

# **Short-Story-Graphics**

<b><u>Name</u></b>	<b><u>Id</u></b>
Joshua Tesfaye	UGR/0359/12
Dagim Fikru	UGR/4328/12

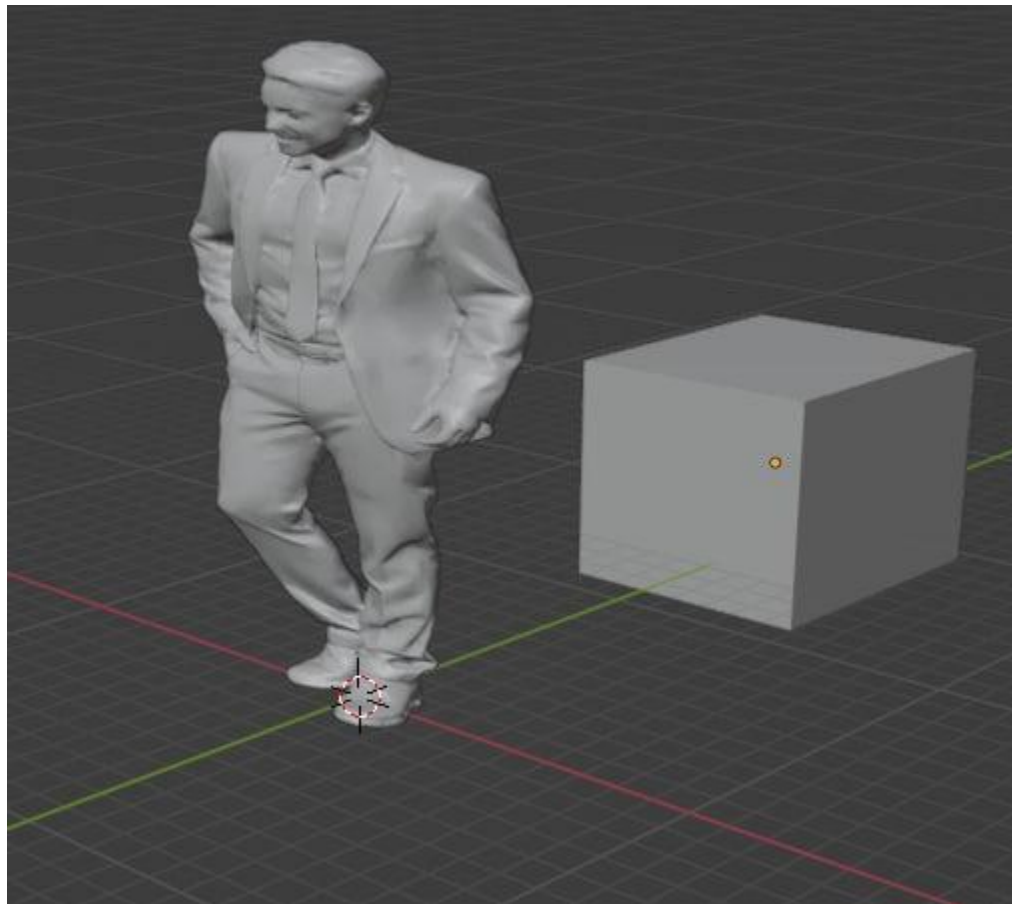
First, we started our work by thinking about what type of story could we make after that we wrote the story as follow:

***The story is 2 of our 3D characters (a man and a woman) chatting about their respective view about the job they have the man is a software engineer whereas the woman is a lawyer there is a little argument as well as sarcasm in their chat (detail in the demonstration).***

