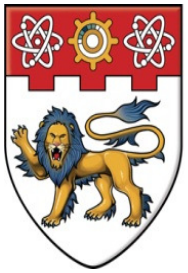


NANYANG TECHNOLOGICAL UNIVERSITY

CZ4004

3D MODELLING AND ANIMATION

PROGRAMMING ASSIGNMENT

3D MESH VIEWER

Prepared By:	
MUHAMMAD SALIHAN BIN ZAOL-KEFLI	U1221712J

Table of Contents

1 Compilation Instructions

1.1 IDE Used

The IDE that I used for the assignment is Microsoft Visual Studio Community 2015. It can be downloaded at https://www.visualstudio.com/ and it is free.

1.2 Third Party Libraries

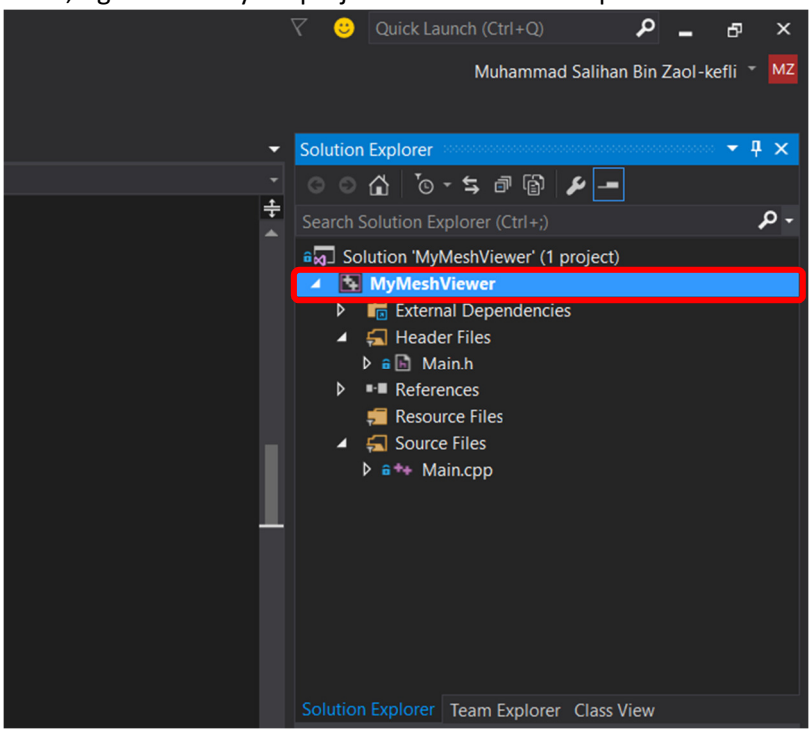
The third party libraries I used for this assignment are GLUT, GLU and OpenGL. Within the zip file, you can find the glu32.dll, glut32.dll and the opengl32.dll files. There are also the glut.h and glut32.lib files that can be found in the include and lib folders respectively.

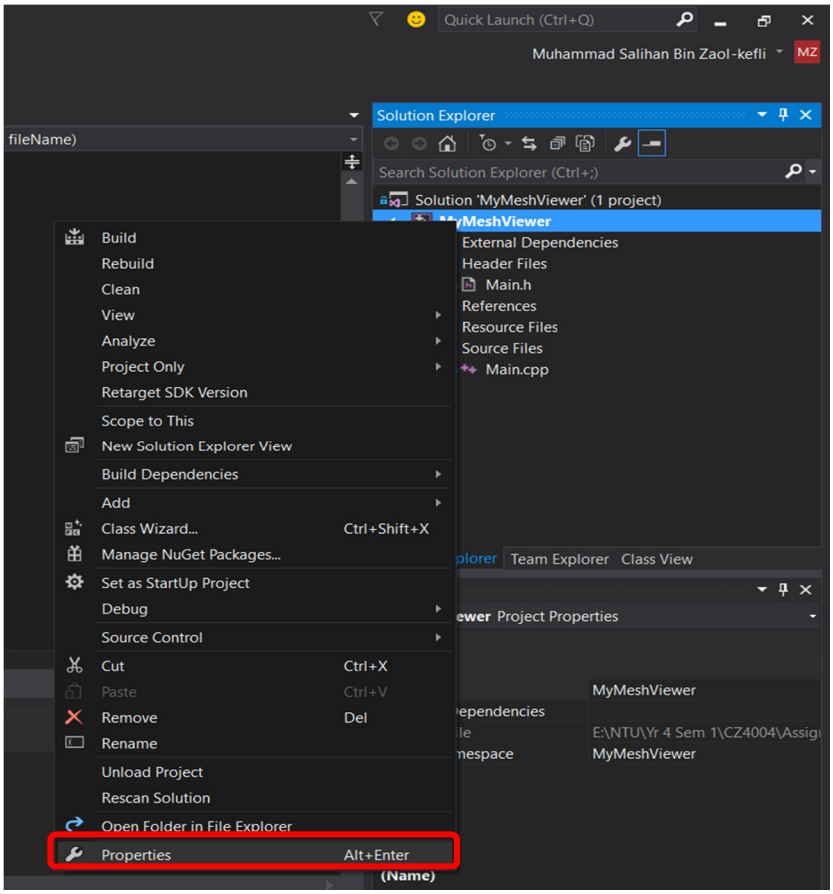
1.3 How to Compile

First, create a new Visual C++ Win32 Console Application project. Give it a name (eg. MyMeshViewer) and choose a directory. Click OK and choose Application settings. Check the Empty project in the Additional options section and then click Finish.

