Expressão Gráfica para Engenharia Elétrica

2021/2

Professora: Raquel Frizera Vassallo

Aluno: Felipe Pereira Umpierre

Segundo trabalho - Animação 3D em Python

Requisitos

Faça uma animação 3D em Python, usando Numpy e Matplotlib. Os requisitos do trabalho são:

- Animar ao menos dois objetos que não sejam simplesmente uma esfera, um cubo ou objeto simétrico. Os objetos devem ser tais que seja fácil perceber as transformações sendo aplicadas e os resultados da sua movimentação.
- Aplicar as transformações ensinadas em sala se aula. Os objetos devem realizar pelo menos translações e rotações em mais de um eixo. Outras transformações, como mudança de escala e cisalhamento, também podem ser utilizadas.
- Utilizar coordenadas homogêneas para aplicar as transformações aos objetos.
- O arquivo principal do código deve ser feito em Python, usando programação direta (*.py) ou Google Colab (*.ipynb). No caso de se usar a programação direta em Python, o arquivo principal deverá ter o nome de main.py e deverá chamar automaticamente qualquer outro arquivo e funções auxiliares necessárias. Quanlquer arquivo adicional como, por exemplo o arquivo com os pontos dos objetos, deverá ser entregue junto com os códigos.
- Os códigos devem estar devidamente comentados para facilitar o entendimento e correção.

Como usar

Instale as bibliotecas necessárias, listadas no arquivo **dependencias.txt**

```
pip install -r dependencias.txt
```

Execute o arquivo **main.py**:

```
<executavel python> main.py
```

Por exemplo, em ambiente Ubuntu:

```
python3 main.py
```

A animação

A animação consiste em 3 dinossauros, gerados a partir do mesmo modelo 3D STL e plotados em cores diferentes, caminhando pelo "chão" até a queda de um meteoro, que os impulsiona pra longe.

Funcionamento do programa

Criação dos objetos

No começo do programa, os modelos 3D em STL que serão plotados e animados são inicializados informando o nome do arquivo e a cor desejada para o construtor da classe `Objeto`:

```
dinossauro = Objeto('dinossauro.stl', 'seagreen')
```

Nesse construtor, o arquivo será carregado e as propriedades internas como a matriz de pontos em coordenadas homogêneas serão extraídas e calculadas

Figura e eixos

A figura e os eixos do `matplotlib.pyplot` são criados com a função auxiliar `utilidades.criar_eixos` e `utilidades.ajustar_eixos` cuida de configurar os tamanhos e a posição de visualização:

```
figura, eixos = criar_eixos()

ajustar_eixos(eixos, azimute=-45, elevacao=30, tamanho_x=500,
              tamanho_y=500, tamanho_z=1000)
```

Transformações

As transformações são definidas por funções do módulo `transformacoes`, tais como `transformacoes.rotacao_z`, que recebem os parâmetros necessários (por exemplo, o ângulo) e retornam uma matriz para ser multiplicada à matriz do objeto transformado.

A aplicação, de fato, acontece no método `Objeto.transformar`, que recebe quantas matrizes forem necessárias e aplica as transformações em sequência (na matriz de pontos e nos vetores):

```
dinossauro.transformar(translacao(10, 10, 10), rotacao_x(30), rotacao_z(45),
...)
```

Opcionalmente, é possivel mover o objeto temporariamente para a origem antes de transformá-lo usando o parâmetro `origem`:

```
dinossauro.transformar(..., origem=True)
```

Nesse caso, a origem é considerada como a posição em que o objeto é inicialmente criado e calculada acumulando e invertendo as translações aplicadas com o tempo.

Plotagem

A plotagem dos objetos é feita pelo método próprio `Objeto.plotar`, que recebe os eixos do `matplotlib.pyplot` criados e desenha as faces e contornos (a partir dos vetores), na cor especificada inicialmente, e também os pontos da matriz, destacados em vermelho. As funções

auxiliares `utilidades.normalizar_eixos` e `utilidades.corrigir_escala` são usadas para evitar distorções.

Animações

As animações são gerenciadas pelo módulo `animacoes`. Nele, a função `animacoes.animar_objetos` é responsável por aplicar animações frame a frame, recebendo os eixos, a própria função de animação e a lista de objetos envolvidos:

```
animar_objetos(eixos, anim_teste, frames=30, intervalo=0.01,
               objetos=(dinossauro, meteoro))
```

Os parâmetros *frames* e *intervalo* controlam a duração da animação como a quantidade de fragmentos dela e o intervalo entre cada um, respectivamente.

Ainda em `animacoes.animar_objetos` ocorre uma iteração de frames em i (de 0 até frames - 1). Em cada iteração, a plotagem é limpa com `plt.cla`, as propriedades dos eixos são reajustadas com `ajustar_eixos` e objetos envolvidos são transformados pela *função de animação* e depois redesenhados com `Objeto.plotar`.

As funções de animação (todas as outras do módulo), recebem a lista de objetos envolvidos na animação, o frame atual `i` e decidem como transformá-los de acordo com ele:

```
if i < frames/3: # até até 1/3 da animação
    ...
elif i < 2*frames/3: # até 2/3 da animação
    ...
else: # restante
    ...
```