

# What it Takes to Make a Game

## Part 1: Why Being an “Ideas” Person isn’t Enough

Perhaps the most common refrain that designers get from laypersons is some iteration of, “I have this great idea. If someone offers to program it, I’ll split the profits with them!” To other average people (gamers or otherwise), this seems reasonable on the surface. After all, a game’s core foundation is just a really interesting idea!

Well... No, Terry. Ideas are a dime a dozen. Everyone has a really cool, great idea. Here, watch:

Hyper-realistic reimagining of Joust, but it’s first-person (or third person) and you’re riding on enormous dragons. The dragons have unique abilities based on the breed, and the player has unique ways of jousting depending on their weapon and/or class. 1v1 or large-scale team battles with lots of dogfighting aerial combat potential.

See? How sick would that be? And that idea literally took me thirty seconds while I was typing this. Not to put too fine a point on it, but ideas are *almost* worthless. I say “almost,” because ideas actually have an incredibly important role in the game design process – Maybe even the *most* important role. But not just any idea works, Calvin. Lemme show you what I mean (but settle in. This is gonna take a minute):

1. Core Concept – This is super valuable because you don’t have a game without it. Every good game on earth was spawned from a good idea; whether it’s a unique concept, a cool spin on an existing concept, or just an extremely streamlined version of a single mechanic that can play Atlas, hoisting the entire game up on its own.
  - a. But unfortunately, it’s also literally the easiest part of the entire process, and it’s also *the fun part* (well, one of the fun parts). If you slap five game designers in a room for a day and tell them to come up with game design concepts, they could have a hundred by the end of the day, most of them would be pretty decent (or better), and they would have had a *great time*.
2. Narrative (macro) – This is *also* super valuable since the absence of a narrative is generally very noticeable for discerning players. Though it’s not technically every game – As a rule of thumb, unless there’s a compelling reason not to have a narrative, you *should* have a narrative.
  - a. Coming up with a game’s narrative is harder, but still pretty easy among ideas for the same reason that the core concept is easy: you’re doing the fun, conceptual stuff that people are generally intuitively good at. We’ve all seen stories we like and played games we think are fun. We’re basically just emulating the traits we enjoy most. While it’s relatively intuitive and your game will probably be intelligible, the narrative still probably won’t be very good, Devin. Because your overall narrative isn’t the hard part...
3. Gameplay Narrative (micro) – ***This***. This is where I see more games fall apart in terms of quality than any other. A game with bad dialogue, confusing characters, unrealistic interactions, unhelpful cues tied in to the gameplay itself - A game can succeed without these things, but I would contend that’s limited mostly to small, self-contained gameplay loops that don’t ***need*** a

narrative since there's never any transitions from plot beat to plot beat or characters with personalities or dialogue. If you want your game to elevate past a mid-2000s Flash game, one of the single most valuable "ideas" you can contribute are things like good writing. Things that aren't intuitively picked up on by most people. The vast majority of people can't write satisfying dialogue or an interesting scene without some training in that field, Samantha. Here's what makes their ideas unique:

- a. You *need* character personality, whether it's developed implicitly through talking to another character, their body language in their surroundings, the way they carry themselves, how they treat their surrounds; or through explicit means. Characters who openly emote towards each other. Conflicts are often either centered around the characters' relationships, or often their relationships **while something is trying to kill them**.
- b. The Narrative "ideas" person is stitching together the characters from scratch. If you hand someone the knitting pattern for a character that you drew up in five minutes, it will take days, weeks, sometimes months to come back with a decent pair of knitted gloves. That's because characters aren't just, "Here's Josh. He likes Hot Topic and he's single and ready to mingle." There might be stuff to work with there, but you just put sprinkles in a bowl and then asked someone else to make ice cream on top.
- c. Narrative Designers have, in my opinion, one of the hardest "ideas" jobs in game design for one critical reason: their role is largely self-contained. If the mechanics people want help figuring out the functionality of their ideas, the programming team is there to talk shop about it- mostly on practicalities and limitations. This creates an idea feedback loop that makes everyone walk away stronger than when they went in. On the other side, Narrative Designers are coming up with the characters, their strengths, weaknesses, hopes, fears, and relationships/histories with the other characters, full character bibles that they almost always have to work on in isolation from other teams because narrative work isn't especially similar to any of the other disciplines in game design. Bouncing ideas off of other people is hard as hell unless they have a decent writing background themselves.
- d. This is compounded by the fact that there's only one thing players feel *more* than a lack of narrative: a **bad** narrative. If you deliver something that is poorly geared toward your intended audience, you've already destroyed the game. If you deliver something that you had to rush through and you miss a glaring characterization error, people will wonder how something so obvious could slip by you; or that you're a bad writer. And unlike a gameplay or programming error, it can be much harder to spot narrative errors even when you see them. Humans evolved to skip over a *lot* of information and their brain fills in the gaps. This means that errors slip through in a way that's more frequent than glaring visual or tactile problems.
- e. To top it off, like game design at large, a huge portion of general audiences think they could write a great book if they just had the time and motivation to type one up. The result is that a lot of people will dismiss, downplay, or belittle the contributions of the writers because "writing is easy." And you'll hear that over and over while a man with a