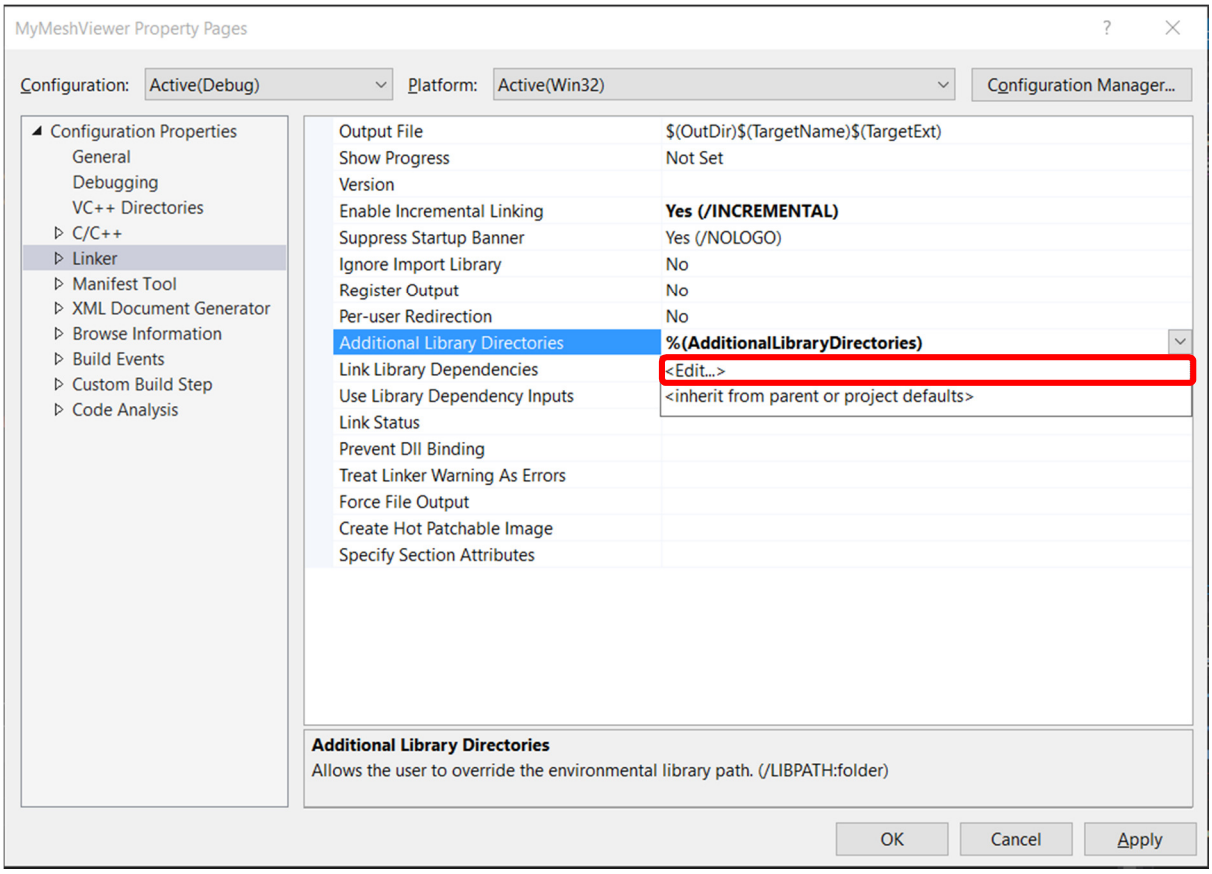
Second, copy all the .dll files, the Main.cpp file, the include folder, the lib folder as well as the TestModels folder into your project directory. Usually the path of this directory will look something like ..\MyMeshViewer\MyMeshViewer\ depending on the name of the project.

Third, right click on your project in the Solution Explorer and click on properties as shown below.





Fourth, a new window should pop up. Navigate to Linker on the left hand side and you should see the below. Click on the Additional Library Directories option and click Edit.



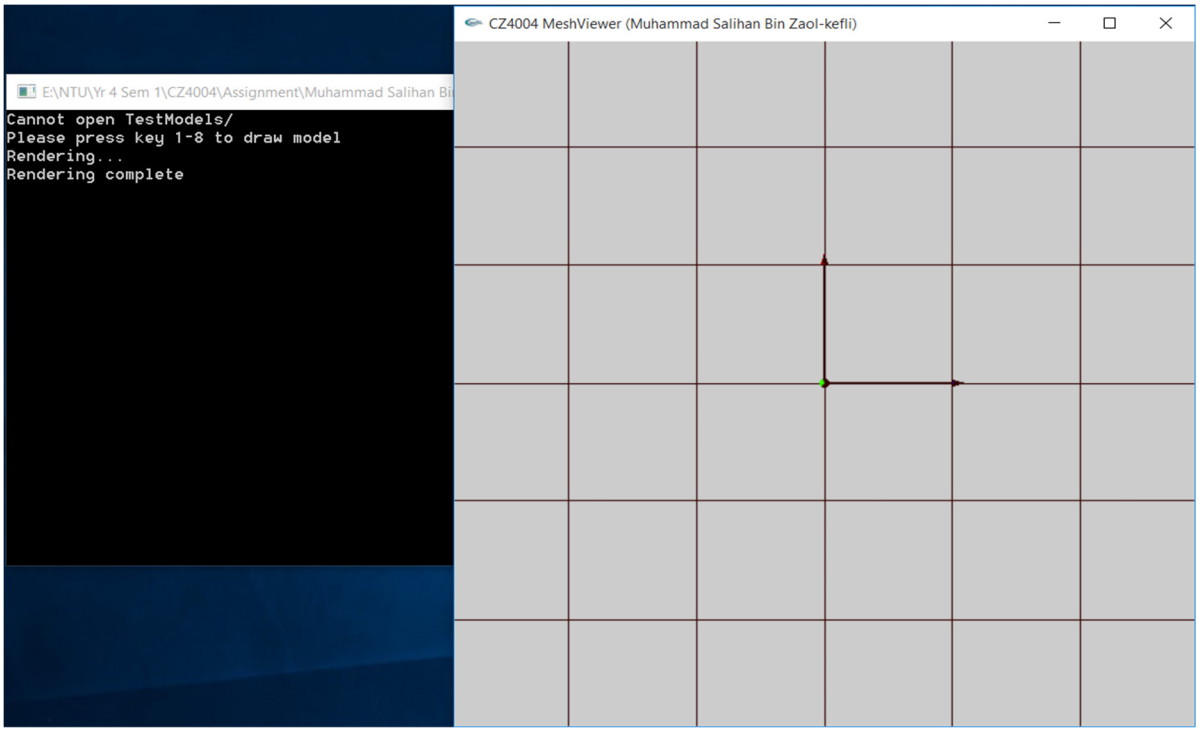
Another small window will pop up. Click the icon that looks like a folder and next to the red cross icon. Navigate to the lib folder from the explorer and click OK then click Apply. The Additional Library Directories field should now reflect the path to the lib folder.

