

# MM 804 - Graphics and Animations

**Link to GitHub:** <https://github.com/Har-Pan/MM-804---Assignment-1.git>

## Download the required packages in your local system with the given commands.

```
pip install -r requirements.txt
```

Downloaded the 3D model from <https://www.thingiverse.com/thing:5167431>

## Information on the stl file

Name: House\_Tower\_Gate.stl

Size: 50.7 MB or 50,672,934 bytes

Type: STL

Vertex Count : 1,517,619

## Setup on Mac and Windows

1. Install Python3 version 3.8.8

Refer the website <https://docs.python-guide.org/starting/install3/osx/>

2. Install the requirements package

```
pip install -r requirements.txt
```

3. Run the Script

```
python visualizer.py
```

## Run Through Code

```
import vtk      #import the vtk model
reader = vtk.vtkSTLReader()  # Read the model with the help of VtkSTLREADER
reader.SetFileName("House_Tower_Gate.stl")  #FileName set to House_Tower_Gate.stl

def rotate_Actor(actor,x,y,z):  # Function to rotate the model at x,y, and z axis.
    actor.RotateX(x)
    actor.RotateY(y)
    actor.RotateZ(z)
    return actor

def set_Prop(prop):  # Function to setup the common property for the model

    prop.ShadingOn()
    prop.SetColor(1, 0, 0)
    prop.SetDiffuse(0.8)
    prop.SetAmbient(0.3)
    prop.SetSpecular(1.0)
    prop.SetSpecularPower(100.0)

def setup_Light():          # Function to setup lightining for the models
    Light = vtk.vtkLight()
    Light.SetLightTypeToSceneLight()
    Light.SetAmbientColor(1, 1, 1)
    Light.SetDiffuseColor(1, 1, 1)
    Light.SetSpecularColor(1, 1, 1)
    Light.SetPosition(-100, 100, 25)
    Light.SetFocalPoint(0,0,0)
    Light.SetIntensity(0.8)
    return Light

def Create_View_Port(read,FileName,Rot_X=-90,Rot_Y=0,Rot_Z=0):  #Create ViewPort
fuction

    set_normal = vtk.vtkPolyDataNormals()  # CreateViewPort function setup normals
which will be used in the mapper.
    set_normal.SetInputConnection(read.GetOutputPort())

    set_map = vtk.vtkPolyDataMapper()      # Now mapper is setup where
vtkPolyDataMapper() fuction is used.
    set_map.SetInputConnection(set_normal.GetOutputPort())

    # Now, different actors would be setup for all the different views
    act_1 = vtk.vtkActor()
```

```

act_2 = vtk.vtkActor()
act_3 = vtk.vtkActor()
act_4 = vtk.vtkActor()

#Actor 1 -> for ViewPort 1 for wireframe representation
set_map.SetInputConnection(read.GetOutputPort())
act_1.SetMapper(set_map)
act_1 = rotate_Actor(act_1, Rot_X, Rot_Y, Rot_Z)
act_1.GetProperty().SetRepresentationToWireframe()

# Actor 2 -> for ViewPort 2 for surface representation (Flat Shading)
act_2.SetMapper(set_map)
property2 = act_2.GetProperty()
property2.SetInterpolationToFlat() # Here the shading is set to Flat
set_Prop(property2)

act_2 = rotate_Actor(act_2, Rot_X, Rot_Y, Rot_Z)

# Now, we setup lights for the other models flat, gourand and Phong.
light = setup_Light()
act_3.SetMapper(set_map)
property3 = act_3.GetProperty()
property3.SetInterpolationToGouraud() # This function will setup shading for the
Gourand View.
set_Prop(property3)
act_3 = rotate_Actor(act_3, Rot_X, Rot_Y, Rot_Z)

act_4.SetMapper(set_map)
property4 = act_4.GetProperty()
property4.SetInterpolationToPhong() # This function will built the shading for
Phong View.
set_Prop(property4)
act_4 = rotate_Actor(act_4, Rot_X, Rot_Y, Rot_Z)

render_1 = vtk.vtkRenderer()
render_2 = vtk.vtkRenderer()
render_3 = vtk.vtkRenderer()
render_4 = vtk.vtkRenderer()

# We set all the viewports for each model and add them to different ports
respectively.
render_1.SetViewport(0, 0.5, 0.5, 1)
render_2.SetViewport(0.5, 0.5, 1.0, 1.0)
render_3.SetViewport(0, 0, 0.5, 0.5)
render_4.SetViewport(0.5, 0, 1.0, 0.5)
render_1.AddActor(act_1)
render_2.AddActor(act_2)
render_3.AddActor(act_3)

