StarCraft's Steps

How A Few Simple Changes

Could Make A Brilliant Game Even Better

By

Dave Howell

Eponym@howell.seattle.wa.us

Summary

Using some fundamental game design principals to make a few subtle but very significant changes in the behavior of a few of the components of StarCraft could have added new strategic decisions for sophisticated players and enhanced game play, without interfering with the overall flow of the game or significantly altering the experience for beginning players. A brief explanation of the relevant design principles is provided; then enhancements to the Terran Scanner and resource mining are provided as examples.

Introduction

Ludology

As John Lassiter has said on many occasions, what makes Pixar so successful isn't the technology, it's the story. Without a good solid story, a movie will be, at best, a "summer blockbuster" that is all but forgotten a year later. Special effects are the sizzle, but the story is the steak.

In computer/video gaming, or indeed, in most forms of gaming (including board games, sports, and gambling), the analog to "story" is the game design; the rules and framework around which the graphics, the AIs, the pieces, the rendering engine, and the other window dressing are arranged. Unfortunately, the principles of good storytelling are far better documented, and taught much more effectively, than principles of good ludology (the art and science of games, from "ludos," the Latin word for "play,"). Indeed, the fact that neither Age of Empire, Black and White, Company of Heroes, nor even Blizzard's own Warcraft III have been able to make StarCraft obsolete, despite its extraordinary age, underscores the fact that most, if not all, of the computer games available today were designed through guesswork and intuition.

Most players, of course, can talk about their favorite games only with the same vague arm-waving. A player might confidently state that StarCraft's Protoss are more fun to play than WarCraft III's Night Elves, or Age of Empire's English army (or vice versa), but will struggle to explain *why*.

The Golden Guidelines of Game Design

I was on a panel about game design at the World Science Fiction Convention in 2002. An audience member mentioned a game that they had problems with, and I responded by saying it violated one of my guidelines for good game design. There was an instant clamber to hear what my other guidelines were. Until that point, they'd just been casual rules of thumb, but the unexpected interest prompted me to try writing them down.

Since then, my handful of guidelines have developed into a pretty comprehensive set of principles and practices for evaluating and improving games. I've woven together ideas from game designers like Reinier Kniezia, Richard Garfield, Greg Costykian, and James Ernest, along with my own thoughts, observations, and experiences as a game designer, game design editor, playtester, and player. Although originally created for analyzing table games, the guidelines have proven very valuable for gaming in a much broader context, including sports, video/console/computer games, and gambling.

Although I've presented the Guidelines at a number of conventions, I haven't yet published any books or articles that lay them out; the written form of the five Axioms and twenty-one Guidelines is still a work in progress. Fortunately, only the first two Axioms and a couple of the Guidelines are needed to discuss some of the opportunities for enhancing StarCraft.

"I'm Not Dead Yet!": The Maintenance of Hope

Less is More: The Fog of War

While "the fog of war" originally came from a quote by a Prussian military analyst, its use in RTS games dates from Chris Crawford's game *Tanktics,* and its current definition as a game mechanism that retains a limited record of what was last directly observed, but does not update until the next successful direct observation, is almost entirely due to its key role in Blizzard's games. Wikipedia notes that *Tanktics* was "criticized for its fog of war system detracting from the fun of the game." I have not had the opportunity to play *Tanktics*, but I strongly suspect that the criticism stems not from a poor game design decision on Crawford's part, but from poor game design understanding on the critic's part. StarCraft's "fog of war" is a critical component of the game's ongoing appeal, because it is the primary mechanism that supports my first axiom of game design.

> **Axiom #1: A game is not fun unless a player believes they have some reasonable chance to win until the moment the game ends.**

A surprising number of people want to argue this point, usually because they've picked out a couple of the words without thinking the entire statement through. I'll briefly discuss some of the more common misunderstandings, using StarCraft to illustrate some of the points.

Assume a player is in the middle of a game of solitaire StarCraft. A 'god's eye view' shows the player has some reasonable perimeter defenses. The computer has a small attack force nearby, and enough extra forces back at its base to pretty much crush the player and end the game. At this point, the outside observer can see that the player has lost the game; they don't have access to enough resources nor the time necessary to build enough defenses to survive, the terrain doesn't provide the necessary opportunities to 'force multiply' the units they have, and the computer hasn't left a weak point that the player could use for a surgical offensive strike. Without the fog of war, a player's reaction at this point would probably be "Oh, great, I'm totally screwed," and to quit the scenario. A tiny minority of players might enjoy continuing to play if they decide to see how long they can survive before they're destroyed, but even these players do not contradict Axiom 1. They have merely redefined "win" from "destroying all the opponent's forces" to "lasting long enough to reach benchmark X".