Fifth, navigate to the C/C++ option on the left hand side. The first field on the right hand side is the Additional Include Directories. Do the same thing as you did in the fourth step but this time navigate to the include folder. Click OK and then click Apply. The Additional Include Directories field should now reflect the path to the include folder.

Lastly, add the Main.cpp file by right clicking the Source Files folder in the solution explorer and choosing the Add->Existing Item option. Right click on the Header Files folder in the solution explorer and choose Add->Existing Item to add the Main.h file which can be found in the lib folder if you want to see the declarations and definitions. Otherwise, you can just add the Main.cpp. Click the button with the play symbol at the top of the IDE to compile and build the program.

2 A Simple Manual to Use the Program

The first thing you should see when the program successfully compiles is the below.



Instructions

- Press keys 1-8 to load a model.
  - 1: bimba
  - 2: bottle
  - 3: bunny
  - 4: cap
  - 5: eight
  - 6: gargoyle
  - 7: knot
  - 8: statute
- Press key ‘p’ to render the model in point cloud mode
- Press key ‘w’ to render the model in wireframe mode
- Press key ‘f’ to render the model in flat shading mode
- Press key ‘s’ to render the model in smooth shading mode
- Press key ‘r’ to reset any transformation
- Press Spacebar key to switch projection from perspective to orthogonal and vice versa
- Press Up Arrow key to increase brightness/intensity of light
- Press Down Arrow key to decrease brightness/intensity of light
- Hold down Left mouse button for rotation
  - Move mouse vertically to rotate about x-axis
  - Move mouse horizontally to rotate about y-axis
  - Move mouse in a circular motion to rotate about z-axis
- Hold down Middle mouse button for translation
  - Move mouse vertically to translate along y-axis

    - Move mouse horizontally to translate along x-axis
- Hold down Right mouse button for scaling
    - Move mouse horizontally
        - Left to scale/zoom out
        - Right to scale/zoom in

