

AUDIO DEVELOPMENT

It's an oft-repeated phrase: 'Games are following in the footsteps of movies.' The comparison is regurgitated so frequently, it's rarely questioned. Yet while the wider issues are open to some debate, the early history of movies shares some remarkable similarities with the early history of videogames.

Consider the first examples of moving footage: blocky, grainy, and black and white. Ditto computer games. Movies were originally silent; the same was true of games (indeed, the computing equivalent of an organ player at the front of the theatre was a C90 cassette on a nearby hi-fi, playing early 1980s pop tunes). And as audio standards improved in films, so the wider entertainment industries began embracing the medium. We're at a similar stage now where EA games boast more licensed music tracks than most compilation albums.

It wasn't like this in the early days, when maverick musicians were offered tiny sections of memory for audio tracks and sound effects. Simple, mono sound effects operating within a few hundred bytes were the audio capabilities of the first wave of dedicated consoles. As a result, most games only boasted a couple of grunts by way of a soundtrack.

Back then, game audio was a luxury few could afford because the system RAM was so low. Specific tools were few and far

between. Coders developed their own routines, which usually involved programming individual notes into a hex editor. It was time-consuming and laborious – but as the challenges were so great, those early musicians pushed each system to its limits, and enjoyed doing so.

It wasn't until 1983 that Matthew Smith broke new ground in so many ways with *Manic Miner*. One of the most important advances was its first use of in-game music – a suitably awful rendition of *In the Hall of the Mountain King* from Edvard Grieg's incidental music to Henrik Ibsen's play, *Peer Gynt*. For the time, it was an impressive achievement, integrating the audio and sound effects on the one mono track.

Rapidly, each new hardware platform began to allow further progression. Three-channel sound appeared on the VIC 20 in 1981. Home computers of the early- to mid-80s even began offering end-user sound programs – the C64, for example, boasted a peripheral that fitted over its standard keyboard to offer a very simple synthesizer.

The most famous musicians of the era were Rob Hubbard, Martin Galway, and Richard Joseph – and the C64 in particular, with its complicated, digitally controlled SID chip, really drove in-game soundtracks. To this day, these original tunes are re-recorded and remixed, with many available on retro music CDs.



BRITNEY
Britney Spears released her own videogame *Britney's Dance Beat* [above], which attempted to piggyback the popularity of the *Dancing Stage* games. While it was a glossy, stylized affair, featuring Britney's most popular songs of the time, it was utterly misjudged, offering a tepid version of a complicated genre. The choreography was lame – surprising, considering the on-stage exertions of Ms Spears – and the game short-lived.



EA
Electronic Arts has invested an extraordinary amount of time and money in its in-game audio. Standardized across a delivery format called 'EA Trax', the firm works with up-and-coming and established brands to provide the in-game soundtrack to the majority of its sports titles such as *NBA Live 2004* [above]. Artists such as Franz Ferdinand, Robbie Williams, and Queens of the Stone Age have all worked with the publisher to include their own songs in EA titles. It's predicted that games are becoming a more powerful audio delivery tool than radio for a certain demographic. The evolution of the service will be to buy the tracks you're listening to.



NOMAD SOUL
French developer Quantic Dream embarked on an interesting collaboration with David Bowie for its *Nomad Soul* (1999). Bowie, along with Reeves Gabrels, wrote the entire soundtrack exclusively for the game, consisting of eight songs: *Thursday's Child*, *Something in the Air*, *Survive*, *Seven*, *We All Go Through*, *The Pretty Things Are Going To Hell*, *Omikron (New Angels of Promise)*, and *The Dreamers*, seven of which appeared later on David Bowie's album, *Hours*. He also featured in the game itself [below] along with his wife, Iman.



MEGA LO MANIA
Mega Lo Mania (1991, above) is widely considered to be the first game that employed professional voice actors. The game [above] hailed from much-loved UK developer Sensible Software, who liked to fool around. The original recordings for the *Mega Lo Mania* game had one of the main characters swearing. This was trimmed for the game to just say, 'No way!' although the original audio files were still stored on the discs.

TALKIES
The advent of the CD (now DVD) as the delivery format of choice allowed developers and audio technicians to experiment with professional voice-acting. This was a radical sea change, yet one which caused unexpected outcry. Games such as *Day of the Tentacle* [above] were often released in two versions: one subtitled and one with spoken audio, which ran the danger of infuriating passionate gamers by featuring voices different from those they had imagined.



The 16-bit machines offered musicians even further options. Sampling – although it had been possible in a very limited capacity beforehand – became a practical way of further enhancing soundtracks. And things improved even further in the late 80s when Sega's Mega Drive offered FM synthesis. It was also around this time that the first PC soundcards were released. Prior to this, developers had to rely on the PC's internal Texas Instrument sound chip – capable of only the most basic of sounds. And the SNES boasted sophisticated audio power for the time – eight independent channels and advanced sample manipulation.

The real breakthrough in music soundtracks, as with so many game innovations, came with the PlayStation, which combined a modest amount of audio RAM – 512K – with serious offline storage. Musicians could opt to load the entire soundtrack and sound effects in at the start, or stream from CD as needed. This former method resulted in an often retro quality: *Final Fantasy VII*, for example, sounded almost SNES-like as its audio was retained in RAM.

These days, audio directors have ever-increasing specifications to experiment with. The PlayStation 2 offers 48 channels of sound, across audio RAM of 2MB. PlayStation 3 will amplify this further. The challenge nowadays is what to do with it all...

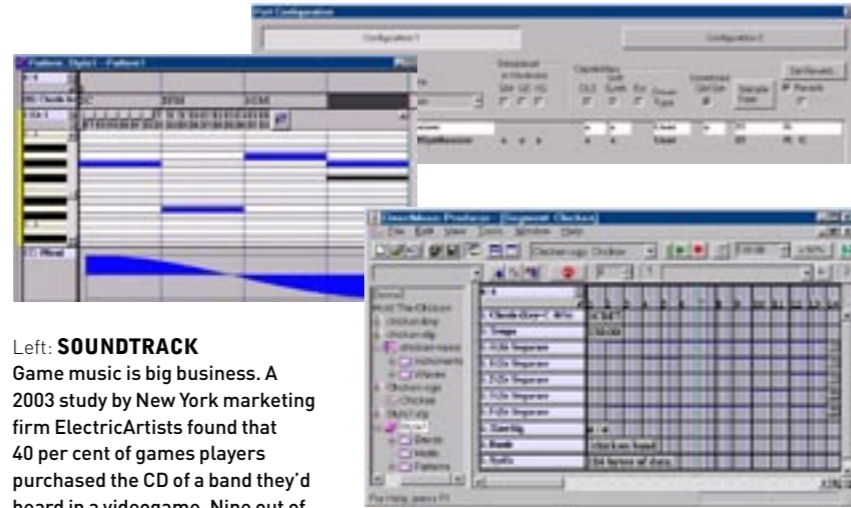
AUDIO TOOLS AND TECHNICAL CHALLENGES The tools used to compose, record, edit, and implement videogame music vary from team to team. Many audio technicians who've grown up with the industry are loyal to one particular package. Like learning a new operating system, sometimes the documented performance benefits are outweighed by the inconvenience of feeling your way around a new software package, particularly in a discipline that embraces so much experimentation.



The truth is, there's no one package endorsed by all. RenderWare's dominance of the independent development sector makes it an obvious choice for those already using other elements of the package. Subsequently, using RenderWare Audio is often a convenient decision – and its diverse feature list, small memory footprint, and portability makes up for the power of an independent suite of software.

DirectMusic, GameCoda, Nuendo, and SoundForge are other popular choices out of a confusing array of competing applications. Aspiring game audio engineers will often learn a tool from a mentor and stick with that for many years. Typically, the features offered by the major packages are present in many others – so portability from platform to platform does need to be considered. Regardless of how competent an operator of various applications an individual can be, all of it is worthless if that program does not integrate effectively with the game engine. For that reason, many large developers working on bespoke engine technology will also create their own music drivers.

The best game audio hails from teams or individuals in continual touch with the design and development team. Long gone are the days where a musician would be presented with a brief which listed sections of music and required sound effects and told simply to get on with it.



Left: **SOUNDTRACK**
Game music is big business. A 2003 study by New York marketing firm ElectricArtists found that 40 per cent of games players purchased the CD of a band they'd heard in a videogame. Nine out of 10 respondents in the survey of 1,000 hard-core gamers said they remembered a game's music after they stopped playing it. Good news for Good Charlotte, who appeared most recently in *NFL Street*.

Above: **DIRECT MUSIC**
Released as part of the DirectX 7 SDK, DirectMusic – the API – and DirectMusic Producer are Microsoft tools designed to plug composers into the heart of its operating systems. Applications that use the DirectMusic application programming interface are not restricted to playing content from DirectMusic Producer. They can play wave and MIDI files, and can construct music and sound effects by playing individual notes on available DLS instruments. However, the full power of DirectMusic can be unleashed with files authored in DirectMusic Producer.

Adaptive or interactive soundtracks offer the greatest scope for creativity. And a modern game can require hundreds of individual sound effects – excluding any recording dialogue. This is why audio requirements need to be drawn up at the design document stage. As well as presenting a framework for the overall structure and volume of sound required, it'll also offer a hint of how realistic any audio aspirations are.

