UMS
UNIVERSITI MALAYSIA SABAH

MCG
Mathematics Computer Graphics
Universiti Malaysia Sabah

UH6461002 MATHEMATICS WITH COMPUTER GRAPHICS

FACULTY OF SCIENCE AND NATURAL RESOURCES,

SEMESTER 1 SESSION 2023/2024

SC40103 SCIENTIFIC DATA VISUALIZATION

ASSIGNMENT 1

NAMES || MATRIC NUMBER:	1. Wong Xiu Jian | BS20110002 2. Yap Jia Jun | BS20110220
LECTURER:	Prof. Abdullah Bade

Table of Contents

Introduction

In the realm of scientific computing and data analysis, visualization plays a crucial role in transforming complex numerical data into comprehensible and insightful visual representations. The Visualization Toolkit (VTK) emerges as a powerful tool in this domain, providing a comprehensive suite of open-source libraries and tools for creating interactive 3D graphics and visualizations.

VTK's capabilities extend far beyond mere rendering of 3D models. It encompasses a wide range of features, including image processing, volume rendering, and advanced modelling techniques. This versatility has made VTK a cornerstone of scientific visualization, enabling researchers and developers to effectively communicate their findings and insights through compelling visual representations.

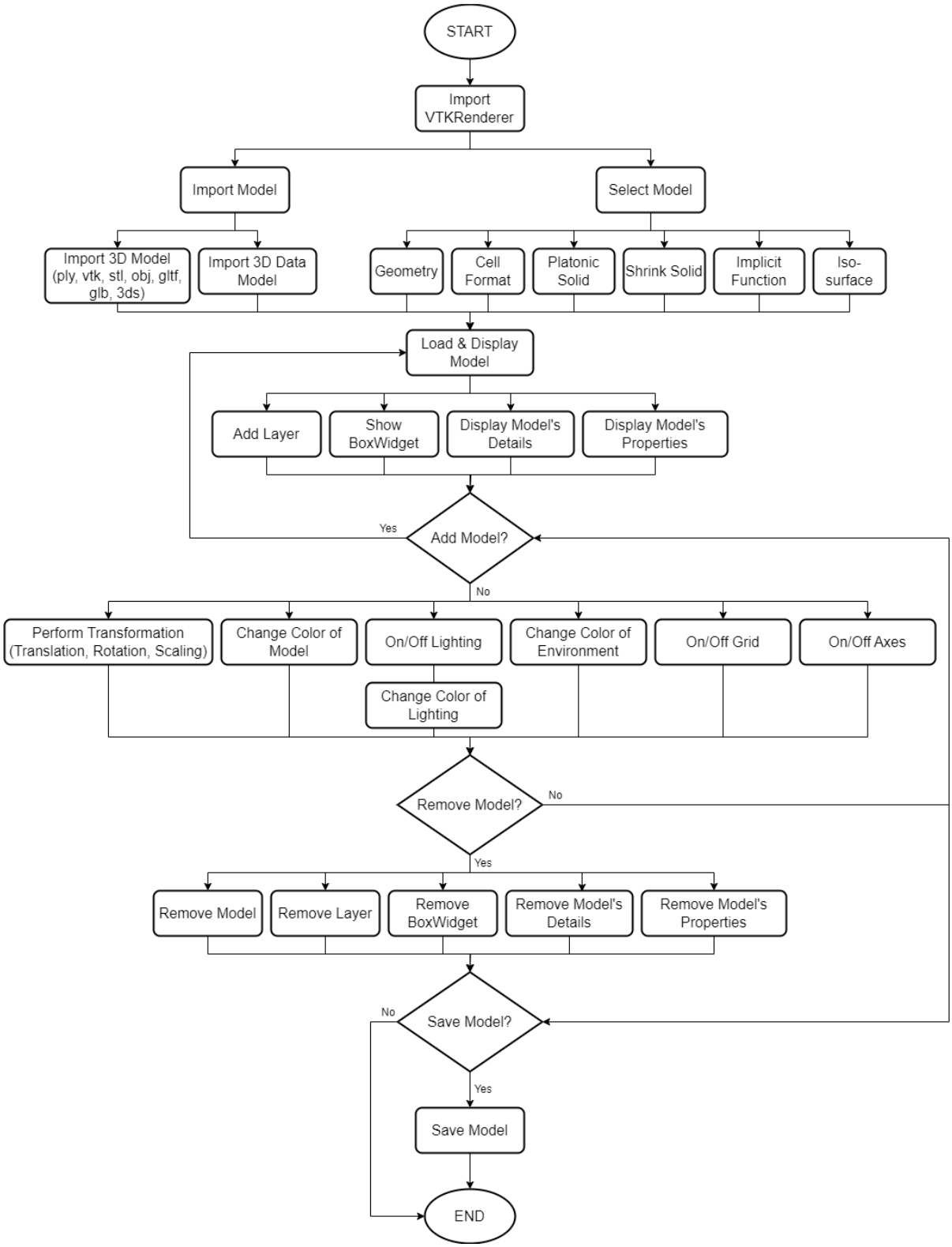
This assignment delves into the foundations of VTK, exploring its capabilities through the creation of a 3D object/geometry visualizer. This visualizer showcases VTK's ability to load, visualize, and manipulate various types of 3D models, including those based on different file formats, geometric primitives, cell formats, sources, parametric objects, implicit functions, and working 3D data.

