

Creatures Sprite breed making tutorial! (ABRIDGED ART ONLY EDITION!)

THIS SHORTENED VERSION OF MY CREATURES BREED MAKING TUTORIAL CONTAINS ALL THE INFORMATION NEEDED TO CREATE ALL THE SPRITES FOR A CREATURES 3/DOCKING STATION ETTIN (plus a bonus tail for them!), INCLUDING EXAMPLE IMAGES AND INFORMATION ON SPRITES AND SPRITE SHEET SIZES. More will be added later for norns and grendels, but the same techniques apply to those aswell, the primary difference is image size.

Things you will need:

Sprite templates. These can be official breed sprites, 3rd party ones, etc.

Some should be supplied with this tutorial, if not I recommend searching eemfoo.org or asking the creatures community! You can also extract your own using programs like Edos or spiritist by using them to export game files to png.

A set of creatures body data of your choice (not in this version of the tutorial but stick to the body data files associated with the breed of the sprite and you're set)

An art program, preferably one with layers and other tools to make your life easier, but mspaint will do at worst.

A program to convert your images into c16 files at the end. Edos and spiritist can extract to and from c16 files. (Preferrably. This covers art but being able to convert it in and out is extremely useful even during the art process)

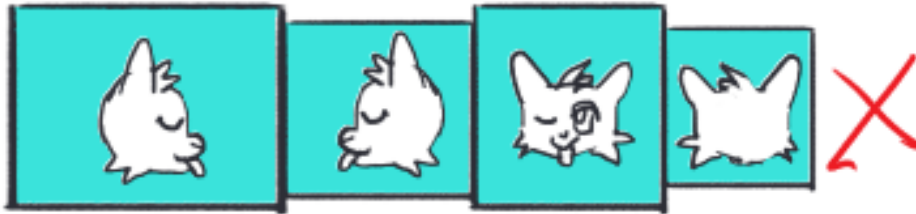
Where to start?

With the sprite art! The other things might sound intimidating or difficult, but sprites are actually 99% of the battle, the rest is largely just forcing things into file formats.

—

HUGE PREFACE before the art happens: Please listen to this it IS important.

If you are making your own sprites and not using a pre-sized template, USE THE OFFICIAL SIZE CONSTRAINTS AND OFFICIAL LIFE STAGES. Mismatched and/or oversized image sizes are a primary cause of random game crashes and jumbled looking/broken creatures. It is not common knowledge but having sprites (heads, for example) of varying image sizes causes bounding box issues when a creature changes expression.



Pictured:

Above, heads all drawn on the same image/canvas size. Imagine a roll of movie or camera film where it switches from image to image.

Below, heads all on different sized images/canvas dimensions. The game engine does not know how to handle the shift in sizes from a physics, placement or body data perspective and the game will crash.

Missing sprites due to not having the correct number for a body part can also do this.



Pictured:

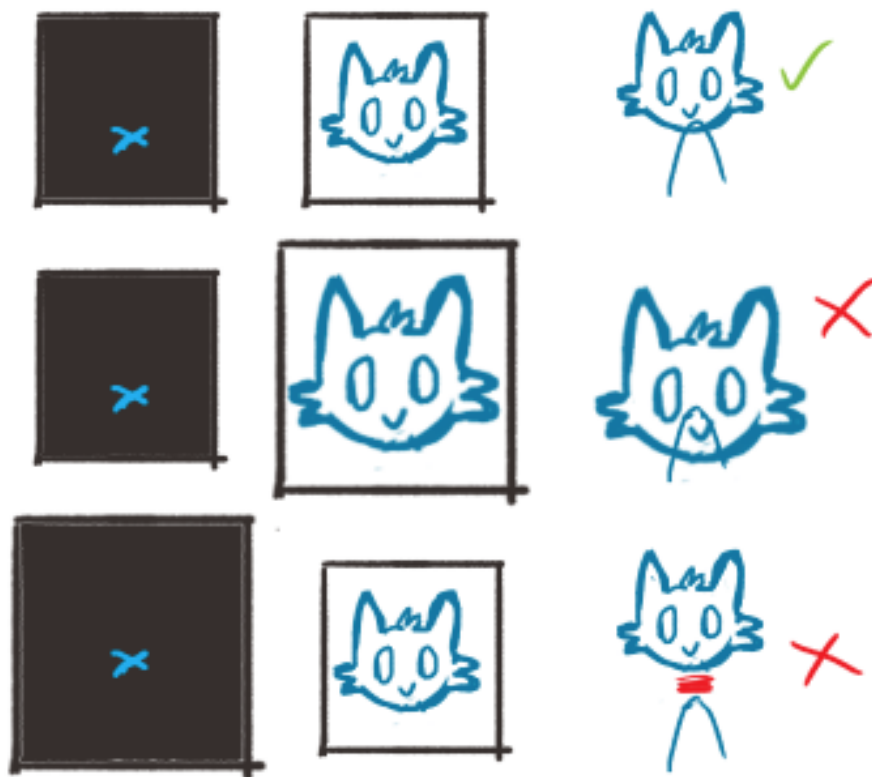
The set of images on top has the correct sprite number in this imaginary example. The set below will cause the game to crash when it tries to call on the missing fourth image in the line-up.

The game engine is not particularly flexible regarding how big a creature can be and the in game environment is designed around fixed sizes. Creatures much bigger than usual are prone to things like breaking door scripts between rooms, falling out of reality, and dying of impact damage and/or starvation from becoming trapped in one place.



Pictured: A simplified example of a creature not fitting safely into the boundaries of a doorway. On a code level, this logic still applies and will result in either the game engine breaking the door in a bid to correct the code error, or a total game crash.

Custom body data is also a rough topic to put it simply. It can be done, and if done extremely carefully and with cross compatibility in mind it can work fine, but it's also a primary cause of glitches and crashes because it functions both as part of appearance, and as part of the collision system. A creature with bigger body data will get stuck in a hypothetical door even if their sprites are small. They will also probably have weird floating body parts- creatures inherit their data and their sprites SEPERATELY so you CANNOT guarantee that custom body data will end up with sprites it plays nice with down the line.



Pictured: A representation of a body data file that writes for the same size as the sprite image, followed by two mismatches that cause misplaced body parts.

I cannot stop you from making entirely custom formats of sprite and body data, but I can do my absolute damndest to discourage you. Most of what you want to do CAN be done with art anyway, from unique limbs to manipulation of perceived size. Extra/custom life stages cause any creature that DOESN'T have this custom life stage to acquire it at that age bracket. This means that you would have to supply this life stage, via art or duplication of other life stage files, for EVERY slot in your game. That's over 100 repetitions. If you make those repetitions, but don't pack them WITH the breed you create, other people will face the exact same problem you did and have to choose between making a hundred and some copies, renames and pastes, or not using the breed. The latter is historically what happens.

'but bigger sprite allocation fixes this' trust me on this. No it doesn't. It is a bandaid at best that gives the engine more leeway but if your creatures head sprite file is twice the size facing forward and it clips into a wall, you better hope you saved really recently and it didn't autosave while closing out because if so you're never getting that world back. I've done a lot of digging into conversions of creatures from previous games into c3/ds and they contain a LOT of this because the c2 engine could handle mis-matched sprite

sizes. From play-testing experience, a lot of c2-to-ds and similar port ups are save file death traps waiting to happen.

ANYWAY GAME SAFETY PSA OVER, ONTO THE ART!

—

You can start on any part of the creature you like! My recommendation would be either getting the worst part done first, the head, or the easiest and arguably most exciting part, the tail! For this tutorial, we're going to start with the head for the adult life stage, because the quicker and easier that can be done, the more likely you are to finish your project and avoid getting discouraged.

(hint: The head sprite files within your game always begin with the letter 'a')

If you're using individual sprites we're only going to worry about the first 16 head images for the moment. On a sprite template sheet, this is the top line of images. This is the head facing right, left, forward and backwards with a neutral expression. This is the primary expression you'll see in your game, and is a great starting point for making the other facial expressions, so we're going to make this first and work from it later.



You'll notice the original sprites have solid pixel edges- this is because of how the game engine works: It cannot do partial transparency in the same way as an art program, and any see-through edges will result in a glitchy area of black pixels. To avoid this, I would recommend either working inside the outline of the official sprite, or making sure to make a hard-pixel-edged selection of your liking and working inside of that instead.

For this example I'm using ettin sprites, and I recommend them as the first breed project you try: The sprites are overall smaller, you don't have to worry about drawing ears at different angles and expressions, and if you don't want to include a tail, you don't have to, as the basic ettin doesn't have one, giving you less work to do.