There are, in fact, quite a few games out there where "last long enough" is the definition of winning. "Long enough" is marked by managing to get further through the level than last time, or to get a higher score than last time, or so on. Because this is such a common definition of winning in video games, digital game players are more likely than, say, card game players, to be willing to switch to this alternate definition of "win-

ning." Generally, though, players will tend to stick to a game's initial definition of "win," and once they no longer believe they can achieve that, the fun is over.

This is why the fog of war is so critical. In our hypothetical StarCraft game, the player does *not* have a reasonable chance to win. They have almost no chance at all given their situation and skill level. But as long as they *believe* they have a chance to win, they'll be able to have fun, and the fog of war conceals the information that would strip them of their belief.

So if not letting a player see their inevitable doom is so important for enjoying StarCraft, then why don't fog-penetrating tools like the Terran Scanners, Zerg Overlords, or Protoss Observers spoil all the fun? Because a player only needs belief in a *reasonable* chance to win, not a likely or guaranteed one, and all of the fog-penetrating tools were very well engineered to limit the amount of information they can provide. Scanners can only hit a few spots before they're drained; by the time they're recharged, the original information is obsolete. Overlords, with a limited range and relatively slow movement, can only safely see the edges of enemy territory. Observers can potentially return information from deep within enemy territory, but if they're spotted, they're so fragile that they're almost instantly destroyed. None of these tools will normally return enough information to utterly crush the hopes of the player, except when the computer is probably only minutes, or seconds, away from pouring those overwhelming forces into the player's lap and bringing that game to an end anyway.

Well-designed Scenarios

Because StarCraft generally meets the conditions for Axiom 1 so well, I'll only comment briefly on one other design component that reinforces it. The pre-constructed scenarios that come with the game could also make or break a player's overall satisfaction with StarCraft. If a player loses a scenario with some idea of what they can do differently that will let them win the next time they try, then they will start their next game believing they have a reasonable chance to win. On the other hand, if scenarios escalate in difficulty too steeply, a player might end up losing that scenario repeatedly, and have no idea what they have to do to change the outcome. I can think of more than a few computer games in my closet where I've hit a "wall" like that, and it almost always means I throw the game in a box and never play it again.

StarCraft's scenarios also generally manage to avoid making the opposite mistake, of not escalating the difficulty *enough* between scenarios. The corollary to Axiom #1 is "A game is not fun unless a player believes they have some reasonable chance to ***lose*** until the moment the game ends." There were a couple of scenarios that failed to provide me with much sense of accomplishment because they seemed too easy. However, this is much less of a problem than ones that are too hard. It's usually not obvious that the scenario's going to be a cakewalk until it's almost over, and, because it was so easy, it tends to be over quickly. As long as there aren't too many over-easy scenarios in a row, this isn't much of a problem.

Different players have differing abilities, of course, so a difficulty jump that's a little too small for one player might be dismayingly large for another. The cost/benefit ratio of too easy vs. too hard clearly indicates that scenario designers should try to err on the side of "smaller," but there are other techniques that can be used to customize the escalation of difficulty to individual players. The inclusion of informal "cheat codes" is one

way that designers tacitly acknowledge that it's very difficult to design a scenario that's not too hard or too easy for *every* player, although skipping *over* a scenario to the one after because it's proven to be too difficult to complete isn't necessarily going to help the player get much more out of the game. A better solution is optional "in between" scenarios that could be bypassed if a player won the previous scenario by a wide enough margin. The old computer game "Populous" used an accelerated version of this system. A player was typically jumped three to six "notches" forward, depending on what their score was for the last game. Once they'd reached scenarios that matched their skill level, they tended to only advance one to three scenario steps after a win.

Decisions, Decisions

While Axiom #1 addresses what can take all the fun out of a game, Axiom #2 explains how to get the fun into a game in the first place.

> **Axiom #2: A game's fun comes from the decisions a player has to make. Fun decisions are relevant, informed, challenging, and verifiable.**

I suspect, when my book on game design is finally written, that the majority of the book will be about engineering good game decisions. Again in the interests of brevity, I'll try to illuminate the key components of this axiom briefly before going on to apply it to parts of StarCraft.

