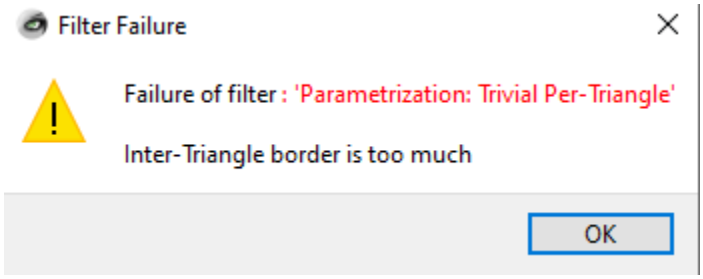
Problems with Meshlab :

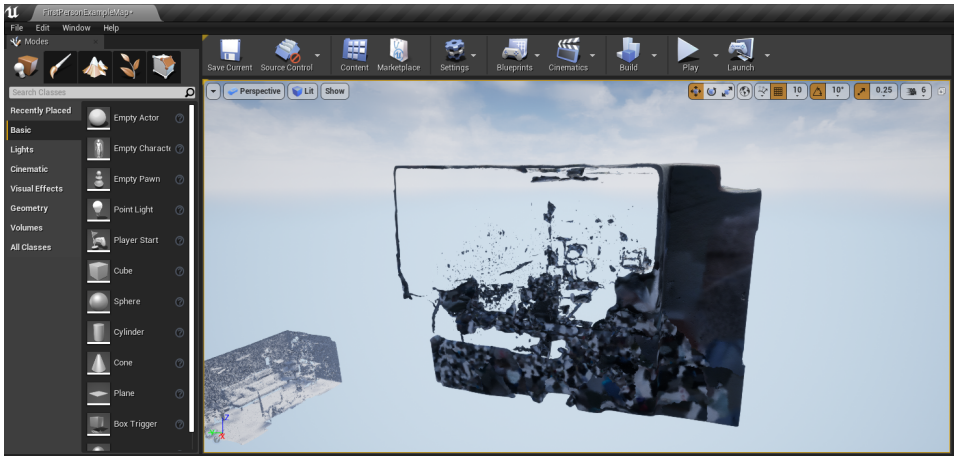
Detailed in the tests below, from our previous meeting on Janurary 29th, 2021. We have conducted long and arduous tests on generating meshes with Meshlab. There are two different types of mesh reconstruction that we can utilize to generate meshes with Meshlab. Ball pivot and poisson reconstruction.

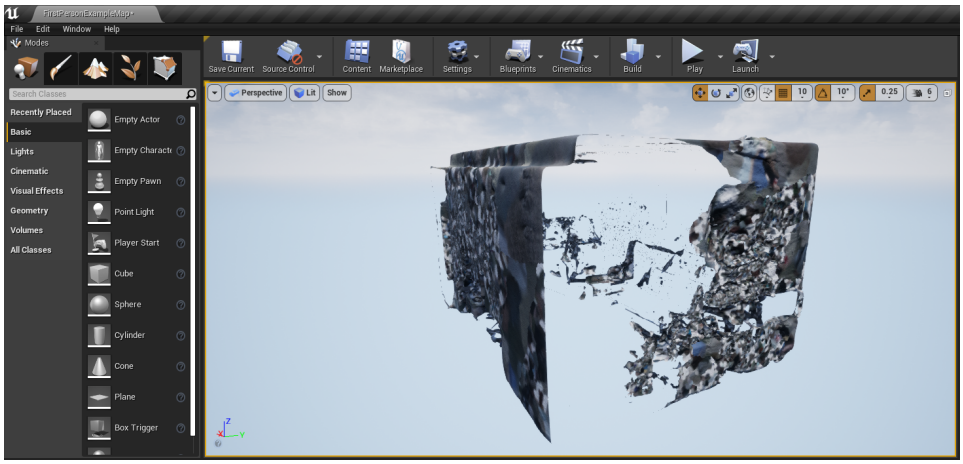
Ball pivot surface reconstruction starts with the normals of the point cloud and uses a triangle based algorithm to reconstruct the desecrated scan. This utilizes three points to form a triangle around an edge within a point cloud. It revolves around edges until all edges have been contacted. When it touches another point, it will then form another triangle and begin that process of edge detection over again. This process of generating triangles with edge detection occurs within a ball of increasing radius from the beginning point.

