**Title: Shadow Mapping**

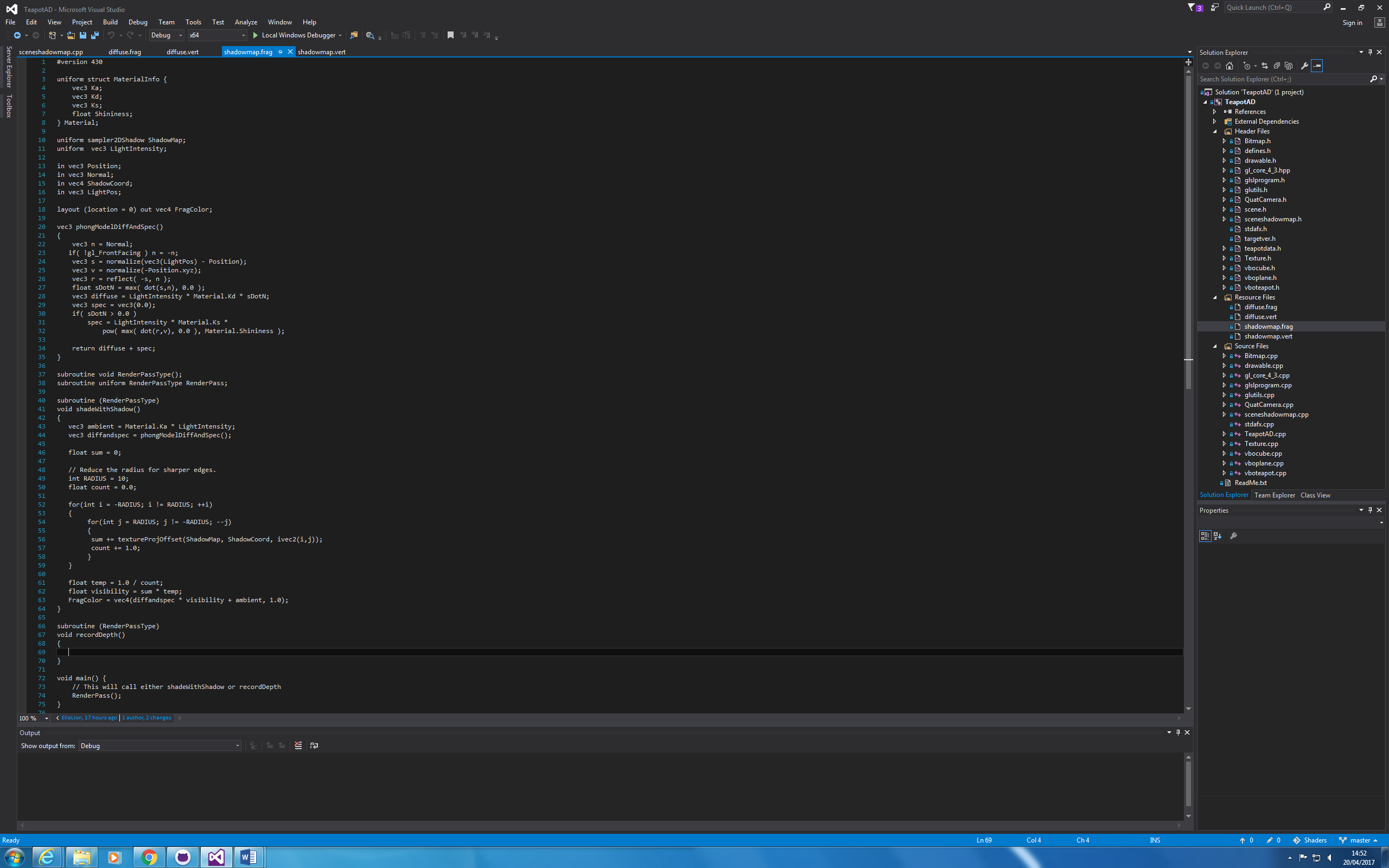
**1.0 Purpose**

To use the given vertex and fragment shaders to implement Shadow Mapping using two passes.

