The Game Development Mindset: 45 Strategies for Growth, Productivity, and Overcoming Failure

By Daniel Doan

Contents

Foreword

I’ve always been fascinated with game design. Ever since I could remember, I would theorycraft about games. I was hopelessly addicted to trading card games as well, and I’d continually question the rules and make my own variants. I would design dozens of board games in my bedroom and invite friends over to play them. None of them would amount to more than just small prototypes, but this served as the roots of my learning as a game designer. And it was through this childhood trial and error that I learned the very first of many important lessons of game design: testing with actual players. I didn't know it at the time, but I was practicing a habit of continual playtesting and iteration.

In my formative years, I got into making custom maps and mods for games such as Age of Empires and Starcraft, and got a few hundred people to play them. Through this experience I learned that getting people to play my creations (marketing) was quite a time-consuming task, something that would inspire my career later on.

In high school, I played several video games at a hyper-competitive level, most notably one called League of Legends, one of the largest online competitive multiplayer games in the world as of this writing. In fact, I took a year after graduating high school in an effort to try to see if my skills were apt enough to become a professional gamer. During my journey, I got the chance to befriend and regularly play with some of the best players in the world, who have played for teams such as TSM and CLG. This stint was short-lived, however, as I realized that the game was slowly consuming my life. But, that’s another story for another time.

Back in college, I made a game called **Generic RPG 1.0**, a tiny text-based combat system, and continued to iterate from there.

This blossomed into a few other side projects that never saw the light of day, but ultimately resulted in the release of **SanctuaryRPG: Black Edition** on Steam, a game that has been played by over half a million people worldwide.

The resulting success catapulted me head first into the industry, and over the past few years I’ve been fortunate enough to have been in contact with many wildly successful game developers, whose brains I have tapped for their invaluable knowledge and expertise.

Working in the games industry has been deeply rewarding for me so far, but it doesn’t come without its many pitfalls.

Game development has also been a very tough journey. My intent with this book is to share with you strategies and shortcuts on how to improve as fast as possible, to get the results that you want to achieve. Both in life as well as in your game development career.

I've boiled down all of my experiences into what I find are the three key elements to success in game development: **Productivity** (getting more done in less time), **Growth** (getting gains while avoiding plateaus), and **Overcoming Failure** (embracing what might hold you back from reaching your goals).

Each of these elements works in tandem with one another to create a pragmatic approach to game development that will keep you moving forward towards success.

Putting these strategies into practice has helped me immensely over the years, and it is my hope that they will help you as well.

Productivity

The first step to becoming a more successful game developer comes from within, internalizing what it means to be productive and practicing good habits that will allow you to push past roadblocks.

Prioritize and schedule the right things.

One of the major improvements that I've made in my own day-to-day life is to assign one priority task to work on each and every day. Although I do plan to complete other tasks during that same day, the priority task is the most important thing to get done. The power of choosing one main task is that it naturally guides your behavior by really forcing you to organize your day around that one responsibility. For instance, in game development, I might decide that a certain section of levels might be imbalanced, so I'll put that on the agenda to get done that day. I may tinker with X, Y, AND Z, but I don't consider the day a success until I accomplish my priority task.

Also, remember to manage your energy and not necessarily your time. If your task really requires your undivided attention, then schedule it for the period during the day where you have energy needed to focus. For example, I've noticed personally that I'm super productive during the mornings but it fades as the day goes on. So, what do I do to work around this? I schedule in the most important tasks in the morning. I reserve my evenings for more creative tasks to keep myself sane. Some developers find their stride midday or in the evening, so the important thing is to find what time works best for you. Nearly every single productivity tactic really harps on time management, but time is useless if you don't have enough energy to complete the task that you're trying to get finished.

Get into the state of flow.

In short, flow is the mental state that you experience during moments of intense focus and peak performance, when you are so focused on the task that the rest of the world fades away. In order to be in this state of flow, a few things need to happen.

First, you need to be fully involved in the task—you're taking action. **Second**, you need to be focused on the challenge that you're trying to overcome. And **Third**, you need to be enjoying the activity— divine blend of hard work and happiness. You're working on something challenging but also fully enjoying the moment.

How do you achieve this? Well, one of the key aspects of flow is immediate feedback. Not only do you need to work on challenges at the right degree of difficulty, but you also need to measure your immediate progress. Getting feedback about how you're performing at every step is crucial (this explains why it's hard to get immersed in a poorly made video game with terrible

UX design). Find a way to measure that feedback by testing the game with real playtesters. Ask the right questions, and you'll find the right answers.

Focus on the 80/20 rule.

Also known as the Pareto Principle, always remember that when trying to improve at something, most of your action probably isn't as helpful as you may think. Always try to measure how effective each of your actions are and double down on what works.

For example, in competitive gaming, if I wanted to increase my skill rating, I would experiment with different things such as time of day (whether I was better in the mornings or afternoons), or different practice routines (should I play 5 matches in a row, or take a break in between each match?). Things like that go a long way toward optimizing your effectiveness. In game development, you should figure out what parts of your game will have the most positive impact on the player and spend your time working on that as much as possible. It's oftentimes easy to fall into the trap of doing something that you *feel* is important, but at the end of the day, it's all about the results. For example, if your target demographic won't care about some element in your game, there's no need to spend hundreds of hours meticulously crafting it. After a while, take a step back and analyze again in order to re-optimize your efforts.

Don't bite off more than you can chew.

