# Things You Should Know About Optimization

This book delves into the intricacies of optimizing content for VRChat, exploring the how and why you should optimize your content. This aims to offer comprehensive knowledge for optimization for those who are unfamiliar. While this book primarily focuses on avatar optimization, many of the same principles can be applied to world optimization as well.


- Basic Optimization
  - Draw Calls (Materials and Meshes)
  - VRAM Usage A.K.A Stop Being Monsters With Your Textures
- Advanced Optimization

# Basic Optimization

Basic optimizations that can be applied fairly easily to anything

# Draw Calls (Materials and Meshes)

Generally, you should keep your material count as low as possible, as every new mesh and material in a scene counts as a new draw call, a draw call is an instruction to the graphics card on what to draw and how to draw it, the more draw calls you have, the more instructions being sent to the graphics card, and therefore the slower it'll be as it's processing all those instructions.

The exact numbers of how many draw calls is "too much" is difficult to gather, because it heavily depends on the computer hardware, however, the lower the better. While many modern computers could run scenes with over 1,000s of draw calls, *that doesn't mean it should*, you should always aim for the lowest amount of draw calls with your worlds and avatars to ensure performance takes little to no hit.

So remember:
1 Mesh = 1 Draw Call, *and* 1 Material = 1 Draw Call

So if we had an avatar with 16 skinned meshes and 32 materials, your avatar would be ***48*** *draw calls*, which is a bit ridiculous for a single person! If you had 10 people in that same avatar, it would end up being ***480*** *draw calls!* Which is typically the amount a rather medium-sized world would likely have!

In general, draw calls add up, and the best way to reduce them is to reduce your materials and meshes, what are ways you can do that you may ask?

# Texture Atlasing

Texture Atlasing should frankly always be done whenever possible. Texture Atlasing is a technique used widely in 3D modeling to optimize models for use in games. There are two ways you can go about this.

## CATS Method

The CATS method is generally easy although not ideal, in the CATS menu, you can go to the optimization tab, click Atlas, and hit a button to save it. The atlas textures it generates are often too large and leave a lot of empty space, although it's better than no atlasing at all.




# Manual Method

This one is a bit complicated to explain in text form, so I made this short video.

https://www.youtube.com/embed/MpC6UvOic6k

# Reduce Meshes

If you have a mesh that doesn't need to be separated on your avatar, join them together. There's no need to have them separated unless you have an animation to toggle on and off a specific mesh, such as a jacket.

> As of the 2022 Unity upgrade, meshes don't need to be split due to shapekeys as this issue was fixed in the Unity version being used. However if your mesh has a lot of shapekeys, you should split for mesh vram reasons, more preferably delete shapekeys you don't use and merge shapekeys that are used together for facial expressions and such.

Basic Optimization

# VRAM Usage A.K.A Stop Being Monsters With Your Textures

With the most recent Developer Update post at the time of writing this, we have learned that Avatar Texture VRAM will now be accounted in the performance ranking.

Below is a table of what the VRAM limits will be according to the dev post.

| | Excellent | Good | Medium | Poor | Very Poor |
|---|---|---|---|---|---|
| PC | <40MB | <75MB | <110MB | <150MB | >=150MB |
| Quest | <10MB | <17MB | <25MB | <40MB | >40MB |

## *What Is VRAM?*

*VRAM* stands for ***V**ideo **R**andom **A**ccess **M**emory* to put it simply, it temporarily stores image data in a place the computer can easily access it to render. VRAM is used to display textures and render 3D models.(more info) VRAM is located on the GPU to reduce the latency compared to regular RAM.

