# Základy počítačové grafiky
# Přednáška 3

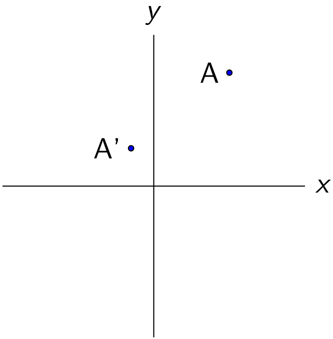
Martin Němec

VŠB-TU Ostrava

2025

# Transformace

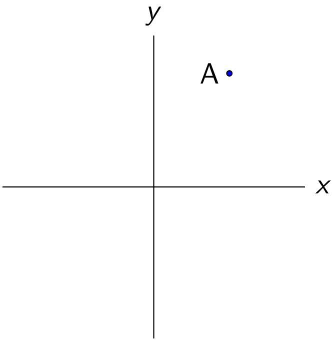
- Co jsou **transformace** intuitivně chápeme i bez VŠ.
- Hlavní teorie byla probrána v předmětu **Lineární algebra** (vektory, matice, transformace, skalární součin atd.).
- My z těchto znalostí budeme vycházet a prakticky je využívat a rozšiřovat.



# Skalár, bod

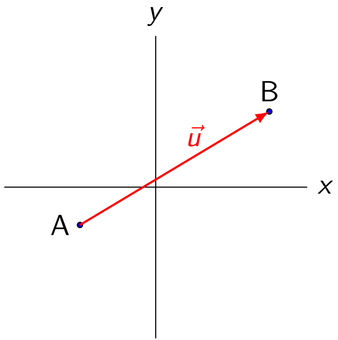
**Skalár** je veličina, která je definována pouze svou velikostí (ve fyzice to jsou např. hmotnost, objem, velikost atd.).

**Bod** základní bezrozměrný útvar, který reprezentujeme v prostoru pomocí trojice reálných čísel $A = [x, y, z]$. Tyto souřadnice udávají polohu v konkrétní souřadné soustavě. V různých souřadných soustavách může mít bod jiné souřadnice.



# Vektor

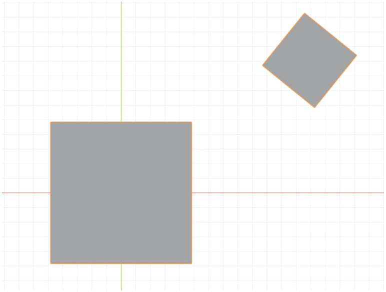
**Vektor** reprezentuje zjednodušeně pohyb z jednoho bodu do druhého (ve fyzice třeba síla a skládání sil).
Souřadnice vektoru tvoří uspořádaná $n$-tice čísel (složky vektoru).
Vektor značíme: $\vec{u} = (x, y, z)$.




- Má velikosti a směr;
- nemá pozici;
- definujeme $\vec{u} = B - A$;
- "jdi deset metru na jih".

# Transformace v PG

- Transformace je zobrazení, které každému bodu $A$ přiřadí jeho obraz, označíme jej bod $A'$.
- V euklidovské geometrii je **afinní transformace** (afinita) taková geometrická transformace, která zachovává linie a rovnoběžnost (ale ne nutně vzdálenosti a úhly).
- Kartézský souřadný systém - osy jsou na sebe navzájem kolmé, se stejným měřítkem.



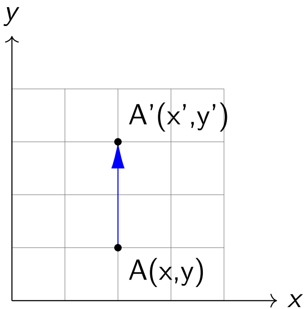
# Translace

**Posunutí** - $T(\vec{d})$, bodu $A[x, y]$ o vzdálenost $d_x$ a $d_y$ na příslušných osách (délky lze zapsat vektorem $\vec{d}$) provedeme tak, že k jednotlivým souřadnicím bodu $A$ přičteme příslušné hodnoty vektoru posunutí $d_x$ a $d_y$.

$$
\begin{aligned}
x' &= x + d_x\,, \\
y' &= y + d_y\,,
\end{aligned}
$$

resp.

