Transformações Geométricas

Transformações Geométricas

- Transformações primárias
  - Translação (T)
  - Escala (S)
  - Rotação (R)
- Transformações secundárias
  - Reflexão (R)
  - Cisalhamento (C)

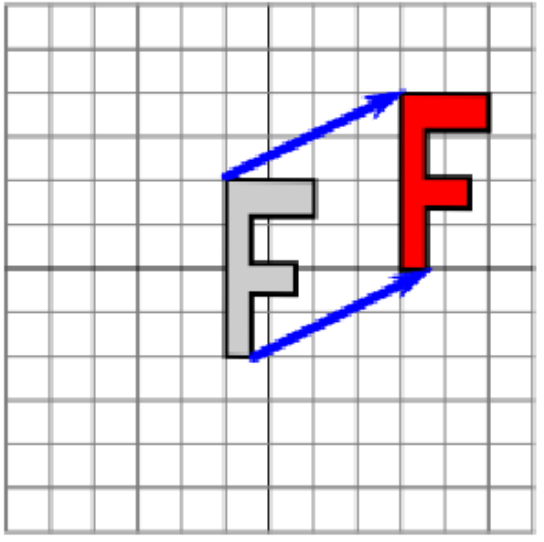
Transformações Geométricas

- Coordenadas Homogêneas:
  - Sistema de coordenadas em geometria projetiva.
  - Um ponto no espaço 2D é uma projeção de um ponto 3D no plano.
- Um ponto 2D em coordenadas homogêneas:
  - Possui três valores: (xh, yh, h).
  - Onde h é um parâmetro homogêneo ($h \neq 0$).
  - Por conveniência, usaremos h = 1:
- Obtemos maior poder de representação.

Translação

- Sejam as coordenadas (x,y,h) e um offset (tx,ty)
- A nova coordenada (x’,y’,h):

$$\begin{bmatrix} x'_h \\ y'_h \\ h \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Matriz de translação}} \begin{bmatrix} x_h \\ y_h \\ h \end{bmatrix}$$



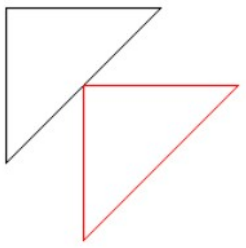
Translação

trinket | Python3 | Run | Share

main.py

```
import numpy as np
from PIL import Image, ImageDraw

def translate(v,tx,ty):
  mt = [
    [1, 0, tx],
    [0, 1, ty],
    [0, 0, 1  ]
  ]
  vt = np.matmul(mt,v)
  return vt.tolist()

im = Image.new('RGB',(400, 300),'white')
draw = ImageDraw.Draw(im)

v1=[150,50,1]
v2=[250,50,1]
v3=[150,150,1]

draw.line((v1[0],v1[1],v2[0],v2[1]),'black')
draw.line((v2[0],v2[1],v3[0],v3[1]),'black')
draw.line((v3[0],v3[1],v1[0],v1[1]),'black')

vt1 = translate(v1,50,50)
vt2 = translate(v2,50,50)
vt3 = translate(v3,50,50)

draw.line((vt1[0],vt1[1],vt2[0],vt2[1]),'red')
draw.line((vt2[0],vt2[1],vt3[0],vt3[1]),'red')
draw.line((vt3[0],vt3[1],vt1[0],vt1[1]),'red')

im.save('a.png')
```

Powered by trinket

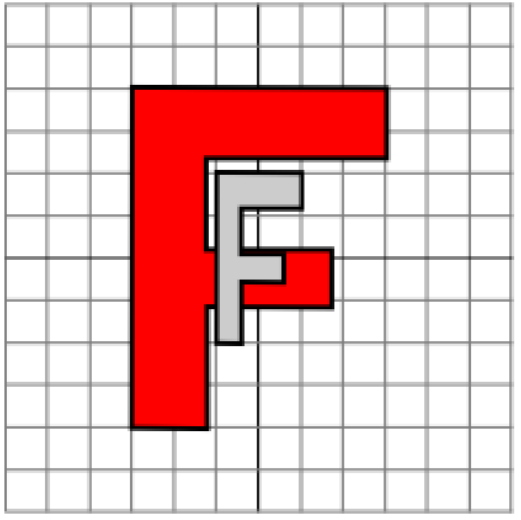


a.png

Escala

- Sejam as coordenadas (x,y,h) e os fatores de escala (sx,sy)
- A nova coordenada (x’,y’,h):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Matriz de escala}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Exercício

