

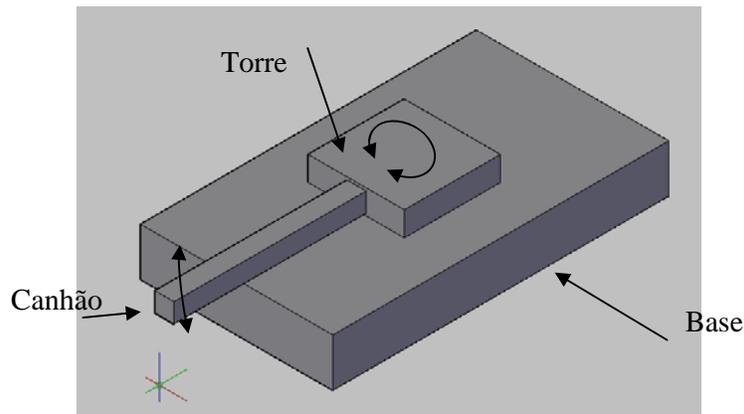
Aula Prática Laboratorial n.º 9

Sumário

Tutorial Nate Robins “*projection*”. Programa C/OpenGL “tanque” baseado no template “*Aula9_template.cpp*”

Tutorial Nate Robins “*projection*”

Execute o tutorial “*projection*” de Nate Robins e veja os projectos “*tanque*” e “*chaimite*” a funcionar.



Projecto “tanque”

1. Crie um projecto do tipo consola com base no “*Aula8_template.cpp*” que foi disponibilizado;
2. Crie a função *desenhaTanque(Tanque t)*, que desenha o tanque com a base centrada na posição $t.x, t.y, 0$ (use a função *cubo()* da aula anterior e a instrução *glScalef(...)* para desenhar os paralelepípedos);

As medidas estão definidas em macros no início do programa:

```
#define RAI0_ROTACAO 20

#define LARGURA_BASE 4
#define COMPRIMENTO_BASE 7
#define ALTURA_BASE 1

#define LARGURA_TORRE 2
#define COMPRIMENTO_TORRE 2
#define ALTURA_TORRE 0.5

#define COMPRIMENTO_CANHAO 4
#define RAI0_CANHAO 0.2
```

3. Alterar a função *tanque* para fazer a rotação da Torre e do Canhão usando os ângulos *t.angTorre* e *t.angCanhao*;
4. Acrescentar teclas para rodar a torre e o canhão;

5. Ponha o tanque a rodar em torno do centro com um raio `RAIO_ROTACAO`;

Projecto “chaimite”

1. Crie um novo projecto do tipo consola baseado no tanque;
2. Acrescente rodas ao tanque com a instrução `gluCylinder(...)` (não se esqueça de o levantar o chão);
3. Crie teclas para mover o tanque como um carro o tanque só pode rodar quando anda;
4. Altere a física do carro para quando não se está a acelerar perder velocidade;
5. Acrescente inclinações à carroçaria do carro quando ele acelera (levanta a frente ou a traseira) e quando vira (inclina-se para um lado ou para o outro)

Observações

`gluPerspective(fov, aspect, near, far)`

Define uma vista em perspectiva e é usado na matriz PROJECTION

`fov` – abertura da lente da câmara em graus

`aspect` – aspecto da câmara/ecrã (largura/altura)

`near, far` – distância dos pontos mais próximo e mais distante que a câmara consegue ver, devem ser maiores que 0 (idealmente maiores que 0.1)

`gluLookAt(eye_x, eye_y, eye_z, center_x, center_y, center_z, up_x, up_y, up_z)`

Coloca a câmara na cena e deve ser usado na matriz MODELVIEW

`eye` – coordenadas onde fica a câmara

`center` – coordenadas para onde está a olhar a câmara

`up` – vector que define a orientação da parte superior da câmara

`quad=gluNewQuadric();`

Cria um objecto quadrático (`GLUquadric* quad`)

`gluCylinder(quad, raio_base, raio_topo, altura, num_fatias, num_aros);`

Instrução para desenhar um cilindro (tronco de cone)

`quad` – objecto quadrático.

Ver:

`gluDeleteQuadric;`

`gluQuadricDrawStyle;`

```
gluQuadricNormals;  
gluQuadricOrientation;  
gluQuadricTexture;  
gluSphere  
gluDisk;  
gluPartialDisk;
```

Exemplo:

```
GLUquadric *quad;  
quad=gluNewQuadric();  
gluQuadricDrawStyle(quad, GLU_FILL);  
gluCylinder(quad, 0.5, 0.5, 0.4, 20, 2);
```

A 2ª e 3ª linhas podem estar na função *Init()* se a variável **quad* for global.