



# PIXEL ART TUTORIAL: BASICS



In this fast-paced tutorial, I show you the basics of making pixel art by walking you through the creation of a **sprite**. Sprites are the images in 2d games that represent the various objects in a game like your player character, monsters, items, etc.

This tutorial is paired with a follow-up tutorial called PIXEL ART: COMMON MISTAKES.

## QUICK LINKS

- Background
- Software
- Other Supplies
- 96 x 96 Sprite
- Choosing a Palette
- Jaggies
- Form and Volume
- Anti-Aliasing
- Selective Outlining
- Dithering
- 32 x 32 Sprite
- File Formats
- Sharing Your Pixel Art

# BACKGROUND

Metal Slug 3 (Arcade). SNK, 2000.

Pixel art, also known as **dot art** in Japan, is a form of digital art where editing is done on the pixel level. It's primarily associated with the graphics of 80s and 90s video games, where commercial artists strained against limited memory and low resolutions to create increasingly eye-catching visuals. These days, it's still popular in games and as an artform in and of itself, despite the possibility of realistic 3d graphics. Why? Well, nostalgia aside, it



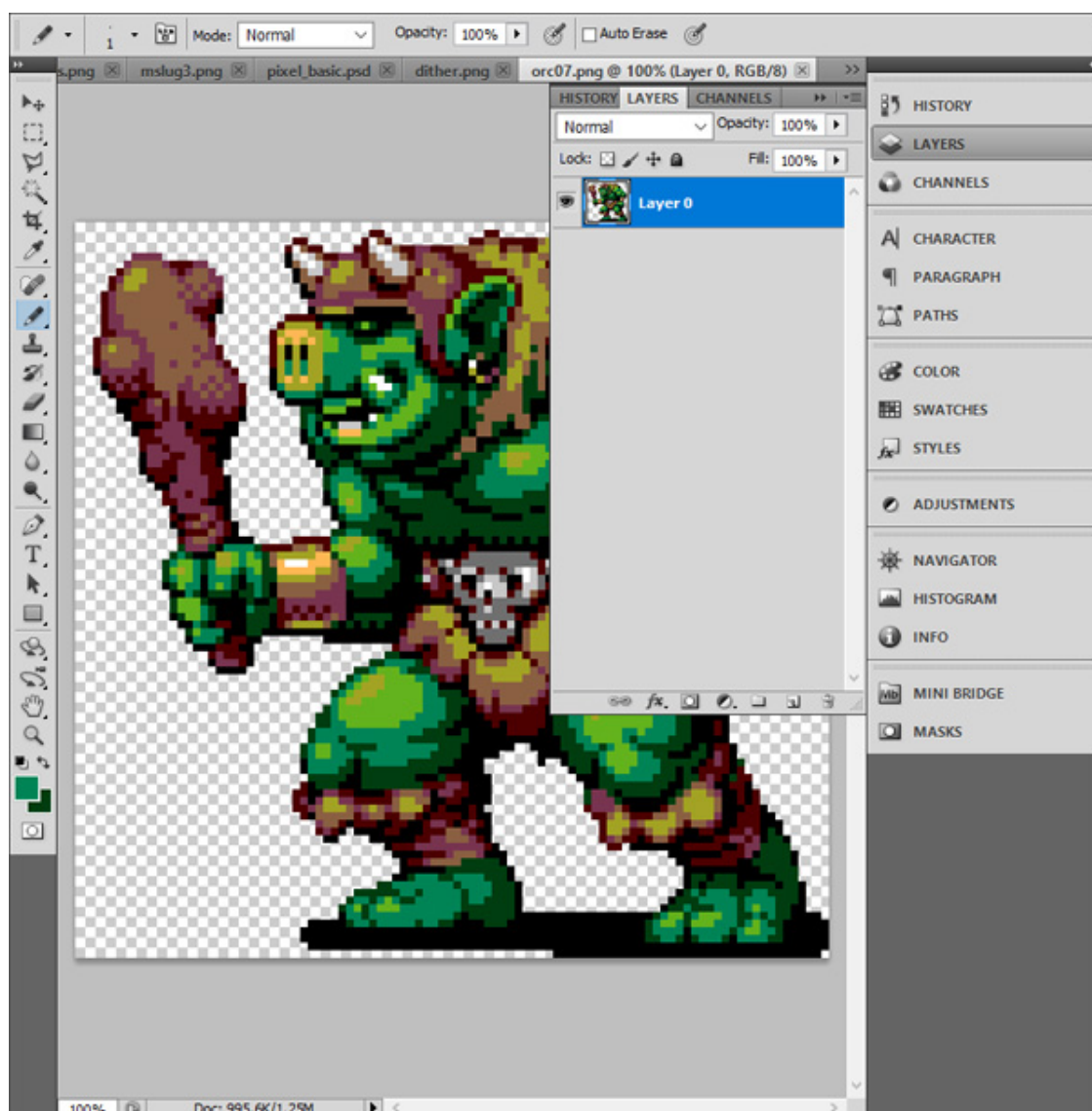
remains a fun and rewarding challenge to create vibrant artwork within such tight constraints. In the same way we admire how a few brushstrokes from a trained hand can represent a form and evoke emotion, so do we admire how a few pixels can combine do to the same.

