3D transformaties

1. 3D transformatiematrices
2. Compositie van transformatiematrices
3. 3D rotatie om willekeurige as
4. Orthonormale basis en veranderen coördinatensysteem
5. gluLookAt()
6. Gimbal lock

3D schaling

$$\mathbf{S} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix}$$

$$\mathbf{p}' = \mathbf{S}\mathbf{p} = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0.25 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.25 \\ 1 \end{pmatrix}$$

Rotatie om z-as in 3D

$$\mathbf{R}_z(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{R}_z(90^\circ) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

Rotatie om x- en y-as in 3D

$$\mathbf{R}_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix}$$

$$\mathbf{R}_y(\phi) = \begin{pmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix}$$

Kolommen van 3×3-matrix `loodrecht` en `eenheidsvectoren`

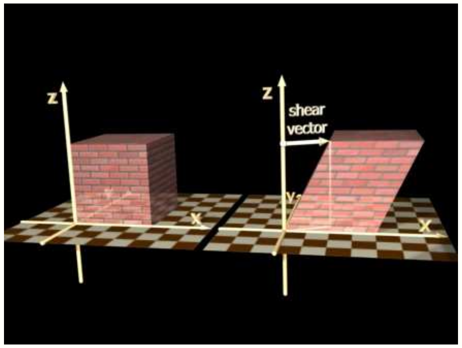
$\Rightarrow$

`3D rotatiematrix` is **orthogonale** matrix

Shear langs x-as

$$\mathbf{S}_{d_y,d_z} = \begin{pmatrix} 1 & d_y & d_z \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{p}' = \mathbf{S}\mathbf{p} = \begin{pmatrix} 1 & d_y & d_z \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x + d_y y + d_z z \\ y \\ z \end{pmatrix}$$



3D transformaties in homogene vorm

3D schaling: $$\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \\ s_z z \\ 1 \end{pmatrix}$$

3D rotatie: $$\begin{pmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x\cos\phi - y\sin\phi \\ x\sin\phi + y\cos\phi \\ z \\ 1 \end{pmatrix}$$

3D x-shear: $$\begin{pmatrix} 1 & d_y & d_z & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + d_y y + d_z z \\ y \\ z \\ 1 \end{pmatrix}$$

Translatie in homogene vorm

3D translatie:
$$\begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + d_x \\ y + d_y \\ z + d_z \\ 1 \end{pmatrix}$$

Voorbeeld schaling om een punt

Bereken de transformatiematrix die een voorwerp schaalt met factoren $(s_x, s_y, s_z) = (2, 3, 2)$ t.o.v. het punt $(1, 1, 0)$.

$T(x_1, y_1, z_1) \cdot S \cdot T(-x_1, -y_1, -z_1) =$

$$\begin{pmatrix} 1 & 0 & 0 & x_1 \\ 0 & 1 & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & -1 \\ 0 & 3 & 0 & -2 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotaties om willekeurige as

3D rotatiematrix om `willekeurige` as:

$$\mathbf{R}_{uvw} = \begin{pmatrix} x_u & y_u & z_u \\ x_v & y_v & z_v \\ x_w & y_w & z_w \end{pmatrix}$$

is **orthogonaal**: `rijen` zijn loodrechte eenheidsvectoren

$$\mathbf{u} = \begin{pmatrix} x_u \\ y_u \\ z_u \end{pmatrix}, \mathbf{v} = \begin{pmatrix} x_v \\ y_v \\ z_v \end{pmatrix}, \mathbf{w} = \begin{pmatrix} x_w \\ y_w \\ z_w \end{pmatrix}$$

Rijen orthonormale basis

1. $||\mathbf{u}|| = ||\mathbf{v}|| = ||\mathbf{w}|| = 1 \Rightarrow \mathbf{u} \cdot \mathbf{u} = \mathbf{v} \cdot \mathbf{v} = \mathbf{w} \cdot \mathbf{w} = 1$

2. $\mathbf{u} \perp \mathbf{v} \perp \mathbf{w} \Rightarrow \qquad \mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{w} = \mathbf{w} \cdot \mathbf{u} = 0$

Voor **orthogonale** matrix $\mathbf{R}_{uvw}$ geldt: `rij gaat over in as`

$$\mathbf{R}_{uvw}\mathbf{u} = \begin{pmatrix} x_u & y_u & z_u \\ x_v & y_v & z_v \\ x_w & y_w & z_w \end{pmatrix} \begin{pmatrix} x_u \\ y_u \\ z_u \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Rij gaat over in as

Orthogonale matrix $\mathbf{R}_{uvw}$:

$$\mathbf{R}_{uvw}\mathbf{u} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{R}_{uvw}\mathbf{v} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{R}_{uvw}\mathbf{w} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$\mathbf{R}_{uvw}$ voert **orthonormale basis** $\mathbf{u}, \mathbf{v}, \mathbf{w}$ over in $\mathbf{x}, \mathbf{y}, \mathbf{z}$-as

Je kunt altijd **rotatiematrix** maken uit `orthonormale basis`!!!

