Just One More Game...

Martin C. Martin*
martin@martincmartin.com

Game designers make many decisions, from basic gameplay mechanics to the details of a particular level. They're all driven by the same goal: to make the game enjoyable. But what makes a game enjoyable? Some things are shared with movies and television. Beautifully rendered landscapes can get us to stop playing and just marvel at the surroundings. A moving story can spur us on and, like a book that we can't put down, we have to know what happens next. Much has been written about these, at least as far back as Aristotle's *Poetics*. But games add something new to the mix, interaction. What is it about interaction that makes a game enjoyable?

Common answers include shaping the unfolding story, the thrill of winning, and simply being able to affect the game's world. But those only hold our interest for so long. "Choose your own adventure" books may allow you to shape the unfolding story, but after a few such choices they no longer hold our interest. Tic-tac-toe becomes boring once we can always play a perfect game. And knocking over tables or breaking crates certainly allows you to affect the world, but after the novelty wears off, there doesn't seem to be much point.

On the other hand, there's one aspect that stays interesting much longer: figuring out how to do better and better. This article explores this aspect of what hooks people on games.

"Now I've Got It, Just One More Game…"

Let's start by looking at a common example in detail. Someone new to first person shooters might first spend time learning how to move and shoot. Because aiming is tricky, they'll often stand still to shoot. But eventually they get frustrated and want to shoot more often *and* dodge more often. So they practice that, and as they get better they think it might give them an edge over other players. At least they won't loose as quickly. They feel that they're starting to get the hang of the game, however tentatively.

That dynamic of continually improving a strategy occurs in game after game. When faced with choices, players don't simply choose at random. Instead, they use a strategy, often unconsciously. Each time they apply it, they see where it succeeds or fails, and they realize some way to improve. And improving gives them an emotional reward, they think "Oh now I know how to do better, just one more game..." Learning the strategy helps make games fun, because it allows the player to become more and more competent in the world. All these goals they're trying to achieve, all these barriers to their progress, are a lot less troublesome once they've got a plan that they think will work.

As they spend more time in the first person shooter, they notice other things. They may be constantly killed when turning corners, and then learn to strafe around them. They may notice that certain spots provide a lot of cover while giving views of a large area, and learn to snipe. They may realize that going in with a teammate really increases your chance of survival, and start exploring team tactics.

Whether the player learns a conscious strategy or hand-eye coordination, they're able to do more in the world, and that feels rewarding. So from now on, I'll use the word "approach" as a general term that includes both hand-eye coordination and conscious, deliberate decision-making.

What's The Approach, and How Do We Improve It?

Looking at existing games in terms of learning an approach explains a lot about why certain genres have withstood the test of time, while others have largely faded. Choose your own adventure books are fun until you realize there's really no way to know which choice is the right one. If you're trapped in underground caves and trying to reach the surface, choosing the route that leads up is as likely to get you killed as the route that leads down. Once the reader realizes that, they just choose options at random, backing up when they hit a dead end.

Tic-tac-toe, on the other hand, is fun while you're learning but before you're perfect at it. Picture someone who has never played the game before, just starting to learn it. The goal is to get three in a row before the other player, so suppose they treat it like a race. They work on getting three in a row as fast as they can, ignoring the other player.

After a couple losses they'll realize they need to block their opponent. And then something interesting happens: in gaining that insight, they all of a sudden realize they can play the game better, which gets them excited. They want to play "just one more game" to try out their new strategy.

In applying the strategy, they quickly discover that they have to choose: when should they block their opponent's lines, and when should they build up their own? Perhaps it takes a game or two to realize, that when their opponent has two in a row, they need to block or they'll loose. Each time they gain some new insight, they want to try it out, realizing that they'll do better in the next game. It's only once they've mastered the game and get no more thrill from improving that the game becomes boring.

What advice does this suggest to game designers? While on this path from the newbie to the elite, the player always has an approach and they're looking for ways to improve it. So our first principle is to ask "what approaches will the player try?" and "after trying them for a little while, how will the player know what to improve?" These two questions are central to game design.