$$A' = A + \vec{d}\,.$$



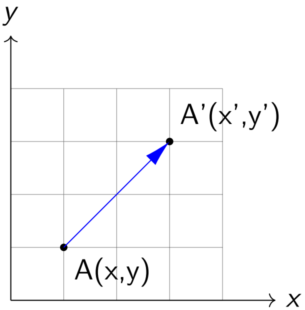
# Změna měřítka

**Změna měřítka** - $S(\vec{s})$, mění velikost objektu v jednotlivých osách. Změny měřítka lze dosáhnout vynásobením každé souřadnice bodu $A[x, y]$ měřítkem pro jednotlivé osy $s_x$ a $s_y$ (lze popsat vektorem $\vec{s}$), čímž získáme transformované souřadnice bodu $A'[x', y']$.

$$\begin{aligned} x' &= s_x x \,, \\ y' &= s_y y \,, \end{aligned}$$

obecně

$$A' = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} A \,.$$



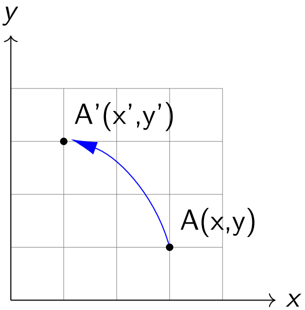
# Rotace

**Rotaci** - $R(\vec{\alpha})$, o úhel $\alpha$ chápeme jako pohyb bodu po kružnici kolem středu rotace. Rotaci o daný úhel lze provádět ve směru hodinových ručiček nebo proti směru hodinových ručiček.

$$
\begin{aligned}
x' &= x\cos(\alpha) - y\sin(\alpha)\,, \\
y' &= x\sin(\alpha) + y\cos(\alpha)\,.
\end{aligned}
$$

$$R_{ccw}(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

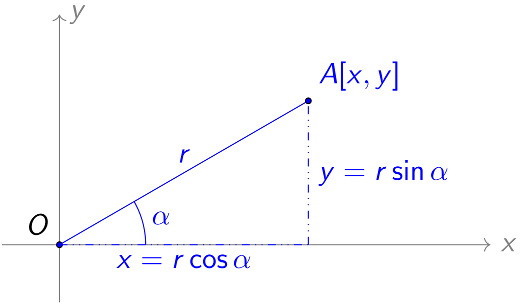
$$R_{cw}(\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$



# Odvození rotace

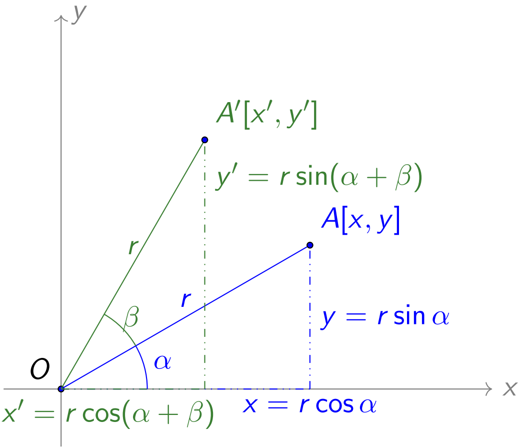
Připomeňme si definici sinu a kosinu v pravoúhlém trojúhelníku.

- Sinus úhlu se rovná poměru protilehlé odvěsny ku přeponě.
- Kosinus úhlu se rovná poměru přilehlé odvěsny ku přeponě.



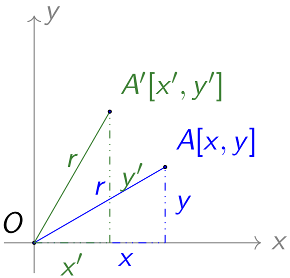

$$\sin(\alpha) = \frac{y}{r}, \qquad \cos(\alpha) = \frac{x}{r}$$

# Odvození rotace



# Odvození rotace

$$\sin(\alpha+\beta) = \sin\alpha\cdot\cos\beta + \cos\alpha\cdot\sin\beta$$
$$\cos(\alpha+\beta) = \cos\alpha\cdot\cos\beta - \sin\alpha\cdot\sin\beta$$