The trick is to minimise how much we work and focus on at once, as it's very easy to get overwhelmed by all the images needed for a creatures sprite breed. Here we're going to make the very first sprite on the sheet (the right facing sprite angled downwards), use it to help make the rotations, and then flip those over and use them to help us make the left facing sprites. It is extremely helpful if your art program has transform tools for this step, especially ones that let you preserve those hard pixel edges. If you find you're terrible at handling selection tools or pixel based work, one method you can use is to make an all black layer with pixel cutouts of the head shapes, and work under it. The game engine perceives solid black and solid white as telling it to make an area see-through, so black can act as a stand in for your blank areas. Also this is a great time to say DON'T use pure black or white in your sprite art for this game! You'll end up with holes in them.

If you're making art with shadows and highlights in it, and you want to skip some work, make the shading top-down so it works when mirrored left and right. If you're trying to make something as 'canon compliant' as possible, then your light source is bright and comes from the top right casting down diagonally.



Pictured: Simple top-down lighting as an optional shortcut, canon-compliant top-right directional lighting, and what will happen if you flip diagonal lighting to try and save time.

You could make an unshaded head, rotate and flip it, and then add shading if you have a strong grasp on directional shadow and a lot of patience.

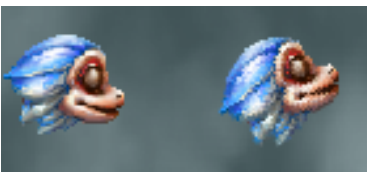
If you have a strong technical knowledge and a program that can use layer effects, you can also apply that here over your work to simulate the directional lighting without having to paint it per-image.

Here's the first head in my example! A contemporary creatures 3 and docking station take on a creature breed called a Bean's Ettin, made for creatures 1. Bean's ettins and the muddled history of whether they're the very first breed of ettin ever made is a topic all it's own.

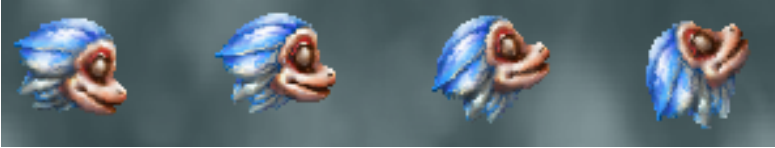


Note I made the eyes open and mouth slightly ajar in this case. This overall outline will help with making later expressions, because it can be coloured in ways that make the eyes and mouth appear more or less open, helping me to change the expression easily later.

When rotating this head to fit the other right-facing angles, expect there to be a bit of blurring and pixel jumbling. There are methods to reduce this, like trying different transformation modes in some programs to see which ones rotate it the cleanest, but no matter what you do you'll probably need to do some touch-ups quickly to make it presentable. Also remember to keep those hard edges no matter what!



pictured rotated before cleaning, and after finishing the turning and cleaning process.



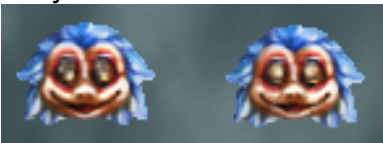
If that felt like a lot, then you now understand exactly why we're going to be exploiting copying and editing images as much as possible instead of drawing them one at a time. For instance, let's cut the work time in half and flip these over and place them over their left-facing examples!



here I added some bare-minimum shading to differentiate them, but this is the entire side-facing part of this step complete now! This is where we make our next new piece of art: The front facing.. face. cx



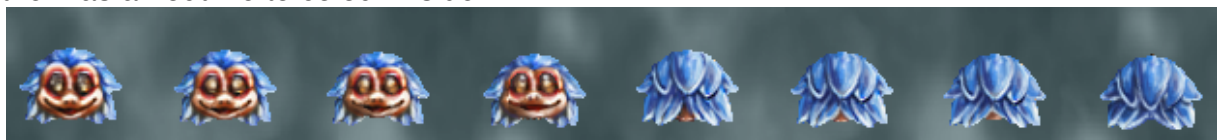
Here's a front-on face for our ettin, looking slightly downward. We're going to do some trickery to use this image as a base for the other front on angles. In this case i'm going to move their eye shine upwards to infer a change in visual direction, move the head's position on the sheet slightly, and grab their snout with the selection tool and move it up a pixel or two. It's the little things that make the difference! You can draw drastic head angle changes individually if you prefer, but we're a long long way from done with even the very first body part, so take every chance you're given to take it easy.



I've also seen use of perspective tools, warp/puppet warp and even placing ears or eyes, noses etc on different layers and moving them around to mimic change in head angle.. Like parallax.



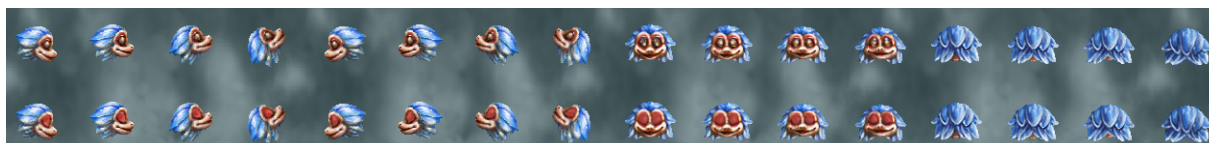
Here's our tilts! I moved the hair around a couple of pixels also to help the appearance of the head movement, but that's not always necessary. We're going to grab these guys and line them up with the backwards facing head tilts they match height with, and use them as an outline to colour inside.



Again, I tweaked a few pixels here and there to show the hair angle changing, but it was much easier while using the front-on sprites as a springboard.



and look at that! That's the biggest part of your entire project done, at least in my opinion. Every head sprite from here is going to be some mutation of these ones. The next 16 images is just the same set but with their eyes shut- you can draw the eyelids closed per expression as we progress, or you can make a layer with eyelids for each angle, and copy it down every time a set of faces needs closed eyes.



There are six expressions that creatures can express using their face sprites: neutral, happy, sad, angry, scared and tired. These have an open and closed eyes variant. If you're able to just paste down eyelids, this means you only have to make the six expression variants, and then duplicate those and paste the closed eyelids over them.



Here's their happy expression! The impression of the open mouth still uses the exact same outline as the default expression does thanks to the leeway given earlier, and similarly the eyes are squinted slightly by raising the cheeks as part of the smile. We're going to follow these basic steps of pasting in the default expression, altering it, and making the closed eyed versions until we've finished all six expressions.



And you've done it! These are all the head sprites you're going to be required to make for the entire breed- unless you want the two biological sexes to look significantly different. You can make separate ones for all life stages, but even official breeds don't do more than make minor tweaks to the adult head sprites and scale them down for the younger stages. The easiest edit is the old age sprites, which are generally a partially desaturated version of the adult sprites. I tend to reduce the saturation by about 30% out of preference, but how they age is entirely up to you! For the head sprite sizes, you can either rescale the entire sheet at once, or rescale the sprites individually. I do the former because it's quicker and easier, but I've included the sizes for all methods of rescaling in the attached sprite size document. Be sure to double check that you're using the scaling sizes relevant to the kind of creature you're making! Ettin sprites have different sizes to grendels and norns etc.

Once you have these sprites done, you can convert them into sprites now and folder them safely for use ingame, or finish all your sprite body parts and then convert, it's up to you! We'll cover sprite conversion in brief later on.

Next up is the body sprite (fittingly, body part 'b'). This is the next largest set of sprites, and it gets increasingly easy from here. To make this as easy as possible, we're going to be grabbing the female adult body sprites and working on the -last- 16 images of the set. This is the bottom row of the sprite sheet and is the body during the last stage of pregnancy. You're probably wondering why we're doing this one first, and it's because

it's much easier to crop away from it to make the previous stages of the body sprite than it is to draw the increasing body size every stage from the beginning.



Pictured: Pregnant female ettin sprites at all angles.