Asking them about existing styles of game play is revealing. Suppose the player finds a safe, and the only thing the player can do is type in a combination and find out whether that's the correct combination. If nothing else in the game relates to the safe, the only viable strategy is to try every possible combination. There's no way to learn from the unsuccessful combinations, no chance to say, "Wait, now I can do it better, just one more game!" After a few such "guess the combination" puzzles, they no longer hold much charm. The player is reduced to following a simple algorithm, just turning the crank.

That's our second principle: "guess the combination" isn't much fun. That's well known, but what is less appreciated is that other forms of interaction amount to the same thing. For example, designers often make a game more challenging by throwing more at a player earlier on. If they don't think of what this does to the strategy, they can turn a fun game into a combination puzzle without realizing it.

Imagine a strategy game where the player must defend three bases. Imagine that the scripted AI character "rushes" the player, sending a wave of units to the player's west most base, then a few minutes later a wave to the east most base, and a few minutes after that to the player's north most base. Suppose that these happen so early that the only way to survive this onslaught is for the player to move all their units to whatever base is about to be attacked.

The player may start by trying to build equal defenses at all three bases, but she will keep losing the game after the first battle. Only when she puts all her units in the west most base will she survive the first wave to fight the second. The only strategy that succeeds is to play the game until you're attacked, reload from a saved game, then move all units to the correct base just before the attack.

The problem is that the knowledge needed to win—the timing and location of the enemy attacks—is specific to a map. It just doesn't transfer to any other map. The only viable strategy is trial and error, just like a "guess the combination" puzzle.

What's worse is that, even if later levels allow more interesting approaches, the player will stick with whatever's worked for them, unless it clearly won't work at all. This brings up a third principle. Once the player has figured out an approach that has some success, they keep applying it to future challenges, rather than abandoning it for something completely new. For the "get attacked/reload/prepare for the attack" approach, there's not much to modify.

You can see this in single player RTS campaigns when a new unit is introduced. Suppose it's an artillery unit, which can take out units from far away. If it simply shows up in the menus without much fanfare, players will stick to the old units at first. If the map can be solved that way but is extremely difficult, they'll just think the game got hard and have no motivation to change. But if there's a target on the other side of a river that must be destroyed to lower a bridge, and none of the old units can hit it, then it's clearly impossible with just the old units. The player will quickly realize that the new unit is the only thing left to try.

The same issues also apply to changing difficulty levels. Game designers often create easier difficulties by simply weakening some aspect of the enemies or strengthening some aspect of the player. If they do this without thinking about the approach, they can end up guiding the player to the wrong approach, making the harder difficulty almost impossible. For example, if an easier difficulty of an action game makes enemies very slow, the player may learn to just run past them. If that strategy only stops being viable at higher difficulties, where the enemies are using advanced tactics, the player is forced to learn the basics of combat in a very difficult setting. That can be very discouraging.

Players who start on an easier mode and settle on an approach will apply it when they first try harder modes. If the approach gives a little success, and "trying harder" works a little better, they may never abandon it. They might think they're just bad at the game, and give up, discouraged.

A forth principle is that it's not good enough for there simply to be a viable approach, the player also needs a way to discover it. Realistic combat flight simulators are an example where your average gamer can't figure out how to improve. Those who just jump in without reading the manual start by simply turning toward the enemy. When the enemy is to their left, they turn left; when they're to their right, they turn right. But planes move much faster than they turn, and at best you get two planes chasing each other's tails, flying in a circle. To succeed, players need to study the manual and practice many classic maneuvers, such as the Immleman or the Split-S, maneuvers that are far from obvious. While some people are that motivated, most aren't. No wonder these are niche games for military lovers.