3 Results

3.1 Point Cloud

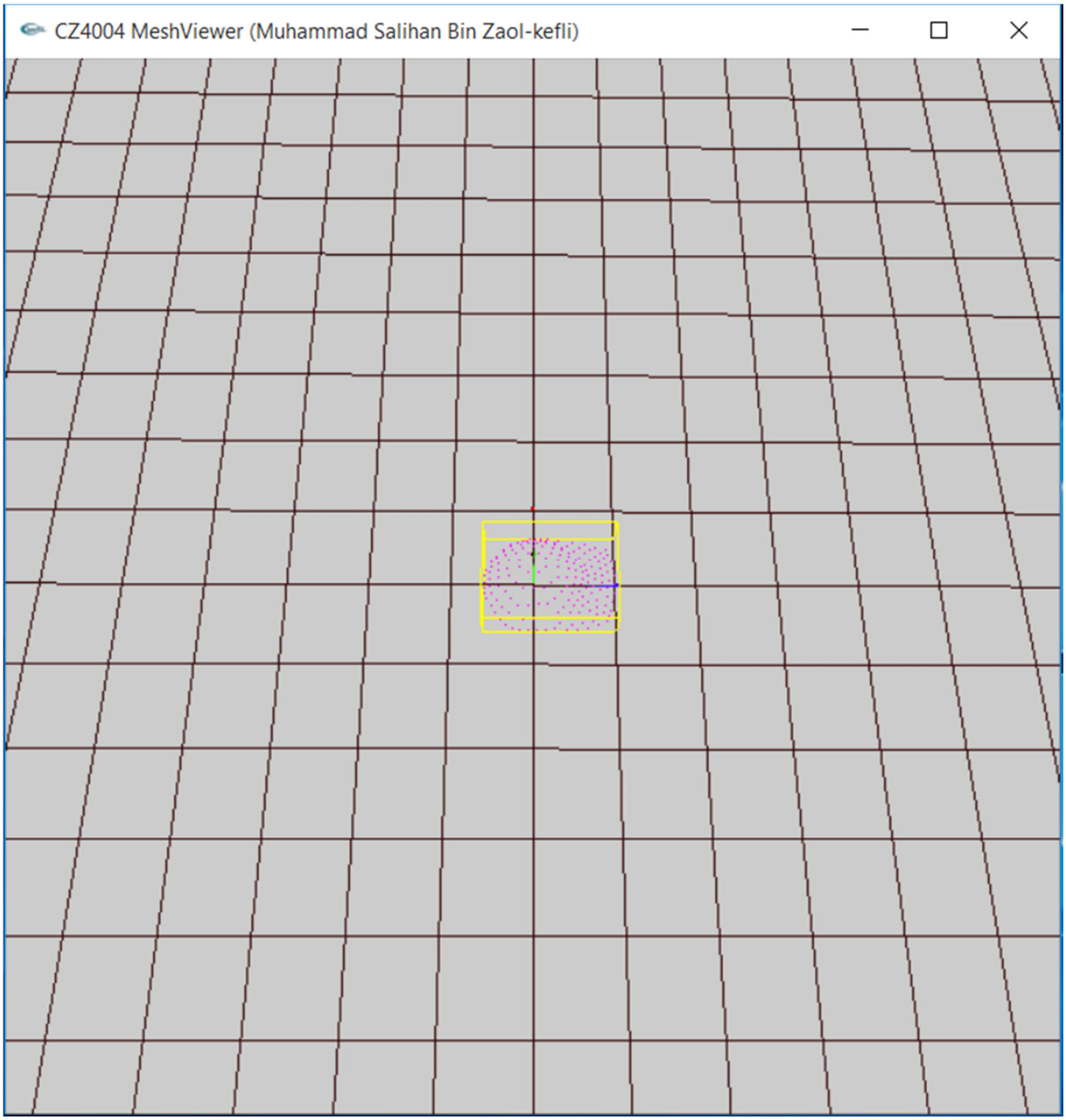


Figure 1 Point cloud rendering of cap model

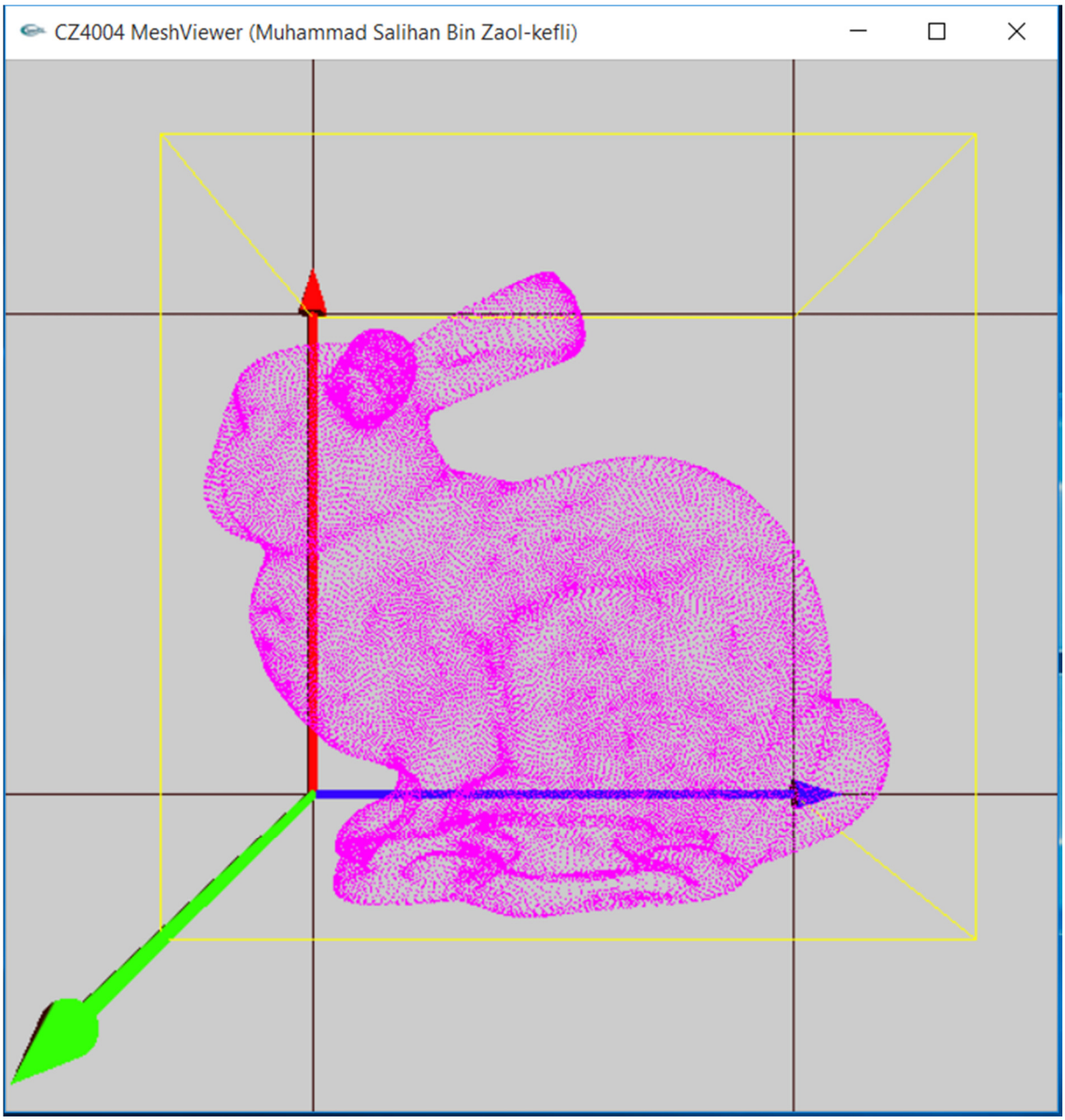