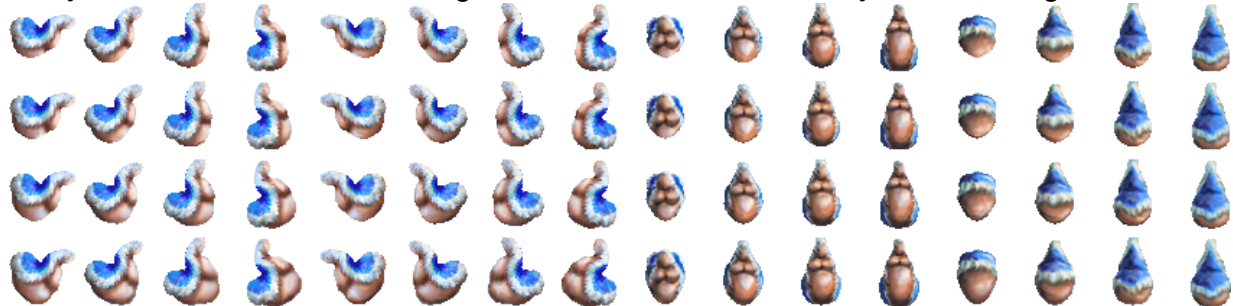
These angles work the same way that the creature heads do: Four facing right, then four left, forward and back. We can do the exact same thing we did before of making the first image, rotating, cleaning and flipping until this row of 16 images is filled out.



Here's our ridiculous first sprite, followed by their equally ludicrous set of tilts.



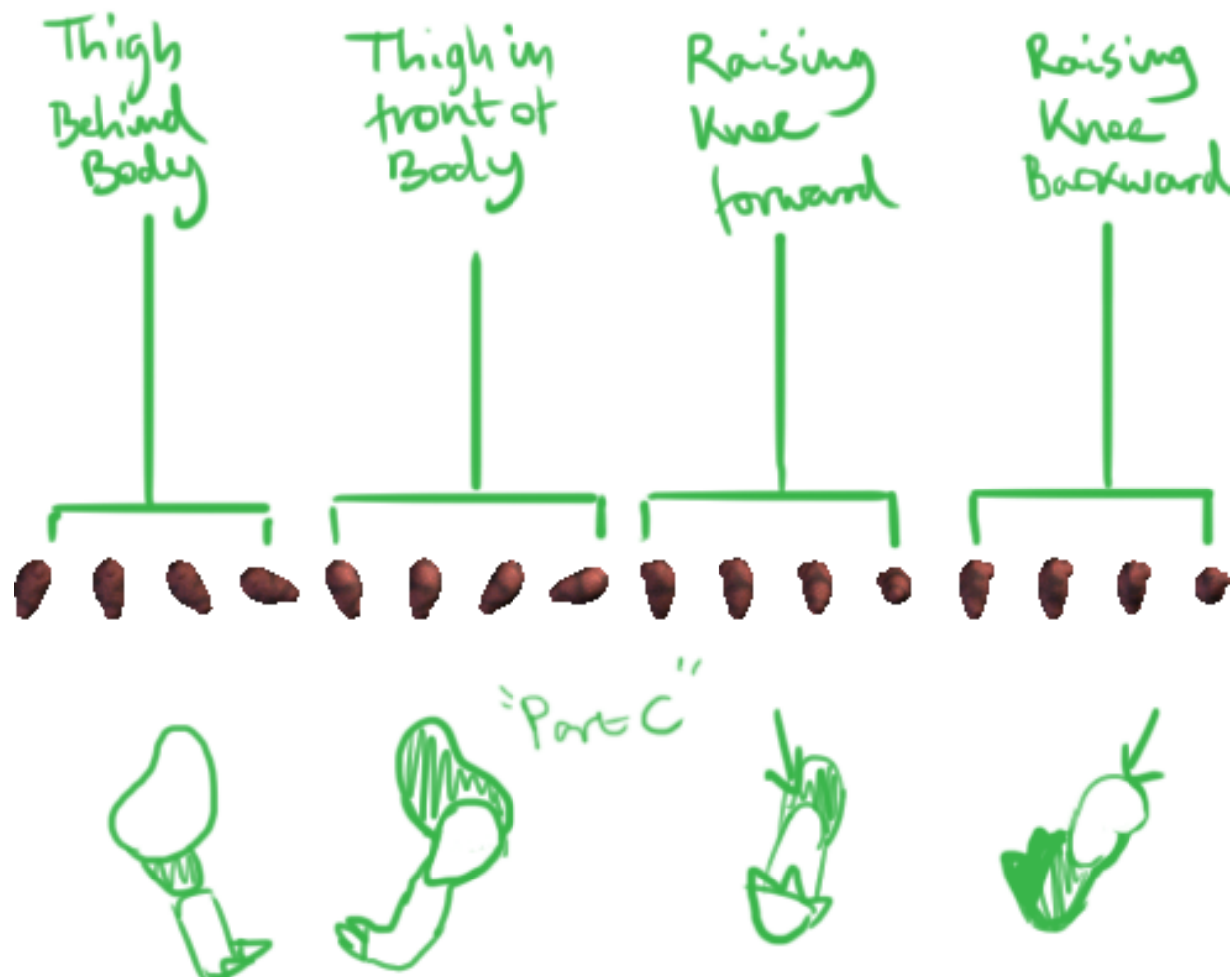
Pasting these over the earlier body sprites means all you have to do is select and cut away the excess, and do shading tweaks if needed. Here's my results doing this:



These sprites, although they include the ones used during pregnancy, are also perfectly acceptable to use for male creatures, as missing sprites can cause ingame issues. You could copy down the top line into every other body sprite line, but it helps to leave the pregnancy stages in as they can be used to diagnose health issues: A male creature showing engorged sprites is producing female pregnancy hormones, for example.

From here on out, the actual images are going to be quick and easy to make, but equally easy to trip up over: Many of these are small vague looking ovals that represent the fronts and back of limb chunks. They're very easy to confuse for one another so I'll mostly be going over which is which while we make these.

In alphabetical order, the next part of the creature's body is part c, which is the left thigh as we view them. These tilt the exact same way that the body does- right, left, forward and back, with the added dimension of paying attention to whether the thigh is the one facing the player/viewer, and keeping track of knees and butts.



Pictured: diagrams that add shading to show when a body part is behind another one, and if they're front and back.

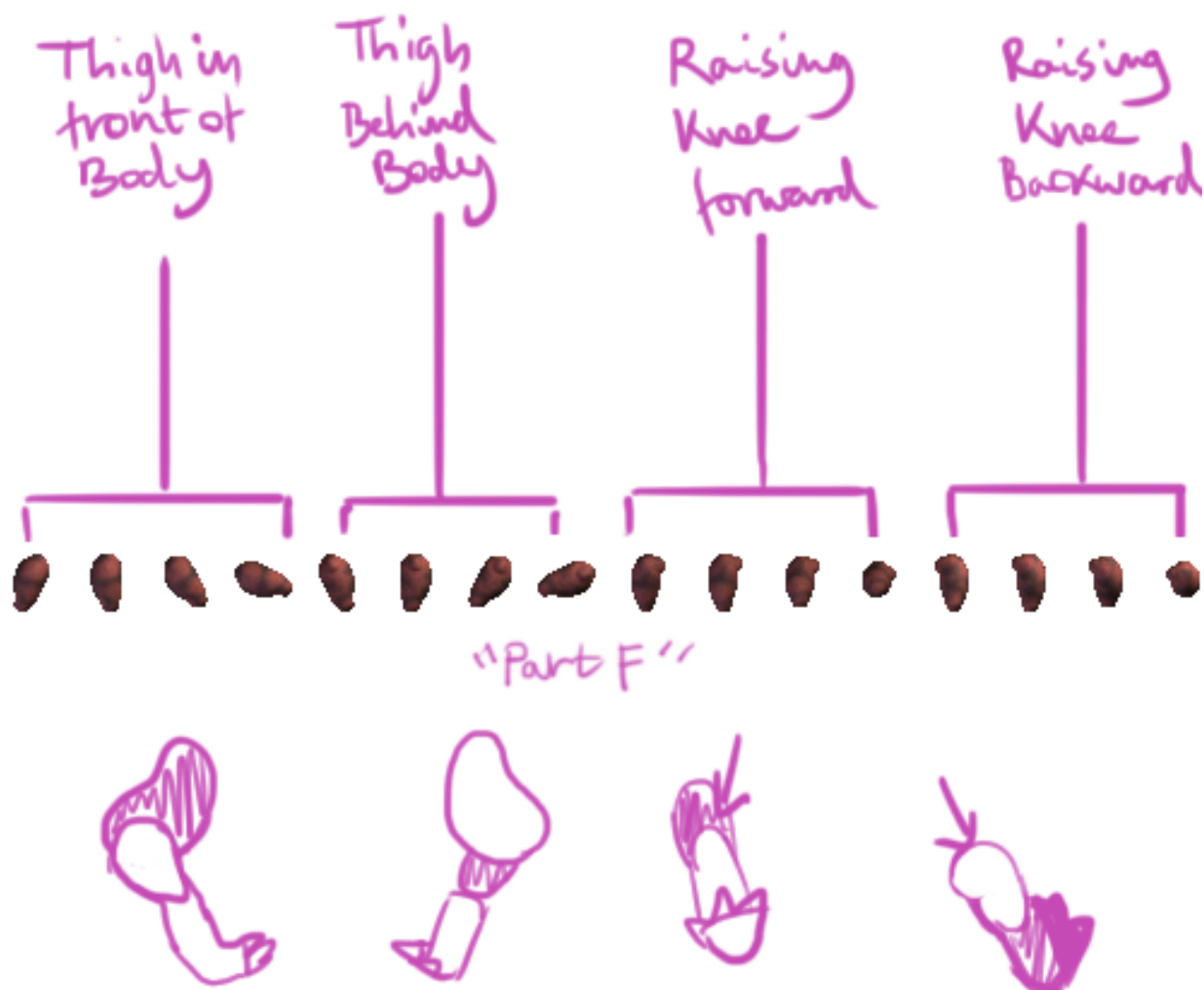
The first four images are of the inner thigh. If your creature has a light belly, their light markings might show on this side of the leg as well, for example. The official sprites make these sprites slightly darker than the next four to infer depth of field. It's not strictly necessary if you're confused by it, but it's an optional touch to add lighting detail! In turn, the next four are in fact not the exact same images flipped, but the leg viewed from the outside. It's a cringe but very straight forward description to say that if you're not sure what to visualise: This is the part the cutie mark goes on with those ponies in that one franchise.