Relevant

Although this can be surprisingly subtle at times, generally, it's a pretty simple criterion. Is the decision I'm currently trying to make going to have an effect on the outcome of the game? Choosing what color to play in chess is relevant; choosing which color pawn to use in Sorry is not.

The most common way to run afoul of this condition is to give a player a decision to make that they later discover wasn't relevant. If I'm trying to capture Kerrigan, and there are three distinct Zerg encampments on the map, I expect that deciding which one to attack first is relevant to the outcome. Finding out later that she wasn't actually on the map at all, and that the three encampments are pretty much identical, certainly casts doubt on the relevance of the original decision, and is definitely not much fun.

Informed

So instead, Kerrigan is definitely somewhere on this map, and if I pick the wrong encampment, I will rouse the Zerg forces and be turned into Bug Chow before I get anywhere close to her. How to choose? I could go with "Eenie, Meenie, Miney, Moe," but that's not going allow this decision to add much fun to the game. I need some kind of data that I can think about to make my choice; I need to be informed. Maybe it's a far away map *ping* during the intro to the scenario, or maybe I get a glimpse of her leaving one of the encampments ("So she must be in one of the other two!"). Maybe I can run a sensor scan on a couple of them, looking for likely aircraft or buildings that she'd be hiding in; anything that will give me some information that I can analyze while attempting to make that decision.

Challenging

Without enough information, I'm just mentally flipping a coin; the decision might be agonizing or trivial, but it's arbitrary and unsatisfying. Too *much* information can also spoil the fun by making the decision too easy. If I can see Kerrigan walk into *that* building right *there*, and can keep watching it while I build up my forces, then deciding where to attack becomes a decision that's relevant, informed . . . and really really obvious; so obvious that it hardly feels like a decision at all. Creating decisions that are challenging but not overwhelming is probably the hardest part of game design.

Part of what makes this the hard part is that, as mentioned earlier in the section about scenarios, how hard a decision should be in order to be challenging but not overwhelming is going to be radically different for different players. At some point, you have to just ensure that most decisions are falling within the same general range of difficulty, and count on players that are too weak or too good for that range to play some other game instead. However, the effective difficulty range of a decision can be greatly expanded by *scaling the size of the consequences inversely with the quantity of information used to make the decision*. I know, another pithy little saying that needs some explanation.

For example, if the colony Kerrigan's hiding out is surrounded by sunken colonies, and it's early in the game before the aerial units are available, deciding to use tanks vs. marines is a fairly easy decision. A complete beginner might go for the units right at hand, the marines, and regret it. If the player includes just a little bit more information into their decision (the capabilities of tanks), the consequences are very significant, and that little bit of information is easy to learn. Another way to say this would be that the "low-hanging informational fruit" will have the biggest effect on the results of the decision.

A better player will give some thought to exactly how *many* tanks they'll need before attacking. Using the wrong unit has much bigger consequences than attacking with barely enough or way too many of the right unit. In StarCraft, "how many" is also frequently connected to "how soon." With StarCraft, "too few" is usually a much bigger problem than "too late," although getting the timing wrong can still make a big difference, especially in how many units you have when you're done. Deciding the quantity needed means analyzing quite a bit more information than merely choosing which units to deploy, but it won't make as big a difference to the final outcome.

An even better player will pull in even more information, and consider the positioning of the tanks. Maybe they'll take advantage of some higher ground for some of them, or move them in with a leapfrog movement. Beyond that, they might plan an attack that includes SCVs and medics as support crew, repairing the tanks on the fly so the tanks can move in even more aggressively, or move in sooner but fewer. Each increase in the amount of information used to make the decision improves the results, but by a smaller amount.

Each of the players is using as much information as they can to make their decision. Although the same information is available to all of them, the beginners don't necessarily understand the relevance of some of it, so they'll filter it out. They'll use the obvious information, which, in this case, is also the most important information. Really good players will be using much more information, and much more indirect and subtle

information, and be able to fine-tune the results. Each player is adjusting the challenge of the decision to their own skill level; they'll each use as much info as they can handle. Because the 'easy' information is also the most important information, the beginners will make the right choice most of the time. Because the 'hard' information still matters when there's only a thin line between success and failure, advanced players will see the benefit and reward from using all the information they can get.

