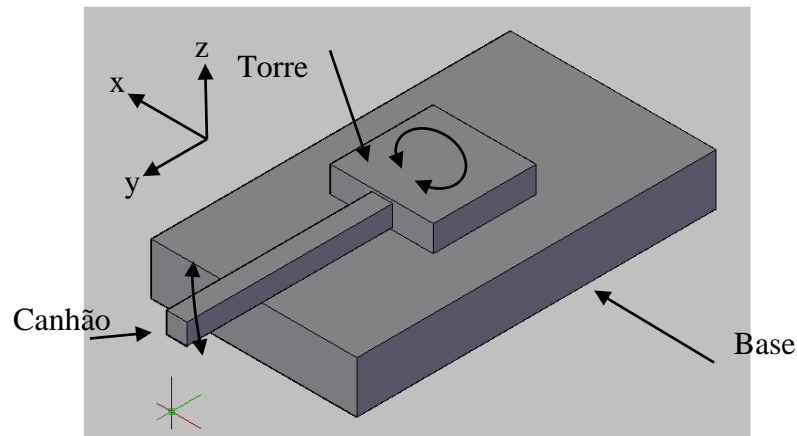


Aulas Práticas Laboratoriais n.ºs 5 e 6

Sumário

Projecto “tanque”. Tutoriais Nate Robins “lightposition” e “lightmaterial”.

Projecto “chaimite”. Tutoriais Nate Robins “texture” e “fog”.



Enunciado

Projecto “tanque”

1. Crie a função `void desenhaTanque(Tanque t)`, que desenha um tanque com a base centrada na posição `t.x`, `t.y`, `0.0` (use a função `cubo()` da aula anterior e a instrução `glScalef()` para desenharmos 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
```

2. Altere a função de modo a implementar a rotação da torre e do canhão usando os ângulos `t.angTorre` e `t.angCanhao`.
3. Acrescente o código das teclas necessárias para rodar a torre (‘z’ e ‘x’) e elevar e baixar o canhão (‘q’ e ‘a’);

4. Ponha o tanque a rodar em torno do centro geométrico do chão da cena. O raio de rotação deverá ser igual a `RAIO_ROTACAO`;

Projecto “chaimite”

1. Modele as rodas do tanque com a instrução `gluCylinder()`. Não se esqueça de elevar previamente o tanque do chão.
2. Acrescente o código das teclas necessárias para conduzir o tanque (avancar, recuar, rodar para a esquerda e rodar para a direita) A rotação só deverá realizar-se se o tanque se encontrar em andamento.
3. Acrescente atrito ao movimento do tanque.
4. Acrescente as transformações necessárias para inclinar o tanque quando este acelera/abrandar (eleva a parte dianteira/traseira) e quando vira (eleva o lado de dentro).

Observações

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

Define uma projecção em perspectiva e é usado na matriz `GL_PROJECTION`

`fov` – campo de visão vertical da câmara (em graus)

`aspect` – *ratio* de aspecto (largura / altura)

`near, far` – distância dos pontos mais próximo e mais distante que a câmara consegue ver. Devem obedecer à inequação $far > near > 0.0$.

```
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 `GL_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_aos)
```

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);  
gluDeleteQuadric(quad);
```