

How to start making pixel art #1

An absolute beginner's guide



Pedro Medeiros

Follow



Jul 21, 2018 · 9 min read

This is a little article on how to start making pixel art, intended for those who are really starting out or never even opened a pixel art software. For now I'll cover only the very basics, how to create a file, setup the canvas size, and work with a color limit.

This article was supported by [Patreon](#)! If you like what I'm doing here, please consider supporting me there :)

Also, this is the part 1 of a series of articles, read the whole series here in the [Pixel Grimoire](#).

Before Starting

Before jumping into pixel art, remember: pixel art is just another art medium, like guache, oil painting, pencil, sculpture or its close cousin mosaic. To make good pixel art you need to be able to make good drawings. In general, this means studying anatomy, perspective, light and shadow, color theory and even art history, as these are all essential for making good pixel art.

Tools

You don't need anything fancy to make good pixel art, and you can do fine even with just a good mouse and free software. My setup includes a small Wacom pen tablet, a good mouse, a good keyboard and my favorite software is **Aseprite**, but you should use whatever your're most comfortable with.

Here's a list of software commonly used for pixel art:

- **[Aseprite](#)**: Great professional editor with many time-saving features (paid)

- **GraphicsGale:** A classic, used in many games. It's a little complex, but full of great features (free)
- **Piskel:** Free online pixel art editor (free)
- **Photoshop:** Powerful image editor not intended to make pixel art but you can set it up to use it (paid)

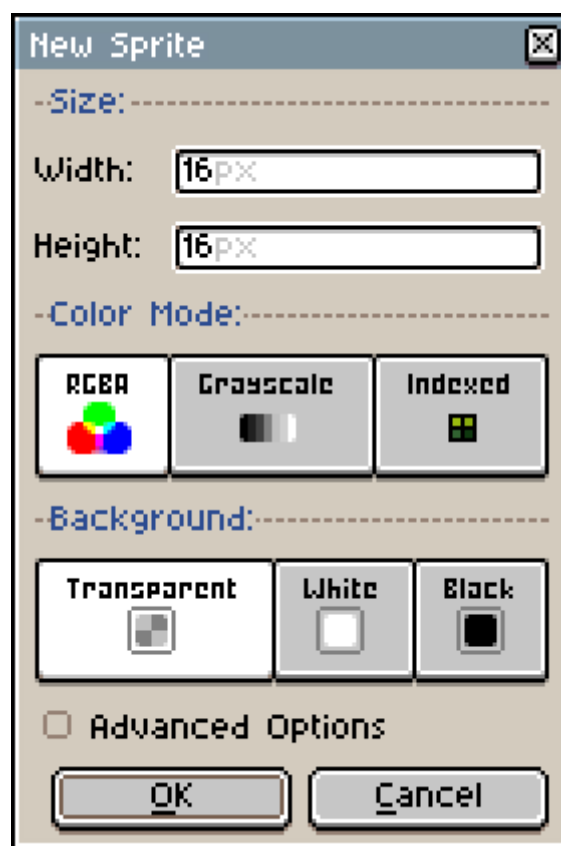
Aseprite

Aseprite is my favorite pixel art software right now. It's incredibly powerful, packed with features and yet simple to use. I chose Aseprite as the software for this tutorial but I'm pretty sure you can adapt it to any other software you use with minimum changes. You can also get the free trial for Aseprite, but keep in mind it won't save your files, which I guess it's OK if you are just practicing.

Making a New File

Just click the "New File..." link in the home screen or go to **File > New File** so we can start drawing.

Let's create a new file. 16 by 16 probably seems a little too small, but I think it's a good starting point. Bigger resolutions can distract you from what you should focus now: understanding the interactions of pixels with their neighbors.



