

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÁ TVORBA 3D MODELŮ V PROGRAMU BLENDER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. VIKTOR JABLONSKÝ

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÁ TVORBA 3D MODELŮ V PROGRAMU BLENDER

AUTOMATIC CREATION OF 3D MODELS IN APPLICATION BLENDER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. VIKTOR JABLONSKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVEL ŽÁK

BRNO 2010

Abstrakt

Tento text vznikl jako součást diplomové práce a popisuje proces vytváření automatického generátoru 3D objektů pro program Blender. Součástí je stručné seznámení s aplikací Blender, dále jsou v dokumentu prezentovány některé existující skripty pro tvorbu 3D modelů, které byly v rámci teoretického studia problematiky otestovány. Po rozboru požadavků na automatický generátor jsou jako vhodná třída vytvářených objektů vybrány různé druhy mečů. V poslední části teoretického úvodu jsou vysvětleny základní pojmy z tohoto oboru, konkrétně názvosloví mečů a příbuzných zbraní. Jsou vyjmenovány nejdůležitější vlastnosti a jejich význam. Tato kapitola zároveň poskytuje zjednodušený typologický průřez skupinou těchto zbraní. Následující kapitoly popisují postup vytváření modelu, hlavní principy a matematickou podstatu použitých algoritmů a metod, návrh a realizaci GUI a způsob testování a optimalizace. Průběžně s výkladem jsou také navrhovány další možnosti rozšíření. Závěr shrnuje dosažené výsledky a hodnotí celkový přínos práce.

Abstract

This text was written as part of a master's thesis. It describes development of an automatic generator of 3D models for Blender. After a brief introduction to the Blender application some of existing scripts for 3D model creation (tested as part of theoretical research) are presented. After an analysis of features required from an automatic generator various types of swords were chosen as the suitable class of objects for the automatic creation. Next part of this thesis presents basic terms in this field, namely the nomenclature of the swords and other related weapons. Their most important properties are listed and explained. The chapter also provides a simplified abstract of the sword typology. Following chapters describe main ideas of creating the model, explain principles and mathematics behind used algorithms and procedures and provide information about realization of the GUI as well as testing and optimisation. As the text explains these issues, suggestions about possible future enhancements are made. Document is concluded with an evaluation of all achievements and argues impact of this work on its field.

Klíčová slova

3D Blender automatické generování model skript Python meč

Keywords

3D Blender automatic creating model script Python sword

Citace

Viktor Jablonský: Automatická tvorba 3D modelů v programu Blender, diplomová práce, Brno, FIT VUT v Brně, 2010

Automatická tvorba 3D modelů v programu Blender

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením pana ing. Pavla Žáka

.....
Viktor Jablonský
24. května 2010

Poděkování

Děkuji vedoucímu své diplomové práce, panu ing. Pavlu Žákovi, za jeho vedení, připomínky a cenné rady. Na úspěšnosti této práce má velkou zásluhu. Dále bych chtěl poděkovat své rodině a přátelům za jejich podporu. V neposlední řadě patří dík také vývojářům Blenderu, kteří odvádějí skvělou práci, a komunitě uživatelů Blenderu, která je vždy ochotná a připravená podat pomocnou ruku.

© Viktor Jablonský, 2010.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Blender a Python	5
2.1 Ukázky (možnosti Blenderu)	5
2.2 Klíčové vlastnosti	5
2.3 Existující skripty	6
2.3.1 Bolt Factory	7
2.3.2 Auto Masonry	7
2.3.3 L-System fixed seed treemaker	7
2.3.4 L-System Buildings	8
2.3.5 Make Human	8
3 Výběr třídy objektů	10
3.1 Výhody plynoucí z automatického generování	10
3.2 Shrnutí požadavků	10
3.3 Vybraná třída objektů	11
4 O meči	12
4.1 Motivační úvod	12
4.2 Anatomie mečů	12
4.2.1 Základní prvky meče	13
4.2.2 Vývoj mečů	14
4.2.3 Netradiční prvky	15
4.3 Dekompozice jako pomůcka při generování	16
5 Dekompozice meče z hlediska modelování	18
5.1 Plánovaný výstup generátoru	18
5.2 Navržené rozdělení modelu	18
5.3 Délkové jednotky, měřítko	19
5.4 Parametrizace čepele	19
5.4.1 Základní rozměry čepele	20
5.4.2 Profily čepele	20
5.4.3 Zúžení čepele	20
5.4.4 Ztupení horního ostří (dvojbřítost x jednobřítost)	21
5.4.5 Drážky	21
5.4.6 Vlastnosti hrotu	22
5.4.7 Ricasso	22
5.4.8 Zakřivení	23

5.5	Typy jílčů	23
6	Vytváření modelu	24
6.1	Princip automatického generování objektů v Blenderu	24
6.1.1	Vytvoření meshe	24
6.1.2	Použití modifikátorů	26
6.2	Princip vytvoření čepele	26
6.3	Složení modelu z vytvořené čepele a hotových prvků	28
6.4	Vytváření čepele detailně	28
6.4.1	Základní rozměry	29
6.4.2	Profily	29
6.4.3	Drážky	32
6.4.4	Podélné a profilové zúžení	33
6.4.5	Ricasso	35
6.4.6	Ztupení horního ostří	35
6.4.7	Hrot	36
6.4.8	Zakřivení	36
6.5	Optimalizace	36
7	Uživatelské rozhraní a realizace generátoru	38
7.1	Výchozí požadavky na uživatelské rozhraní	38
7.2	Výsledný vzhled	39
7.3	Způsob implementace	40
7.4	Použité nástroje a technologie	40
7.5	Přechod na další stabilní verzi Blenderu (2.6)	40
7.6	Testování, zpětná vazba	41
8	Závěr	43
A	Obsah CD	46
B	Meče - Slovníček pojmů	47
C	User Manual	49
C.1	Introduction	49
C.2	First steps	49
C.3	Advanced settings	49
C.4	Using the source code, extending functionality	50
D	Ukázky renderovaného výstupu	52

Kapitola 1

Úvod

Myšlenka automatizovat nějaký proces tak, aby probíhal bez zásahu člověka, byla odjakživa hnacím motorem pokroku. V počátcích dějin techniky si konstruktéři a vynálezci kladli za cíl osvobození dělníků od úmorné dřiny, později se stal středem pozornosti vyšší výkon a lepší zisk. S příchodem počítačů inženýři dostali možnost dát "obyčejným" lidem mocné nástroje, které jim každodenně pomáhají vykonávat činnosti, jež by jinak sami nesvedli. Automatické generování 3D objektů naplňuje všechny výše uvedené aspekty - ušetří práci umělcům i peníze vývojářům a běžnému uživateli PC poskytne příležitost získat model, který by mnohdy sám vůbec nedokázal vytvořit.

Automatické vytváření obsahu (ať už pro animovaný film, počítačovou hru, fyzikální simulaci či cokoliv jiného) je navíc i jistý krok kupředu. V budoucnosti lze takový generátor nejen využívat pro ulehčení práce, ale také na něm stavět a zařazovat jej jako jednu z mnoha komponent do složitějších, ambicióznějších projektů.

Tento dokument má čtenáře postupně provést celým procesem vytváření generátoru 3D objektů pro program Blender, od stručného seznámení, přes návrh a realizaci až k závěrečnému testování a zhodnocení.

V úvodní části se nachází krátké seznámení s programem Blender a popis jeho důležitých vlastností. Je také přiblíženo vytváření skriptů a skriptovací jazyk Python, protože to jsou klíčové nástroje pro vypracování zadaného úkolu.

Poznámka o používání odborných pojmů

V celém dokumentu se objevuje řada odborných termínů nejen z oblasti počítačové grafiky. Pokud to situace umožní, přidrhuje se text českých překladů. U některých pojmů se však v praxi používají téměř výlučně anglické varianty (dokonce i v odborné literatuře), např. u slova *mesh*. V takovém případě dokument zůstává u používanějších verzí, aby čtenář nebyl zbytečně zmaten.

Co se týká odborných pojmů z oblasti počítačové grafiky, předpokládá se, že čtenář rozumí alespoň těm základním (což by mělo být pro čtení tohoto textu zcela dostačující). Pro snazší orientaci v názvosloví spojeném s třídou objektů, na jejichž generování je tento projekt zaměřen, nabízí příloha stručný slovníček pojmů. Na jeho konci se také nachází vysvětlivky některých speciálních pojmů zavedených pro potřeby této práce.

O autorství přiložených obrázků

Pokud není u obrázků výslovně uvedeno jinak, je autorem veškerého grafického obsahu autor tohoto textu. Všechny převzaté obrázky jsou použité se souhlasem jejich autora nebo

v souladu s případnou licencí. U každého takového obrázku je uveden odkaz na zdroj.

Poznámka k použité literatuře

Veškeré citované zdroje, u kterých je uveden internetový odkaz, byly dostupné minimálně k datu odevzdání této diplomové práce.

Kapitola 2

Blender a Python

Motto:

Blender is about people.

- Ton Roosendaal

Autorem výše uvedeného citátu [2] není nikdo jiný, než duchovní otec a vedoucí vývojář Blenderu. Tento výrok velmi dobře ilustruje přístup Blenderu k uživateli. Jedná se o open-source (a tedy zdarma dostupný) software pro modelování a vykreslování 3D grafiky a animace. Za jednu z klíčových předností lze označit ovládací rozhraní aplikace, které je poměrně netradiční a někdy vzbuzuje nedůvěru začínajících uživatelů. Ti pokročilí na něj ovšem zpravidla nedají dopustit. Mezi další přednosti patří dostupnost na řadě platform (Windows, Linux, Mac OS X a Solaris) [9] a opravdu velká skupina příznivců z řad uživatelů i vývojářů.

