A Beginner's Guide to Indie Development : r/IndieDev

r/IndieDev•10 mo. ago

RoGlassDev

A Beginner's Guide to Indie Development

Who am I?

I've been developing games for about 15 years (since I was a teenager) using various editors, engines, coding languages, etc. I see myself as a designer first and foremost, but I've done art and programming for my games for a long time as well. I've worked as a solo dev, on an indie team of 4, at a larger indie company, and at a AAA company. After my recent venture at the AAA company (don't want to name it and cause drama) where both projects my team was working got canceled, I decided that after 5 years of not shipping a game, the only way to get one done was to do it myself. So here I am with a Early Access, almost ready to full release game called RoGlass. It's a tile-base Roguelite Puzzle game, and if you're interested in checking it out, here's the link: https://store.steampowered.com/app/2582540/RoGlass/

Why am I making this post?

I'm relatively new to Reddit, but I have seen a lot of devs (especially newer ones) on here struggling with where to start, how to get a game idea made, how to deal with motivation, etc. There's a lot of different aspects to game development and I wanted to share my experiences through some bullet point lists of things I did well, things I could have done better, and things I wish I knew before hand. Hopefully they help some of you!

I want to make a game, but where do I start?

1. First off, you need an idea. If this is your first game, make it as **SIMPLE AS POSSIBLE**. If you think it's too simple, simplify it more. Most beginner devs overscope their first game and never finish it. Some people start by replicating a game like Pac Man or Asteroids. Even simple games take a lot of effort to make and you'd be surprised how hard it is to make a game look and feel exactly like another.

2. Once you have an idea of what you'll want to make, research online to find an engine that will support that idea. Try different ones out and get a feel for what you like. I personally really enjoy the visual scripting of Unreal Engine's blueprint system.

3. Start by learning the basics of your engine of choice through step by step tutorials, but after you've got a decent grasp on how things work, stop watching random tutorials.

4. **Tutorial Hell** is where a lot of newer devs get stuck. They feel like they keep learning new things, but can't actually do anything with the knowledge they've gained. This is because every problem requires different solutions in game development. Also, step by step tutorials make it difficult to retain the process of how things can be done. You'll still need the fundamentals they teach to even navigate the engine though.

5. So what do you do if tutorials stop helping? Start by figuring out exactly what you're trying to do and look up how to do those very specific things. The better you can define what you're trying to do, the easier it will be to actually get it done.

6. Keep working on your game, share it with others as soon as you've got something working to get feedback as early as possible. I know it's scary and criticism can hurt, but no dev can make the perfect game straight from their head or in a bubble with no feedback.

7. Share, collect feedback, iterate, repeat. It will take a lot of tries to get your idea to work, so don't be afraid to throw things away. **Almost every game I've made barely resembles the idea I started with.** RoGlass was initially supposed to be a Rube Goldberg Idle game that you needed a certain amount of points per second to complete each level. Don't be afraid to start over if things aren't working out how you envisioned, but also, don't give up on an idea too quickly before you've given it a chance to work.

8. Don't look for funding. There are free tools and resources EVERYWHERE. You don't need a big budget to make a game, you just need time and effort. I've used Unreal5, Paint.NET, Audacity, Google Docs, etc. that are all free to use for my game RoGlass.

9. BONUS: If you've got a good grasp on how to make games, try starting with some game jams. Time, theme, mechanic, and control restrictions can lead to some fantastic ideas while keeping scope really small.

I've been working on my game for a while, but I am losing motivation. What should I do?

1. **Show off what you're making.** One of the biggest mistakes I made was not getting my game in front of people earlier on. Showing people what you're working on can really help boost motivation. It's really rewarding to talk with someone about an idea, implement it, and show it to them. If you're worried **"it isn't ready yet,"** then show it to friends and family. Game devs like to noodle their game forever, but eventually you need to show it to others. That feeling of "it isn't ready" very rarely goes away completely, so just suck it up and get it out there.

2. **Be open to new ideas, even if you don't think they'll work right away.** Game developers quite often have insights that you wouldn't have thought of, but so can non-game developers. Try to look at things from different perspectives and see if they can work for you game, but don't accept every idea people throw at you. Your vision is the key to keeping the game as one solid idea. A lot of friends/family might not be game developers (and not understand the impact of their ideas) and people you share with might have their own ideas of how your game should work.

