

Exercício sobre Shaders

Implementar programa estilo screen saver

Baseado no código de exemplo disponibilizado pelo professor (`exemplo_03.c`), altere a matriz de transformação para retirar a parcela da rotação e manter apenas a parcela de translação. Calcule a movimentação (`dx` e `dy`, posições 12 e 13 da matriz) tal que, enquanto o triângulo não colide com as bordas da janela, a sua posição em `x` e `y` é modificada pela atualização da matriz de translação acumulando a direção de deslocamento atual ($\cos(\alpha)$ e $\sin(\alpha)$). Quando o triângulo colide com algum dos seus vértices com as bordas da janela, ele deve ter sua direção alterada, para a direção refletida a partir da direção anterior e do ponto de colisão. Utilize a fórmula ao lado para calcular a reflexão da direção.

Converter ângulo para radianos:
 $\text{rad} = 60 / 180 * \pi$

$$R_{\text{out}} = 2N(N.R_{\text{in}}) - R_{\text{in}}$$

$$\begin{aligned}\theta &= 60^\circ \\ x &= \cos(\theta) \\ y &= \sin(\theta) \\ x &= \cos(\text{rad}) = 0.5 \\ y &= \sin(\text{rad}) = 0.87\end{aligned}$$

Onde: N é a normal no ponto de colisão e R_{in} representa o vetor da direção de deslocamento atual.

Dicas:

- A normal da borda pode ser definida com os pontos das extremidades dos eixos e a origem: $(-1, 0)$, $(0, 1)$, ...
- $N.R_{\text{in}}$ é o produto escalar entre os vetores N e R_{in} . Sendo que R_{in} deve ser o vetor invertido ao da direção!
- Três pontos importantes a serem resolvidos no código: detecção da colisão com as bordas, implementação da reflexão e a alteração da matriz de translação com a progressão do triângulo na direção em que ele está no momento.
 - Para colisão, teste a posição dos vértices do triângulo + translação, contra os limites das bordas.
 - Para reflexão, implementar a equação acima.
 - Para translação, dada uma direção em ângulo, calcular o deslocamento com $\cos(\alpha)$ e $\sin(\alpha)$.