Ale síla Blenderu nespočívá jen ve skvělé dokumentaci, podpoře komunity, multiplatformnosti, uživatelské přívětivosti a dostupnosti. Jedná se totiž o velice všestranný nástroj, který lze používat mimo jiné i pro vytváření modelů a map (resp. úrovní/levelů) k existujícím 3D aplikacím (nejen hrám) a tato díla pak mohou být jednoduše exportována díky skriptům - a zde se ke slovu dostává jazyk Python. Přes skripty psané v Pythonu se dá ovládat prakticky celý Blender - lze importovat a exportovat data, automatizovat činnosti v uživatelském rozhraní, vytvářet či měnit modely,...

2.1 Ukázky (možnosti Blenderu)

Kvality Blenderu lze dobře demonstrovat pomocí nějakého již hotového díla a spíše než slovní popis se nabízí vizuální ukázka. Jakých výsledků je možné v tomto programu dosáhnout, ukazuje obrázek 2.1. Na svých webových stránkách [15] autor dokonce vysvětluje, jakým způsobem se k výsledku dopracoval.

Velice pěknou kolekci hotových děl je možné nalézt v galerii na stránkách nadace Blenderu [7]. Na diskusních fórech věnovaných Blenderu [8] se také nachází mnoho hotových příkladů a ukázek ke stažení a prostudování.

2.2 Klíčové vlastnosti

Podrobné vypisování vlastností Blenderu je nad rámec tohoto textu. Navíc jsou tyto informace dostupné z mnoha knižních i internetových zdrojů. Zde se tedy uvádí jen bodový



Obrázek 2.1: Grapes od Lucase Vebera, renderováno pomocí Blenderu - použito s laskavým svolením autora. Zdroj: [15]

seznam těch nejdůležitějších (z hlediska prezentované práce), čerpaný z [9]:

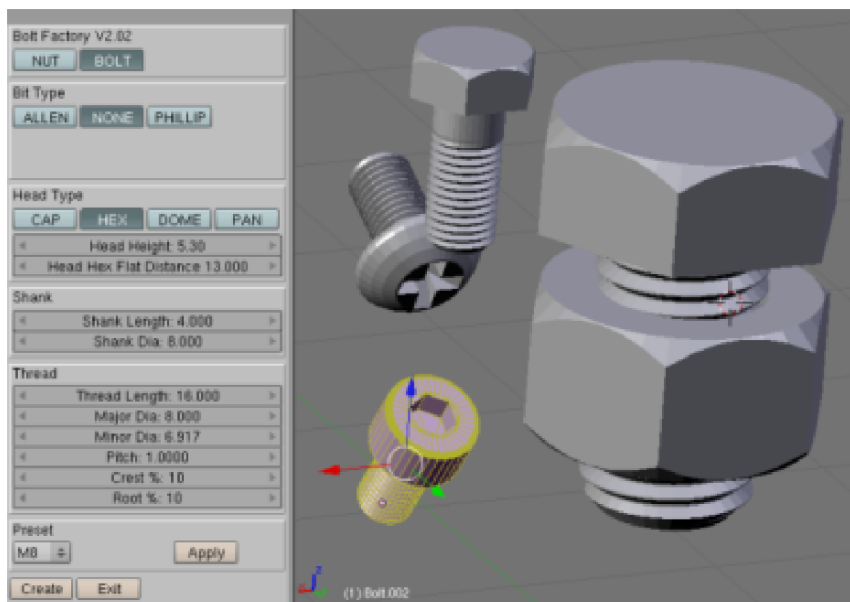
- Flexibilní a plně konfigurovatelné pracovní prostředí
- GUI pro skripty v Pythonu
- Řada typů 3D objektů (NURBS, B-spline,...)
- Modifikátory meshů
- Pokročilé metody renderování
- Široká paleta nástrojů pro texturování
- Podpora čtení a zápisu pro širokou paletu 2D a 3D formátů
- a mnoho dalších...

2.3 Existující skripty

Následuje konkrétní seznámení s již existujícími skripty pro Blender, které mají blízko k zadání této práce. V oficiálním katalogu [18] (odkud se také nachází níže uvedené skripty) jsou jednotlivé položky rozříděné do několika skupin, podle svojí funkce (např. animace, CAD nástroje, export, import, průvodci,...). Výše uvedený zdroj je pouze jediným z mnoha míst, kde se dají sehnat užitečné skripty, avšak poskytuje dostatečně velký výběr, pro potřeby této kapitoly zcela postačující. V případě hlubšího zájmu se čtenáři doporučuje použít internetové vyhledávače a diskusní fóra věnovaná Blenderu, např. [8].

2.3.1 Bolt Factory

Skript, jehož autorem je Aaron Keith, vytváří modely různých druhů šroubů a matek. Uživatel má k dispozici mnoho různých parametrů, které může nastavovat pomocí grafického rozhraní. Dá se ovlivnit např. výška, typ a průměr hlavy, délka a průměr dřívku, průměr závitů, ... Ačkoliv práce s Bolt Factory je poměrně jednoduchá a přímočará, výsledky vypadají velice dobře, jak je možno si všimnout na obrázku 2.2.



Obrázek 2.2: Ukázka práce se skriptem Bolt Factory

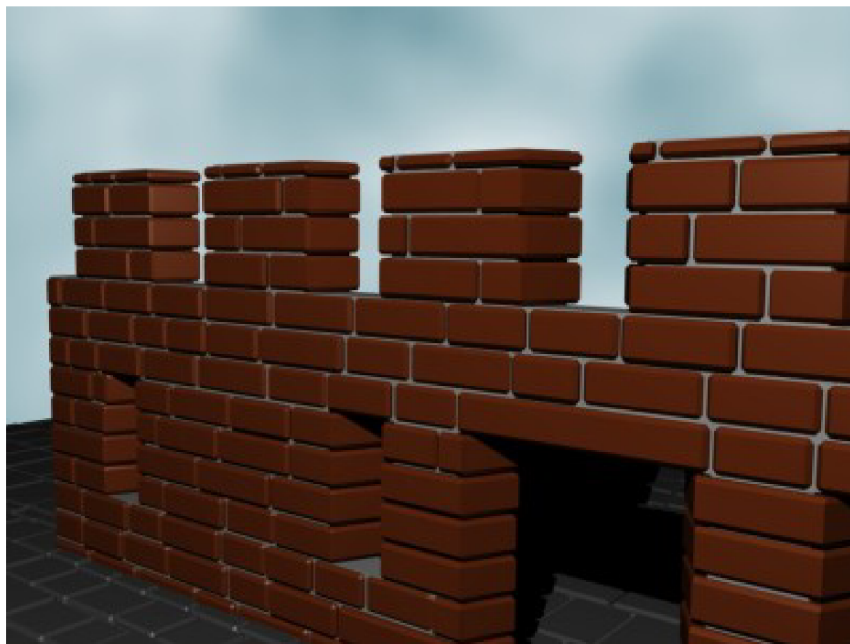
2.3.2 Auto Masonry

Auto Masonry vytváří kamenné nebo cihlové valy, zdi, cesty, atp. Lze dokonce použít křivku, podle které se objekt vytvoří (a tedy usnadňuje například generování klikatých dlážděných cest). Velmi rychle se tak dají získat i kamenné budovy či celá města. Poslední verze byla již dokonce přepracována pro Blender verze 2.5 (v době psaní této práce se za poslední stabilní verzi považuje 2.49). Autorem je Paul Spooner. Práce, jejíž výsledek ilustruje obrázek 2.3, netrvala se skriptem více než několik minut, mimo jiné i díky existenci grafického rozhraní pro nastavení parametrů. Ručně vytvořeny byly pouze materiály a jejich barvy.

2.3.3 L-System fixed seed treemaker

Původní skript Armagana Yavuze, upravený Jean-Michel Sorelem, generuje rostliny pomocí tzv. L-Systému. Skript je integrován do *.blend* souboru, součástí je i několik předvytvořených druhů listů a scéna je také přednastavena pro renderování. Pomocí různých parametrů lze zvolit charakteristiky rostliny (např. rozložení listů na větví, jakým způsobem se kmen větví, ...) a ovlivňovat výsledný tvar (zakroucení, délka, ...). Stromek vytvořený tímto nástrojem ukazuje obrázek 2.4.

Podstatou původního skriptu je generování dvou *meshů*; jeden pro strukturu (kmen a větve) a jeden pro listy nebo plody (tudíž nelze mít oboje najednou). Verze Jean-Michel



Obrázek 2.3: Renderovaný výstup skriptu Auto Masonry

Sorela tento nedostatek odstraňuje tím, že umožňuje zachovat strukturu rostliny a tak dovolu-
je přidávat více prvků (např. plody). Toho je dosaženo díky neměnnému *seedu* a možnosti
ukládat množinu parametrů do souboru. [13]

L-Systém je systém založený na gramatikách (na principu přepisování řetězců), vynale-
zený Aristidem Lindenmayerem (v roce 1968). Později se zjistilo, že se jedná o nástroj
užitečný pro počítačovou grafiku. Przemyslaw Prusinkiewicz pak stál v čele snahy vyvinout
komplexní systém pro modelování rostlin a jejich růstu. [1] Zde prezentovaný skript těchto
technik využívá.

2.3.4 L-System Buildings

L-Systém, zmiňovaný výše, se hodí nejen pro modelování organických objektů. Důkazem je
například skript v katalogu nazývaný L-System Buildings. Zajímavé je, že jako elementární
bloky používá předem vytvořené *meshe*, díky čemuž je vysoce flexibilní. Ukázky výsledků,
kterých lze dosáhnout, jsou k vidění na fóru Blender Artists, v tématu věnovaném přímo
tomuto skriptu (viz. [12]).

2.3.5 Make Human

Přestože skript Make Human je již zastaralý a nepracuje se na něm, byl zahrnut do tohoto
seznamu jako ukázka toho, kam až může podobný projekt dospět. V současnosti je totiž
k dispozici samostatná freeware aplikace Make Human, která z něj vychází. Lze ji definovat
jako nástroj k vytváření modelů lidského těla s možností nastavit mnoho parametrů (věk,
proporce, pohlaví...). Od svého vzniku prošla rozsáhlým vývojem a poslední verze dává
velice zajímavé výsledky, dokonce umí vytvořit i armaturu pro budoucí animaci postavy.
Pro podrobnější informace viz. domovská stránka projektu Make Human. [14]



Obrázek 2.4: Stromek vytvořený pomocí L-System fixed seed treemaker

Kapitola 3

Výběr třídy objektů

Před zvolením konkrétní třídy objektů, která se bude vytvářet, je třeba objasnit, co se od generátoru očekává a jaké jsou vlastně jeho možnosti. Částečně to již bylo nastíněno v úvodu, ale je nutné se na celou problematiku podívat konkrétněji.

