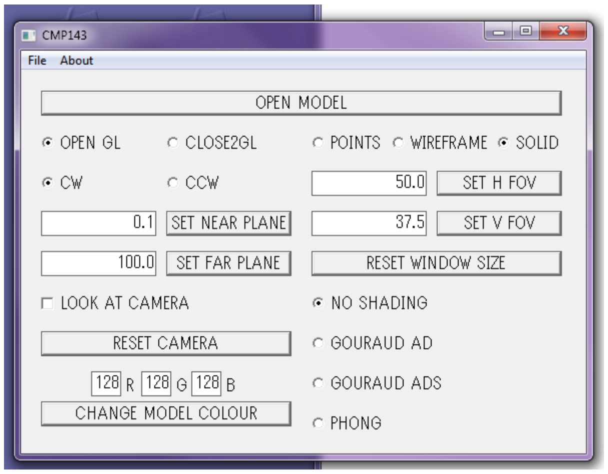
Universidade Federal do Rio Grande do Sul
INF01009 – Computação Gráfica
Aluno

Instituto de Informática
2023/1
Pedro Company Beck – 00324055

Assignment 2

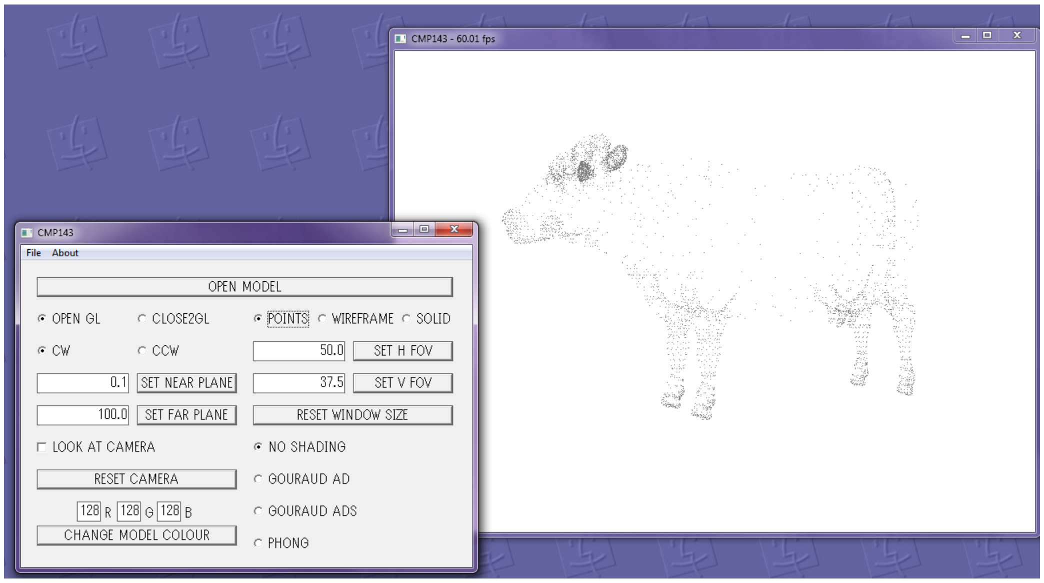
New controls were added to the interface, as shown below:



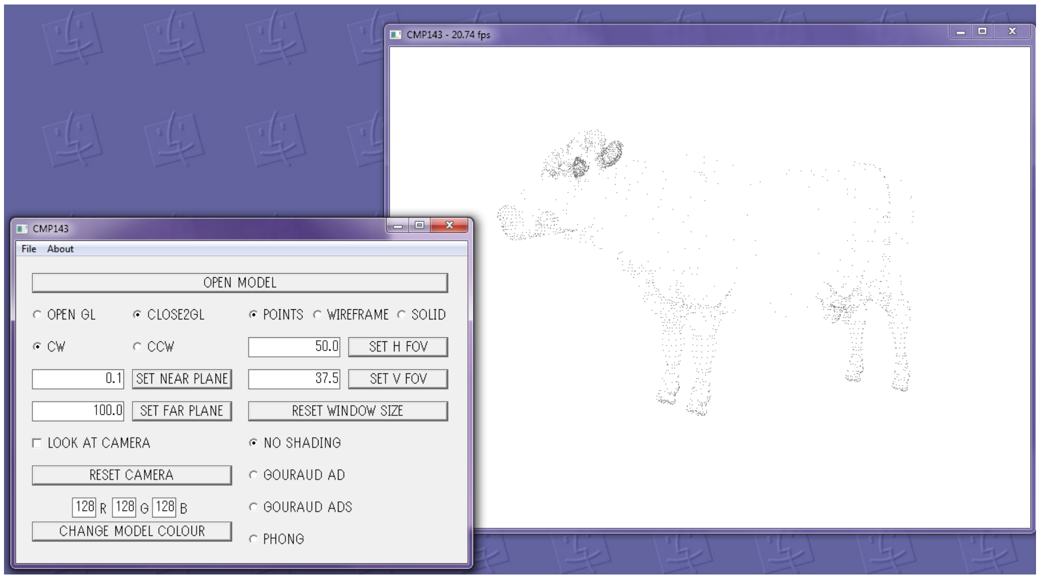
[✓ OPEN GL] CLOSE2GL
POINTS WIREFRAME [✓ SOLID]
[✓ CW] CCW
50.0 SET H FOV
0.1 SET NEAR PLANE
37.5 SET V FOV
100.0 SET FAR PLANE
☐ LOOK AT CAMERA
[✓ NO SHADING] GOURAUD AD GOURAUD ADS PHONG
128 R 128 G 128 B

1. Close2GL part

   a) Points
   The interface has a toggle for the user to choose to see the model as points, wireframe or as a solid. This functionality works both for OpenGL and Close2GL, as shown below:



[✓ OPEN GL] CLOSE2GL
[✓ POINTS] WIREFRAME SOLID
[✓ CW] CCW
50.0 SET H FOV
0.1 SET NEAR PLANE
37.5 SET V FOV
100.0 SET FAR PLANE
☐ LOOK AT CAMERA
[✓ NO SHADING] GOURAUD AD GOURAUD ADS PHONG
128 R 128 G 128 B