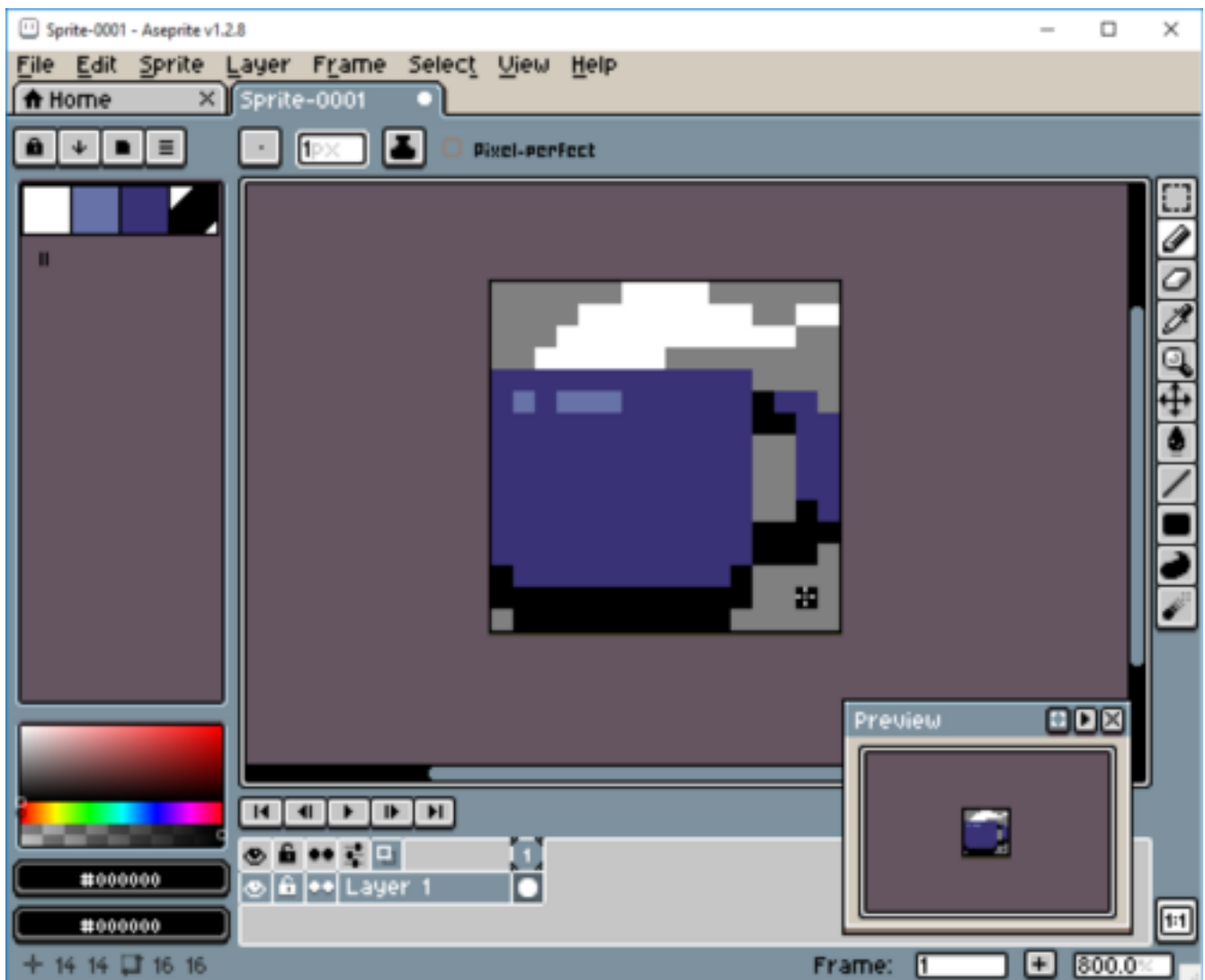
Aseprite 'New Sprite' dialog

You can leave the color mode in **RGBA**, that is the most simple and intuitive for now. Some pixel artists like to work with an **indexed** palette which allows some pretty cool color tricks, but comes with some drawbacks too.

Keep the background **transparent** or **white**, it won't change much for now. Just make sure that Advanced Options is unchecked (but feel free to experiment with them later) and you are good to go!

Let's Draw!

There are lots of toolbars and menus there, but don't worry, we just need a few buttons for now. The main tool is the **Pencil**, that should always be kept with 1 pixel of width, and it will be how we place our pixels on the canvas. Just click the button, or press **B**, and click on the screen to place down a pixel of the selected color.



