

MG Interview: Dave Scheifler

Interview by Christopher Breen

In the course of reviewing Mission: THUNDERBOLT for MacUser magazine, I had occasion to contact its creator, Dave Scheifler. (Okay, I was stuck on Region 10 with a Terrorist Collaborator that I couldn't seem to get around, I had a deadline, and I needed assistance.) Dave proved to be a helpful and thoughtful correspondent. I got past the Terrorist, learned patience from the Icky Lumps, and finally accomplished my mission (and met my deadline!).

Revealing a generous nature, Dave — laughing in the face of online charges — consented to this bi-coastal quiz for Inside Mac Games.

IMG: What is the history of Mission: Thunderbolt?

Scheifler: Thunderbolt began in September of 1986, while I was working for Digital Equipment Corporation. I'd had a couple of false starts on other types of games in previous years, but this instance proved to be for real. Part of the motivation, to be quite honest, was sheer frustration at the human interface and playability of the other home computer and CRT (display terminal) based games. They were painfully nonintuitive, irritating in how they presented information, and the mechanics of how a person interacted with those games was simply dreadful. Well, to be honest, I figured I could do a better job of it, what with having been writing software for some 15 years by that time. Apologies if that sounds arrogant — it probably was at the time. Anyway, my central focus was to explicitly design ease-of-play into my game so that a player could concentrate on *playing* the game and having fun, rather than fighting the game itself.

I chose a science fiction theme because most everything else seemed to be fantasy based, and more of the same would be less interesting to write, and to play, than some other genre. Within a couple of months my friends were beating on me with requests to try it out — even though almost nothing existed at the time. Nevertheless, I spun off a baselevel, crossed my fingers, and was pleased when the responses were favorable. Thereafter, besides adding features, I continued to give special attention to modifying aspects of the game's interface that were annoying to people or detracted from the playing experience. That outlook on game design has stayed with me through the years, and is applied just as scrupulously to each new venture.

I named the game "DoomsDay 2000". DOOM for short. After many more months of coding, and extensive playtesting to achieve good play balance, Version 1.0 was officially released in December 1987. It was conceived as a series of five missions, the first being "Operation Thunderbolt." And in subsequent years I produced the next two missions: "Operation Iron Maiden" and "Operation Tsunami." They were all released for Digital Equipment Corporation's VAX/VMS systems and used a Digital VTxxx compatible terminal for the display; and if the VT supported soft fonts, then the game downline loaded a special set of monochrome fonts that gave pictorial representations of creatures, loot, and terrain, rather than symbols from the extended ANSI character set. Apple actually formally acquired a copy from me to help shake out a terminal emulator for communication with VAXs, since DOOM pretty much took advantage of every VT capability. DOOM still exists today, and has spread all over the world via Digital's engineering network. So if you have access to a VAX/VMS system, and have a friend in Digital who can grab the software for you, then you can get an early start on Thunderbolt's follow on missions today — albeit in a more primitive form.

It was the popularity of DOOM, and direct encouragement from friends and associates, that resulted in my doing a commercial version of the game. As a side note, it was sufficiently popular that there were several thefts of my sources via network penetration. In one instance the routing came through one of our offices in Seattle; a Digital security guy called me to let me know what had happened. It's not known whether the culprit(s) got the sources off the E-net, but it caused me to begin encrypting sources from then onward. But encryption only slowed 'em down a bit. A year or two later there was another break-in. It was done by the same fellow who stole the VMS operating system sources from Digital's STAR cluster in New Hampshire. What I found interesting at the time is that Digital's security was actually watching and recording his activities remotely as he tinkered around in our VINO cluster in Massachusetts where I did my game development — which evidently contributed to the fellow's eventual arrest. You can read about him in a book called "CYBERPUNK, Outlaws and Hackers on the Computer Frontier," by Katie Hafner and John Markoff. See "Part One: Kevin the Dark-Side Hacker."

Yet another side note: I used a "RoboCop" in my game as early as October 1986, well before the movie of the same name came out, but being ignorant of such things, I never registered it as a trademark! When the Studio (or whoever) TM'ed the name, I changed mine. Live and learn....

It took about a year to port DOOM's first mission over to the Macintosh, and I sold it via mail-order for half a year as "DoomsDay 2000." It was then published by Casady & Greene as "Mission: THUNDERBOLT" in January 1992. The core of the game remained the same across the port, but the look became much more colorful and interesting (the detail creature art was done by John Calhoun of "Glider" fame), and the inclusion of sound effects had a dramatic impact as well. Perhaps the word "port" is misleading in one key area: the entire human interface was scrapped and rewritten from scratch. What I find gratifying is that unless I mention its history, people tend to think it was written for the Mac initially, which, given Thunderbolt's origin, is great because it means I was successful in creating a "Macintosh" product.