As humans, we usually choose projects based on what we would like to see happen. However, it is important to perform a realistic assessment of your resources and abilities before you set about trying to achieve whatever it is that you're going after. You need to ask yourself if you have the potential to develop the technical ability and willpower to finish the project. You also need to ask yourself whether or not completing your game is within your financial means.

For example, in game development, if you decide to make a game purely for creative reasons, make sure that you have enough in the bank to sustain yourself during and after the development cycle. Everything you need to do for your game project needs to be carefully analyzed. Are the things you want to have done feasible and cost-efficient for you or your team to perform? If there is something beyond your means, do you personally know someone that can help you? There is no shame in asking others for help or assistance, just make sure to plan out what you need ahead of time. If you can't do something, or can't find someone willing to help you for the right price, then maybe you should decrease the scope of your project.

The easiest way to complete a game is to revisit games you have made in the past. You should know what has worked for you and what has not. In this scenario, you probably also know how long it would take for you to complete a similar project and the associated costs. Building something you've already done in the past is the most practical plan because you are already familiar with the efforts involved. However, there no great reward without risk there is no reason as to why you should not take risks every once in a while and try something completely new. You'll need to plan out your actions far ahead of time and be prepared for new or unexpected costs during the development cycle. Be realistic with your game projects, and don't bite off more than you can chew. Make sure that you have enough resources to turn your idea into a tangible reality. Work on projects that you're already familiar with, and make sure that you can see it through to completion. The last thing you want to do is start another project that'll never see the light of day, so think about how you will finish a project and what steps you'll need to take to get there. At one point in my career I was working on three different games at the same time. Needless to say, none of those games ever saw a proper release.

Enjoy the process, not just the goal.

A large part of competitive video games is winning (some may argue it's the only thing that really matters). The pursuit of conquering your enemies. The distant hum of the crowd as you make a crucial last-minute play, sealing the opposing team's fate. The glory of victory. Now it's fine to set a goal to win, but remember that the process is much more important than the goal itself.

If you'll excuse me, I'd like to get a little meta: My goal right now is to finish writing this book, but if I don't actually enjoy the process, it's going to be torture and I'll more than likely give up before it gets done. The same thing goes for careers. If you want to be a successful game developer more than you want to do the work that it takes to get there, the chances of success are slim to none. If you don't enjoy the process, either learn how to, or find a different goal to accomplish. Successful people need to be truly exceptional at whatever it is that they're doing, and the only way to do great work is to love what you do.

This is not to say that there will be times where it gets rough or you think you might hate what you're doing...

Set new personal challenges.

Every so often during my work (or hyper-competitive gaming sessions), I might feel really bored or uninterested. Upon introspection, most of the time it's due to me having recently accomplishing some goal that I've set for myself and then a subsequent lack of setting another one in its place. These personal challenges give me some kind of target to hit and make the daily grind much more rewarding. Remember to set more than one personal goal (a few micro and one macro). For instance, if you were trying to finish developing an indie game, you could set monthly, weekly, and daily goals. This way, you're always progressing toward some kind of milestone to keep you moving forward.

Stay in your optimal zone, and track your progress.

When trying to improve your craft, remember that what's measured is improved. In competitive gaming, I always made an effort to improve my score. Day after day, month after month, it was all about improvement. There's one small caveat to this rule though, and that's . . . striving to remain in the optimal zone of difficulty.

For example, imagine that you're playing a game of Chess. If you try to play a serious match against a 6-year-old, you'll probably quickly become bored. It's way too easy (unless you were unlucky in your choice of 6-year-olds). On the opposite end of the spectrum, if you're facing off against a seasoned Grandmaster . . . you'll quickly find that you'll become demotivated for an entirely different reason: the match is too difficult.

Now, compare these experiences to playing Chess against someone who is around your skill range. You have a chance of winning, but only if you really try. Your focus increases, distractions fade away, and you find yourself completely immersed in the task at hand. In game development, make sure that you're continually challenging yourself with tasks that are just above your reach. If you've only made a text-based RPG, it makes no sense to try to jump to creating AAA games in CryEngine. Conversely, it wouldn't make sense to just keep making simple text-based games over and over again. Tasks that are below your current abilities are boring. And Tasks that are way beyond your current abilities can be discouraging and disheartening. Try to focus on the sweet spots: asks that are right on the border of success and failure are extremely motivating to our brains. At our cores, we want nothing more than to truly master a skill that is just beyond our current horizon—just remember that this is a moving target, so it's imperative to reflect on our current skill level from time to time to remind ourselves how far we've come.

Make your daily routine more engaging.

Because of the various metrics worth considering in what we do every day, people often have conflicting mindsets for what makes a day productive. In my opinion, however, the most important by far is long-term progression. A day with no progression will be one that quickly ruins your motivation to actually continue doing what you're doing. Back when I was waist-deep in competitive gaming, there were days where I would lose most of my competitive games, which would result in me ending up in some kind of lower skill bracket. Not only was I getting frustrated with losing, I found myself demoted, furthering my frustrations.

One way I could have addressed this was to factor in a progression system that didn't rely on a single metric. I was way too focused on "gaining a higher rank" that I tunnel-visioned. I was no longer focusing on any other metric besides winning games. Had I known this, I would have kept track of other metrics like accuracy or APM or engaged in another activity outside of competitive play that could result in the progression that I was looking for. As a general rule, it's important for you to have current and available progress visualized so that goals are recognizable as achievable.

Practice the art of saying no.