OPEN GL [✓ CLOSE2GL] [✓ POINTS] WIREFRAME SOLID

[✓ CW] CCW

0.1 SET NEAR PLANE · 50.0 SET H FOV · 100.0 SET FAR PLANE · 37.5 SET V FOV

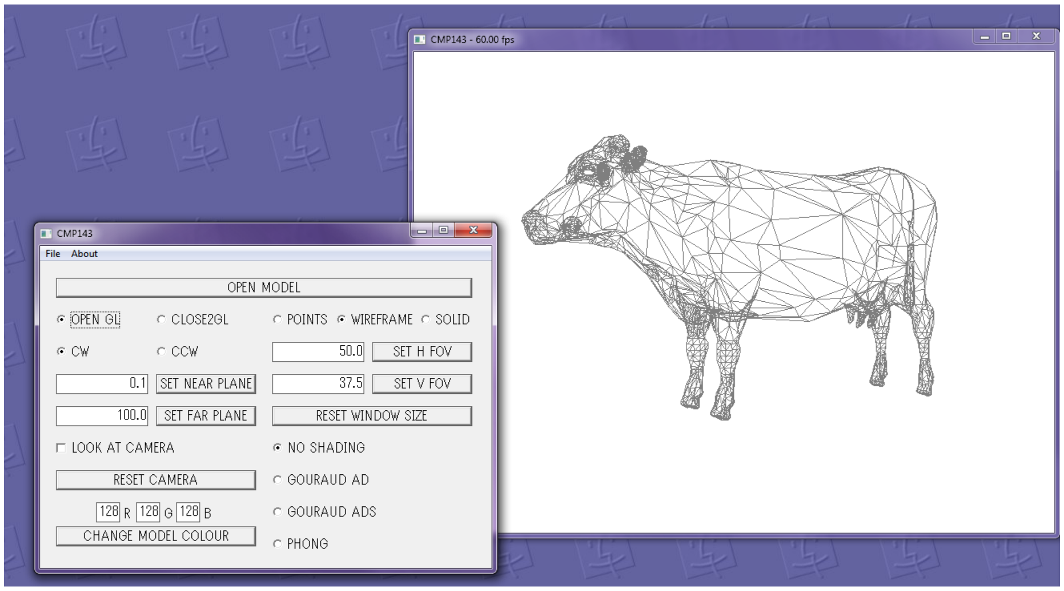
LOOK AT CAMERA

[✓ NO SHADING] GOURAUD AD GOURAUD ADS PHONG

128 R 128 G 128 B

As it can be seen comparing the two pictures above, the Close2GL shows fewer points. This is the case because of the backface culling done to discard the vertices in which the fragment normals face away from the camera.

b) Wireframe
When the toggle is set to "Wireframe", these are the results obtained:



[✓ OPEN GL] CLOSE2GL POINTS [✓ WIREFRAME] SOLID

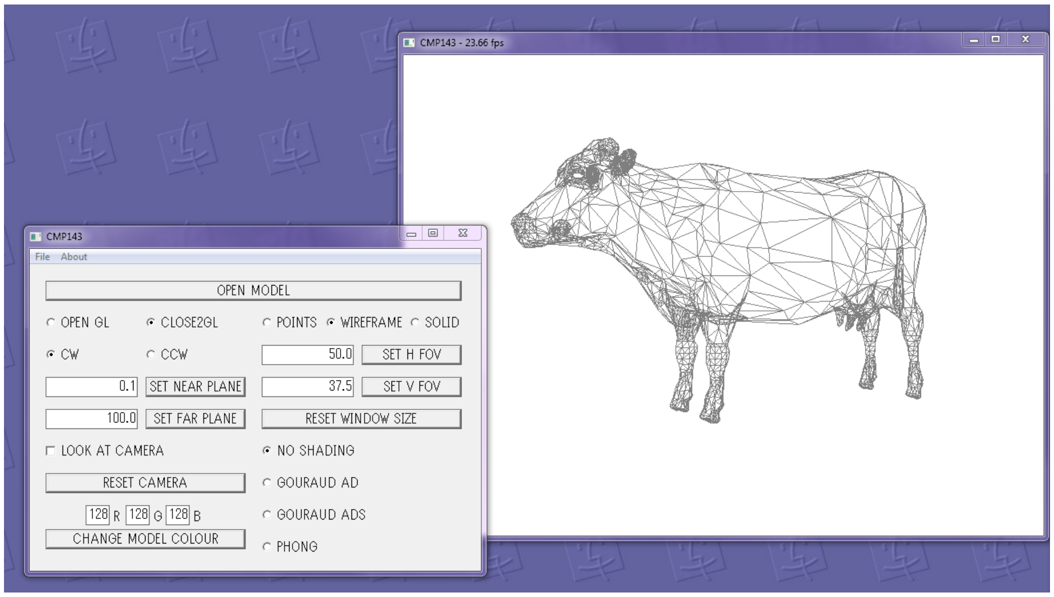
[✓ CW] CCW

0.1 SET NEAR PLANE · 50.0 SET H FOV · 100.0 SET FAR PLANE · 37.5 SET V FOV

LOOK AT CAMERA

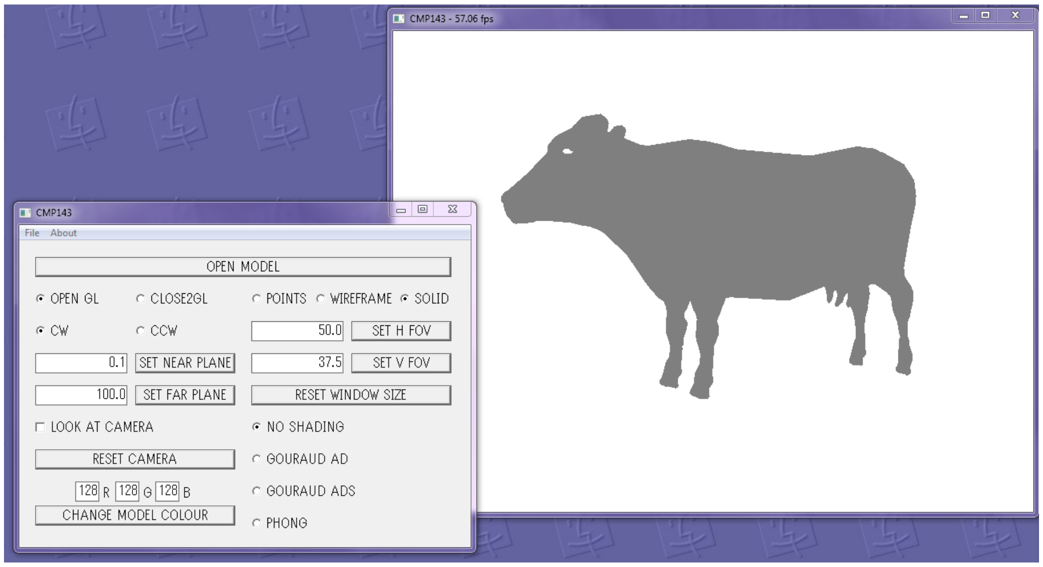
[✓ NO SHADING] GOURAUD AD GOURAUD ADS PHONG

128 R 128 G 128 B

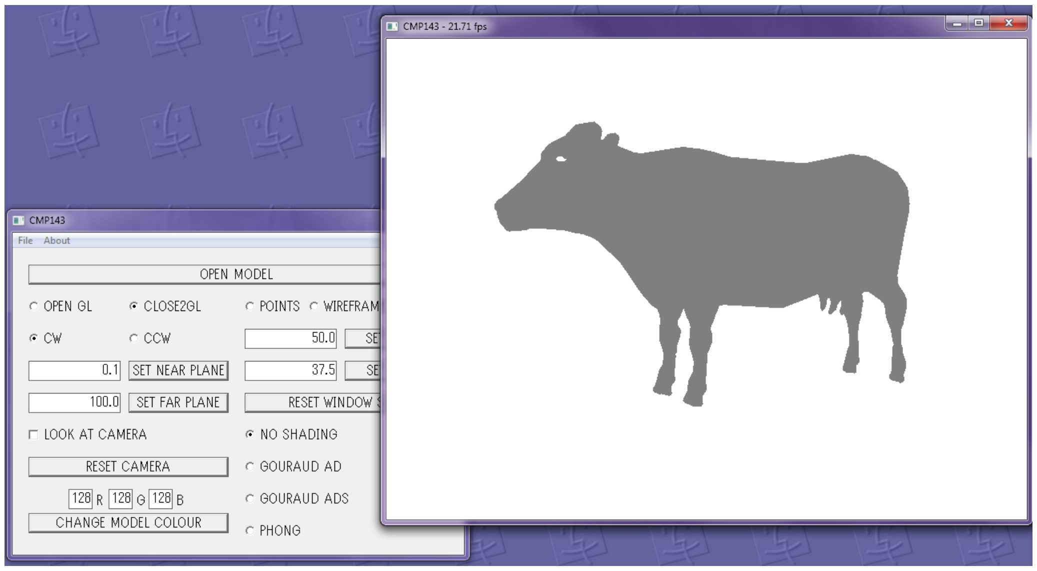


OPEN GL [✓ CLOSE2GL] POINTS [✓ WIREFRAME] SOLID
[✓ CW] CCW
50.0 SET H FOV
0.1 SET NEAR PLANE 37.5 SET V FOV
100.0 SET FAR PLANE
☐ LOOK AT CAMERA [✓ NO SHADING] GOURAUD AD GOURAUD ADS PHONG
128 R 128 G 128 B

c) Solid
When the toggle is set to "Solid", these are teh results obtained:



[✓ OPEN GL] CLOSE2GL POINTS WIREFRAME [✓ SOLID]
[✓ CW] CCW
50.0 SET H FOV
0.1 SET NEAR PLANE 37.5 SET V FOV
100.0 SET FAR PLANE
☐ LOOK AT CAMERA [✓ NO SHADING] GOURAUD AD GOURAUD ADS PHONG
128 R 128 G 128 B



OPEN GL [✓ CLOSE2GL] POINTS WIREFRAM

[✓ CW] CCW 50.0

0.1 SET NEAR PLANE 37.5

100.0 SET FAR PLANE

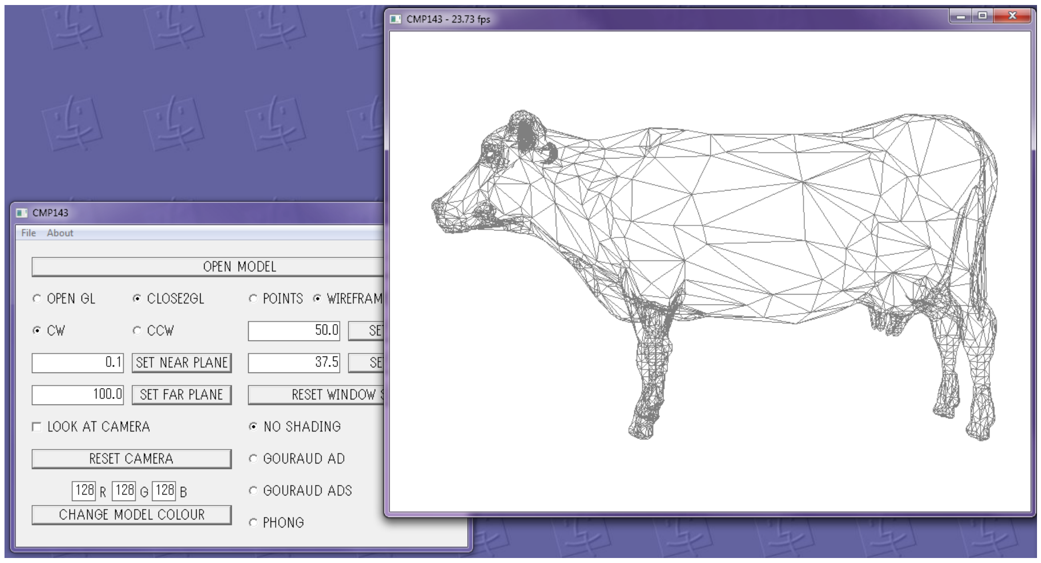
LOOK AT CAMERA [✓ NO SHADING]

GOURAUD AD

128 R 128 G 128 B GOURAUD ADS

PHONG

d) Translation
The Close2GL implementation supports translation as well, compare the following two images, of the start position (first image) and the image after a translation on the X-axis (second image).



OPEN GL [✓ CLOSE2GL] POINTS [✓ WIREFRAM]

[✓ CW] CCW 50.0

0.1 SET NEAR PLANE 37.5

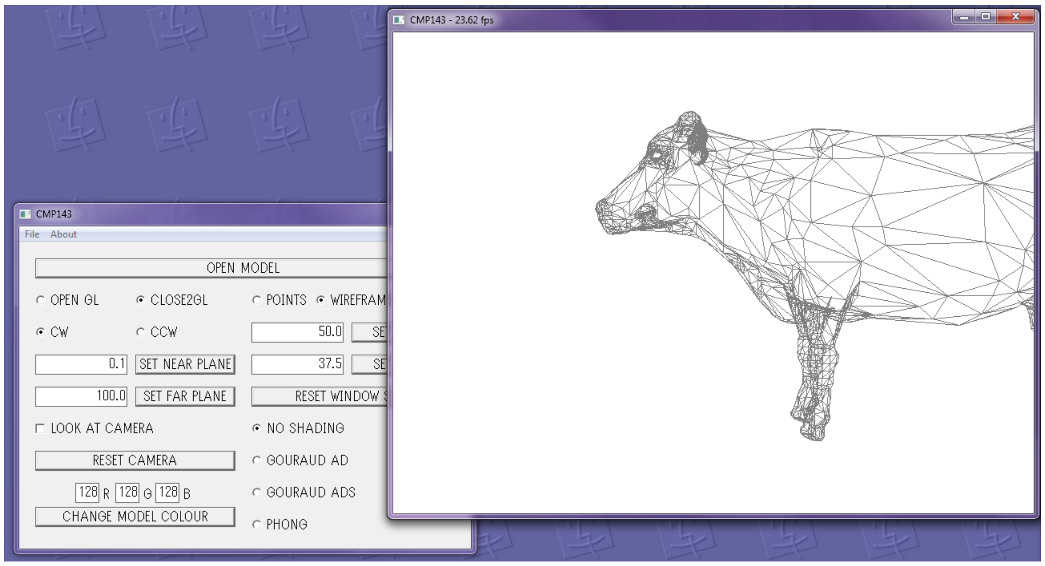
100.0 SET FAR PLANE

LOOK AT CAMERA [✓ NO SHADING]