Here's the thigh sprites I've got for the bean's ettins. For these I've opted to have some of their belly skin travel onto the inner thigh area, and some of their back coloration flowing down onto their outer leg fur. Using the size and shape of the blue patch to show angle was useful, but not really necessary.

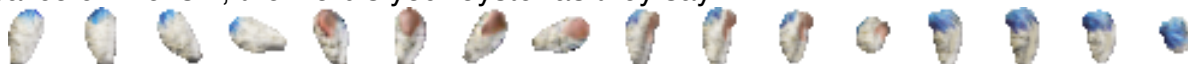


From here you can do one of two equally valid paths: You can either continue down the

alphabet of body images to part 'd' which is the left shin, or jump forward to make the other thigh of your creature which is part 'f'. To show the difference between the two thigh sprites, I'm going to quickly go over c vs f.



As you can see, for reasons I can't really explain, part 'f' goes in a different order to part c, in that the thigh that appears above the body is first, followed by the one obscured behind the body (the inner thigh one). The rest of the opposing pairs of body parts follow this pattern too, and I'll make sure to point out the confusing ones along the way. For these thigh images, if you haven't been worrying about directional shading, you can flip, cut and paste the images from 'c' to fill out part 'f'. I'll be doing the same but adjusting for light direction- If you want to go even harder and make both limbs unique, this is your chance to give your creature asymmetry: Things like cow splotches, freckles, calico/chimerism, the word's your oyster as they say!



Handily you can look at the placements of blue, white and tan on these example sprites to see the change in arrangement on this second thigh.

After the thighs is the calf areas (or lower legs if you prefer). They follow a similar pattern, of course, and are just as confusing to look at- This is piece 'd'.

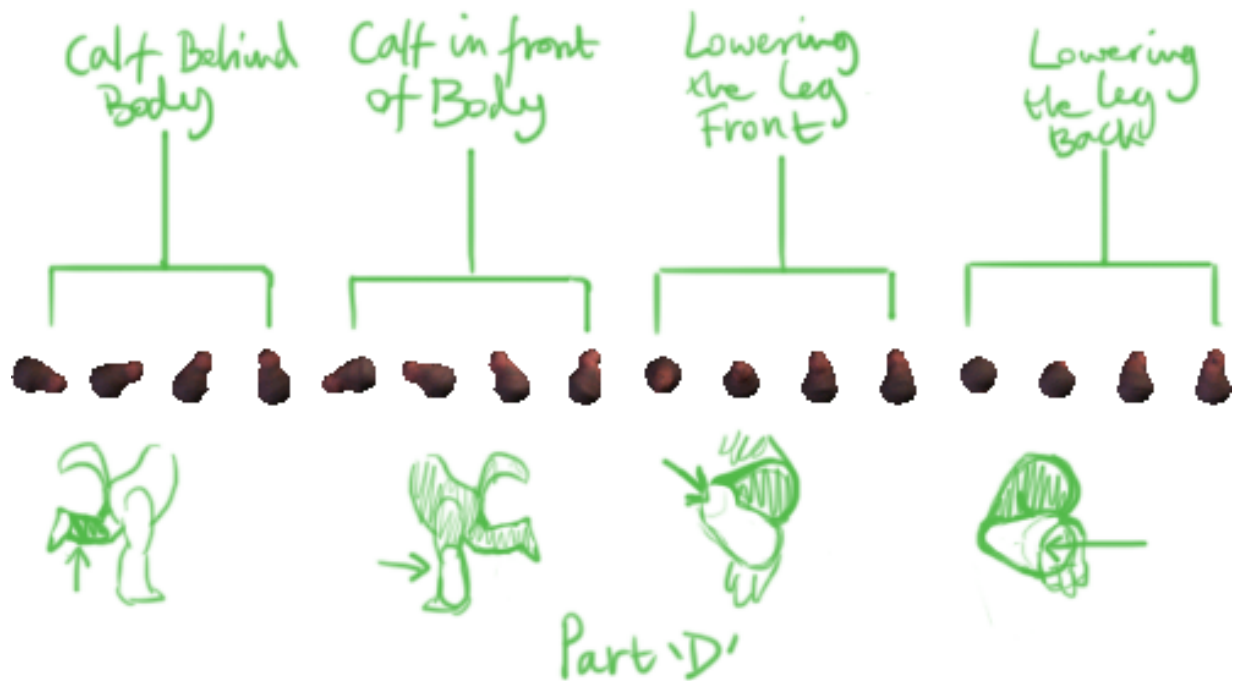


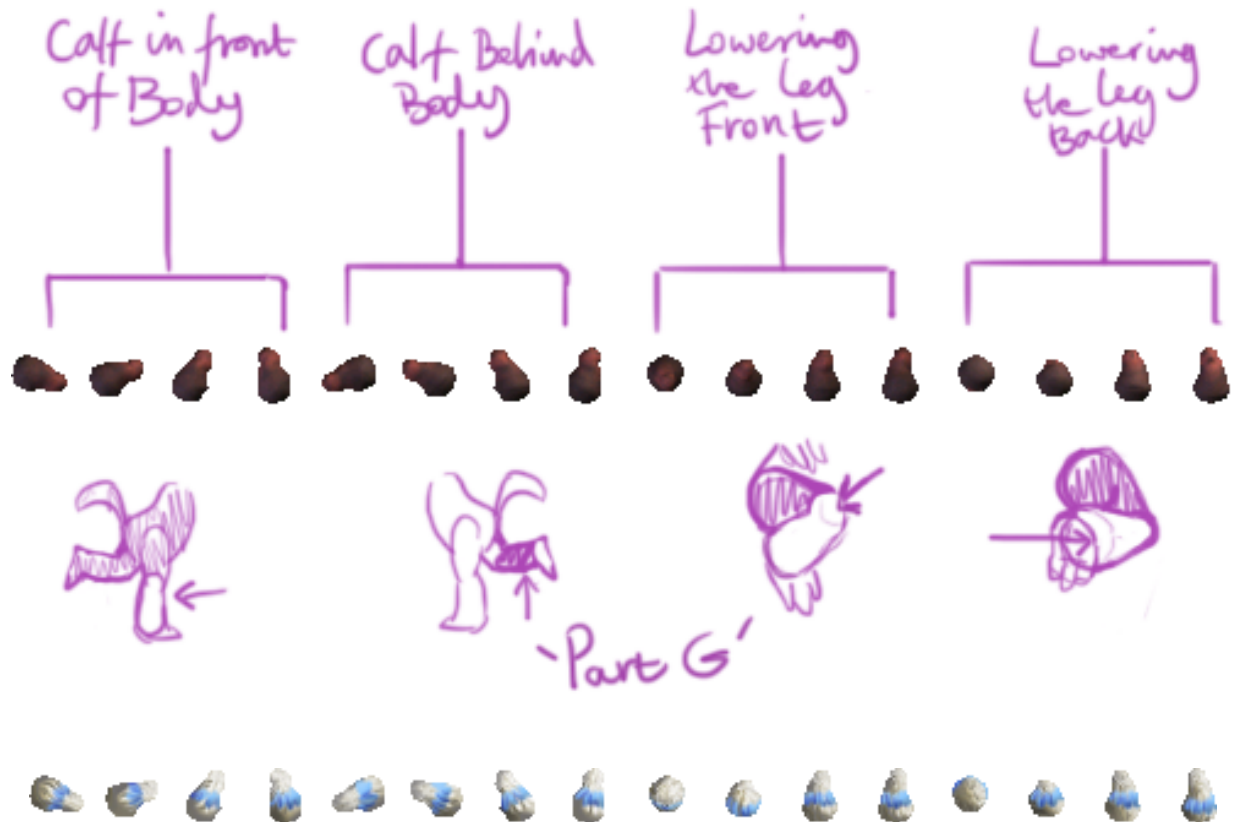
pictured: first 4 are lower leg facing right from behind the body, second set of 4 are facing left and layer overtop of the body, followed by a front facing set starting knee-up and lowering down, then the last four are a shot of the bottom of the leg (ankle) lowering down to a straight leg position.

