



What HCI Designers Can Learn From Video Game Designers

<i>Randy Pausch (chair)</i>	<i>University of Virginia</i>	pausch@virginia.edu
<i>Rich Gold</i>	<i>Xerox PARC</i>	richgold@parc.xerox.com
<i>Tim Skelly</i>	<i>Microsoft Research</i>	timsk@microsoft.com
<i>David Thiel</i>	<i>Microsoft Research</i>	dthiel@microsoft.com

contact:
 Randy Pausch
 Department of Computer Science
 Thornton Hall
 University of Virginia
 (804) 982-2211

ABSTRACT

Computer users have tasks they need to perform, and are therefore motivated to overcome poorly designed interfaces. With video games, there is no external motivation for the task — if the game's interface is not compelling and entertaining, the product fails in the marketplace. Many aspects of game design, such as an *attractor mode* to draw users toward the game, have direct relevance to other domains, such as information kiosks. This panel will consist of video game designers who will relate their design methodologies, techniques, and other experiences which will help HCI designers create more compelling, engaging, and effective interfaces.

The panel will consist of panelists' presentations followed by a large allocation of time for interaction with the audience's questions. The panelists' presentations will include demonstration examples drawn from coin-operated and computer-based games.

KEYWORDS

video games, design process, direct manipulation, input devices, interface design, design methodology, consumer acceptance

PANELIST'S STATEMENTS:

Randy Pausch: Why Is This Topic Relevant?

As the computer and entertainment industries continue to merge, interface designers need to be aware of divergent paradigms and techniques, which will come from a much wider variety of sources than in the past. At the same time, user expectations are rising. Pong existed over twenty years ago, and we are now designing for a generation of computer users who literally grew up on video games. I believe the HCI community can benefit from the lessons learned in the

video game industry, and I am not at all surprised that corporations like Xerox and Microsoft are now readily hiring interface designers whose primary expertise has formerly been in the gaming and toy industries.

Randy Pausch is an Associate Professor of Computer Science at the University of Virginia. He is an NSF Presidential Young Investigator and a Lilly Foundation Teaching Fellow; his main research area is Virtual Reality.

Rich Gold: Spreadsheets, Princesses and Altered Users

An interface is where one system touches or interacts with another system. For instance, we can think of a child as interfacing to a doll when cuddling it. However this is not the way we usually use the term in computer science where it has come to mean a third system, usually of widgets and control devices, that sits between the user and the "real" program. The doll/child relationship is different: the child does not interface with the exterior of the doll to reach some deeper "dollness", and I believe the same holds for video games.

There are two areas where we can profitably talk about interfaces and video games. First, almost all video games are interfaces between the child and an elaborate scoring system. This system is tuned such that as the game progresses it is harder and harder to score points but the points are of larger and larger values. If done just correctly the effect is a near drug-like high in the child. Most games, it should be noted, fail at this task and hence at the store shelves. Second, there is a hardware device called a joystick (or pad or Glove) which interfaces between the child and the game. A good joystick does not automatically give a high score, but closes the eye/game/hand feedback loop in such a way as to provide pleasure. Again, we find good and bad sticks in about the same ratio as good to bad spreadsheets.

However, the child playing a video game is a dynamic system that over time will get better at playing games, will score higher points and will reach deeper levels even if the game interface remains constant. This implies actual changes to the child and the question can be raised as to the effect of these changes in interfacing to the rest of the



culture he or she is embedded in. It can be said that "a video game's job is done when it is purchased from a rack at Toy's R Us, but the user it creates will last a life time." It is becoming belatedly understood that a computer program's success must be judged in a context that extends well beyond the desktop and so it is not clear that by finding the Princess we have found an interface.

