Wireframe

LIFTING THE LID ON VIDEO GAMES

Mystery Shopper

Mystery and making in Mineko's Night Market

GAME-CRAFTING TIPS/ADVICE/GUIDES

Make a score attack game
Cram an action game into 1kB of code

Horror city planning
Simple ways to make scarier locations

Amazing sound design
Create worlds with sound and music

Subscribe today
12 weeks for £12*

£3
ALL FORMATS

Wireframe
LIFTING THE LID ON VIDEO GAMES

Degrees of
Separation
Chris Avellone on
narrative design

Material
worlds
Making games
with wool and clay

PR
FE

THE DEVS
WITH SON

Mystery
Shopper

Mystery and making in
Mineko's Night Market

Issue 8 £3
wfmag.cc

GAME-CRAFTING TIPS/ADVICE/GUIDES

**Make a score
attack game**
Cram an action
game into 1kB of code

**Horror city
planning**
Simple ways to make
scarier locations

**Amazing
sound design**
Create worlds with
sound and music

9 772631 672000
08

£3
ALL FORMATS

ame
ON VIDEO GAMES

METRO EXODUS
POST-APOCALYPTIC TRANSPORTING

HOCK
OID BASTARDS

The ugliness of photorealism

Can we stop calling triple-A games 'beautiful' as a reflexive move? This is probably an unreasonable ask, and there's likely some beauty to be found in some of these big, big games. But every year we get the same major studios racing towards a chasm of hyper-fidelity. Throwing more money, time, and bodies into the problem, as though they're trying to build a bridge just stable enough to cross into the next hardware cycle.

And every year, we get critics, and gamers, and marketing departments all enraptured with the beauty of this aggressive arms race towards a perfect representation of reality in a virtual space. They use words like beautiful, pretty, and gorgeous – can I just say, it makes my skin crawl?

What we're all responding to (because none of us is really immune, not at first blush anyway) is the power of spectacle. It's designed to override our critical faculties, to arrest us in an often unearned awe. And while the technical ingenuity and extensive labour of the developers – who work for massive corporations to produce massive games on massive budgets so they can garner massive sales for their executives and shareholders who in turn get massive returns – is profound and something to be applauded, we owe them, ourselves, and this industry so much more.

The issue of photorealism in games is a matter of personal taste, a question of aesthetics. I'd argue that what people are responding to is the technical mastery. That we've been conditioned to believe more polygons, more particles, newer more advanced shaders are what is crucial and amazing. That there really hasn't been a beautiful game in the triple-A space for a while now, because how can something be beautiful when we know the practices involved in its creation, when we see how these companies executives behave, and are rewarded for fractionally better margins.

Beauty can rise up out of ugliness. It is possible. But I can't help but find the reflexive need to fall over these
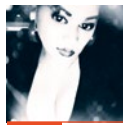
DIA LACINA

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**

#08

massive commercial ventures and characterise them as 'beautiful' extremely distasteful. We should know better.

Most, if not entirely all, game developers who work at triple-A studios don't enjoy many protections from abusive crunch practices: massive layoffs (as we've just seen at Activision Blizzard) to make earnings calls sound better, or the ability to petition for the kinds of bonuses or residuals they deserve from shipping a title that sells. We know that labour conditions in this industry suck because we know how quickly people burn out and leave. We watch the latest Game of the Year brought to life only through burning the passion these developers have in making games as fuel.

When you think about the human cost of these games – is beauty a word that is even appropriate? Sure, *Red Dead Redemption 2* may have an incredible number of trees that look ripped from a photograph, snow may clump on the branches in a realistic way, and sheer rocky peak towers may poke ominously through volumetric clouds in the distance. But how many people burned out in making that scene, left before the game shipped, and had their named stripped from the development credits entirely because that's Rockstar policy? How many birthdays and anniversaries are missed? How many relationships collapse under the pressure of crunch? What happens to worker health, mental and physical, as they're continually pushed beyond what we should consider reasonable?

Plenty of people will argue that great art requires sacrifice. And that's bullshit – the kind that reduces people down to their capacity to create value for a corporation. There's no game worth the kinds of sacrifices these companies routinely ask of their employees.

Instead of calling these games beautiful, let's acknowledge their human cost. Let's stop equating the spectacle of money with aesthetic greatness. Let's stop calling things undeservedly beautiful, and call them what they really are – expensive. It's an expense we can't afford to keep paying. ⓦ

Contents

Contents

6

Attract mode

Interface

44

Contents

12

18

63

Toolbox

Rated

WELCOME

As I write this, the internet's still abuzz with the news that Activision Blizzard has laid off eight percent of its staff – amounting to almost 800 employees worldwide. On Twitter, people who until only a couple of days ago were still staff at one of the giant's various divisions are sharing their stories of shock and disappointment – unsurprising, given that Activision made the announcement during an earnings call with investors, and staff were reportedly informed of their fate only afterwards.

To add insult to injury, the staff cuts are being made despite the firm's 'record' profits during the latter part of 2018 – according to boss Bobby Kotick, Activision didn't realise its 'full potential' last year and so, as the firm focuses its attention on its biggest franchises, hundreds of staff are being let go.

The news has rightly sparked debate about the way staff in gaming's largest companies are treated. While job losses in the industry are nothing new, the severity of Activision's cuts, and the cold way they were reportedly announced (staff had little more than rumours to go on for several weeks, according to Kotaku) is troubling to say the least. Firms like Activision may have to answer to their investors, but they should also, surely, have a duty of care to the workers that make these companies their fortunes in the first place.

Ryan Lambie
Editor

Cute is a goal

Mineko's Night Market brings mystery, commerce and cats together – we spoke to Meowza Games about its gorgeous adventure

There are times when you're idly browsing a list of upcoming games, and one just pops right out at you, rattles your visual cortex around a bit, and generally sets up shop in the back of your brain where it will stay for the foreseeable future. *Mineko's Night Market* is such a game, because first of all – just look at it. Channelling the best of Studio Ghibli, *Animal Crossing* and cats, this arrives as the full debut from husband and wife duo Meowza Games – or Brent and Brandi Kobayashi, if you want to be familiar.

> **"Making it from scratch again definitely falls into the category of a 'technical challenge'"**

Mineko sees players taking control of the titular little girl, who finds herself in the middle of a mysterious set of occurrences on the island she inhabits. Locals have been telling the tale of a legendary Sun-Cat known as Abe for many years, but recently… well, it seems Abe might be more than just a legend. There's a story of some 20-or-so hours to get stuck into, and players will be perfectly able to jump into *Mineko* and blaze their way through a linear story if they so choose.

But that would be missing a lot of the point of the game, with the Night Market part of the title in particular looking like it will demand a fair bit of attention for those of us raised in the ways of *Animal Crossing*, *Stardew Valley* and *Recettear: An Item Shop's Tale* (or the more recent *Moonlighter*). That is to say, those of us who get our kicks from engaging in naked, boiled-down capitalism in our games.

The market in the game is a place in which you will be able to sell and trade pretty much anything you can make (yes, there's crafting, too) or forage, building on top of the story-led core to make something more freeform and relaxing. Don't want to press ahead unearthing the mystery of that large cat in a mask? Go fishing for three hours. Easy.

It's been a big part of the drive for the Kobayashis to make sure *Mineko* is a game that can be played in the way players want to tackle it. It doesn't have to be stressful, it doesn't have to take over and demand all of your attention for hours, days, weeks at a time. But at the same time, there's no denying there's a lot of appeal in losing yourself in this Japanese-inspired island and its mysteries within. Plus there's loads of cats everywhere, and you've got to love that.

And it's not just an adventure mixed with a market, either – *Mineko* is riddled with side quests and minigames, aiming to always have something at hand to keep you entertained. From races to sumo (suit) matches, there's plenty going on in this beguiling, beautiful world. Though we have to point out that at the time of writing, there's no battle royale mode.

So, utterly enamoured with our early look at *Mineko's Night Market*, we decided to grab some collaborative answers from Brent and Brandi Kobayashi about the game, its inspirations, and just what sort of threats and crude elements could possibly lie underneath such an adorable facade.

Mineko's Night Market
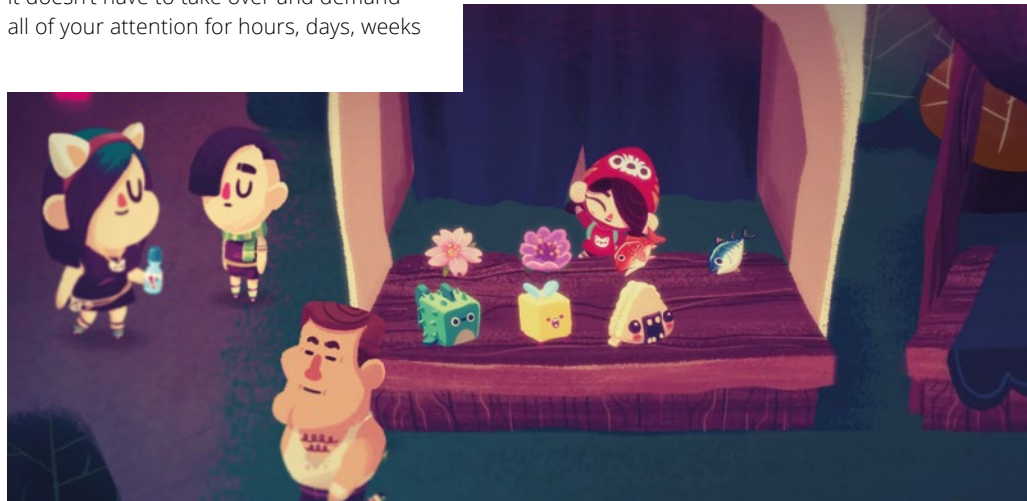
Is developing a game like this a case of making lots of smaller games inside a bigger one?

'Making lots of smaller games inside a bigger one' is basically our game design document. The game features 16 unique Night Market Main Events (one for each week of the year), all crafting workbenches include a small minigame to complete recipes, plus a lengthy storyline that's filled with varied objectives. We wanted to fill the game with surprises around every corner. We highly recommend not designing games this way.

How much variety are we looking at for players?

Although the story rests on the backbone of a linear story, we wanted to include a number of different ways for the player to achieve this end. We want the player to feel they could specialise in the aspect, or aspects, of the game they feel most compelling. For example, if the player really enjoys fishing, they could simply fish and sell their catches to vendors/the market and make most of their money doing just that. It was important for us to allow the player to feel they could continue to live their life to their playing style and pace.

How deep is the crafting system? At the same time, is it accessible for those of us who might just want to make the same thing over and over to sell?

This was one of the design decisions we went back and forth on the most during development. We started by designing these multistep *Cooking Mama*-type games for all the crafting workbenches because it seemed like a fun idea. We soon realised that we started to also fall into the category of players who want to craft the same things over and over, and the length and repetition of the minigames was feeling extremely cumbersome. We scaled back on the crafting workbench minigames to make them more akin to *WarioWare*-style microgames, so that there is still a bit of interaction/skill involved, while hopefully removing the dread of going through a gauntlet of games to create every single crafted [item]. ➡

ⱽ **It's not a bustling metropolis, but the game is full of life.**

As you're running a market, have you had to create a viable in-game economy? If so, how's that been for you?

Balancing money in the game is definitely one of the greatest design challenges to date, mostly due to the other townsfolk outside of the player who have their own money, supplies and needs. Balancing the world's economy would be much easier if no-one else had money. I guess I am starting to understand the mindset of the world's billionaires.

When the trailer warns of 'crude humour', what are you referring to? Just how saucy can a game as adorable as this be?
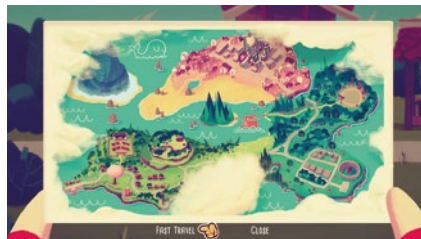
Comical butts, references to butts, and the occasional butt.

Some influences are apparent from the get-go, but are there any we might not have picked up on?

It's difficult to pinpoint specific references, but we wanted to capture the spirit of

▲ How many times can we say 'cute'?

> ## "We were adamant that the player should be able to complete the story at whatever pace they desire, but at the same time not exclude those who only want to run through the story [quickly]"

Japanese culture and the varied aspects to it; a culture that has a fascinating ability to preserve traditions while simultaneously adapting to modern times. A few of our biggest sources of inspiration are the charm and aesthetics of Ghibli films and the humour and child-like naivety of old Japanese children's cartoons like early *Doraemon* episodes. We sort of wanted to create the feeling of living inside a Japanese cartoon.

How deep do the Studio Ghibli influences go?

Not a whole lot outside of the general Japaneseyness.

▲ Brent's art style will be familiar to some.

How did the distinctive art style come to be? Was it always intended to be utterly adorable?

We've always had an affinity for cute. There are those who take the term 'cute' when referring to their art as an insult or a lower-grade form of compliment, but we will proudly wear 'cute' on our sleeve and are appreciative and thankful for anyone who finds our work to be cute. Cute is a goal.

Who is Abe? What is he? Where does he come from?

Abe is a mythical Sun-Cat that the people on the island have worshipped for hundreds of years. What was always thought to be a long-told myth has the town puzzled, as sightings of Abe have begun to surface among the residents of the island in recent days. We don't want to reveal the whole story of Abe, since that is basically Mineko's quest for the entire game!

How difficult is it to weave a narrative into a game like this? (We noted it'll actually have an overarching plot, unlike *Animal Crossing*, say.)

The most difficult part was figuring out the pacing of the storyline, since we were adamant that the player should be able to complete the story at whatever pace they desire, but at the same time not exclude players who only want to run through the story and not be tied to a heavy grind to play it out. I mean, when you play the

∧ Mineko isn't afraid of a shift down the pit.

main storyline in a game like *Skyrim*, the actual game is, what, three minutes long? We – hopefully – achieved a lot of this by tying much of the story components to the in-game calendar and filling the game with other side objectives in between.

Those besuited FBI/*Men In Black*-style fellows are a mite worrying – are we looking at genuine peril in the game?
Spoiler: Mineko won't die.

What engine have you used to bring *Mineko* to life? How's the experience been working with it?
We use Unity. It's not perfect, but the ubiquity and familiarity made it an easy choice for us. It's helpful for collaboration since a lot of developers already know it, it has a lot of third-party support for the things we don't know how to make, and plenty of documentation/support for issues we run into.

Have there been any big technical (or other) challenges along the way?
Whenever making a game composed of a billion sprites, you're always going to hit performance issues that we deal with on a near-daily basis. Also, we made the decision early in development to switch from a completely 2D game to a 2D/3D game where we impose 2D 'billboarded' sprites on a 3D plane and had to remake the game from scratch again. Making the game from scratch again definitely falls into the category of a 'technical challenge', I would surely say.

∨ The game is inspired by – and riffs on – Japanese styles, culture and traditions.
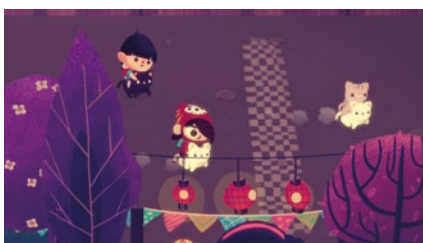
How did you keep up the motivation, going back and starting the whole project again like that?
It breathed new life into the game's design direction we hadn't initially considered. Originally, the game was a simpler point-and-click time management game that ran on a loop of resource harvesting in the day, then selling/buying goods at night. Admittedly, the game may have had a much cleaner design to begin with, but there was a lot of unsatisfactory rigidity to the design, so we experimented with the player direct-controlling Mineko in a 3D space. We found that this allowed for much more interesting possibilities for resource gathering, and in the process found that it just made the world that much more fun to explore and tell a story through. We went all-in with this direction to emphasise more of the focus on the exploration and narrative.

Thinking broadly, what are your hopes – dreams, even – for *Mineko's Night Market*? What do you want it to achieve?
There are so many games out there now fighting for players' valuable video gaming hours. We've never set out to create a game that the player should play forever, nor should they. They should also live their real life. Hopefully those who play our game can share a laugh with us, feel touched by the story, and then go outside and look at a tree. Trees are cool.

Finally, will *Mineko* feature a battle royale mode? This has to be asked about every game these days. We suggest it be called Last Cat Standing, which is simplistic, but gets the message across.
Not exactly battle royale, but we've honestly considered a multiplayer cat racing game supplemental or game mode! Unfortunately, we've backburnered multiplayer ideas for now, but we promise to give you credit for Last Cat Standing if we do. Oh wait, what about Cat-tle Royale? ⓦ

Mineko's Night Market **releases 2019 on PC, Mac and Switch.**
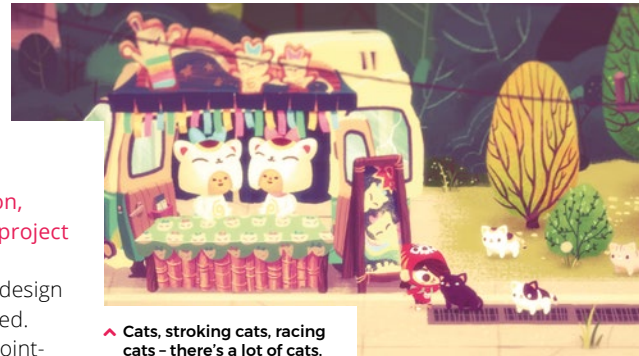
∧ Cats, stroking cats, racing cats – there's a lot of cats.

CATS ROOL; DOGS DROOL

There's a distinctly cat-heavy theme to *Mineko's Night Market*, which is of course utterly adorable. One thing has to be raised, though: what about dogs? What did those poor, slobbering idiots do to be left out in the cold like this? Brent clears it up: "We have absolutely nothing against dogs. Their inclusion was just too much of a safety concern for the cats."

And keeping on the theme of asking the Important Questions, we hit Brent with a doozy: just what is his favourite genre of cat? "Literally, the best question we've been asked," he laughs, "That grey one with the darker grey stripes but their paws and belly are all fluffy and white. I think that's the official breed name."

Journey from the East

Tales of the Neon Sea shows the Chinese dev scene is in rude health

Info

GENRE
Adventure

FORMAT
PC

DEVELOPER
Palm Pioneer

PUBLISHER
Zodiac Interactive

RELEASE
TBC 2019

D eveloped with a love of science fiction at its core, *Tales of the Neon Sea* brings a pixel art aesthetic to the realm of neo-noir cyberpunk we're all familiar with. "We grew up on genre staples such as *Blade Runner* and *Ghost in the Shell*, each with their own variation on an alluring and terrifying cyberpunk future," says Tian Chao, producer on the game. "At the same time, growing up in China during the ongoing economic boom has given many of us the feeling of actually living in such a world of towering skyscrapers and hazy neon streets! Even Denis Villeneuve, director of the recent *Blade Runner* sequel, referenced Beijing during the design of that movie. This gives us a unique perspective to inject into the now familiar cyberpunk vision."

Bringing this vision to life hasn't been the smoothest of journeys for developer Palm Pioneer, though, as the simple fact is there are more constraints and roadblocks when it comes to making games in China. Governmental controls and vetting procedures mean there can be stifling effects on the output of a studio, and the studio is no stranger to this impact.

"The development of the game has not always been smooth," Chao explains. "Under the constraints of both development time and local Chinese policies, we're constantly trying to inspire the team's creative vision, overcome any bumps in the road that we meet, while also polishing the quality of the game and trying to get the name out there."

But while Chinese government censorship is the go-to reason for the problems a studio might face, it's actually more nuanced than that: "During the game development process, the reconciliation between Eastern and Western cultures and the overseas promotion all posed a challenge for us," Chao says. "Fortunately, we received a lot of support and help from various parties, and we have successfully participated in international game exhibitions such as PAX and Gamescom. The positive feedback and comments players provide at these events greatly increased our confidence and drive during development."

It also helps that *Tales of the Neon Sea* looks fantastic, while its familiar sci-fi setting means it's universal enough to break down cultural barriers. "In the game," Chao explains, "players will find many puzzles and stories with oriental characteristics, but the oriental elements are only part of the *Tales of the Neon Sea* world view. Other elements in the game also occupy a very important position. We believe that the integration of these elements creates an interesting and unique experience."

⌄ **Seedy and grimy, yet at the same time bright and cute.**
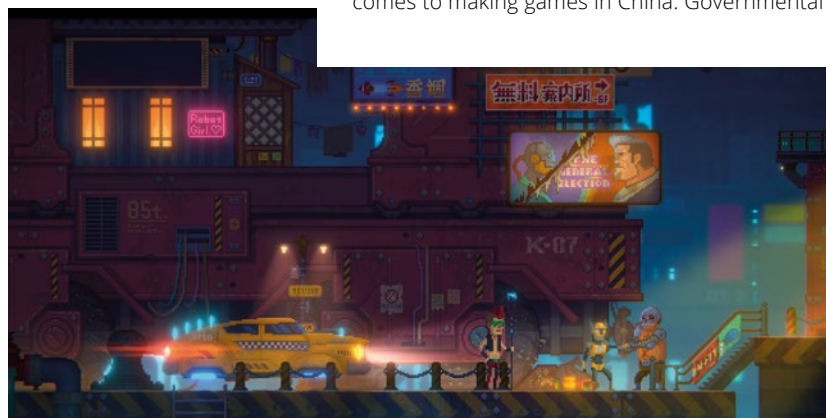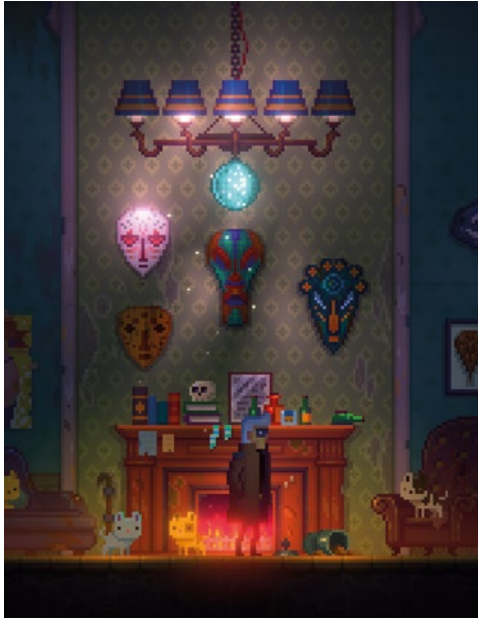
⌃ *Neon Sea* takes in all manner of eclectic influences.

Admittedly, a neo-noir cyberpunk adventure game telling a mysterious story full of twists and turns might not sound like the most unique of experiences, but *Neon Sea* is still uplifted, both through its usage of pixel art and its superb use of lighting effects, which really bring the seedy, futuristic city to life.

All of this has been achieved using Unity, and the experience has been a positive one for Palm Pioneer: "In terms of Unity, it has been integral in turning our dreams into reality," Chao says. "The custom functions and cross-platform capabilities helped to accelerate multiplatform development, and enabled us to design more content for [the game]."

There's a lot of effort and optimism going into *Tales of the Neon Sea*, as well as a quiet confidence the game can do well on the global stage. It's a big change from the Chinese development scene of just a few years ago, as

Palm Pioneer is well aware: "A few years ago, most of China's gamers and development teams focused on mobile games, and a large number of so-called 'pay to win' games existed," Chao tells us. "In the past few years, the situation has begun to change, and many players and development teams are starting to pay attention to and get involved with PC and console games. Focus is now being placed on the gameplay and fun of the game itself, and many excellent titles have already emerged. So I think this is a good environment for an independent studio like ours.

"We are very optimistic about the future because we focus on doing what we love," Chao continues. "Making a good game has always been our team's dream, and we'll continue to strive towards this goal."

> **"We're optimistic about the future, because we focus on doing what we love"**

Tales of the Neon Sea releases in 2019 on PC.

⌄ Pixel art is often pleasing to see, but seldom rendered with such wonderful lighting.

Devil May Cry 5

We go hands-on with the latest entry in Capcom's demon-slaying series

Info

GENRE
Action-adventure

FORMAT
XBO / PS4 / PC

DEVELOPER
Capcom

PUBLISHER
Capcom

RELEASE
8 March 2019

The first *Devil May Cry* introduced a whole new style of action game back in 2001. Showcasing a more cinematic style of combat, the game ranked you on how ineffably cool you looked while smacking demons about left, right and centre. It debuted during the era of nu metal and *The Matrix*, and rode the same wave of balletic action and downtuned guitars. But since then, the likes of *Bayonetta* and Ninja Theory's excellent (although reportedly slow-selling) reboot of *Devil May Cry* have taken some of the thunder. It remains to be seen whether *Devil May Cry 5*, directed by series veteran Hideaki Itsuno, can still show off some new moves almost 20 years later.

