Optimizing My Project

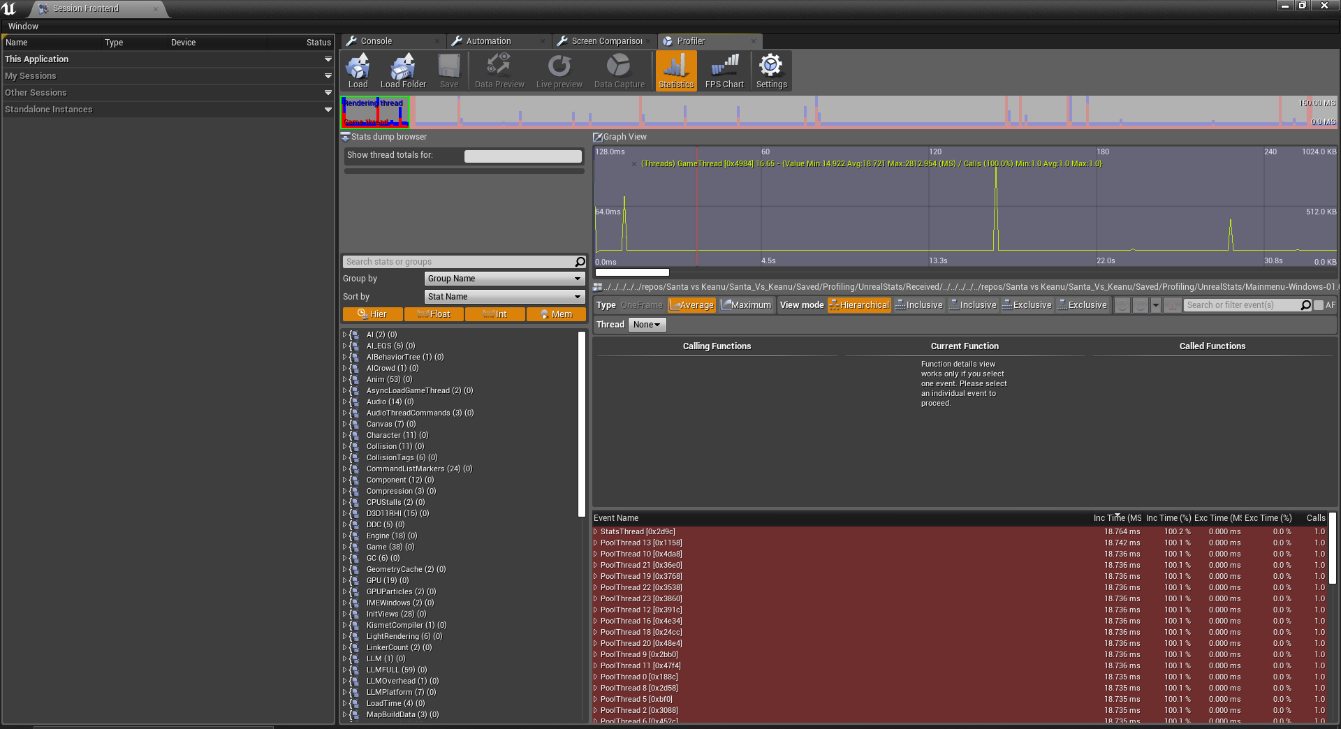
Alper Mimoglu  
 January 04, 2021

# Introduction

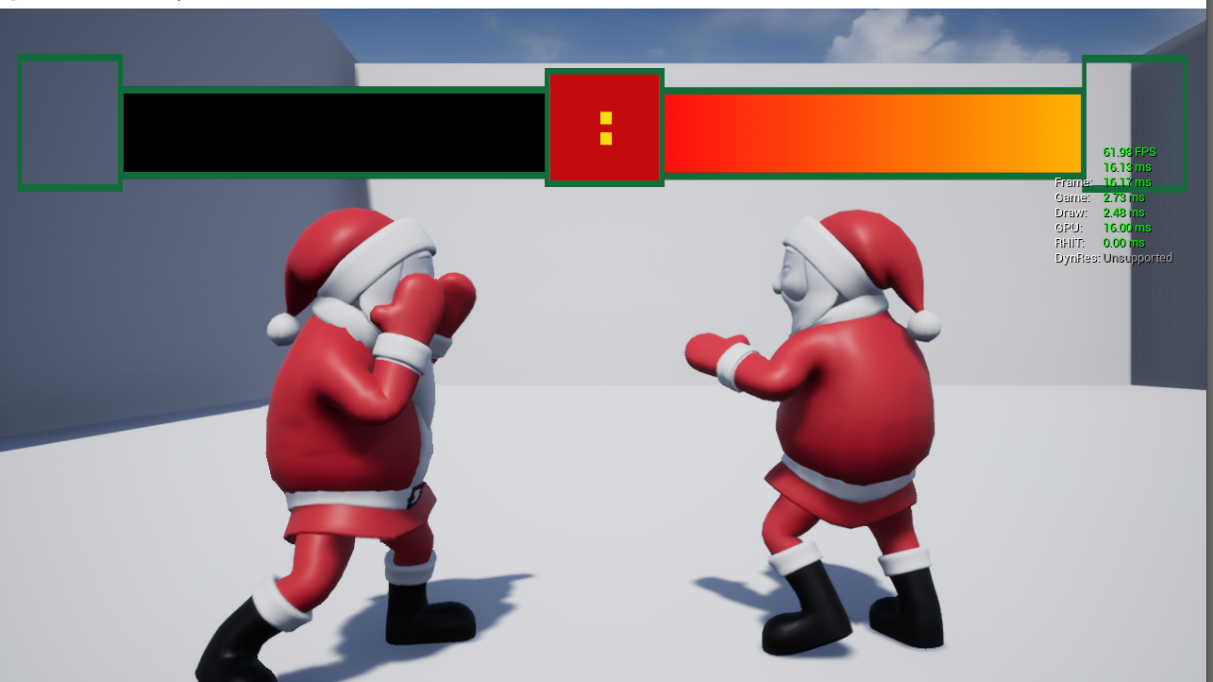
In this Journal I am going to go though my project linked down below and optimize it to see what I could do differently. I am going to achieve this by using the project frontend on Unreal Engine 4 and use the graphic visualizer to see how much resources is my project using.