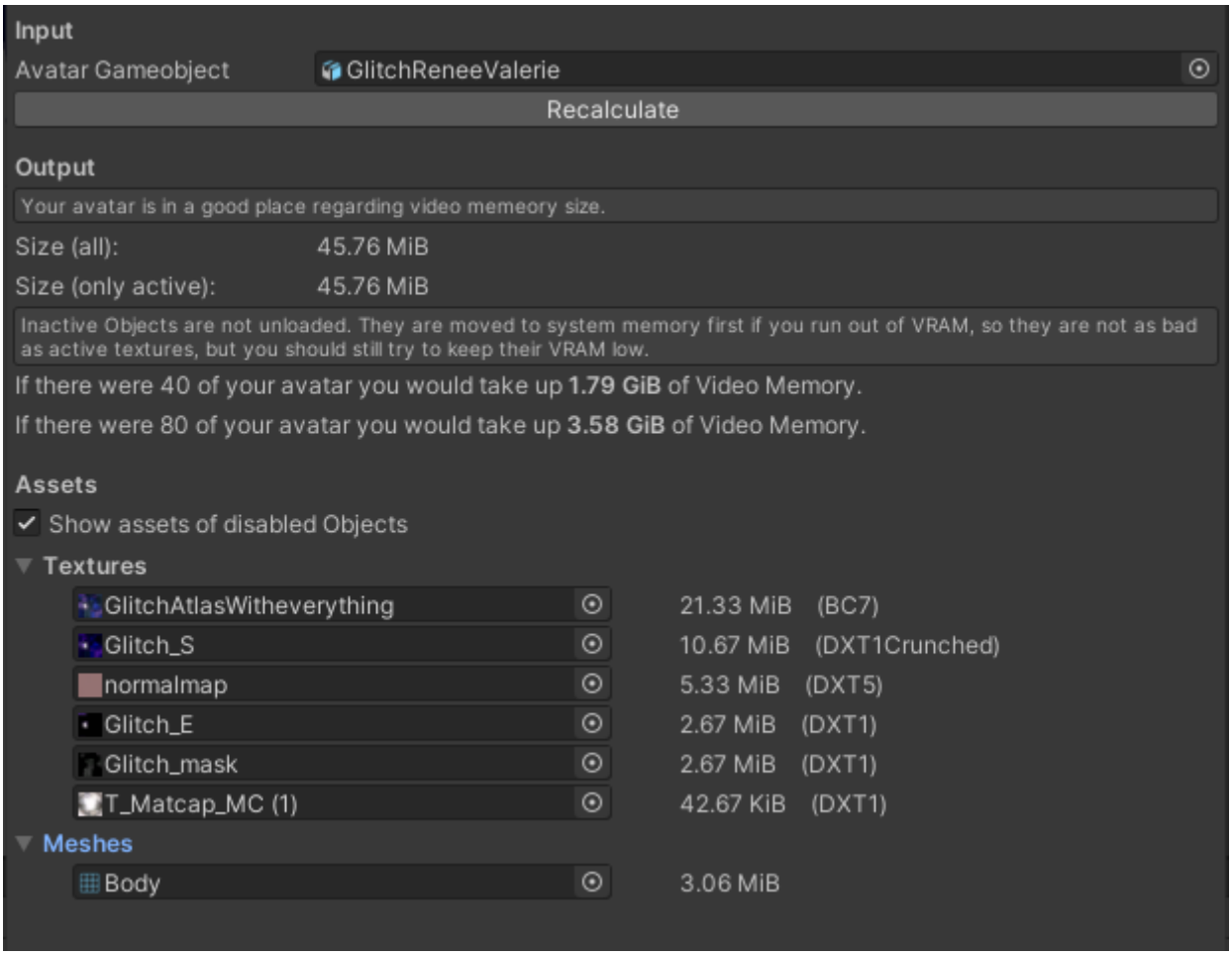
According to Steam's January 2023 hardware survey, the majority of Steam users have 8 GB of VRAM, VRChat is basing the performance rankings on this data, along with taking off 2GB to account for VR compositing and your OS rendering.

## *How Do I Reduce My Avatar's VRAM Use?*

> ***Do NOT turn off Mipmaps! You may have heard this can reduce VRAM but it can cause performance and visual quality to take a hit! Poiyomi's post here goes more in-depth with it.***

**Overall, I highly recommend reading *Razgriz's Optimizing Texture Memory* post as well, it goes into much more depth about this topic than I have.** ~~go beg them to put it on VRCLibrary.~~ ***~~(don't actually beg them obviously lol)~~***