This is harder than it seems, but it's an invaluable way to free up time that you otherwise wouldn't have. The difficulty comes from emotional values that make a person fear rejection or incompetence if they speak up. Don't get side-tracked if you need to dedicate time to your important tasks—and definitely don't slow down because someone else wants to use you for something. If your task is truly important, it should be prioritized over anything else that could be coming up. If you want to become one of the best players in the world in a competitive video game (like I once did), you need to focus on that to the exclusion of other things that are competing for your attention. If you want to create and release a masterpiece of an indie game, make sure to cram that into the schedule as much as possible.

Don't let yourself get sidetracked.

If you want to get great at something, nothing will ever take the place of practice. But remember, don't overburden yourself. You might have hundreds of things to do, but the key to doing them all is to focus on a few specific things on a daily basis and complete those. Yes, this will leave the other tasks hanging but multi-tasking means less focus and very little progress for any of the tasks.

If you try to multitask, one thing leads to another and another and another. And at the end of the day, nothing has really been accomplished, or even worse, more work was added than done. Narrow down your daily focus to only a few goals at a time and get those accomplished. Sometimes the best solution is to simply ignore some things. "Ignore" is a rather nasty word, but if you don't learn to slow down the effects of distractions on your work, your important tasks won't be receiving the attention that they deserve. Learn to postpone doing things that don't need to be done immediately in favor of your important daily routines and your diligent practice.

Keep yourself focused.

So, most people don't have trouble with focusing, per se: they have trouble with deciding. One technique that has always worked for me is to streamline my tasks so that I only tackle one at a time. For instance, in competitive gaming, if I wanted to get better at a certain mechanic, I would focus exclusively on practicing that mechanic for hours on end—to the exclusion of everything else—and set a deadline for it.

Have you ever had a task that you absolutely had to get done? You probably finished it because the deadline ultimately made the decision for you. Maybe you procrastinated beforehand, but once things became extra urgent and you were forced to make a decision, you took action and did it.

Make yourself extra efficient by only focusing on one thing at a time. Sometimes, instead of doing the difficult work of choosing only one thing to focus on, we often convince ourselves that multitasking is the far superior option. This is ineffective; I mean technically we're capable of doing two things at the same time, such as watching TV while working, or answering an email while chatting up someone on the phone. What's really impossible, though, is fully concentrating on two tasks at once. You're either listening to the TV or paying attention to meticulously crafting the email or talking to your best friend on the phone. During any instant, you're concentrating on either one or the other. Multitasking is detrimental to improvement, as you're not fully focused.

Be mentally prepared.

When you're tired, sleepy, or agitated, you'll perform worse than usual. In competitive gaming, this is called going on tilt: it's never really a good idea to play when you're in this state no matter how committed you are to the game. In short, you're just going to have a bad time all around. In game development, stop forcing yourself to work on your game if you're feeling burned out. This may make you resent game development and quit altogether. Instead, take a break and go do something else for a while. This can be hours, days, or even months (I personally had to take a month-long break while developing SanctuaryRPG in order to stay sane, but it's totally up to your needs).

Practice the art of consistency.

Remember that discomfort is temporary and that you'll never regret good work once it's done. We want our work to be great, but we don't want to struggle through our work. We want flat stomachs and rock-hard abs but we don't want to grind through yet another workout. We oftentimes want the final result but not the countless failed attempts that precede it. No grind, no gold.

Despite our natural resistance to it, I personally have never found myself feeling worse after the hard work was done or the goal accomplished. There have been days when it was really hard to start or push through, but it was always worth doing. Have the courage to show up and do the work, even in an average manner. It's better than not showing up at all. In order to write this book I forced myself to write a few hundred words a day, every day. Life is a constant balance between slipping into the ease of distraction or defeating the pain of discipline. Our lives and even our identities are defined by this balance. Life is the sum of a hundred thousand internal battles that we have with ourselves daily, and tiny decisions about whether to stay the course or give up. Life isn't a dress rehearsal. Spend your time in a way that will make your future self proud. Show up.

Growth

The second step to becoming a successful game developer comes from leveraging all that you have towards growth. Nearly all independent game developers start at the bottom, and growth is crucial once you've established a strong work ethic.

Be prepared to suffer.

As with anything worth doing, it will probably require a lot of suffering. Trying to get good at competitive games is oftentimes akin to trying to break through a brick wall with a spoon; it may take a quite a bit of time but you'll succeed eventually. My business partner and I once started on a game development project and poured several hundreds of hours into it; we never got past the prototype stage. There's no beating around the bush with this one: game development is hard, and if you're working on a larger project with a scope of a year or two, expect to suffer. If you're looking to make a game, you should have a clear vision and enough motivation to make it through the pain of setbacks and hurdles and anything else life will try to throw at you. Having to delay your gratification for a long time is tough, and not everyone will be able to make it past this stage, so prepare for it.

Set your own milestones.

When I first started gaming on a competitive level, I had a single goal in mind: I wanted to be one of the best *League of Legends* players in the world. I'm not sure what my young mind was thinking at the time, but it felt right. As I started climbing and getting better at the game, I realized that no matter how good at the game I got, there was always someone who was better than me that I was envious of. This never changed, even as I approached the upper echelons of competitive play. Bottom line, you'll never be "good enough," whatever good enough may be—and even if you think you are, the passing of time will ensure that you'll get surpassed if you don't continually seek to improve yourself.

In game development, make sure to set goals that are relevant to you. Want to create the highest rated game in your genre? Want to make your first six figures from a single game? Set the goal and work to accomplish it. Then, create the next one and repeat.

Even if you're not ready, start anyway.