3.1 Výhody plynoucí z automatického generování

- Rychlé vytvoření modelu
- Zpřístupnění modelovacího nástroje širšímu okruhu uživatelů
- Možnost rychle a snadno získat velké množství podobných modelů

Dále je nezbytné se zamyslet nad oblastmi použití a skupinami uživatelů, kteří z něj mohou potenciálně těžit. Jsou to především:

- Animátoři a tvůrci CG filmů
- Herní vývojáři (3D grafici, návrháři úrovní,...)
- Nezkoušení a mírně pokročilí uživatelé Blenderu (studijní a jiné účely)

3.2 Shrnutí požadavků

Má-li být vybrána konkrétní třída objektů jako cíl tohoto projektu (tak, aby byl smysluplný), musí být respektovány určité všeobecné požadavky, kladené na vybranou skupinu:

- Objekty by se měly používat poměrně často (nemá velký smysl automaticky generovat věc, kterou pak není potřeba opakovaně vytvářet)
- Tvar (případně struktura) objektu je netriviální (takže strojové vytváření skutečně ušetří nezanedbatelné množství času)
- Objekt se dá vhodným způsobem parametrizovat (tzn. mimo jiné i to, že třída objektů není příliš konkrétní)
- Existuje poptávka po automatickém generátoru objektů dané třídy (a nebo se předpokládá)
- Nejsou dostupné žádné jiné kvalitní nástroje, které by už tuto úlohu zvládaly (nemá smysl znovu vytvářet něco, co je už hotové a dobře to funguje)

3.3 Vybraná třída objektů

Skupinou, která byla nakonec zvolena, a která dobře odpovídá výše uvedenému, jsou různé druhy mečů. Není to samozřejmě jediná možnost a při výběru přicházelo v úvahu mnoho dalších, stejně vhodných alternativ, ale jedním z rozhodujících faktorů se stal autorův osobní zájem o zbraně a válečnictví, které nejlépe reprezentuje svojí elegancí a mytologickou symbolikou právě meč. Motivační úvod do problematiky mečů (kterému je věnována část jedné z následujících kapitol) snad navodí podobný zájem i ve čtenáři.

Důvody volby budou nyní rozvedeny trochu konkrétněji než v předchozím odstavci. Meč se objevuje v počítačové grafice poměrně často, nejvíce asi ve fantasy počítačových hrách a strategiích. Mají-li vývojáři do hry zakomponovat třeba i jen deset různých typů, je to již pro 3D grafika časově náročný úkol. Automatizace se tu přímo nabízí. Existují různé typy mečů (scimitary, jezdecké šavle, vikingské meče, katany, rapíry,...) a tudíž lze doufat v poměrně různorodý výstup generátoru. Zejména křížové záštity, hlavice a ochranné koše jílců bývaly v historii bohatě zdobené, proto jejich automatické (či poloautomatické) vytváření znamená profesní výzvu (z hlediska modelování ve 3D).

Konečně i užitečnost generátoru lze poměrně jasně vidět. V době psaní této práce se autorovi nepodařilo nikde nalézt jakýkoliv automatický generátor 3D modelů mečů a tak zde existuje určitý potenciál k naplnění. Kromě již výše zmíněných vývojářů počítačových her by se generátor mohl hodit i tvůrcům CG filmů - přinejmenším jako prostředek k prototypování (což posiluje samotný Blender, který je jako nástroj pro svojí jednoduchost k podobným účelům s oblibou používán). Pak je tu také velice početná skupina příznivců fantasy a hráčů Pen&Paper RPG her. Který z nich by neuvítal možnost dát svému oblíbenému meči ze hry či knihy viditelnou podobu? (Proto by měl generátor poskytovat i možnost vytvořit nerealistický, ale vizuálně přitažlivý meč - více v kapitole věnované parametrizaci.)

Poznámka k termínu "meč"

Přísně vzato je *meč* definován jako chladná zbraň, jejímiž hlavními znaky jsou přímá linie čepele, dvojbřitost, jílec v ose čepele a prostá záštita. [4] Protože od generátoru mečů pro program Blender se požaduje poněkud širší záběr, očekává se, že bude umět vytvářet i modely příbuzných zbraní, například šavlí (které mají ostří typicky pouze na jedné straně čepele, což odporuje výše uvedené definici). Kvůli jednoduchosti je tedy v tomto textu poněkud nepřesně označována celá skupina takových zbraní souhrnným pojmem *meč*.

Kapitola 4

O meči

4.1 Motivační úvod

Meč je symbolem války, práva, věrnosti a cti. Představuje nezbytný artefakt pro rytíře a rytířské řády a vůbec velmi často slouží jako ceremoniální nástroj. Je mu věnováno výsadní postavení v mnoha oblastech umění - v literatuře, sochařství, poezii... Nejen v pohádkách a v mytologii se mu dávají jména a přisuzuje magická moc. Jedná se pravděpodobně o nejpopulárnější zbraň všech dob, která provází lidstvo téměř po celou jeho zaznamenanou historii - na bojišti se objevil dokonce ještě za druhé světové války!

Repliky mečů se vyrábějí po celém světě a ačkoliv kvalita kusů zhotovených pomocí moderních technologických postupů si ani v nejmenším nezadá s těmi tradičně vyráběnými, stále je velká poptávka po mečích zhotovených s pomocí kladiva a kovadliny. Lze v tom snad spatřovat důkaz toho, že meč, ačkoliv jako nástroj války byl už překonán, stále vzbuzuje úctu a obdiv lidí. A ani dnes, v informačním věku, nesmí chybět při žádné vojenské přehlídce.

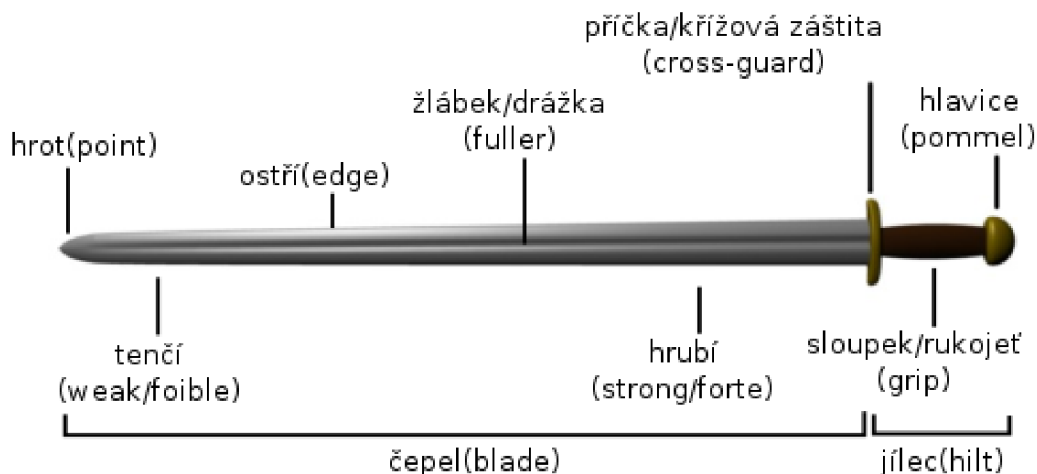
Následující text má čtenáře seznámit se základními vlastnostmi a terminologií těchto zbraní, aby bylo možné se na ně odkazovat v dalších kapitolách, a také poskytnout alespoň minimální¹ přehled typů mečů, které by měl generátor umět vytvářet.

4.2 Anatomie mečů

Tento druh zbraně se vyráběl a používal už od doby bronzové. Vyvinul se z dýky poté, co byl objeven materiál, který svými vlastnostmi umožnil delší čepel. Nejstarší zbraně, které se dají označit za meč, se datují okolo roku 3300 př. Kr. [19] Proces prodlužování ostří vyvrcholil s příchodem železa a oceli, kdy již mohly být vyráběny prakticky použitelné dlouhé meče. Se změnou stylu boje se také měnil tvar čepele i celková hmotnost.

Co se týká jílce, jeho původním účelem bylo poskytnout možnost pevného sevření zbraně a ochranu proti sklouznutí ruky na ostří či vyklouznutí meče z ruky. Za dlouhá staletí výroby mečů se přizpůsoboval potřebám různých úchopů (např. pro boj obouručně) a postupně se také vyvinuly sofistikovanější ochranné prvky - záštita se prodlužovala, ohýbala směrem k hrotu, k hlavicí meče, či stranou [4], v některých případech se dokonce zcela spojila s hlavicí (z důvodu potřeby chránit ruku před bodnými a sečnými útoky) a vznikal tak prostor pro rafinovanější tvary a zdobení.

¹Úplné popsání všech typů a variant mečů by svým rozsahem vydalo na samostatnou práci.



Obrázek 4.1: Části meče - české a anglické názvy

Přestože to na první pohled nemusí být patrné, na meči lze při podrobnějším zkoumání rozlišit mnoho vlastností a jemných detailů, které určují jeho výsledný tvar, vzhled a účel. Z nich je nezbytné vycházet, aby se zajistila variabilita výsledných modelů. Další text popisuje nejdůležitější části, které budou později významné při parametrizaci. Přehled a stručné definice všech pojmů se nachází také ve slovníčku pojmů v příloze.

4.2.1 Základní prvky meče

Prakticky každý meč (bez ohledu na svůj účel či období, z něž pochází) se skládá z *čepel* a *jílce*. Jednotlivé součásti jsou ukázány na obrázku 4.1 (v závorkách jsou uvedeny anglické překlady). Jílec je typicky tvořen *příčkou*, *sloupkem* a *hlavicí*.

