



Computação
Gráfica

Cálculo das Vistas

Métodos Auxiliares

Objeto Observador

Visualização

Refêrências

Fim da Apresentação

O método "mostrarFacesPrincipais" recebe o próprio objeto como parâmetro e um ponto chamado "pontoObservador", que representa o observador do objeto.

O cálculo das faces visíveis é realizado utilizando o método *Back-face culling*. O objeto é mostrado centralizado através da perspectiva isométrica.

```
def mostrarFacesPrincipais(self, pontoObservador):
    obj = self
    facesVisiveis = []
    for face in obj.faces:
        pontoMedioFace = self.getCentroFace(face)
        vetorObservador = [pontoObservador.x - pontoMedioFace.x, pontoObservador.y - pontoMedioFace.y, pontoObservador.z - pontoMedioFace.z]
        vetorAB = [face.vertices[0].x - face.vertices[1].x, face.vertices[0].y - face.vertices[1].y, face.vertices[0].z - face.vertices[1].z]
        vetorAC = [face.vertices[0].x - face.vertices[2].x, face.vertices[0].y - face.vertices[2].y, face.vertices[0].z - face.vertices[2].z]
        vetorNormal = [vetorAB[1]*vetorAC[2] - vetorAC[1]*vetorAB[2], vetorAB[2]*vetorAC[0] - vetorAC[2]*vetorAB[0], vetorAB[0]*vetorAC[1] - vetorAC[0]*vetorAB[1]]
        print(vetorNormal, vetorObservador)
        produtoEscalar = vetorObservador[0]*vetorNormal[0] + vetorObservador[1]*vetorNormal[1] + vetorObservador[2]*vetorNormal[2]
        if produtoEscalar > 0:
            facesVisiveis.append(face)

    obj.girarY(45)
    obj.girarX(35.26)
    obj.transladar(590, 260, 0)

    for face in facesVisiveis:
        for aresta in face.arestas:
            pygame.draw.lines(screen, BLACK, False, aresta.get2D(), 1)
    obj.transladar(-590, -260, 0)
    obj.girarX(-35.26)
    obj.girarY(-45)
```

Alguns métodos auxiliares são utilizados para o cálculo das faces visíveis, bem como para a representação gráfica do objeto e do observador em 2D.

```
def mostrar2D(self):  
    for aresta in self.arestas:  
        pygame.draw.lines(screen, BLACK, False, aresta.get2D(), 1)  
  
def getCentro(self):  
    somaX = 0  
    somaY = 0  
    somaZ = 0  
    quantidadePontos = len(self.vertices)  
    for p in self.vertices:  
        somaX += p.x  
        somaY += p.y  
        somaZ += p.z  
    return ponto(somaX / quantidadePontos, somaY / quantidadePontos, somaZ / quantidadePontos)  
  
def getCentroFace(self, face):  
    somaX = 0  
    somaY = 0  
    somaZ = 0  
    quantidadePontos = len(face.vertices)  
    for p in face.vertices:  
        somaX += p.x  
        somaY += p.y  
        somaZ += p.z  
    return ponto(somaX / quantidadePontos, somaY / quantidadePontos, somaZ / quantidadePontos)
```

Cálculo das Vistas

Métodos Auxiliares

Objeto Observador

Visualização

Refêrências

Fim da Apresentação

O objeto observador é definido apenas para representação, mas o ponto do observador, que é utilizado para cálculos em si, é o ponto chamado de "obs" na imagem abaixo.

```
obs = ponto(1280, 720, 500)
pontosObs = [ponto(obs.x - 5, obs.y - 5, obs.z), ponto(obs.x + 5, obs.y - 5, obs.z), ponto(obs.x - 5, obs.y + 5, obs.z), ponto(obs.x + 5, obs.y + 5, obs.z)]
arestasObs = [aresta(pontosObs[0], pontosObs[1]), aresta(pontosObs[0], pontosObs[2]), aresta(pontosObs[2], pontosObs[3]), aresta(pontosObs[1], pontosObs[3])]
objetoObservador = objeto(pontosObs, arestasObs, [])
```

