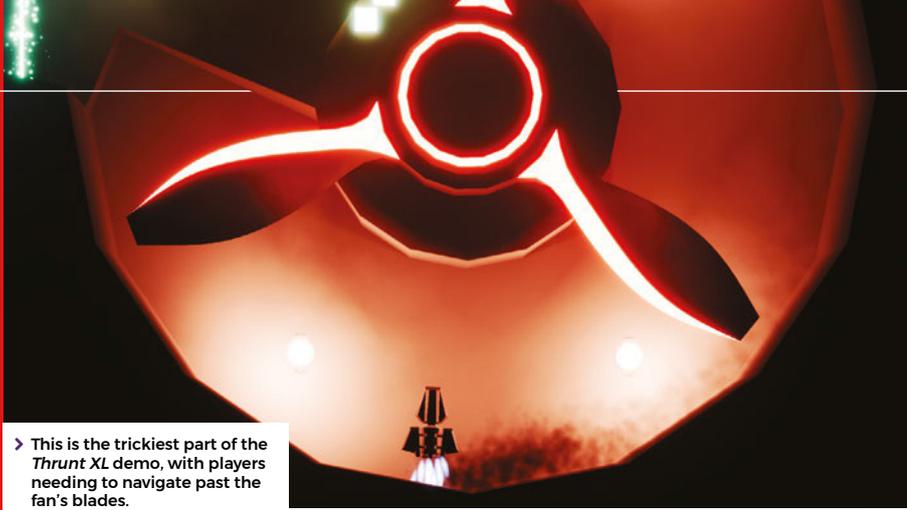
CHOICES, CHOICES

Once Skellern decided to experiment with an existing game engine, he looked around to see which suited him best. Initially, he preferred Unity over Unreal Engine – because, he says, "the documentation for it was way better in 2015, though that's less the case now."

After creating a few rudimentary games in Unity over the course of an afternoon, Skellern ended up creating *Thrust*, the predecessor to *Thrust XL*. "When I realised I could develop decent games so quickly using Unity, I was converted. I also realised engines don't prevent you from doing your own thing – you just have to work within their frameworks."

The original *Thrust* was created partly as a learning experience and as a way for Skellern to introduce himself to a more ambitious gaming project. As an against-the-clock racing game boasting 13 monochrome levels with the same design DNA as the sequel, it provided Skellern with a welcome release from the rigours of his day job as a programmer of medical technology.

"My initial idea was to spend a month making it and then see where it got to, but I was quite enjoying playing my own game so I fleshed it out and made a bunch more levels," he says. Sticking to his experimental approach, Skellern steered away from fancy graphics "because it was more about learning the engine and techniques of gameplay."



► This is the trickiest part of the *Thrust XL* demo, with players needing to navigate past the fan's blades.



► Phantom Aspect has deliberately removed the player's ability to fire back, making *Thrust XL* about control and agility.



▼ Barry Skellern, also known as Phantom Aspect.



It's why *Thrust* was entirely black and white and didn't use any advanced engine features.

Inevitably, work, family, and friends have affected the amount of time Skellern has available for development. As a consequence, game creation has meant some late nights. But, rather than stick with Unity, Skellern rose to the challenge of learning Unreal Engine properly, and he ended up using the latter to create *Thrust XL*. "Unity had given me great knowledge about how games fit together," he says. "I still like Unity and will no doubt use it again, but I wanted to broaden my horizons."

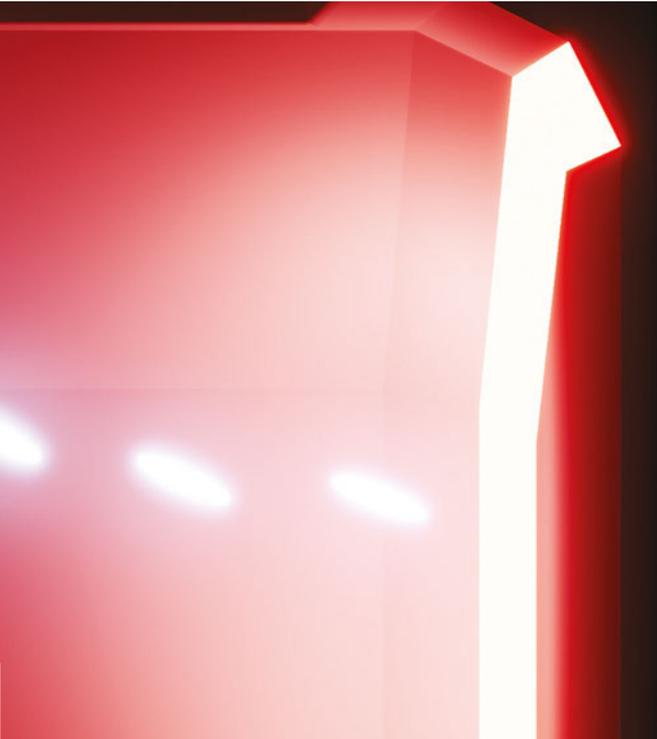
AN UNREAL EXPERIENCE

Unreal Engine has been around since 1998, making its debut with Epic Games' first-person-shooter, *Unreal*. It has since proven to be a versatile engine, used to create titles within many other genres such as fighting games and stealth. Just like Unity, it provides standard components for codeless games creation, which reduces barriers to entry. But any engine requires some knowledge of programming logic, and Unreal Engine arguably has a steeper learning curve than Unity.

Another learning curve

Another way of opening and closing doors within a game such as *Thrust XL* is to define animations in a 3D package: in such a case Unreal Engine would trigger the animation following a collision event attached to a switch. It's also possible to create a curve – as you can see in the image at the top of page 49. In this case, an animation curve is overlaid on the level in the background and it can be used to determine, for example, how you want the vertical position of an obstacle to change over time.

"It's an animation that you're authoring directly within the engine rather than from your 3D software," says Skellern. "You could trigger that to run over a certain period of time and have a door open with a specifically authored motion."



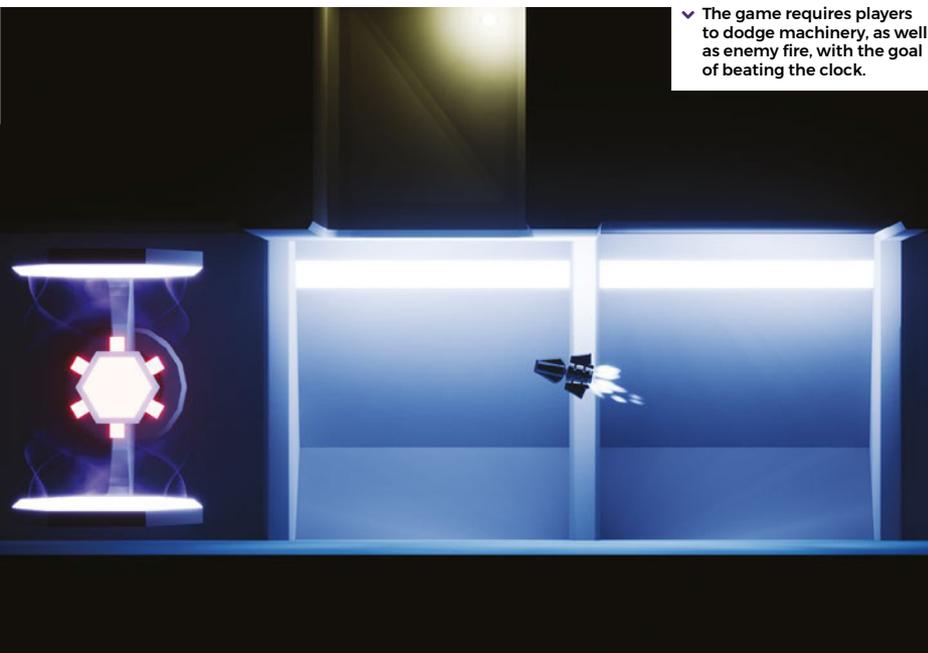
“Unreal Engine offers a visual scripting system called Blueprint which allows for quick development of game logic without using C++,” Skellern explains. “So while there’s no programming involved, the same logical process goes into designing the games. Users benefit from not having the problem of forgetting to close brackets or omitting a semicolon at the end of a line or something like that. There’s no issue with a compiler giving you a big spew of error messages, which can be a big time sink for new coders. Blueprint minimises problems, even though the typical maths things that you have to do when coding a game are no different whether you type them in or drag boxes.”

“Users benefit from not having the problem of forgetting to close brackets or omitting a semicolon at the end of a line”

Blueprint has allowed Skellern to make swift progress. “You can get 80 percent of the way with a game’s code using just 20 percent of the boxes, so you very quickly learn the

common patterns and the common nodes you need and how to wire them up,” he says. Development entails creating a rough outline of each level using the indie version of the 3D modelling package MODO Indie, defining the graphical assets and the collision shapes (“there’s an element of designing the gameplay within MODO,” Skellern says). These are then imported into Unreal Engine, where they have their movement and custom behaviour defined in Blueprints.

