

CSI4130 Computer Graphics

Winter 2020: Assignment 3

Due: Wednesday, March 23th, 2020, 11:00pm in Virtual Campus
University of Ottawa - Université d'Ottawa

Jochen Lang

1 Material and Lighting [10 in total]

This assignment is based on the box laboratories with materials and lighting of March 12th, 2020. You should use this code as a start for your WebGL2 solution. (Your solution must be WebGL2 and not use THREE).

2 Spheres [2]

Create a class for modelling a sphere as shown in Fig. 1. You can reuse much of the `BoxShape` and associated classes from the laboratory. Your sphere should enable drawing with smooth and flat shading. You will need to use two different face lists. You are required to use `element draw`.

A list of vertex position and an indexed face list are given in the Appendix.

3 Spinning Grid of Spheres [1]

Use the sphere from Question 2 and place them in a 5×7 frontoparallel grid. Use one `element draw` for all the spheres. Use an animation similar to the lab to have all the spheres spinning while using instanced drawing. (Note: If you can't get the sphere to work, you can use the cubes from the lab).

4 Light Source [3]

Use a single directional light source and a material for the spheres (see also Question 5). Give the user the ability to change the ambient, diffuse and specular light source strength through the

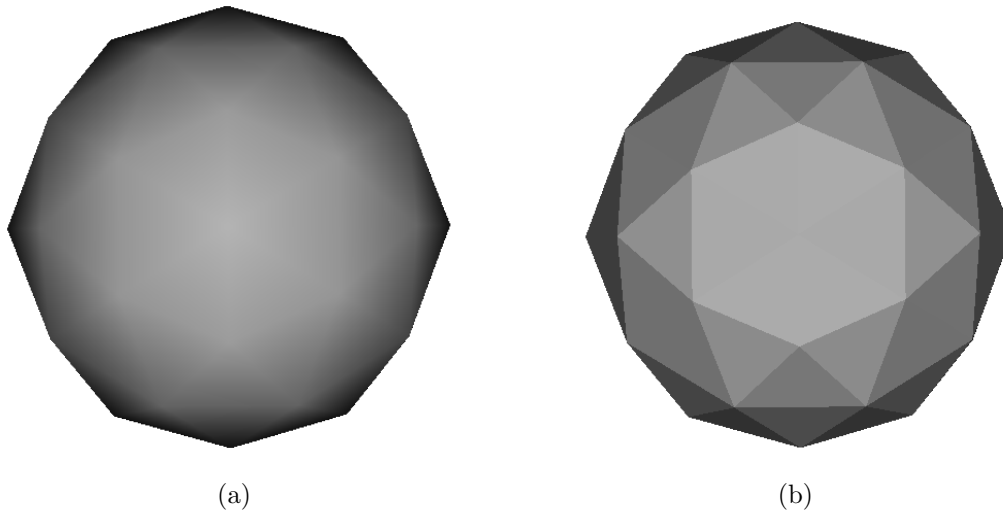


Figure 1: Sphere (42 vertices, 80 faces) rendered with smooth shading in Fig. 1(a) and flat shading in Fig. 1(b)

dat.gui interface. Please remove the second light source and the options to switch to point or spot light source. Please remove the unnecessary light parameters from your javascript and shader code. Calculate the lighting model per fragment.

The user should also be able to change the direction vector of the light source. Control the position of the light sphere with 'a', 'w', 's' and 'd' keys.

5 Materials [3]

Add controls for a single material for all spheres. The user should be able to control the emissive, ambient, diffuse and specular colours, as well as the shininess exponent of the Blinn-Phong (OpenGL hardware) model.

Vary the material for each sphere based on the instance id. The emissive, ambient, diffuse and specular colours should vary in the first four rows of the grid, respectively. Scale the colour parameters from left to right by $s_{colour} = \begin{bmatrix} 0.5 & 0.9 & 0.95 & 1 & 1.05 & 1.1 & 1.5 \end{bmatrix}$. Make sure to clamp the values of each channel at 1. In the fifth row of the grid vary the shininess parameter (exponent) in a logarithmic scale $s_{log} = \begin{bmatrix} 0.1 & 0.2 & 0.5 & 1 & 2 & 5 & 10 \end{bmatrix}$.

