

Computação Gráfica (IME 04-10842)

Trabalho 2

Transformações Afim

Observações:

Este trabalho pode ser feito por uma dupla de alunos.

Como resposta do trabalho, você deve submeter um programa em Python (.py), contendo a função descrita abaixo, indicando os respectivos autores/alunos.

Para o caso de o trabalho ser feito por uma dupla, ambos os alunos devem submeter respostas (idênticas).

As únicas bibliotecas Python permitidas para implementação da função são o Numpy e o Matplotlib.

Enunciado:

- 1) Crie uma função em Python que implemente a animação de um objeto geométrico dentro de uma figura do Matplotlib.
- 2) A função deve ter como único parâmetro uma lista de pontos que representam o objeto, organizada em um *array* Numpy.
Por exemplo, um quadrado unitário centrado nas coordenadas $[4.5, 4.5, 1]^T$ é representado pelo *array*: `po = np.array([[4,4,1],[4,5,1],[5,5,1],[5,4,1]])`.
- 3) Na animação, o objeto geométrico deve girar sobre o seu centro e, simultaneamente, girar sobre a origem do espaço Afim, ou seja, sobre o ponto $[0, 0, 1]^T$.
- 4) Além disso, durante a rotação sobre a origem, o objeto deve ter seu tamanho reduzido à metade, e retornar ao tamanho original ciclicamente.
- 5) O usuário deve poder controlar a velocidade angular de ambas as rotações, usando as seguintes teclas:
 - Seta para cima (\uparrow): aumenta a velocidade da rotação sobre o centro do objeto.
 - Seta para baixo (\downarrow): diminui a velocidade da rotação sobre o centro do objeto.
 - Seta para a esquerda (\leftarrow): aumenta a velocidade da rotação sobre a origem do espaço.
 - Seta para a direita (\rightarrow): diminui a velocidade da rotação sobre a origem do espaço.
- 6) A qualquer momento, o usuário deve poder alterar o tamanho/escala do objeto com os botões do mouse:
 - Botão esquerdo: aumenta o tamanho do objeto em 50%.
 - Botão direito: diminui o tamanho do objeto em 50%.
- 7) Para encerrar a animação, o usuário deve clicar na tecla 'Esc'.
- 8) Os movimentos do objeto devem ser suaves, ou seja, ele não deve mudar de posição abruptamente.
- 9) Para implementar as interações com o usuário, use funções call-back do Matplotlib, como os exemplos no arquivo *call_back.py*, em anexo.