

# An Interpreted Demonstration of Computer Game Design

Chuck Clanton  
Aratar, Inc.  
220 Downey Street  
San Francisco, CA 94117 USA  
+1 415 753 5855  
chac@aratar.com

## ABSTRACT

Perhaps you believe computer games have no relevance to HCI, or perhaps you are just reluctant to let your kid blast you out of warp space in seven seconds flat to find out why they spent \$49.95 for a dumb game. This is your chance to find out whether game designers know something you do not, without embarrassing yourself in front of your kid.

## Keywords

CD-ROM games, entertainment software.

## INTRODUCTION

Some years ago, I asked myself why millions of people spent their own hard-earned cash to buy *Myst*. Microsoft Word may sell more copies, but those purchases are not so eagerly concluded. I spent over a hundred hours watching people play *Myst*, several other popular games, and a few failures. I learned that *Myst* is a great game for a lot of reasons, but most importantly because of the way its designers thought about it. Though games have been used for training applications, the way game designers think seems of more general benefit. I needed to design games to learn to think like that. I will demonstrate both popular and not so popular games to illustrate some game design principles that were eye opening for me. I hope you find them thought provoking.

## THE USER INTERFACE OF A GAME

The HCI of a game can be divided into three levels. *User interface* is typically used for the perceptual and motor level: how the joystick works, what instruments are displayed on the screen, etc. To avoid confusion, I will call this the *game interface*. The “physics” of the game world comprise the *game mechanics*: what jumps and dodges your proxy character can do, the flight dynamics of your star fighter. This resembles the functionality in an application’s UI. Finally, every game makes the player strive to achieve a goal. Discovering and overcoming these constitutes the *game play*. This corresponds to the user’s tasks and subtasks in application software.

Most games, including the most successful, have game interface flaws obvious to a CHI designer. Flaws in game mechanics have a greater impact on game play. The CHI community has knowledge and methodology that could improve game interface and mechanics, making them accessible to more players. However, game designers and publishers alike are adamant that game play is the deciding ingredient of a good game. *Resident Evil* has a seriously flawed game mechanic but great game play and it’s one of the most wildly successful games of the last few years. Why? Game play is about overcoming problems, so problems in the game interface and mechanics can be masked by the fun of solving problems in the game.

People enjoy solving problems. Different kinds of problems for different people and genders, but we all love to overcome meaningful obstacles. Obstacles are the designer’s friends. Understand them, balance them, tune them, but do not annihilate them. Make them enhance rather than detract.

The game design community is devoted to iterative design and testing to optimize for fun. The primary role of the game designer is to repeatedly imagine playing the game and to play test game fragments when they are available to tune the design to be more fun. If CHI designers spent less time optimizing legibility of icons and more time making applications engaging, we might be pleasantly surprised at the results. Fun may be the most important human factor.

## GAME GENRES

We demonstrate games from different genres because they require different perceptual and cognitive skills and thus illustrate different principles. *Adventure games* focus on navigating a world by expressing intent (click on the door to leave), and solving key and combination lock puzzles. Keys are kept in an inventory for later use on matching locks. *Phantasmagoria*, *Myst*, and *Riven* are graphic adventures. *Action games* require perceptual and motor skills to drive, fly, and run around and act in a world. Different tactics are needed to defeat different enemies, and a strategy must be devised to win. Action games include shooters, sports, flying and driving games. *Road Rash* and *Resident Evil* are examples. *Strategy games* involve command and control over an army in a world. Resources must be managed, and a strategic thinking is required to win. Some simulation games and real-time strategy games like *Red Alert* fall in this genre. The greatest challenge in real-time strategy games is the player’s limited capacity for attention. *Miscellaneous* We also will look at *You Don’t Know Jack*, a quiz game, and *Diablo*, a role-playing game where you construct your surrogate in an ancient mythic world.

## HOOK ‘EM FAST AND HARD

*Establish the quest.* Most games have an alluring goal that is quickly discovered and easily stated. In *Red Alert*, conquer the world. This overarching goal provides context to learn the game. Goals can change but are needed early.

*Provide a gentle on-ramp.* Learnability is a major CHI issue. Games do it well for novices, experts, and a broad range of ages (but not genders). *Road Rash* is renown for its “on ramp”. Everyone wins the early races easily due to clever motorcycle racing dynamics that favor the player.

*When players select a difficulty, they accept it.* Setting the game difficulty from “Easy” to “Hard” changes the game play and sometimes the game mechanics. In an easy *Red*

*Alert*, attacks come later and are slower and less ferocious. The player who picks a hard setting says "Eek, it's too hard!" and plays again.

Let each player progress at their own rate. Games are usually divided into levels. After winning a level, the player moves to the next level—a fresh new environment with new and harder challenges. In *Road Rash*, the player wins and can afford a better bike and can enter harder races. Level designers are specialists. Great ones are rare and sought after. Application designers should take note.

