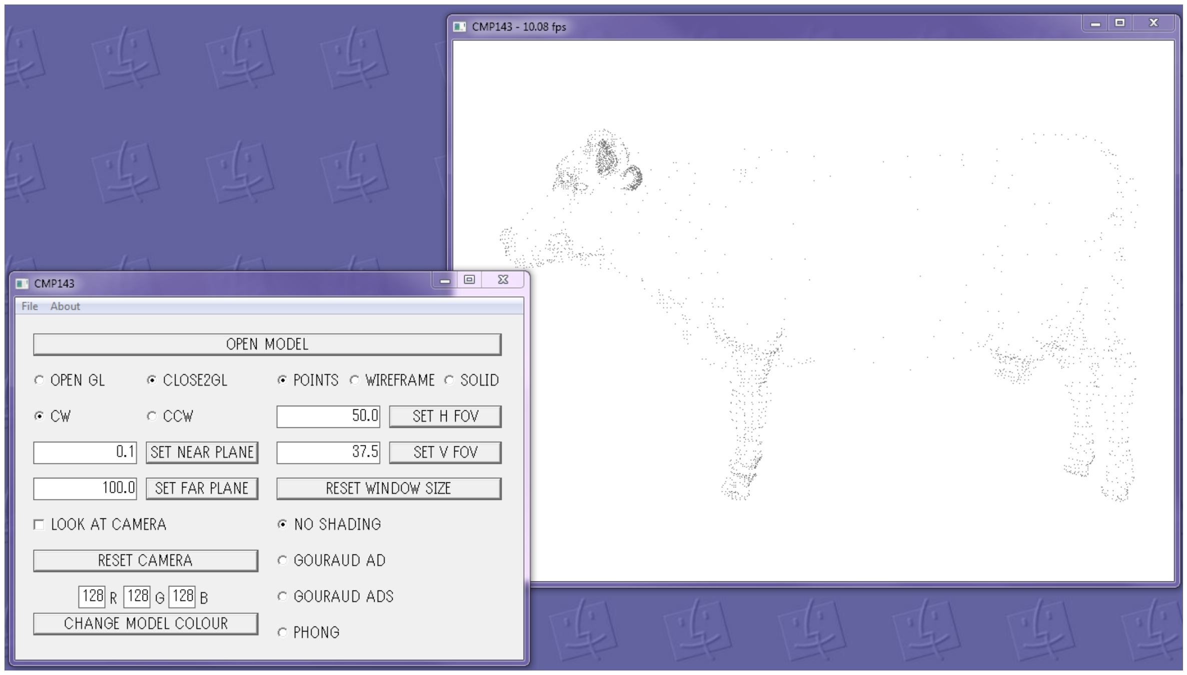
Universidade Federal do Rio Grande do Sul — Instituto de Informática
INF01009 – Computação Gráfica — 2023/1
Aluno — Pedro Company Beck – 00324055

Assignment 3

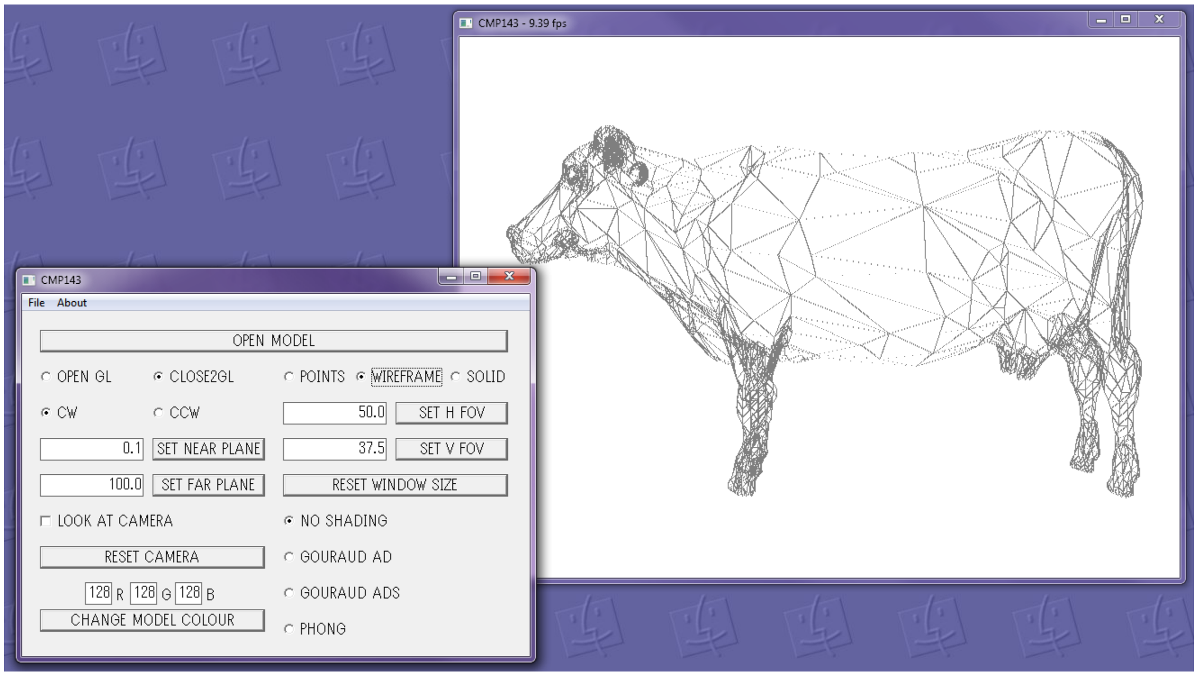
Rasterization was implemented in Close2GL, and it's being done by making color and z buffers, then mapping the color buffer as a texture to two triangles that will be shown on the screen. Each pixel of the color buffer has a z value that is compared for every fragment to check whether it will be visible or not. The implementation can be checked in the source code that will be provided with this report.