Willekeurige en orthogonale matrix

Voor **elke** 3-3 matrix geldt: `as gaat over in kolom`

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \end{pmatrix}$$

Voor **orthogonale** matrix $\mathbf{R}$ geldt: `rij gaat over in as`

$$\begin{pmatrix} x_u & y_u & z_u \\ x_v & y_v & z_v \\ x_w & y_w & z_w \end{pmatrix} \begin{pmatrix} x_u \\ y_u \\ z_u \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$\mathbf{R}$ orthogonaal $\leftrightarrow \mathbf{R}^T = \mathbf{R}^{-1}$

Als $\mathbf{R}$ `rotatiematrix`, dan

- $\mathbf{R}^T$ ook `rotatiematrix`
- $\mathbf{R}^T = \mathbf{R}^{-1}$

Inverse van orthogonale matrix is **Transpose**

- $\mathbf{R_{uvw}}$ voert `orthonormale basis` $\mathbf{u}, \mathbf{v}, \mathbf{w}$ over in $\mathbf{x}, \mathbf{y}, \mathbf{z}$-as
- $\mathbf{R}^T_{\mathbf{uvw}}$ voert `orthonormale basis` $\mathbf{x}, \mathbf{y}, \mathbf{z}$ over in $\mathbf{u}, \mathbf{v}, \mathbf{w}$-as

$\mathbf{R}_{\mathbf{uvw}}^T$

$\mathbf{R}_{\mathbf{uvw}}^T$ voert `orthonormale basis` $\mathbf{x}, \mathbf{y}, \mathbf{z}$ over in $\mathbf{u}, \mathbf{v}, \mathbf{w}$-as

$$\mathbf{R}_{uvw}^T \mathbf{x} = \begin{pmatrix} x_u & x_v & x_w \\ y_u & y_v & y_w \\ z_u & z_v & z_w \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} x_u \\ y_u \\ z_u \end{pmatrix} = \mathbf{u}$$

Wat is rotatiematrix om vector **a** over hoek ϕ?

1. Maak **orthonormale** basis $\mathbf{u}, \mathbf{v}, \mathbf{w}$ met $\mathbf{w} = \mathbf{a}$

2. $\begin{pmatrix} - & \mathbf{u} & - \\ - & \mathbf{v} & - \\ - & \mathbf{a} & - \end{pmatrix}$ voert $\mathbf{u}, \mathbf{v}, \mathbf{a}$ over in basis $\mathbf{x}, \mathbf{y}, \mathbf{z}$

3. Roteer om $\mathbf{z}$-as over ϕ

4. Roteer $\mathbf{x}, \mathbf{y}, \mathbf{z}$ terug naar $\mathbf{u}, \mathbf{v}, \mathbf{a}$

Rotatiematrix om vector **a** over hoek ϕ

$$\begin{pmatrix} | & | & | \\ \mathbf{u} & \mathbf{v} & \mathbf{a} \\ | & | & | \end{pmatrix} \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} - & \mathbf{u} & - \\ - & \mathbf{v} & - \\ - & \mathbf{a} & - \end{pmatrix}$$

met $\mathbf{u}, \mathbf{v},$ **a orthonormale basis**

1. Maak orthonormale basis

Gegeven: vector $\mathbf{a}$

- Normeer $\mathbf{a}$: $\mathbf{a} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$
- Construeer $\mathbf{tmp}$ uit $\mathbf{a}$ met kleinste waarde gelijk aan 1
- $\mathbf{u} = \frac{\mathbf{tmp} \times \mathbf{a}}{\|\mathbf{tmp} \times \mathbf{a}\|}$
- $\mathbf{v} = \mathbf{a} \times \mathbf{u}$

1. Rekenvoorbeeld

Gegeven: vector $\mathbf{a} = (2, 3, 1)$

- Normeer $\mathbf{a}$: $\|\mathbf{a}\| = \sqrt{4 + 9 + 1} = 3.7 \Rightarrow \mathbf{a} = \begin{pmatrix} 0.54 \\ 0.8 \\ 0.27 \end{pmatrix}$
- $\mathbf{tmp} = (0.54, 0.8, 1)$
- $\mathbf{u} = \frac{\mathbf{tmp} \times \mathbf{a}}{\|\mathbf{tmp} \times \mathbf{a}\|} = (0.83, -0.56, 0)$
- $\mathbf{v} = \mathbf{a} \times \mathbf{u} = (0.15, 0.22, -0.96)$

Rekenvoorbeeld stappen 1 t/m 4

Wat is rotatiematrix om vector $\mathbf{a} = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$ over hoek $45°$?

