



Computação
Gráfica

Estrutura do Objeto

Rotação

Redimensionamento

Translação

Cisalhamento

Reflexão

Referências

Fim da Apresentação

Cada objeto é representado por uma classe chamada objeto, que recebe em seu método construtor uma lista de pontos (vértices do objeto) e uma lista de arestas (linhas que compõem o objeto)

```
class objeto:
    def __init__(this, pontos, arestas):
        this.pontos = pontos
        this.arestas = arestas
```

```
'''Meu Quadrado'''
pontoA = [0, 0]
pontoB = [0, 100]
pontoC = [100, 100]
pontoD = [100, 0]
pontos = [pontoA, pontoB, pontoC, pontoD]
arestas = [[pontoA, pontoB], [pontoB, pontoC], [pontoC, pontoD], [pontoD, pontoA]]
quadrado = objeto(pontos, arestas)
```

Estrutura do Objeto

Rotação

Redimensionamento

Translação

Cisalhamento

Reflexão

Referências

Fim da Apresentação

A classe possui duas funções para mostrar o objeto na tela: Uma para mostrar na cor padrão (preto) e outra para mostrar na cor vermelha. O desenho é feito a partir da lista de arestas. Cada aresta é desenhada individualmente.

```
def mostrar(self):  
    for aresta in self.arestas:  
        pygame.draw.lines(screen, BLACK, False, aresta, 1)  
  
def mostrarRED(self):  
    for aresta in self.arestas:  
        pygame.draw.lines(screen, RED, False, aresta, 1)
```

Estrutura do Objeto

Rotação

Redimensionamento

Translação

Cisalhamento

Reflexão

Referências

Fim da Apresentação

A classe objeto possui um método chamado "rotacionar" que recebe uma quantidade de graus como parâmetro e realiza a rotação do objeto em relação à origem da tela.

```
def rotacionar(self, graus):  
    rad = 2 * math.pi * graus/360  
    cos = math.cos(rad)  
    sen = math.sin(rad)  
    for ponto in self.pontos:  
        ponto0 = ponto[0] * cos - ponto[1] * sen  
        ponto1 = ponto[0] * sen + ponto[1] * cos  
        ponto[0] = ponto0  
        ponto[1] = ponto1
```

Estrutura do Objeto

Rotação

Redimensionamento

Translação

Cisalhamento

Reflexão

Referências

Fim da Apresentação

A classe objeto possui três métodos para redimensionamento. Um para o eixo X, outro para o eixo Y e outro em ambos os eixos (mudar escala).

```
def redimensionarX(self, proporcao):  
    for ponto in self.pontos:  
        ponto[0] *= proporcao
```

```
def redimensionarY(self, proporcao):  
    for ponto in self.pontos:  
        ponto[1] *= proporcao
```

```
def redimensionar(self, proporcao):  
    for ponto in self.pontos:  
        ponto[0] *= proporcao  
        ponto[1] *= proporcao
```

Estrutura do Objeto

Rotação

Redimensionamento

Translação

Cisalhamento

Reflexão

Referências

Fim da Apresentação

A classe objeto possui um método para translação. Ele recebe os valores de translação em x e em y e realiza a translação.

```
def transladar(self, x, y):  
    for ponto in self.pontos:  
        ponto[0] += x  
        ponto[1] += y
```

Estrutura do Objeto

Rotação

Redimensionamento

Translação

Cisalhamento

Reflexão

Referências

Fim da Apresentação

A classe objeto possui um método para cisalhamento. Ele recebe os valores de cisalhamento em x e em y e realiza o cisalhamento.

```
def cisalhar(self, x, y):  
    for ponto in self.pontos:  
        ponto0 = ponto[0] + x * ponto[1]  
        ponto1 = ponto[1] + y * ponto[0]  
        ponto[0] = ponto0  
        ponto[1] = ponto1
```

Estrutura do Objeto

Rotação

Redimensionamento

Translação

Cisalhamento

Reflexão

Referências

Fim da Apresentação

A classe objeto possui três métodos para reflexão: Um para refletir no eixo X, outro para refletir no eixo Y, e outro para refletir nos dois eixos. Os valores negativos resultam em uma inversão de lado na tela.

```
def refletirX(self):  
    for ponto in self.pontos:  
        ponto[0] = largura_tela - ponto[0]  
  
def refletirY(self):  
    for ponto in self.pontos:  
        ponto[1] = altura_tela - ponto[1]  
  
def refletir(self):  
    for ponto in self.pontos:  
        ponto[0] = largura_tela - ponto[0]  
        ponto[1] = altura_tela - ponto[1]
```


Computação Gráfica

Estrutura do Objeto

Rotação

Redimensionamento

Translação

Cisalhamento

Reflexão

Referências

Fim da Apresentação

Transformações na Reta

Referências

Slides - Transformações Geométricas em C.G. - Parte 2, Aura Conci
<http://www.ic.uff.br/~aconci/CG-TEMA4-parte2-2020.pdf>

Pygame - Documentação
<https://www.pygame.org/docs/>

Computação Gráfica

Estrutura do Objeto

Rotação

Redimensionamento

Translação

Cisalhamento

Reflexão

Referências

Fim da Apresentação

Transformações na Reta

Aluno

Pedro Henrique Mendes Pereira

Disciplina

Computação Gráfica - 2021.1

Professora

Aura Conci

Curso

Ciência da Computação

UFF – Niterói