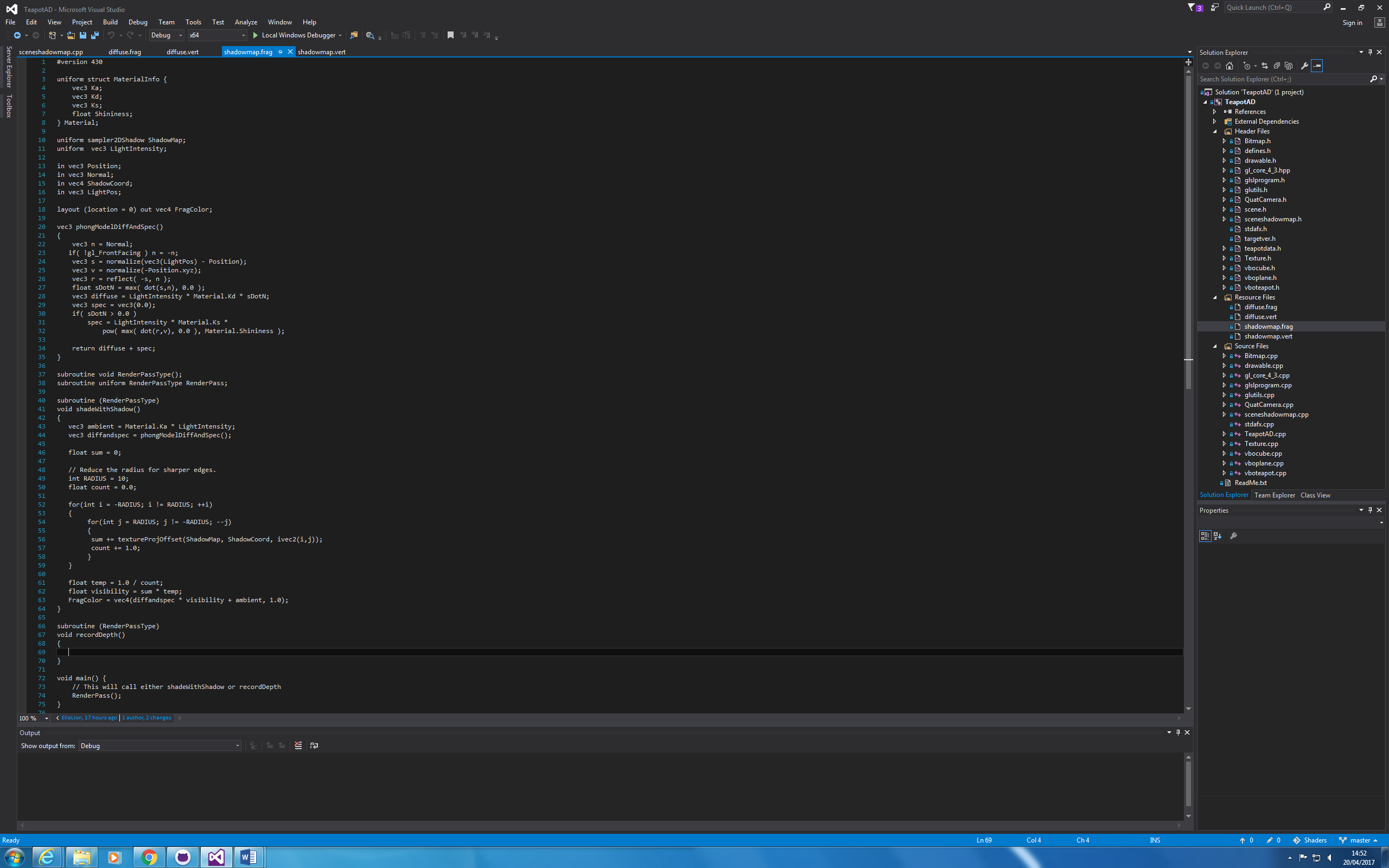
**2.0 Method/Results**

I firstly started in the fragment shader to complete a set of to-do tasks.

