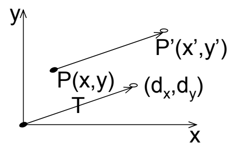
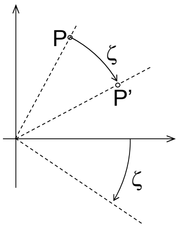
## Geometriai transzformációk



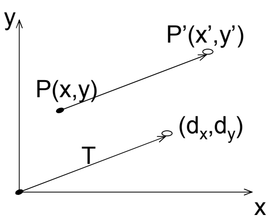

## Bevezetés - Transzformációk

A Számítógépes Grafikában használatos 2- és 3- dimenziós transzformációk:

- eltolás
- nagyítás, kicsinyítés (skálázás)
- forgatás

## Pont 2D eltolása

Hosszak és a szögek változatlanok




$x'=x+d_x$

$y'=y+d_y$

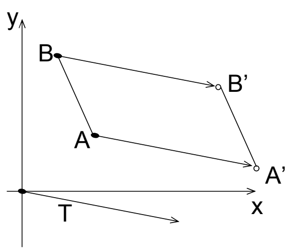
(oszlop-)vektorokkal:

$$P=\begin{pmatrix}x\\y\end{pmatrix}\quad P'=\begin{pmatrix}x'\\y'\end{pmatrix}\quad T=\begin{pmatrix}dx\\dy\end{pmatrix}$$

P'=P+T

## Szakasz 2D eltolása

Elegendő az új végpontokat számolni




A'=A+T

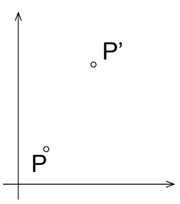
B'=B+T

## 2D nagyítás/kicsinyítés

A szögek változatlanok

Szokták a kettőt együtt **SKÁLÁZÁSKÉNT** említeni

Origóból történő nagyítás




$x'=x\cdot s_x$

$y'=y\cdot s_y$

(oszlop-)vektorokkal:

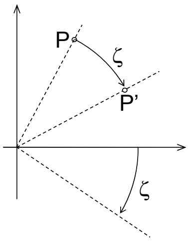
$$P=\begin{pmatrix}x\\y\end{pmatrix}\quad P'=\begin{pmatrix}x'\\y'\end{pmatrix}\quad S=\begin{pmatrix}S_x & 0\\0 & S_y\end{pmatrix}$$

P'=S·P

## 2D forgatás

A hosszak és a szögek változatlanok

Origó körüli forgatás




$x' = x\cdot\cos\zeta - y\cdot\sin\zeta$

$y' = x\cdot\sin\zeta + y\cdot\cos\zeta$

$$\begin{pmatrix}x'\\y'\end{pmatrix}\quad R=\begin{pmatrix}\cos\zeta & -\sin\zeta\\ \sin\zeta & \cos\zeta\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}$$

P' = R · P

## Homogén koordináták

(x, y) jelölése homogén koordinátákkal: (x, y, w)

Egyenlőség:

(x,y,w) = (x',y',w') , ha van olyan $\alpha$ :

hogy $x' = \alpha \cdot x$, $y' = \alpha \cdot y$, $w' = \alpha \cdot w$

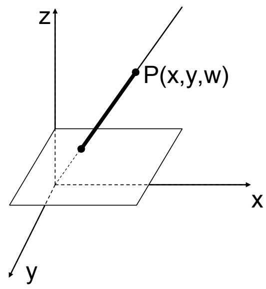
pl: (2, 3, 6) = (4, 6, 12)

Egy ponthoz végtelen sok (x,y,w) tartozik.

Ha w=0, akkor (x,y,w) végtelen távoli pont

(0,0,0) nem megengedett!

