

UNIVERZITA PARDUBICE
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2010

Jiří Popelka

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Lineární transformace a promítání v počítačové grafice
Jiří Popelka

Bakalářská práce
2010

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří POPELKA**
Osobní číslo: **I07759**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Lineární transformace a promítání v počítačové grafice**
Zadávající katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

V úvodní části práce je nutné provést přehled problematiky lineárních transformací 2D a 3D grafiky (včetně skládání transformací) a promítání 3D do 2D s použitím homogenních souřadnic a transformačních matic. Dále je potřebné přehledově přiblížit problematiku vlivu parametrů transformačních matic na chování daných transformací a metod promítání. Cílem bakalářské práce je dále návrh a vytvoření programové aplikace jako učební pomůcky využitelné v předmětech počítačové grafiky. Účelem aplikace je demonstrování vlivu parametrů na transformace a na výsledek příslušné promítací metody. Aplikace musí umožňovat ovládání změn hodnot parametrů transformací a promítacích metod s okamžitou reakcí v zobrazení objektu v grafickém prostředí.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **ŽÁRA, J. A KOL.** Moderní počítačová grafika. Brno, Computer Press, 2004.
2. **Ing. Veselý, P,** Přednášky z předmětu Počítačová grafika

Vedoucí bakalářské práce:

RNDr. Miroslav Benedikovič
Katedra softwarových technologií

Datum zadání bakalářské práce: **15. ledna 2010**

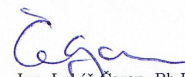
Termín odevzdání bakalářské práce: **14. května 2010**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2010

Prohlášení autora

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 16.5.2010

Jiří Popelka

Poděkování

Tímto bych rád poděkoval svému vedoucímu práce panu RNDr. Miroslavu Benedikovičovi za vedení práce a cenné rady při jejím vypracování.

Anotace

Práce je zaměřena na počítačovou grafiku, konkrétně na lineární transformaci ve 2D grafice a promítání v 3D grafice. Zahrnuje historii a současnost počítačové grafiky a metody uplatňující se při řešení požadavků. Součástí je i aplikace demonstrující popsané teoretické předpoklady.

Klíčová slova

lineární transformace, počítačová grafika, projekce

Title

Linear transformation and projection in computer graphics.

Annotation

Work is focused on computer graphics, namely on a linear transformation of 2D graphics and projection in 3D. Includes history and current computer graphics methods and applying the requirements of the solution. There is also an application demonstrating the theoretical assumptions described.

Keywords

linear transformation, computer graphics, projection

Obsah

1	Úvod	9
2	Transformace	10
2.1	Homogenní souřadnice	10
3	Geometrické transformace ve 2D	12
3.1	Posunutí	12
3.2	Otáčení.....	13
3.2.1	Kolem počátku.....	13
3.2.2	Kolem libovolného bodu	13
3.3	Změna měřítka	14
3.3.1	Souměrnost.....	14
3.4	Zkosení	16
4	Geometrické transformace ve 3D	17
4.1	Posunutí	17
4.2	Otáčení.....	17
4.3	Změna měřítka	18
4.4	Souměrnost.....	18
4.5	Zkosení	18
4.6	Skládání transformací.....	19
5	Promítání	20
5.1	Rovnoběžné promítání	20
5.2	Středové promítání	20
6	Delphi 7 jako vývojové prostředí	22
6.1	Object Pascal	23
7	Aplikace	24
7.1	Zadání.....	24
7.2	Implementace.....	24
7.2.1	Souhrn formulářů.....	24

7.2.2	Adresářová struktura	24
7.2.3	Funkce a procedury využitě pro transformace v 3D	27
7.2.4	Funkce a procedury využitě pro transformace v 2D	28
7.2.5	Společné funkce a procedury	29
7.3	Ovládání aplikace	29
8	Závěr.....	33
	Seznam použité literatury.....	34
	Přílohy	35
 Seznam obrázků		
	Obrázek 1 - Posun ve 2D.....	12
	Obrázek 2 - otočení ve 2D	13
	Obrázek 3 - změna měřítka ve 2D	14
	Obrázek 4 - souměrnost kolem osy x.....	15
	Obrázek 5 - souměrnost kolem osy y.....	15
	Obrázek 6 - středová souměrnost.....	15
	Obrázek 7 - zkosení ve 2D	16
	Obrázek 8 - jednobodové středové promítání	21
	Obrázek 9 - dvoubodové středové promítání	21
	Obrázek 10 - třibodové středové promítání.....	21
	Obrázek 11 - Delphi 7	22
	Obrázek 13 - Třída polygonu ve 2D	25
	Obrázek 14 - Třída tělesa ve 3D	26
	Obrázek 15 - Aplikace - Úvodní formulář	29
	Obrázek 16 - Aplikace – dynamická transformace 2D.....	30
	Obrázek 17 - Aplikace - Statická transformace 2D.....	31
	Obrázek 18 - Aplikace - Dynamická transformace 3D	32
	Obrázek 19 - Aplikace - Statická transformace 3D.....	32

1 Úvod

Závěrečná práce je zaměřená na způsoby provádění transformací ve vektorové grafice, především na maticové transformace a způsoby promítání 3D objektů na dvourozměrnou obrazovku. Téma jsem si vybral, neboť mi přišlo zajímavé vytvářet aplikaci pro demonstrační či učební účely.

V této práci postupuji od teorie lineární transformace, až k její aplikaci, která ji názorně demonstruje. V druhé kapitole se zabývám transformací jako takovou a zaváděním homogenních souřadnic. V následující kapitole si představíme základní transformační operace pro dvourozměrný prostor. Čtvrtá kapitola je pak rozšířením do třetí dimenze a znázorněním transformačních matic pro tento prostor. Je zakončena vysvětlením skládání transformací pro vytvoření četných možností operací s transformovanými objekty a základy pro promítání objektu na obrazovku počítačů obsahuje pátá kapitola.

Od šesté kapitoly je práce zaměřena na návrh aplikace. Je zde krátce představeno vývojové prostředí, ve kterém aplikace vznikla a programovací jazyk, kterým byla sepsána. V poslední části představuji aplikaci samotnou a vysvětluji, jakým způsobem ji lze ovládat.

V závěru své bakalářské práce se zamýšlím nad možností rozšíření aplikace v dalších oblastech promítání objektu.

2 Transformace

Transformace geometrických objektů jsou operace, které jsou v počítačové grafice jedny z nejvíce používaných. Tyto transformace můžeme rozdělit do dvou skupin. První skupinu nazýváme **lineární transformace**, mezi které řadíme zejména: **posun, otočení, změnu měřítka a zkosení**. Dále mohou vzniknout operace vzniklé spojením výše jmenovaných transformací. Druhou skupinu nazýváme **nelineární transformace**. Nelineární transformace se nejčastěji v počítačové grafice využívá u warpingu obrazu. Speciální transformace, které se aplikují k převádění objektů z vícerozměrného prostoru do prostoru s méně rozměry, nazýváme **projekce**. Projekci nejčastěji využíváme k převodu trojrozměrné scény na stínítko obrazovky.

Pod geometrickou transformací rozumíme, možnost realizace na všechny body objektu. Tím docílíme změnu jeho současné polohy. Velmi často je tato operace využívána v počítačové animaci v trojrozměrném prostoru, při psaní textu na křivku v desktop publishing (DTP) nebo při pohybu objektu po určité dráze. Pokud chceme výhodněji reprezentovat objekt, abychom ho mohli dále určitým způsobem zpracovávat, například chceme-li vypočítat objem objektu, využít ho pro projekci nebo viditelnost, musíme podrobit transformaci souřadnicovému systému.

Pokud podrobíme transformaci bod P se souřadnicemi $[X, Y]$ ve dvou rozměrech, respektive $[X, Y, Z]$ v rozměrech třech, v kartézském systému souřadnic, vznikne bod P' , který má souřadnice $[X', Y']$, resp. $[X', Y', Z']$. Jestliže aplikujeme transformaci na celý objekt, znamená to, že podrobíme této transformaci, všechny body, které daný objekt tvoří.

2.1 Homogenní souřadnice

Abychom si značně ulehčili výpočty transformací, zavádíme homogenní souřadnice, díky kterým můžeme lehce reprezentovat všechny body objektu. Nejčastěji používané lineární transformace můžeme díky homogenním souřadnicím vyjádřit pomocí jedné jediné matice. V nehomogenních souřadnicích toto vyjádření není možné. Jiné využití matice, při transformaci v homogenních souřadnicích, je tzv. perspektivní promítání. Pokud chceme využít skládané transformace, o kterých jsem se zmínil výše, je realizace pomocí matic velmi výhodná. Skládání matic se totiž realizuje jako násobení matic, které je implementací mnoha knihoven pro práci s maticemi. Stejně tak jako skládání transformací je realizováno pomocí násobení matic, je inverzní transformace zastoupena inverzní maticí atd.

