

Univerzita Palackého v Olomouci

Pedagogická fakulta

Katedra technické a informační výchovy

Přemysl Geryk

Počítačová grafika a animace v 3D prostoru

Bakalářská práce

Olomouc 2013

vedoucí práce: Mgr. Jan Kubrický

Prohlašuji, že předložená práce je mým původním autorským dílem, které jsem vypracoval samostatně. Veškerou literaturu a další zdroje, z nichž jsem při zpracování čerpal, v práci řádně cituji a jsou uvedeny v seznamu použité literatury.

V Olomouci dne

.....

(podpis)

Děkuji vedoucímu práce, Mgr. Janu Kubrickému, za odborné vedení práce a cenné rady poskytnuté při vypracování této práce. Také bych chtěl poděkovat svým rodičům za podporu a podnětné připomínky k této práci.

Obsah

Úvod.....	5
I. Teoretická část	6
1 3D grafika a animace	6
2 Techniky modelování trojrozměrných objektů.....	7
3 Vlastnosti povrchu a textury	8
4 Techniky používané k tvorbě 3D animací	10
5 Osvětlení a renderování 3D scény	12
6 Techniky synchronizace animace	14
7 Software pro 3D animaci - Blender	16
7.1 Historie.....	16
7.2 Charakteristika	16
8 Programovací jazyk Python a rozhraní MIDI.....	17
8.1 Programovací jazyk Python	17
8.2 MIDI (Music Instrument Digital Interface)	17
II. Praktická část	18
9 Návrh scény	18
10 Modelování	19
10.1 Postava	19
10.2 Baskytara	21
10.3 Klavír	23
10.4 Bicí.....	25
10.5 Scéna.....	27
10.6 Světla	28
11 Materiály a textury	30
12 Příprava animačních akcí.....	31
13 Návrh skriptu	34
13.1 Soubor midi.py.....	34
13.2 Soubor __init__.py.....	36
14 Hudební podklad a synchronizace	38
15 Renderování	40
Závěr	41
Seznam zdrojů a literatury	42
Seznam příloh	43
Anotace	50

Úvod

S vývojem počítačových technologií se rozšířily, a stále rozšiřují, také možnosti počítačové grafiky. Ta našla své uplatnění v množství různých odvětví a oborů od designu přes zábavní, stavební, elektrotechnický průmysl, strojírenství až po lékařství. Současná technologie nejenže usnadňuje práci vývojářům, grafikům a designérům, ale vytváří prostor pro nové obory nebo ty současné posouvá na úplně jinou úroveň. Ještě větší možnosti nabízí práce s grafikou v trojrozměrném prostoru, která dnes zasahuje do většiny oborů lidské činnosti. To se děje především v oblasti návrhů, což je běžnému člověku skryto. Kde je však viditelná pro všechny, je zábavní resp. filmový průmysl, a to hlavně ve formě animací.

Hlavním cílem této bakalářské práce je naprogramování skriptu, řešícího problém se synchronizací animace pomocí midi souboru a vytvoření ukázkové animace. A to konkrétně v jazyku Python s implementací pro software pro tvorbu 3D animací, Blender. Dílčím cílem je poskytnout stručný náhled do problematiky tvorby 3D animací.

I. Teoretická část

1 3D grafika a animace

3D grafika je významným odvětvím počítačové grafiky. Zkratka 3D (3 dimensional) značí, že se jedná o práci ve trojrozměrném prostoru. Ten lze popsat např. kartézskou soustavou souřadnic nebo pomocí sférického (azimutálního) systému. V praxi se většinou setkáme s jejich kombinací. U animace přibývá čtvrtý rozměr, kterým je čas.

Principem 3D grafiky je práce s trojrozměrnými geometrickými daty a jejich následné renderování (viz kapitola Renderování) do 2D obrázků, které již nenesou trojrozměrná data, neboť slouží k zobrazení na dvourozměrných médiích (papír, počítačový monitor, filmové plátno aj.). Nejpopulárnějším využitím 3D grafiky je dříve zmíněná tvorba animací.

Historie 3D grafiky se datuje od roku 1960, kdy William Fetter¹ při popisu své práce pro Boeing zavedl pojem počítačová grafika, což je považováno za základní milník historie počítačové grafiky. Je zároveň autorem prvního modelu lidské postavy, vytvořeného pro ergonomické studie kokpitů roku 1964. Prvním interaktivním grafickým systémem byl Sketchpad² a prvním komerčním softwarem pro animaci plně renderovaných polygonů byl Synthavision společnosti MAGI [1, s. 15-25].

V 80. letech 20. století nastává velký boom v oblasti 3D grafiky a animace. Díky příchodu nových výkonnějších a dostupnějších technologií si počítačová grafika nachází místo v komerční sféře.

¹ William Fetter - designér firmy Boeing

² Sketchpad - autorem je Ivan Sutherland z institutu MIT

2 Techniky modelování trojrozměrných objektů

Vytváření skulptur, popis prostoru, a rozmístování virtuálních trojrozměrných objektů, prostředí a scén počítačovým systémem se označuje za modelování [1, s. 103].

Tyto objekty jsou vytvářeny člověkem pomocí softwarových modelovacích nástrojů nebo na základě počítačových simulací a matematických popisů. Existují dva typy reprezentace těles:

- **Obrysová** - těleso je popsáno jako mnohostěn určený svými hranicemi (body, hrany, plochy) a je duté. Jedná se o nejčastější reprezentaci těles určenou především pro následné renderování, kde není objem tělesa podstatný.
- **Objemová** - tento typ reprezentace pracuje i s informací o objemu. Těleso tedy přestává být pouhou schránkou. Jeho využití je především při simulacích.

Mezi základní objekty či nástroje všech modelovacích programů lze zařadit tzv. geometrická primitiva. Jejich nabídka se samozřejmě v každém programu mírně liší, ale téměř vždy jsou zastoupeny rovina, krychle, koule, válec, kužel a z dvojrozměrných objektů křivky, kružnice či mnohoúhelníky. Označují se za primitiva, neboť se jedná o jednoduché objekty, matematicky snadno popsatelné. Komplexních modelů lze dosáhnout pomocí booleovských operací nebo postupnou úpravou jejich tvaru pomocí standardních nástrojů jako jsou řez, vysunutí, posun, změna velikosti, rotace aj. Velmi oblíbeným výchozím primitivem je krychle, kvůli její jednoduchosti a symetričnosti ve všech třech osách. Díky této skutečnosti se ujal termín "box modeling", který označuje právě techniku, při které se jako výchozí objekt využívá geometrické primitivum, zejména krychle.

Jako zajímavou techniku je třeba zmínit procedurální modelování. Její podstata spočívá v tom, že objekty takto vytvářené lze popsat jednodušší či složitější matematickou funkcí nebo algoritmem, často s několikanásobnou rekurzí, které by pomocí technik geometrického modelování bylo možno vytvořit jen těžko nebo vůbec ne kvůli jejich složitosti a množství. Tato technika se týká především fraktálních obrazců, částicových systémů s přirozeným růstem (tráva, rostliny a další přírodní tvary) nebo simulací látek (kapaliny, textil, mlha atd.) [1, s. 128-166].

3 Vlastnosti povrchu a textury

Vzhled prostředí a objektů je určen procesem stínování, který je založen na simulaci odrazů světla matematickými výpočty. Podle použitého algoritmu se rozlišuje několik typů stínování. Difuzní stínování je nejjednodušším typem. Pro výpočet využívá pouze normály jednotlivých polygonů, na jejichž základě přiřadí každému polygonu konstantní hodnotu stínování. Zrcadlové stínování k výpočtu používá normály polygonů i normál vrcholů a vytváří interpolaci mezi nimi. Na základě této interpolace o úhlu dopadu světla se vypočte hodnota stínování pro každý bod povrchu. Vyhlazené stínování, též zvané Gouraudovo, vychází z myšlenky zprůměrování normál sousedních polygonů. To má za následek přesáhnutí hranic polygonů a vytvoření dojmu hladkého přechodu stínování mezi polygony. Stínování zastínění okolím je technikou vytvářející jemné postupné stínování. Při výpočtu pracuje s přímým i odraženým světlem a počítá množství světla, které se na povrch nedostane.

Základními vlastnostmi všech povrchů jsou barva, odrazivost a průhlednost. Barvu lze definovat pomocí řady barevných modelů (RGB, HSL, CMYK), a to buď jako jednu barvu, barevný přechod nebo lze využít obrazových map (textur). Difuzní a zrcadlová odrazivost určuje matnost či lesklost povrchu a míru jeho zrcadlového efektu. Intenzitu odrazivosti lze obdobně jako barvu definovat obrazovou mapou. Vhodným nastavením a kombinací s barvou lze simulovat velké množství materiálů jako je dřevo, plast, lak, kov apod. Průhlednost je klíčovou vlastností pro simulaci materiálů, jako je sklo nebo voda. I průhlednost lze samozřejmě definovat obrazovou mapou. Je třeba zmínit, že velkou roli při definování průhlednosti a odrazivosti hraje index lomu světla, který ve značné míře dodává materiálům na realistickém vzhledu.

V souvislosti s použitím obrazových map se mluví o texturování, a to vizuálním nebo prostorovým. Do kategorie vizuálních textur patří výše zmíněné barevné mapy, mapy odrazivosti a mapy průhlednosti, které ovlivňují pouze vzhled. Oproti tomu prostorové textury mají vliv na tvar objektu. Do této kategorie se řadí mapy hrbolati a posunu. Mapy hrbolati (bump maps) sice neovlivňují tvar objektu přímo, ale mění směr normál. Tím dochází ke změnám stínování a je vytvořena iluze hrbolatého povrchu. Nevýhodou této techniky je, že při detailních záběrech ztrácí povrch svůj hrbolatý vzhled. Druhou možností jsou mapy posunu, které v závislosti na intenzitě barvy přímo posouvají jednotlivé body podél zadané osy. Často se mapy posunu

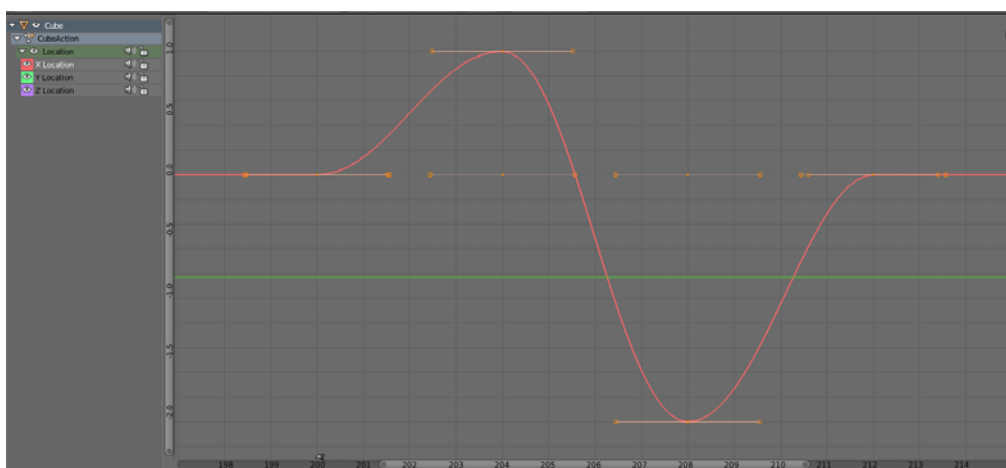
používají pro tvorbu trojrozměrných krajin s horami a údolími, ale stejně dobře jsou použitelné např. na modelování soustružených drážek či vinutí drátu.

Obrazové mapy lze na objekty umístit několika různými projekčními metodami. Jednou z nejpoužívanějších metod je rovinná projekce, která promítá obraz na rovinu. Z toho vyplývá, že se výborně hodí na ploché objekty, neboť nedochází k deformaci obrazu (pokud je povrch objektu rovnoběžný s promítací rovinou). Při použití na zakřivené objekty se obraz deformuje a dochází k efektu připomínajícímu promítačku. Velmi podobnou metodou je projekce na krychli, která se chová jako projekce rovinná s tím rozdílem, že promítá obraz na všech šest stran krychle. Projekce na válec a na kouli jsou si velmi podobné. U obou se obrazová mapa obalí kolem celého objektu, až se protilehlé strany mapy setkají. Rozdíl je v tom, že u válcové projekce se může promítnout obrazová mapa ještě na horní a spodní stranu, kdežto u projekce na kouli se mapa napne směrem k pólům, kde se sepne. Velmi efektivní metodou je umístování obrazových map na objekt pomocí UV souřadnic. Tato metoda funguje na principu rozložení sítě objektu a jeho promítnutí do roviny, která používá označení souřadných os jako U a V, neboť X a Y jsou použity pro definování 3D prostoru. Následné umístění obrazové mapy již probíhá rovinnou metodou. Pro usnadnění existují různé nástroje pro rozdělování objektů na menší celky a jejich úpravy [1, s. 261-291].