Close2GL now supports rasterization of a model as points:



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

As wireframe:



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

[✓ CW] CCW

0.1 SET NEAR PLANE

100.0 SET FAR PLANE

50.0 SET H FOV

37.5 SET V FOV

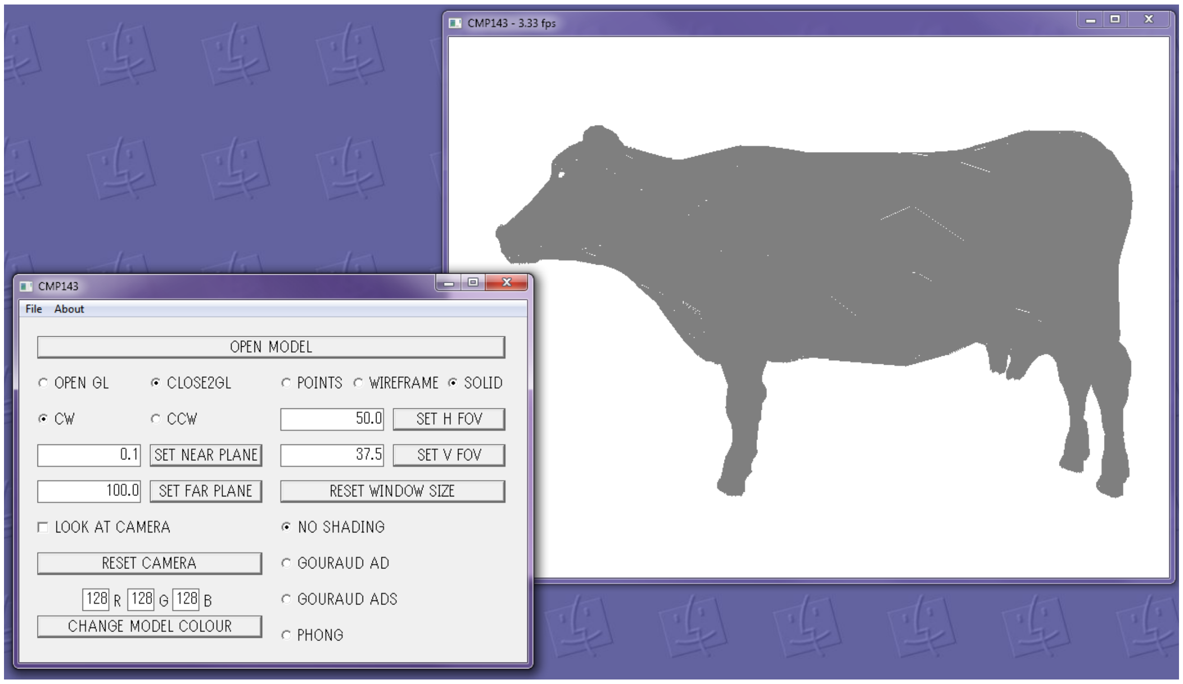
☐ LOOK AT CAMERA

[✓ NO SHADING] GOURAUD AD GOURAUD ADS PHONG

128 R 128 G 128 B

unfortunately, some of the edges are not entirely connected as lines, this is due to the rasterization function only printing the first and last pixels of the row that’s being drawn at any moment.