It's important to consider which sections of the game actually require music. Like enthusiastic kids colouring with crayons, it's often possible to cram too much into a finite space, ruining the overall effect. Music pacing is a mechanic gamers understand; when the score goes quiet, they know they're in trouble.

Movie musicians score over rough cuts. While this is rarely possible in games – test levels often bear scant relation in all but the most basic mechanics to finished code – it's vital to see the game in production throughout. Integral teams have this

Below: **FABLE**
Movie composer Danny Elfman recently composed the main theme for Lionhead's *Fable*. It was an extravagance, for sure, and of questionable contribution to the game's overall success. But that's not the point – not only will Elfman's involvement encourage other high-profile composers to tackle the medium, but Lionhead's attitude to audio is encouraging for interactive musicians who must often fight to get a larger share of the total budget.



Below: **SOUNDFORGE**
Soundforge is the industry standard for editing sound files and has long had excellent support for MP3. It's a meaty program with an intuitive interface and superior speed – vital when processing modern soundtracks. It can batch-process audio files and can also accept video files. Loop creation and spectral analysis capabilities complete the picture along with the capability to rip from and write tracks to a CD.



Right: **NUENDO**
Aimed at the audio for film, video, and interactive media markets, the Nuendo Media Production System is a modular system based on an audio software application for the Macintosh and Windows 98/NT/2000 platforms, which includes several hardware accessories. Nuendo features up to 200 tracks of 24 bit/96 kHz digital audio, advanced-featured surround mixing, a Video-Track, and MIDI-Tracks, along with the most comprehensive functions for digital audio available.



advantage over external musicians, but even storyboards and movie files when available can provide inspiration on the central themes, if not the timing.

Frequently, traditional rules need to be ripped up. Composition classes often emphasize the importance of repetition – and videogame music may have historically been famous for tunes based around a single central theme – but tastes have changed. Modern gamers expect diversity. A 30-hour game needs more than 30 minutes of music.

Don't be afraid to experiment. Sound can often hail from unusual sources. Specific sound-effect libraries can prove expensive, while home-grown audio can be used in unusual ways – standard animal noises reprocessed in imaginative ways, for example, can provide effective monster noises.

This type of cost-conscious attitude is often vital. Research has shown that videogame players can rate the audio track forming as much as 30 per cent of their overall enjoyment of a given game. However, typically, funds are all too often allocated to other areas over and above sound – with often less than five per cent of the total development budget under the control of the audio director.

But don't let all this discourage you. Game audio is being increasingly recognized – both by the industry itself and those outside of it. BAFTA (the British Academy of Film and Television Arts), for example, acknowledges and honours the very best in-game audio. There is a trend for top talent such as Danny Elfman – alongside contemporary pop musicians – to be eager for their work to be featured in games. It was recently suggested that modern gamers hear more audio through computer and videogames than anywhere else. Maybe the Buggles were right after all: Videogames killed the radio stars.

INNOVATIVE USE OF GAME AUDIO



Above, left to right: Three images from *Rez*, developed and published by Sega.



REZ

A game that has to be heard as well as seen, *Rez* divided the gaming community with its innovative use of visuals and audio – but to those in the know, it's regarded as one of the most important games of the current era. Hailing from Sega's United Game Artists stable, who had previously developed the insanely catchy and stylish *Space Channel 5*, it's an audio-visual tour de force. Players assume the role of a hacker sent into a computer to eliminate a virus. So far, so *Tron*. But *Rez* excels through the simplicity of the game, the beautiful design, and the interactive audio.

Each level begins with a simple techno beat. Enemies appear on screen and can be shot individually, or grouped together in a high-scoring chain. Gameplay-wise, it's as on the rails as you get – but the real interaction comes from the TV's speakers. Every time an enemy is targeted, it emits a beat – be that a synth, a rim-shot, or any other type of electro pulse. As they're destroyed, they also emit a sound – not an explosion as such, but yet more beats. Everything is timed to perfection. Rather than producing a rambling, jumbled mess, the techno soundtrack works brilliantly, even appealing to those who despise the music-game genre.

As the level design and enemy frequency increase in intensity, so does the underlying beat. Players discover themselves tapping their feet along with their fingers, as the hypnotic fusion of simple vector graphics, metamorphosing upgrades and incessant, pulsating audio combine to create a unique gaming experience.

Rez can be completed within an hour. But that's not the point. Most who fall for its seductive beauty find themselves treating it more like an album than a game – simply skipping back to the start and listening to it all over again.

FIFA SERIES

On the pitch, the *Pro Evolution Soccer* series beats the *FIFA International Football* series about a million-nil. But the latter's enduring populist appeal is undoubtedly down to EA's trademark polish, and this is nowhere more evident than in the sophisticated commentary technology.

The 2004 incarnation of the series is reputed to have used over 20,000 separate lines of commentary, split between the resident pundits. As a feat of logistics it's impressive enough, but the skilful weaving of these disjointed lines into coherent and sensible observation is what really astounds.

Coding routines to recognize on-screen player actions and pull voice from a bank of sounds is a challenge. Konami's *Pro Evolution* series fails miserably, and is nothing more sophisticated than streaming 'He shoots!' (or similar) whenever the shoot button is pressed. When players are in the box, this is obviously relevant. But when the button is hit accidentally from inside the player's half, it's as amateurish as it is inappropriate.

EA, however, has implemented a sophisticated system that analyzes context as well as button presses and joystick waggles – with almost total success. The banter between the two men on the microphone appears almost natural – certainly as lifelike as we can expect from an audio system stringing random lines of text together. Little touches, such as player names recorded in various levels of excitement and the illusion of memory, as commentators recall earlier misses, complete the illusion of spontaneity. And add to this the authentic crowd noises – complete with celebratory cheers and derisive boos – and it's the total package.

Below: *FIFA 2005*, published by EA.



MONTY ON THE RUN

When Commodore 64 magazine *Zzap! 64* awarded *Monty on the Run*'s music 98 per cent, it was more a commentary on the state of in-game audio than the quality of that particular soundtrack – amazing as it was for the time. It was, however, an indication of the shape of things to come, as its composer, Rob Hubbard, went on to dominate the airwaves for a generation of games. While Rob's innovative methods for squeezing the utmost from simple hardware aren't relevant for today's powerful home consoles, they do offer thoughts for writing contemporary music for limited specifications, such as the emerging mobile platforms, for example.

Back in the day when simple sound chips offered little scope for high-quality audio, Rob crammed so much life into the three-channel audio that it compensated for the game's derivative platform approach – possibly the first example of music genuinely enhancing the overall experience.

Rob built his own music utility, which was able to take hi-fi samples and compress and arrange them into code suitable for the 64's SID chip. For *Monty on the Run*, he managed to squeeze in pleasing violin samples and a guitar solo; literally unheard-of in those days. While these restrictions don't apply to today's audio technicians, the spirit of experimentation lives on.

Below: *Monty on the Run*, published by Gremlin.



A prolific composer, Rob was responsible for more than 75 Commodore 64 games between 1985 and 1989, producing classics such as *Sanxion*, *Zoids*, and *International Karate+*. His distinctive style – which he applied brilliantly to diverse genres, both gaming and musical – proved so popular, that his credits on a game could boost sales.

Rob still composes, but mainly for his own pleasure. In 1989 he joined Electronic Arts, providing soundtracks for its turn of the decade titles. He remains there to this day under the guise of audio technical director.

GTA: VICE CITY

Yet another example of the GTA series as trail-blazer. The in-car radio has been used throughout the series, but it really came into its own in *Vice City*, a game which showed the industry how to cooperate effectively with the music business.

As *Vice City* is based in the '80s – the decade much of its audience grew up in – the soundtrack is comprised almost entirely of original '80s music. Of course, the developer was helped by the enormous success of *GTA:III*, which dominated the charts the year prior to *Vice City*'s release. Its sales obviously convinced the record labels of the benefit of widespread in-game distribution, rather than simply demanding significant and restrictive licensing fees.

Above: *GTA: Vice City*, developed by Rockstar Games and published by Take 2 Interactive in 2002.

The soundtrack took eight months to assemble – a huge time to allocate to audio. The decision to pursue such a wide range of music was dictated by the game's free-roaming style of gameplay. *GTA III* boasted a soundtrack of around three and a half hours, which is lengthy in anyone's book – but consider the fact that players could enjoy upward of 100 hours roaming Liberty City and one can appreciate how its open-ended structure could have been compromised by repetitive audio.

Vice City's soundtrack spanned seven CDs when it was released separately, each crammed with a distinctive style that reflected the playlists of the individual music radio stations. Users were given the option to change radio stations – though not individual tracks – at will, the idea being to create an audio canvas that would reflect the type of action the player was involved in. Players soon decided upon favourite stations for particular pursuits, be they cruising along the beach, tearing through streets on high-speed pursuits, or simply causing random havoc. Sure, it's interactive audio in the most basic sense, but the way it's done is an absolute triumph.

RICHARD JOSEPH, AUDIO DIRECTOR, ELIXIR STUDIOS

Richard Joseph started out in the games industry in 1986, working with publisher Palace Software. Since then, he's worked variously for The Bitmap Brothers, Millennium, and in a freelance capacity before joining Elixir Studios where he occupies the position of audio director.

