



### Easy Start



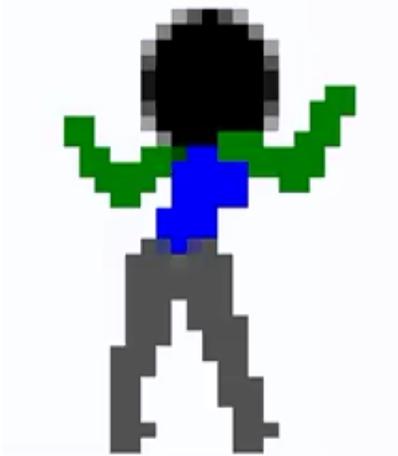
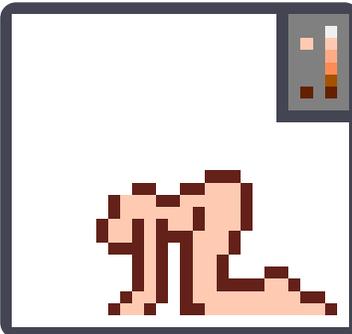
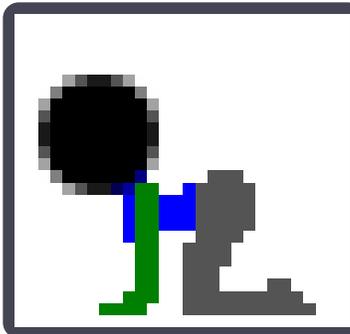
I almost never make *sprites* completely from scratch. Usually, I start with another game character, remove their hair & outfit, then build from there. That makes things much easier since you don't have to figure out the new character's proportions, you'll simply re-use proportions that you already know will work.

### Reduce, Reuse, Rebuild!



You can re-use character bodies by swapping heads & changing outfits. Some sprite artists go even farther, but this has always been enough for me.