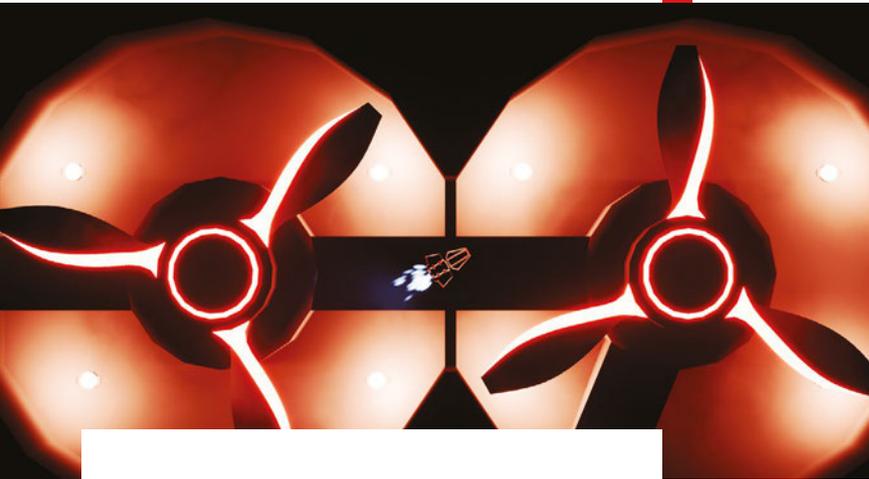
▼ The game requires players to dodge machinery, as well as enemy fire, with the goal of beating the clock.



“Once the block-out of the level feels right and it’s OK to fly through without any of the doors in place and so on, I add the doors and wire them up in Blueprint so that the game knows how to respond when certain events occur,” he says. “If a switch is triggered, the game has to know which doors ➔

▼ Constructing Blueprints to update individual objects is quite simple, but understanding how they interact can be a challenge.





open, which doors close, and which obstacles to switch on and off. Since I work on my own, I can use my own instinct, and I can work out what does and doesn't feel right."

Solo work certainly has its advantages. "I have full control over everything that goes into the game and I don't have to have meetings with myself to decide what I can do," he says.

As for disadvantages – well, the sheer wealth of skills a solo developer needs is an obvious challenge. "You have to be a generalist across lots of different skills,

and obviously, you can't do everything perfectly," Skellern says. "For *Thrust XL*, I've done everything from the graphics, level design, and UI to the music and sound effects, and I know that some of these are less polished compared to other things. I think my UI skills are one of my weaker points, definitely, and if I was in a team then I would probably prefer to have someone else do that."

Working alone, he adds, also makes it easy for him to lose sight of how hard the game is. "I'm doing my best to address accessibility so that players aren't prevented from enjoying the game due to avoidable exclusionary features," he says. "Obviously, this is a challenge in a game that's designed to be difficult, but it's one that I take seriously, and I always welcome suggestions."

TIME IS PRECIOUS

It's also been important for Skellern to understand what can be achieved within a reasonable time frame, since he's acutely aware the project could consume many more months than he would like. To that end, he began work on *Thrust XL* in November 2017, and he doesn't envisage finishing it until the summer of 2019. But he was keen not to overcomplicate things by adding, for example, lots of textures in the game.

▼ Your ship in *Thrust XL* is agile but fragile – touch any of the walls as you fly around and it's curtains.

"When I was showing the game, players would come back even after they'd finished the demo, determined to identify a racing line and complete it in a faster time"

"There are two reasons for this," he begins. "First, it would take me a lot longer because the UV mapping and the texturing that goes into the 3D modelling process is a lot more involved. Second, I don't think the quality would be up to scratch – I think I would fail to achieve the look I like if everything was texture-mapped.

"So in experimenting with different looks and different styles, I realised that if I get rid of all of the textures, it makes everything a lot quicker to work with. There's also very little colour in the game materials – pretty much all of the walls are grey and the colour actually comes from saturated lighting, relying on Unreal Engine's light-baking and post-process effects to find a characteristic but efficient style."

It's a great style, achieved through technological trickery rather than artistry, although there is art in getting the lights to blend and the colours to be cast the way Skellern likes them. Ultimately, keeping the graphics low-poly and low-textured is an efficiency-driven look, but it's one he's happy with. "There's a bit of pragmatism in that choice, which

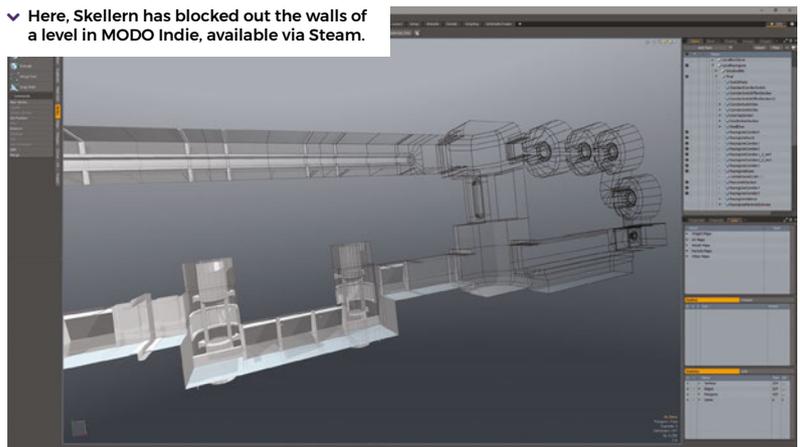
I think is necessary as a one-man indie producer," he says. "You have to slash something or the game will never come out."

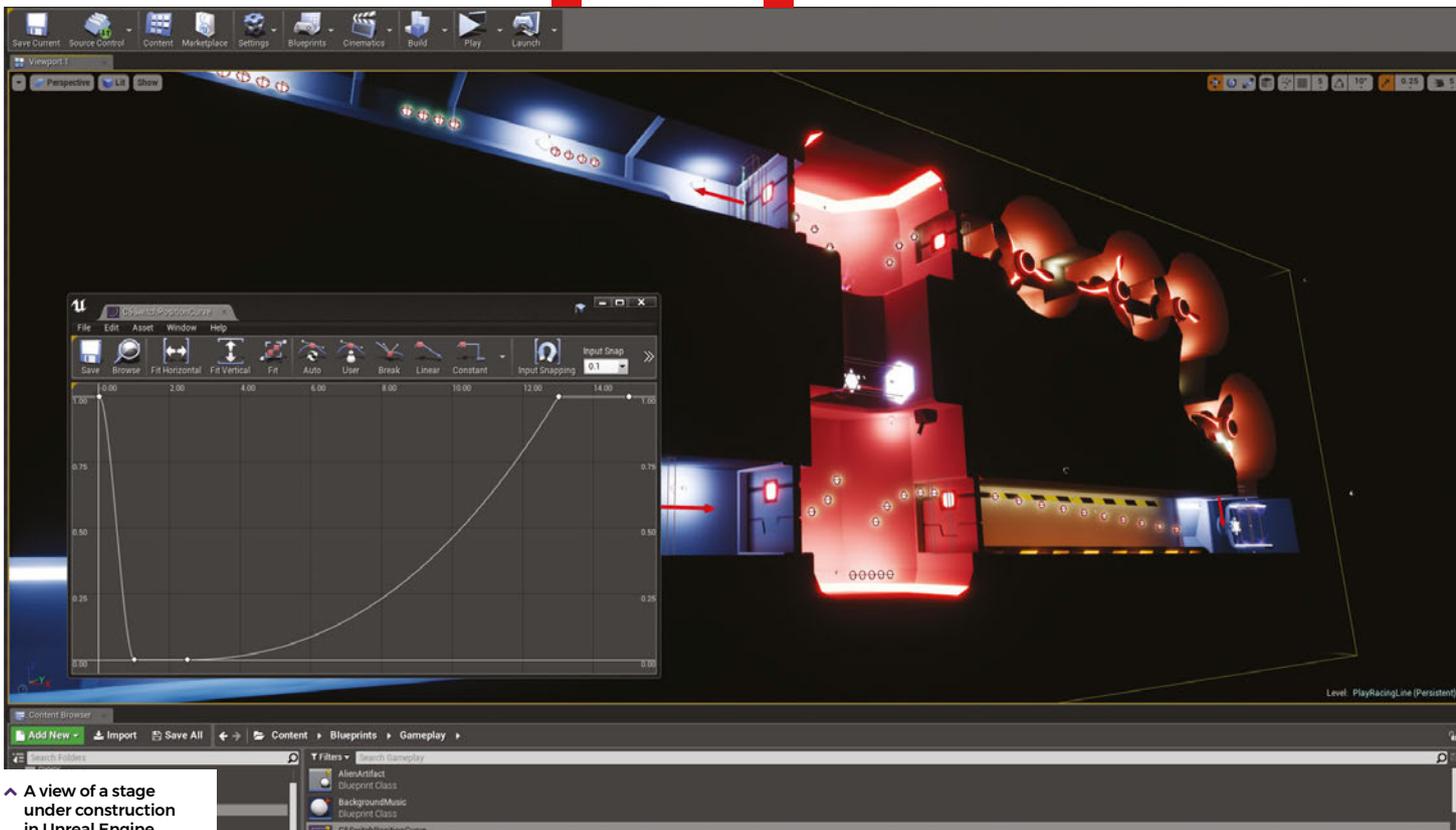
Some cuts, however, have made the gameplay

better: Skellern decided to omit weapons from the game, making it entirely about progressing from one level to the next without being hit.

"I felt games like *Gravitar* and *Thrust* were sluggish because you could never thrust the way you wanted to shoot, or shoot the way you wanted to thrust

▼ Here, Skellern has blocked out the walls of a level in *MOD0 Indie*, available via Steam.





