

The Convergence Conundrum

© Copyright 2001 Noah Falstein

nf@theinspiracy.com

Convergence. It has a commanding ring to it, a sense of pathways coming together, leading us into a glorious future. It's peanut butter and chocolate coming together in a serendipitous collision, computers, telephones, and modems creating the World Wide Web. But let's not forget that convergence is also what happened to the Titanic and an iceberg.

In my career as a game designer I've always embraced variety and change. A love of change is practically a prerequisite for a long career in this industry, as it has changed so much over the past twenty-five years. I've been witness to a fair number of convergences, of computer games with film, with novels, with the Internet, with education, with wireless communications, and with corporate training. Some have had the unexpected synergy of the world wide web. Others have had more of a cold North Atlantic overtone to them.

It's important to realize that computer and video games represent a convergence in and of themselves. There are few careers that bring together as many wildly divergent disciplines as electronic game development. In order to make a successful computer game, you need to combine the logical, systematic disciplines of computer programming and project management with the creative disciplines of art, music composition, and writing, and the commercial factors of sales and marketing. When I began creating computer games in the mid-70's they were almost exclusively one-man jobs (and I don't use the male gender in its inclusive sense). Now, although it is still possible to do interesting games single-handedly, large Hollywood-style teams are the rule, and the mix of divergent talents and disciplines has expanded even further. The new generation of Massively Multiplayer Online Role-Playing Games (MMORPG's) even have social scientists and psychologists on their development teams to help craft their complex social infrastructure.

But the basic blend of software engineering and entertainment is at the core of even the simplest one-person games. It's an uneasy mix at times. In the early 80's, there were several attempts to apply the management techniques of business software engineering to game development. The biggest motivation was to quell the runaway schedules and budgets that have always plagued game development. I was working at the time for a major toy company that had recently had their first big hit in the electronic game field, and was eager to break into the burgeoning field of video game development. Our new "Advanced Research" group had worked with Texas Instruments and their new home computer, the TI 99/4, and was making games for it, but when our boss wanted us to make schedules, we had trouble setting deadlines. I remember a meeting that I'll relate, changing some details to avoid embarrassing those involved. Our dozen or so twentysomething engineers met with a management consultant I'll call Jim, who had been working in software since the mid-50's when computers were the size of dinosaurs, and not much faster. He had a straightforward approach to solid deadlines. First, we had to

set some well-defined specifications.

"O.K.," said Jim, "Let's make a list of the technical specifications this software has to have. It has to run on the TI hardware. What's the processor speed?"

Soon we had a list of the technical specs, ROM and RAM, I/O and the like. Then it was time for the software.

"This first game, Alien Attack. What are its primary objectives?"

My fellow engineers and I were a bit nonplused. Finally, someone ventured, "We want it to be scary."

Jim's confident smile faded. "Well, O.K." He thought for a moment. "How do you make it scary?"

Harry, the engineer in charge of the project, had been working on that. "Well, there are these eggs. And they're hatching into dangerous space aliens. But the eggs have numbers on them, like timers, and they're counting down, starting at 10, and when they hit 0 they hatch. It builds tension, you need to hide behind the eggs at first to be safe from the aliens, but then the eggs hatch into aliens themselves. It's scary."

Jim's brows creased. "So how do you know if it's scary enough? When are you done?"

Harry thought about that for a while. "Well, it's scary so that the player feels good when they blow up the aliens. It's fun. Maybe we're done when it's fun?"

That brought an enthusiastic chorus of murmurs from the rest of us, "Yeah!" "That's it!" "Fun is what it's all about."

Jim was frowning now. "But how do you quantify that? How do you know if it's fun enough?"

And that was indeed the problem. We didn't have an answer then. Often, we still don't have one now, twenty years later. That has been the bane of many a computer game schedule – the game is designed according to the original specifications, but it's just not fun enough. It's not that the concept of fun itself is completely subjective. On the contrary, in my dealings with filmmakers years later at Lucasfilm and Dreamworks, I found that, although far from a science, much has been learned about those same fuzzy issues like "scary" and "fun" in filmmaking. Alfred Hitchcock, for example, had the mechanisms of suspense honed to a fine art – and what is suspense but "scary fun"? One filmmaker told me about some basic principles of using sound design to build suspense. Sounds are scarier when they seem to come from behind the viewer, triggering basic survival reactions. Having specific music that plays quietly before something scary happens increases our anticipation and magnifies our reaction. But having music STOP playing shortly before something scary happens works on an even more subconscious level. We know something is about to happen, but we don't know why. That it itself is scarier.

Some of those techniques have been directly applicable to computer and video games. There are in fact many techniques that have been successfully adapted to games from software engineering, education, and Hollywood video and sound production. But Hollywood, for example, has had the advantage of relatively fixed technologies for many decades. Since the late 1930's they have been making essentially modern movies that can still have the ability to captivate and entertain us. The cameras have become lighter, color has become ubiquitous, and special effects have improved, but the basic techniques have not changed.

Game development is by comparison still in the process of integration. We've had changes in just the last twenty

years that are much more fundamental than those that film has gone through. In that time we've gone from text descriptions to 3D worlds with lighting and textures, from electronic beeps and boops to full surround-sound orchestral soundtracks, from one person in front of a lonely TV screen to tens of thousands playing simultaneously. And the size and cost of top-notch games has ballooned accordingly – and yet, with the Internet, it's still possible for a talented individual to create a fun, simple game in a few months and distribute it worldwide. Games may be on your desktop computer, or on a console plugged into your TV, in an arcade or on a hand-held dedicated system, or even on your PDA or cell phone.