And as a solid:



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]

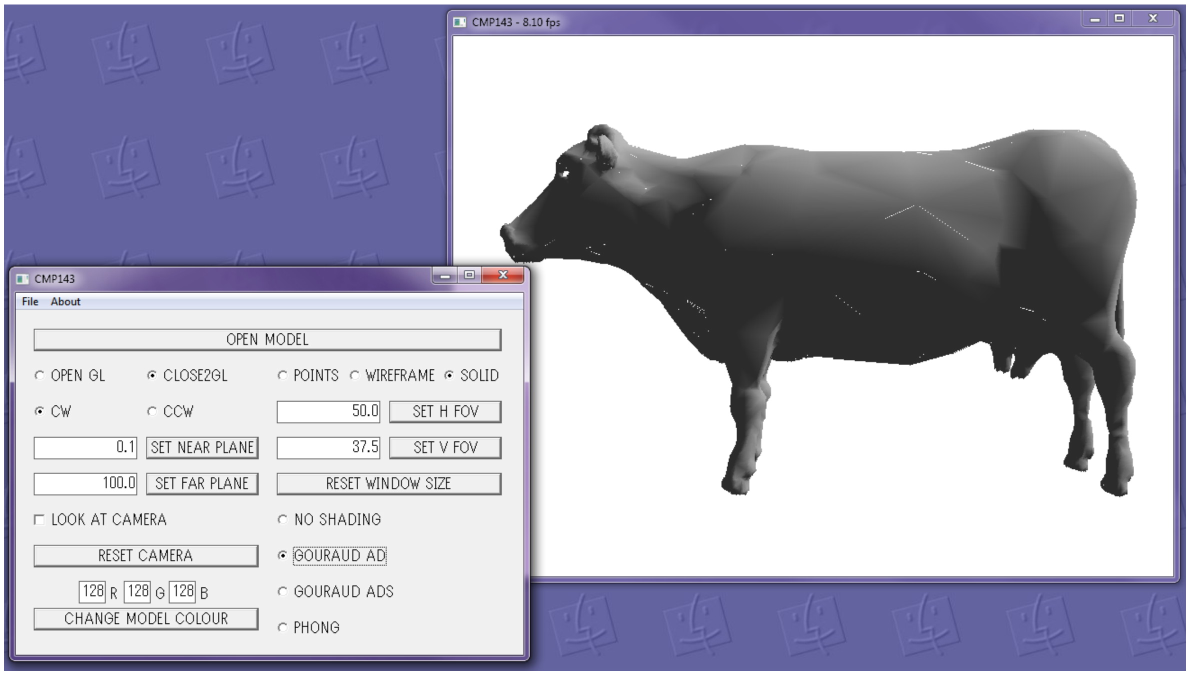
GOURAUD AD

128 R 128 G 128 B GOURAUD ADS

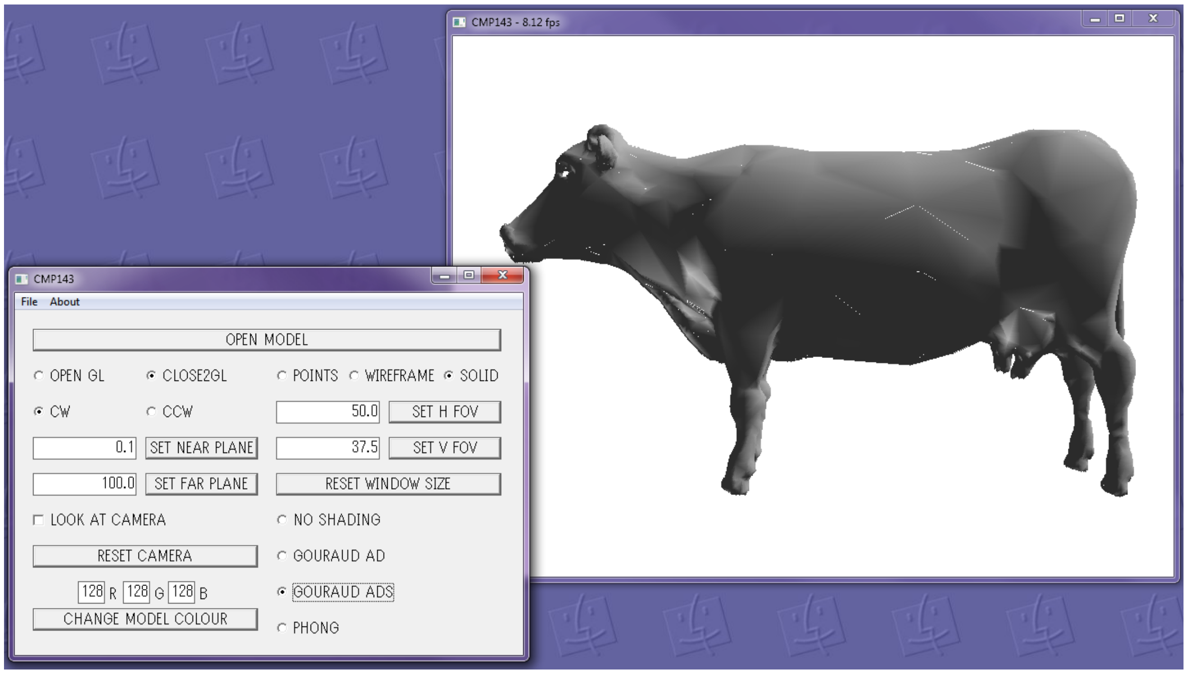
PHONG

unfortunately, there are some visible artifacts. This may have been due to the usage of the `floor()` function to approximate the