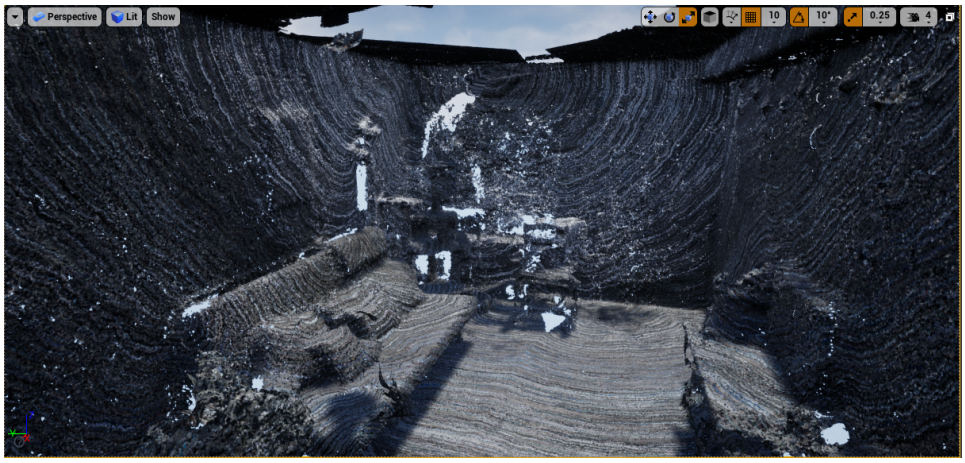
Poisson surface reconstruction is used to reconstruct 3d surfaces from point samples given to the program. It allows for the fitting of scanned data, the filling of surface holes within that scanned data and meshing of existing 3d models. Generally speaking, for both ball pivot and poisson surface reconstruction, as the number of vertices gets larger so does the number of faces. This in turn increases the amount of triangles required to generate an accurate mesh for the scan. There comes a point where, in Meshlab, there are too many faces and too many triangles for its parameterization trivial per triangle texture generation method.



