|  |
| --- |
| Hogeschool van Amsterdam |
| Advanced Graphics Programming |
| [Type the document subtitle] |
|  |
| **bart dikmans** |
| Student Nummer: 500681415 Opleiding: Game Development Telefoon: 06-46270911 E-mail: bart.dikmans@hva.nl |

# Inhoudsopgave

[Inhoudsopgave 1](#_Toc447550675)

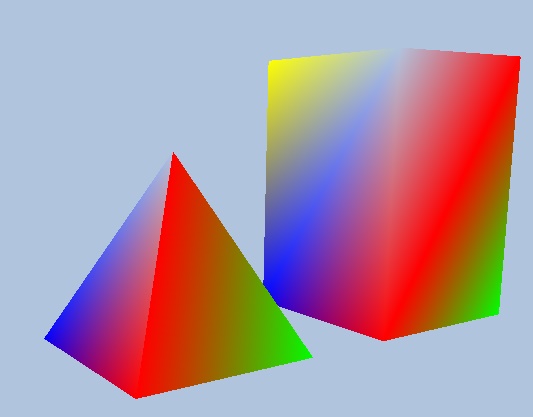
[Assignment 1 2](#_Toc447550676)

[Assignment 4 4](#_Toc447550677)

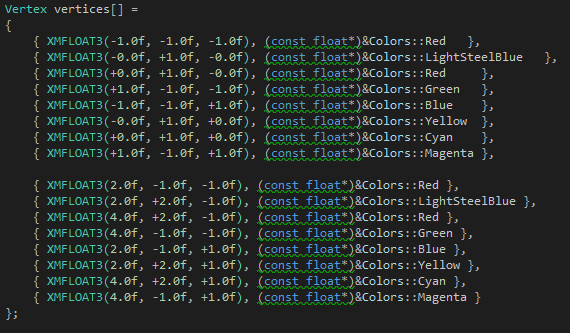
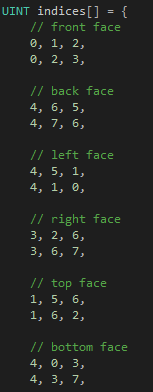
[Assignment 5 6](#_Toc447550678)

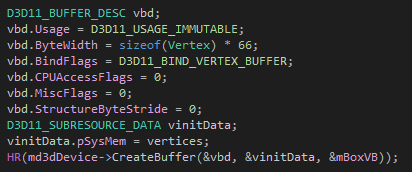
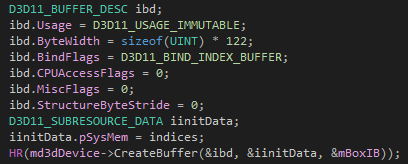
[Assignment 6 8](#_Toc447550679)

# http://puu.sh/o5KH5/bfd70880bc.pngAssignment 1

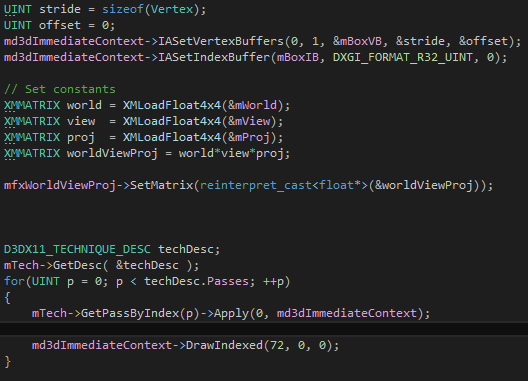


Output of my two programs. Left is the creation of a cylinder and 2 spheres through the Geometry generator and on the right a manually constructed pyramid and a rectangle.