We can't really divulge much about the plot except to say that it's set some time after the fourth game. Initially, you'll be taking control of Nero, nephew to the original demon slayer Dante; he plays like your bread-and-butter hack-and-slasher, but with a few new abilities

to mix things up. You can now charge up your gun to fire off multiple bullets to keep your combo going, but the biggest change is Nero's ability to change arms during combat. Bereft of his demon arm, Nero now has interchangeable prosthetics courtesy of mechanic companion, Nico. Equipped with a limited stock of limbs, Nero can either use his mechanical arm to mix up attacks in combos, or sacrifice it to remain on the offensive. There are several types, each with different properties – electric, explosive, and so forth. In echoes of Ninja Theory's *DmC*, Nero can now hook onto targeted enemies and, depending on their size, either drag them or the hero up close to launch a melee combo.

The general flow remains the same as previous games: take out waves of demons in the fanciest way possible to rack up red orbs, which can be exchanged for new abilities. One welcome addition, though, is that targeting an enemy will show you how much life it has before you attempt to pummel it to dust. There are also some light but welcome puzzling segments between each battle.

We also got to play as Dante for a few missions, and he handles as well as he ever has. There are multiple fighting styles that can be switched on the fly with the directional pad and shoulder buttons. This new ease of access means you can mix up your fighting style to your heart's content, tying together all manner of attacks into a pleasing display of violent gymnastics.

There's also a new character named V, who looks like Kylo Ren fronting a Linkin Park tribute band. Rather than attacking directly, V can

⌄ The photo-real look results in some sharply textured but repetitive backdrops.
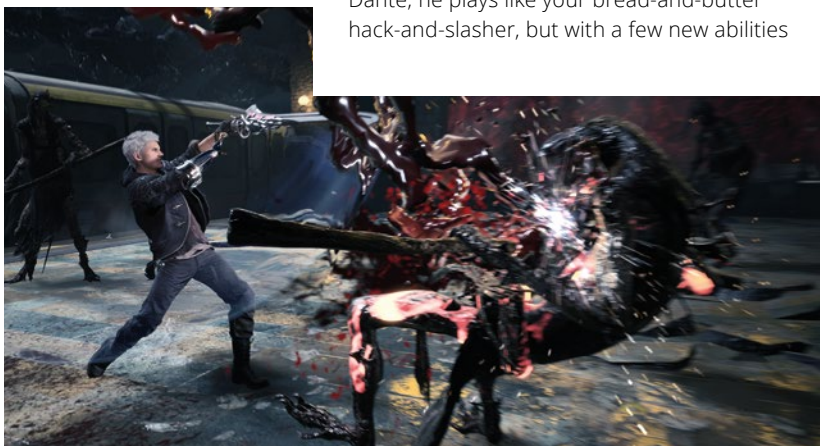
summon a giant shape-shifting panther for close-up attacks, and use his crow companion to shoot at long range. V will only get up close to finish off demons with his cane. It's an unusual approach, and one that takes a bit of time to really get used to, but once it does, it feels pretty satisfying.

Visually, however, *Devil May Cry 5* introduces an uneasy mix between over-the-top Gothic stylings and near-photorealistic environments, which produces a slight uncanny valley effect. Set mostly in a fictional city called Redgrave, the game strongly resembles London, with real-world landmarks recreated in Capcom's fancy RE Engine: red telephone boxes, for example, double up as checkpoints where you can buy new skills and supplies with the red orbs you earned in battle. The problem with these realistic environments, though, is that their blandness can lead to confusion. At one stage, I found myself lost in a hotel, unable to differentiate one corridor from another. Clicking in a thumbstick activates the destination view, which helps direct you to the next goal, but it seems more like a sticking plaster than a fix.

Devil May Cry 5's biggest issue, at least in the demo build we tried, is that the combat feels somewhat weightless. As Nero's motor-powered sword slices through demons, there's no sense of connection; no satisfying crunch of steel against bone. There's still space for experimentation within combos – chaining attacks together has long been the series' high point – but the insect-like demons at the

> ### "There are multiple fighting styles that can be switched on the fly"

beginning provide little challenge, and it's not until a boss appears at the end of the mission that the demo really begins to engage.

The sequel has plenty to discover in terms of new characters and attacks, but going on what we've played so far, it's perhaps too close to the PlayStation 2 entries in its structure. Moments between combat just aren't that inventive when compared to *DMC*, which turned environments into mini puzzles for you to traverse. More confusingly, there's a moment of nudity so gratuitous that it soured the experience; for a series that never took itself seriously, it's a moment that jars, especially given its context.

As a long-time fan of the series, I remain optimistic that the final release will pleasantly surprise me overall. The lack of satisfying feedback from attacks is cause for concern, but ultimately, our hands-on only represented a small slice of the finished game. Here's hoping *Devil May Cry 5* can live up to the series' proud legacy. ⓦ
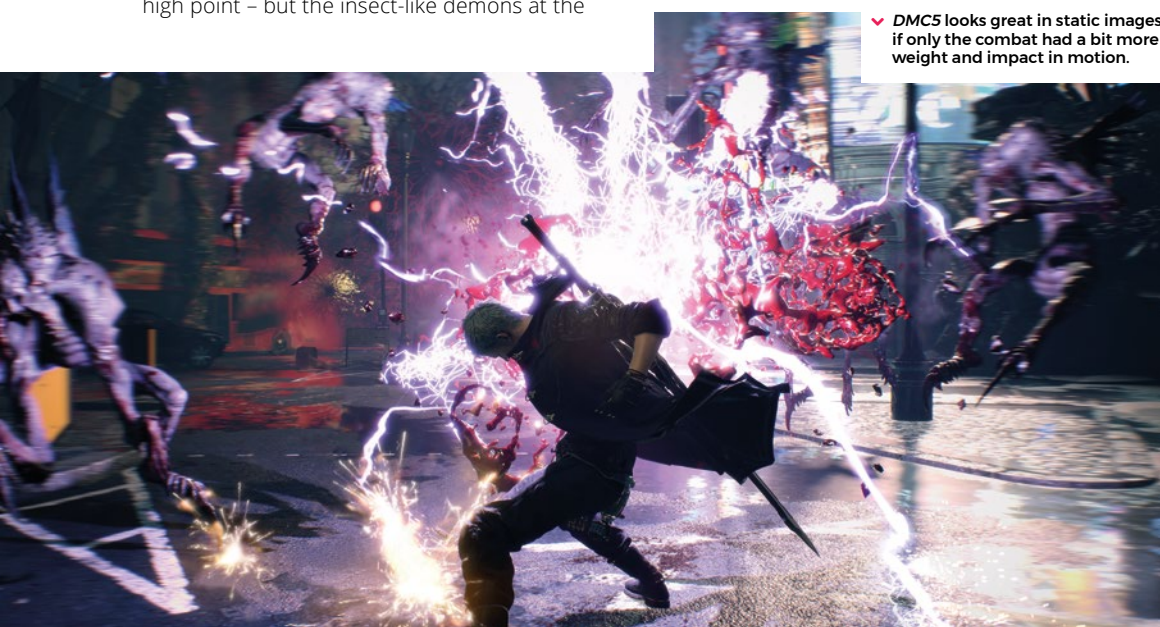
RE-START YOUR ENGINE

Devil May Cry 5 is another great showcase for Capcom's new RE Engine. First seen in *Resident Evil 7* and more recently the excellent *Resident Evil 2* remake, the RE Engine gives *DMC5* the most open environments yet seen. Redgrave (not London, honest guv) features a variety of locations: crumbling streets, demonic structures and benighted rooftops, which provide some really nice vistas. The character models in *DMC5* are also worth a mention: Nico in particular is really expressive. The three games mentioned here may lie in very different genres, but the engine steps up to the unique demands of each title.

Headlines
from the virtual front

03

02

01. Direct from Nintendo

Sitting around the fireplace together and taking in what Nintendo has to offer in its most recent Direct broadcast has become a rite of passage for owners of the venerable manufacturer's hardware, and February's edition was no different. Ninty announced two huge Switch releases, *Super Mario Maker 2*, sequel to the superb DIY Mario title of 2015, and *The Legend of Zelda: Link's Awakening*, a remake of the fantastic Game Boy original. Both are set for release later in 2019, and we intend to have our grubby mitts all over them.

Details for other titles we already knew about followed, including the remaster of *Assassin's Creed III* and a Switch version of *Hellblade: Senua's Sacrifice*. But possibly the most exciting element, though, was the launch of *Tetris 99* on Switch; a free battle royale version of Tetris. It sounds strange, but it's absolutely compelling.

01

02. Baldur's console

In news sure to delight those of us who like fun things and the opportunity to move around with said fun things, developer Beamdog has teamed up with publisher Skybound to bring a range of classic CRPGs to console. What this means is that for the first time, console owners will be able to play the likes of *Baldur's Gate*, *Neverwinter Nights* and *Planescape: Torment*.

For those unaware, that's three of the best games ever made – and they'll be available on Switch, so portability comes into play in a big way. We don't want to editorialise too much here, but it's difficult not to be excited. Release dates weren't announced at the time of writing, but it looks like the games will show up in 2019.

03. Starbreeze's woes continue

The ongoing saga of Starbreeze's difficulties – essentially entering bankruptcy proceedings towards the end of last year – has seen one of the publisher's titles passed back to its licence holder. *System Shock 3*, currently in development at OtherSide Entertainment, was sold back to the developer early in February. Starbreeze had picked up the publishing rights back in 2017 and had, since that time, invested over £9 million into production of the sequel to the cult classic sci-fi RPG/FPS hybrid. The company said it expects to fully recoup costs associated with the investment, though the price of handing publishing back to OtherSide has not been revealed.

"I believe this is the best solution for us, although it is sad that we cannot complete the project with OtherSide," said Mikael Nermark, acting CEO at Starbreeze.

Renowned Atari ST programmer Steve Bak has died aged 66

Capcom adds dozens of classic game soundtracks to Spotify

04

06

05
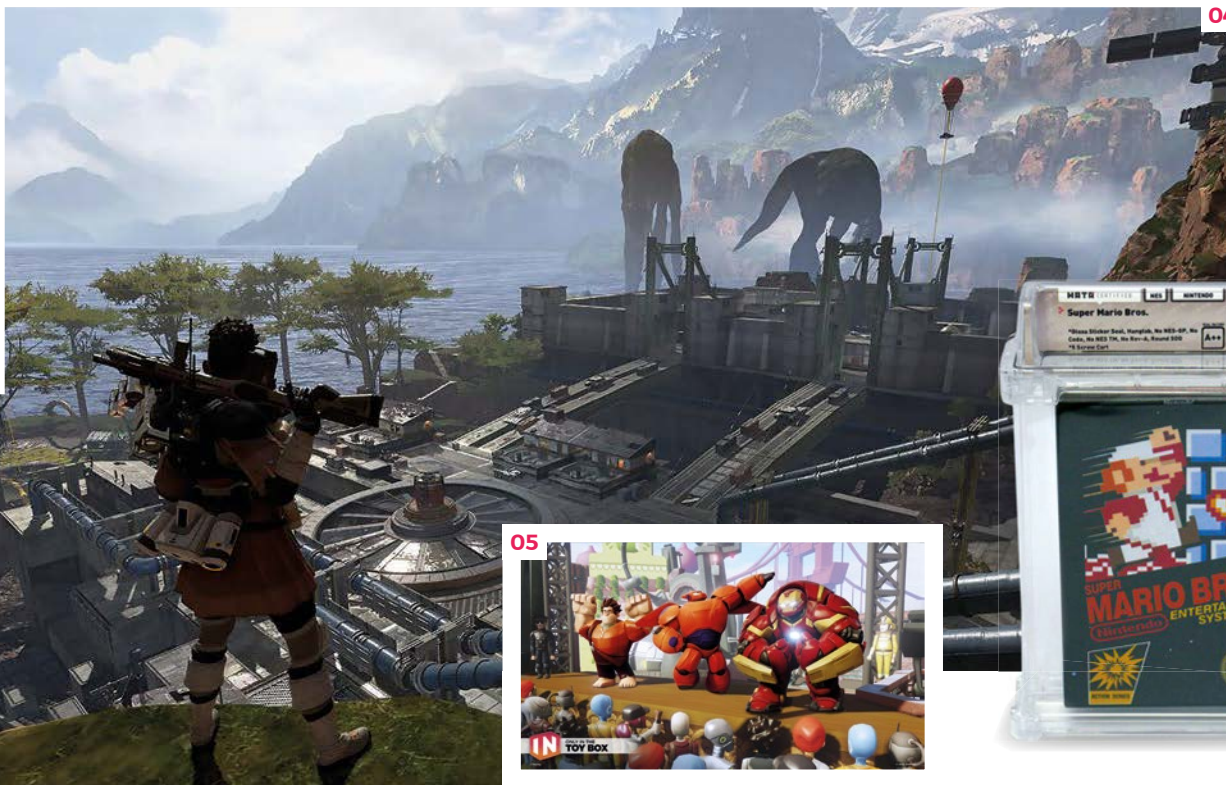
04. Surprise Legends

EA and Respawn pulled a bit of a fast one on the whole world by launching a surprise spin-off of the *Titanfall* franchise, *Apex Legends*. This free-to-play FPS incorporates elements of MOBAs, and slots neatly into the ever-growing pantheon of battle royale titles, slapping players into teams of three and having them battle (royally) to the last team standing against up to 57 other players (or 20 teams total).

While the surprise launch does mean work on *Titanfall 3* has been put on the back burner for the time being, *Apex Legends*' immediate success – 25 million players in its first week – has bumped the player count in Respawn's previous, the excellent *Titanfall 2*, up to higher levels than they have been.

05. Disney: 'We're not great at making games'

In a quarterly earnings call, Disney CEO Bob Iger made a surprising admission to those listening: he's never considered the company to be all that good at making or publishing video games. "We've tried our hand in self-publishing," the chief said. "We've bought companies. We've sold companies. We've bought developers. We've closed developers... And we found over the years that we haven't been particularly good at the self-publishing side."

Instead, Iger confirmed Disney was to remain firmly in the camp of licensing out its properties, like *Star Wars* to EA, and focusing its creative juices on the things it is good at – theme parks, cruise ships and making movies.

06. Super (expensive) Mario

A sealed, early print run copy of the original *Super Mario Bros* on NES has sold at auction for a whopping £78,000, purchased by a group of retro gaming collectors who apparently didn't know the game came bundled in for free with plenty of NES consoles.

Okay, Nintendo may have reprinted the original *Mario* adventure repeatedly between its 1985 launch and 1994, meaning it's not in the slightest bit a 'rare' game, but this version stood out as an early version printed for test launches of the NES in Los Angeles and New York, complete with absolutely pristine condition box, sticker and even shrink wrap. So probably worth more than the £2 you pay on eBay for regular cartridges.

THQ Nordic/Koch Media acquires Kingdom Come developer Warhorse Studios

This image appears online; haunts dreams

Incoming

Sayonara Wild Hearts ⬆

From Swedish developer Simogo comes a truly manic-looking 'pop album video game' that looks like a cross between *Child of Eden* and an acid trip in a trendy nightclub. There are people on motorcycles, flashing colours and a banging soundtrack; *Sayonara Wild Hearts*' creators cite *Akira*, *Gradius* and *OutRun* as influences, so we're on board already.

Release date: TBC 2019

The Last Night ⬆

Neon signs and flying cars are well-worn cyberpunk staples, but *The Last Night*'s deployment of them is quite stunning. It's a cinematic platform game that mixes pixel art and 3D environments to eye-catching effect, while its platform-adventure action feels of a piece with such nineties genre offerings as *Flashback* and *Beneath a Steel Sky*.

Release date: TBC

Necrobarista ⬆

We need at least two cups of coffee to get us going in the morning, and as it turns out, so do dead people. Set in Melbourne, *Necrobarista*'s about a cafe where, for one night only, the deceased can enjoy a final hot beverage with the living. It's a visual novel with a clean anime look – joined, we hope, by a mystery plot that's appropriately rich and dark.

Release date: TBC 2019

Pathologic 2 ⬇

More open-world horror, this time about a remote town in the midst of a terrifying plague. The game itself has been affected by a different kind of crisis; in a late 2018 blog post, studio Ice-Pick Lodge revealed that economic problems in its home country of Russia had affected *Pathologic 2*'s early development, and that it will now be released in three parts. The first game was an overlooked, inventive chiller; as such, we've high hopes for this sequel.

Release date: TBC 2019

Manifold Garden ⬆

Developer William Chyr studied physics at school, and briefly worked at a nuclear physics institute in Italy before he got into art and then game design. We mention this because it helps explain why *Manifold Garden* looks so extraordinary: like the work of M.C. Escher, it fuses art with mind-bending mathematics. At its core, Chyr's game is a first-person puzzle game where you explore looping webs of impossible architecture; one of the central challenges, as we understand it, is figuring out how to traverse the game's shifting gravity fields. What looks like a wall from one angle can look like a floor from another; it's like *Inception*, but without Hans Zimmer's parping soundtrack or Michael Caine. "I see the game as a metaphor for the last 400 years of physics," Chyr explained at a recent GDC talk. "At the beginning you're learning how gravity works, and then by the end of it, you gain an understanding of the shape of the universe." It's a bit more involved than your typical match-three puzzler, then.

Release date: TBC 2019

Among Trees ⬎

Sick of the daily grind, the noise, the smog, the ubiquitous posters for West End musicals? Then let *Among Trees* take you back to the refreshing simplicity of nature: fulfil all your dreams of living in a log cabin, felling trees, and growing herbs in a window box. It's another survival game, essentially, but with a more relaxed vibe than most. There is a grumpy-looking bear wandering about in the woods, though, so we probably shouldn't get too complacent.

Release date: TBC 2019
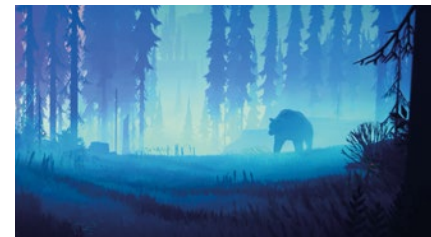
Chernobylite ⬎

The 1986 Chernobyl disaster has long held a curious sway over game designers, with such disparate titles as *Atomic Runner*, *S.T.A.L.K.E.R.* and *Call of Duty 4* all referencing the incident to some extent. So does *Chernobylite*, currently in development at The Farm 51, the Polish studio who brought us the VR action thriller, *Get Even*. It's a free-roaming horror game that takes place in a fantastical version of the abandoned Pripyat – this one's full of angry monsters and creepy dolls with green, glowing eyes. Interestingly, the developers have travelled to the real Chernobyl exclusion zone to gather images and research, so the setting should feel grounded in the real world, even as hellish things come crawling out of the ether.
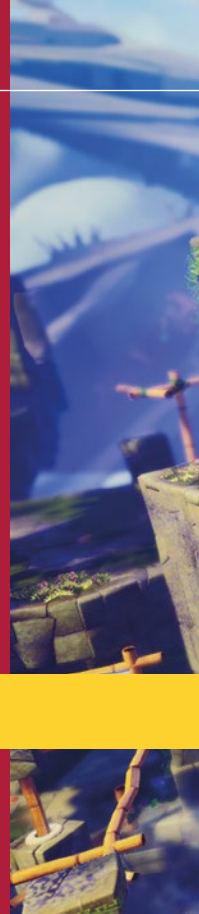
Release date: TBC 2019

STAYING INDEPENDENT

THE EVOLUTION OF SUMO DIGITAL

How collaboration and adaptation turned the Sheffield-based studio into the UK's largest independent developer

△ *Snake Pass* recalls the colourful 3D platformers of the N64 era, but plays quite unlike anything else.

⌄ **Unsurprisingly, Sumo's success with Sega's IP in kart racing finds the studio back in the driving seat for the upcoming** *Team Sonic Racing*.

ndependence can mean different things to different people. In game development, it can turn into its own genre or studio philosophy. You hear stories of industry veterans tiring of the triple-A machine striking out on their own, or the DIY hobbyist creating something quirky after many years in their own time and out of their own pocket. A studio with over 500 staff, who not only work on big licences but also with some of the biggest publishers in the business, then might not immediately strike you as emblematic of the independent spirit.

"For us, independent isn't defined by the type of games we make. It means we're not part of a publisher," says Paul Porter, Managing Director of Sumo Digital, one of the biggest independent game developers in the UK. "Independence gives us the freedom to work with a wide range of publishers across all genres and platforms. Our versatility and flexibility are skills that we've developed by being independent."

Yet it's also likely you won't have heard of the Sheffield-based studio, as it seldom gets the same attention as, say, a celebrated in-house team at a major publisher or a tiny, trailblazing indie. Instead, Sumo occupies a position between these poles: it's a studio-for-hire. Often overlooked, these developers are nonetheless essential to the industry. If a publisher holds an

exciting and lucrative licence, or a game needs to be ported to a wide variety of platforms, or a troubled production needs more hands on deck, a dependable studio-for-hire is who they turn to. With over 15 years of experience, Sumo has become one of the safest pair of hands in the industry.

MADE IN SHEFFIELD

Sumo was founded in 2003 by Porter and a few other former staff at Infogrames Sheffield, though you might be more familiar with its former guise as Gremlin, who had been around since the eighties and also gained acclaim for the *Sonic the Hedgehog*-inspired platformer, *Zool*. The perils of acquisition, however, often mean a studio's fate is outside its control. In 2003, the French publisher Infogrames closed many of its studios, including Infogrames Sheffield. Sumo essentially arose from those ashes, and it was perhaps no surprise that one of its goals was not to repeat the same experience.

"We learned a lot from our time with Infogrames and we wanted to create a stable video game developer that had the flexibility and opportunity to work on different titles," says Porter. "At the same time, we had all worked for a large publisher for the previous decade, so we also understood their ➡

‹ While *OutRun 2* was Sumo's first breakthrough title, its first game was actually a rather forgettable football title for Codemasters.
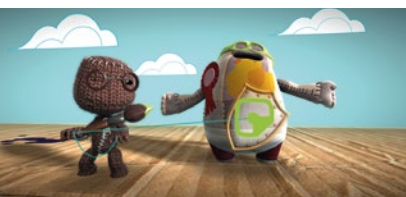
∧ While taking the helm of *LittleBigPlanet 3*, Sumo were also responsible for creating DLC for its predecessor.

∨ Paul Porter, Managing Director of Sumo Digital.

demands and requirements, and had a very collaborative approach."

Instead of the plucky upstarts trying to build a new IP from scratch, then, Sumo's approach would see it fostering collaborative long-term relationships with clientele, representing some of the industry's biggest names, to work on their existing properties, all the while remaining independent. Today, it's not just its Sheffield headquarters but rather a network of studios around the UK, working on a variety of titles from action-adventure *Crackdown 3* for Xbox One to an FPS set in the *EVE* universe, *Project Nova*.

ROLLING START
Before amassing the diverse portfolio it enjoys now, Sumo carved out something of a specialty in the arcade racing genre. Its big break was bringing Sega's arcade sequel, *OutRun 2*, to Xbox. The port was a huge success, which led to the studio also being responsible for porting the updated *OutRun 2 SP* to PS2 as well as reimagining the first game, merged with the sequel, which became *OutRun 2006:*

Coast 2 Coast. The latter's challenge wasn't just reproducing an arcade-quality experience for the home but also for the PlayStation Portable, for which Sumo would have to develop its own fully customised engine.

This established the studio's reputation as a reliable go-to, not just for arcade racers, but also with the expertise to adapt console hits for portable, which included PSP ports of *TOCA Race Driver 3* and *Split/Second*. More importantly, it was the beginning of a beautiful friendship with Sega. "Since *OutRun 2*, the Sega relationship has been very important to us," Porter explains. "We wanted to repay the faith that Sega showed in us at every opportunity."

That opportunity meant a diversion from arcade racers to arcade sports with the *Virtua Tennis* franchise, which led to work on *Sega Superstars Tennis*, a simple but fun game that was better known for being an excuse to play with Sega's wonderful cast of characters on the court. But if that game hasn't stood the test of time, then it was certainly a stepping stone to one of Sega's most successful forms of fanservice: *Sonic & Sega All-Stars Racing* and its sequel, *Sonic & All-Stars Racing Transformed*.

Once again leveraging its arcade racing expertise, Sumo was able to create a worthy competitor to *Mario Kart*. Aside from developing a slick kart racer, *All-Stars Racing* also gave Sumo the licence to work with iconic characters across Sega's history, from *Jet Set Radio* to *Shenmue* to *NiGHTS into Dreams*, all of which required collaborating with Sega's many different departments.

"Publishers and developers have to take risks with every title, so being entrusted with a triple-A game or renowned IP is a big responsibility," says Porter. "We've built up a lot of knowledge and experience from Sega over the years." Even though the *All-Stars* brand belongs to Sega, Sumo has managed to make the series its own.

MORE THAN JUST RACING
Recent years have also seen Sumo diversifying beyond the racing genre, from a sojourn into fitness on Xbox One to collaborating with IO

⌃ Sumo acquired indie studio The Chinese Room, creators of *Everybody's Gone to the Rapture*, in August 2018.