Another consequence of having decisions with this pyramid of information relevance is that when a beginner fails to make the right choice, they'll usually fail by a small margin. The information they *did* use will get them close. A "near-miss" will encourage them to try just a little bit harder; if they just get a little bit better, then they'll get it right.

Let's consider some counter examples. If the obvious information is overwhelmingly relevant, then once you've gotten to the point that you can process that much info, coming to the right conclusion is easy, which is to say, "not challenging." Which is to say, not fun. On the other hand, if there's an enormous mass of information, and it's all equally important to finding the successful solution, then even players who can process *most* of the data will still have failures as bad as they did when they first started. They might be getting better every time they try, but if their success rate seems stuck at "no better than blind guessing," they'll probably get discouraged and quit trying before they master this problem.

As it happens, most decisions, both in games and in life, tend to be ones where the quantity of information used to make the decision will affect the consequences, but as the amount of info goes up, the effect of each new piece is less than the one before. However, *understanding* that this is the case means being able to deliberately engineer a game to provide as many decisions as possible that will be as challenging as possible to as wide a range of players as possible. As you'll see in a later section, it is this specific component of decision engineering that underlies the little 'tweaks' that could have been made to StarCraft to make it an even better game.

Verifiable

The last component isn't very hard, but is frequently overlooked. Once I've made a decision, how do I know if it was the right one? If I attack the middle encampment, and Kerrigan's not there, then I've verified that I chose poorly. If I drop a nuclear bomb right on the edge of visible territory where some scourge are hovering, I might never know if they were just the edge of a huge cloud that I obliterated, or just some stragglers hanging out over otherwise empty terrain. Even if I win the game, if I don't know *why* I won, then a lot of the fun of winning is replaced by frustration. Fortunately, StarCraft tends to have very few decisions that aren't at least partially verifiable.

Scanners, Mark II

With Axiom #2 in mind, let's take a closer look at the Terran Scanner. It's the gizmo that most effectively compromises the fog of war, so I thought I'd see if it couldn't be modified to reduce its ability to run afoul of Axiom #1 while improving its support of Axiom #2 at the same time.

I started by considering the ramifications of giving the scanners some kind of maximum range. A max range of 128 squares, for instance, would allow a scanner to scan an entire 256 by 256 map only if it were positioned dead center. This immediately provides some new interesting decisions. Before, where you built a scanning tower didn't matter. With a range limit, there's a reason to put them closer to the front, or even to fly a command center to some fairly inaccessible plateau in order to get a good site for a scanner. Conversely, the enemy has to decide how much of a threat a forward-positioned scanning tower represents, and might choose to target it specifically. But we can do much better than this.

One of the distinctive characteristics of a lot of the game design of StarCraft is the wide use of quantized effects. The scanner, for instance, has a clearly defined diameter. Anything inside the circle, you see. Anything outside, you don't. If your flyer is 7 spaces from a missile tower, then it hits you. If you're 8 away, it can't. The very title of this white paper was chosen because of the myriad "hard edges" in StarCraft. Replacing some of the binary on/off quantization with linear scaled effects adds some rich second-order layers of information, making some previously uninteresting decisions (to reasonably good players) very interesting indeed.

Thus, my next approach was to limit the scanner through some form of linear decay. There are a number of options for making the scanner decay from "100% useful" to 0% besides having it be 100% at 127 spaces and 0% at 128. One possibility would be to have the precision of the placement get worse as the epicenter of the scan became further away. I tend to suspect, though, that clicking right *here* and having the scan instead be centered *there* could prove very frustrating to players, however much it might reflect real-world behavior. Also, "getting lucky" with your scan isn't really consistent with the overall feel of StarCraft, in part because of all the other quantized behaviors in the game.

Another option would be if the scan results became more "blurry" or "fuzzy" with distance. However, there aren't any existing mechanisms to support that kind of partial ambiguity, and adding it would be pretty complex.

Let's look at what happens to game play if the *radius* of the scan decreases with distance. The current scan has about a 10 space radius. Imagine a Mark II version that has a scan radius of 15 when the epicenter of the scan is the com tower itself, and drops linearly to a radius of 1 at 192 spaces away.