Figure 2 Point cloud rendering of bunny model

3.2 Wireframe

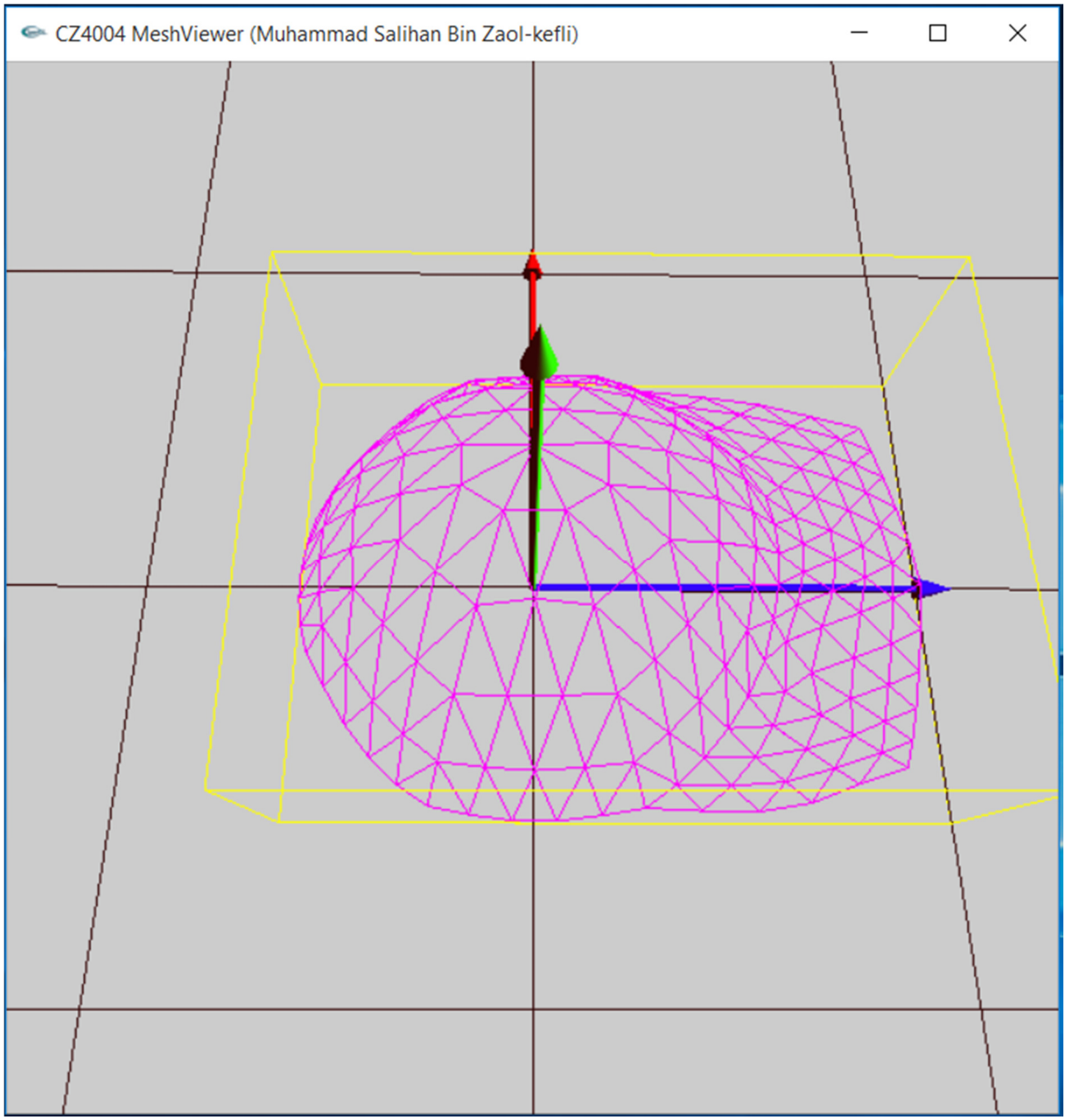


Figure 3 Wireframe rendering of cap model