Interactive on maps for its sandbox franchise, *Hitman*. But when a studio-for-hire acquires a reputation for one genre, how does it get the chance to flex its muscles in other areas?

"Sometimes, in building a new relationship, or working on a different genre, we might work on a small part of a game – proving that we have the skills and expertise – and then progress onto a bigger project," Porter explains. "In other cases, we create prototypes that clearly demonstrate the technical and creative skills of the people here, and can develop a project from conception through to release."

Games are, after all, an evolving medium, and as such, individual developers at Sumo are constantly learning and sharing experiences. "I think it's in a developer's DNA to want to push themselves to not only be the best they can be, but to learn and try new things," says Porter. Equally important in diversifying the studio's capabilities is recruiting new talent, from seasoned genre specialists to graduates, to promising members of modding communities.

The latter was key when Sumo led on the development of *LittleBigPlanet 3*. Taking the reins from original creator, Media Molecule, it was the studio's first major breakout from its racing stable. Suffice to say, taking charge of one of Sony's biggest franchises was a huge responsibility. As one of the senior designers, Brad Davey, recalls, "That was a difficult franchise to get your head into, but Sony gave us that time to get our heads around it." It was also invaluable that one of the requirements was for Sumo to hire experts from the community to help boost that understanding, which was how Seb Liese joined the team as a level designer.

Working as a teacher at the time, Liese had dreamed about making video games ever since the days of Maxis's ambitious life simulation

game, *Spore*. "I heard that people who were very good at *LittleBigPlanet* were getting hired, so I bought a PlayStation and *LBP* and just started creating hours and hours a day to try and get a job through that," he tells us. It certainly paid off, as his presence in the community caught Sony's attention, who sent him to Sumo. Davey recalls looking over Liese's levels at the time: "He broke the game to make things in *LBP2* that we didn't think were possible!"

It can be all too easy to overlook the work of a studio-for-hire when compared to the visions of an in-house team or the freedoms of a small, passionate indie, especially if the studio's bread and butter involves converting existing games. But as Porter emphasises, "Creativity within constraints is still creativity."

"It's definitely still very challenging, and there's a lot of options to be creative within a limited space," Liese adds. "You get a franchise, you have certain rules you have to follow, you have this little box you have to stay within, but I find it very fun to be as creative as possible within those boundaries."

An outsider having the opportunity to take the lead and go wild with an established IP, whether *LittleBigPlanet* or *Sonic & All-Stars*, illustrates just the kind of creativity that can be achieved. For Davey, it's all about building trust. "Sony trusted us to not go off the rails, but do some unexpected stuff with *LBP3*, like the Adventure, Inventory and hub world, and the whole sense of adventure we tried to instil in that game, because that wasn't really what ➧

> **"Publishers and developers have to take risks with every title, so being entrusted with a renowned IP is a big responsibility"**

CHINESE WHISPERS

Sumo's independence hasn't prevented its parent company The Sumo Group from bringing other studios into its fold. But it came as a surprise when last year it acquired The Chinese Room, the award-winning developer of *Everybody's Gone to the Rapture*. Paul Porter assures us, however, that it's not being absorbed or rebranded. "We're committed to continuing creating original IP, which The Chinese Room has a proven track record for," he says. "We want them to retain their identity and vision. Being part of the Sumo family just means they'll have the support and resources they didn't have before."

⌄ *TOCA Race Driver 3* was among many of Sumo's PSP racing ports.

IT'S THE LITTLE THINGS

Self-publishing *Snake Pass* brought its own challenges. Perhaps the most unexpected involved a patch which changed the game's icon on the Switch home screen to an image of a snake's head. The presentation drew ire from Switch owners unenthused with what looked too much like a mobile app – an issue other third-party publishers continue finding themselves falling foul of. "I think we were ground zero for that," Brad Davey recalls. "Stuff like that is usually outside of our remit, but now it was all on us. It seems like such a trivial thing, but we had so many conversations about that!"

▼ Seb Liese, Senior Designer at Sumo Digital.

that franchise was at that point. If we build that trust, we get a lot of creative freedom to bring out a personal touch."

INDIE PASS

Yet both Liese and Davey would taste first hand the freedom and challenges of making an 'indie' game with a little 'i'. In 2017, Sumo released the innovative 3D platformer, *Snake Pass*, which began from an idea from Liese. As one of the design leads, Davey recalls how there had been talk for some time among the top brass about creating an original IP. "When Ian Livingstone joined Sumo as a new board member, he really pushed forward with that idea," he says. "And that led to us finally doing our first game jam, which we've wanted to do for years."

The game jam was held in 2015, providing an opportunity for the studio's talented developers to experiment with their own prototypes in the hope that one would have the potential to become a full game. By then, Liese had also been given some time to learn Unreal Engine 4. Naturally, it helped that he used to own a pet snake (aptly named Solid Snake), but the game's concept emerged from a happy accident that occurred while trying to make a physical rope in UE4. "While creating that rope, at some point I

forgot to attach it to the ceiling. I saw it fall on the floor in this nice turd shape in a simple satisfying motion," he recalls. "When I saw that, I wondered if I could control this rope as a character. And that was the first step of *Snake Pass*."

After encouragement from other enthused colleagues, Liese submitted the idea to the game jam and won. Within a few months, the team had a working demo to present at the indie-focused Rezzed show in London. Based on the extremely positive reaction from the show crowds, it only took the weekend after for Porter to announce that Sumo would take *Snake Pass* into production.

Despite looking like a colourful vibrant throwback to the 3D platformers of Rare's yesteryear, the game was also innovative thanks to its tricky mechanic: learning how to move (and 'think', as the tagline went) like a snake around different environments. Essentially, *Snake Pass* was a 3D platformer where you couldn't jump, though it also took inspiration from modern experimental indies that delighted in frustrating players with unwieldy controls. "We definitely looked at *Octodad*, but we wanted it to be the *Octodad* which you could also perfectly master," explains Liese. "When you know the snake, you can 100

> ## "We get a lot of creative freedom to bring out a personal touch"

percent the game, you can determine exactly what the snake does at any time."

It's certainly unlike anything Sumo has ever had the chance to make, while the decision to self-publish also taught the company things it had never previously considered. "The self-publishing side was completely new to us," says Porter. "Having worked so closely with publishers over the years, we're used to hitting deadlines for shows and marketing plans, but doing all of this ourselves – registering on digital stores, designing our booths, the branding – certainly gave us a greater insight and appreciation of all the work that they do."

Although *Snake Pass* was a multiplatform release, the timing was also fortunate to make it one of the very first titles for Nintendo Switch: one running on Unreal Engine 4, no less. "Epic was pretty much making Unreal ready for making Switch games as we were trying to finish *Snake Pass*, so we had a lot of contact with Nintendo and Epic," says Liese.

The Switch may be on the lower end of hardware specs, but it's testament to Sumo's work that *Snake Pass* performs well on the platform, while on the opposite end of the spectrum, the team has also leveraged the power of Xbox One X to optimise the game for 4K.

"When we started development, the Switch wasn't known about," Porter tells us. "But within days of receiving it, we were able to leverage our cross-platform expertise and had *Snake Pass* running beautifully. That was one of those moments where years of experience and hard work by lots of people come together to produce something magical."

JACK OF ALL TRADES

The success of *Snake Pass* has naturally been a boost for Sumo. It was one of the first indie games to launch on Switch, topped the eShop chart across many territories, and got in early on the system's indie gold rush. After over a decade of working on other publisher's properties, it's also an IP the studio can proudly show off without the usual licensing restrictions.
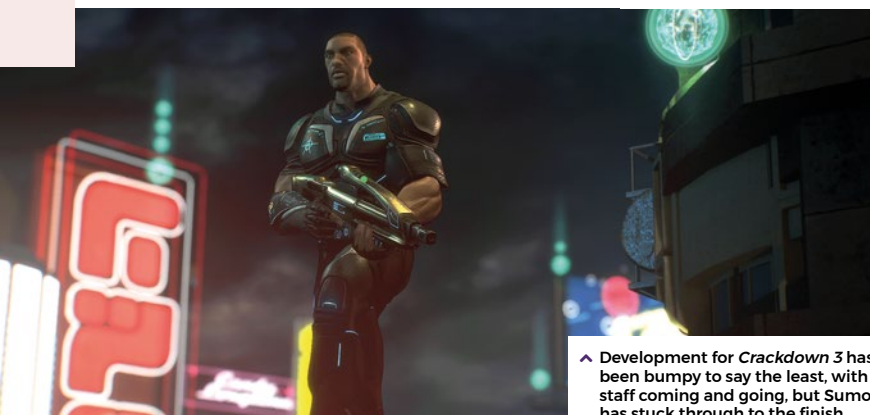
Whether all this means Sumo's interested in pursuing the smaller, indie side of development further remains to be seen. Davey mentions that there have been other game jams since, but it's more likely that new, original IPs will become the remit of latest member of the Sumo family, The Chinese Room (see 'Chinese Whispers'). Porter, meanwhile, is keeping the company's plans close to his chest. The upcoming slate for publishers like Sega, Microsoft and CCP Games, on the other hand, suggest that it's business as usual for the studio-for-hire.

If anything, making smaller games might just be simply one of many pies Sumo has its fingers in. A recent acquisition of the shuttered CCP Newcastle – the developer behind early virtual reality hit, *EVE Valkyrie* – may even be a hint that it's considering adding VR to its portfolio. Whatever the future, then, Sumo's versatility should mean that it'll remain independent – and in demand – for a long time to come.

"Just like the industry as a whole, we're never going to stand still – whether it's actively looking for new opportunities or reacting to opportunities that come up," says Porter. "We've laid solid foundations, and will stay true to our roots of working with great publishers on great games." ⓦ

SUMO INDIA

Sumo may be a British company, but in 2007, it also founded a studio in India. Although much of the country's development talent tends to be used as outsourced labour for big-budget studios in the West, the team in Pune is an integral part of Sumo's operations, collaborating with its sister UK studios across multiple projects. Sumo India may not have led on creating its own title, at least as yet, but it certainly consists of talented coders, artists, QA testers, engineers and designers that may conceivably do so in the future.

> Sometimes a studio-for-hire is called in to assist with a discrete part of a game. In *Hitman 2*, Sumo was responsible for the Colombia map.

˄ Development for *Crackdown 3* has been bumpy to say the least, with staff coming and going, but Sumo has stuck through to the finish.

Interactive

Defying physics in Just Ski

We catch up with Jeff Weber, who makes physics-based games in his Wisconsin basement

It was about 15 years ago when Jeff Weber first began looking around for a new hobby. He'd just moved from his birthplace of Michigan to Wisconsin, and wanted to find something creative to do in his spare time; at first, he purchased a radio-controlled helicopter, but found that it somehow didn't fill the gap. "I got the helicopter, and it was in a million pieces," Weber recalls. "I found no desire to put it together."

Already a developer of business apps by trade, Weber began to think instead about game design. Partly inspired by the 2001 browser title, *Ski Stunt Simulator* (see boxout), Weber started making his own 2D physics-based games in his spare time, each one building on the last. In the 2008 browser game *Diver*, the player uses a diving board to bounce a stick figure to a location in the water marked out by two buoys; assuming the stick figure doesn't make a fatal collision with a cliff, the player's given a mark out of ten for accuracy. In the 2014 smartphone title, *Krashlander*, the player guides a fearless skier around a silhouetted obstacle course of platforms and cliffs, essentially using the skier as a wrecking ball against alien invaders.

Weber's evolving brand of physics games are, he says, a challenging yet enjoyable way of giving his maths skills a workout. "The math is the biggest challenge," Weber tells us, "but it's also what makes physics-based game development so fun. I have a Bachelors degree in mathematics and I didn't really start using my math knowledge and skills until I started making games. I will sometimes spend days scratching math equations out in my notebooks trying to get something in the game to behave the way I want. It's very rewarding to finally figure out the math behind some game mechanic you have in your head."

> **"I will sometimes spend days scratching math equations out in my notebooks"**

FINE-TUNING

Like so many indie developers, Weber now uses Unity; before that, he made several games in Truevision3D and Microsoft's Flash competitor, Silverlight – *Diver* and its sequel, *Diver 2*, were both built in the latter. "Once Silverlight died,

^ *Krashlander* sees a lone skier attempt to thwart an alien invasion. Mostly, though, the skier tends to fall into bottomless pits.

I finally moved to Unity," Weber says. "I had my eye on Unity for quite a while and it turned out to be a perfect fit. It had a physics engine built in, supported C#, and was just generally easy for me to get things done."

One of the things that keeps games like *Diver* and *Krashlander* engaging is the way the little stick figure character doubles up and flails its limbs as it collides with the scenery. The animation is handled by the physics engine rather than drawn by hand, but that doesn't mean Weber doesn't spend time adjusting and fine-tuning things behind the scenes.

"All the sprites are animated using physics," Weber explains. "All the body parts are rigid bodies, and they are connected together with various joints. Then I write the math code to make them control the way I want. I do spend a *lot* of time tweaking physics parameters to get things to feel just right. It's just part of the deal when making physics games."

With game development still a hobby, Weber only has an hour or so each day to work on his projects; if he's at work, he'll often find himself reaching for a pen and paper to jot down ideas as they pop into his head. ("I'm patient," Weber says, "but man, I wish I could spend more time making games.")

Weber's most recent game is *Just Ski*, which offers a further twist on the tumbling physics of his earlier titles: by sliding down on the screen and then flicking up just before the skier performs a jump, the figure will crouch and then, if the timing's right, perform a satisfying mid-air spin. It's a mechanic Weber discovered through experimentation, and it's an idea he wants to iterate on further in future games. Since *Just Ski* came out last year, however, Weber has seen growing competition in the places where he sells his games, and he's now thinking about turning to a publisher to help market them to a wider audience.

"The last year or so, since the release of *Just Ski* on Steam, has been interesting," Weber says. "A lot has changed in the indie scene and competition is crazy... Marketing is my least favourite part of game development, and the competition is greater with every release. I am seriously considering a publisher with my next game. I'd suggest others consider it as well, given the current state of the industry."

Getting an indie title noticed may be tough these days, but Weber says he'll still be sending his hapless stick figures flying over vertiginous drops for a good few years to come. "I doubt I'll ever move to far away from physics-based games," he says. "I just have too much fun making them." ⓦ

SKI SUNDAY

Developed by Michiel van de Panne and Cedric Lee in 2001, *Ski Stunt Simulator* was a browser-based game programmed in Java. It made clever use of mouse gestures to alter its skier's stance; this, combined with a range of different courses and challenges, resulted in an absorbing physics simulation with some truly bone-shattering mishaps. "I've been fine-tuning, tweaking and modifying this style of control ever since I started making games," Jeff Weber says of the relatively obscure title that first inspired him. "*Ski Stunt Stimulator* had and still has a huge influence on my games."
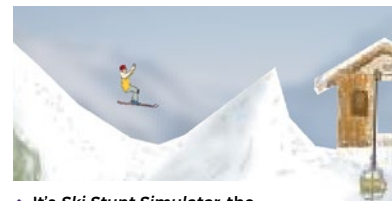
^ It's *Ski Stunt Simulator*, the 2001 browser game that sparked Jeff Weber's hobby in the first place.
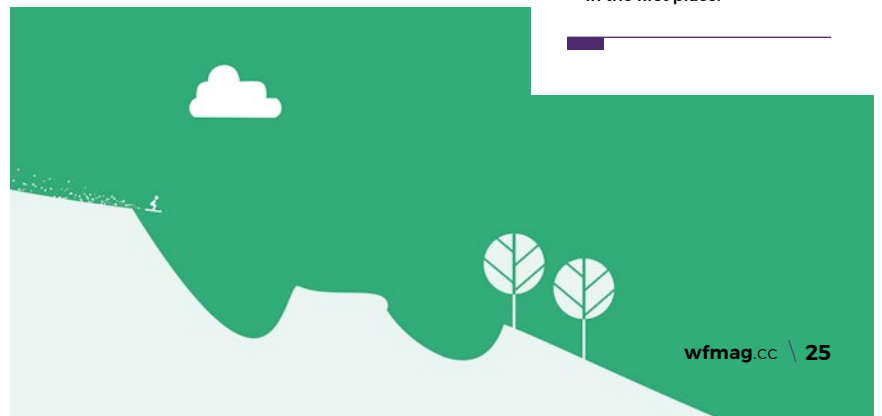
Adverts? Badverts, more like

STEVE MCNEIL

Steve is a partnered Twitch streamer who can't quite understand what all the fuss is about.

At the end of last year, Twitch began to run adverts for a New Year's Eve livestream at Times Square, hosted by one of the site's most successful streamers, Tyler 'Ninja' Blevins. Ads are nothing new on Twitch, but this one was different. Rather than promoting a new game, it promoted a livestream by Twitch themselves. What really got people's backs up, though, was that it ran these ads, which featured Ninja, on rival streamers' channels, including high profile ones such as Dr Disrespect and BikeMan.

If you're not into Twitch, you'll just have to take my word for it that those are real people and that, despite your gut instinct, they are legitimate, successful personalities in their chosen field. BikeMan (I know, I know) took to Twitter to claim the ads were a "direct conflict of interest". Dr Disrespect did the same, telling Twitch, "Don't ever do it again. Ever."

Twitch did do it ever again, ever. The very next month in fact, promoting a *Fortnite*/NFL Pro Bowl event hosted by another streamer, Imane 'Pokimane' Anys. Of course, Twitch is a highly competitive platform, with many people struggling to find an audience, and so anything which attempts to funnel viewers to a small elite group of streamers is going to cause frustration, but, as someone whose background is in comedy and, latterly, television, it's a frustration I struggle to empathise with. In comedy, if someone is funnier than you, they get more laughs and are more likely to get booked for more gigs. In television, it's accepted that, during the breaks for *Saturday Night Takeaway*, they're going to run adverts for *Dancing On Ice*. To my knowledge, Ant and Dec have yet to call out Jayne Torvill on social media. Although I would very much like to see that, please.
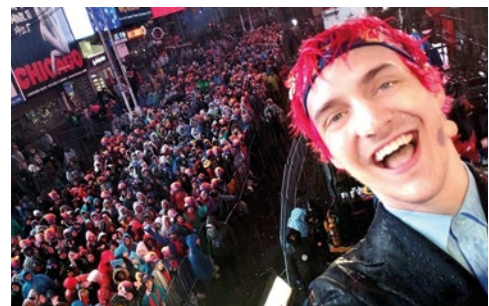
The common point I'm lumbering towards is that if you're good at the thing you do, people will watch your thing. If Dr Disrespect, a man who makes millions a year doing this, is worried about an advert for a show where a man with

⌃ Well, it pays more than designing levels for *Call of Duty*, which this man used to do…

red hair will unsuccessfully attempt to get people who've never heard of him to do a dance in the rain (YouTube search: 'ninja floss cringe'), he should perhaps focus on improving the quality of his output, rather than the nature of the ads on his channel from which he himself derives income.

Twitch have since promised not to run these sorts of ads in the future as a result of the backlash, which is indicative of the level of mutual dependency between the platform and its streamers. And yet, whilst the Amazon-owned Twitch is undoubtedly 'big business' and I'd hate to side with 'the man, if I'm honest, I struggle to feel too sorry for a millionaire who plays video games for a living and wears sunglasses indoors. Anyone that wears sunglasses indoors deserves everything they get. ⓦ

⌃ The irrepressible Tyler 'Ninja' Blevins. His New Year's Eve livestream lasted 12 hours.

> **"I struggle to feel too sorry for a millionaire who plays video games for a living and wears sunglasses indoors"**

Toolbox

The art, theory and production of video games

∧ Get to the chopper: squeeze a score attack
game onto the BBC Micro. See page 34.

∨ Find out how to make shudder-inducing
horror locations on page 28.
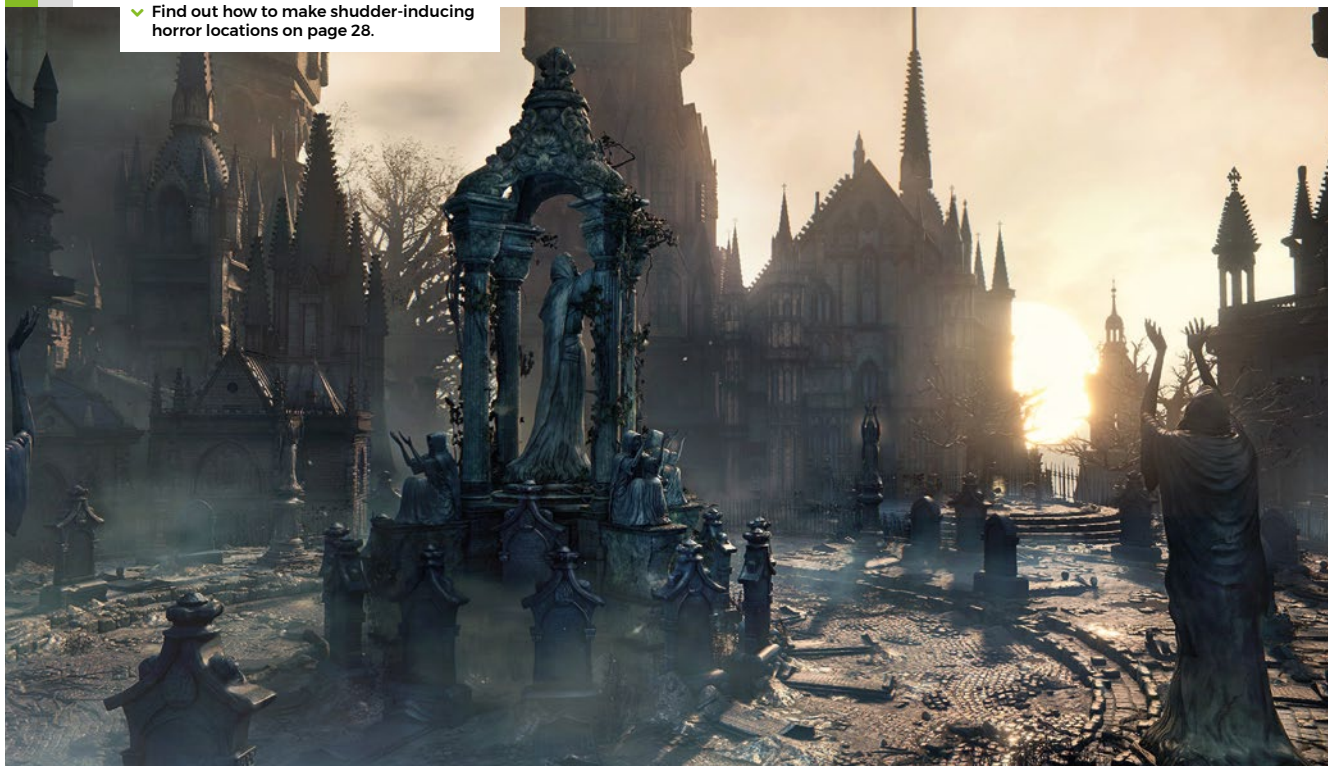
CityCraft: how to plan horror cities

Want to make your players shudder? Here are three tips for making a truly frightening horror setting

AUTHOR
KONSTANTINOS DIMOPOULOS

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

Confusing Geometries

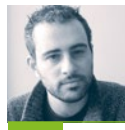Remember how director Stanley Kubrick's 1980 horror movie *The Shining* made you feel? Just how unsettling the Overlook Hotel was? Well, it wasn't all down to the sharp shadows, foreboding soundtrack and masterful cinematography. The hotel itself was wrong: its layout, architecture and geometry could never exist in reality, and this is a trick we can use to make our own cities subtly disturbing. An excellent video on *The Shining*'s spatial trickery can be found on YouTube: **wfmag.cc/overlook**

U sing environments to add texture to disturbing stories, and setting to scare people, is a far from new idea. The Romantics placed castles on remote cliffs; urban terrors have been hiding in generic metropolises since the 20th century; and contemporary horror games have all paid careful attention to their setting. Crafting a suitably atmospheric space is, after all, crucial when it comes to horror, and the richness of urban settings – their sheer genre potential – cannot be ignored.

Decaying provincial towns, dead civic husks, remote mountain villages, and vampire-infested cities can all create an atmosphere of dread. So if your aim is to make your location horrify, possibly confuse or unsettle, here are some helpful tips.

ELDRITCH, SCARY CITIES
A common technique is to exploit the familiarity players have with cities, and subvert their

expectations. Ignore key functions (think of a settlement without roads, for example), play with scale (impossibly large squares, and tall edifices), and strive for oddness; players are bound to notice if enough civic elements are missing or out of place. Horror writer H.P. Lovecraft masterfully introduced the oddness of the title town in *The Shadow Over Innsmouth* by mentioning its lack of a Chamber of Commerce and a public library. In game worlds, we have to make certain any purposeful omissions are both noticed, and not registered as something we overlooked.