If you're doing challenging work, you'll never feel truly ready. A side effect of doing truly meaningful work is that you're simultaneously pulled by excitement and pushed by confusion. You're bound to feel unprepared, unqualified, and most definitely uncertain. But seriously, what you have right now is enough. You can delay or plan all you want, but you have what it takes to get started. It doesn't matter what you're trying to do, actually; you just need to get started. We all start in the same place, with no contacts, and no experience. The difference is that the winners chose to start anyway. No matter what your situation is, start before you feel ready.

Be okay with never feeling fulfilled.

In competitive gaming, no matter how good I got I always felt like I was never good enough. I'd read success stories and get motivated, but at the end of the day, I had to admit that there were aspects of my play that still needed to be worked on. There's definitely something to be said in game development about looking at successful developers for inspiration, but you'd be remiss not to feel like you're only a single person in the grand scheme of things.

When you read a really great success story, either you're motivated to do better than them, or you're demoralized and feel like you have a super long way to go. I have to admit that a lot of people do lean towards the latter. At some point, everyone will deal with demoralization. I personally try to overcome this by studying the habits of success, and trying to emulate them. What exactly did he do, and how did he do it? What mechanics make his game work? Look at others as chances to learn, and you'll naturally improve.

Stop training and start doing.

If you want to get really good at let's say… basketball, you can spend all the time in the world practicing free throws, but that isn't going to prepare you for the intensity of a real game. However, this doesn't mean that training is a bad thing. Practicing free throws enough times will make it so that during a real match, sinking that basket will be the least of your concerns since you've gotten it down. Your mind is able to focus on all of the other aspects of the game, thus improving your overall skill level. In game development, you'll need to focus on honing your core fundamentals. If you're a programmer, get super comfortable with whatever engine or framework that you're working with before trying to learn other frameworks. If you're an artist, really focus on mastering one particular style. Focus on your core fundamentals, and the rest will follow suit.

Clearly identify your obstacles.

To improve, you have to figure out what's holding you back. Start with yourself. Is there a situation that regularly causes stress in your day? This kind of frustration can actually be an indicator that a bottleneck exists somewhere in your routine. Focus on isolating that particular problem and make a conscious effort to address it.

In my competitive gaming prime, I identified that a large part of what was holding me back was my inner feelings of deep frustration when I lost a match. This really stunted my progress, as my emotions were acting as a bottleneck that put me in the wrong headspace to learn from my mistakes. And during game development, my personal biggest bottleneck was art: I've always been horrendously bad at it. But I wasn't going to let that stop me from creating what I wanted though, and I used placeholder art while building out the core gameplay, only recruiting an artist toward the end of the development cycle of my first game.

Don’t get ahead of yourself.

Recently, a friend and I were talking about the human will towards mastery. The conversation initially started as a discussion on the attraction people have toward more and more specific tools as they advance in a particular craft. For instance, I was noting how as I got deeper and deeper into programming and game development, I get pickier about the type of keyboard that I use, and I find that I'm accumulating a variety of neat software tools and shortcuts as well that I wouldn’t have understood the use of when I was starting out.

Sometimes, people gather tools in the absence of experience. I think of all of the amateur developers who think that they need a top-of-the-line system to even begin learning programming. Perhaps they think that if they have the tools, they’ll become experts by simply looking the part or using the right tools. I mean, it certainly doesn’t hurt to have functioning tools, but it will become a bottleneck if you let it hold you back mentally.

Push your unfair advantages.

Figure out what you’re good at and double down on it. Whether that’s a particular playstyle in competitive gaming or a certain framework in game development, make sure that it fits what you’ve already spent years getting used to doing. For instance, I grew up designing board games and modding computer games and followed that up with creating software for other companies. Then I finally transitioned into game development. It was a much more natural progression than just jumping right in. Remember that there are many different roles within the industry. For example, there’s no need to force yourself to become an artist if you think you’re not experienced or motivated enough to do it. I know I’m certainly not.

Stop making excuses for yourself.

So, let’s talk about skill. For the purposes of this discussion, we’ll define skill as applied knowledge alongside talent and execution. Skill is, at its core, relative. Maybe being on the top of a certain leaderboard in a video game is most skillful for you. Maybe it’s winning the match, or maybe it’s trying different or creative strategies. However, one thing will without a doubt lead to failure, and that thing is to look at others and go, “That’s not skillful. I’m better than that person because I don’t do the cheap, no-skill thing that they rely on in order to win.” Now, whatever they’re doing to get ahead may or may not be objectively skillful, or according to your personal definition, but worrying about such things is a complete waste of your time. What you should worry about is how to adapt to that instead of complaining. This mental block will actually prevent you from adapting. Remember, treat everything that bests you as skillful. Why? Because it was proven to be superior and generated results. Remember that the fastest way to improve is to take a look at the person in the mirror and start from there. In game development, don’t go around looking to make excuses and call others out on their strategies for success, no matter how “skill-less” you perceive their actions. I’ve been called a ton of names because I bought a licensed art pack for my second game, Overture. Does this really matter in the long run? Not really. There are many ways towards success. Do what you can with what you have, and leave the judgement at home.

Leverage the power of marginal stacked gains.

It's far too easy to overestimate the importance of the singular defining moment and underestimate the value of consistently making better decisions every day. Almost every habit that we have is the result of many micro-decisions over time, and yet we easily forget this when we want to make a change. We may convince ourselves that change is only really meaningful if there's some large outcome associated with it. Whatever it is, we oftentimes put immense pressure on ourselves to make some overt and sweeping improvement that everyone will instantly notice.