The visualizer seamlessly integrates with 3D Studio scenes, incorporating multiple actors, and offers a user-friendly interface built upon PyQt5. Furthermore, it provides detailed information about the loaded models, including the number of surfaces and points, volume of object, total surface area, and enables colour and lighting adjustments.

To enhance user interaction, the visualizer incorporates basic transformations triggered through a menu or mouse clicks. Additionally, it offers the ability to save and write modified models, preserving the user's customizations.

Through this assignment, we embark on a journey to uncover the depths of VTK's capabilities and harness its power to transform scientific data into meaningful visualizations.

Flow Chart


START
Import VTKRenderer
Import Model
Select Model
Import 3D Model (ply, vtk, stl, obj, gltf, glb, 3ds)
Import 3D Data Model
Geometry
Cell Format
Platonic Solid
Shrink Solid
Implicit Function
Iso-surface
Load & Display Model
Add Layer
Show BoxWidget
Display Model's Details
Display Model's Properties
Add Model?
Yes
No
Perform Transformation (Translation, Rotation, Scaling)
Change Color of Model
On/Off Lighting
Change Color of Environment
On/Off Grid
On/Off Axes
Change Color of Lighting
Remove Model?
No
Yes
Remove Model
Remove Layer
Remove BoxWidget
Remove Model's Details
Remove Model's Properties
Save Model?
No
Yes
Save Model
END

System Architecture

The 3D object visualizer utilizes the VTK library for rendering and the PyQt5 library for its user interface. It can load, render, and interact with 3D objects, geometry, and 3D data using VTK readers. Additionally, the visualizer allows users to manipulate objects interactively using the widget box for transformations. Users can modify the color and lighting of objects and the background environment. Users can extract information about the loaded 3D model, such as volume, total surface area, number of surfaces, and number of points. Finally, users can save and export the modified model.

1. 3D Reader
   This visualizer allows users to load 3D file format such as PLY, VTK, STL, OBJ, GLTF, 3DS, and Digital Imaging and Communications in Medicine (DICOM). Users can load these models by clicking the button provided in the user interface and select the file or folder they want to import.

2. Geometry Loader
   Users can add the 3d geometry object into the render scene. Theses geometries included basic geometry such as cube, sphere, cone, cylinder, torus, cell format models, platonic solid, shrink solid, geometry created by implicit function, and iso-surfaces models.

3. Layer
   All the loaded geometries/objects/models will be show at the layer frame. Users can switch the models that loaded on the scene anytime by selecting them.

4. Statistics
   Visualizer provides a frame to show the details of the loaded model, such as name, volume of the object, total surface area, number of surfaces, and number of points. This information will be calculated once the model is loaded and will be store in the dictionary to reduce the work of system when changing object frequently.

5. Environment
   Users can change the background color of the scene by selecting the color or type the color in this frame. This frame also provides function to show grid and axis to ease users to identify direction in the scene.