Presto! Scanning just became much more interesting. Mark II's are more dangerous to attackers because they cover more territory when scanning nearby, so an advanced Terran player has to consider if that benefit means they should build com towers in more of their bases, or even build extra command centers in order to have towers in more bases. When choosing where to go 'fishing' in dark areas of the map, the more distant areas might be more important, but it's harder to find your enemy if the size of your view area is smaller. It becomes relevant *which* tower is used to scan which areas. An attacker might choose to change which part of a base to attack based on how far from that edge the com tower is located.

None of these issues radically effect how a beginner or even an intermediate player would use communication towers, but the advanced players have more information to consider in order to use the scan as effec-

tively as possible. A game design difference that could be implemented in just a minimal amount of extra code, and wouldn't have increased the computing requirements of the original game to any significant degree, would have notably increased the strategic options of advanced players, making using or dealing with communication towers more challenging. More fun.

Mine Over Matter

Replacing quantized with linear effects in the scanner is fairly interesting. Replacing quantized effects in the behavior of resources is ground-shaking.

Currently, the default characteristics are more or less as follows: a resource site will have eight to ten beds of crystals, each bed containing 1500 points. There's one vespene geyser, containing 8000 points of gas. Each bed can be worked by one worker unit at a time, and that worker unit takes about 5 seconds to extract 8 points from a crystal bed or about 2 seconds from a geyser (on a speed setting of "5", or one notch faster than the middle). These characteristics don't change until the remaining points in the resource reach zero. At that time, the output from the crystal bed goes to zero, and that from a geyser goes to 2.

One mildly interesting decision under the current conditions is how many workers to build. Let's say that after mining, a worker takes eight seconds to travel to the drop-off point and back. Up to a certain point, the more workers you have, the faster the resources come in, until you get close to saturation. Once there are enough workers to have one working every bed plus enough extra to include workers moving to or from the dropoff facility, there's nothing to be gained by adding any more workers. With the numbers used above, that's about 20 workers. At the beginning of the game, resources spent building workers means resources not spent building fighters, but since workers are usually pretty inexpensive, deciding when to build more workers, and when you have enough, is not generally an especially interesting decision for a good player. Especially with resource clusters that are vulnerable to attack, getting the resources converted from points in the ground to points in storage as fast as possible is pretty much a no-brainer, and once the crystals are gone, it's very rare that it's worth maintaining the base just for the two points per trip from a depleted geyser.

Even within the current conditions, it's possible to add some interesting complexity to resource clusters. Although I don't recall seeing this done in any of the 'official' scenarios, when I build scenarios by hand, I sometimes create resource beds where the points-per-bed numbers vary widely. If a cluster contains crystal beds with point values of 300, 600, 1000, 1200, 1600, 2500, 3500, and 5000, then the point at which it's no longer worth it to stick around becomes much less clear, and the decision of when to abandon this base becomes much more interesting (i.e., fun). At first, all eight beds are funneling crystals into storage. With enough workers to max out the harvest rate, the 300 point bed is gone in about 3 minutes, and all the beds with 1200 or less have been completely consumed in a little over 12 minutes. The production rate was 768 points per minute at first, but after 12 minutes, it's dropped to half that amount. But there's still 7,800 points of crystals to be mined! If all the beds had been the default 1,500, then at this point, there would be only 2,784 points remaining, and the bed will be completely consumed in less than 4 more minutes. The

staggered beds won't even be down to 2,700 points for another 15+ minutes. With the staggered beds, there's a lot more still waiting to be mined, but because the output is lower, it may not be worth trying to defend it from attack. In fact, although the staggered beds have 15,700 total points of crystal, and the 'traditional' beds had only 12,000, because it will take so much longer to extract the resources, it might not be worth trying to mine them in the first place. Whether to go there, and when to leave, becomes an interesting, fun decision.

These decisions become even richer if productivity decay is built into the beds as an inherent property, which can be done in two different ways. As a bed's total point value goes down, the number of points a worker gets with every trip can get smaller, or the amount of time the worker spends mining before leaving can grow longer. I think the second option is more interesting because of its effect on the optimum worker body count. Assume that a "new" crystal bed requires a worker to take three seconds to extract 8 points of crystal. If the trip time is still eight seconds, then eight crystal beds would support up to 30 workers; 3⅔ per bed. That's 50% higher than the initial extraction rate of the current conditions, and a 'strip-and-go' attack on an open resource cluster becomes a much more viable strategy than is currently the case.

The downside is that in just a minute or so, the maximum number of workers that can effectively mine those resources starts to drop. The extra workers are now just bouncing around and getting in the way. The fewer workers are assigned in the first place, the longer they'll be working with maximum efficiency, but the longer the whole operation is vulnerable to disruption. So how many is the right number? It depends! An interesting decision!