Improving by just a small margin isn't immediately notable, but it can be just as meaningful in the long run. This pattern also works the exact same way in reverse: If you find yourself engaged in bad habits or poor results, it's most likely not because something dramatic happened overnight. It's usually the sum of many tiny choices here and there that eventually leads to a problem. When focusing on marginal changes, there is basically no difference between making a choice that is one percent worse or better, but as time goes on, these small changes compound and there's a large gap between people who make positive ones versus negative ones. Small choices may not seem like make that much of a difference, but the reality is quite opposite.

Take constructive criticism, even if you feel a little defensive.

Remember, no matter what you do, you'll always have haters. You'll always do something that someone else disagrees with, regardless of your intentions. In online multiplayer games, especially team games, if you do something that your teammates don't necessarily agree with, or you mess up somehow, you'll get a ton of flak for it. Try not to take it personal. Also, keep in mind that as humans we're oftentimes oblivious to our own mistakes, so we need someone to point out our flaws to us. A lot of what it takes to get better is practice after knowing what you're doing wrong.

In game development, if your game gets even a modicum of popularity, expect to get haters. I've been told that I'm the worst developer in the world, I've been told that I should get out of the industry because I'm worthless, I've been told that I'm the bottom of the barrel when it comes to game development. Is this true? I mean, to some degree. I'll fully admit that on some days I really do feel like I'm a talentless game developer because SanctuaryRPG hasn't shipped a million copies. But really, is it completely true? Not necessarily. You see, sometimes when I release a game, it ships with bugs that may temporarily break the game for players. Some of these players have gotten extremely upset to the point of verbally abusing me over the internet. As a result of this, I've taken a lot more measures to reduce the number of bugs that ship with major updates. Remember, it's all about extracting the valuable insight behind criticism—be in constructive or abusive. You're not a perfect person, and you must realize your flaws and address them accordingly.

Maintain the right attitude, and your life will be much easier.

You might still be able to get better without having the right attitude, but it'll probably be a much slower and potentially painful process. So, what's the right attitude you may ask? Well, we can start by defining the wrong attitude. David Sirlin, a notable game designer, has actually defined the a person with a bad attitude: a scrub. Don't be one. A scrub will blame others for their mistakes—they'll never admit that they make any mistakes, always have some kind of excuse for poor performance, and never think about how to fix their mistakes.

With this kind of mentality, you'll lose the game before it even starts. When you lose, the blame rests on you—yes, even if it's a team game. It's your responsibility to find your mistakes, learn from them, and fix them. This attitude of consistently workshopping your own behavior is crucial. In game development, always try to figure out what you might be doing wrong long before you ship a game with those issues. Get as much feedback as you can from family, friends, and strangers alike. Have them point out potential mistakes that you might be making, and genuinely listen to their advice.

Your own skills have their limits.

It's human nature to believe that we are really good at something that we're terrible at. This is known as the Dunning-Kruger effect. The sooner that you realize this, the faster you will improve. People who are unskilled tend to suffer from base cases of illusory superiority, rating their own skill as above average. Usually much higher than in actuality. In contrast, highly skilled people oftentimes underrate their abilities. If you want a good example of this, look at shows such as American Idol. Many of these people are just horrible at singing, yet they believe that they're good enough to be a professional singer. When you first do something, you won't think you're great at it. Give it a bit longer, and suddenly you think you know everything. You haven't yet learned enough to become a true expert, but you're a hell of a lot better than a rookie. It's easy for you to see the difference between you and a new player, but the difference between you and the real expert can seem invisible.

We're oftentimes completely incapable of understanding what makes the master better than us, because we haven't yet learned the seemingly insignificant and less obvious things that the expert has. Remember, in order to improve, you must approach every obstacle or defeat as, "This is due to my own lack of skill." Unless you're literally the most skillful person on the planet at whatever it is that you're trying to do, there are people who are better than you, and you'll need to learn from them in order to get better.

Go with the flow, not against it.

In competitive gaming, it was always important for me to figure out what the current strategy trends were and adapt to them rather than continuing using old strategies and techniques.

Picture that you're a soldier on a medieval battlefield. You're trying to enter a castle that lies ahead. Perched atop are dozens of archers with their bows drawn. You'll need to figure out how to get past the archers—there's no way around it. You're going to be facing archers, and it's up to you to determine what you want to do. Don't show up to the castle gates looking for a melee fight: adapt to survive. In game development, if you know that there are already a bunch of generic looking platformers that aren't performing very well, you'd be crazy to attempt to create yet another one and hope for success. The "meta" is always changing, and you must adapt to it. Even if your studio focuses on a specific genre, it never hurts to prototype out some mechanics or features found in what's currently the genre *de jour*.

Continually optimize your time.

There's no substitute for putting in the time, but you shouldn't just be playing just to play. We have a finite number of tries per day, and a finite number of days in a lifetime. You have to continually figure out how to improve where you're currently at. If you're trying to get better, it doesn't make sense to continue to do the same things over and over again, hoping that things will change for the better. In game development, think about what you're doing each and every day. Is that optimal? Will that help you reach your goals?

Discover and double down on your core competencies.

When I first started playing my competitive game of choice, I just played. It was great to discover new things in the game, and pretty soon I discovered that I was great at a certain subset of playable avatars in the game. I discovered my preferred playstyle, and I stuck to that playstyle for thousands of matches In fact, in my climb, I regularly just played the same one or two characters over and over again in order to master them.