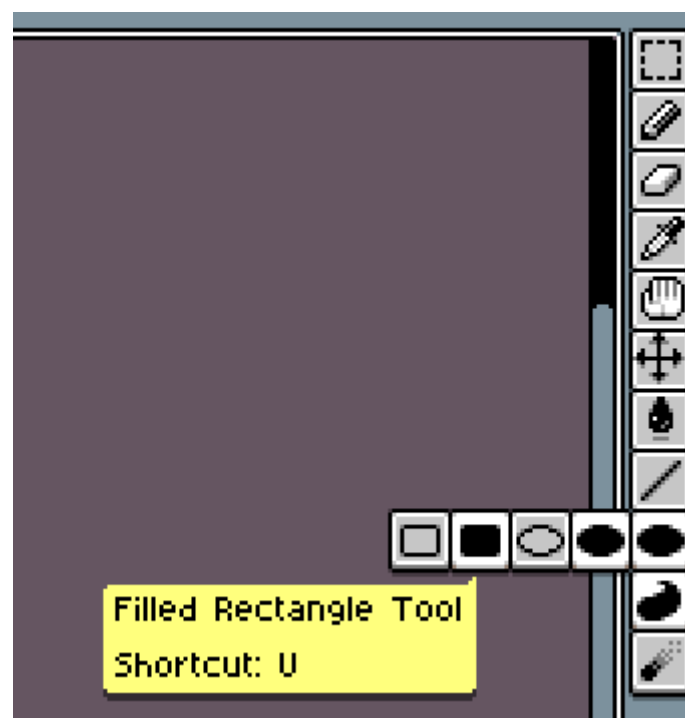
Aseprite workspace

On the left you can see your color palette, with some of the default colors. Let's change those to another, simpler set. Click on the third icon on top of the color palette (*Presets*) and choose **ARQ4** (a really good palette made by [Endesga](#)), that's the one you will be using for your first sprite.

Now, only using the 4 colors on the top left, try drawing a **mug**.

Feel free to use mine as an inspiration, but also try making it unique. If you make a mistake, **alt+click** on an empty area or outside of your drawing and you will "pick" the transparent color and you can use it to erase pixels. Alternatively you can click on the **Eraser** or press **E** to select it.

You will probably notice that working in such a low resolution is very different from regular drawing. Everything needs to be calculated, and each pixel you place is a big choice you need to make. That's the thing you will need to get used to.



You can also experiment with the other buttons in the toolbar. It's worth noticing that some buttons will open more options when pressed. Just avoid the blur tool for now, as it adds more colors and we don't want that yet.

Next, let's make more sprites! Try drawing a **skull**, a **sword** and a **human face**. This time without my pixel art reference. If you feel that the sprites simply won't fit in the canvas, that's absolutely normal, try abstracting something to a single pixel and try again. It's very hard to work with such a low resolution and it feels like a puzzle

sometimes. Here's another article I wrote about working with low resolutions for Kano: [\[link\]](#).

If you want, here's my versions of those sprites, just please make sure to finish yours before looking at them [\[skull, sword and human face\]](#).

This is always a good exercise. If you want to keep practicing, try making even more drawings with those constrains.

Saving Your File

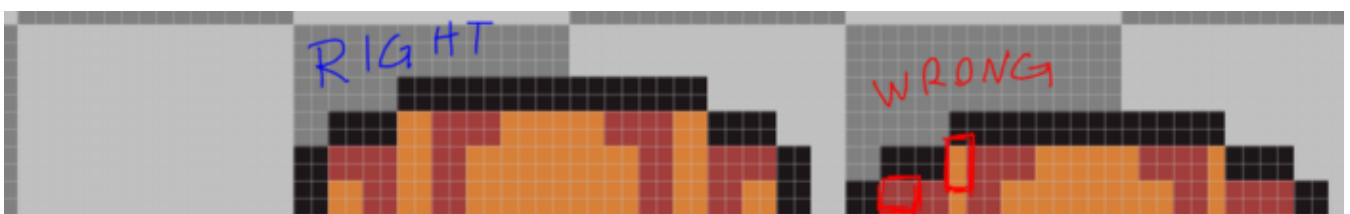
To save your file press **Control+S** (or go to **File>Save As...**), choose a file name and location and just hit save.