```

```

render_4.AddActor(act_4)

render_2.AddLight(light)

renderWindow = vtk.vtkRenderWindow()
renderWindow.SetSize(700, 600)
renderWindow.AddRenderer(render_1)
renderWindow.AddRenderer(render_2)
renderWindow.AddRenderer(render_3)
renderWindow.AddRenderer(render_4)
renderWindow.Render()

# Finally the image would be saved as a Jpeg with the help of JPEGWriter()
function

w = vtk.vtkWindowToImageFilter()
w.SetInput(renderWindow)
w.Update()
j_w = vtk.vtkJPEGWriter()
j_w.SetInputData(w.GetOutput())
j_w.SetFileName(FileName)
j_w.Write()

# Create_View_Port saves our results for different angles as jpg file.

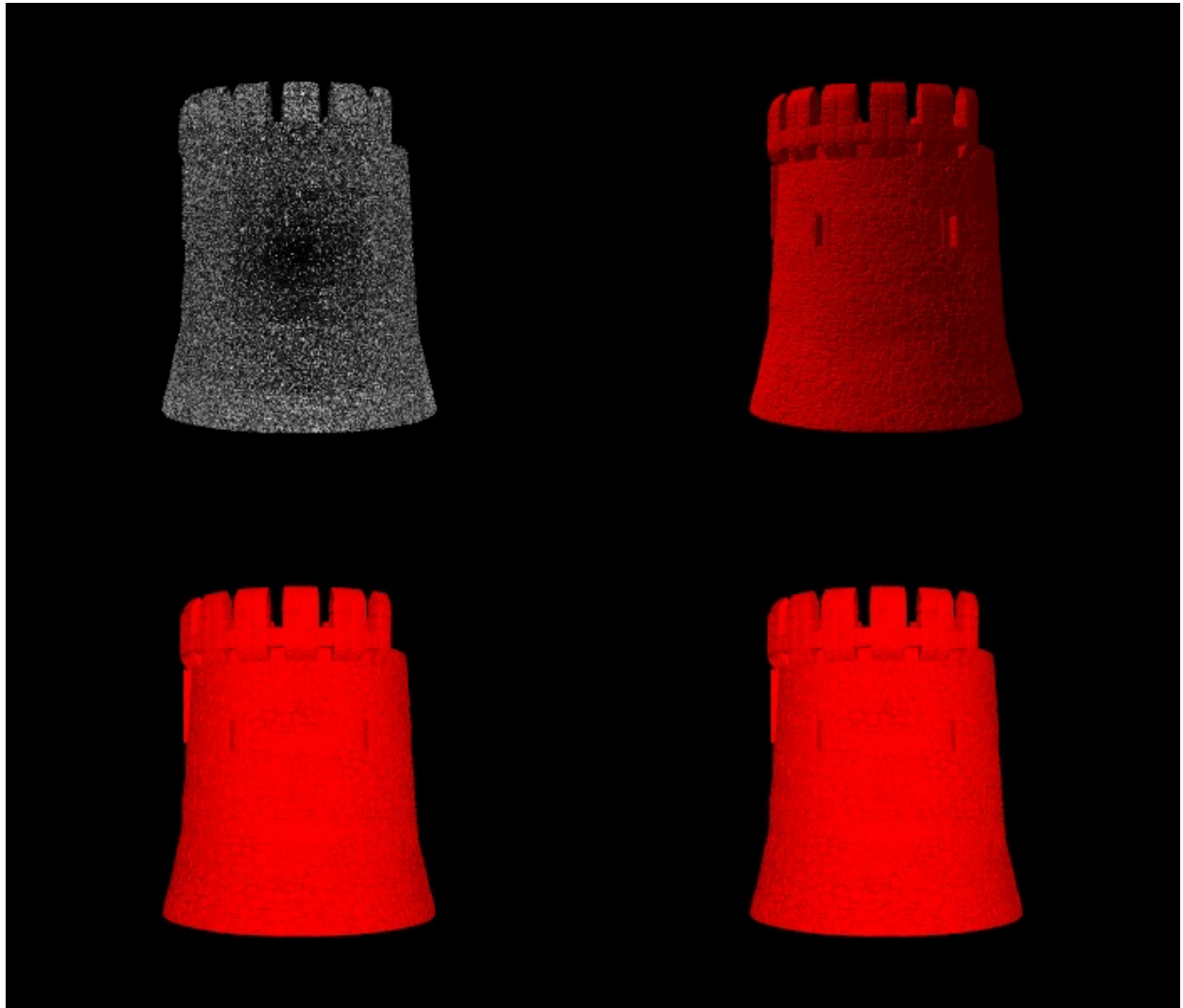
Create_View_Port(Rot_X=-90, Rot_Y=0, Rot_Z=0, read=reader, FileName='First_angle.png')

Create_View_Port(Rot_X=-90, Rot_Y=0, Rot_Z=180, read=reader, FileName='Second_angle.png')

```

## OutPut

## First Angle Output



## Second Angle Output

