Importing and Displaying 3D Objects in Python using OpenGL

In Task 1.2, you had used OpenGL with Python to superimpose a 3D Model of a teapot on top of an ArUco marker.
In this tutorial you will learn how to import a 3D model in .obj format into Python and superimpose that model on top of the ArUco marker.

To implement this, you are required to open the ***GLteapot.py*** python file which you modified in Task 1.2 and make changes to that file.

Please do the following steps:
1) Go to the **Resources** folder given in this Task and copy the *Crow_Model.zip* and the *objloader.py* python file to the same directory as your *GLteapot.py* file.
2) Extract the contents of the *Crow_Model.zip*. You will find three files in the zip file. They are *Crow.blend*, *Crow.obj* and *Crow.mtl*. Make sure that all these file are in the same directory as the *GLteapot.py* file
3) Open the GLteapot.py file and make the following changes:
   a) Importing the objloader.py module

```
import numpy as np
import cv2
import cv2.aruco as aruco
import math
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
from PIL import Image
import pygame
from objloader import *   <----- Add this line
```

   b) Initialising a global object

```
texture_object = None
texture_background = None
camera_matrix = None
dist_coeff = None
cap = cv2.VideoCapture(0)
crow = None                   <----- Add this line
```

```
def init_gl():
   global texture_object, texture_background
   global crow              <----- Add this line
   glClearColor(0.0, 0.0, 0.0, 0.0)
   glClearDepth(1.0)
   glDepthFunc(GL_LESS)
```

```
    glEnable(GL_DEPTH_TEST)
    glShadeModel(GL_SMOOTH)
    glMatrixMode(GL_MODELVIEW)
    glEnable(GL_DEPTH_TEST)
    glEnable(GL_LIGHTING)
    glEnable(GL_LIGHT0)
    texture_background = glGenTextures(1)
    texture_object = glGenTextures(1)
    crow = OBJ('crow.obj', swapyz=True)<----- Add this line
```

Calling the OBJ function from objloader.py will import the .obj file to Python

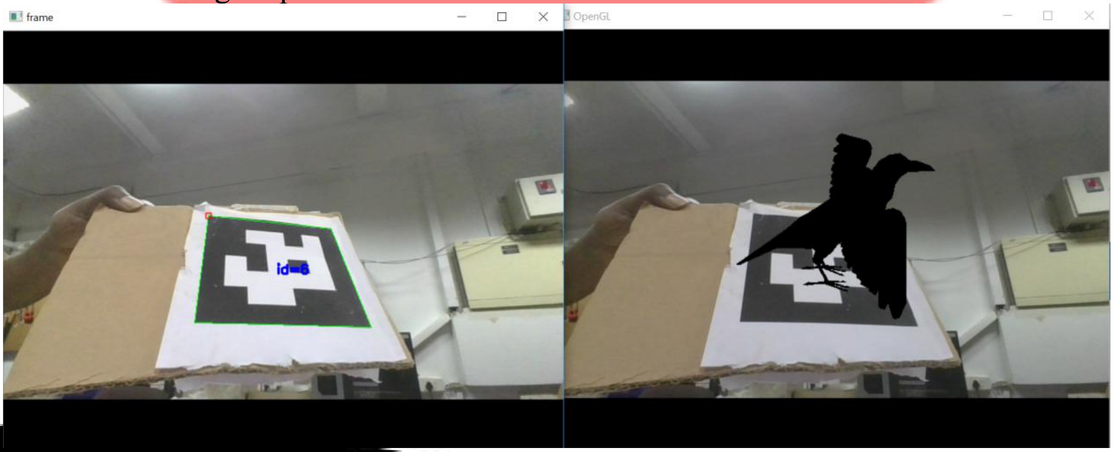
c) Modifying the Overlay function

```
def overlay(img, ar_list, ar_id, texture_file):
        for x in ar_list:
                if ar_id == x[0]:
                        centre, rvec, tvec = x[1], x[2], x[3]
        rmtx = cv2.Rodrigues(rvec)[0]
        view_matrix =
        view_matrix = view_matrix * INVERSE_MATRIX
        view_matrix = np.transpose(view_matrix)


        #init_object_texture(texture_file) <-- Comment this
        glPushMatrix()
        glLoadMatrixd(view_matrix)
        #glutSolidTeapot(0.5)               <-- Comment this
        glCallList(crow.gl_list)          <----- Add this line
        glPopMatrix()
```

The overlay function that you modified in Task 1.2 should work fine. You just have to modify these three lines as depicted.

4) Once you have made these changes you are free to run your code. You should get the following output.



More Resources

- Augmented Reality using OpenCV, OpenGL and Blender by RD Milligan
- Pygame OBJFileLoader