4 Techniky používané k tvorbě 3D animací

Animovat v podstatě znamená vdechnout život do neživého předmětu, obrázku nebo kresby (anima v latině znamená duše). Animace je umění vyjadřující pohyb pomocí obrázků, které nepocházejí přímo z reálného světa. Iluze pohybu je při ní vytvářena na základě velmi rychlého zobrazování sekvence velkého počtu statických obrázků nebo snímků [1, s. 305].

Klíčování je základní animační technikou. Spočívá v definování animované sekvence pomocí klíčových snímků a následnou interpolací mezi nimi. Jako klíčové se zpravidla volí extrémy daného pohybu. Jakmile jsou určeny klíčové snímky, následuje doplnění mezifází. Interpretuje se zanesením do Gaussovy roviny, kde vodorovná osa značí čas a svislá hodnotu parametru (např. poloha objektu v ose x). Klíčovými snímky se proloží tzv. interpolační (IPO) lineární nebo obecná křivka, kterou lze následně upravovat (Obr. 1).



Obr. 1 - IPO křivka

Pohyb po dráze je další ze způsobů animace objektů v trojrozměrném prostoru. Prvním krokem je mít objekt, který se má pohybovat, a křivku, která určuje dráhu. Druhým krokem je přiřazení objektu ke křivce a nastavení rychlosti pohybu. To se většinou zadává pouze dvěma body, a to počátkem a koncem. Výsledný pohyb bývá v této chvíli v čase lineární. V případném třetím kroku tedy v případě potřeby dochází úpravou interpolační křivky k nastavení požadovaného zrychlení v pohybu, je-li to třeba [1, s. 341-349].

Pohybová kostra je podpůrným aparátem k deformaci složitějších modelů (např. postav či zvířat). Je založena na definování zjednodušené kostry pomocí kostí a kloubů a jejich hierarchické struktury, která je velmi podobná té skutečné. Tato kostra je následně přiřazena modelu, který se deformuje podle této kostry. Přirozené lidské či zvířecí pohyby jsou však natolik komplikované, že jejich dokonalosti nelze dosáhnout pouze pomocí kostry. Proto se v praxi často používají i svalové aparáty propojené s těmito kostrami [2, s. 52-57].

Dopředná kinematika (*forward kinematics - FK*) je sice jednoduchou, ale poměrně pracnou technikou pro animování složených postav a objektů. Její největší výhodou je prakticky absolutní kontrola nad daným objektem. Princip této techniky spočívá v tom, že se pozice a úhel otočení každého kloubu, který se v daném klíčovém snímku má pohybovat, určuje zvlášť. I přes svou časovou náročnost je nenahraditelnou, neboť nabízí možnost velmi přesného nastavení, které u inverzní kinematiky nebo pohybu po dráze nelze dosáhnout.

Inverzní kinematika (*inverse kinematics - IK*) patří mezi pokročilé animační techniky. Používá se především pro animaci postav nebo pospojovaných objektů, které mají definovanou hierarchickou kostru. Hierarchické řetězce určené pro IK se skládají z kořene řetězce (obvykle první kloub řetězce, jehož poloha není IK ovlivněna), koncového efektoru (kloub určující umístění konce pohybujícího se řetězce) a mezičlánků. Když se efektor posune, IK dopočítá polohy a otočení kloubů mezi efektozem a kořenem. Použitím omezení pohybu jednotlivých kloubů se zamezí nepřirozeným či nechtěným pohybům [1, 373-376].

5 Osvětlení a renderování 3D scény

Renderování (rendering) je proces vykreslující výsledné 2D snímky na základě 3D modelů, jejich polohy, povrchových vlastností, nastavení světel ve scéně a umístění kamery. Pro požadovaný vzhled scény se používají různé techniky simulující šíření světla. Mezi nejčastější patří:

- **Trasování paprsků** - funguje na principu sledování paprsků světla pro každý pixel ve scéně. Z praktického hlediska má smysl sledovat pouze paprsky, které dosáhnou kamery (směr sledování je od kamery ke zdroji světla). Tato technika pracuje s povrchovými vlastnostmi objektů (odrazivost, zrcadlení, index lomu, průsvitnost, barva aj.) a počtem dovolených odrazů či průchodů paprsků (tzv. hloubka trasování). Z toho plyne, že čím větší je hloubka trasování, tím detailnější a realističtější obrázek vznikne.
- **Radiozita** - technika používaná při globálním osvětlení scény. Princip spočívá v rozdělení všech objektů ve scéně na mnohoúhelníky (tzv. clustery), které se dělí na zdroje světla, povrchy přijímající a povrchy blokující světlo. S využitím iteračních technik se počítá množství světla přeneseného mezi jednotlivými povrchy v tolika opakováních, dokud není energie světla povrchy úplně pohlcena. S použitím radiozity lze dosáhnout fyzicky velmi přesného rozptylu světla ve scéně. Nedosahuje však tak realistického vzhledu jako v případě trasování.

Většinou se renderovací techniky navzájem kombinují a scéna se renderuje v několika samostatných vrstvách s různými parametry (např. místnost a nábytek pomocí radiozity a hlavní postava trasováním). Tím lze dosáhnout zvýšení realistického vzhledu výsledného snímku a zároveň kratšího času potřebného k jeho renderování [1, s. 189-192].

Pro uspokojivý výsledek renderování je důležité správné nasvícení scény. Tradičními technikami nasvícení scény je třibodové osvětlení, které používá klíčové, výplňové a uvolňovací či okrajové světlo, a čtyřbodové osvětlení, které navíc ještě zahrnuje osvětlení pozadí. Klíčové světlo je ve scéně dominantním osvětlením objektu zájmu a určuje celkovou náladu scény. Úlohou výplňového světla, také označovaného

za okolní, je zejména klíčové světlo, jím vrhané stíny a pomáhá definovat barevný tón scény. Výplňová světla nevrhají stíny. Uvolňovací a okrajová světla se používají ke zdůraznění tvaru a objemu hlavního objektu zájmu. Základními parametry určujícími vlastnosti světla jsou poloha, barva, intenzita, dosvit, slábnutí a úhel světelného kuželu. Základními typy světelných zdrojů jsou všesměrové, kuželové, nekonečné a plošné světlo. Liší se od sebe hlavně úhly, které svírají jejich paprsky [1, s. 236-250].

6 Techniky synchronizace animace

Snímání pohybu (motion tracking) je technikou používanou především k tvorbě realistických pohybů, která se objevuje hlavně ve filmovém průmyslu. Záznam pohybu lze realizovat různými způsoby a za pomoci různých technologií, které je předurčují k použití v různých oblastech. Protetické snímání je založeno na principu převodu kinetické energie na energii elektrickou za použití potenciometrů umístěných na pohyblivé konstrukci. Omezení této technologie je zapříčiněno objemností potenciometrů, potažmo samotné konstrukce, což omezuje pohyb samotné postavy. Tato technologie má však výhodu velké přesnosti získaných dat, díky čemuž má své uplatnění v lékařství. Jako pravděpodobně nejrozšířenější se dá označit technologie optického snímání. Herci (postavy) mají na sobě umístěny optické senzory nebo značky na specifických místech těla (kloubech) herce. Počet těchto značek se pohybuje v rozmezí od 12 do 70 a celý pohyb je zaznamenáván několika kamerami z různých úhlů, aby se předešlo překrývání těchto značek. Výsledný záznam je poté zpracován a pohyb jednotlivých značek je vyhodnocen vzhledem ke statickým bodům scény a použité perspektivě. Tato technologie neomezuje hercův pohyb a díky několika kamerám je dosti přesná, proto i hojně využívána v různých odvětvích. Současné softwary pro snímání pohybu jsou dokonce částečně schopny pracovat i bez těchto značek a sledovat specifické body na těle člověka (konečky prstů, oči, paže, ramena atd.) Mezi další technologie snímání pohybu patří akustické a magnetické senzory, gyroskopické senzory, akcelerometry aj., často používané v mobilních telefonech, tabletech či herních konzolách.

Animace řízená uživatelem se týká především oblasti real-time animace. Na tomto způsobu jsou založené počítačové a konzolové hry, ale objevuje se i v jiných interaktivních aplikacích. Princip je založen na spouštění animačních akcí na základě hodnot získaných z různých vstupních zařízení (myš, klávesnice, herní ovladače či pohybové senzory). Vzhledem k tomu, že se jedná o renderování v reálném čase, je třeba vytvořit modely tak, aby byly maximálně realistické a zároveň minimálně náročné na výpočetní čas.

Animace řízená pomocí midi je technikou navazující na animace řízené uživatelem. Využívá se při ní jako vstupního zařízení midi rozhraní. Je tedy možné tohoto rozhraní využít pro real-time animaci nebo jako vstup použít midi soubor.

Jednotlivým animačním akcím jsou přiřazeny konkrétní midi události (note on, note off, změna hlasitosti aj.). Technika s použitím midi souboru se hodí především k tvorbě animací doprovázených hudebním podkladem. Stejně dobře se ale dá využít v jakékoliv animaci, kde se často opakují akce s nutností přesného načasování. Midi se pro tento účel velmi dobře hodí, neboť jej lze snadno vytvořit, modifikovat a hlavně pro každou notu určuje velmi přesně v čase. Proto je možné pomocí midi výrazně zkrátit čas potřebný pro tvorbu animací s velkým množstvím akcí se zárukou přesné synchronizace v časové ose. Minimálně takto lze vytvořit přesně časovaný základ animace a jednotlivé pohyby následně přizpůsobit ručně [1, s. 376-384].

7 Software pro 3D animaci - Blender

7.1 Historie

Program Blender, za jehož vznikem stojí spoluzakladatel holandského animačního studia NeoGeo, Ton Roosendaal, vznikl v roce 1995 jako interní firemní software. Roosendaal měl myšlenku o šíření Blenderu zdarma a společně s malou skupinou bývalých spolupracovníků založil nekomerční společnost Blender Foundation, jejímž cílem bylo nalézt způsob, jak vyvíjet Blender jako Open Source projekt. Založením sbírky "Free Blender" získali 100 000 € potřebných k odkoupení práv a 13. října 2002 byl Blender vydán pod licencí GNU-GPL. Od té doby pokračuje vývoj tohoto softwaru velmi rychlým tempem.

7.2 Charakteristika

Blender je integrovaná sada nástrojů umožňující tvořit v široké oblasti 3D grafiky. Navíc je Blender multiplatformní program a velikost instalačního souboru je okolo 10 MB. Zaměřený na profesionály a umělce, může být Blender používán pro tvorbu 3D vizualizací, obrázků nebo HQ videa, které po spojení vytvoří 3D interaktivní obsah, který lze otevřít i bez pomoci Blenderu.

Hlavní rysy:

- *Plně integrovaná sada obsahující širokou paletu nástrojů nezbytných pro 3D tvorbu, jako je: modelování, UV mapování, texturování, animování, simulace, skriptování, renderování, post-produkci videa a tvorbu her.*
- *Multiplatformní aplikace s rozhraním založeným na OpenGL, kterou lze provozovat na: Windows, Linux, OSX, FreeBSD, Irix, Solaris a mnoho dalších.*
- *Umožňuje rychle a produktivně vytvářet velmi kvalitní 3D architekturu.*
- *Malý spustitelný soubor pro snadnější distribuci [8].*

8 Programovací jazyk Python a rozhraní MIDI

8.1 Programovací jazyk Python

Jazyk Python začal vznikat v roce 1989 ve výzkumném ústavu v Amsterdamu. Při jeho zrodu stál Guido van Rossum a je vidět, že u návrhu dostatečně přemýšlel. Vznikl promyšlený jazyk, který je stále ve vývoji. Jméno dostal podle pořadu BBC Monty Python's Flying Circus [9].

Python je jednoduchý a přehledný programovací jazyk, který je interpretovaný, což znamená, že je multiplatformní. Je expresivní, což znamená, že stejný program má v Pythonu výrazně kratší kód než třeba v C++ nebo Javě. Jednou z velkých výhod Pythonu je široká nabídka základních knihoven, které pokryjí většinu běžných operací a zkrátí samotný kód na pouhých pár řádků. Stejně tak nabízí širokou škálu specializovaných knihoven pro specifické oblasti. Python je určen především pro objektově orientované programování, stejně dobře ale může být použit k programování procedurálnímu [3, s. 1].