The barrier to entry for pixel art is also relatively low compared to painted or 3d graphics, making it a nice option for indie game developers seeking to bring their ideas to life. But make no mistake, that in no way implies that it's easy to actually FINISH a game with it. I have seen many an indie Kickstart their pixel art Metroidvania thinking they have a year to finish when in reality it's more like six years. Pixel art at the level most people want to do it is time-consuming and there are very few shortcuts to making it. At least with a 3d model you can rotate it, deform it, move its limbs around, copy animations from one model to another, etc. High level pixel art almost always requires a lot of painstaking pixel placement on every frame.

With that warning out of the way, a little bit about my style: I primarily use pixel art for making video games, and it's from video

games that I draw most of my inspiration. In particular, I'm a fan of the Famicom/NES, 16-bit consoles, and 90s arcade games. My favorite games of that era had pixel art that I would describe as colorful, bold, and clean... but not so clean that it was stiff or minimalistic. That's the style that I modeled my own after, but you could easily apply the ideas and techniques in this tutorial to something completely different. Study a variety of artists and make pixel art what you want it to be!

# SOFTWARE



The basic tools required for pixel art are zoom and the Pencil for pixel placement. Also helpful are line/shape tools, select/move tools, and a Paint Bucket for quick fills. There are many free and paid software options you can use that have these tools. I'll outline some of the most popular ones here (including what I use).

## **PAINT**

If you're on Windows, its built-in paint program is bare bones but has all of the above tools you'd need to make pixel art.

## **PISKEL**

A surprisingly robust pixel art editor that runs in your browser! Can export to PNG or animated GIF, as well as saving locally to your browser. This seems like a great starting option.

## **GRAPHICSGALE**

GraphicsGale is the first standalone editor I remember hearing about that was designed just for pixel art and featured animation tools. Created by a Japanese company called HUMANBALANCE, it became freeware in 2017 and is still widely used despite Aseprite's growing popularity. Unfortunately, it's Windows only.

## **ASEPRITE**

This seems to be the most popular editor available right now. Packed full of features, actively developed, and available for Windows, Mac, and Linux. On top of that, it's open source and can be used for free if compiled from the source code. If you're serious

about making pixel art and don't already have an editor you're attached to, this is probably the way to go.

## GAMEMAKER STUDIO 2 💰💰+

GameMaker Studio 2 is an excellent 2d-focused game-making tool that includes a decent Sprite Editor. If you're interested in making pixel art for your own games, it's very convenient to do it all in the same software. I am currently (in 2019) using it to make UFO 50, a collection of 50 retro games. I mostly use GameMaker's Sprite Editor for sprites and create my tilesets in Photoshop.

## PHOTOSHOP 💰💰💰+

Since Photoshop is an expensive, subscription-based app that's not designed around pixel art, I don't recommend it unless you already have it for painting or image manipulation. It can get the job done for static sprites and pixel illustrations (like the ones I've made for this tutorial), although it's pretty cumbersome compared to focused apps like GraphicsGale or Aseprite..

# OTHER SUPPLIES

My pixel art setup. Very black, I'm now noticing...

## DRAWING TABLET 💰💰+

I highly recommend a drawing tablet for any kind of digital artwork



in order to prevent repetitive stress injuries to your wrists. RSI is much easier to prevent than to fix. Once they start feeling sore, you're already headed downhill (my days of drawing with a mouse have made it hard to play any games that require mashing buttons). So start taking care of yourself early - it'll be worth it! I'm currently using a small Wacom Intuos Pro.

## WRIST GUARD

If getting a tablet is not possible, at the very least get a wrist guard. My favorite is the Mueller Green Fitted Wrist Brace. I've found other brands to either be uncomfortably tight or not supportive enough for me. You can order wrist guards easily online.



# THE 96 X 96 SPRITE



Final Fight (Arcade). Capcom, 1989. ([Source](#))

Let's begin! For this tutorial, we're going to start by creating a 96x96 pixel character sprite. I've chosen to make an orc, but feel free to pick something else! I've placed my finished orc in a screenshot of Final Fight above to give you a sense of scale - this is a large sprite for most retro games (the screenshot is 384x224).