This means that as game developers, we've had a tough time coming up with standards. For one thing, the ultimate goal is still that damnably elusive "fun". For another, the range and diverse capabilities of delivery platforms, constantly changing and improving, thwarts efforts to standardize.

Game companies have for the most part learned to live with this uncertainty, and even to impose some higher-order structure on the chaos. It's encouraging that people who do their apprenticeship in game development begin to build an appreciation for certain universalities. It is still often a matter of subjective experience and personal knowledge, but gradually the patterns and techniques of the industry are being set down. To return to the original metaphor of The Titanic, we may not have learned how to always chart courses through clear water, but we've mapped out many of the biggest icebergs. Curricula of game development are beginning to be taught in colleges and universities worldwide, books are being written, and as the population of developers ages, it is likely that more will turn to teaching to pass on what they've learned.

But this lack of externally validated standards still tends to trouble those who seek to apply game techniques to education and business. Even the language can be a problem. I was the third employee at Dreamworks Interactive, and I saw that company grow rapidly, integrating people from a business software background at Microsoft, a film background from Dreamworks SKG, and a game background from many entertainment software companies. Early on, we realized that there were some basic terms that meant different things to each group. One of the most fundamental was the term "development". To most game industry veterans, development refers to the entire production process of a game, particularly including the main phase of integrating programming and artwork. To movie industry workers, development is what happens *before* production, when a movie is being discussed and the initial concept written down. And to Microsoft, developers are programmers, and a "development issue" is a problem in the code. It took a while to sort out the confusing in some of our early meetings that resulted from that particular term.

Educators tend to have a love/hate relationship with games. The love comes from the realization that the most effective learning happens when the students are actively engaged in what they do, enjoying themselves and eagerly seeking knowledge. Games are good at this – in fact, Marshall McLuhan once said, "Anyone who makes a distinction between education and games doesn't know the first thing about either." But the hate comes from the amount of time those same students are willing to devote to playing games where the learning seems to be exclusively about which special combinations of martial arts moves are required to dispatch each opponent in the most bloody way possible. Games certainly tap into very primal learning mechanisms. The most popular games and indeed the most generally popular forms of all entertainment tend to be closely allied to basic issues of survival and reproduction – hence the widespread interest in issues of sex and violence. Although much has been made of the darker uses of this interest, let's not forget that there are positive sides too. Many of Shakespeare's works are deeply involved with sex and violence as well – much of great literature in general, in fact. A well-designed game may be able to teach very effectively, precisely because it plugs into basic human drives. The trick is in what they are teaching.

Many of the best educational games I have seen use a process I've often heard called "stealth education". They are, at their hearts, entertaining games. They use all the techniques we game designers have composed and divined over the years to make them fun. But in order for the player to maximize their fun, these games are designed so that it becomes desirable to learn more about the subject in question. These techniques have been used for some years to teach basic skills like math, reading, and geography – the Carmen SanDiego series being one popular example. But the range of this kind of game is expanding.

For instance, one game called Hungry Red Planet, that at this writing is nearing completion, is designed to teach principles of good nutrition. But it does so by giving the player control over a series of settlements on Mars, letting them build farms and support structures, explore the landscape, and protect themselves from dangers. But the best way to do these things is to maintain good nutrition for your settler's meals, and the best way to do that is to learn as much as you can about actual nutrition facts built into the game. The game, Hungry Red Planet, is soon to go into a testing phase funded by an NIH grant to see if it is in fact an effective teaching method, and a search is on for a publisher to bring it to the public as well.

This is not an isolated example – an increasing number of games are being developed to help with social problems and to educate through entertainment. They can be slow to be adopted by traditional educators, but are often avidly played by children on home computers, and may soon have a significant impact.

Another area that has been slow to adopt game technology techniques is traditional business. Despite the billions of dollars spent each year in corporate training budgets, only a small fraction is devoted to game-based learning. Here, the biggest hurdle seems to be the negative connotations that games have in terms of an implied triviality or frivolity. But convergence is beginning to be seen in the corporate arenas too. The US military was one of the earliest adopters of this technology – perhaps because wargames have been an accepted part of military strategy for over a century, and perhaps because of the particularly potent ability of computers to accurately simulate combat vehicles and strategic encounters in a safe and inexpensive manner. I witnessed one rather novel use of commercial game technology in business several years ago when I helped design a training simulation for Shell Oil's exploration and production arm. They had taken the game Quake, known in the games industry as a "First Person Shooter" and used the modeling extension provided. They modeled not simply another game level to hunt opponents, but instead an actual North Sea oil platform that was under construction. The idea was to train employees destined for that platform in a safer (and cheaper) facility on the mainland. Guns were replaced with fire extinguishers, providing for an active focus for the players that would teach them safety procedures while letting them absorb the physical layout of the platform in a way that simple study of blueprints would never have matched. Ironically, a fall in world oil prices prompted Shell to cancel this along with some other promising projects, but the concepts are still sound – and oil prices are on the rise again.

So, will the convergence of game techniques with other areas like education and business training result in happy synergy, or cold disaster? At heart, I'm an optimist. We may have our own icebergs to dodge, but the final payoff of delivering people to new lands of knowledge and enlightenment is worth the risk.