Over the course of his career, Richard has composed and implemented soundtracks for over 100 games, including such seminal classics as *The Chaos Engine*, *Speedball 2*, *Gods*, *James Pond*, *Mega Lo Mania*, and *Sensible Soccer*. His company Audio Interactive produced the soundtrack to *Theme Park World*, which won a BAFTA award. Most recently, Richard and the team at Elixir have completed the score to *Republic: The Revolution* and *Evil Genius*, which was also nominated for a BAFTA.

**How did you get involved in writing music for computer and videogames?**

Just prior to joining the industry, I bought a Sinclair Spectrum. It absolutely fascinated me. I was keen on music technology anyway – always experimenting with synthesizers, tape recorders, things like that. This was before MIDI came out; we were playing around with various early computer-based keyboards like Emulators and Fairlights, though they were very expensive. It was an interesting time. The technology was very basic. Yamaha brought out a thing called the CX5M, which I've still got. That was relatively cheap thing – something like £500 or so. It was a dedicated music computer, with eight mono-channels using FM synthesis.

I'd been recently commissioned to write 100 tunes for children's story tapes. I spent most of the early time doing those and experimenting with this new technology. Then, one day, I picked up a copy of *Melody Maker* and in it was an advertisement for a games musician. That kind of thing was unheard-of in

those days – there must have been only five people or so in the country doing it. So I saw this advert and thought: 'Right, I've got 100 tunes; I'll find the best 10 and take those to them.'

I got the job and dumped everything else I was doing before, which was the usual musician stuff of trying to get into TV, radio, adverts, film...that world, as such.

How did things differ back then?

Back then, I'd use something like 3K to do a whole soundtrack, then the programmer Richard Leinfellner would come on the phone and say 'snip, snip, snip,' and I'd need to get it into 2K. And that was for audio and sound effects.

We were always very ambitious at Palace. Take *Barbarian*, for example. Steve Brown, the designer, wanted the soundtrack to sound like a *Conan* movie. What they wanted me to do was make the Commodore 64 sound like an orchestra, which, of course, was impossible. However, you could use various devices to get the feel across, to emulate the sound of an orchestra – and that's what I tried to do.

Back then I very much had to prove myself – I was competing against someone like Rob Hubbard, who was a genius. So every single thing I did I'd labour over. Take *Stifflip and Co*, for instance. With that, I tried to compose a very traditional English-sounding audio track. It wasn't the sort of thing you'd hear at the time.

You were among the early pioneers of interactive music, particularly with *The Chaos Engine*. How did that come about, and what did you learn about creating interactive pieces of music?

We realized when we were creating the game that it was perfectly suited to small sections of music. The game had many small areas that needed to be completed in sequence, and each of those reached a climax. So we thought it'd be nice to have different music for each section.

During the game's development, I'd spent a serious amount of time listening to early Prodigy. Analyzing it, I'd

Far right: *The Chaos Engine*, developed by The Bitmap Brothers.

Right: *James Pond 2*, developed and published by EA.

Below: *Evil Genius*, developed by Elixir Studios.



figured out it worked in simple one-bar phrases. That was all they needed for each section: simple one-bar motifs.

I felt it was possible to pull that down even further, into two beats – that is, a soundtrack that could change every two beats. It really suited *The Chaos Engine* to have music like that, so you could go from one section to another and within just two beats the whole sound had changed.

We also made it so that if you returned to an area you'd previously completed – that is, collected everything you needed to, and shot every enemy – the game would move to a slower version of the original music which said aurally: 'Hey, I've done this bit.' We pieced together a load of different ideas and it just gelled.

But what we learned was that this wasn't an interactive soundtrack in the way many people think of it. I firmly believe

that a truly interactive soundtrack – one that puts layer upon layer – is not practical. It doesn't work.

I've heard soundtracks where they attempt to do this. You'll be walking along and all of a sudden, a hi-hat will come in or something. You can't do this with music. If you take bits out or add bits in, it's not the same. Music is the sum total of what you're hearing. It's built specifically to be that piece of music.

Of course, tastes will change. I believe in a generation's time the way we listen to music will be radically different – and this kind of layered music will become more acceptable. You'll be able to have contrasting melodies – say, one 'good' piece of music over a 'bad' piece – played simultaneously. It'll be new music that we've never heard before. People will be able to work interactively with it because they'll have grown up with it. It'll sound horrible to us, of course, but those growing up with it will find it perfectly acceptable.

Does it pay to use off-the-shelf tools or your own proprietary technology?

I tend to favour our own technology. My experience of modern sound tools is not great. I got hold of DirectMusic, installed it, started working on it, and dumped it virtually straight away. I felt it was a program that had been put together by people that weren't writing music for games, or to entertain other people. It was like an academic exercise more than anything else. I know some people who have attempted to make careers using it, but I firmly believe they're barking up the wrong tree.

Q&A RICHARD JOSEPH

I will work with postproduction stuff like *Nuendo* and *SoundForge*. But mainly we work with our own proprietary technology. Our programmer Andy Mucho has written some amazing drivers that'll power the audio. We've just worked on two realtime strategy games. I call them four-dimensional games, because you've got your width, height, and depth – but you've also got time. It's not like a film, which is entirely linear – something could literally go on forever. And things requiring audio cues can happen anywhere at any time.

What we've had to do is build software that will cope with this. If you've got a thousand things going on in front of you, you've got to be able to make sure that only the most relevant sounds are heard at any given time. You don't want people to go and switch it off.

Andy's an amazing programmer. With *Evil Genius*, it was the first time I've seen a magazine comment specifically on the audio technology. And that's immensely rewarding.

What are the things to bear in mind when recording dialogue?

You've got to be careful of actors hamming it up just because they're doing a game. Consider that you might hear a phrase

a dozen times – the skill comes in making sure that the delivery of it is something that you hardly notice.

A game is developed over a long period of time. You have to keep going back and changing dialogue. So it's really important to make sure all your settings are the same every time you go back into the studio, otherwise you'll get a completely different tone. Another important thing to remember is to ensure that key voices are performed by people who know what they're doing.

Can a good soundtrack really make a bad game better? Can a bad soundtrack really detract from a good game?

Yes, to both questions – although it's made more difficult because music's always a matter of taste. For *Republic*, everyone loved the soundtrack – we got a BAFTA nomination for it, and it was fantastic. But then I went up on a forum and saw a post that said: 'I think the music sucks.' What I realized was that this person hates orchestral music. For him, the music didn't work and it ruined the game.

Below: *Sensible Soccer*, developed by Sensible Software.

Bottom: *Theme Park World*, developed by Bullfrog, with a BAFTA-winning soundtrack.

Below left: *Republic: The Revolution*, developed by Elixir Studios, with a BAFTA-nominated soundtrack.



Take *Doom 3*, for example. You play it and you pick up on all the sound in the background. All the rooms have a pitch or a sound to them. It's like 'Music Concrete' (experimental music using tapes and electronics from the 50s and 60s); it's music that isn't music. And it works. So well, in fact, that I think the developers actually ended up replacing the original music. Which appears to have been a great move. It's making a lot of others doing game audio think innovatively about the way they create their own soundtracks.

It takes a bold decision to do something as different as this. With *Gods*, we decided to have no music at all in the game, because we really wanted to create a 'You're there' atmosphere. You don't have music in real life. And in *The Chaos Engine*, we decided to have no bullet sounds – again, something not considered the norm.

But that's what I'm about; experimenting with weird ideas. One of my favorite soundtracks is *Mercenary* on the C64, which has no music and virtually no sound. But it's so atmospheric. It's perfect.



Above: *Speedball 2*, developed by The Bitmap Brothers.

Can any traditional musician write game soundtracks?

It's difficult. Movies and TV are linear experiences. Each moment is timed to perfection. You'll tug at the heart strings of someone and then maybe throw something else straight at them. But all the time, the music is timed perfectly to the flow of the film. But with games, the player is directing it.

Take something like a thirdperson game. You can have a bit of music that represents going to the next puzzle, which can be a linear piece. But then when the player reaches this puzzle, hours could be spent trying to solve it and immediately the linearity drops away.

People often get stuck over this idea of interactive music. But it's quite simple to produce a never-ending piece that reflects the state of the game and still remains musically 'correct'. We do this by creating a pool of musical phrases, which we splice together at random. If you have enough of these you can create the illusion of a never-ending piece. In simple terms, all you then need are three states – neutral, good, and bad. It's then just a case of swapping between these three pools depending on how well you're doing in the game.

Is the tendency to create your own sound effects, or buy them off the shelf?

We do both. I'll try wherever possible to go out and do location recordings. At the beginning of the year, for example, I went to France to record a load of frogs in the forest. If you can do it yourself, it's best to get it that way. You've a much better idea of the material you're after.

Sound-effect libraries exist on the Internet. You can get hundreds of thousands of effects and samples. The BBC Sound Library's up there for instance. So if I'm after a particularly specialized sound, I'll look there first. You have to be careful, though. There was a tale recently of someone paid to go to Hollywood and get effects from a top postproduction house. And the results were rubbish.

You've got to be creative, though. Look at *Star Wars Episode One*. For the Pod Racer sound effects, they actually used a lion's growl as part of the engine sounds. You wouldn't realize this, of course, but it adds an intensity to the effect. A lot of this creativity is rubbing off on games – people from film schools are bringing these techniques. The most important thing to do is to make sure that if a player hears something hundreds of times, they are never bored of it.