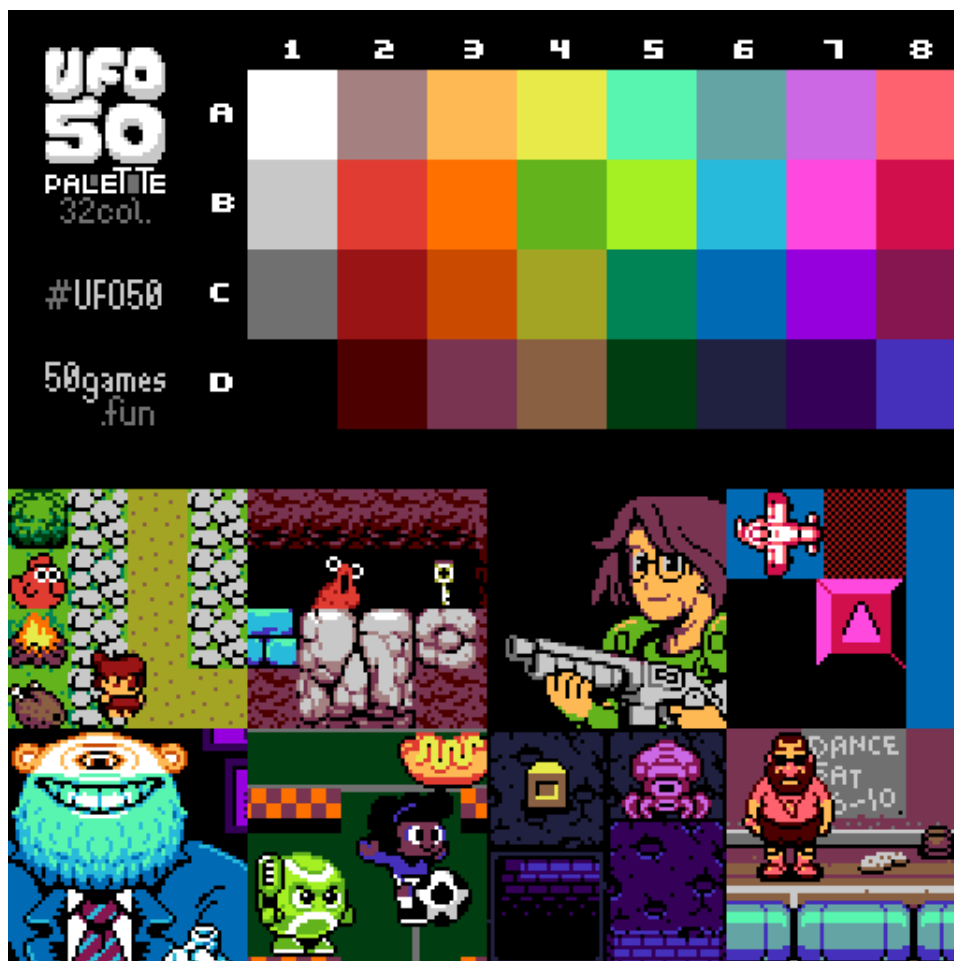
The reason we're starting with such a large sprite is that I find it easier to show off the techniques we're learning. Pixelling larger sprites also feels more analogous to traditional artforms like drawing or painting, which might be more familiar to you. After we get the basic tools under our belt we can start working smaller.

## 1. CHOOSING A PALETTE

Pixel art is defined by its constraints. A pixel has much more meaning in pixel art than other digital mediums and the same is true for colors, which you ultimately want to constrain, as well.



So yes,  
a color  
palette  
is



important and helps define your style. BUT, for beginning pixel artists, I think it's best to put any theorizing about palettes aside and just pick an existing one (or even a few colors at random) so that you can start pixelling. One nice thing about pixel art is that it's very easy to swap palettes at any point, so there's no need to let this decision paralyze you before you start putting some dots down.

For this tutorial, I'll be using the 32-color palette we created for UFO 50. 32 colors is a popular choice for pixel art palettes, but 16 colors is also common. This particular palette was designed for a fictional console that would have lied somewhere between a Famicom and a PC Engine. You're welcome to use it freely to quickly bypass this step! (Or not! This tutorial is not dependent on the palette at all.)