In game development, it's important to know what you're good at—and what you're not good at. For example, I'm horrible at worldbuilding. My core competency revolves around designing intricate gameplay systems, making core game loops more engaging, and keeping players playing for as long as possible. I'm also good at aesthetic design since I've flirted with photography and graphic design in the past. However, I know very little about pixel art—my skills are negligible at best. I tried to learn pixel art at one point, I truly did, and as a result of that I actually am a relatively advanced beginner. But alas, I knew that pixel art wasn't my calling. I almost exclusively rely on the art of others in my game development projects, simply because I know my limitations. Remember that everything you do is a trade off. It's impossible to have the best of both worlds because you're ultimately giving up one thing for another every time you make a decision.

Be deliberate with your practice.

Ultimately, when I was playing competitive games, the goal was for me to get better as a player. Oftentimes I would meet people who had been playing the game for years and hadn't really improved, people who never had the correct mindset toward improvement.

Remember that deliberate practice is when you work on a skill that requires a handful of practice sessions to master. If it takes longer than that, you're probably working on something that's much too complex, so you need to break it down into smaller and more measurable chunks. Once you master this small behavior, you can move onto practicing the next small task. Rinse and repeat. Deliberate practice is necessary to transcend what many gamers refer to as "ELO Hell," or in more layman terms, a plateau—a state where players feel that their progress is stunted for some reason or another. Focus on deliberately improving one sub-aspect of whatever it is that you're trying to accomplish, then repeat. Purposeful practice has well-defined and specific goals that'll give you a way to objectively judge whether each practice session has resulted in success.

Remember, it's all about getting immediate feedback, so that you know exactly what to do to improve. It'll push you out of your comfort zone. It also requires your full attention: distractions are dangerous when trying to engage in this form of practice. Also, having some kind of coach, a third party to keep you focused and hold you accountable, really helps expedite this process. In addition to motivating you, their job is to break down particular aspects of key skills for development towards your overall performance goals. This is crucial for the momentum that you're trying to achieve.

Cultivate your passions.

When it comes to getting better at something, there's just no substitute for putting in the time and effort to make it happen. The problem is that, unfortunately, putting in the time and effort is hard. We only have so many hours in a day to spend, and if you're not passionate enough about something, you probably won't make the time for it.

When it came to competitive gaming, I was beyond passionate. In my prime, I often dedicated over twelve hours a day to practicing my craft. In game development, make sure that you're sufficiently passionate about your project. Stop abandoning your prototypes mid-development and stick to something that you just can't stop thinking about. You'll know it when it hits you.

Engage with people better than you.

Call them talented friends, call them mentors… it doesn't matter. There's just no other way. Learning from people who have already done what you want to do, or who have achieved what you're dreaming of achieving is one of the most effective shortcuts to getting better.

When I played competitively, after every single match, I would send friend requests to the top performing players and figure out if they wanted to practice with me. Remember to make use of other people's knowledge, and don't expect to be able to think of everything yourself. Stand on the shoulders of giants. You can also learn a ton from watching other people play. You'll get to see how they deal with unique situations, and you might discover things that you'll never in a million years figure out on your own. In game development, you need to find someone who has been there and done that. Learn from them relentlessly. Reach out to them. Talk to them. Give value to them, and they'll give value in return. And when you're a giant yourself, you can pay it forward and let new developers stand on *your* shoulders.

Trust in the process of repetition.

Whenever you put in consistent action and learn from your mistakes, you'll progress. I only got pretty good at competitive gaming because I had played thousands of matches over a span of several years. There's no other way.

For example, let's take a look at photography. If you want to be a photographer, you could try to take one photo a day and make it perfect, or you could take dozens of photos, do this daily, learn from your mistakes, and improve. If you want to be a successful game developer, you could scheme and brainstorm and try to plan out the perfect game—or you could just try to make a basic game, learn from your mistakes, and then experiment with new ideas and games until you hit something big.

It's not the quest to achieve some goal that'll make you better or determine your worth: it's the skills you develop from doing enough of the work. In other words, when thinking about your goals, don't just consider the outcome. Focus on the practice and piles of work that are prerequisite for the success. When you look at things this way, you'll start to realize putting in the time is infinitely more important than just simply setting a goal. The goal is just an event, something that you can't predict. But the work is what can make the event happen. If you ignore the outcomes and focus only on drilling the skills, you'll still get the results. Build habits and the outcomes will follow suit.

Reward yourself liberally.

Ideally, hard work toward your goal should be its own reward. This is usually true, but it certainly helps to have some kind of motivation to actually perform hard work consistently. For example, in the case of full time indie game development, having a few rest days on a weekly basis gives you small, local goals to work toward while you pursue the large ones. If you feel like being extravagant, you could buy yourself a brand new development computer once your indie game takes off. Having a reward anywhere between these events will give you something to break down your long term goals into more manageable ones, and as a result, you reduce the likelihood of being overwhelmed. This isn't to say you should reward yourself for everything you do on the way to your goal. While they're obviously a good thing, having too many or having rewards that break habits are not.

The key here is to make sure the rewards have meaning. If you reward yourself with an entire 2-liter of soda and a large order of bacon fries for every day you stick to your diet, you'll make your diet meaningless. Similarly, if you buy a new fancy pair of $2,000 yoga pants to reward yourself for saving up loose change, you’re invalidating your savings goals.