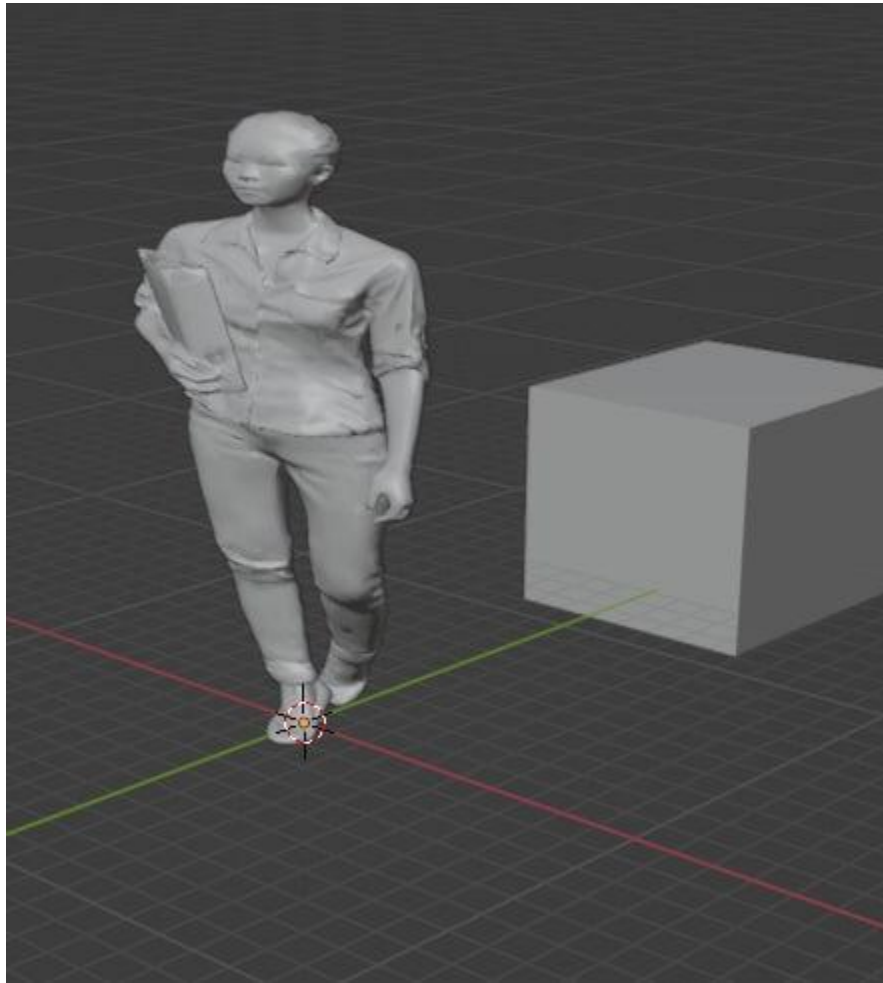
After writing the story we started making models of our characters by using software used to create and animate 3D models called **blender**.

The models we created are shown below:

**1. Man model:**



## 2. Girl model:



## 3. The building Model:



After creating the models we set up our working environment to render the models we use by using OpenGL with pygame.

First, we create the camera to view the objects on the screen.

In the **camera.py** file we have Camera class to create different objects in it.

In the Camera class we have instance variables:

- **self.camera\_pos = Vector3([0.0, 15.0, 50.0])**
- **self.camera\_front = Vector3([0.0, 0.0, -5.0])**
- **self.camera\_up = Vector3([0.0, 1.0, 0.0])**
- **self.camera\_right = Vector3([1.0, 0.0, 0.0])**
- 

The above variables controll the view of a camera as specified in their assigned vectors.

The other instance variables are:

- **self.mouse\_sensitivity = 0.25**
- **self.jaw = -90**
- **self.pitch = 0**

The above variables play crucial role in the mouse movement from a user.

**For detail check *camera.py***

Now the hard job comes in, rendering the models we created to our OpenGL window. We do this in the file **main.py**. The functions of each part of the code is mentioned below with each code snippets.

### **1. Importing:**

```
from camera import Camera
from ObjectLoader import ObjLoader
from TextureLoader import load_texture_pygame
import pyrr
from OpenGL.GL.shaders import compileProgram, compileShader
from OpenGL.GL import *
import pygame
import os
import time
from playsound import playsound
```

***The import statements are much of self descriptors***

## 2. Creating Story class:

```
class Story:
    def mouse_look(self, xpos, ypos):
        global first_mouse, lastX, lastY

        if self.first_mouse:
            lastX = xpos
            lastY = ypos
            self.first_mouse = False

        xoffset = xpos - lastX
        yoffset = lastY - ypos

        lastX = xpos
        lastY = ypos

        self.cam.process_mouse_movement(xoffset, yoffset)

    def __init__(self) -> None:

        vertexPath = "shaders/vertex.txt"
        fragmentPath = "shaders/fragment.txt"

        with open(vertexPath, 'r') as f:
            vertex_src = f.readlines()

        with open(fragmentPath, 'r') as f:
            fragment_src = f.readlines()
```

This class contain different function with different functionalities one of the function is to control the mouse and read the shader And fragment files ***vertex.txt*** and ***fragment.txt*** files respectively.