## 2. A CRUDE OUTLINE

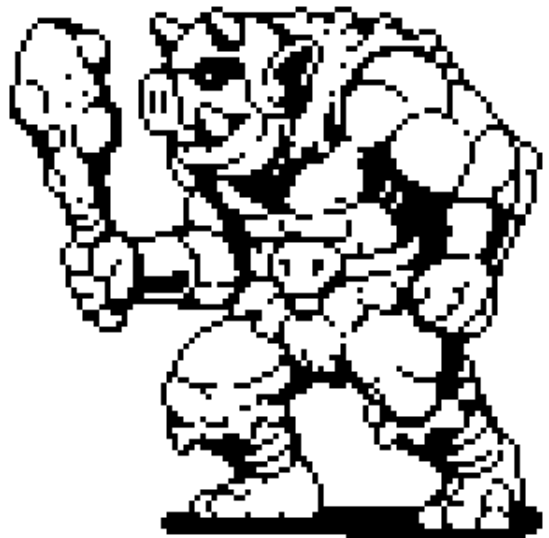
We'll start our sprite by dragging the Pencil tool around, drawing a sketch the same way we'd draw one with pen and paper. There is definitely an overlap between pixel art and traditional art, especially with larger sprites like this one.



From what I've noticed, strong pixel artists are at least pretty good at drawing and vice versa. So it never hurts to improve your drawing skills, as well.

### 3. CLEANING UP THE OUTLINE

Next, we're going to clean up the outline by removing stray pixels and reducing every line to a single pixel in thickness. But which pixels do we remove, exactly? To answer that, we need to learn about pixel lines and "jaggies".



### JAGGIES

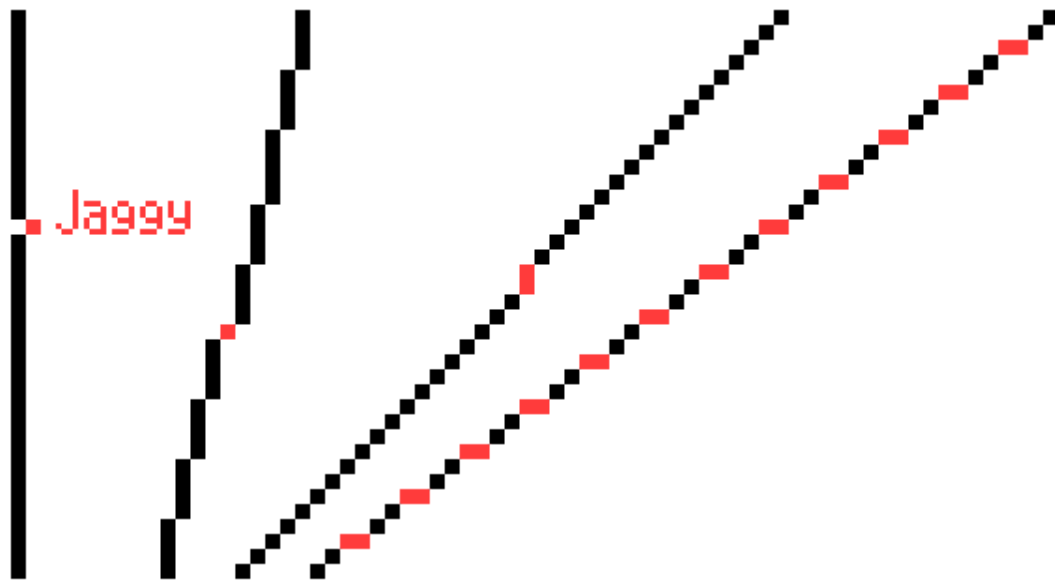
There are two basic lines that we need to learn how to make in pixel art: straight and curved. With a pen and paper, this is

mostly an issue of muscle control, but we're working with little blocks of color, which creates a new kind of challenge.

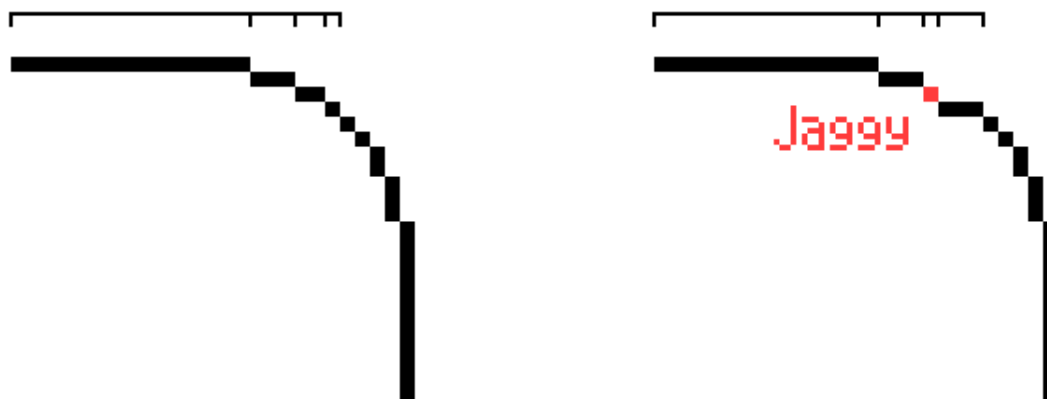
The key to making nice pixel lines is reducing the number of **jaggies**: single pixels or small segments of pixels that break up the consistency of a line. Since a single pixel in pixel art has a great impact on the overall image, jaggies can be an eyesore. Imagine drawing a straight line on a piece of paper when all of a sudden someone slams the table - that little uncontrolled squiggle is sort of what a jaggy in pixel art can feel like.

