


1. Projeto Relógio

1.1. 


1.2. 

1.3. 


1.4. 

1.5.  função na linha 271 do código, vê o comentário (estruturas e funções comentadas no glm.h também)

(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.  não avancei com esta por estar dependente da anterior

(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.  usei a função que tínhamos de fichas de anteriores


(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.  usei a função que tínhamos de fichas de anteriores

(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 e 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`).