6. Transformation

   Users can do transformation such as translation, rotation, and scaling on the models by changing the values in the transformation frame or dragging the box widget on the model.

7. Box Widget

   When the object is loaded, the box widget will applied on it to assist users to perform transformations.

8. Color Object

   Color of the object/geometry/model can be change by selecting the color under the properties frame.

9. Lighting

   Users can on or off the lighting effect applied on the models loaded on the scene, various types of color provided to change the lighting.

10. Save Model

    Users can save the modified models in vtk format.

Algorithm Used

3D Reader

1. Open file dialog to get open file/folder name.
2. Split the file name to get object name and type of file.
3. If PLY file type
    3.1 reader= vtkPLYReader
4. If VTK file type
    4.1 reader= vtkPolyDataReader
5. If STL file type
    5.1 reader= vtkSTLReader
6. If OBJ file type
    6.1 reader= vtkOBJReader
7. If GLTF file type
    7.1 reader= vtkGLTFReader
8. If 3DS file type
    8.1 reader= vtk3DSImporter
9. If DICOM file type
    9.1 reader= vtkDICOMImageReader
10. Get output port from reader
11. Set a mapper to the output port
12. Assign mapper to actor
13. Add actor to renderer

Geometry Loader

1. Select type of geometry to load
2. Get output port from the geometry source
3. Set a mapper to the output port
4. Assign mapper to actor
5. Add actor the renderer

Layer

1. If an object is loaded
    1.1 Set solid_name, actor, details to dictionary
    1.2 Show list of objects loaded

2. If object is deleted
    2.1 Delete dictionary[ID]
    2.2 Delete actor
    2.3 Delete layer

Show Details

1. The object is loaded
2. Polydata= actor.GetMapper.GetInput
3. Set polydata into triangle filter
4. Set the filter into mass properties
5. Get volume of object
6. Get total surface area of the object
7. Get number of surface from the polydata
8. Get number of points from the polydata

Environment

1. Choose the color
2. SetBackground of the renderer to the selected color
3. If Grid check box is open
    3.1 Show grid
4. If Axe check box is open
    4.1 Show x-y-z axis

Transformation

1. Get the value tx, ty, tz, rx, ry, rz, sx, sy, sz for translation, rotation and scaling value respectively
2. Input the values into vtkTransform
3. SetUserTransform(vtkTransform)
4. Transform the box widget along with the object

Box Widget

1. The object is loaded
2. Create box widget
3. If the box widget is drag
    3.1 Transform =BoxWidget.GetTransform

3.2 Actor.SetUserTransform(Transform)

Color Object

1. The color is selected
2. Get the properties of actor and set its color to the selected color

Lighting

1. Color of light is selected
2. light_actor= vtkLightActor
3. light_actor.SetLight(selected color)
4. Add light to renderer

