Exercício 1: Mude a cor de fundo para branco

Mudança que Fiz:

- Na função init(), alterei a cor de fundo de azul para branco.
- Original: glClearColor(0, 0, 1, 1) (azul: R=0, G=0, B=1, A=1)
- Modificado: glClearColor(1, 1, 1, 1) (branco: R=1, G=1, B=1, A=1)

Exercício 2: Mude a rotação do eixo Y para o eixo X e veja o que acontece

Mudança que Fiz:

- Na função draw(), troquei o eixo de rotação de Y para X.
- Original: glRotatef(r, 0, 1, 0) (rotaciona no eixo Y)
- Modificado: glRotatef(r, 1, 0, 0) (rotaciona no eixo X)

Exercício 3: Agora mude a rotação do eixo Y para o eixo X e Y e veja o que acontece

Mudança que Fiz:

- Na função draw(), ajustei a rotação para ocorrer nos eixos X e Y ao mesmo tempo.
- Original: glRotatef(r, 0, 1, 0) (eixo Y)
- Modificado: glRotatef(r, 1, 1, 0) (eixo combinado X e Y)

Exercício 4: Mude a cor do triângulo para preto

Mudança que Fiz:

- Na função draw(), mudei a cor do triângulo de amarelo para preto.
- Original: glColor3f(1, 1, 0) (amarelo: R=1, G=1, B=0)
- Modificado: glColor3f(0, 0, 0) (preto: R=0, G=0, B=0)

Exercício 5: Altere os vértices X, Y para um número maior e teste o triângulo

Mudança que Fiz:

- Na função draw(), aumentei os valores dos vértices do triângulo.

- Original:
    - glVertex3f(0, 1, 0) (topo)
    - glVertex3f(-1, -1, 0) (inferior esquerdo)
    - glVertex3f(1, -1, 0) (inferior direito)
- Modificado (multipliquei por 2):
    - glVertex3f(0, 2, 0) (topo)
    - glVertex3f(-2, -2, 0) (inferior esquerdo)
    - glVertex3f(2, -2, 0) (inferior direito)

Exercício 6: Atualize o ângulo de rotação para transações para o lado esquerdo ou no sentido horário. (OBS: no código original ele gira anti-horário). O que precisou ser alterado?

Mudança que Fiz:

- Na função main(), inverti a direção da rotação.
- Original: r += 3 (aumenta o ângulo, gira anti-horário)
- Modificado: r -= 3 (diminui o ângulo, gira horário)

O que Alterei: Eu mudei o sinal de r += 3 para r -= 3. Antes, o triângulo girava no sentido anti-horário, mas agora ele gira no sentido horário, como se estivesse indo para o lado oposto.

Exercício 7: Altere a posição inicial do triângulo. Atualmente, ele inicia em x=-1.5 e y=0. Modifique para que ele comece centralizado (x=0, y=0). O que acontece com a exibição ao iniciar?

Mudança que Fiz:

- No início do código, ajustei as variáveis de posição inicial.
- Original: x = -1.5, y = 0
- Modificado: x = 0, y = 0

O que Aconteceu ao Iniciar: Eu coloquei o triângulo no centro da tela, e ele apareceu bem no meio logo que o programa começou, em vez de estar mais para a esquerda como antes.

Exercício 8: Mude a escala inicial do triângulo. No código original, ex=1, ey=1, ez=1. Altere para ex=2, ey=2, ez=2. Como a mudança da escala afeta a exibição do triângulo?

Mudança que Fiz:

- No início do código, alterei as variáveis de escala.
- Original: ex = 1, ey = 1, ez = 1
- Modificado: ex = 2, ey = 2, ez = 2

Como a Mudança Afetou: Eu aumentei a escala, e o triângulo apareceu duas vezes maior desde o início, tanto na largura quanto na altura. O ez = 2 não mudou nada visível porque o triângulo é 2D.

Exercício 9: Modifique a movimentação do triângulo. No código original, pressionar A move para a esquerda e D move para a direita. Inverta os controles para que A mova para a direita e D mova para a esquerda. Explique o que foi alterado no código para isso acontecer.

Mudança que Fiz:

- Na função main(), troquei as direções das teclas A e D.
- Original:

```
if event.key == K_a:
        x += -0.2 # Esquerda
if event.key == K_d:
        x += 0.2 # Direita
```

- Modificado:

```
if event.key == K_a:
        x += 0.2 # Direita
if event.key == K_d:
        x += -0.2 # Esquerda
```

Explicação do que Alterei: Eu inverti os sinais: antes, 'A' diminuía o x (movia para a esquerda), e 'D' aumentava (movia para a direita). Agora, 'A' aumenta o x (vai para a direita), e 'D' diminui (vai para a esquerda). Troquei os valores de incremento para o oposto.

Exercício 10: Adicione um controle de zoom com as teclas 'Z' e 'X'. O objetivo deste exercício é permitir que o usuário aproxime e afaste o triângulo usando as teclas: 'Z' para aproximar (trazendo o triângulo para

frente no eixo Z), 'X' para afastar (empurrando o triângulo para trás no eixo Z).

Mudanças que Fiz:

1. **Adicionei uma variável para a posição Z:**
    - No início do código: z_pos = -6
2. **Declarei z_pos como global:**
    - Na função main(): global x, y, r, ex, ey, ez, z_pos
3. **Atualizei a translação em draw():**
    - Original: glTranslatef(x, y, -6)
    - Modificado: glTranslatef(x, y, z_pos)
4. **Adicionei controles de zoom em main():**

```
if event.key == K_z:
        z_pos += 0.2 # Aproxima
if event.key == K_x:
        z_pos -= 0.2 # Afasta
```

Explicação: Eu criei a variável z_pos para controlar a posição no eixo Z, começando em -6. Usei ela na translação para substituir o valor fixo. Com 'Z', eu aumento o z_pos, trazendo o triângulo mais perto (ele fica maior), e com 'X', eu diminuo, afastando-o (ele fica menor).