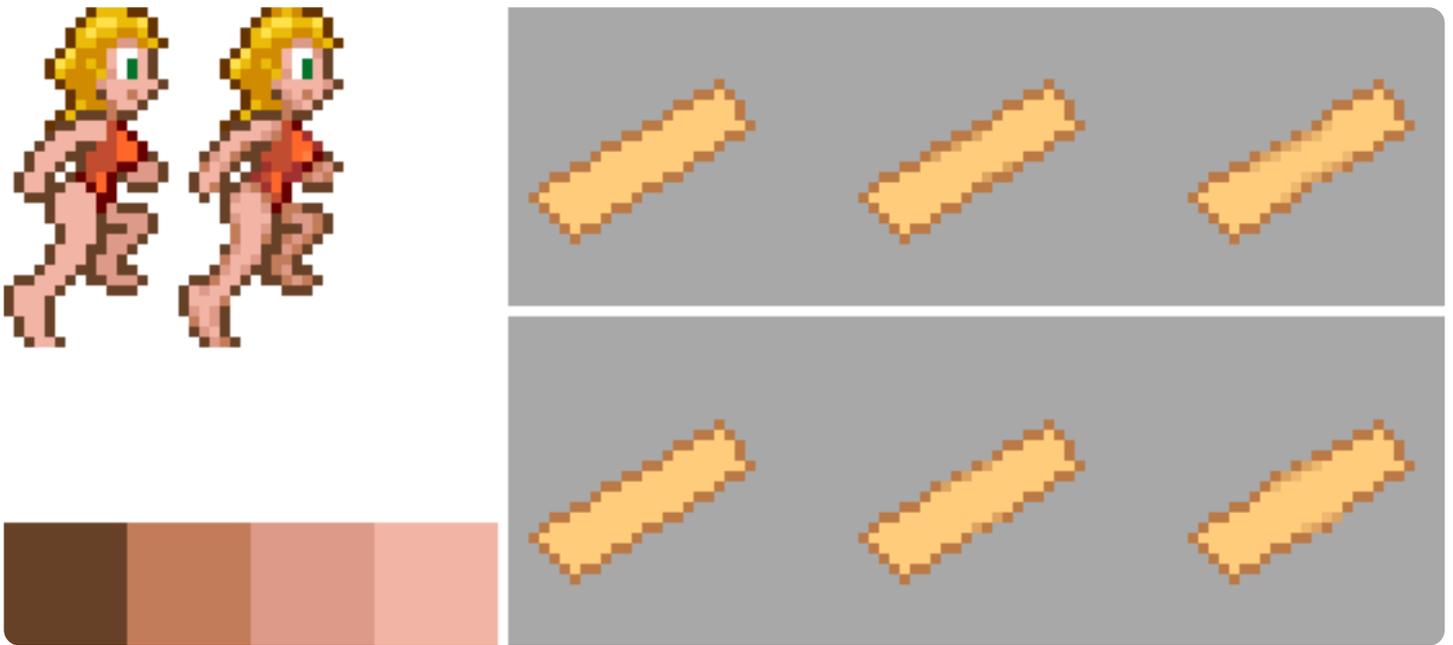
## My Process for New Poses



If I need a completely new pose, I figure out the approximate size of the head and then draw each limb in a single color. I add an outline. If there's enough room, I might add some very basic shading. Then I adjust the body's shape by adding manual anti-aliasing to the edges. I add the character's head. And then I add an outfit... sometimes.

<< In this video, you might notice moments when pixels are flickering back and forth. The video is playing at 8x speed, and I occasionally look at a tiny preview of the sprite and toggle the changes back and forth trying to decide if the changes I just made look better or worse. With sprites, it only matters what they look like at their intended size. (I suppose you could just blur your eyes)

## Anti-aliasing



The goal of anti-aliasing is to describe the various shapes in the sprite down to a subtle level of detail. (It's possible to describe changes in shape that are even smaller than a pixel)

Anti-aliasing is one of the most important skills you can learn because it's the only way to suggest subtle details and subtle changes in shape in an already very tiny image. Use it to clarify how much a shape curves, especially in situations where using a whole pixel would be too much. It can also adjust the apparent thickness of limbs. If you want to show a slight outward bulge, lighten some of the pixels in the outline. If you want to show a slight inward crease or inward curve, slightly darken the pixels along the inside of the outline. Focus on clarifying the shapes first. If you have enough room you can consider adding shading first, and then adding anti-aliasing. I recommend limiting the shades to 4 levels of brightness. This includes the outline. Using fewer than this makes it difficult to adequately describe details. Using more makes the process more cumbersome than it needs to be, without much improvement in quality. Do not add anti-aliasing to the outside of the outline unless you're working with an image format that supports translucent pixels. (Alpha channels)



It's interesting to note that small sprites naturally have a cartoony look because they have limited shading, selectively exaggerated features, and use outlines to distinguish them from the background.

## Outlines & Shading



The purpose of the outline is to provide contrast. It guarantees that the sprite will be visible against any background. Unlike traditional art, a sprite is a moving image, so its background will be changing constantly. The outline's color will generally be a much darker version of the color you're outlining. Be careful with dark shading, unless you're shading flowing cloth, because it tends to hide details. I usually keep the colors pretty bright.

## Proportions



eyebrows start  
1/2 down head

Head = 2/3 of upper half  
Halfway = waist  
Groin = 1/4 from waist

RPG Proportions

Chrono Proportions

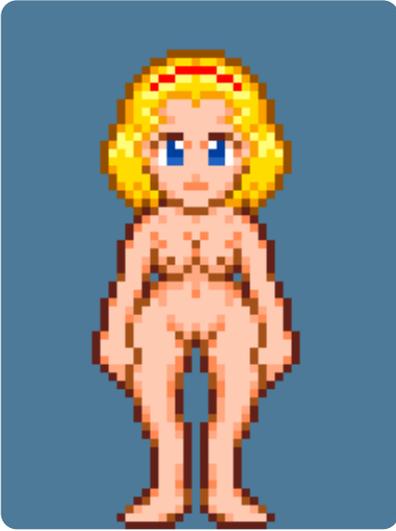
Paying attention to proportions is helpful even with small sprites like these. But the smaller the sprite, the more you need to compromise on realistic proportions. The eyes should still be centered on the head, unless it's tilted up or down. But the size of the head compared to the body will generally be either 1/3 to 1/2 the size of the entire sprite. This is because you need to be able to clearly see the character's eyes, which convey the most expression.

## Why 3D Doesn't Work Well



This is also why creating small sprites in 3D doesn't work too well. It's because the expression gets lost in the blurred pixels. And at that size, you cannot afford the blurry or misplaced pixels you get from 3D. At small sizes, every pixel counts.