VIDEOGAME NARRATIVE

From an early age, we need to be told stories. Whether it's parents educating children with simple tales featuring colourful characters, or playground battles and heroic rescues – if we're not being told tales, we're inventing them for our own distraction.

In the earliest days, computer and videogame developers reserved narrative effort for the adventure or role-playing genres. Primitive text-based adventures lacked the sophistication of even the most basic of stories, creating a world restricted to simple location descriptions. Nonetheless, anecdotes abound of grown-up gamers recalling extraordinary vistas and panoramas illustrated by the computer – only for them to realize that they originated from simple text-based games. Their imaginations had coloured in complex backgrounds to the words.

DRAGONS AND PRINCESSES

This lack of narrative endeavour in the early computer and videogames can be easily explained – quite simply, there was barely enough processing power to move the characters, let alone illustrate their adventures. Sure, princesses were saved and dragons slain, but that, literally, was the start and end of the story.

It could be argued that all games tell stories. And to a point, it's true. *Space Invaders* boasts a basic plot, and one on which Devlin and Emmerich would happily base a £40 m movie.

A tenuous link between games and movies, sure – but there's a more concrete relationship between the two. The explosion of the medium into the public consciousness has resulted in a strange trading of blows between the movies and videogames. However, that's not stopped Hollywood professionals from aiming to work at least briefly in the games industry. Spielberg, the Wachowski Brothers,

and George Lucas have all tried their hands at making games. And the main roles they occupy are in direction and scriptwriting, two of the most important narrative skills.

While the arrival of movie talent has met with varied success, there can be no doubt that standards in game narrative are improving. And this is necessary for the medium to be taken seriously. Often, videogame stories are rubbished as clichéd and derivative. The industry itself seems to dismiss the importance of plot, choosing instead to emphasize in the box-blurb the number of levels and weapons.

THE QUEST

The adventure genre still leads the way in videogame narrative – though that has diversified over recent years to include more action-orientated titles. *Deus Ex*, *Tomb Raider*, *Broken Sword*, *Final Fantasy*, and *Metal Gear Solid* are all very different games, but the central quest theme is constant in all.

The rare exceptions prove narrative is an area worth exploring. Good stories appeal to both male and female players – and when many games struggle to recoup their initial investment, it's important to ensure their target market is as wide as it can be.

```

* ADAMS ADVENTURE * (VERSION 1.0/416)
(C) ADVENTURE BOX 3435 LONGWOOD FL 32750

THIS PROGRAM WILL ALLOW YOU TO HAVE AN
ADVENTURE WITHOUT EVER LEAVING YOUR
CHAIR! YOU WILL FIND YOURSELF IN A
STRANGE NEW WORLD. YOU'LL BE ABLE TO
EXAMINE, TAKE AND OTHERWISE MANIPULATE
THE OBJECTS YOU FIND THERE. YOU WILL
ALSO BE ABLE TO TRAVEL FROM LOCATION TO
LOCATION.

I'LL BE YOUR PUPPET IN THIS ADVENTURE.
YOU COMMAND ME WITH 2 WORD ENGLISH
SENTENCES. I DO HAVE OVER A 130 WORD
VOCABULARY SO IF A WORD DOESN'T WORK
TRY ANOTHER!

SOME COMMANDS I KNOW: HELP, SAVE GAME,
QUIT, SCORE, TAKE INVENTORY.

THE AUTHOR HAS WORKED OVER A YEAR ON
THIS PROGRAM SO PLEASE DON'T COPY OR
ACCEPT A PIRATED COPY! NOW HIT RETURN!

```



ADVENTURELAND

The first of the Scott Adams text adventures, *Adventureland* (1978) used simple two-word commands throughout the game (top). Scott Adams' series is one of the most fondly remembered of the early adventure genre, although his reputation began to wane as he tackled licensed properties towards the mid- to late 80s.

THE HOBBIT

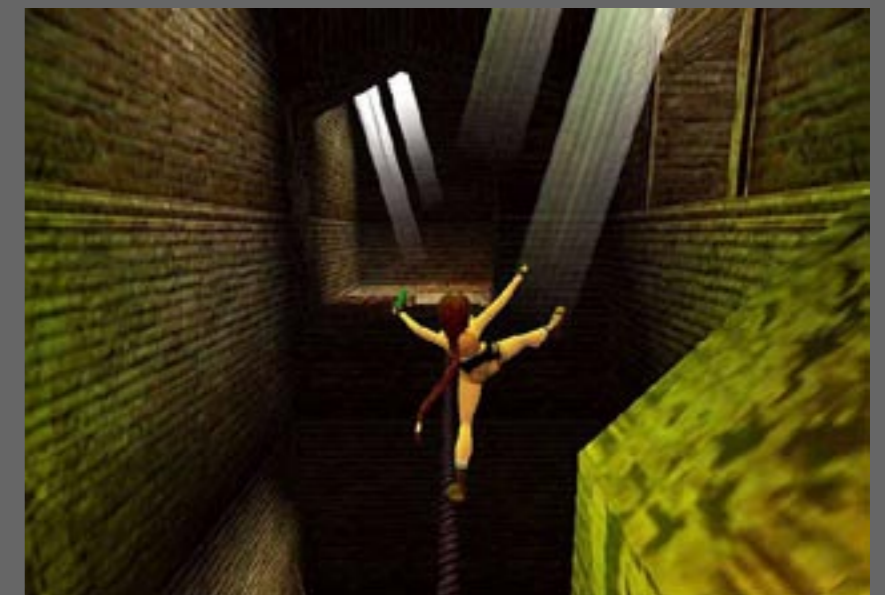
The Hobbit (1982), obviously licensed from J.R.R. Tolkien's *Lord of the Rings* prequel, was a fairly straightforward interpretation of the tale. The game was first developed for the Apple TRS-80, and re-issued on the Spectrum to take advantage of its superior graphics (above). The original Spectrum version was called 1.1 to make it appear enhanced. A version 1.2 was subsequently re-released with several bugs fixed.

CITY OF HEROES

It can work both ways. Some games prove so popular, they are turned into books – an endorsement of the narrative if ever there was one. The most recent example of games being licensed into a series of books is NCSOFT's *City of Heroes* (top right). The books are scheduled to appear in 2005. The game features 20 different ongoing story arcs as villain groups menace Paragon City and react to player victories and defeats.

TOMB RAIDER

Most games tell stories using linear sequences or cut-scenes to top and tail the sections under the control of the player. Although cinematic techniques have improved over time, it's remained this way since the industry experimented with interactive fiction. Bold attempts have been made to try to break free from this rigid format – but as games such as *Tomb Raider* (above right) have proved, it can work very well.



LUDOLOGISTS VS. NARRATOLOGISTS

There are those who claim that computer and videogames should be about the experience of playing rather than the pursuit of narrative. The opposition argue that games should reward players not just in the form of subsequent levels and challenges or an end-of-level ditty, but something more meaningful.

These two camps are divided, respectively, into Ludologists and Narratologists. And an example of their differences is illustrated by imagining their views on *Tetris* (1985) and *Myst* (1995). It is widely agreed that *Tetris* is one of the greatest games ever created. Narratologists would question its worth as there is no intentional storyline.

On the other hand, *Myst* – which typical gamers despise due to its utter linearity (not to mention obscure design) – would be claimed by Narratologists to be of significantly more merit, due to its imaginative dreamworld and intellectual challenge.

The truth, as ever, is somewhere in-between. Games should play well, above all else. But if they can tell stories, too, then so much the better.

NARRATIVE PROCESS Boy lives happily with his uncle and aunt. Then they are murdered. In pursuit of retribution, boy becomes powerful Jedi, hangs out with scum and villainy, nearly makes out with his sister and ends up with 50 per cent fewer hands in the sequel. The end.

Had George Lucas not spent a significant amount of time drafting the structure to the *Star Wars* universe, there's no doubt that the movies would not have enjoyed such enduring appeal. And the same goes for games. If you're serious about creating a narrative that will engage, then the structure needs to extend far beyond the actual start and end of the story.

There are three vital documents necessary in developing a strong in-game narrative. The first is the story, told from start to finish. This, typically, will be under 20 pages long, and will detail the events specific to the game. Locations, actions, 'inciting incidents' – all need to be scribbled down in chronological order. After revisions – and there's no limit on these – you should have a plot that is interesting, exciting, and features the correct pacing necessary to hold your audience's attention for the duration of the game.

THREE-ACT STRUCTURE

Games should adhere to a structure – for many, the three-act structure is suitable. Don't be discouraged or frustrated by the word 'structure' – it's something that works for a reason. This structure includes elements such as the 'inciting incident', which is an event or action that radically changes the balance of the protagonist. In a movie, the inciting incident needs to happen within the first quarter, but in a game many argue that it must come much sooner – ideally in the introduction.

If the inciting incident doesn't happen immediately, the game needn't be pedestrian up until this point. *Prince of Persia: The Sands of Time* – a game with a wonderful narrative, and perhaps the most perfect ending ever featured in a videogame – kicked off with a dramatic set-piece, but it wasn't until those sands of time were unleashed and the wider picture revealed that the game truly began.