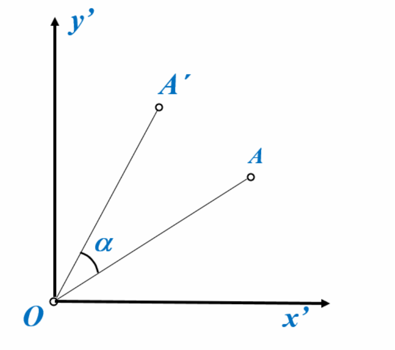


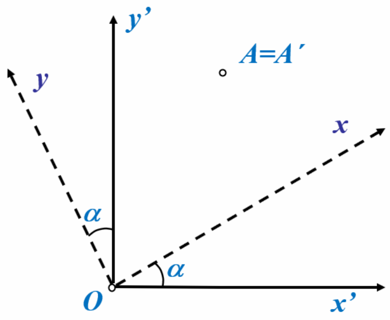

$$x' = r\cdot\cos(\alpha+\beta)$$
$$x' = r\cdot(\cos\alpha\cdot\cos\beta - \sin\alpha\cdot\sin\beta)$$
$$x' = r\cdot\cos\alpha\cdot\cos\beta - r\cdot\sin\alpha\cdot\sin\beta$$
$$x' = x\cdot\cos\beta - y\cdot\sin\beta$$

$$y' = r\cdot\sin(\alpha+\beta)$$
$$y' = r\cdot(\sin\alpha\cdot\cos\beta + \cos\alpha\cdot\sin\beta)$$
$$y' = r\cdot\sin\alpha\cdot\cos\beta + r\cdot\cos\alpha\cdot\sin\beta$$
$$y' = y\cdot\cos\beta + x\cdot\sin\beta$$

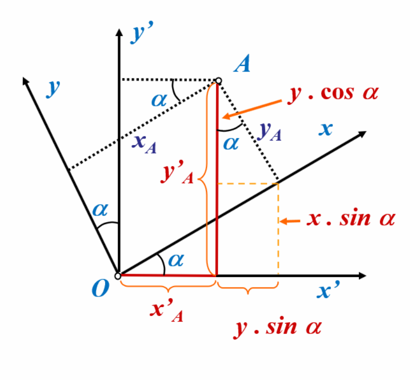
# Odvození rotace

Rotaci si můžeme znázornit jako rotaci bodu kolem středu souřadné soustavy, ale i jako rotaci souřadné soustavy.



Odvození rotace s využitím rotace souřadné soustavy.




$$x' = x \cdot \cos\alpha - y \cdot \sin\alpha$$
$$y' = y \cdot \cos\alpha + x \cdot \sin\alpha$$

# Transformace a maticový zápis

Translace

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

Rotace

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Změna měřítka

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

# Afinní transformace

Afinní transformace zachovává lineární vztahy mezi body (nezachovává obecně úhly ani délky). Zachovává:

- Rovnoběžnost přímek – přímky zůstávají přímkami, rovnoběžné přímky zůstávají rovnoběžnými.
- Poměr délek na přímce (kolinearita) - například střed úsečky zůstane středem i po transformaci.

Mezi afinní transformace patří (translace, rotace i změna měřítka).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

# Afinní transformace

Společně lze tedy všechny tři transformace zapsat obecně jako

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

zkráceně:

$$X' = \mathbf{A} \cdot X + \vec{d}$$

Pozor na variantu, kdy je bod reprezentován řádkovým vektorem:

$$\begin{bmatrix} x' & y' \end{bmatrix} = \begin{bmatrix} x & y \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} + \begin{bmatrix} d_x & d_y \end{bmatrix}$$