Restricting visibility, and then revealing arrestingly majestic urban views – a technique city builders employed heavily during the Middle Ages – can enhance your city's atmosphere, and even set up jump-scares. Furthermore, a well-timed showcasing of the true scale of a city's weirdness can be profoundly unsettling. A panoramic view finally showing the city's pentagram structure, or presenting players with a burned-out wasteland surrounding the district they are in can be sublime.

Frogwares' forthcoming *The Sinking City* **is a unique example of urban focused cosmic horror. (Full disclosure: I worked on the game's pre-production for a few months.)**
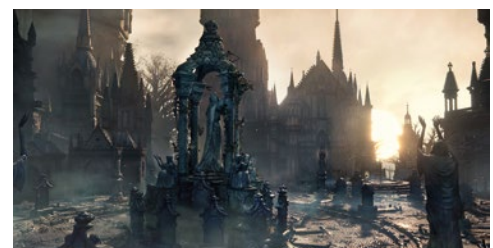
The Gothic altars and spindly towers of Yharnam show just how effectively architecture can evoke horror.

Few things are as scary as an abandoned or partially destroyed city; it hits too close to our sensibilities – think of the deserted Pripyat or war-torn Homs. We instinctively empathise with the plight of their former residents. We are reminded that death can be violent and undeserved, that desolation can remove all traces of our existence. Rusted doors, weed-infested parks, crumbling roads, rotting bridges and decrepit husks of buildings can be employed to that effect.

Constructing the identity of a place shouldn't exclusively rest on its built environment. The dark rumours, ominous legends and whispered hearsay surrounding it can conjure sinister images before players see anything. Venice, you see, is not just the town, but the myth and legend surrounding it. The songs, films, stories and texts capturing it. The Cradle level from *Thief: Deadly Shadows* is a fine example of this technique.

Other scary touches can include roads that disappear after an initial visit, or alleys that never appear on any map. Street art and graffiti can suggest supernatural threats, street names can change or simply sound evil, vehicles resembling hearses can set a solemn mood, and background screams or areas of absolute, unnatural silence can keep tensions high.

INVERTING CITY DESIGN IDEAS

Inverting the rules and goals of good city design is another great way of achieving unease. Go for the opposite of what city planners aim for, and stress agoraphobia, claustrophobia, confusion or even acrophobia. Force players into tiny confined spaces, ridiculously crowded squares, and make them walk narrow paths along deep chasms. Extremely large plazas, uncomfortably narrow sidewalks and confusing road networks are more subtle but also effective.

Constricted footpaths and alleys that barely allow a single person through (see **Figure 1**) can feel even worse with the help of urban fauna such as cockroaches, mice and wild dogs that can be shown or heard. Traditional proportions of road width to building height (often a cosy 1:2) can easily be broken with buildings too high or roads too wide. The intense verticality of a city, and the creation of urban canyons with leaning, unpainted houses will suggest danger. Add in winding routes, dead ends, steep ramps and stairs, dense vegetation and impractical landscaping, and you'll have a thoroughly uncomfortable environment.

"Go for the opposite of what city planners aim for"

And what about the land uses planners usually try to hide? Sanatoriums, jails, asylums, graveyards, junkyards and factories can all paint an area as – at the very least – unpleasant.

CREEPY ARCHITECTURE

A glimpse at *Bloodborne*'s Yharnam is proof enough that architecture can convey horror. Gothic and Gothic Revival buildings of black stone are packed with disturbing details, and their ill-lit interiors are perfectly eerie. Many cultures employed architectural elements and decorations to ward off evil (crosses, gargoyles, and the like), but inverting such principles can provide your town with imposing buildings that act as huge conductors of ancient energies.

To return to Lovecraft's Innsmouth, the hanging of decorative arrangements of seashells created both the sense of cohesive space, and of discomfiting unfamiliarity.

Playing with building scale, or the scale of openings aside, another trick involving architecture is the use of totally foreign elements. The black church in John Carpenter's 1994 film, *In the Mouth of Madness*, was instantly off-putting because it was a distinctly un-American building in the midst of a traditional all-American town. It was as unsettling as discovering a huddle of Neolithic huts hiding beneath modern-day Hyde Park. ⓦ

Geomancy explained

The Mayans, the ancient Egyptians, and all kinds of strange, mythical civilisations have used geomancy in their city planning. This means that civic space is organised not only according to human needs, but according to their beliefs. Such cities are defined by celestial alignments, flows of energy, solstices, divine geometries and elaborate astronomic charts. Their roads are more likely to follow the path of the sun, and their geometric centres are bound to have cosmological significance.

Figure 1: Claustrophobic, canyon-like roads (less than a metre in width, lined with disproportionately tall walls) rarely soothe.

Structurally Sound: how musical game worlds are made

88 Heroes composer Mike Clark explains how music and sound intertwine to create atmospheric game worlds
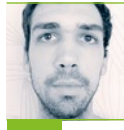
AUTHOR
MIKE CLARK

Mike Clark is a music composer from Southampton who's composed for indie games and performed live as an electronic music producer.
clarkmelodies.com

Undertale's soundtrack blends analogue synth instruments, like sine and square waves, with a plethora of real instruments to help create lots of emotion.

You can listen to the entire *Celeste* soundtrack by Lena Raine on Bandcamp for free.

Music for video games is often underappreciated. When I first started writing music in my bedroom, it took me a while to realise how much I was influenced by the worlds that came from my tiny CRT. A couple of years ago, I was lucky enough to be approached by Bitmap Bureau, an indie startup who hired me to compose the music for their first game, *88 Heroes*.

88 Heroes is a platformer styled like a Saturday morning cartoon. Interestingly, cartoon soundtracks have a lot in common with those for stage productions: short musical cues accompany the actions on screen, so if someone violently falls downstairs, you hear a piano rolling down the keys. This is called 'mickey mousing' in cartoons, but we hear similar things in film soundtracks. Take *Raiders of the Lost Ark*, scored by John Williams: for every heroic rope swing, leap of faith or close encounter with danger, the main theme can be heard powering through the dissonances and changing rhythms. It fills the audience with hope and becomes synonymous with the lead character – we want to see him succeed. Let's not forget the title theme. Every time you see the *Star Wars* logo, does that grand title theme play in your head? It's the same with video games. The challenge here, of course, is that players often leave the title screen after three seconds.

Three seconds is all you need, though. Take *Super Mario World*'s soundtrack, composed by

Koji Kondo. Many of its levels have the same leading melody, which changes subtly in tonality and rhythm to create the appropriate mood. The most repeating part of the melody is four bars long, but we hear it in so many forms that we only need the first two bars to know where it's from. In classical music, this is called 'variations on a theme'. In video games, we call it a 'sonic identity'.

HOW A PICTURE SHOULD 'SOUND'

Sonic identity informed my approach to the *88 Heroes* soundtrack. The title screen tells us that an unknown group is going to save the day.
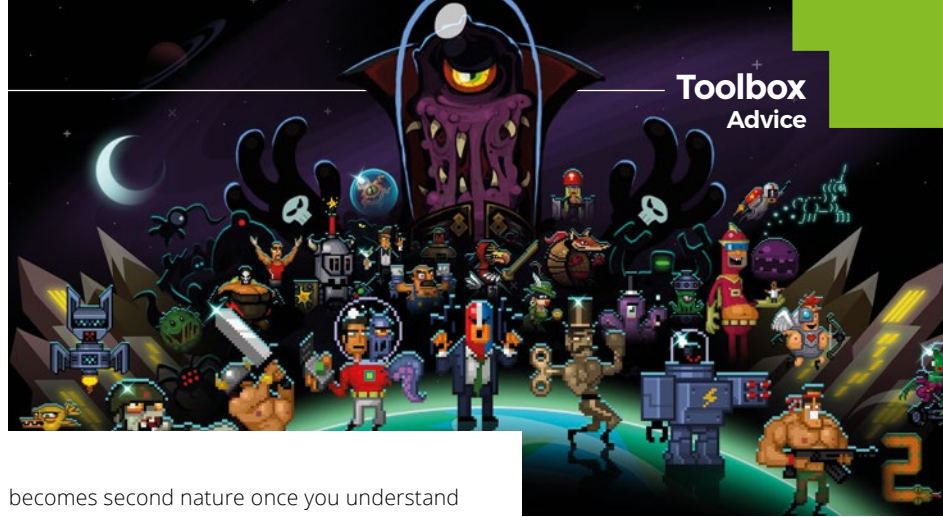
"I decided that game music should be music that uses an ensemble of instruments that you wouldn't usually hear anywhere else," Koji Kondo once said.

I first thought about unlikely heroes who end up on an adventure, and *Back to the Future*, scored by Alan Silvestri, sprang to mind. The second inspiration came from traditional superheroes, like *Superman*. I composed a melody which travels between the first and fifth notes in the scale (in this case C and G) with little flourishes of the notes in-between. It's a triumphant, heroic melody.

This concept helps to connect these worlds beyond their visuals. It took a long time for games to evolve into the cohesive open-world sandboxes or MMOs we see today; the technology that masked loading screens to create a seamless experience was unheard of in the 1990s, so a melody that you hear in different 'costumes' gives these games a sense of cohesion.

INTELLIGENT INSTRUMENTS

What if you have levels (or worlds) so big that some areas need to be loaded? That's where non-linear composition comes in. *Banjo-Kazooie*, released for the N64 in 1998, was among the first 3D games to feature dynamic music. It used a technique called MIDI channel fading. MIDI stands for Musical Instrument Digital Interface; think of it as a universal language for music that is played back in real time by the hardware. As you walk into caves, fly in the sky or move near certain characters, instruments fade in and out using the different MIDI channels to mimic the atmosphere, give the player an audio cue, and build and release tension.

Learning how to write music that changes as you play might seem impossible at first, but it becomes second nature once you understand the relationship between every instrument in your composition. Many digital audio workstations, like Logic and FL Studio, let you import MIDI data for a song (so you have all the notes in front of you) and set the instruments yourself. Try slowly fading out or muting certain tracks altogether, and listen to how the mood changes. What could this change represent in a video game? It's like when you're riding Yoshi in many of the *Mario* games; the fast bongos come in to represent the quick-footed dinosaur as he dashes at high speeds.

Music is used to evoke emotions that wouldn't be possible with visuals alone. *Beep: A Documentary History of Game Sound* shows a six-second video of a boat accompanied by two soundtracks; one is a light and happy guitar piece, the other a grating, scary, orchestral dissonance. Through these two extremes, the music creates the mood by itself. I remember playing *Metroid Prime* and finding the Chozo Ghost enemies rather scary, not because of their appearance, but because of the unnerving music that accompanies them.

> **"Music and sound design are one and the same"**

Music and sound design are one and the same. Think about what feelings you can create by taking music away entirely – it's a great way to create tension before a boss battle or pivotal plot point, and it really works. In *Undertale*, scored by Toby Fox, there are times when the music stops so abruptly during NPC dialogue that you feel shivers down your spine.

So, what if you're trying to come up with some game music, and you have writer's block? Well, the next time you play a new game, turn the sound off. As you're playing, focus on how the story, art or characters make you feel, and focus on the emotions the game is trying to convey. Then, think of a time when a song made you feel happy, sad, joyful, anxious, or even frightened. Maybe you can use the music to create the mood you want for that game, as opposed to what the game makes you feel. By finding these emotions and understanding how they can change, you'll be able to write a score that helps strengthen the immersion, escapism and player investment in your game. Ⓦ

⌃ The initial idea was to have different music playing for each character. Sadly, that would have taken far too long, but one or two characters do still have their own theme.

⌃ Grant Kirkhope, the composer and sound designer for *Banjo-Kazooie*, sent five tapes to the company before he even heard back.
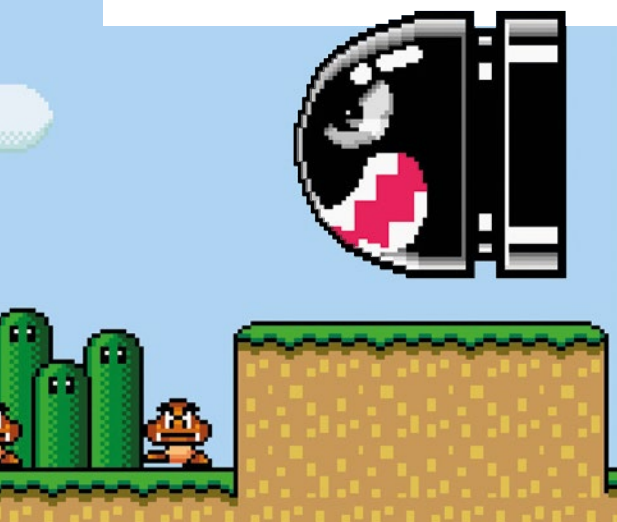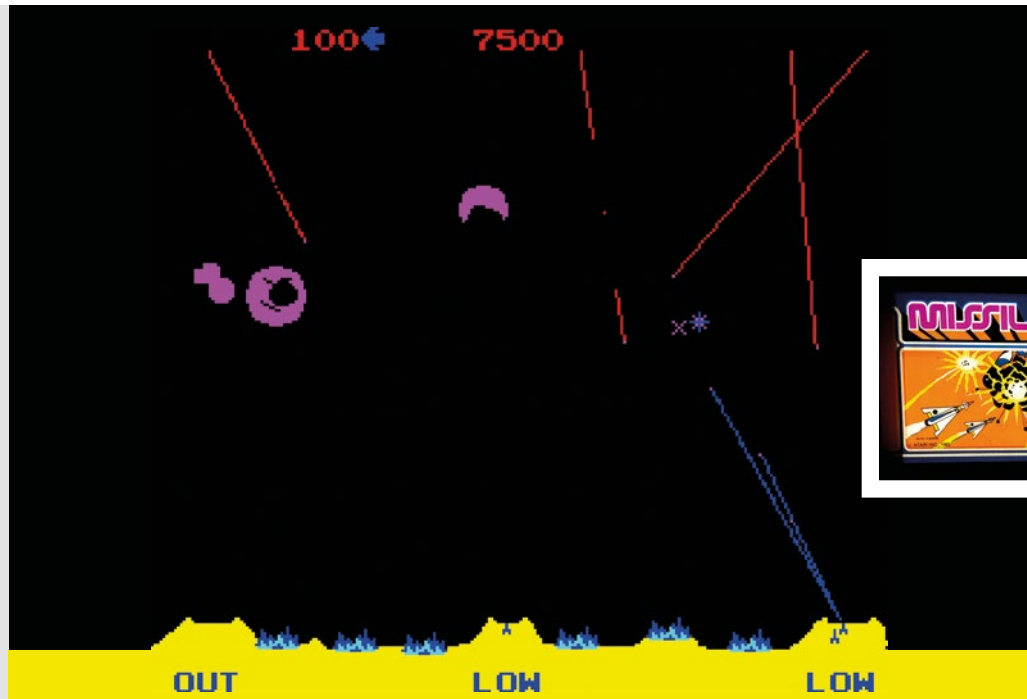
∧ The original *Missile Command*: simple, addictive, disquieting.

Missile Command's scary vapour trails

Vapour trails helped make Missile Command an arcade classic. Here's how to code your own

AUTHOR
RYAN LAMBIE

Rarely has a single vector line represented something so horrifyingly momentous. In Atari's *Missile Command*, each thin red band streaking down the screen represented the vapour trail of a deadly nuclear warhead, fired by an unnamed country at a row of six cities squatting vulnerably at the bottom of the screen. The challenge was to defend those cities from certain destruction; enemy warheads had to be taken out by moving a cross-hair with a trackball, and firing a dwindling stock of counter-missiles. Time the firing of a missile correctly, and skilful players could catch several falling warheads in the burst of a single counter-missile. Get the timing wrong, though, and you'd watch in horror as a warhead continued its silent fall towards one of your urban centres.

Missile Command took a similar concept to *Space Invaders* (destroy the objects descending from above), but added a new, ingeniously button-pushing Cold War twist, underlined by its now infamous use of the words, 'The End', which appeared when the player's cities were wiped out.

NIGHTMARE SCENARIO
Indeed, developer David Theurer evidently recognised *Missile Command*'s nightmarish quality: as he later revealed in an interview with Polygon, he suffered from recurring nightmares about nuclear war for months.

Released in 1980, a period of renewed tensions between the West and the Soviet Union, *Missile Command* had a hint of realism that was unlike other arcade games of its era. Its concept was reportedly inspired by a magazine article about satellite defence systems, and on reading it, Atari coin-op designer Steve Calfee called Theurer into his office and told him to make a game that

involved looking down at a radar screen and defending American cities from Russian missiles. From this somewhat loose brief, Theurer created an arcade classic: a lean, intense coin-muncher whose ingenuity is easily overlooked amid all the blasting.

Missiles are randomly spawned at the top of the screen in waves of three or four, and are set to fall towards a random target at the bottom. The missiles are represented by a single, flashing pixel, while the vapour trail – a vector line that follows the missile's x-y trajectory – helps the player track the threat as it hurtles down the screen. Theurer gradually intensified the action through similarly simple yet effective means: enemy missiles will sometimes split up into further warheads, and their speed and number increases in each round.

Just to add to the air of grim inevitability, there's no way to win in *Missile Command* – all you can do is try to gain a new high score before the world ends. ⓦ

Missiles and trails in Python

Written by the great Daniel Pope, here's a code snippet that illustrates *Missile Command*'s vapour trails in Python. To get it running on your system, you'll first need to install Pygame Zero – you can find full instructions at **wfmag.cc/XVIIeD**

```python
import random
from collections import deque
from itertools import tee
from math import sin
WIDTH = 800
HEIGHT = 400

GRAVITY = 5
TRAIL_LENGTH = 400

TRAIL_BRIGHTNESS = 100
FLARE_COLOR = (255, 220, 160)

missiles = []


class Missile:
    def __init__(self, x, vx, y=0, vy=20):
        self.x = x
        self.y = y
        self.vx = vx
        self.vy = vy
        self.trail = deque(maxlen=TRAIL_LENGTH)
        self.t = random.uniform(0, 3)

    def step(self, dt):
        self.t += dt
        uy = self.vy
        self.vy += GRAVITY * dt
        self.y += 0.5 * (uy + self.vy) * dt

        self.x += self.vx * dt
        self.trail.appendleft((self.x, self.y))

        # If the trail is off the bottom of the screen
        # kill the missile
        if self.trail[-1][1] > HEIGHT:
            missiles.remove(self)
            return

    def draw(self):
        for i in range(len(self.trail)):
            if i + 1 == len(self.trail):
                break
            start = self.trail[i]
            end = self.trail[i + 1]
            c = TRAIL_BRIGHTNESS * (1.0 - i / TRAIL_LENGTH)
            color = (c, c, c)
            screen.draw.line(start, end, color)
        screen.draw.filled_circle((self.x, self.y), 2, FLARE_COLOR)
```

```python
        # This small flickering lens flare makes it look like
        # the missile's exhaust is very bright.
        flare_length = 4 + sin(self.t) * 2 + sin(self.t * 5) * 1

        screen.draw.line(
            (self.x - flare_length, self.y),
            (self.x + flare_length, self.y),
            FLARE_COLOR
        )


def draw():
    screen.clear()
    for m in missiles:
        m.draw()


def update(dt):
    for m in list(missiles):
        m.step(dt)


def new_missile():
    m = Missile(x=random.randrange(600, 800), vx=random.uniform(-70, -10))
    missiles.append(m)


new_missile()
clock.schedule_interval(new_missile, 5)
```

Download the code from GitHub: **wfmag.cc/wfmag8**

NUCLEAR CODE

Daniel Pope's code not only shows how to recreate *Missile Command*'s falling warheads in Python, but it also adds a few modern flourishes that simply wouldn't have been achievable on the hardware available to David Theurer in 1980. The ends of the trails gradually decrease in brightness as they descend, while a flickering lens flare effect helps pick out the position of the warhead itself.

Head to Wireframe's GitHub repository at **wfmag.cc/wfmag8**, and you'll find a longer version of the same code – **missiles-wobbly.py**. This uses a perlin noise function to create a more sophisticated kind of vapour trail which curls delicately as the missile falls.

Dan's longer code features curling vapour trails.

Squeezing the Beeb: make a helicopter score attack game

A challenging score attack game squeezed into just 1kB on the BBC Micro? Eben shows you how

AUTHOR
EBEN UPTON

Eben is the creator of Raspberry Pi and a co-founder of the Raspberry Pi Foundation.

Squeezing better performance out of fixed platforms has been an important part of game development since the early days. Comparing launch titles with those created towards the end of a platform's life demonstrates how much latent power developers are able to wring out of a piece of hardware once they understand its capabilities and limitations. It remains a relevant skill today.

Older machines provide a perfect, constrained environment to practise this sort of optimisation, either for education or recreation. This is particularly true if you also impose some artificial restrictions, like total memory footprint, display update rate, and so on. In this article, we'll look at how to fit a score attack game into a kilobyte of 6502 machine code, running 'in a frame' on the venerable BBC Micro.

THE PLATFORM

The BBC Micro was one of the more successful 8-bit computers of the 1980s. Like many of its contemporaries, it was based on the MOS Technology 6502 processor. The baseline Model B shipped with 32kB of RAM, and could output 8-colour graphics at a screen resolution of 160×256 pixels. There is no sprite, playfield, or other display acceleration hardware, making

it a very 'one-dimensional' optimisation target: all backgrounds and moving objects need to be drawn to the framebuffer in software, and getting a good result is all about squeezing bytes and cycles out of your program.

Display memory is arranged in a rather idiosyncratic fashion. In 8-colour mode, each byte contains two horizontally adjacent pixels interleaved in bits (6,4,2,0) and (7,5,3,1). Video memory starts at address $3000 (where $ indicates a hexadecimal number), and pixel-pair byte addresses look like the table in **Figure 1**.

The screen is split vertically into 32 8-pixel-tall rows. Each row is 8×160/2 = 640 = $280 bytes in size, so the second row down starts at address $3280, and the entire framebuffer occupies 20kB. Within a row, pixel pairs are stored in columns of eight, so the second pixel pair across in the first row starts at address $3008. This is convenient for software which wishes to draw 8-pixel tall characters aligned within rows, but creates fun challenges if you want to plot a sprite at an arbitrary location.

THE GAME

Let's see how this mechanic might work in pretty much the simplest possible environment. In our game, the player will fly a helicopter around the screen, collecting coins, avoiding birds,

▾ Avoid the birds, grab the coins, score points: *Copter* has all the ingredients of an effective arcade game, squeezed into a few bytes.

$3000	$3008	$3010	$3018	$3020	$3028	$3030	...
$3001	$3009	$3011	$3019	$3021	$3029	$3031	...
$3002	$300a	$3012	$301a	$3022	$302a	$3032	...
$3003	$300b	$3013	$301b	$3023	$302b	$3033	...
$3004	$300c	$3014	$301c	$3024	$302c	$3034	...
$3005	$300d	$3015	$301d	$3025	$302d	$3035	...
$3006	$300e	$3016	$301e	$3026	$302e	$3036	...
$3007	$300f	$3017	$301f	$3027	$302f	$3037	...
$3280	$3288	$3290	$3298	$32a0	$32a8	$32b0	...
$3281	$3289	$3291	$3299	$32a1	$32a9	$32b1	...
$3282	$328a	$3292	$329a	$32a2	$32aa	$32b2	...
...	...	...	...	...	...	...	...

△ **Fig 1: A handy table of pixel-pair byte addresses.**

and trying not to fall in the sea. The helicopter consumes fuel when climbing, and must regularly refuel on a platform which moves from side to side in the sea.

The score attack element comes in how coins are scored. The first coin collected during a flight places one point in a bonus pot; each subsequent coin doubles the pot and when the helicopter next lands, the pot is added to the current score.

To make things more fun, we'll set ourselves a few arbitrary restrictions:
- Machine code should fit in less than 1kB
- Graphic assets should fit in less than 1.5kB
- The game should run 'in a frame', i.e. in less than 20 ms for a 50Hz PAL display
- It shouldn't use operating system services

Let's take a look at a few places where we've had to work to squeeze out speed and/or size.

(FAIRLY) RANDOM NUMBERS

When the player collects a coin, we need to generate a pseudorandom location (x, y) on the screen for its replacement. As is often the case in game development, we place a premium on speed over quality in our random numbers.

The function **rand**, taken from **wfmag.cc/ LMDaHd** and reproduced below, implements a Galois linear-feedback shift register (LFSR). This occupies 17 bytes, and generates a new 8-bit random number in an average of 132 cycles; the resulting sequence repeats with period 65535 (the theoretical maximum).

```
rand
   ldy #8         ; 2 bytes, 2 cycles
   lda lfsr+0     ; 2 bytes, 3 cycles
l17
   asl            ; 1 byte, 2 cycles
   rol lfsr+1     ; 2 bytes, 5 cycles
   bcc s19        ; 2 bytes, 2.5 cycles
   eor #$2d       ; 2 bytes, 1 cycle
s19
   dey            ; 1 byte, 2 cycles
   bne l17        ; 2 bytes, 2.875 cycles
   sta lfsr+0     ; 2 bytes, 3 cycles
   rts            ; 1 byte
```

A faster, larger LFSR implementation for 6502 can be found at **wfmag.cc/FRxvS**.