8.2 MIDI (Music Instrument Digital Interface)

Jak již samotný název napovídá, MIDI je komunikačním rozhraním vyvinutým pro komunikaci mezi digitálními hudebními nástroji a počítačem. Komunikace probíhá v bytekódu posíláním standardizovaných instrukcí. Tento proud instrukcí lze samozřejmě zaznamenat a uložit do souboru SMF (standard MIDI file) s přesně daným formátem dat. Každý takový soubor obsahuje hlavičku s údaji o formátování a jednu nebo více stop (Track), každá s maximálně 16 kanály (channel) [10].

Každá stopa se skládá z řady MIDI událostí (MIDI-events), z nichž nejdůležitější jsou Note-on (0x9) a Note-off (0x9). Za těmito událostmi následují základní vlastnosti noty v pevně daném pořadí a rozsahu. Méně často se pak objevují události změny tempa, výšky noty, nástroje aj [11].

Zápis Note-on / off má tuto strukturu:

Note-on	Channel	Note Number	Velocity
0x9	0 - 15	0 - 127	0 - 127
Note-off	Channel	Note Number	Velocity
0x8	0 - 15	0 - 127	0 - 127

II. Praktická část

Jelikož práce s grafikou je především o praxi, která staví na teoretických základech, obsahuje i tato práce část praktickou. Ta má spíše popisný charakter a přibližuje celý postup tvorby animace řízené pomocí midi souboru. Celý popis je členěn do kapitol v návaznosti na předchozí teoretickou část. Obsahově zahrnuje všechny kroky od prvotního návrhu, přes modelování, programování, tvorbu animačních akcí až po finální rendering.

9 Návrh scény

K této práci mě inspirovala série videí s názvem Animusic³, která jsou založena na synchronizaci animace s midi podkladem. Cílem tedy bylo naprogramovat v Pythonu skript pro software Blender, který by umožnil něco podobného, a k tomu vytvořit ukázkovou demo animaci. Abych se vyhnul přílišné složitosti, použil jsem v animaci pouze 3 nástroje (bicí, baskytara a klavír). Tyto nástroje doplňují dvě figuríny člověka a skupina světél, na kterých lze předvést, že řízení pomocí midi se dá využít i na jiné modely než hudební nástroje a zároveň vytvoří náladu celé scény. Všechny tyto objekty bylo nutno umístit do vhodného prostředí. Navrhl jsem vlastní pódium, které je na pomezí klubové scény a koncertního pódia. Dále bylo nutno vytvořit hudební a midi podklad, který by zapadal do tohoto konceptu.

³ Animusic je americká společnost zabývající se tvorbou 3D vizualizací hudby a animací založených na MIDI.

10 Modelování

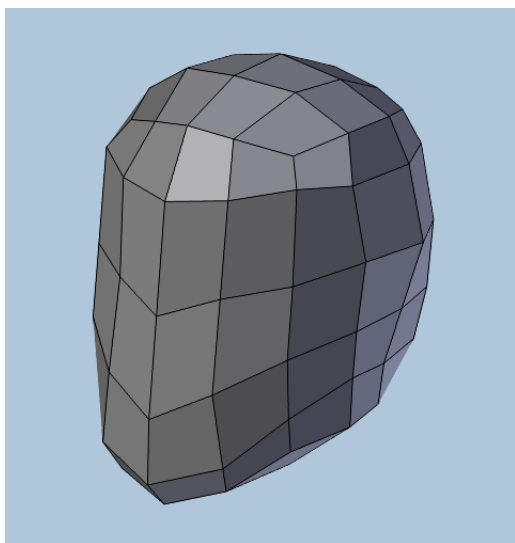
Tato kapitola shrnuje postupy použité k modelování jednotlivých objektů, a to od výběru vhodných referenčních obrázků, přes konkrétní nástroje, techniky a postupy až k závěrečným detailům.

10.1 Postava

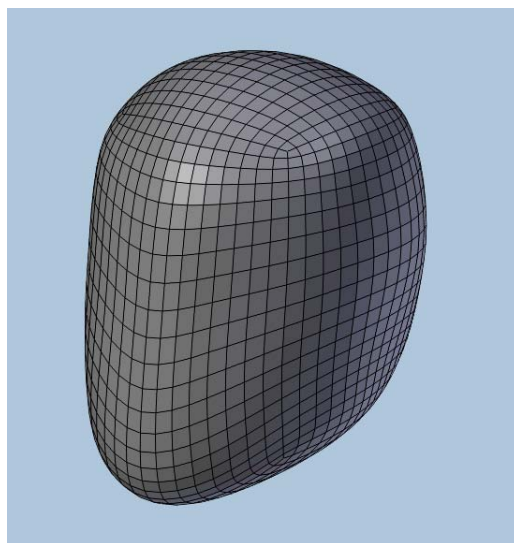
Pro model postavy člověka jsem zvolil blokový vzhled připomínající figurínu, tvořenou 49 oddělenými částmi. Jako předlohu jsem použil skicu člověka (viz Příloha č. 3) nabízející pohled zepředu, zboku a zezadu. Tato postava je tvořena ze samostatných bloků a referenční obrázek slouží pouze k dodržení proporcí lidského těla.

Jako výchozí objekt pro všechny části jsem použil krychli, která je díky pravidelnosti a rovnoměrnému rozložení bodů velmi vhodná. Jejich osm bodů samozřejmě není pro vymodelování požadovaného tvaru dostačujících, a proto je potřeba jejich počet zvýšit příkazem *Subdivide* nebo pomocí nástroje *Loop cut*.

Při modelování hlavy vycházím z krychle. Jelikož se jedná o symetrický objekt, odstranil jsem body na levé polovině krychle a použil modifikátor *Mirror*, jenž automaticky vytváří symetrický obraz poloviny pravé. Umístil jsem jednotlivé body v prostoru tak, aby jejich spojnice co nejlépe kopírovaly tvar hlavy na referenčních obrázcích. Na hotový tzv. "low-poly" model (Obr. 2) jsem použil příkaz *Smooth* a modifikátor *Subdivision surface* a doladil jednotlivé body tak, abych dosáhl co nejlepšího výsledku (Obr. 3).

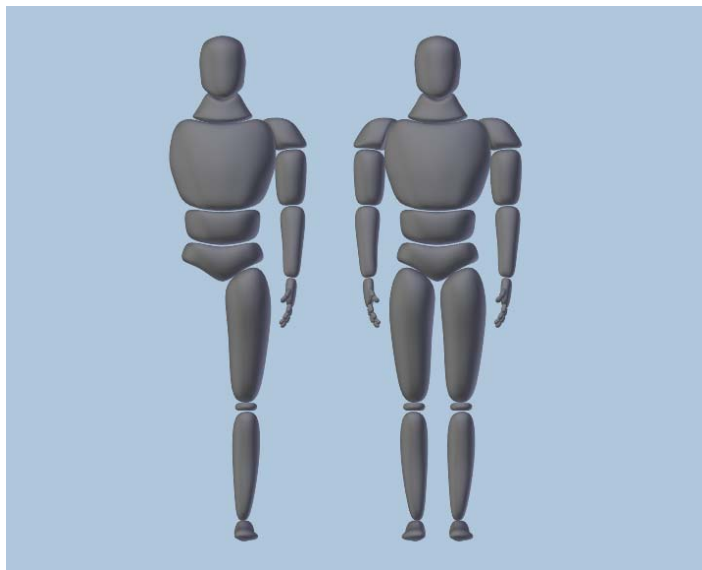


Obr. 2 - Low-poly model hlavy



Obr. 3 - Model hlavy s modifikátorem

Všechny zbývající nepárové části jsem tvořil stejným postupem s použitím většího počtu bodů. Části, které jsou v páru (bloky tvořící ruce a nohy), jsem modeloval pouze na jedné straně. Základem mi byla opět krychle. Po dokončení této jedné strany jsem vybral všechny bloky tvořící ruku a nohu a příkazem *Duplicate* je zduplikoval (s 3D kurzorem jako středem) a nakonec příkazem *Scale along global X* s hodnotou -1 vytvořil jejich zrcadlový obraz na druhé straně těla (Obr. 4).

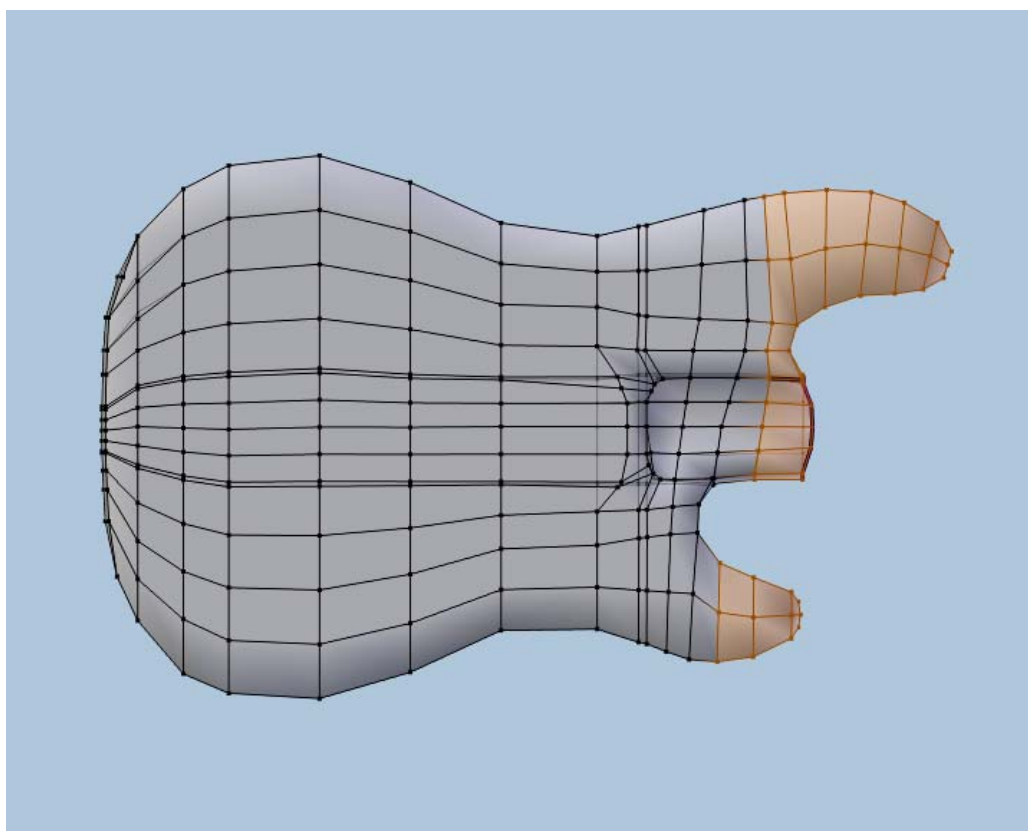


Obr. 4 - Model postavy před a po duplikaci párových částí

10.2 Baskytara

Baskytara je poměrně jednoduchým objektem na modelování, neboť se jedná v podstatě o tvarovanou desku. Na druhou stranu obsahuje více drobných prvků (šroubky, ladící mechanika apod.), které se mohou zdát nepodstatné, modelu však dodávají realistický vzhled. Jako předlohu jsem zvolil model baskytary Fender Vintage Modified Precision Bass (viz Příloha č. 3).

Při modelování těla baskytary jsem opět vycházel z krychle jakožto základního tvaru. Jelikož se však nejedná o symetrický tvar, nelze využít modifikátoru *Mirror*. Za použití příkazů *Subdivide*, *Loop cut* a *Grab* jsem co nejpřesněji kopíroval tvar předlohy. Při modelování jsem dbal na rovnoměrné rozložení bodů a udržení co nejpřímějších linií, čímž se snažím předejít nepřírozeným ohybům a zvlněním při následném použití modifikátoru *Subdivision surface*. Vyčnívající části baskytary jsem modeloval použitím příkazu *Extrude*, abych se vyhnul příliš ostrým úhlům mezi liniemi či jejich překrývání (Obr. 5).



Obr. 5 - Model těla baskytary (část modelovaná pomocí Extrude oranžově)

Pro modelování krku jsem zvolil jako základ půlválec, který bylo nutno mírně zploštit. Hlava baskytary a oblast spoje s tělem baskytary má obdélníkový průřez, kterého jsem snadno dosáhl vybráním příslušných bodů a použitím příkazu *Scale along global Y* s hodnotou 0. Hlavu baskytary jsem, podobně jako tělo, vytvaroval tak, aby co nejpřesněji kopírovala předlohu.