IMG: Was Mission: Thunderbolt (or its predecessor, DoomsDay 2000) your first game?

Scheifler: As a matter of fact, the very first program I ever wrote was a space war game back in the early Seventies. The screen had a sun in the middle, and a pair of space stations for refueling and re-equipment at opposite corners. Two players fought each other in real time, while dealing with the strong gravitational pull of the sun. Docking maneuvers were tricky, and could result in the destruction of both you and your station.

What makes it special in my mind's eye is that this was in "the olden days" on a Digital Equipment Corporation LINC-8 computer with 4k of memory, an embedded CRT, multiple DECtape spindles, and an ASR-33 Teletype with papertape reader/punch. I'd actually come into work every evening and *toggle* my code in via the LINC-8's switches ... because I hadn't realized that the machine had things called editors and assemblers! I'd memorized all the octal mnemonics for the opcodes as well as the field layouts. Wonderful naiveté. I have lots of good memories of my awakening into the delightful and intriguing world of computers.

Although I gave no thought to it at the time, I wouldn't be surprised if unconsciously I was actually inspired by Steve Russell, the guy who did the very first computer space war game. So computer gaming is where it all really started for me. I subsequently met Steve many years later while working for Digital, became friends, and we played together in a weekly role playing game for a time.

IMG: How did you begin as a programmer, and what motivated you to be a game designer? In particular, why did you create this particular kind of game?

Scheifler: My boss saw my SpaceWar game, realized I had potential, showed me how to use the LINC-8's programming tools, and pointed me at a manual. In six months I was creating utilities to help them with their efforts. This was in St. Louis at Washington University's BioMedical Computer Laboratory (BCL). Across a couple of years I basically advanced from being merely a computer monitor boy in their cardiac care unit, to doing systems programming and porting operating systems to new platforms.

After several happy years at BCL, I went off and got a couple of degrees in computer science and joined Digital out in California. I did customer support for their DECsystem-10 and DECSYSTEM-20 customers for two years and then moved to Massachusetts to their Large Computer Group doing software development for the -20s. Eventually I shifted over to VAX/VMS when that became the (then) central focus of the corporation. Several years of programming later, I shifted again and climbed up the management ladder because I'd found that I also enjoyed directing and guiding projects. I rose from project leader to supervisor to manager and then, eventually, senior manager — most recently responsible for Digital's Fault Tolerant Software Engineering group in Marlboro.

But my roots were that of a bit-banger. I had originally been self-taught, then built a strong foundation through formal education, and later at Digital learned the discipline and processes that differentiate a "software engineer" from a "programmer." Anyway, back in 1986 I decided to spend my off hours getting back in touch with programming by learning C; I was a manager at the time. To really get to know the intricacies of a new language you have to use it. So...this was the other half of the motivation for why I began writing DomsDay 2000.

Given the limitations of display terminals back then, DOOM naturally assumed typical board game qualities in that the playing area was a two dimensional grid, with game pieces advancing from tile to tile. And that fundamental design carried over to Thunderbolt. Had there been more sophisticated technology available to me at the time, Thunderbolt would undoubtedly have a different look than it does today.

Lastly, it reflects the kind of game that I like to play. Good thing too, given all the years I've spent working on it thus far. In fact I still enjoy playing it. I've got a great character at the moment — Colonel Steel — she recently scrounged a blaster rifle and is about to explore Region 12, having successfully neutralized an icky lump infestation that threatened to

overwhelm the regions above.

IMG: Why did you choose to initially port Thunderbolt to the Mac rather than the PC?

Scheifler: Two reasons. First, I knew people who had Macs, and others who had PCs, and so had informally become familiar with both of them. I concluded that if things went down the tubes I'd rather be stuck with a Macintosh at home than a PC. The PC was great for playing games, but if you wanted a machine that was easy to use and could do useful work as well as play games, then the Mac was the way to go.

The other factor is that the PC *did* have scads of games, whereas the Mac had almost nothing. But having watched the Mac market for a year or so I'd notice the gradual increase in games and the clamoring for more. So I figured that by going with the Mac there'd be a smaller market but perhaps correspondingly deeper penetration into that market. The next step would be to move the game over to the PC.

With my Windows port now well into Beta Test, my view of the PC has naturally changed as I became PC literate. The introduction of Windows by Microsoft made a really big difference. Sure, I've heard the comments that it's only a band-aid on top of DOS, but let's face it — it's a pretty nifty band-aid! Windows, together with its wide range of high quality products, makes it a very good platform — arguably several steps behind the Mac in some ways, yet equal to or perhaps ahead in others! By the way, it's a real credit to the Microsoft folks that the port to Windows took just three months. Their C libraries were quite robust and provided capabilities equivalent to what I used on the Mac. Made the job fairly straightforward.