For complex simulation games, the causes of your current problems can be far from obvious, so the game must provide some feedback. Why isn't anyone moving into your new SimCity neighborhood? You can check the police coverage, traffic jams, and many other reports to find out. Without those, the player would have no idea how the world works, and the game's responses would seem random. With those, the player can learn "ah, I need to separate the

industrial and residential zones to cut down on pollution. Now I've got it! Just one more game..."

For games like SimCity or Civilization with complicated dynamics, there's often a temptation to provide advisors who suggest what to do next. For example, a military advisor might lean toward using force, while a diplomat would favor political pressure. While this greatly reduces the number of actions for the player to consider, how will the player choose which advisor to listen to? If the game is too complex to understand in the first place, then simply telling the player some things he or she can do doesn't help their understanding. Perhaps this is why information displays have been central to these games from the beginning, while advisors have always been an experimental, secondary feature.

More Existing Gameplay

With these principles in hand, this section looks at existing games from our new point of view. These design principles explain a lot about the success or failure of many types of game play. For example, what is the skill or strategy learned in an adventure game? Let's start with the dialog. If the dialog is restricted to selecting options from a list, most players simply choose each item in turn until they find a sequence that advances the story. Any idea about which option is best simply suggests which option to try first. But there's no real penalty for choosing a couple wrong answers before getting the right one.

What's more, any reasonable approach would be wrong as often as right. Each character has their own personality, so while asking the nervous pirate about a recent failure may cause him to break down, asking the same thing of the bored bartender will have no effect. And will breaking down make the character more likely to tell you something, or reduce them to a blubbering pile, unable to speak? The rules are different for every character, so that finding the way through one tree doesn't help you at all with the next dialog. On top of that, designers of decide the make the game more challenging by hiding the correct answer in a non-obvious choice. Dialog trees simply become a case of trying all possibilities, making them a fancy kind of combination puzzle. "More challenging" doesn't mean "more fun."

For the adventure game as a whole, outside of dialogs, the situation isn't much better. Sometimes what to do next is obvious, such as giving the membership card you just found to the clerk who asks for it. Other times it comes with a little thought. But there are always puzzles that are so out of the blue, you would never think of them on your own. Maybe it's an action that you're just not used to associating with the object, such as using a wet towel to "flick" someone. Or maybe it's something that wouldn't work in the real world, such as drawing a realistic looking mustache on a passport photo with a magic marker. What can you do when you're at an impasse like that? Your only option is to try every object on every other object. Which reduces it to a combination puzzle.

Multi-user dungeons (MUDs) and MMORPGs are at the other end of the spectrum. The first few levels are spent getting the hang of the interface and basic game mechanics. But as soon as you get used to those, you've gained enough levels that the monsters you've been fighting don't give any experience points, and the next harder monsters are too hard, so you have to form a party. This brings a whole new set of social skills to learn. And on it goes, each new level bringing new abilities, spells, monsters and items, so that most users spend the entire game in the learning curve. And these games are known for being some of the most addictive out there.

Perhaps the single issue that designers feel most passionately about is whether games should allow "save anywhere" or have save points. Many feel strongly one way, whereas others feel just

as strongly the other. From most points of view, this is a minor issue: it doesn't make the world more or less interactive, it doesn't change the puzzles, units, or core gameplay. Yet it changes the process of learning an approach, sometimes drastically. With "save anywhere," you can try each challenge multiple times, saving after the first time you solve it, even if it was by chance. Save points, on the other hand, force a player to play most of the game over and over, until they can succeed most times. Either way allows the player to learn ever greater approaches, but how well they must be learned, and therefore how difficult they must be, are very different in the two cases.

Practical Steps

"Because of too much involvement, [game designers] are unable to objectively comprehend how the actual players would feel when they play the game for the first time."

- Shigeru Miyamoto, Nintendo, creator of *Donkey Kong, Mario Bros.,* and *Zelda.*

Looking at games in terms of the player's approach, the skill or strategy they build up over time, suggests a few practical steps. Intended approaches can and should be up front in the design document. Doing so helps focus designers on this core dynamic, helping them think of design decisions in terms of "what's the approach?" and "how does the player know what to improve?"