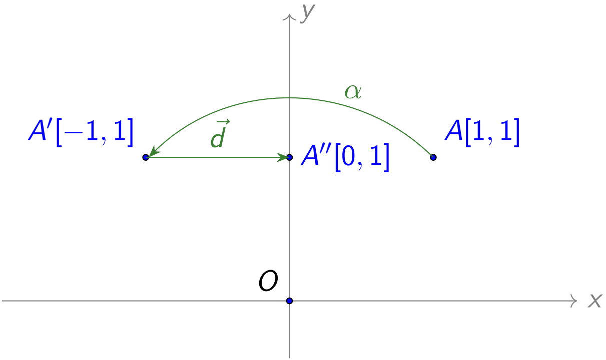
zkráceně:

$$X' = X \cdot \mathbf{B} + \vec{d}$$

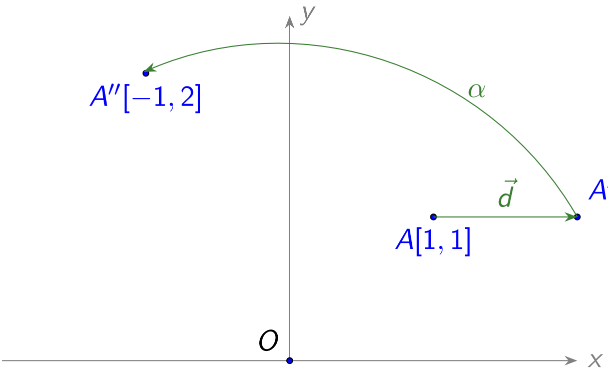
Jaký je vztah mezi oběma maticemi (transpozice matice)?

# Rotace a posun

Co když budeme chtít více transformací? Proveďte rotaci bodu $A = [1, 1]$ o úhel $\alpha = 90^\circ$ proti směru hodinových ručiček a posuňte o vektor $\vec{d} = (1, 0)$.



Co když transformace přehodíme? Proveďte posun bodu $A = [1, 1]$ o vektor $\vec{d} = (1, 0)$ a rotaci o úhel $\alpha = 90^\circ$ proti směru hodinových ručiček.



# Rotace kolem obecného bodu

Pokud je středem rotace obecný bod $S$, musíme nejprve provést posunutí tak, aby byl bod $S$ ve středu souřadné soustavy. Následně provést rotaci o daný úhel $\alpha$. A ještě provést zpětné posunutí.

$$X' = X + (-\vec{d})$$
$$X'' = \mathbf{A} \cdot X'$$
$$X''' = X'' + \vec{d}$$

zkráceně:

$$X' = (\mathbf{A} \cdot (X + (-\vec{d}))) + \vec{d}$$

# Skládání transformací

Pokud skládáme více transformací (nevíme dopředu jaké) platí obecně

$$
\begin{aligned}
X' &= \mathbf{A_1} \cdot X + \vec{d_1} \\
X'' &= \mathbf{A_2} \cdot X' + \vec{d_2} \\
&\dots \\
X^n &= \mathbf{A_n} \cdot X^{n-1} + \vec{d_n}
\end{aligned}
$$

zkráceně:

$$X^n = \mathbf{A_n} \cdot (\dots(\mathbf{A_2} \cdot (\mathbf{A_1} \cdot X + \vec{d_1}) + \vec{d_2})\dots) + \vec{d_n}$$

# Transformace v prostoru

Obecný zápis afinní transformace v prostoru.

$$X' = \mathbf{A} \cdot X + \vec{d}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a_{xx} & a_{yx} & a_{zx} \\ a_{xy} & a_{yy} & a_{zy} \\ a_{xz} & a_{yz} & a_{zz} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}$$

Translace i změna měřítka jsou jednoduché, jak to je s rotací?

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

# Rotace ve 3D

V prostoru umíme sestavit jednoduše rotaci kolem všech tří souřadnicových os.

Rotace kolem osy $x$ (roll) - souřadnice $x$ se nemění, body se otáčejí v rovině $yz$. Analogicky rotace kolem $y$ (pitch) a $z$ (yaw).

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

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

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

# Jak to je v počítačové grafice?

Doteď se jednalo pouze o opakování lineární algebry a středoškolské matematiky.

A jak je to tedy v počítačové grafice?

# Projektivní prostor

Myšlenkou je reprezentace bodu v prostoru o jednu dimenzi větším.