1. `orthon.` basis $\mathbf{u}, \mathbf{v}, \mathbf{a} = \begin{pmatrix} 0.83 \\ -0.56 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.15 \\ 0.22 \\ -0.96 \end{pmatrix}, \begin{pmatrix} 0.54 \\ 0.8 \\ 0.27 \end{pmatrix}$

2. $\begin{pmatrix} - & \mathbf{u} & - \\ - & \mathbf{v} & - \\ - & \mathbf{a} & - \end{pmatrix} = \begin{pmatrix} 0.83 & -0.56 & 0 \\ 0.15 & 0.22 & -0.96 \\ 0.54 & 0.8 & 0.27 \end{pmatrix}$

   voert $\mathbf{u}, \mathbf{v}, \mathbf{a}$ over in $\mathbf{xyz}$-as

3. Roteer om $\mathbf{z}$-as over $45°$

$$\begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.71 & -0.71 & 0 \\ 0.71 & 0.71 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

4. Roteer $\mathbf{x}, \mathbf{y}, \mathbf{z}$ terug naar $\mathbf{u}, \mathbf{v}, \mathbf{a}$

$$\begin{pmatrix} | & | & | \\ \mathbf{u} & \mathbf{v} & \mathbf{a} \\ | & | & | \end{pmatrix} = \begin{pmatrix} 0.83 & 0.15 & 0.54 \\ -0.56 & 0.22 & 0.8 \\ 0 & -0.96 & 0.27 \end{pmatrix}$$

Rotatiematrix om vector $(2,3,1)$ over hoek 45°

$$\begin{pmatrix} 0.83 & 0.15 & 0.54 \\ -0.56 & 0.22 & 0.8 \\ 0 & -0.96 & 0.27 \end{pmatrix} \begin{pmatrix} 0.71 & -0.71 & 0 \\ 0.71 & 0.71 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.83 & -0.56 & 0 \\ 0.15 & 0.22 & -0.96 \\ 0.54 & 0.8 & 0.27 \end{pmatrix} =$$

$$\begin{pmatrix} 0.79 & -0.06 & 0.61 \\ 0.32 & 0.9 & -0.32 \\ -0.53 & 0.44 & 0.72 \end{pmatrix}$$

Vector $(2,3,1)$ gaat over `in zichzelf`

$$\begin{pmatrix} 0.79 & -0.06 & 0.61 \\ 0.32 & 0.9 & -0.32 \\ -0.53 & 0.44 & 0.72 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}$$

Draaihoek van $R_x(\frac{\pi}{3})$

draaihoek aflezen uit diagonaal matrix:

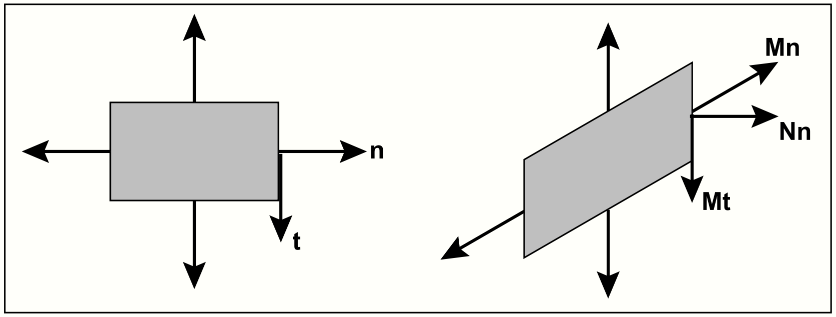
$\cos(\textbf{hoek}) = (\texttt{trace}(R) - 1)/2$

$\cos(\textbf{hoek}) = 1/2 \Rightarrow \textbf{hoek} = \pi/3$

Willekeurige rotatie-matrix:

1. **hoek** uit diagonaal van matrix
2. **rotatie-as** met behulp van eigenwaarde 1 en bijbehorende eigenvector

Transformeren van normaalvectoren



Als **normaalvector** $\mathbf{n}$ getransformeerd met

- $\mathbf{M}$, dan $\mathbf{Mn}$ `niet` loodrecht op rechthoek
- Welke matrix $\mathbf{N}$ levert $\mathbf{Nn}$ **wel** loodrecht op rechthoek?

Transformeren van normaalvectoren

Vergelijking vlak $n_1 x + n_2 y + n_3 z + n_4 = 0$

$$\mathbf{n} = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{pmatrix} \text{en} \quad \mathbf{p} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Vlak: $\mathbf{n}^T \mathbf{p} = 0$ (**vectorproduct**)

Waar gaat n in over?

$$\begin{tikzcd}
\mathbf{p} \arrow[r, dash, "M"] \arrow[d] & \mathbf{p}' \arrow[d] \\
\mathbf{n} \arrow[r, dash, "?"] & \mathbf{n}'
\end{tikzcd}$$

$$\mathbf{n}' = (\mathbf{M}^{-1})^T \mathbf{n} \qquad \text{als } \mathbf{M}^{-1} \text{ bestaat}$$

Als $\mathbf{M}$ transformatiematrix van **punt**,
dan $(\mathbf{M}^{-1})^T$ transformatiematrix van **normaalvector**

Rekenvoorbeeld

Gegeven: matrix $\mathbf{M} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ (**y-shear**)

Gevraagd: matrix $\mathbf{N}$ voor normaalvectoren?

$$\mathbf{N} = (\mathbf{M}^{-1})^T = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$$

Transformeer $\mathbf{n} = (0, 1)$

$$\mathbf{N}\mathbf{n} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

1. Transformaties van vectoren

Gegeven $\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ en $\mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ en afbeelding M:

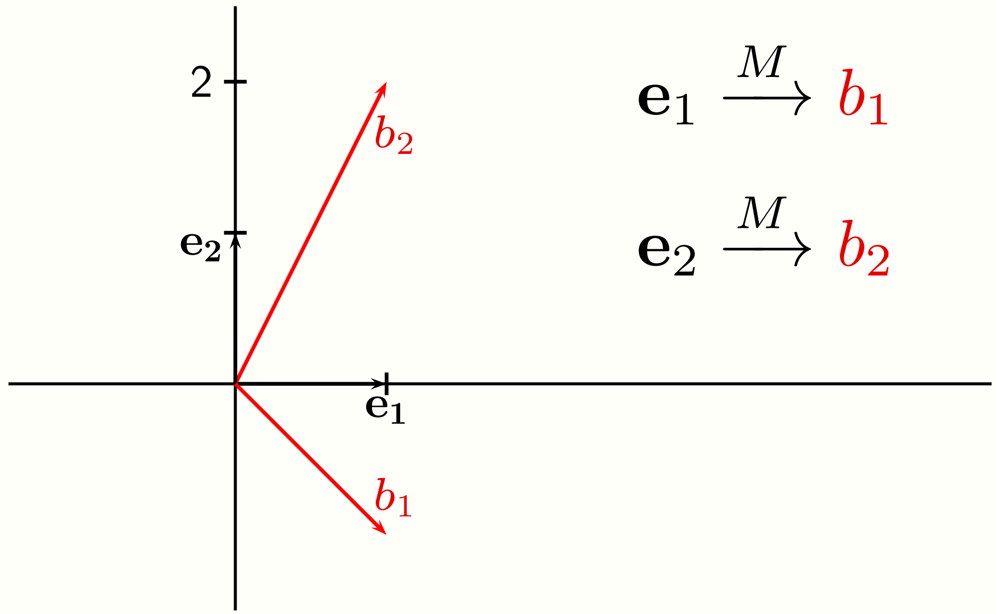
$M\mathbf{e}_1 = b_1 \qquad M\mathbf{e}_2 = b_2$

Wat is M?

$$M \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_1' \end{pmatrix} \qquad M \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} b_2 \\ b_2' \end{pmatrix} \Rightarrow$$

$$M = \begin{pmatrix} b_1 & b_2 \\ b_1' & b_2' \end{pmatrix}$$

1. Transformaties van vectoren



$$\mathbf{e}_1 \xrightarrow{M} b_1$$

$$\mathbf{e}_2 \xrightarrow{M} b_2$$

$$M = \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix}$$

2. Veranderen coördinatensysteem

Kies als basis $B = \{b_1, b_2\}$ met $b_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ en $b_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

Stel de coördinaten van punt P t.o.v. B zijn $P_B = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \Rightarrow$

$$P = \beta_1 b_1 + \beta_2 b_2 = \begin{pmatrix} b_1 & b_2 \\ | & | \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = M \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}$$

$P = M \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}$ zijn coördinaten van P t.o.v. $\mathbf{e}_1, \mathbf{e}_2$

$$P = b_1 + b_2 = 1 \begin{pmatrix} 1 \\ -1 \end{pmatrix} + 1 \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}_B = \begin{pmatrix} 2 \\ 1 \end{pmatrix}_E$$

2. Veranderen coördinatensysteem

$$M_{B\to E}$$

$$\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix}_{\mathbf{e}_1,\mathbf{e}_2} = M \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}_{b_1,b_2}$$

Dus M zet een punt t.o.v. b_1,b_2 in een punt t.o.v. $\mathbf{e}_1, \mathbf{e}_2$

$$\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}_{b_1,b_2} = M^{-1} \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix}_{\mathbf{e}_1,\mathbf{e}_2}$$

Dus M^{-1} zet een punt t.o.v. $\mathbf{e}_1, \mathbf{e}_2$ in een punt t.o.v. b_1,b_2

1. Transformaties van vectoren

$$\mathbf{e}_1 \xrightarrow{M} b_1$$

$$\mathbf{e}_2 \xrightarrow{M} b_2$$

2. Veranderen coördinatensysteem

$$\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}_{b_1,b_2} \xrightarrow{M} \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix}_{\mathbf{e}_1,\mathbf{e}_2}$$

Rekenvoorbeeld

$$\mathbf{e}_1 \xrightarrow{M?} b_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad \mathbf{e}_2 \xrightarrow{M?} b_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \Rightarrow M = \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix}$$

$$\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}_{b_1,b_2} \xrightarrow{M} \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix}_{\mathbf{e}_1,\mathbf{e}_2}$$

Gegeven $P = \begin{pmatrix} 1 \\ 1 \end{pmatrix}_{b_1,b_2}$

Wat is P t.o.v. $(\mathbf{e}_1, \mathbf{e}_2)$?