front-butt sneers at your game demo and points out a “plot hole” in the five minutes he got to play. See what I mean about the macro writing being easier, Bradley?

4. Gameplay Mechanics – These are ideas too, but there are a *fucking lot of them* in most games. These are the building blocks for how every piece of the game interacts with every other piece. I’ll give you two clear examples using my sick-as-hell Joust idea:
  - a. If my Fire Dragon breathes fire on your Dragon, what happens? Yes, I’m asking you, Trevor. **What happens?** Put on the spot, most people would rattle off a pretty decent list of interactions of the top of their head, but they inevitable miss a fair number of mid-level mechanics, at least one *super* important high-level mechanic, and then a dozen low-level mechanics. It takes a team a week or two tirelessly working out the answer to each of those questions: Does the fire hurt the dragon since it has dragon scales? What if it just causes a debuff to the dragon? Does the debuff do anything else meaningful? What happens to the rider? Well, what kind of armor were they wear? Does the gear have stats? Does the player’s class provide any resistances or abilities to allow them to mitigate the attack? How much HP does the rider have? How much damage does the fire do? Does the fire damage the gear? Do you lose gear entirely if it gets sufficiently harmed? What about *the weapon*? Should a player have to continue playing the game down such a handicap? What if we turn that handicap into an opportunity whereby the player has the option to fly over other jousters and leap onto their dragon, engaging them in 1v1 combat to take control of their dragon and their weapons? It never ends, Kristen!
    - i. This giant chunk of mechanical questions is valid – Every idea there is a topic that needs raised and answered. I came up with them quickly and it resulted in an absolute deluge of information. And unlike an overflow of core concepts (which are all just cool thought experiments), these questions create immediate design concerns because their interactions can drastically increase or decrease the size/scope of the game, even to the point of changing its genre or theme to fit new mechanics that the designers fell in love with. Importantly, these mechanical issues are “easy” in that they’re simple; the ones a Gameplay Designer can come up with in thirty seconds after hearing about a mechanic.
    - ii. You have to have an eye for mechanical failure points because it’s your job to spot an issue *before it becomes an issue*. That’s how you save a team hours and hours of wasted work on a build that doesn’t do the *only thing it’s supposed to do* just because Garry forgot to finish altering dragon flight physics to sync up with player physics, so the whole game is just jousters ***flying around, attached to nothing but their own instant fear of their very immediate mortality*** while the dragons rotate around, ping-ponging about the screen until someone busts a gut laughing. But that brings us back around to your job, Arnold.
  - b. If my Teleport Javelin passes through your Phase Armor as you re-materialize, how does the Joust game know how to handle this edge case scenario? That’s our job too. And that’s a much scarier, weirder job because you have *no clue* what’s actually making it happen. And if it creates a crash state as it happens, it’s so damn hard to even figure out

**what** went wrong, let alone how to fix it. This whole mechanical step, you're generating and using a slew of ideas; but they're usually not as sexy and fun as the core concept or overarching narrative ideas. These ones are granular, dirty, and the don't tend to play nice together, which means you have to be ruthless in chopping away the bits that don't fit with the other bits (even if you like them in a vacuum). You have to be able to discern those issues as quickly as possible, construct an analysis for how to translate that to the team, and then you have to work as a team to figure out a thorough, fast, and doable fix given the catastrophic nature of the bug. These ideas? These ideas are hard, but they're the engineering behind the scenes that makes your plane take off without losing wheels or having its door fly off mid-flight, Monica. You **need** good nuts-and-bolts people on your team, and they're a step past "ideas," and into statistics, psychology, math, and attention to detail that catches the trouble spots quickly.

