10 Ways to Improve your Indie Game Development

All credits go to original topic on www.gamedev.net

Step 1: Foster a community developed for – and by – your fans

The most important word here is foster. Just because you've thrown up a blog or a forum and you've got a twitter account, doesn't really mean you're fostering a community. To build the community and have it last, you've got to be involved with it. This means listening to, responding to, and even implementing the community's suggestions for the game. If you've drawn people to your site that are bothering to post, tweet or comment on your game, chances are they've got some good ideas. Clearly not all of them are, but every now and then there's a gem that'll actually improve the game. Don't be scared of trying a few of these ideas – the community will appreciate the fact that you're paying attention.

Big companies generally have the budgets to go beyond these basics – they create art packs and/or music for their community to build their own fansites. This is huge – you're basically giving your community the opportunity to do your marketing FOR YOU. Let them do it – give them the art, screenshots, wallpapers, mp3s. Whatever it takes to get your game out there – after all, there's thousands of games coming out a year, yours needs to stand out. So let your community work for you.

Step 2: Don't worry about graphics, but DO pay attention to art style

Many indie games do this right (Braid, Terraria, Revenge of the Titans, Castle Crashers, Frozen Synapse, Lume, etc etc). There's a ton of great examples where in lieu of big-budget graphics, they've opted for 8-bit looks or just a clean, stylized look. This is great for several reasons – it makes your game more recognizable and also makes your game far less hardware-dependent.

Where this might be a negative is if your look strays too far into a design that some might consider childlike or 'kid-friendly'. Bold, colorful graphics are great, but you might turn off some people who just prefer a more "adult game" look. Regardless of how many people think your game is the best-looking one they've ever seen, by making sure your game looks unique, you're going to guarantee that it will at least stand out, which gives it a far better chance of being remembered by your target audience

Step 3: Mods, mods, MODS

This is going to give your game longevity out the wazoo. Heck, I’ve replayed games for hours and hours just to try out new mods. Notable games that have a great mod scene are Half Life 2 (2006), Stalker: Shadow of Chernobyl (2007), Oblivion (2006), Torchlight (2009). Heck, even Morrowind (2002) has a dedicated mod community that’s as strong as ever. There’s many more than just these few examples, but the key here is longevity. Granted, Oblivion and Morrowind are huge games, expensive games, but even in the most recent super-ridiculous Steam Sale, they only dropped in price by $6 and $5, respectively. They’ve been lower in the past, but they maintain a reasonably high price for old games, mostly due to the fact that they can get it. They’re well supported, they (by now) run on almost all hardware, and staggeringly enormous amounts of mods are available for them. You can quite literally play these games for years. Only Half Life 2 has a mod community that rivals them.

Step 4: Let me play it first – the disappearing demo

This is something some devs do really, really well – Spiderweb Software and Soldak Entertainment come to mind here. Not only do they both provide comprehensive demos for every game they make, Soldak goes the extra mile and updates the demo with pretty much every patch they apply to the full game, so when you download one of their demos, you know it’s going to be representative of what you’re buying. Kudos to you for that, Soldak.

Demos are going the way of the dodo in the era of 10+ gigabyte games. That’s not surprising, but if you’re an indie developer, this is your chance to get your game out there into people’s hands. This may not always result in a sale, but if it does, that’s not only cash in your pocket, but also a potential one-person marketing team. If they like it, they’re probably going to tell their friends about it or possibly even join and become active in your community. This is worth its weight in gold.

Step 5: Stay focused on your simple, unique idea and implement it well

This is a tricky thing to nail down. You’re developing the game, so clearly you’re doing this. And in some ways, it flies in the face of my suggestion to take your community’s suggestions and implement them. I guess what I mean is this: make sure that whatever the core of your game is, do that one thing really well. In Braid, it was time-manipulation in a platformer. This could have bombed, if they’d over-complicated it, but they stuck to the basics and polished it until perfect. That’s what is going to make your game different from all the others, so make sure it works.

Step 6: Make your website look professional enough for people to give you money

This is something that can be overlooked quite easily. It’s also something that isn’t necessarily a hard-and-fast rule. But the way I see it, if I’m going to give you $10 or $20 of my money – or more – I want to feel like you are going to be around for a little while. I don’t want you and/or the site to disappear. This goes hand-in-hand with creating and supporting a community. If you’re working on your game a bunch and want it to be awesome, this is really a no-brainer. If web’s not really your thing, you’re more of a programmer, put some feelers out to the community and see if you can get some volunteers to add a level of polish.

Step 7: Keep working on it and keep those patches coming

This, like above, is going to give your audience and potential customers a lot of confidence. They want to feel like you’re working to make this the best game ever. If you don’t patch it regularly, and especially when it’s necessary, kiss a lot of possible purchases goodbye. Nothing kills enthusiasm more than a buggy game that’s not getting the love it needs. People will talk, negative reviews will get posted. Start working on patches regularly, even if it’s not perfect, people will start talking about that. This exact thing happened to Star Ruler last year. By most accounts, buggy and unfinished out of the gate. This was admitted by the developers, as they simply ran out of cash. People took them on their word that they’d keep working on the game and bought it. Since that time, they’ve sold enough to be around a year later with reports of the games’ current state being a massive improvement.

Step 8:Design your game for netbooks and other really low end hardware

This is sort of a given for most Indie developers simply due to the fact that the majority of them are developing low-poly count, reasonably non-hardware-hungry type games. Generally they’re slightly more casual. That being said, some of them aren’t. However, with the explosion of netbooks and now handheld devices, be they smartphone or tablet, I have a ‘gaming platform’ with me at all times, pretty much. The more platforms you can get your game on, the more it will sell.

Step 9: Leverage other people's work

It was very interesting to work in the game industry until about the early 2000’s. We were figuring out new rendering technology, how to solve physics, collision etc. At some point in early 2000 or so these all sort of became solved problems. There really isn’t any value in solving them again (back to point one). You might be really interested in how physics simulations work, but you need to decide if you want to play with a physics engine or make games for a living. There are tons of middleware out there - if it solves a problem for you, use it, instead of wasting your time reinventing the wheel. This is a great lead-in to our next topic.

Step 10: Make it addictive

This is one that can’t always be implemented. And of course, all designers probably want this. Get ‘em hooked on a simple but addictive gameplay element and you’ve pretty much guaranteed yourself a bundle o’ dough. Hell, who woulda thunk a game with a unicorn mascot based around a game that became popular with a daytime TV show (Pachinko) would help get their company sold for over a billion dollars? I probably wouldn't have put money on it if someone had said it that way… but if you play Peggle… then you start to understand why. That crap is addictive like heroin added to your morning cup of coffee. Heck, I've put more hours into Zuma-alikes than I'd care to admit.

And frankly, although I seriously dug Crackdown, the thing that kept me playing for hours and hours on end were those goddamn orbs. I just.. needed…. to… get… one… more…

Addict me, I'm begging you.

From:
http://code4all.nl/dokuwiki/ - **Code 4 All**

Permanent link:
http://code4all.nl/dokuwiki/10_ways_to_improve_your_indie_game_development

Last update: **2014/01/09 15:19**