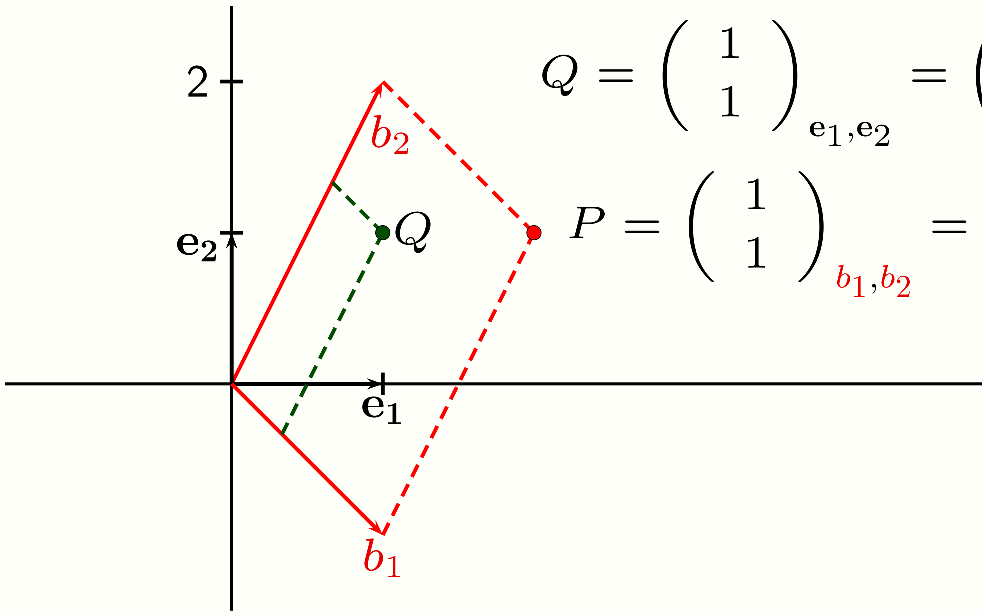
$$P = \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}_{b_1, b_2} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}_{\mathbf{e}_1, \mathbf{e}_2}$$

Gegeven $Q = \begin{pmatrix} 1 \\ 1 \end{pmatrix}_{\mathbf{e}_1, \mathbf{e}_2}$

Wat is Q t.o.v. (b_1, b_2)?

$$Q = \frac{1}{3} \begin{pmatrix} 2 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix}_{\mathbf{e}_1, \mathbf{e}_2} = \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix}_{b_1, b_2}$$

2. Veranderen coördinatensysteem



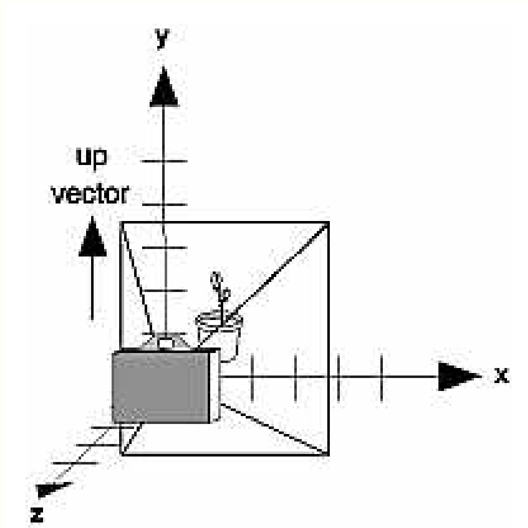
$$Q = \begin{pmatrix} 1 \\ 1 \end{pmatrix}_{\mathbf{e}_1,\mathbf{e}_2} = \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix}_{b_1,b_2}$$

$$P = \begin{pmatrix} 1 \\ 1 \end{pmatrix}_{b_1,b_2} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}_{\mathbf{e}_1,\mathbf{e}_2}$$

Positioneren van de camera

gluLookAt **default positie**: camera in oorsprong, kijkend in negatieve z-richting, en positieve y-as rechtop

```
gluLookAt (0.0, 0.0, 0.0, 0.0, 0.0, -100.0, 0.0, 1.0, 0.0);
```

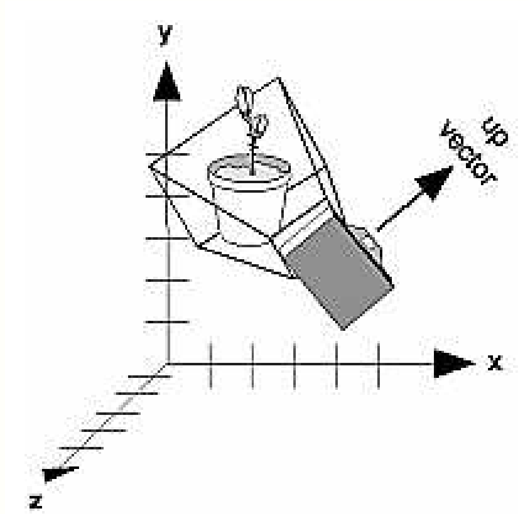


z-waarde kan elke negatieve waarde van z zijn

Zelf gedefinieerde positie

positie: camera in $(4, 2, 1)$, kijkend naar $(2, 4, -3)$, orientatie vector $(2, 2, -1)$

```
gluLookAt(4.0, 2.0, 1.0, 2.0, 4.0, -3.0, 2.0, 2.0, -1.0);
```



gluLookAt / m4.lookAt

Wat doet gluLookAt(eye, lookat, up)?
Bouwt matrix die wereldcoördinaten omzet in eye/camera-coördinaten

eye/camera coördinatensysteem:
$n = eye - lookat$
$u = up \times n$
$v = n \times u$