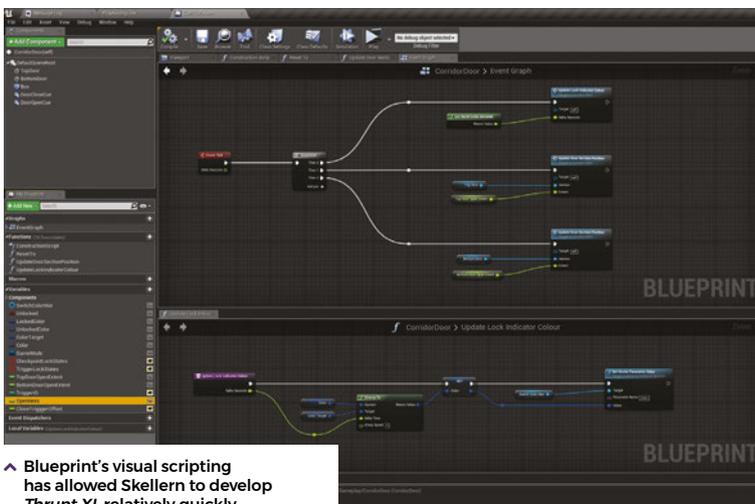
▲ A view of a stage under construction in Unreal Engine.

Code with nodes

Unreal Engine uses a visual scripting system called Blueprint, which revolves around a set of nodes. These are connected by wires which show the path of a given action.

In the example shown below, the Event Graph at the top of the screen is concerned with updating the state of a door within *Thrust XL*. So if a player flies in front of a switch, three pins (0, 1 and 2) are called upon, one after another.

Firstly, this updates the lock indicator colour from red to green, as defined in the bottom window. It then moves to the next two pins, individually causing the top and bottom pieces of the door to slide back.



▲ Blueprint's visual scripting has allowed Skellern to develop *Thrust XL* relatively quickly.

– you always had to face the way you want to fire, and that was rarely the way you wanted to go,” he explains. “By taking the weapon off the ship and explicitly saying you can’t fight back and you can only race past using your agility, the game feels freer and more smooth. You’re not getting bogged down in corners and having to work out how you can fly across the path of a gun turret while simultaneously shooting it.”

Skellern is now aiming to nail down 20 levels by the summer of 2019 while working on ways for players to upgrade their ship. “There’s not a lot to do in terms of the functionality of the game,” he reveals. “Now it’s mostly content creation, but that takes time.”

To drive people to keep playing, Skellern has added target times and medals – rewards that positively encourage speed-running. “When I was showing the game, players would come back even after they’d finished the demo, determined to identify a racing line and complete it in a faster time,” he says.

Thanks to the use of a game engine and some canny design choices, though, players will be able to race through *Thrust XL* very soon. “Unreal Engine has helped me speed up development,” Skellern concludes. “It’s a very good tool.”



Developer Profile

SNK

40 years old, and it only died once: a modern tale of (limited) success

Celebrating 40 years in the game this year, SNK is a legend in the world of development, publishing, and hardware. It had a hand in some of the finest arcade series of all time, crafted some spectacular home hardware, and has a bevy of lesser known cult hits under its belt to boot. At the same



time, it's unlikely most of you have ever owned any SNK hardware, or any of its initial, non-re-released, games.

The Shin Nihon Kikaku ('New Japan Project') Corporation was founded on 22 July 1978, though an earlier non-gaming entity had existed since 1973. Headed by Eikichi Kawasaki, the firm made strides into the then-nascent coin-op market, bringing its own brand of vertical and side-scrolling shooters to the masses.

It wasn't until the introduction of new hardware – the Neo Geo Multi-Video System (MVS) – in 1990 that the company really hit its stride. Focusing on pure arcade quality, SNK would develop or publish some true greats of the coin-op's golden age. *The King of Fighters*, *Metal Slug*, *Samurai Shodown*, *Fatal Fury*, *Shock Troopers*, and more came out of SNK's doors through the nineties – franchises still played and loved to this day.

SNK brought the arcade home with its AES – a console version of the MVS – and even tried its hand at releasing a CD version of the Neo Geo, as well as a couple of handhelds for good measure. Each had its good points, but mostly these off-shoots were consigned to the scrapheap not too long after launch. With success limited to arcades and relatively low sales of home units, things didn't actually go so well for SNK financially, and an uncaring parent company soon pushed the studio to the brink.

SNK died in 2001. Following bankruptcy proceedings, the company ceased operation on 22 October 2001. Eight days later, its remaining intellectual property

► The NeoGeo Mini marked SNK's 40th anniversary.



was bought by a company formed a couple of months prior, called Playmore – headed up by one Eikichi Kawasaki. SNK's founder had seen the writing on the wall and left his baby, though with the greater good in mind. Within a couple of years, things were mostly back to normal, the company didn't reside under an uncaring parent company, and SNK Playmore would quickly crack on with attempts to re-establish itself as an arcade force to be reckoned with.

Despite some success, though, it wasn't to be: SNK's output slowed to a trickle, support for the Neo Geo was finally ended in 2007 (with its final game releasing in 2004), and a succession of arcade releases failed to find much footing. Revisiting its most famous console in 2011 with the

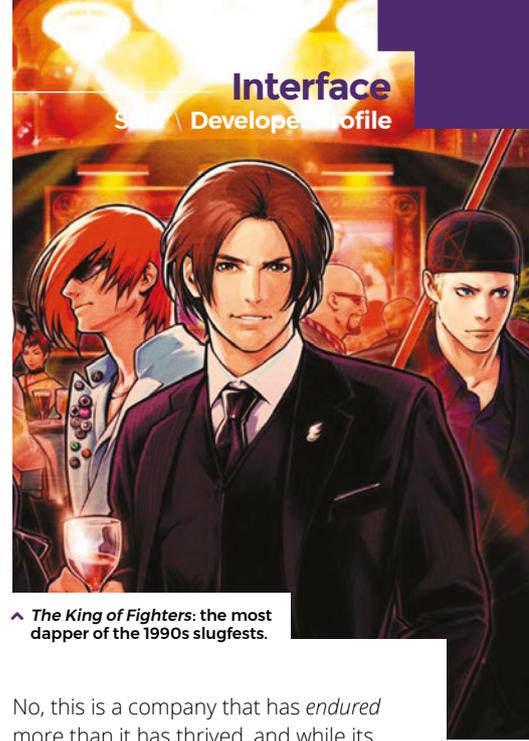
Neo Geo X, SNK hoped to offer players an affordable, impressive piece of portable hardware on which to enjoy the best of its back catalogue. But it didn't work out, and the handheld was abandoned by SNK after less than a year on sale.

Even with the best of intentions, SNK's output began to wither, relying increasingly on retro compilations for contemporary consoles and slot machines using the company's licences. It took a £49.5m takeover by a Chinese consortium, prompting Kawasaki's retirement, to kick things up a notch once more. Soon after the takeover was complete in 2016, *The King of Fighters XIV*

“While its glory days do seem to be firmly in the rear-view mirror, by no means is it a forgotten publisher”

was announced and a new SNK – shorn of its ‘Playmore’ – emerged. One focused on using its assets to make games, not slot machines, and one attempting to become the force it once was in the world of gaming.

It would be a lie to say ‘40 years on and SNK goes from strength to strength’ or any similar arrangement of words.



▲ *The King of Fighters*: the most dapper of the 1990s slugfests.

No, this is a company that has *endured* more than it has thrived, and while its glory days do seem to be firmly in the rear-view mirror, by no means is it a forgotten publisher. *The King of Fighters* sees sequels created, while the ACA Neo Geo collection from Hamster is bringing a range of classics from the arcade and ludicrously expensive home console to a new generation of players – and this time for about a tenner, instead of a grand.

All the same, while SNK's tagline might be ‘the future is now’, that future seems resolutely in lockstep with the 40 years previous. Whatever comes next, it'll be fun to see what the company comes up with to delight – and disappoint – its fans over the next four decades. 🍷

▼ SNK founder Eikichi Kawasaki, here promoting *Lee Trevino's Fighting Golf*. [Image courtesy of Brian Hargrove]

Homeward bound

SNK's Neo Geo Advanced Entertainment System (AES) console launched in 1990 with a focus on renting units to the likes of hotels and bars. SNK soon realised, though, that regular people were interested in picking up the console – even with it costing £500 (a staggering £980 adjusted for inflation). With no major third-party publishers on board and a limited selection of games, somehow, some way, the AES managed to survive multiple generations against competitors, as well as the bankruptcy of the company that made it.

From its release in 1990, the AES managed a 17-year run before SNK finally discontinued support for the machine in 2007. Again a resounding success for endurance rather than a straightforward triumph, the AES nevertheless goes down in history as one of gaming's most resilient machines.



「ファイティングゴルフ」を間に、SNKの川崎社長とリー・トレビノ

Neo Geo's 10 Overlooked Games

Some Neo Geo greats that tend to get the short end of the stick



01

Top Hunter: Roddy & Cathy

MVS / AES / Neo Geo CD – 1994

Largely flying under the radar on release, *Roddy & Cathy* has enjoyed something of a reappraisal in recent years. Its recent re-release through Hamster's ACA Neo Geo project has opened this twee little two-plane platformer up to a whole new generation, and it's definitely worth a go. One of the Neo Geo's most overlooked classics.



02

Shock Troopers

MVS – 1997

Riffing on Capcom's *Commando* and *Mercs*, *Shock Troopers* nonetheless ended up being a standout of the Neo Geo era – overlooked in large part thanks to its lack of an AES home release. It might be pretty basic – it's run and gun, after all – and the sequel might be better, but this is still worth a go.