Save Model

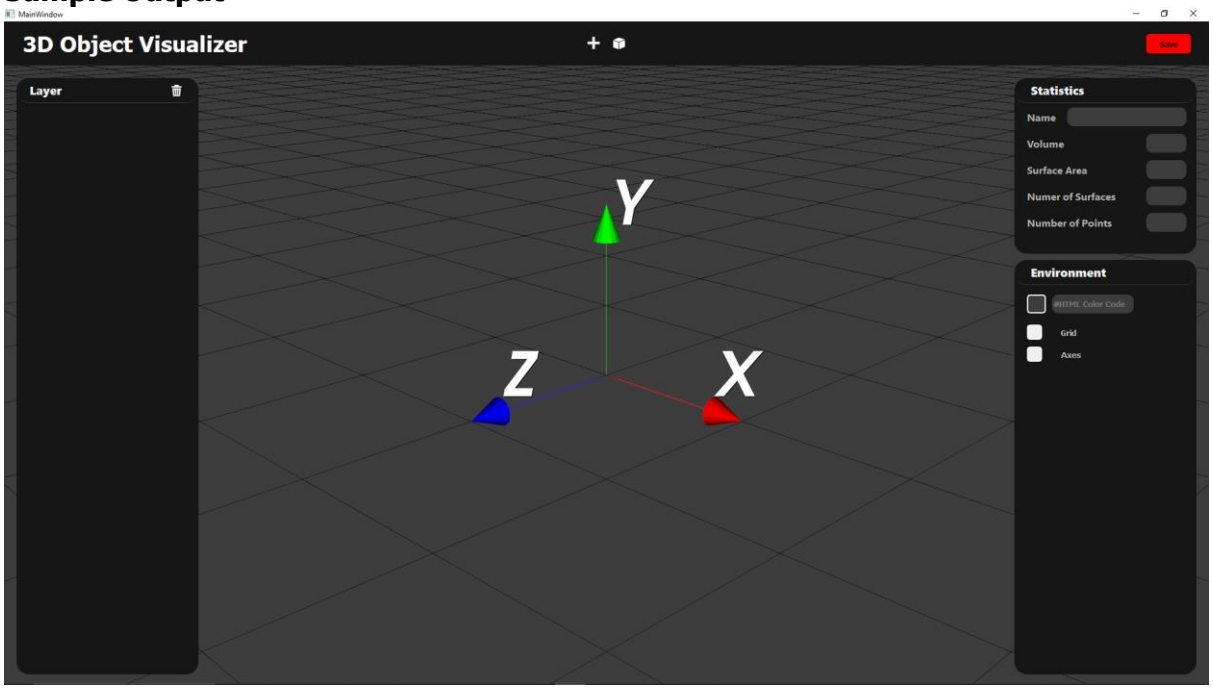
1. If there is an actor on scene
2. Get actor color
3. Save actor color
4. Get actor transformation
5. Save actor transformation
6. Get polydata of actor
7. Save model

Strength and Uniqueness

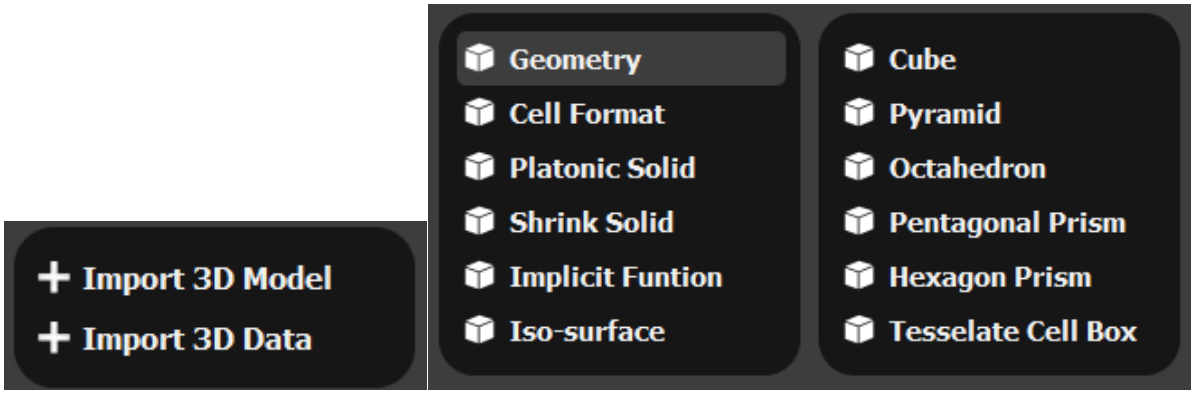
This 3D visualizer has the following strength and uniqueness.

1. It can import various types of 3D file format.
2. It can import 3D data.
3. It can load, visualize and display models such as geometric object, cell format, sources format, parametric objects and model based on implicit function and iso-surface.
4. Able to import 3D Studio scene that includes multiple actors.
5. Able change the background color of the scene.
6. Able change the color and light of the model.
7. Able to select and delete models loaded in scene.
8. Able to show details of the model.
9. Able to perform transformations by menu and clicking mouse on the model.
10. Able to save and write the modified model.
11. It can show grid and axis.

