**Linear Texture Filtering Feature:**

Once again, I started off by reading most of the slides and getting a grip on the concept; it became pretty clear to me that I should interpolate different pixel colour values and take their average.  
At first I tried without using weights and taking the average of all neighbouring pixels and the original pixel itself.

Attempt #1:  
  
*(Left: Point sampling, Right: ‘Linear’ sampling 4x)*  
As you can clearly see, there is some linear filtering going on, but the colour became a lot darker and the actual rendering was a little bit offset. This is due to taking the average of every pixel and maybe some values on the actual diffuse map might not be used.

Attempt #2:  
  
*(Left: Point sampling, Right: Linear sampling 4x + weights + original pixel)*Now, I’ve been playing around with fixed weight values and this seemed to fix the darkness issue, also linear sampling is being applied correctly as well.  
And lastly, the original pixel colour value is now being applied fully, so the actual colour isn’t as different than before, only big noticeable and positive difference is the blurriness smoothly blending in with the adjacent colours of the mesh.

Extra attempt:  
  
*(Linear sampling 8x + weights)*  
I could go overboard with this result by sampling from all 8 neighbouring pixels, but this resulted too much in a blurry mess. Just like we’ve seen in GP2’s post processing.  
I decided to stick with the former method of sampling the 4 adjacent pixels and taking their respective weighted average.

The performance impact would also start to show the more samples I did per pixel.  
I’ve added the option to toggle the sampling state through the SceneManager.  
And the final render would look like this:

*Point:*  
 *Linear:*  


  
*Point vs Linear*

Sources:  
<https://cglearn.codelight.eu/pub/computer-graphics/textures-and-sampling>  
<http://www.essentialmath.com/OtherPubs/Texture_Filtering.pdf>  
<https://web.eecs.umich.edu/~jjcorso/t/598F14/files/lecture_0924_filtering.pdf>