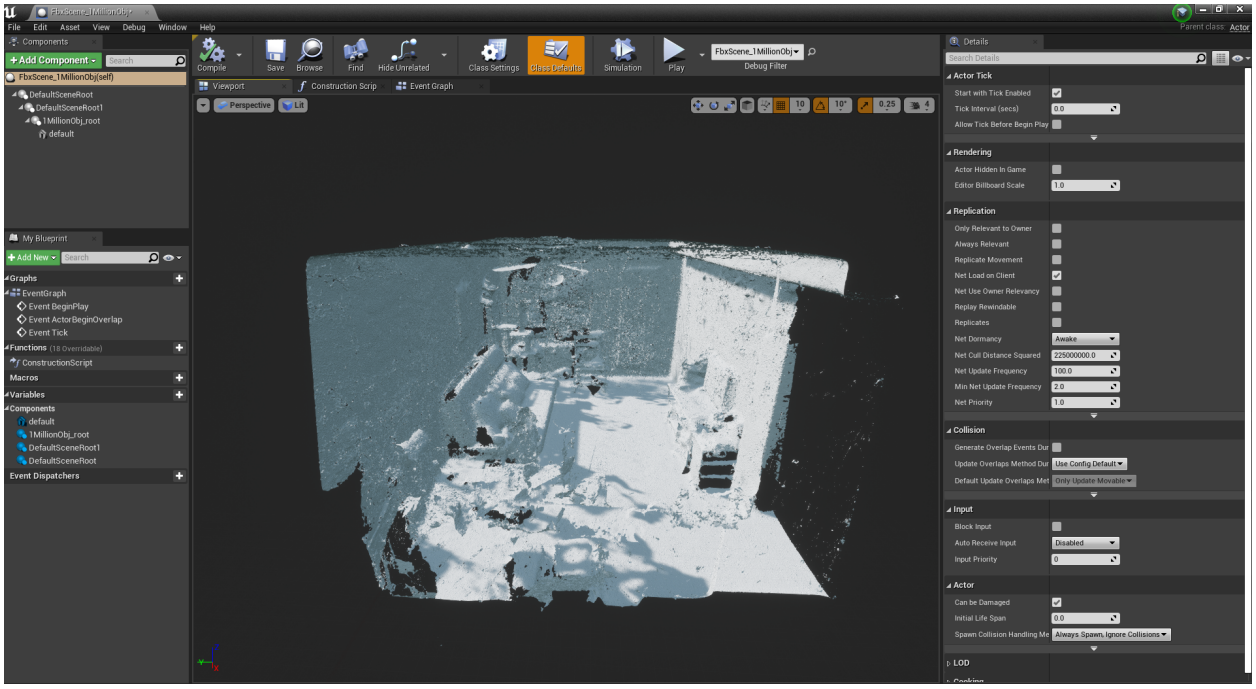
This means that the texture and mesh we are attempting to generate from Meshlab is too large for the program to handle. One quick mitigation that we can do to generate at least some kind of texture for our OBJ is to resize the resolution of our texture. However it is impossible to know what resolution our scan is going to need. Even if we found out what resolution our scans need, it would be impossible to determine that for each and every new individual scan coming into Meshlab. Here is the result of resizing the resolution texture for our scan in Meshlab. This is an exported OBJ of that.







As you can see, resizing the texture resolution causes the texture for the object to either not exist or completely be distorted in an unfixable matter. This is unavoidable. These are each three different trials of attempting to resize the resolution with a stable model : this is the stable model that the textures are attempting to be applied to :



It is also noteworthy that Meshlab takes **a very long time** to generate any kind of texture or mesh without crashing. It takes on average at least **one hour to three hours** to generate a mesh depending on the size of the scan you put into Meshlab. Therefore if we were to use Meshlab in the end, we would have to essentially ask our users, who would be using our application, to wait upwards of an hour to generate mediocre textures and meshes for their scan.

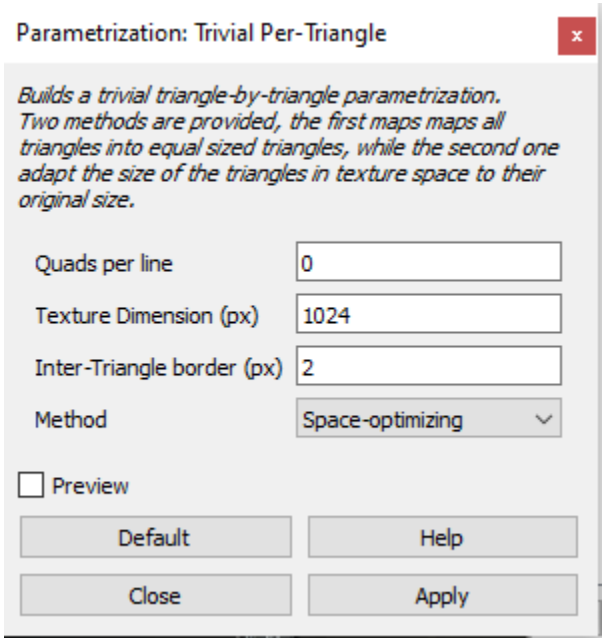
<u>Tests with self-scanned environment and Meshlab</u> :