Tvar plastového krytu, nacházejícího se pod strunami, jsem modeloval z primitiva *Plane* a pomocí *Extrude* jej "vytáhl" do prostoru. Struny jsou jednoduchým válcem. Jejich zakroucení okolo ladící mechaniky jsem vytvořil příkazem *Spin*. Značky na hmatníku, pražce a šroubky jsou taktéž jednoduché válce, které jsem pouze kopíroval příkazem *Duplicate*. Při modelování ladící mechaniky a kobylky jsem použil jako základ krychli a pro dosažení požadovaného tvaru příkazy *Grab* a *Extrude*. Na všechny části jsem opět použil *Smooth* a modifikátor *Subdivision surface*.

10.3 Klavír

Klavír byl na modelování poměrně náročný, kvůli velkému počtu dílů. Jako předlohu jsem zvolil model od firmy Fazziolli, která na svých webových stránkách nabízí dostatek obrazových materiálů z různých pohledů a v dobré kvalitě. Pro modelování klavíru jsou podstatné čtyři pohledy, a to shora, zepředu, zezadu a z boku. Bohužel nabízené fotografie neobsahují kolmý pohled z boku, pouze pod úhlem, proto jsem použil zjednodušené zobrazení uvedené v dokumentaci tohoto modelu (viz Příloha č. 3).

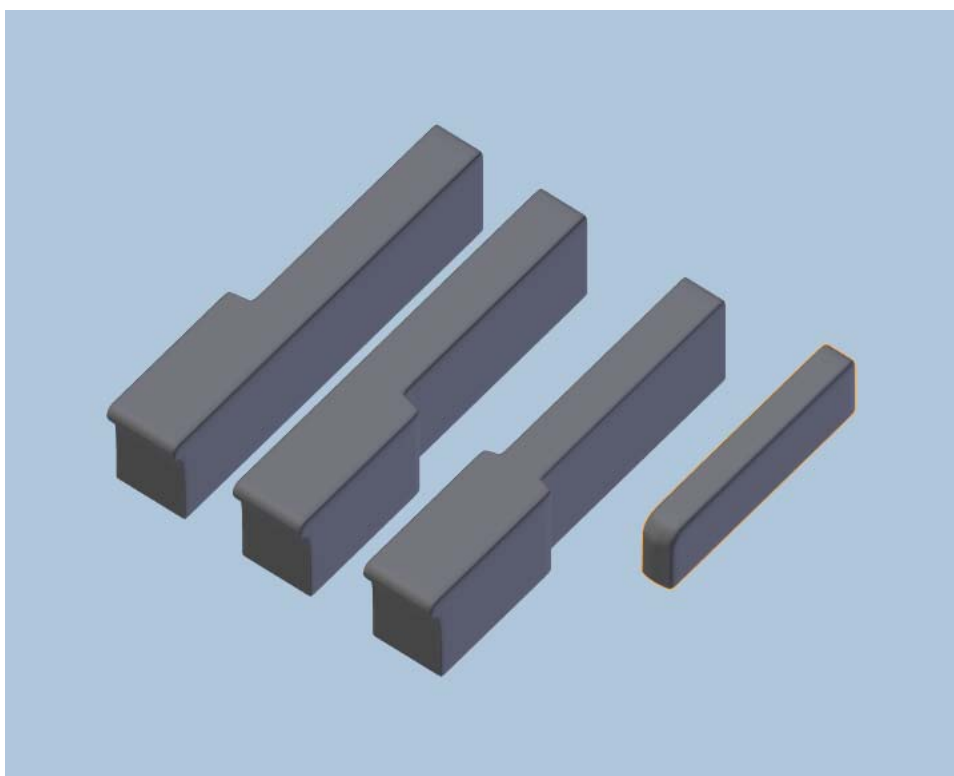
Při modelování jsem opět použil techniku box-modelingu. Základem je tedy krychle, ze které jsem pomocí nástrojů *Subdivide*, *Extrude*, *Grab* a *Scale* modeloval objekt spodní desky klavíru. Dále jsem vybral okrajové polygony a pomocí *Extrude* je o kousek vysunul směrem nahoru. Na obou stranách jsem odebral z výběru několik krajních polygonů a ostatní opět pomocí *Extrude* vysunul nahoru. Tentokrát již šlo o vysunutí boční stěny do konečné výšky podle předlohy. Následně jsem s použitím nástrojů *Loop-cut* a *Grab* modeloval krajní části bočních stěn. Vybráním odpovídajících polygonů a opět pomocí *Extrude*, *Grab* a *Rotate* jsem vymodeloval desku oddělující prostor klaviatury, lištu na přední straně a bloky po stranách klaviatury. V posledním kroku při modelování těla klavíru jsem vybral jeho dno a posunul je nahoru, přibližně do dvou třetin výšky.

Jako další jsem modeloval nohy klavíru, které jsou spojené s deskami pod přední a zadní částí klavíru. Desku jsem modeloval z krychle a použitím *Loop-cut* si připravil místa, ve kterých se připojují nohy. Nyní už stačilo tyto body vybrat a pomocí *Extrude*, *Scale* a *Grab* je vymodelovat do požadovaného tvaru podle předlohy z předního a bočního pohledu. Střední část sloužící k uchycení pedálů se velmi podobá dvěma nohám s tím rozdílem, že jsou na konci propojené. Podle předlohy jsem tedy podobným způsobem jako dříve modeloval tuto část. Jejich propojení jsem vytvořil vybráním vždy čtyř bodů, které mají být propojené a příkazem *Fill* je propojil a plochou vyplnil prostor mezi nimi.

Klaviatura je poskládána z množství kláves, které se opakují a mohou mít pouze jeden ze čtyř tvarů (Obr. 6). Černé klávesy jsou všechny stejné a jedná se o kvádry se zaoblenými hranami. Bílé klávesy jsou ve své podstatě kvádry s výřezem na jedné nebo obou stranách a vysunutou přední hranou. Na základním kvádru jsem si tedy pomocí

Loop-Cut připravil plochy pro následné vysunutí pomocí *Extrude*. Po nachystání všech čtyř typů kláves, jsem pomocí *Duplicate* a *Grab* vytvořil celou klaviaturu podle předlohy. U všech kláves jsem umístil pomocí *Set origin* počátky na jejich zadní hranu, což mi umožnilo snadnější manipulaci při animování. Posledním dílem je horní deska, která má stejný tvar jako spodní deska klavíru jen je kratší. Vytvořil jsem ji tedy jako kopii, umazal nepotřebné body široké přední části a vzniklé díry vyplnil pomocí *Fill*. Víko klaviatury má tvar ohnuté desky, takže jsem znovu vyšel z krychle a použitím *Extrude* a *Rotate* vytvaroval požadovaný ohyb.

Na všechny části jsem použil modifikátor *Subdivision surface*. Vzhledem k tomu, že jsou všechny hrany zaoblené, jejich zosťnění jsem dosáhl pomocí dvou technik. První je přidání řezu (řezů) v těsné blízkosti hrany. Druhou je vybrání příslušné hrany a nastavení hodnoty *Mean Crease*, která určuje váhu daných bodů při použití modifikátoru *Subdivision surface*.



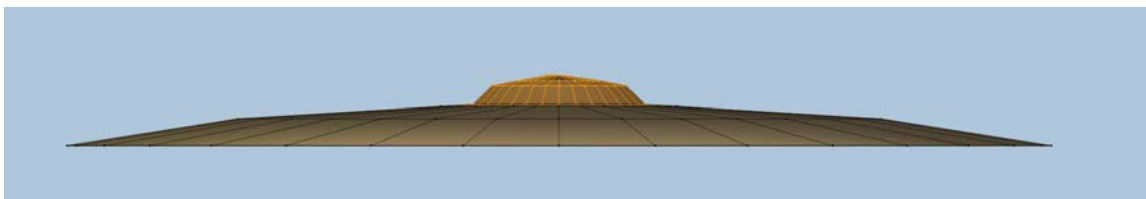
Obr. 6 - Čtyři možné tvary kláves

10.4 Bicí

Sada bicích je ze všech modelů v této animaci nejsložitější, a to nejen kvůli počtu dílů, ze kterých se skládá. Bicí jsem modeloval bez konkrétní předlohy, vzorem mi však byl model Pearl Vision VB. Pro správné pochopení následujícího textu je uveden popis jednotlivých dílů a součástí sady bicích (viz Příloha č. 4). Celá sada se skládá z pěti bubnů (velký buben, malý buben a 3 tomy), dvou činelů (crash a ride), hi-hat, celkem tří stojanů, stojanu na hi-hat s pedálem, pedálu pro velký buben a židle.

Základ bubnu je válec bez horní a dolní stěny, který jsem pomocí *Extrude* a *Scale* upravil pro vytvoření tloušťky korpusu. Ráfek jsem modeloval z primitiva *Circle*, a pomocí *Extrude* vytvořím prstenec. Ten jsem nástroji *Grab*, *Scale* a *Loop-cut* ohnul kolem hrany korpusu. Šest bodů po obvodu jsem lehce přizvedl, abych vytvořil plochy pro uchycení šroubů. Velký buben nemá kovový, ale dřevěný ráfek, který jsem vytvořil pomocí *Loop-cut* a *Scale*, což vytváří iluzi, že zde ráfek je, ale ve skutečnosti je to stále jeden objekt. Na korpusu je umístěno šest mušlí na každé straně. Krychli jsem nástroji *Subdivide* a *Grab* vytvaroval do zaobleného tvaru a pomocí funkce *Spin* vytvořil dalších pět kopií kolem korpusu. Šrouby spojující plošky na ráfku s mušlí jsem modeloval jako válec, který má na straně u ráfku vymodelovanou hlavičku hranatého tvaru a opět duplikoval pomocí *Spin*. Ráfek, mušle a šrouby jsem zkopíroval na druhou stranu korpusu pomocí *Duplicate* a *Mirror*. Posledními částmi jsou blány, které jsem modeloval jako *Circle* a mírně ohnul jejich okraje. U velkého bubnu je spojení šroubu s ráfkem tvořeno úchytem připomínajícím hák. Ten jsem modeloval z krychle pomocí *Extrude*. Každý buben má svůj objekt *Empty*, který je předkem všech jeho dílů, což usnadňuje následnou manipulaci. Ostatní bubny jsem vytvořil duplikací všech dílů a následnou změnou jejich velikosti

Činely mají tvar dvou kulových vrchlíků posazených na sobě (Obr. 7). Modeloval jsem tedy jeden činel z *Circle* pomocí *Grab* a *Loop-cut* a opět použitím *Duplicate* a *Scale* vytvořil ostatní činely jako kopie.

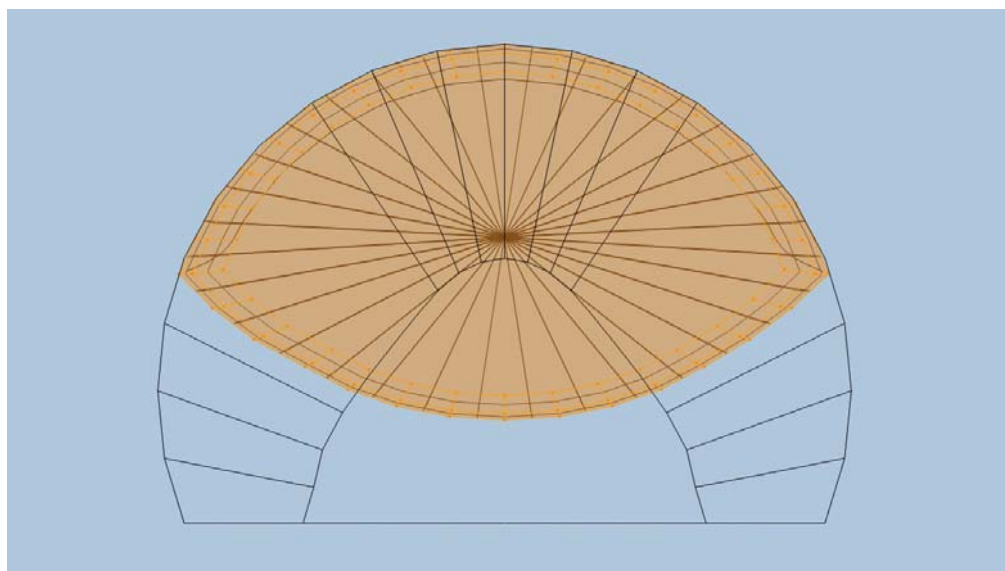


Obr. 7 - Činel (boční pohled)

Stojany na činely, židli a malý buben jsou v základu totožné, pouze mají jinou velikost. Stojany jsou složeny z tyče, tří noh, objímky s úchyty pro nohy, objímky s úchyty pro vzpěry, tří vzpěr, kloubu a trnu, na kterém je umístěn samotný činel. Tyč a trn jsem vytvořil jako válec a u trnu pomocí *Loop-cut* a *Scale* vytvořil podložku pro činel. Při modelování objímky jsem vytvořil pouze jednu třetinu a pomocí *Spin* a *Remove doubles* dotvořil výsledný tvar. Protože jsou nohy složeny ze dvou symetrických částí, modeloval jsem pomocí *Extrude* pouze jednu a druhou vytvořil nástrojem *Mirror*. Zbývající dvě nohy jsem vytvořil opět nástrojem *Spin*. Vzpěry jsem modeloval úpravou krychle a kloub jako kouli. Ramena ke stojanu pro malý buben jsem modeloval stejným způsobem jako nohy, pouze do jiného tvaru. Sedátko židle je pouze válec upravený nástrojem *Scale*.