03

Waku Waku 7

MVS / AES / Saturn – 1996

A frantic cartoon you control, masquerading as a one-on-one fighter, *Waku Waku 7* actually acted as a sequel to the much more staid *Galaxy Fight*. It has more in common with pinball than it does *Fatal Fury*, and it's easy to see why it slipped through the cracks. That doesn't mean it should have, mind.



04

Stakes Winner

MVS / AES / Neo Geo CD – 1995

In 1995 we wanted to race Ridges, not horses. Which, given the power of hindsight, was a bit of a faux pas on our collective part. While *Ridge Racer* has aged rather poorly, *Stakes Winner's* mix of skilful horse racing and (very mild) equine management has stood the test admirably.



05

Twinkle Star Sprites

MVS / AES / Neo Geo CD / Saturn – 1996

A competitive vertical shoot-em-up, *Twinkle Star Sprites* tasks players with avoiding or destroying waves of enemies – as you'd expect. But on destroying them, a fireball attack is launched at the other player, which they then have to dodge or deflect. It's a recipe for raucous fun, and a genuinely distinctive feature for the genre.

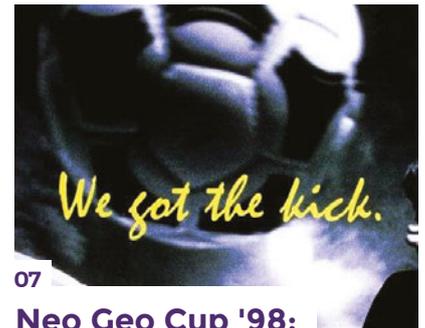


06

King of the Monsters 2 – The Next Thing

MVS / AES / Neo Geo CD – 1992

The first game acted as a Kaiju wrestling simulator, while this oft-overlooked sequel mixes it up a bit, removing pins and the ropes while throwing in some side-scrolling sections, more moves, and a gorgeous look to things. Also, you can throw the Eiffel Tower at your opponent. Games are art.



07

Neo Geo Cup '98: The Road to the Victory

MVS / AES – 1998

A FIFA World Cup 1998 spin-off of *Super Sidekicks 3*, *The Road to the Victory* is understandably overlooked in the most part. Who remembers those limited release tie-ins anyway? It's a shame, though, as this is arguably the best football game on the Neo Geo – made even more shame-y as it's unlikely to ever see a re-release.



08

Cyber-Lip

MVS / AES / Neo Geo CD – 1990

Magician Lord is the more remembered launch title for the Neo Geo, but *Cyber-Lip* definitely deserves a look. It pales in comparison to the *Contra* series it so liberally takes inspiration from, and you can't shoot diagonally for some reason, but nothing stops this from being fine fun for an hour or so.



09

Pulstar

MVS / AES / Neo Geo CD – 1995

Tough, riddled with power-ups, involving face-offs against giant enemy bosses, being in a wee little spaceship – there's a lot about *Pulstar* that screams 'R-Type clone'. And that's fair. Still, the rendered graphics, resulting in a *Donkey Kong Country* in space look, make *Pulstar* an overlooked standout in the Neo Geo's library.



10

Magical Drop 3

MVS / AES / Saturn / PSone – 1997

Cross *Bust-a-Move* with *Columns* and you end up with *Magical Drop*, the third game of which was easily the best in the series, and a fine puzzler to this day. Why was it overlooked? Well, puzzle games rarely have the impact of shooters and brawlers, and that's what the Neo Geo was known for.

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

Hitman 2

Info

GENRE
Stealth

FORMAT
PS4 (tested) /
XBO / PC

DEVELOPER
IO Interactive

PUBLISHER
Warner Bros.
Interactive
Entertainment

PRICE
£49.99

RELEASE
Out now

REVIEWED BY
Adam Barnes



▲ Part of the fun of *Hitman* is finding a ridiculous costume to brutally murder in, though some of them – like this one – are tied to particular assassinations.

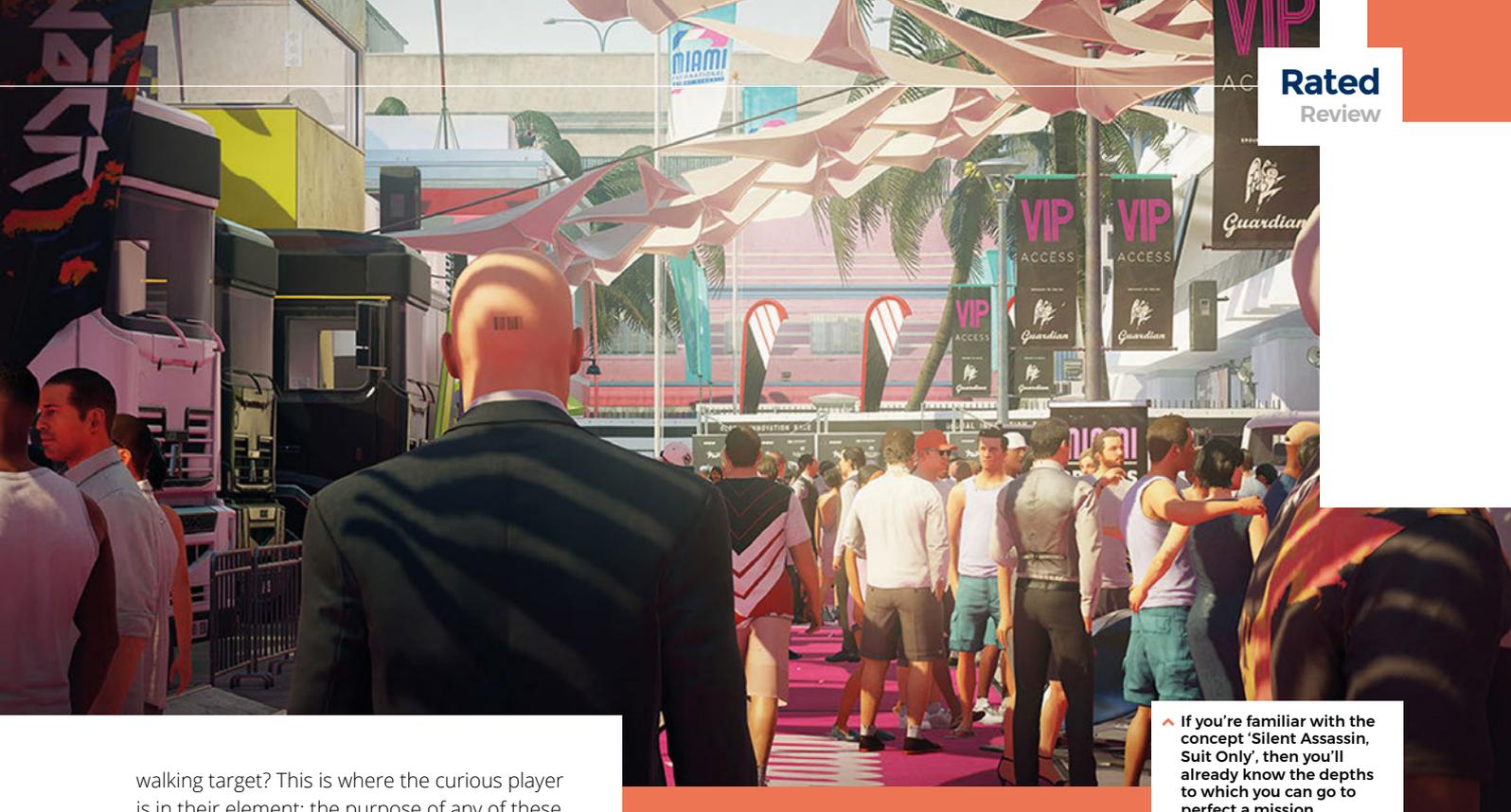
Invisible assassins aren't supposed to leave a mark

Everyone agreed, in the end, that an episodic approach to *Hitman* worked in its favour. This is a series that thrives on the player building an intimate understanding of each mission, all in pursuit of assassination perfection. The last outing enabled that, doling out the levels in piecemeal to essentially enforce this unique 'play, kill, repeat, perfect' mechanic. That's changed: *Hitman 2* is a package of missions as the games ever were, so now the onus is squarely on you to get better at the game.

And in fact, that is almost required to get any enjoyment out of it. *Hitman 2* hasn't altered the template in any way: you're still playing Agent 47, you're still set assassination targets to find and kill, and you're still given a multitude of means of doing so. Each time you first load into a stage, it's more about playing for reconnaissance than the hunt, to doss around these sprawling sandboxes as you learn what is and isn't possible. Much of the same smart system elements of the last game have carried over and continue to nudge the uncreative player towards particular actions. Huge boosts in points for certain kills give pointers for things to look out for, while interacting with the environment in particular ways rewards the curious and prompts further ways to secure the mark. It's a franchise built for

the inquisitive, and that's perhaps more integral than ever with this entry.

This is primarily because of the scale of the maps. This time around, they're surprisingly vast environments; ones that might not seem so grand at first but quickly open up to be varied, often interconnected puzzles. The first attempt will be a case of fumbling around the different locales, mentally piecing together the individual mechanisms available to give you a picture of what you have at your disposal and the sorts of obstacles you'll face. It builds up a good degree of intrigue for experimentation, wherein lies the ultimate critique cop-out: *Hitman 2* is for a very particular sort of gamer and nothing has – or perhaps ever will – been done for it to appeal to anyone outside of that. It's a game that compels players who like to mess around with systems, whose own innate curiosity drives the game – rather than a set of objective markers and run-to waypoints. Getting into the pit crew area of the first 'proper' stage of Miami is incredibly easy, for example, and through no effort at all you'll soon find your target, Sierra Knox, bursting onto the racetrack in her soon-to-be-a-fireball vehicle. But what if you learn that there's a set of three gas pipes that can be tinkered with beside the racetrack? Or that the trophy cup itself can be poisoned? Or how, if left unprompted, Knox will actually finish the race anyway and become a



▲ If you're familiar with the concept 'Silent Assassin, Suit Only', then you'll already know the depths to which you can go to perfect a mission.