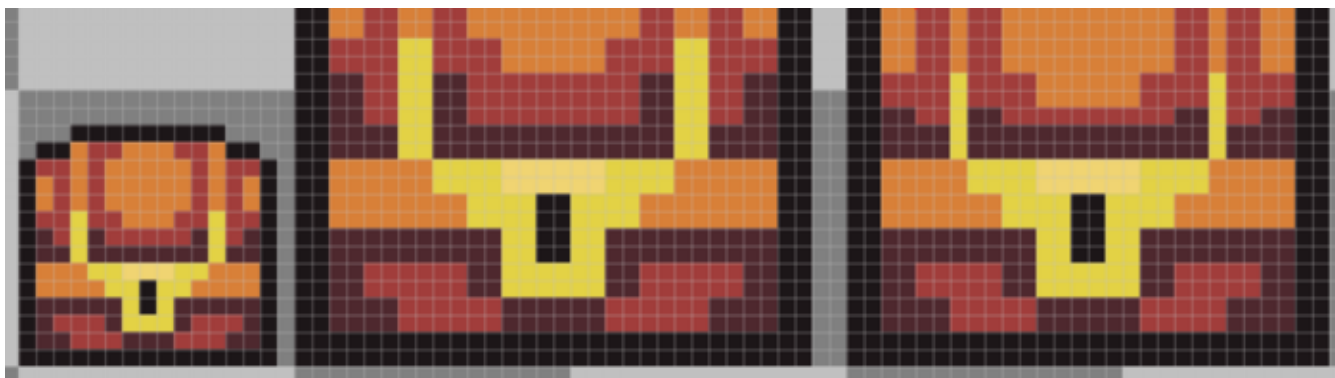
Don't forget that in the trial version of Aseprite saving is disabled!



Aseprite Export File dialog

You will see that Aseprite can save in a variety of formats, but I always recommend keeping a **.ase** version of every file you make. Just like in Photoshop you would keep a **.psd** file. When exporting for web or games, you can use **Control+Alt+Shif+S** or **File>Export**.





Why you should never resize a pixel art partially

Aseprite has this really good **Resize** feature in the export window. It only scales your sprite in round numbers, which is perfect. If you rescale your sprite 107%, for example, it will break pixels everywhere and it will be a mess, but if you scale it 200% each pixel will now be 2 pixels wide and tall, so it will look nice and sharp.

A Bigger Canvas

Now that you got the basics, like creating a new file, saving and drawing into the canvas, let's try drawing on a slightly bigger canvas, **32 by 32** pixels. We'll also use a bigger palette now, try the [AAP-Micro12](#) (by [AdigunPolack](#)). This time we will draw a shovel.

Unlike the 16 by 16 sprite, we can actually fit some outlines here, so let's start with that. Here's my process breakdown:

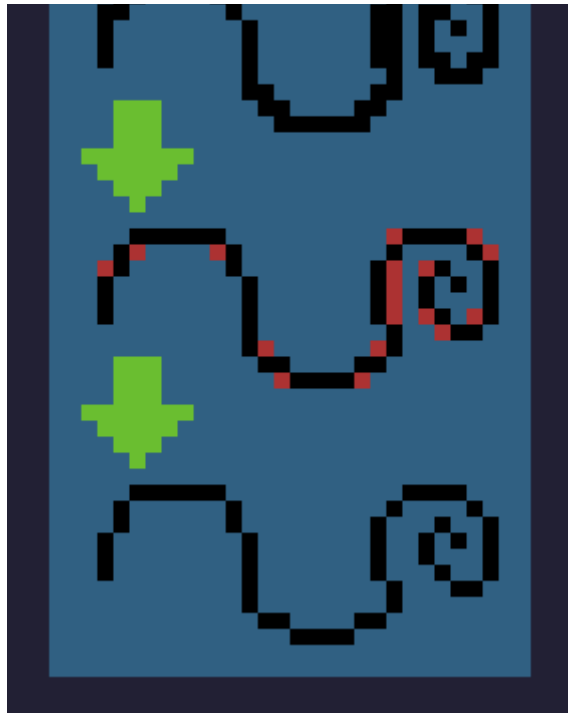
Step 1: Lines



Step 1

This line style is what we call a pixel perfect line, it's only 1 pixel wide and it connects diagonally with other pixels. When making lines like that we avoid unintentional edges, like here:





Aseprite also has a really good feature on the brush settings to do that almost automatically: with your **brush tool** selected, click the **Pixel-perfect checkbox**. Just don't forget to toggle it off when not working with outlines because it will probably annoy you.