DRAWING SPRITES

The BBC Micro's unusual video memory layout creates a number of challenges when drawing sprites. In general, we need to handle the (very common) case when our sprite overlaps more than one row, and to construct pixel pairs from individual pixels in a different way for sprites at even and odd x locations. To simplify things a little, we'll limit ourselves to sprites which are 8 pixels tall (so they overlap only one or two rows), and 7 pixels wide (so that for each frame we can store one 8x8-pixel sprite for even locations, and another for odd locations with the pixel shuffling 'baked in').

We'll provide support for three plotting modes: store, pad, and blend. Store copies the sprite data directly to video memory; it's used to draw coins, and to erase the helicopter at its previous position. Pad is identical to store but draws two columns of cyan pixels to the left and right of the sprite data; it's used to draw birds, which travel horizontally at one pixel per frame, and so can be made self-erasing. Blend uses the two most significant bits of each byte of sprite data as a mask, to control which pixels are written, and to support pixel-perfect collision detection; it's used to draw the helicopter, and the landing pad.

Let's take a quick tour of the function plot. This takes four arguments:
- A sprite index in A
- An operation code (**O_PAD**, **O_STR**, **O_BLN**) in X
- An x coordinate in Y (because, why not?)
- A y coordinate in zero-page location w8 ($70)

We call plot sufficiently often with **O_STR** and **O_BLN** that it's worth adding entry points that ➡

⌄ **Ah, the blocky, lovable BBC Micro: mainstay of schools up and down the UK.**

load X for us: the cost of the entry points is paid for by the LDX instructions saved elsewhere. We use the 2-byte BIT hack here. The plot function contains a branch, at label m02, which is executed each time plot wishes to draw some pixels; the operation codes are carefully chosen so that overwriting the operand of that branch with an operation code makes the branch point to the piece of code which performs the appropriate operation.

Once we've modified the branch, we're done with X, so we zero it and use it to clear some workspace registers, and the hit value used for collision detection.

```
plot_s              ; store entry point
    ldx #O_STR
    .byte $2c       ; 2-byte BIT hack

plot_b              ; blend entry point
    ldx #O_BLN

plot                ; generic entry point
    stx m02+1       ; modify branch

    ldx #0          ; clear workspace
    stx w5
    stx w3
    stx w1
    stx hit         ; and hit flag
```

Next, we'll calculate the source address for the sprite data. Remember, we store two copies of each sprite: one for even x coordinates, followed by one for odd x coordinates. We store x (which is held in Y, remember) in w7, shift it right (so x is now expressed in pixel pairs, with the LSB shifted into the carry flag), and then add $1300/32 = $98, and the carry. The function **asr16** multiplies everything by 32, storing the source address in w1,w0.

```
sty w7
    lsr w7          ; divide by two
    adc #$98        ; add sprite base and LSB
    ldy #5          ; multiply by 32
    jsr asl16
```

Now we'll do the same for the destination address into w3,w2. This looks pretty gruesome, but really just implements the following formula:

addr   = $3000+x*8+(y&7)+(y>>3)*$280

...where x is expressed in pixel pairs. Most of the effort is in computing the last term. In fact, there's already a handy 'multiply by 640' table in the OS 1.2 ROM at address $C375, but remember we decided not to use any OS services. In an amusing demonstration of what happens when a 'fixed' optimisation target isn't really fixed, the designers of the BBC Master moved this table without realising how many games relied on it: hilarity ensued.

```
lda w8              ; compute y&7
    and #7
    sta w6

    lda w8          ; compute y>>3
    lsr
    lsr
    lsr
    sta w2
    asl             ; compute (y>>3)*5
    asl
    adc w2
    ldy #7          ; w3,w2 = (y>>3)*$280
    jsr asl16
    lda w7          ; w5,w4 = x*8
    ldy #3
    jsr asl16
    adc w6          ; add terms
    adc w2
    sta w2
    lda w3
    adc w5
    adc #$30
    sta w3
```

Finally, we're ready to draw some pixels, by calling one of the three plot kernels either one or two times. The kernels expect to be called with Y=0, the carry flag cleared, and the following arguments:
- The source address in w1,w0
- The destination address in w3,w2
- Number of bytes to process per column in w4
- Number of columns to process (=4) in w5
- Number of bytes to skip per column (=8-w4) in w6

We've set almost all of these up already; the only remaining tasks are to set up w4 and w5, and to clear Y. The code to do this is located at

▲ Altogether, *Copter*'s graphics only take up a piffling 1.5kB of memory.

the label **stripe**: we call it once, then perform the adjustments needed to move to the next row, and then fall through into it a second time if work remains to be done. The **bvc pad** instruction has been modified at the start of the plot function to point to the appropriate kernel.

```
lda #9          ; carry clear
  sbc w6
  clc
  jsr stripe

  lda w4        ; advance source ptr
  adc w0        ; no carry
  sta w0
  lda w2        ; advance destination ptr
  and #$f8
  adc #$80
  sta w2
  lda w3
  adc #$02
  sta w3
  ldy w4        ; swap w4 and w6
  lda w6
  sty w6
  beq out       ; exit if no bytes left

stripe
  sta w4
  lda #4
  sta w5
  ldy #0
m02
  bvc pad       ; to be modified
```

To understand how a kernel works, let's take a look at a simple version of the store kernel. To save space, the actual program implements this by modifying the blend kernel at runtime to skip the masking and hit-test code.

The outer loop iterates w5 times (generally four). On each iteration, we load X with the number of bytes to process, copy that many bytes from the source to the destination, and then add a value to Y to skip a certain number of bytes.

```
store
loop0
ldx w4          ; bytes to process
loop1
```

THE BIT HACK

The most notorious 6502 hack uses the BIT instruction (which tests the value at a memory location, but does not modify any registers) to provide multiple entry points to a function. Suppose we have a function that we want to enter either with X=2 or X=3. We might write:

```
enter2
  ldx     #2
  bne     body      ; 2 bytes
enter3
  ldx     #3
body
  :       :
  rts
```

But we can save a byte by rewriting this as:

```
enter2
  ldx     #2
  .byte   $2c       ; BIT opcode
```

```
enter3
  ldx     #3
  :       :
  rts
```

This is really devious. If we enter at **enter3**, the code runs as normal.

But if we enter at **enter2**, then the processor sees that $2c byte along with the following two bytes ($a2, $03), which make up the second LDX instruction, as a 3-byte BIT instruction like so:

```
enter2
  ldx     #2
  bit     $03a2
  :       :
  rts
```

The second LDX doesn't happen, and X remains set to 2.

⌄ **Released in 1988, *Exile* was a classic commercial example of a game that squeezed the Beeb.**
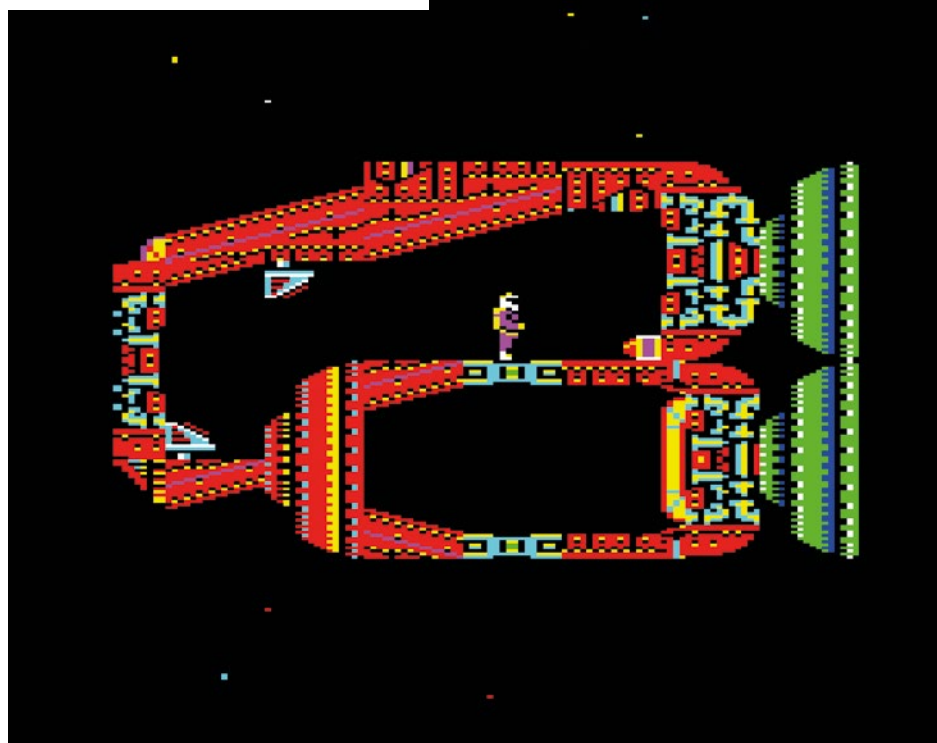
ADDRESSES AND ADDRESSING MODES

While the 6502 is an 8-bit processor, addresses are 16-bit values in the range 0…65535 (or $0 to $ffff in hexadecimal). The 256 bytes of memory are known as zero page: instructions that access zero page are smaller and execute faster, and some addressing modes (see below) make use of values stored there; for this reason, machine-code programs store commonly used values there. On the BBC Micro, locations $50…$af are generally free for use by user programs. Addressing modes determine how the 6502 finds data in memory. Here are a few examples:

```
lda     #42             ; immediate: load A with 42
and     $8000           ; absolute: and A with value at address $8000
 ora    $8000,X         ; indexed: or A with value at address $4000+X
 xor    ($44),Y         ; indirect: exclusive or A with value at;
                        ; address formed by taking the 16-bit value
                        ; at address $44…$45 and adding Y
```

CARRY PROPAGATION

The 6502 does not provide an ADD instruction. Instead, the ADC instruction adds a value to A, and also adds in the value of the 1-bit accumulator flag C. To ensure we add the right value, we can use the CLC (clear carry) instruction, like this:

```
clc
adc     #42
```

But this costs a precious byte and two precious cycles! Often we find we can infer the value of the carry flag by careful analysis of our code. So for example, if the value at location $77 is known to be a multiple of 2, we can comment out the CLC instruction in this fragment:

```
lsr     $77
; clc
adc     #42
```

This sort of analysis tends to generate very brittle code, but can be a good final polishing step.

```
lda (w0),y     ; copy a byte (two pixels)
sta (w2),y
iny
dex
bne loop1
tya
clc
adc w6         ; bytes to skip
tay
dec w5
bne loop0
rts
```

RACING THE RASTER

There's a long-standing obsession with running games at the full frame rate supported by the display device. Early console platforms tend to do this, because they lack a framebuffer and use dedicated hardware to compose the display on the fly.

Platforms which operate by first generating an image in memory and then scanning it out (most home computers, modern consoles) can trade scene complexity for frame rate by scanning the same image out for several frames and spreading the work of generating the next image across those frames Nonetheless, games which run at the full frame rate appear smoother to the eye, with the result that 50Hz (or 60Hz) is often used as a marketing slogan; Team17 was famous for promoting their late-period Amiga games in this way.

Our game must run in a frame for a more prosaic reason: the 20kB framebuffer occupies more than half the total memory in the system, so it is not feasible to double buffer the display (generate one image while displaying another). We will be updating the framebuffer live as the hardware is scanning it out to the display, and must avoid having the raster intersect with an object during the period after it has been erased at its previous location, but before it has been redrawn: this would result in an unsightly flickering effect. It's much easier to manage the position of the raster, and so avoid flicker, if you're running in a frame.

We solve the flickering problem in different ways for different objects:

- Coins do not move, and each frame is drawn over its predecessor with an opaque background. There is no separate erase and redraw step, so no flicker can occur.
- Birds move horizontally, but the pad operation causes it to leave a solid trail of cyan pixels behind it. Again, no explicit erase is required, so no flicker can occur.
- The sea and moving platform are at the bottom of the screen, and will always have been updated before the raster reaches that point.

The helicopter is the most complex case. We must erase it, process the coins and birds, and finally redraw it at its new location. There is therefore a significant window of vulnerability for flicker to occur. Our solution is to package all of the remaining work (drawing the score, bonus and fuel bar, and the sea and the platform) into a function, **canmove**. If the helicopter is in the bottom half of the screen, we process it first before the raster can reach it, and only then call **canmove**; conversely, if the helicopter is in the top half of the screen, we call **canmove** first to allow the raster time to reach the bottom half, and only then process it.

CONCLUSION

So there we have it: a score-attack game for a 37-year-old piece of computer hardware, and a

^ **Three frames of animation are all you need to create a tiny helicopter with a spinning rotor blade.**

demonstration of how much you can squeeze into a kilobyte of 6502 machine code. You can build it from source with the ATasm assembler like so:

```
atasm -r wireframe.s -owfm.bin
```

Copy the resulting binary to a BBC Micro floppy (or, more realistically, an emulator SSD file) disk with the name WFM, along with the graphics asset file located at **wfmag.cc/wfmag8**. Then fire up the Beeb, load the disk, and type:

```
MODE 2
*LOAD GFX 1300
*LOAD WFM 880
CALL &880
```

Use **Z** and **X** to move left and right, and **SHIFT** to thrust upward. When you inevitably die, press **SPACE** to start a new game. The game can be played online at **wfmag.cc/copter**. Ⓦ

Credits: Graphics by Sam Alder and Alex Carter, based on resources developed for an event at the Centre for Computing History in Cambridge.

KEY CODE

The full Copter source code is reproduced in compact form on the right, with the key below outlining which bits do what:

1. Constants, zero-page map

2. Initialise zero-page, hardware; clear screen

3. Top of main loop: wait for vblank

4. Update animation frame and platform position

5. Possibly do moveable work

6. Check if player has died

7. Run player physics

8. Erase player

9. Run coins

10. Run birds

11. Draw player

12. Do moveable work if not done

13. End of main loop

14. Moveable work: first draw water and platform

15. Draw score and bonus

16. Draw fuel bar

17. Read key

18. Collision detection

19. Do player physics for one axis

20. Galois LFSR random number generator

21. Draw 4x8 pixel digit

22. 16-bit variable shift left

23. Plot function with entry points

24. Kernels

25. Zero-page and hardware initialisation data

Copter code

Here's the complete Copter listing. Don't want to type it all out? You'll also find everything you need on GitHub: **wfmag.cc/wfmag8**

```
1   S_COPL = 12
    S_COPR = 14
    S_COIN = 24
    S_LAND = 28
    S_CYAN = 40

    L = S_LAND/2
    M = L+3

    WATR = $1700
    DIGS = $1840

    O_PAD = p-p
    O_STR = s-p
    O_BLN = b-p

    KEY_Z  = $61
    KEY_X  = $42
    KEY_SP = $62
    KEY_UP = $00

    p_fuel = $50
    p_live = $51
    b_y = $53
    b_x = $58
    b_w = $5d
    l_dx = $62
    p_dx = $63
    da = $64
    lfsr = $66

    w0 = $68
    w1 = $69
    w2 = $6a
    w3 = $6b
    w4 = $6c
    w5 = $6d
    w6 = $6e
    w7 = $6f
    w8 = $70
    w9 = $71

    hit = $77
    mframe = $78
    sframe = $79
    wframe = $7a
    scr = $7b
```

```
    spacer = $7e
    bonus  = $7f

    l_x = $82
    c_x = $83
    c_y = $87
    p_xl = $8b
    p_yl = $8c
    p_xh = $8d
    p_yh = $8e
    p_vxl = $8f
    p_vyl = $90
    p_vxh = $91
    p_vyh = $92
    p_xo = $93
    p_yo = $94
    p_hit = $95
    b_ox = $96
    b_dx = $9b

    hwdat = $66
    hwadr = $6c

    *=$880

2     sei

    top
     ldx #$4f
    l00
     lda zpage,x
     sta $50,x
     dex
     bpl l00

     lda #$3c
     ldy #0
    l01
     sta ($52),y
     iny
     bne l01
     inc $53
     bpl l01


    l02
     inx
     lda hwdat,x
```

```
     ldy hwadr,x
     sta $fe00,y
     bne l02

     sta spacer

3   loop
     lda #2
    l03
     bit $fe4d
     beq l03
     sta $fe4d

4    lda mframe
     and #$0c
     cmp #$0c
     bne s00
     lda #$04
    s00
     sta sframe

     lda mframe
     lsr
     bcs s02
     lda l_x
     cmp #122
     lda l_dx
     bcc s01
     eor #$ff
     adc #0
     sta l_dx
    s01
     adc l_x
     sta l_x
    s02

5    lda p_yh
     php
     bmi s03
     jsr canmove
    s03

6    jsr detect
     lsr p_hit
     jsr detect

     lda p_yh        ➡
```

```
cmp #232
bne s05
lda p_live
bpl die
lda mframe
lsr
lda p_xh
bcs s04
adc l_dx
sta p_xh
s04
sec
sbc l_x
cmp #3
bmi die
cmp #29
bmi live
die
plp
l04
lda #KEY_SP
sta $fe4f
lda $fe4f
bpl l04
jmp top
live
ldx #254
jsr add24
sty p_fuel
iny
sty bonus+1
sty bonus+2
sty p_vxl
sty p_vxh
beq s07
s05
```

7

```
ldy #0
lda #KEY_Z
jsr inkey
bpl s06
dey
lda #S_COPL
sta p_dx
s06
lda #KEY_X
jsr inkey
bpl s07
iny
lda #S_COPR
sta p_dx
```

```
s07
lda #153
ldx #0
jsr axis

ldy #1
txa
jsr inkey
bpl s08
lda p_fuel
beq s08
dec p_fuel
dey
dey
s08
lda #232
inx
jsr axis
```

8

```
ldy p_xo
lda p_yo
sta w8
lda #S_CYAN
jsr plot_s
```

9

```
ldx #3
l05
stx w9
lda c_y,x
asl a
sta w8
bcc s09
lda mframe
and #$18
lsr a
lsr a
lsr a
adc #S_COIN
ldy c_x,x
bcc s10
s09
l06
jsr rand
and #120
cmp #120
bpl l06
ora #$84
ldy #3
l07
cmp c_y,y
beq l06
```

```
dey
bpl l07
sta c_y,x
l08
jsr rand
asl
cmp #154
bcs l08
ldy c_x,x
sta c_x,x
lda #S_CYAN
s10
jsr plot_s
ldx w9
dex
bpl l05

ldx #4
```

10

```
l09
stx w9
lda b_ox,x
cmp b_w,x
lda b_dx,x
bcc s11
eor #2
sta b_dx,x
clc
s11
adc b_ox,x
clc
adc #$ff
sta b_ox,x
adc b_x,x
tay
lda b_y,x
sta w8
lda b_dx,x
adc sframe
ldx #O_PAD
jsr plot
ldx w9
dex
bpl l09
```

11

```
ldy p_xh
sty p_xo
lda p_yh
adc #8
sta p_yo
sta w8
lda p_dx
```

```
adc sframe
jsr plot_b
lda hit
sta p_hit
```

12

```
plp
bpl s12
jsr canmove
s12
```

13

```
inc mframe
jmp loop
```

14

```
canmove
lda wframe
clc
adc #8
cmp #$c0
bcc s13
lda #0
s13
sta wframe
and #$c0

ldy #127
l10
sta m00+1
ldx #63
l11
m00
lda WATR,y
sta $7d80,y
sta $7e00,y
sta $7e80,y
sta $7f00,y
sta $7f80,y
dey
dex
bpl l11
lda m00+1
clc
adc #64
cpy #0
bpl l10

lda #248
sta w8
lda l_x
adc #8
tay
pha
```

```
lda #L
sta w9
l12
asl
jsr plot_b
pla
adc #7
tay
pha
inc w9
lda w9
cmp #M
bne l12
pla
```

15

```
lda #$f0
sta da
ldx #7
l13
stx w4
lda scr-1,x
pha
asl a
asl a
jsr digit
pla
lsr a
lsr a
jsr digit
ldx w4
dex
bne l13
```

16

```
lda p_fuel
lsr a
lsr a
tay
clc
l14
lda #$3c
dey
dey
beq s14
bmi s15
eor #$15
s14
eor #$2a
s15
sta $3123,x
sta $3124,x
sta $3125,x
```

```
txa
adc #8
tax
bne l14
rts
```

17

```
inkey
sta $fe4f
lda $fe4f
and p_live
rts
```

18

```
detect
lda p_hit
and #$15
beq none
cmp #$14
beq none
cmp #$11
beq coin
cmp #$15
beq coin
lsr p_live
none
rts

coin
lda p_yh
ror
clc
adc #4
and #$f8
adc #4
ldx #3
l15
cmp c_y,x
bne s16
and #$7f
sta c_y,x
ldx bonus+2
beq zero
ldx #2
add24
sed
clc
ldy #2
l16
lda bonus,x
adc bonus,y
sta bonus,x
dex
```

```
      dey                sta w0             ora w3             and #7         24                   sta hit
      bpl l16            lda p_vxh,x        sta (da),y         sta w6                pad            commit
      cld                cmp #$80           pla                                     p                eor (w2),y
      rts                ror A              lsr                lda w8                 jsr col         ora (w0),y
 zero                    ror w0             lsr                lsr                                    and #$3f
      inx                cmp #$80           dey                lsr                    lda w2        fast
      stx bonus+2        ror A              dex                lsr                    adc #8        f
      rts                ror w0             bpl l19            sta w2                 sta w2           sta (w2),y
 s16                     sta w1             inc m01+1          asl                    txa           bits_11
      dex                lda p_vxl,x        tya                asl                    tay              iny
      bpl l15            sec                bpl l18            adc w2                 adc w3           dex
      rts                sbc w0                                ldy #7                 sta w3           bne l23
                         sta p_vxl,x        lda da             jsr asl16                               tya
19                       lda p_vxh,x        sbc #16            lda w7                 jsr store        clc
 axis                    sbc w1             sta da             ldy #3                                  adc w6
      sta w2             sta p_vxh,x        rts                jsr asl16         col                   tay
                         rts             22                    adc w6                 ldx w4           dec w5
      tya             20              asl16                    adc w2                 lda #60          bne l22
      beq s17          rand              asl                   sta w2            l21                   rts
      cmp #$80           ldy #8            rol w1,x            lda w3                 sta (w2),y    bits_x1
      ror                lda lfsr+0        dey                 adc w5                 iny              bcs bits_11
      pha             l17                  bne asl16           adc #$30               dex              lda (w2),y
      and #$80           asl               sta w0,x            sta w3                 bne l21       bits_01
      clc                rol lfsr+1        inx                                        rts              and #$aa
      adc #$40           bcc s19           inx                lda #9                                   cmp #$28
      adc p_vxl,x        eor #$2d       out                    sbc w6             store                bvc decide
      sta p_vxl,x     s19                  rts                 clc             s                    bits_10
      pla                dey                                   jsr stripe            lda #f-slow       and #$55
      adc p_vxh,x        bne l17        23                                           .byte  $24        cmp #$14
      sta p_vxh,x        sta lfsr+0     plot_s                 lda w4                                   bvc decide
 s17                     rts               ldx #O_STR          adc w0            blend
                                           .byte $2c           sta w0         b                    25
      clc             21                                       lda w2                 tya           zpage
      lda p_xl,x      digit             plot_b                 and #$f8                                .word $ffff
      adc p_vxl,x        and #$3c          ldx #O_BLN          adc #$80               sta m03+1        .word $3000
      sta p_xl,x         clc                                   sta w2                                  .word $40b0
      lda p_xh,x         adc #<DIGS     plot                   lda w3            l22                    .word $d020
      adc p_vxh,x        sta m01+1         stx m02+1           adc #$02             ldx w4             .word $2210
      cmp w2                                                   sta w3            l23                    .word $5701
      bcc s18            ldy #15           ldx #0              ldy w4               lda (w0),y          .word $6607
      ldy #0          l18                  stx w5              lda w6            m03                     .word $4941
      sty p_xl,x         ldx #3            stx w3              sty w6              bpl fast             .word $6137
      sty p_vxl,x     m01                  stx w1              beq out           slow                   .word $0c01
      sty p_vxh,x        lda DIGS          stx hit                                 asl a               .word $3000
      cmp #248        l19                                   stripe                  bmi bits_x1
      lda w2             pha               sty w7              sta w4               lda (w2),y          .word $1454
      bcc s18            and #3            lsr w7              lda #4               bcs bits_10         .word $037f
      tya                asl               adc #$98            sta w5            bits_00                .word $aa02
 s18                     asl               ldy #5             ldy #0               cmp #$3c
      sta p_xh,x         sta w3            jsr asl16        m02                 decide                  .word $2021
                         asl                                 bvc pad             beq commit           .word $4043
      lda p_vxl,x        asl               lda w8                                                      .byte $4d    ⓦ
```

Network and learn with
Ukie Hub Crawl

Want to meet other developers, and learn more about securing investment? Then check out Ukie's series of free Hub Crawl events taking place all year

GET INVOLVED

Do you have an online course you'd like to share with readers? Have you created an online resource that other game developers might find useful? Maybe you have a local code club you're keen to promote? If you have something you'd like to see featured in the Directory, get in touch with us at **wfmag.cc/hello**.