The inciting incident will set up what many call the 'obligatory scene'. That's where the main character is going to resolve the tale – be that through conflict (that is, defeating those who sought to harm him) or restoration (putting things

Below: *Prince of Persia's* success was obviously not just down to its narrative. However, those who completed the game were treated to a textbook example of how to weave a decent story into an even more decent game. Perfectly paced, the game kicks off with a dramatic opening that doubles as a tutorial. Then, at the end of the first act – roughly an eighth of the way in – the central quest is revealed. *Prince of Persia* uses the classic

double-act mechanic, advancing the plot as dialogue is exchanged both in-game and through brief cut-scenes. It also cleverly integrates brief glimpses into the future, both to offer hints at obstacles yet to be overcome, and also to sow the seed of events yet to happen. The ending – far from being the disappointment of the majority of titles – is satisfying and surprising, and offers a strong reason to play through once again.



Left: Of course, the horror setting helped. But beneath *Resident Evil's* gory façade was a structure that would allow developer Capcom to extend the game into a hugely successful series encompassing many different genres beyond the straight action adventure. Generic and clichéd in places and with the occasional telegraphed revelation, the game nonetheless retains player interest far beyond the climactic ending – and its follow-ups have generally enhanced the original story.



Left: A genre in which it's difficult not to leave the player satisfied is the classic murder mystery. With 1991's *Cruise for a Corpse*, Delphine advanced the plot not so much by solving common object-based puzzles, but by talking to the right people about the right subjects at the right time, thus collecting clues. The game's clock advances in ten-minute intervals whenever an interesting clue is discovered, allowing the developers to integrate a crude timetable of where the assorted characters would be next. This is a clever narrative tool, abusing realtime principles in favour of allowing the player to progress at his own pace.



Left: A bold attempt at weaving a new interactive narrative into an existing world was Gathering of Developers' three-part *Blair Witch* series of games. All published in 2000 – a feat of logistics only possible by offloading the licence to three development teams – they attempted to expand on the myth in the original movie. Despite some continuity errors between the three titles, it was an interesting experiment – although the movie sequel released the same year and the overall quality of the trio diminished the reputation of the franchise. Each game was set in a different historical period, offering the writers a blank canvas upon which to interpret the film's folklore. Subsequent titles such as *Enter the Matrix* and *Knights of the Old Republic* followed this lead, with varying levels of success.

PASSIVE? OR INTERACTIVE?

There are two main methods of advancing the plot – through passive cut-scenes or as a natural part of the gameplay. Both have their benefits. Cut-scenes offer the director more creative freedom; by snatching control of the players, there is no need to worry about them doing things that might break the flow of these pivotal moments.

There has been a tendency over recent years to make these cut-scenes much more elaborate, with some of them running into tens of minutes. If this is your preference, go make a movie. While there is no doubt that gamers often admire lavish cut-scenes, they are ultimately frustrated by spending significant amounts of time not operating the joypad.

Exposition can be handled more smartly and seamlessly in-game, through dialogue or commentary. Text needs to be punchy and to the point, though there is room occasionally for the odd flowery moment if it cements a personality trait rather than specifically telling the story.

Ultimately there is neither a right way nor a wrong way – and, ideally, a game should utilize a mixture of passive and active exposition. However, if exposition is handled clumsily, you run the risk of boring the player with noninteractive sections. Be warned.



back to how they were). It's called 'obligatory' for a reason – the audience expects it, and will be left naturally disappointed if the ultimate conclusion doesn't meet that expectation.

The end of every 'act' should build to a peak, ensuring the pacing rises and falls. These peaks should build towards the end of every act and drop off at the beginning of the next section. It goes without saying that the end of Act 3 should boast the largest climax of them all.

Once you're happy with your plot, it's important to draw a detailed timeline of what every character featured in the game is doing and where that character is at any one time. Often this information won't be used directly – but the audience can sense whether you, as a storyteller, really know your world. If you don't, they quickly stop believing. It will also provide you with natural references to integrate into the script.

Finally, it's useful to have a background document that details every character's history, along with a summary of the overarching plot. This will provide a rich source of additional information that can be introduced into the script where appropriate, and provide the writers with the background that they need to get the characters' dialogue and voices right.

NARRATIVE CASE STUDIES Integrating narrative into interactive entertainment is a difficult skill. While there are many examples of competent storytelling within the art of videogames, the following four examples have been highlighted because they illustrate individual facets – both from a ‘how to’ as well as a ‘how not to’ perspective.

METAL GEAR SOLID 2 (2002)

Following the global success of *Metal Gear Solid* on the PlayStation in 1998, Hideo Kojima set out to better the original in every way. With the benefit of a new, more powerful platform in the PlayStation 2, many argue that he lost sight of what made the original so revered. Weighty cut-scenes – some demanding close to 30 minutes of viewing – and laboured text-based conversations using the in-game communication device, tested players’ patience, and the resulting game divided opinion among the gaming fraternity.

The question of gaming and identity was extrapolated throughout PlayStation 2. Cleverly, Kojima surprised fans by switching characters one-third of the way in. Instead of playing the cold, hard central character Solid Snake, MGS2 forced gamers to assume the role of almost his direct opposite – the blonde, sensitive Raiden. And while many felt cheated that they weren’t once again able to play videogaming’s equivalent of Rambo, it afforded a perspective of Snake that would not have been possible. You could appreciate the character more by not controlling him.

However, nothing could forgive the messy, rambling plot. Arbitrary twists and turns confused all who played it, resulting in an ending as unsatisfactory as it was convoluted. The attack of September 11, 2001, on the World Trade Center resulted in the development team dropping a climactic cut-scene showing the Metal Gear Ray’s destructive route through New York. The game simply cut from one location to another. Players were so confused by this point in the game, they simply shrugged it off. An essay in how not to do it – despite being a valiant effort.

Above right: *Metal Gear Solid 2*, developed by Konami.

Right: *The Broken Sword* series, developed by Revolution.

THE BROKEN SWORD SERIES (1992-2003)

Revolution founder Charles Cecil has been an exponent of videogame narrative for the duration of his time in the industry, and his tireless campaigning for gaming to adopt the grammar of movies has inspired a young generation of game designers. The *Broken Sword* series is his most famous creation, and is widely regarded as one of the greatest examples of videogame narrative.

This success is due to two main factors. In the trilogy to date, the narrative has drawn on real-life mysteries and legends – the Knights Templar in the first, the Mayan Prophecies in the second, and the Voynich Manuscript in the third. This has not only provided the series with a solid backdrop, but also piqued the interest of many who have played the games. Many *Broken Sword 1* players, for example,

subsequently went on to find out more about the Knights Templars and the Holy Grail.

More obviously, much of *Broken Sword’s* success can be attributed to the will-they-won’t-they? relationship of its central duo. George Stobbard and Nicole Collard’s continual fighting is typical of many male/female pairings, but it’s rare to see such interest handled so eloquently in a videogame.

Of course, the game’s style helped enormously. But rather than rely on long cut-scenes to advance the plot, Cecil keeps the action in-game and dialogue-driven. Even in the third game’s attempt to drive drama through video sequences, players needed to remain alert, as life-or-death decisions were usually not far away. While these ‘action sequences’ were not entirely successful, they do provide alternative thinking for those wishing to avoid lengthy cut-scenes.



PLANESCAPE TORMENT

Planescape Torment was based around an interesting concept. A traditional role-playing game set in the *Advanced Dungeons & Dragons* world – with its familiar rules and structure – it ran the danger of disappearing in a sea of similar titles. But what made the game stand out was the quality of the plot and the strength of its dialogue – two things rare in the world of videogames.

Assuming the role of a character who cannot die, *Planescape’s* world is structured but non-linear. Players opt to progress as they see fit, treating NPCs with contempt, admiration, or something in between. This could have resulted in a jumbled mess of conversational anomalies, but the sophisticated storytelling engine permitted the developers a huge degree of freedom – freedom that is handed over to players as they journey through *Torment’s*

world, forging friendships, alliances, and creating grievances as they see fit.

The player’s moral alignment and affiliation with different factions are impressively flexible and have a significant impact on the course of the game. Characters can switch character class allegiances on whims – unusual for a genre that typically restricts such decisions. The immortal premise actually works for the game, rather than against it. What’s more, the lead character’s immortality means that players rarely abuse the Save or Load system, maintaining the solidity of the game world.

Planescape marked a bold step in videogame narrative, and one which was not entirely commercially successful. A shame, as its brave attempt to introduce intelligent role-playing should have been rewarded with much more than a swift appearance in the bargain bin.

Left: *Deus Ex*, developed by Ion Storm.

Right: *Planescape Torment*, developed by Black Isle.



DEUS EX

Warren Spector has almost single-handedly taken it upon himself to pioneer the non-linear action adventure, with *Deus Ex* proving to be his most successful attempt to date. While the standard plot – shady government agency, viruses, terrorism, and so on – could have proved derivative initially, it was the storytelling skill and implied freedom that really captured the imagination. Most impressively, he did this in a firstperson game – typically the most linear of all genres.

Of course, no computer or videogame could ever be truly non-linear. But Spector and his team structured the game’s back-end to offer the illusion of freedom, advancing the same story through many different mechanics. Players with a penchant for gunplay could find NPCs reacting to them differently later on, aware of their hardball reputation. Brilliantly, the game’s expansive back-story is never fed down the throats of its players. A wealth of background information can be found in various documents and computer terminals – as well as from in-game characters. Kojima would have forced players to read every single character. Spector has enough confidence in both his players’ intelligence and his own storytelling ability to permit a greater degree of freedom.