Příčka, někdy označovaná jako křížová záštita, dosedá na konec čepel (tam kde začíná jílec), odděluje čepel od sloupku a její primární funkcí je zabránit sklouznutí ruky na čepel (při bodání) a ochrana ruky s předloktím při boji. Někdy se místo, kde leží příčka, nazývá *rameno* (anglicky "shoulder") meče. S postupem času docházelo ke změnám ve stylu boje a to si vyžádalo lepší způsob ochrany ruky - vznikaly komplexnější druhy záští, jako jsou pěstní oblouky, drátěné koše atd.

Sloupek (rukojeť) je tvořen tzv. *trnem* (či řapem, anglicky "tang"), který je vytažen přímo z čepel, je tedy její součástí. Trn se obkládá dřevem, kostí, parohem, slonovinou či jiným materiálem, aby se dal dobře držet v ruce. Sloupek může být také potažen či ovinut textilem, kůží, drátem apod.

Hlavice pak ukončuje jílec i celý meč. Kromě zajištění a upevnění jílice na trn slouží také jako opora pro pěst svírající meč a jako protizávaží těžké čepeli. Nezanedbatelný je také její ozdobný význam. Zejména ve starší literatuře se někdy označuje jako *jablko* [20], snad pro svůj charakteristický kulovitý tvar.

Čepel tvoří celou sečnou/bodnou část meče. Při jedné nebo obou hranách čepel se táhne *ostří* a to buď po celé její délce, nebo jen v určité části. U konce čepel (na opačné straně, než je hlavice meče) se nachází *hrot*, který může být špičatý či zakulacený (podle toho, k jakému účelu má meč sloužit).

Na čepeli lze najít tzv. *hrubí*, což je její masívnější část blíže k jílici, která bývá nejpevnější

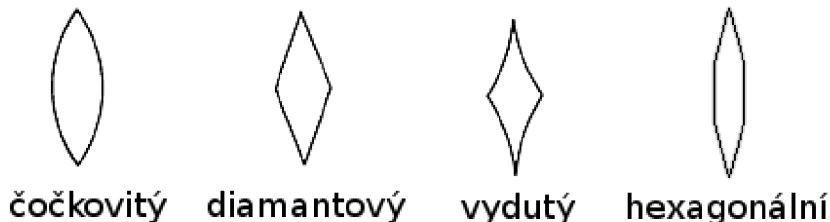
a nejodolnější částí zbraně. Naproti tomu *tenčí*, nacházející se na straně hrotu, zhruba v první třetině čepele, je lehčí a pružnější (zejména je to patrné na rapírech a kordech).

Na některých mečích je zřetelně viditelná *drážka* (či drážky), táhnoucí se po plochých stranách čepele. Také se občas označuje jako "žlábek". Jejím hlavním účelem je odlehčit meč při současném zachování pevnosti. Bývá různě široká i hluboká, jednoduchá či dvojitá, zpravidla se vyskytuje na obou stranách čepele (buď ve stejné výšce na obou stranách, nebo na jedné straně výše a na protilehlé níže - pak mluvíme o protilehlých drážkách), a nemusí vždy nutně vést po celé její délce. Nejčastější varianty ukazuje obrázek 4.2.

Typ drážky také výrazně ovlivňuje *profil čepele* - takto se označuje tvar řezu vedeného kolmo na směr čepele. Raně středověké meče, určené především k sekání, mívaly čočkovitý profil ostří, zatímco v pozdějších dobách se objevují také diamantové, vyduté či hexagonální [17] - viz. obrázek 4.3.



Obrázek 4.2: Varianty drážek (znázorněno v profilu čepele)



Obrázek 4.3: Různé profily čepele

4.2.2 Vývoj mečů

Tvar a rozměry čepele i jílice doznávaly s postupem času výrazných změn. Následující text poskytuje přehled takových variant, které budou hrát klíčovou úlohu při parametrizaci (parametry musí být zvoleny tak, aby byly tyto varianty dosažitelné jejich různým nastavením). Jedná se tedy o hrubý **přehled výstupu** generátoru.

Bude také podrobněji vysvětlena problematika záštít. Jedná se zcela jistě o nejkomplexnější část meče (především u složitých jílců rapírů.) a nezle vyloučit, že počet parametrů zde poroste nad únosnou mez². v takovém případě by zřejmě bylo vhodnější poskytnout v rámci generátoru pouze několik předvytvořených variant.

²S největší pravděpodobností se to ukáže již během parametrizace, event. až během návrhu řešení.

Meče ve starověku

Pro toto období jsou charakteristické kratší čepele ve tvaru listu, tzn. jsou rozšířené v polovině bližší k hrotu. Dalšími typickými znaky jsou prakticky chybějící záštita jakéhokoliv druhu, žádná hlavice a celková velikost jílce pouze o něco větší než pěst. V případě, že jílce nakonec nebudou parametrizovány, ale zahrnuty jako předvytvořené modely, by proto bylo vhodné poskytnout alespoň jeden odpovídající prototyp.

Vikingské a středověké meče

Tato "třída" mečů je asi nejznámější a také se nejčastěji objevuje ve filmech a uměleckém stylu fantasy. Význačným je pro ně tvar kříže (díky výrazné příčce), sloupky jsou pak v zásadě variace na jedno téma, s výjimkou uzpůsobení pro boj obouručně. Tato skupina je nejbližší obecnému popisu uvedenému výše.

Rapír a kord

S tímto typem zbraní se začal klást větší důraz na obranu pomocí meče a také na bodné útoky. Čepel se stávala elegantnější - byla odlehčena drážkami a výbrusy³ a prodloužena. K sekání sloužil především hrot a tenčí (jež zůstávalo ploché, zatímco část blíže k jílcu se zesilovala kvůli odolnosti). Typické jsou obdélníkové, diamantové až čtvercové profily, u kordů pak také trojúhelníkové. Z nenabroušené části čepele (u ramene) se zrodilo *ricasso*, jež se uchopovalo palcem a ukazováčkem (příp. i prostředníčkem), čímž se docílilo lepší ovladatelnosti. Důsledkem toho se také zkracovala rukojeť.

Největších změn ale doznaly ochranné prvky. Příčka se rozšířila, tak vznikl tzv. *quillion*. Mohl být rovný či zahnutý (nejčastěji ve tvaru písmene S) - horní větev se stáčela směrem k hrotu, aby pomohla zachytit soupeřovu čepel, zatímco dolní, někdy protažená až k hlavici, utvořila *pěstní oblouk* chránící celou ruku. Kromě *drápků* sloužících k ochraně prstů spočívajících na *ricassu* přibyl i *hřbetní prstenec* (k ochraně hřbetu ruky) a proti němu *palcový kroužek*. Výše zmíněné části jsou vyznačeny na obrázku 4.4. Jako jistá jednodušší alternativa se objevuje *ochranný koš*, drátěný či z plechu, ve tvaru např. misky nebo zvonce. [4]

Zbojříři občas kombinovali oba systémy a vznikaly složité, avšak funkční a bohatě zdobené prvky ochrany. Misky a zvonce se v některých případech zdobily perforacemi.

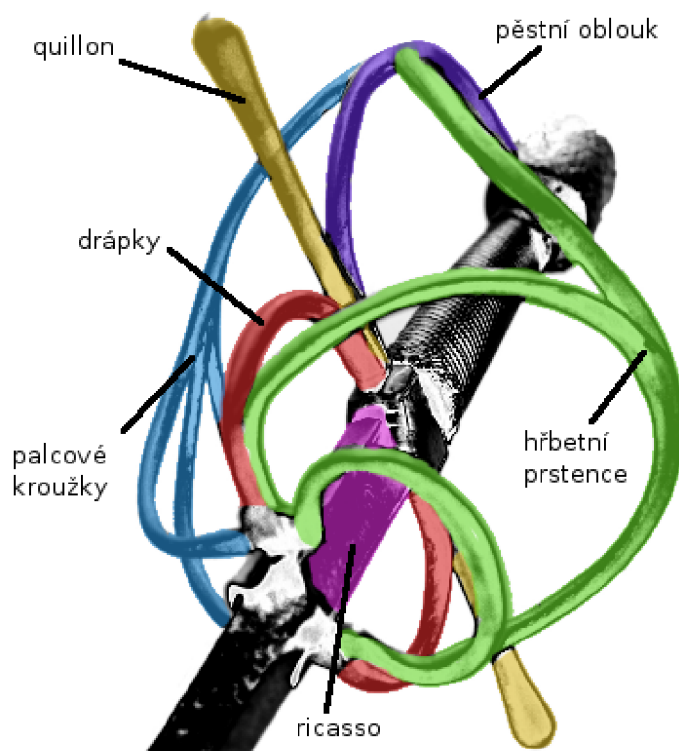
Šavle a palaše

V období rapírů si vojenské verze nových typů mečů zachovaly masívnější čepele a větší tzv. pádnost (zjednodušeně řečeno vyšší razanci při seku). Široká čepel, vhodná k sekům i bodům, byla často opatřována pouze jedním ostrím a je typickým znakem této skupiny zbraní. Pro parametrizaci důležitou vlastností je i vzhůru zakřivená čepel u šavlí (na rozdíl od rovných palašů) a také tzv. *falešné ostrí*, tedy druhé ostrí na horní straně čepele, které se však netáhne po celé její délce, ale nachází se jen u hrotu. [4]

4.2.3 Netradiční prvky

Kromě výše uvedených znaků se občas vyskytují jak u reálných předloh, tak u smyšlených mečů (z oblasti fantasy), různé neobvyklé atributy. Patří mezi ně například zuby na části

³To platí především pro hrubí meče.



Obrázek 4.4: Jilec rapíru - ochranné prvky. Vytvořeno autorem tohoto textu, fotografie na pozadí čerpána z [21] pod licencí CeCILL.

čepel (většinou to bývá v oblasti hrubí, ale není to podmínkou). Z hlediska parametrizace je nutné předpokládat možnost jejich výskytu na jedné či obou stranách čepel a to po celé délce či jen na části. Zuby mohou mít také různou hustotu, velikost či sklon.