Bod P s kartézskými souřadnicemi $[X, Y, Z]$ ve třech rozměrech má uspořádanou čtveřici čísel $[x, y, z, \omega]$, kterou nazýváme pravoúhlé homogenní souřadnice, pro které platí:

$$X = \frac{x}{\omega}, Y = \frac{y}{\omega}, Z = \frac{z}{\omega}, \omega \neq 0.$$

Homogenní souřadnice jednoznačně určují bod P . Souřadnice ω je tzv. váhou bodu. Transformovaný bod P' s kartézskými souřadnicemi $[X', Y', Z']$ má homogenní souřadnice $[x', y', z', \omega']$.

Nejčastěji volíme váhu bodu $\omega = 1$, čímž dosáhneme, že homogenní souřadnice bodu jsou $[X.\omega, Y.\omega, Z.\omega, 1]$.

Lineární transformaci bodu $P = [x, y, z, \omega]$ na bod $P' = [x', y', z', \omega']$ zapisujeme pomocí matice o rozměrech 4×4 a označujeme \mathbf{A} , přičemž v posledním sloupci jsou nuly a na posledním řádku v tomto sloupci píšeme jedničku:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & 1 \end{bmatrix}.$$

Pro převod souřadnic vynásobíme transformační matici s homogenními souřadnicemi bodu P :

$$P' = [x' \quad y' \quad z' \quad \omega'] = PA = [x \quad y \quad z \quad \omega] \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & 1 \end{bmatrix}.$$

Pro transformaci bodu $P = [x, y, \omega]$ na bod $P' = [x', y', \omega']$ ve dvou rozměrech používáme podobnou matici:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{bmatrix}.$$

Pro převod souřadnic platí opět:

$$P' = [x' \quad y' \quad \omega'] = PA = [x \quad y \quad \omega]\mathbf{A}.$$

3 Geometrické transformace ve 2D

3.1 Posunutí

První z hlavních geometrických transformací je posunutí (move), neboli translace. Translaci bodu P vyjádříme vektorem posunutí:

$$\vec{p} = (X_T, Y_T) = (X' - X, Y' - Y).$$

Po provedení transformace dostaneme bod P' , který má souřadnice:

$$X' = X + X_T$$

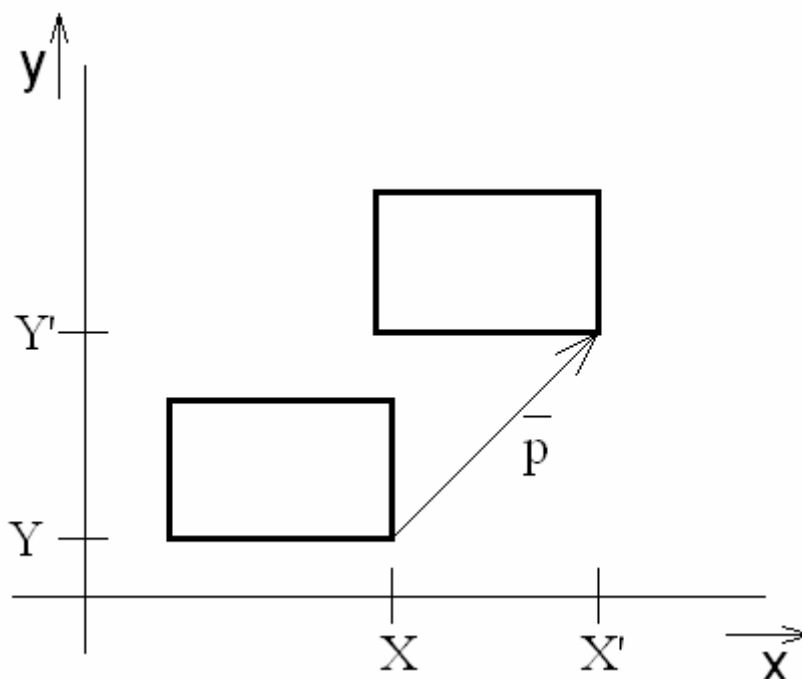
$$Y' = Y + Y_T.$$

Při použití maticové transformace se uplatňují pouze koeficienty a_{31} a a_{32} . Matice posunutí má potom tvar:

$$\mathbf{A}_M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ X_t & Y_t & 1 \end{bmatrix}.$$

Inverzní matice je ve tvaru:

$$\mathbf{A}_M^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -X_t & -Y_t & 1 \end{bmatrix}.$$



Obrázek 1 - Posun ve 2D

3.2 Otáčení

3.2.1 Kolem počátku

Jako druhou geometrickou transformaci si popíšeme otáčení, kterou jinak nazýváme rotace. Pokud aplikujeme otočení na bod P , dostaneme bod P' , který je otočen o úhel α kolem počátku $O = [0, 0]$ se souřadnicemi:

$$X' = X \cdot \cos \alpha - Y \cdot \sin \alpha$$

$$Y' = Y \cdot \sin \alpha + X \cdot \cos \alpha .$$

Pokud chceme aplikovat otočení pomocí transformace matic, definujeme si matici otočení ve tvaru:

$$\mathbf{A}_R = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

Inverzní matice je ve tvaru:

$$\mathbf{A}_R^{-1} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

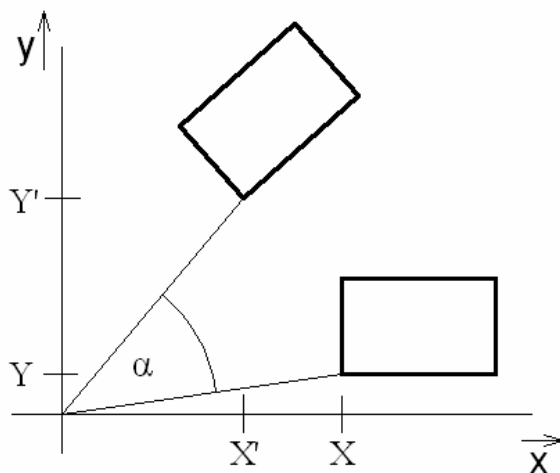
3.2.2 Kolem libovolného bodu

Chceme-li otočit geometrický objekt kolem libovolného tělesa, například kolem jeho středu, musíme použít skládání transformací. Nejprve posuneme střed $S = [S_x, S_y]$ do počátku $O = [0, 0]$. Poté provedeme rotaci a následně znovu translaci zpátky do původního bodu. Pro bod P' potom platí:

$$P' = P \cdot \mathbf{A} ,$$

kde:

$$\mathbf{A} = \mathbf{A}_M \cdot \mathbf{A}_R \cdot \mathbf{A}_M^{-1} .$$



Obrázek 2 - otočení ve 2D

3.3 Změna měřítka

Změna měřítka, neboli scale, je v podstatě změna velikosti objektu. Tato transformace se provádí ve směru os. Je-li hodnota koeficientu pro změnu měřítka v rozmezí 0 až 1, jedná se zmenšení objektu, na který se transformace uplatňuje. V případě, že tento koeficient je větší než jedna, dochází ke zvětšení transformovaného objektu. Záporná hodnota koeficientu udává zvětšení nebo zmenšení v opačném směru.

Pro změnu měřítka bodu P se používají rovnice:

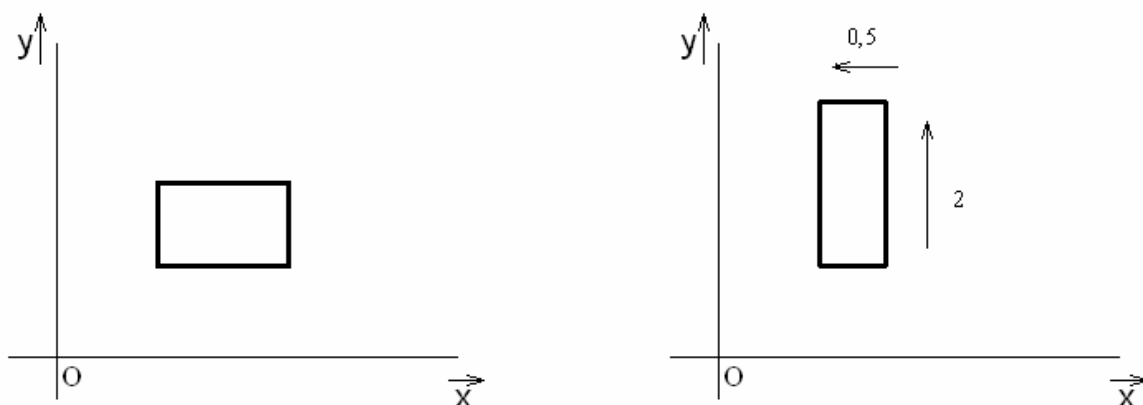
$$X' = S_x \cdot X$$

$$Y' = S_y \cdot Y,$$

kde S_x je koeficient změny měřítka ve směru souřadnicové osy x a S_y koeficient ve směru osy y.