Cut-scenes are rarely intrusive – a necessity, thanks to the game’s perspective. Switching from firstperson to thirdperson would detach players from the in-game world, shattering the feeling of existing within a realistic environment. Indeed, it’s the consistency of the story and the aplomb with which multiple solutions were designed and implemented for almost every section that enabled players to become so immersed in a rich, atmospheric tale.

Q&A

CHARLES CECIL, REVOLUTION SOFTWARE

Charles Cecil began writing games in 1981. Since its inception, his company, Revolution Software, has pioneered the use of traditional storytelling techniques in interactive entertainment, most successfully with the *Broken Sword* series of adventures, which built on the narrative success of such titles as *Lure of the Tempress* and *Beneath a Steel Sky*.

What do you think of the standard of storytelling in games?

The quality of storytelling in videogames is generally quite poor. That's not because the people who tell the stories in the games lack talent, but because the narrative in an interactive environment has an additional dimension over linear narrative. The opportunities and constraints of this extra dimension need to be understood both creatively and conceptually.

The wrong solution is simply to force a linear story into a game through cut-scenes rather than through integrating the story into the gameplay. That's where many games go wrong – not least because the end-result usually involves lengthy exposition scenes, which, frankly, quickly bore the player.

The development of a grammar for interactive storytelling is vital in order to move the art-form forward. There are two schools of thought. I follow the line of: look at the way films work, understand their grammar, and then work out how to apply that to games. The other school says: forget about other media, interactive entertainment has nothing to do with linear movies. We're our own medium, so let's invent our own rules as we discover them.

The reason that I think they are wrong is simple. If you look at great artists in any field, they only tend to break the rules after they have mastered them. Picasso, for example, was a brilliant classical painter, but only after he mastered his craft did he go on to experiment. Without his mastery of the rules of his art, his improvisation would have been meaningless.

What are the essential ingredients of a good narrative?

You have to tell a great story and present it within a structure that is logical and makes sense. Jean-Luc Godard summed it up well when he said that a story has to have a beginning, a middle, and an end – but not necessarily in that order. And that's the heart of it. You have to present a story with a structure – the design of the structure is up to you.



What, ideally, should come first: the plot, or the gameplay mechanic?

It really depends on the game. There is no doubt that plot must support gameplay – gameplay is king. But if the plot and the gameplay are genuinely intertwined then they will need to be designed in parallel. Ideally elements of the plot would develop systemically – so the author designs the rules for the plot which the player then creates! To an extent this happened in *Grand Theft Auto III* – with an overarching linear plot running in parallel with player-created systemic sub-plots. This is one of the reasons why the game was so well received. In doing this, we got a glimpse of the future of storytelling in the interactive medium.

But isn't GTA's narrative actually fairly basic? It's a series of missions interspersed with cut-scenes...

Clearly, in *GTA*, you had the overall linear narrative, but within it you had player-created systemic substories. That's what makes the game so appealing; the freedom of the systemic, within the structure of the linear. That's why it worked.



Top: *Broken Sword II: The Smoking Mirror*, by Revolution Software.

Above: *Gold and Glory: The Road to El Dorado*, by Revolution Software.

I would argue that its staccato storytelling is part of the game's strength. The player can play the game at their own pace, and the pace rises and falls with a great rhythm throughout each mission. By way of example, I remember a particularly good mission in which you have to hijack a van. Finding it and tailing it is fairly low-key, but then you ram it and it accelerates to escape, then you hear sirens, then the police cars arrive and while you continue ramming the van, they are ramming you. All hell has broken loose. And then you steal the load, outrun the cops, and normality returns.

So the pace really varies, like it does in any good narrative. But it varies under the player's direction, and that's part of the brilliance.

Is there anything specifically that we can learn from the structure of movies?

Absolutely. There are several great books on this subject – I would recommend *Story* by Robert McKee, or David Freeman's book *Creating Emotion in Games*, which specifically takes film techniques and applies them to games. I have attended several excellent lectures by Robert McKee and he concentrates on structure and how to create an emotional and engaging story. One element that is of relevance is what he calls the 'expectation gap' – the difference between what the protagonist, and therefore the audience, expects and what then happens. The expectation gap creates interest and emotion – and therefore drama. If a game does not create expectation gaps then it feels flat. A criticism of *GTA3* is that there were few expectation gaps – what you expected to happen generally did happen. Had they addressed this, the game would have been even better.

However we are still pioneering this exciting new art-form. Games are in their infancy, which puts us in the great position of being able to make mistakes, provided they are not too serious and we continue to innovate.

You've been developing narrative-led video games since the early 1980s. What are the most important things you've learned about conveying the background story?

In an adventure there is a lot of story to tell. A golden rule is to hide the exposition. We do this by integrating the story and the gameplay – which means that our linear cut-scenes need only be relatively short. We also realized very early on that if you have two protagonists, then those characters can exchange dialogue that can be witty, fun, and also convey exposition. So when George and Nico – the main characters of the *Broken Sword* series – exchange quips, we're actually furthering the story in a subtle way.

You can hide exposition in several ways. One effective way is to integrate the narrative with the gameplay itself – so



Above: *Broken Sword: The Sleeping Dragon*.

Opposite page, top to bottom: *Lure of the Temptress*, *In Cold Blood*, *Who Wants To Be a Millionaire?*, all by Revolution Software.

as the player advances through puzzles, the story is gradually revealed.

The other way is, instead of having a few great chunks of exposition, you actually break it up into lots of little sequences. So you're continuing to entertain the player, hopefully making them laugh, hopefully making them gasp, but at each point the story is being drip-fed, so we don't need the extraordinary lengthy cut-scenes that have become prevalent in many Japanese games. I used to assume that Japanese gamers liked lengthy cut-scenes – but recently a Japanese journalist complimented me on writing games that hid their exposition!

Have you been able to exploit the technology at your disposal better as processing power has improved?

Absolutely in terms of graphics, physics, AI, etc. In terms of narrative, I think that our creativity has limited our progress rather than a lack of processing power. Technology is way ahead of what we've needed, and we're still catching up with it.

What we could do in our first game *Lure of the Temptress* is not that much different from what we can do now. We designed the game so that the player could go and talk to a huge number of characters about what was relevant at that point, but as the game advanced, their dialogue was replaced to reflect new situations. And that was quite interesting because it meant you could go back and talk to people, and they would say completely new things based upon the fact that the world had moved on.

The other thing that people really liked was that we had characters who'd walk around doing their own thing – a system that we called Virtual Theatre. I've always regretted the fact that we didn't really exploit Virtual Theatre in later games. But the way that characters had freedom to do what they wanted to conflicted with the adventure idea of a multi-linear narrative in a nonlinear environment. I've always been aware that Virtual Theatre was a glimpse of what was possible. Our future adventure games will definitely revisit this idea of characters having their own independent lives.

The Broken Sword series has always been applauded for its narrative. What do you think is it about the game that appeals?

One key element is that we feature authentic ancient artifacts and draw on historical references that allow us to blur the boundary between fact and fiction. For our first *Broken Sword* game, *Broken Sword – Shadow of the Templars* (1996), the Knights Templar provided a wonderfully rich subject – there was a historical conspiracy, chivalry of the knights (tempered with barbarity), a fabulous treasure that was lost and never found, and the Order really could exist in some form of secret society today. All of these things combined and culminated in a very exciting way.

It is interesting to note that Hollywood is making a plethora of films that feature the subject matters that have previously been covered by the *Broken Sword* games. In November 2004 the movie *National Treasure* was released, in which the protagonist (Nicolas Cage) finds himself in a race to find a long-lost treasure taken from Europe to America by Templars. In 2005 the Ron Howard directed movie *The Da Vinci Code* (based on the novel by Dan Brown) will feature a French female and American male protagonists in a race to find a Templar secret (sounds familiar?) that threatens to rock the Catholic Church. The subjects covered by *Broken Sword* games have a habit of foreshadowing the zeitgeist – which is why, I believe, they feel fresh and cutting-edge.

The influx of talent from external industries has met with mixed results. What are the important things to remember when dealing with skills from traditional scriptwriting and filmmaking?

It is absolutely vital that the vision-holder – or the auteur (if you want to be pretentious) – needs to keep control of and drive the vision. And that vision-holder must understand and maximize the gameplay experience. Provided the script-writer, the artists, the directors, the cinematographers, and the orchestrators work under the direction of the person driving the vision, then they can be enormously valuable. If they take over and say 'Well, actually, we can do better than you,' then you have a disaster on your hands. So the key is having one person who drives the vision, but controls and uses the people around as part of a valuable creative team.