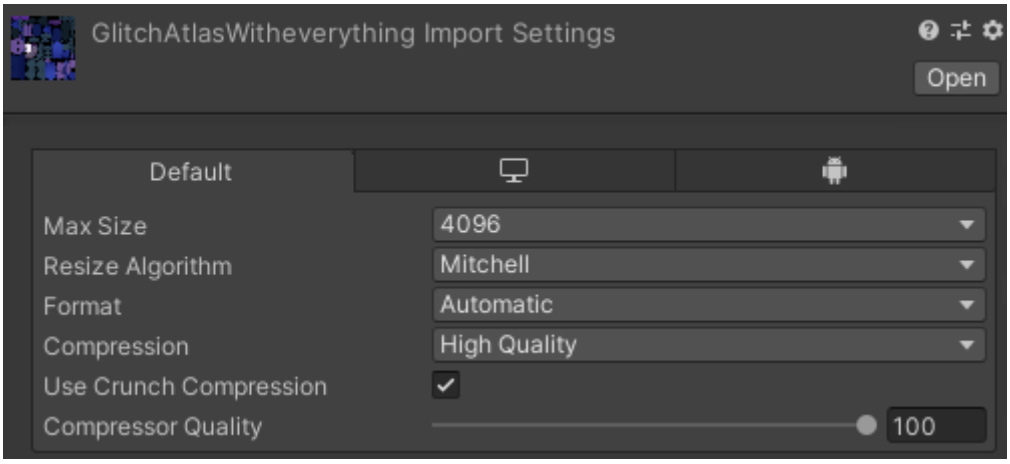
First of all, I recommend downloading *Thry's Avatar Performance Tools,* this has a very helpful tool that'll help you reduce your VRAM usage, you can access this tool by going to ***Thry>Avatar>VRAM*** to open the window once you've installed the UnityPackage. Drag your avatar into the input, and then hit "Recalculate" It'll then show you how much VRAM your avatar is taking up as you can see below with an older model of mine.



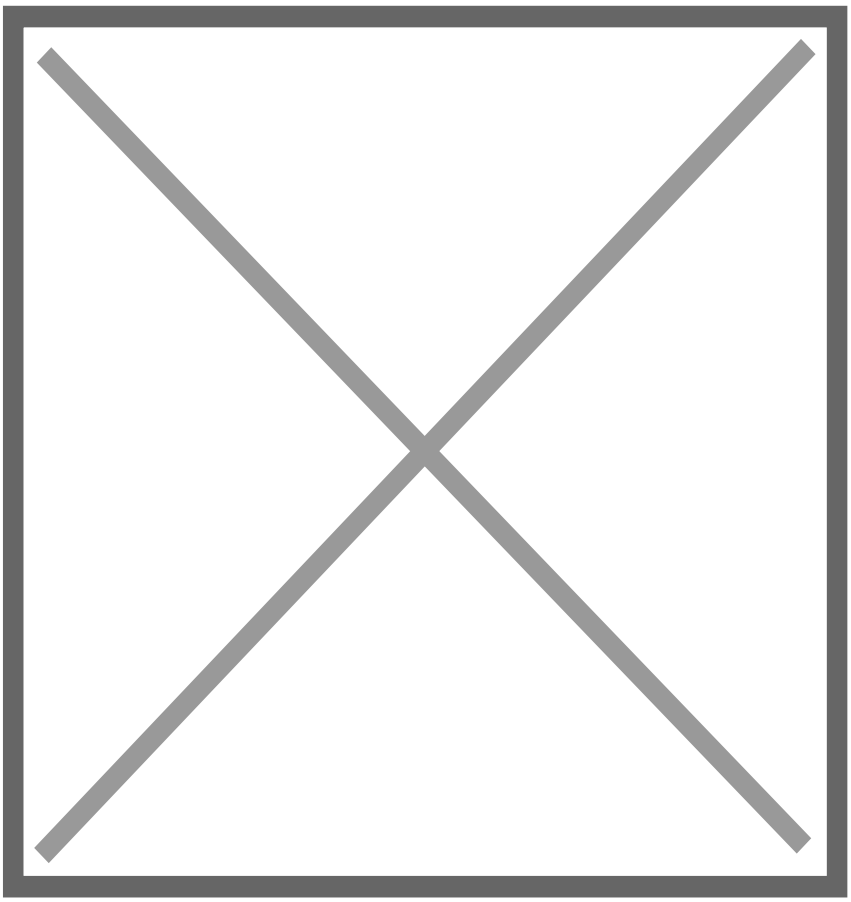

This avatar has alright VRAM usage, it would get put into "Good" performance ranking, however, we can get it much lower than this very easily! Let's go into the settings of the texture taking up the most space on our model here.