IMG: Who is MegaCorp? And, other than John Calhoun's graphics, who else has a hand in your creations?

Scheifler: MegaCorp International was incorporated in July of 1991. "MegaTech International" had been the original name used in DOOM, but when my attorney's firm did a name search it turned out that another company already had that name, here in Massachusetts, in fact. Hence the shift to MegaCorp. In both instances, the name was created to convey the sense of a globe-spanning conglomerate which has an ongoing presence within the context of the game. Just for grins I made it the name of my real company as well.

Who are we? Well, we're a small company in more or less central Massachusetts with aspirations toward being a leading light in the home entertainment field one day. Thus far (this is really just a running joke with a friend) when people call in it's the janitor who typically answers the phone (when the answering machine doesn't) and the guy tends to gripe a lot about his slave-driver Corporate President. Me, of course, in both instances ;^)

Over the years I have especially valued the advice and consultation of a close circle of friends whom I've known for well over a decade now, all of whom have been in the computer industry for as long, or longer, than I. Also, that role playing game that Steve Russell and I were in was science fiction based (Jim Burrows and Earl Wajenberg were the games-masters), and it too was a source of inspiration over the years.

IMG: One of the big differences between MT and other games is that the player is offered a number of options for solving a particular problem. Also, a real-time element is built in that requires the player to repeat certain tasks in order to succeed (knocking down walls for instance). While this speaks to the particular character of Mission: Thunderbolt, it also says something about your ideas of how games should be put together. How does MT reflect your outlook on game design and, more broadly, your view of the world?

Scheifler: The game, while just a game of course, nevertheless embodies certain aspects of life — elements that make life interesting and challenging, and thus hopefully make the game interesting and challenging as well.

Here's a mundane example. Let's say your car breaks down on Route 128. What do you do? If you have the tools and the expertise you can try to make on-the-spot repairs, and if successful, then away you go, continuing your journey. If repair proves impractical through want of tools or knowledge or sheer obstinacy on the part of the vehicle, then you can try to hitch a ride...if there is other traffic on the road. If there's no traffic, or you don't feel comfortable with thumbing a ride, you can walk along the road looking for a motorist's callbox or gas station. Or you can head off into the country in search of a residential area where some homeowner might lend a hand or let you use the phone.

This all means you have a problem to overcome and you can select a solution. And if that doesn't pan out you can devise another. You aren't locked into dealing with the problem in just one way. There are alternatives. Of course, one doesn't always have such latitude. Sometimes there *is* only one viable solution to a given problem. But generally that's not the case.

That's a key "real life" element that I find interesting in games. The ones that pretty much lead you around by the hand through a set story with little or no option to really influence the action tend to be boring for people like me who want active participation. And, no, a simple sequence of decision trees, leading to a couple of different endings, is not what I'm getting at here. Rather, if I, the player, am not clever enough, or resourceful enough, to overcome a defined problem in quite the way the designer(s) intended, then there should be inherent flexibility in the design to allow alternative solutions — not just an online hint for a "correct" solution. If I decide to head off at a tangent from that road, looking for a homeowner, I then want to be able to do just that and explore the consequences of my choice, rather than being forced by rigid structure to instead flag down another driver.

Thunderbolt has a first cut at this philosophy. For example, if you find yourself unable to locate a way into an as-yet-unexplored area, you can hunt for hidden entries, ignore it altogether, or even pick up a handy blunt instrument and bash down the nearest wall. Similarly, if you find that you get eaten alive every time you try to venture down to the next region via the stairs, there are various alternatives depending on what equipment you have or can scrounge or how crafty you are. One of the alternatives is to go find some place safe on *this* region, pound a hole in the floor, and then drop through. Situations like this abound in the game, with resolution being governed by player resourcefulness rather than predetermined solutions. And discovering what *can* be done is part of the Thunderbolt playing experience. I expect to evolve this concept further over time.

Real life is reflected in other ways too, one of which you mentioned. Go grab a crowbar from the basement and try to bash a passage through one of your walls. Non-trivial, ain't it? Takes time and effort and determination and patience — just like in Thunderbolt. But your effort is rewarded by your being able to successfully accomplish that task. And if you find a better tool, the job even gets easier. This sort of time and effort dependency also exists for a meta reason: if you could just click a button once and have the problem go away, it'd be kind of moot as an obstacle — both meaningless and inconsistent — hence you have to work at it.

Hand-in-glove with open-ended choices is the changing nature of Thunderbolt from game to game. For instance, resources that were readily available for one character may be in short supply or nonexistent for the next, thereby directly encouraging player ingenuity and initiative.

