

Relatório do Trabalho de Computação Gráfica

Pedro Henrique de Assis Fukuda
Rinaldo Reverte Mendes Junior

RGA: 2016.1904.016-6
RGA: 2015.1904.011-3

Linguagem Utilizada: Python 3.8

Versão Opengl: OpenGL version string: 4.6 (Compatibility Profile) Mesa 20.2.3
GLSL: Version 330 core (3.3).

Sistema Operacional: Manjaro 20.2 Nibia (Linux).

Parâmetros: Seguimos os parâmetros definidos pela descrição do trabalho. Para os axis usamos os parâmetros (-2, 2, -2, 2, -2, 2) que foram comentados no grupo de Telegram.

Bibliotecas Utilizadas

import sys - Utilizada para o método sizeof().
<https://python.readthedocs.io/en/latest/library/sys.html>

import numpy as np - Utilizada para os métodos np.array() e dot().
<https://pypi.org/project/numpy/>

import math - Utilizada para os métodos sen() e cos().
<https://python.readthedocs.io/en/latest/library/math.html>

import pywavefront - Utilizada para ler, extrair informações e tratar o .obj.
<https://pypi.org/project/PyWavefront/>

import cmd - Utilizada para o terminal de leitura de comandos (Classe ComandosTerminal). <https://docs.python.org/3/library/cmd.html>

import pyrr - Utilizada para criar a matrixOrtho (Projection).
https://pyrr.readthedocs.io/en/latest/api_matrix.html

import glm - Utilizada para criar a matrixView (View). <https://pypi.org/project/PyGLM/>

Bibliotecas da Opengl

```
from OpenGL.GL import *  
from OpenGL.GL import shaders  
from OpenGL.GLU import *  
from OpenGL.GLUT import *
```

<https://pypi.org/project/PyOpenGL/>

Bibliotecas para Manipulação de imagens (captureScreen())

```
from PIL import Image  
from PIL import ImageOps
```

<https://pypi.org/project/Pillow/>

Informações Zip

O zip contém uma pasta chamada Trabalho que segue a seguinte hierarquia (Caixa alta representam pastas):

```
TRABALHO
  SHADERS
    axis.fp
    axis.vp
    flat.fp
    flat.vp
    luz.fp
    luz.vp
    none.fp
    none.vp
    phong.fp
    phong.vp
    smooth.fp
    smooth.vp
  Trabalho3.0.py
  cone.mtl
  cone.obj
  torus.mtl
  torus.obj
  cube.mtl
  cube.obj
  sphere.mtl
  sphere.obj
```

Os arquivos .mtl são gerados pelo Blender com informações do material dos objetos.

Implementações Adicionais

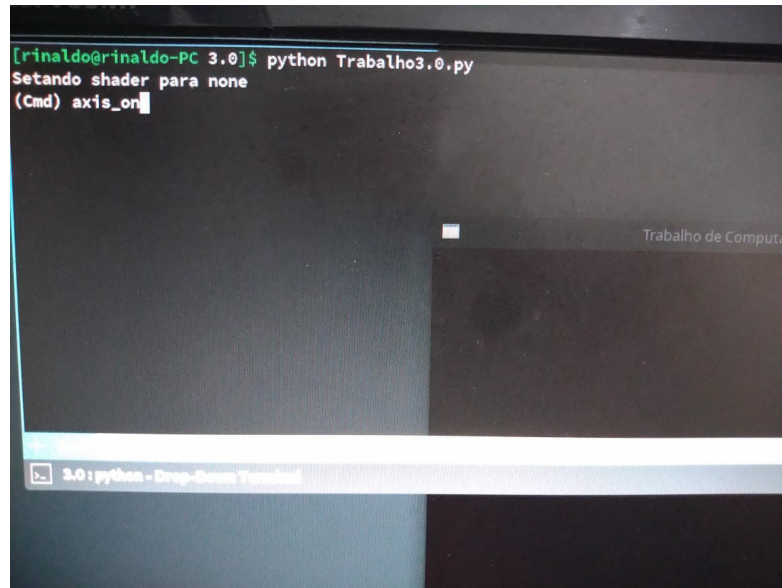
Foi criado o método `setCurrent()` que seleciona um objeto para ser o corrente, e as seguintes transformações funcionam sem a necessidade de passar o parâmetro 'nome': `rotate()`, `scale()`, `translate()`, `shear()` e `color()`.

Para alguns comandos de terminal, uma mensagem customizada é mostrada no terminal dando informações sobre o comando o feito.

Execução

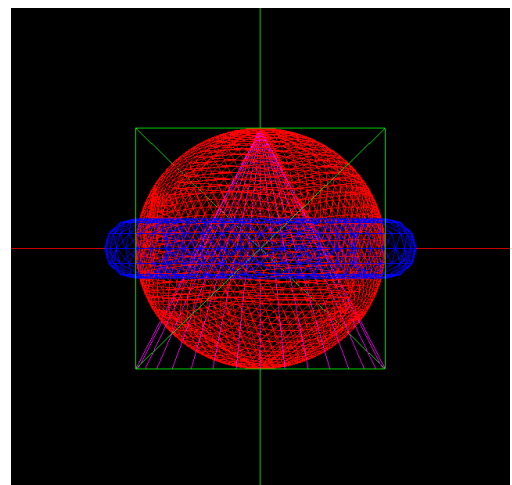
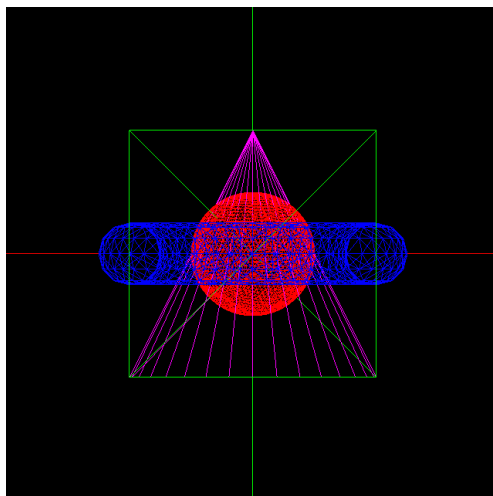
Execução: python Trabalho3.0.py (Em um terminal linux).

Em nosso trabalho para executar comandos de terminal, é necessário selecionar a tela de desenho e apertar a tecla 'm' para abrir o shell, onde os comandos descritos na descrição do trabalho podem ser executados, 1 de cada vez.



Informações Adicionais

O sphere.obj aparentemente possui uma escala diferente da definida na descrição do trabalho (é um pouco menor). Chegamos a esta conclusão ao comparar com o resultado da 'multiple_wire.png' em scripts_testes.



Ao realizar o comando `wire_on` é possível notar que todos .objs desenhados possuem um vértice no centro deles. Não fomos capazes de tratar isso pois a biblioteca `pywavefront` já trata os .objs e fornece as informações de desenho automaticamente.