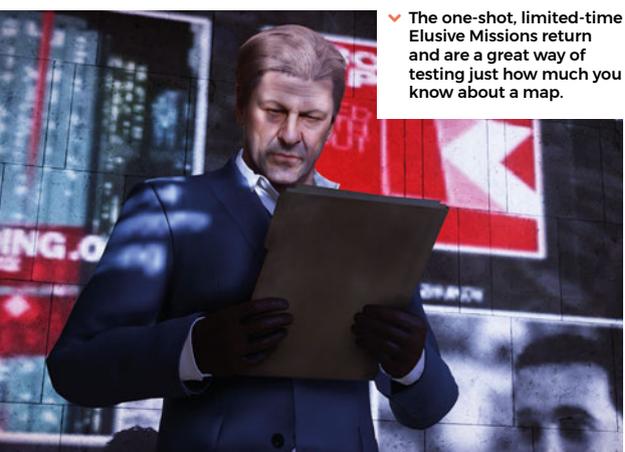
walking target? This is where the curious player is in their element: the purpose of any of these potential routes are clear, but flawlessly executing them is still a mystery. *Hitman 2* has embellished on this franchise staple more than ever, with each mission providing a healthy supply of suggested routes and story tidbits to track down, but it's down to the player to want to experience those.

There's still a familiar amount of improvisation, too, as once-infallible plans inevitably fail, more interesting prospects present themselves, or unexpected NPC reactions make the anticipated impossible. There's a beauty in this, in allowing *Hitman 2*'s missions to play on however crudely the ball lies. But to get the most out of the game – to rinse it of everything it can offer – it really requires a desire to see a plan through to 'proper' completion. Yet that's where some of the frustrations are born, and in truth they're not ones alien to the franchise.

There's a clumsiness to everything: non-player characters that just stop working, with only a reload or a broken neck the solution; targeted

“It's a franchise built for the inquisitive, and that's perhaps more integral than ever with this entry”

✓ The one-shot, limited-time Elusive Missions return and are a great way of testing just how much you know about a map.



button prompts unexpectedly switching the moment an NPC passes by, causing a fistfight when all you wanted to do was read a newspaper; guards getting riled the moment you cross some invisible barrier into 'trespassing', forcing you to hide in a bin somewhere for five minutes.

Then there's the fact that a lot of the actual assassinations are still prescribed in many ways; the sandbox environments give some amount of player agency, but the actual hits themselves don't feel as creatively rewarding if they're executed as IO Interactive intended.

Instead *Hitman 2*, like most other *Hitman* games before, is more memory game than puzzle-solving, about practising enough times to become the perfect assassin rather than improvising a solution of your own devising.

And perhaps that's the only real criticism of *Hitman 2*: it's just not brave enough to mix up the formula that has admittedly worked so well for it for years. If 2016's *Hitman* was a return to form, resetting the template for what IO wants the franchise to be, then *Hitman 2* is the natural, if unsurprising, extension of that: bigger, better, more things to do. This is engrossing stealth and the best example of the series' playful, experimental nature... and that's enough. But it will always be enough for the very gamers that *Hitman* is popular with; it's not until these cat and mouse playgrounds can shake off these ingrained issues that *Hitman* will ever reach beyond that. 🗡️

HIGHLIGHT

There's always been a real separation in *Hitman* between starting a level for the first time and being prepared enough to aim for that revered Silent Assassin, Suit Only ranking. The larger, more expansive, and interconnected maps make that divide all the greater and feel more rewarding for spending hours – and it is hours – unearthing every secret each level hides.

VERDICT

A truly definitive *Hitman* experience, with all the ills and thrills that come with it.

82%

Info

GENRE
Racing game

FORMAT
PS4 (tested) /
XBO / PC /
Switch

DEVELOPER
Caged Element

PUBLISHER
Wired
Productions

PRICE
£32.99

RELEASE
Out now

REVIEWED BY
Jack Yarwood

HIGHLIGHT

Grip: Combat Racing provides an almost unprecedented amount of freedom for a racer, with its grip mechanic letting you drive pretty much anywhere – assuming you have enough speed. It's possible to be shoved off course by an opponent, only to latch onto the narrow edge of the track and continue racing unharmed.

▼ You'll be more susceptible to spinning out when travelling fast. To counter this, stop accelerating and your vehicle will come back under your control.



Grip: Combat Racing

Oh, what a feeling – when we're driving on the ceiling

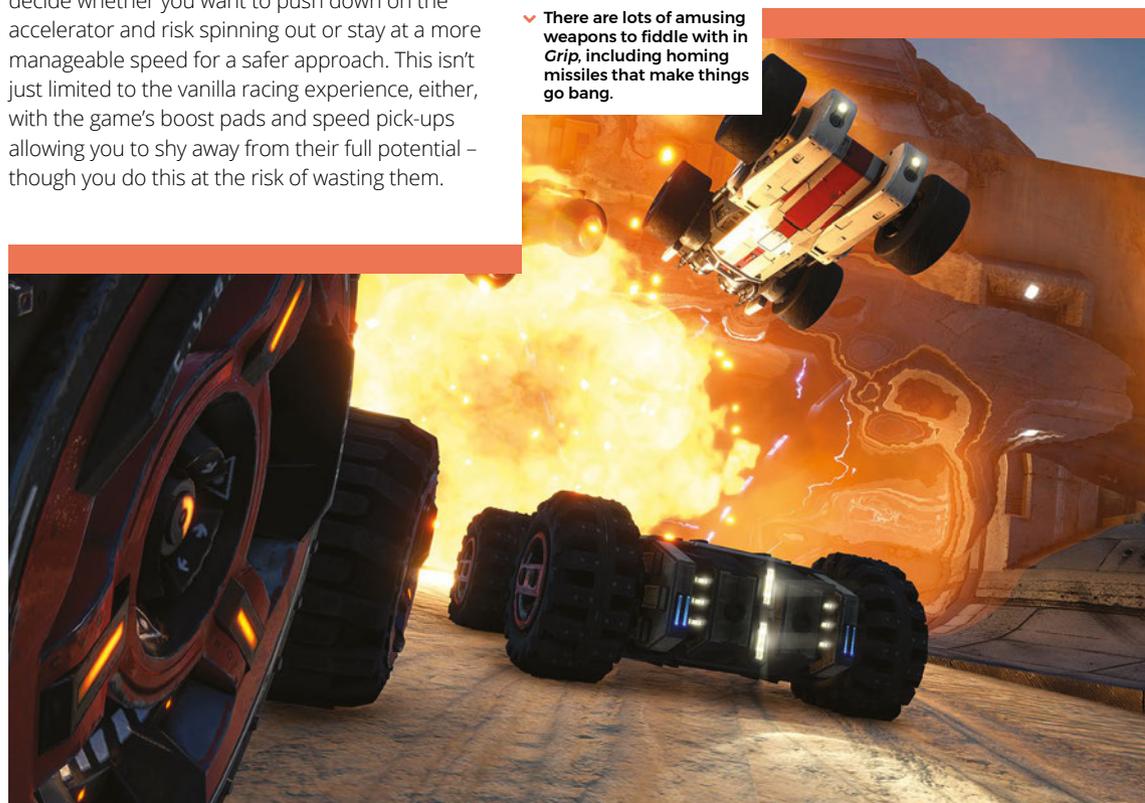
In my first few moments playing *Grip: Combat Racing*, Caged Element's chaotic arcade racer, I reached insane speeds, rammed players off the map and into the abyss, and fired bursts of explosive weapons at my competition. Unlike a lot of other contemporary racing games, *Grip: Combat Racing* is designed to be played aggressively, with the brakes being more of a formality than a necessity. Taking inspiration from the *Rollcage* games from Psygnosis, it puts the same focus on maintaining your speed and trying to remain in control of your vehicle as you power through the different maps, driving along any and all surfaces.

Given its emphasis on speed, you might expect the experience to feel a bit unwieldy, but it is surprisingly intuitive. At all times, you can ease off the accelerator to regain greater control of your vehicle, with the game operating on a risk-and-reward-style system. Fallen behind? Then you can decide whether you want to push down on the accelerator and risk spinning out or stay at a more manageable speed for a safer approach. This isn't just limited to the vanilla racing experience, either, with the game's boost pads and speed pick-ups allowing you to shy away from their full potential – though you do this at the risk of wasting them.

The flipping mechanic is another satisfying feature. Whereas in other games, crashing or falling on your roof would result in a game over or cause a huge delay to your lap time, here you can simply carry on, as the vehicles are designed to work on both sides. This is especially useful for those moments when you unintentionally flip upside down during a jump or fall from the ceiling onto your roof. But it can also be used to perform stylish stunts as you decelerate mid-air and perform a backflip or two, before continuing on without a scratch.

The variety of the maps is also pretty commendable. The game features four distinct planets, with futuristic cities, icy tundra, grassy plains, and industrial deserts. All of these maps have their own unique obstacles to avoid, including gun turrets, steep cliffs, and sharp turns. But they are designed with the same key principle in mind:

▼ There are lots of amusing weapons to fiddle with in *Grip*, including homing missiles that make things go bang.