Let's look at some examples:

## STRAIGHT LINES



## CURVED LINES

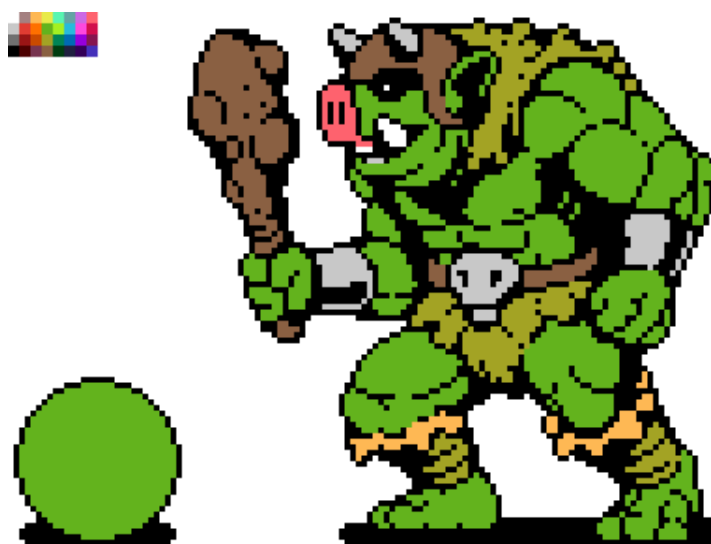


With curved lines, jaggies crop up when the length of the line segments don't grow or shrink in a consistent manner.

At this point, you probably think jaggies are worse than stepping in gum, but in actuality, it's impossible to avoid them entirely unless your pixel art is made only of the simplest shapes. Any of your favorite retro games will have jaggies. The goal is simply to minimize them while expressing what you want to express.

## 4. APPLYING THE FIRST COLORS

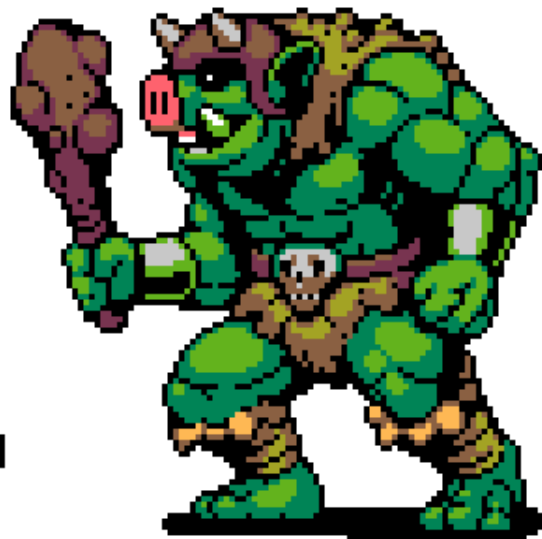
With your Paint Bucket or some other fill tool, color your character in! A palette will make this part simpler, and if your paint software doesn't support palettes, you can always paste your palette into the image itself (as I've done here) and select colors using the Eye Dropper tool.



In the lower-left, I've also introduced a familiar friend, The Ball, to give a quick look at what's going on in each step.

## 5. SHADING

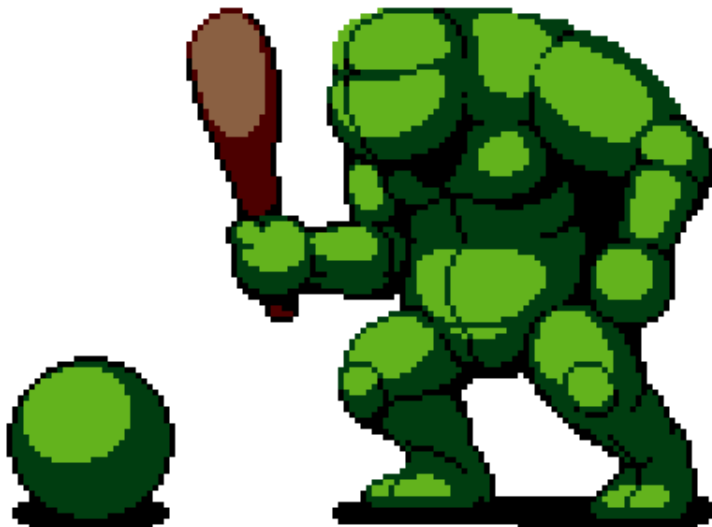
It's time to shade! The basic idea is that we're going to add darker colors (called **shades**) to the sprite to simulate shadow, thereby making the sprite look 3d instead of flat. For this tutorial, let's assume that there



is a single light source above the orc and slightly in front of it, so everything on top and/or in front is bright. We'll add our darker colors to the bottom and back of the orc.