Matice pro změnu měřítka má tvar:

$$A_S = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$



Obrázek 3 - změna měřítka ve 2D

3.3.1 Souměrnost

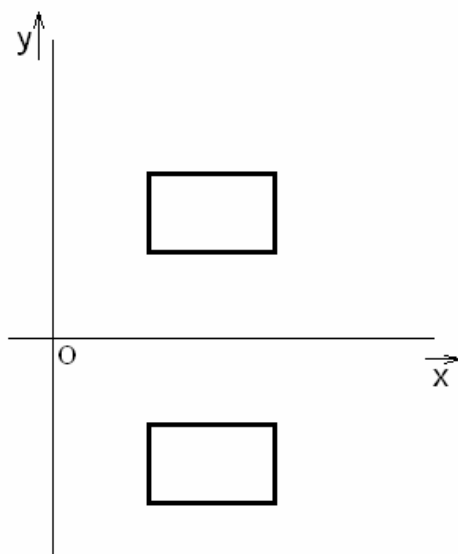
Souměrnost je zvláštním případem změny měřítka. Koeficienty pro změnu nabývají hodnoty -1 nebo 1. V daných kombinacích těchto koeficientů, můžeme rozdělit souměrnost na dvě skupiny. Souměrnost kolem počátku, neboli středová souměrnost. Druhou možností je osová souměrnost, kde je objekt souměrný podle souřadnicové osy x nebo podle souřadnicové osy y.

Koeficienty pro jednotlivé souměrnosti jsou:

Souměrnost podle osy x

$$S_x = 1$$

$$S_y = -1.$$

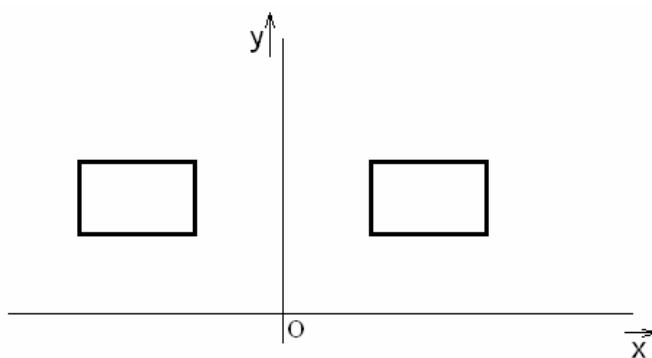


Obrázek 4 - souměrnost kolem osy x

Souměrnost podle osy y

$$S_x = -1$$

$$S_y = 1.$$

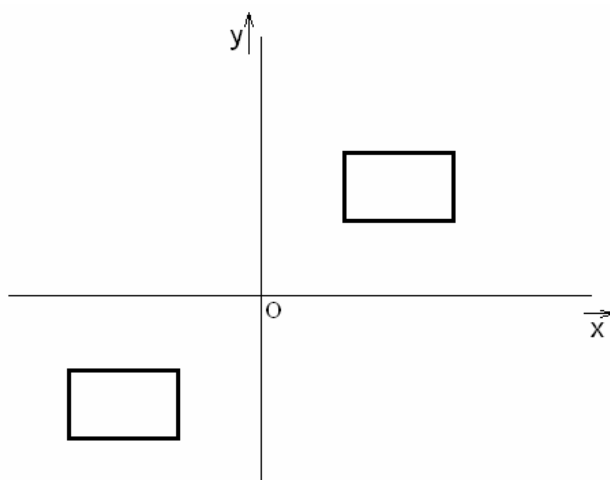


Obrázek 5 - souměrnost kolem osy y

Středová souměrnost

$$S_x = -1$$

$$S_y = -1.$$



Obrázek 6 - středová souměrnost

3.4 Zkosení

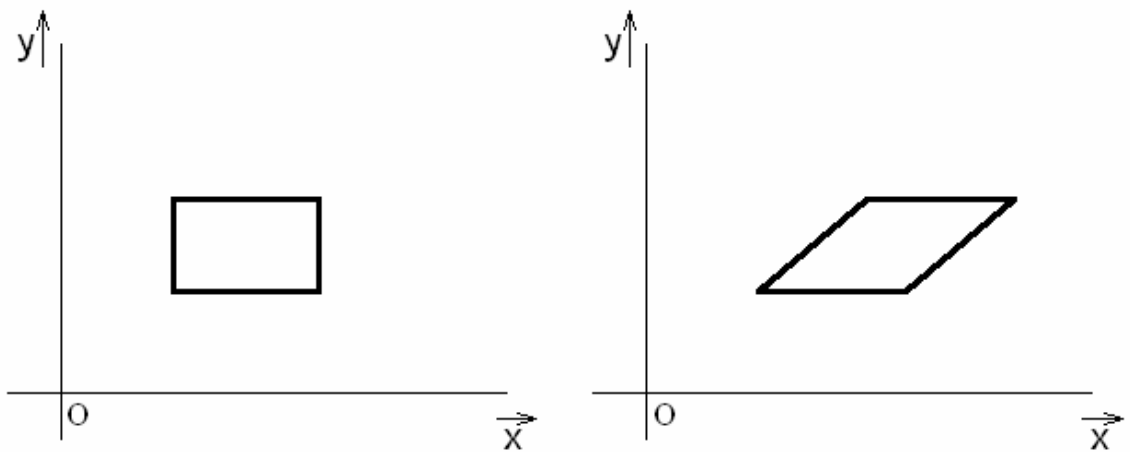
Zkosení, neboli shear provádíme ve směru souřadnicových os. Míru zkosení udávají koeficienty pro zkosení S_x a S_y . Aplikováním transformace zkosení na bod P , dostaneme bod P' , který má souřadnice:

$$X' = X + S_x \cdot Y$$

$$Y' = Y + S_y \cdot X.$$

Matice zkosení je tedy ve tvaru:

$$\mathbf{A}_{Sh} = \begin{bmatrix} 1 & S_y & 0 \\ S_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$



Obrázek 7 - zkosení ve 2D

4 Geometrické transformace ve 3D

V podstatě se transformace ve třech prostorech, provádí obdobně jako ve dvou prostorech. Namísto matice 3 x 3 používáme matice o rozměrech 4 x 4.

4.1 Posunutí

Stejně jako ve dvou rozměrech, i zde je posunutí dáno vektorem posunutí, kterému je přidána další souřadnice. Pro vektor posunutí platí:

$$\vec{p} = (X_t, Y_t, Z_t) = (X' - X, Y' - Y, Z' - Z).$$

Transformační matice je ve tvaru:

$$\mathbf{A}_M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ X_t & Y_t & Z_t & 1 \end{bmatrix}.$$

Posunutím bodu P , dostaneme bod P' o souřadnicích:

$$X' = X + X_t$$

$$Y' = Y + Y_t$$

$$Z' = Z + Z_t.$$

4.2 Otáčení

Otáčení ve 3D provádíme kolem základních souřadnicových os x , y , z . Pro transformaci nám bohužel nestačí jedna matice. Matice otočení se liší osou, podle které chceme otáčet.

Matice otočení kolem osy x je ve tvaru:

$$\mathbf{A}_{Rx} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Matice otočení kolem osy y má tvar:

$$\mathbf{A}_{Ry} = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

A pro osu z je ve tvaru:

$$\mathbf{A}_{Rz} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 & 0 \\ -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

4.3 Změna měřítka

Může být opět prováděna ve směru všech os a je dána hodnotou koeficientů změny měřítka S_x , S_y , S_z .

Matice pro změnu měřítka je:

$$\mathbf{A}_{Sc} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

4.4 Souměrnost

Stejně jako ve dvou prostorech, tak i ve 3D je souměrnost zvláštní případem změny měřítka. Souměrnost má tři variace. Souměrnost podél os, souměrnost podél roviny a souměrnost kolem počátku.