- Rozšíření o jednu dimenzi (expanze z 2D do 3D, popř. z 3D do 4D).
- Bod $X[x, y]$ v homogenních souřadnicích $X_h[wx, wy, w]$, kde $w \neq 0$.
- Bod se souřadnicemi $A_h = (x_h, y_h, w)$ má kartézské souřadnice $A_k = [x_h/w, y_h/w]$.
- Nejčastěji volíme homogenní souřadnici $w = 1$.

Uspořádanou čtveřici $[x_h, y_h, z_h, w]$ nazveme pravoúhlé homogenní souřadnice bodu $A$ v projektivním rozšíření $E_3$, jestliže pro souřadnice bodu $A$ bude platit:

$$x = \frac{x_h}{w}, y = \frac{y_h}{w}, z = \frac{z_h}{w}$$

# Homogenní souřadný systém

Srovnání zápisu translace, rotace a změny velikosti.

Kde je výhoda?

Afinní prostor
Kartézský souřadný systém

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

Projektivní prostor
Homogenní souřadný systém

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & d_x \\ a_{21} & a_{22} & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_h \\ y_h \\ 1 \end{bmatrix}$$

# Převod bodu

Co se děje s bodem $A[2,2]$, když při převodu volíme jinou homogenní souřadnici?

- $A[2,2]$, $w = 1$ ... $A_h = [2,2,1]$
- $A[2,2]$, $w = 2$ ... $A_h = [4,4,2]$
- $A[2,2]$, $w = 1/2$ ... $A_h = [1,1,1/2]$
- $A[2,2]$, $w = 10$ ... $A_h = [20,20,10]$

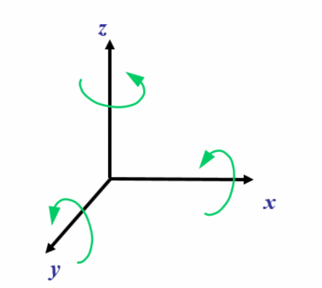
# Translation

Matici posunu (translace) v projektivním prostoru (homogenní souřadný systém) $\mathbf{T}(a, b, c)$ zapisujeme jako:

$$\mathbf{T}(a, b, c) = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

# Rotation

Matice rotace v projektivním prostoru $\mathbf{R}(\alpha)$:




$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\mathbf{R}_y(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\mathbf{R}_z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

# Scale

Změnu měřítka $\mathbf{S}(a, b, c)$ v projektivním prostoru (homogenní souřadný systém) zapisujeme jako:

$$\mathbf{S}(a, b, c) = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

# Rotace kolem obecného bodu

Jak se změní rotace kolem obecného bodu ve 2D při použití projektivního prostoru?

$$X' = \mathbf{M}_{(a,b)}(\alpha) \cdot X\,.$$

$$\mathbf{M}_{(a,b)}(\alpha) = \mathbf{T}(a,b)\mathbf{R}(\alpha)\mathbf{T}(-a,-b)\,.$$

To znamená

$$\mathbf{R}_{(a,b)}(\alpha) = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix}\,.$$

# Příklad na rotaci

Vypočtěte rotaci bodu $A = [5, 4]$ o úhel $\alpha = 90^\circ$ kolem bodu $C = [2, 3]$.

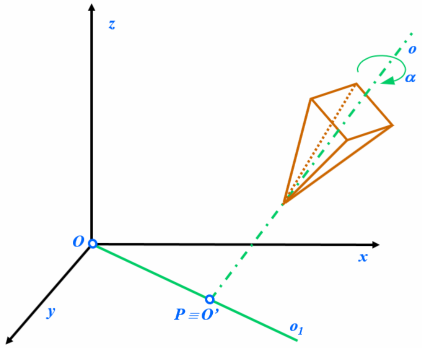
$$\mathbf{R}_{(a,b)}(\alpha) = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos 90 & -\sin 90 & 0 \\ \sin 90 & \cos 90 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}.$$

$$\mathbf{R}_{(a,b)}(\alpha) = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}.$$

$$A' = \mathbf{R}_{(a,b)}(\alpha) A = \begin{bmatrix} 0 & -1 & 5 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ 1 \end{bmatrix}.$$

Bod $A'$ má souřadnice $[1, 6]$.