## FORM AND VOLUME

If you're having trouble with this part, you may need to practice thinking about drawings as **forms** with **volume**, instead of simply lines and color. Forms

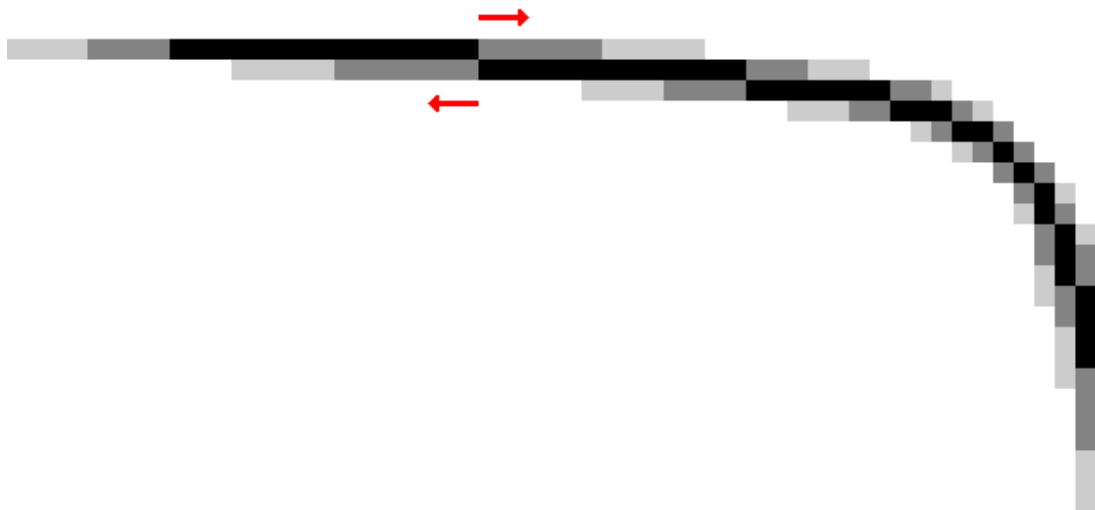


exist in a three-dimensional space and can have volume that fills up that space. By shading, we're bringing out that volume.

It may help you to visualize your character without all of its details and pretend that it's made out of clay instead of pixels. By shading, you're not just adding color - you're sculpting out a form. A well-defined character has details that do not obscure the basic forms - if you squint, a few large clusters of light and dark should still emerge.

## ANTI-ALIASING

Every time I introduce a new shade of color, I do some **anti-aliasing** (also known as **AA**), which is a way to smooth out blocky pixels by putting "in-between" colors at the corners where two line segments meet:



The gray pixels soften the "breaks" in our line. The longer the line segment, the longer the segment of AA we use to soften it.

To the right you can see what AA looks like as applied to our orc's upper arm. I use it to smooth out the lines that define the curvature of his muscles.

Be careful not to anti-alias the outside of a sprite used for a game or anywhere you don't know what color the background

is going to be. If, for example, you anti-alias on a light background then that anti-aliasing will stand out on a dark background.



## 6. SELECTIVE OUTLINING

Up until now, our outline has been pure black, which gives the sprite an overall cartoony look. It's also creating a lot of harsh segmentation. For example, the black lines on the arm are defining the



musculature in an extreme way, making them look less like they are all part of the same body part.