Players are slippery beasts, however, and don't always use the approach you so carefully laid out for them. As Mr. Miyamoto points out, game designers are often far too close to the game to see how others relate to it. Instead, we can take advantage of a growing trend and use play testing not simply to find bugs, but as an integral part the design process.

In developing Half-Life, Valve took outside volunteers—not QA people but regular gamers, who hadn't played earlier builds—and watched them play the game. Other than starting the game for them, the observers were not allowed to say or do anything. As Ken Birdwell says, "Nothing is quite so humbling as being forced to watch in silence as some poor play-tester stumbles around your level for 20 minutes, unable to figure out the 'obvious' answer that you now realize is completely arbitrary and impossible to figure out."

The Sims 2 team called this "Kleenex Testing:" "The key to fresh feedback is using each tester only once: just like Kleenex. The most important part of Kleenex testing is finding people who can play a game with someone looking over their shoulders and while voicing the thoughts that go through their heads."

What sort of thoughts? Here's how one player might start with Space Invaders. This also brings together a lot of the themes I've been talking about, and provides a concrete example of learning an approach:

> *The player started by shooting blindly into the crowd. Most shots hit, but a few went straight between columns of invaders. That reduced her firing rate, so she started paying a little more attention to where she was aiming, trying to line up with a column.*
>
> *Next she hit on the idea of taking out an entire column at a time, mainly because aligning with a column is tricky, so she wanted to milk each alignment for everything it was worth. She didn't often get the entire column in one go, because the column moves slowly, so she moved a little when the column was half done. She also tried to get back to the column after dodging incoming missiles. That taught her some fine motor skills for positioning her base. Before long she was leading targets and finding the trade off between dodging and shooting.*

As the session went on she discovered that if she shot a hole through the middle of a bunker, she could take down invaders without having to dodge as much. As the invaders got closer and closer to the bottom of the screen, she realized she should take out the ones on the bottom first. Also, taking out ones on the edges gave each row further to travel, slowing their descent even more. When she was down to the last few invaders, she learned to get out of the way while her shot traveled up the screen and scoot back when it was almost gone.

After getting a handle on all the skills above, those infrequent fast missiles from the invaders became a pain. Because they're too fast to dodge, she looked for cover whenever she wasn't firing.

Space Invaders has some other interesting properties too. There are a couple design decisions that seem superficially minor, but that have a big effect on how someone learns. By making ammunition scarce, the player will naturally focus on making each shot count. That encourages them to think about aiming and leading the target, skills that are essential on later levels. If faster shooting were allowed, the player might focus on shooting as fast as possible, even if that meant they couldn't shoot as accurately. When they got to a later level where accuracy was required, they might get stuck or at least take a long time to learn.

In fact, Space Invaders does limited ammunition one better. Because there can only be one shot on screen at a time, the firing rate is slower when you miss than when you hit. That means sloppy aiming slows everything down, focusing you on where every shot goes. When your accuracy improves, the game picks up pace.

Also, having the invaders only shoot sparingly in early levels focuses the player on shooting columns first, and dodging second. This gives players a chance to get the hang of the small movements of the base needed to track a column, which does a lot to encourage better dexterity.

Realism

How a new feature affects game play isn't simply how it changes the rules; it's how it changes the approaches that players try, and their understanding of how to improve them.

Take realism in games. Discussions of game AI, such as the Game Developer's Conference roundtables, often contain a mention that AI should be fun and not smart. But how do you translate this into practice? In a shooter, is a well coordinated enemy more fun than mindless hordes? In a strategy game, should units have morale?

Without any framework in which to think about these questions, the discussion ends up on realism. Imagine an RTS that simulates the physics of missile flights and evasive maneuvers to determine whether the missiles hit. If the most successful approach turns out to involve micromanaging the motion of each of your units, then the winner might simply be whichever player can click fastest. And let's face it, that's not much fun.