Add a checkbox in dat.gui to switch between flat and smooth shading.

6 Bonus: Bump Mapping [2]

Add another row of sphere for which you use bump mapping. The *heights* of the bumps should vary as the colours from left to right.

7 Submission

You are allowed to use any library also used in WebGL2 laboratories. You are allowed to reuse code from the laboratory. Please submit your webpage, javascript and shader code in one *zip archive*. As in the lab, point your webpage to

```
"../lib/webgl-utils.js",  
"../lib/webgl-debug.js",  
"../lib/cuon-utils.js",  
"../.. /node_modules/gl-matrix/gl-matrix.js", and  
"../.. /node_modules/gl-matrix/dat.gui/build/dat.gui.js".  
(Please do not submit those).
```

Appendix

The 42 vertex position of the sphere are given as:

0.00000	0.85065	0.525731
0.00000	0.85065	-0.525731
0.00000	-0.85065	0.525731
0.00000	-0.85065	-0.525731
0.85065	0.52573	0.000000
0.85065	-0.52573	0.000000
-0.85065	0.52573	0.000000
-0.85065	-0.52573	0.000000
0.52573	0.00000	0.850651
-0.52573	0.00000	0.850651
0.52573	0.00000	-0.850651
-0.52573	0.00000	-0.850651
0.00000	1.00000	0.000000
0.50000	0.80901	0.309017
0.50000	0.80901	-0.309017
-0.50000	0.80901	-0.309017
-0.50000	0.80901	0.309017
0.00000	-1.00000	0.000000
0.50000	-0.80901	-0.309017
0.50000	-0.80901	0.309017
-0.50000	-0.80901	0.309017
-0.50000	-0.80901	-0.309017
1.00000	0.00000	0.000000
0.80901	-0.30901	-0.500000
0.80901	0.30901	-0.500000
0.80901	0.30901	0.500000
0.80901	-0.30901	0.500000
-1.00000	0.00000	0.000000
-0.80901	-0.30901	0.500000
-0.80901	0.30901	0.500000
-0.80901	0.30901	-0.500000
-0.80901	-0.30901	-0.500000
0.00000	0.00000	1.000000
-0.30901	-0.50000	0.809017
0.30901	-0.50000	0.809017
0.30901	0.50000	0.809017
-0.30901	0.50000	0.809017
0.00000	0.00000	-1.000000
-0.30901	0.50000	-0.809017
0.30901	0.50000	-0.809017
0.30901	-0.50000	-0.809017
-0.30901	-0.50000	-0.809017

and the corresponding 80 faces:

12	13	14
12	15	16
17	18	19
17	20	21
22	23	24
22	25	26
27	28	29
27	30	31
32	33	34
32	35	36
37	38	39
37	40	41
35	25	13
16	29	36
14	24	39
38	30	15
19	26	34
33	28	20
40	23	18
21	31	41
1	12	14
12	0	13
14	13	4
0	12	16
12	1	15
16	15	6
2	17	19
17	3	18
19	18	5
3	17	21
17	2	20
21	20	7
4	22	24
22	5	23
24	23	10
5	22	26
22	4	25
26	25	8
6	27	29
27	7	28
29	28	9
7	27	31
27	6	30
31	30	11
8	32	34
32	9	33
34	33	2
9	32	36
32	8	35
36	35	0
10	37	39
37	11	38
39	38	1
11	37	41
37	10	40
41	40	3
0	35	13
35	8	25
13	25	4
0	16	36
16	6	29
36	29	9
1	14	39
14	4	24
39	24	10
1	38	15
38	11	30
15	30	6
2	19	34
19	5	26
34	26	8
2	33	20
33	9	28
20	28	7
3	40	18
40	10	23
18	23	5
3	21	41
21	7	31
41	31	11