While the original Bean's ettins have white lower legs, I've added a blue band to these to help show the angle of the limb here.

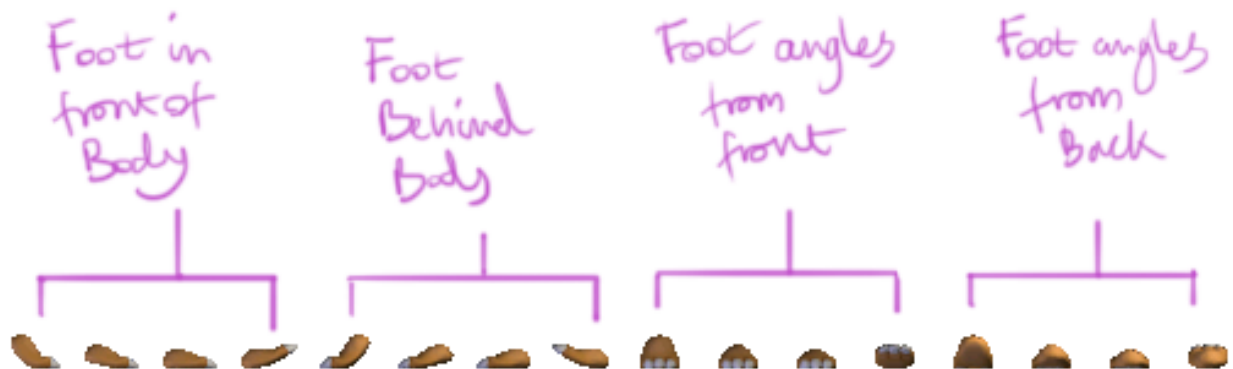
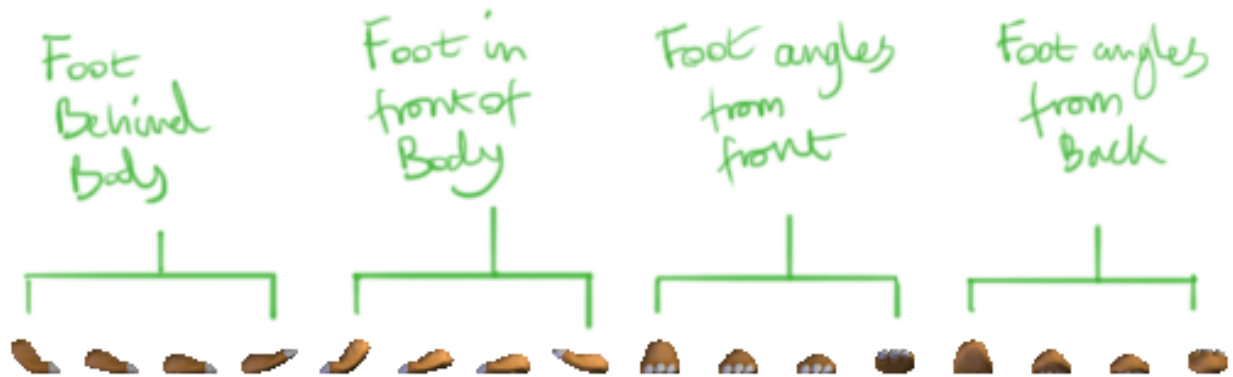


And here's piece 'g', It's easy to get tripped up here with how similar the two pieces are, but the best part to focus on is making sure that the last 8 images are flipped the right way. Referring back and forth with the original sprites here frequently is a good tip.





Now you're almost done with the legs! The feet come next and are both small (easy to finish quickly) and forgiving (you get some creative freedom), so at this point it's again going to be making sure things face the right way. I've given creatures two left feet before and I'm here to try and help you avoid that fate.



This is foot 'e' and it's fairly self explanatory in that once again the first four sprites are the inner side of the foot, the outer facing side, front then back. One method to avoid worrying about flipping the front and back foot sprites for foot sprite set 'h' is just to make the foot symmetrical. Here's an example of a foot design that would work pasted into both 'e' and 'h':



In my case, I'm working on a creature themed after creatures 1 and 2, where the foot sprites are hand-like with three 'fingers' and a distinctly thumb-like toe on the inside.



Here's my 'e' art, followed by 'h' to show you my ettin's foot sprites with the 'thumb' to give an idea of the directions in which the foot sprites face.



And after this, that's the legs done!

There are either two (for an ettin with no tail) or four (for tailed creatures in general) pieces of art left to do for your sprite breed, and only one of those is a confusing nugget shape: The upper arms, sprites 'i' and 'k'.

The first one alphabetically is piece 'i':



The first 8 images are what you'd expect, aka the inside followed by outside views of the upper arm. After that it deviates from the way the feet and knees were sprited by started with the arm facing forward and 'swung back'. I'll include diagrams but envision this as the arm as you swing it back behind you while walking. The next three frames are that arm being swung forwards and up into the air. The last four are the same action but flipped and viewed from behind.

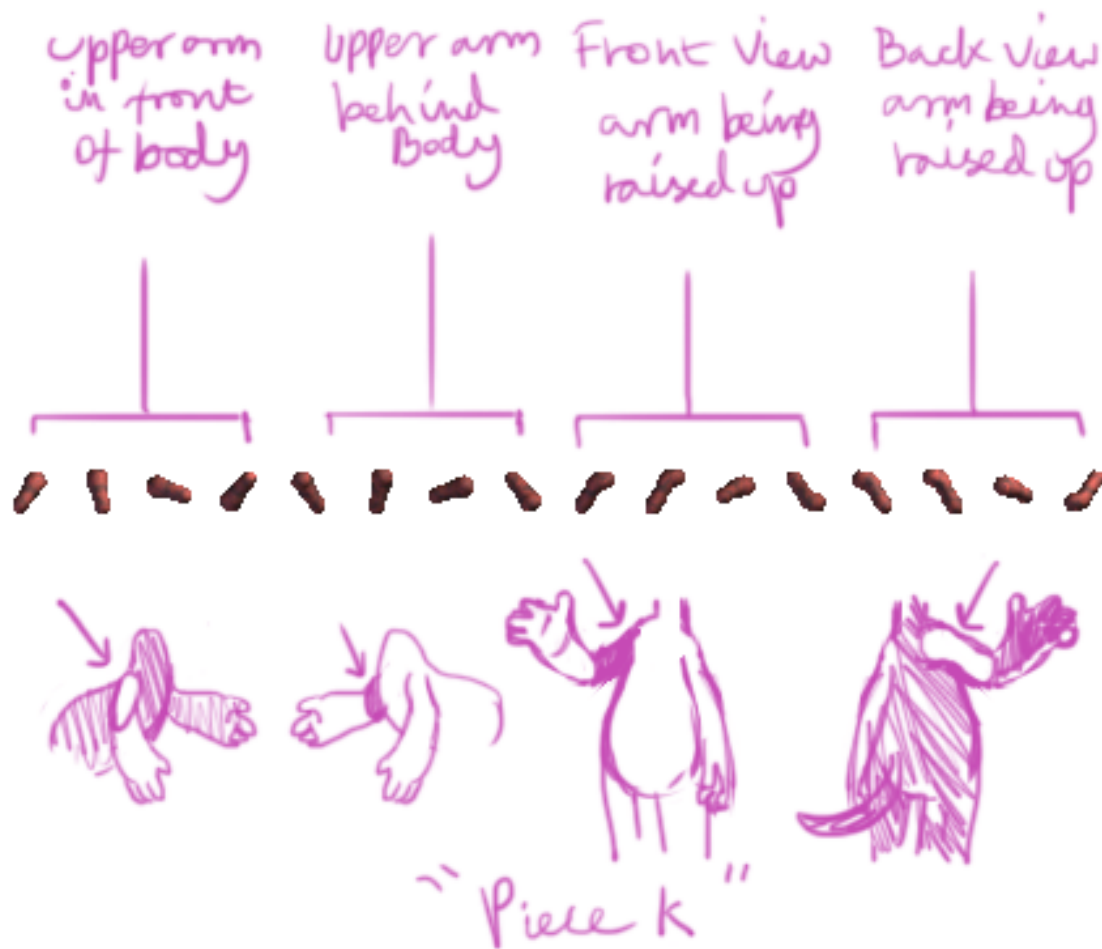
upper arm
behind
Body

upper arm
in front
of body

Front view
arm being
raised up

Back view
arm being
raised up





To help disambiguate these arm pieces, which genuinely look like macaroni, Here's some additional diagrams, followed by my sprites. Similarly to the upper thigh sprites, I've added a small blue marking area to help define the top of the arm (the shoulder) where it joins the torso, and help visualise how it tilts.