Aseprite Pixel perfect algorithm

Step 2: Base colors



Step 2

The good thing about having only so few colors to choose from is that you won't be overwhelmed by too many options. That's why it's much harder to work with a lot of colors, if you have a color in your palette there's no excuse not to use it at its best. Try

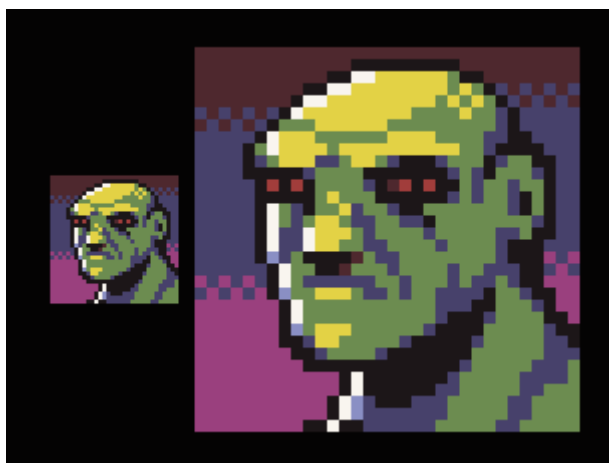
to think of it as a puzzle, experiment a lot, even weird or unusual combinations until you find what you believe is the “best match” for each area.

Step 3: Shading



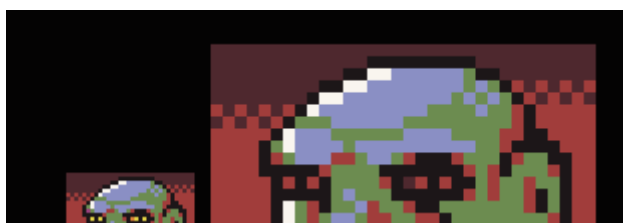
Step 3

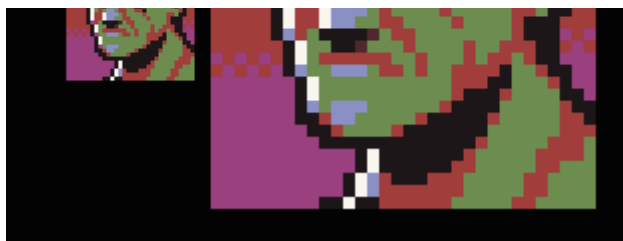
Use your palette to make light and shadow in creative ways. Since you are working with a very restricted palette, you won't have every hue with different brightness, so you will have to improvise.



Improvising shade tones with different hues

In the example on the left I'm using the same palette you are, the **AAP-Mini12**. When I drew this green dude I didn't have any light green color, so I went with the nearest hue I had available, which was yellow. The same thing happened with the shadow, I chose blue because it was the closest dark one. But what if I went the other way? I could get a brighter blue and darker red, right? Well, not really:





Shade tones using inverted hue

It's a cool effect, but clearly there's something wrong. Usually you will want the cold hues to be your shadows and warm hues to be your key light, or they might look weird. This is not a stone-written-rule or anything, there are many exceptions, but when not sure, just go with it.