# of Vertices in Scan	# of faces generated	Able to set parameterization per triangle	Can texture colors be mapped	Face creation method used	Texture outcome
54,718	101,729	Yes	Yes	Ball Pivot	Spikey and inaccurate. Texture is not applied to the scan and cannot be viewed in other programs.
54,718	126,602	Yes	Yes	Poisson Reconstruction	Textures are accurately mapped onto

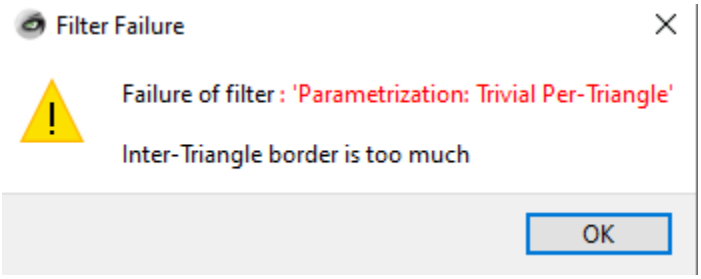
					the 3d model however texture is blurry.
389,175	715,393	Yes (Takes a very long time)	Yes	Ball Pivot	The same outcome as the first Ball pivot : spikey and inaccurate texture mapping.
1,279,652	2,186,398	No	No	None	No outcome.
2,215,564	3,885,134	No	No	None	No outcome.
61,688,364	Meshlab Crashes before faces can be made.	No	No	None	No outcome.

The reason why some of these in the table above have “none” listed for their face creation method is because they, 100% of the time, will crash Meshlab. Resulting in **NO mesh** or textures being generated for the object.

- Screen Poisson surface reconstruction creates “watertight” surfaces from the original point set. This generally tends to create smoother surfaces and textures.

