

Mude a cor de fundo para branco

```
glClearColor(1, 1, 1, 1) # Define a cor de fundo azul (R=0, G=0, B=1, A=1)
```

trocando os valores do `glClearColor` podemos trocar a cor do fundo, colocando todos os valores em 1 para ter um fundo branco.

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

```
glRotatef(r, 1, 0, 0)
```

alterado o valor de 0,1,0 para 1,0,0, para que a rotação seja realizada no eixo X em vez de Y

Agora muda a rotação do eixo Y para o eixo X e Y e veja o que acontece

```
glRotatef(r, 1, 1, 0)
```

trocando os valores para 1,1,0 podemos realizar a rotação do eixo X e Y ao mesmo tempo

Mude a cor do triângulo para preto

```
glColor3f(0, 0, 0) # Define a cor do triângulo como amarelo (R=1, G=1, B=0)
```

trocando os valores para 0,0,0 podemos definir a cor do triângulo como preto

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

```
glVertex3f(3, 4, 0) # Define o vértice superior
```

```
glVertex3f(2, 2, 0) # Define o vértice inferior esquerdo
```

```
glVertex3f(4, 2, 0) # Define o vértice inferior direito
```

alterado os valores de X e Y

Atualize o ângulo de rotação para girar mais rápido para o lado esquerdo

Ou no sentido horário. (OBS: no código original ele gira anti-horário). O que precisou ser alterado?

```
r -= 15
```

o valor de r foi alterado de 3 para 15 para aumentar a velocidade do giro, e o sinal de positivo para negativo para alterar o sentido de anti-horário para horário

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?

ao alterar os valores para 0,0 o triângulo inicia centralizado

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?

```
ex = 2 # Escala no eixo X
```

```
ey = 2 # Escala no eixo Y
```

```
ez = 2 # Escala no eixo Z (não afeta 2D)
```

o triângulo aumenta de tamanho

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.

```
if event.key == K_d: # Se a tecla A for pressionada
```

```
    x += -0.2 # Move o triângulo para a esquerda
```

```
if event.key == K_a: # Se a tecla D for pressionada
```

```
    x += 0.2 # Move o triângulo para a direita
```

foi trocado o K.a para K.d e o K.d para K.a para inverter a movimentação do triângulo

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).

```
if event.key == K.x: # Scroll para baixo
```

```
    ex -= 0.2 # Diminui a escala do triângulo no eixo X
```

```
    ey -= 0.2 # Diminui a escala do triângulo no eixo Y
```

```
    ez -= 0.2 # Diminui a escala do triângulo no eixo Z (não afeta 2D)
```

```
if event.key == K.z: # Scroll para cima
```

```
    ex += 0.2 # Aumenta a escala do triângulo no eixo X
```

```
    ey += 0.2 # Aumenta a escala do triângulo no eixo Y
```

```
    ez += 0.2 # Aumenta a escala do triângulo no eixo Z (não afeta 2D)
```