## Compression & Anti-aliasing



The image's compression format will affect how you can anti-alias it. I recommend using the PNG file format, which supports multiple types of compression and transparency and is very widely supported. A PNG file can be either lossless with a huge filesize (24-bit,) or palette-based with a tiny filesize (8-bit.) Additionally, PNG files can use either boolean transparency (pixels are either fully opaque or fully invisible,) or they can use alpha channels (which allows pixels to be translucent.)

In Adobe Photoshop, a GIF or an 8-bit PNG cannot have these partially translucent pixels, so you would only be able to place anti-aliasing on the inside of the outline. But if you use any other program, you won't have that limitation. With translucent pixels you're able to display full anti-aliasing both inside and outside of the character's outline. I recommend this because an 8-bit PNG with an alpha channel gives you both flexibility and maximum compression. Photoshop cannot directly create these, but a program called **PNGquant** can convert a 24-bit PNG with an alpha channel into an 8-bit PNG with an alpha channel. (But don't throw away your source files. Photoshop can't open 8-bit alpha PNG files either.)

So unless you're using PNGquant, do not use anti-aliasing on the outside of the character's outline. However you can still imitate anti-aliasing on the inside of the outlines by placing somewhat darker pixels at the corners of the outline when it tries to curve. That way, the inner parts of the outline look a little smoother. You can see what I mean if you look at some of my characters.

## Convenient Preview in Photoshop

When you're creating anti-aliasing manually, it's very important for you to see how it actually looks at the intended size, because it's all about *implying* and *suggesting* the changes in shape. But these impressions are only apparent when you view the sprite at a small size.

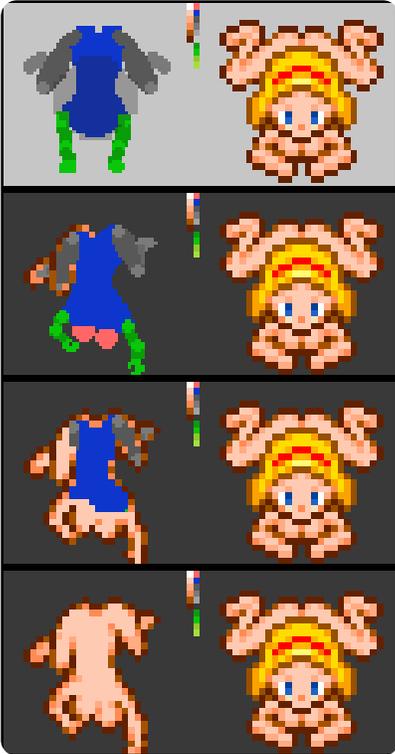


In Photoshop, you can view an image at many different zoom-levels at the same time. To do this, go to the top menu, and select:

Window -> Arrange -> New window for ????

The choice (or choices) at the bottom will duplicate the image's window. These duplicates update in real-time because they're all the same file, so Photoshop won't remind you to save until you try to close the last one for a given file.

## Anti-aliasing and Animation



To anti-alias an animated sprite, you want to start creating the sprite at double-size until you have the silhouette figured out. Then overlay a 2x2 grid, turn on snapping and set your brush to 2px size. Each click will fill a grid tile. This allows you to manually control how the final sprite will look at half-resolution by drawing over the shapes. The trick to anti-aliasing an animated sprite is to follow the same consistent rules when you outline the sprite in each animation frame:

1. When the sprite's silhouette has only 1 pixel in a grid-tile, fill that tile with the darkest color.
2. When the sprite's silhouette fills 2 pixels of a grid-tile, fill that tile with the 2nd darkest color.
3. When the sprite's silhouette fills 3 pixels of a grid-tile, fill that tile with the 3rd darkest color.
4. When the sprite's silhouette fills all 4 pixels of a grid-tile, fill that tile with the lightest color.

But don't fill in the fully-filled grid tiles until after you're done outlining, because trying to fill too soon will distract you from the task of drawing the outline itself. Another rule-of-thumb is I also prefer to only let the two darkest colors touch the background. This means that I outline any lighter colors with dark pixels, and just treat those colors as inner details instead of the outline. After outlining and filling the sprite, I like to add additional detailing, shadows, and smoothing to the inner parts of the outline using the 2nd lightest color. Every animation frame should be outlined and filled from scratch. As long as you follow the same outlining rules you'll end up with nice consistent results, and the subtle details of the original double-size sprite will be mostly preserved. Here's a narrated video demonstrating the whole process.