'l' followed by 'k'- I was able to mostly reuse the first 8 images as the arms are identical on both of their side views.



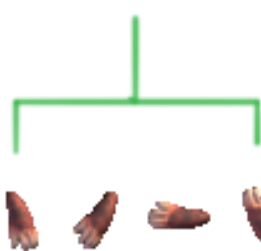
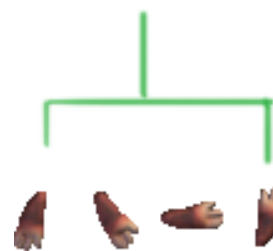
The last part of the arms is the lower part. This sprite includes the forearm, wrist and hand all in one sprite image allowing for a large amount of creative freedom. In my instance as I'm working off of an existing design, these arms are going to look very normal, but there's nothing stopping you from making the lower arm a pair of pincers, wings, or tentacles if you so desired!

Lower arm
Behind
Body

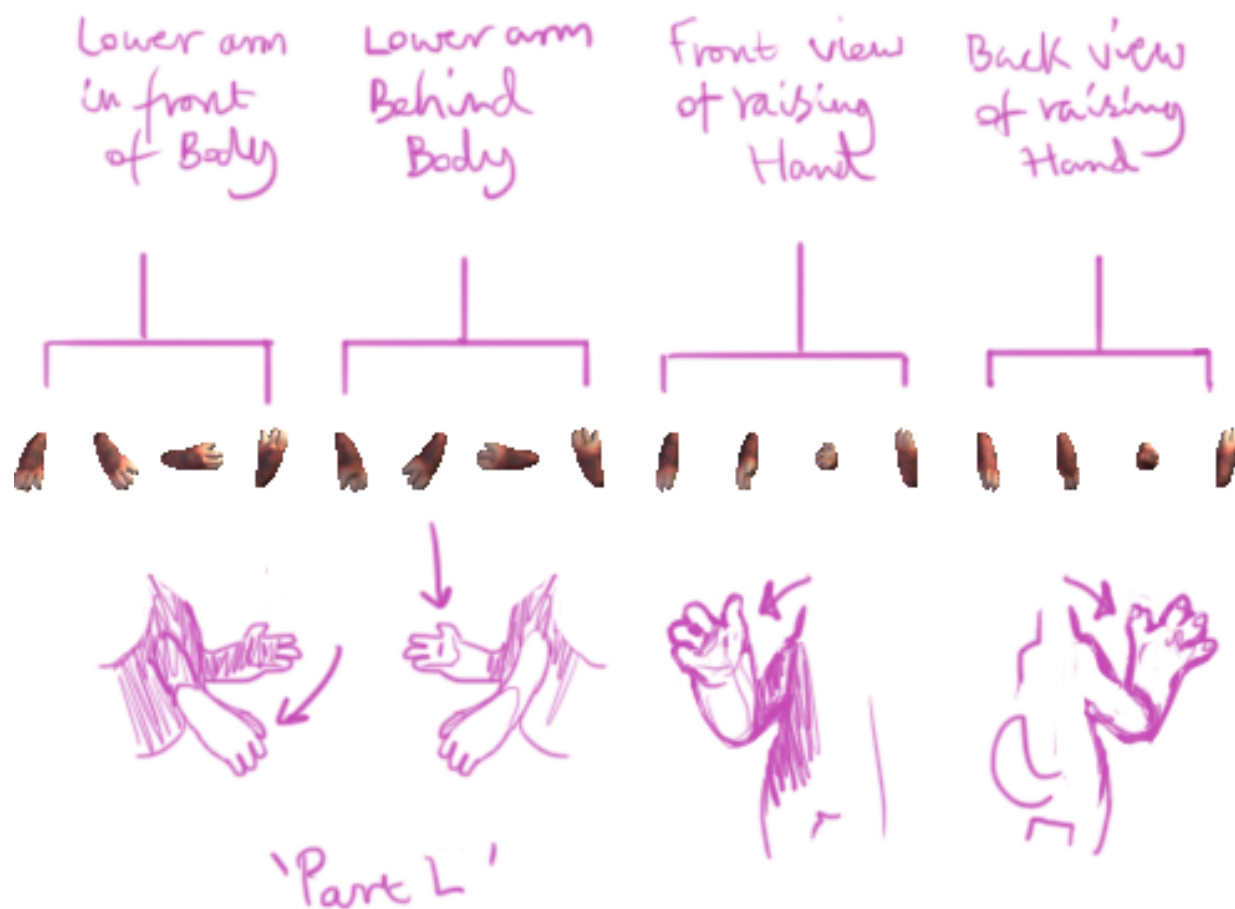
Lower arm
in front
of Body

Front view
of raising
Hand

Back view
of raising
Hand



'Part J'



The arms are a really good example of a sprite where, if you only did top-down shading, you could get away pretty much entirely with mirroring images from one set to the other, provided you're very careful to line them up in the right places. It would have been lovely if part 'l' was just a complete flip horizontal of part 'j', but it's more like the pieces are flipped but then completely reorganised, causing the same need for caution we had with all those little similar-looking leg pieces before.. At least this time there's a very obvious hand in the images to track what you're doing with.

Here's my sprites, again with blue color band to help show the direction and tilts:

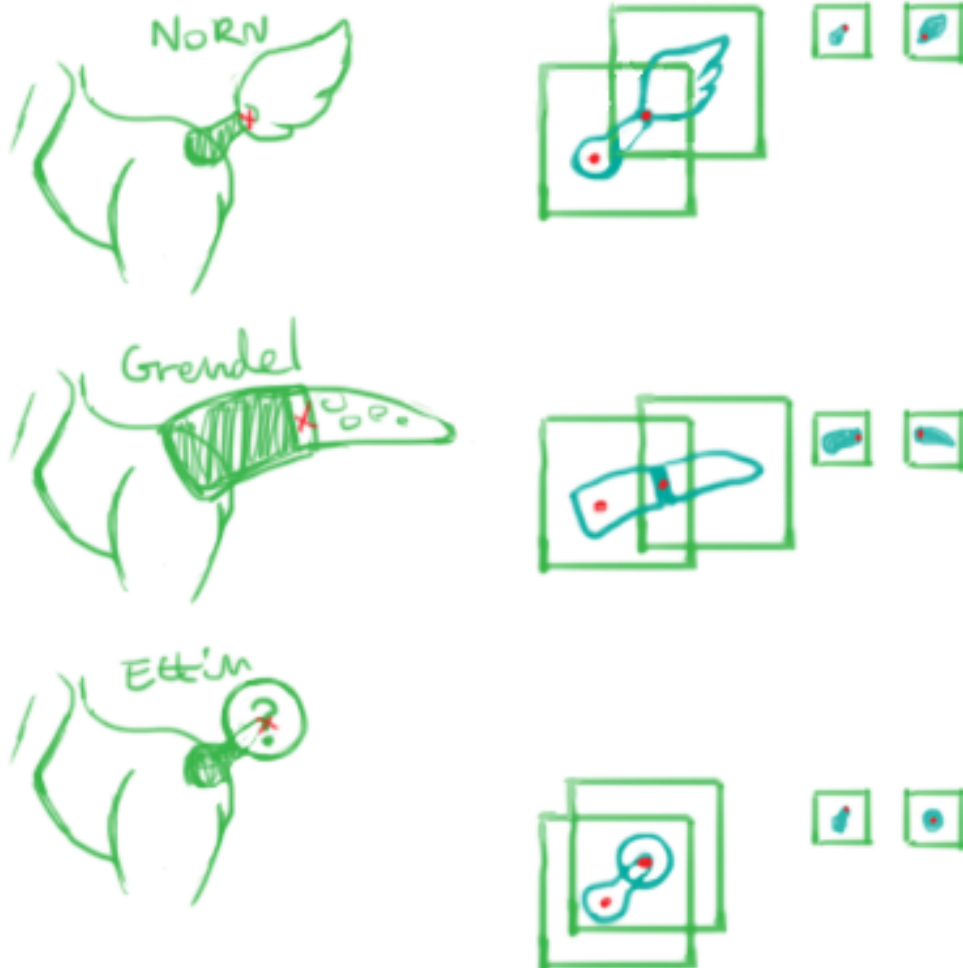


That's all the hard parts of your sprite breed complete! The very last step is the two tail segments, if you're opting to give your ettin (or other creature) a tail. Considering how many hours of work can go into all the previous steps, there's no judgement being cast if you choose to follow the classic ettin design and don't include a tail. You could even go and add a little patch of fluff on their body sprite where the tail is, like a rabbit, or a ridge of fur down the back. Just keep in mind that tailless creatures can inherit a tail from a parent who has one! So a tail stacked on another tail might look a little goofy if

you go all in and draw the tail on the body.