> VRAM usage does ***NOT*** equal the file/download size of the texture! The VRAM use of a texture is in its uncompressed state**, meaning crunch compression will not affect VRAM usage, however, please still use crunch compression to reduce your**

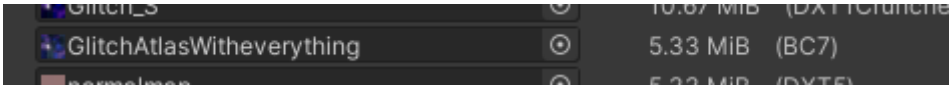
> **avatar's download size.**




As you can see, it's set to 4k, *but does it really need to be?* Frankly, no. Below is a gif showing the difference between 4096 and 2048 on my model. The only difference it particularly makes is on a very small part of my model that has a bit of detail on the horns, which is largely unnoticeable.



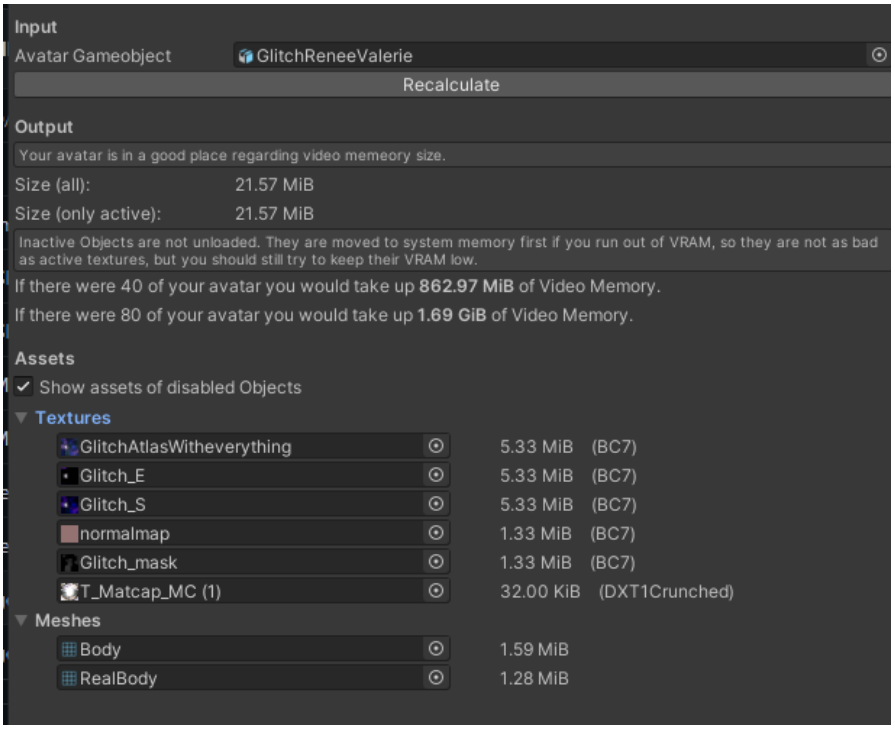
*Do you know what is noticeable though?* The VRAM usage! Let's head back to Thry's VRAM Calculator.



As you can see, the VRAM usage dropped significantly, from 21.33MB to 5.33MB, which is a ***16MB*** difference, for VERY little quality loss. Now my avatar only uses 29.76MB and fits snugly into *excellent*. However, we can take this a step further by doing it with the rest of the textures. I also went ahead to split the face of the mesh from the rest of the body and removed all the shapekeys on the body to lower the VRAM usage of the meshes. This doesn't do too much on low-poly models like mine, but can have a tremendous effect on high-poly models.

**Note on matcaps and similar textures**

If you've downloaded a pack of matcaps somewhere, chances are they aren't configured correctly and are set to 2048x2048 in Unity. Check the original file to see it's resolution, the majority of matcaps tend to be around 256x256 and 512x512 in my experience. This has occasionally made it's VRAM usage go up. Usually not by much, but if you're trying to get every byte you can, make sure each texture is at it's appropriate resolution.




After recalculating, you can now see we've gone from 45.76 MB to 21.57MB, which is roughly a 53% decrease!

So, in short, you should reduce the resolution of your avatar's textures, you can often get away with making many kinds of textures lower res very easily, especially masks which you can often lower the resolution on a ton and there will be little to no visual difference.

> **NOTE:** You do not have to change the source image's resolution, changing the resolution in Unity is enough. This allows you to avoid doing anything destructive to your textures! This is because Unity essentially recodes the entire file to be whatever specific size and other settings it's set to.

# Advanced Optimization

These optimization techniques may be more difficult for newbies and potentially time-consuming for those who haven't learned them yet.