## Kapcsolat 2D és 3D közt




$(t \cdot x,\ t \cdot y,\ t \cdot w)$

egyenes a 3D térben

$$\left(\frac{x}{w}, \frac{y}{w}, 1\right)$$

P vetülete a w=1 síkon

A végtelen távoli pontok nincsenek a síkon

## 2D eltolás - matematikailag

$P' = T(d_x, d_y)\, P$ ,

ahol $T(d_x, d_y) = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix}$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Ismételt eltolások (kompozíció):

$$P \xrightarrow[T(d_{x1},\, d_{y1})]{} P' \xrightarrow[T(d_{x2},\, d_{y2})]{} P''$$

$P' = T(d_{x1}, d_{y1})\, P$ ,

$P'' = T(d_{x2}, d_{y2})\, P' = T(d_{x2}, d_{y2})\, (T(d_{x1}, d_{y1})\, P)$

$$T(d_{x1}+d_{x2}, d_{y1}+d_{y2}) = \begin{pmatrix} 1 & 0 & d_{x1} \\ 0 & 1 & d_{y1} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & d_{x2} \\ 0 & 1 & d_{y2} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & d_{x1}+d_{x2} \\ 0 & 1 & d_{y1}+d_{y2} \\ 0 & 0 & 1 \end{pmatrix}$$

## 2D skálázás - matematikailag

$P' = S(s_x, s_y)\, P$ ,

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Ismételt skálázások (kompozíció):

$$P \xrightarrow[S(s_{x1},\, s_{y1})]{} P' \xrightarrow[S(s_{x2},\, s_{y2})]{} P''$$

$P' = S(s_{x1}, s_{y1})\, P$ ,

$P'' = S(s_{x2}, s_{y2})\, P' = S(s_{x2}, s_{y2})\, (S(s_{x1}, s_{y1})\, P)$

$$S(s_{x1}s_{x2}, s_{y1}s_{y2}) = \begin{pmatrix} s_{x2} & 0 & 0 \\ 0 & s_{y2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_{x2}s_{x1} & 0 & 0 \\ 0 & s_{y2}s_{y1} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

## 2D forgatás - matematikailag

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\zeta & -\sin\zeta & 0 \\ \sin\zeta & \cos\zeta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$P' = R(\zeta)\, P$ $\qquad$ ($R(\zeta)$ ortogonális)

Ismételt forgatások:

$P' = R(\zeta_1)\, P$,

$P'' = R(\zeta_2)\, P' = R(\zeta_2)\, (R(\zeta_1)\, P) = R(\zeta_1 + \zeta_2)\, P$

**Bizonyítás:** Házi Feladat

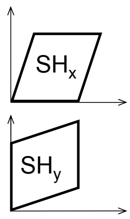
## 2D nyírás – affin transzformációk

A hosszak és a szögek változhatnak

Párhuzamos egyenesek képe párhuzamos

$$SH_x = \begin{pmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad SH_y = \begin{pmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x+ay \\ y \\ 1 \end{pmatrix}$$




**Affin transzformáció:** eltolások, skálázások, forgatások és nyírások tetszőleges számú és sorrendű egymás utáni alkalmazásával kapott transzformáció

## 2D transzformációk kompozíciója
### 1. példa

Forgatás $\zeta$-val egy tetszőleges P(x,y) pont körül.

- a) eltolás P-ből O-ba T(-x,-y)
- b) forgatás az origó körül R($\zeta$)
- c) eltolás O-ból P-be T(x,y)

$$\begin{pmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos\zeta & -\sin\zeta & 0 \\ \sin\zeta & \cos\zeta & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} \cos\zeta & -\sin\zeta & x(1-\cos\zeta)+y\sin\zeta \\ \sin\zeta & \cos\zeta & y(1-\cos\zeta)-x\sin\zeta \\ 0 & 0 & 1 \end{pmatrix}$$

## 2D transzformációk kompozíciója
### 2. példa

Nagyítás egy tetszőleges P(x,y) pontból:

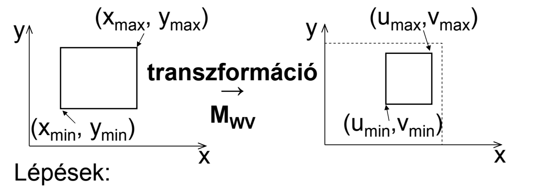
$$T(x,y) \cdot S(s_x,s_y) \cdot T(-x,-y) =$$

$$\begin{pmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{pmatrix} =$$

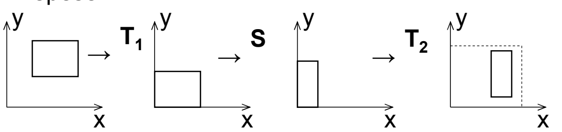
$$= \begin{pmatrix} s_x & 0 & x(1-s_x) \\ 0 & s_y & y(1-s_y) \\ 0 & 0 & 1 \end{pmatrix}$$

## 2D transzformációk kompozíciója
### 3. példa / 1

„Világ koordináta rendszer” „Képernyő koordináta rendszer ”




Lépések:




## 2D transzformációk kompozíciója
### 3. példa /2

$$M_{wv} = T(u_{min}, v_{min}) S\left(\frac{u_{max}-u_{min}}{x_{max}-x_{min}}, \frac{v_{max}-v_{min}}{y_{max}-y_{min}}\right) T(-x_{min}, -y_{min}) =$$

$$= \begin{pmatrix} 1 & 0 & u_{min} \\ 0 & 1 & v_{min} \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & 0 \\ 0 & \frac{v_{max}-v_{min}}{y_{max}-y_{min}} & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & -x_{min} \\ 0 & 1 & -y_{min} \\ 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & -x_{min}\frac{u_{max}-u_{min}}{x_{max}-x_{min}}+u_{min} \\ 0 & \frac{v_{max}-v_{min}}{y_{max}-y_{min}} & -y_{min}\frac{v_{max}-v_{min}}{y_{max}-y_{min}}+v_{min} \\ 0 & 0 & 1 \end{pmatrix}$$

## 2D transzformációk kompozíciója
### 3. példa /3

$$M_{wv} = \begin{pmatrix} \frac{u_{max}-u_{min}}{x_{max}-x_{min}} & 0 & -x_{min}\frac{u_{max}-u_{min}}{x_{max}-x_{min}}+u_{min} \\ 0 & \frac{v_{max}-v_{min}}{y_{max}-y_{min}} & -y_{min}\frac{v_{max}-v_{min}}{y_{max}-y_{min}}+v_{min} \\ 0 & 0 & 1 \end{pmatrix}$$

tehát

$$P' = M_{wv}P(x,y) = \begin{pmatrix} (x-x_{min})\frac{u_{max}-u_{min}}{x_{max}-x_{min}}+u_{min} \\ (y-y_{min})\frac{v_{max}-v_{min}}{y_{max}-y_{min}}+v_{min} \end{pmatrix}$$

## Általános kompozíció mátrix

Könnyű látni, hogy skálázások, forgatások, nyírások és eltolások kompozíciója skálázás/forgatás utáni eltolásként értelmezhető:

**r**: skálázás/forgatás, **t**: eltolás

$$M = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} r_{11} & r_{12} & 0 \\ r_{21} & r_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

Az **M** alakú mátrixokat **kompozíció mátrix**oknak nevezzük

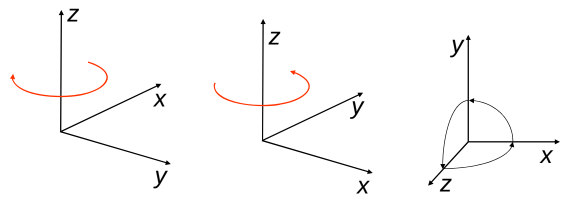
## Gyorsítások

$$M = \begin{pmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{pmatrix}$$

**M·P számításakor:**

1. 9 szorzás és 6 összeadás helyett elegendő
   $x' = x \cdot r_{11} + y \cdot r_{12} + t_x$
   $y' = x \cdot r_{21} + y \cdot r_{22} + t_y$
   kiszámítása, ami csak 4 szorzás és 4 összeadás
2. Ha kicsi $\zeta$ szöggel forgatunk, akkor $\cos\zeta \approx 1$, így
   $x' = x \cdot \cos\zeta - y \cdot \sin\zeta \approx x - y \cdot \sin\zeta$
   $y' = x \cdot \sin\zeta + y \cdot \cos\zeta \approx x \cdot \sin\zeta + y$,
   ami csak 2 szorzás és két összeadás
   (+ sin kiszámítása)