### KEEP 'EM HOOKED

Spread clues, tools, and obstacles out but not too much. Discovering locks or keys, weapons or enemies, locations or maps is fun, and anticipates coming game play. When the time or distance separating a lock from a key becomes greater than the import of the one found first, the fun is often diluted. Players criticize *Riven* for its difficulty due to the distance between the combinations and their locks. The mechanical plausibility of *Riven*'s puzzles prevents this distance from feeling like a cheap design trick.

Avoid lengthy dead ends. *Silent Steel* is a strategy game with an interactive dialogue game mechanic. I was stuck late in *Silent Steel* and called the help line. They could not help because an earlier unfortunate choice that was not visualized in my current situation was crucial. Worse, an early choice in some games leads to defeat much later. It is unfair to need to know the end of the game to play the beginning correctly. Results of actions should be visible and should be sufficiently localized that the player can explore and overcome obstacles without lots of tedious backtracking. (The ability to backtrack after dying can cleverly be made a game problem by converting Save Game to a limited resource, as in *Resident Evil*.)

Pressure can be fun. Action games are fast paced. Adventure games are not. Testosterone-enriched teens like action. Others often prefer adventures. The heart pounding "action" end game in *Phantasmagoria* adds time limits to its adventure game mechanics to give its adventure game audience a special treat.

Give hints not answers. Being stuck is no fun, so games often have a help system. If help gives the complete solution, players use it EVEN THOUGH IT SPOILS THEIR GAME. *Phantasmagoria* provide hints not solutions to avoid this paradox. Well written hint books and web sites give just enough help to get the player unstuck.

Avoid linear, monotonous pacing. Problems should get harder but not monotonously so. Easier problems are interspersed to build the player's motivation and confidence for the harder ones. Parallel problems allow the player to work on something else when feeling stuck on another problem. Parallel problems can be designed to offer subtle help with each other when solved, especially if they require very different types of thinking. This balances the game for players with different types of skills (e.g. visual versus auditory thinking).

Reward game play with media. Interactive choices in games often have non-interactive illustrations and rewards. Shoot at an enemy and see it die. Find the troll and learn where the treasure is. Good games favor interactivity over

non-interactive elements. The "Hollywood mistake" of too little game play eliminates the compelling reason to sit in front of a computer rather than a TV. *The Beast Within* is an interactive video game with a rich narrative with puzzles as obstacles like a Hollywood film. No other game has succeeded with this much non-interactive material. This is treacherous design territory. Application software fails at the other extreme. Few cool illustrations of what I did, no victory celebrations. Where is the fun in that?

Confusion is not fun. Keep the problems in the game play, not the game interface or mechanic. First person follow, where your proxy is on the screen, displays information intuitively. In *Tomb Raider*, the player can see our heroine get hurt and limp after a fight. However, *Resident Evil* has a serious flaw because it matches a first person follow camera (screen left may be proxy left or right) to a first person aiming mechanic (left button fires at screen and proxy left). Which way should the left button fire? *Tomb Raider* solves the problem with automatic aiming (but the player can shoot and kill an unseen enemy).

Frustration can be fun. Accomplishment feels good in proportion to difficulty. The trick is to make obstacles appear harder than they are, and not be so hard the player quits in frustration. Failure must be fun too. Application software typically suffers from failing to let the user discover then solve a problem. In *Riven*, a player can push a button that unlocks a door elsewhere without first seeing the door. That player missed the fun of discovering how to unlock the door. Clues should not be too subtle. Remember, the player has not lived with them as you have.

Trial-and-error is not fun. The player must believe that if they had just been a bit smarter, a bit quicker, a bit more skillful, they would have won. Losing less and less badly is addictive. In *Silent Steel*, I made an innocuous choice and without warning, I lost the game. I could only know that this choice was fatal by making it. That feels unfair. Worse, I lost control when I made this choice in violation of the game's prior mechanics. I screamed at this game.

It's fun to be known. It's great when *You Don't Know Jack*'s announcer says, "That's your third miss, player number 2. Shape up!" or late on Friday night, "What are you loser's doing at home? You should be out partying!"

Make a great game and players will master its complexity.

*Myst* is the top-selling computer game and has a very simple and intuitive game interface and mechanic. The cursor's rollover feedback shows what you can do (an outward arrow over a door to leave the room). Most adventures like *Phantasmagoria* change mechanics for different puzzles. This adds little difficulty for gamers. Action games like *Quake* have a more complex game mechanic and interface with hand-eye skills for running and shooting, and can change with new vehicles or weapons. The action gamer must invent different tactics for each enemy and devise a strategy to win the game.

The *Red Alert* player forages for resources, builds armies and fortifications, scouts territory, defends against attack, and plans and launches attacks. Complex but very fun!

*YOUR MISSION, should you decide to accept it...* The designers job is to create fun game play, and insure that the game interface and game mechanic do not interfere.