Další možností je zvlnění čepel ve směru kolmém na její směr (kdy ostří není rovné, ale postupuje jako vlnka směrem od ramene k hrotu), různé výbrusy, vývrty a výřezy v čepeli (kvůli odlehčení) za zmínku stojí i např. rozšíření čepel jen v určité části⁴ a např. pouze na jednu stranu, unikátní zakřivení čepel i jílice, rytiny a nápisy na čepeli...

Které z těchto prvků byly zahrnuty jako možnosti generátoru, je uvedeno a zdůvodněno v části věnující se parametrizaci.

4.3 Dekompozice jako pomůcka při generování

Výše uvedené poznatky napovídají, že meč lze rozdělit na několik částí, které vykazují shodné či podobné vlastnosti. To je možné zúročit v podobě dekompozice generovaného objektu na jednodušší části, vytváření každé z nich samostatně a jejich následného složení. Struktura kódu bude přehlednější a celý postup se také značně zjednoduší. Zavedení jisté hierarchie a rozřídění prvků usnadní i orientaci v uživatelském rozhraní (což jistě ocení koncoví uživatelé).

V případě nutnosti lze také vyčlenit některé složité podčásti z parametrizace a zahrnout je do celku jako sadu předvytvořených prototypů, které se při procesu vytváření pouze

⁴To by mohlo být užitečné například i pro vytváření čepelí typických pro starověk

proporciálně upraví, aby korektně zapadaly do modelu. Blíže se této problematice věnuje další kapitola.

Kapitola 5

Dekompozice meče z hlediska modelování

Následující text přibližuje klíčové myšlenky vhodného rozdělení modelu tak, aby bylo maximálně usnadněno jeho generování. v úvodní části je naznačeno, co by měl generátor umět vytvořit. Dále je navrženo logické rozdělení modelu a nastíněno podrobnější rozpracování této problematiky (jako výhled do budoucna). Druhá polovina kapitoly seznamuje čtenáře s potřebnými parametry, vysvětluje je a uvádí možný rozsah jejich hodnot. na závěr jsou představeny typy jílců, které budou zahrnuty do nabídky předem vytvořených komponent (tento seznam se snaží respektovat všechny typy mečů, které by měl generátor umět vytvořit).

5.1 Plánovaný výstup generátoru

Aby generátor pokryl co nejširší výběr mečů a dovolil tak vytvářet většinu běžných exemplářů, musí zvládnout minimálně následující typy:

- Starověké meče (např. gladius či spatha)
- Středověké meče (vikingske, obouruční, tzv. "bastard", ...)
- Rapíry
- Kordy
- Šavle a palaše
- Některé další, méně typické (např. tzv. "flamberge")

Z výše uvedených požadavků plyne, že kromě volby typu profilu meče a druhů žlábků by měl mít uživatel možnost nastavit především podélné zúžení, délku ricassa (nenaostřená část čepele), způsob naostření meče (tj. např. zda bude ostří dvojité), tvarově a rozměrově přizpůsobit hrot a vybrat (případně ručně nadefinovat) zakřivení čepele.

5.2 Navržené rozdělení modelu

V kapitole 4 byl položen teoretický základ pro rozklad modelovaného objektu na menší části. Ideální rozdělení, které z těchto znalostí vyplývá, však svým rozsahem není možné

realizovat v rámci jedné diplomové práce. Bylo tudíž přikročeno ke kompromisnímu řešení, kdy se čepel vytváří na základě parametrů a následně se k ní přiřadí jílec, který si uživatel zvolí z nabídky předem vytvořených komponent.¹

Automatické generování jílců na základě vstupních parametrů je samozřejmě teoreticky možné. Vzhledem k rafinovanosti tvarů ochranných prvků by autor práce uvažoval o použití gramatik pro generování základních tvarů. Následné zdobení, např. rytiny, by se vzhledem k optimalizaci modelu řešilo ideálně použitím textur a/nebo techniky *normal mappingu* (Kterou autor považuje pro tuto specifickou aplikaci vhodnější než *bump mapping* - Blender však podporuje oba dva přístupy [10]). Tyto textury by opět mohly být buď předem vytvořené, nebo procedurálně generované.

Jak je vidět, existuje zde obrovský potenciál pro budoucí rozšíření. Přestože do současné verze (odevzdávané s touto diplomovou prací) nejsou zahrnuta, později mohou být doplněna s cílem zjemnit logické rozdělení vytvářených objektů a poskytnout tak uživateli širší paletu nástrojů a lepší možnost přizpůsobení modelu. Následující seznam podčástí meče slouží jako nástin této myšlenky (jedná se o části meče, které by se generovaly samostatně):

- Čepel
- Křížová záštita
- Sloupek
- Hlavice
- Hřbetní prstenec
- Pěstní oblouk
- Palcový kroužek
- Ochranný koš

5.3 Délkové jednotky, měřítko

Jedna délková jednotka v Blenderu nemá pevně určený význam (přesněji neodpovídá jí konkrétní reálná jednotka míry, např. metr). Je to dáno mj. tím, že není určen primárně jako CAD-ový systém pro technické vizualizace, ale spíše se zaměřuje na vizualizace umělecké.²

Proto ani generátor mečů nemá přesně určené míry. pro přibližný odhad skutečných rozměrů lze v rámci generátoru považovat jednu jednotku Blenderu za 10 cm. Není to však podstatné, protože při vytváření mečů záleží především na rozměrech relativních (tzn. např. délka x šířka čepel). Výsledný 3D model lze v Blenderu podle libosti jednoduše zvětšit či zmenšit.

5.4 Parametrizace čepel

Nyní bude podrobně vysvětlena problematika parametrizace. Jsou zde uvedeny všechny použité parametry, jejich význam a rozsah vstupních hodnot.

¹Tím pádem bude nutné distribuovat jako generátor mečů celý *.blend* soubor, nikoliv pouze zdrojový kód Pythonu. Podrobněji se této problematice věnuje kapitola 7.

²Přesto existují iniciativy pro přiblížení Blenderu typickému "CAD stylu" práce, podrobnější informace jsou k dispozici např. na webových stránkách [6].

5.4.1 Základní rozměry čepele

Parametry pro základní rozměry čepele jsou tři: délka, šířka a tloušťka. Jejich vyznačení a orientace v souřadném systému je dobře vidět na obrázku 6.2.

Délka - Určuje délku čepele, od ramene (tj. místa kde navazuje jílec) k hrotu. do této délky se nezapočítává samotná délka hrotu, neboť parametry hrotu se zadávají zvlášť (je to z důvodu lepší přehlednosti). Jediné teoretické omezení tohoto parametru je na kladné hodnoty, při realizaci však bude vhodné nastavit nějaký horní limit, aby se zamezilo zbytečnému generování bizarních tvarů. Hodnota je absolutní (tzn. v jednotkách Blenderu).

Šířka - Rozměr ve křížové záštity, neboli šířka od ostří k ostří (resp. od ostří ke hřbetu, pokud není ostří oboustranné). Podobně jako v případě délky musí být hodnota parametru nenulová, při realizaci by měl z praktického hlediska být horní limit zhruba 7x menší než horní limit pro délku. Hodnota je absolutní (tzn. v jednotkách Blenderu).

Tloušťka - Tloušťka čepele ve směru kolmém na její šířku. Hodnota musí být nenulová a její horní limit by neměl přesáhnout 20% maximální povolené šířky čepele (kvůli zachování realistických proporcí meče). Hodnota je absolutní (tzn. v jednotkách Blenderu).

5.4.2 Profily čepele

Uživatel si může zvolit jeden z následujících tvarů profilu:

- Čočkovitý
- Diamantový
- Vydutý
- Hexagonální

Jak vypadají jednotlivé profily je ukázáno na obrázku 4.3 v podkapitole 4.2.1.

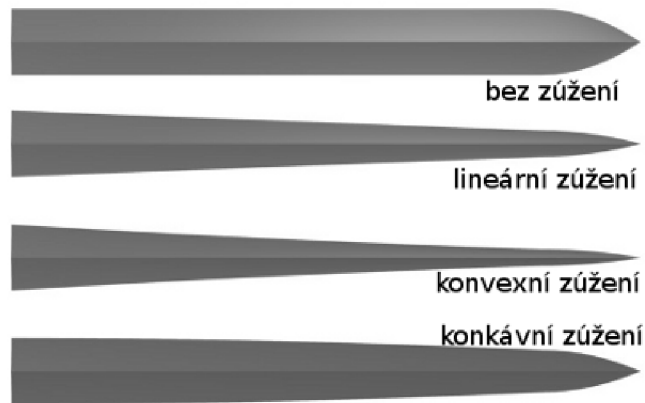
5.4.3 Zúžení čepele

Co se týká zúžení čepele, rozlišuje se podélné a profilové zúžení. Je to míra, s jakou se čepel zužuje směrem k hrotu. Podélné zúžení ovlivňuje šířku čepele, zatímco profilové její tloušťku. Protože profilové zúžení čepele není okem příliš dobře viditelné, lze tento parametr zanedbat a u profilového zúžení použít stejné hodnoty, jako u podélného. pro nastavení podélného zúžení jsou nutné dva parametry: míra a průběh zúžení.

Průběh zúžení - Způsob zužování čepele. pro napodobení většiny mečů stačí tři: lineární, konvexní a konkávní. Lineární průběh je konstantní, tzn. že od ramene ke hrotu meče se čepel zužuje stále se stejnou mírou. Konvexní zúžení je nejmarkantnější u ramene a zeslabuje se směrem k hrotu, zatímco konkávní naopak u hrotu nabírá na síle.

Míra zúžení - Udává, jak bude zúžení razantní. Dolní hranicí je 0, což znamená, že se čepel nebude zužovat, ale zachová si konstantní šířku po celé svojí délce. Maximální hranicí představuje stav, kdy se čepel na svém konci zúží do jediného bodu. s ohledem na uživatele by tento parametr měl být relativní, tzn. že horní limit reprezentuje číslo 1.

