

Aula Prática Laboratorial n.º 2

Sumário

Template de programa C/OpenGL. Projectos “polígono” e “relógio”. Tutorial Nate Robins “*shapes*”.

Enunciado

Projecto “polígono”

1. Crie um projecto com base no *template* de programa que foi disponibilizado.
2. Analise o referido *template* e observe atentamente o seu funcionamento.

Projecto “relógio”

1. Crie uma função `void poligono(GLint n, GLfloat x0, GLfloat y0, GLfloat r)` que desenhe um polígono regular de n lados, com o centro geométrico no ponto de coordenadas x_0, y_0 e distância aos vértices r . Use o código criado na aula anterior acrescentando o bloco `glBegin() / glEnd()`.
2. Use a referida função para desenhar um círculo na janela principal da aplicação. Este irá servir de modelo ao mostrador do relógio analógico que se pretende construir.
3. Na função `key()` configure duas teclas para aumentar e diminuir o número de lados do polígono.
4. Crie a função `void mostrador()`. Use as funções do OpenGL para modelar, com base no desenho de pequenos segmentos de recta, as doze marcas principais do mostrador do relógio. Use segmentos um pouco maiores para as marcas correspondentes às 3, 6, 9 e 12 horas.

5. Use as funções do OpenGL para modelar, com base no desenho de segmentos de recta de diferentes espessuras, os três ponteiros do relógio: segundos, minutos e horas. Para cada ponteiro, recorra às equações paramétricas da circunferência para calcular as coordenadas de um dos vértices. O outro vértice será sempre coincidente com o centro do mostrador.
6. Na função `main()`, remova o indicador de comentário (“//”) existente no início da linha `//glutTimerFunc(estado.delay, Timer, 0);`
7. Use a função de temporização do GLUT para construir o mecanismo interno do relógio. A função `Timer()` deverá alterar apenas os valores das variáveis globais, encarregando-se o *callback* de desenho de desenhar o relógio em conformidade com os novos valores.

Tutorial Nate Robins “*shapes*”

Execute o tutorial “*shapes*” de Nate Robins.

Observações

1. As equações paramétricas da circunferência são as seguintes:

$$\begin{cases} x = r * \cos(t) + x_0 \\ y = r * \sin(t) + y_0 \end{cases}$$

em que:

(x_0, y_0) são as coordenadas do centro da circunferência;

r é o raio da circunferência;

$0 \leq t < 360^\circ$.

2. A correspondência entre diferentes unidades de medida de ângulos é a seguinte:

$$180^\circ = \pi \text{ radianos}$$

3. Ao proceder à animação tenha em atenção dois aspectos importantes:

- a) a marca das 12 horas está localizada na posição do mostrador correspondente a um ângulo de 90° (e não 0°);
- b) os ponteiros do relógio movem-se em sentido retrógrado (e não em sentido directo).

Funções

- **`void glPointSize(GLfloat size);`**

Especifica a dimensão em *pixels* dos pontos desenhados. Esta função deverá ser colocada fora do bloco `glBegin()` / `glEnd()`. O valor inicial é 1.0.

- **`void glLineWidth(GLfloat width);`**

Especifica a espessura em *pixels* das linhas desenhadas. Esta função deverá ser colocada fora do bloco `glBegin()` / `glEnd()`. O valor inicial é 1.0.