We zoeken rotatiematrix R die: $u \xrightarrow{R} x \quad v \xrightarrow{R} y \quad n \xrightarrow{R} z$
R voert dan **wereld**- over in **camera**-coördinaten

gluLookAt / m4.lookAt

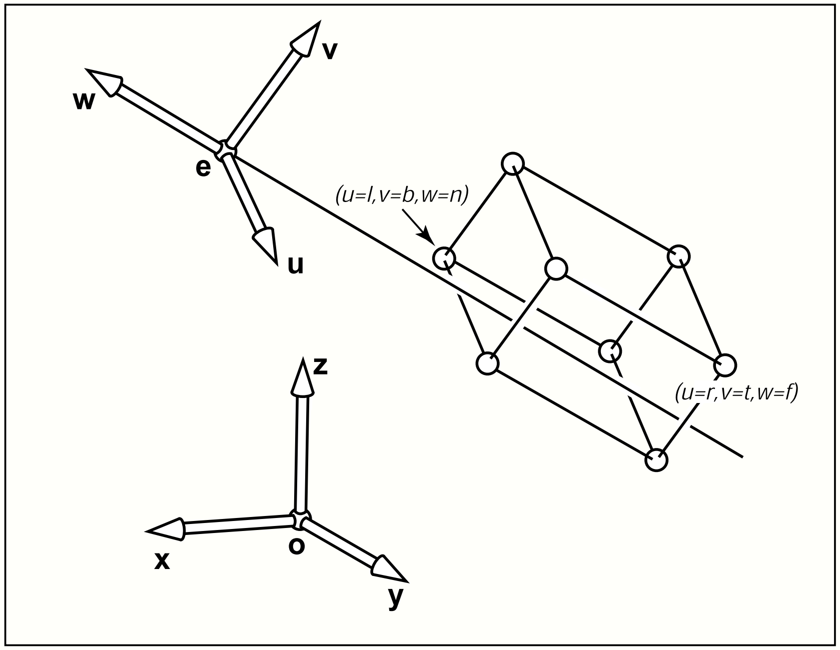
We zoeken rotatiematrix R die: $u \xrightarrow{R} x \quad v \xrightarrow{R} y \quad n \xrightarrow{R} z$

We weten dat u, v en n orthonormaal stelsel van vectoren

$$R = \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ n_x & n_y & n_z \end{pmatrix}$$

$$\begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ n_x & n_y & n_z \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Transformatie die wereldcoördinaten overvoert in cameracoördinaten



Eerst translatie van $\mathbf{e}$ naar oorsprong, dan gevonden rotatie R

Transformatie die wereldcoördinaten overvoert in cameracoördinaten

$$\begin{pmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -e_x \\ 0 & 1 & 0 & -e_y \\ 0 & 0 & 1 & -e_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Wat doet gluLookAt(oog, lookat, up)?

```
 eye = [4; 4; 4];
look = [0; 1; 0];
up = [0; 1; 0];

n = eye - look; n = n/norm(n);
u = cross(up,n); u = u/norm(u);
v = cross(n,u); v = v/norm(v);
dx = -eye' * u
dy = -eye' * v
dz = -eye' * n
V = [u' dx; v' dy; n' dz; 0 0 0 1]
```

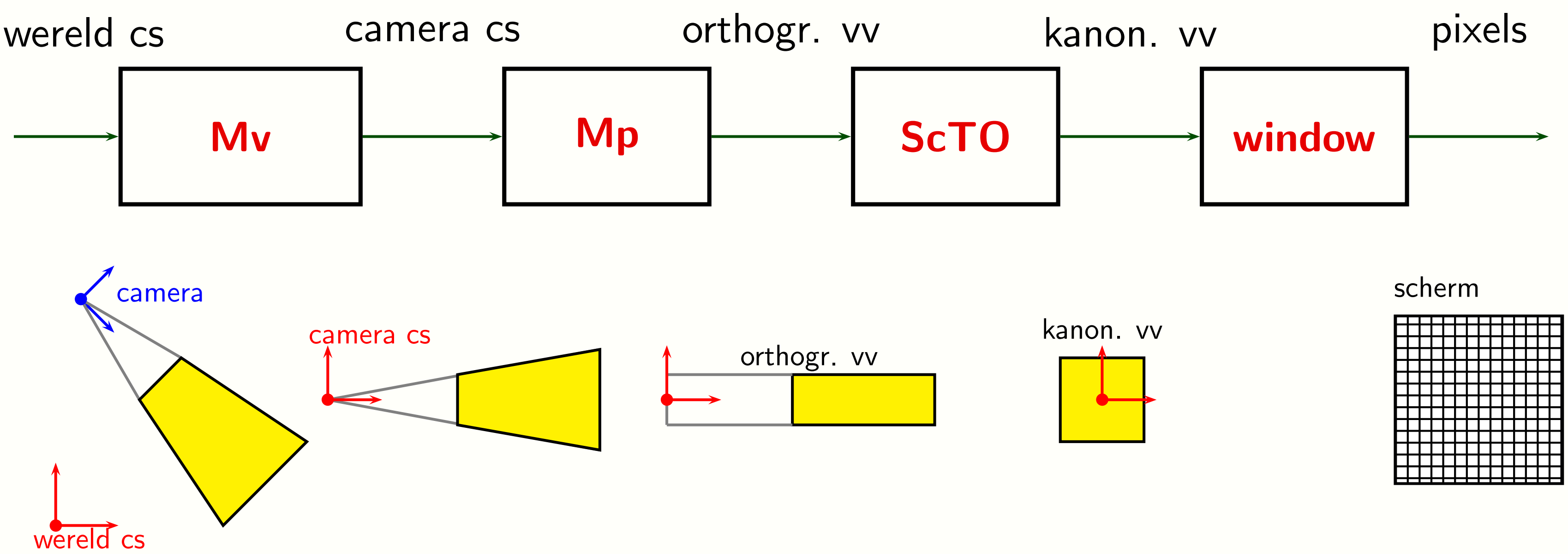
gluLookAt

```
gluLookAt(4,4,4,0,1,0,0,1,0)
```

```
glGetFloatv(GL_MODELVIEW_MATRIX,mat)
```