Na obrázku 5.1 jsou ukázané všechny varianty podélného zúžení.



Obrázek 5.1: Možné varianty podélného zúžení - shora dolů: bez zúžení, lineární, konvexní, konkávní.

5.4.4 Ztupení horního ostří (dvojbřítost x jednobřítost)

Především šavle a palaše se vyznačují jedním ostřím, proti kterému (na druhé straně čepele) je tupá hrana, nazývaná hřbet. Tuto vlastnost lze simulovat ztupením (dalo by se také říci "obroušením") horního ostří meče.

K tomu stačí jeden číselný parametr, který relativně vyjadřuje míru tohoto "obroušení". Jeho hodnoty se mohou pohybovat mezi 0 a 1, přičemž 0 značí nulové (tedy žádné) ztupení, zatímco hodnota 1 reprezentuje "obroušení" horního ostří až do poloviny šířky čepele.

5.4.5 Drážky

V podkapitole 4.2.1 byly představeny základní typy drážek (jsou také znázorněné na obrázku 4.3). pro jejich specifikaci v uspokojivém rozsahu je nutné vybrat jeden ze tří základních typů: jednoduchá drážka, dvojitá a protilehlé drážky. Další parametry jsou číselné a vyjadřují: šířku, hloubku a místo začátku a konce drážky.

Typ drážky - Výběr jednoho ze základních typů: jednoduchá drážka, dvojitá drážka, protilehlé drážky.

Šířka drážky - Určuje rozměr drážky ve směru šířky čepele. Hodnota je relativní k šířce čepele a musí být kladná. Protože vyjadřuje poloměr kružnice, podle které se drážka aplikuje na profil, může být maximální hodnota větší než 1 (přestože je parametr relativní) - souvisí to se způsobem, jakým se výpočet deformace modelu provádí (podrobné vysvětlení je uvedeno v podkapitole 6.4, zabývající se detailním popisem způsobu generování čepele). Během testování se jako vhodná maximální hodnota ukázalo číslo 2,5.

Hloubka drážky - Vyjadřuje, jak hluboko do materiálu čepele je drážka "vykrojena". Parametr je opět relativní, a to vzhledem k tloušťce čepele. Minimální hodnota je 0 a značí, že drážka nemá žádnou hloubku (a tudíž nebude vidět). Maximální hodnota 1 pak reprezentuje hloubku rovnající se polovině tloušťky čepele. v praxi by to znamenalo, že v nejhlubším místě drážky má čepel nulovou tloušťku, což je nesmyslné. pro realizaci se tedy zdá být vhodné omezit maximální hodnotu tohoto parametru na 0,95.

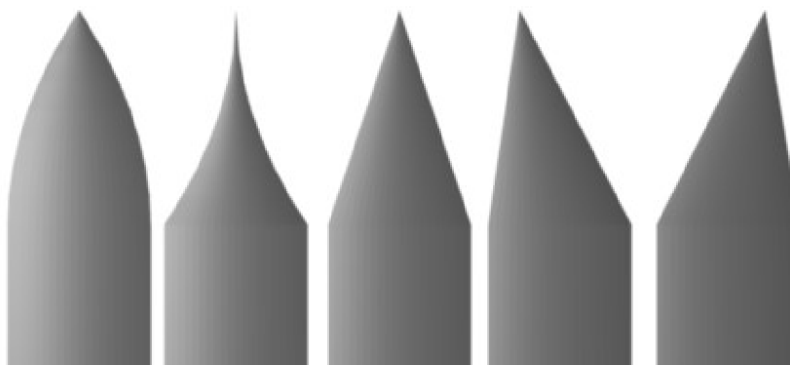
Začátek drážky - Udává vzdálenost, v jaké začíná drážka (směrem od ramene meče). Hodnota parametru je relativní k délce čepele, kdy 0 znamená začátek přímo v místě kde jílec navazuje na čepel a 1 reprezentuje místo, kde začíná hrot meče. Zejména z estetických

důvodů se při testování ukázalo rozumné omezit obor hodnot poněkud přísněji, na 0,0-0,99.

Konec drážky - Představuje vzdálenost, ve které končí drážka (směrem od ramene meče) a může být chápána podobně, jako parametr *začátek drážky*. Hodnota je také relativní vzhledem k délce čepele, s teoretickým omezením 0 až 1. v praxi se pak nejvíce osvědčilo omezení na hodnoty 0,01 - 0,99.

5.4.6 Vlastnosti hrotu

Pro rozšíření možností generátoru je nezbytné umožnit nastavení vzhledu hrotu meče v jistém smyslu slova nezávisle na zbytku čepele. pro dosažení většiny potřebných tvarů hrotu jsou nutné tyto parametry: délka, typ zúžení a asymetrie. Vybrané tvary ukazuje obrázek 5.2.



Obrázek 5.2: Některé varianty hrotů. Zleva: konkávní, konvexní, lineární, lineární vychýlený nahoru, lineární vychýlený dolů.

Délka hrotu - Určuje délku hrotu (tato délka není započítána do délky čepele). Jedná se o absolutní hodnotu (tzn. v jednotkách Blenderu) a teoreticky by měly být hodnoty parametru omezeny na kladná čísla. v praxi by potom, s ohledem na zachování proporcí, neměla horní hranice přesáhnout maximální délku čepele (ačkoliv to samozřejmě teoreticky může být dovoleno). Během testování se jako ideální ukázala nejvyšší dovolená hodnota 2,9.

Typ zúžení - Vyjadřuje průběh zužování hrotu. Prakticky každý hrot by se na konci meče měl zakončit v jednom bodě, proto není nutný parametr určující míru zúžení (jako tomu je u podélného zúžení čepele) - ta je vždy maximální. Tudíž stačí jen výběr ze tří typů: lineární, konvexní a konkávní. Význam je stejný, jako v případě *průběhu zúžení* čepele.

Asymetrie - Vychýlení konce hrotu směrem k hornímu či dolnímu ostří. U některých mečů je hrot zahnutý směrem nahoru (poměrně často si toho lze všimnout u šavlí) a proto je tento parametr celkem významný. Jeho hodnota je relativní, pohybuje se mezi 0 a 1, kde 0 značí maximální vychýlení směrem dolů (k dolnímu ostří), 0,5 představuje střed čepele a 1 reprezentuje největší možné vychýlení nahoru. Vychýlení větší než šířka čepele je teoreticky možné, avšak z hlediska funkčnosti přebytečné, tudíž nežádoucí (tohoto efektu se snadněji a intuitivněji dosáhne pomocí zakřivení čepele - viz. dále).

5.4.7 Ricasso

Ricasso je nenaostřená část čepele a často se objevuje především u šavlí a palašů. Není však neobvyklá ani u středověkých mečů. pro základní vyjádření jeho vzhledu jsou nutné mini-

málně následující parametry: délka, šířka a tloušťka. do budoucna je možné tuto paletu ještě rozšířit, např. o jiné profily (v současné verzi je zahrnut pouze nejtypičtější - obdélníkový).

Délka ricassa - Určuje délku ricassa (ve směru čepele). Hodnota je relativní vzhledem k délce čepele, kdy 0 znamená, že ricasso se vůbec nevytvoří, zatímco hodnota 1 by znamenala že ricasso zabere celou délku čepele. v praxi je maximální hodnota parametru omezena na 0,4, což je hodnota zvolená na základě testování s ohledem na realističnost a estetický vzhled.

Šířka ricassa - Udává šířku ricassa (v podobném významu, jako je určena šířka čepele). Parametr je však relativní vzhledem k šířce čepele. pro hodnotu 0 by byla šířka nekonečně malá, proto je dolní hranice v praxi omezena na 0,2. Maximální šířka je teoreticky neomezená, z praktického hlediska je však nastavena na 2 (tedy dvojnásobek šířky čepele).

Tloušťka ricassa - Představuje tloušťku ricassa, podobně jako je určena tloušťka čepele. Parametr je však relativní právě vzhledem k tloušťce čepele. Teoretické omezení je opět pouze na kladná čísla, avšak při testování byly meze nastaveny na hodnoty 0,1 - 2,0 (kdy 2,0 znamená tloušťku dvakrát větší, než je tloušťka čepele).

5.4.8 Zakřivení

Parametrizovat zakřivení by bylo složité a vzhledem ke své matematické podstatě by to neúnosně zvyšovalo nároky na uživatelské pochopení problematiky. k dosažení co největší flexibility je proto vhodné použít pro specifikaci zakřivení křivky. pro nejčastější typy zakřivení budou k dispozici předem vytvořené vzory, nicméně uživatel by měl mít možnost po vytvoření danou křivku ještě upravit.

Tím dostane do ruky absolutní kontrolu nad zakřivením modelu a možnost volného experimentování s různými tvary. Vzhledem k jednoduchosti práce s křivkami v Blenderu by tedy vlastní nastavení křivky nemělo činit potíže ani 3D modelářům - začátečníkům.

5.5 Typy jílců

Pro každou kategorii mečů je typický poněkud jiný jílec - jsou tedy nutné různé modely minimálně pro: starověké meče, středověké meče (zde by bylo vhodné rozlišit meče jednoruční a obouruční), šavle a palaše, rapíry a kordy. Generátor by měl být realizován tak, aby se daly přidávat další modely bez nutnosti většího zásahu do zdrojového kódu.

V budoucnu může být pro každý typ v nabídce více exemplářů. Co se týká samotné realizace modelu, je rozumné při jejich vytváření oddělit v rámci *meshe* části jako jsou ochranné prvky, sloupek, hlavice, atp. (například pomocí tzv. *vertex groups*). To následně usnadní jejich rozdělení, pokud by se zavedla možnost poskládání jílice z různých komponent a bude možné pro toto vylepšení snadno rychle přizpůsobit i stávající modely.

Kapitola 6

Vytváření modelu

V první části této kapitoly jsou popsána obecná východiska a postupy vedoucí k vytvoření modelu meče. Druhá polovina pak podrobně popisuje algoritmy a matematickou podstatu generování čepele. v samotném závěru je nastíněn způsob optimalizace vytvářeného modelu z hlediska počtu polygonů.