Otočte objekt o úhel $\alpha$ okolo obecné osy $o$




Výsledná transformace $\mathbf{M} = \mathbf{T}_2 \cdot \mathbf{R}_5 \cdot \mathbf{R}_4 \cdot \mathbf{R}_3 \cdot \mathbf{R}_2 \cdot \mathbf{R}_1 \cdot \mathbf{T}_1$

# Nekonečno

Co se děje s bodem $A_h[2, 2, w]$, když budu měnit jenom homogenní souřadnici?

- $A[2, 2, 1]$, ... $A_k = [2, 2]$
- $A[2, 2, 1/10]$, $w = 2$ ... $A_k = [20, 20]$
- $A[2, 2, 1/100]$, $w = 1/2$ ... $A_h = [200, 200]$
- ...
- $A[2, 2, 0]$, $w = 10$ ... $A_h = [\infty, \infty]$

Dovolují popsat bod v nekonečnu?

# Násobení matic

Násobení matic není komutativní (komutativita $X \cdot Y = Y \cdot X$), co to znamená?

Příklad:

- Bod $A[1,1]$ posuneme o $\vec{v}(1,0)$ a provedeme rotaci proti směru hodinových ručiček o $90^\circ$. Jaké má transformovaný bod souřadnice?
- Bod $A[1,1]$ zarotujeme proti směru hodinových ručiček o $90^\circ$ a posuneme o $\vec{v}(1,0)$. Jaké má transformovaný bod souřadnice?

# Hlavní výhoda?

Jakou hlavní výhodu přináší použití projektivního prostoru v počítačové grafice?

# Transformace v OpenGL

Vytvoříme si matici 4x4 (transformaci) pomocí knihovny GLM (popřípadě sami).

```
// Construct identity matrix
glm::mat4 M = glm::mat4(1.0f);

M=glm::rotate(glm::mat4(1.0f),angle,glm::vec3(0, 1, 0));
M=glm::rotate(M, angle, glm::vec3(1, 0, 0));
M=glm::translate(glm::mat4(1), glm::vec3(0, 0, myView));
M=glm::scale(glm::mat4(1.0f), glm::vec3(0.5f));
```

# Transformace v OpenGL

Upravíme vertex shader tak, že si uvnitř vytvoříme uniformní proměnnou, kam budeme naši matici posílat. A následně každý vrchol touto proměnnou vynásobíme. Pozor na správné násobení!

```
#version 330
layout(location=0) in vec3 vp;
uniform mat4 modelMatrix;
void main () {
   //Why vec4(vp,1.0)?
   gl_Position = modelMatrix * vec4(vp,1.0);
}
```

# glGetUniformLocation

Vrací hodnotu (integer) reprezentující pozici uniformní proměnné.

```
GLint glGetUniformLocation(program,  name);
//program -  id shader programu
//name - nazev promenne

//Priklad
GLint id = glGetUniformLocation(sid, "modelMatrix");
```

Nyní pošleme na uniformní proměnnou modelMatrix ve vertex shaderu matici M.

```
//Render
glUseProgram(programID);
glUniformMatrix4fv(id, 1, GL_FALSE, value_ptr(M));
//location, count, transpose, *value
```

Lokální souřadná soustava

# Souřadné soustavy v OpenGL

Lokální souřadná soustava - vyznačení středu souřadné soustavy.

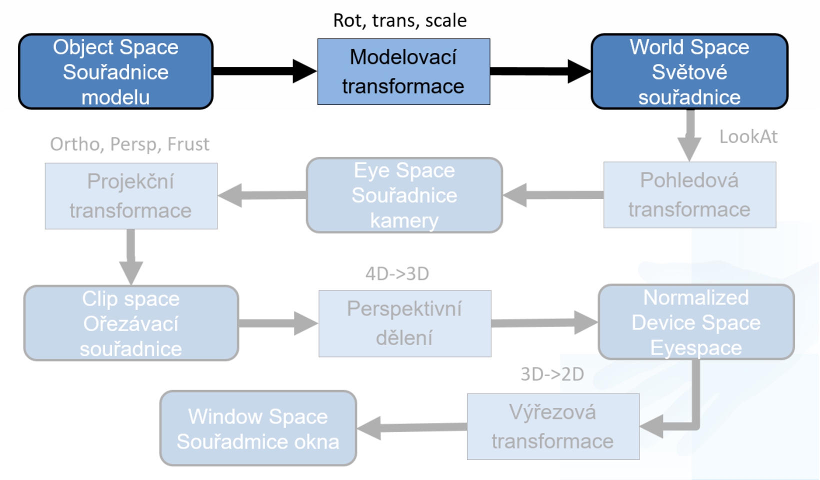
# Souřadné soustavy v OpenGL

Globální souřadná soustava - rozmístění modelů.