$$\begin{pmatrix} 0.7071 & 0 & -0.7071 & 0 \\ -0.3313 & 0.8835 & -0.3313 & -0.8835 \\ 0.6247 & 0.4685 & 0.6247 & -6.8716 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$$

Grafische pijplijn



Grafische pijplijn

To specify **viewing**, **modeling**, and **projection** transformations, you construct a 4×4 matrix M, which is then multiplied by coordinates of each vertex v in the scene to accomplish the transformation

$$v' = Mv$$

Welke ModelViewmatrix onstaat hier?

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glMultMatrixf(N);
glMultMatrixf(M);
glMultMatrixf(L);
glBegin(GL_POINTS);
glVertex3f(v);
glEnd();
```

$$v' = NML(v)$$

This means that **last** transformation command called in your program is actually **first** one applied to the vertices

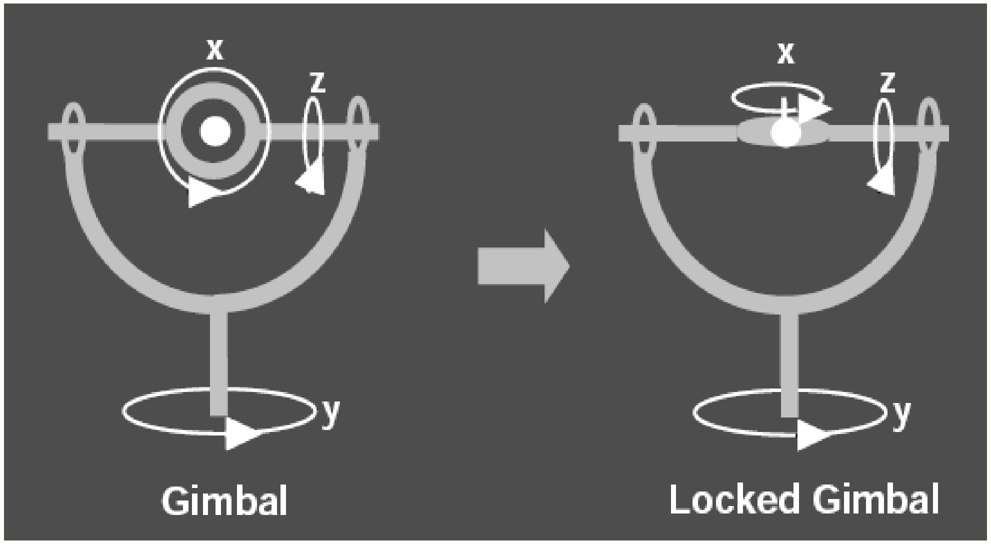
Wat is gimbal lock?

http://en.wikipedia.org/wiki/Gimbal_lock

- **Gimbal lock** is fenomeen van twee rotatie assen van object die in dezelfde richting staan
- het betekent dat je object niet roteert zoals je zou verwachten
- Gimbal lock is frustrerend probleem van iedere CG artiest
- Gimbal lock treedt op bij animatie van object met rotatiematrices met Euler hoeken

Gimbal lock

Gimbal lock ontstaat als assen van twee gimbals in `dezelfde richting`: één vrijheidsgraad verloren

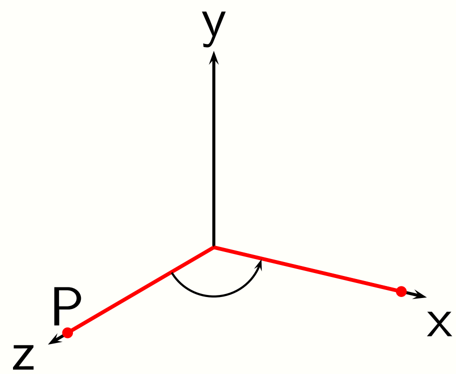


Gimbal is hardware implementatie van `Euler hoeken`

Wiskunde van gimbal lock

Gimbal lock ontstaat als `as` wordt overgevoerd in `andere as`

Bijvoorbeeld rotatie over 90° om y-as voert z-as over in x-as $\Rightarrow$ rotatie om x-as $=$ rotatie om z-as



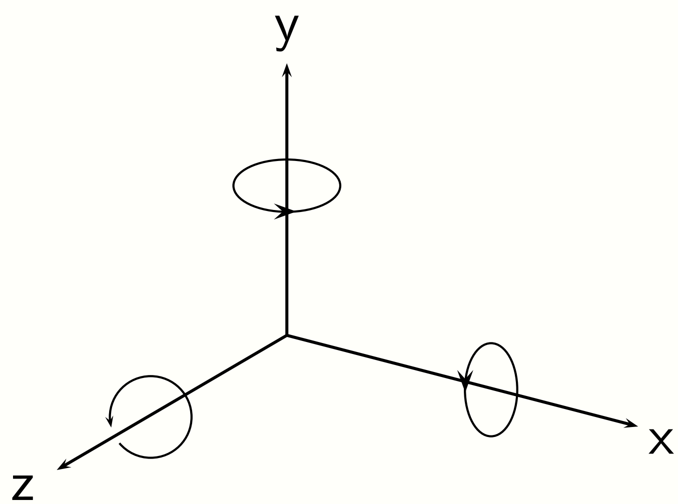
3D rotatiematrices

$$R_x(p) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos p & -\sin p \\ 0 & \sin p & \cos p \end{pmatrix}$$