In shooters there's talk of learning, shell shock, emotions such as stress, panic, morale, even personality types with an accurate distribution of traits. Realism for its own sake can interact with gameplay in essentially arbitrary ways, making the game more annoying or inscrutable as often as more fun. How can a focus on learning an approach help? As an example, take alliances between players and the AI in strategy games. How should the alliance logic work on the AI side? First, it must be understandable. We've all seen games that act schizophrenic, offering a peace treaty one moment and demanding a tribute the next. With alliances so unpredictable, the player simply can't figure out how to manage the alliances, so they shy away from using them at all.

If the politics of alliances isn't the main focus of the gameplay, it's probably best to keep it simple, and keep the logic transparent. Attitudes could be based on relative military strength of the two sides, which will mostly change slowly, removing schizophrenia. If joint military campaigns are too complicated, simple non-aggression pacts might do. If the player gets lots of warning before an alliance is broken, with an explanation as to why, the system will be transparent. Then the main question becomes: how can the player use an alliance in an approach? Even if it's as simple as keeping a second front at peace for a few rounds, this can become an invaluable part of a game.

Every so often a simulation game comes out that gives you a bunch of basic building blocks, and an open ended environment to see what combinations you can make. For example, a dancing game might map the four limbs to four buttons, and simply give you a way to move them however you see fit. That lets you play around and create whatever dance moves you like. But without a goal to work toward and some constraints along the way, there's nothing to learn. I call these games MacPaint style games, after the early digital paint program. MacPaint was too simple to be used for serious work, much the way that a dancing sim couldn't be used to choreograph real dances. After noodling around for a few minutes, most people find the boring. But adding a little structure can make it a lot of fun. Games like Dance Dance Revolution or Tony Hawk don't allow just any combination, and force you to get into the music or the motion in order to progress. Skateborading games like Tony Hawk are still open ended and are often played among friends, with everyone showing off moves to each other. Far more so than with paint programs.

Summing Up

When people play a game, there are reasons behind their actions. In a shooter, if they don't seem to be hitting the enemy, they may get closer to it. If players take too much damage, they may hang back so they're harder to hit. In a role playing game they may focus on improving a spell caster to unlock the more powerful spells, or focus on the fighter that's been the most valuable party member so far. They always have an approach, and by applying it they can discover its bigger problems, along with some ways to improve it. They feel rewarded and competent when these approaches succeed, and change them when they don't.

Designers often think in terms of the final approach, the final skill or strategy a player will need. They focus on things like game balancing or making sure there is no one approach that trumps all the others. However, it's learning the approach that's the fun part. Tic-tac-toe becomes boring once you can win every time. Playing a game is a journey to a final approach, and the path is at least as important as the destination.

People who didn't grow up playing games sometimes decry them for their violence or "dumbing down" our children. But really, games are about learning. They're all about improving your competency through applying yourself and reflecting, through hours and hours of practice. Even if the thing that's learnt is trivial, games are about gaining confidence and a feeling of being good at something. That's what really sets games apart from TV, movies, and books. How can we get non-gamers—the general public, the politicians—to see the benefits of games, as well as the problems?

For More Information:

Birdwell, Ken. *The Cabal: Valve's Design Process for Creating Half-Life.* Game Developer Magazine, Dec. 1999, pp. 40-50.

Koster, Ralph. *Theory of Fun for Game Design.* Paraglyph Press, 2004.
"EA Sports Fight Night 2004" postmortem in Game Developer, June/July 2004.

Tsunada, Kudo. *Postmortem: The Control System of* EA Sports Fight Night 2004. Game Developer Magazine, June/July 2004.

Bradshaw, Lucy *et al. Postmortem: Avoiding Sequelitis in* The Sims 2. Game Developer Magazine, January 2005. Especially "What Went Right" 3, Kleenex Testing and "What Went Wrong" 1, Noisy Feedback.

[Maybe Also:]

- **What Video Games Have to Teach Us About Learning and Literacy** by James Paul Gee.
- Noah Falstein's chapter in an upcoming book on a similar topic.