# Souřadné soustavy v OpenGL

Globální souřadná soustava - provedení transformací.

# Souřadné soustavy v OpenGL



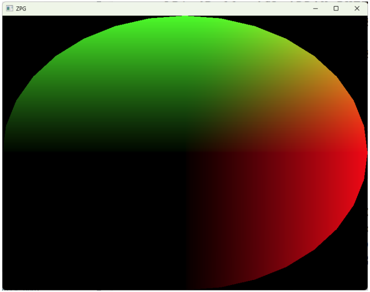
# Další modely (pozice+normála)

```
//vertex buffer object (VBO)
GLuint VBO = 0;
glGenBuffers(1, &VBO); // generate the VBO
glBindBuffer(GL_ARRAY_BUFFER, VBO);
glBufferData(GL_ARRAY_BUFFER, sizeof(sphere), sphere, GL_ST

//Vertex Array Object (VAO)
GLuint VAO = 0;
glGenVertexArrays(1, &VAO); //generate the VAO
glBindVertexArray(VAO); //bind the VAO
glEnableVertexAttribArray(0); //enable vertex attributes
glEnableVertexAttribArray(1);
glBindBuffer(GL_ARRAY_BUFFER, VBO);
glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE,
        6 * sizeof(float), (GLvoid*)0);
glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE,
        6 * sizeof(float), (GLvoid*)(3 * sizeof(float)));
```

# Další modely (pozice+normála)

```
glEnable(GL_DEPTH_TEST);
while (!glfwWindowShouldClose(window)) {
        // clear color and depth buffer
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        glUseProgram(shaderProgram);
        glBindVertexArray(VAO);
        // draw triangles
        glDrawArrays(GL_TRIANGLES, 0, 2880);
```

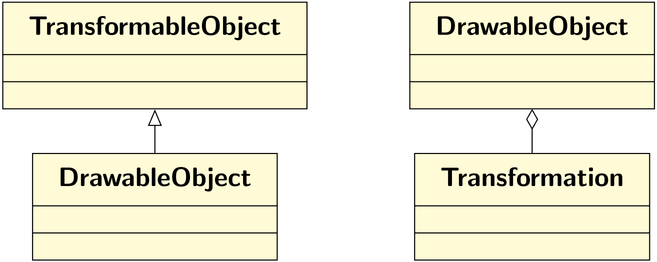


# Použití transformaci v projektu

Použít transformace v projektu lze více způsoby. Zaměříme se na následující dva:

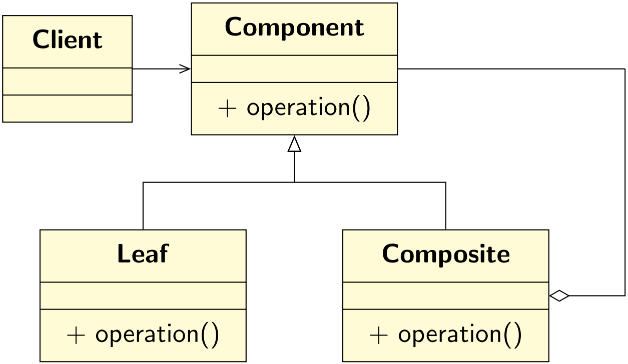
- DrawableObject je transformovatelný (dědičnost)
- DrawableObject má transformace (agregace/kompozice)

Obě varianty mohou být správné (reprezentují jiný přístup) a mají své výhody a nevýhody.



# Composite Pattern

Snahou je přistupovat stejným způsobem ke složeným (kompozitním) a jednoduchým objektům. Příkladem je souborový systém, kde máme složky a jednotlivé soubory.



# Závěr

**Dotazy?**