

OpenGL Python

Lucas e Meani

Objetivo

- Criar uma sala de estar de uma casa com objetos com diversas texturas.

Dentro da sala contém os seguintes objetos:

Sofá, esteira, televisão, rack, ambiente interno

Objeto com 1 textura

```
✓ def inicializaParede():
    global Vao_parede, parede_indices, parede_texture, parede_buffer
    parede_indices, parede_buffer = ObjLoader.load_model("meshes/cube.obj")

    #Vao do objeto
    Vao_parede = glGenVertexArrays(1)
    glBindVertexArray(Vao_parede)

    #VBO do objeto. Ao invés de termos 1 VBO para cada informação (vértices, texturas, normais, etc.), este exemplo utiliza
    #1 único VBO com todas as informações dentro
    bvbo = glGenBuffers(1)
    glBindBuffer(GL_ARRAY_BUFFER, bvbo)
    glBufferData(GL_ARRAY_BUFFER, parede_buffer.nbytes, parede_buffer, GL_STATIC_DRAW)
    # vertices
    glEnableVertexAttribArray(0)
    glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, parede_buffer.itemsize * 8, ctypes.c_void_p(0))
    # texturas
    glEnableVertexAttribArray(1)
    glVertexAttribPointer(1, 2, GL_FLOAT, GL_FALSE, parede_buffer.itemsize * 8, ctypes.c_void_p(12))
    # normais
    glVertexAttribPointer(2, 3, GL_FLOAT, GL_FALSE, parede_buffer.itemsize * 8, ctypes.c_void_p(20))
    glEnableVertexAttribArray(2)

    #carrega os arquivos de textura
    parede_texture = glGenTextures(1)
    load_texture("textures/floor.jpg", parede_texture)
```

Objeto com mais de uma textura, usando glob

```
def inicializaEsteira():
    global Vao_esteira, esteira_indices
    esteira_indices, obj_buffer = ObjLoader.load_model("meshes/esteira.obj")

    #Vao do objeto
    Vao_esteira = glGenVertexArrays(1)
    glBindVertexArray(Vao_esteira)

    #VBO do objeto. Ao invés de termos 1 VBO para cada informação (vértices, texturas, normais, etc.), este exemplo utiliza
    #1 único VBO com todas as informações dentro
    bvbo = glGenBuffers(1)
    glBindBuffer(GL_ARRAY_BUFFER, bvbo)
    glBufferData(GL_ARRAY_BUFFER, obj_buffer.nbytes, obj_buffer, GL_STATIC_DRAW)
    # vertices
    glEnableVertexAttribArray(0)
    glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, obj_buffer.itemsize * 8, ctypes.c_void_p(0))
    # texturas
    glEnableVertexAttribArray(1)
    glVertexAttribPointer(1, 2, GL_FLOAT, GL_FALSE, obj_buffer.itemsize * 8, ctypes.c_void_p(12))
    # normais
    glVertexAttribPointer(2, 3, GL_FLOAT, GL_FALSE, obj_buffer.itemsize * 8, ctypes.c_void_p(20))
    glEnableVertexAttribArray(2)

    #carrega os arquivos de textura
    for obj in glob.glob("textures/esteira/*.jpg"):
        load_texture(obj, glGenTextures(1))
```

Modificação load_texture: erro out of index

```
with open(file, 'r') as f:
    line = f.readline()
    while line:
        values = line.split()
        if len(values) == 0:
            line = f.readline()
            continue

        if values[0] == 'v':
            ObjLoader.search_data(values, vert_coords, 'v', 'float')
        elif values[0] == 'vt':
            ObjLoader.search_data(values, tex_coords, 'vt', 'float')
        elif values[0] == 'vn':
            ObjLoader.search_data(values, norm_coords, 'vn', 'float')
        elif values[0] == 'f':
            for value in values[1:]:
                val = value.split('/')
                ObjLoader.search_data(val, all_indices, 'f', 'int')
                indices.append(int(val[0])-1)

        line = f.readline()
```

Draw dos objetos

```
glBindVertexArray(Vao_parede) #ativamos o objeto que queremos renderizar

#chao
especificaMaterial(0.2, 0.2, 0.2, 0.8, 0.8, 0.8, 0.1, 0.1, 0.1, 32)
transformacaoGenerica(0,0,1,10,.1,10,0,0,0)
glDrawArrays(GL_TRIANGLES, 0, len(parede_indices))

#teto
transformacaoGenerica(0,70,1,10,.1,10,0,0,0)
glDrawArrays(GL_TRIANGLES, 0, len(parede_indices))

#parede tras
transformacaoGenerica(3,1,0,10,3.5,.1,0,90,0)
glDrawArrays(GL_TRIANGLES, 0, len(parede_indices))

#parede <
transformacaoGenerica(1,1,100,.1,3.5,10,0,90,0)
glDrawArrays(GL_TRIANGLES, 0, len(parede_indices))

#parede >
transformacaoGenerica(1,1,-100,.1,3.5,10,0,90,0)
glDrawArrays(GL_TRIANGLES, 0, len(parede_indices))
```

Draw dos objetos

```
#rack
glBindVertexArray(Vao_rack) #ativamos o objeto que queremos renderizar
especificaMaterial(0.2, 0.2, 0.2, 0.8, 0.8, 0.8, 0.1, 0.1, 0.1, 32)
transformacaoGenerica(1,-18,90,.096,.096,.096,0,90,0)
glDrawArrays(GL_QUADS, 0, len(rack_indices))

#sofa
glBindVertexArray(Vao_sofa) #ativamos o objeto que queremos renderizar
especificaMaterial(0.2, 0.2, 0.2, 0.8, 0.8, 0.8, 0.1, 0.1, 0.1, 32)
transformacaoGenerica(0,0.1,-5,1.5,1.5,1.5,0,180,0)
glDrawArrays(GL_QUADS, 0, len(sofa_indices))
```

Draw dos objetos

```
#tv
glBindVertexArray(Vao_tv) #ativamos o objeto que queremos renderizar
especificaMaterial([0.2, 0.2, 0.2, 0.8, 0.8, 0.8, 0.1, 0.1, 0.1, 32])
transformacaoGenerica(-55,-60,28,.042,.042,.042,90,0,0)
glDrawArrays(GL_QUADS, 0, len(tv_indices))

#esteira
glBindVertexArray(Vao_esteira) #ativamos o objeto que queremos renderizar
especificaMaterial(0.2, 0.2, 0.2, 0.8, 0.8, 0.8, 0.1, 0.1, 0.1, 32)
transformacaoGenerica(30,1.5,-30,.2,.14,.2,0,90,0)
glDrawArrays(GL_QUADS, 0, len(esteira_indices))
```