10.5 Scéna

Jak bylo zmíněno výše, pódium jsem tvořil podle vlastního návrhu. Celá scéna se skládá pouze ze tří objektů. Zvolil jsem půdorys kruhové úseče (Obr. 8 - černě), tzn. že se jedná o válec otevřený z jedné strany, což vytváří prostor potřebný pro pohyb kamer a umístění světel. Druhým objektem je samotné pódium, u kterého jsem opět vycházel z válce. Pódium má spíše půdorys, jehož zadní strana kopíruje tvar stěny a přední strana je zakulacena s větším poloměrem (Obr. 8 - oranžově). Za pomoci nástrojů *Loop-cut*, *Grab* a *Scale* jsem modeloval lištu podél horní přední hrany. Posledním objektem dotvářejícím prostor scény je závěs podél zadní strany pódia, který jsem vytvářel vybráním příslušných bodů na objektu zdi a jejich duplikací. Nástroji *Scale* a *Grab* jsem přizpůsobil velikost a pozici a pomocí *Subdivide* znásobil počet bodů, které budou potřebné při následující aplikaci textur.



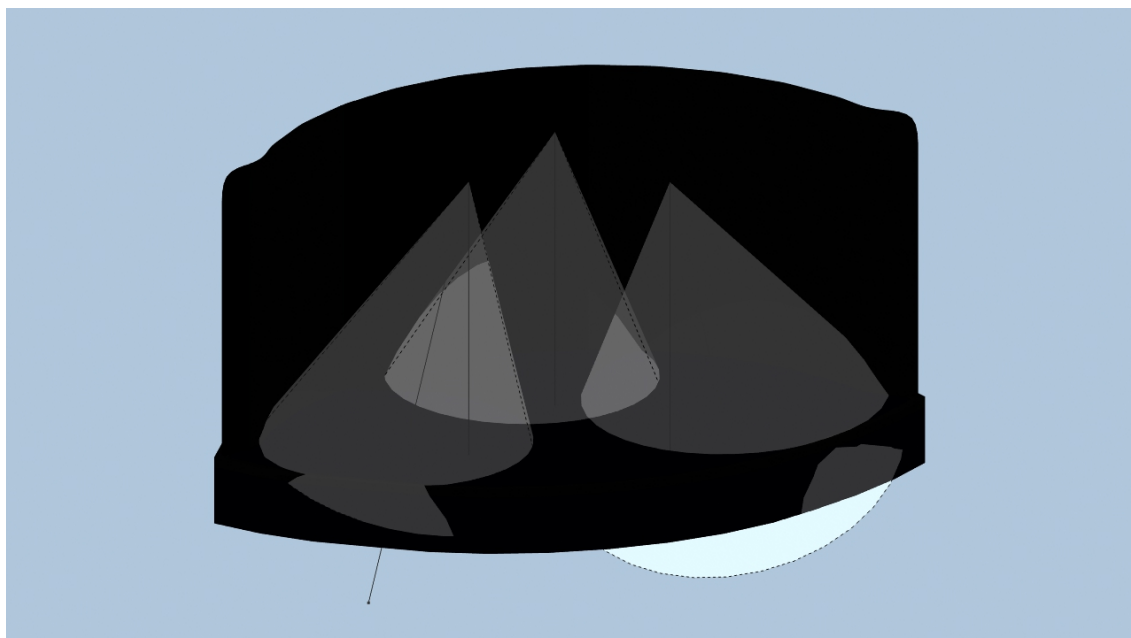
Obr. 8 - Půdorys scény a pódia

Všechny popisované objekty (postava, nástroje a scéna) jsem vytvářel jako samostatné projekty (soubory typu *blend*). Po dokončení všech modelů jsem je spojil do jednoho projektu pomocí *Append*. V této zkompleťované scéně jsem ještě modeloval paličky jako válce upravené pomocí *Loop-cut* a *Scale*. Druhým objektem přidaným až v této fázi byl popruh k baskytaře. Ten jsem modeloval z *Plane* pomocí *Extrude* do požadovaného tvaru kolem těla baskytaristy. Na tento pás jsem použil modifikátory *Sub-surf.* a *Solidify*, kterým jsem vytvořil tloušťku.

10.6 Světla

Při návrhu osvětlení scény jsem vycházel z potřeby mít v průběhu animace naprostou kontrolu nad osvětlením, které obsáhne funkce světla hlavního, výplňového i efektového, a aby zároveň bylo možno dosáhnout světelných efektů používaných v reálném světě. Základní koncept tedy obsahuje hlavní osvětlení pódia, bodové nasvícení nástrojů a barevná efektová světla ve čtyřech barvách. Celkem je na scéně 25 světel, z čehož některé jsou ovladatelné samostatně a ostatní jsou sdruženy do skupin po dvou nebo třech světlech, což redukuje počet ovladatelných prvků zhruba na polovinu.

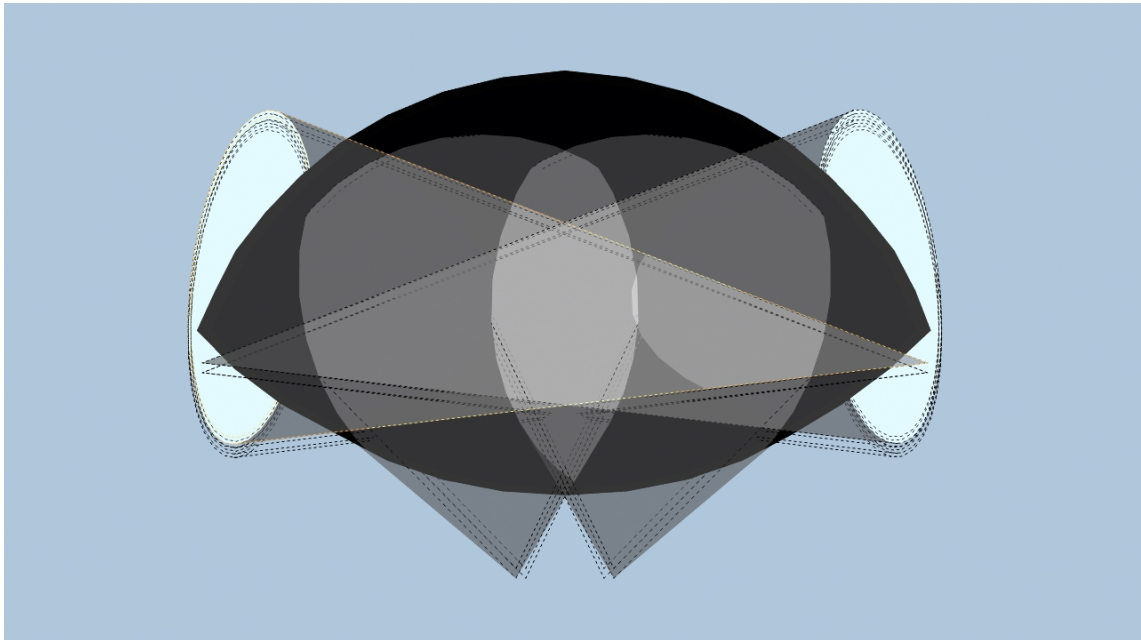
Do kategorie hlavního osvětlení spadají dvě skupiny světel. První skupinou jsou tři kuželová světla, ovladatelná samostatně a každé umístěné nad jedním nástrojem, směřující téměř kolmo k zemi. Jejich barva je bílá, vyzařovací úhel 35° , vrhají stín a způsobují odlesky (Obr. 9). Pro dosažení zajímavějšího vizuálního efektu mají nastavenou viditelnost (*Halo*) na 0,5 %. Druhou skupinou jsou tři kuželová světla umístěná nahoře před pódium. Jejich vlastnosti jsou stejné jako u předchozí skupiny, pouze místo ostrých okrajů mají nastaveno slábnutí směrem k okrajům (*Blend*) na 30% a jejich vyzařovací úhel je 55° .



Obr. 9 - Bodové osvětlení nástrojů

Funkci výplňových světel plní tři bílá světla umístěná nahoře na zadní stěně. Opět se jedná o bílá, kuželová světla, se slábnutím směrem k okrajům na 50 %, jejich vyzařovací úhel je 45° a nevrhají stín ani odlesky.

Skupinu efektních světel tvoří barevná světla umístěná na stranách pódia a nahoře před pódium (Obr. 10). Jedná se o kuželová světla s vyzařovacími úhly 60° (boční světla) a 90° (přední světla), která opět mají nastaveno slábnutí směrem k okrajům na 20 % a viditelnost na 0,2 %. Vzhledem k tomu, že do této skupiny patří celkem šestnáct světel, jsou sdruženy do skupin po dvou světelných podle jejich barvy a umístění. Celkem jsem použil čtyři barvy (žlutá, červená, zelená a modrá), což dává dohromady osm podskupin barevných světel.



Obr. 10 - Rozmístění barevných světel

11 Materiály a textury

Většina povrchů je tvořena jen pomocí základních povrchových vlastností, na některé jsou však aplikovány barevné mapy nebo obrazové mapy posunu a hrbolosti. V tabulce (Tab. 1) je uveden výpis použitých materiálů a jejich vlastností⁴ v následujícím formátu:

Barva - slovní pojmenování barvy / intenzita barvy v procentech

Odrazivost difuzní - slovní pojmenování barvy / intenzita v procentech

Odrazivost zrcadlová - barva / intenzita / Fresnelova difrakce (lom světla)

Průhlednost - hodnota alfa v procentech / Fresnelova difrakce

Tab. 1 - Seznam materiálů

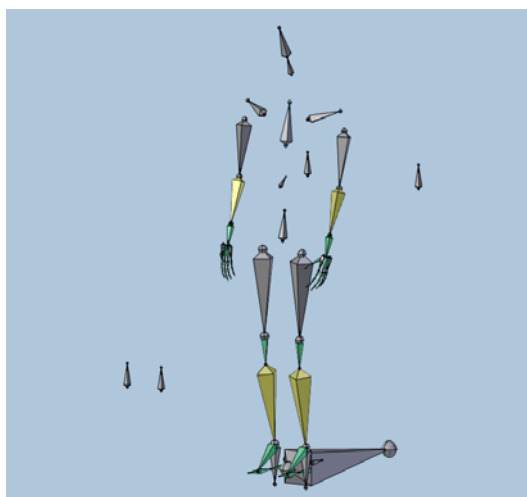
Název	Barva	Odrazivost difuzní	Odrazivost zrcadlová	Průhlednost
Blána černá	černá / 10%	bílá / 70%	šedá / 8% / F0,4	-
Blána čirá	šedá / 50%	bílá / 60%	šedá / 22% / F0	20% / F2
Blána čirá.1	bílá / 50%	bílá / 60%	šedá / 10% / F0,37	95% / F1,42
Dřevo světlé	béžová / 80%	béžová / 50%	-	-
Dřevo tmavé	hnědá / 19%	béžová / 50%	-	-
Kov	bílá / 86%	šedá / 50%	bílá / 100%	-
Kov - struny	šedá / 80%	bílá / 50%	bílá / 58% / F0,68	-
Klávesa b.	šedá / 20%	bílá / 50%	šedá / 32% / F1,27	-
Klávesa č.	černá / 80%	bílá / 50%	šedá / 32% / F1,27	-
Lak bílý	bílá / 50%	bílá / 64%	bílá / 11% / 0,43	-
Lak černý	černá / 80%	šedá / 10%	bílá / 8% / F1,52	-
Lak červený	červená / 15%	růžová / 50%	bílá / 25% / F1,18	-
Lak klavír	černá / 93%	bílá / 40%	šedá / 30% / F1,41	-
Lišta	černá / 11%	šedá / 10%	-	-
Mosaz	okrová / 60%	okrová / 50%	okrová / 32% / F0	-
Plast černý	černá / 30%	šedá / 72%	šedá / 3% / F0,4	-
Plast šedý	šedá / 51%	šedá / 46%	-	-
Podlaha	modrá / 50%	modrá / 5%	-	-
Textil	černá / 30%	bílá / 10%	-	-
Závěs	černá / 20%	-	-	-

Pro materiály světlého a tmavého dřeva a činely jsem aplikoval barevnou mapu *Wood*. Texturu *Wood* jsem dále použil ještě na struny jako mapu odlesků a na závěs jako mapu posunu. Na materiál textilu jsem aplikoval barevnou mapu *Clouds* a na závěs odleskovou mapu *Noise*. Na dno klavíru jsem metodou UV aplikoval barevnou mapu vycházející z referenčního obrázku.