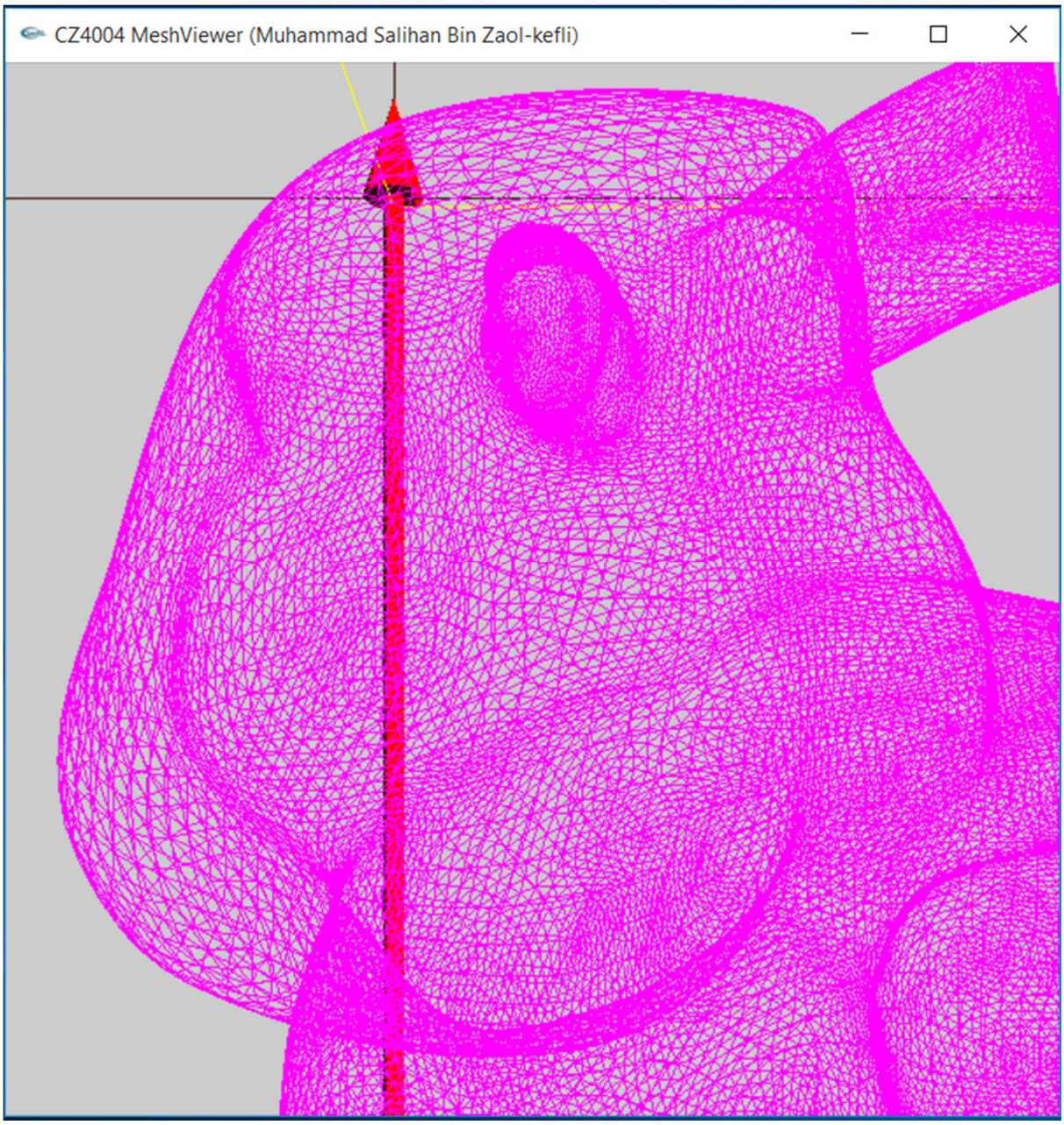


Figure 4 Wireframe rendering of bunny model

3.3 Flat Shading

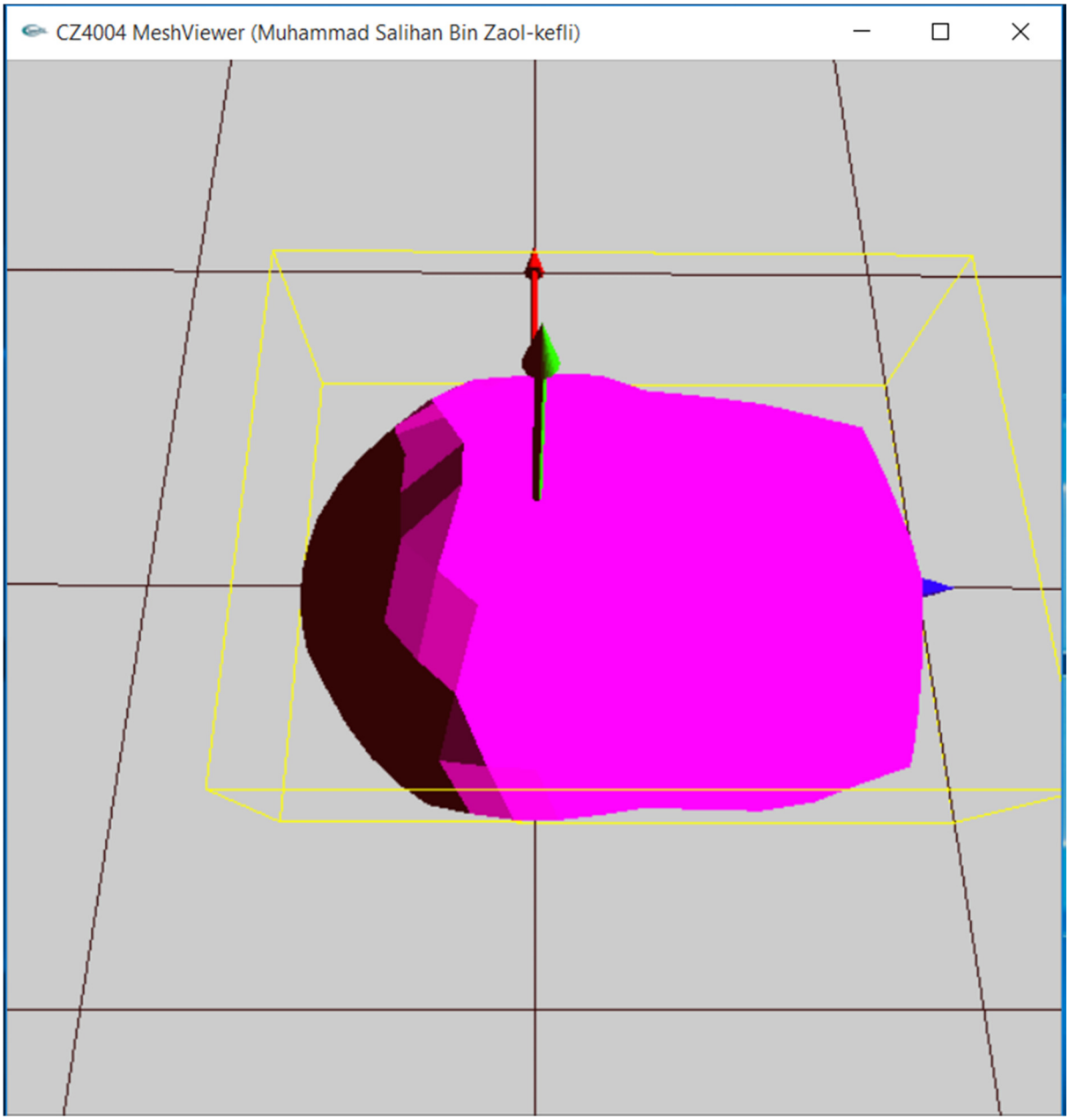