6.1 Princip automatického generování objektů v Blenderu

Při ručním modelování je v Blenderu nejčastěji využívána hraniční polygonální reprezentace (někdy se také označuje jako B-rep [3]). Geometrie objektu je popsána tzv. *meshem*, který je tvořen vrcholy (anglicky *vertexy*) v prostoru a čtvercovými, případně trojúhelníkovými plochami (*faces*) určenými právě pomocí těchto vrcholů. pro automatické generování je tento přístup obzvláště výhodný, neboť se s ním dobře pracuje při použití analytické geometrie.

Velmi často se automatický model generuje právě na základě nějaké matematické rovnice. Zajímavých tvarů je možné dosáhnout například pomocí rovnic odvozených z rovnice kružnice. Zajímavé příklady lze najít na mnoha webových stránkách, např. [16].

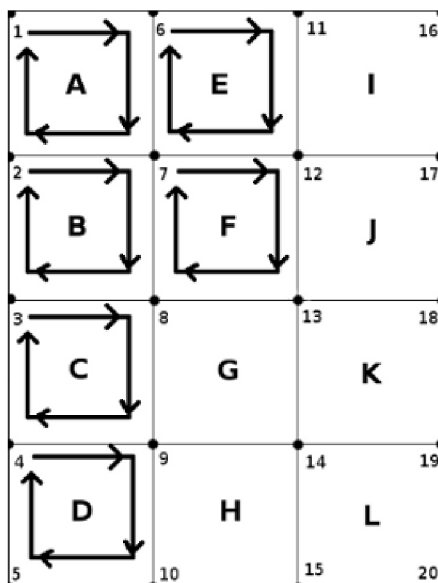
6.1.1 Vytvoření meshe

Jak již bylo řečeno, základním a prakticky jediným krokem pro vytvoření geometrie modelu je vygenerování *meshe*. Jeden z možných postupů je napřed vygenerovat všechny body ohraničující objekt a následně jejich spojením vytvořit síť čtverců (anglicky *faces*).¹ Takový přístup ovšem ztroskotá, pokud je topologie modelu komplikovanější a není předem známá (například ze znalosti osmi vrcholů pro krychli nelze bezpečně vyvodit, že všechny její stěny jsou vyplněné). Proto je nezbytné pamatovat při návrhu algoritmu pro vytváření modelu na tyto nástrahy.

Protože však topologie čepele podobná úskalí neskrývá, dá se výše popsaný způsob s výhodou aplikovat (jeho předností je v první řadě jednoduchost a přímočarost). Algoritmus vytvoření čtyřúhelníkových stěn se nejlépe demonstruje na rovné ploše, princip však zůstane stejný i pro geometrii čepele.

Celá situace je znázorněna na obrázku 6.1. na základě známého pořadí generovaných vrcholů (na obrázku vyznačeny čísla), lze čtyřúhelníkové stěny (v obrázku označené písmeny A-L) vytvořit průchodem přes seznam vrcholů. Nechť délka toho seznamu je n . v případě

¹Čtyřúhelníky jsou před trojúhelníky upřednostňovány především kvůli lepší kontrole modelu. Tato praxe je při ručním modelování mnohokrát osvědčená a proto by ji měl dodržovat i automatický generátor, mimo jiné pro případ, kdy chce uživatel automaticky získaný model dále ručně upravovat.



Obrázek 6.1: Postup vytváření čtyřúhelníkových stěn (anglicky *faces*)

pravidelného pravoúhlého objektu (jakým zvolená ukázková plocha je) se dá zjistit počet vrcholů v jednom sloupci (v obrázku jsou to vrcholy 1-5 pro nejlevější sloupec, 6-10 pro další,...) - označeno písmenem x - a také počet vrcholů v jednom řádku (y). Celkový počet stěn p se spočítá podle vztahu 6.1.

$$p = (x - 1)(y - 1) \quad (6.1)$$

V tomto případě je tedy $x=5$, $y=4$ a $p=(5-1)(4-1)=12$. Podstatou algoritmu je průchod seznamem vrcholů, kdy se ke každé stěně přiřadí čtyři odpovídající body. Vznikne tím seznam *faces*, který Blenderu spolu se seznamem vrcholů stačí k tomu, aby model zobrazil. Nejpřímochařeji lze algoritmus vyjádřit následujícím pseudokódem:

```

for (a=0; a<x-1; a++) //průchod přes sloupce
  for (b=1; b<y; b++) //průchod přes řádky
  {
    v1 = a*x+b // index prvního vrcholu (v seznamu vrcholů)
    v2 = (a+1)*x+b // ...druhého...
    v3 = (a+1)*x+b+1 // ...třetího...
    v4 = a*x+b+1 // ...čtvrtého...
  }

```

Například první *face* na obrázku je tedy tvořen vrcholy 1,6,7,2 (tak jak naznačuje šipka v obrázku). Druhý má vrcholy s čísly 2,7,8,3 atd. Postupně se tím vygenerují všechny stěny. Výhodou je, že pro aplikování na vytváření čepele bude tento algoritmus vyžadovat pouze minimální úpravy (sít vrcholů totiž bude víceméně pravidelná, jak je demonstrováno dále v tomto textu).

6.1.2 Použití modifikátorů

Blender také dovoluje aplikovat na objekt různé modifikátory, například *Subdivision surface* pro zjemnění sítě modelu, *Mirror* pro symetrické zrcadlení podle zvolené osy, ... pro generátor mečů je v současné podobě podstatný pouze jediný modifikátor, a to je deformace podle křivky (v Blenderu nazývaný *Curve*).

Díky API Blenderu je možné při běhu skriptu na existující objekt ve scéně jednoduše aplikovat libovolný modifikátor. Lze měnit i parametry tohoto modifikátoru, což se hodí především pro přizpůsobení křivky rozměrům čepele. Podrobněji se použití modifikátoru *Curve* věnuje jedna z následujících podkapitol.

6.2 Princip vytvoření čepele

Ze všech možných způsobů, které se pro vytváření čepele nabízejí, je rozumné vybrat ten nejjednodušší. Jedna možnost by byla například vygenerovat základní tvar meče a následně jej deformovat, upravovat a přidávat detail, podobně jako při ručním vytváření. Podobně postupuje i kovář, když vyrábí skutečnou zbraň - vyková přibližný tvar a pak jej opracovává a odstraňuje přebytečný materiál.²

Existuje však způsob vytvoření modelu, který je z programátorského hlediska průhlednější. v dalším textu bude označen pojmem *vytažení z profilu*. Základní myšlenkou je fakt, že každý meč má typický základní tvar profilu čepele, a to po celé své délce (což platí v drtivé většině případů).

Vytažení z profilu

Čepel se dá ze základního tvaru vytvořit "vytažením" v podélném směru (pro další vysvětlení se předpokládá, že v tomto směru probíhá osa y). Podobně se v Blenderu dá při ručním modelování použít nástroj *extrude*. Elegance metody spočívá v tom, že jakékoliv deformace prakticky redukuje ze 3D do 2D! (Bude vysvětleno v zápětí.)

Samotné vytažení z profilu by ovšem velmi dobrý výsledek nedávalo - například z diamantového profilu by tím vznikla obyčejná čtyřhranná "tyč". Je jasné, že to jsou různé strukturální vlastnosti meče, které v každém místě čepele ovlivňují její tvar - a tedy i tvar profilu.

Profilový řez

Povedeme-li v kterémkoliv místě čepele (na ose y) kolmý řez, dostaneme dvojrozměrnou plochu, která připomíná tvar profilu čepele, ale jisté detaily se liší (např. strany tohoto rovnoběžníka tvoří úsečky, ale jsou v nich "prohlubně" způsobené vlivem drážky). Tato plocha - resp. tvar - bude v textu dále nazývána *profilový řez*.

Při realizaci je profilový řez reprezentován seznamem vrcholů - ty vlastně určují tvar profilu. Počtem bodů tvořících profilový řez se navíc dá snadno ovlivnit výsledná kvalita modelu (a tedy počet polygonů).

²Moderní výroba mečů je v tomto ohledu podobná - z hotového kusu oceli se vyřízne přibližný tvar, který se následně brousí.

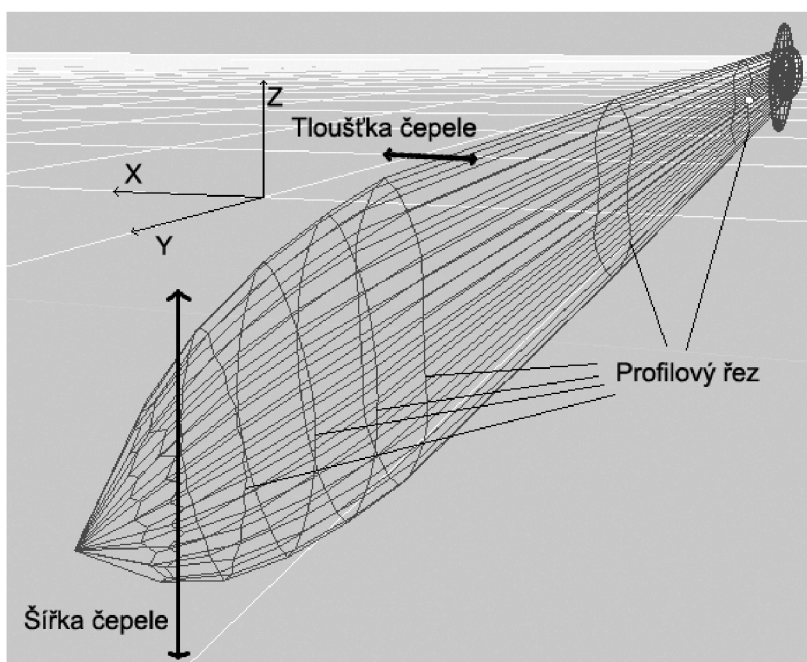
Deformační funkce

Protože parametry popisující tvar a vzhled meče jsou dobře známe (byly popsány v podkapitole 5.4), dá se snadno vypočítat, jak bude v libovolném bodě osy y vypadat profilový řez. k tomuto výpočtu slouží tzv. *deformační funkce*.