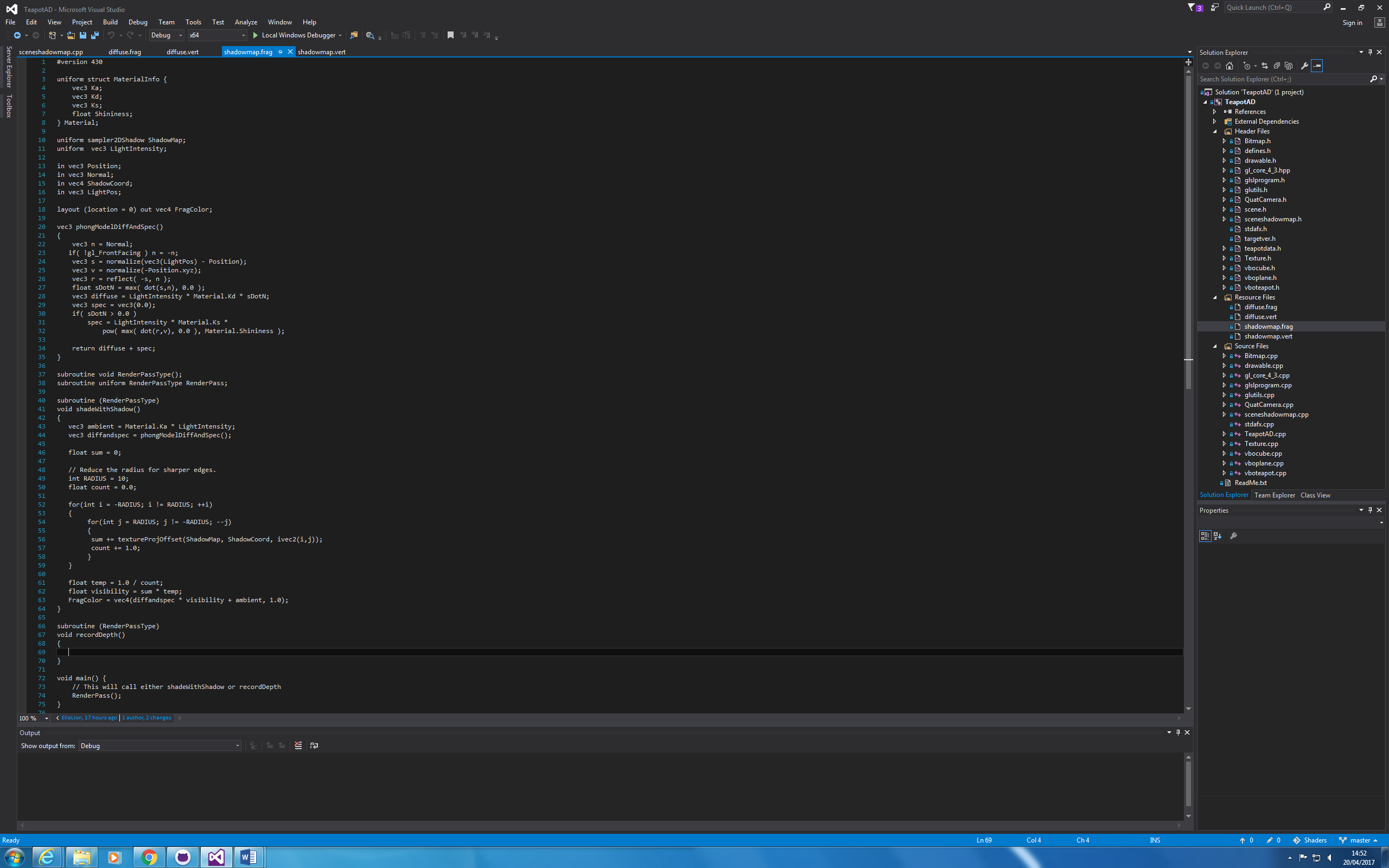
I had to create ambient light using LightIntensity and the material’s Ka value, this was done as below;