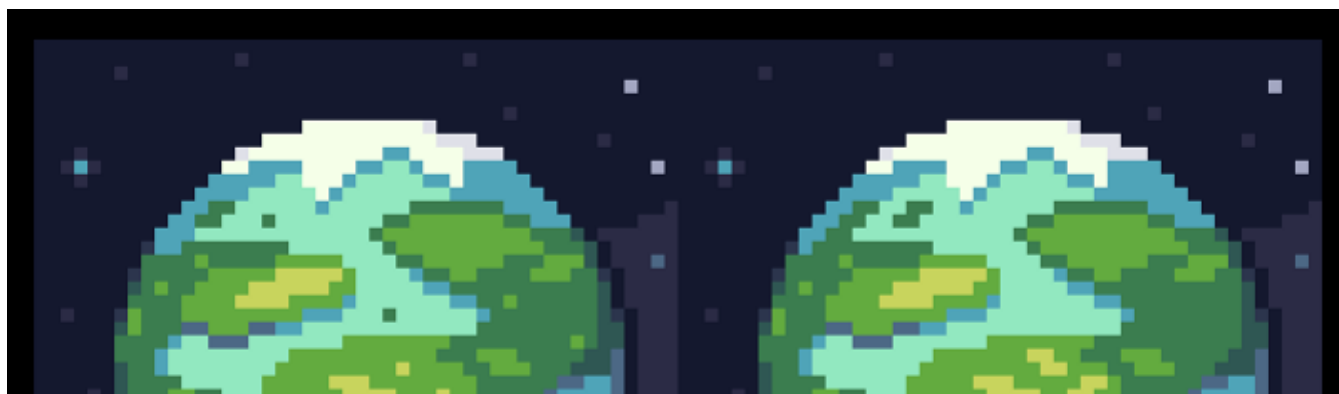
Step 4: Anti-alias and polish

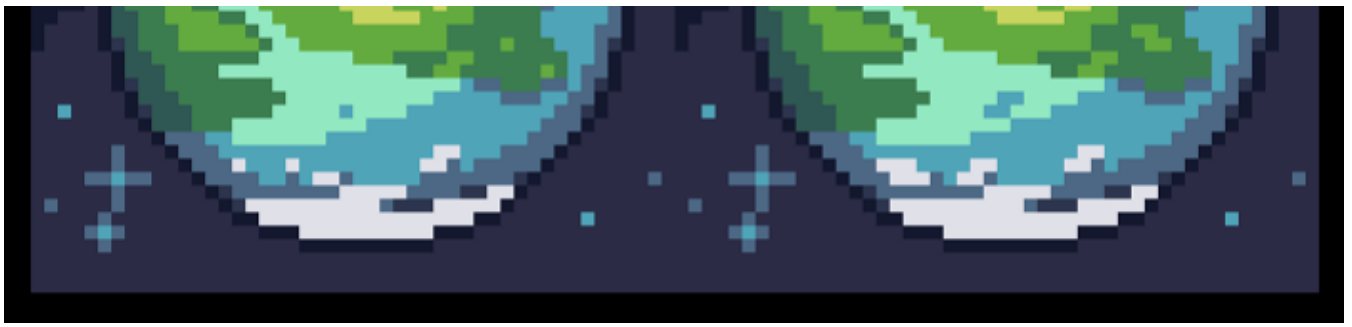


Step 4

This is the part of the drawing where you try to make the pixels a little less “pointy”. Manual anti alias is a complex subject, and we probably will need a whole article to discuss just that, but the theory is, you will use mid tones to simulate “half pixels” and soften the edges. But don't worry too much about this yet, for now focus on making your sprite as readable as possible.

Another good idea in this step is to hunt down some orphan pixels to reduce noise. Orphan pixels are pixels that are not part of a bigger group of pixels of the same color and are not part of the anti-alias, like this:





Removing orphan pixels

You see the little 1-pixel-islands on the left? Those are orphan pixels, as you can see the planet looks much better after we merge those pixels with some other nearby pixels of the same color.

And what about the stars in that example? Well, they are there to prove that orphan pixels are not always bad, those stars work exactly as intended, creating a noise texture and bringing up the contrast in the background.

The idea is not to mindlessly remove orphan pixels, but to through them and ask yourself: does this pixel really need to be alone?

Now What?

Now it's time for you to experiment with more colors and bigger resolutions! But go slowly, maybe 48 by 48 and 16 colors and so on. If you are really starting out I would avoid animation for now and focus on getting comfortable with static images first.

I selected some other pixel art guides that I really like if you want to do some research:

- [Pixel art tutorial by Cure](#)
- [Pixel art tutorial by Derek Yu](#)
- [Pixel art tutorial by Arne](#)

I also make some tutorials about specific topics or aspects of pixel art and game design, you can see them all here:

- [My Patreon page](#)
- [A compiled list of all my tutorials](#)

Keep reading the part 2 [here!](#)