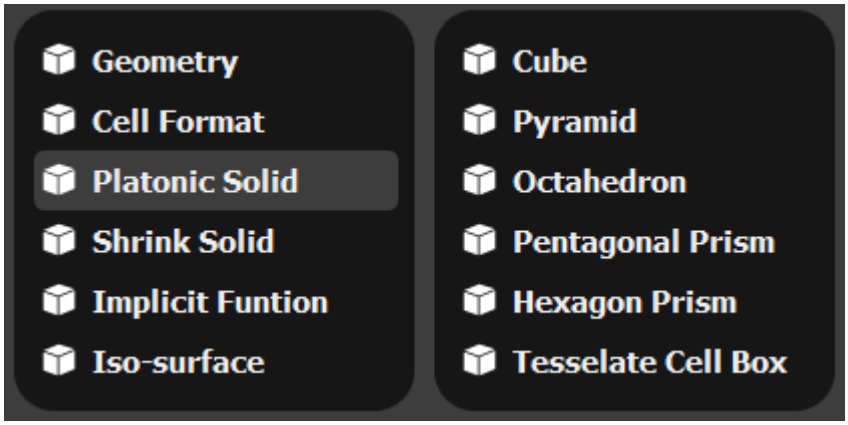
Sample Output



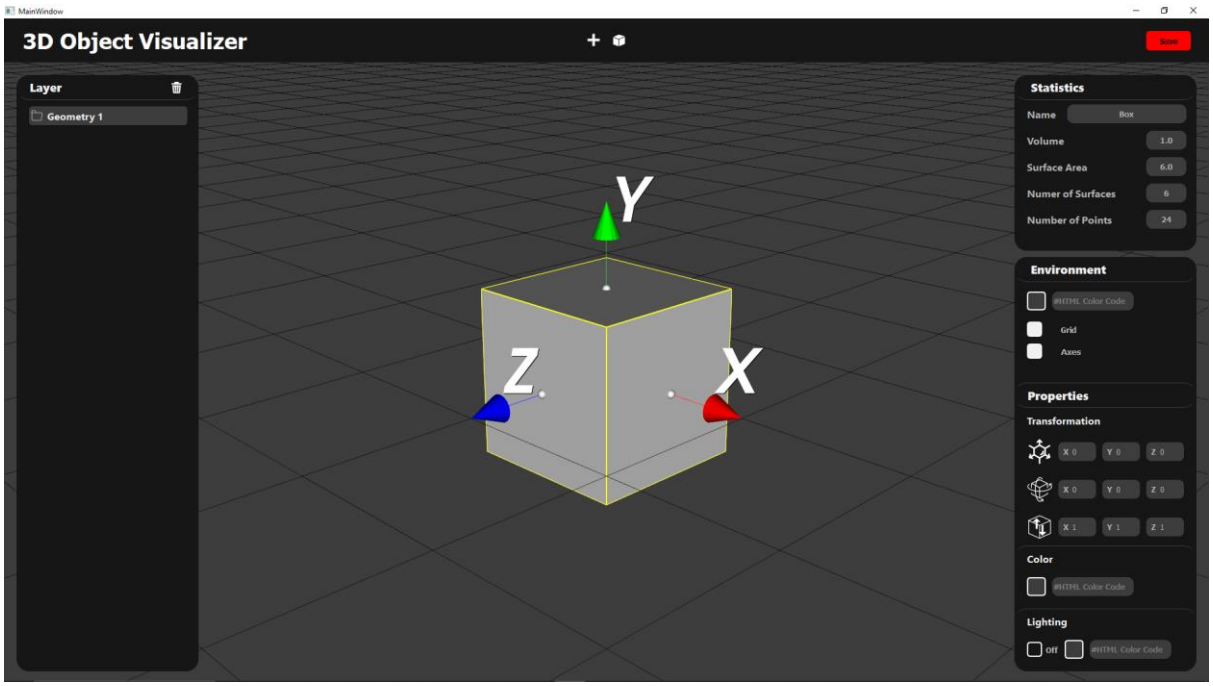
The main graphics use interface of our visualizer