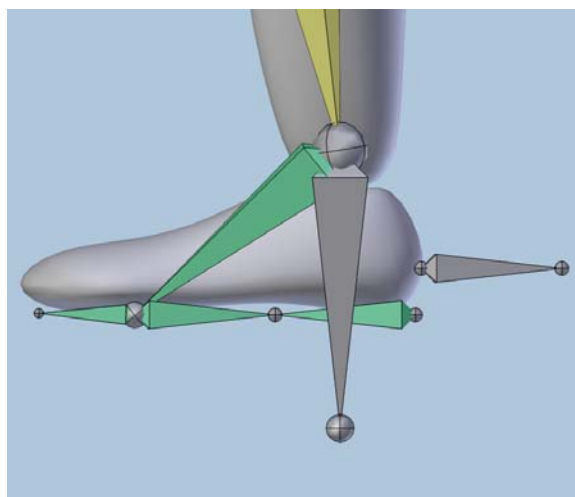
⁴ Některé materiály byly upraveny ve složené scéně vzhledem k celku. Hodnoty v tabulce se tedy mohou lišit od těch, které jsou použity v jejich samostatných projektech.

12 Příprava animačních akcí

Základem pro animování postavy je kostra. Vytvořil jsem kostru inspirovanou *DVD tutoriálem Lee Salviminiho* [12] a vybral si z něj pouze nezbytně nutné části. Jako základ jsem vytvořil *Meta-Rig* pomocí stejnojmenného addonu a umístil jednotlivé klouby na správné pozice tak, aby odpovídaly způsobu pohybu postavy. Dále jsem rozdělil stehenní kosti, tak abych získal kosti představující kolena a přiřadil všechny bloky tvořící tělo ke kostře funkcí *Set Parent To Armature - with automatic weights*. Nyní jsem ve *Weight Paint* módu ručně doladil přiřazení bodů k jednotlivým kostem tak, aby nedocházelo k nepřírozeným deformacím. Když jsem měl připraveny kosti zajišťující deformaci, bylo potřeba vytvořit řídicí kosti pro IK a několik pomocných kostí. IK efekty pro ruce jsou kosti umístěné na stejném místě jako kosti dlaní. IK efekty pro nohy jsou kosti vycházející z kotníků a směřující dolů. Pomocné kosti jsou tři pro každé chodidlo (Obr. 11) a umožňují jeho přirozený pohyb a deformaci. Dalšími pomocnými kostmi jsou tzv. *Pole target* umožňující směřování kolena a loktů při používání IK. Kosti IK, *Pole target* pro lokty a kost s názvem *hips* jsou vázány na kost *Root*, představující centrum celé kostry. *Pole target* pro kolena jsou vázány na kost *foot_IK*. Kostem potřebným k ovládní postavy jsem přiřadil alternativní křivky (*custom shapes*), které později při animování umožňují větší přehlednost, neboť veškeré ostatní kosti nemusí být zobrazeny.



Obr. 12 - Kostra postavy



Obr. 11 - Kostra chodidla

Nejjednodušší na animaci byl klavír, protože akce pro všechny klávesy obsahují stejný pohyb. Jedná se o rotaci kolem lokální osy X o 5° , krátké zastavení a rotaci zpět. Celý pohyb je dlouhý 9 framů (dále jen f).

Animační akce jednotlivých bubnů jsou změnou velikosti o 1 % ve všech směrech a trváním 5 f. Cílem této změny je pouze zdůraznění úderu. Akce pro činely spočívají v rotaci o $1-8^\circ$ kolem horizontálních os a délkou 4 f pro hi-hat, 6 f pro crash a 7 f pro ride. U pedálu hi-hat je akce dlouhá 4f a jedná se o rotaci o 8° a zpět, u pedálu velkého bubnu jde o rotaci hřídelky o 45° krátké zastavení a pohyb zpět, kde celý pohyb trvá 7 f.

Pohyby postavy bubeníka jsou nejsložitějšími v celé scéně, neboť se jedná o komplexní pohyby a přirozenost lidského pohybu je vždy složitá. Po několika pokusech jsem našel správné časování úderu s délkou 6 f. Ten se skládá z pohybu dolů v trvání 2 f a pohybu zpět s trváním 4 f. Pro obě ruce je pohyb stejný a u jednotlivých bubnů se liší pouze polohou ruky a natočením hrudníku a ramen. Úder na crash je 6 f dlouhý, začíná nad činelem a končí v místě nad malým bubnem. V tomto případě tedy neobsahuje pohyb zpět. Údery na ride a hi-hat trvají 5 f a jsou podobné úderům na bubny s tím rozdílem, že mají menší amplitudu pohybu a jsou tedy jemnější. Pohyb pravé nohy při každém úderu na velký buben trvá 7 f a mírně se liší od pohybu rukou. Začátek pohybu je nad pedálem, trvá 2 f, než se dostane do nejnižšího bodu, v něm setrvá 2 f a poté se vrací nahoru zároveň s pedálem. Poloha chodidla na konci je však níže než na začátku. Pohyb levé nohy trvá celkem 6 f. Během prvních 2 f se chodidlo zvedne a dojde k otočení v kotníku, další 2 f je pohyb dolů, kdy dochází k sešlápnutí pedálu špičkou, a během posledních 2 f se chodidlo otočením v kotníku vrací zpět do původní polohy. Paličky se u všech úderů pohybují zároveň s dlaní a směřují šikmo vzhůru. V momentě úderu se mění jejich rotace, tzn. směřují vodorovně nebo mírně dolů, a při pohybu zpět se navrací do původního otočení. Doplnujícím pohybem postavy bubeníka je pokyvování hlavou o délce 13 f.

Pohyb u baskytary je pouhé kmitání strun. Pro dosažení tohoto pohybu jsem vytvořil jednoduchou kostru, která umožňuje pohybovat strunami (Obr. 13). Samotná akce pohybu struny je dlouhá 6 f a skládá se z pohybu nahoru, dolů, nahoru a zpět do výchozí polohy.



Obr. 13 - Kostra strun baskytary (oranžově)

Postava baskytaristy obsahuje dvě hlavní oblasti akcí (levá a pravá ruka) a jednu doplňkovou (levá noha). Akce pravé ruky (brnknutí) je složena z pohybu dolů (pod strunu) a z pohybu zpět do výchozí pozice, při němž se nepatrně ohnou prsty. Celá akce trvá 6 f a u každé struny je akce stejná, jen se liší ve výchozí pozici. Pohyb levé ruky je dlouhý 7 f a skládá se z přimáčknutí struny, podržení a puštění. Podobně jako u pravé ruky se pro každou strunu a každou notu liší pouze výchozí pozice. Doplňkovou akcí je klepání nohou dlouhé 8 f, vytvořené rotací v kotníku, což ve výsledku vytváří pohyb nahoru a dolů.

13 Návrh skriptu

Při hledání možností jak synchronizovat animaci pomocí midi souboru jsem objevil skript *Midi_driver*⁵, který toto umožňuje. Ten je však určen pro Blender 2.49, čímž se stává nepoužitelným ve verzi 2.64, v níž došlo k přechodu na Python 3 a změně API. Řešením tedy bylo naprogramovat vlastní skript určený pro aktuální verzi. Princip je přečtení midi souboru a následná synchronizace akcí v NLA editoru podle zadaných kritérií. Celý skript sestává ze dvou částí, kde každá je umístěna v samostatném souboru. V následujícím textu stručně popisuji objektovou strukturu a funkce obou částí.

13.1 Soubor *midi.py*

Soubor *midi.py* má funkci externího modulu a obsahuje část kódu sloužící ke zpracování samotného midi souboru. Výše zmíněný *Midi_driver* spadá pod licenci GNU-GPL a jeho část řešící čtení a zpracování midi funguje ve své podstatě jako samostatně fungující modul, byť je součástí jednoho souboru. Po prostudování tohoto řešení jsem dospěl k závěru, že lze tuto část s několika úpravami velmi dobře použít v mém řešení.

Třídy:

- **Note**
 - Defínuje objekty s atributy nesoucími informace: výška tónu, nástroj, kanál, čas začátku a konce, počáteční a koncovou hlasitost, a zda je objekt vybrán.
 - Tato třída obsahuje pouze konstruktor a komparátor.
- **MidiFile**
 - Tato třída obsahuje konstruktor zajišťující načtení vstupního midi souboru do paměti a následně volá metody *readHeaderChunk*, *readTrackChunk* a *calcRealTime*.

⁵ *Midi_driver* vytvořil Paweł Adamowicz (dostupný z <http://blendit.xaa.pl/?p=middrv&l=eng>)

- *readHeaderChunk* přečte hlavičku souboru, zjistí, zda se jedná o midi a jeho správný formát a uloží informace o počtu stop (tracků) a časování (bit-time).
- *readTrackChunk* zajišťuje rozpoznání jednotlivých midi událostí a v případě rozpoznání noty nebo změny nástroje volá metodu *handleMidiEvent*, která ji dále zpracuje a přidá do seznamu *noteList*. Pokud byla rozpoznána událost *note-off*, je nalezení příslušné noty, ke které náleží, v seznamu *noteList* zpracováno zavoláním metody *finishNote*.
- *calcRealTime* je voláno po až přečtení celého souboru a naplnění *noteList*-u, ve kterém převádí všechny časové hodnoty z hodnoty *bit-time* na milisekundy.
- *strToNumber* a *vStrToNumber* převádějí předaný řetězec znaků pevné nebo variabilní délky na jeho číselnou hodnotu.
- *selectNotes* mění atribut *isSelected* u not v seznamu, které odpovídají zadaným kritériím
- *extractSelectedNotes* vrací nový seznam not, který již obsahuje jen objekty s příznakem *isSelected*
- **MidiFormatError**
 - Zachycuje tyto tři chyby čtení souboru:
 - nejedná se o midi soubor
 - nepodporovaný formát midi
 - nepodporovaný časový formát

13.2 Soubor `__init__.py`

Soubor `__init__.py` slouží k inicializaci celého skriptu a umožňuje jeho implementaci jako addon-u. Tato část skriptu zajišťuje vytvoření uživatelského rozhraní (Obr. 14), výběr souboru a jeho předání k dalšímu zpracování a samotnou synchronizaci animačních akcí. Slovník v úvodu tohoto souboru obsahuje základní informace o skriptu.



Obr. 14 - GUI skriptu

Třídy:

- **Interface**
 - V úvodu je určeno v kterém okně a na které kartě se interface vykreslí.
 - `draw` je metoda sloužící k vykreslení definovaných objektů (textová pole, tlačítka, seznamy atd.)
- **MidiFileSelector**
 - Jedná se o operátor, spouštěný stisknutím tlačítka, který spouští "průzkumník souborů" a vybraný soubor předává k dalšímu zpracování.
- **Synchronize**
 - Jako první volá třídu `MidiFile` z modulu `midi.py` a předá jí cestu vybraného souboru jako parametr.

- Dále volá metodu `selectNotes`, které předává parametry definované uživatelem a zavoláním metody `extractSelectedNotes` získá seznam pouze vybraných not, se kterým může dále pracovat.
- Pro aktivní objekt vytvoří nový *Track* v NLA editoru s názvem `Midi_Track`. Pokud nastane případ, že se následující akce překrývá časově s tou předcházející, vytvoří se `Midi_Track2`. V případě, že jsou v danou chvíli více než dvě překrývající se akce, zpracují se pouze dvě a další se přeskočí. Tento stav může nastat při nevhodně zvolených vstupních parametrech.
- Poté prochází seznamem, přepočítává čas z milisekund na framy a kopíruje původní akci do vytvořeného *Tracku* na správná místa.
- Mimo tyto třídy je na konci blok kódu upřesňující vlastnosti jednotlivých objektů rozhraní a metody umožňující registraci a odregistraci těchto tříd jako součástí aplikace.