Here’s an example of a good reward: If you wanted to play video games but knew you had to clean your house first, reward yourself with a bit of game time in between each full room cleanup, like the bathroom or the kitchen. This breaks up the daunting task of cleaning an entire house and breaks it into smaller chunks that you can tackle. The video games are a good reward here because they’re only possible after actually completing the goals you set out to do in the first place. Similarly, for every milestone you accomplish in development, try and set yourself a reward. Maybe it’s a social outing, a small trip, a vacation day or a nice meal.

Stop reading so much, and take more action.

Okay, so don’t stop reading altogether (because reading is awesome), but the fact is that there’s nothing better than the art of doing. Remember that reading articles, brooks, or watching videos can never replace the process of actually doing something. When I first started out with programming, I would watch hundreds of programming tutorials online and consider myself pretty knowledgeable. When it came time to actually write code though, my brain kept tripping all over itself . . . until it learned how to fail gracefully and adapt. It wasn’t until I had actually started the art of coding that I really progressed; watching video after video without actually doing the work didn’t do much for me at all.

When you find yourself hitting some kind of plateau, that’s when you start hitting the books again. Technical books to help you overcome technical hurdles, motivational books to help fuel your inner drive for success, and fun books to help you relax and recharge.

Seek out experts in their fields.

This may sound kind of trite, but the first step to getting to where you want to be is finding out the most efficient route to your intended destination. When you’re first starting out with something, you might feel a bit overwhelmed. Perhaps you feel like there is too much to do and too few hours in the day. You may be overwhelmed by not knowing what metrics to track in order to move forward. If you’re ever thinking to yourself, "Wow, I could really use someone to help me figure out the best way to do this," then you should consider getting a mentor.

Going solo can cost much more in wasted time and money, even though you may think that you’re saving money by skimping out in the short term. How would it feel to have a sounding board for honest feedback on your ideas and goals, and a partner to support you in the process of ideation, implementation, and execution? Find someone or a group of people who have already done what you want to do. Frankly, sometimes we think we know the right path to take for growth, but we come to find that it ultimately isn’t working. In order to get results, we may need guidance from someone who can see things from a much more objective view.

If you’re looking for a bit of marketing or game development advice, feel free to send me an email: daniel@blackshellmedia.com – I promise to personally read and reply to every email that gets sent my way.

Overcoming Failure

The third step to becoming a successful game developer comes from maintaining that momentum in the face of setbacks and obstacles. Remember that productivity provides the initial surge, and growth provides the leverage, but mastering this third step is crucial, as how you deal with setbacks will determine whether or not you'll eventually succeed.

Don't blame yourself too much.

It's okay to blame external circumstance, but never let that upset you. If it does, take a breather and come back to the game when you're more calm. Regardless of how poorly your team did, there is always some aspect of your performance that can be improved upon: it's up to you to figure out what that was. To keep things in perspective, remember that you learn so much more after a loss than after a win. In order to improve, you'll never want to feel complacent in your skills. Losing allows you to reflect on what could potentially be improved on, rather than just the feeling of being "good enough" when you win. In game development, making a horrible game isn't an excuse for you to quit making games entirely: it's a chance for you to put your ego aside and think about what you could do better.

Learn how to fail gracefully.

Minimize the power of defeat in your mind. No one likes losing or failing, but success should be defined as failing up. Try to think of what it feels like to lose. Why does it make you feel so bad? Is it because your peers will mock you? Is it because you won't be perceived as "good"? Diffuse those thoughts and prove to yourself that they're inaccurate. Remember that winning or losing doesn't define everything about you. In video games, just because you've lost this particular round doesn't mean that you'll lose the next match. There are many different factors that go into each situation. You know, maybe the odds just weren't in your favor, or maybe you made a few mistakes that compounded, but this certainly doesn't make you a loser. In game development, the logic should be the same. Just because you finished dead last in a game jam doesn't mean that you should give up your dream, or feel bad about yourself. Remember, it's not about where you're at, but it's about how fast you can improve. Use every single failure as an opportunity to grow—you'll need to learn how to embrace failure to get ahead. You aren't going to hit a homerun every time, or even most of the time—you're just preparing yourself and training to get better, which is what'll get you to that potential success in the first place.

Be quick to move on.

Remember that people will never define you for how many times you fall but how many times that you stand up afterward. Now, this doesn't mean that you should completely ignore failure, but make sure to not let failure define you. In order to succeed at something, we must first acknowledge that failure is part of the process. The master has failed more times than the beginner has even tried. The ability to fail gracefully and learn from failure separates the rookie from the adepts, and this point is not to be taken lightly.

For the longest time I couldn’t forgive myself for releasing objectively bad indie games between my two commercial releases. I had been working on a few games titled Blue Whale Save and Silent Vac, and both were quite terrible in retrospect. Did I make a mistake releasing them? In retrospect, perhaps. However, dwelling on whether or not this was a mistake and feeling bad for myself simply doesn’t contribute to anything positive. Whenever you think that you’ve made a mistake or you think that you’ve failed in some capacity, simply make a mental note of why, and move on. Don’t dwell on it. It doesn’t help.

Never beat yourself up.

Don’t be the guy who punches himself in the face after failing.

I’ve actually been wanting to tell this story for a while: So back in the days where people actually frequented the local arcades, there was a guy that I played against in Street Fighter a few times. Let’s call him Randy. Every time Randy would lose, he would punch himself in the head—he was literally beating himself up for his failure. After one of these incidents, I got the nerve to ask him why he was hitting himself. He told me that he really wanted to hit me, but decided to hit himself instead because his parents told him that he couldn’t hit others. I felt sorry for the guy, but I could relate. I’m not sure what happened to Randy after all these years, but I definitely felt his pain. I, too, have wanted to punch myself in the face after failing at something so miserably.