> You can reach some incredible heights in the game. Just make sure you nail the landing!

to be a playground for players who want to go fast. Each level is filled with large stretches of flat terrain to build up speed, but there are also pathways and shortcuts that you can discover if you're going fast enough. Some of these shortcuts appear to be more intentional than others, however.

While playing, I lost control of my vehicle in the air, only to latch onto a wall and drive through a tiny gap in the terrain, emerging unscathed on the other side. This shortcut clearly wasn't intended, but the potential to grip onto the side of walls and onto ceilings encourages you to take these strange twists and turns along the side of mountains or balance on the edge of tracks to find hidden routes.

The game is also stacked with tons of additional features. Not only is there split-screen co-op and the option to move the competition online, but there are also other game modes for you to work through. These include arena deathmatches where players can try to destroy each other with an assortment of devastating weapons, including homing missiles and Gatling guns; carkour (or car parkour); and ultimate races, which are determined on points as opposed to just your final position. These modes offer a collection of different settings to experiment with, but the core game always retains that same focus on speed and aggression.

Arguably the best way to uncover these modes is through the campaign, where you need to work through a bunch of different races and challenges – the goal being to level up your reputation and unlock new vehicles, advancing upwards through the game's numbered tiers. It's in the campaign mode that *Grip: Combat Racing* introduces a rivalry system. After completing all the challenges in a tier, you're given one last challenge: a duel against the vehicle that you sparred with the most over the last few events. This final obstacle usually takes the form of a race and results in some of the most brutal competitions, as you both try to one-up each other on the way to the finish line. It's a simple yet efficient way to make the player

more invested, and helps to create more of a narrative between them and the other faceless racers.

The camera, meanwhile, sticks out as one of the game's few glaring faults. For the most part, the camera is functional and unobtrusive, but when you're pushing those higher speeds inside caves or tunnels, snapping rapidly between ceilings and floors, it can become quite disorientating and confusing to know which way you should be turning or where you are in relation to the rest of the level. This isn't necessarily game-breaking by any means, but it does force you to be more cautious in confined spaces than you might like to be.

As a fast arcade racer with tons of invention, though, *Grip: Combat Racing* fits the bill. Its turn of speed is exhilarating, while its mechanics create a tense climate where players are constantly on the brink of glory... or a humiliating collision. 🗣️

✔ The best way to get a head start on your competition is by hitting the accelerator when you see the word 'Go'. This gives you a little bonus boost that you can use to pull ahead early.



“Grip is designed to be played aggressively”



> Gripping onto the walls is all dependent on the amount of speed you've built up. If you're going too slow, you won't make it far before falling off again.

VERDICT

Grip: Combat Racing is a thrilling successor to the *Rollcage* games.

81 %

Rated Review



HIGHLIGHT

Getting good at a character requires understanding their Quirk, allowing for experimentation that feels a lot like how the characters progress in the show itself.



Review

My Hero One's Justice

Even if you haven't heard of the anime, the fighting's good

▲ Iida can close distances very quickly, while Bakugo does better with some distance between him and the enemy.

Info

GENRE
Fighting

FORMAT
Switch (tested)
/ XBO / PS4
/ PC

DEVELOPER
Byking Inc.

PUBLISHER
Bandai Namco

PRICE
£49.99

RELEASE
Out now

REVIEWED BY
Joe Parlock

VERDICT

One for fans rather than newcomers, this is an impressively quirky and technical fighting game.

68%

It's one of the hottest properties around, so it's no surprise that the school-for-superheroes anime *My Hero Academia* would eventually receive the video game treatment. What is surprising is that it's a remarkably solid fighting game.

My Hero One's Justice is a 3D fighter in the same vein as the *Dragon Ball Xenoverse* and *Naruto Ninja Storm* series, where two characters duke it out with the use of their Quirks (*MHA's* superpowers). The fighting is deceptively simple, relying on three attack buttons, and modifiers that can turn all three into special abilities and devastating Quirk attacks. As some Quirks work much better at a distance, while others are only viable up close, fights focus on understanding the limitations of a Quirk, manoeuvring, and poking at the enemy's defences rather than memorising optimal combos and nailing button inputs as is standard for the genre.

Compared to other fighting games, *My Hero One's Justice* is a bit slim in its offering of game modes. Alongside the standard online, arcade, and mission modes, there is a story mode that makes the weird decision of not starting at the beginning of the story. This is one area that will alienate newcomers to *My Hero Academia*, as it glosses over almost all character motivation for the entire first series of the anime and drops players straight into the climax of a second series story arc. The story's told through a slideshow of anime stills with the odd fight crammed in the middle, though, meaning those who'd rather stick to the other modes until they're all caught up won't be missing much.

The roster features a decent spread of the most popular characters, be they students, Pro Heroes, or Villains. Their Quirks are balanced nicely against one another, allowing each character to fill vital roles in both their primary fighter and assistance roles. For example, Bakugo is a highly versatile mid-range fighter, Aizawa can grab and erase the enemy's Quirk, while All Might is an absolute beast up-close. It would have been nice to see some secondary characters given the spotlight, as the *MHA* universe is full of interesting characters who aren't explored enough in the anime or manga, such as Class 1-B, or the teachers of UA. If the game wants to skip vital story for newcomers, then only going for the most popular characters for its cast is a waste.

My Hero One's Justice is a surprise, then. What initially presents itself as a bare-bones tie-in game quickly unfolds into an engaging, strategic fighting game with a unique cast. It's certainly not one for newcomers to the *My Hero Academia* world, but those who've already been sucked in to Midoriya's story will appreciate how detailed and true to the source material this adaptation can be. 🍎



✓ Making clever use of each character's Quirk is the difference between a good fighter and a great Hero.

HIGHLIGHT

The zombie designs are stellar, looking just like the Walkers from the TV show. Complete with dynamic damage modelling, every slice of the machete reveals more decomposing viscera in all its gruesome glory.

Review

Overkill's The Walking Dead

Unlike the zombies, this shooter isn't at all staggering

The *Walking Dead* has struggled to make the leap to games. TellTale's excellent adventure series came close, but financial woes overshadowed a lot of achievement there – and the less said about 2013's *Survival Instinct*, the better. Now Overkill (of *Payday 2* fame) is tackling Robert Kirkman's zombie epic in its inimitable co-op FPS style. Unfortunately it hasn't quite nailed it.

Much like *Payday 2*, equipment and abilities can be swapped out before challenging missions to specialise in stealth or firepower, and further differentiate each survivor's unique traits through skill points. This granular customisation system helps make two players using the same character still fill different roles in the team, so nobody's forced into characters they dislike. Unlike *Payday 2*, *The Walking Dead* features a global sound system, where gunshots, traps, car alarms, and more can all attract increasing amounts of unwanted attention, requiring communication and teamwork to avoid unending rivers of the dead. That, or a lot of firepower.

Some of these systems aren't communicated well, though. Rescued survivors can be sent on missions, or assigned posts at the home base, but this isn't explained in the tutorial, despite how much of an impact it has on abilities. What alerts zombies is also unclear, as characters can shout to each other from across the map with

no repercussions, but running too close to an idle zombie will send them after you. *The Walking Dead* really could do with an overhaul of its UI to overcome these problems as it obfuscates integral elements of working as a successful team.

The DC streets and buildings are all gorgeously dilapidated, but the real stars are the zombies (sorry, 'Walkers') themselves. All rendered in impressively large numbers, guts and bones burst out from putrefying flesh. Rather than being simple cannon fodder, each one feels like a threat in its own regard. Overkill has done a fantastic job of recreating the oppressive threat they pose in

the source material.

Despite all this, it's impossible to get away from the game's biggest problem: it's a technical mess. Long loading times, poor performance, slow matchmaking, buggy AI, and badly placed enemy spawns plague *The Walking Dead*,

making seeing the good bits difficult.

Overkill has a history of this, with *Payday 2* also being released in a poor state before patching pulled it back, but that took several years on PC and has yet to happen on some consoles.

Looked at just right, Overkill's *The Walking Dead* is a great advancement over *Payday 2*. It features tense zombie survival, great co-op play, and a haunting new take on a world many are already familiar with.

Unfortunately, like a walking corpse, the infestation of bugs makes getting close enough to see that difficult. 🗣️

“Slow matchmaking, buggy AI, and badly placed enemy spawns plague *The Walking Dead*”

Info

GENRE

Co-op first-person shooter

FORMAT

PC (tested) / XBO / PS4 (console version Feb 2019)

DEVELOPER

Overkill Software

PUBLISHER

Starbreeze Publishing, 505 Games

PRICE

£46.49

RELEASE

Out now

REVIEWED BY

Joe Parlock

VERDICT

Some fantastic ideas and great co-operative play, but technically it feels like it's held together with gum and spit.

49%

Rated
Review



HIGHLIGHT

War Stories cast a welcome light on largely forgotten WWII operations, and the objective-based missions offer a fine blend of action and exploration, often set against stunning backdrops. It's such a shame you're forced to play alone when the vast environments all but beg collaborative play.

▼ Daily assignments encourage you to switch up your weapons and play a different way – not always to your advantage, mind.



Review

▲ Not quite the white Christmas we were hoping for... and Santa probably shouldn't go near those chimneys.

Battlefield V

Tanks for the memories (even though they weren't so great)

Info

GENRE

First-person shooter

FORMAT

XBO (tested) / PC / PS4

DEVELOPER

DICE

PUBLISHER

Electronic Arts

PRICE

£59.99

RELEASE

Out now

REVIEWED BY

Vikki Blake

VERDICT

Fun with friends, but a little onerous on your own.