Rich Gold is a composer and cartoonist who co-founded the League of Automatic Music Composers, the first network computer band. As an internationally known artist he invented the field of Algorithmic Symbolism, an example of which, *The Party Planner*, was featured in *Scientific American*. He was head of the sound and music department of Sega USA's coin-operated video game division and the inventor of the award winning *Little Computer People* (Activision), the first fully autonomous computerized person you could buy. For five years he headed the electronic and computer toy research area at Mattel Toys and was the manager of, among other interactive gizmos, the *Mattel Power Glove*. After working as a consultant in *Virtual Reality* he joined Xerox PARC, where he is now a researcher working on ubiquitous computing, an artist in residence program, tiny hand held devices and the philosophy of stuff.

Tim Skelly: The Design Process & Usability Testing

There are many diverse categories of video games: arcade game beat-em-ups, computer-based graphic adventures, puzzles, simulations of societies and simulations of flight. The 'task' underlying these ornate and diverse interfaces is to help the user reach a kind of focused, highly engaged state of mind, the 'Flow' state. It is a pleasant and self-motivating condition that would be desirable in any user. The way each video game facilitates reaching that state is similar from one game to the next: Step one, present a goal; Step two, provide clear-cut feedback to the user as to their progress towards the goal; Step three, (and this is the step at which good video games excel) constantly adjust the game's challenges to a level slightly beyond the current abilities of the player. That's it — simple, but still a formidable design challenge. How each game designer deals with that challenge usually depends on the economic realities of the marketplace. Luckily, video games are virtually all interface. No one can complain that the product is shipping without enough development resources devoted to UI.

PC-based games live or die in the same marketplace as more conventional software. Developers of this class of games, like their 'productivity tool' counterparts, must contend with marketing promises and guesses, and ever-shifting deadlines. As such, an examination of PC-based game development offers few novel guide posts towards effective UI. Coin-operated arcade games exist in the same economic world as jukeboxes, pinball machines and vending machines. How does UI development differ

here? Video game players are users that will turn away from a bad product at the drop of a quarter. Because of this, coin-operated game distributors will not buy arcade games that do not have promising coin-collection reports. This means that arcade game developers must put their machines through the most brutal usability testing in the software world — prototype games on location with real players spending real money. For an arcade game to show large weekly earnings, the game interface must cater to a player's every need. Players must be able to immediately understand how to operate the game, and user help (if necessary at all) must be fully 'context sensitive.' Success rates improve when products undergo exhaustive testing and iterative redesign. Designers of coin-operated games and programmers follow these vital development disciplines, not because they think it is a good idea, but because manufacturers of coin-operated video games can't even give away their failures.

Tim Skelly is a Researcher with Microsoft's Advanced Technology group. A fifteen year veteran of the video game industry, his credits include the world's first co-operated video game, *Rip-Off*, the FASA *BattleTech* Center and SEGA's *Sonic the Hedgehog II*.

David Thiel: Effective Use of Sound in Games

All video games are not paragons of virtue in their use of sound. They are just a venue where a lot of implementation energy has produced a considerable body of work in a new field. Much can be learned from the things that video games (and pinball machines) have done with auditory feedback, both right and wrong.

Within the framework of the previous statement I will discuss some of examples that deal with issues emerging from the following questions:

- Is there a difference between interactive sound and traditional music and sound post-production?
- What are the challenges and constraints that interaction places on sound designs for user interfaces?
- What kind of sound production capability is nominal for good user interface interaction?
- Does music have a place in interface?
- Who is equipped to do it?

David Thiel joined Microsoft Research in July 1993. David has designed and implemented sound drivers used in over 70 interactive entertainment products; in all cases providing sound effects, voice and music. In 1990 David was the project manager and designer for four video games using a novel gesture-based input device. He demonstrated these games at the *Virtual Reality Gallery of SIGGRAPH*. Adlib and Yamaha have contracted David as an FM synthesis consultant. He most recently wrote the English manual for Yamaha's FM synthesizer chip OPL3. David has experience in all forms of commercial synthesis and has written sound drivers that provide responsive sound support for asynchronous interaction.