Hodnoty koeficientu pro změnu měřítka v případech souměrnosti:

Středová souměrnost	$S_x = -1$	$S_y = -1$	$S_z = -1$
Souměrnost podél osy x	$S_x = 1$	$S_y = -1$	$S_z = -1$
Souměrnost podél osy y	$S_x = -1$	$S_y = 1$	$S_z = -1$
Souměrnost podél osy z	$S_x = -1$	$S_y = -1$	$S_z = 1$
Souměrnost podél roviny xy	$S_x = 1$	$S_y = 1$	$S_z = -1$
Souměrnost podél roviny xz	$S_x = 1$	$S_y = -1$	$S_z = 1$
Souměrnost podél roviny yz	$S_x = -1$	$S_y = 1$	$S_z = 1$

4.5 Zkosení

Zkosení, neboli deformace objektu ve třech rozměrech, se provádí vždy ve směru rovin xy nebo xz nebo yz. Míru zkosení nám udávají koeficienty zkosení S_x , S_y , S_z . Matice transformace zkosení ve 3D má následující tvar.

Pro rovinu xy:

$$\mathbf{A}_{Sh_{xy}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ S_x & S_y & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pro rovinu xz:

$$\mathbf{A}_{Sh_{xz}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ S_X & 1 & S_Z & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pro rovinu yz:

$$\mathbf{A}_{Sh_{yz}} = \begin{bmatrix} 1 & S_Y & S_Z & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

4.6 Skládání transformací

Často potřebujeme vykonat transformaci, která není mezi základními operacemi. Například otočení kolem libovolného bodu, zvětšení měřítka polygonu vzhledem ke středu a spousta dalších operací s objektem. Abychom docílili požadovaných výsledků, musíme použít skládání transformací. Nejedná se o nic jiného, než o násobení matic v daném pořadí. Výsledkem této operace je transformační matice, kterou nakonec aplikujeme na daný objekt. Matice jednotlivých transformací násobíme zprava, abychom dodrželi posloupnost operací.

Chceme-li například změnit měřítko objektu, vzhledem k jeho středu, musíme provést následující transformace:

- posun objektu o vzdálenost středu do počátku souřadnicového systému,
- aplikace změny měřítka,
- posun objektu zpět na původní souřadnice (inverzní transformace posunu).

Výsledná transformační matice bude mít podobu:

$$\mathbf{A} = \mathbf{A}_M \cdot \mathbf{A}_S \cdot \mathbf{A}_M^{-1}.$$

5 Promítání

Promítání neboli projekce, je nástroj pro zobrazování vícerozměrných objektů do prostoru o méně rozměrech. Nejčastějším převodem bývá převod trojrozměrných objektů na zobrazovací zařízení obrazovek, které zobrazují pouze rozměry dva. Při tomto převodu však nastává zkreslení informace o objektech.

Existuje mnoho metod, jak takové převádění realizovat. Tímto problémem se zabývá deskriptivní geometrie. V počítačové grafice se nejčastěji používají dvě metody promítání. **Rovnoběžné** a **středové** promítání na jednu průmětnu. Při promítání musíme samozřejmě brát v potaz, že těleso má i další vlastnosti. Například pokud se nejedná o mřížkový model, musíme uvažovat, které hrany jsou viditelné, popřípadě jakou barvu mají stěny.

5.1 Rovnoběžné promítání

Tento způsob promítání se nejčastěji využívá v technickém kreslení, neboť zachovává rovnoběžnost stran. Rovnoběžné promítání dělíme na **pravouhlé**, které se nejčastěji využívá v počítačové grafice. Tj. takové promítací paprsky, které svírají s průmětnou úhel 90° , a **kosouhlé** pro ostatní úhly.

Pokud chceme promítnout těleso do roviny xy , stačí zanedbat souřadnice pro osu z . Transformační matice vypadá následovně:

$$P_{xy} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Tímto získáme půdorys. Pokud chceme jiný pohled, musíme nejdříve zvolit odpovídající transformaci pro otočení a posun objektů, vynásobíme s maticí P_{xy} . Výslednou transformační matici aplikujeme na všechny objekty.

5.2 Středové promítání

Jinak řečeno perspektivní promítání, se používá především v oblastech, které se věnují zachycení objektů reálného světa. Obrazy středového promítání jsou velmi podobné těm, které lidské oko dokáže zachytit. Proto se s velkým uplatněním využívají především v architektuře nebo virtuální realitě. Při tomto způsobu promítání, vycházejí všechny promítací paprsky z jednoho bodu, tzv. středu promítání. Tím je zapříčiněno, že není zachována rovnoběžnost objektu. Je-li objekt vzdálenější od středu promítání, je jeho průmět menší.

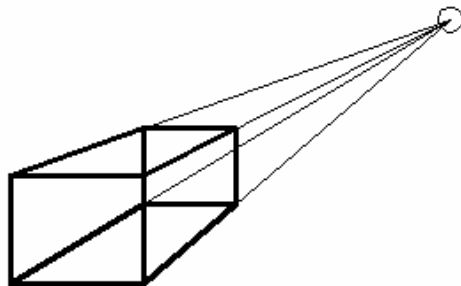
Matice pro perspektivní promítání vypadá následovně:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/d \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

kde d , je vzdálenost středu promítání od průmětny.

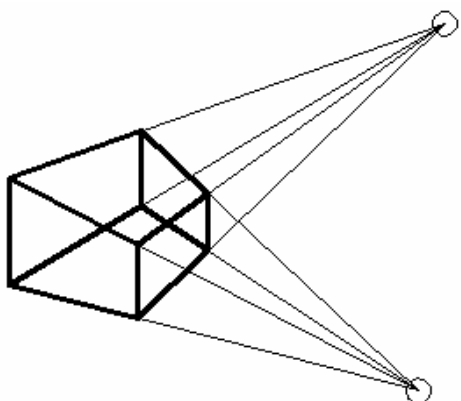
Rozlišují se tři případy pro perspektivní promítání v závislosti na jeho středu:

- **jednobodové** – průmětna protíná jednu souřadnicovou osu a všechny úsečky kolmé na průmětnu míří do jediného bodu.



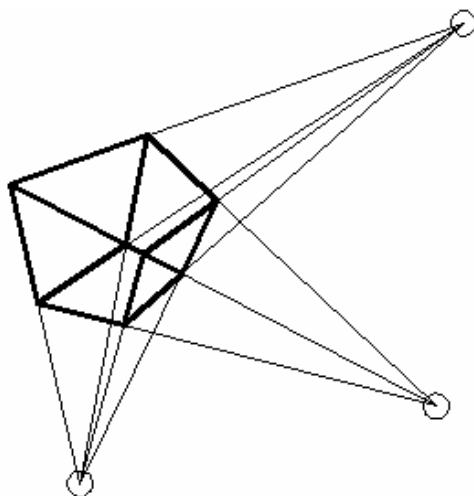
Obrázek 8 - jednobodové středové promítání

- **dvoubodové** – protíná dvě souřadnicové osy a úsečky míří do dvou bodů.



