

# Technická univerzita v Liberci

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B 2616 – Elektrotechnika a informatika

Studijní obor: 2612R011 – Elektronické informační  
a řídicí systémy

Vytvoření 3D modelu lidského těla a jeho animace  
pro zobrazování znakové řeči

Creation of 3D human model and animation for  
display sign language

## BAKALÁŘSKÁ PRÁCE

Autor:

**Tomáš Petráček**

Vedoucí bakalářské práce:

Ing. Josef Chaloupka, Ph.D.

Konzultant

Ing. Jindřich Žďárský, Ph.D.

**V Liberci dne 13. 5. 2008**

-----zde patří zadání bakalářské práce-----

## Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že souhlasím s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mě požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do její skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

## Abstrakt

Tato bakalářská práce popisuje vytvoření 3D modelu lidského těla s následnou možností animace pro využití k vizualizaci znakového jazyka. Program byl vytvořen za pomoci jazyku pro popis virtuální reality VRML a univerzálních knihoven jazyka C++, které jsou volně dostupné. V projektu je rozvedena tematika grafických transformací v prostoru a metodika tvorby modelu a programu pro vytvoření pohybu.

Klíčová slova: znaková řeč, znakový jazyk, grafika, grafické transformace, matice transformací, quaternion, VRML, C++, visual studio, model těla, pohyb těla.

## Abstract

This studie describe creation of 3D model of human body with possibility of animation and vizualization of sign language. Program was made with help of language for description virtual reality VRML and universal librares language C++, which are freelly available on website. This project eleborates subject graphics transformations in space and methods of creation model and also program for creating movement.

Keywords: sign speech, sign language, graphics, graphics transformation, matrix of transformation, quaternion, VRML, C++, visual studio, model of human body, movement body.

# Obsah

ÚVOD .....	7
1 ZNAKOVÝ JAZYK A MOŽNOSTI JEHO VIZUALIZACE .....	8
1.1 SYSTÉMY PRO POPIS VIZUALIZACE ZNAKOVÉHO JAZYKA .....	8
1.2 PODOBNÉ SYSTÉMY .....	8
1.2.1 Projekt Musslap – Překlad do znakového jazyka.....	9
1.2.2 DePaul University American sign language project .....	10
1.2.3 Text-To-Sign language synthesis tool.....	11
1.3 ZNAKOVÝ JAZYK .....	12
1.3.1 Vývoj znakového jazyka .....	12
1.3.2 Lingvistika znakového jazyka.....	13
1.3.3 Prstová abeceda .....	14
2 MODEL LIDSKÉHO TĚLA .....	16
2.1 MODEL ČLOVĚKA .....	16
2.2 SYSTÉM KOSTÍ .....	16
3 ANIMACE VE 3D PROSTORU .....	19
3.1 MATEMATICKÝ POPIS TRANSFORMACE BODU V PROSTORU.....	19
3.1.1 Popis pomocí matic .....	19
3.1.2 Skládání transformací.....	21
3.1.3 Nekomutativnost násobení matic .....	21
3.1.4 Rotace kolem obecné osy .....	22
3.1.5 Popis rotací pomocí quaternionu.....	23
3.2 POHYBY MODELEM.....	24
4 REALIZACE SYSTÉMU .....	25
4.1 DIRECTX .....	25
4.1.1 Skinned mesh .....	26
4.2 TVORBA MODELU VE VRML.....	27
4.2.1 Jazyk pro popis virtuální reality .....	27
4.2.2 Model a jeho tvorba v jazyce VRML.....	28
4.3 TVORBA ANIMAČNÍHO PROGRAMU .....	29

---

4.3.1	Popis programu .....	29
4.3.2	Vykreslení kostry modelu .....	31
4.3.3	Realizace rotace bodu kolem obecného vektoru .....	32
4.3.4	Tvorba souboru s animací .....	32
4.3.5	Podpora programu .....	32
5	ZÁVĚR.....	33
5.1	BUDOUCNOST PROJEKTU .....	33
5.1.1	Propracování modelu.....	33
5.1.2	Animace znaků .....	33
5.1.3	Instalační balíček .....	34
5.1.4	Tvorba databáze .....	34
5.1.5	Syntéza řeči .....	34
	SEZNAM POUŽITÉ LITERATURY .....	35
	PŘÍLOHA.....	37

## Úvod

V dnešní moderní době je využití techniky, počítačů, každodenní praxí většiny lidí. Ještě v nedávné době byly některé projekty, které využívaly výpočetní techniku, nerealizovatelné díky nedostatečnému výkonu tehdejších počítačů. Avšak díky rychle se zvyšujícímu výkonu výpočetní techniky se rozvinuly některé vědní obory a na základě nového potenciálu technických prostředků vzniká mnoho nových. Velmi rychle se rozvíjejícím oborem je počítačová grafika.

Tato bakalářská práce popisuje využití trojrozměrné počítačové grafiky k zobrazení modelu lidského těla, k vizualizaci znakového jazyka a k pohybu obecně .

Cílem této práce je nalézt vhodnou formu pro vytváření slov ve znakovém jazyce s následnou prezentací. Pomocí programu by uživatel měl být schopen ovládat jednotlivé části lidského těla, které jsou spojeny s vizualizací znakového jazyka, zejména pak vhodné ovládání horních končetin a prstů.

Tato bakalářská práce obsahuje pracovní postup při vytváření výsledného programu a problémů spojených s jeho realizací. Tento projekt může být jedním z podkladů pro vytvoření uceleného systému pro vizualizaci českého znakového jazyka. Na tento projekt je možno v budoucnu navázat, zejména díky nabytým zkušenostem z tvorby a vizualizace pohybu v trojrozměrném prostoru.

# 1 Znakový jazyk a možnosti jeho vizualizace

Znakový jazyk je přirozený a plnohodnotný komunikační systém tvořený vizuálně pohybovými prostředky, tj. tvary rukou, jejich postavením a pohyby, mimikou obličeje, pozicemi hlavy a horní části trupu. Znakové jazyky vznikaly spontánně v komunitách neslyšících, kde se dále vyvíjejí. Do takovýchto komunit můžeme dále zařadit kromě neslyšících a nedoslýchavých také tlumočníky, přátele a rodinné příslušníky neslyšících. Znakový jazyk používají také lidé s poruchou řeči.

Znakový jazyk je často mylně označován termínem znaková řeč. Znaková řeč je zákonem o znakové řeči definovaný termín, který označuje jak český znakový jazyk, tak i znakovanou češtinu. Znakovaná čeština je však umělý jazykový systém, který využívá gramatické prostředky českého jazyka, který je artikulován současně se znaky českého znakového jazyka. Ve znakované češtině je každé slovo nahrazeno znakovým ekvivalentem. Existují i znakované systémy pro jiné mluvené jazyky. Těmto umělým systémům však neslyšící často nerozumí, protože zpravidla neovládají dobře gramatiku daného mluveného jazyka.

## 1.1 Systémy pro popis vizualizace znakového jazyka

Vytvoření programu, kterým by byl znakový jazyk univerzálně překládán, je velice složitá a zdouhavá věc. K tomuto cíli se lze pouze přiblížit. Existuje mnoho možných variant a kombinací slov, které bychom potřebovali vyjádřit či přeložit. Část slov by sice šla univerzálně přeložit do znakované češtiny, ale výsledek by nebyl zcela úplný. Program by musel mít vytvořenou obrovskou databázi slov jak ze znakované češtiny, tak z jednotlivých písmen prstové abecedy. Tyto dvě databáze by byly programem paralelně využívány a data z nich by byla mezi sebou různě kombinována. Např. pro zobrazení slova Jihlava by program využil obě dvě databáze, Jihlava se dá složit z písmen J a I prstové abecedy a znaku hlava.

Problém by však nebyl opět zcela vyřešen. Znakovaná forma českého jazyka totiž nezná skloňování, či časování sloves. Proto by musel být použit další systém, který by tento problém odstranil a před vstupem slov do překladače by tato slova převedl do základního tvaru.

## 1.2 Podobné systémy

Problémem překladače a vizualizace znakového jazyka se ve světě zabývá mnoho institucí. Jejich systémy jsou pojaty různě. Některé z nich mají dokonale propracované modely, jiné zase velmi kvalitní databázi znaků. Všechny mají ale jeden společný cíl – pomoci neslyšícím



lidem ve světě moderních technologií. V následujících odstavcích je uvedeno několik systémů, které jsou velmi podobné cíli této bakalářské práce.

### 1.2.1 Projekt Musslap – Překlad do znakového jazyka

Tento projekt je vyvíjen Západočeskou univerzitou v Plzni, konkrétně oddělením umělé inteligence na katedře kybernetiky. Projekt byl odstartován v roce 2004 a tento rok by měl být dokončen. Překlad do české znakové řeči je jen jednou z mnoha částí velkého projektu Musslap<sup>1</sup>. Cílem celého projektu je podpořit systémy používající rozpoznávání mluvené řeči rozpoznáváním vizuálním a na výsledek aplikovat metody vyhledávání informací. Část tohoto projektu řeší překlad psaného textu do znakové řeči. Systém se jmenuje TTSS<sup>2</sup> a je určen k vytváření znakových promluv z psaného textu. Systém se skládá ze dvou částí.

- Zpracování vstupního textu a jeho překlad do znakové řeči
- Syntéza znakové řeči

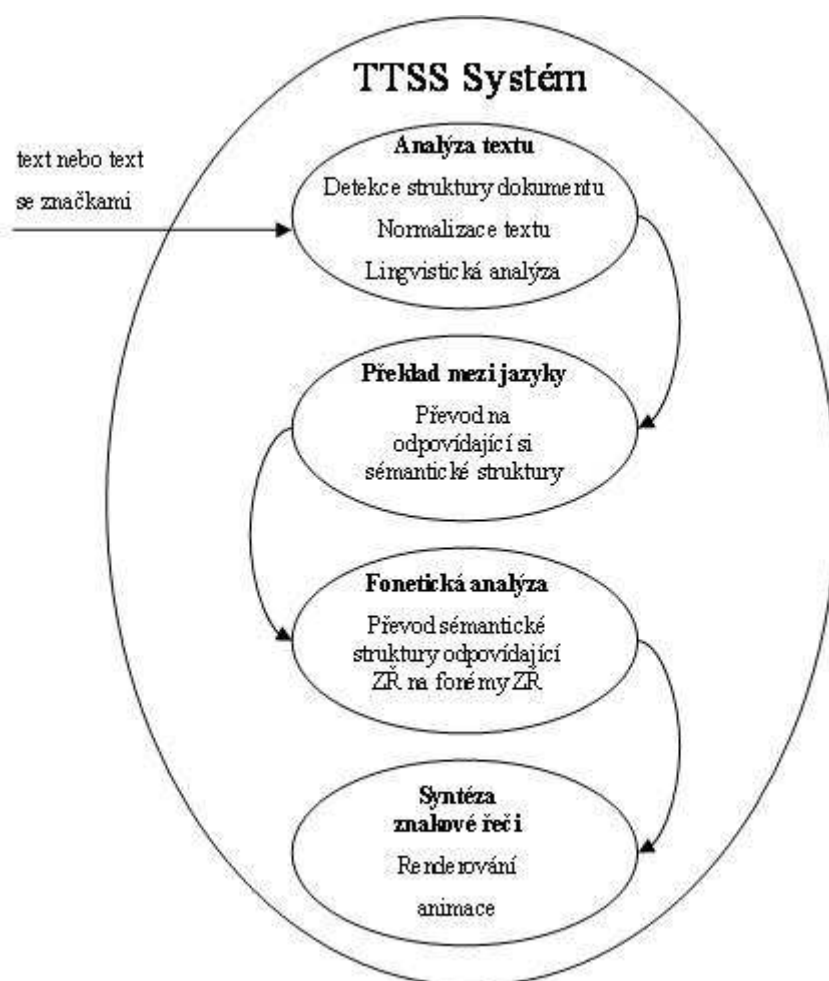


**Obr. 1:** Grafické podoba modelu

---

<sup>1</sup> Multimodal Human Speech and Sign Language Processing for Human-Machine Communication

<sup>2</sup>Text to sign speech



**Obr. 2:** Podrobné schéma TTSS systému

K vlastní reprezentaci výsledné vizualizace projekt využívá model Hiro, navržený podle standardu H-Anim<sup>3</sup>, vytvořený v jazyce VRML. Úplná specifikace definuje 89 kloubních spojení a 89 segmentů těla připojených k těmto kloubům. Ne však všechny definice musejí být dodrženy. Takto strukturovaný počítačový model je vhodný použít při animaci znakové řeči. Výsledné animace jsou vcelku dobře čitelné a jednotlivé části rukou dobře propracované. Model dokonce využívá některé doplňující (nonhanded<sup>4</sup>) prvky, jako je mimika obličeje či pohyby rtů. Jediné, co by se dalo modelu vytknout je stínování, které zde není řešeno vůbec.

### 1.2.2 DePaul University American sign language project

Systém je vyvíjen DePaul University v americkém Chicagu již od roku 1999. Podobně jako předchozí projekt řeší překlad psané formy jazyka do podoby znakové. Tentokrát však

<sup>3</sup> norma pro modely postav a jejich animace. Více na : <http://www.h-anim.org>

<sup>4</sup> anglický výraz pro vyjádření nemanuálních prvků znakové řeči

z angličtiny do americké znakové řeči. Problém je však složitější o to, že americká znaková řeč se liší od znakované podoby klasické angličtiny. Výsledný model, kterým je prováděna vlastní vizualizace, je dost propracovaný. Pěkné grafické zpracování modelu a časově realistické pohyby jsou velmi elegantní a snadno čitelné.



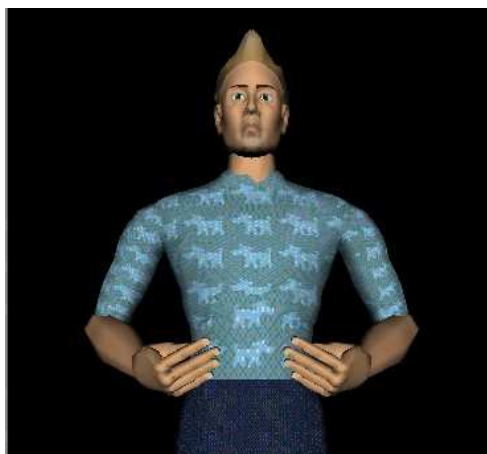
**Obr. 3:** Grafické zpracování modelu od DePaul University

### 1.2.3 Text-To-Sign language synthesis tool

Projekt je vyvíjen institutem informatiky a telematiky, konkrétně centrem pro výzkum a technologie v Řecku. Projekt se opět zabývá překladem anglického jazyka do anglické znakové řeči. V současné době je vytvořena základní databáze slov a jejich znakových ekvivalentů. K prezentaci animací projekt využívá prostředků jazyka VRML a v něm vytvořeného modelu podle normy H-ANIM. Výsledný projekt je díky univerzálnosti jazyka VRML umístěn na web. Uživatel zadá do vyhledávače slovo, které chce zobrazit. Pokud je toto slovo v databázi, dostane na výběr mezi třemi avatary<sup>5</sup>. V prezentaci projektu se dočteme, že projekt by měl mít v budoucnu uplatnění např. v televizních zprávách, kdy by automaticky překládal mluvenou formu do formy znakové. Grafické zpracování modelu je plně dostačující a pro vizualizaci jednoduchých znaků dobře využitelné.

---

<sup>5</sup> Běžně používaný termín pro označení virtuální postavy



**Obr. 4:** Grafická podoba projektu Text-To-Sign language

## 1.3 Znakový jazyk

### 1.3.1 Vývoj znakového jazyka

První písemné zmínky o znakovém jazyce pocházejí ze 17. století ze Španělska. V roce 1620 Juan Pablo Bonet v Madridu publikoval dokument, který popisoval výuku neslyšících lidí. Tento dokument je považován za první moderní dohodu fonetiky a logopedie. Určuje metody pro vzdělávání neslyšících za použití manuálních znaků ve formě abecedy hluchoněmých, za účelem zlepšit komunikaci mezi neslyšícími nebo němými lidmi. V 18. století publikoval Charles – Michel de l'Épée svoji abecedu pro neslyšící, která však vycházela z jazyka navrženého Bonetem. Jím vytvořený způsob komunikace je v základu nezměněný a používá se dodnes. V roce 1755 Abbé de l'Épée založil v Paříži první veřejnou školu pro neslyšící děti.

Prokazatelně nejslavnějším absolventem této školy byl Laurent Clerc, který později odešel do Spojených Států a spolu s Thomasem Hopkinsem Gaillaudetem založili v Hartfordu první americkou školu pro neslyšící. Edward Miner Gallaudet (syn Thomase Gaillaudeta) založil ve Washingtonu v roce 1857 první akademii pro hluchoněmé. Z té se v roce 1864 stala Gallaudet University.

Obecně má každý mluvený jazyk jako protipól znakový jazyk. V každém společenství lidí, kteří mluví stejným jazykem se vyskytují hluchoněmí. Ti tvoří pro danou oblast specifický znakový jazyk. Jazyk je tím specifičtější, čím více jsou od sebe dané oblasti vzdáleny, ať už geograficky nebo kulturou. Navíc může být každý znakový jazyk ovlivněn mluveným jazykem, ze kterého vychází. Proto odlišnost dvou znakových jazyků z různých oblastí může být značná. Přesto, když se setkají uživatelé dvou různých znakových jazyků, je

komunikace mezi nimi značně jednodušší, než když se setkají dva lidé hovořící různými mluvenými jazyky. V tomto ohledu znakový jazyk otevírá cestu k mezinárodní komunitě neslyšících. Znakový jazyk však není univerzální a existuje mnoho znakových jazyků, které jsou spolu navzájem většinou nesrozumitelné. Umělé znakové jazyky, protipóly mluvených jazyků, slouží hlavně pro dorozumívání slyšícího s neslyšícím, protože pro slyšícího člověka je mnohem jednodušší nahradit mluvená slova znaky. Vedle těchto národních znakových jazyků existuje mezinárodní znakový jazyk, který vznikl přirozeným vývojem z mezinárodního systému Gestuno<sup>6</sup>, který neměl žádná gramatická pravidla. Postupem času se z Gestuna a z gramatik znakových jazyků vytvořil Pidgin<sup>7</sup>, který se v současnosti již dá považovat za plnohodnotný znakový jazyk. Mezinárodní znakový jazyk se používá mimo jiné např. při Deaflympiádě a při setkání Světové federace neslyšících. Mnoho neslyšících však mezinárodní znakový jazyk neovládá.

### 1.3.2 Lingvistika znakového jazyka

Znakový jazyk je z lingvistického hlediska stejně bohatý a komplexní jako kterýkoliv mluvený jazyk. Profesionální lingvisté studovali mnoho znakových jazyků a ve všech z nich našli většinu jazykových komponentů, které se nacházejí v mluvených jazycích.

Znakový jazyk není pouze vyjádření mluveného slova pomocí posunků. Znaky jsou velmi specifické a abstraktní, často nemají žádný vizuální vztah k jejich mluvené interpretaci.

Znakový jazyk je stejně jako mluvený jazyk uspořádán pomocí základních bezvýznamných jednotek do plnohodnotných smysluplných celků. Základní prvky jednotlivých znaků jsou:

- orientace ( dlaně )
- umístění artikulace
- pohyb ( paže nebo dlaně)
- nemanuální prvky ( pohyb horní poloviny těla, mimika obličeje)

Stejně jako v mluveném jazyce, můžeme i v jazyce znakovém vyjádřit naši myšlenku nejen pomocí řeči, ale i pomocí dalších doplňkových faktorů. Například pouhá změna tónu hlasu na konci věty změní větu oznamovací na větu tázací či rozkazovací. Toto se v jazyce znakovém vyjadřuje pomocí výrazu v obličeji nebo změnou důrazu při interpretaci znaku

---

<sup>6</sup> mezinárodní znakový systém

<sup>7</sup> komunikační systém s nezřetelnou gramatikou, v níž se spojují prvky dvou nebo více přirozených jazyků

pomocí rukou. Při rozhovoru, kde je použit znakový jazyk, se proto nemůžeme vizuálně zaměřit pouze na pohyb rukou, ale musíme člověka mluvícího znakovým jazykem vnímat jako komplexní celek.

### 1.3.3 Prstová abeceda

Prstová abeceda je vizuálně motorická (stejně jako znakový jazyk). Využívá formalizovaných a ustálených postavení prstů a dlaně jedné ruky k zobrazování jednotlivých písmen abecedy. Její produkce je lineární. Pro její použití je nutná znalost písemné formy mluveného jazyka.

Prstová abeceda je integrální součástí českého znakového jazyka a totální komunikace, kde se využívá k odhláskování cizích slov, jmen, odborných termínů a pojmů, pro které dosud nejsou ustáleny znaky českého znakového jazyka a k vizualizaci tvarů slov (koncovek) mluveného jazyka.

Existují jednoruční i obouruční prstové abecedy a v jednotlivých zemích se více či méně liší. V České republice se převážně používá obouruční prstová abeceda. Jednoruční prstová abeceda se používá na východní Moravě a na Slovensku.



**Obr 5.** Ukázka písmene A v jednoruční prstové abecedě



**Obr 6.** Ukázka písmene A ve obouruční prstové abecedě

Jednoruční prstová abeceda pochází ze Španělska, kde ji nacházíme ve spisech z r. 1535 a 1575 jako tajnou mluvu. Tímto způsobem měla být znázorněna písmena latinské abecedy polohou prstů. Jednoruční prstovou abecedu využil jako první k výuce a komunikaci s neslyšícími Španěl Bonet v roce 1620. U nás se prstová abeceda poprvé začala používat ve výuce v roce 1957.

## 2 Model lidského těla

Cílem této bakalářské práce je vytvořit trojrozměrný model lidského těla s možností libovolně pohybovat s horními končetinami tak, aby bylo možné vyznakovat libovolný znak ze znakového jazyka. Vytvořeným modelem by mělo být možné zcela bez omezení pohybovat a otáčet tak, aby si uživatel mohl nastavit pro něj nejvhodnější zobrazení. Zároveň by měl být kladen důraz na jednoduchost a dostupnost. Pro zobrazení je proto použit grafický systém VRML<sup>8</sup>. Při použití této platformy pro zobrazení 3D objektů, je třeba nainstalovat podporu tohoto jazyka. Využití systému je velice široké a praktické. Je vhodný pro spolupráci s internetovými prohlížeči, tedy po umístění animace na web, si může uživatel jednotlivé grafické práce prohlížet a používat na internetu.

### 2.1 Model člověka

Model lidského těla je jen jednou z mnoha částí potřebných k vytvoření komplexního nástroje pro tvorbu a prezentaci jednotlivých posunků ve znakovém jazyce. Je však neméně důležitým prvkem, z hlediska prezentace je dokonce základním stavebním kamenem celkového dojmu. Při hledání vhodného modelu si lze vybrat mezi dvěma faktory: reálný model s vyššími nároky na použitý hardware oproti jednoduchému modelu s možností funkčnosti i na starších počítačích a operačních systémech ( Windows ). Pro zobrazení znakového slova jsou především potřebné dostatečně propracované horní končetiny a obličej pro případné doplňující informace o daném slově.

Vhodným kompromisem při výběru dvou již zmíněných faktorů je grafický formát VRML, jazyk určený pro popis virtuální reality. Lze v něm vytvořit téměř jakýkoliv grafický objekt či grafickou scénu. Byl zvolen díky své hardwarové nenáročnosti a možnosti umístění výsledků na web. Výsledný grafický vzhled se sice nedá srovnávat s jinými platformami typu DirectX nebo OpenGL, ale pro tuto potřebu je plně dostačující.

### 2.2 Systém kostí

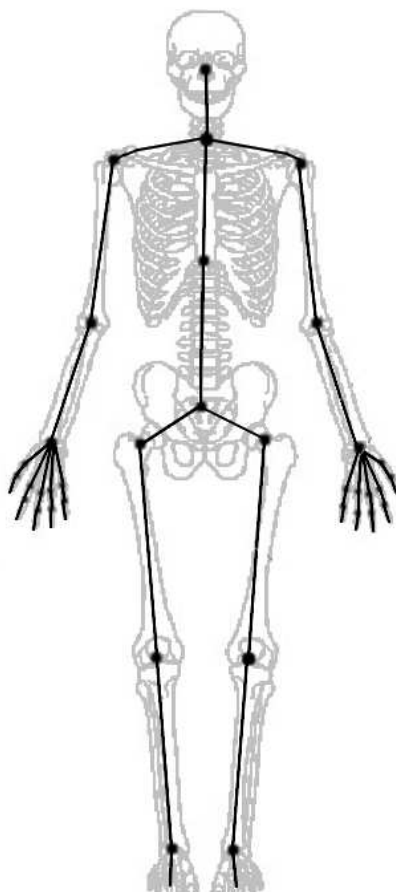
Výsledný model by se měl co nejvíce podobat lidskému tělu. Proto je v modelu použit systém kostí, který je až na výjimky shodný s kostrou člověka a je vytvořený podle normy H-ANIM. Výjimky tvoří zejména některé části na lidském těle, které lze nahradit monolitickými celky. Jedná se například o zápěstní kůstky, chodidla, či páteř, která není tvořena jednotlivými

---

<sup>8</sup> Virtual reality model language – jazyk pro popis virtuální reality



obratli, ale je rozdělena pro jednoduchost do tří částí. Naopak některé části musejí být stejné jako na skutečné kostře člověka. Již z principu samotné interpretace znakové řeči je jasné, že se jedná o horní končetiny. Ramenní kloub, ze kterého vychází pažní kost, je spojena přes loket s kostmi předloktí. Předloktí je provázáno přes zápěstí s jednotlivými články prstů.



**Obr 7.** Kostra modelu lidského těla

Před samostatným popisem modelu je třeba definovat některé standardně užívané pojmy v oblasti 3D grafiky.

- **vertex**<sup>9</sup> - jedná se o bod v prostoru definovaný třemi souřadnicemi (x, y, z) v kartézské soustavě.
- **face, polygon** - plocha v modelu definovaná zpravidla třemi vertexy, polygon je definován i více než třemi.

---

<sup>9</sup> pojmy jako vertex a další se zpravidla nepřekládají a budou zde uvedeny v anglickém jazyce

- **joints** - kloub, slouží pro animování modelu, je definován polohou v kartézském souřadném systému.
- **bone** - prvek modelu, který se přímo využívá pro zobrazování částí modelu. Kost je definována jako spojnice mezi dvěma sousedícími klouby.

Model lidského těla, vytvořený v jazyce VRML, je složen ze systému kostí a kloubů. V modelu je zvolen referenční bod (kloub), který je v kartézském systému umístěn v počátku. Jedná se o kloub pelvis<sup>10</sup>, tomuto kloubu jsou všechny ostatní klouby a kosti podřazené. Jednotlivé kosti jsou spojeny pomocí kloubů. S kostmi jsou provázány vertexy, které vytvářejí povrch modelu. Při pohybu jednou kostí pohneme i všemi vertexy, které jsou s danou kostí asociovány. Kostí v modelu mají dále definovaného svého potomka. Pohneme-li nadřazenou kostí, svoji pozici podle této kosti změni všechny kosti podřazené. Tím lze docílit rozpohybování modelu.

---

<sup>10</sup> v anglickém jazyce výraz pro pánev

## 3 Animace ve 3D prostoru

### 3.1 Matematický popis transformace bodu v prostoru

Pro popis animace v prostoru je třeba si objasnit některé pojmy související s transformací bodu. Teoreticky si lze zvolit jeden z mnoha matematických nástrojů, ale ne všechny jsou pro tuto potřebu použitelné. Při výpočtech je třeba zohlednit i zaokrouhlovací chyby, které by při nevhodném použití daného matematického nástroje mohli výsledek zcela znehodnotit.

#### 3.1.1 Popis pomocí matic

Transformace bodu v prostoru může být zobrazena zápisem

$$\vec{v}' = \mathbf{M} \cdot \vec{v} \quad (1)$$

Transformovaný vektor  $\vec{v}'$  se spočte jako součin transformační matice  $\mathbf{M}$  a původního vektoru  $\vec{v}$ . Transformace v rozšířeném trojrozměrném prostoru je určena soustavou rovnic

$$\begin{bmatrix} v_1' \\ v_2' \\ v_3' \\ v_4' \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \quad (2)$$

Zde je ve stručnosti uvedeno několik speciálních matic, které určují nejznámější transformace v trojrozměrném prostoru:

- translace

je transformace posunutí, kde

$$\vec{p} = (p_x \quad p_y \quad p_z) \quad (3)$$

je vektor posunutí, který určuje jakým směrem a o jakou vzdálenost bude bod posunut.

Transformační matice posunutí  $\mathbf{T}$  a inverzní matice  $\mathbf{T}^{-1}$  jsou

$$\mathbf{T}(p_x, p_y, p_z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p_x & p_y & p_z & 1 \end{bmatrix} \quad (4)$$

$$\mathbf{T}^{-1} = \mathbf{T}(-p_x, -p_y, -p_z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -p_x & -p_y & -p_z & 1 \end{bmatrix} \quad (5)$$

- Změna měřítka<sup>11</sup>

je transformace používaná pro změnu velikosti zobrazovaného objektu a je dána maticí

$$S(s_x, s_y, s_z) = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad S^{-1}(s_x, s_y, s_z) = S\left(\frac{1}{s_x}, \frac{1}{s_y}, \frac{1}{s_z}\right) \quad (6)$$

koeficienty  $s_x \neq 0$ ,  $s_y \neq 0$  a  $s_z \neq 0$  jsou použity ve směru příslušné souřadnicové osy. Pomocí měřítkových koeficientů lze realizovat některou z transformací souměrnosti v prostoru (středovou souměrnost, souměrnost podle roviny a osovou souměrnost). Například souměrnost podle roviny  $xy$  můžeme realizovat pomocí koeficientů  $s_x = 1$ ,  $s_y = 1$  a  $s_z = -1$

- Rotace

otáčení ve třech rozměrech může být jedním z podpřípadů otáčení kolem jednotlivých souřadnicových os. Matice  $\mathbf{R}_x$  reprezentující otáčení okolo osy  $x$  o úhel  $\alpha$  a odpovídající inverzní matice  $\mathbf{R}_x^{-1}$  jsou

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) & 0 \\ 0 & -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$\mathbf{R}_x^{-1}(\alpha) = \mathbf{R}_x(-\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Obdobně jsou sestaveny matice pro otáčení kolem osy  $y$ , resp.  $z$ :

$$\mathbf{R}_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$\mathbf{R}_z(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

---

<sup>11</sup> běžně se používá výraz scaling

### 3.1.2 Skládání transformací

Velmi důležitým prvkem v matematickém popisu transformace bodu je složení několika transformací. To nám umožní provést několik transformačních operací najednou. Např. máme matice transformací  $\mathbf{A}_1$  a  $\mathbf{A}_2$ . Ty by měly být aplikovány na bod  $B$ , reprezentovaný vektorem  $\vec{b}$ , pak lze provést

$$\vec{b}_1 = \mathbf{A}_1 \cdot \vec{b} \quad (11)$$

$$\vec{b}_2 = \mathbf{A}_2 \cdot \vec{b}_1 \quad (12)$$

nebo lze transformace  $\mathbf{A}_1$  a  $\mathbf{A}_2$  složit do jedné  $\mathbf{A}$

$$\mathbf{A} = \mathbf{A}_1 \cdot \mathbf{A}_2 \quad (13)$$

a transformovat

$$\vec{b}_2 = \mathbf{A} \cdot \vec{b} \quad (14)$$

### 3.1.3 Nekomutativnost násobení matic

Při násobení dvou matic je potřeba pamatovat na to, že násobení matic není komutativní operace, na rozdíl od obvyčejného násobení dvou čísel. Obecně totiž nemůžeme zaměnit pořadí matic, neboť bychom dostali dva různé výsledky. Toto pravidlo lze porušit pouze při násobení některých speciálních matic (např. jednotková matice).

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \neq \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad (15)$$

Na jednoduchém příkladu je uveden výsledek špatného pořadí vynásobení dvou matic rotace. Na stole leží kniha. Hřbet knihy je na levé straně kolmo na hranu stolu a titulní strana knihy směřuje nahoru. Osa  $x$  je na hraně stolu, osa  $y$  prochází hřbetem knihy, osy  $xy$  tedy definují rovinu stolu. Osa  $z$  prochází kolmo k rovině stolu. Řekněme, že chceme knihu nejprve otočit o  $90^\circ$ <sup>12</sup> kolem osy  $z$ . Kniha bude ležet hřbetem u hrany stolu. Poté knihu otočíme okolo osy  $x$  o  $180^\circ$ , kniha bude směřovat hřbetem od hrany stolu směrem k pozorovateli a titulní stranou dolů. Tuto jednoduchou transformaci lze reprezentovat vztahem:

$$\mathbf{T} = \mathbf{R}_z(\gamma) \cdot \mathbf{R}_x(\alpha) \quad (16)$$

Pokud by bylo zaměněno pořadí transformací, situace by vypadala následovně. Nejprve bychom knihu nechali otočit kolem osy  $x$  o  $180^\circ$ , hřbet knihy bude na levé straně stolu a titulní strana bude směřovat dolů. Následně knihu otočíme kolem osy  $z$  o  $90^\circ$ . Kniha je sice

---

<sup>12</sup> Pro otáčení v kladném směru použijeme pravidlo pravé ruky (palec ve směru osy a prsty ve směru otáčení)

titulní stranou dolů stejně jako v předešlé transformaci a její hřbet bude opět u hrany stolu, bude však směřovat směrem od pozorovatele. Výsledek je tedy skutečně odlišný, tzn.

$$R_z(\gamma) \cdot R_x(\alpha) \neq R_x(\alpha) \cdot R_z(\gamma) \quad (17)$$

### 3.1.4 Rotace kolem obecné osy

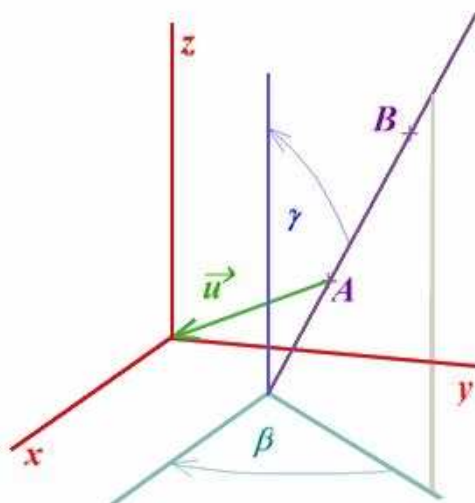
Otáčení kolem obecné osy v prostoru lze realizovat složením několika dílčích transformací kolem jednotlivých os  $x$ ,  $y$ ,  $z$  (viz. vztahy 7-10). Nalezení příslušných transformací (úhlů otočení) však není jednoduché. Osu rotace  $O$  určíme pomocí dvou bodů  $A = [a_1; a_2; a_3]$ ,  $B = [b_1; b_2; b_3]$  a ztotožníme ji s některou souřadnou osou, v našem případě s osou  $z$ . Je zřejmé, že osu nejprve posuneme o vektor  $\vec{u}$ , čímž bod  $A$  ztotožníme s počátkem. Dále je třeba osu rotace otočit o úhel  $\beta$  daný vztahem

$$\beta : \operatorname{tg}(\beta) = -\frac{b_2 - a_2}{b_1 - a_1} \quad (18)$$

kolem osy  $z$ . Po provedení této operace bude osa rotace ležet v rovině  $xz$ . Ke ztotožnění s osou  $z$  zbývá provést rotaci kolem osy  $y$  o úhel  $\gamma$

$$\gamma : \operatorname{tg}(\gamma) = \frac{\sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2}}{b_3 - a_3} \quad (19)$$

Osa rotace je nyní ztotožněná s osou  $z$ . Následně je třeba provést rotaci kolem osy  $z$  o daný úhel  $\alpha$  a inverzními transformacemi se z osy  $z$  vrátit k původní ose  $AB$ . Vše je znázorněno na obr. 4.



**Obr. 8** Rotace kolem obecné osy

Rotaci kolem osy  $AB$  o úhel  $\alpha$  lze dostat složením těchto transformací:

$$R_{AB}(\alpha) = T(AO) \cdot R_z(\beta) \cdot R_y(\gamma) \cdot R_z(\alpha) \cdot R_y(-\gamma) \cdot R_z(-\beta) \cdot T(OA) \quad (20)$$

### 3.1.5 Popis rotací pomocí quaternionu

V matematice se pojem quaternion označuje nekomutativní rozšíření komplexních čísel. Je tedy definován jako číslo s jednou reálnou a třemi imaginárními složkami. Zatímco komplexní čísla jsou vytvořena z reálných čísel a přidaného prvku  $i$ , který splňuje podmínku  $i^2 = -1$ , tak quaternion je zapsán jako

$$w + ix + jy + kz \quad (21)$$

kde

$$ij = k, jk = i, ki = j, ji = -k, kj = -i, ik = -j, i^2 = j^2 = k^2 = ijk = -1$$

Z výše uvedeného je zřejmé, že quaterniony nejsou komutativní, což je dobře využitelné zejména pro rotaci v prostoru.

Dále bude použit jednotkový quaternion, který je definován jako

$$w^2 = x^2 = y^2 = z^2 = 1 \quad (22)$$

Po aplikaci quaternionu do obecné transformační matice dostaneme rotační matici reprezentovanou jednotlivými složkami quaternionu

$$\mathbb{R} = \begin{pmatrix} w^2 + x^2 - y^2 - z^2 & 2xy - 2wz & 2wy + 2xz & 0 \\ 2wz + 2xy & w^2 - x^2 + y^2 - z^2 & 2yz - 2wx & 0 \\ 2xz - 2wy & 2wx + 2zy & w^2 - x^2 - y^2 + z^2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (23)$$

a tu použijeme obvyklým způsobem.

Další důležitou operací pro skládání rotací je násobení quaternionů. Princip násobení je stejný jako u násobení matic s tím rozdílem, že pořadí zápisu má opačný smysl.

$$q = q_2 \cdot q_1 \quad (24)$$

Operace násobení quaternionů  $q_1 q_2$ , kde  $q_1 = a + \vec{u}$  a  $q_2 = t + \vec{v}$ , je definována jako

$$q_1 q_2 = at - \vec{u} \cdot \vec{v} + a\vec{v} + t\vec{u} + \vec{u} \times \vec{v} \quad (25)$$

kde operace  $\vec{u} \cdot \vec{v}$  je skalární součin a  $\vec{u} \times \vec{v}$  vektorový součin dvou vektorů. Po úpravě:

$$q_1 q_2 = w_1 w_2 - x_1 x_2 - y_1 y_2 - z_1 z_2 + i(y_1 z_2 - z_1 y_2 + w_1 x_2 + x_1 w_2) + j(z_1 x_2 - x_1 z_2 + w_1 y_2 + y_1 w_2) + k(x_1 y_2 - y_1 x_2 + w_1 z_2 + z_1 w_2) \quad (26)$$

Inverzní rotaci, stejně jako u maticového tvaru, lze získat inverzí quaternionu  $q^{-1}$ , ta je definována

$$q^{-1} = (w + ix + jy + kz)^{-1} = w - ix - jy - kz \quad (27)$$

Jsou známy i další matematické operace s quaterniony, ale pro popis rotace bodu v prostoru nám postačí pouze tyto základní.

### 3.2 Pohyby modelem

Aby byl naplno využit potenciál jazyka VRML, je potřeba nejprve porozumět struktuře a implementaci animací v tomto jazyce. Platforma VRML využívá k animaci modelu záchytné body, klouby tzv. *joints*. Jak je uvedeno v kapitole 2.2 k animaci se využívá vlastnosti kostí *children* a *parent*. Například při předpažení v modelu pažní kostí, resp. změnění pozice loketního kloubu, je změněna i pozice a orientace všech kostí a kloubů této kosti podřazených, tzn. předloktí, zápěstí a všechny kosti prstů. Pro úspěšné popsání všech prvků animace je potřeba znát:

- kloub, který chceme animovat
- vektor, kolem kterého bude daný joint rotovat
- úhel, o který bude joint posunut vzhledem k vektoru rotace
- keyframe – póza modelu definovaná v čase

Každý kloub použitý v modelu má definovány tyto čtyři informace a z těchto čtyř základních vstupních dat dokáže daný pohyb reprezentovat. První zásadní článek je kloub, který chceme nechat rotovat. Potřebujeme znát jeho pozici v kartézském souřadném systému zadanou pomocí tří složek  $x$ ,  $y$ ,  $z$ . Tento kloub je zpravidla první kloub za osou rotace, tzn. chceme-li pohnout loktem, jedná se o loket, ale osa rotace bude procházet kloubem ramene.

Osa rotace je určena vektorem rotace, který je reprezentován třemi složkami  $v_x$ ,  $v_y$ ,  $v_z$ . Pro snadnější ovládání programu je tento vektor automaticky navrhnut podle kosti *parent*, avšak uživatel si může vektor libovolně upravovat a měnit.

Další nedílnou součástí je úhel rotace. V programu je reprezentován pomocí radiánů a dodržuje zásady konvence uvedené výše.

Poslední částí je *keyframe*. Každý joint obsahuje pole *keyframe* a všechny prvky tohoto pole obsahují data vektoru rotace a úhlu. Každá animace, která je do modelu implementována, je rozdělena právě na daný počet *keyframe*. V praxi to znamená čím bude v animaci více *keyframe*, tím bude animace kvalitnější a propracovanější. Musíme si však uvědomit, že tvorba animace s vysokým počtem *keyframe* je vzhledem ke grafickému zpracování modelu zbytečná a je proto zapotřebí najít vhodný kompromis. Pro potřeby této práce by mělo stačit něco kolem 6 až 10 animačních snímků.



## 4 Realizace systému

### 4.1 DirectX

DirectX je multimediální knihovna vytvořená firmou Microsoft, která je využívána pro bezplatné použití při vývoji her a jiných grafických aplikací pro operační systém Windows. Tato knihovna je stručně řečeno rozhraní mezi softwarem a hardwarem, konkrétně mezi Windows a grafickou kartou, zvukovou kartou a vstupními periferiemi, jako je klávesnice, myš a další. Samotná knihovna DirectX obsahuje mnoho podknihoven, zde je uvedeno několik základních:

- DirectDraw
- Direct3D
- DirectInput
- DirectSound
- DirectPlay
- DirectMusic
- DirectShow

V knihovně DirectDraw jsou základním stavebním prvkem plochy<sup>13</sup>, na tyto plochy můžeme libovolně kreslit několika způsoby nebo třeba vkládat obrázky v nejrůznějších formátech. Teoreticky můžeme vytvořit nekonečné množství používaných ploch, počet je však omezen velikostí dostupné grafické a systémové paměti. Minimální počet vytvořených ploch jsou dvě - přední a zadní. Přední plocha je viditelná na obrazovce monitoru, zadní však ne. DirectShow nám však nabízí volbu, která plocha bude přední, a která zadní. Toto se dá s výhodou využít k zobrazování různých grafických efektů.

Knihovna Direct3D je základní knihovnou DirectX. Byla vytvořena tak aby realizovala co nejvěrněji reálný prostor. Obsahuje například různé typy osvětlení scény, nastavení kamery, vlastnosti animace, stínování, textury a mnoho dalších grafických vymožeností.

DirectInput slouží ke komunikaci mezi periferiemi a knihovnou DirectX. Ukládá například hodnotu stlačené klávesy na klávesnici nebo polohu myši.

---

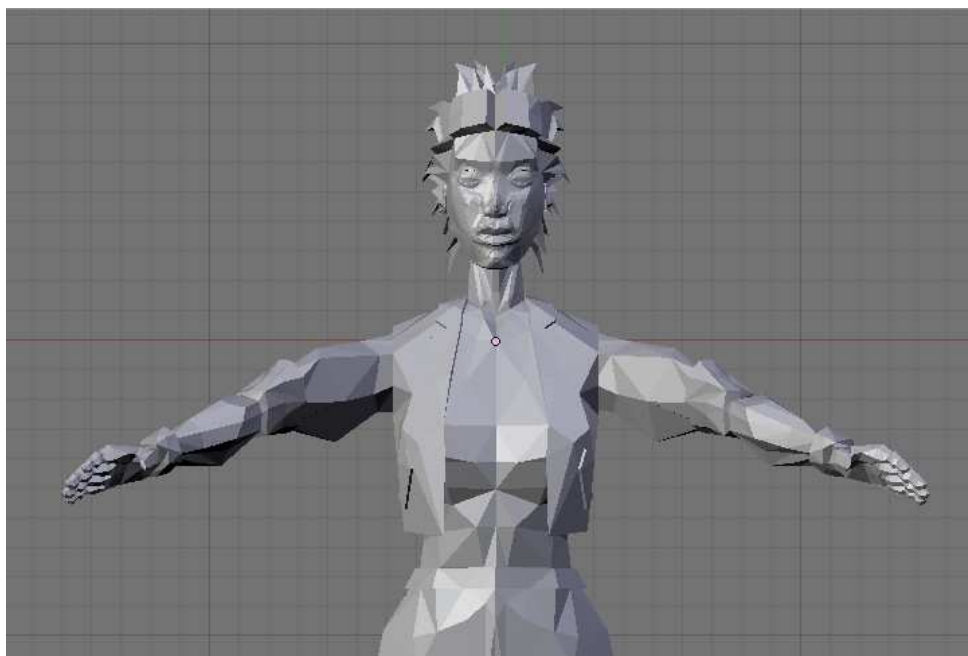
<sup>13</sup> V literatuře běžně označováno termínem surface

Knihovny DirectSound a DirectMusic, slouží k přehrávání zvuku a hudby.

Více zde nebudu vlastnosti a teorii DirectX rozvádět, další informace lze nalézt na webových stránkách Microsoftu<sup>14</sup>, kde je vše rozvedeno do detailů.

#### 4.1.1 Skinned mesh

Původním cílem bylo vytvořit model a jeho následnou animaci právě za pomoci knihoven DirectX. Microsoft nabízí na svém webu jako podporu pro svůj produkt DirectX nejrůznější předpřipravené příklady a programy. To je i příklad Skinned mesh. Jedná se o předpřipravený projekt vytvořený v jazyce c++ winApi, který vypadal, že bude obsahovat vše potřebné pro účely této práce.



**Obr. 9:** model Skinned mesh

Model využívá výše popsaný systém kostí. Pohyb je tedy realizovaný pomocí struktury parent a child bone, což se pro naši potřebu velice hodí. První problém s tímto modelem však nastal téměř na začátku práce. Při analýze kostry tohoto modelu bylo zjištěno, že model nemá zcela identickou kostru jako standart H-Anim. Asi největší odchylka byla zjištěna v oblasti prstů. Model má totiž jednotlivé články prstů propojené, tzn. že pohneme-li posledním článkem ukazováčku, pohne se i poslední článek všech ostatních prstů. Tento nedostatek by se dal odstranit předěláním modelu. Model jde totiž exportovat do grafického editoru Blender.

---

<sup>14</sup> Dostupné z [http://msdn.microsoft.com/cs-cz/library/bb205066\(en-us\).aspx](http://msdn.microsoft.com/cs-cz/library/bb205066(en-us).aspx)

Muselo by se však udělat více úprav. Model je až zbytečně moc propracovaný v oblasti obličeje a hlavy, to má za následek velikou hardwarovou náročnost a zahlcení procesoru zbytečnými operacemi. Mnohem významnější problém, díky kterému jsem nakonec od realizace pomocí SkinnedMesh ustoupil, je složitost implementování animace. Všechny informace o modelu jsou obsaženy v jediném souboru. To znamená, že obsahuje veškeré nastavení světel a kamery. Je v něm obsažen samotný model se všemi svými vertexy, klouby i kostmi a v neposlední řadě informace o animaci. Bohužel se nepodařilo zjistit, jakým způsobem je animace v souboru zapsána a implementována. Hlavní cíl této práce - rozpohybovat model – nebyl proto naplněn, a proto bylo od použití tohoto modelu upuštěno.

## 4.2 Tvorba modelu ve VRML

### 4.2.1 Jazyk pro popis virtuální reality

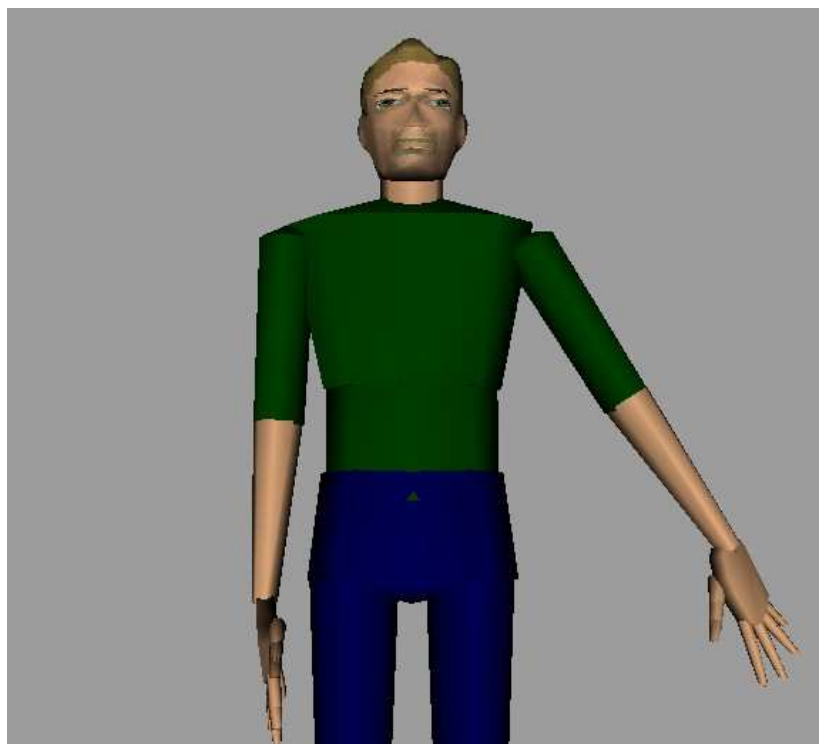
Jazykem VRML jsou definovány způsoby zápisu virtuálních světů do souborů v textovém tvaru. Tento jazyk je tedy současně i předpisem pro zapisování informací určitého typu. Můžeme ho proto v tomto smyslu přiřadit k formátům GIF, BMP nebo JPEG u zápisu obrázků či MPEG nebo AVI u zápisu videa. Jazyk VRML má jednu zásadní výhodu – nebyl vyvinut jednou firmou, ale je výsledkem společné práce mnoha firem a programátorů po celém světě. Má v sobě tedy předpoklady pro všeobecné využití jako univerzálního standardu pro popis virtuální reality.

Základní vlastnosti VRML:

- Virtuální svět tvořený 3D objekty, může být kombinován s multimediálními doplňky, např. obraz, video, zvuk.
- Při tvorbě virtuální reality můžeme využívat prvky uložené na disku našeho počítače nebo kdekoli na síti internetu. Tyto prvky můžeme navíc mezi sebou libovolně kombinovat.
- Animace, manipulace a interakce s virtuálními objekty je zajištěna jednotným a přehledným způsobem.
- Vytvořené virtuální prostředí lze vkládat do www stránek.
- Jazyk VRML umožňuje spolupráci s dalšími programovacími jazyky (Java, JavaScript).
- Výsledná virtuální realita je uložena pouze v textovém tvaru.

#### 4.2.2 Model a jeho tvorba v jazyce VRML

Jazyk VRML umožňuje skládání jednotlivých prvků a částí v jeden veliký celek. Této vlastnosti využívá i model vytvořený v tomto jazyce. Každá část těla modelu je zpracována a vymodelována samostatně a tyto části jsou potom pomocí programového zápisu složeny ve výsledný model. Celý model a všechny jeho jednotlivé části jsou dostupné na internetových stránkách<sup>15</sup>. Protože zdrojový kód modelu byl napsán tak, aby si při každém přehrání scény s modelem program stáhl potřebné prvky z internetu, což bylo velmi nepraktické, musel být program lehce upraven. Všechny potřebné prvky jsou uloženy v adresáři *bone* a ve zdrojovém kódu jsou přepsány odkazové cesty prvků, vedoucí k tomuto adresáři. Tato jednoduchá úprava výrazně zrychlila celou proceduru zobrazení modelu.



**Obr 10. :** Konečná podoba modelu

Model je vytvořen podle standardu H-Anim a obsahuje celkem 46 kostí. Názvy kostí mají zvolenou strukturu takto:

- kost se jmenuje podle části těla, kde se nachází
- pokud se jedná o tzv. párovou kost, je před jménem uvedeno slovo left nebo „L“ resp. Right nebo „R“

---

<sup>15</sup> Domovská stránka <http://ligwww.epfl.ch/~babski/StandardBody/Animation.html>

- články prstů jsou navíc označeny zkratkami *ext*, *base*, *mid*
- paže a trup jsou navíc označeny slovy *lower* a *upper*

Zdrojový kód popisující definici lokte a pažní kosti vypadá takto:

```
DEF hanim_1_elbow Transform {center 0.206777 1.166288 -0.023045
    children [ # --- Childrens of hanim_1_elbow
    DEF hanim_1_forearm Transform {
        children Inline {
            url "Bob_left_lower_arm.wrl" } ]}}
```

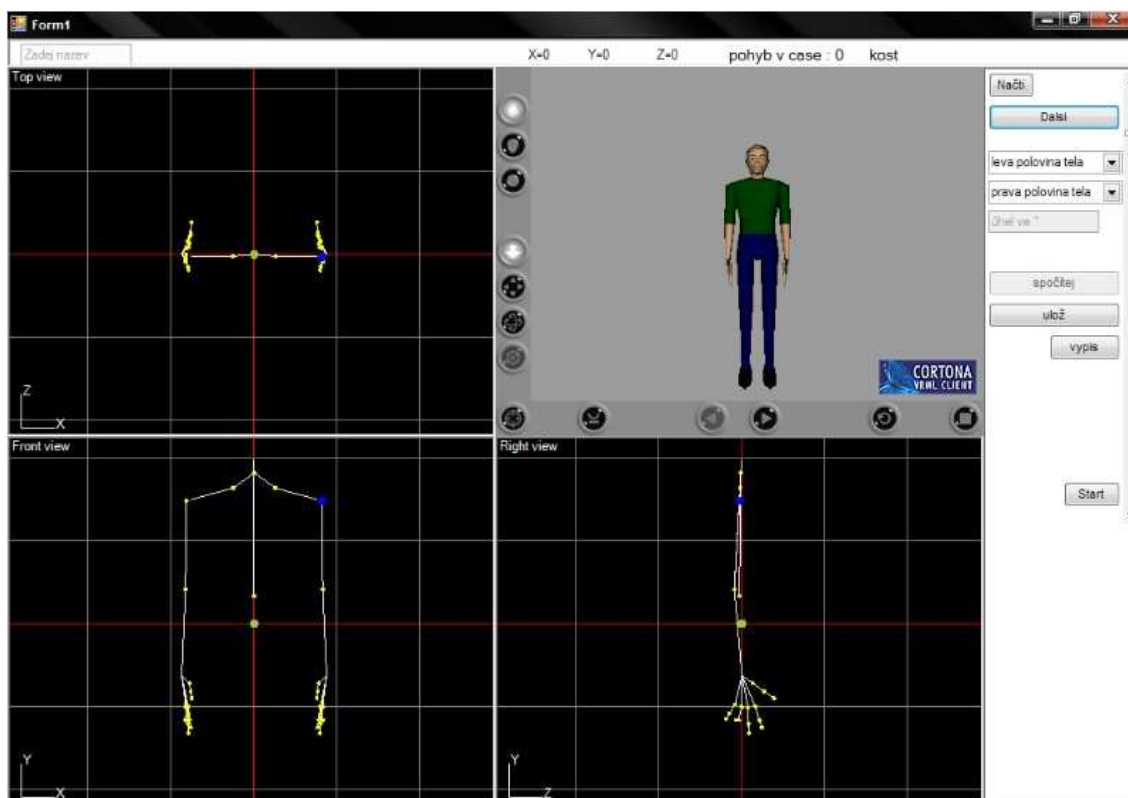
### 4.3 Tvorba animačního programu

Nyní je vytvořen model člověka, který splňuje naše požadavky. Kromě základního souboru, ve kterém jsou obsažena data o modelu, je potřeba vytvořit i soubor s daty o animaci.

Aby bylo možné tento soubor vygenerovat a model tím rozpohybovat, je třeba do vstupního souboru animace importovat data o pohybu jednotlivých kloubů. Všechny potřebné parametry jsou již uvedeny v kap. 3.2 a právě tato data budou programem počítána.

#### 4.3.1 Popis programu

Program je vytvořen ve vývojovém prostředí Microsoft Visual Studio C++ 2008 jako windows form apliation. Nejprve byl řešen problém, jakým způsobem bude uživateli model zobrazen. Pro účely animace nám postačí jednoduchý model, proto jsem zvolil pouze kostru modelu se zvýrazněnými klouby. Navíc program zobrazuje pouze klouby, které lze animovat. Je to z důvodu, že znakový jazyk využívá ke svým potřebám hlavně horní končetiny. Není proto zapotřebí zobrazovat všechny kosti a klouby. Pro dobrou orientaci při tvorbě animace resp. jakékoli grafické práce je velice důležité, jakým způsobem bude daná scéna zobrazena. V našem případě je pracovní okno rozděleno do 4 částí. Na tři pohledy 2D a to: top, front a right a na pohled celkový, který by uživateli měl dát ucelenou orientační informaci.



**Obr. 11:** rozmístění pracovních oken v programu

Další důležitou vlastností je dobrá ovladatelnost. Ovládání programu bylo zvoleno pouze pomocí myši s jedinou výjimkou a to zadávání úhlu otočení pomocí klávesnice.

**levé tlačítko myši** – umožňuje posun modelu

**pravé tlačítko myši** – provádí změnu polohy vektoru rotace, pokud je tento vektor zvolen

Dále program obsahuje několik akčních tlačítek.

**načti** – na pracovní plochy se načte model

**start editace/další** – po stisku tlačítka start editace může uživatel začít pracovat, následně bude tlačítko vykonávat příkaz *další*, na tento příkaz se uživatel posune o jeden keyframe dopředu

**spočítej** – provede se výpočet rotace a vykreslení změny na pracovní ploše

**ulož** – provede se uložení daného pohybu do souboru s animací, na toto tlačítko je nutno kliknout po každé změně v modelu

**start** – zobrazí scénu s modelem

**výběrové pole levá a pravá polovina těla** – provede se výběr kloubu, který chceme transformovat

**posuvný prvek** – slouží k přiblížení/oddálení kostry modelu

**vstupní pole** – slouží k zadání úhlu rotace ve °

### 4.3.2 Vykreslení kostry modelu

Jako zobrazovací prvky 2D pohledu jsou použity komponenty `pictureBox`. Pro kreslení na plochu obecně se v jazyce `c++` používá knihovna `graphics`. S touto knihovnou jsme schopni vykreslit jednoduché prvky, např. pouhé body, čáry, kružnice, čtverce. Knihovna `graphics` podporuje také barvy a vkládání obrázků v různých formátech. Pokud je například na plochu potřeba vykreslit obyčejnou čáru, bude použita funkce `DrawLine`, do které budou zadány vstupní parametry typ pera a  $x$  a  $y$  pozici počátku a konce. Obdobně lze vykreslit např. obdélník nebo elipsu. Více informací o problematice kreslení na plochu lze nalézt například na internetových stránkách<sup>16</sup>.

Aby mohla být kostra modelu vykreslena na pracovní plochu, potřebujeme mít definovány všechny klouby modelu, postačí nám k tomu načtení kloubů do pole. Poté jsou klouby spojeny čarami funkcí `DrawLine` a spoje, klouby, jsou zvýrazněny barevnými body.

Nyní je model zobrazen v základním zobrazení, které však není vyhovující pro tvorbu animace. Jelikož jednotlivé kosti modelu mají nepoměrnou délku kostí (pažní kost oproti článku prstu), bylo potřeba udělat vhodnou editační funkci zobrazení. Jedná se o funkci posunu a zoomu. Funkce posunu je realizována pomocí levého tlačítka myši, pokud uživatel táhne myší, model se mu posouvá daným směrem. Jedná se o jednoduchou transformaci posunutí, která je uvedena v kapitole 3.1.1. Po změně polohy myši jsou všechny zobrazené klouby přepočítány a následně znovu zobrazeny. Pro funkci zoom je použito transformace `scaling`. Postup výpočtu je obdobný jako u `translace`. Po aplikaci těchto dvou funkcí je uživateli umožněno optimálně si nastavit zobrazení kostry modelu.

Aby byla získána základní prostorová představa o modelu a vytvořené animaci, bylo zapotřebí realizovat zobrazení modelu v prostorovém promítání. Tento problém je vyřešen díky vlastnosti jazyka VRML, který umožňuje spolupráci s webovými prohlížeči. Platforma `.NET C++` obsahuje komponentu `webBrowser`, tedy okno webového prohlížeče. Do adresy, kterou bude toto okno zobrazovat je vložena cesta, která odkazuje na soubor s naším modelem a model je v tomto okně zobrazen ve 3D prostoru. Toto řešení je velice praktické. Uživatel si může rozpracovanou animaci prohlédnout, záleží sice na daném VRML prohlížeči, ale u většiny z nich se jedná o kliknutí na model, poté se animace spustí.

---

<sup>16</sup> [http://www.c-sharpcorner.com/UploadFile/mahesh/gdi\\_plus12092005070041AM/gdi\\_plus.aspx](http://www.c-sharpcorner.com/UploadFile/mahesh/gdi_plus12092005070041AM/gdi_plus.aspx)

### 4.3.3 Realizace rotace bodu kolem obecného vektoru

Po vykreslení kostry na pracovní plochu bylo třeba vyřešit zobrazování transformovaných bodů. Aby byla získána základní představa o tom, jaký pohyb byl vlastně vytvořen, musí být realizována rotace bodu kolem obecného vektoru. Rotace je realizována podle matematického postupu uvedeného v kapitole 3.1.4. Výpočetní funkce byla rozdělena na několik menších funkcí z důvodů snadnější realizace. Ukázka funkce je uvedena v příloze [1].

### 4.3.4 Tvorba souboru s animací

Pro vytvoření určitého souboru póz, tedy animace, je potřeba postupovat podle uvedeného postupu:

- 1) v červeném editačním poli nahoře je potřeba zadat název souboru s animací
- 2) načíst kostru modelu tlačítkem načti
- 3) kliknout na tlačítko start editace animace
- 4) upravit 2D pohledy na kostru modelu
- 5) výběr kloubu, který chce transformovat
- 6) úprava rotačního vektoru
- 7) zadání úhlu
- 8) klik na tlačítko spočítej
- 9) uložení pohybu
- 10) pokud chce vytvořit další pózu, klikne na tlačítko další

Všechny potřebné informace: vektor rotace, číslo pózy a vybraný kloub, jsou zobrazeny v informačním panelu nad pracovní plochou.

### 4.3.5 Podpora programu

Pro program byl zvolen programovací jazyk, který bude mít do budoucna zajištěn vývoj a podporu. Byl tedy zvolen standard jazyka C++ NetFramework. Protože se nejedná o standardně dodávaný doplněk pro windows, je potřeba si toto rozšíření na počítač doinstalovat. Pro zobrazení grafické scény s modelem je nainstalován prohlížeč VRML. Prohlížečů virtuální reality existuje mnoho, ne všechny jsou však vhodné. Doporučuji například Cortona VRML client od Paralel graphics nebo Flux player od Media Machines.