pixel values to integers and some tests were made, however, the cause is still unknown.

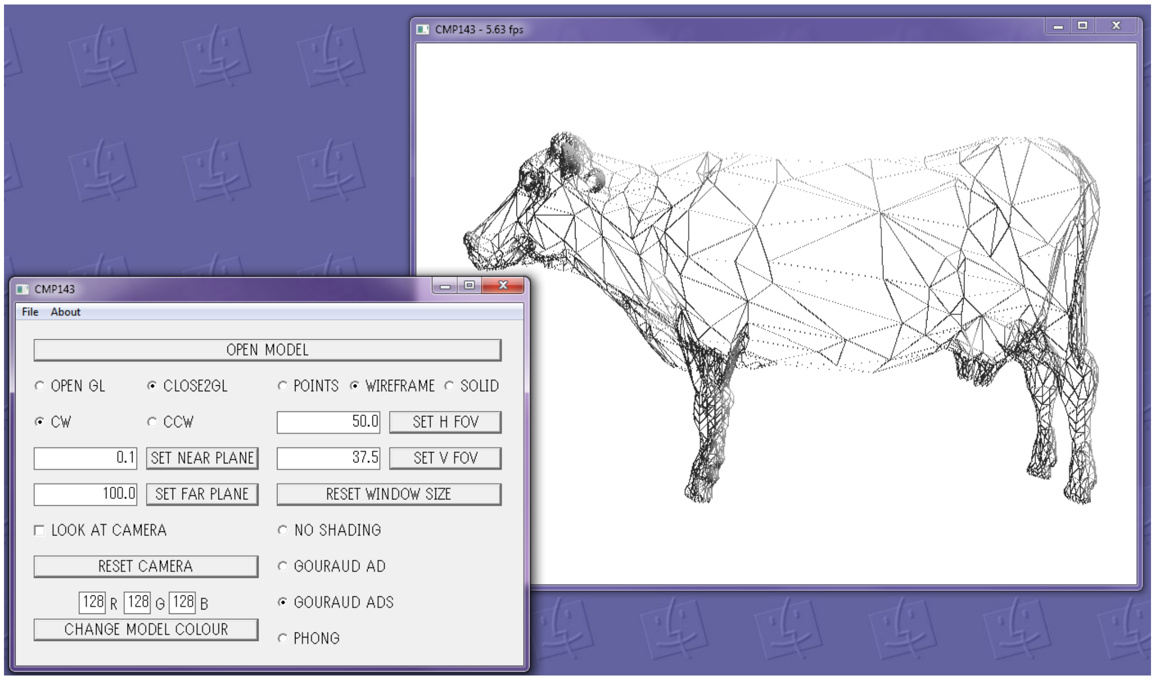
Close2GL also supports rasterization with Gouraud AD and Gouraud ADS (or Phong) lightning models:



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
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