Obrázek 9 - dvoubodové středové promítání

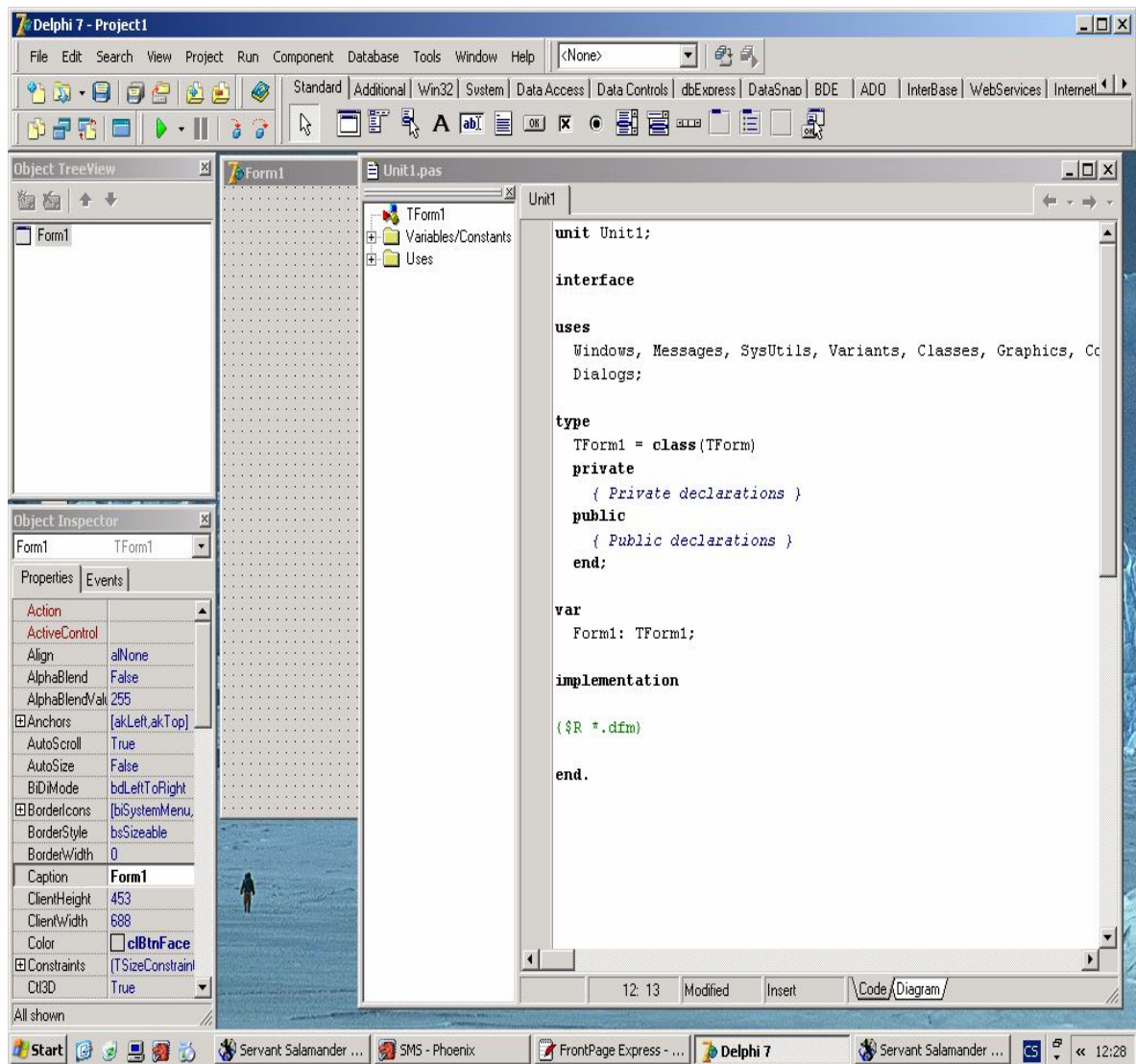
- **tříbodové** – protíná všechny tři souřadnicové osy a úsečky míří do tří hlavních bodů.



Obrázek 10 - tříbodové středové promítání

6 Delphi 7 jako vývojové prostředí

Pro realizaci demonstračního programu lineární transformace jsem se rozhodl použít Borland Delphi 7. Toto vývojové prostředí využívá vyššího programovacího jazyka Object Pascal. Syntaxe Object Pascalu vychází z Pascalu a je velmi příznivý k objektově orientovanému programování. Stejně jako většina vývojových prostředí, má i delphi 7 nástroje pro diagnostiku ladění vytvářených aplikací i jejich zdrojových kódů. Pro vývoj grafických aplikací využíváme komponenty, které jsou už přednastavené, nebo můžeme vytvořit svojí vlastní. Ačkoliv je práce v tomto vývojovém prostředí intuitivní a snadná, žádná z verzí se nedočkala stálého vydání.



Obrázek 11 - Delphi 7

6.1 Object Pascal

Jak už bylo řečeno, jeho syntaxe vychází z jazyka Pascal a umožňuje využívání objektů při objektově orientovaném programování. Nerozlišuje velká a malá písmena v zápisu syntaxe, čímž má značně menší možnosti v použití námi zvolených proměnných. Zároveň klade na identifikátory podmínky. Pro jejich názvy se mohou použít pouze písmena, číslice a podtržítka, přičemž názvy nesmějí obsahovat české znaky a první musí následovat písmeno.

Stejně jako ostatní programovací jazyky, také Object Pascal má vyhrazená klíčová slova, která nemůžeme zvolit jako identifikátory. Na konci každého příkazu, kromě výjimky před slovem *else*, musí být středník. Pro lepší orientaci ve zdrojovém kódu, vznikly určité zásady, které by se měly dodržovat. Například názvy identifikátorů by měly být mnemotechnické, dle jejich významu. Dále odsazovat vnořené bloky nebo psaní každého příkazu na nový řádek. Většinu těchto pravidel má uživatel zafixovanou a dělá je automaticky, i když při vývoji malých aplikací na ně neklademe takový důraz.

7 Aplikace

V této kapitole začnu popisovat demonstrační aplikaci, která je nezbytnou součástí pro vypracování bakalářské práce. Stručně rozeberu její návrh a poté způsob orientace v aplikaci a ovládání transformací.

7.1 Zadání

Aplikace musí umožňovat demonstraci vlivu parametrů transformace a výsledek příslušné promítací metody. Aplikace musí dále zprostředkovat ovládání změn hodnot parametrů v transformaci, s okamžitou reakcí v zobrazení objektu v grafickém prostředí.

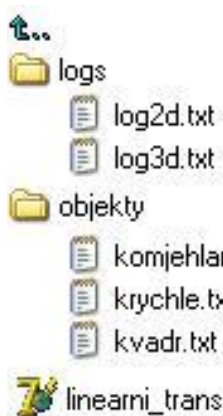
7.2 Implementace

Pro realizaci aplikace jsem použil Delphi 7 od společnosti Borland, kterou jsem představil v kapitole 6 Vývojové prostředí. Aplikaci tvoří celkem 7 formulářů, na kterých jsou umístěny různé komponenty standardně obsažené v Delphi 7.

Souhrn formulářů:

Form1	Úvodní formulář po spuštění aplikace
Form2	Hlavní formulář pro práci s transformacemi ve 2D
Form3	Hlavní formulář pro práci s transformacemi ve 3D
Form4	Formulář pro nastavování rozsahu os
Form5	Formulář pro nastavování barev
Form6	Formulář obsahující informace o zobrazených objektech
Form7	Formulář pro zobrazení log souborů.

7.2.1 Adresářová struktura



logs	- adresář pro zapisování log souborů
log2d.txt	- historie provedených transformací ve 2D
log3d.txt	- historie provedených transformací ve 3D
objekty	- adresář s předdefinovanými tělesy v text. souborech
komjehlan.txt	- definice bodů pro komolý jehlan
krychle.txt	- definice bodů pro krychli
kvadr.txt	- definice bodů pro kvádr
linearni_transformace.exe	- spouštěcí soubor aplikace

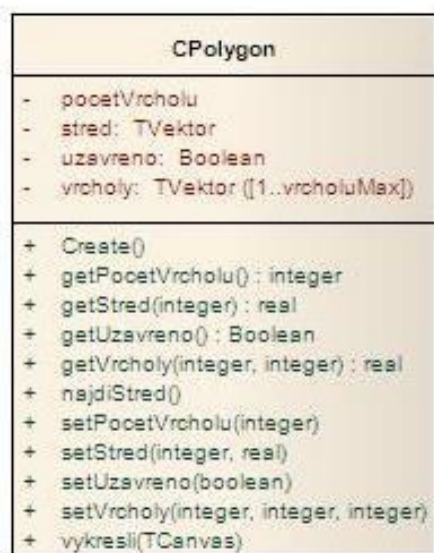
Obrázek 12 - Adresářová struktura

Toto je samozřejmě popis adresářů pro výslednou aplikaci. Při vývoji vznikají i jiné soubory. Každý formulář má svůj unit se zdrojovými kódy. Každý unit pak má svoji zálohu atd.

Seznam tříd a jejich metod

Třída CPolygon

- je třídou pro vytvoření objektu zastupující polygon, na který budeme chtít aplikovat transformace.