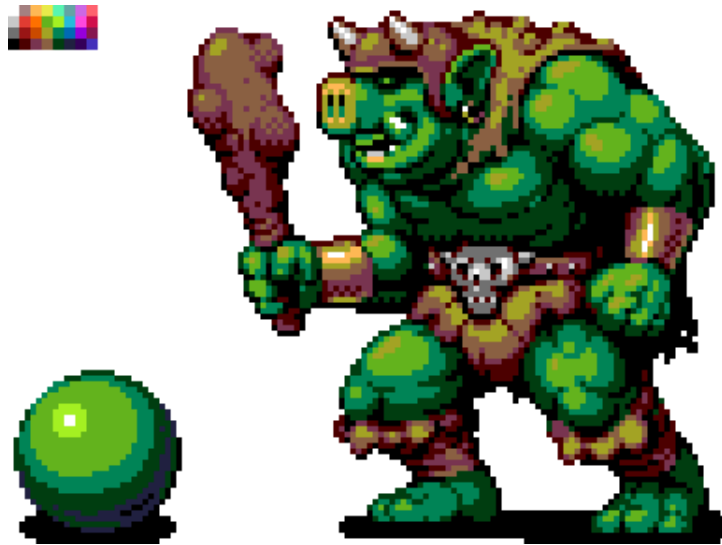
To give the sprite a more naturalistic look and to soften the segmentation (in order to bring out our character's basic form), we can use a technique called **selective outlining** or **selout**. Selout means replacing a lot of the black outline with lighter colors. Toward the top, where the light is hitting our sprite, we will use the lightest colors or, where the sprite meets the negative space, we may remove it entirely. For segmentation (e.g. for muscle, fur texture, etc.), we can use our darker shadow colors instead of pure black.



I've also added another level of even darker shadow to the orc in this step. So there are now three shades of green on our orc's skin. This new shade of green can be used for selout and further anti-aliasing.

## 7. FINAL TOUCHES

At the end, we can add highlights (the brightest spots on our sprite), details (earrings, studs, scars), and continue to make adjustments until we're happy with it (or need to move on, as is often the case!).



A couple other things to try at this stage: flipping your artwork horizontally is a powerful trick in digital artwork that often exposes flaws in proportions and shading. Another trick is to remove the color from your artwork (i.e. set the **saturation** to zero) to see whether your shading still reads well.

## DITHERING

Until now, we've mostly been shading with large, unbroken clusters of darker color. However, there's another technique, called **dithering**, that allows us to bridge two different shades of color without adding a new shade. Take a look at the following example:



At the top is a **gradient** that moves from dark to light, using hundreds of different shades of blue. In the middle, we've reduced the number of colors to 9, but that's still a lot of shades for a single color. It's also created a distracting effect called **banding**, where, because of the thick, uniform bands of color, our eyes begin to focus on the lines where the colors meet instead of the colors themselves.

Finally, at the bottom we've applied dithering, which mitigates the banding effect and only uses 2 colors! The idea is to create **noise** of varying densities to simulate the gradation of color. It's very similar to a technique called "halftone" that's used in printing. Or "stippling" in illustration and comics.

I use dithering sparingly - on the orc I added only a little bit for texture. Some pixel artists don't use dithering at all. Some use

it extensively and make it look quite good. In general, I think it works best on large areas of a single color (take a close look at the sky in the Metal Slug 3 screenshot from above) or in places that we want to look rough or bumpy (like dirt, perhaps). If you like how it looks, experiment with it and find out how to make it work best for you!

If you want to see dithering used extensively and done well, study the games of the Bitmap Brothers, a UK game studio from the 80s, or the games on the PC-98, a Japanese computer (please note that many PC-98 titles are NSFW):



Kakyusei (PC-98). Elf, 1996. (Source) There are only 16 colors in this image!

## 8. ONE LAST LOOK



One of the dangers of pixel art is that, due to its constrained, grid-like nature, it is easy to feel like you can get it "just right", and you may find yourself spending an excessive amount of time tweaking your sprites at the end. In some ways it feels like a puzzle to solve, and that can be very addictive. As a result, pixel art tends to attract perfectionists, so please be careful about lingering on a single sprite for too long. In game development, a single static sprite is just one small piece of a very complex arrangement of pieces and it's important not to lose sight of the bigger picture, so to speak.

Even if you aren't making pixel art for games, it's good to be able to say "This is good enough!" and move on. The best way to

improve your skills is to see the entire process from beginning to end as many times as possible, on as many different subjects as possible. At the very least, leaving a piece behind for awhile will let you look at it with fresh eyes!

# THE 32 X 32 SPRITE

We created a large 96x96 sprite first because at that size it still feels like drawing and painting, but with pixels. The smaller a sprite gets, the less your sprite looks like what it's supposed to represent and the more responsibility each individual pixel has.

In Super Mario Bros., Mario's eye is just two pixels stacked on top of each other. So's his ear. And his creator, Shigeru Miyamoto, explained that the reason why he has a mustache is because they needed it to distinguish his nose from the rest of his face. So one of Mario's most iconic features was not just a character design choice but also a pragmatic one! Proving the old adage that necessity is the mother of invention... and giving us further insight into why pixel art is so interesting.