Figure 5 Flat shading rendering of Cap model

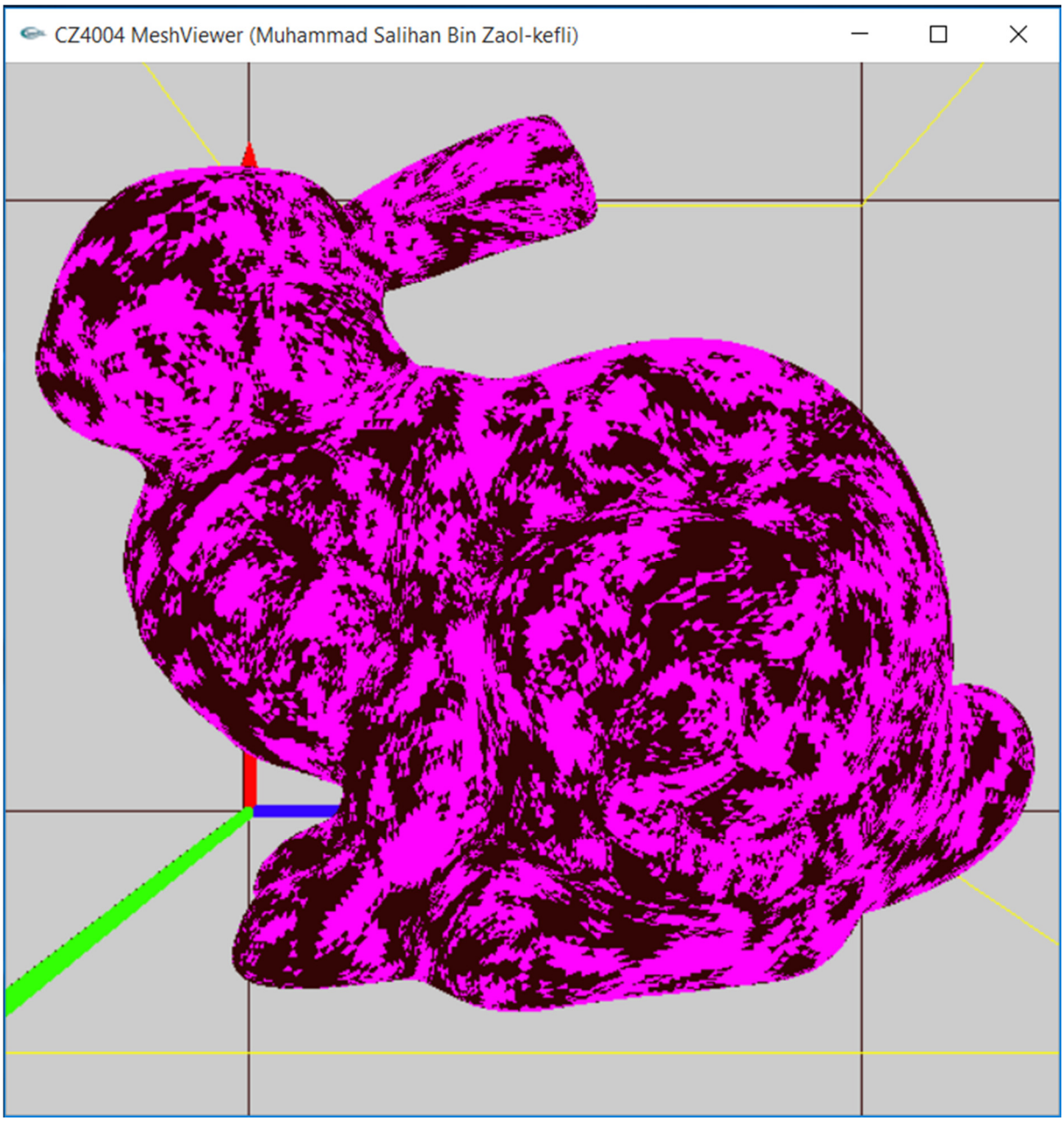


Figure 6 Flat shading rendering of bunny model

3.4 Smooth Shading

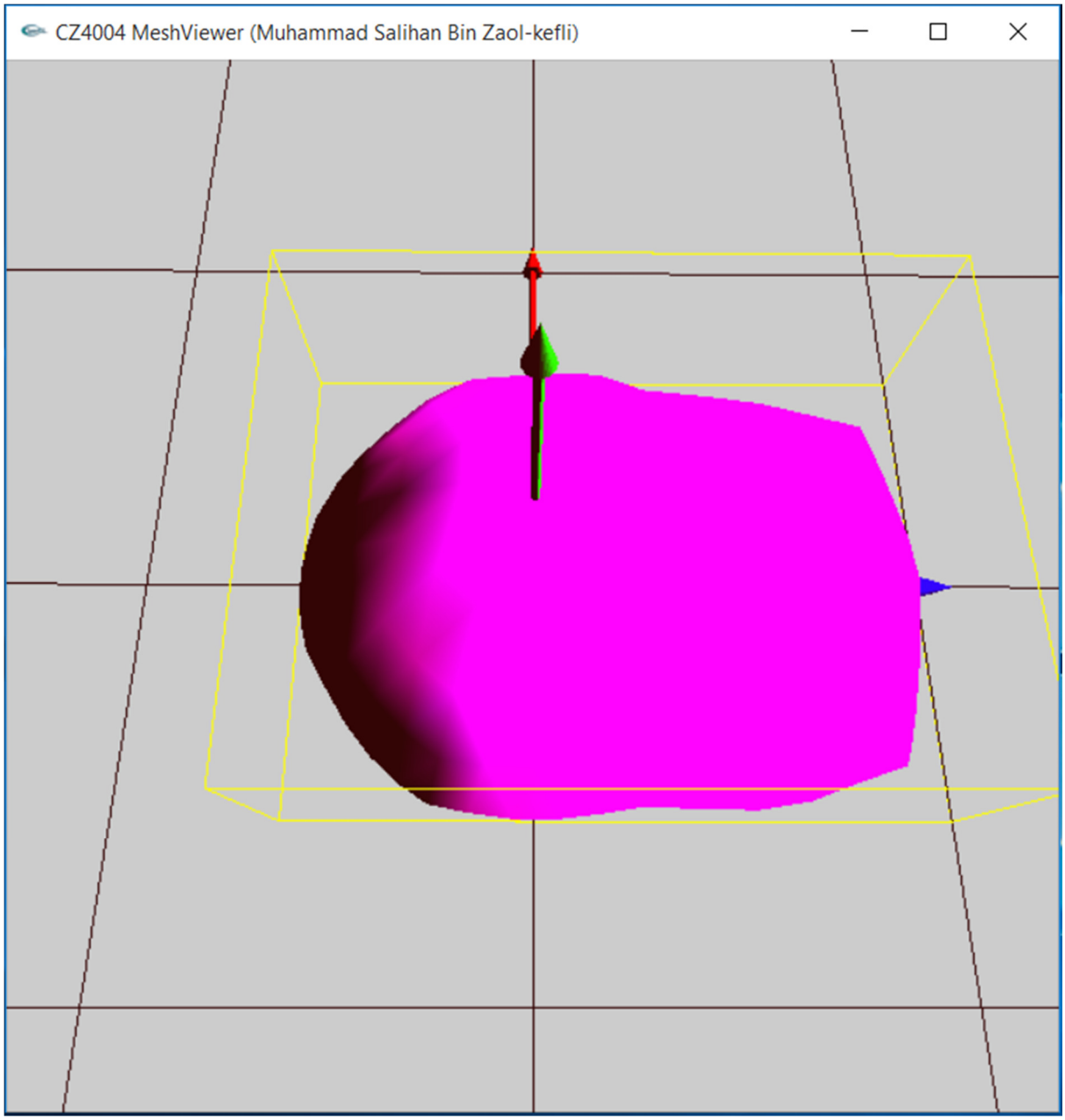