Obrázek 13 - Třída polygonu ve 2D

Metody

Create

- konstruktor objektu

najdiStred

- spočítá souřadnice středu zvoleného polygonu

setStred(index:integer;hodnota:real)

- nastaví hodnotu atributu stred

getStred(index:integer)

- vrátí hodnotu atributu stred

vykresli(Canvas: TCanvas)

- vykreslí polygon zadaný body

setUzavreno(stav:boolean)

- nastaví hodnotu atributu uzavreno

getUzavreno

- vrátí hodnotu atributu uzavreno

setPocetVrcholu(cislo: integer)

- nastaví hodnotu počtu vrcholů do atributu

getPocetVrcholu

- vrátí počet vrcholů

setVrcholy(cisloVrcholu,hodnotaX,hodnotaY:integer)

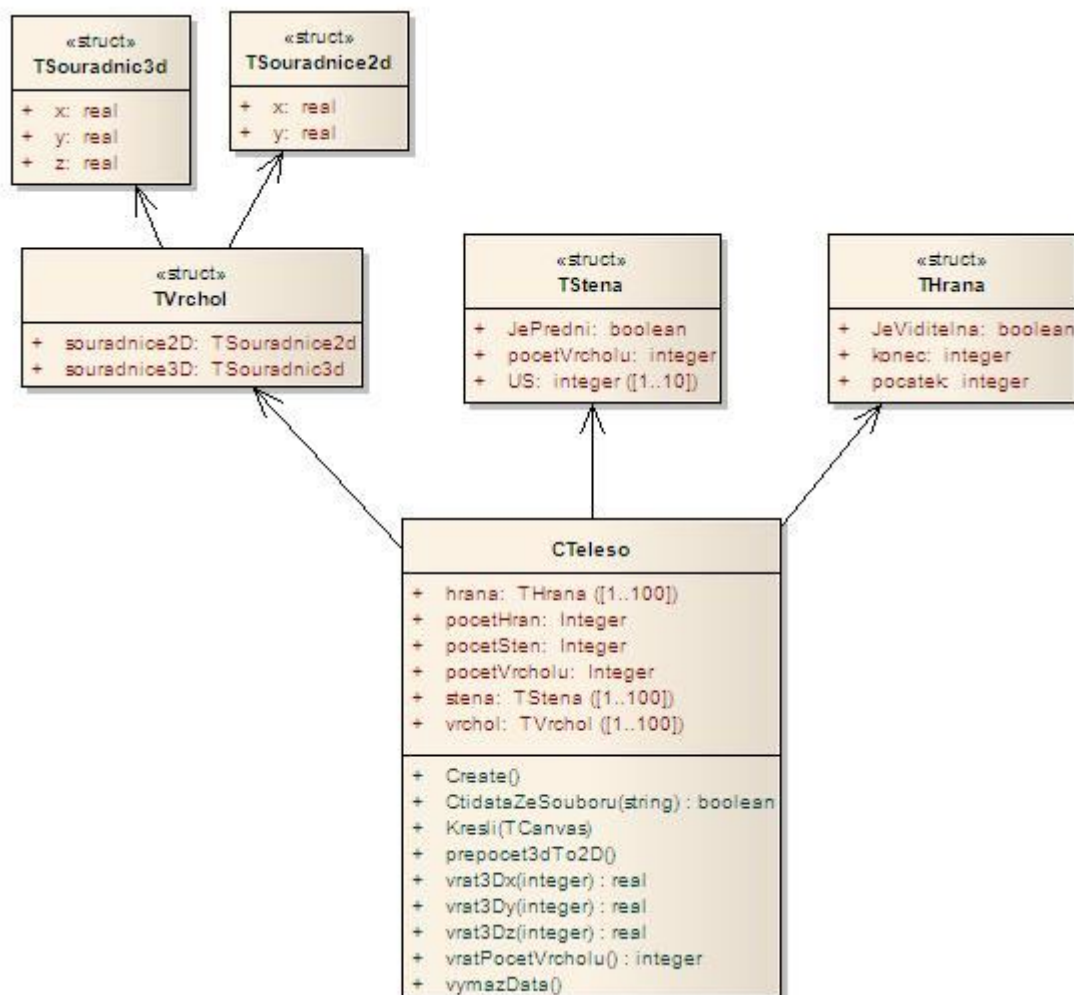
- nastaví souřadnice vrcholů

getVrcholy(cisloVrcholu,hodnota:integer)

- vrátí souřadnice vrcholů.

Třída CTeleso

- zastupuje objekt tělesa, na který budeme aplikovat transformace.



Obrázek 14 - Třída tělesa ve 3D

Metody

Create

- konstruktor objektu

vymazData

- vymaže data v objektu

CtidataZeSouboru(jmeno:string)

- načtení dat ze souboru (počet vrcholů, souřadnice, atd.)

prepocet3dTo2D

- přepočítá souřadnice pro zobrazení na obrazovku

Kresli(canvas:TCanvas)

- vykreslí těleso

vratPocetVrcholu

- navrátí počet vrcholů

vrat3Dx(cisloVrcholu:integer)

- vrátí reálnou souřadnici vrcholu tělesa na ose x

vrat3Dy(cisloVrcholu:integer)

- vrátí reálnou souřadnici vrcholu tělesa na ose y

vrat3Dz(cisloVrcholu:integer)

- vrátí reálnou souřadnici vrcholu tělesa na ose z

7.2.2 Funkce a procedury využití pro transformace v 3D

nactiSoubor(jmenoSouboru:string);

- načte zadaný soubor s daty pro těleso

maticeOtoceniX(uhel:real)

- nastaví parametry matice pro otočení ve směru osy x

maticeOtoceniY(uhel:real)

- nastaví parametry matice pro otočení ve směru osy y

maticeOtoceniZ(uhel:real)

- nastaví parametry matice pro otočení ve směru osy z

maticeMeritko(_x,_y,_z:real)

- nastaví parametry matice pro změnu měřítka

maticePosunu(_x,_y,_z: real)

- nastaví parametry matice posunu

maticeZkoseniYZ(_y,_z:real)

- nastaví parametry matice pro deformaci ve směru s yz

maticeZkoseniXZ(_x,_z:real)

- nastaví parametry matice pro deformaci ve směru s xz

maticeZkoseniXY(_x,_y:real)

- nastaví parametry matice pro deformaci ve směru s xy

X_2D(x,y,z : real)

- převede souřadnice x z 3D do 2D

Y_2D(x,y,z : real)

- převede souřadnici y z 3D do 2D

RealToPixel_x(x:real)

- převede reálné souřadnice x na hodnotu pixelu

RealToPixel_y(y:real)

- převede reálné souřadnice y na hodnotu pixelu.

7.2.3 Funkce a procedury využití pro transformace v 2D

Real2Pix_x(Xr:real):integer;

- přepočít reálných souřadnic na pixelové

Real2Pix_y(Yr:real):integer;

- přepočít reálných souřadnic na pixelové

Pix2Real_x(Xp:integer):Real;

- přepočít pixelových souřadnic na reálné

Pix2real_y(Yp:integer):real;

- přepočít pixelových souřadnic na reálné

pomer(WindMax,WindMin:integer; RealMax,RealMin:real):real;

- výpočet poměru stran nutného k přepočtu souřadnic

maticePosunu (prirustekX, prirustekY : real)

- nastavení parametrů matice posunu

maticeOtoceni(uhel :real)

- nastavení parametrů matice otočení

maticeMeritko(hodnotaX, hodnotaY : real)

- nastavení parametrů matice pro změnu měřítka

maticeZkoseni(hodnotaX,hodnotaY:real)

- nastavení parametrů matice posunu.

7.2.4 Společné funkce a procedury

kresliOsy;

- vykreslení os

zapisLog(nazevSouboru:string;text:string)

- zapsání do log souboru

soucinSouradnic(s:TVrchol;maticice:TMaticice;rozmer:byte)

- funkce pro násobení souřadnic

maticovaTransformace(transfMat:TMaticice)

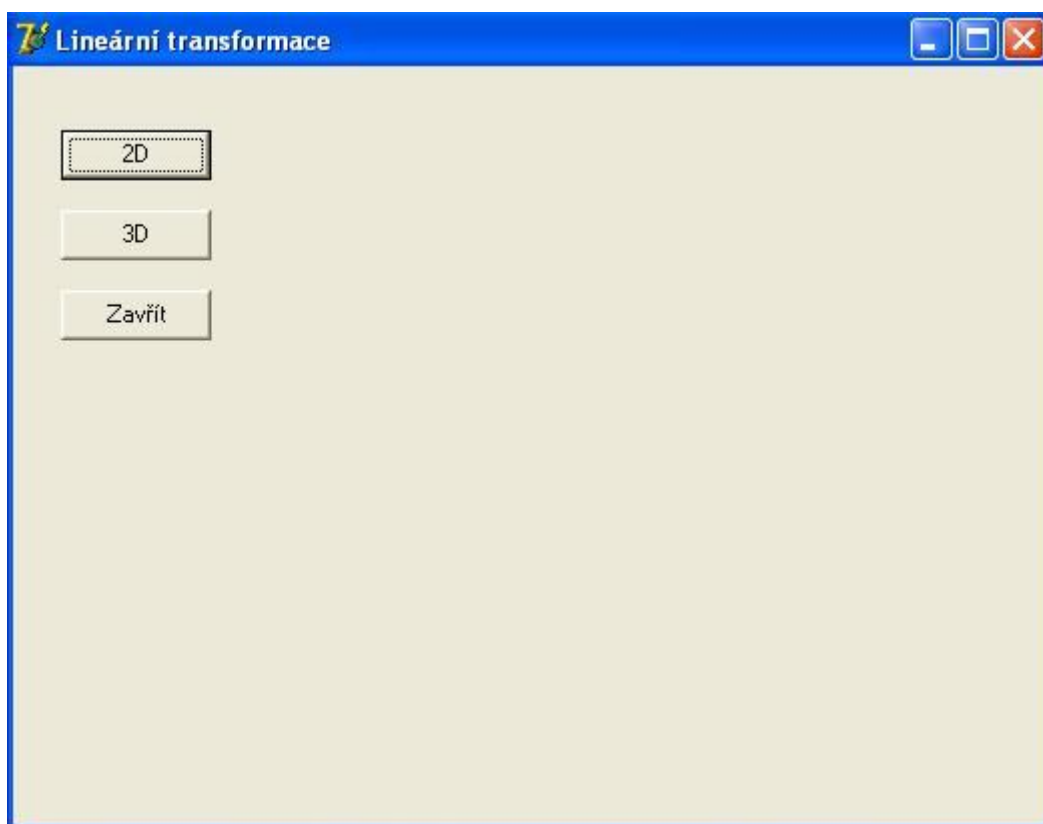
- procedura provádějící maticovou transformaci na objekt

soucinMatic (maticice1,maticice2:TMaticice;rozmer:byte)