14 Hudební podklad a synchronizace

Hudební podklad je mým vlastním dílem, určeným konkrétně pro tuto animaci. K tvorbě podkladu jsem použil program FL Studio 9, který slouží ke skládání hudby na PC. Při zápisu not používá systém podobný midi a umožňuje využívat banky midi zvuků nebo je nahradit zvuky reálnými či zvukovými syntezátory. Tento software také nabízí snadný export do formátu midi.

Zvukový podklad k ukázkové animaci obsahuje celkem 5 stop. Po exportu do midi je jejich struktura následující:

- **Track 1 - prázdný**
- **Track 2 - Bicí**
 - midi channel 10, instrument - Standard drum set (128)
 - obsahuje většinu not hraných na bicí
- **Track 3 - Bicí**
 - midi channel 10, instrument - Standard drum set (128)
 - obsahuje pouze noty na bubny, které by měly být hrané pravou rukou
- **Track 4 - Baskytara**
 - midi channel 2, instrument - Acoustic bass - finger (33)
- **Track 5 - Klavír**
 - midi channel 1, instrument - Acoustic grand piano (0)
- **Track 6 - Doplnkové akce**
 - midi channel - 3, instrument - FX 2 - soundtrack (97)
 - je použit pouze pro časové určení pomocných akcí a v hudebním podkladu není slyšet

Samotná synchronizace jednotlivých animačních akcí pomocí skriptu je velmi jednoduchá. V prvním kroku jsem vybranému objektu přiřadil akci, která se má synchronizovat. V druhém kroku jsem v GUI skriptu zadal cestu k midi souboru, definoval všechny potřebné parametry, podle kterých se má akce synchronizovat a tlačítkem *Sync* spustil proces synchronizace.

15 Renderování

Proces renderingu je přímo závislý na výpočetním výkonu počítače a s nižším výkonem rapidně narůstá čas potřebný pro výpočet. Vzhledem k omezením daným PC, na kterém jsem vytvářel ukázkovou animaci (Intel Core2 duo (64 bit, 2 GHz) a 4 GB RAM), jsem byl donucen zvolit přiměřený kompromis mezi kvalitou a časem potřebným pro rendering.

Výsledné video jsem renderoval v jedné vrstvě v šesti průchodech (*Z*, *Diffuse*, *Specular*, *Shadow*, *Reflection* a *Refraction*). Výstupní rozlišení jsem zvolil HDTV 720p (1280x720, 24 fps) s kódováním AVI Raw (nekomprimované AVI). Celá animace se zpracovávala technikou trasování paprsků s vypnutým výpočtem odražených paprsků. Zahrnutí odražených paprsků světla by sice animaci přidalo na realistickém vzhledu, ale úměrně s počtem odrazů narůstá i čas potřebný pro výpočet každého snímku (až několikanásobně). Zároveň jsem použil vyhlazování hran s jedenáctinásobným oversamplingem a filtrem Mitchell-Netravali. Jiné filtry hrany příliš rozmazávají nebo naopak zvýrazňují, což by v tomto případě působilo nevhodně. Poslední fází celého procesu renderingu je post processingové zpracování obrazu. Kvůli náročnosti na výpočet jsem jej omezil pouze na dithering s hodnotou 1, který přimícháním malého množství šumu do obrazu omezuje slévání barev.

Proces renderování využívá procesor na 100 % a tím činí PC v danou dobu nepoužitelným k jakékoliv jiné činnosti. Z tohoto důvodu jsem animaci renderoval po částech dlouhých 100, 200 nebo 300 framů, což mimo to umožnilo i včasné zaznamenání nedostatků a jejich snadnější nápravu. Vyrenderované části jsem zkompletoval dohromady a přidal k nim zvukovou stopu a titulky v trial verzi stříhového programu Adobe Premiere CS4. Z důvodu kompatibility jsem finální video exportoval ve formátu HDTV s kódováním MPEG2 (viz Příloha č. 1).

Závěr

Hlavním cílem bakalářské práce bylo naprogramovat skript, umožňující synchronizaci animace na základě midi souboru, a vytvořit ukázkové animace jeho aplikace. Jelikož se jedná o první verzi skriptu, pokrývá pouze oblast pohybových akcí objektů. V dalších verzích by bylo možno rozšířit jeho funkčnost propojením s vlastnostmi materiálů, světel či přímými zásahy do IPO křivek. Již v této verzi je však skript užitečným nástrojem při tvorbě animací. Ukázková animace předvádí atraktivní formou jeho využití a zároveň prezentuje možnosti Blenderu jako softwaru dostupného běžnému uživateli PC.

Dílčím cílem bylo poskytnout náhled do problematiky tvorby 3D animací. Teoretická část této práce poskytuje stručný souhrn poznatků a technik z oblasti 3D grafiky a animace. Velmi dobře tedy může sloužit jako doplňující materiál při výuce předmětů, týkajících se této oblasti tvorby. Celkově byla tato práce technicky a časově velmi náročná, splňuje však vše, co si klade za cíl. Stejně dobře ji lze využít jako ukázkou vlastní tvorby 3D animace.

Seznam zdrojů a literatury

Monografie:

- [1] KERLOW, Isaac Victor. *Mistrovství 3D animace: ovládněte techniky profesionálních filmových tvůrců*. Vyd. 1. Brno: Computer Press, 2011, 496 s. ISBN 978-80-251-2717-9.
- [2] RATNER, Peter. *Mastering 3D animation*. 2nd ed. New York, NY: Allworth Press, 2004, xiii, 354 p. ISBN 15-811-5345-7.
- [3] SUMMERFIELD, Mark. *Programming in Python 3: a complete introduction to the Python language*. 2nd ed., Fully rev. ed. Upper Saddle River, NJ: Addison-Wesley, c2010, xvi, 630 s. Developer's library. ISBN 03-216-8056-1.
- [4] VAN GUMSTER, Jason. *Blender for dummies*. Indianapolis: Wiley Publishing, c2009, xii, 388 s. ISBN 978-0-470-40018-0.
- [5] HESS, Roland. *The essential Blender: guide to 3D creation with the open source suite Blender* [online]. San Francisco, CA: No Starch Press, 2007, 371 s. ISBN 978-1-59327-166-4. Dostupné z: http://wiki.blender.org/index.php/Doc:2.4/Books/Essential_Blender
- [6] POKORNÝ, Pavel. *Blender: naučte se 3D grafiku*. 1. vyd. Praha: BEN, 2006, 247 s. ISBN 80-730-0203-5.
- [7] PILGRIM, Mark. *Ponořme se do Python(u) 3: Dive into Python 3*. 2nd ed., Fully rev. ed. Praha: Cz.Nic, c2010, 430 s. CZ.NIC. ISBN 978-80-904248-2-1.

Ostatní zdroje:

- [8] Doc:CZ/2.6/Manual. In: *BlenderWiki* [online]. Dostupné z: <http://wiki.blender.org/index.php/Doc:CZ/2.6/Manual>
- [9] ŠVEC, Jan. Létající cirkus. In: *Root.cz* [online]. 2001. Dostupné z: <http://www.root.cz/clanky/letajici-cirkus/>
- [10] An Introduction to MIDI. MIDI MANUFACTURERS ASSOCIATION. [online]. 2009. Dostupné z: <http://www.midi.org/aboutmidi/intromidi.pdf>
- [11] MIDI File Format. In: *The Sonic Spot* [online]. 2007. Dostupné z: <http://www.sonicspot.com/guide/midifiles.html>
- [12] SALVEMINI, Lee. CG MASTERS. Ninja character creation: Volume 2 - Rigging [DVD]. 2011.
- [13] BLENDER. *Blender Documentation contents: Blender v2.59.0 - API documentation* [online]. 2011. Dostupné z: http://www.blender.org/documentation/blender_python_api_2_59_0/contents.html

Seznam příloh

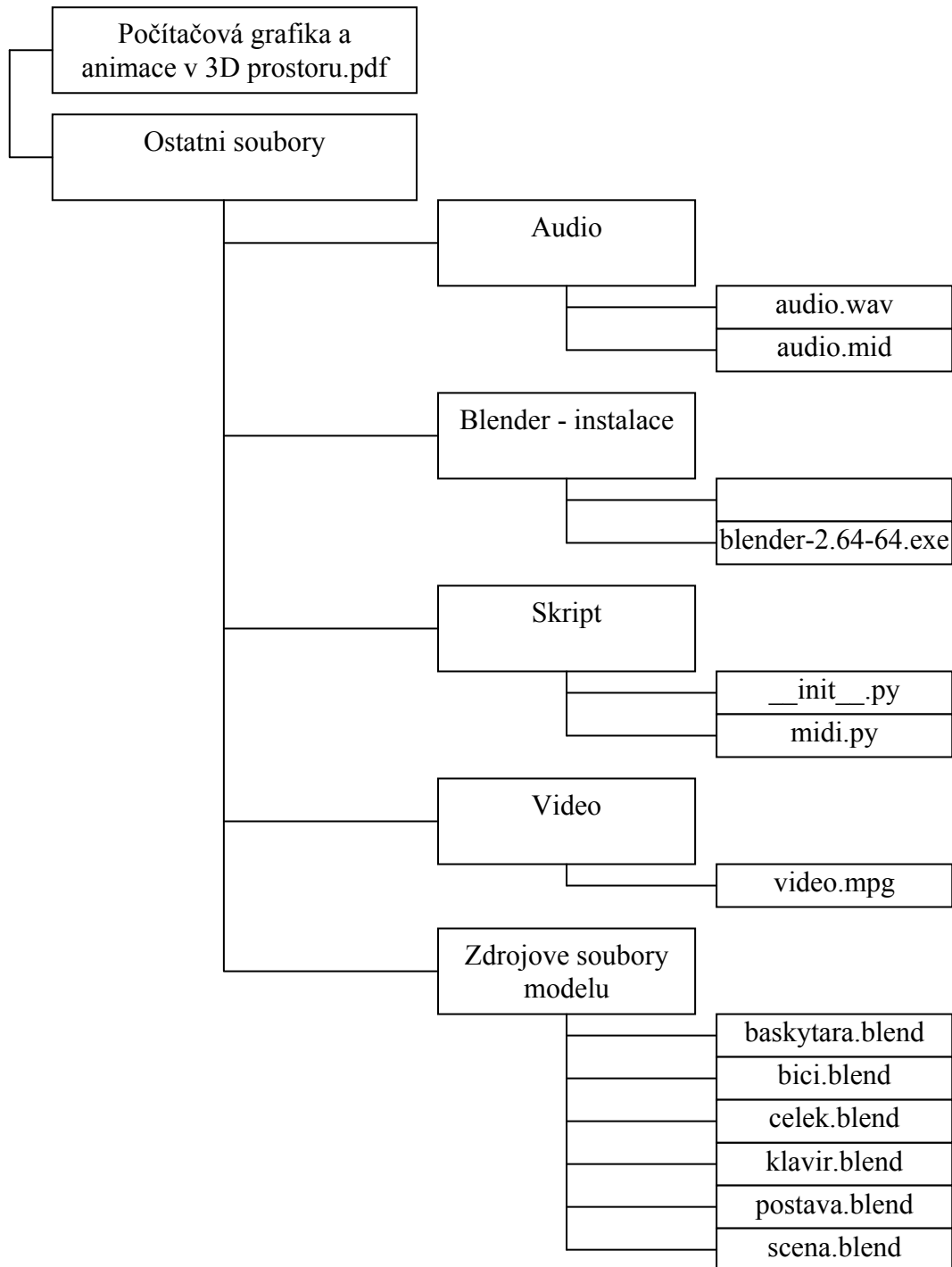
Příloha č. 1 - Struktura CD

Příloha č. 2 - Slovníček pojmů

Příloha č. 3 - Referenční obrázky

Příloha č. 4 - Bicí (popis)

Příloha č. 1 - Struktura CD



Příloha č. 2 - Slovníček pojmů

Python - interpretovaný programovací jazyk

MIDI - digitální rozhraní hudebních nástrojů

3D - three dimensional = trojrozměrný

Třída - základní konstrukční prvek objektově orientovaného programování (předpis pro objekty)

Metoda - funce pracující s daty tříd a objektů

GUI - graphic user interface = grafické uživatelské rozhraní

Post processing - postupy používané pro vylepšení kvality videa a obrázků (následné zpracování)

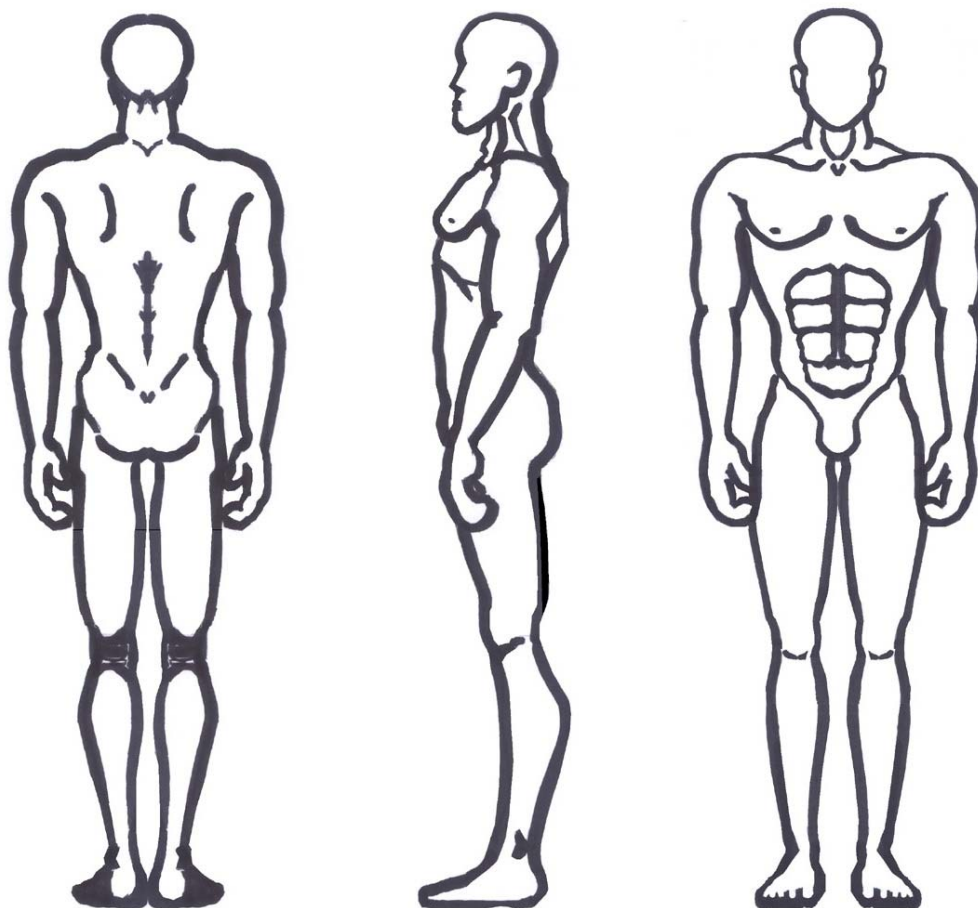
GNU-GPL - všeobecná veřejná licence pro svobodný software

Addon - software rozšiřující a doplňující konkrétní aplikaci

Multiplatformní - software spustitelný pod více než jednou platformou (operačním systémem)

Oversampling - technika zdokonalování obrazového nebo zvukového záznamu v digitální podobě

Příloha č. 3 - Referenční obrázky



Obr. 15 - Skica postavy (převzato z http://fc00.deviantart.net/fs31/i/2008/198/0/6/Male_Anatomy___Broad_Shoulders_by_sunandshadow.jpg)



Obr. 16 - Baskytara Fender (převzato z <http://assets.fender.com/fri/39005a2c4c3b361cd7cdaca5241f0ca4/generated/86ceb3c861ac3d7b0737b2ccde488ca3.png>)



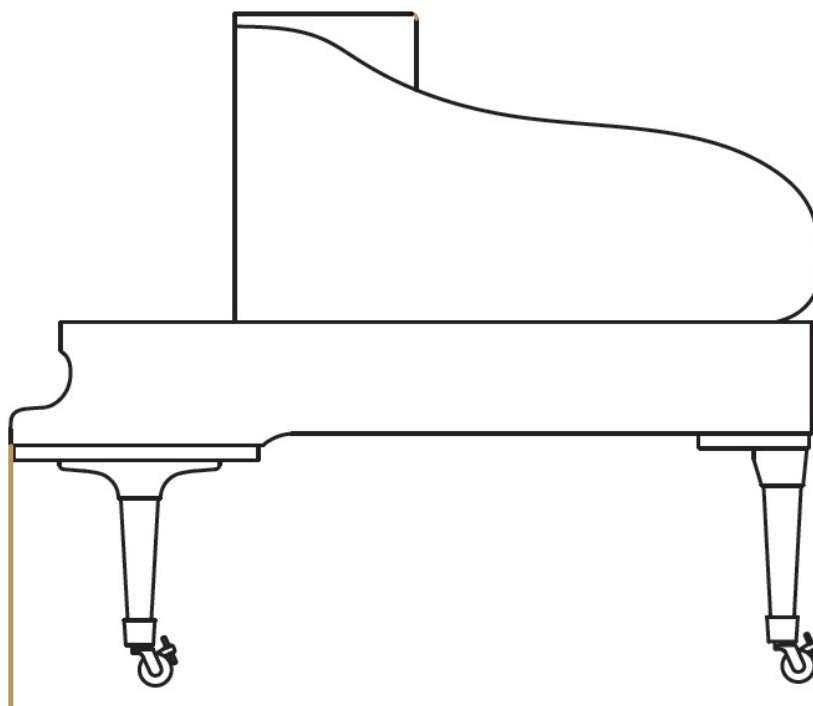
Obr. 18 - Klavír Fazioli - zhora (převzato z <http://www.fazioli.com/sites/default/files/f228-06.jpg>)



Obr. 17 - Klavír Fzioli - zepředu (převzato z <http://www.fazioli.com/sites/default/files/f228-03.jpg>)

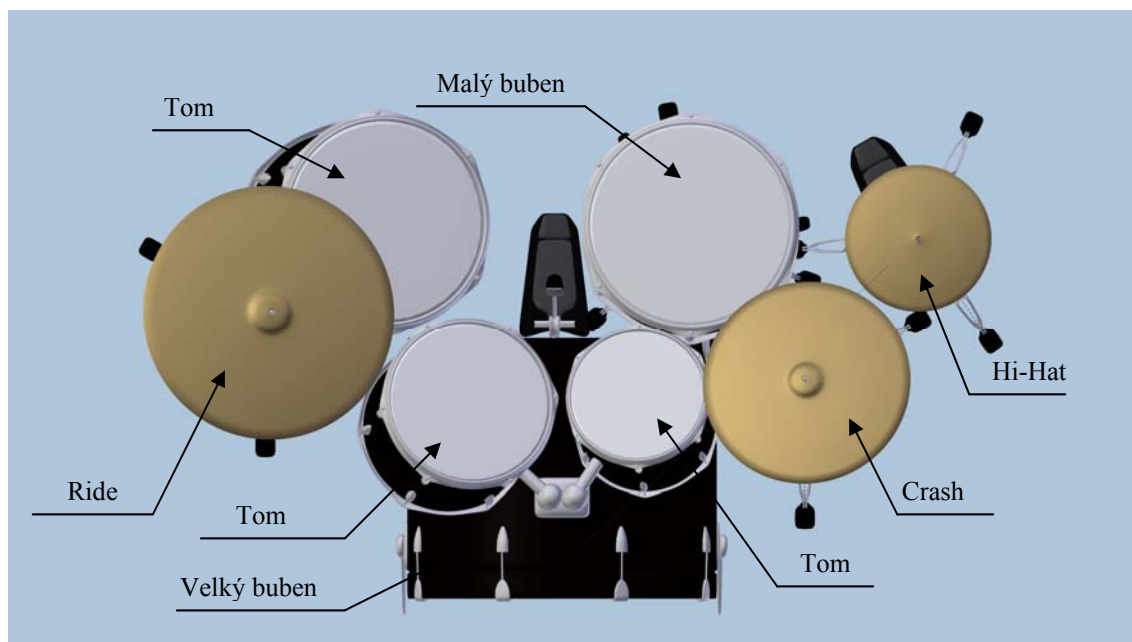


Obr. 19 - Klavír Fazioli - zezadu (převzato z <http://www.fazioli.com/sites/default/files/f228-02.jpg>)

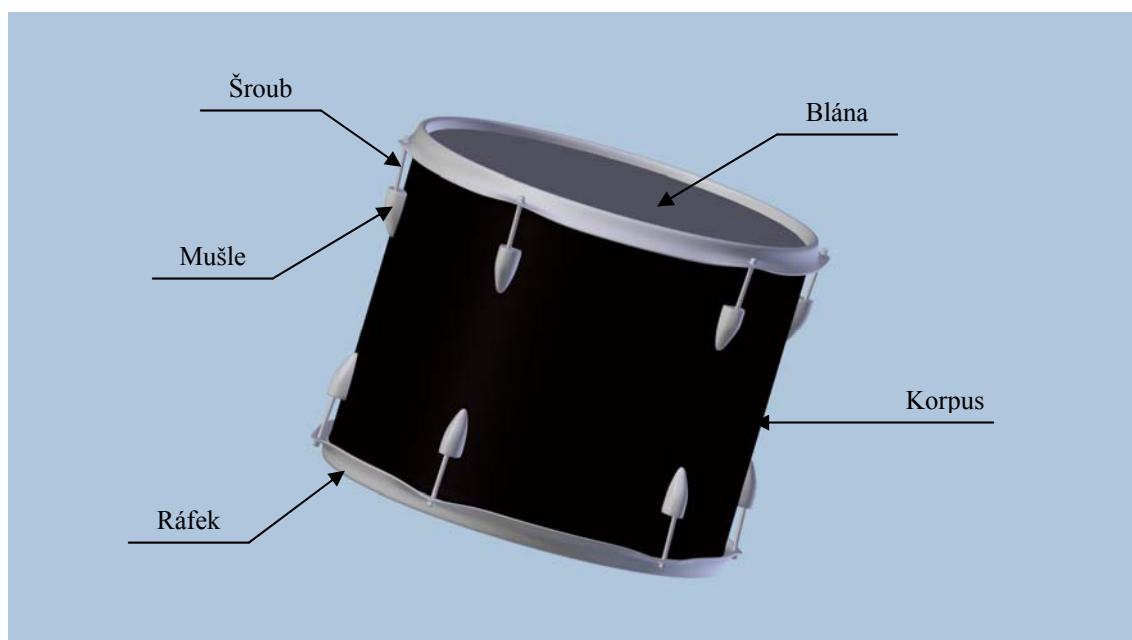


Obr. 20 Klavír Fazioli - z boku (převzato z http://www.fazioli.com/sites/default/files/schede-prodotto/fazioli_F228_ITA_0.pdf)

Příloha č. 4 - Bicí (popis)



Obr. 21 - Popis bicí sady



Obr. 22 - Popis částí bubnu

Anotace

Jméno a příjmení:	Přemysl Geryk
Katedra:	Katedra technické a informační výchovy
Vedoucí práce:	Mgr. Jan Kubrický
Rok obhajoby:	2013

Název práce:	Počítačová grafika a animace v 3D prostoru
Název v angličtině:	Computer graphics and animation in 3D space
Anotace práce:	Tato práce se zaměřuje na problematiku vytváření 3D animací a jejich synchronizaci na základě MIDI souboru. První část práce poskytuje teoretický základ potřebný pro tvorbu 3D animací. Druhá část stručně popisuje celý proces tvorby ukázkové animace od návrhu, přes vytváření modelů, programování skriptu až po výsledné video.
Klíčová slova:	počítačová grafika, 3D animace, 3D modelování, programování, MIDI
Anotace v angličtině:	This work focuses on the creation of 3D animations and their synchronization based on the MIDI file. The first part provides the theoretical background needed for creating 3D animations. The second section briefly describes the entire process of creating a sample animation from design, through modeling, script programming to the final video.
Klíčová slova v angličtině:	computer graphics, 3D animation, 3D modeling, programming, MIDI
Přílohy vázané v práci:	Příloha č. 1 - Struktura CD Příloha č. 2 - Slovníček pojmů Příloha č. 3 - Referenční obrázky Příloha č. 4 - Bicí (popis)
Rozsah práce:	50 s.
Jazyk práce:	český