OPENGL / PYTHON

Renaud.Seguier@centralesupelec.fr

With
prerequisites_opengl.py
you are able to display a pyramid

Now, you will try to move it and apply a texture coming from an image.

Herafter some documentation

OPENGL
http://www.opengl.org/sdk/docs/man/

GLUT
http://www.opengl.org/resources/libraries/glut/spec3/spec3.html

GLU
https://www.khronos.org/registry/OpenGL/specs/gl/glu1.3.pdf

Starting from prerequisites_opengl.py code follow those steps

A) Modify the perspective projections settings (near and far clip clips) to make sure they are useful. Place the pyramid at Z = -20, 0, 90, 110 to check the “clipping planes”.

B) Apply a rotation and a translation to the pyramid.To have an action on the matrix which controls the rotation of the objects, just after cleaning the buffer in the display function, insert the following commands

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
```

Then use the fonctions

```
glTranslatef();
glRotatef();
```

C) Manipulate the position of the pyramid using the keyboard.
You can use the keyboard function like that

```
def keyboard_simple(key, x, y):
    key_in_str = key.decode('utf-8')
    print(key_in_str)
```

If GLB_Xcursor and GLB_Y_cursor are the coordinates of your pyramid, increment or decrement them on each keystroke. You can do the same with the orientation. Do not forget to force the display like wat is done at the end of the display function.

D) Install Opencv for Python
In the terminal :

Pip install opencv_python

And import it in the code with

```
import cv2
```

We are going to apply a texture to the triangles. It is saved in the im.bmp image.
Declare the following global variables

texture size
GLB_large_text,GLB_haut_text=256,256;

Build the following function which allows you to associate a bmp file with a texture taken into account by OpenGl

```
def LoadTexture():
    pMatrice = cv2.imread('im.bmp')
    texID = glGenTextures(1)
    glBindTexture(GL_TEXTURE_2D, texID)
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR)
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR)
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, GLB_large_text, GLB_haut_text, 0,
GL_RGB, GL_UNSIGNED_BYTE, pMatrice)
```

You just need to call this function as follows during initialization (init) after allowing OpenGl to manipulate textures:

glEnable(GL_TEXTURE_2D);
LoadTexture();

In the display function, the coordinates of the texture change between 0 and 1.
In this display function, rather than specifying the color of each vertex, it is necessary to give its reference coordinate in the texture. for example

```
Utext = 55.0 / GLB_large_text
Vtext = 162.0 / GLB_haut_text
glTexCoord2f(Utext, Vtext)
glVertex3f(1.0, 0.0, 0.0)
```

Which means that the vertex coordinate (1, 0, 0) is associated with the pixel (Utext, Vtext) of the image. Start displaying a bit of texture in a triangle.