- funkce pro násobení matic

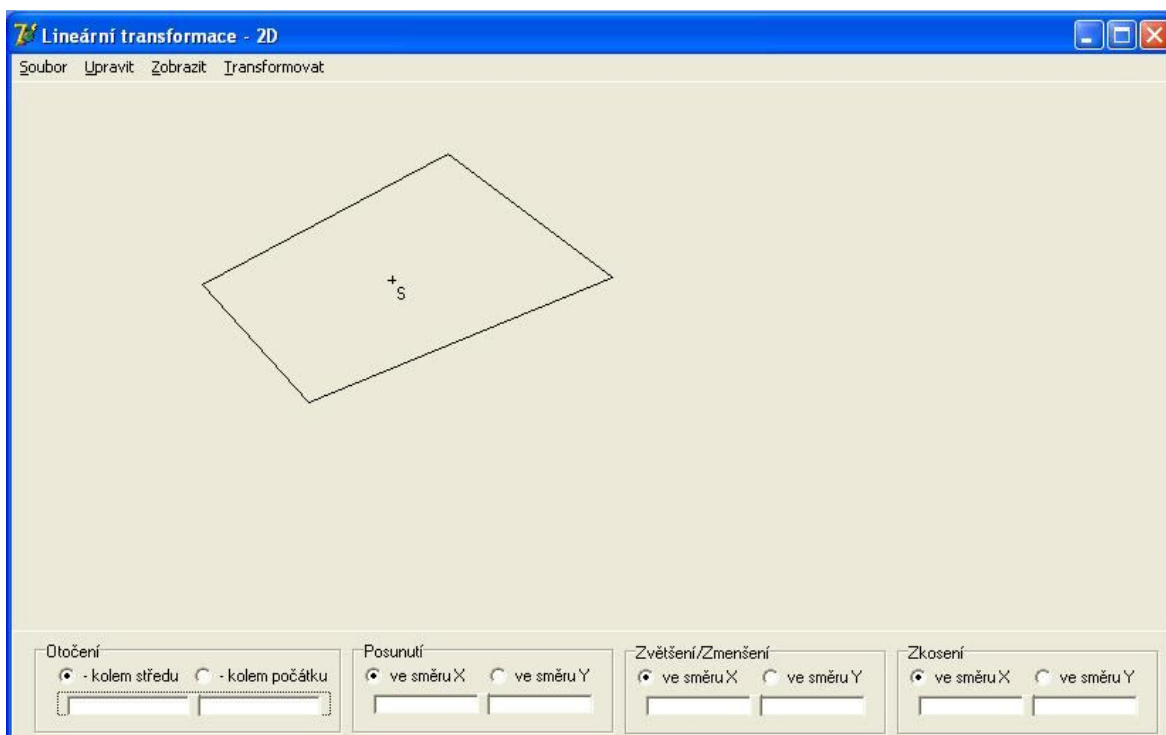
7.3 Ovládání aplikace

Aplikaci jako takovou spouštíme .exe souborem v hlavním adresáři nazvaný: „linearni_transformace“. Ihned se zobrazí úvodní menu, kde si můžeme vybrat transformace ve 2D nebo 3D, popřípadě ukončit aplikaci.



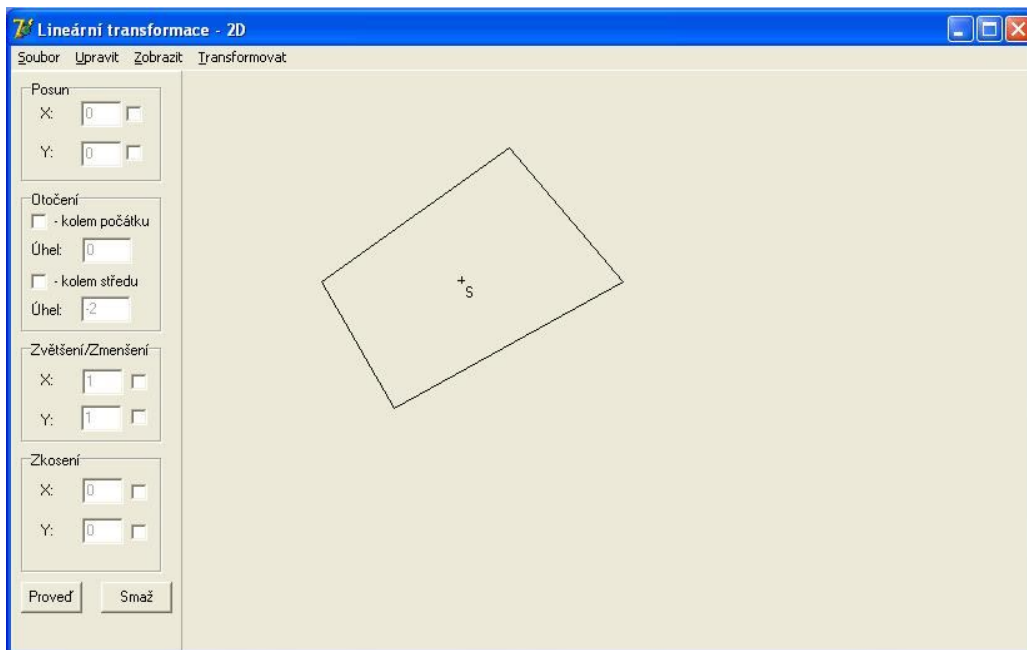
Obrázek 15 - Aplikace - Úvodní formulář

Pokud zvolíme transformace ve 2D, stiskem tlačítka 2D, objeví se nám nový formulář. Ihned vidíme menu aplikace v horní části, posuvníky v dolní části, pro tzv. dynamickou práci s transformovaným objektem, a prostor mezi posuvníky a menu, který tvoří kreslicí plochu. Chceme-li začít vytvářet polygon, který budeme následně transformovat, stačí kliknout levým tlačítkem myši na libovolné místo na kreslicí ploše. Tak definujeme první bod polygonu. Zvolíme-li další bod, uvidíme, jak se postupně bude polygon vykreslovat. Stisknutím pravého tlačítka myši se polygon automaticky uzavře a vypočítá se jeho střed.



Obrázek 16 - Aplikace – dynamická transformace 2D

Nyní můžeme začít transformovat, případně celý polygon smazat přes tlačítko v menu: Upravit -> Smazat. Pro transformaci můžeme využít dva módy. První, který se objeví při spuštění, je dynamický. Akce se provádějí za pomoci posuvníků v dolní části a přepínáním ve směru os, v jakém se mají vykonávat. Druhý mód je statický. Zde si můžeme zadat přesné hodnoty, které chceme předat jednotlivým parametrům transformačních matic. Přepínání mezi módy se ovládá přes tlačítka v menu: Transformovat -> Staticky nebo Dynamicky. Zvolením statického způsobu transformace se objeví nový panel s editačními prvky, kde zadáváme reálná čísla pro parametry. Zaškrtneme tu akci, kterou chceme aplikovat. Pokud je vše nastaveno podle požadavků, stačí stisknout tlačítko: Proved' a vykoná se složená transformace podle navolených hodnot. Zároveň se zvolené transformace po stisknutí tlačítka uloží do souboru na disk. Lze je tak zpětně prohlížet. Stisknutím: Zobrazit -> Logfile se tento soubor načte a vypíše do zobrazeného formuláře.