GOURAUD AD

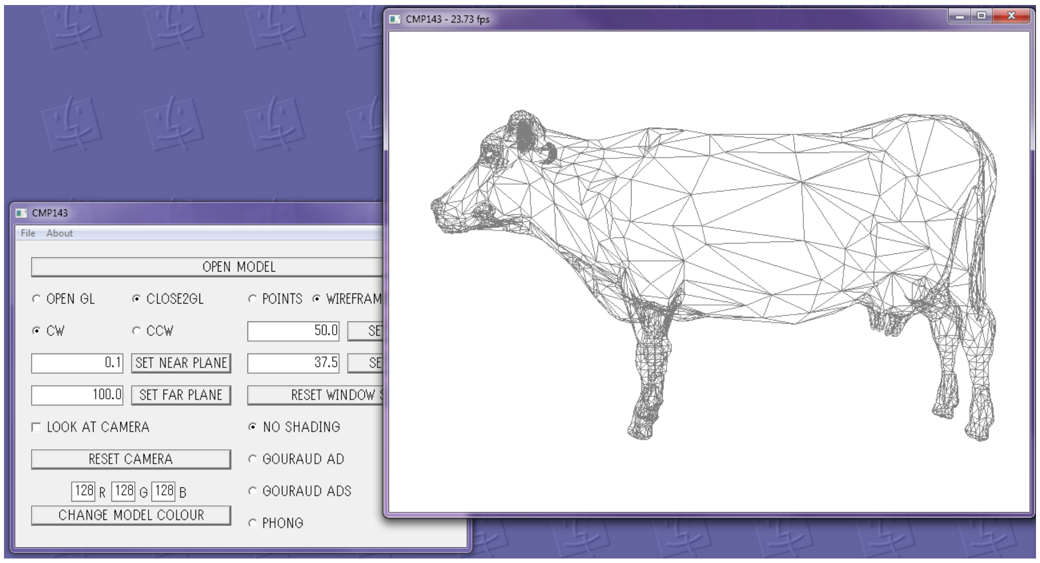
128 R 128 G 128 B GOURAUD ADS

PHONG

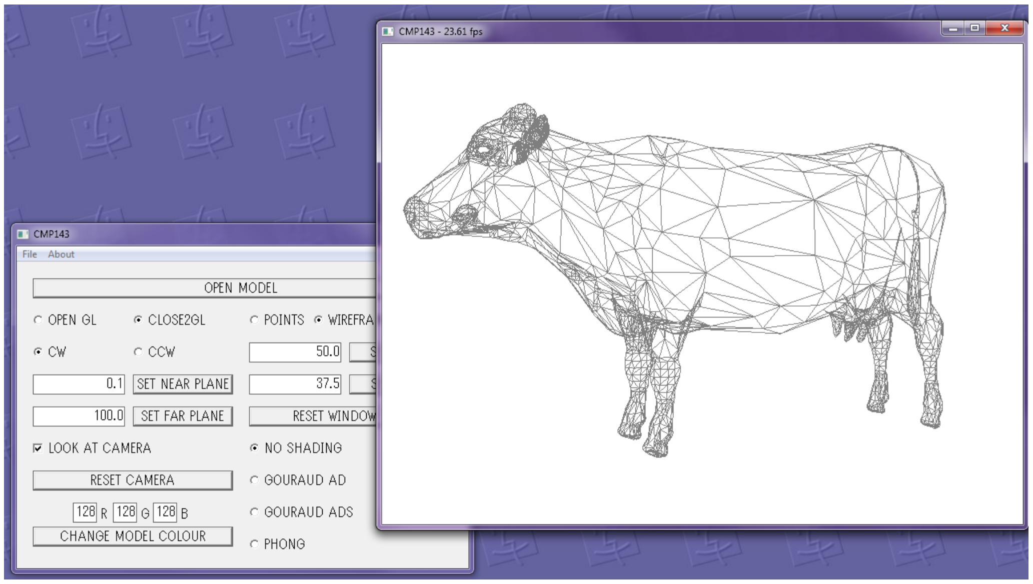


○ OPEN GL [✓ CLOSE2GL] ○ POINTS [✓ WIREFRAM]
[✓ CW] ○ CCW 50.0
0.1 37.5
100.0
☐ LOOK AT CAMERA [✓ NO SHADING]
○ GOURAUD AD
128 R 128 G 128 B ○ GOURAUD ADS
○ PHONG

e) Supports translation while keeping fixed look at point?
If "Look at Camera" is selected, the camera will translate while also keeping fixed at the point in the center of the object, as shown below:

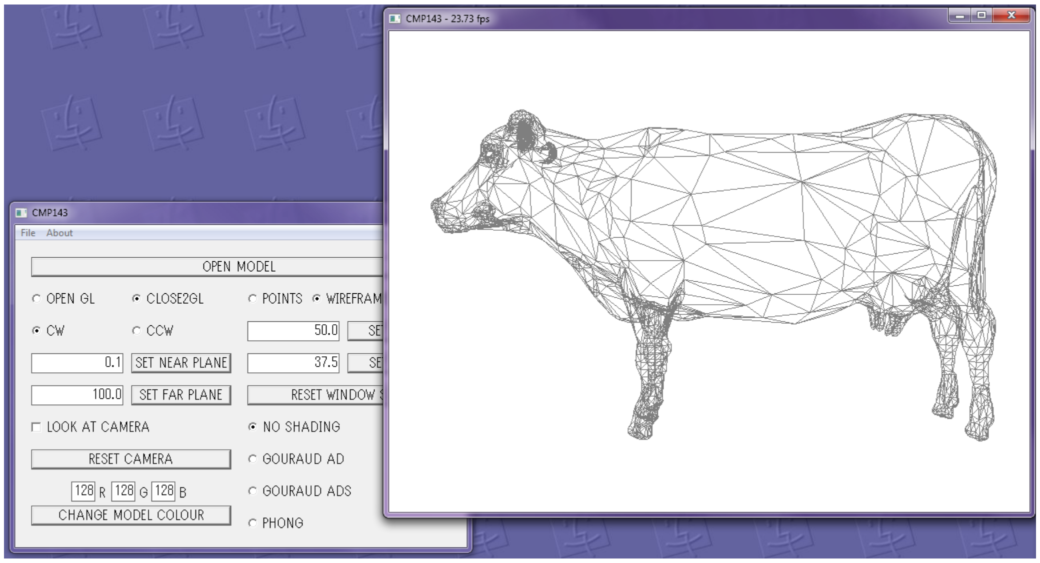


○ OPEN GL [✓ CLOSE2GL] ○ POINTS [✓ WIREFRAM]
[✓ CW] ○ CCW 50.0
0.1 37.5
100.0
☐ LOOK AT CAMERA [✓ NO SHADING]
○ GOURAUD AD
128 R 128 G 128 B ○ GOURAUD ADS
○ PHONG

