

## A Guide to Polygon Pixel Characters

### Software and plugins:

Blender for the modelling and aseprite for the textures. As for plugins, a key one for me is texel density checker and although I haven't yet got Primbase to work for myself I imagine it would also be very useful.

### Setup:

Although characters may be more complex than a sign, this video's fundamentals are what I have been using all the way through my admittedly short 6 months of time making polygon pixel models.

▶ Pixel Perfect Texture & Easy Unwrapping! in Blender 2.8 My Workflow Tutorial.

### Resolution:

Basically this boils down to how big you want the character to be. A smaller character will often need more primitive shapes to maintain a cohesive look.

ArtOfSully has been making characters for two years and in that time he's gone through two different sizes illustrating this point the best so below I'll do a brief comparison to show what I mean exactly:



POKÉMON @ARTOFSULLY



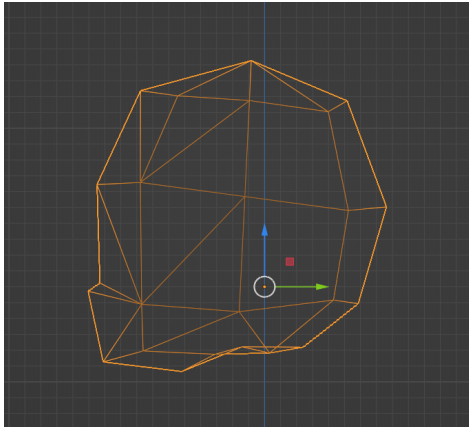
Sully's older style is a relatively high resolution being about 128 pixels high, with this level of detail comes the opportunity to convey more to the observer but comes with it's own issues.

As Sully said himself, a resolution of this scale means that it starts to drift away from being pixel art and more to a low resolution texture and so part of the original intended style was lost.

With Sully's new designs, they rely far more heavily on flat textures to convey what previously may have been done with polygons. This is in large part due to the massive downscale to around 48 pixels high. Other details have also been removed, a good example is the head which has become far simpler, removing the nose and shortening the head.

Although it may appear that I'm favouring the lower resolution style of Polygon Pixel art with these examples, that's not the case. Smaller styles lack the ability for greater expression and characterisation. Put simply, the resolution that works best is entirely dependent on what you yourself want to convey.

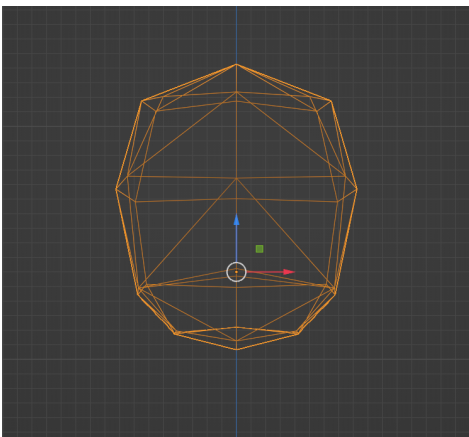
## Modelling and Texturing:



The head is likely the most difficult shape to get right so here's a guide on how I learnt to do mine, hopefully it saves you the weeks of trial and error it took me.

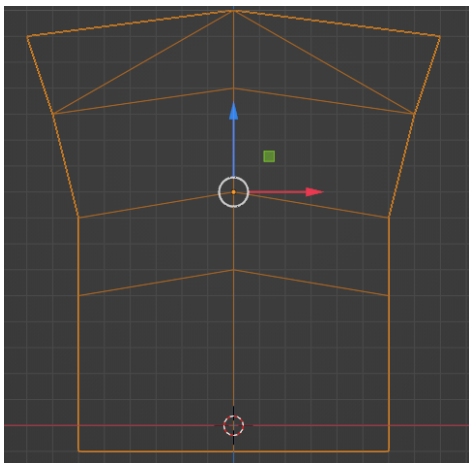
The base primitive you will want to use is a sphere with 8 segments and 5 rings. After that, it's just a process of adjusting the topology. You may notice the front faces have been converted to tris, this is just to give me some more control of the silhouette and the way the textures act on the mesh.

With textures, one square on the grid is equal to one pixel so make sure you're looking at the grid for reference.



When it comes to texturing the head, or any other complex shape, you need to avoid creating obvious seams. The best way I found to do this is to flatten a group of polygons on the mesh in one axis, then unwrap the texture, and then extrude the mesh back out using a copy of the original model as reference.

The only issue that can occur with this is pixel warping but I find that it's hardly ever noticeable.

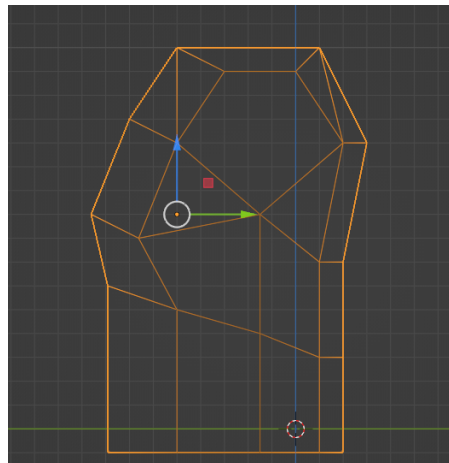
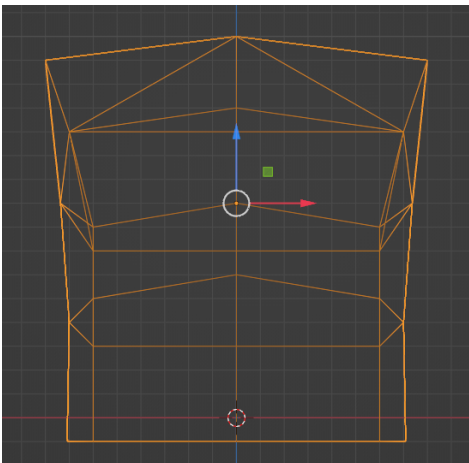


This body is rather easy, I edit a plane to get the shape seen in the first image and then extrude it twice. It's then edited to get the shape you see in the lower images.

I use the same texturing technique that I use on the head for the body to avoid seams.

The arms, legs and neck are all cylinders with minor alterations.

I don't have any good instructions on how to do the feet or hands unfortunately since they're usually more trial and error than anything else.



**Further Reading:**

**My Sketchfab Page:**

<https://sketchfab.com/HenryLutece>

**Sully's Sketchfab Page:**


<https://sketchfab.com/artofsully>

**Red Alchemy's Polygon Pixel Art:**

<https://redalchemy.tumblr.com>

**Youtube tutorials:**

**Setting up:**

 **Pixel Perfect Texture & Easy Unwrapping! in Blender 2.8 My Workflow Tutorial.**

**PS. I didn't realise how few sources there are that I know of this style, hopefully more people will get involved with it over time.**