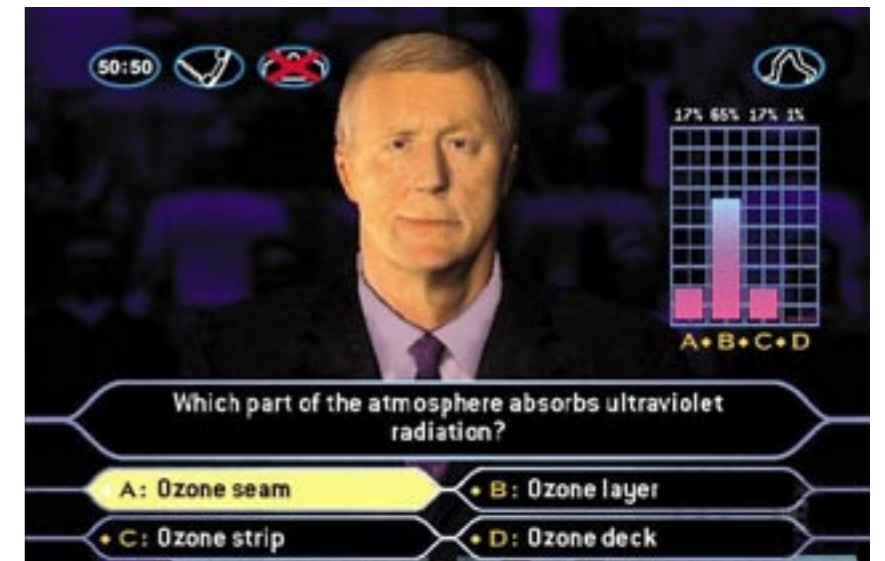
Back in the early 1990s bosses at publishing companies decided that people didn't want to play games, but instead wanted to watch 'interactive movies'. And to make them, control of the vision was passed from game designers, who understand the interactive experience, to filmmakers who don't. The result was an unmitigated disaster, which is a shame because the idea of creating a game that is as emotionally compelling as a great movie is a worthy goal.

How do you base a game around an existing narrative – for example, a film or book licence?

The easiest way of adapting a licence, of course, is to write a side-scrolling jumping game or a firstperson shooter, but that's missing the point!

How to write a narrative-driven game based on an existing story is far from obvious. A few years ago we were approached to write an adventure game based around Dreamwork's animated movie *Gold and Glory – Road to El Dorado*. I was pleased with the solution that we found.

The movie featured two happy-go-lucky chancers who set off in search of gold and adventure. Building on the characters, one of the fun elements of the movie, I decided to set the game after the movie ended and have the two protagonists stranded on a raft, telling their versions of the story to a girl they wanted to impress. In re-telling their highly exaggerated versions of the story, we were able to imply that part of their versions were true and part were fiction – hence, able to work in original events that stuck to the vision of the film. It was fun because we could tell the same story, but in a different way.



TESTING

Testing is perhaps the most unfairly maligned part of the development process. Although Quality Assurance (QA) departments are intimately involved throughout development, the position of tester is frequently considered to be just an entry-level position, or a stepping stone to a position of greater responsibility. Consequently, testers, and the testing process, are all too frequently looked down upon by other elements of the development team, and blamed for their apparent lack of rigour by customers who encounter a bug or glitch in a released title.

VALUABLE FEEDBACK

But the dogged foot soldiers that are found in QA departments are crucially important to the development effort: they provide valuable feedback from a position that isn't so close to the process that they're blinded by it; they ensure that a game's functionality isn't compromised by bugs and glitches; and by helping games pass the approval processes laid down by manufacturers, and meet the strictures of content rating systems, they can ultimately save publishers and



Halo: Combat Evolved, developed by Bungie and published by Microsoft.

developers money. In order to fulfil this valuable role, testers work fiendishly hard, playing games to excess with the unnatural (and frequently not very enjoyable) intention of breaking them. And yet in worst-case scenarios their feedback is ignored but they're still blamed for any bugs in the final game.

QA resources are typically provided by both the developer and the publisher, with the average team consisting of around ten testers at a developer, supporting a publishing QA department of around 200 people, whose efforts are divided across various projects at any one time. While early testing is typically directed towards more design-related issues, such as nuances of control and interface, by the end of production, this will generally have given way to more technically weighted feedback, with the majority of effort devoted to identifying and prioritizing bugs and glitches, and recording them using bug-tracking tools. The final element of the QA process also comes towards the end of production, when testers make sure that the game meets the guidelines laid down by the manufacturers for their hardware approval.



Above: *Jak and Daxter*, developed by Naughty Dog.
Left: *Ratchet & Clank*, developed by Insomniac Games.

QUALITATIVE TESTING

The initial stages of QA involvement can occur very early during development, providing scope for testing departments to provide useful feedback regarding design. While the rest of the team is producing early milestone builds, or, even earlier, technical prototypes and/or vertical-slice demos, testers will diligently and exhaustively play through these sections of the game. They will then provide comments about such issues as interface and control by liaising with the producer, who communicates this feedback to the rest of the team where necessary. It's rare, though, for QA departments to continue to have significant input into major design issues after a game is feature-locked (*see over*).

Nevertheless, forms of qualitative testing can continue: although it's not widely considered to be an essential component of production, focus group testing can provide valuable information that's beyond the scope of the typical QA team. And although creative firebrands might balk at the prospect of subjecting their grand designs to the critical faculties of the unwashed masses, posing the right questions of focus groups can give designers a lot to think about and learn from.

During a talk at GDC Europe in 2002, Bill Fulton, the founder of the Microsoft Games User-Testing Group, outlined various improvements that were made to the company's titles in the wake of focused consumer research. Most notably, the group discovered, after experimentation, that the preference of players for inverted or uninverted pitch controls in firstperson shooting games is a deeply ingrained one that's not easy for players to re-learn, and very few people were equally at home with either control method. Consequently the option to invert the pitch control was integrated into the opening cut-scenes in *Halo: Combat Evolved*, a game that benefits enormously from an almost sublime attention to detail.

Other games to benefit from focus testing include *Ratchet & Clank*, and *Jak and Daxter*, both developed under the auspices of the Cerny Method (*see page 51*). But there are drawbacks to focus testing if the wrong questions are asked, and the process has to be closely monitored to make sure that participants don't try to second-guess the basis of any questions, giving the answers that they think they are expected to give.

TECHNICAL TESTING The real substance of a QA team's efforts don't start until after the development team has locked down the set of features that will be going into the final game. It's at this point that the testing process shifts to the real meat and drink of any QA team: bug hunting.



CLASSIFYING BUGS

QA departments generally divide bugs across four categories, indicating their severity and the priority with which they need to be fixed:

A-CLASS BUGS

These include bugs that cause the game to crash or lock up, or that would cause the game to fail to receive console manufacturer approval. Also, bugs that cause installation to fail, or features that don't function.

B-CLASS BUGS

Although these are also undesirable, they're more likely to feature in a released game if deadlines are pressing. They include major graphical glitches, such as scenery pop-up and frame-rate issues as well as front-end mistakes and localization errors.

C-CLASS BUGS

These frequently fall by the wayside unless they can be quickly and easily solved, and consist of minor slipups such as spelling errors and minor graphical or audio glitches.

D-CLASS BUGS (ALSO KNOWN AS WISH CLASS BUGS)

Typically these are actually suggestions rather than bugs, including items such as an unintuitive front-end, or an unappealing colour scheme and so on.

The fundamental task of the tester is to root out bugs that range from minor art glitches to wholesale crash bugs, making sure that these are effectively communicated to the programming team who then need to remedy them. It goes without saying that none of these ought to appear in the finished product; but with the funding for development frequently incumbent upon developers meeting milestone requirements, testers need to be engaged in the pursuit of glitch-free perfection as early as possible.

Thus testers are required to meticulously play through unfinished early builds of the game code, frequently for a stamina-sapping length of time. Some testers will need to draw up a table or checklist

to assist them to methodically test every permutation of gameplay variables. This could, for example, be as simple as testing every possible combination of character model and weapon in an arena combat game, or the more complicated task of going through every possible conversation with every different character in a complicated RPG. Others will need to test the game's compatibility with every different variation of hardware (which, of course, requires much less work for console versions of games), while others still will need to simply play through the game.

The need to track the fate of any bugs that are discovered requires specialized tools for logging and prioritizing them. Testers will typically use video recordings of game sequences, or screenshots to demonstrate the nature of the bug or glitch, so any tool needs to be able to attach such associated media files with the bug entry. Once the tools are in place, the testers are then free to do their work: any glitches are entered into the logging tool; the game's producer will then assign that bug a priority depending on how pressing the need is for it to be fixed, and allocate programming resources; finally, the fixed code will be reassigned to the testers, for the cycle to start again.

HARDWARE APPROVAL

Perhaps the most important facet of QA is ensuring that the finished game meets the approval procedures of the console manufacturers. Although this step isn't necessary at all for PC games, most games these days are developed for one or more console (currently PlayStation 2, Xbox, and GameCube). Each hardware manufacturer has its own set of hoops that developers and publishers need to jump through before they will allow a game to grace their hardware, such as Sony's Technical Requirements Checklist (or TRC). Indeed some manufacturers even use automated tools to test the console approval compliance status of games.

These guidelines and restrictions ensure that each console is home to no objectionable content, and that the consumer's experience with such things



Opposite, above and below: Atari scored a couple of own goals with *Driv3r* and *Enter the Matrix*. Both titles were bugged and glitchy, perhaps because they were rushed out to meet deadlines imposed by the commitments of financial reporting. *Driv3r* featured flying cars performing doughnuts in the sky, and missions that would complete at random, and although *Enter the Matrix* couldn't quite match showstoppers such as these, it was nevertheless very rough around the edges.