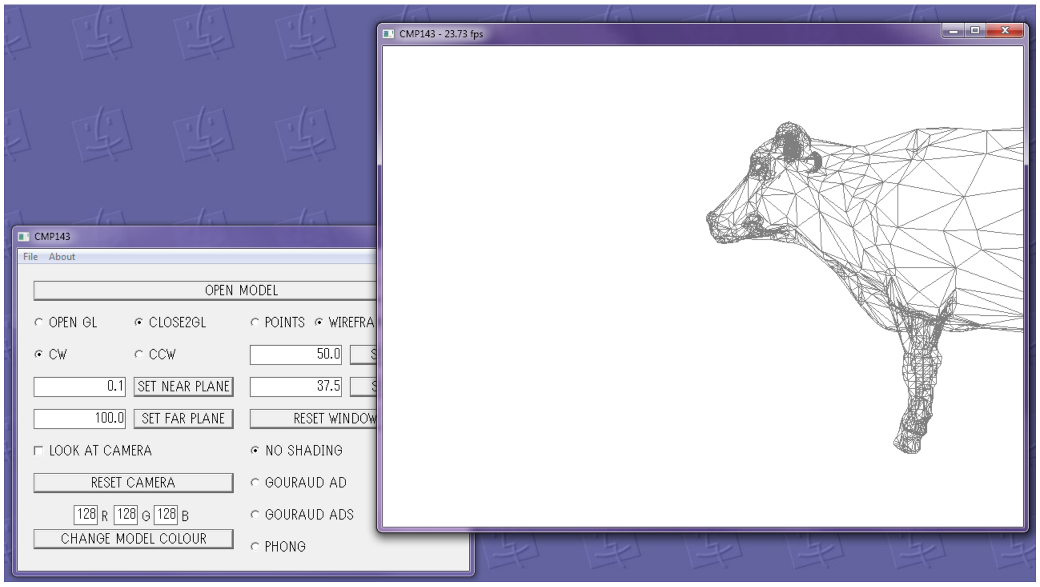


OPEN GL [✓ CLOSE2GL] POINTS [✓ WIREFRA]
[✓ CW] CCW
0.1 SET NEAR PLANE 50.0
100.0 SET FAR PLANE 37.5
[✓ LOOK AT CAMERA] [✓ NO SHADING]
GOURAUD AD
128 R 128 G 128 B GOURAUD ADS
PHONG

f) Rotation
Rotation is also supported, as shown below:



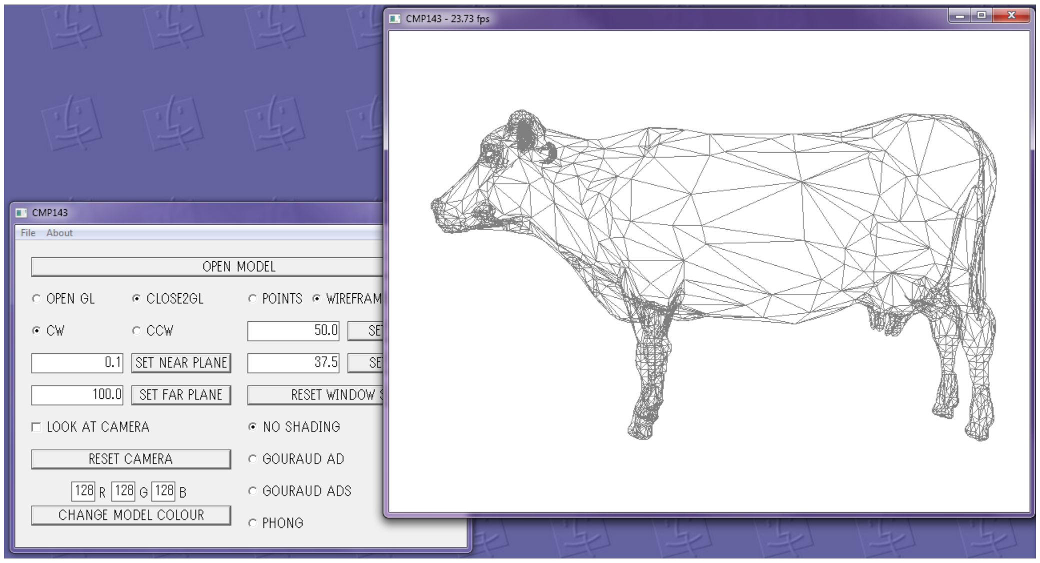
OPEN GL [✓ CLOSE2GL] POINTS [✓ WIREFRAM]
[✓ CW] CCW
0.1 SET NEAR PLANE 50.0
100.0 SET FAR PLANE 37.5
LOOK AT CAMERA [✓ NO SHADING]
GOURAUD AD
128 R 128 G 128 B GOURAUD ADS
PHONG

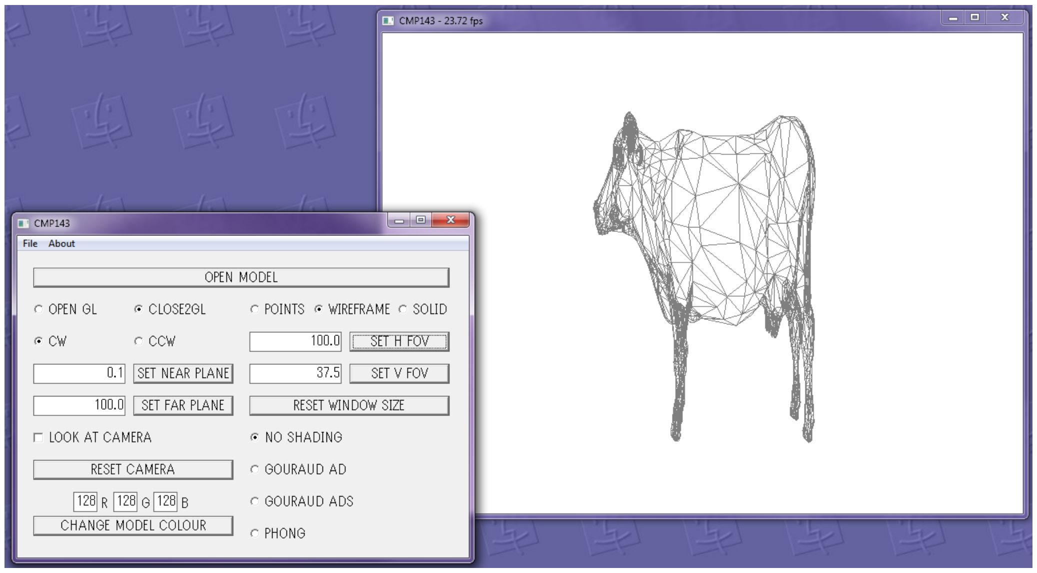


g) Backface culling
As seen on item 1.a), backface culling is performed to not send the vertices belonging to fragments that would be faced away from the camera to the GPU to render. The source code is included with this report and the implementation can be found on the `GLuint BuildTriangles(ModelObject model)` function.

