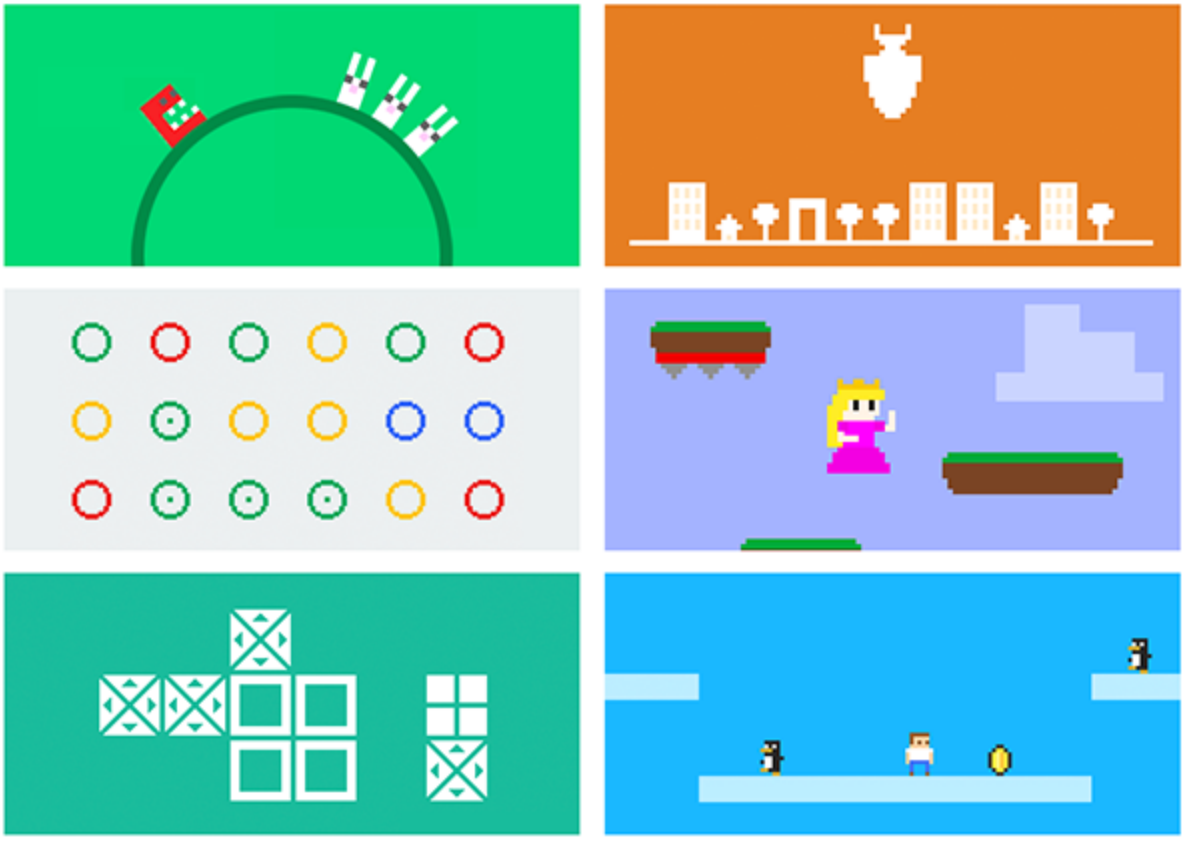
How To Start Making Games

By Thomas Palef for lessmilk.com



Book released in March 2014 - version 1.1

Introduction

As a developer and a gamer I always wanted to make games, but I never actually did. In order to change that I threw myself a public challenge: build a new game every week in HTML5. I now have 10 finished games on my website lessmilk.com, and I plan to keep going because it's a super interesting project.

Now that I've gained some experience with lessmilk, I decided to write this short free ebook to give you some advice on how to start making games.

Start Really Small

I remember that a few years ago I was excited to build my first game ever. My idea was simple: make a clone of the original Zelda, the one that came out on the NES.

After two weeks of work, I had a green character moving around in an empty world. The game was boring, full of bugs, and I had no more motivation to finish it. By wanting to do something big, I ended up with basically nothing.