5. Sound Design – These people are providing an absolute necessity to any game; an imperative set of "ideas." But their role is also highly specialized. Most people couldn't sit down with Audacity and a mixer program do anything other than make Mozart turn in his grave. As such, people don't question the Sound Designers much. Despite that, the value placed on the sound design of a game can often be dismissed itself, but it's a dire mistake, Natasha. These designers have to stitch together the theme of the game, the characters, the mechanics, and the narrative all at once. The game only starts to come together when there's excellent sound design that makes the protagonists feel more relatable and lovable; that makes the villain feel threatening and unstoppable; that makes the battles *bop*; and that makes the dramatic moments feel awe-inspiring. When a character is dying and you hear their last words in a way that haunts you. All of that is the weave between narrative, art, sound, and music – glued together exclusively by the sound designer. These people take a lot of inspiration and direction from every other member of the team, but almost no one ever understands their craft aside from them, so their unique importance can't be overstated.
  - a. As a short example, there's one thing that can quickly and easily separate a game with a good mixer from a bad one. It's whether the gameplay loop has music that is pleasing to listen to on repeat- possibly for hours at a time. It has to feel unintrusive, thematically appropriate, and it has to fit the tone of what you're doing. Those are actual, technical skills used in the service of **Ideas**.
6. Programmers – The programmers are creating "ideas" to solve problems non-stop. They're figuring out how to make a nebulous object do the thing they need it to do, and then they need to add a layer and try again. And then when something breaks, they have to figure out **exactly** where it broke. And then they have to do that every single time the gameplay team adds another convoluted mechanic. They have to identify the strengths and weaknesses of a given approach to solving a problem in the nuts and bolts of the game. If their ideas are sloppy, the game's code will be sloppy. This can make it buggy, choppy, or even unable to be fixed if someone *wanted* to touch up their mess. There may be more of them than most other roles, but if they don't work as a cohesive unit, planning their ideas together, the foundations of your game can easily wind up unstable and headed for certain disaster.

7. The Director – She has to keep track of how each department is coordinating together to avoid miscommunications; but she’s also going to be the judge for the first visual tests and the first mechanical tests and the first narrative tests and the first audio tests.
  - a. The Director has to pull from her own pool of knowledge and wisdom cobbled together across her career to do something no artist ever wants to do: she tears things out. She says, “I’m picking this one and we’re moving on.” Because if the Director doesn’t do that, the project stalls. It stalls over a point that people get hung up on because they’re passionate about an idea, but it *needs* to be done, Melanie, otherwise the game never gets made. Their creativity is in overdrive, conceptualizing each piece of the game, laid out in front of her. She has to say, “which of these elements work together and which ones don’t? Is the issue artistic in nature? Mechanical? Or are there some outside sociocultural factors that make the game’s overall direction incompatible with the current climate?” The Director has to contain all of that and make all of those decisions for the game’s health and well-being, but she also has to block out rough storyboarding/sketch ideas to give the visdev team an idea of her (and/or the team’s) vision. She has to proactively approach the gameplay team about mechanical improvements or the narrative team about the best overall layout for the story beats taken in context of the other game’s elements. The Director’s ideas encompass ***everything***, Jeff.
8. Art – Your Art team does a lot of what the sound team and the Director do, but on a more front-facing level, making sure every part of the process has a cohesive visual flow ***or*** conveys a specific message to the players. Art can be used to communicate a truly ludicrous amount of information with the most subtle colors, objects, expressions, and animations. Every one of those is a thought- an idea.
  - a. The marketers and producers are artists like this too – their ideas generate hype for the game; they give the audience a taste of the game’s feel. And if the marketer’s ideas are bad, the game’s trailer will be boring; the ads will target the wrong communities; and the promotional material will be ignored every time you try to pitch your game.

Those are your “ideas” people.

And from here, I’ve built up the question and answer like Coke with mentos shoved in them: ready to burst in your face like the appendix you used to have.

What’s the difference been their ideas and yours, oh hypothetical gamer mine. Don’t worry, Steven. I’ve got you covered.

The answer is execution. All of those roles I listed? They execute on their ideas. They turn them from a concept into a real, working part of the game itself. With each of them, the game would fail. Not because you don’t have enough good ideas, but because you couldn’t do anything with those ideas. They’re grains of sand in a desert, getting all up in your shoes and your face, and then you’ve got sand in your mouth. No one wants sand in their mouth, *Chet*. So stop asking for people to make your game for you.