- Implemente a função escala e exiba o resultado

trinket Python3 Run Share

main.py

```
1  import numpy as np
2  from PIL import Image, ImageDraw
3
4  def escala(v,sx,sy):
12
13 im = Image.new('RGB',(400, 300),'white')
14 draw = ImageDraw.Draw(im)
15
16 v1=[150,50,1]
17 v2=[250,50,1]
18 v3=[150,150,1]
19
20 draw.line((v1[0],v1[1],v2[0],v2[1]),'black')
21 draw.line((v2[0],v2[1],v3[0],v3[1]),'black')
22 draw.line((v3[0],v3[1],v1[0],v1[1]),'black')
23
24 vt1 = escala(v1,0.5,0.5)
25 vt2 = escala(v2,0.5,0.5)
26 vt3 = escala(v3,0.5,0.5)
27
28 draw.line((vt1[0],vt1[1],vt2[0],vt2[1]),'red')
29 draw.line((vt2[0],vt2[1],vt3[0],vt3[1]),'red')
30 draw.line((vt3[0],vt3[1],vt1[0],vt1[1]),'red')
31
32 im.save('a.png')
```

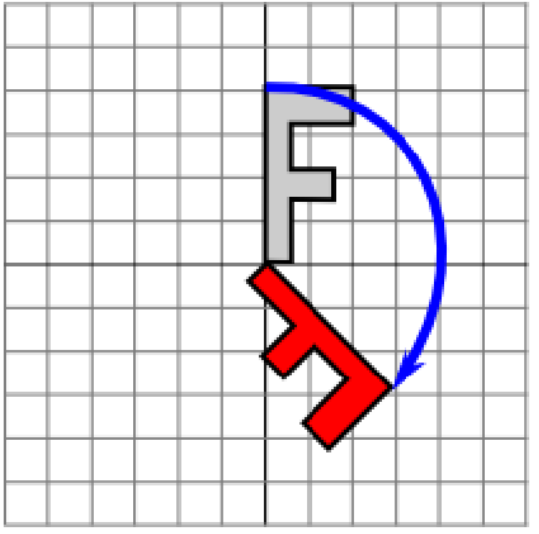
Escala

- sx e sy devem ser maiores que zero.
  - Se sx > 1 e sy > 1 o objeto aumenta.
  - Se sx < 1 e sy < 1 o objeto diminui.
  - Se sx = sy a escala é uniforme.
  - Se sx ≠ sy a escala é diferencial.

Rotação

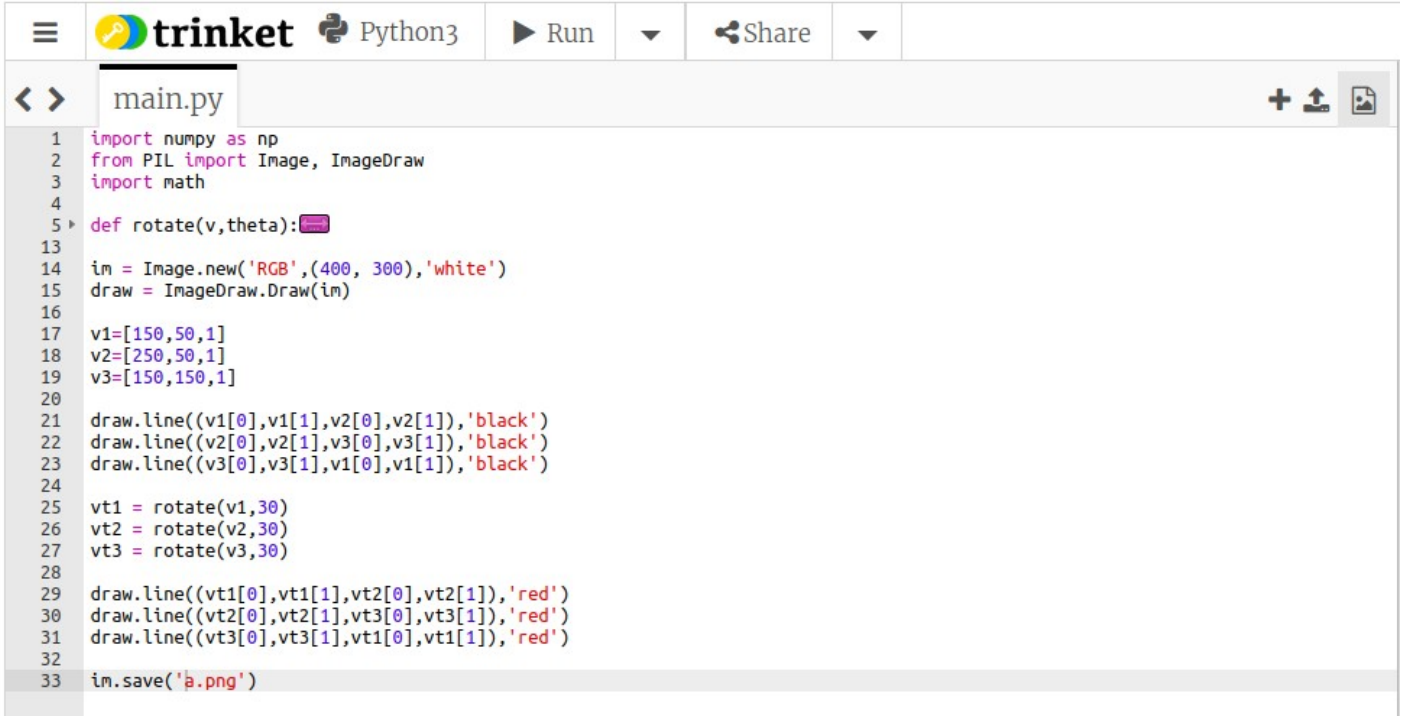
- Sejam as coordenadas (x,y,h) e um ângulo Θ de rotação:
- A nova coordenada (x’,y’,h):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\operatorname{sen}\theta & 0 \\ \operatorname{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Matriz de rotação}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Exercício

