THE 4 Fs OF GAME DESIGN

FAIL FASTER, AND FOLLOW THE FUN

In 2001, I attended a workshop at GDC taught by one Marc "MAHK" LeBlanc. Over the course of those two days, I would be exposed to a great deal of development wisdom (and if you haven't attended one of Marc's sessions, what the hell have you been doing? Go, people! Go!), but there were five words in particular that Marc shoved down our collective throats that have completely altered the course of my career.

FAIL FASTER. FOLLOW THE FUN.

» Those words are a complete methodology for how to not be eaten by snakes, and perhaps succeed, when pushing into the wilderness of the New.

Marc repeated the phrase endlessly, until there was absolutely no chance that anyone in that room would ever forget them. Here, I'm going to continue the cycle of abuse, and do my damnedest to inflict the same brain wound on as many readers of this magazine as I can. It worked on me, so maybe it will work on you.

So, full credit goes to Marc LeBlanc. Also, fail faster, people. Fail faster! And, follow the goddamn fun.

SCOPE: WHEN TO APPLY THE 4F METHOD

» Before the brain wounding, though, let's talk about limits. This "4F" method does not necessarily apply to everything.

What we will be discussing is most useful when you are exploring the New. Though failing faster does have some application when making small iterations on a known design, you can actually succeed without doing it if you have a strong template in front of you (heresy!).

HOWEVER.

» If you are making something new from scratch that you have no reasonable guide for, then your situation is somewhat more dire. In the end, all of the success you will have in "finding the functional New" will come from the moments when you first fail (and you will fail; in fact, that's the next step), quickly recover, and then follow what shards of fun you may have managed to discover. Everything else will be a waste of time and resources.

FAIL EARLY, FAIL OFTEN

» We game developers hate this word, generally.

Oh, sure, if we are sitting around a table sipping the mind-altering beverage of our choice amongst our industry comrades, we will happily extol the virtues of rapid prototyping, praise Valve for its highly iterative approach, and shake our heads in horror at the tales of They Who Just Do Not Get It.

Go ahead, try it for yourself: Walk up to your scrum leader or manager or whatever and tell her that you have just failed. Big time. You built something that totally sucks.

Did she smile? Pat you on the back? Congratulate you?

If so, then awesome! You can stop reading now if you want. (In fact, you are probably on my team. Get back to work!)

Myself, I've had to fight mightily to make Marc's words a reality in my day-to-day work. Our highly pressurized game development culture has precious little room for processes that make things that aren't fun. Solutions, people! We need solutions! Winter is coming!

But... we do know about this one, don't we? The first time a new design is put into play, there is a (scientifically verified) 99.97% chance that it will suck. Anyone building new features must account for this in their process. This doesn't happen because we're dumb. It happens because designing fun games is a hard thing to do.

GAME DESIGN: NOT ROCKET SCIENCE

» I had an executive producer back in ye olden days who was fond of the phrase "It's not rocket science, people!" When we would hit something seriously tricky, and were up against the wall, he would say this, and we would all laugh, reminded that what we were doing wasn't so hard after all (I mean, rocket science, right?! Come on!).

I've decided since then that he was completely ass-over-teakettle wrong. Now, he might have been right way back in the 1990s, when video games were built out of sticks and glue, but in the current era that sentiment is so horribly outdated as to be dangerous.

I propose to you that the underlying technology that drives most interactive systems is at the very least as complicated as the systems that put rockets into space. On top of that, the goal of most of this technology is to elicit human emotion, a task that has consistently eluded casual reproduction for thousands of years (and still defies most of our attempts to codify it).

Designing a fun game is hard. Much harder, in fact, than rocket science. (When John Carmack needs a relaxing break from game development, you know what he

RED STEEL 2.

ILLUSTRATION BY JUAN RAMIREZ

does? Yep, rocket science.) It's hard enough, in fact, that exactly zero game developers can get it right the first time, every time. None. So. Embrace failure. Because it is inevitable. So really, you might as well.

BE WILLING TO FAIL PUBLICLY

» We know that making games is hard, and we know that rapid iteration is the cure, so why don't more teams embrace it? Simple: Very few people want to look dumb in front of their peers.

This is the heart of the matter. This, right here, is big enough to effectively be the entire problem. I'm not kidding. If you can get past this one, much of the rest of the 4F approach follows naturally.

When I started my current project, I printed a big sign that said "FAIL FASTER" and hung it over my desk. I think my team initially thought I was literally insane. From the outside, it looked like the act of someone who wanted to end his career. "You're... not actually going to leave that there, are you?" someone asked me, I think a tad fearful that my madness might reflect poorly on him. And reasonably so! I was nervous when I put it up there!

(Now, of course, many months later, when I am frustrated that our current review wasn't amazing, they point at that sign. "Well, at least we failed fast!" they say, lobotomizing my concerns. I love them for this.)

Anyone with a resume to protect will naturally be suspicious of being asked to fail. If you want to try to implement a successful

iteration culture on your team, you ignore the natural terror of failing publicly at your peril.

FAIL FAST II: FAIL FASTER

» The bright side of failure, when accompanied by fair, critical evaluation as to the reasons behind it, is that it is the most certain road toward success with the New. But how many iterations does it take? Yeah, we don't know. So, the solution is to make your failures as fast as possible.