**https://drive.google.com/file/d/1DbrHF3re2iegL-yat\_KY9F2NNt451ivl/view?usp=sharing**

# Before Optimizing

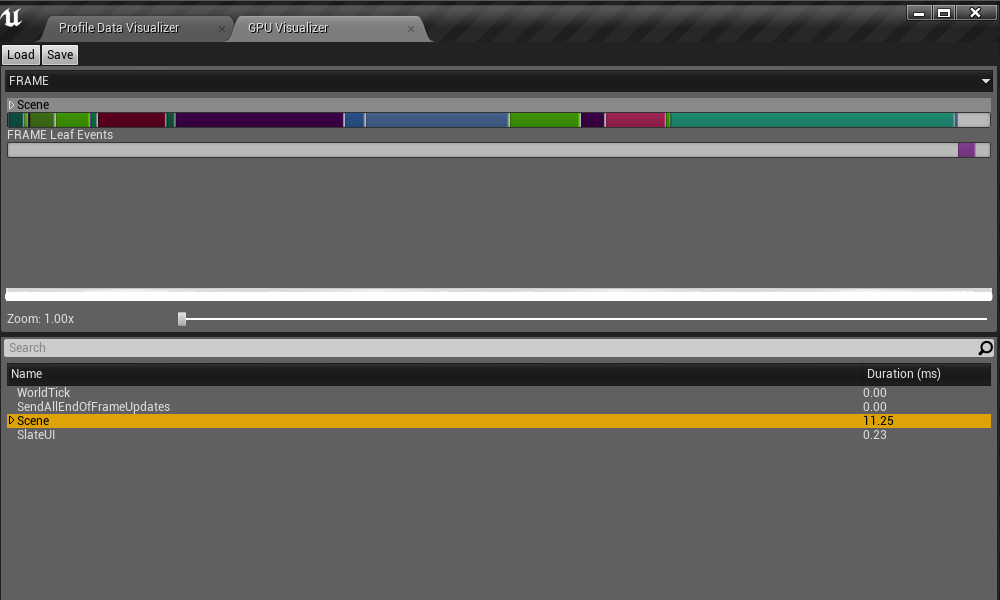
I recorded a base line before I optimized it and it was shocking to see that my project took a lot to run it here is an image on how it looked before I started optimizing:

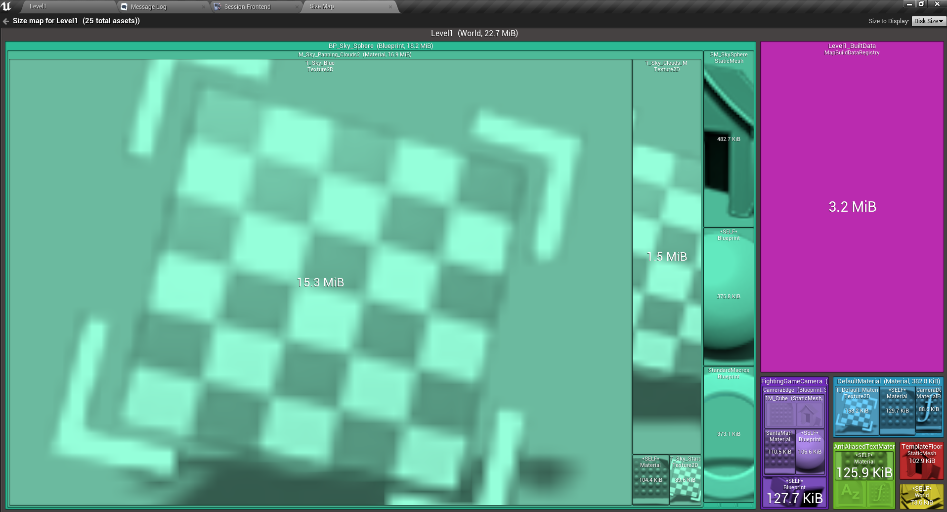
As you could see there are allot of spikes showing up in the profiler and you could actually feel it in the game when the spikes happen I found that the game thread was taking up a lot of performance so I delved into after I done this I went into standalone play mode in unreal to check if my project was Graphics based or CPU based, It turned out that my game was graphics based I know this because I went into my console in game and turned on the fps counter as well as the consumption in milliseconds as you could see here:



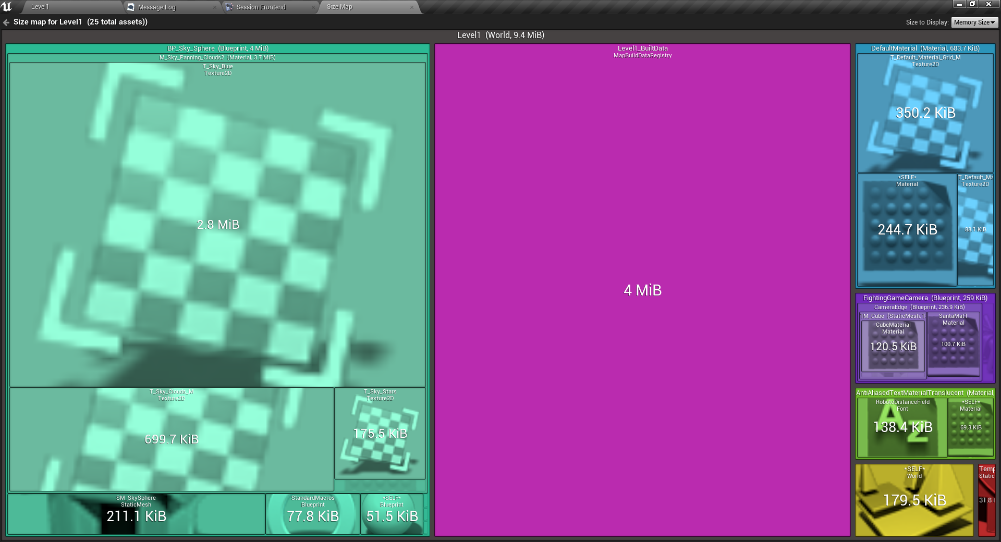
In this image if you look at the green writing there is an fps counter and below it is a millisecond tick for my one it was going at 16.13ms and to determine if this project is CPU or GPU bound is to look at the process that is matching the 16.13ms for this project the GPU matched up with that if it was CPU bound then where it says Game and Draw would be equal to 16.13 or 16ms then I would know that my project was CPU bound.

After this I pressed (Ctrl+Shift+,) to bring up the Project Visualizer and for GPU bound games this is the best way to optimize the project because it shows elements in the scene and you could sort it from high to low to see what is taking a lot of resources, without doing anything to my project this is how my Project Visualizer looked:

  
By doing this in the standalone play mode it only shows what the actual project is using and not the editor usage if I done it in editor the number would be higher and less accurate on the built project itself. As you could see in this image that my current scene alone was using 11.25ms and you can see by the bars that post processing is the largest majority of that then is lighting and after that would be the internal engine stuff that I can’t change if I did I might brake Unreal Engine its self.

Then I looked at the disk size map feature of unreal to see if there was any physically large assets in the project but there wasn’t any large assets to lower as you could see here: 

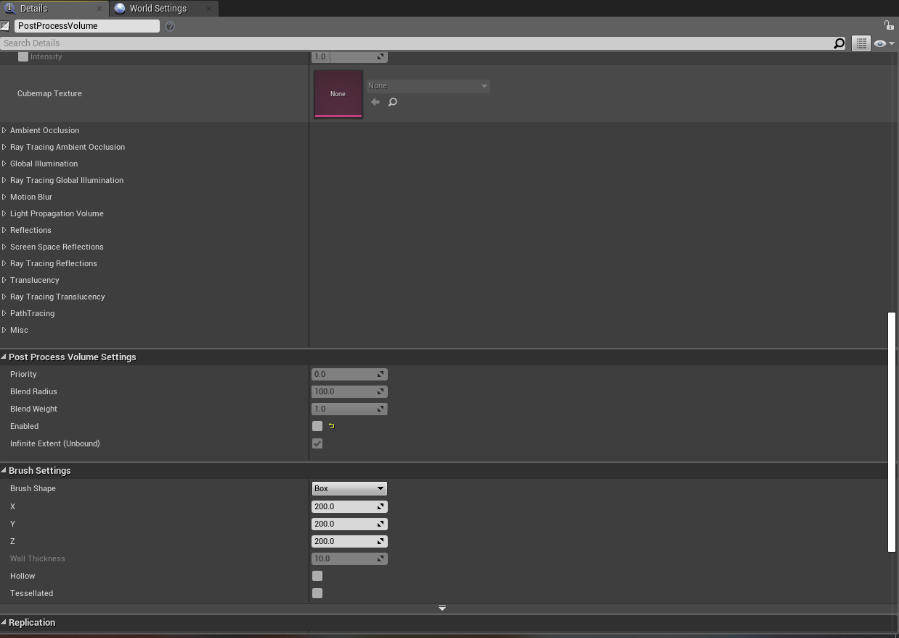
The largest asset I have is the 15.3MB of disk space if this was a bigger project this tool would be able to identify if an artist put a 4k texture and it would take you to the asset that would have that texture and you could optimize it from there this tool also shows how much memory each asset is using the way to see this is clicking where it says disk size and there will be an option for memory size which looks like this:



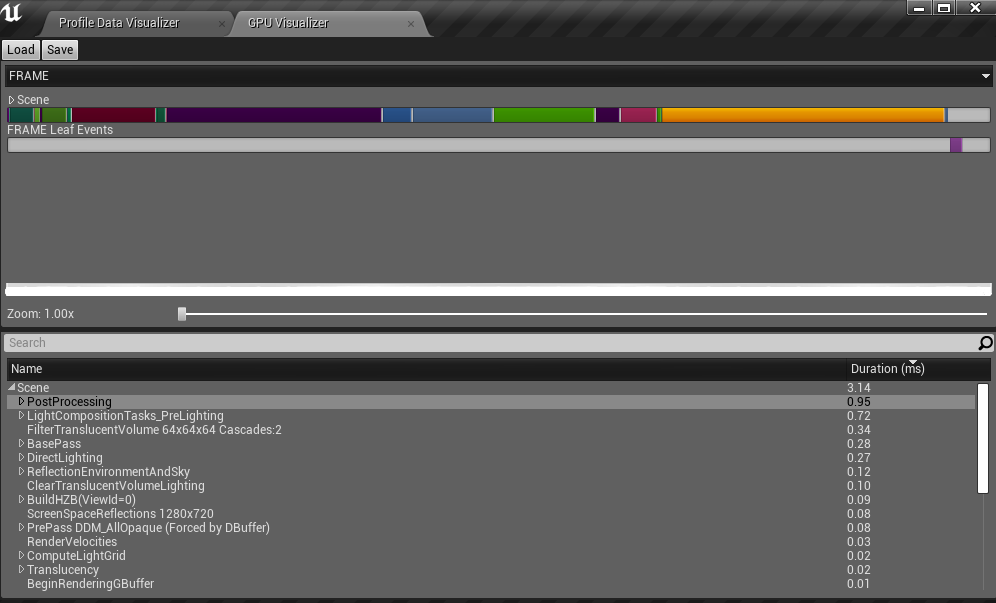
This looks similar to the disk setting and they have similar functionality but this shows how much ram everything is using in the scene when it is running, for me there wasn’t anything taking up a gig of ram the most it was using was 2.8MB of ram which isn’t too much but if this was a bigger project it would show what is using a lot of ram so you could optimize it this is a very powerful tool for bigger projects because it takes you directly to what is taking up the ram to optimize it which makes optimizing in unreal easier for me who has never done this before.

# What I Done to Optimize the project

# 3.1 Post processing Optimization

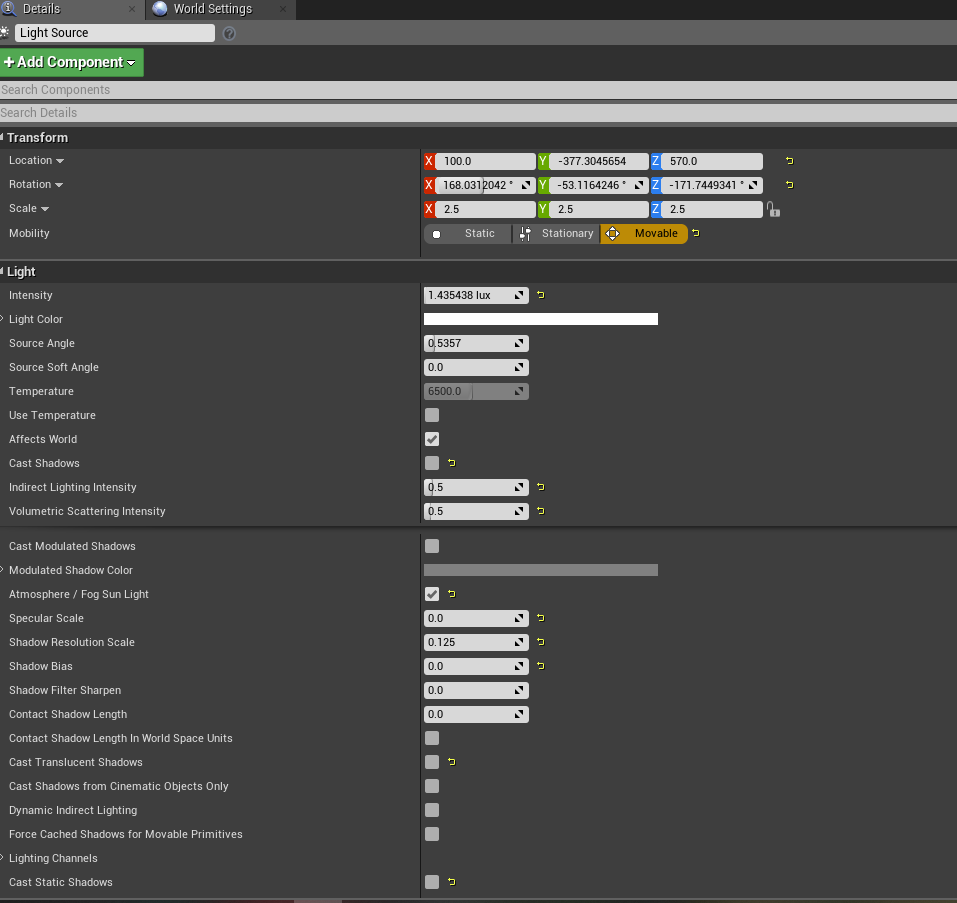
So the first step I took to optimize my project is to look at the post processing to see if my project needs it so I turned it off and it looked the same I screen shotted me turning off the post processing 