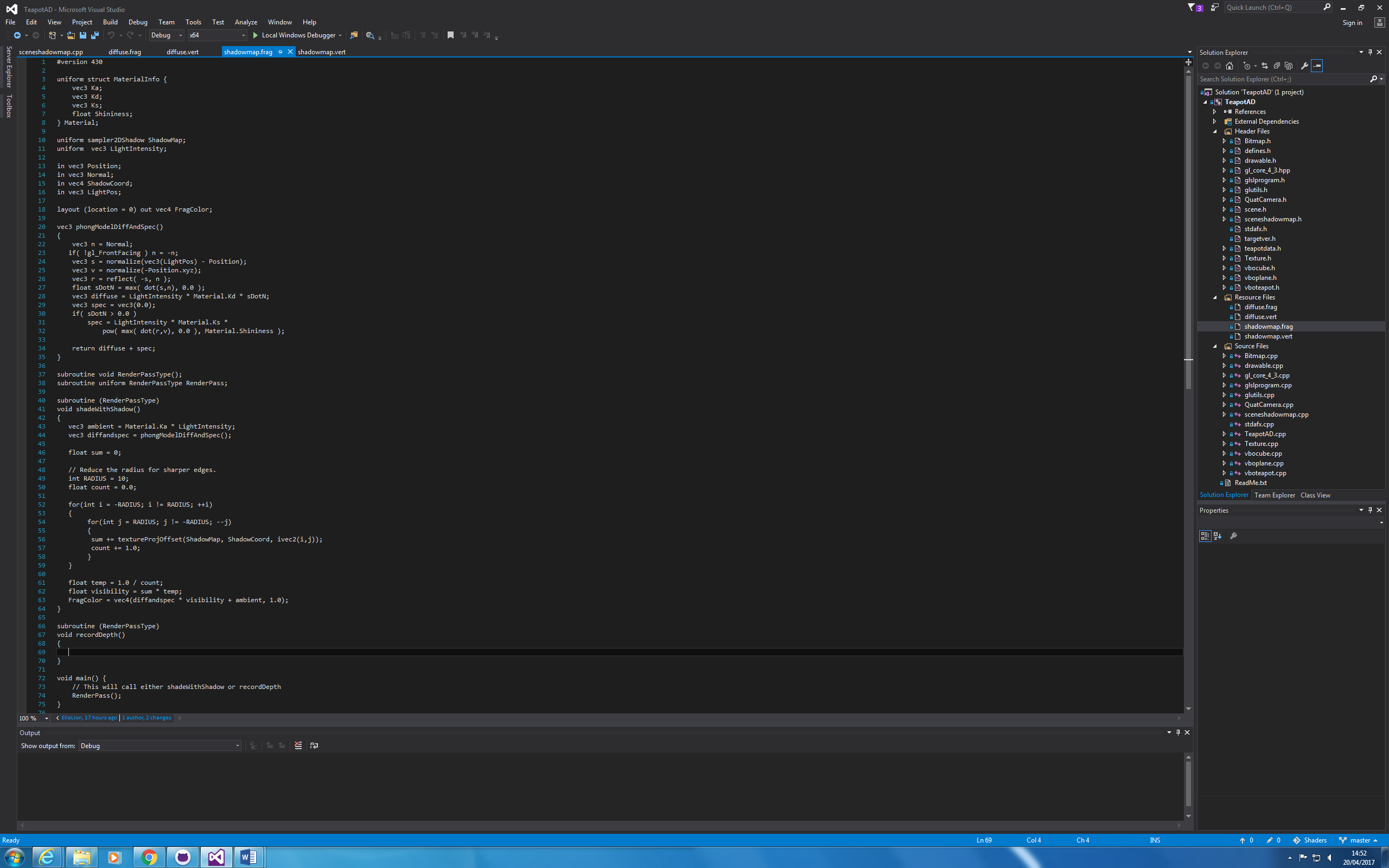
I then went onto calculating the diffuse and specular lighting;