And what about resource beds being mined by the enemy? Currently, as long as the crystal beds are still mostly present, it's worth it to try to kill off the workers and slow the output, if not to take over the resources entirely. With linear production output decay, it becomes more important to know when the enemy first moves on a resource cluster, and how long they've been there. The "zergling swarm" attack is pretty darn boring, and I've been very happy, based on some of the design decisions made for WarCraft III, to see that Blizzard apparently thinks so, too. With some thoughtful placement of linear decay resources in a scenario, a peculiar new variant of that attack, the "locust swarm" of defenseless but voracious *miners* becomes a radical new strategy, with (unlike the zergling swarm) a number of interesting options for counter-strategies as well.

Grinding to a Halt

I have played a few games of StarCraft where it became clear that I would eventually be crushed like a bug, but the computer opponent would probably make me wait quite a while before finally putting me out of my misery, especially if I kept putting up a fight. These experiences are pretty rare, though, and (to me, at least) don't represent a design flaw serious enough to need fixing.

A slightly more common and much more annoying outcome is the resource-starved stalemate. We're both spread across the map, the computer's got no mobile units left except for some workers huddled around

vespene geysers, and I've got two hydralisks, three zerglings, and a mutalisk. I can either surrender, and tell myself that I really actually won, and the computer was just too stubborn to admit it, or spend hours running my meager forces back and forth, trying to take out buildings one at a time without losing any of my mobile units, waiting between each attempt for them to heal back up to full strength.

With linear decay resources, the stalemate problem almost disappears. It's nearly impossible to have a map completely devoid of resources. Resource clusters that were abandoned earlier in the game because it wasn't worth it to mine there become valuable once again if the end game turns out to have two almost perfectly evenly matched forces. Even a slight difference in the rate at which resources continue to arrive can fairly quickly turn a stalemate into a victory. If I can get just enough crystals to turn that mutalisk into a guardian, I'm golden; the hydralisks can defend it from the unit or two that the computer can field before the guardian has destroyed enough workers to cut the computer's resource input to zero, leaving me just the mopping-up.

Conclusion: Tip of the Iceberg

There are a number of other units and properties that become more interesting and more fun if steps become ramps: if quantized behavior is replaced with linear decay. Missile Turret and Goliath missile accuracy, the range at which a Hydralisk starts chasing after an enemy unit, or the radius and duration of a Defiler's plague attack (the further away the target, the wider it spreads, but the quicker it goes away), for example.

Another tool for rounding off the edges of steps is noise. AI strategies, in particular, would be significantly more challenging, and more effective, if the AIs were "less perfect;" if there was a small amount of randomness mixed in to their decisions. A number of unit and weapon behaviors would also give advanced players more to worry about (in a good way) if they had small amounts of noise added to the algorithms.

I'd also originally intended to cover the myriad end-game effects that adding a small unit maintenance cost would have on the game. WarCraft III added a maintenance cost mechanism, but in a way that made for a fairly uninteresting decision: bring your mobile unit strength to 39 or 79, mine as fast as you can, then build the rest of the units up to the maximum 100 points. A system with both units and buildings incurring a maintenance cost but keeping the cost much lower, deducting it from banked resources instead of the production stream, and creating linear decay/degradation effects in units that weren't being maintained, would let beginners safely ignore this effect, but, especially in a longer-running game with seriously restricted resources, would become a very important part of advanced players' planning, especially if "decommission" is added to the current option of "destroy."

Unfortunately, while adding such a system should still represent a pretty modest amount of extra coding, the ramifications to the advanced player are complex enough that it would probably more than double the length of this white paper. Although they're just the tip of the iceberg, hopefully the provided examples

illustrate how much more fun is still waiting to be unleashed in the already extraordinarily good fundamental design of the StarCraft/WarCraft games.

About the Author

Dave Howell was Employee #7 at Wizards of the Coast, and was a frequent early playtester of their games, as well as being the Production Manager for Magic: the Gathering, head of the Book Publishing division, and creator and team lead of the online customer support system. He's also been a frequent "Guinea Pig"/playtester and occasional co-designer for Cheapass Games, and has presented "Golden Guidelines for Game Design" at three World Science Fiction Conventions and at GameStorm in Portland, Oregon (twice).