After I done this, I reran the Project Visualizer and there was an improvement as you could see in the image below:

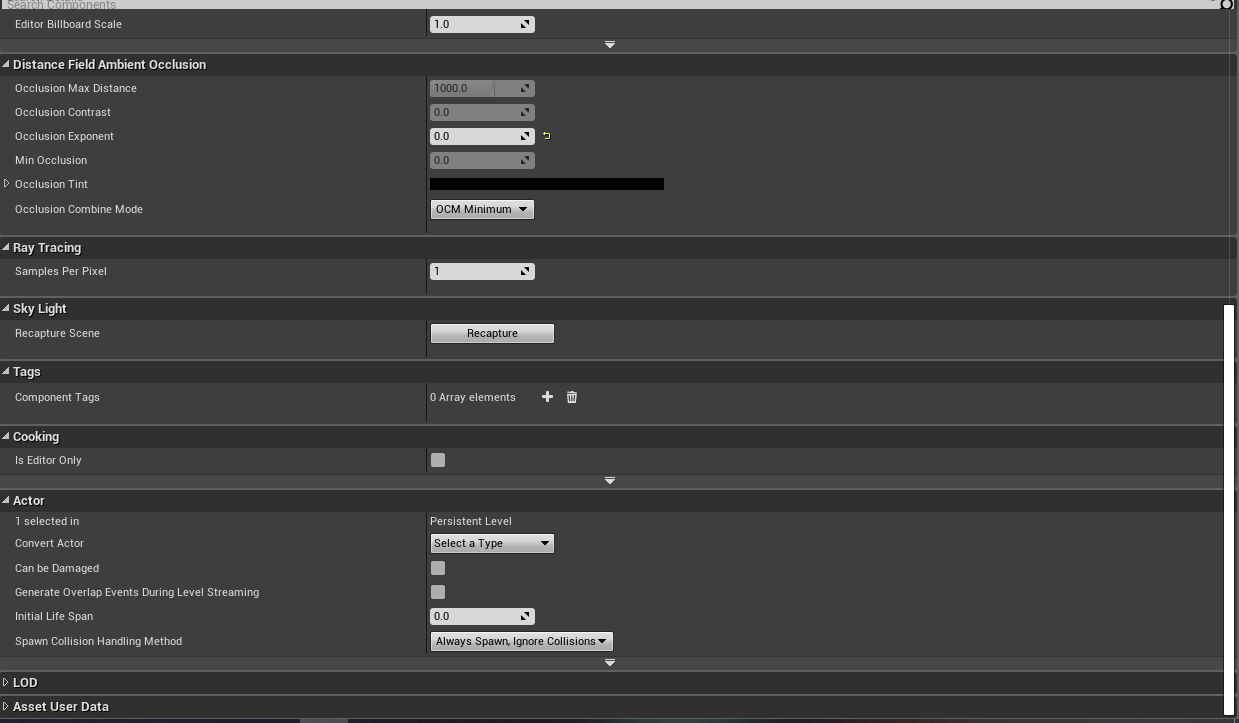
  
  
so just by removing post processing the usage for my project went down by 8.11ms to 3.14 which is a very drastic drop just by removing a component which was not very necessary for the overall project.