- Implemente a função rotate e exiba o resultado:

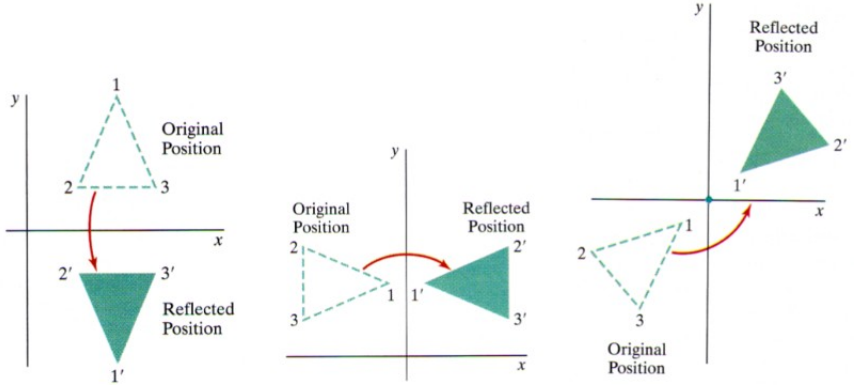



```python
import numpy as np
from PIL import Image, ImageDraw
import math

def rotate(v,theta):

im = Image.new('RGB',(400, 300),'white')
draw = ImageDraw.Draw(im)

v1=[150,50,1]
v2=[250,50,1]
v3=[150,150,1]

draw.line((v1[0],v1[1],v2[0],v2[1]),'black')
draw.line((v2[0],v2[1],v3[0],v3[1]),'black')
draw.line((v3[0],v3[1],v1[0],v1[1]),'black')

vt1 = rotate(v1,30)
vt2 = rotate(v2,30)
vt3 = rotate(v3,30)

draw.line((vt1[0],vt1[1],vt2[0],vt2[1]),'red')
draw.line((vt2[0],vt2[1],vt3[0],vt3[1]),'red')
draw.line((vt3[0],vt3[1],vt1[0],vt1[1]),'red')

im.save('a.png')
```

Trnasformações secundárias

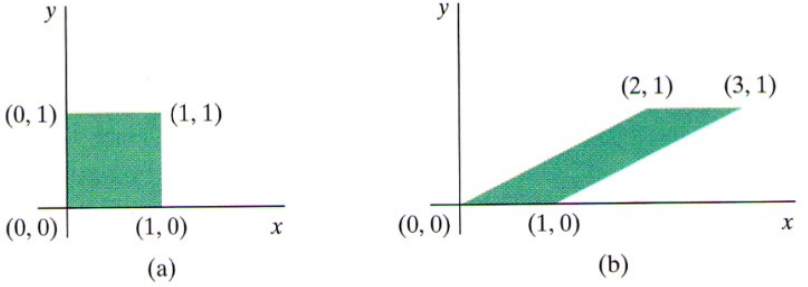
- **Reflexão**
  - Matrizes de reflexão

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- **Cisalhamento**
  - Matriz de cisalhamento

$$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Concatenação

- Transformações arbitrárias
  - Múltiplas transformações
  - Translação, escala, rotação
- Custo de formar uma matriz M = ABCD é menor que o custo de aplicar separadamente cada transformação a todos os vértices

Laboratório

- Aplique transformações sucessivas sobre os vértices do triângulo dos exemplos anteriores até que o resultado final se assemelhe a imagem a seguir: