



Computação Gráfica

Computação Gráfica

Introdução

Construção do Objeto

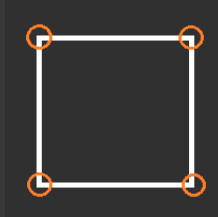
Funções Auxiliares

Execução do Código

Fim da Apresentação

Desenhando em 2D com Pontos

Definir a figura que será criada e mapear seus vértices



Utilizar uma lista de pontos para definir os vértices da imagem

```
pontos = [[0, 0], [0, 10], [10, 10], [10, 0]]
```

Utilizar uma lista de elementos que representam dois pontos que, quando ligados por uma linha reta formam uma aresta

```
arestas = [[[0, 0], [0, 10]], [[0, 10], [10, 10]], [[10, 10], [10, 0]], [[10, 0], [0, 0]]]
```

Computação Gráfica

Introdução

Construção do Objeto

Funções Auxiliares

Execução do Código

Fim da Apresentação

Desenhando em 2D com Pontos

Recebe a lista de pontos e a lista de pares de pontos (que representam as arestas)

Define o maior valor de x e o maior valor de y entre os pontos

Chama a função responsável por criar uma matriz que comporte os pontos

Chama a função responsável por desenhar as arestas e os pontos

```
def criar_objeto(pontos, arestas):  
    maior_x = 0  
    maior_y = 0  
    for i in range(len(pontos)):  
        if pontos[i][0] > maior_x:  
            maior_x = pontos[i][0]  
        if pontos[i][1] > maior_y:  
            maior_y = pontos[i][1]  
  
    retangulo = gerar_matriz(maior_x + 1, maior_y + 1)  
    retangulo = desenhar_linhas(arestas, retangulo)  
    return retangulo
```

Computação Gráfica

Introdução

Construção do Objeto

Funções Auxiliares

Execução do Código

Fim da Apresentação

Desenhando em 2D com Pontos

Funções simples de gerar matriz e Função de desenhar linhas (apenas perpendiculares, e paralelas aos eixos, por enquanto)

```
def gerar_matriz(largura, altura):  
    matriz = []  
    for i in range(largura):  
        coluna = []  
        for i in range(altura):  
            coluna.append(False)  
        matriz.append(coluna)  
    return matriz
```

```
def desenhar_linhas(arestas, retangulo):  
    for i in range(len(arestas)):  
        if arestas[i][0][0] == arestas[i][1][0] and arestas[i][0][1] != arestas[i][1][1]:  
            for j in range(abs(arestas[i][0][1] - arestas[i][1][1]) + 1):  
                if arestas[i][0][1] <= arestas[i][1][1]:  
                    retangulo[arestas[i][0][0]][arestas[i][0][1] + j - 1] = True  
                elif arestas[i][0][1] >= arestas[i][1][1]:  
                    retangulo[arestas[i][1][0]][arestas[i][1][1] + j - 1] = True  
        elif arestas[i][0][1] == arestas[i][1][1] and arestas[i][0][0] != arestas[i][1][0]:  
            for j in range(abs(arestas[i][0][0] - arestas[i][1][0]) + 1):  
                if arestas[i][0][0] <= arestas[i][1][0]:  
                    retangulo[arestas[i][0][0] + j - 1][arestas[i][0][1]] = True  
                elif arestas[i][0][0] >= arestas[i][1][0]:  
                    retangulo[arestas[i][1][0] + j - 1][arestas[i][0][1]] = True  
    return retangulo
```

Computação Gráfica

Introdução

Construção do Objeto

Funções Auxiliares

Execução do Código

Fim da Apresentação

Desenhando em 2D com Pontos

Função responsável por mostrar o objeto na tela (provisória)

```
def mostrar_objeto(objeto):
    for i in range(len(objeto)):
        for j in range(len(objeto[0])):
            if objeto[i][j] == False:
                print(" ", end="")
            else:
                print("# ", end="")
        print("")
```

Criação e exibição do objeto

```
objeto = criar_objeto(pontos, arestas)
mostrar_objeto(objeto)
```

Vizualizaçãodoobjeto deacordocomasdiretrizes
definidas pela função "mostrar_objeto"

[illegible]

Computação Gráfica

Introdução

Construção do Objeto

Funções Auxiliares

Execução do Código

Fim da Apresentação

Desenhando em 2D com Pontos

Aluno

Pedro Henrique Mendes Pereira

Disciplina

Computação Gráfica - 2021.1

Professora

Aura Conci

Curso

Ciência da Computação

UFF - Niterói