To create the pyramid and rectangle one must first construct the vertices and after that connect the vertices with multiple indices like the following:

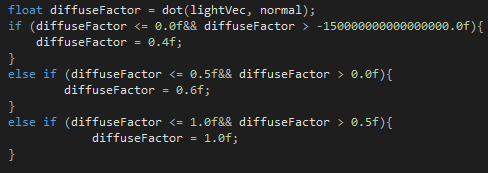
These two arrays will be inserted into vertices and indices buffers to be used to create the objects in the drawScene function. Following are the insertion into the vertex (left) and indices (right) buffers

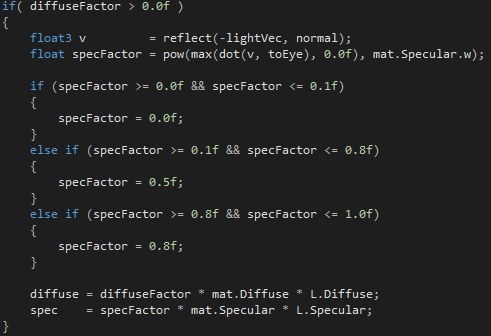
After creating the buffers the following is used to draw the objects stored in the buffers



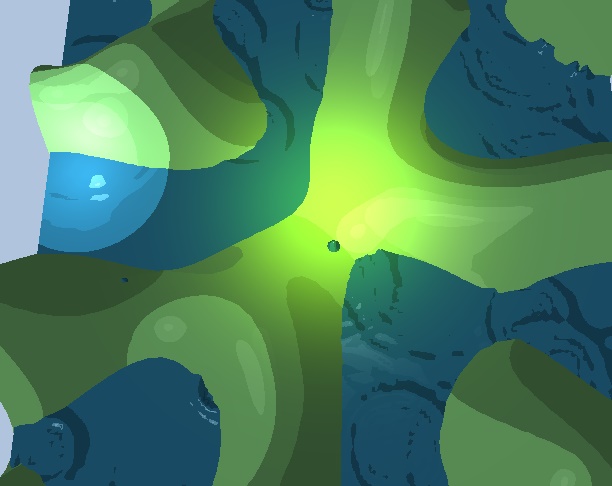
# Assignment 4

In assignment 4 I changed the shader of every lightsource to a toon shader. This was done by adding the following lines of codes to every lighting source which exists in the Lighthelper.fx file.



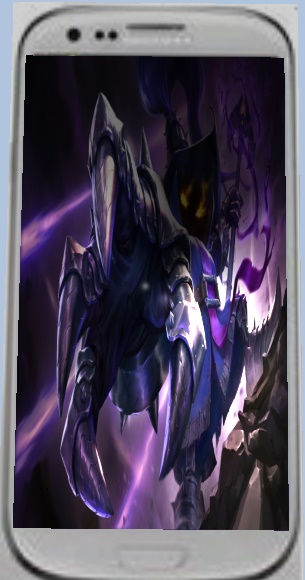


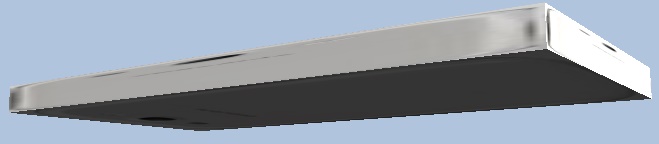
This code means that the instead of the lighting being smooth it now has multiple threshholds in which it sets itself to a set number. This creates multiple noticeable layers in the scene which can be seen below



In this image there are 3 different lighting sources which all have the beforementioned code included to create the multiple layers.

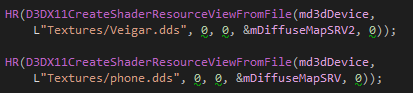
# Assignment 5

For this assignment I created a box with a quad and textured both of them to make the end result look like a mobile phone with a picture of a League of Legends champion. In the following 2 images the end result can be seen. In the first image there is a top down view which shows the low quality phone and champion. In the second image there is a view from the side and bottom.



At first I create a box and a quad using the same methods used in assignment 1. I add a small detail to create the textures on the objects.

Using the following lines of code I initialize 2 different Diffuse Maps with 2 different .dds textures. These .dds textures are created by saving any image in the directx texture tool.

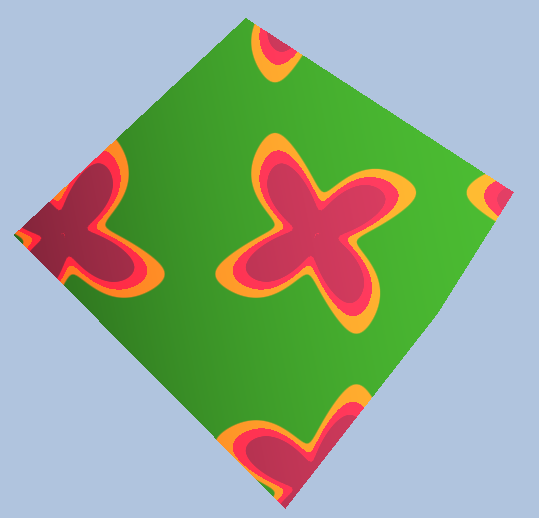


After this during the drawscene method I start of by creating the objects the same way as in assignment 1 except for the following line of code.

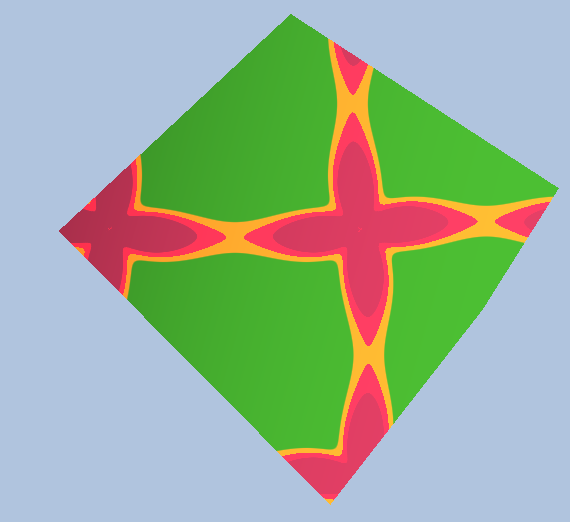
http://puu.sh/o5Mku/202ef6f86f.png

This line adds the diffusemap to the object which results in the object having a texture.

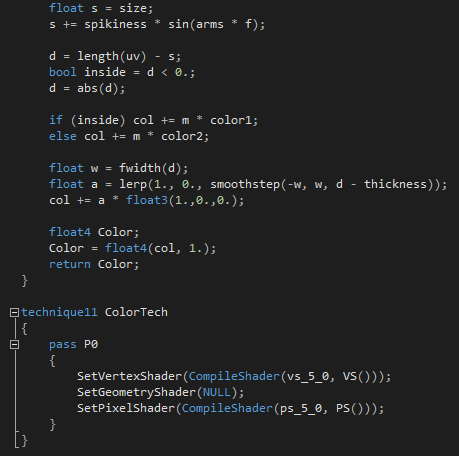
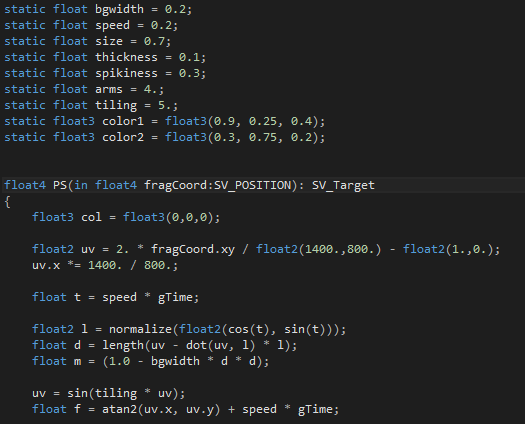
# Assignment 6

During assignment 6 I added a shader to a box creating the following effect.

This effect creates a small rotating and expanding/retracting windmill. One of the following steps can be seen in the next screenshot



This shader can be obtained by implementing the following code in the color.fx file



The shader requires the current pixel (which is the fragCoord:SV\_POSITION) aswell as as the static floats declared above the PS float 4. Using these variables one can adjust the almost everything in the shader like the amount of spikes or the colors. After going through multiple calculations the float4 PS returns a RGBA value which is given to the pixel in the ColorTech technique in the setPixelShader method.

This shader was imported from shadertoy however shadertoy uses multiple different variables which are not usable in the luna projects. Some examples of this are the usage of Vector3 and Vector4 in the shadertoy shaders, these have to be converted to float3 or float 4’s. I got the information from the following link:

<https://alastaira.wordpress.com/2015/08/07/unity-shadertoys-a-k-a-converting-glsl-shaders-to-cghlsl/>

The following screenshot is part of the list on the website in case the website does not load.

