

Computação Gráfica - PL8

Prof. S.M. Jesus

Thomaz Rodrigues Lima - a90985

p8-1.py: Antes da modificação os triângulos estavam da seguinte forma:

```
position_data = [p0, p3, p1, # Triângulo 1  
                 p1, p3, p4, # Triângulo 2  
                 p1, p4, p2] # Triângulo 3
```

Visualmente, essa estrutura forma três triângulos conectados, onde p0 e p2 estão nas extremidades superiores, p3 e p4 estão na parte inferior, e p1 é um ponto central.

Agora, após a troca de p0 e p2 na lista position_data:

```
position_data = [p2, p3, p1, # Triângulo 1  
                 p1, p3, p4, # Triângulo 2  
                 p1, p4, p0] # Triângulo 3
```

No caso, só o Triângulo 2 irá aparecer, pois p2,p3,p1 formam uma linha, e p1,p4,p0 também.

p8-3.py: A cor na superfície da esfera é determinada pela fórmula **vec3 color = mod(position, 1.0);** no fragment shader. A **função mod(position, 1.0)** calcula o resto da divisão das coordenadas da posição dos vértices por 1.0, gerando valores positivos até 1. Isso cria um gradiente de cores na esfera, que muda conforme a posição dos pontos. Ou seja, mesmo que haja uma rotação, a priori, a posição dos pontos é a mesma e **NÃO HÁ** mudança de cor.

p8-4.py: A variável **uniform time** é usada para criar uma animação de variação de cores e ondulação na superfície da esfera. Ela altera a posição dos vértices no Vertex Shader e a cor no Fragment Shader, com a finalidade de gerar efeitos baseados no tempo.