## **3.2 Lighting Optimization**

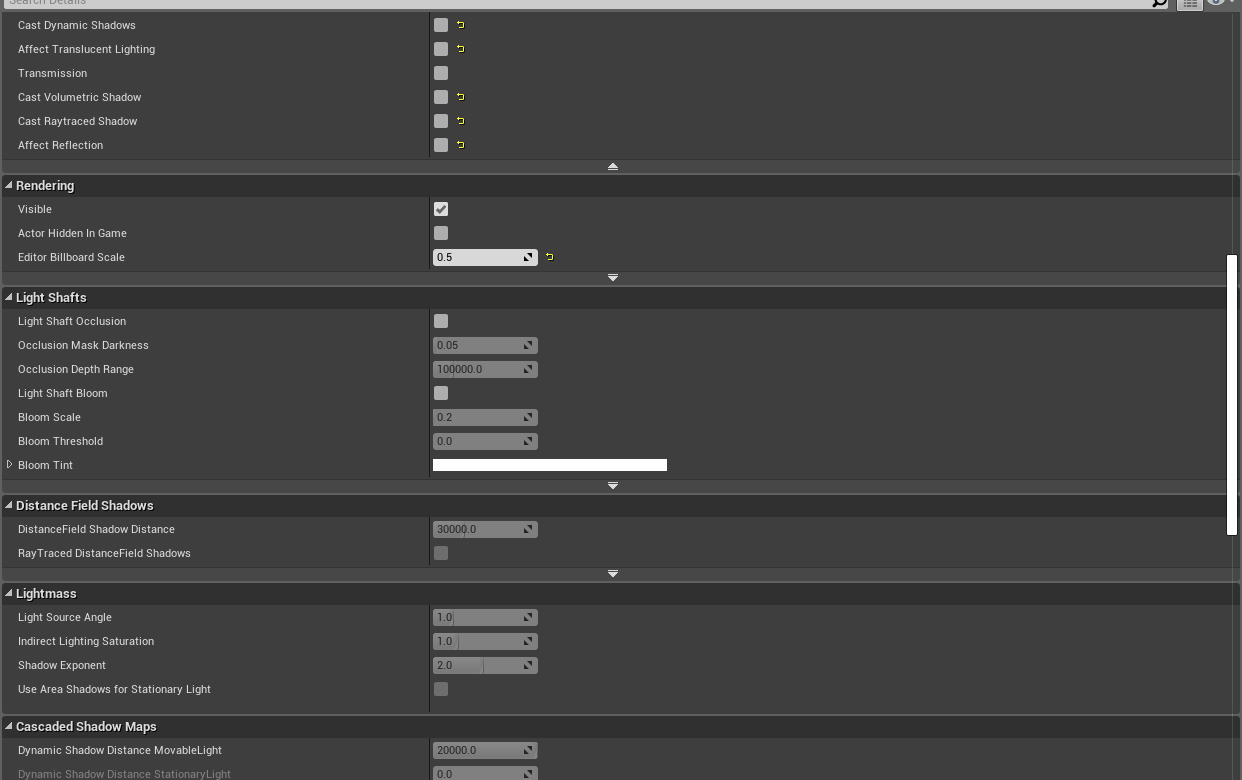
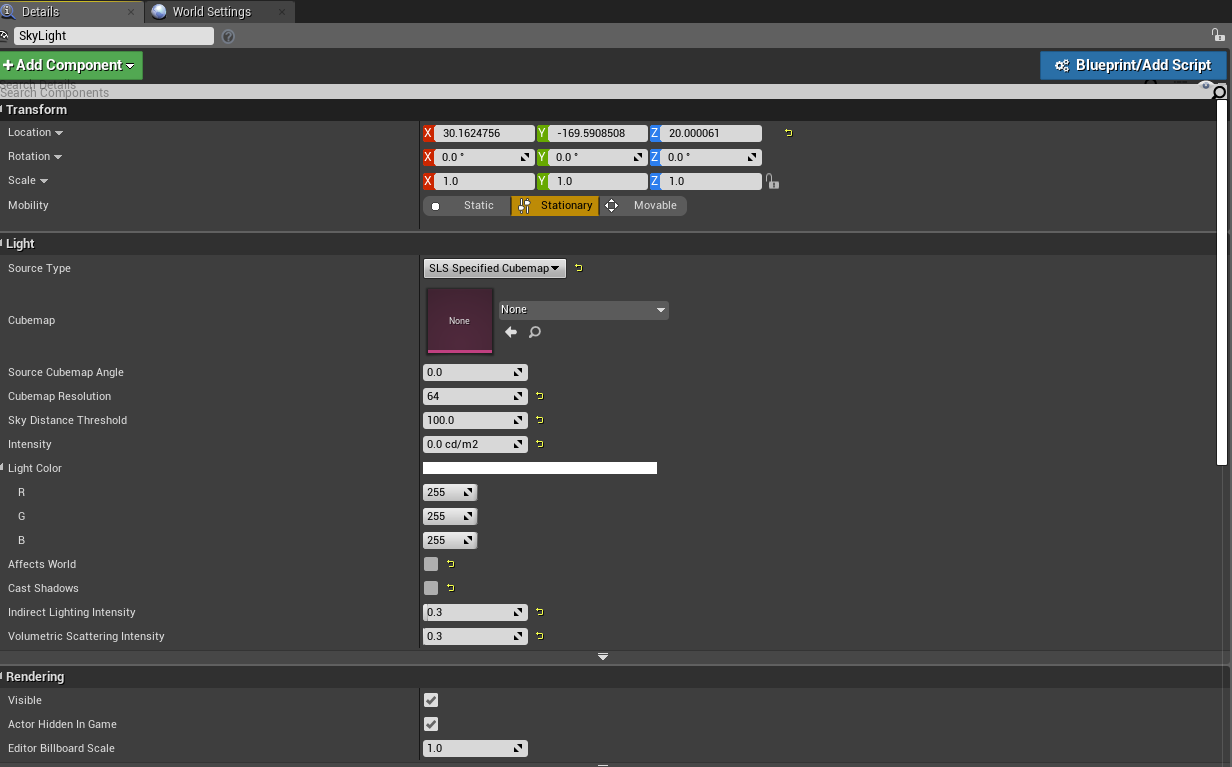
When I looked at the graph after I turned off post processing I saw that lighting was still taking up a lot so of resources, so I decreased the settings here is what I have changed in the lighting:



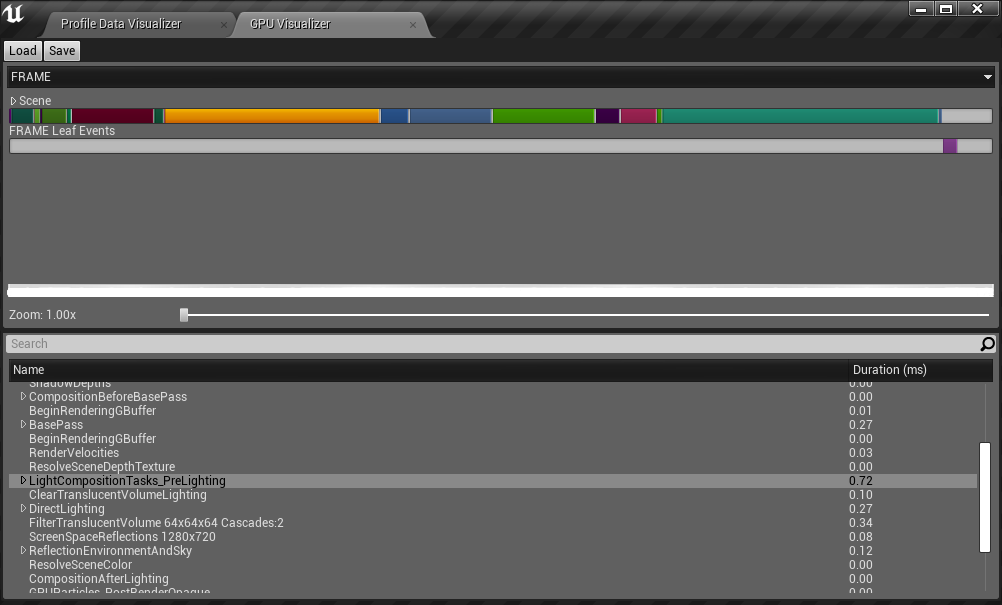
First I started with the light source I started turning random setting and looked at the editor view if it was having any effect if not I would turn it off or put it to zero as you could see in the pictures above and below



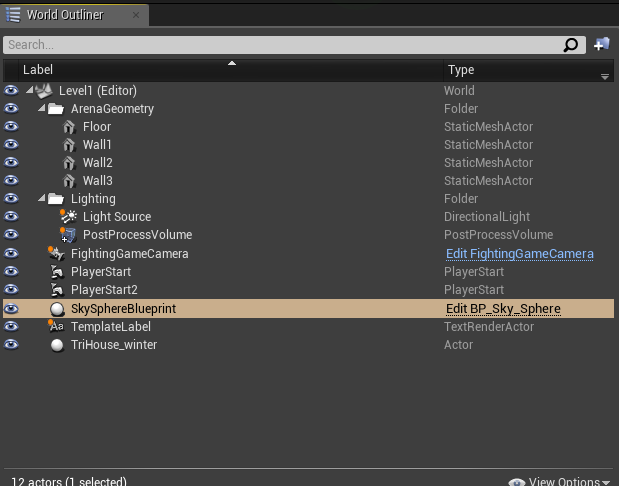
After I done the lighting I went over to the Skylight which is the sky box in unreal and turned off some setting and lowered overs I pretty much done it the same way as I did the previous lighting by lowering setting or turning them off to see if it effects the level and how it looks



After I done this, I opened up my Project Visualizer and It make a minimal difference as you could see below:

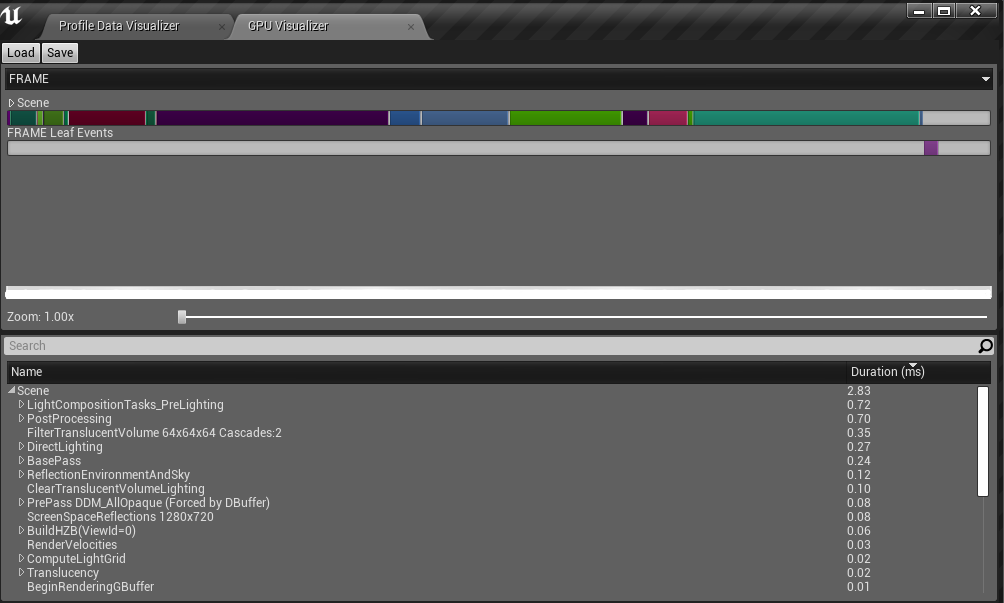


If you compare it to the previous one the boxes which are purple and dark blue which are the lighting there is not that much of a usage increase but there is still some if I had a more expensive project probably small differences make up a lot of performance so after this I was seeing if there was any way I could get it lower, then I went through my project visualizer as you could see below:



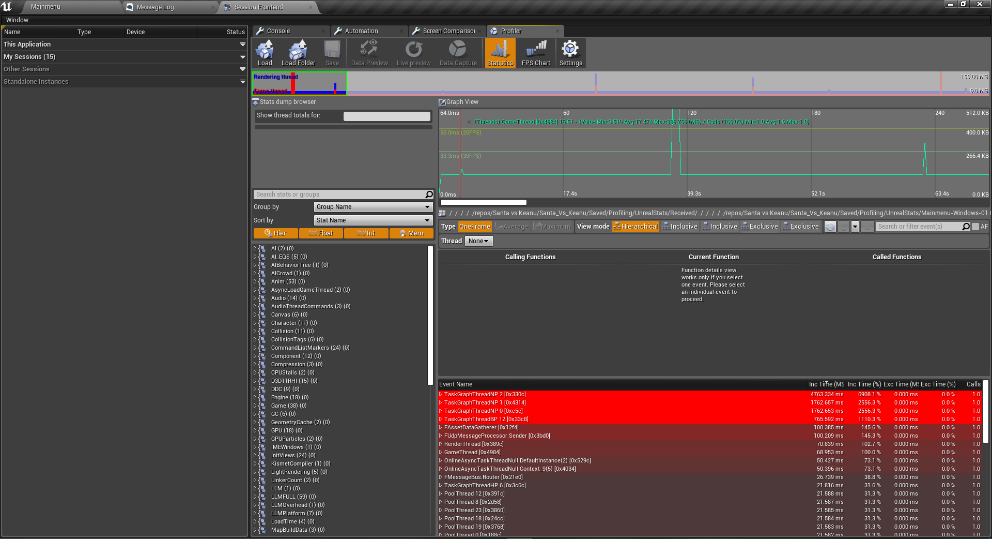
I saw that there was a lot of useless stuff in my world outliner, so I deleted the following items from the scene which were TriHouse\_winter, Template Label, Sky sphere. After I done this, I reran again the built project to open the Project Visualizer and this time I saw a difference in the Visualizer it drop drastically as you could

see here:



It went from 11.25ms down to 2.83ms that is a difference of 8.42ms saved on the overall project which is huge just by deleting stuff that the project does not need made it load faster and made it smoother to play.

Then I reran the Session Frontend profiler I noticed that there were less spikes in there now than it was when I started as you could see here:



There are some spikes but it’s unreal stuff that I can’t change because I don’t know how to read it properly, but I could see compared to the one I started off with the spikes on this one is smaller and more spaced out, so I think my optimization made a difference.

# The Tools I used and where I found them

# The tools I have used to optimize my project are the Graphics Visualizer, the Session Frontend profiler and Size Maps which I learnt from James and Lucy in there session on optimization I also researched into optimization for my C++ code but it was way to complicated so I decided not to do it my CPU usage wasn’t that high anyway to go through my C++ code but this video[1] is good at explaining benchmarking C++ code if I was in the mood to possibly destroy my project then I would probably do it but I have referenced it for anyone who wants or if I would want to do it in the near future.

# Conclusion In conclusion just by getting rid of unneeded actors in my project I have managed to improve load times of my project and I got rid of the stutter that happened when I would load a scene and I have now learnt how to properly optimize a project so if I am on a bigger project in the future I now have acquired the skills to do it and optimization in unreal is so much easier with the built in tool that show you exactly what is using what which was help full for beginner to start optimizing there project.

# References

[1] TheChernoProject. (2019, August 07). BENCHMARKING in C++ (how to measure performance). Retrieved January 04, 2021, from https://www.youtube.com/watch?v=YG4jexlSAjc