100.0 SET FAR PLANE

50.0 SET H FOV

37.5 SET V FOV

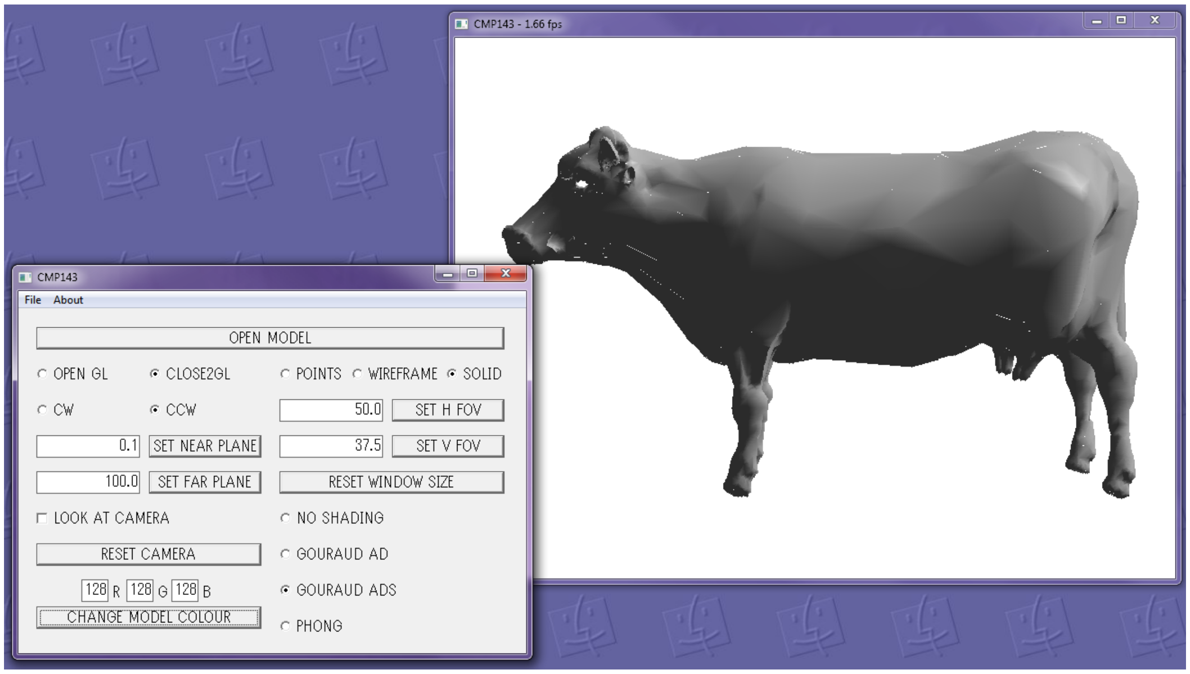
☐ LOOK AT CAMERA

NO SHADING / GOURAUD AD / [✓ GOURAUD ADS] / PHONG

128 R 128 G 128 B

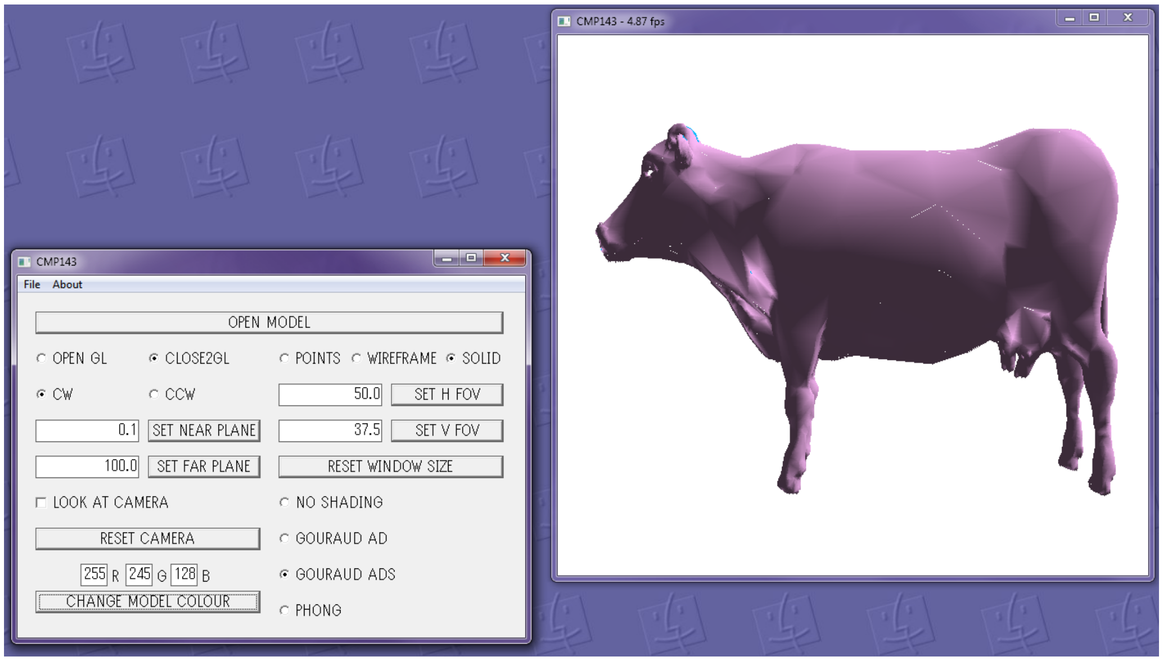
These were implemented using Gouraud shading.