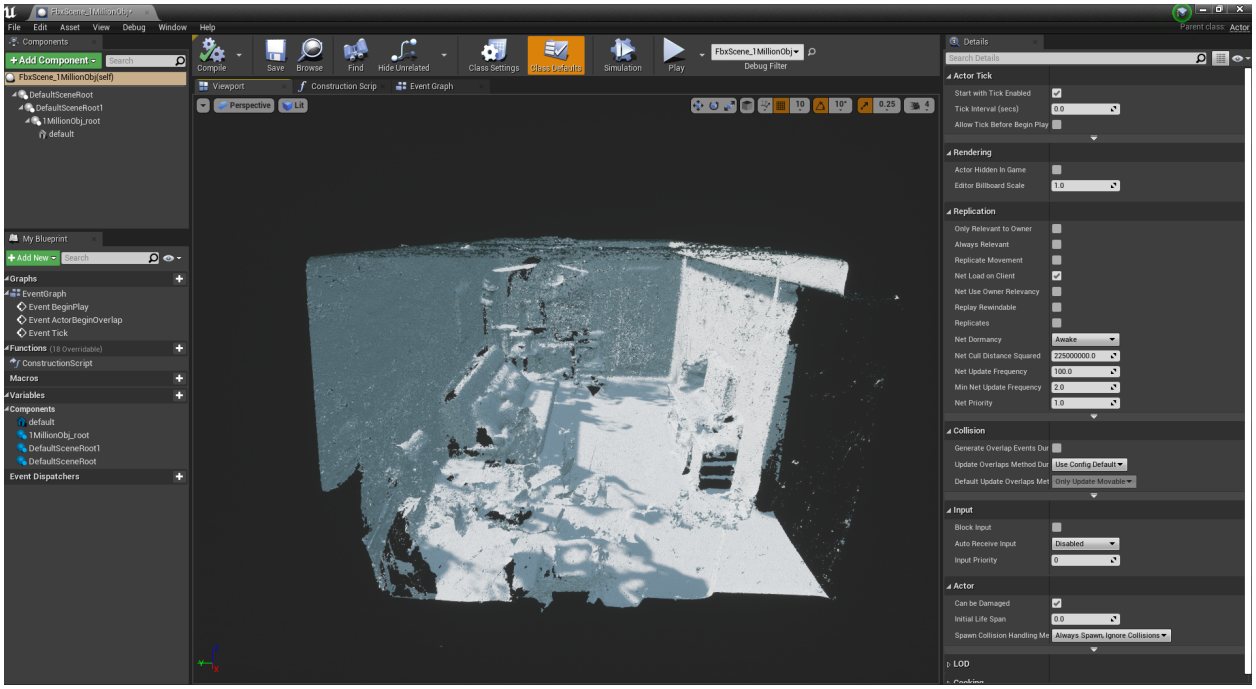


This sets up the texture for the 3d object. It is based solely on how many triangles there are between points within the point cloud that were generated from either of the surface reconstruction methods used. If there are too many faces then this issue comes up :

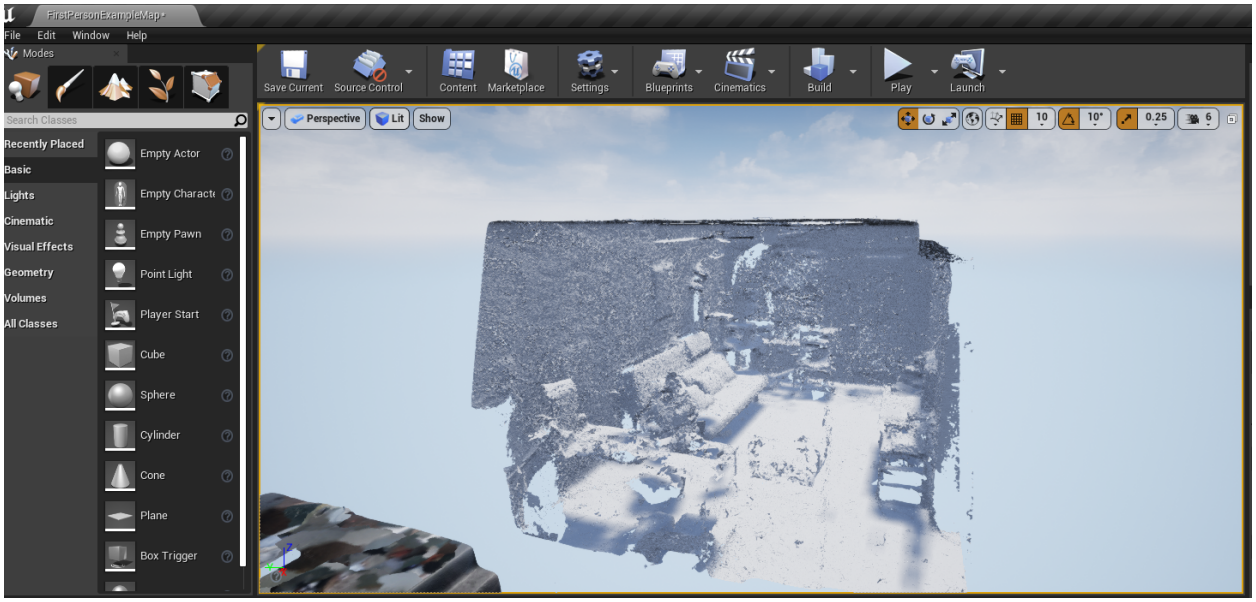


Our resource : [How-To: A Simple Way to 3D Scan an Environment – Intel® RealSense™ Depth and Tracking Cameras]

The screenshots below are the OBJ and material results of importing them directly from Meshlab into Unreal after generating meshes with Meshlab.



1,279,652	2,186,398	No	No	None	No outcome.



2,215,564	3,885,134	No	No	None	No outcome.

Ball Pivot info source :
[The ball-pivoting algorithm for surface reconstruction - Visualization and Computer Graphics, IEEE Transactions on (poly.edu)]