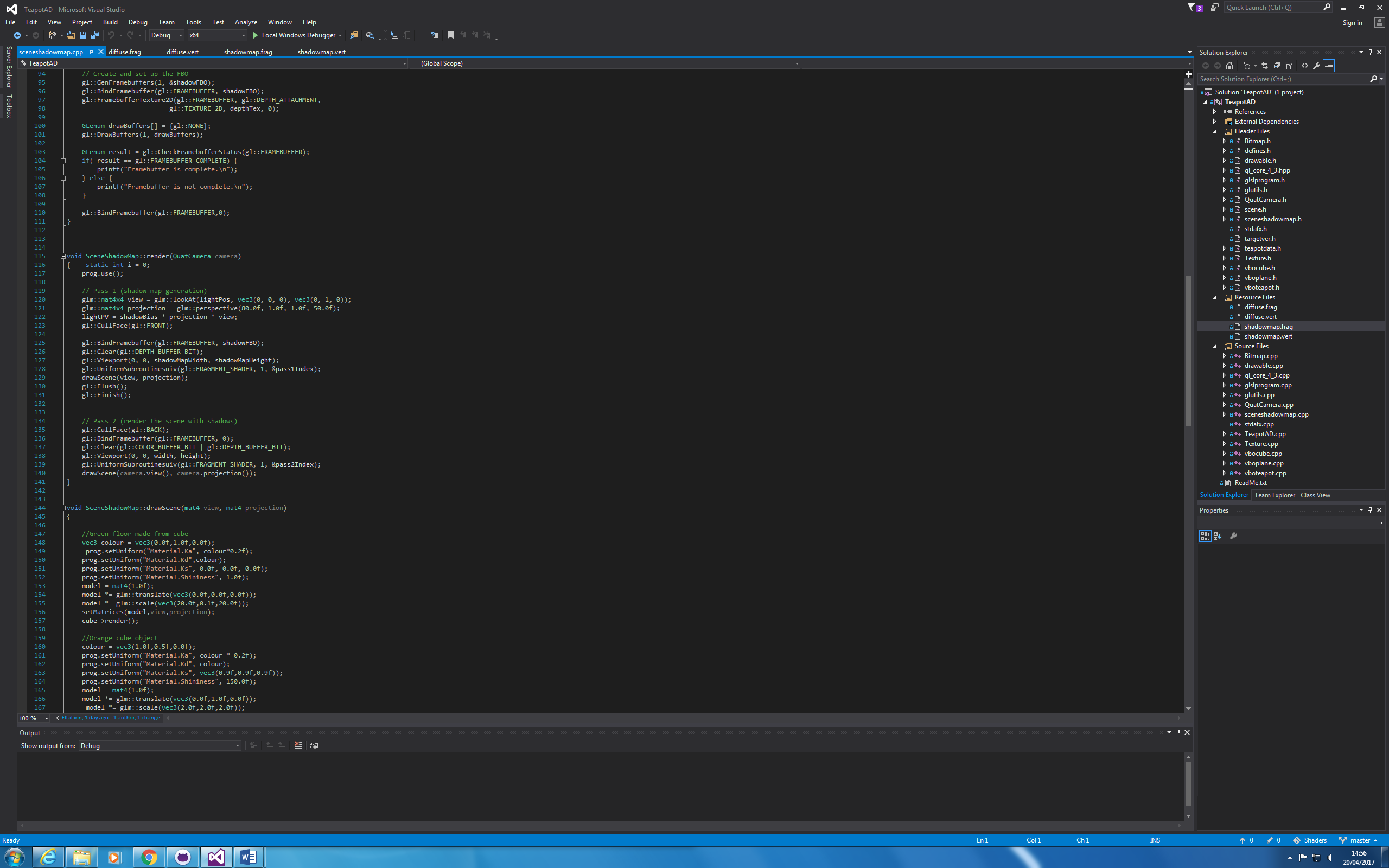
Then found a value for shadow, this was done by sampling ShadowMap using a different function called tectureProj;



Lastly, I set the FragColour, incorporating ambient, diffuse and specular, and shadow;