Cálculo das Vistas

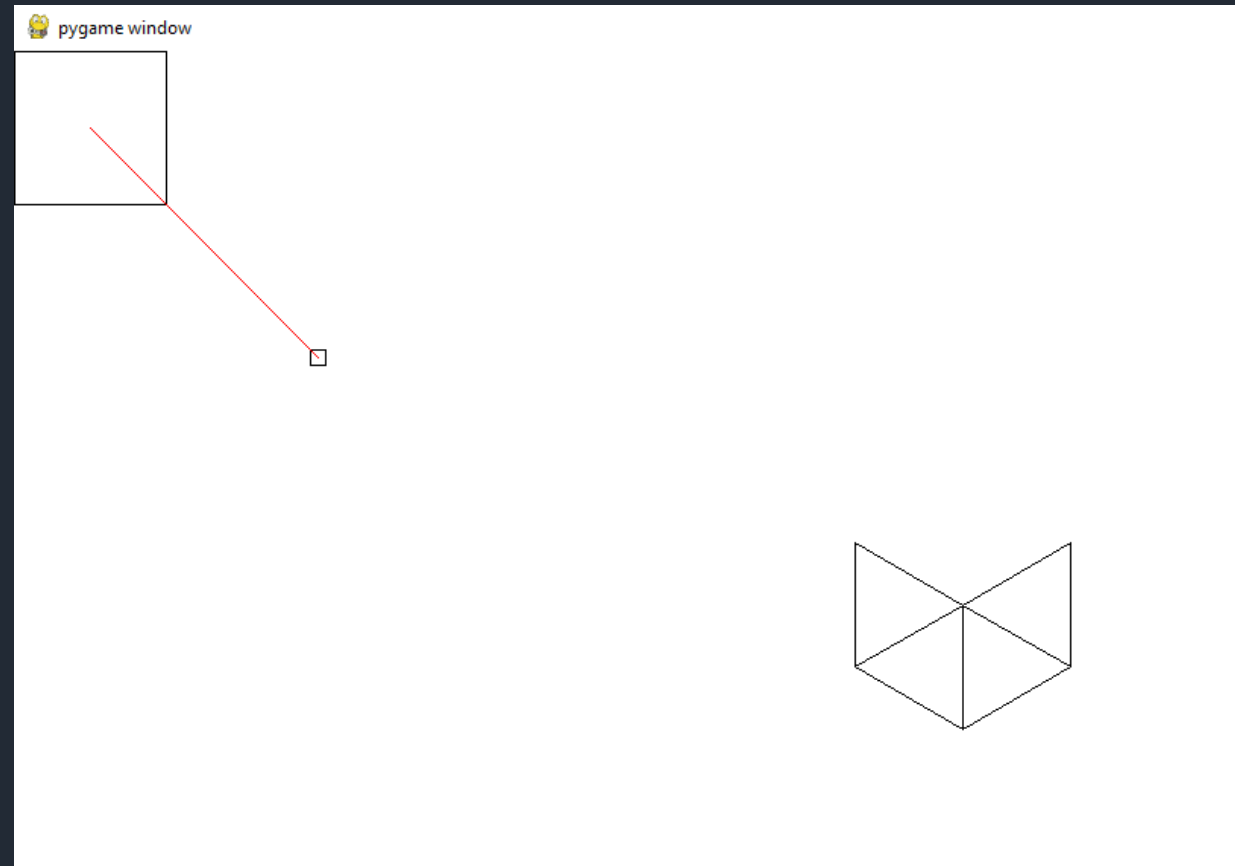
Métodos Auxiliares

Objeto Observador

Visualização

Refêrências

Fim da Apresentação



A visualização é dada por duas vistas:

Uma vista 2D, com o observador em relação ao centro do objeto, com vista para a face central do objeto.

Uma vista isométrica, mostrando as faces visíveis do objeto.

Computação Gráfica

Objeto 3D - Faces Visíveis

Cálculo das Vistas

Métodos Auxiliares

Objeto Observador

Visualização

Refêrências

Fim da Apresentação

Referências

Livro - Teoria da Computação Gráfica 1a Edição, Aura Conci

https://drive.google.com/file/d/15VeqNBTyThiO-OSWQ2N_6iP9DAgWvUej/view?usp=sharing

Computação Gráfica

Objeto 3D - Faces Visíveis

Cálculo das Vistas

Métodos Auxiliares

Objeto Observador

Visualização

Refêrências

Fim da Apresentação

Aluno

Pedro Henrique Mendes Pereira

Disciplina

Computação Gráfica - 2021.1

Professora

Aura Conci

Curso

Ciência da Computação

UFF – Niterói