

## 16 Rotações

---

Rotações no espaço levam um objeto de uma posição caracterizada pelas coordenadas  $(x_1, y_1, z_1)$  a uma nova posição caracterizada pelas coordenadas  $(x_2, y_2, z_2)$ . A transformação de coordenadas que leva de uma posição a outra é dada por:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}$$

Os valores dos coeficientes  $a_{ij}$  dependem do ângulo de rotação e do eixo em torno do qual a rotação é realizada. Para rotações de ângulos  $\alpha$ ,  $\beta$  e  $\gamma$  em torno dos eixos  $x$ ,  $y$  e  $z$ , respectivamente, as matrizes de rotação são:

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix} \quad \mathbf{R}_y = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \quad \mathbf{R}_z = \begin{pmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

O script abaixo preenche uma matriz de rotação para um ângulo de rotação aleatoriamente escolhido em torno de um eixo também aleatoriamente escolhido.

exemplo-16-1.html

```
<script>
var x1, y1, z1;
var x2, y2, z2;

// gera um ângulo de rotação entre 0 e 90 graus
var ang = Math.random()*Math.PI/2;
with (Math) {
  var cosAng = cos(ang);
  var sinAng = sin(ang);
}

// gera um índice que define o eixo de rotação
var ind;
do { ind = Math.ceil(Math.random() * 3); } while (ind==0);
```

```

var rot = new Array();

switch (ind) {
  case 1: // roda em torno do eixo x
    var eixo = 'x';
    rot[0] = [1,0,0];
    rot[1] = [0,cosAng,sinAng];
    rot[2] = [0,-sinAng,cosAng];
    break;
  case 2: // roda em torno do eixo y
    var eixo = 'y';
    rot[0] = [cosAng,0,sinAng];
    rot[1] = [0,1,0];
    rot[2] = [-sinAng,0,cosAng];
    break;
  case 3: // roda em torno do eixo z
    var eixo = 'z';
    rot[0] = [cosAng,sinAng,0];
    rot[1] = [-sinAng,cosAng,0];
    rot[2] = [0,0,1];
    break;
  default: // matriz identidade
    rot[0] = [1,0,0];
    rot[1] = [0,1,0];
    rot[2] = [0,0,1];
    break;
}

x1 = 1; y1 = 0; z1 = 0;
x2 = rot[0][0] * x1 + rot[0][1] * y1 + rot[0][2] * z1;
y2 = rot[1][0] * x1 + rot[1][1] * y1 + rot[1][2] * z1;
z2 = rot[2][0] * x1 + rot[2][1] * y1 + rot[2][2] * z1;

document.write("A rotação de (");
document.write(x1.toFixed(1) + ", ");
document.write(y1.toFixed(1) + ", ");
document.write(z1.toFixed(1) + ") ");
document.write("de ");
document.write((ang*180/Math.PI).toFixed(0) + "° ");
document.write("em torno do eixo <i>" + eixo + "</i> ");
document.write("leva a (");
document.write(x2.toFixed(1) + ", ");
document.write(y2.toFixed(1) + ", ");
document.write(z2.toFixed(1) + ").");
</script>

```

### Resultado:

A rotação de (1.0, 0.0, 0.0) de 21° em torno do eixo y leva a (0.9, 0.0, -0.4).

O ponto  $(x_1, y_1, z_1)$  armazena as coordenadas antes da rotação, e o ponto  $(x_2, y_2, z_2)$  armazena as coordenadas depois da rotação. O eixo em torno do qual será efetuada a rotação é escolhido através do parâmetro  $r$ : se  $0 < r \leq 1$ , a rotação se dará em torno do eixo  $x$ ; se  $1 < r \leq 2$ , a rotação se dará em torno do eixo  $y$ ;  $2 < r \leq 3$ , a rotação se dará em torno do eixo  $z$ . Para obter números que satisfaçam estas condições, o resultado da chamada a `Math.random()` é multiplicado por três e passado como argumento ao método `ceil` do objeto `Math`, que devolve o menor número inteiro maior ou igual ao argumento (do inglês *ceiling*,

teto). A instrução está em um `do {...} while (...)` para evitar o raro evento de um zero (que não teria significado) ser retornado: a chamada é feita enquanto a variável `ind` for igual a zero.

O seno e o cosseno do ângulo serão utilizados várias vezes no preenchimento da matriz de rotação e pode ser interessante calculá-los previamente, principalmente em situações em que a velocidade dos cálculos é importante. O trecho do script que faz isso ilustra um recurso que economiza digitação: a instrução `with (objeto) { ... }`, que assume que os métodos chamados no bloco delimitado pelas chaves pertencem ao objeto passado como argumento da instrução.

O objeto `Array()`, que armazena a matriz (bidimensional) de rotação, é inicialmente criado sem que o número de elementos seja especificado. A definição do número de elementos em cada dimensão é feita na atribuição de valores aos elementos. São três elementos na primeira dimensão – `rot[0]`, `rot[1]` e `rot[2]` – e três elementos na segunda dimensão – indicados pelo número de itens separados por vírgula dentro dos `[...]` (colchetes). Esta estratégia, apesar de simples e direta, pode deixar o código confuso pois não deixa explícitas as dimensões da matriz no momento de sua criação, o que pode dificultar a interpretação do programa a longo prazo ou por outros programadores.

## Exercícios

1. Substitua o trecho do script que calcula as coordenadas  $(x_2, y_2, z_2)$  por instruções que façam o mesmo, mas utilizem o produto de uma matriz quadrada  $3 \times 3$  por uma matriz coluna  $3 \times 1$ .
2. Modifique o script acima para que imprima o resultado de duas rotações sucessivas de ângulos aleatoriamente escolhidos em torno de eixos aleatoriamente escolhidos (por exemplo,  $17^\circ$  em torno do eixo  $z$  seguido de  $54^\circ$  em torno do eixo  $y$ ).
3. Use um laço `for` para modificar o script de forma que imprima as coordenadas rodadas para rotações sucessivas de  $0^\circ$  a  $90^\circ$  em intervalos de  $10^\circ$  em torno do eixo  $x$  (ou seja, os valores de  $(x_2, y_2, z_2)$  para  $0^\circ, 10^\circ, 20^\circ, \dots, 90^\circ$ )