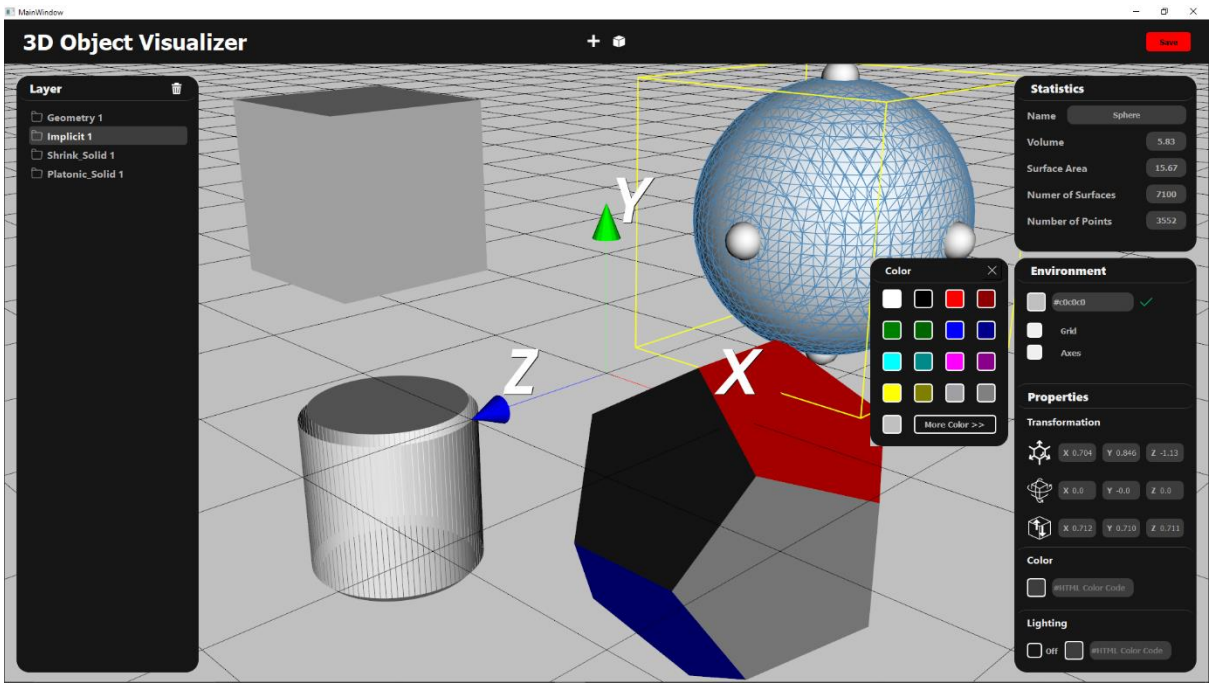
The drop-down menu for import model (Left). The drop-down menu for select build-in model (Right).



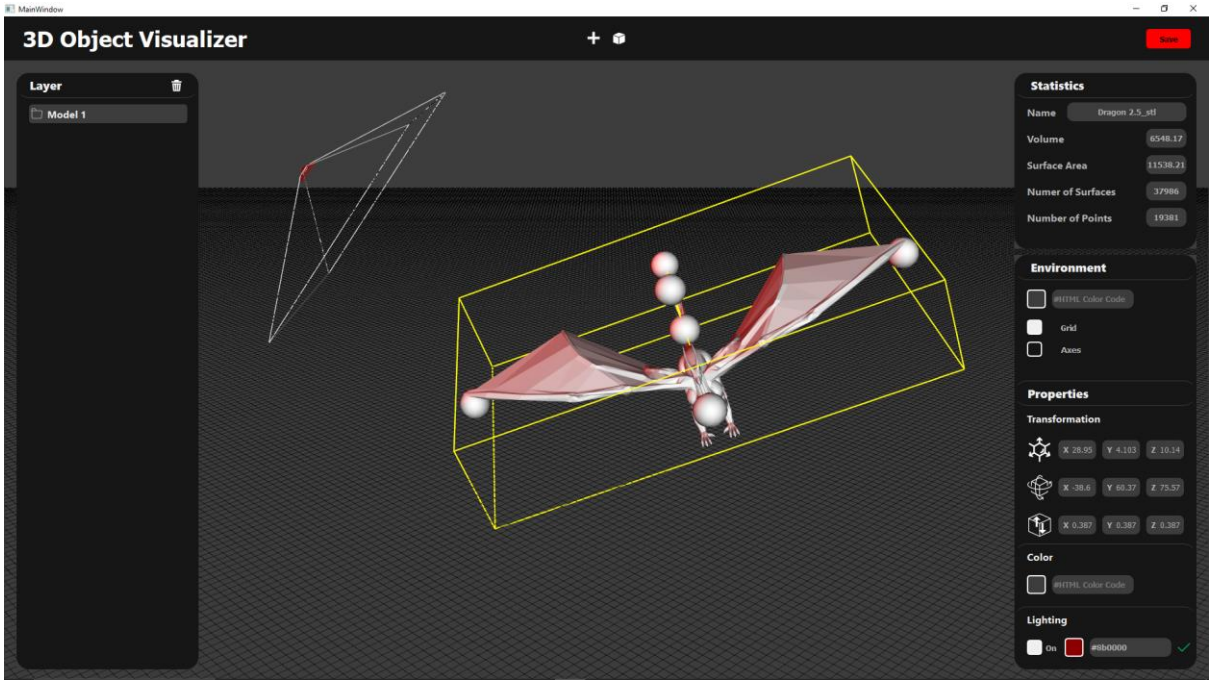
The drop-down menu for select Platonic Solid.