## 3D koordináta-rendszerek




balkezes
bal-sodrású

jobbkezes
jobb-sodrású

## 3D transzformációk - homogén koordináták

**(x,y,z)** megadása homogén koordinátákkal: **(x,y,z,1)**

**(x,y,z,w) = (x',y',z',w')**, ha van olyan $\alpha$, hogy

**$x' = \alpha \cdot x$, $y' = \alpha \cdot y$, $z' = \alpha \cdot z$** és **$w' = \alpha \cdot w$**

Ha **w≠0**: $\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}, 1\right)$ a szokásos jelölés

Ha **w=0**: **(x,y,z,0)** végtelen távoli pont

**Kapcsolat: (x,y,z)** - egyenes a 4-dimenziós térben aminek a **w=1** -re való vetülete a homogén koordináta

## 3D eltolás

$$T(d_x,d_y,d_z) = \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

mert $$T(d_x,d_y,d_z)\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x+d_x \\ y+d_y \\ z+d_z \\ 1 \end{pmatrix}$$

$$T^{-1}(d_x,d_y,d_z) = T(-d_x,-d_y,-d_z)$$

## 3D skálázás (nagyítás/kicsinyítés)

$$S(s_x,s_y,s_z) = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

mert $$S(s_x,s_y,s_z)\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \cdot s_x \\ y \cdot s_y \\ z \cdot s_z \\ 1 \end{pmatrix}$$

$$S^{-1}(s_x,s_y,s_z) = \begin{pmatrix} 1/s_x & 0 & 0 & 0 \\ 0 & 0/s_y & 0 & 0 \\ 0 & 0 & 1/s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

## 3D forgatások

A z-tengely körül

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

Az x-tengely körül

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

Az Y-tengely körül

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

## 3D nyírás (Z mentén)

$$SH_{xy}(sh_x, sh_y) = \begin{pmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

mert

$$SH_{xy}(sh_x, sh_y)\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x + sh_x z \\ y + sh_y z \\ z \\ 1 \end{pmatrix}$$

## 3D kompozíció-mártix

Skálázás/forgatás utáni eltolásként értelmezhető:

$$M = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} & R & & T \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$P = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ 1 \end{pmatrix} \qquad MP = R\begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} + T$$

**Ilyen módon hatékonyabban számítható!**

## 3D - síkok transzformációi

A sík egyenlete: **Ax + By + Cz + D = 0**

Legyen **P** a sík tetszőleges pontja!

Ha $P = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$, akkor $N = \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix}$ a sík normálisa,

hiszen $\mathbf{N^T P = 0}$

Ha a sík pontjait **M**-el transzformáljuk, akkor hogy transzformálódik a sík normálisa?

## 3D - síkok transzformációi

Legyen **P** tetszőleges pont a síkban! Ekkor $\mathbf{N^T P = 0}$.

Melyik az a **Q** mátrix, amelyre $\mathbf{(Q N)^T (M P) = 0}$?

Ha $\mathbf{M^{-1}}$ létezik, akkor

$$\mathbf{((M^{-1})^T\ N)^T (M P) = N^T ((M^{-1})^T)^T M P = N^T P = 0}$$

$$\Downarrow$$

$$\mathbf{Q=(M^{-1})^T}$$

$$\mathbf{N' = (M^{-1})^T\ N}$$

(NEM BIZTOS, hogy létezik normális! Pl: vetítés)

## 3D koordináta-rendszerek váltása /1

$\mathbf{P^{(j)}}$: a **P** pont az **j** koordináta-rendszerben

$\mathbf{M_{i\leftarrow j}}$: transzformáció, amely

a **j** koordináta-rendszerbeli pontokat

az **i** koordináta-rendszerbe viszi át

Ekkor

$$\mathbf{P^{(i)} = M_{i\leftarrow j}\, P^{(j)}}$$

Ha $\mathbf{P^{(j)} = M_{j\leftarrow k}\, P^{(k)}}$, akkor

$$\mathbf{P^{(i)} = M_{i\leftarrow j}\, P^{(j)} = M_{i\leftarrow j}\, (M_{j\leftarrow k}\, P^{(k)}) = M_{i\leftarrow k}\, P^{(k)}}$$

ahol

$$\mathbf{M_{i\leftarrow k} = M_{i\leftarrow j}\, M_{j\leftarrow k}}$$

## 3D koordináta-rendszerek váltása /2

Továbbá

$$\mathbf{M_{i\leftarrow j} = M_{j\leftarrow i}^{-1}}$$

pl:

a) Ha $\mathbf{M_{i\leftarrow j}}$ **=T(tx,ty)**, akkor $\mathbf{M_{j\leftarrow i}}$ **=T(-tx,-ty)**.

b) Ha **R**: jobb-kezes koordináta-rendszer

**L**: bal-kezes koordináta-rendszer, akkor

$$M_{R\leftarrow L} = M_{L\leftarrow R}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

## 3D - transzformációk alakja

(Különböző koordináta-rendszerekben)

$\mathbf{P}^{(j)}$: pont a **j** koordináta-rendszerben

$\mathbf{Q}^{(j)}$: transzformáció a **j** koordináta-rendszerben

Melyik az a $\mathbf{Q}^{(i)}$, amelyre

$$\mathbf{Q}^{(i)}\,\mathbf{P}^{(i)} = \mathbf{M}_{i\leftarrow j}\,\mathbf{Q}^{(j)}\,\mathbf{P}^{(j)}?$$

Mivel $\mathbf{P}^{(i)} = \mathbf{M}_{i\leftarrow j}\,\mathbf{P}^{(j)}$, ezért

$$\mathbf{Q}^{(i)}\,\mathbf{M}_{i\leftarrow j}\,\mathbf{P}^{(j)} = \mathbf{M}_{i\leftarrow j}\,\mathbf{Q}^{(j)}\,\mathbf{P}^{(j)},$$

$$\Downarrow$$

$$\mathbf{Q}^{(i)}\,\mathbf{M}_{i\leftarrow j} = \mathbf{M}_{i\leftarrow j}\,\mathbf{Q}^{(j)}$$

$$\mathbf{Q}^{(i)} = \mathbf{M}_{i\leftarrow j}\,\mathbf{Q}^{(j)}\,\mathbf{M}_{i\leftarrow j}^{-1}$$

## Mátrix műveletek (OpenGL)

OpenGL-ben oszlopfolytonosan tároljuk a mátrixokat

Az egység mátrix:

$$\begin{pmatrix} a_1 & a_5 & a_9 & a_{13} \\ a_2 & a_6 & a_{10} & a_{14} \\ a_3 & a_7 & a_{11} & a_{15} \\ a_4 & a_8 & a_{12} & a_{16} \end{pmatrix}$$

```
GLfloat M[] = {
   1.0, 0.0, 0.0, 0.0,
   0.0, 1.0, 0.0, 0.0,
   0.0, 0.0, 1.0, 0.0,
   0.0, 0.0, 0.0, 1.0}
```

Új aktuális mátrix betöltése:

```
void glLoadMatrix{fd}( T M[16] );

glMatrixMode(GL_MODELVIEW);  // típus
glLoadMatrix(M);             // betöltés
```

## Mátrix műveletek (OpenGL)

Az aktuális mátrix legyen az egységmátrix:

```
void glLoadIdentity(void);
```

Az aktuális mátrix szorzása:

```
void glMultMatrix{fd}( T M[16] );
```

Pl.:

```
GLfloat M[] = {
           1.0, 0.0, 0.0, 10.0,
           0.0, 1.0, 0.0, 0.0,
           0.0, 0.0, 1.0, 0.0,
           0.0, 0.0, 0.0, 1.0}
glMatrixMode(GL_MODELVIEW);
glMultMatrix(M);
```

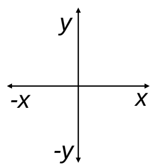
A szorzat lesz az új aktuális mátrix

## Koordináta transzformációk (OpenGL)

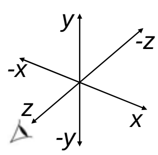
- **Nézeti (Viewing)**
  - a néző (kamera) helyének a megadása
- **Modell (Modeling)**
  - az objektumok (modell) mozgatása
- **Modell-nézet (ModelView)**
  - a nézeti és a modell transzformációk együtt
- **Vetítési (Projection)**
  - a nézet vágása és látótérbe méretezése
- **Ablak**
  - az eredmény ablakra való leképezése

## Nézeti koordináták (OpenGL)

- A megfigyelő nézőpontja kezdetben (0, 0, 0)
- A megfigyelő a z tengely negatív irányába néz. Virtuálisan rögzített koordináta rendszer



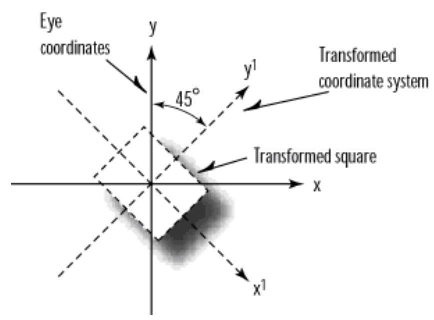

Ahogy a megfigyelő látja a modellt




Így látnánk oldalról a megfigyelőt a pozíciójának a *z* tengely irányába történő elmozdítása után

## Nézeti koordináták (OpenGL)

A nézeti koordináta rendszer elforgatása $45^0$-kal az ótamutató járásával megegyező irányban



## Nézeti (Viewing) transzformáció (OpenGL)

Ez hajtódik végre először, ezt kell legelőször definiálni

Nézőpont meghatározása
- Kezdeti nézőpont (0, 0, 0)
- **`gluLookAt`** paranccsal módosítható

## Nézeti (Viewing) transzformáció (OpenGL)

```
void gluLookAt(
    GLdouble eyex, GLdouble eyey,
    GLdouble eyez,
    GLdouble centerx, GLdouble centery,
    GLdouble centerz,
    GLdouble upx, GLdouble upy,
    GLdouble upz)
```

**`(eyex, eyey, eyez)`** a szem pozíciója

**`(centerx, centery, centerz)`** referenciapont, ahová a szem néz

**`(upx, upy, upz)`** felfelé mutató vektor (up-vektor,VUP)

Pl.:

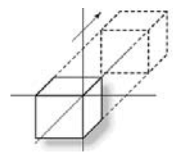
```
gluLookAt(0.0,0.0,2.0, 0.0,0.0,0.0,
    0.0,1.0,0.0);
```

## Nézeti (Viewing) transzformáció (OpenGL)

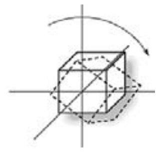
A **`gluLookAt`** eljárás kiszámítja a megadott nézeti transzformáció inverzét, majd az aktuális mátrixot megszorozza a kapott inverz transzformációs mátrixszal

Az altuális mátrix mód a **`GL_MODELVIEW`** legyen!

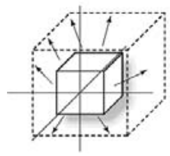
## Modell (Modeling) transzformáció (OpenGL)



eltolás (transzláció)



forgatás (rotáció)



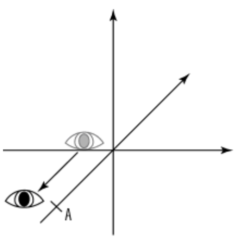
skálázás

A modell vagy egy részének a transzformálására használjuk

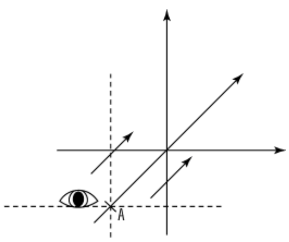
A csúcspont (vertex) koordinátákat transzformálja