Obtaining hardware approval isn't just about technical glitches and bugs. Both *Broken Sword III: The Sleeping Dragon* (above) and *Metal Slug 3* (right) were refused approval by Sony for the U.S. market because they were considered too old-fashioned for the brand.



as memory card interfaces are consistent and trouble-free. While most manufacturers provide training seminars for QA teams to attend in order to make sure that developers and publishers are au fait with the precise nature of these requirements, there's no substitute for experience. Indeed since the console approval process can last up to two weeks, and typically takes more than one submission, failing to meet approval guidelines can prove to be very costly to publishers who have to revise marketing and distribution schedules. Allied to the fact that it's not uncommon for a game to fail on a single bug, the process demands the utmost attention to detail.

NOTORIOUS BUGS It should come as no surprise to find that the occasional bug can rear its ugly head in games released to retail. Equally, it's hardly shocking to find that glitches are more common in the sorts of big, expansive, emergent videogames that are more difficult to test. But this does present the irony that many of the most bug-ridden games are also the best available.

SOME GAMES THAT CONTAINED BUGS

KNIGHTS OF THE OLD REPUBLIC

LucasArts (2002)

Although it's possible to play through BioWare's magnificent *Star Wars* RPG without encountering a single glitch, the range of bugs discovered within *Knights of the Old Republic* (below) is enough to make even a Wookiee flinch. Indeed, in one of the game's more memorable malfunctions, the lead character's entire party of allies would slowly be transformed into Wookiees. Other anomalies included the appearance of a 'Galaxy Droid' who could escort the party to any planet in the game, and floating and invisible characters. Again, however, the number of bugs is directly related to the sheer range and variety of interactions that the game world affords the player.

GRAND THEFT AUTO III

Rockstar Games (2001)

A game that allowed players to explore an entire city across three islands, which gave players the freedom to interact with this world in an apparently endless variety of ways, was inevitably going to feature the occasional glitch. Nevertheless, the extent and variety of the glitches found in *GTA III* were exceeded in scope only by the ambition of the game itself. From randomly missing traffic, to typical 'falling through scenery' bugs, the game had them all. There were also some neat glitches that actually added to its appeal, such as being able to get to the third island early (by exploding a car near a roadblock), and being able to play with your character dressed in a prison uniform.



SUPER MARIO SUNSHINE

Nintendo (2002)

Given Nintendo's reputation for staunch quality control, and the solidity and bug-free nature of its predecessor, it was disappointing to learn that *Super Mario Sunshine* (left) was afflicted with bugs and glitches aplenty. Now, by any ordinary yardstick, many of these were truly inconsequential – Mario appearing to walk on air, or sink into the ground in certain instances, for example. But Nintendo had historically set its standards so high that they became painfully apparent. This was compounded by more serious errors (such as falling through the bottom of the galleon in Pinna Park, which was probably the most oft-cited niggle of Nintendo's disappointed fan following).



AND SOME THAT DIDN'T

ETERNAL DARKNESS: SANITY'S REQUIEM

Nintendo (2002)

The crash bug is truly the bane of the developer's (and tester's) existence. Pity the gamer then, who, for no apparent reason, is forced to witness their game resetting itself. Unless they're playing the magnificent *Eternal Darkness*, which uses simulated technical flaws and glitches to evoke a deeply unsettling atmosphere as your character's sanity gradually dissipates. Thus, during key sequences depicting the gradual onset of psychosis, the player's TV appears to change video modes; items appear to go missing from the inventory; the game

will apparently reset; and, most unsettling of all, the game appears to delete your Save Game after ending a game session.

METAL GEAR SOLID 2: SONS OF LIBERTY

Konami (2001)

In a similar vein, towards the end of Hideo Kojima's masterpiece, the depiction of the gradual disintegration of an AI entity is conveyed by simulating various types of glitchy behaviour – most notably, by presenting the player with a 'Fission Mailed' screen (so it's actually obvious to the astute gamer that it's not an authentic Mission Failed moment – although it's still pretty disconcerting).



Q&A

ASAD HABIB, QA MANAGER, KUJU ENTERTAINMENT

Over a career spanning seven years, Asad Habib has tested games for the likes of Gremlin Interactive, Atari, and currently Kuju Entertainment. Over this time he has tested games on a variety of hardware platforms, ranging from the Sega Saturn to Sony's PlayStation 2, and even mobile phone platforms.

What sort of skills do you need to make a good tester?

Being able to cope with repetitive tasks is a key part of being a tester. Some people might like the idea of playing games, but imagine being saddled with a game from a genre that you didn't like; you'd still have to test the game. Day in, day out, and not for days or weeks, but for months and months!

How important is testing and QA to the overall development process?

Very important! Just look at the lambasting the latest *Tomb Raider* title received in the specialist and mainstream press due to its bugs. A bug-ridden product can ruin the enjoyment of a game and undo months and months of hard work in development.

How is testing integrated into the rest of the development process? What points of contact and interaction are there, for example, between QA/test staff and the rest of the development team?

QA staff are always expected to provide good communication and be open with the development team, simply because they are the front line for the artists and programmers, so it is vital that any problems are communicated in the proper channels as early as possible. Test integration as early as possible is key, as this makes the whole test team feel a part of the development of the game and not a separate entity, as can sometimes be the case. For QA departments at large publishers it is sometimes even more isolated, working with developers that are located across the world, which presents a bigger challenge. The main points of contact would typically be a lead tester liaising directly with the producer at first.



How well is the testing process usually integrated into the overall development process, in your experience?

It varies wildly. Sometimes the testing process is, literally, a piece of paper with a list of bugs written on it. At the other end of the scale there are projects with 20 people working on 10,000 bugs using a third-party bug tracker. Obviously the more complex and bigger the game the greater the testing resources required.

Usually there are two sides to the Quality Assurance programme. Development QA is literally the front line of the development team. The programme starts very early during the project, and can range from providing input for a first pitch (if a demo is made, for instance) to the more common requirement of checking milestones before they are delivered to the publisher.



Left and below: *FireWarrior*.

Far left: *Advance Wars: Under Fire*; both developed by Kuju Entertainment.





Development QA consists of the early checking of prototypes. This involves such things as checking the frame rate, checking the controls, the art and overall look and feel, and sound as well as individual engine components (for example, the physics in a racing game or the character interaction in an RPG).

Publishing QA, on the other hand, tends to focus on the last three months of the development cycle and in particular the requirement to pass the game through the approval processes of third-party manufacturers such as Sony and Microsoft. This requires comprehensive test plans, and the coordination of each of the third-party manufacturers' standards and guidelines checks (TRC checks for Sony; TCR for Microsoft; and Lot checks for Nintendo). Typically at this stage the onus is on mass bug-finding: collision checks, falling through the world/map checks, crash bugs, licensing issues (very important), graphical glitches, control system bugs, front-end anomalies. Literally the whole gamut of bug testing.



What do testers actually do? What makes up their average day?

Find bugs! Typically a tester will come in, in the morning and load up his or her e-mail and bug-tracking software. Assuming a version is ready for him in the morning he will check against the list of marked fixes on the buglist, which the development team should have done. The first priority is then systematically going through the list to see which bugs are still in and which bugs have passed. Once this is done, any new bugs found along the way are also inputted.

Does the development industry place enough importance on the testing process and the people responsible for it?

In a short word, no, which is a shame. Tight budgets and timescales always cut into QA time on a product. But when QA departments input bugs, they also record suggestions for improvements that could be made. It would be an interesting exercise to compare this list to the problems that people complain about upon release. You might be surprised at the similarity. Companies that do invest time and people on QA usually reap the rewards in the shape of better review scores and so on.



How important is a quality-testing department to making sure that console approval and rating processes run smoothly?

This is the life-blood of publishing QA. Any department that regularly fails at console approval is soon an ex-QA department. Typically a game can expect to pass the second time around. Successful liaison with the console approval QA departments is key. Manufacturers now offer training seminars to communicate standards, and some have automated tools for testing their console approval compliance status.

Do you have any memorable or informative experiences, or examples of interesting glitches that got through?

One that stands out was for a soccer game that was due for release and submission. The producer came down and said he wanted me to test it before they sent it in the afternoon. Most of the testing had been done. When I got the game I wondered what would happen if I fouled a player off the pitch? Would the game be stupid enough to break? Surely not. So I positioned a man outside the boundary, selected another player and made him dive in. Immediately the ref gave a free kick. Outside the soccer field!

Or there's the time I was woken up at 2 a.m. to ferry a mastered game to Sony, only to receive another phone call at 4 a.m. saying a bug had been found so could I please return to pick up the new version. That was a long day.



How important is outsourcing to the localization process?

If a sufficiently big internal localization team is not built up then outsourcing localization is key for titles. There is a definite need for them simply because not all organizations are big enough for in-house localization. Good communication with the outsourcing team or teams at all stages of the localization process saves time later on. For example a simple task, such as cheats not being sent through early enough, might mean the localization testers spend a full day getting to the end of the game, when instead it could have been done in half an hour. Therefore checklists with all the important information needed by localization teams are not uncommon in QA departments.

What are the most common mistakes that developers make when working with QA departments?

Not giving enough time early on due to milestone commitments. All too often QA gets put on a back burner.

Among the games that Asad Habib has tested are: From left to right: *Driver 2*, developed by Reflections Interactive; *FireWarrior*, *Advance Wars: Under Fire*, and *Reign of Fire*, all developed by Kuju Entertainment.