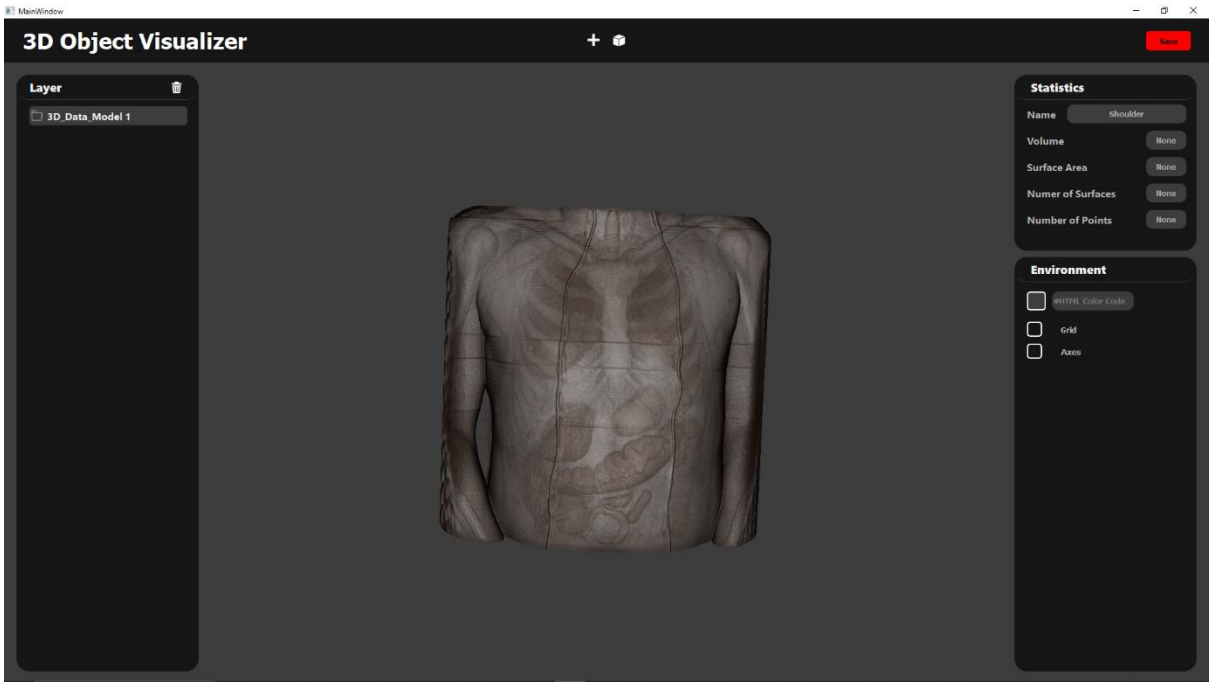
The interface after importing a model and displays. Left side is the frame to display the layer of model indicates that there are how many models in the scene. Right sides consists of two frame where one is to display the details of the model and another is store functional feature such as change color, on/off grid, axes and lighting and perform transformation.



User can import multiple models and manipulate each model by selecting the model in the layer frame.



The diagram shows the lighting actor and the color of lighting can be changed.



The diagram shows the imported model in 3D Data.