## Modell-nézeti dualitás (OpenGL)

A nézeti és a modell transzformációk duálisak, ezért elegendő csak a modell koordináta rendszert transzformálni




nézeti koordináta rendszer mozgatás




modell koordináta rendszer mozgatás

## Modell transzformációk (OpenGL)

**`glMatrixMode(GL_MODELVIEW);`**

**`void glRotate{fd}(T a, T x, T y, T z);`**

**`a`**: forgatás fokban; **`(x, y, z)`**: forgási tengely

pl. 45 fokos forgatás az *x*-tengely körül:

**`glRotated(45, 1.0, 0.0, 0.0)`**

**`void glTranslate{fd}(T x, T y, T z);`**

**`(x, y, z)`**: az eltolás vektora

pl.: *x*-tengely mentén 50 egységgel való eltolás

**`glTranslated(50, 0, 0)`**

**`void glScale{fd}(T x, T y, T z );`**

**`(x, y, z)`** skálázás mértéke a tengelyek mentén

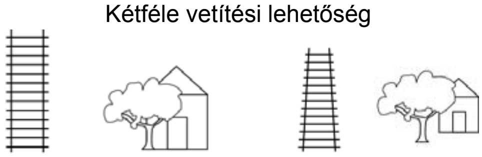
pl.: **`glScaled(0.5, 0.5, 0.5)`**

0.5-szörös uniform nagyítás

## Vetítési (projection) transzformáció (OpenGL)

```
glMatrixMode(GL_PROJECTION);
```

Kétféle vetítési lehetőség



ortografikus és perspektivikus

Megadjuk a látóteret is

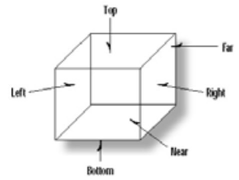
Végrehajtás:

új projekciómátrix =
projekciómátrix · specifikált mátrix

## Ortografikus vetítés (OpenGL)

```
void glOrtho(double left, double right,
        double bottom, double top,
        double near, double far);
```

Orthogonális (ortografikus) vetítés vágási terének megadása




2D eset:

```
void gluOrtho2D(
        double left, double right,
        double bottom, double top);
```

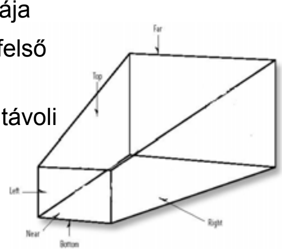
## Perspektív vetítés (OpenGL)

```
void glFrustum (double left, double right,
        double bottom, double top,
        double znear, double zfar);
```

**left, right**: a bal és jobb oldali vágósík *x* koordinátája

**bottom, top**: az alsó és felső vágósík *y* koordinátája

**znear, zfar**: a közeli és távoli vágósík *z* koordinátája.




**Nézőpont**:
az origó: (0, 0, 0)

## Perspektív vetítés (OpenGL)

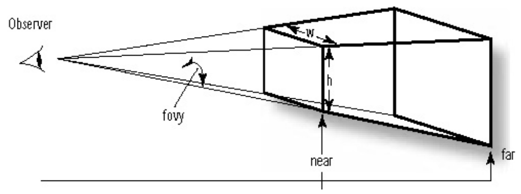
Szimmetrikus látótér megadása:

```
void gluPerspective (double fovy,
    double aspect, double near, double far);
```

**fovy**: a látótér szöge y irányban

**aspect**: w/h

**near, far**: a vágósíkok távolsága a megfigyelőtől



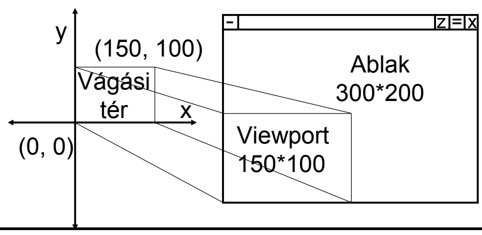

## Ablak (OpenGL)

2D-s leképzés az ablak egy téglalap alakú (viewport) részébe:

```
void glViewport(GLint x, GLint y,
    GLsizei width, GLsizei height);
```