Handily, the original Bean's ettin in creatures 1 had a tail, so they can be used to go over the steps for tail making.

When it comes to tails in creatures 3/docking station, there are two main forms they can come in: Tails that use norn data, and tails that use grendel data. The easiest way to visualise these is that a norn tail has a short base sprite and usually a longer tail tuft at the end, a bit like a cartoon donkey might. The grendel tail is a thick reptilian style tail that sticks out fairly straight behind the creature. If your desired tail doesn't fit either of these visual criteria, you may have to look into which sprites have the best size constraints for your project and then get a little inventive with the art. There is also a secret third option: Ettin tail data. Ettins have digital relics of a tail they once had, before it was cut from the game: A few blank sprites here, a few floating pixels there, a few data files to show where these parts would hook up. They're actually unique to the other two tail types and I'll go over them further in a visual comparison.

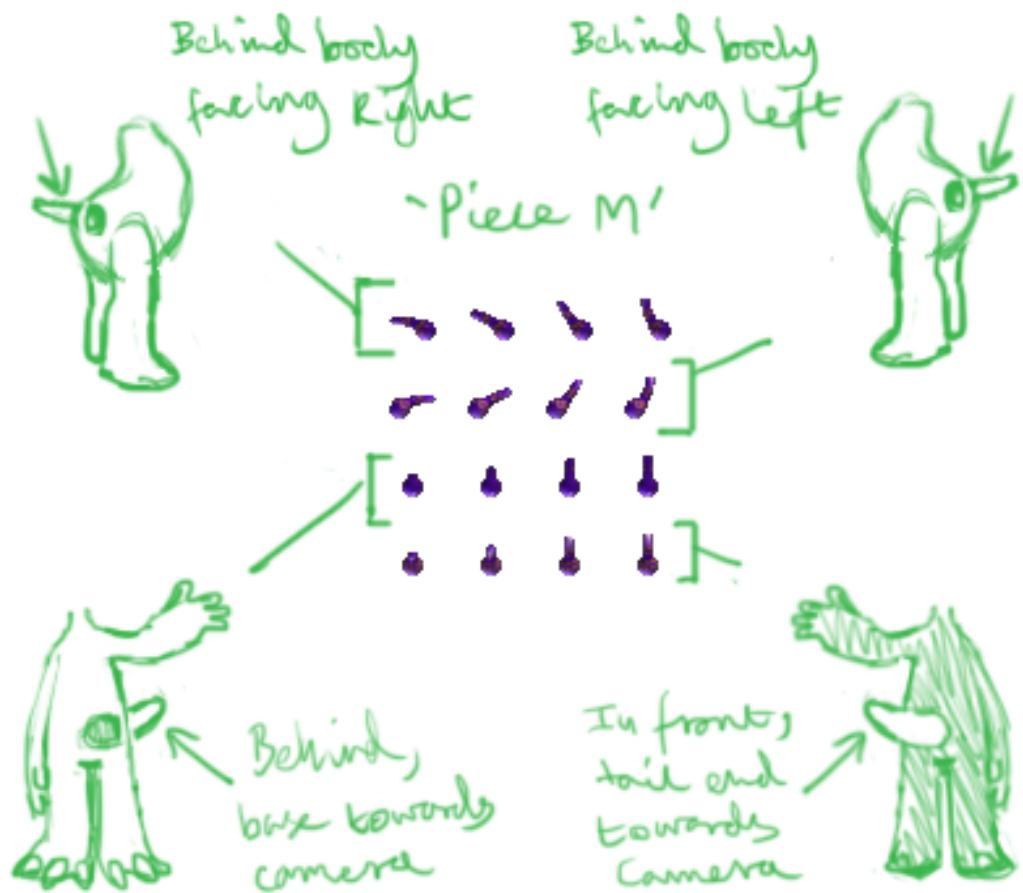


Pictured: Norn, grendel and ettin tails showing the overlaps and hookup-points between their tail base sprite and tail end sprite. If you envision the tails as two square pieces of

paper, the length of the tail is dictated by both how big the squares are, and also how much the squares overlap each other. Each type of tail is sliding different sized squares of paper around over themselves in different ways. The most outright useful things for you making a sprite breed to know though are:

- Norn tails have a shorter/smaller base sprite than grendel tails, and most of what you see is the end sprite of their tail due to the base of the tail being largely obscured behind the body of the creature.
- Grendels have overall the longest tails with the biggest sprite space. They are slightly more rigid compared to the norn tail.
- Ettin tails, or what's left of them, are structured similarly to Norn tails at the base, but the end of the tail is attached by the very middle of the tail end sprite. This gives the least room for tail length, but more room for rounded or short fluffy tails in theory because you can use the entire square space of the sprite without having to worry about where the end of the tail tip lines up.

For the Bean's ettins I'm working on, their sprites were originally done over c1 norn tails, which were often drawn onto the body sprite. There's no direct equivalent in c3/docking station, but I'm going to use the c3/ds norn tail sprites as the nearest equivalent.



All creature tail sprites (both parts 'm' and 'n') are organised the same, no matter if they're norn, grendel or ettin. Starting with 'm', the first four are as you would expect by now, facing right behind the body. What's new is that the side facing tail sprites are - always- behind the body sprite to make them appear as though they grow cleanly from the torso. The last tail images are the tail being raised up from a lowered position, from the front and behind. If your creature has a belly marking of some kind that could run down the tail, it would show on the last four images.



The bean's ettin tail base I've made is a slightly fluffier version of the official norn one. All that really 'matters' with the tail 'm' sprites is that they start and end in the same

points as the official ones do: Where they connect to the body and where they connect to the tail tip, like a game of 'join the dots'. This ensures the body, tail and tail tip connect and animate together properly. With the tail end, part 'n' you have even more freedom, only restricted to making sure that the base of it lines up with the official sprites.



These are the tail tip sprites for the bean's ettin!



The bases of these tail tufts are pixel identical to the official norn 'n' sprites, but everything else within the sprite space is different. One thing I found useful while trying to figure out how they would move in game was opening up an image file of the 'm'

sprites and just pasting and waving around the 'n' sprites overtop of them and seeing roughly how they connect to each other.



an example of lining up the sprites to see an approximate of the end result. I opted to make the tail tip a little brighter after comparing this version of the sprites to the tail base and body.

Now all the adult creature sprites are 100% done! Yes, you've done the worst bit!

Anything past here is down to image scaling or optional editing. Generally, sprite breeds have a small visual change between adult and old creatures, and between the two sexes as well.

The adult sprites of your sex of choice are the most useful sprites to have on hand for making the rest of the breed: They're already at the maximum size and detail you're expected to make. Using them as a base for the rest of the work is practically mandatory to my own dev process. An example of this is that I have one breed that I made unique sprites for every life stage, and it took me a month of solid work. Comparatively, streamlining the process like this, I can make a new complete breed in two or three days.

On that note, you can follow along with me making the female Bean's ettin! In this case, this little fella I just finished is the male, primarily because there's much better

documentation of the male Bean's ettin design than there is the original female. In some previews, caps, etc, the female is white furred. In others she's solid brown or reddish hued. Here I've taken the male head sprite and adjusted the eye and accent colors to match the red-brown look and the original red eyes of the female. I also went back at this point and added a lavender strip from the original design to both sexes of sprite to make them gel together better.