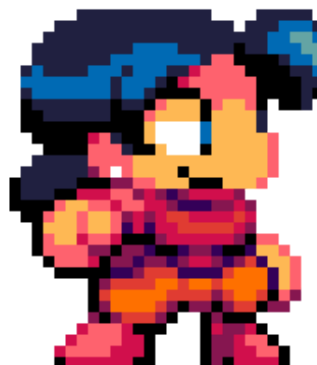
With all that in mind, the basic steps we'll take to create a 32x32 sprite are actually pretty similar to the 96x96 sprite: sketch, color, shade, and then polish. For the initial sketch however, I often use colored shapes instead of drawing the outline because at this size, color plays a larger role in defining the character than outlines. If we look at Mario again, he doesn't have an outline at all! And it's not just his mustache that's putting in work - his sideburns define his ears, his sleeves define his hands, and his overalls more or less make his entire body intelligible.



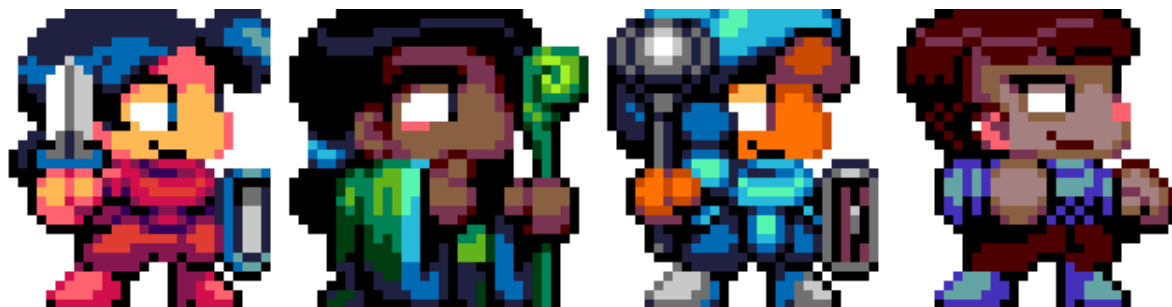
Making  
small  
sprites is  
about

making compromises. If you add an outline around something, you may lose the room to shade it. If your character has well defined arms and legs, the head will probably need to be smaller to make room for them. By using color, selout, and anti-aliasing effectively, however, you can make your canvas feel larger than it actually is.

For small sprites, I tend to favor **chibi** (or **super-deformed**) designs where the characters are cute and have large heads and eyes. It feels like a great way to create expressive characters in a limited space (it's also an appealing art style regardless). But perhaps you're more interested in bringing out a character's



mobility or brute strength instead, in which case you may choose to focus less on the head in favor of a more powerful-looking body. Ultimately, it's up to your preferences and your project!



The full party assembled!

## FILE FORMATS



Enough to send chills up any pixel artist's spine.

The above is what will happen if you save your artwork as a **JPG**, a **lossy** file format. What that means is that data is actually lost when you save it, due to the way the file is **compressed** (to



reduce file size). Practically speaking, your nice, crisp pixel art will end up looking blurry and you won't be able to get its original palette back easily.

The recommended **lossless** file format for static pixel art is PNG. For animations, animated GIFs are the most popular format.

# SHARING YOUR PIXEL ART

Sharing your pixel art on social media is a great way to get feedback and meet other pixel artists (don't forget to use the **#pixelart** hashtag!). Unfortunately, social media websites tend to convert PNGs to JPGs without asking, tarnishing your artwork as it goes public. On top of that, it can be hard to figure out what it was about your image that triggered the conversion!

To help out, here are some tips on how to keep your pixel art crisp on various social media sites. Note that these sites change their algorithms frequently and this section may not always be completely up to date.

## TWITTER

The key to keeping PNGs intact on Twitter is to make sure they're either less than 256 colors or less than 900px on the longest side. ([Source](#)) I would also upscale your images to at least 512x512 px, making sure that you are upscaling at a clean multiple (200% and

not 250%, for example) and preserving hard edges (called "Nearest Neighbor" in Photoshop).

Animated GIFs must be less than 15 MB to post on Twitter. As for quality, the prevailing theory is that they should be at least 800x800 px and looping animations should be looped three times, with the final frame of the GIF displayed at half the length of every other frame. However, it's unclear how necessary all these steps are as Twitter continues to update how they display images. At the very least, I would make sure the animations are at minimum size. ([Source](#))

## INSTAGRAM

As far as I can tell, there is no way to post lossless images to Instagram, but you can improve the look by scaling up your artwork to at least 512 x 512 pixels.

That's the end of this tutorial!

Click on the hand to check out the next tutorial: **PIXEL ART: COMMON MISTAKES**



[Return to Top](#)

Copyright © 2019-2021 Derek Yu. All rights reserved.

Last updated: Jan 28th, 2020.