What is it?
Hosted by the Association for UK Interactive Entertainment (or Ukie to its friends), Hub Crawl is a series of informal events designed to bring indie developers together.

Join fellow games developers and publishers and identify new opportunities and strategies to support the growth of your business.

What will I get out of it?
This year's Hub Crawl will focus on helping games businesses get investment-ready. This will consist of a series of free events, located all around the UK, bringing games businesses together to learn about types of investment, with practical guidance on business planning, pitching and growth.

Each event will feature a series of short, informative sessions from industry experts and this will be followed by a games panel featuring companies from the local area. Content will change at each event so make sure you check out all the events in your region.

After each session there'll be a chance to network, share ideas and have a drink with your peers.

Where's it held, and how much is it?
This is a free event open to anyone involved in the games industry. You'll find confirmed dates and locations in the table below, and more dates will be announced as the year progresses. You can keep up to date with Hub Crawl events in your area and more at **ukie.org.uk/events-and-training**.

Location	Date
Brighton, Unity Technologies	4 March
Leeds, Avenue HQ	11 March
Liverpool, Baltic Creative	12 March
Leamington Spa, Playground	2 April
Bristol Games Hub	25 April

> **Find out more about securing investment for your indie game, and meet like-minded developers, at a Ukie Hub Crawl near you.**

Wizball

How songwriting experience can transform game design

Sensible Software's Wizball is a C64 classic, but it wouldn't have been the same without the studio's other passion: making music

WRITTEN BY
Kim Justice

In the beginning, you bounce almost aimlessly in a grey and sparse world, as you wonder if the game actually loaded properly.

The year 1987 was a high point for the Commodore 64. The system had been around for a few years, and coders were wringing the absolute best work from it; it was the computer's prime, evidenced by the likes of *Auf Wiedersehen Monty*, *Pirates!*, *IK+,* and *The Last Ninja*. Adding to this millionaire's row of classics was a strange yet innately playable game from an Essex studio by the name of Sensible Software – more accurately, the duo of artist-designer Jon Hare and programmer Chris Yates.

The game was called *Wizball*, and it truly put them on the UK software map. Having set the stage with 1986's *Parallax*, a title snapped up by Ocean Software on the company's very first day of business, Sensible's success – in particular, the critical adoration that was poured onto *Wizball* upon its arrival – showed that there were still plenty of young teams willing to keep the C64's line of great software going. But as young as Sensi might have been as a games company, their story actually goes back further, to a history that not only informed their games, but changed how they approached their creation when compared to other studios.

MOVING PICTURES
Jon Hare and Chris Yates were friends of friends whose paths crossed in the early eighties on the

> **"I went to Chris Yates' house and he'd done this weird spinny-ball thing"**

way to a Rush gig. Their shared interests in both progressive rock and similar school subjects quickly resulted in a firm friendship, as well as the formation of a band called Touchstone. Hare reminisces that, "we started writing music together very fast, and found a man who was a drummer… well, he said he was a drummer, but he was actually using saucepans for drums. Our first gig was terrible – he couldn't even drum to *We Will Rock You*, which is pretty basic."

The band would perform in Chelmsford, Southend-on-Sea, and various other towns on the Thames estuary for around five years, developing something of a following, and performing shows that could have only come from the creative minds of a bunch of youngsters with absolutely no money and a penchant for mocking each other.

In a fit of teenage angst over the break-up of his first relationship, Hare composed an earnest and quiet ballad called *Narcissus*, which, when performed, Yates would interrupt by playing a speaker-breaking widdly guitar solo. Hare and Yates would also wear rubber masks and dressing gowns as a budget tribute to Peter Gabriel.

"We wanted to make stuff that was successful," Hare points out with a hint of wistfulness, "but we also wanted to do art stuff, purely because we wanted to do it… and this would also be important to the games later on."

Now we're cooking on gas. Enemies everywhere, because we're starting to make things vibrant. And there's Mount Rushmore, for some reason.

SENSIBLE STARTERS

In the world of Sensible Software, there's no firm separation between the music years and game years – Touchstone was still ongoing, as a two-piece, while Yates taught himself programming by endlessly 'renting' ZX Spectrums from Littlewoods and Kays catalogues and returning them before he had to pay for them. The knowledge Yates gained from this landed him a programming job at LT Software in Basildon, and he was quickly able to get Hare on board after he impressed the bosses with some pixel art.

After making the Spectrum game, *Twister: Mother of Charlotte* for System 3, they left following a dispute with LT over the division of profits – and thus, Sensible Software was born in 1985. "Chris and I learnt that LT, who commissioned us, took 85 percent of the money even though we'd done all the work," Hare says. "And we were like, 'OK, we're not doing that again!'"

They quickly established a relationship with Manchester's Ocean Software, finding a champion for their games in Gary Bracey, the company's mercurial software manager. The pair also met a young Irish musician named Martin Galway, who would be tasked with soundtracking what Sensi put onto computer screens.

MAKE A JAZZ NOISE HERE

Sensible Software's approach to making games was different from the processes that are more common today. With no producer to oversee their work and no milestones to reach except for a firm deadline, they didn't plan an awful lot of their games out on paper, preferring to improvise on the screen itself. *Wizball* started out as little more than a circle and a line – there were no project guidelines or design documents to speak of, purely one of Yates' programming experiments.

Says Hare: "I went to Chris Yates' house and he'd done this weird spinny-ball thing, and he said, 'I've got this ball – it was just a circle and

The box art for *Wizball* was, as with most Ocean games, created by the late, great Bob Wakelin.

a line of land at this stage – you spin it and it bounces, and I'm not sure what to do with it.'"

The Sensible approach was to gradually add to this on the spur of a moment, in much the same way a band writes and then arranges a song. That circle and line can almost be thought of as the initial riff, something that comes into a musician's head and has to be hurriedly transferred to a musical instrument or hummed into a Dictaphone.

From here, Sensible Software's approach was iterative – they would continuously look at what they'd done and add more and more touches to it each time. First, there were the initial ➡

Balls quite literally fly in all directions as *Wizball* informs you, in no uncertain terms, to prepare yourself for some gamey goodness.

Gary Bracey, Ocean's lead software producer, had a rare nose for both the commercially successful and the unique outlier.

MAN AT THE TOP

Ocean's Gary Bracey transformed his company's fortunes with licensed games like *RoboCop*, but his creative streak and love for unique games made for a great relationship. Hare says: "Gary was a great champion for us – precisely what I believe we're missing in modern publishers. He presented us and protected us within the organisation as a creative force. He didn't feel the need too much to tell us what to do, and he was extremely supportive, which was great for us."
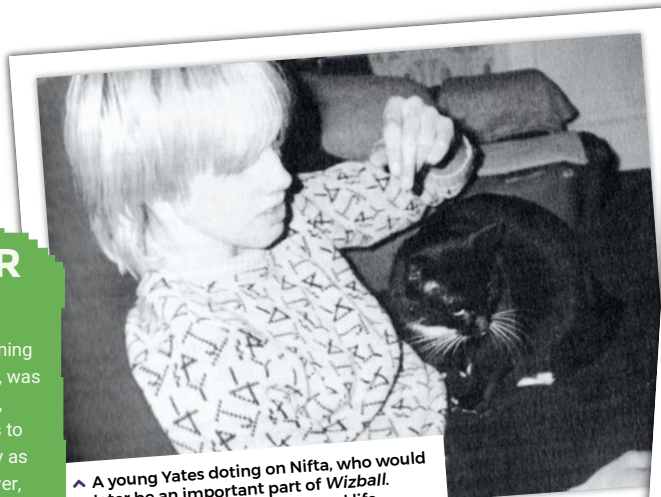
⌄ A young Yates doting on Nifta, who would later be an important part of *Wizball*. Fortunately not spherical in real life.

⌄ A big old flash of colour greets another filled paint pot. If this were a video, there would probably be an epilepsy warning.

⌃ Hare in 1987 outside a programmers' den that not only housed Sensi, but various other 1980s micro-luminaries.

brushstrokes: a circle that could be a placeholder for a ship, and a line that represents the ground. These could have been the foundation stones for a typical arcade shoot-'em-up, a genre which inspired Sensible greatly.

"We got all our inspiration from arcade machines," Hare recalls. "We'd be in the arcades, or pubs and chip shops where these things were everywhere, and we'd play them. Chris in particular was into games like *Nemesis* [also known as *Gradius*] and *Salamander* that were quite big at the time."

Wizball was particularly inspired by Williams' classic, *Defender*, but the unconventional movement of the ship – it's a ball that bounces, don't forget – also brings to mind Archer MacLean's *Dropzone*, a title where it's functionally essential to master both the X and Y axis.

With *Wizball*, it was decided that the player should work towards power-ups that would not only give the ball better weapons, but also allow

a much freer and traditional control over it. In a power-up system inspired by *Gradius*, the player can keep a stock of pick-ups and choose when to power their ball up, although the initial power-ups that give the player control over vertical and horizontal motion are the most essential.

THOSE WERE THE DAYS

This iterative process was, in many ways, the mark of a different time. Games weren't 'produced' in the same way they are now, and coders and designers were left to their own devices. It's time that Jon Hare looks back on fondly. "There wasn't such a thing as a game producer – the word didn't exist," Hare says. "You didn't have written game design plans – certainly in the UK we didn't – it was a bit more like constantly iterating, and all the way through Sensible Software this follows through."

After forming *Wizball*'s general structure – and giving the initial circle a smiley face and a spinning movement – the duo continued to add new flourishes, like so many harmonies, solos, and middle eights.

Unlike a typical shooter, the aim in *Wizball* is to pick up drops of colour and use them to colour in the world; you also have a smaller

⌄ A human Wiz mixes up paint in his lab, while you choose a power-up that'll stick with you, even after your inevitably stupid death.
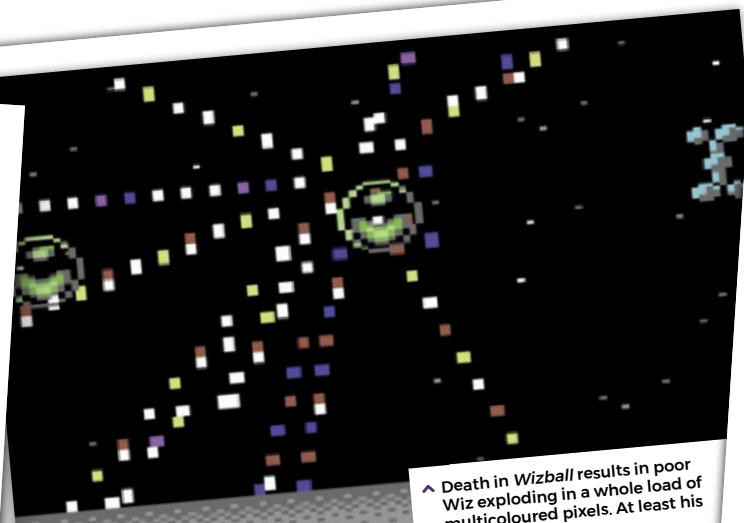
^ Death in *Wizball* results in poor Wiz exploding in a whole load of multicoloured pixels. At least his demise was pretty.

^ Hare and Yates with Martin Galway, who briefly became a fully-fledged member of the group before amicably leaving for Origin in the USA.

⌄ A young Hare and Yates plot their path towards world domination. Computer games probably not involved at this stage.

STATE OF PLAY

As someone with over 30 years and counting of industry experience, Hare says it's a shame that it's so difficult for those breaking into the industry today. "Generating a new brand… the failure rate is about 99 percent right now. Even when people feel like they've got their break making games, it doesn't mean that the game is going to be finished, or come out, or be a success… it's so hard now, and from the perspective of when it was easy, it's almost heartbreakingly hard. It doesn't need to be that way."

ball – later dubbed the Catellite – specifically designed for this purpose.

The addition of bonus stages, switching between different levels of a stage, and the hidden time limit you have on a level before the arrival of the dreaded Filth Raid, were all added before Hare and Yates even thought of a plot to tie the whole game together. The evil wizard, Zark, has drained all the colour out of Wizworld, and it's up to Wiz and his equally spherical cat, Nifta (named after Yates' real-life pet), to get rid of all the grey.

Hare recalls: "I'd been interested in colour from the lighting stuff I'd studied in college, so I said, 'Why don't we do this RGB mixing kind of thing?' Although we weren't sure whether it would be with paint or light… so in the end we did this thing where it starts off black and white and you add colours, and then put the story around it. This is the way these kinds of things often work – the mechanics drive what the narrative you make up around it is." ➡

^ Back in the office, Hare and Yates set about pounding *Wizball* into shape.
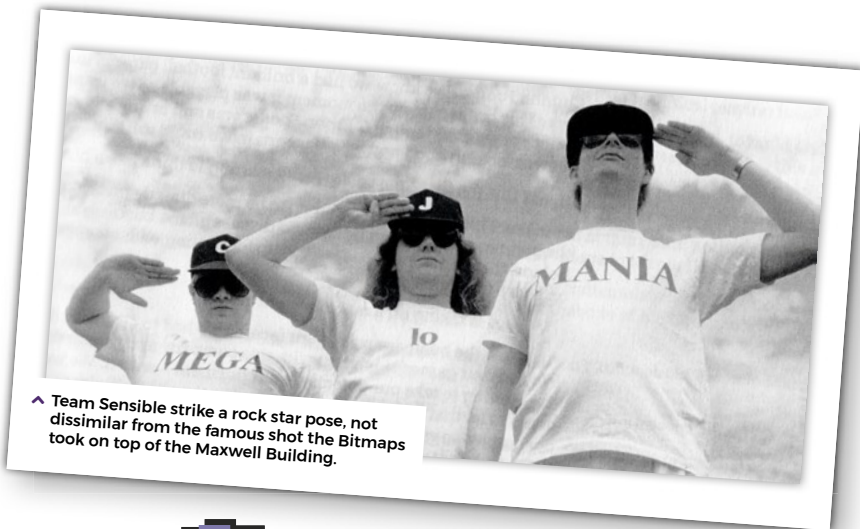
^ Team Sensible strike a rock star pose, not dissimilar from the famous shot the Bitmaps took on top of the Maxwell Building.

FRIENDLY RIVALRIES

The character that Sensible showed in their games was also shown in their press interviews at the time, and they would often be fighting for inches with fellow Essex studio, The Bitmap Brothers, for the spot of essentially being the rock stars of computer gaming. The unspoken rivalry would often come through in passive aggression – Hare recalls that they were named 'The Bitmap Wankers' in Sensi's Filofax – but the pair would later establish a firm alliance at the head of publishing house, Renegade Software, with Hare and Bitmaps founder Mike Montgomery beginning a lasting friendship.

MEATY BEATY BIG AND BOUNCY

When playing *Wizball*, there's a sense of being part of the creative process. Essentially, you start with an unfinished game: a bouncing ball that is only controllable in a very basic sense. After shooting down a bunch of stationary enemies, you get the pick-ups required to control the ship properly; from there, you can start to paint the world in. You're right at the front of the game, flying on top of it. If *Wizball* were a band, Hare and Yates would be the rhythm section and you, the player, would be the vocalist. Seeing as vocals are usually the last thing to be recorded in a studio, this way of looking at the game is justifiable, if not a little oblique.

Through all of the interesting touches, the game itself doesn't let up – the cocktail of classic shoot-'em-up mechanics never ceases to be engrossing, from the almost aimless fumbling and bouncing around at the beginning to the tight and fast experience you discover once you've fully got your motion. Thanks to this, *Wizball* became a critical if not massively commercial success, and it would be widely

⌄ The Filth Raid are fast, they shoot everywhere and are generally best avoided, so don't stick around.

regarded as one of the greatest games ever released on the C64.

Jon Hare's opinion of the game is equally positive. "*Wizball* is our best C64 game," he says. "I think it's when we found our feet as a creative force, which goes back to the music stuff – we already worked together as a pair anyway, and then you added Martin [Galway]'s music over the top. I remember the hairs on the back of my neck standing up when I first heard his music for Parallax, thinking 'this isn't normal!'. The C64 had such warmth in its sound and was essentially a proper instrument in its own right, and Martin's work was incredible."

PASSING YOURSELF COMING IN

Sensible's way of creating things wasn't just limited to *Wizball* – indeed, it would use the same philosophy to create games that were not only famed for their playability, but also for their character and humorous touches. Hare and Yates would continue to run through their games at every step; one of them would come up with some new idea, or a new riff on what was already there. And sometimes, this riff might change a project in a highly significant way.

> ### "When playing *Wizball*, there's a sense of being part of the creative process"

Where other teams might have given such things more thought to see if they could play with the original design, Hare and Yates would just go ahead and implement them – it's not as if they usually had an extensive design document to work from anyway. This practice reached its summit with the spiritual sequel to *Wizball*, 1992's *Wizkid* – something approaching an art game that still ended up being released by Ocean.

At other times, this experimental approach resulted in entirely unrelated new ideas. While working on the strategy title, *Mega-Lo-Mania*, and waiting for a build to compile, Hare had the idea of turning the game's map into a crude football pitch – he even used the game's little character sprites as players. This would stick with him, and become the genesis for what was perhaps the studio's seminal title: *Sensible Soccer*.

"We would say that what you throw away is as important as what you keep," Hare says. "So a
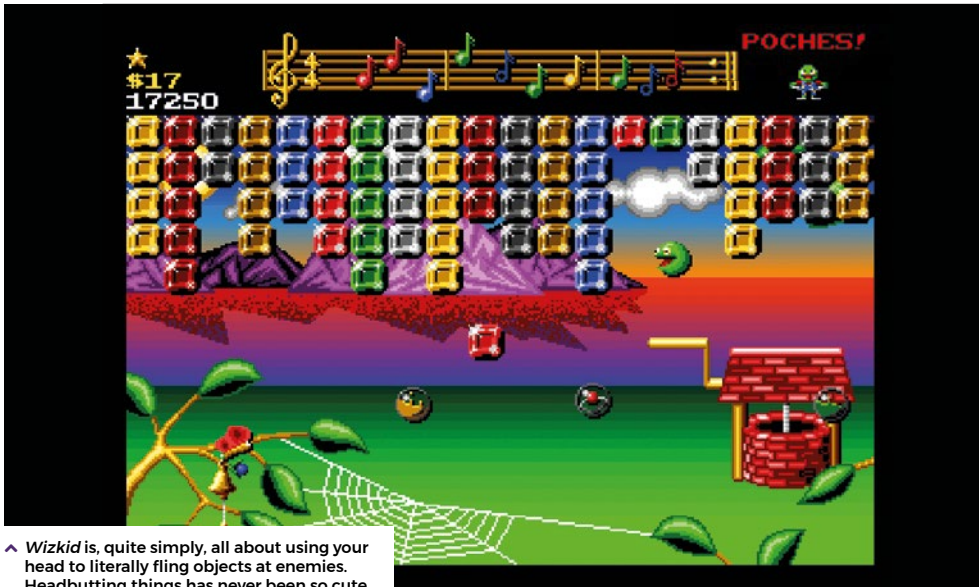
▲ *Wizkid* is, quite simply, all about using your head to literally fling objects at enemies. Headbutting things has never been so cute.

THE SECOND ALBUM

Wizkid, Sensible's 1992 sequel to *Wizball*, offers a whole new concept: a take on *Arkanoid* filled with typical Sensi flourishes and flat-out weirdness. "*Wizkid* was a good example of what we did," explains Hare. "We balanced our commercial games like *Sensible Soccer* with games like *Wizkid*, which is a very out-there art game. Both of them are very good in their own ways, and we were lucky to be able to do that and have Ocean, Renegade, and later Virgin [publishers of *Cannon Fodder*] supporting us for these different sorts of games."

lot of the things we tried in a game we'd just go, 'Nah, bin it' or 'Let's go back'. We did a lot of what we called 'backwards to go forwards' work. We were very much treating it like music – 'Let's try this out; no, that didn't work, but that thing you did before was good, try doing it again, doubling it up…' When we settled on it, Chris would write some code, I would do art, the words in the game… we'd go away and come back – which is how we were writing songs."

SUPERSTAR HEROES
When people think of Sensible Software and their games, music is often one of the first things that springs to mind – after all, their games were often soundtracked by indelibly catchy original songs composed by Hare and their official music man, the late Richard Joseph, or in the C64 days they would have appropriately epic SID tunes from Martin Galway (who would also briefly join the studio in the 1980s), as was the case with *Wizball*. Even *Narcissus*, the angsty teenage power ballad mentioned at the start of this article, would take on a new life in a video game, used as the recruitment screen music for 1993's *Cannon Fodder*.

The composing of music would be important all the way to the end, where it would be the central part of *Sex 'n' Drugs 'n' Rock 'n' Roll*, the Viz-esque rock opera adventure game that was sadly never released. But people might not have figured that rather than just being catchy tunes in their games, the process of making music was something that Sensible transposed to the very creation of their games, too – if it hadn't been

for those first years when Hare and Yates were more interested in writing songs and found an iterative process that worked for them, their games would have been completely different.

There's no end of stories in which people messed around with music and bands before they headed into the world of game development – with varying degrees of success – but in the case of Sensible, the band never really broke up. It was this unique approach to game design, particularly in the case of their breakthrough title, *Wizball*, that helped them become one of the great British games studios. In closing, Hare shares the moment when they suddenly figured out that, with *Wizball*, "we had total freedom."

"Ocean liked *Parallax*," Hare recalls. "They wanted our next game and didn't know what it was, it could have been anything – it was a bit like getting an album deal. They just wanted our next thing, and they trusted you." Ⓦ

▼ Only Hare and Yates – or rather Jovial Jops and Kuddly Krix, their oft-used nicknames – are good enough for high scores.

▼ We've got our cat, and we're in control! Careful, though: much like *Gradius*, one death will result in the loss of all those power-ups.

▲ *Assassin* **brought a console-like action experience to the Amiga.**

▽ **Team17 turned from Amiga to console with the PlayStation-exclusive shooter,** *X2*.

Developer Profile

Team17

Looking back to the earlier years of a British success story

▲ **Designed by Rico Holmes, top-down shooter** *Alien Breed* **was one of the Amiga's best action games.**

t might be a British indie publishing powerhouse these days, valued in the hundreds of millions range and popping out releases from all manner of different developers, but Team17 wasn't always a sure thing. It's a developer/publisher that's had its share of highs, lows, and a fair bit of time spent in the wastelands of mediocrity since its birth in 1990. So while we all know where the company stands – and how impressive it is – these days, we thought it would be fun to look back to those earlier, formative years for the Amiga's best friend. Though not Amiga Power's best friend…

The studio's formation runs – in abridged form – thus: chain of computer shops in the UK forms publishing house, 17-Bit Software, in order to get exclusive dibs on games it signs for the shop.

Young Swedish developer catches eye of publishing house; is introduced to other art-and-design talent to make future games look as good as they play in new dev studio, Team 7. Idea spreads that maybe 17-Bit should make its own games in-house. Discussions happen, 17-Bit and Team 7 merge, Team17 is born.

There's a lot more to it than that, of course, but the simple fact is this was a merger brought about not by necessity or greed, but through a straightforward series of occurrences. It all just made sense. Combining the UK wing's eye for industry trends with the Swedes' ability to knock out better games than you'd expect on a small budget resulted in early hits like Team17's first title as an entity, *Full Contact*. This one-on-one fighter rode *Street Fighter II*'s coat-tails straight into the middle of a bunch of

▲ Team 17 chief executive officer, Debbie Bestwick.

eager, hungry Amiga owners, and sold incredibly well considering it was a budget title from an unknown studio.

This initial success was repeated plenty of times in Team17's first few years, and by 1993 half of all Amiga game sales could be attributed to Team17, whether down to titles it had developed internally or just published, like 1992's *Strider*-alike, *Assassin*. Hitting the top of the charts near enough every time, bagging high scores across the board, and bringing the Amiga (and, let's not forget, the PC – though usually in years-later MS-DOS ports) a bunch of game genres we figured were going to remain the exclusive domain of consoles and never make their way to Commodore's machine. Life was good.

DIET OF WORMS

It wasn't all candy and roses, mind – the aforementioned *Amiga Power* magazine was notably severe in its treatment of Team17's games, and spoiled the party to the point that the studio actually launched legal action against the publication in 1995. Would you be surprised to hear said lawsuit went nowhere? And, to be fair, *ATR: All Terrain Racing was* rubbish.

