Multimédia I - Licenciatura em Engenharia Informática
2021/2022

Ficha Prática n.º 2

Importante:

A ficha deve ser realizada em **grupo (2-3 alunos)**, seguindo as seguintes normas:

- A constituição dos grupos deve ser a mesma da ficha anterior;
- A submissão da ficha prática deve ser através do **Canvas**, em *assignment* próprio, e deverá consistir num ficheiro **ZIP** com uma pasta contendo o **código-fonte**.
- A data-limite para entrega da ficha prática é **24 de outubro de 2021 às 23:55**;
- A apresentação da ficha prática decorrerá nos dias 21 e 26 de outubro, no horário das aulas práticas;
- No dia da apresentação, **TODOS** os elementos do grupo deverão estar presentes. Os elementos ausentes serão classificados com 0 valores;
- A apresentação e discussão poderá ser realizada individualmente.

1. Projeto Relógio

1.1. **(0.5 valores)** Crie uma cópia de um projeto das aulas anteriores, onde já tenha o Visual Studio Code configurado corretamente. Deste modo, não é necessário configurar novamente o IDE.

1.2. **(0.5 valores)** Abra a pasta do novo projeto no Visual Studio Code e apague todos os ficheiros excepto a pasta *.vscode*

1.3. **(0.5 valores)** Faça download do ficheiro "template_ficha02.zip" e coloque o conteúdo na pasta do novo projeto. Deverá ficar somente com um ficheiro *"template_ficha02.c"*.

1.4. **(0.5 valores)** Renomeie o ficheiro *"template_ficha02"* para *"relogio.c"*

1.5. **(2.5 valores)** Crie uma função genérica para criar um polígono:

```
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 *x0*, *y0* e distância aos vértices *r*.

1.6. **(2.0 valores)** Use a função da alínea anterior para desenhar um círculo na janela principal da aplicação. Este irá servir de modelo ao mostrador do relógio analógico que se pretende construir.

1.7. **(2.5 valores)** Crie a função `void mostrador()`

Use as funções do OpenGL para modelar, com base no desenho de pequenos segmentos de recta, as 60 marcas principais do mostrador do relógio, correspondentes aos minutos. Use segmentos um pouco maiores para as marcas correspondentes às horas.

1.8. **(2.5 valores)** Crie a função `void ponteiros()`

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. Use as equações paramétricas da circunferência para localizar um dos vértices dos ponteiros (o outro vértice é sempre coincidente com o centro do mostrador).

1.9. **(6.0 valores)** Use a função de temporização do GLUT para construir o mecanismo interno do relógio. A função *timer* só altera as variáveis globais, o *callback* de desenho é que se encarrega de desenhar o relógio segundo os novos valores.

1.10. **(2.5 valores)** Adicione duas teclas para permitir aumentar e diminuir, respetivamente, a velocidade do timer (*delay*).

Observações

1. As equações paramétricas da circunferência são as seguintes:

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

em que:

(x_c, y_c) são as coordenadas do centro da circunferência;

r é o raio da circunferência;

$0 \leq t < 2\pi$ *radianos*.

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

$180^o = \pi$ 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 90º (e não 0º);
    b. Os ponteiros do relógio movem-se em sentido retrógrado (e não em sentido direto).

4. Funções:

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

Especifica a grossura das linhas desenhadas.
Esta instrução tem que ser colocada fora do bloco *glBegin/glEnd*.
O valor inicial é 1, sendo afetado pela opção:
glEnable/glDisable(GL_LINE_SMOOTH)

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

Especifica a diâmetro dos pontos.
Esta instrução tem que ser colocada fora do bloco *glBegin/glEnd*.
O valor inicial é 1, sendo afetado pela opção:
glEnable/glDisable(GL_POINT_SMOOTH)