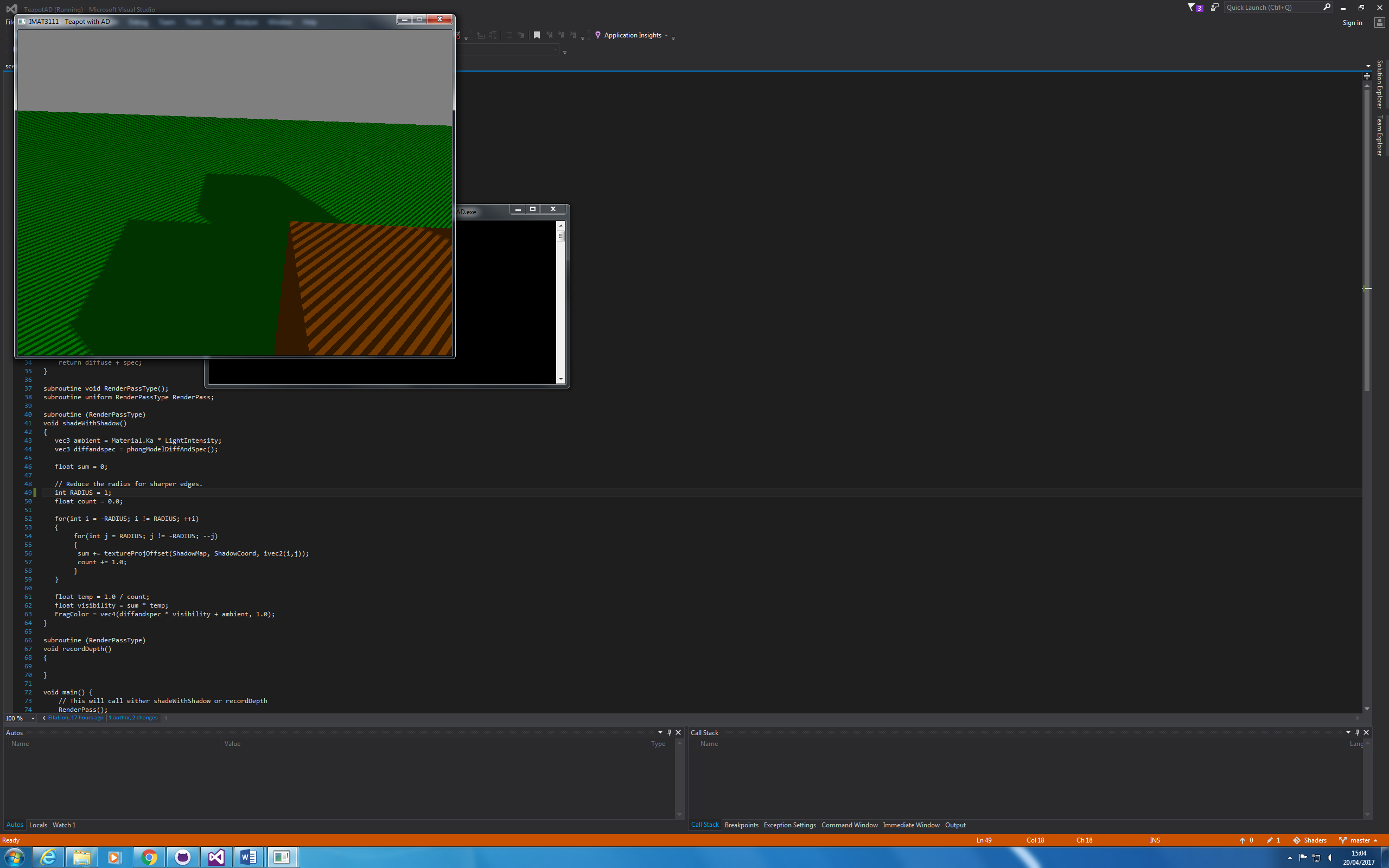
I then moved onto the to-do list in the cpp file provided, this was asking me to; set up a view matrix, suggest light is aimed at world origin, set up a projection matrix, and calculate variable lightPV.