63%

A melancholy air permeates *Battlefield V*. It's in the stories that turn the lens away from the heroism of war and instead concentrate on the faces of the soldiers wielding the weapons.

These vignettes – called War Stories – pick up where *Battlefield 1* left off, providing a limited buffet from which you can select solo stories of derring-do. Like its predecessor, *BFV* tries – and mostly succeeds – to humanise war, and these individual missions are cinematic and painstakingly crafted as they reveal (and revel in) the lesser-known battles of World War II. But the cinematics and soaring soundtrack sit at odds with the game itself; on one hand, *BFV* wants you to give WWII the grim circumspection it deserves. On the other, it's egging you on with headshot bonuses.

While the three War Stories at launch offer six hours of individual play, the meat of *BFV* lies in its myriad multiplayer modes. But the co-operative Tides of War mission hasn't launched at the time of writing, and Battle Royale – the priority for every developer except DICE, apparently – is also locked. This means *Battlefield V* is chiefly a multiplayer game with a handful of solo missions that, ironically enough, would be more enjoyable if you had a buddy or two in tow.

But my, it's gloriously good fun. Fully destructible set-pieces ensure even protracted battles avoid tedium, your environment ever-

changing around you. Dynamic weather can almost switch the fortunes of a match in an instant, obscuring sniper sightlines and helping you get the drop on your enemy. And while the character classes feel a little unbalanced, the gunplay remains meaty and satisfying.

The trouble is, it lacks polish. Soldiers struggle to scale tiny ridges. Sometimes you can't revive, or you'll pick up a weapon instead of your fallen comrade. Waiting to parachute in takes too long. Dying bodies flail about with aggressive ragdoll animations. Just eight multiplayer maps are in rotation, which simply isn't enough variety. Stealth is encouraged, but rarely successful, and the astonishing accuracy of your AI foes becomes frustrating. And the loading times? I could live with it if matchmaking was consistent, but all too often your squad waits for several minutes awaiting entry... and then one or two squaddies get unceremoniously kicked out, forcing you all to try again.

When you can get into a match, though, you'll find the magic of *BFV* lies not just in the number of kills you notch up – you can climb the leaderboard simply by working towards your in-game objectives. It's a great way for players with a low kill:death ratio to participate, and it's never better than when you're with your pals.

Another month or two's bug-squishing and polish could've elevated *Battlefield V* from a good game to a great one. As it stands, it's difficult to justify the premium price tag for a limited, lacklustre experience. 🗿



Review

◀ The controls are beautifully calibrated – even in its largest form, the bird remains highly responsive. Power-ups that increase movement speed aren't worth bothering with.

Black Bird

From Japan comes an exquisitely made 2D shooter

Since striking out alone, designer Yoshiro Kimura (*Little King's Story*, *Chulip*) has been quietly making outstanding, idiosyncratic mobile games such as *Dandy Dungeon* and *Million Onion Hotel*. The third game from Kimura and his team at boutique developer Onion Games marks a change of format, but it's cut from a similar cloth: *Black Bird* is at once immediate and surprisingly deep, with sturdy foundations underpinning its surface weirdness.

In some ways, it's the studio's most conventional game to date, as a side-scrolling shooter with elements of *Defender* and, more obviously, *Sega's Fantasy Zone*. It's almost an evil alter ego of the latter, in fact, trading those vibrant colours for a much darker palette befitting the game's doom-laden atmosphere. Your motivation is established in a matter of seconds, as you see a young girl collapse and die. Lying in the dirt, she's ignored by most passers-by, with one man callously poking her prone body with a cane. Suddenly, she undergoes a startling transformation into a phantom crow, setting out to exact a violent revenge on the inhabitants of this unfeeling world.

Your ultimate goal is to destroy well-armoured towers stationed at regular intervals on the game's wraparound levels. All the while, you'll fend off a variety of smaller enemies on the

ground and hovering in various dirigibles. Anything you destroy drops green jewels, which will boost your size and attack power once you've collected enough of them. Then, once the last tower falls, an end-of-level boss arrives.

If you're rushing, you can race through all four stages in well under half an hour. But there's a benefit to taking your time. Shooting successive enemies steadily tops up a combo meter that resets if you don't find more targets quickly. As such, you'll need to make sure you've got a

supply of regular enemies while you're taking out the towers, before racing along to the next. Use one of your limited supply of smart bombs – normally reserved for tight

spots – when you've maxed out the combo meter and you'll let off a ferocious screech that engulfs everything in a swirling vortex, releasing a cascade of gems and significantly boosting your score.

Like the best shooters, it strikes a fine balance between risk and reward. Those gems shrink when they bounce, encouraging you to get close to enemies so you can pick them up sooner, though that naturally increases your chance of taking a hit. Delay destroying the final tower, meanwhile, and eventually a large enemy will arrive on the scene to chivy you along.

There's more to discover once you've defeated the final boss, where the game eventually matches the intensity of its 'bullet hell' peers. An unlockable True Mode cranks up the pace and projectile volume, with fresh enemy types and new endings determined by your score. That might not be incentive enough for some players to stick with a game that's admittedly slight in stature. But it would be ignorant indeed to dismiss this deceptively smart shooter as shallow. 🗨

"Black Bird strikes a fine balance between risk and reward"

✔ Even at its most hectic, *Black Bird's* action remains consistently readable, those red bullets and arrows standing out against the near-monochromatic backgrounds.

03461960
080 x 021
080 x 022
080 x 023



HIGHLIGHT

There's much to admire about *Black Bird's* presentation, from the spectacular, screen-blurring effects of a smart bomb to its odd little flourishes like *Million Onion Hotel's* returning asparagus rockets, and the way enemy patterns are timed to match the soundtrack. In short, it's a wonderfully reactive world, rich in offbeat character.

Info

GENRE

Scrolling shoot-'em-up

FORMAT

PC / Switch

(both versions tested)

DEVELOPER

Onion Games

PUBLISHER

Onion Games

PRICE

£14.99

RELEASE

Out now

REVIEWED BY

Chris Schilling

VERDICT

Short but exquisitely fashioned and highly replayable, *Black Bird* injects a classic genre with eccentric charm.

84%



HIGHLIGHT

Arca's Path envelops you in its soothing dreamscapes, accentuated by the ambient score composed by electronic musician Raffertie. Even your ball's presence causes the environments to crackle with life in a neat stop-motion effect. It's a place you can unwind in at the end of the day.

Review

Arca's Path VR

A classic puzzler goes hands-free

▲ The one thing your controller can do is pause the action to better take in the view.

Info

GENRE
Puzzle

FORMAT
PSVR [tested] / Rift /
Vive / Steam VR /
Gear / Go / Focus

DEVELOPER
Dream Reality
Interactive

PUBLISHER
Rebellion

PRICE
£13.99 (console/PC)
/ £7.99 (mobile)

RELEASE
Out now

REVIEWED BY
Alan Wen

VERDICT

Arca's Path is an old-school puzzler given a VR spin – just not as innovative or intuitive as you'd think.

72%

Accessibility can mean different things. For Nintendo, the Wii Remote provided intuitive gesture-based controls, while Microsoft's Adaptive Controller allows players with disabilities to customise controls according to their needs. In the case of *Arca's Path*, its use of VR removes the controller altogether, leaving players to literally use their heads.

On its own, *Arca's Path* looks like a prettier version of the arcade classic *Marble Madness*, where you're rolling a ball through labyrinthian levels to reach the goal. But instead of conventional controls, head-tracking means simply gazing at a point will have the ball rolling there. It's not quite the VR telekinesis you expect, as you're not directly controlling the ball, but rather using your head to control a cursor that magnetically pulls it along. Looking just a little in front of it will steer it at a gentle pace, while looking further away pulls it at a higher speed.

If you're liable to motion sickness, the velocity fortunately doesn't ramp up too much, even when rolling downhill or navigating a half-pipe. Indeed, there's an overall dreamlike calm to *Arca's Path*. Without a timer counting down, you're free to take your time negotiating your run through these ethereal mazes that get progressively longer.

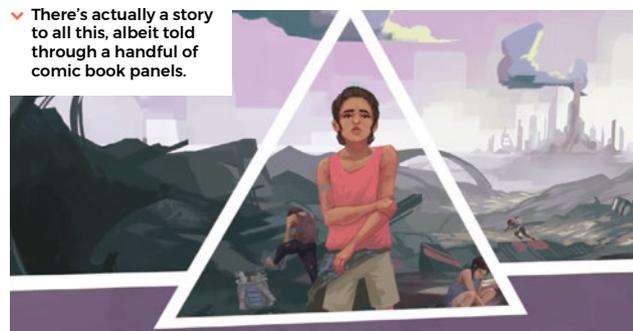
Gradually, a handful of new mechanics and hazards are introduced, and it's not long before you're crossing paths without any barriers preventing you from tumbling into the abyss or encountering floors that crumble away if you roll over them too fast. Fortunately, the game