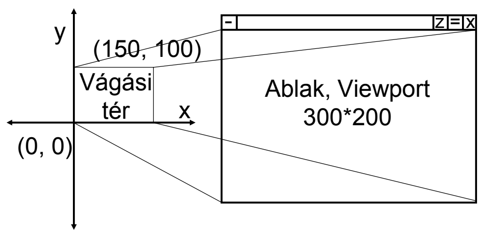
**x, y**: a viewport bal alsó sarka az ablakban,

**width, height**: a viewport mérete pixelben,




## Ablak (OpenGL)

**Alapértelmezés: (0, 0, winWidth, winHeight)**, ahol **winWidth** és **winHeight** az ablak méretei



## Perspektív vetítés (OpenGL)

Pl.: Módosítsuk viewport-ot és a vágási teret perspektív vetítésnél

```
void ChangeSize(GLsizei w, GLsizei h){
     GLfloat fAspect;
     if (h == 0)  h = 1; // ne ossszunk 0-val
  // az ablakon beállítjuk a viewport-ot
     glViewport(0, 0, w, h);
     fAspect = (GLfloat)w/(GLfloat)h;
  // vetítési mátrix
     glMatrixMode(GL_PROJECTION);
     glLoadIdentity();
  // vágási tér megadás, perspektív vetítés
     gluPerspective(60.0f, fAspect,
          1.0, 400.0);
     glMatrixMode(GL_MODELVIEW);
     glLoadIdentity();
}
```

## Transzformációs mátrixok (OpenGL)

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ w_0 \end{pmatrix} \rightarrow \begin{pmatrix} \text{Modelview} \\ \text{mátrix} \end{pmatrix} \rightarrow \begin{pmatrix} x_e \\ y_e \\ z_e \\ w_e \end{pmatrix} \rightarrow \begin{pmatrix} \text{Projekció} \\ \text{mátrix} \end{pmatrix} \rightarrow$$

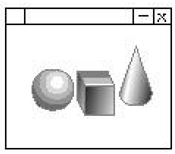
Eredeti koordináták — Szem koordináták

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ w_c \end{pmatrix} \rightarrow \begin{pmatrix} \text{Perspektív} \\ \text{osztás} \end{pmatrix} \rightarrow \begin{pmatrix} x_c / w_c \\ y_c / w_c \\ z_c / w_c \end{pmatrix} \rightarrow \begin{pmatrix} \text{Viewport} \\ \text{transzf.} \\ \text{mátrix} \end{pmatrix} \rightarrow$$

Vágási koordináták — Normalizát koordináták

## Transzformációs mátrixok (OpenGL)

$$\rightarrow \begin{pmatrix} x_c / w_c \\ y_c / w_c \\ z_c / w_c \end{pmatrix} \rightarrow \begin{pmatrix} \text{Viewport} \\ \text{transzf.} \\ \text{mátrix} \end{pmatrix} \rightarrow$$



Normalizát (inhomogén) koordináták

## Mátrix vermek (OpenGL)

Mátrix módok: **`GL_TEXTURE, GL_MODELVIEW, GL_COLOR, GL_PROJECTION`**

Minden mátrix mód számára van egy mátrix verem

Az aktuális mátrix a verem tetején lévő mátrix. A műveletek:

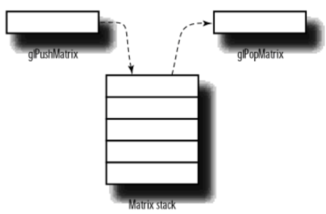
```
void glPushMatrix( void );
void glPopMatrix( void );
```

## Mátrix vermek (OpenGL)

**`glGet(GL_MAX_MODELVIEW_STACK_DEPTH)`**
(Microsoft: maximális mélység 32)

**`glGet(GL_MAX_PROJECTION_STACK_DEPTH)`**
(Microsoft: maximális mélység 2)

**`GL_STACK_OVERFLOW, GL_STACK_UNDERFLOW`**




**Alapállapot:**
egységmátrix,
**`GL_MODELVIEW`**

## Feladat (OpenGL)

Rajzoljuk meg egy dobókocka perspektivikus képét!