Still, the honeymoon period was over by the mid-nineties, and Team17's wilderness era was about to begin... at least after one small release.

> ### "To be fair, *ATR: All Terrain Racing* was rubbish"

Worms wasn't supposed to be as successful as it ended up being. It was meant to sell about 50 or 60 thousand copies, total, not the millions it went on to sell around the world on most formats you could think of. A daft little one-man project for the Amiga blew all the way out of hand, and soon enough became – logically – quite the cash cow for Team17. It was onto a good thing; why bother mixing it up and taking risks with projects that might not pay off?

Sure, there were other releases – the odd new game here and there, a few remasters or reboots of the early classics – but Team17 very much became 'the *Worms* studio'. Since 1995 there have been over 25 *Worms* titles and spin-offs, and very few of them have resulted in a level of success remotely similar to that of the original few releases.

Should Team17 have stuck with this approach, we surely wouldn't be where we are now. But a pivot to more publishing-focused duties, a restructuring – which saw Debbie Bestwick, present since day one, move to take full control of the company – and a flotation on the stock market has repositioned Team17 as a cornerstone of the British independent publishing scene. The company went from plucky upstarts, to Amiga heroes, through *Worm*-y villains, and rebirthed itself as something absolutely vital in modern gaming. If that's not a success story, we don't know what is. ⓦ

Team(17) for hire

In the dark ages of Team17's worst *Worms* addiction, there was the odd blip here and there where a game without the aggressive annelids in it would surface. And, rather oddly, a few games even popped up made by Team17, but based on pre-existing licences. It was something of a short-run experiment, with only three of these games from other franchises appearing. Still, we got to see Team17's take on *Lemmings* for PlayStation formats in 2006, a regrettable foray into the world of *Army Men* in the same year with *Major Malfunction*, and one of the worst games ever made in 2009's *Leisure Suit Larry: Box Office Bust*. All things considered, it was probably for the best to abandon the contract work approach.
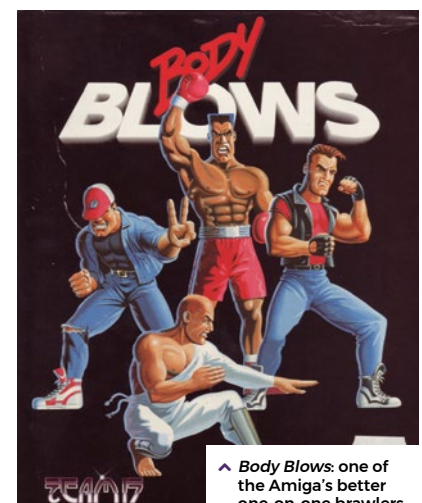
▲ *Body Blows*: one of the Amiga's better one-on-one brawlers.

Seven-year itch
10 games from Team17's first 7 years
The Amiga did get a hell of a lot of love back then
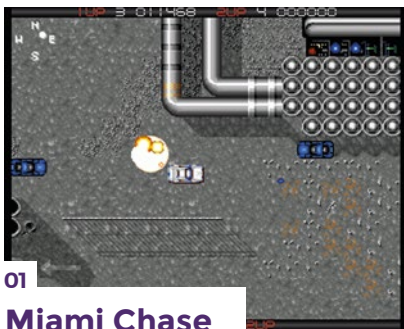
01
Miami Chase
Amiga — 1990

Team17 didn't exist yet, but the seeds were sown with this *Miami Vice*-inspired racer. Developer Team 7 was formed thanks to 17-Bit Software's Martyn Brown, who saw potential in programmer Andreas Tadic's earlier work. *Miami Chase*, published by Codemasters, led to the Team 7 and 17-Bit merger, so we have it to thank for Team17's formation.
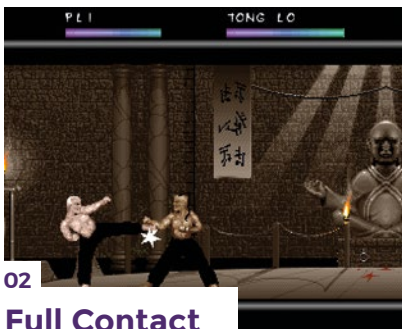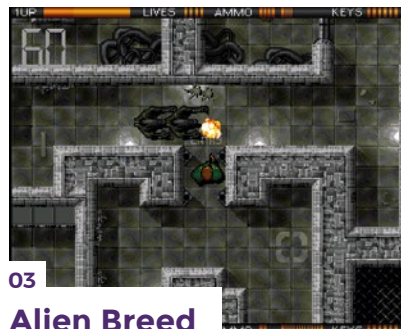

02
Full Contact
Amiga — 1991

The birth of Team17 needed a special launch to back it up, and *Full Contact* was just that. OK, so it's of its time, but this one-on-one fighting game combined pacey action with beautiful graphics and fluid animation. Plus it cost a tenner – an absolute bargain for the time. The newly-formed company couldn't have knocked it out of the park any harder.

03
Alien Breed
Amiga / PC — 1991

Answering the question of 'what if *Gauntlet*, but *Aliens*?', the first *Alien Breed* offered intense challenge and frenetic action. Its multiple sequels continued in the same tough, top-down shooter format, with the series eventually branching out into the *Doom* clone territory with *Alien Breed 3D* and its sequel – both of which helped the Amiga go out on a high.

04
Project-X
Amiga / PC — 1992

Another stroke of genius from Team17, *Project-X* was a shoot-'em-up influenced by *Gradius* and *R-Type*. The Amiga had seen arcade ports before, but few original arcade-style shooters, and *Project-X* both looked the part and, were it a game you had to pay for each play, would have greedily gobbled up all your 10p pieces. Because it was too hard.

05
Superfrog
Amiga / PC — 1993

Once again, Team17 looked at what was elsewhere, saw a gap on the Amiga, and plugged it with its own version. *Superfrog* was a challenger to *Sonic* and *Mario*, and while it could never hold a candle to the competition, it was still a good laugh. Also, along with *Zool* and *RoboCod*, it acted as a lesson in product placement. Ah, delicious Lucozade.

06
Body Blows
Amiga / **PC** – **1993**

Again, Team17 looked to the consoles, saw something popular (*Street Fighter II*), and brought it back to the Amiga. And PC. But mainly the Amiga. *Body Blows* was brilliant fun, a fast-paced fighting game made specifically for Commodore's machine so not suffering the ignominy of cutbacks others ported to the Amiga had seen. All together now: NINJAAA!

07
Arcade Pool
Amiga / **PC** – **1993**

A lo-fi release on Team17's part, *Arcade Pool* was… well, just that. It was a pool game with arcade leanings – rather than the sim-oriented approach of, say, *Jimmy White's 'Whirlwind' Snooker*. Simple, with accurate physics and engaging mechanics, it performed surprisingly well for something that didn't take its inspiration from a popular console game of the time.
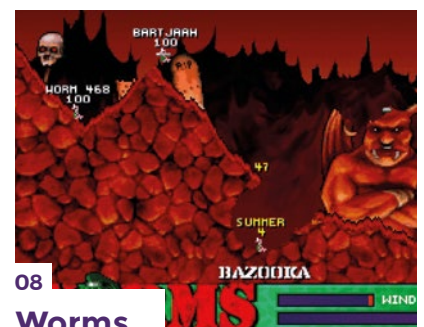
08
Worms
All of them – **1995**

Andy Davidson's *Total Wormage* became Team17's meal ticket for a long time, and while it's easy to decry the company for relying so heavily on this one franchise, you can at least see why it did. *Worms* took the *Scorched Earth* template, added a bunch of weapons, threw in a sense of humour, then sat back and let you have a metric worm-ton of fun. A superb game.
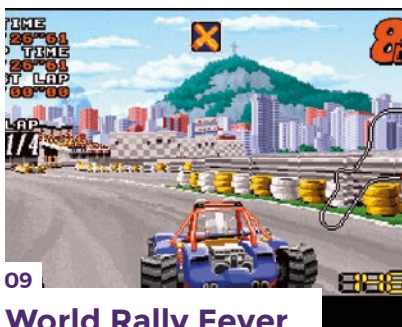
09
World Rally Fever
PC – **1996**

Worth featuring here not necessarily because of its quality – though *World Rally Fever* was a good, fun, arcade-style racer – but because this marked the first time in its existence Team17 had *not* released a game on the Amiga. The era wasn't quite over, but with Commodore's machine suffering a comedy of errors and dying spectacularly, it was an understandable move.

10
Worms: The Director's Cut
PC – **1996**

Team17's last Amiga game was a fantastic love letter to the system. An exclusive sequel with an enhanced engine, new weapons and more tweaks under the bonnet, *The Director's Cut* was the original *Worms* games' pinnacle. Naturally, after all the care and attention, it sold a reported 5000 copies worldwide. Sigh.

Try 12 weeks
for only £12

Crackdown 3

△ The auto-lock aim takes a lot of skill – and enjoyment – out of the combat sequences, especially when there's a lot of enemies running at you.

Too little, too late, too dated

△ The cel-shaded, stylised cutscenes are easily one of the best things about *Crackdown 3*.

Crackdown 3 is late and – good lord – does it show.

From the over-the-top introductory sequence to the bombastic set-pieces to its own self-assured jokes, *Crackdown 3* thinks it's a whole lot cooler than it actually is. It's the gaming equivalent of a one-night stand from 2010 that's unexpectedly cartwheeled back into your life again. You'd kinda-sorta liked them at the time, admittedly, but now you've grown up a little, moved on... whereas One-Night Stand is in the same clothes, sporting the same haircut, and using all the same old lines, perplexed and confused that you're no longer sniggering at the jokes they unashamedly nicked from old *Scrubs* episodes.

And that's the problem here, really. It's not that I hate the steady rinse-and-repeat of jailbreaks and orb collecting and liberating locations from all the evil Bad Guys; it's just that I don't particularly enjoy them, either. Despite its punk-*Tron* aesthetic and balls-to-the-wall over-the-topness, *Crackdown 3* is astonishingly dull, lacking both ambition and creativity as it festers away in its nine-year-old rut.

Yes, it offers a cool play-your-own-way progression system, inviting you to move through each of the city's districts at your own pace as you clear out the bad guys, liberate the

good guys, poke your head into alleyways, and leap across rooftops in search of those elusive orbs. But since 2010's *Crackdown 2*, sandbox play has moved on in huge, impressive and genre-redefining ways. We have the beautifully realised worlds of *Red Dead Redemption 2* and *Assassin's Creed Odyssey* now. There's a very real chance we're becoming incredibly spoilt, I suppose, but *Crackdown 3*'s sandbox just isn't big enough or exciting enough to compare to its peers in any way that isn't unfavourable.

Even the story's a bit meh. You play as an agent sent to restore order throughout New Providence after an evil megacorp assumes control with a regime of poverty, oppression and lashings of propaganda. Any chance of meaningful social commentary, though, is lost in *Crackdown*'s brutally unsubtle delivery.

Rather than walking through the poverty-stricken shanty towns at the edge of the city and reflecting on the evils of Terra Nova's corporate greed, instead I'm just leaping from one rusty, corrugated steel roof to the next, scanning the horizon for the next orb to add to my collection. I intercept the cynical, cold communications from one SuperBad to another, but I'm too busy climbing the tower and trying not to fall off – mid-air movement is alarmingly fickle, and there's a very real chance I'll slip to my doom if I mistime this jump – to pay the message

New Providence is a great place to explore, with plenty of secrets and collectibles hidden around the map.

There's plenty of vehicles to get you around, although towards the end of the game, you might find it quicker to jump from building to building.

any attention. And to be honest, if the game isn't all that bothered whether I'm taking out the bad guys or the innocent bystanders that accidentally stumble into my path, why should I be? Occasionally, the game will make a half-hearted attempt to take me down if I play a tad too chaotically, but this is easily contained – or avoided altogether.

It's all the more frustrating, then, that the visuals used to tell this lacklustre story – the cel-shaded, comic book-esque sequences – are truly glorious. While this, too, is getting to be something of a well-trodden trope by now, these cutscenes are done with enough care and flair to stand out in stark contrast to much of *Crackdown 3*'s other humdrum elements… of which there are plenty, unfortunately.

The combat – if repetitive – feels smooth and solid enough, and there's a good range of weapons from which you can pick 'n' mix the right combination for you, and the different flavours of enemies you'll bump into in the distinct districts of New Providence, but auto-lock all but obliterates any need for skill.

"For all its boom and bluster, *Crackdown 3* is unremarkable"

Initially, it was loads of fun, swaggering through the shoot-shoot sequences, mowing through bad guys and revelling in being a badass… but it doesn't take long for this, too, to feel unnecessarily repetitive as the same waves of identikit enemies throw themselves in front of my hail of toxic gunge.

As for the rest of it? I mean, it's fine, I guess. It's not terrible, but it's not particularly engaging, either. Vehicle sequences are fine. The missions peppering the map are fine, if repetitive. The multiplayer Wrecking Zone is a little less than fine – it's chaotic, but not in a good way, and the novelty of disintegrating everything you see grows old pretty quickly – and even the Terra Nova boss battles you take on once you've weakened their hold are nothing to write home about, either. It's mechanically sound, and it doesn't take itself too seriously – but there are plenty of games that do this already.

Crackdown 3 is at its best when you actively avoid the chores it throws at you and simply explore, ignoring the intrusive map icons and instead leap untethered through the nooks and crannies of its bright, bolshy world. While some might have hoped its continual delays would've resulted in more shine and polish, instead *Crackdown 3* has dropped like an old relic misplaced in time and space, outclassed at every level, in almost every way, by its more contemporary peers. For all its boom and bluster, *Crackdown 3* is unremarkable and, unfortunately, forgettable, too. ⓦ

This boss fight wasn't anywhere near as cool as this screenshot suggests it could've been.

HIGHLIGHT

Crackdown 3 is never better than when you're soaring across the skies and moving from building to building, scouring the world for secrets. The game's curiously addictive collectibles – a rainbow smorgasbord of orbs that operate as the game's progression system – are a sweet palate cleanser to its otherwise rinse-and-repeat activities.

VERDICT

Some say better late than never… sadly, we'd disagree.

53%

HIGHLIGHT

After relaxing genetic laws in America, my friend Zhi emails me. We've hung out at my apartment. They've opened up to me. We've sung Disney karaoke. This time, Zhi's tone is frantic. Their skin is greying from gene therapy. *Spinnortality* is great at showing you the human consequences of your business decisions.

Despite its heavily abstracted interface, your actions often have very human consequences.

Review

Spinnortality

Ah, everyone decided to wear trenchcoats for casual Friday

As what is often jokingly referred to as 'the darkest timeline' of current events spreads into 2019, the original tenets of cyberpunk continue to look less like a paranoiac's fever dream, and more like accurate predictions for the future. So it's no surprise to welcome something like *Spinnortality* to the world. We are, chronologically speaking, sixteen years away from William Gibson's *Neuromancer*. We have entered the year *Blade Runner* was set. I flew back from L.A. last week, and though I was thankfully spared a Voight-Kampff test at airport security, the sense of each one of my hurried cigarettes mingling with a noirish fog of international instability and uncertainty was eerily tangible.

As CD Projekt Red's *Cyberpunk 2077* looms, we're also faced with the ironic facade of a product marketed on an aesthetic critique of the very systems necessary for its existence, and one that it subtly attests to the legitimacy of. A multimillion dollar blockbuster requiring 3TG conflict minerals both to produce and run, the profits from which will enrich the company's shareholders. Like all entertainment products of its size, for some it will be an outlet, an

expression, a work of art. For most of us though, it's just another opiate. Something to distract us while social media companies undermine democracy, wealth inequality grows, and human rights become trite afterthoughts in the steady pursuit of profit. Gnarly augmentations aside, cyberpunk is starting to look as tame in comparison to reality as Good Charlotte do to GG Allin.

Cyberpunk management sim, *Spinnortality*, is a one-man, crowdfunded project. So if nothing else, it's got that 'punk' suffix down pat. It's how this novel and well-written, but ultimately sedate and simple strategy expresses that 'Cyber', however, that produces both its most interesting moments and biggest flaws.

"All we ask is that you make us immortal," request a boardroom of shadowy figures in *Spinnortality*'s introduction. Actually, the first thing they say is, "Welcome to the boardroom, Godzilla," because I have a thing about naming myself Godzilla in business simulation games. These five clandestine titans of industry, like all obscenely wealthy sociopaths, are having difficulty coming to terms with the fact that they're not actually gods. My task, as head of Wireframe.Inc, is to fix this slight genetic hiccup for them.

Time in *Spinnortality* is divided into turns, each of which represent six months. Each turn,

When launching products, you'll need to be mindful of a nation's political and social leanings.
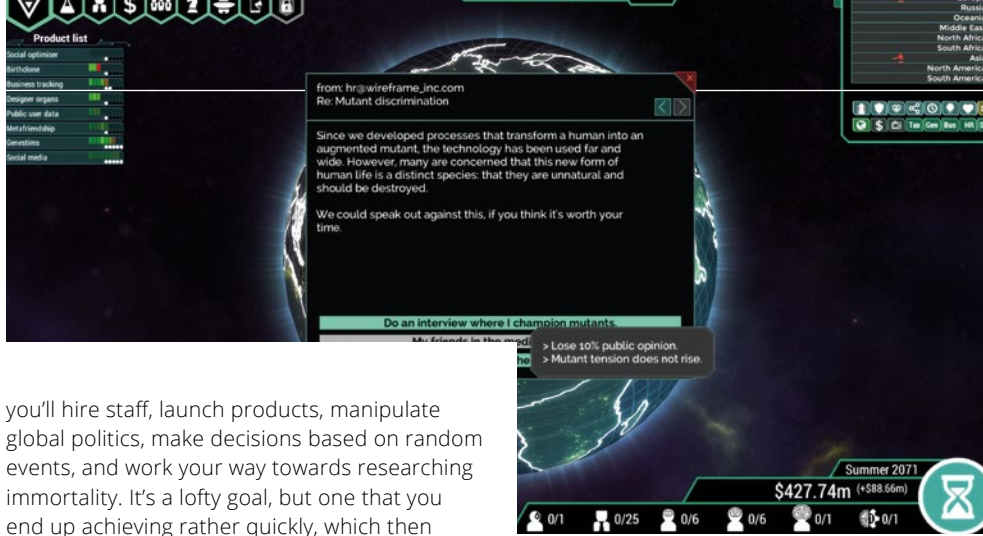
you'll hire staff, launch products, manipulate global politics, make decisions based on random events, and work your way towards researching immortality. It's a lofty goal, but one that you end up achieving rather quickly, which then leads you to a choice of four victory conditions: Imperial, Consumerism, New World and Humane, all of which revolve around what sort of governments control the globe, how much these governments like you, and what their attitude toward taxation is.

So, you launch products by researching and predicting trends, make as much money as possible, and use that money to influence nations to your own, likely nefarious, ends. At the same time, you'll have to deal with global events, inefficiency and corruption within your own company, and the ever-present whims of the boardroom. In terms of play, this generally comes down to balancing numbers, planning ahead, and making sure to diligently and repeatedly research, launch and relaunch products.

Product hawking is later replaced by a wider and more complex long game of buying shares and bribing governments to get certain laws passed, and this is all bolstered by an additional layer of resource acquisition and risk/reward options for corporate subterfuge. *Spinnortality* is both pleasingly complex and elegant in its implementation of these many interplaying systems, and learning how to balance your economic and political goals is rewarding. Once it shows its hand, however, a honking great, neon-glowing issue starts to surface. Namely, that the actual process of progressing through the game just isn't as compelling as it needs to be to play in anything longer than short bursts without feeling a bit burnt out with it all.

Having to constantly launch and relaunch new products, for example, starts to grate after the initial novelty of reading the flavour text. Researching always leads you to fully comprehensive information on the most profitable options, so it seems like an 'auto-launch' option would have worked just as well here. As interesting as the game's writing is, I often found it difficult to find the motivation to go through the motions of progress once I'd worked out what I needed to do.

When *Spinnortality* is at its best though, it uses these simple systems to make witty, scathing critiques of late-stage capitalism's farcical excesses. It manages to turn the phrase 'systemic critique' into a design philosophy as well a political one. Simply put, the game becomes much tougher if you decide at any point you'd rather leave the world better than you found it.

Each turn, a new event pops up, and sometimes, you're given a choice that feels like no choice at all. You might destabilise a country in order to build up a more politically advantageous regime in its ashes. Months later, you'll receive an email from a woman who broke her nose in the ensuing riots. Do you apologise, or deny responsibility? Both will affect the public's opinion of you, which in turn will piss off the boardroom and edge you that little bit closer to losing that game, but looking the other way penalises you slightly less. Numerically, it's a simple choice. Do the right thing, and you'll have to make sacrifices. Ultimately, this 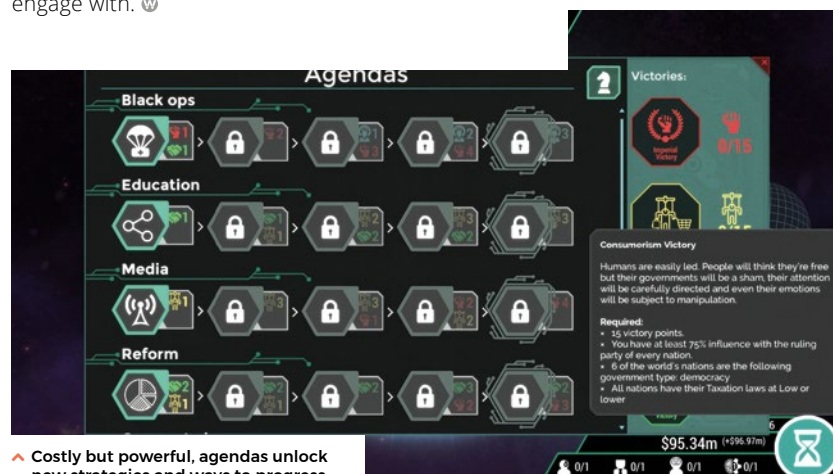scarily on-point simulation makes its points well, but sometimes embodies the circular drudgery of the mindset it parodies too closely to be as fun to play as it is interesting to engage with. Ⓦ

"Do the right thing, and you'll have to make sacrifices"

^ **Costly but powerful, agendas unlock new strategies and ways to progress.**

Pixar finally makes its debut, with worlds based on *Toy Story* and *Monsters Inc.* both being highlights of the whole game.

HIGHLIGHT

Thanks to a shift to Unreal Engine 4 part way through development, *KHIII* looks absolutely sublime, and at times the worlds can be indistinguishable from their source materials. The magic abilities are visual feasts, and the engine's ability to support more enemies than previous entries is put to good use. If nothing else, *KHIII* is really good at being a spectacle.
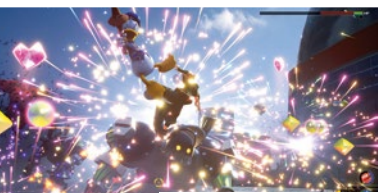
This is actually a calmer moment in the average *KHIII* encounter.

Info

GENRE
JRPG / Action

FORMAT
PS4 (tested)
/ XBO

DEVELOPER
Square Enix

PUBLISHER
Square Enix

PRICE
£49.99

RELEASE
Out now

REVIEWED BY
Joe Parlock

VERDICT

Kingdom Hearts III is a functional bookend to a long-running series, but in its plot-driven heavy lifting it loses the magic and whimsy of its predecessors.

64%

Kingdom Hearts III

A long-awaited conclusion mislays some of the old magic

After 18 years of narrative threads being tangled together like cables at the bottom of a junk drawer, *Kingdom Hearts III* races towards the final confrontation. Sora, Donald, and Goofy must travel to more Disney-flavoured worlds in a last-ditch attempt to gain enough strength before Master Xehanort gathers his forces and darkness prevails. It's all very high stakes, but does this final chapter deliver what we've been waiting for all these years? Mostly not.

Kingdom Hearts III's big problem is, with just seven Disney worlds and no *Final Fantasy* characters, it's forgotten the main thing that pulled players into the series way back in 2001. It doesn't help that the worlds used never hit the highs of previous games' Space Paranoids, Timeless River or Neverland. The inclusion of Pixar is long overdue, but was anybody really asking for a world based on *Pirates of the Caribbean 3* or *Big Hero Six*?

When it doesn't feel sparse, *Kingdom Hearts III* feels incredibly busy. The core of the game is still light JRPG mechanics and frenetic hack-and-slash encounters, but the series has been picking up new systems almost constantly over the years, and *KHIII* hasn't done much spring cleaning to make it legible for the player.

Alongside keyblades and magic, you've got *Dream Drop Distance*'s shotlock (guns) and freeflow (parkour) systems, reprisals (counters), weapon form changes, new theme park-inspired special attacks, a new open-world Gummi Ship system, the Gummiphone... the systems never end. As gorgeous as the game is (and boy, does it look fantastic with its change to Unreal Engine 4), fights are a meaningless – and patronisingly easy – blur of lights and sounds as Sora spams out every special move possible. It's a stark contrast to the highly choreographed and iconic fights of the previous games.