$$R_y(h) = \begin{pmatrix} \cos h & 0 & \sin h \\ 0 & 1 & 0 \\ -\sin h & 0 & \cos h \end{pmatrix}$$

$$R_z(r) = \begin{pmatrix} \cos r & -\sin r & 0 \\ \sin r & \cos r & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$R_x(90^\circ)$ voert y-as over in z-as

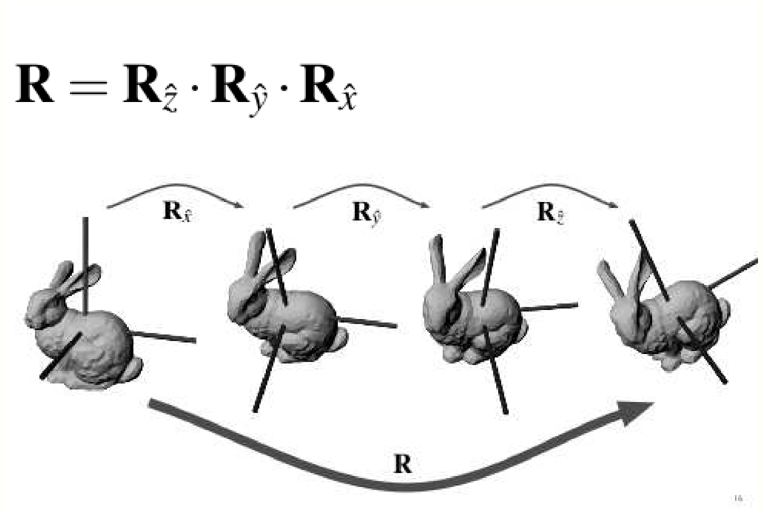


$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos 90^\circ & -\sin 90^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Euler

Willekeurige rotatie kan samengesteld worden door 3 rotatiematrices om as:

$$R = R_y R_x R_z$$



Voorbeeld: $R_z(r)R_x(90^\circ)R_y(h)$

Bekijk de volgende 3 rotaties:

1. rotatie om y-as met hoek h
2. rotatie om x-as met hoek 90°
3. rotatie om z-as met hoek r

$R_z(r)R_x(90^\circ)R_y(h)$

$$\begin{pmatrix} \cos r & -\sin r & 0 \\ \sin r & \cos r & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos 90^\circ & -\sin 90^\circ \\ 0 & \sin 90^\circ & \cos 90^\circ \end{pmatrix} \begin{pmatrix} \cos h & 0 & \sin h \\ 0 & 1 & 0 \\ -\sin h & 0 & \cos h \end{pmatrix} =$$

$$\begin{pmatrix} \cos(r+h) & 0 & \sin(r+h) \\ \sin(r+h) & 0 & -\cos(r+h) \\ 0 & 1 & 0 \end{pmatrix}$$

Deze matrix is maar van **één hoek** (`r + h`) afhankelijk $\Rightarrow$
Dit heet **gimbal lock**: één `vrijheidsgraad` is verloren

Opgave 12

1. Gegeven een orthonormale basis $\mathbf{u}, \mathbf{v}, \mathbf{w}$. Wat voor matrix is $\mathbf{M_{uvw}} = \begin{pmatrix} x_u & y_u & z_u \\ x_v & y_v & z_v \\ x_w & y_w & z_w \end{pmatrix}$ en waar worden $\mathbf{u}, \mathbf{v}, \mathbf{w}$ in overgevoerd?

2. Wat is de inverse van $\mathbf{M_{uvw}}$?

3. Welke matrix voert de vectoren $\mathbf{u}, \mathbf{v}, \mathbf{w}$ over in de vectoren $\mathbf{a} = (1, 2, 3)$, $\mathbf{b} = (1, 1, 1)$ en $\mathbf{c} = (2, 1, 0)$?