h) Discarding vertices behind or on the same plane as the camera
This was also implemented on the `GLuint BuildTriangles(ModelObject model)` function from the `main.cpp` file.

i) Change Hfov
This is supported, and the results are shown below:




CMP143 - 23.72 fps

CMP143

File About

OPEN MODEL

OPEN GL [✓ CLOSE2GL] POINTS [✓ WIREFRAME] SOLID

[✓ CW] CCW 100.0 SET H FOV

0.1 SET NEAR PLANE 37.5 SET V FOV

100.0 SET FAR PLANE RESET WINDOW SIZE

LOOK AT CAMERA [✓ NO SHADING]

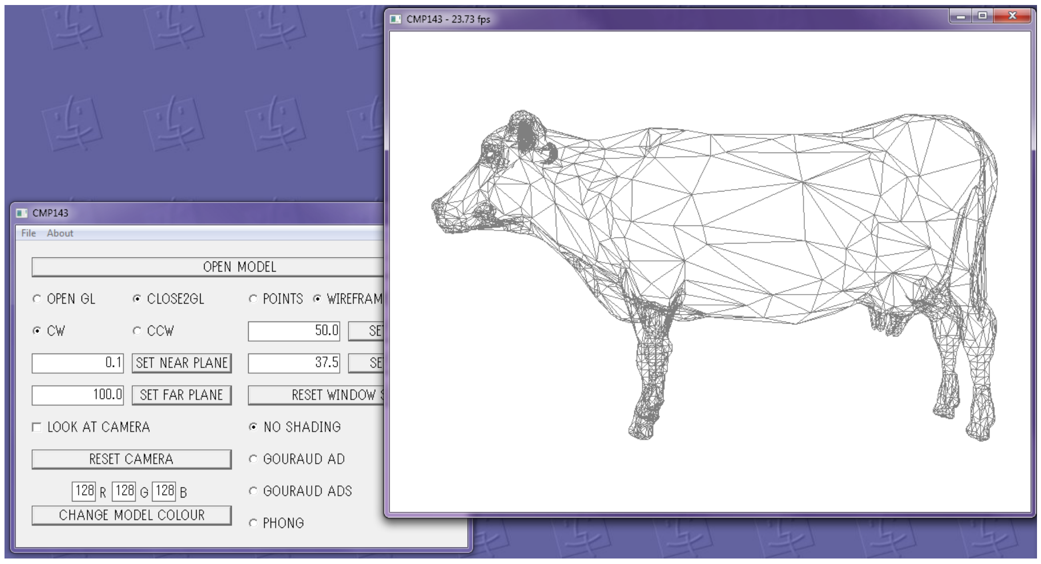
RESET CAMERA GOURAUD AD

128 R 128 G 128 B GOURAUD ADS

CHANGE MODEL COLOUR PHONG

In this case, the Horizontal FOV was changed from 50° (base value) to 100°. The image looks squished horizontally due to that.

j) Change Vfov
This is also supported, and the results are shown below:


CMP143 - 23.73 fps

CMP143

File About

OPEN MODEL

OPEN GL [✓ CLOSE2GL] POINTS [✓ WIREFRAM

[✓ CW] CCW 50.0 SE

0.1 SET NEAR PLANE 37.5 SE

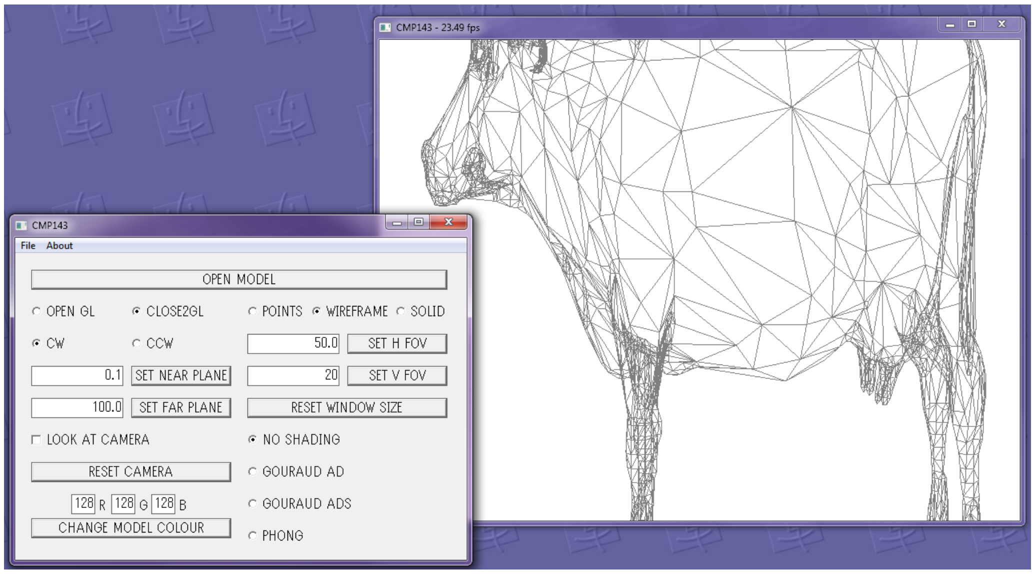
100.0 SET FAR PLANE RESET WINDOW S

LOOK AT CAMERA [✓ NO SHADING]