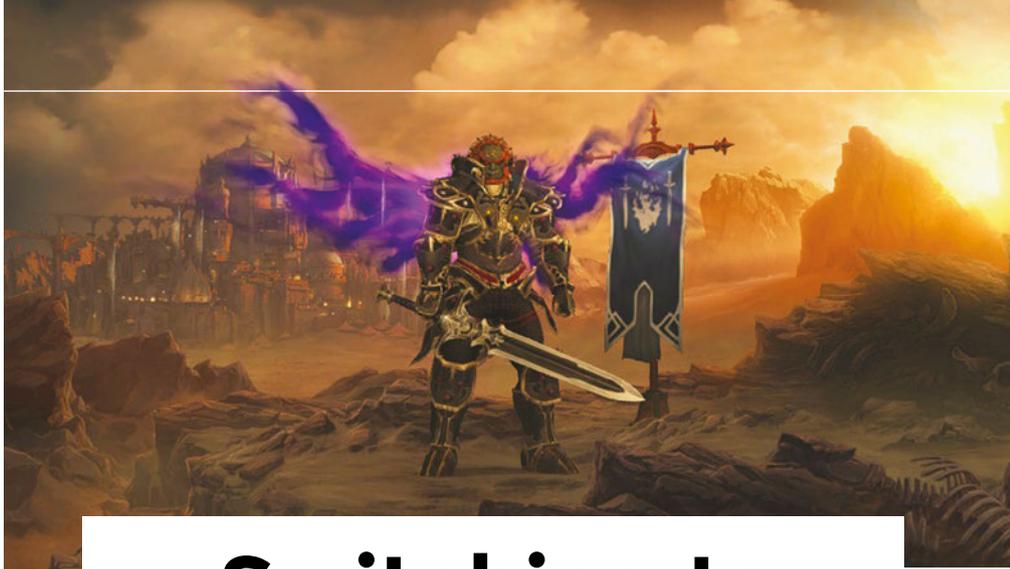
is generous with its checkpoints, and things like activated switches or collected crystals will remain in place when you retry. For the most part, then, *Arca's Path* feels like an escapist playground, albeit one with plenty of peril lying beneath its light narrative.

Whether its gaze-based controls are more intuitive and innovative than your thumbs is debatable; there's the occasionally tricky path where my head's spatial awareness failed me in a way that using a control stick or mouse wouldn't have. There's no denying the novelty of head-tracking as a control function, but it's ultimately neither as clever or inventive as, say, the world-tilting antics of *Super Monkey Ball*.

Like many VR games, it's also rather short at just 25 levels. There is, however, incentive to replay if you want to collect all of the crystals, which unlocks a time trial mode for that level. This should give *Arca's Path* the challenge and longevity others – like cool-headed speedrunners – may crave. 🎮

▼ There's actually a story to all this, albeit told through a handful of comic book panels.





Switching to social in Diablo 3

The devil – in your hands – is an idle plaything

There's one big reason we can't quite get enough of *Diablo 3* on Nintendo Switch right now, and it isn't anything to do with what's here in the overall package.

It's not that *Diablo 3: Eternal Edition* isn't a nifty little box of treats on Switch – it comes with the *Reaper of Souls* and *Rise of the Necromancer* expansions, a few Nintendo-specific treats like a Ganondorf – of *Zelda* fame – transmogrification (you can just call it a 'costume'), and is pretty much the ultimate collection of the six-year-old game.

But that pales in comparison to the fact that this is a portable version of *Diablo 3* you can play with up to three others on one machine. Most of us never actually knew we wanted that. Turns out, we did.

Bringing four people together in front of one machine happened before on the console ports, but never before have we been able to take that one machine on the go wherever we choose. The living room. The kitchen. Outside. Back inside again. We were waiting for a package, the overall 'staying in the house' thing had to be adhered to – but the point remains: this is *Diablo* on the go.

Even discounting the multiplayer aspect, it's a surprisingly fresh feel behind the game, again thanks in no small part to its portability. Pick it up, play it for a bit, whack it back in your bag, and forget about it until you want another 20 minutes. It's *ideal* for the format.

OK, so if you're a *Diablo 3* stalwart and have been playing it non-stop for six years, you're unlikely to see much point here. But those of us who dipped in at launch then forgot to ever go

back are left with a remarkably fun game that supports local co-op in the best fashion: everyone crowded around, bickering, and taking their time in the inventory. How it should be.

Controls on a single Joy-Con are, as you'd expect, a mite pernickety – even so, they work. You can play the game. And *Diablo 3* is such a laid-back experience in the most part, away from the hardcore difficulties at least, that it

doesn't matter so much if you're flubbing your inputs every now and then.

The game runs well, only slowing every now and then when things are at their most hectic, and even on the small screen in portable mode you're

able to figure out who each player is controlling and what you're doing at any given time. That ease of play really does help out here.

It's just *such* a well-fitting game, it's honestly surprising. The number of folks who scoffed at – or outright ignored – Blizzard's announcement that it was bringing *Diablo 3* to Switch was not inconsiderable. Turns out we're all a bit egg-y faced after the mini-revelation of *Diablo* Switch's greatness.

When Nintendo announced the Switch back in 2016, there was some confusion as to what it would be – who it would appeal to. Would it be riddled with casual, lightweight games like the original Wii? Would it be a place for Nintendo's greatest franchises to flourish? Turns out it was both of those, plus an indie machine, *plus* a new and exciting way to play half-a-decade-old PC action-RPGs.

And if that doesn't surprise you, you're probably not paying enough attention. 🗨

"It's just such a well-fitting game, it's honestly surprising"

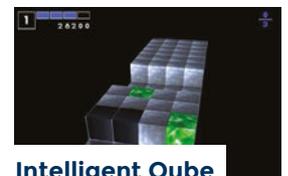
Wireframe Recommends



Tekken 3

PLAYSTATION CLASSIC

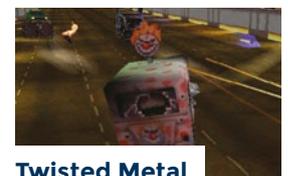
The PlayStation Classic is out now, and it turns out *Tekken 3* is still cracking good fun. There are legitimate arguments to be made that this is the best of the franchise.



Intelligent Qube

PLAYSTATION CLASSIC

Kurushi in Europe, the deal is still the same: avoid the incoming blocks. Like many a solidly designed puzzle game, *Intelligent Qube* has aged brilliantly.



Twisted Metal

PLAYSTATION CLASSIC

The clowns-and-coppers vehicle shooter, however, has not aged so well. All the same, in split-screen it's still a lark – if only not quite as impressive as we remember.



Half-Life

The legendary FPS made many leaps, but leaving you front-and-centre in the story was one of its biggest achievements

VALVE / 1998 / PC, PS2, MAC

Two decades on from release, it can be difficult to pick out the individual elements introduced (or at least popularised) by *Half-Life*. So ubiquitous are its individual features in modern gaming, you'd be forgiven for thinking they'd been around all along.

One area the game did innovate in – and one it's easy to remember – is in its presentation of the story. Up until *Half-Life*, games would wrench control from the player and force them to watch as a black-bordered cutscene played out, demanding their attention as interminable exposition flowed.

Just two months before *Half-Life*'s release in November 1998, the PlayStation classic *Metal Gear Solid* hit. While itself innovative and inspirational to future generations of games, it was

one of the absolute worst offenders when it came to cutscene crimes, with some of these mini-movies running upwards of ten minutes in length. That's a long time to be sat doing nothing.

With this context in mind, it helps to hammer home just how much of a mould-breaker *Half-Life* was. Its cutscenes – and let's just stick with calling them that, because they were – didn't take the player out of it. You still had control; things continued happening around you while you spent your time bunny-hopping around nearby furniture. It was still interminable exposition, but it didn't have the massive negative of ripping you out of the game at the same time.

Half-Life was a step up for video game narrative, offering a level of immersion – and polish – we hadn't been exposed to before. Immediately, other games, especially first-person shooters, looked a bit childish and silly by comparison. Tough guys with big muscles making sub-Schwarzenegger level quips in action scenes the player had no control over suddenly lost their lustre.

Of course, it wasn't just in the presentation of the story where *Half-Life* strived for a level of immersion. Other things we now take for granted were brought to the fore by Valve's classic, like fallen soldiers dropping their weapons where they died, those same soldiers demonstrating self-preservation instincts, or health stations being in logical positions rather than floating

medikits in the middle of nowhere. *Half-Life*'s manual specifically pointed these elements out, which to modern ears must sound ludicrous – but in the hazy mists of 1998, this was a new approach, worth remarking on.

It has to be equally rewarding and galling for features your game introduces – or, again, popularises – to become omnipresent. On one hand, you've changed gaming. On the other hand, 20 years later, plenty of people have forgotten – or never even known – you did it to begin with. Fortunately, *Half-Life* has a lot of people flying its flag: we were there when it changed gaming forever, and we remember it well. 🎮

“Up until *Half-Life*, games would wrench control from the player and force them to watch as a cutscene played out”

Next Issue

ON SALE 20 DEC

DISCO ELYSIUM

Behind the scenes of a
ground-breaking detective RPG

Don't
miss out!
Subscribe
PG54

Editorial

Editor

Ryan Lambie

Email ryan.lambie@raspberrypi.org

Features Editor

Ian Dransfield

Email ian.dransfield@raspberrypi.org

Sub Editors

Phil King and David Higgs

Design

criticalmedia.co.uk

Head of Design

Lee Allen

Designers

Harriet Knight, Mike Kay

Contributors

Adam Barnes, Lottie Bevan, Vikki Blake, David Crookes, Rik Cross, Konstantinos Dimopoulos, Andrew Fisher, Stuart Fraser, Dia Lacina, Joe Parlock, Chris Schilling, Howard Scott Warshaw, Alan Wen, Jack Yarwood

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:

rpihelp@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., 30 Station Road, Cambridge, CB1 2JH. 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).

Also

- ▶ Destiny's long-lost soundtrack
- ▶ What a video game producer does
- ▶ Using SIN and COS functions in your games
- ▶ Breaking into the games industry



wfmag.cc