Figure 7 Smooth shading rendering of Cap model

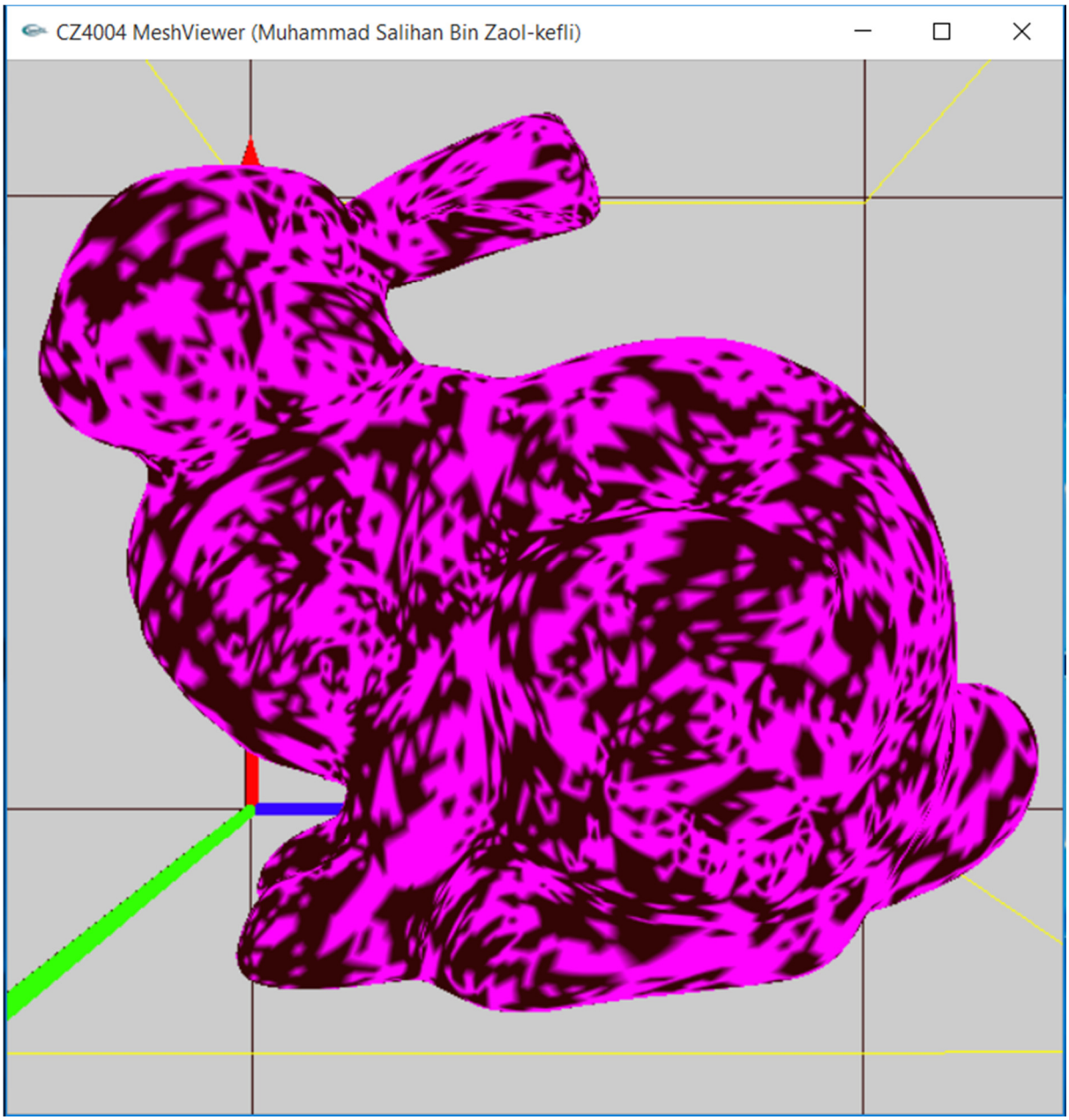


Figure 8 Smooth shading rendering of bunny model