3. **AVOID IMPOSTER SYNDROME** - This is probably the hardest hitting thing in game development, especially when working solo, **ESPECIALLY** when working on your first game. If you're working on games, **YOU ARE A GAME DEVELOPER.** Sure, success might be measured in different ways, but don't constantly compare yourself to others. Don't compare your game to others and think "mine is so much worse." Game development is all about personal growth, so compare yourself to yourself and you'll find that you are steadily improving over time. When you look back at how far you come after each game, and over several years, you'll see how much you've progressed.

4. **Create a production plan.** It sounds scary, but can be as simple as writing out stuff you need to get done for your game. Keep the tasks small in size and break bigger tasks into smaller ones. "Make the HUD" can be broken into "design the HUD layout on paper," "code the health bar," "code the energy bar," etc. Having concrete tasks that you can achieve in a short amount of time (between a few hours and a few days) can help boost motivation a ton and keep you on track to actually completing your game.

5. **Avoid feature creep.** Staring at the same thing day in and day out makes it feel boring after a while. You might be losing motivation because you think your game is boring, purely because you've spent hundreds of hours working on it and playing it. This causes some people to try to pivot to new ideas for their game to spice up the work flow. "What if there was a lumberjacking mini-game?!" and **BAM**, you just veer off into the metaphorical forest. **Don't add anything to your game that doesn't make it better.** Try to minimalize adding new features/systems until you've gotten the core of the game completed.

6. **Don't beat yourself up.** Your first game will almost certainly be bad. Your second too. It takes time to hone your skills, just like any other profession. No one is going to be amazing the first time they do something. Giving up or becoming complacent are your only enemies. As long as you keep trying to improve, you'll get better.

7. **Keep a good schedule and get good sleep. No, really.** It's pretty tough keeping a consistent schedule as a solo dev because you can work whenever you feel like working, sleep whenever you want, and be an absolute degenerate at times. Having a consistent schedule can help you focus yourself for when you should be working and let you relax when you need a break. Set times for when you should be working and times where you shouldn't be thinking about your game at all. Sometimes it's hard to stop thinking about ideas, coding problems, etc. and you can't shut it off. Allow yourself to have free time without worrying about your game. Everyone needs breaks.

I've been working on games as a hobby, but now I want to actually sell them. Where do I start?

This goes back to the idea phase. Why do you want to make games? For me, my most recent game was a push to get something out there after having multiple projects shut down on me. If your main goal is to be a developer as a living, then there are a few things you should do:

1. **First and foremost, make sure you can support yourself.** Don't quit your day job. If you're vehement on burning your savings, make sure you have enough runway for AT LEAST a year, but ideally 2+. Games take a long time to make, and your first commercial game probably won't be enough to sustain you. You can't make games if you don't have a place to live and food to eat!

2. **Look at the games that are popular.** What genres are people in love with at the moment? Also, keep your focus on games that are made with similar team sizes and resources. If you're a solo dev, don't look at the 4X game with 200 people working on it as something to go for.

3. **Try to start with an idea that is really marketable.** My puzzle game RoGlass has nice looking stained glass tiles, but no creatures, characters, or environments to show off. I hadn't thought about this in the early stages, but once I started marketing, I realized how much harder it was going to be. You don't have to have an idea that's perfect to market (as I said earlier, ideas will change anyways), but you should at least consider it as soon as possible.

4. **Don't just copy something else.** You might really enjoy Dark Souls, but trying to replicate Dark Souls will only end in failure. Take what you like about a genre or 1-3 specific games, and put a spin on them.

5. **On the flipside, don't make something so out of left field, people won't comprehend it.** It's good to be creative, but being too out there will confuse your

players and shrink your target audience. A lot of devs agree that the best approach is roughly 80-90% familiarity, with 10-20% innovation.

6. **As stated before, keep your game's scope small.** You can always expand on an idea in the future (you almost certainly will), but if you start too big, you'll never finish. I was working on a game for 2 years and had to scrap it because I knew that the goals I had for the game wouldn't be realized for at least another 1-3 years (and the quality I wanted wasn't even guaranteed).