## 3. Setting the camera:

```
# CAMERA settings
self.cam = Camera()
self.WIDTH, self.HEIGHT = 1280, 720
lastX, lastY = self.WIDTH / 2, self.HEIGHT / 2
self.first_mouse = True
pygame.init()
self.window = pygame.display.set_mode((self.WIDTH, self.HEIGHT), pygame.OPENGL |
                                       pygame.DOUBLEBUF | pygame.RESIZABLE) # |pygame.FULLSCREEN
pygame.mouse.set_visible(False)
pygame.event.set_grab(True)
```

This simply adjust the camera(the class is in the **camera.py** file) and create the pygame window.

#### 4. Loading 3D objects:

```
self.building_indicies, self.building_buffer = ObjLoader.load_model(
    "meshes/building.obj", False)
self.woman_indicies, self.woman_buffer = ObjLoader.load_model(
    "meshes/3dwoman.obj")
self.man_indicies, self.man_buffer = ObjLoader.load_model(
    "meshes/3dman.obj")
self.floor_indicies, self.floor_buffer = ObjLoader.load_model(
    "meshes/floor.obj")

shader = compileProgram(compileShader(
    vertex_src, GL_VERTEX_SHADER), compileShader(fragment_src, GL_FRAGMENT_SHADER))
```

In this we loaded our models to **OpenGL**.

**The next step is building a vertices and textures.**

After loading the objects we create vertices, normals and textures for our man, woman, floor and building objects.(for detail check **main.py**)

#### 5. the main function:

```
def main(self):
    # font = pygame.font.SysFont("Times New Roman, Arial", 30)
    # text = font.render("Abebe beso bela", True)
    running = True

    while running:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                running = False
            elif event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE:
                running = False

            if event.type == pygame.VIDEORESIZE:
                glViewport(0, 0, event.w, event.h)
                projection = pyrr.matrix44.create_perspective_projection_matrix(
                    45, event.w / event.h, 0.1, 100)
                glUniformMatrix4fv(self.project_location,
                                    1, GL_FALSE, projection)

        keys_pressed = pygame.key.get_pressed()
        if keys_pressed[pygame.K_a]:
            self.cam.process_keyboard("LEFT", 0.08)
        if keys_pressed[pygame.K_d]:
            self.cam.process_keyboard("RIGHT", 0.08)
        if keys_pressed[pygame.K_w]:
            self.cam.process_keyboard("FORWARD", 0.08)
        if keys_pressed[pygame.K_s]:
            self.cam.process_keyboard("BACKWARD", 0.08)
```








This function is used to display the window as well as control the movement of screen by using keyboard.

## 6. Play the audio:

The last step is playing the audio or sound of our characters:

```
while(self.runonce == True):  
    playsound('audio/1.mp3')  
    playsound('audio/2.mp3')  
    playsound('audio/3.mp3')  
    playsound('audio/4.mp3')  
    playsound('audio/5.mp3')  
    playsound('audio/6.mp3')  
    self.runonce = False  
  
pygame.quit()
```

# References

1.  How to Play Audio Files in Python : playsound Module → this video helped us to insert audio to our story
2. <https://free3d.com/> → this helped us to get different object models
3.  OpenGL in python e15 - loading 3D .obj files  
 OpenGL with Python 5: Loading Obj Models  
 OpenGL with Python Tutorial 9: Loading Obj Models (deprecated) → these videos helped us to load the objects to opengl
4.  OpenGL in python e07 - texturing a cube → this video helped us to create textures to the models
5.  Camera/View Space // OpenGL Beginners Series → this series helped to do different parts of the project
6.  OpenGL with Python 4: Applying Transformations → this video helped us to do transformations.