RESET CAMERA GOURAUD AD

128 R 128 G 128 B GOURAUD ADS

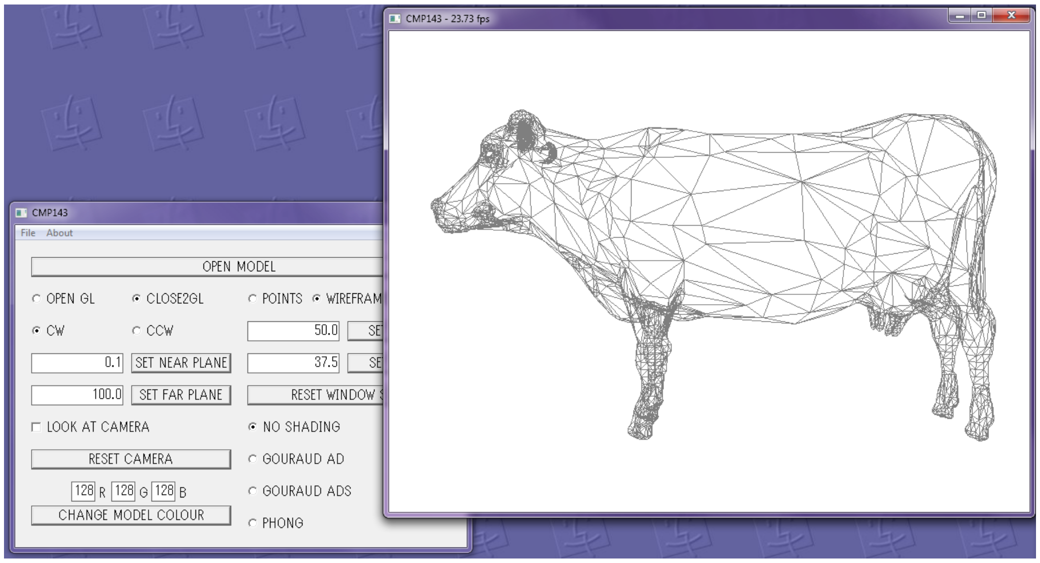
CHANGE MODEL COLOUR PHONG



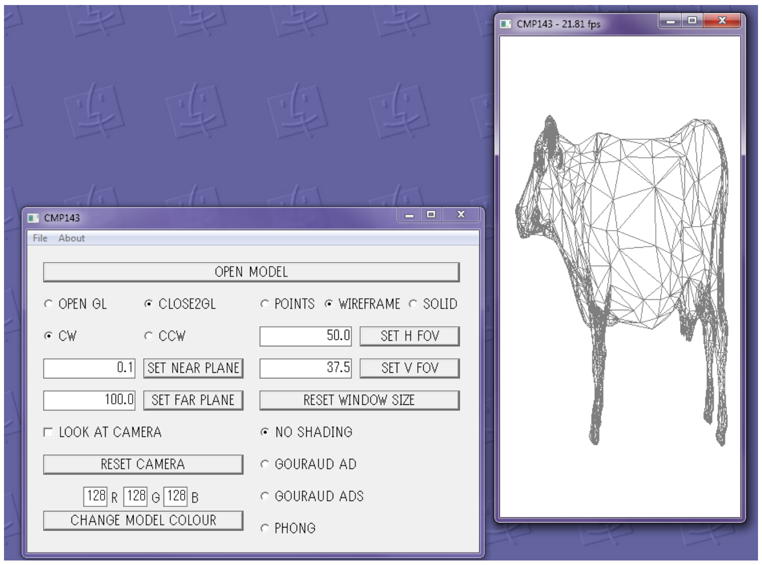
[OPEN GL ✓ CLOSE2GL] [POINTS ✓ WIREFRAME SOLID]
[✓ CW] CCW
50.0 SET H FOV
0.1 SET NEAR PLANE
20 SET V FOV
100.0 SET FAR PLANE
☐ LOOK AT CAMERA
[✓ NO SHADING] GOURAUD AD GOURAUD ADS PHONG
128 R 128 G 128 B

In this case, the Vertical FOV was changed from 37.5° (base value) to 20°. The image looks stretched vertically due to that.

k) Works after the window has been resized?
This is supported, and the results from changing the window size from the default 800x600 size to having fewer horizontal pixels is shown below:



[OPEN GL ✓ CLOSE2GL] [POINTS ✓ WIREFRAM]
[✓ CW] CCW
50.0
0.1 SET NEAR PLANE
37.5
100.0 SET FAR PLANE
☐ LOOK AT CAMERA
[✓ NO SHADING] GOURAUD AD GOURAUD ADS PHONG
128 R 128 G 128 B

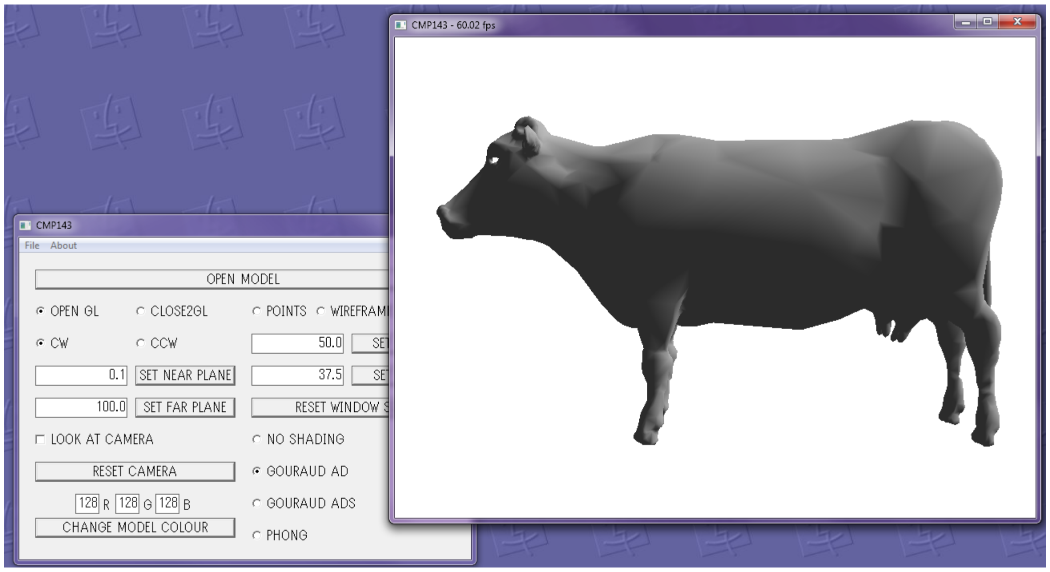


OPEN GL [✓ CLOSE2GL] POINTS [✓ WIREFRAME] SOLID
[✓ CW] CCW
50.0 | 0.1 | 37.5 | 100.0
☐ LOOK AT CAMERA
[✓ NO SHADING] GOURAUD AD GOURAUD ADS PHONG
128 R 128 G 128 B

2. Renderings from OpenGL and Close2GL are synchronized?
As shown on items 1.a), 1.b), and 1.c), the OpenGL/Close2GL toggle can be switched without affecting the position of the vertices shown. With the exception of the points mode, in which OpenGL seems to (incorrectly) show, in that mode, vertices that would be otherwise removed by backface culling, the visualizations are identical to each other.

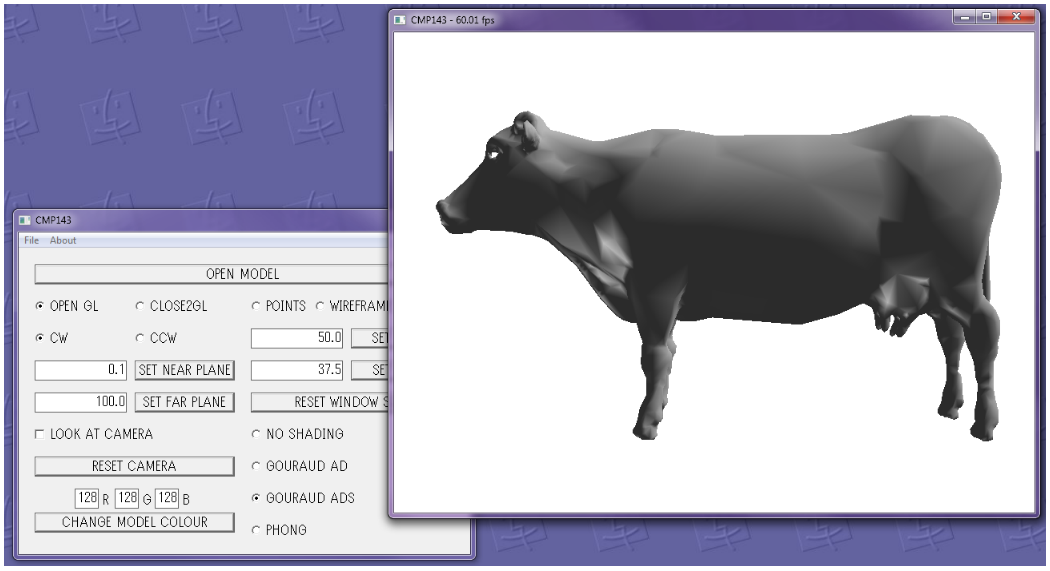
3. OpenGL part
In OpenGL mode, the shaders were implemented and can be selected with the shader toggle on the bottom right of the interface.