It works with both clockwise and counterclockwise orientation of the vertices:



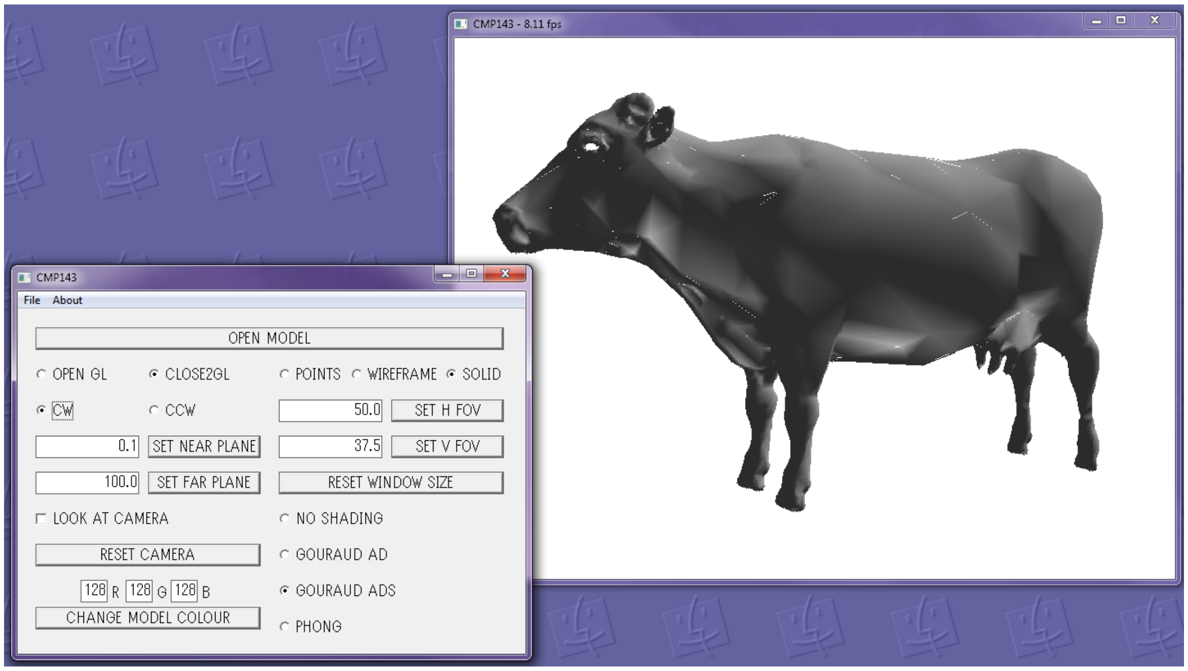
OPEN GL [✓ CLOSE2GL] POINTS WIREFRAME [✓ SOLID]
CW [✓ CCW] 50.0 SET H FOV
0.1 SET NEAR PLANE 37.5 SET V FOV
100.0 SET FAR PLANE
LOOK AT CAMERA NO SHADING
GOURAUD AD
128 R 128 G 128 B [✓ GOURAUD ADS]
PHONG

And it works if the screen is resized or the color of the model is changed:



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
255 R 245 G 128 B [✓ GOURAUD ADS]
PHONG

It also works fine if the camera orientation changes:


CMP143 - 8.11 fps

CMP143

File About

OPEN MODEL

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

RESET WINDOW SIZE

☐ LOOK AT CAMERA

NO SHADING

RESET CAMERA

GOURAUD AD

128 R 128 G 128 B

[✓ GOURAUD ADS]

CHANGE MODEL COLOUR

PHONG