7. **Keep things within your skillset.** You'll improve over time and it's good to push your skills, since it will allow you to become better at them. However, if you aren't much of an artist, don't try to emulate art from professionals who've done it for over a decade (my mistake from the point above). **Lean into your strong points and let them support your weaker ones.**

8. **A consistent theme and style is key.** If you're really lacking in a skill, you might want to ask for help or outsource. If you don't have resources for that, you will have to rely on what you can do and improve over time. Dream Quest is a perfect example of a game that looks like programmer art, but the gameplay is really solid and the style is consistent. Don't try to throw a bunch of high quality assets into your game if the rest of it can't match. This will look much worse than just setting the quality bar low and keeping it consistent. **There are way too many "asset flips" out there, don't be one of them.**

9. **Get a Steam page (or some sort of store page) up and running AS SOON AS POSSIBLE**. Not only does this help get initial feedback on the idea/aesthetic, but it allows you to start getting a fan base right away. I made the mistake of releasing in Early Access shortly after making my Steam page and didn't put any marketing efforts in before hand. Steam success stories rely on the snowball effect. The better your game does initially, the more that Steam shows it off. If you already released for Early Access, it isn't too late, but the process will be much harder.

10. **Market the hell out of your game as soon as you have ANYTHING to show.** It's fine if it isn't finished, just get stuff out there. Building up a fanbase is much harder to do after the fact and wishlists can help propel your game into the snowball status.

11. **No one has the magic formula for what makes a game successful or how to successfully market a game.** If anyone says they have a guaranteed success method, they're full of it. The most important thing to remember is that the more effort you put in, the more you get out. Trying more ideas and methods can only

improve your wishlists, sales, etc. A lot of people view games as a success or a failure and 99.9% of games start off in the failure category. Instead of viewing it this way, look at it as a ramp of progress. The more you push, the higher you go.

12. **Price your game at a reasonable price and keep in mind that most people buy games at a discount.** If you start with a discount, then Steam (or whatever store you use) takes a cut, and finally add in taxes, you'll end up with a lot less than you'd think. Many indie devs price their games at too low of a price and not only will you not get any profit, but you also will get lumped into the "cheap game" category. People will think your game is low quality because of the low price. Don't undersell your game, but also make sure the quality and content are a fair deal.

What makes a good game developer?

1. **Always be learning.** The moment you aren't learning anymore is the moment you've stopped pushing your limits and/or won't step outside of your comfort zone. Every game I work on teaches me countless new things. It's very exciting and terrifying that our career is never certain, but being too complacent can be your downfall. Your game dev journey can only end when you stop making games

2. **Perseverance and drive are key.** The most important aspect for any solo developer is self motivation. As long as you can stay motivated and keep pushing forwards, you'll consistently improve and actually make things. **Excuses are barriers that stop you from continuing your game dev journey.** Raw willpower is what makes or breaks people, because the resources available nowadays are endless. **Giving up is the only true way to fail.**

3. **Stay happy and healthy.** A lot of people think that "I'll just slave my life away and my soul will shine through my game" but that rarely works out how you'd imagine it. We're all human beings, and we need food, sleep, socialization, etc. to stay happy, healthy and sane. Don't push yourself so hard that you get burned out. Overexerting yourself when you're not physically or mentally up to it can lead to horrible mistakes and you'll end up creating more work for yourself in the future.

4. **Have thick skin. Criticism will come.** There are no perfect games, and your game is no exception. You need to take criticism on the chin and figure out how to improve your games from it. If someone says "your game sucks," ask them why and how to improve. Read between the lines of feedback you get to figure out what the core problems actually are. "The fireball is annoying to use" might actually mean "I'm tired of clicking to cast fireball so many times." When you devote your life to making a creation and someone trash talks it, it feels like they are insulting your pet or

child. **Taking criticism too personally will shut you off from improving your games.** On the other hand, some people are just haters. Let them hate and be on their way, don't let it get to you.

5. **Don't be afraid to ask for help.** A lot of people are scared that they won't get accepted, they will look incompetent, or people won't respect them if they ask for help. Put your ego aside and reach out when you need it. There are a lot of extremely friendly people out there who want to help each other, so reach out. Respect others and others will respect you.
6. **Focus on how to make ideas better instead of reinventing the wheel.** Ideas are a dime a dozen, but execution is key. If certain mechanics work really well in a game, figure out ways to improve them. Don't be afraid to borrow ideas and put your own twist on them. If you close off your view to everything else, you might make something from your own mind, but it's probably been done before (and better). Humans have improved over time by learning from each other, and game development is no different.
7. **Research early and often.** Play the competition. Your "competition" might actually end up becoming your ally. People who like specific genres try to find games in a similar vein. If your game offers a similar feeling as another with a new twist, those players will be MORE inclined to play your game. This is much more true for indie games than AAA games (where massive marketing budgets, launch times, etc. compete for player attention). You also have to keep in mind that you can buy 5-20 indie games for the price of 1 AAA game.

If you have any questions, feel free to comment below or message me directly.

I hope this post can be of use to some of you all and best of luck on your game dev journeys!