Avoiding spoilers, wrapping up the years-running story is one of *Kingdom Hearts III*'s strengths. *Kingdom Hearts* has always been about emotion and aesthetics over logical consistency, so it doesn't make a lick of sense, but that doesn't really matter. It manages to pull all these disparate plot points together in a cohesive and affecting way, tug on heartstrings, fall in love with characters all over again, and leave you singing *Simple and Clean*. Which is all it really needed to do.

Kingdom Hearts III is like pizza; there is no bad pizza, and there is no bad *Kingdom Hearts*. Chilling with Sulley and hearing Elsa belt out *Let it Go* make your heart feel warm and fuzzy, but in its focus on wrapping up the story, it's lost the magic of its predecessors. Mechanically messy and painfully small, at the very least it finally gives us some closure on friends we made almost 20 years ago. ⓦ

> **"There is no bad pizza, and there is no bad *Kingdom Hearts*"**

> Hot damn, heroes, that was UH-MAY-ZING! Your action-packed antics have utterly undone that old hag's veil of evil!

Review

▲ The goddess of compassion Guanyin pops up as your occasional guide. Despite her traditional appearance, her dialogue skews more on the contemporary side.

Unruly Heroes

Ex-Rayman devs work their magic on a classic Chinese tale

Unruly Heroes isn't so much a retelling of the great classical novel of Chinese literature, *Journey to the West*. Rather, it's a reworking that turns its central cast into video game mascot characters for a 2D platformer. Aside from that time Damon Albarn made an opera out of it, *Journey to the West* isn't all that well known *in* the West, although it's inspired countless loose adaptations, from *Dragon Ball* to Ninja Theory's post-apocalyptic retelling, *Enslaved*. And with that in mind, Magic Design Studio sticks much closer to the original's picaresque adventure through Chinese folklore.

It does look an absolute treat, with a hand-painted style that's like concept art come to life. If it makes you think of *Rayman Legends*, then that's because the studio consists of former members of those games' design team. Even though this is made with Unity rather than the UbiArt Framework, the commonalities are clear.

While your four distinct heroes lend themselves to couch co-op, flying solo still lets you switch between characters with a tap of the left bumper, which comes in handy in a tag-team fighting game when a character with low health needs swapping out. Some puzzles are also designed around specific character skills; Monkey can use his staff as a makeshift bridge, while Pigsy inflates himself to float upwards to otherwise unreachable areas. These are quite simple, though, as they can only be activated by a nearby statue resembling the relevant hero.

Along with some chase sequences and challenging multistage boss fights, *Unruly Heroes* offers fairly standard platforming fare, although its melee combat lacks the punchiness of *Guacamelee*, and knockbacks are irritatingly common when rushed by smaller enemies. Checkpointing can be inconsistent, where dying in one section might let you respawn as another hero where you left off, while another may force you to restart a bit further back. Having to fill a gauge by defeating enemies before you can activate a checkpoint also seems a frustrating requirement, mostly felt in the late game where more precarious platforming challenges mean you'd rather not dither about attacking enemies.

> *"Unruly Heroes* has the spice of variety on its side"

Yet *Unruly Heroes* has the spice of variety on its side, and any whiffs of annoyance are quickly forgotten as another cracking idea emerges to mix things up. You might find yourself taking control of a defeated foe's abilities, or encountering puzzles where you have to pay attention to both foreground and background, or riding a celestial cloud that can shoot lightning bolts beneath it. Journeying into the Underworld was a personal highlight, where hell is both depicted as a bureaucracy and an unusual take on a typical minecart level.

For a game consisting of no more than 30 levels, it's generous with its inventiveness, still chucking new mechanics at you in the final stretch. Besides, any platformer with art direction that makes you forget you're in yet another ice or fire-themed world is doing something right. Ⓦ

Info

GENRE
Action platformer

FORMAT
Switch (tested) /
XBO / PC

DEVELOPER
Magic Design Studios

PUBLISHER
Magic Design Studios

PRICE
£17.99

RELEASE
Out now

REVIEWED BY
Alan Wen

VERDICT

Unruly Heroes is a beautiful platformer with enough inventive variety that prevents occasional annoyances from breaking its spell.

75%

HIGHLIGHT

Art Sqool automatically saves your assignments to a portfolio folder on your computer, meaning you can keep your most absurd creations forever. Perhaps Glander's next game will provide a similarly surreal gallery of these drawings.

Your Q-shaped lecturer will grade your creations based on colour, composition, linework and approach. Sadly, none of this actually matters.

A Photoshop-style toolbar lets you diversify your creations with new swatches and brushes, all of which can be found on campus.

Review
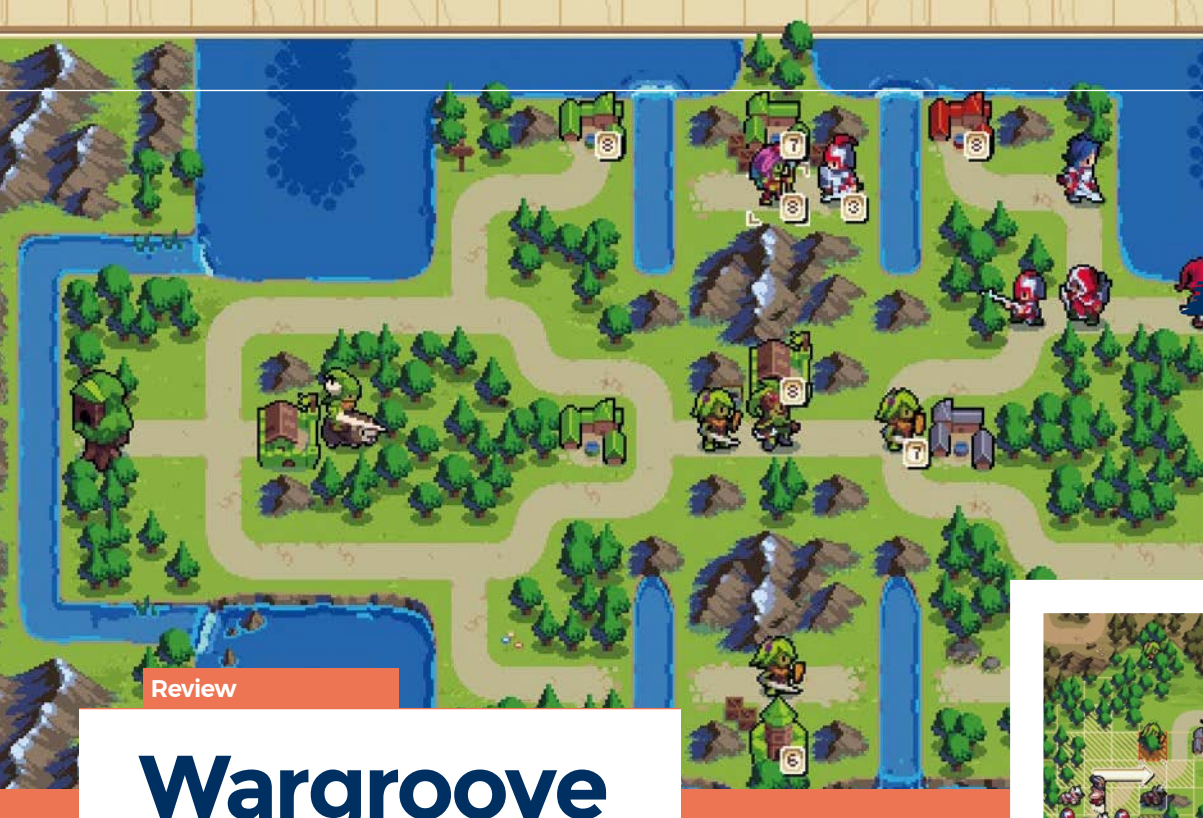
Art Sqool

Julian Glander's virtual campus is way too cool for sqool

VERDICT

Art Sqool's bonkers world demands attention, but it lacks basic features that would make it worthwhile.

69%

There's something ironic about reviewing Julian Glander's *Art Sqool*. You almost feel like you've fallen into his trap. The New York City-based artist has waded into the world of video games with his latest endeavour, preceded by galleries worth of shiny blobs and silky textures, glossy internet-age art that has made Glander a viral hit, landing him illustration gigs at The New York Times and Adult Swim.

Art Sqool slides you into the chunky shoes of Froshmin, a new student on this delightful pastel campus that is brimming with muses, from wireframe frogs to wingding billboards. Your eyes will flicker constantly between the imposing form of these creations and the lilac static of the sky.

Glander's creations, combined with the surreal score, curate an arresting atmosphere that is difficult to ignore. *Art Sqool*'s soundtrack feels hybridised from the work of artists like Iglooghost, George Clanton and Angelo Badalamenti; it's moody and ethereal but constantly transforming, with a variety of instruments being thrown into the mix when you least expect it. This is important, as it never feels like it's repeating when you're wandering around campus.

Your goal is to follow assignments given to you by your AI mentor that range from absurd to verbose, from a meticulous recreation of an air conditioner patent drawing to something that simply makes you smile. Thanks to the ambience, the latter assignment becomes an oddly introspective task.

The shining achievement of *Art Sqool* is how it makes you feel comfortable with just creating something. Every creative spirit knows the pain of staring at a blank canvas and self-flagellating over where to begin. In *Art Sqool*, it's hard not to want to capture something that reflects the rubberised forests and chewy chess boards in front of you.

Better still, you can keep your creations once you wrap up your studies, as each assignment is saved in a little portfolio folder on your computer, ready to be shared on social media.

Yet, *Art Sqool* unfortunately falls apart when it ventures into the tricky world of game design. Despite Glander's artistic talent and good intentions, *Art Sqool* is not intuitive to control. The camera's locked to your movement, which makes appreciating some of the best exhibits on campus more difficult than it should be. The canvas and art tools, meanwhile, take up far too much of the screen.

The only way to navigate to higher planes is by jumping, which looks awkward and feels like a sloppy compromise. On top of this, the AI adjudicator has a real sense of judgement, meaning you can just fire in scribbles and pass the 50 assignments with disappointing ease.

Maybe the missing features are intentional, and Glander is just repackaging the veracious message that art is never finished, just abandoned. Unfortunately, beyond being a delightfully silly jaunt into the left side of your brain, *Art Sqool* fails to fully find its feet as a compelling video game. ⓦ

Weather conditions from fog and sandstorms to rain and wind can shift the dynamics of a carefully planned advance.

Review

Wargroove

Groove is in the art... of war

A glance at a few screenshots should tell you what *Wargroove* is trying to be. Fantasy setting aside, everything from its character design and colour palette to its grid-based maps and combat animations bears a clear resemblance to turn-based strategy classic, *Advance Wars*. And so it is in practice. If you're a veteran of Intelligent System's series, you'll find *Wargroove* uncannily familiar.

In fact, Chucklefish have transplanted the *Advance Wars* rule set and style almost wholesale into their game. It's not only the basic concept of trying to wield the strengths of your army's units against the opposing force's weaknesses that's revived. It's that every unit type in *Wargroove* is a near identical twin of one in *Advance Wars*, and matches up against every other type in the same way. Yet while *Wargroove* lacks creative ambition, this detailed replication gives it strong foundations. There's even something admirable about its commitment to the source material.

Besides, it does introduce some extra strategic layers of its own. The most obvious change from *Advance Wars* is that army commanders actually take to the battlefield, acting as super units that can deal serious damage and unleash special powers, with the risk of instant defeat should they fall. Meanwhile, each ordinary unit type can trigger a critical attack by fulfilling a certain condition, such as standing next to another of the same unit or charging in from maximum range.

It all adds up to an experience that feels expertly assembled and balanced. But that's not always quite enough. It doesn't help that the genre has moved more in recent years towards tight, lean designs that inject a sense of urgency and make every turn count. In a world of *XCOM 2* or *Into the Breach*, *Wargroove*'s core loop can feel flabby and lacking intensity. In campaign mode particularly, missions can drag, despite the game's attempts to vary objectives. Inching towards a victory that's seemed inevitable for over 20 minutes is never a joy.

It's disappointing as well that the impact of commanders is ultimately quite muted. The only thing that differentiates them is their chargeable powers, or 'grooves', and these often prove unimportant, especially compared to their potentially game-changing equivalents in *Advance Wars*. The rhythm of battles thus becomes more monotone, and choosing a commander is less tactically interesting than it might have been.

If the notion of a new *Advance Wars* appeals, however, none of this should be too off-putting. The addictive old processes of building and positioning units to counter your opponent remain highly absorbing. As an homage to a past great, *Wargroove* remains welcome, even if that reverence stifles its self-expression. ⊚

Caesar the dog is the stand-out character in an otherwise fairly bland line-up.

VERDICT

Wargroove successfully recreates a winning formula, but plays things a little too safe.

72%

HIGHLIGHT

While the mood is serious and hard-boiled as your senses numb to the rising body count, the glitchy physics will occasionally throw a curveball, spinning the last faceless goon you shot before rag-dolling him into space. Hey, at least It's a good laugh.

If you can make out all the bullets and even your own cross-hair amidst these effects, chances are you're superhuman.

Review

The Hong Kong Massacre

More like John Woe, am I right?

VERDICT

A terribly disappointing cross between *Hotline Miami* and John Woo flicks with the wit and execution of neither.

44%

As its title suggests, *The Hong Kong Massacre* takes inspiration from the Hong Kong action films of the early nineties, where the game is also set. But if you're imagining a charismatic leading man to rival Chow Yun-Fat, big shots pointing guns in each other's faces, or a bonkers, memorable set piece, you'll find none of that here.

What first-time developer VRESKI does take from the John Woo playbook is the ability to slow down time, to give the ultra-violent spray of bullets some balletic grace. Except in video games, this has also been done to death, from *Max Payne* to *Vanquish* to *Red Dead Redemption* – hell, there was even *Stranglehold*, which was itself a direct sequel to Woo's 1992 action opus, *Hard Boiled*.

What *The Hong Kong Massacre* tries to do is marry this with indie hit *Hotline Miami*'s top-down perspective, twin-stick controls and merciless one-hit death. It doesn't, however, share any of that game's vibrant style, opting instead for a dreary hard-boiled tone, with a one-note plot about an ex-cop waging a vendetta against a crime syndicate. It's forgettable stuff, and the occasional – shockingly low-res – CGI cutscenes do little to give personality to a protagonist who looks so bland it's almost offensive.

The photorealistic aesthetic also does the game no favours, with the recycling of identikit environments, from back alleys, rooftops, restaurants and drug dens, barely distinguishable from one another, but also impacting on readability. Because once the guns go off, so do spark and smoke effects, all of which conspire to frustrate your ability to parse just where the bullets are coming from or where the hell your little white cross-hair has even disappeared to.

Slowing down time can only get you so far, so tapping R1 allows you to dive under gunfire, hop over tables or jump through windows in true action-movie style. Executing these moves, however, locks you into a canned animation whose timing is difficult to gauge; often, you won't know whether you can move again until a bullet snuffs you out. All of this confirms just how ill-suited the game is to a top-down perspective.

When each level can be beaten in a minute or two, upping the difficulty to ensure a player doesn't blaze through the game in one breezy sitting makes sense. Adding enemies with psychic powers who can shoot with 100% accuracy through obstacles you thought were cover, however, is a major test of the player's patience.

There may be incentives for replaying levels and beating them within a set time, without slowing down or with perfect accuracy, but when it's all underpinned by the same tedium and frustration, slowing time feels more like a curse when you'd rather just fast-forward. Ⓦ

While blurry close-ups of the last goon you bump off in a level might be a stylistic choice, low-res cinematics doesn't seem to be deliberate.
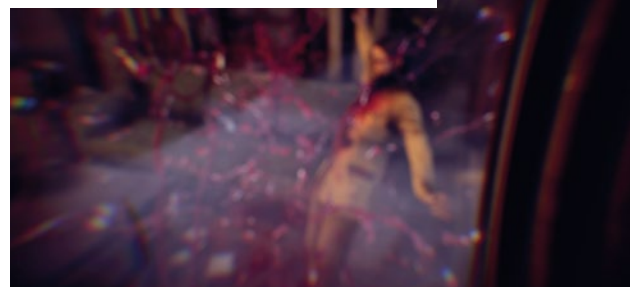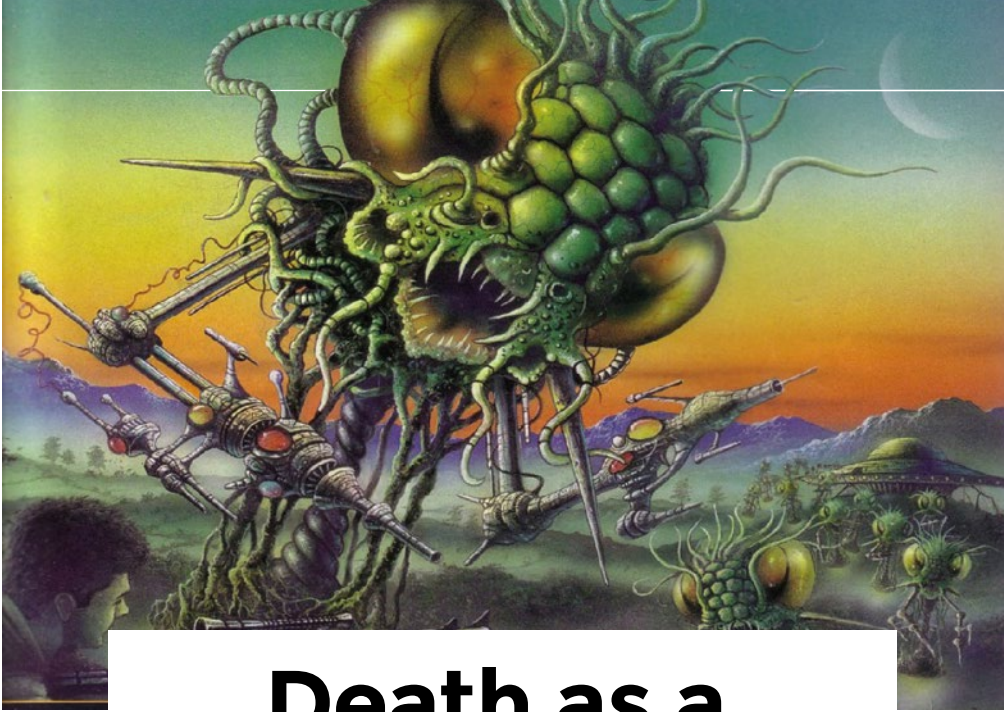
Death as a learning experience

The allure – and challenge – of UFO: Enemy Unknown is still strong

It's been 25 years since *UFO: Enemy Unknown* launched, and in that time what we've mainly done is argue about whether that's what it should be called, or if we should go with the American title of *X-COM: UFO Defense* (or *X-COM: Enemy Unknown*, as it was in the PAL PlayStation version). Why do we do this? Well, because if we focus on anything else, all that will happen is we'll get angry and terrified in equal measure. Still. After 25 years. Because what a game *UFO-COM* is.

It's a turn-based strategy game with management aspects baked in – build your base, research the tech, turn your rag-tag paramilitary into a bunch of world-saving badasses (with awful hair), that sort of thing. But what's always set *UFO* apart is just how unforgiving it was, is, and forever will be. It's easy to forget until you go back, lured in by the stench of nostalgia, only to have your best troops killed when they take their *first step* off the landing craft. Fair? No. Character-building? Absolutely.

This isn't something that's dissipated with age, either. In a time of *Dark Souls* and, well, the remake of *XCOM*, people are keen to be challenged. *UFO* doesn't challenge you. It kicks you in the face and laughs, and you can't do anything about it. Dominance firmly established, the tension kicks in. You get better, some of your team survives, you research armour and can survive longer, the tide starts to turn… then the Chryssalids show up, implant your troops with parasites, and your whole world turns upside down again. What a game!

But there's always one thing holding *UFO* back for me – even though I can, will, and do re-play it on a regular basis, especially when a new turn-based strategy game has popped up (say hello to *Wargroove* on page 63), I really don't think this is one that has aged well. Bear with me here; it's not that the actual mechanics haven't stood the test of time: they're still exceptional – if sometimes unfair. No, it's all about that user interface.

Icon-based controls, clumsy clicking and re-clicking, positioning and navigation handled by… well, mid-nineties AI… doesn't bode well for a positive user experience, at least not for the newcomers. I struggle to see many younger players bothering with much more than half an hour's aborted attempt at getting to grips with *UFO*, because quite frankly, it's a bit of a pain in the backside.

But what that does do is make the Gollop brothers' finest hour all the more special – all the more of an elite little club for us to be haughty in. We can all share war stories, safe in the knowledge that few of those young whippersnappers will invade our midst.

We can revel in the hilarity of not getting that 'floss' dance, and chuckle at how, back in our day, looking up and down in an FPS was a luxury.

UFO: Enemy Unknown has ended up being, somewhat ironically, a bit of a safe space for us. What a game! Ⓦ

> **"Fair? No. Character-building? Absolutely"**

Wireframe Recommends
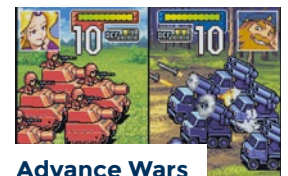
Advance Wars
GBA, DS, 3DS
In hibernation it might be, but the *Advance Wars* games *still* rank up there as some of the best the turn-based strategy genre has to offer. Why there hasn't been a new one in years, we do not know.

Final Fantasy Tactics
PS1, PSP, IOS, ANDROID, GBA
The original, *War of the Lions* do-over, and *Advance* spin-off all get a look-in here. One of the smartest, deepest turn-based strategy series ever made, and still superb fun whatever format you play on.
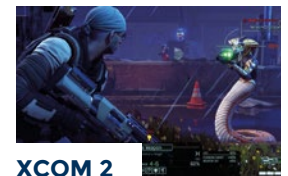
XCOM 2
PC, PS4, XBO
The first was great; the second truly amazing. This is what happens when actual fans of the original *UFO* get to make a reboot. Intense, smart, difficult and fun, nobody expected things to be this damn good.

Time Crisis

It only took one step to reinvigorate the flagging light gun genre

NAMCO / **1995** / **ARCADE, PS1**

By the mid-nineties we'd had our fill of light gun games. Greats arrived after the period, of course, but generally speaking we were pretty over it. *Time Crisis* popped up from behind some boxes in the back and shouted "ACTION!" in our faces, though, and suddenly the plastic guns and dodgy accuracy genre was vital again.

Mad Dog McCree tried to make us care by letting us shoot people acting poorly as cowboys. *Virtua Cop* succeeded in bringing a wealth of challenge to the light gun genre, but its perfection of the existing form did little to evolve it. Namco's first *Time Crisis* greeted players arriving in the bustling arcades of the era with something different: a pedal.

You were still dragged around a level on rails, with the same sets of enemies leaping out with little regard for their own well-being – such is the life of a nameless henchperson – but that pedal was a sea change. With your foot off it, you were hidden in cover, impervious to attacks and automatically reloading behind the crate/wall/ornamental fountain which would be your home for the next few seconds.

Pressing a foot down would launch the player up into the fray, allowing them to pop off shots at the massed bad guys and, thus, progress through the level. The only thing was, being out of cover meant you were always vulnerable to attack. So, of course, you'd spend all your time with your

> **"So it was that a simple pedal was able to revitalise a genre that had peaked"**

foot *off* the pedal, right? Of course not – it was called *Time Crisis* for a reason, said reason being you were very much up against the clock.

So it was that a simple pedal was able to revitalise a genre that had peaked; by instilling in it the ability to provide sanctuary, to ratchet up the tension, to throw in an extra element of both strategy and skill, and simply by giving the player more control over a light gun game than they'd ever known before. One wide metal plate was all it took – another daft, clunky peripheral added to a genre that made its name on other daft, clunky peripherals.

The effect was diluted once *Time Crisis* arrived on the PlayStation, with Namco's official G-Con 45 (or Guncon if you weren't in the EU) not including a pedal, instead relying on a button press for all those added extra elements. Obviously it worked fine and didn't take away from the mechanics of the game in any way, but it always felt like something was missing. Fortunately, plenty of third-party peripheral-makers were on the case, and a few of us were able to source our very own gun-pedals at home.

It's strange to look back on what *Time Crisis* did for the light gun genre, living as we now do in a world devoid of these games. The simple addition of a flat bit of metal to slap your foot on made Namco's clock-fancying blaster so *crucial* a game. Now? Well, there's nary a bright orange handgun to awkwardly point at a screen. ⓦ

Next Issue

Lost Words

Beyond the Page

Rhianna Pratchett talks exclusively about her narrative-driven adventure

Don't miss out! Subscribe PG **54**

Also

> The secret of making a great Zelda-like adventure

> Rig and animate a third-person character in Unity

> Inside the game-changing puzzler, Baba Is You

> Why user research is game design's best-kept secret

wfmag.cc