So if I had only a single piece of advice to give to people interested in making games, it would be to start really small. Look at my first game on lessmilk, it's so simple that it's almost not even a game. But I had fun while making it and I learned a lot. Most importantly, I actually finished it.

Choose The Right Framework

Nowadays there are plenty of frameworks to make games, and that's great. Which framework should you choose? As you may guess, it depends. Here are some tips on how you can find your answer:

- First you should ask yourself some questions about what you want, like:
    - Do you want to make 2D or 3D games?
    - Do you want them to run on console, mobile devices or desktop browsers?
    - Can you afford to pay for the framework?

- Once you answered all these questions, start reading about the frameworks on the internet and choose 2 or 3 that appear to best fit your needs.
- Then write an extremely simple game with each one of them.
- And finally keep the framework that you feel the most comfortable with.

It's a process that takes time, but it's worth it. Why? Because if you get to the point where you realize that you picked the wrong framework, much of what you learned previously will go to waste.

Graphics And Sounds

For me, making the graphics and sounds for my games was the scary part. They both have a super important role in a video game, and I have no knowledge on how to do any of those things. So what can you do?

The good news is that you don't need to be a designer nor a musician to make a good game. There are plenty of resources available online that you can use:

- For sound, there's an awesome tool called Bfxr. If you know how to press a button, then you can make sound effects for your games.
- For graphics, there are plenty of free sprites available for you, like those on the OpenGameArt website.

Of course, nothing stops you from learning how to make your own sounds or graphics. For example I decided to do all of the sprites in my games. Because of this I spend way too much time in Photoshop trying to make decent sprites, but at the same time I'm practicing a new interesting skill.

Keep Iterating

When making a game, you have to think about a lot of things: ideas, gameplay, art, level design, CPU usage, etc. So how can you get everything right? The answer is quite simple: keep iterating.

Instead of trying to do everything at the same time, try to divide the games in much smaller

parts. For example, if you plan to do a platformer like Mario you should do something like this:

- v0.1: just the player (a black square) moving in a super simple level (just a line)
- v0.2: make the level a bit more complex
- v0.3: add some enemies (red squares standing still) to make the player die
- v0.4: add an end to the level, and make a simple second level
- v0.5: tweak the jump physics, and polish the first level
- Etc.

As you can see, each iteration is small and focused on one or two things. At each version you can actually play your game, feel what's wrong, and fix it. And after 10, 20, or 50 iterations, you'll end up with a game you like.

Juicify The Game

One common problem with amateur games is that they often feel wrong, and because of this they are not fun to play. It turns out there is an easy fix to this problem: "juicify the game". Let me explain.

The basic idea is to add animations, transitions, and delays to the game. These are just aesthetic changes, but they will make the game feel more responsive and less boring. This is a vast subject that I cannot cover here, but if you're interested you should definitely watch this 15 minutes Youtube video that shows how juiciness works.

User Testing

Once your game is nearly finished, let some of your friends and family test it. Make sure to be there while they play, because you will most definitely discover that your games has flaws. Here are some examples:

- If people struggle to find a way to start the game, don't say "but it's obvious, it's right there!". Instead, change the game to make it even more obvious.
- If they keep dying on the first level, it's not because they are weak, but because you are now super strong at your own game. So just change the level to make it easier.

- If they keep pressing the wrong keys, maybe you should rethink the controls of your game.

Doing this was eye opening for me. If you really listen to people's feedback, this simple technique will greatly improve the quality of your games.

Publish Your Game

When your game is 100% finished, the next step is to release it so other people can play it. Sounds simple, right? Well, answer this question: how many new games were released today? Probably hundreds. So when your game comes out, you are directly competing against hundreds of games, plus the thousands that already exists! With that much competition, getting people's attention and making them play your own game is not easy.

Here are some of the things you should do in order to get your game noticed: start a blog, be active in game related communities, send press releases, use social medias, etc. As you can expect this is quite time consuming, that's why you should probably start doing all of this way before your game is ready. This part is often overlooked by developers, but if you want your game to be played, you'll have to do it.

Conclusion

Making games may be hard, but it's also an incredibly fun thing to do. Seeing a game slowly taking life is an amazing process. So if this is something you'd like to do, my advice is to go for it!

If you want to learn how to actually build a game from scratch, I wrote a few technical tutorials on my website, click here to read them.