Obrázek 17 - Aplikace - Statická transformace 2D

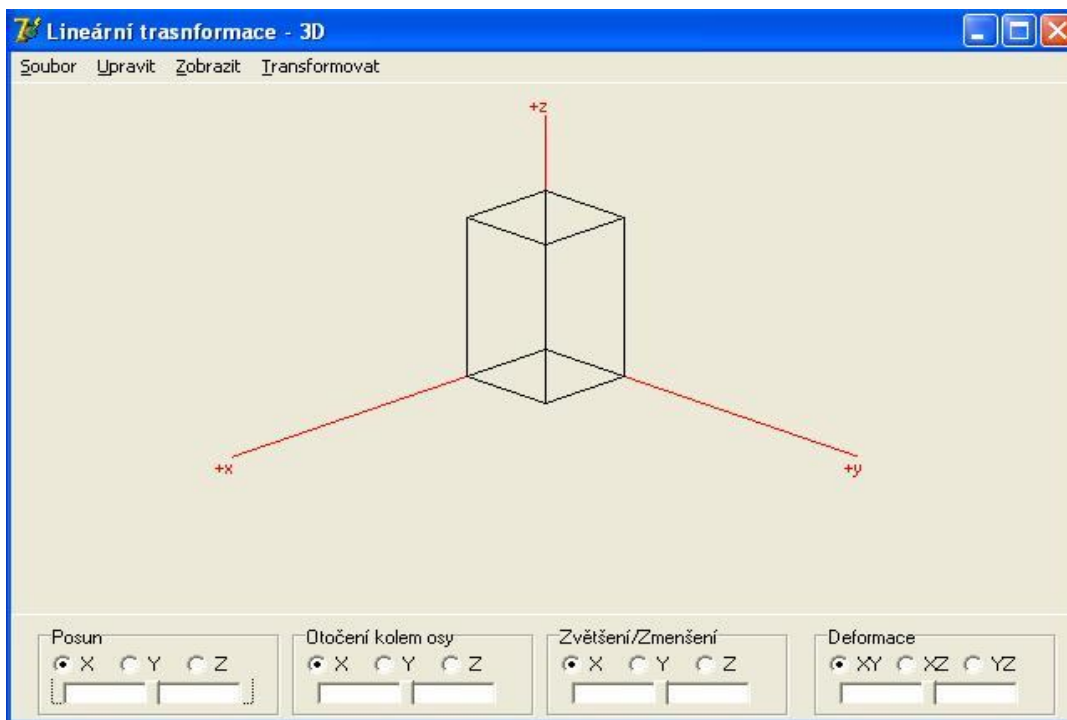
Položka: Zobrazit, umožňuje mimo zobrazování historie transformací také nastavení viditelnosti os nebo zobrazení informací o polygonu, rozsahu os nebo informací o poslední použité transformační matici.

V položce: Upravit, nalezneme možnosti pro nastavení rozsahu os, nastavení barev polygonu i os a také již zmiňovanou možnost smazání kreslicí plochy. To nám umožní nové zadání libovolného polygonu.

Nabídka: Soubor, skrývá dvě tlačítka. Menu, které nás vrátí na úvodní formulář a tlačítko: Zavřít, které zavře aplikaci.

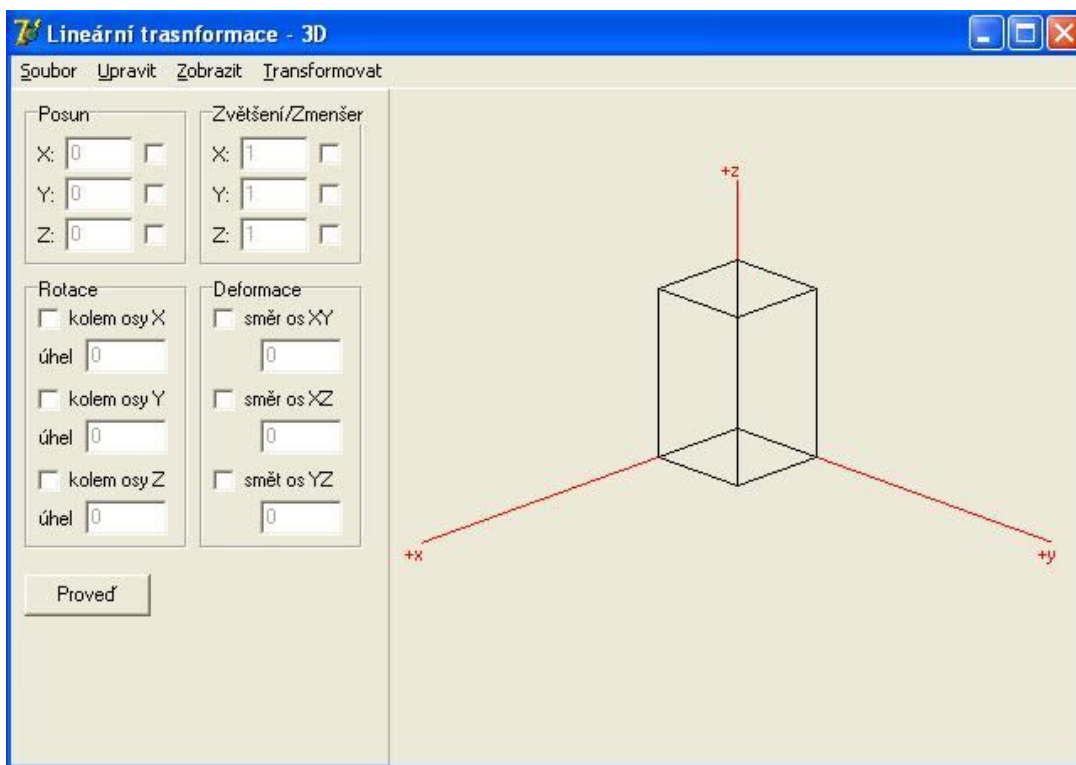
Stiskneme-li tlačítko 3D, dostaneme se do formuláře pro transformace těles.

Stejně jako v předchozím případě, i zde můžeme vidět v horní části menu, v dolní posuvníky a mezi nimi kreslicí plochu, kde se budou vykreslovat 3D tělesa. Souřadnice vrcholů těles získáváme ze souboru, kde je striktně dán vzhled zápisu. V adresáři: Objekty, jsou předdefinována tři tělesa. Krychle, kvádr a komolý jehlan. Pokud chceme zobrazit vlastní definované těleso, lze tak učinit přes menu aplikace, v položce: Soubor -> Těleso -> Načti ze souboru. Na stejném místě jsou i předdefinované tvary. Pokud si zvolíme jeden z nich, ihned se nám vykreslí uprostřed kreslicí plochy v počátku souřadnic x , y , z .



Obrázek 18 - Aplikace - Dynamická transformace 3D

Nyní můžeme opět aplikovat transformace pomocí dynamického zadávání přes posuvníky. Můžeme smazat těleso, změnit jeho barvu nebo barvu os v nabídce Upravit. V nabídce Zobrazit jsou opět možnosti zobrazování os, zobrazení informací o tělesu nebo zobrazení historie aplikovaných transformací. Pro statické zadávání opět platí přepnutí přes: Transformovat -> Staticky.



Obrázek 19 - Aplikace - Statická transformace 3D

8 Závěr

V rozsahu zadání bakalářské práce byla vypracována výše představená aplikace. Dalším jejím možným rozšířením je propracovanější grafický design. Dále aplikace nabízí prostor pro rozvoj dalších oblastí promítání, jakými jsou: zobrazování stěn u těles, viditelnost hran, s kterou také souvisí vykreslování objektu a os v závislosti na sobě samých. Ovládání pohledu na těleso by mohlo být další možnou nadstavbou aplikace.

Závěrem nutno dodat, že během vývoje nevznikly žádné vážnější problémy. Nejprve byla aplikace vytvořena pro přímé zadávání hodnot parametrů matic. Později přišel nápad měnit tyto hodnoty pomocí posuvníků pro okamžitou reakci na zobrazovaném objektu, kde byl problém s výstupem hodnot komponentů trackbar.

V tuto chvíli aplikace splňuje požadavky zadání a je tak použitelná pro demonstraci lineární transformace a vlivu parametrů transformačních matic na ni.

Seznam použité literatury

[1] ŽÁRA, Jiří; BENEŠ, Bedřich; FELKEL, Petr. *Moderní počítačová grafika*. Vyd. 1. Praha 4 : Computer Press, 1998. 448 s. ISBN 80-7226-049-9.

[2] MOFFATT, Neil. *Delphi Basics* [online]. c2010 [cit. 2010-05-07]. . Dostupné z WWW: <<http://www.delphibasics.co.uk/>>.

Přílohy

Jako přílohu bakalářské práce přikládám CD, které obsahuje:

- Dokument ve formátu PDF
- Dokument ve formátu DOC
- Zdrojové kódy aplikace
- Spustitelný exe soubor aplikace