## 5 Závěr

V rámci této bakalářské práce byl vytvořen systém pro zobrazení modelu lidského těla s možností vytvářet jednotlivé znaky znakového jazyka. Systém umožňuje vytvářet jednoduché znaky, které lze uložit a následně zobrazit na trojrozměrném modelu lidského těla. Program je napsán v jazyce C++, konkrétně na platformě NetFramework. Systém je proto možné používat pouze na zařízení, které má tento standard nainstalovaný. Pro zobrazení scény s modelem je použit jazyk pro popis virtuální reality VRML.

Navržení uceleného nástroje pro vytváření jednotlivých slov znakové řeči je velmi složitý problém. V průběhu řešení této práce se ukázalo, že výsledný produkt bude velmi rozsáhlý systém, který je zapotřebí dále zdokonalovat. Program, navržený pro tvorbu znaků, není zrovna ideálním řešením, protože na vytvoření základní databáze slov znakového jazyka bychom potřebovali mnoho času. Výsledek této bakalářské práce nelze tedy chápat jako ucelený hotový produkt, ale spíše jako jeden z mnoha prvků velkého celku, který by sloužil k vizualizaci znakového jazyka.

### 5.1 Budoucnost projektu

#### 5.1.1 Propracování modelu

Vytvořený model pro vizualizaci znakového jazyka je sice plně dostačující, ale z grafického hlediska nepříliš dokonalý. Zvláště po grafické stránce by model potřeboval vylepšit, aby byl pro uživatele příjemnější. Při komunikaci neslyšících je neméně důležitým prvkem mimika obličeje a pohyby rtů. Tento model daný problém neřeší vůbec, do budoucna by bylo proto vhodné model zdokonalit v obličejové části a problém odstranit.

#### 5.1.2 Animace znaků

Pro zobrazení jednodušších znaků je navržený program dostačující. Avšak při tvorbě složitějších slov, či slovních spojení narazíme na problém. Bylo by dobré vyřešit propojení jednotlivých slovních částí tak, aby na sebe volně navazovaly. Jak je uvedeno v kapitole 1.1, některá slova jsou složena z více znaků. Ty jsou však mluvčím znakovou řečí spojena dohromady stejně jako při mluvené formě. Programem pro tvorbu animací lze jednotlivé znaky pouze vytvářet, ne však editovat, či kopírovat a používat pro jiná podobná slova. Program by bylo proto dobré rozšířit o editační mód.

Další prvek, který by bylo zapotřebí promyslet a zdokonalit, je reálnější zobrazení pohybů. Při zobrazování se totiž využívá jednoduchého pohybu po kružnici, reálný pohyb rukou při znakování je složen z mnoha pohybů. Program by bylo vhodné doplnit o předpřipravené typy pohybů právě po těchto křivočarých drahách. To by však vyžadovalo mnohem větší počet zobrazených póz. Pohyb po křivce by musel být rozdělen na malé, částečné pohyby po přímkách, které by spojeny v jeden celek utvářely dojem pohybu po křivce.

### **5.1.3 Instalační balíček**

Jelikož program není napsán pro platformu standardně dodávanou v balíčku služeb systému windows a pro zobrazení animace, je zapotřebí nainstalovat program pro prohlížení. Bylo by dobré vytvořit jakýsi instalační balíček, který by všechny tyto produkty nainstaloval do počítače najednou. Nastal by zde zřejmě problém s kolizí o autorských právech na instalované produkty. Určité řešení by bylo například v postupné instalaci jednotlivých služeb a balíčků, která by probíhala sama a automaticky.

### **5.1.4 Tvorba databáze**

Dalším problémem je vytvoření rozsáhlé databáze slov a jejich znakových ekvivalentů. Na tvorbu takové databáze není program navržen. Vytvoření finální podoby jednoho slova se všemi náležitostmi je dost časově náročné. Tento problém by mohl být vyřešen pomocí snímání pohybů rukou, prstů a obličeje skutečného člověka. Model pro zobrazení na počítači by mohl zůstat stejný, muselo by se ale vyřešit snímání pohybů a jejich převod do grafické podoby. Bylo by zřejmě potřeba vytvořit program, který by tento problém řešil. S tímto nástrojem by tvorba databáze slov a znaků byla mnohem efektivnější a rychlejší.

### **5.1.5 Syntéza řeči**

Dalším vhodným rozšířením by bylo doplnění modelu o syntézu řeči. Při animaci daného znaku by se zároveň přehrávala jeho mluvená forma. Toto by bylo využitelné například v interaktivním programu pro výuku znakového jazyka.

## Seznam použité literatury

- [1] Yeats S., Holden E. J., Owen R.: *Real-Time 3D Graphics for Human Modelling and Teaching Sign Language*, ICCVG 2002, Zakopane
- [2] Mgr. Marie Růžičková: *Učíme se českou znakovou řeč*, SEPTIMA 1997, Praha
- [3] Obrátil J.: *Vizualizace znakové řeči*, TUL 2006, Liberec
- [4] Potměšil M.: *Všeobecný slovník českého znakového jazyka*, 1 vydání A-N. Fortuna 2002, Praha
- [5] Žára J.: *VRML97 Laskavý průvodce virtuálními světy*, Computer Press 1999, Brno
- [6] Žára J., Beneš B., Sochor J., Felkel P.: *Moderní počítačová grafika*, Computer Press 2004, Brno
- [7] Kruglinsky D., Shepherd G., Wingo S.: *Programujeme v Microsoft Visual C++*, Computer Press 2000, Praha
- [8] Humanoid animation working group, H-Anim, [Online; navštíveno 25.4. 2008]. Dostupné z: <http://www.h-anim.org>
- [9] Západočeské univerzita v Plzni, Syntéza znakového jazyka, [Online; navštíveno 29.4. 2008]. Dostupné z: <http://musslap.zcu.cz/cs/synteza-znakoveho-jazyka>
- [10] Matišek D., Geometrické transformace, [Online; navštíveno 2.3. 2008]. Dostupné z: [http://mathonline.fme.vutbr.cz/pg/flash/PgDelphi/08\\_TRANSFORMACE.htm](http://mathonline.fme.vutbr.cz/pg/flash/PgDelphi/08_TRANSFORMACE.htm)
- [11] Tišnovský P., VRML: Jazyk pro popis virtuální reality, [Online; navštíveno 2.5. 2008]. Dostupné z: <http://www.root.cz/clanky/vrml-jazyk-pro-popis-virtualni-reality>
- [12] Babski C., VRML Animation Gallery, [Online; navštíveno 10.4. 2008]. Dostupné z: <http://ligwww.epfl.ch/~babski/StandardBody/Animation.html>
- [13] Informatic & Telematic Institute., VSigs, [Online; navštíveno 10.4. 2008]. Dostupné z: <http://vsigns.iti.gr:8080/VSigns>
- [14] kolektiv autorů, Znakový jazyk, [Online; navštíveno 15.4. 2008]. Dostupné z: <http://ruce.cz/clanky/7>

- [15] Wikipedia, Sign language, [Online; navštíveno 15.4. 2008]. Dostupné z: [http://en.wikipedia.org/wiki/Sign\\_language#History\\_of\\_sign\\_language](http://en.wikipedia.org/wiki/Sign_language#History_of_sign_language)
- [14] Inc. Vcom3D, Authoring with studio, [Online; navštíveno 15.4. 2008]. Dostupné z: <http://www.vcom3d.com/Studio.htm>.
- [15] Microsoft Corporation, SkinnedMesh Sample, [Online; navštíveno 10.2. 2008]. Dostupné z: [http://msdn.microsoft.com/en-us/library/bb147387\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/bb147387(VS.85).aspx)

## Příloha

[1] Ukázka zdrojového kódu funkce pro výpočet rotace v prostoru.

Rotace je realizována podle matematického postupu uvedeného v kapitole 3.1.4. Výpočetní funkce byla rozdělena na několik menších funkcí z důvodů snadnější realizace.

```
public: void GenRot(){
u[0]=-a[0]; u[1]=-a[1]; u[2]=-a[2];
inictrans(); // nastavení translace T(u=AO)
beta = (-Math::Atan((b[1]-a[1])/(b[0]-a[0]))); // výpočet úhlu beta
gama=Math::Atan(Math::Sqrt((Math::Pow(b[0]-a[0],2)+Math::Pow(b[1]-
a[1],2)))/(b[2]-a[2])); // výpočet úhlu gama
inicrotz(); //nastavení rotace R(z,beta)
Multip(trans,rotz;mat); // přiložení rotace R(z,beta)
inicroty(); // nastavení rotace R(y,gama)
Multip(mat,roty;mat); // přiložení rotace R(y,gama)
alfa = Math::PI*(uhel)/180;
inicrot(); // nastavení rotace R(z,alfa)
Multip(mat,rota;mat);
gama = -gama;
inicroty();
Multip(mat,roty;mat);
beta = -beta;
inicrotz();
Multip(mat,rotz;mat);
u[0]=a[0];u[1]=a[1];u[2]=a[2];
inictrans();
Multip(mat,trans;mat);
MultipMV(mat,bod;vyslbod);
```