Paper prototyping is a pure incarnation of the 4F approach. Even so, I've seen people polish the crap out of their paper prototypes before testing them, and then discovering that their laminated, be-graphicked cards are terribly unfun. "Don't polish a turd," Marc LeBlanc told us during his course. "It's a dirty job, and in the end all you have is a shiny turd." The trick to escaping this trap is to develop an instinct for the shortest-possible road to prove a design idea, and a willingness (nay, an urge!) to share your work well before it is "done."

Years ago, I was directing a team working on a shootery-thing, and we needed some new gameplay. (It was, in fact, my first real-world opportunity to try to put the Fail-Faster-Fun-Following system into play.)

I told my designers that we were going to do rapid level prototyping. I'm not sure what they thought I meant by "rapid," but when I explained that I expected them to build fully functional levels in one-day cycles, their faces turned gray. Yet we still met each day at 4 p.m., and played each other's work.

Day one was a disaster. Only two guys had levels (and I was one of those two), and none of what we had was fun. They were merely "professionally executed demonstrations of existing mechanics" that lacked love. But we were brainstorming; no criticisms or name-calling were allowed. The people who didn't bring levels to the meeting were not allowed to speak. If you wanted to share your opinion, you had to bring a level.

The next day, we had levels from everyone. Still nothing really fun, but all the designers presented their work with something more resembling a competitive spirit. My guys were not dumb (as is the case with most game developers, in my experience), and had quickly picked up on the fact that the best designs would simply win the meetings.

So far, we had been failing plenty fast. I was proud! Now we had to find the fun.

FOLLOWING THE FUN

» The "win the meeting" thing happened the next day. One of our guys got frustrated with the lack of awesome in the first set of daily levels, and designed something I can only call "Rocket-pocalypse." He had put the player up on a platform overlooking an endless enemy charge, and had assigned the action button to fire a metric ton of rockets straight down from the heavens on the field whenever the player so desired.

We played nothing else for the whole session. The team loved it, I loved it. It was really, really fun.

The fact that this mechanic had nothing to do with the actual game was irrelevant. The designer knew that he was making something for the trash bin, but it didn't matter. After all, it only took one day, we were doing research, and most importantly, it was super fun. After that, the "fun target" had been set. "Beat John's Rocket-pocalypse" was now everyone's goal. Whatever they made needed to be at least as fun as that.

This is what is meant by "following the fun." It means, in reviews and in your design planning, only allowing yourself to pursue existing implementations that rock.

Heh. There's a pretty good chance right now that you're thinking, "Well, yeah, we do that all the time. I've heard that before. What's the big deal?"

Over the years, I have worked with hundreds of devs. Again and again and again, I have encountered smart, creative, dedicated, talented people who are producing mediocre gameplay. Why?

Usually, it's because (for reasons of environment, training, experience, constraint, or whatever) they have accepted, consciously or unconsciously, a fun-quality bar that is lower than what will make the audience want to keep playing. Often, ideas like, "Yeah, it works!" or "It's pretty fun!" are the guiding criteria for success, especially when time is burning.

I call this "foolish optimism." If you played a game, and your response was, "It's pretty fun!" would you recommend it to a friend with enough passion to convince her to buy it?

There's a better criteria, and it is what drives the idea of following the fun. Think about the game you are making. Maybe you're working on a feature, or a level. When you playtest it, do you often find yourself continuing to play long after you validated the feature? Just because it's fun? Do other people do this?

If not (and if we're talking about a key feature that you need to be fun for your game to work), then you probably haven't found the fun.

Revise, redesign, or cut that feature.

This is hard to do. In fact, if it isn't hard, then that's a sign that you're not moving your quality bar high enough. Following the fun requires a commitment to actual fun.

Of course, that doesn't mean that everyone's iterations need to be perfect. Learning to see through the imperfections into the underlying fun in a demonstration is one of the most important skills a game developer can develop. (So get started on that.)

IN THE FIELD

» I used the 4F process during the production of RED STEEL 2, and the results were unanimously positive. We had the daunting problem of solving first-person melee combat (with guns) with the Wii MotionPlus, without any prior successful examples to work from. We went through whole new approaches to combat at a stunning rate, sometimes prototyping complete combat systems in a single week, learning what sucked, and moving on. Using this approach, we designed a one-of-a-kind first-person motion-control sword-and-gun combat system, and we did it in nine months. IGN called the game "one of the top titles on the Wii." I credit the team's embracing of the Fail Faster philosophy with that success. (I wish it had sold more, but I suppose you can't have everything.)

In my current project, I have a relatively small team working

on some new stuff. I have applied, almost to the letter, the same process that I described to you above: 24-hour test cycles of whatever-in-the-hell-you-can-come-up-with-as-fast-as-possible, and instilling in the team the belief that I will never crucify them for failing, that the only sin is the sin of failing to share your failures quickly enough. They have embraced it, and the speed with which we have been making progress exploring the New in our chosen category is inspiring to behold.

GAME JAM SESSIONS

» One last thing: If you've been involved in a game jam, and have made it this far in the article, you may have noticed that the Fail Faster approach could be described as a game jam with a producer.

Yes. Yes, it is. That is how I make games these days. And I hope you'll join me, because we need more and better New, now more than ever.

JASON VANDENBERGHE *is a creative director at Ubisoft, which he has to admit doesn't exactly suck. You can read his intermittent blog and various scribblings at www.darklorde.com. He can be reached by email at jason.vandenberghe@ubisoft.com.*

MARC "MAHK" LEBLANC'S *work can be digestified at www.8kindsoffun.com. And, thanks again, Marc.*