Pictured: comparison of the color changes between male and female sprites. This can be done by hand painting, but it's extremely quick and easy if you have a program that has hue shifting abilities to try out different recolours of your first set of sprites sheets! For me I did my best to select the blue areas of all the male sprites and then make a separate layer of that area filled in a brownish color. After playing with layer modes for a little, the setting 'hue' worked best at preserving the general lighting while changing the color of the fur.

The old sprites are also generally very easy to make: The official in game breeds for c3/ds use desaturated versions of the adult sprites for the elderly ones. If you have an art program with saturation options (usually a part of where you mess with hue shifting) then this is almost a 'one click and done' kind of thing. Failing that, if you have layers available you could try something like an off-white or a mid grey above your original sprites (to produce the 'greying' look) in a very low opacity to try and preserve some of the shading. Another idea could be manually adding a few streaks of white into their fur! I used the lilac area from my adult sprites coloured light grey for this along with overall fur area desaturation.



Pictured: comparison of the adult and elderly male bean's ettin head sprites. The same image filter processes were done to copies of all the adult sprites to make the old ones.

Before we get to resizing sprites for the smaller life stages (or working over smaller sizes, if you prefer that) this is a good point to pause and organise your files some: If you haven't saved out your work files into pngs by now, this is the time to start doing it and putting them in nice organised folders. Otherwise you're about to be completely swamped with images! I like to folder my images by age, sex and filetype, an example would be a folder full of all the saved out sprite sheet pngs for old females. Another way could be to organise them by body part, with all the ages and sexes of that part saved into the folder so you just dip into the folder for the relevant part you wish to convert to c16 format.

If you get confused or lost, a good rule of thumb when working with sprite sheets is that there are 28 per life stage, (14 of those male, 14 female), so you can always count your files if you're worried about if you've grabbed all of them!



pictured: eight folders each containing 14 sprite sheets in png format. This is much easier to keep track of than if all 112 sprite sheets in one folder!

Now, the baby and teen life stages... There are two ways to make these. One is to make them over existing sprite sheets for those life stages, if you have them (these should be available alongside this tutorial!) and making them the same way you did the adult sprites. This can be a good method if you imagine the younger life stages as having distinct appearances like very different markings, colors or outlines to each other. The other way is to take the adult sprites with or without alterations and then shrink them to the size of the younger life stages. I'm going to be doing the second approach with small color and marking adjustments. If you choose to do the resizing option, follow along, and if you're making them over the younger stage sprite sheet bases, I'll meet you further down for sprite file conversion.



Pictured: An example of how markings and color might change over the life-stages of a Norn.

As a preface: Ettins, norns and grendels all have slightly differently sized sprites throughout their lives, so when referencing pixel dimensions and image sizes, make sure you're referencing the creature genus you want!

In this case I'm making ettins on regular ettin sized sprites, with the exception of the tail sprites, which are norn sized. I'll tackle the tail first to handle the outlier and count as the example piece. Included should be some text documents listing the dimensions of the individual sprites (say, one tail image from the sheet at one angle) and the dimensions of the sprite sheets at different ages. I'll mention the relevant ones here to make sure you can follow. Norn tail bases are the same size for adults and old creatures at 30x30px. On the included sprite sheet they're arranged in a 4x4 grid, meaning that the sheet is 120x120px in size. I mention this because the provided sprite sheets have specific dimensions, but if you're extracting your own sprite sheets for a specific project, they may be arranged differently, in which case you will have to calculate the sheet size based on the sprite arrangement. For example, if my tail sprite sheet was arranged in an 8x2 grid, it would be 240x60px and you'll have to figure out the scaling for the smaller stages that way.

Teen norn tail bases are 24x24px, and the sheet is 96x96px. Resizing my tail base sprites down to this size using the 'nearest neighbour' setting gets me this:



A little boxy, but as this sprite is buried mostly in the body of the creature, this will do just fine without edits in this case. This is a good time to check whether or not your art program has a resize setting like this that preserves hard edges. Sometimes programs will have a 'lock transparency/alpha' function alongside this or instead of this. Either way, on this small easy to resize and undo body part you may want to try some different methods to resize to get what you like: The absolute, single most important part is to avoid semi transparent edges, as those will come out as a dark pixel mess in-game. You could clean the edges manually, but you would likely be in for a world of hurt and frustration. At very worst if your art program doesn't have this kind of resize function, you can probably grab a second one or an image viewer with scaling options just for the purpose of resizing.

The norn tail end sprites start at 32x32px in a sheet of 128x128 for adult and old stages, and are 25x25px in a sheet of 100x100 for teen.

My result on a quick resize is this:



Which is ok, but a little boxy on a few edges, which is easy to adjust!



Not a large difference especially when they'll be in motion in game, but you know how it is- I can see it therefore I care. XD This said, try to limit your adjustments to a few tasks that are simple to do- maybe a few tweaks with the selection tool, or a couple of adjustments with the brush in a single color. Otherwise you'll be adjusting forever: "Perfect is the enemy of good".. It's also the enemy of finished.

For the baby sized tail you can either take your teen sized sprites and shrink them further, or start from the largest size again. The results are usually pretty similar, so feel free to experiment!

Here are my female tail sprites in the baby size, after a couple of tiny tweaks:



pictured: Super tiny tail sprites! This is effectively the smallest that sprites get in game, so don't worry about having to handle anything smaller than this.

All the resizing from here is the same method, and references the ettin pixel size dimensions, excluding the tail obviously. I'm going to paste them from my documentation here for ease of reference:

ETTIN INDIVIDUAL SPRITE SIZES:

HEAD (A)

Grid Setup: 16 x 12

Baby Sprite Size: 60 x 60

Adolescent Sprite Size: 70 x 70 (sheet size 1120x840)

Adult/Old Sprite Size: 80 x 80

BODY (B)

Grid Setup: 16 x 4

Baby Sprite Size: 40 x 40

Adolescent Sprite Size: 47 x 47 (752x188)

Adult/Old Sprite Size: 54 x 54

UPPER LEG (C & F)

Grid Setup: 16 x 1

Baby Sprite Size: 24 x 24

Adolescent Sprite Size: 28 x 28

Adult/Old Sprite Size: 32 x 32

LOWER LEG (D & G)

Grid Setup: 16 x 1

Baby Sprite Size: 24 x 24

Adolescent Sprite Size: 28 x 28

Adult/Old Sprite Size: 32 x 32

FEET (E & H)

Grid Setup: 16 x 1

Baby Sprite Size: 24 x 24

Adolescent Sprite Size: 28 x 28

Adult/Old Sprite Size: 32 x 32

UPPER ARM (I & K)

Grid Setup: 16 x 1

Baby Sprite Size: 20 x 20

Adolescent Sprite Size: 23 x 23

Adult/Old Sprite Size: 27 x 27

LOWER ARM AND HAND (J & L)

Grid Setup: 16 x 1

Baby Sprite Size: 24 x 24

Adolescent Sprite Size: 28 x 28

Adult/Old Sprite Size: 32 x 32

SPRITE SHEET SIZES

baby head: 960x720

teen: 1120x840

adult (and old """): 1280x960

baby body: 640x160

teen: 752x188

adult: 864x216

thigh:

384x24

448x28

512x32

calf:

384x24
448x28
512x32

foot:

384x24
448x28
512x32

upper arm:

320x20
368x23
432x27

lower arm/hand:

384x24
448x28
512x32

After resizing there's just a couple of miscellaneous files to make, and you're done! Egg glyphs are the preview images that the egg layer in game uses to show you what breed you have selected: You need one for male and one female. They are commonly a front-on face sprite of your breed scaled to fit, but don't have to be, as long as they're the official 59x52px dimensions. A small aesthetic note is that some of that sprite is overlapped by the egg layer buttons on the right. So it's best to keep the image on the smaller side, as if you use the entire width there may be some image cutoff.

An example of an official pair of egg glyph images, here are the male and female c3 ettin egg layer pics:



Pictured: male and female c3 ettin egg glyph images.

The baby life stage faces fit into the egg glyph size pretty well- here's an example of the male and female beans ettin baby sprites as egg glyphs, with a little additional male/female sex indicator.



So we have all these images, now what? Well I'd love to go into further detail but I'm gathering up resources to package in with the larger tutorial (genomes, data, programs, links) and I don't have those finished yet. I wanted to make sure people have SOMETHING for the art process side of things regardless though. Be sure to check for

wikis, sites, and the community for help and further tutorials! If you're lucky someone will compile your breed for you if you ask nicely enough. c:

Hang in there- Jesseth