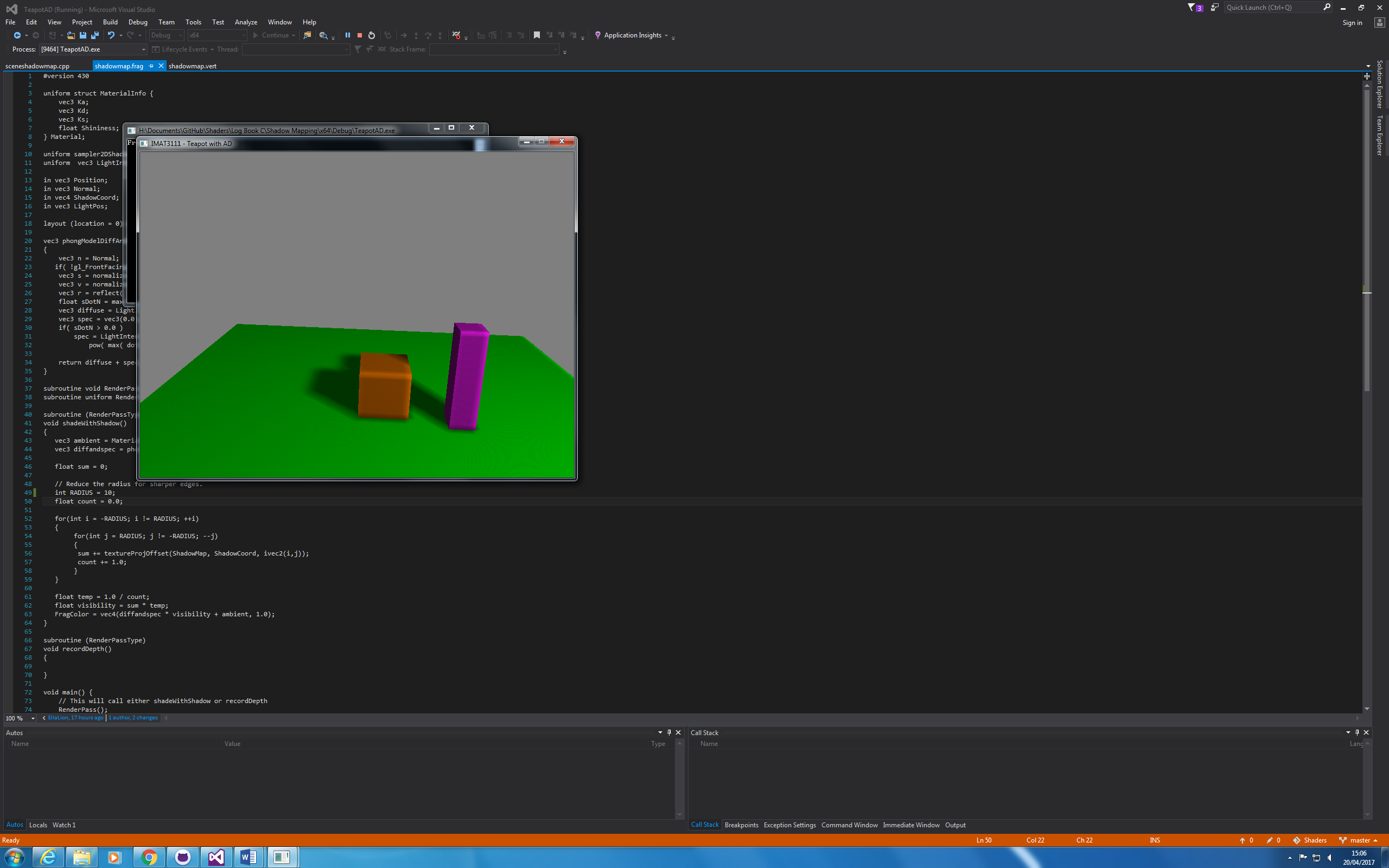
As you can see I have used Face Culling to get rid of the ‘acne’ problem.

I have also used PCF to soften some shadow edges.

**3.0 Comments/Conclusion**



With Acne, without shadow smoothing



Without acne, shadows substantially softened