   a) Gouraud Shading AD
   This is the result obtained with Gouraud Shading AD:



[✓ OPEN GL] CLOSE2GL POINTS WIREFRAM
[✓ CW] CCW
50.0 | 0.1 | 37.5 | 100.0
☐ LOOK AT CAMERA
NO SHADING [✓ GOURAUD AD] GOURAUD ADS PHONG
128 R 128 G 128 B

b) Gouraud Shading ADS
   This is the result obtained with Gouraud Shading ADS:



CMP143 - 60.01 fps

[✓ OPEN GL] CLOSE2GL POINTS WIREFRAM

[✓ CW] CCW 50.0 SE

0.1 SET NEAR PLANE 37.5 SE

100.0 SET FAR PLANE

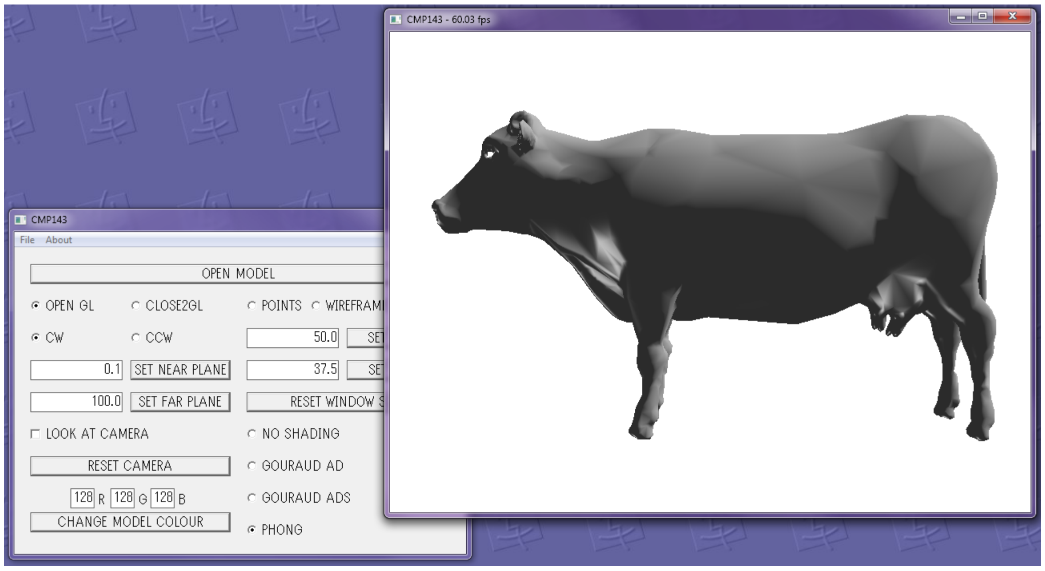
LOOK AT CAMERA NO SHADING

GOURAUD AD

128 R 128 G 128 B [✓ GOURAUD ADS]

PHONG

c) Phong Shading
   This is the result obtained with Phong Shading:



CMP143 - 60.03 fps

[✓ OPEN GL] CLOSE2GL POINTS WIREFRAM

[✓ CW] CCW 50.0 SE

0.1 SET NEAR PLANE 37.5 SE

100.0 SET FAR PLANE

LOOK AT CAMERA NO SHADING

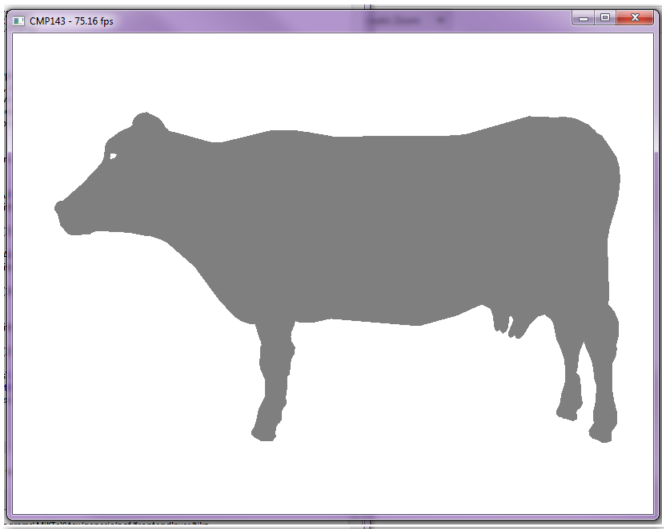
GOURAUD AD

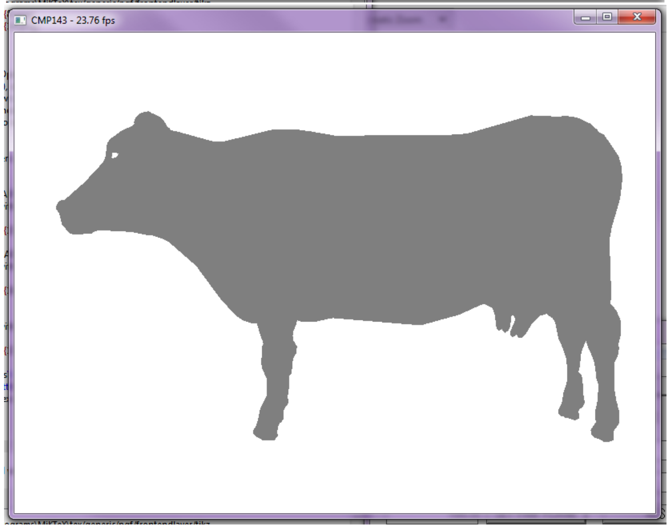
128 R 128 G 128 B GOURAUD ADS

[✓ PHONG]

d) Use of Subroutines
   As it can be seen in the `triangles.frag` and `triangles.vert` files, subroutines were implemented in both the fragment and vertex shaders to select the desired shader with the information sent from the interface.

4. FPS rate
   The FPS rate is shown on the title of the OpenGL window, on the top left corner, next to the name of the program.





The first image shows the frame rate with OpenGL, it is capped at around 75 fps, the frequency of the monitor currently being used. The second image shows the frame rate using Close2GL, which is at about 23 fps, as it can be seen some performance is lost while doing all the transformations in the CPU instead of doing it on the GPU.

The source code is included with this report.