So then, how do you suppress that primal urge to beat yourself up? Distance your ego from your actions. A neat psychological trick that I’ve learned is to properly frame failure. So for instance, if you failed at something, it’s not necessarily “you” that failed, but rather your actions that failed. Failing doesn’t mean that you’re a failure, it just means that you need to take a different course of action in order to rectify your mistakes. Problem solved.

Manage your emotions.

Managing emotions is a key element for success, as it’s super easy to be derailed by a poor mood or mindset. Successful people are able to channel anger and frustration into their work, keyword here being channel and not repress. When negative emotions creep in, make the conscious effort to acknowledge them and remind yourself that you can get past them. You may not be able to immediately change the condition that you’re in, but you can absolutely change the way that you deal with these emotions in a positive way.

In game development, if you find that you get frustrated or angry when trying to fix a bug in your code, perhaps you can turn that into a blog post or funny meme. My whole schtick with adding potatoes into my game SanctuaryRPG essentially came from wanting to entertain myself in moments of sheer frustration. Or let’s say that you’re angry because someone is flaming you in an internet forum. No worries, just remind yourself that at the end of the day, you can't lose an argument that you choose not to participate in—simply just step away from the situation.

Lower your confidence a little.

Yep, you read it right. It sounds counterintuitive, but it can actually be incredibly helpful. In order to succeed, you'll need to have high self-confidence to get things done, but if you selectively lower your confidence in certain areas, you might find that success may come a bit easier. Lowering your confidence may make you pay attention to feedback and help you to be more self-critical.

If you honestly think that you're perfect, you probably won't be at all receptive to any kind of feedback. Easing on your confidence level can make you prepare more and work harder. For example, if you're not convinced you're going to absolutely crush it at something, you're much more likely to spend more time practicing.

Embrace imperfection.

While rewards are a strong motivator to continue onward toward a goal, roadblocks on the way to success can completely crush your dreams. Everyone suffers setbacks in their personal journey, and this has been true for as long as there have been people around with dreams and aspirations. Thomas Edison threw several thousand swings at getting the lightbulb down, and although almost all of those swings were misses, there was a single success that mattered greatly in the end. The developers behind the hit mobile franchise Angry Birds made 51 games, almost went broke in the process, but hit paydirt on their 52nd game.

Many people give up their dreams when things don't work out the first time around. They forget that not everything will come easily. Given enough time, every single one of us will slip up while trying to change our habits in one way or another. When this happens, just acknowledge that a lapse did in fact occur; figure out why and move on. If you're obsessing over failures, you're not working toward your goals: the two are mutually exclusive.

It's much easier to forgive rather than berate yourself, as the former takes just a moment of willpower while the latter consumes all of your time and energy. Making mistakes or failing to adhere to a schedule is only human: there will always be a step or two or three backward as you progress forward. If you continue to work through these events, you'll come out ahead compared to what happens when you punish yourself for your failures, real or perceived.

Embrace the struggle.

I'm struggling as I write this very sentence. I am struggling today as you're reading this very sentence. If you've ever struggled to stay consistent with something you truly care about, maybe my struggle will resonate with you as well: It's been a while since I started Black Shell Media, a video game publishing and marketing company. During these mostly glorious, sometimes deeply frustrating days, I have made it my personal mission to add value to the world by helping indie game developers become more successful.

I have written article after article, book after book, week after week, month after month, year after year. But sometimes I struggle. Sometimes I don’t feel like writing. Sometimes I don’t feel like I have any great ideas, and sometimes I feel like giving up. But I remind myself that it takes grit to succeed.

Keep your eyes on the prize.

I’ll always remember the moment where I reached Master Tier in Starcraft 2. I’ll never forget the thrill of competing against the best players in the world in League of Legends. The satisfaction of releasing SanctuaryRPG to critical acclaim on Steam is just more than words can ever hope to describe. In game development, the daily grind can be a bit too much to bear sometimes, but trust me, it’s worth it in the end—even if it takes many failures to get there.

Once I started posting about SanctuaryRPG during its early development stages, the feedback that I got from people who thoroughly enjoyed the game was absolutely incredulous. Every person that buys or plays SanctuaryRPG represents a person who set aside time to support me and play my game, and those who have left a review have taken even more of their precious time to help spread the word—that feeling of love is really unreal. Watching a community grow around your baby and hearing from journalists and fans who play your game is one of the most rewarding life experiences I’ve ever had the pleasure of going through.

When I worked on SanctuaryRPG in my friend’s dorm room back in college, I never really thought that I would be creating a cult hit. I never thought that I would be the brainchild of a game that over half a million people have played. If you’re ever feeling like making a successful indie game is too daunting, just remember that you’re getting the chance to make someone’s year with your creation. It’ll be worth it. Trust me: I’ve been through it.

Afterword

Thank you so much for supporting my work. If this book has helped you in any way, please send me an email to let me know: daniel@blackshellmedia.com. I look forward to hearing from you and learning about your journey.

If you've found this book helpful, please let me know by leaving a rating on Amazon! It only takes less than a minute and would be incredibly helpful to me. To leave a review, go ahead and **click on this link** and you'll be taken to a page where you can leave your starred rating and tell the world what you thought about the book. If the link doesn't work for any reason, you can also find the book by searching the title in the Amazon store. Thank you so much again, I really appreciate it!