Ideally these two factors together provide a high degree of replay value. Which was also a design goal. Personally, I find it difficult to shell out \$50 for a game that I finish in a couple of days, if replay is meaningless when its “solutions” are known. From what I’ve heard so far, Thunderbolt seems to be achieving the replay design goal fairly well.

IMG: Suppose you could design a game where you were not hampered by the limitations of current technology. What would it be?

Scheifler: No doubt about it. The holodeck aboard the Enterprise on “Star Trek: The Next Generation.” Total immersion in the gaming experience. You are there! And you become as fully absorbed in its pseudo-worlds as that time when you stayed up all night immersed in the best novel you have ever read — but gloriously more so. And there would be an entire library of scenarios providing mystery, adventure, love, and pathos. Or ad hoc scenarios ad-libbed and guided by your heart’s desires via artificial intelligence. It would be the ultimate in computer gaming.

Barring that (grin), I’d design a game composed of a helmet or goggles with active matrix color that spanned the peripheral vision of your left eye all the way to the farthest extent of the peripheral vision of your right eye so that where ever you looked you saw more of the game world, and as you turned your head the point of view would smoothly change accordingly. There’d be other devices worn or manipulated to enable you to fully interact with the world and its content, and sensory input to provide tactile feedback, with CD quality sound to help bring it to life in all the myriad ways of a real environment. The game, and hence its storyline, would have tremendous depth and breadth, ideally embodying all of the elements we have today — flight simulators and gothic mysteries, high magic and arcane sciences, and dazzling combat sequences ranging from the primitive to ultra high tech — either as individual scenarios or interwoven to provide months if not years of ongoing entertainment value. Yet also loosely structured when desired to provide open-ended decision making on the part of the player, to thereby draw the player into the story and making them fully a part of its unfolding ... because some people like to have a story told to them and others want to be active participants and influence its outcome. Lastly you could play any role within the game — the hero or wingman, the valued assistant or the worldly wise adventurer — based on personal interests or abilities; and of course with high replay value, in that one can choose a different role each time and/or have the game evolve such that each replay offered new twists and challenges to be overcome by the more experienced players ...

Sorry, I seem to be rambling and hand-waving. Of course I am talking “virtual reality” here. But that has become a much over-used and abused catch phrase these days, so I’ve been avoiding it. Anyway, notable academic research, as well as industry leaders, have been pursuing the key elements for several years and I have no doubt that it will be pulled together into a plethora of consumer products during this decade and into the next century. Probably sooner than we think. The 1980s saw an incredible evolution in graphics, the emergence of high quality sound tracks, and extraordinary simulation technology. The ‘90s are continuing that evolution, particularly with 3D representations. I’d like to evolve the look and feel of my own games to take advantage of the revolutionary advancements that are even now occurring in the industry as well as the trends of the future.

To better try to answer your question, there is no one specific future game that I dream about doing per se, but rather I’d like to pull together emerging technologies and do a major advancement in the industry, incorporating some of my design philosophy. With respect to current technology, I’d sure love to meld Origin’s 3D graphics from their Ultima Underworld series (or top-down views from The Black Gate) with Thunderbolt — that’d be a dynamite

game!

IMG: What are you currently working on and what might we see from you in the future? You mentioned Mission #4 in one of your communications. *Mission #4*?! My god, how far ahead have you planned? George Lucas planned nine Star Wars and so far has done three. Is the Mission series intended to be your life's work?

Scheifler: You got me to chuckle on that one. George and I are tied with respect to deliverables (as I mentioned there are three missions already out for the VAX), but he makes lots more money than I do!

When I started Thunderbolt/DOOM back in '86, I created a storyline that spans five missions: the first of which I've shipped on the Mac and will be released for Windows sometime this fall, two more have been heavily "alpha tested" (that's a pun by the way) under VMS and are ready for porting (one is in the same complex and the other is at MegaCorp's Far East manufacturing facility), and the last two which are still in the concept stage (most likely on a near-earth space station and the other at the bad guy's home planet). After all, there has to be *something* for you to do with that anti-matter bomb you retrieve in Thunderbolt!!!

Of course, capitalism being the way it is, they'll only see the light of day if a lot of people buy the first mission and holler for more. Hopefully your review in last October's issue of MacUser, and having won MacWorld's 1992 Macintosh Game Hall of Fame Award for best role-playing game, will help in that regard!

And, yes, I'd love for it to be my life's work. Okay...here comes the great revelation. Last August I turned 40. Call it a career crisis, or a mid-life crisis, or (as I like to think of it) as pursuing a once in a life time opportunity...I left Digital this past January after 17 years of service in order to devote full-time to MegaCorp International. My goal is to have a thriving home entertainment business within five years. But, who knows, a year from now I could be looking for a regular job again ;^)

Fingers firmly crossed!