Výhodou je, že vzhledem k podstatě profilového řezu se všechny výpočty provádějí pouze ve dvou rozměrech - v rovině kolmé na osu y . To jednak urychluje výpočet a navíc se značně zjednodušuje návrh deformačních algoritmů. pro každou významnou tvarovou charakteristiku meče (tedy pro konkrétní skupinu parametrů) je dobré mít samostatnou funkci, aby se jejich efekty daly vzájemně libovolně kombinovat.

Vstupem funkce je seznam vrcholů v aktuálním profilovém řezu (tedy v tom, který se má deformovat), poloha řezu na ose y a hodnoty příslušných parametrů (např. pro deformační funkci počítající vliv podélného zúžení jsou to parametry *míra zúžení* a *průběh zúžení*). Funkce spočítá a aplikuje deformace a vrátí správně upravený profilový řez (jehož tvar přesně odpovídá tvaru meče v daném místě osy y).

Jak jsou jednotlivé profilové řezy čepele deformované drážkou je naznačeno na obrázku 6.2. Základním tvarem je čočkovitý profil, což je na všech řezech v obrázku stále patrné, nicméně změny způsobené drážkou jsou jasně viditelné. Směrem ke hrotu se však vliv zmenšuje, s tím jak drážka končí (na obrázku je přechod pozvolný, protože profilových řezů je z důvodu přehlednosti vloženo málo - v praxi je hranice konce drážky zpravidla ostrá a dobře viditelná).



Obrázek 6.2: Princip vytvoření čepele - profilové řezy.

Složení tvaru z profilových řezů, jemnost modelu

Pro získání popisu čepele pomocí vrcholů tedy stačí vložit v ose y několik profilových řezů, které jsou příslušně deformované pomocí výše popsaných funkcí. na vytvoření stěn *faces* se

pak jednoduše použije algoritmus popsany v podkapitole 6.1.1 (s několika málo jednoduchými úpravami, konkrétně např. modifikace algoritmu pro vytvoření trojúhelníků u hrotu meče, kde se na samotné špičce spojují všechny body do jediného *vertexu*). na výsledek se ještě aplikuje zakřivení pomocí modifikátoru, o což bude podrobněji popsáno později.

Na základě počtu vrcholů v profilovém řezu a také tím, jak hustě se vkládají profilové řezy na ose y , se určuje výsledná kvalita modelu. Čím více bodů profilového řezu a čím více profilových řezů na ose y , tím detailnější je model (následkem čehož má samozřejmě větší počet polygonů).

Velikost kroku vkládání bodů (profilových řezů v modelu i vrcholů v rámci výchozího profilového řezu) by měla být ve skriptu nastavitelná např. pomocí konstanty. z hlediska implementace to není žádný problém.

Výše uvedený přístup se dá velice snadno a také účinně optimalizovat, o čemž bude řeč v závěru této kapitoly.

6.3 Složení modelu z vytvořené čepel a hotových prvků

Protože uživatel by mohl chtít meč ještě upravovat, je příhodnější ponechat čepel i jílec jako oddělené objekty (spojit se dají snadno a rychle, u rozdělení to naopak nemusí vždy platit).

Co se týká úpravy velikosti jílice podle čepel, není to nezbytně nutné, dokonce to snad lze považovat za kontraproduktivní. Pokud by totiž jílec změnil velikost relativně k rozměrům čepel, přijde uživatel o možnost ovlivnit poměr velikostí čepel a jílice (všechny meče by se tím prakticky "normalizovaly" na standardní proporce).

V budoucnu by se daly doplnit parametry pro nastavení velikosti jílice, ale vedlo by to pravděpodobně jen k proporcionálně deformovaným jílcům a kýžený efekt by to stěží přineslo. Podobná vylepšení mají smysl až s přechodem ke skládání jílců z dílčích komponent (záštity, sloupky, hlavice - jak o tom byla řeč v podkapitole 5.2).

Dosáhnout při realizaci správného zarovnání jílice na začátek čepel bude jednoduché při dodržení pravidla, že čepel se začíná generovat v počátku souřadného systému. Tato podmínka je výhodná i pro generování čepel, neboť začít s vytažením čepel v bodě 0 je naprosto přirozené a zcela jistě i nejkorektnější řešení. Jílec se tedy vždy umístí na stejné souřadnice.

6.4 Vytváření čepel detailně

Následuje popis principů vytváření jednotlivých prvků čepel (matematická podstata, algoritmy). Jejich realizace ve zdrojovém kódu se může v určitých případech mírně odchylovat (především z důvodu optimalizace, přehlednosti kódu nebo způsobu práce s API Blenderu), nicméně tyto úpravy jsou z hlediska vysvětlení problematiky nepodstatné, neboť základní myšlenky a metody řešení se nemění.

Na tomto místě je nutné připomenout, že veškeré transformace se díky metodě *vytažení z profilu* provádějí ve 2D (tedy v rovině) a také že orientace vytvářeného objektu v prostoru je následující: délka čepel leží v ose y , šířka v ose z a tloušťka v ose x . Vše je dobře patrné na obrázku 6.2.

6.4.1 Základní rozměry

Požadované délky čepele se dosáhne jednoduchým vkládáním profilových řezů³ podél osy čepele (které odpovídá osa y). Hustotu vkládání těchto řezů ovlivňuje jednak parametr nastavující jemnost modelu v ose y a jednak optimalizační algoritmus.

Správná šířka čepele se dá získat jednoduchou úpravou výchozího profilového řezu (který slouží jako obecná šablona pro vytváření konkrétních profilových řezů vkládaných do modelu). Výchozí profilový řez se vždy vytváří normalizovaný na velikost 1 ve směru šířky čepele (tedy v ose y). pro získání šířky konkrétního profilového řezu tedy stačí pouze vynásobit všechny z -ové souřadnice vrcholů vkládaného profilového řezu šířkou čepele.⁴

Přestože tloušťka čepele může (a měla by) být považována za jeden ze základních rozměrů, bude popsána v následující podkapitole věnující se profilům čepele (tloušťka je totiž jedním z faktorů které určují tvar profilu meče a v modelu se projevuje právě prostřednictvím tvaru profilového řezu).

6.4.2 Profily

Získání výchozího profilu je nutné pro vytvoření i té nejzákladnější čepele. Realizovaný generátor musí umět všechny čtyři základní typy (které byly popsány výše). v budoucnu lze samozřejmě funkčnost skriptu doplnit o další, méně obvyklé typy profilů, např. trojhranný. Nyní bude podrobně vysvětleno, jakým způsobem se základní typy profilů dají získat.

Čočkovitý

Jak napovídá samotný název, tento profil svým tvarem připomíná čočku (přesněji - z hlediska optiky - bikonvexní spojnou čočku). Při popisu jeho vytváření tedy může být s výhodou použito terminologie optických čoček: zakřivení čočkovitého profilu popíšeme pomocí poloměru křivosti r . Jak jej spočítat bude uvedeno vzápětí.

Tloušťka čočky je zároveň tloušťkou čepele (a tedy i rozměrem profilového řezu v ose x). pro další výpočty je však vhodné uvažovat pouze polovinu tloušťky - označuje ji písmeno D .

Výška čočkovitého profilu při generování (budiž označena písmenem L) musí být normalizovaná, tudíž je rovna 1. Výška L čočkovitého profilu je vlastně délka tětivy kružnice o poloměru r s počátkem v ohnisku čočky. Tětiva je totožná s vertikální osou čočky, jak je dobře patrné na obrázku 6.3.

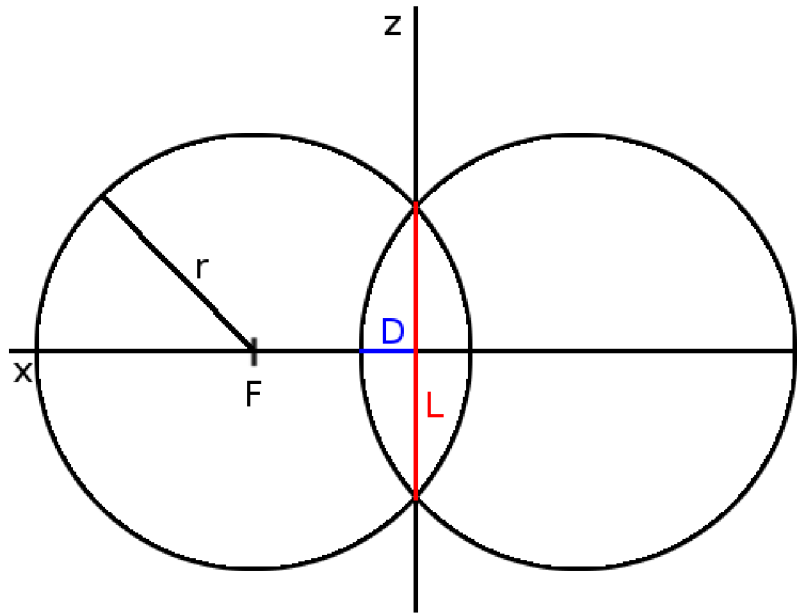
Poloměr kružnice (a tedy výše zmiňovaný poloměr křivosti r) se dá spočítat ze vzorce 6.2.

$$r = \frac{\frac{L^2}{2} + D^2}{2D} \quad (6.2)$$

Souřadnice vrcholů v ose z se generují se zvoleným krokem (tak aby bylo možné nastavit jemnost modelu v ose z). Protože šířka čepele (tzn. rozměr v ose y) je normalizovaná, generují se z -ové souřadnice v intervalu $z = -0,5$ až $z = 0,5$. pro každý krok se vypočte hodnota x -ové souřadnice (vzhledem souměrnosti podle osy z budou dvě - kladná a záporná) s pomocí vztahu 6.4 odvozeného z obecné rovnice kružnice.

³blíže vysvětlení tohoto a dalších termínů je možné nalézt ve slovníčku pojmů v příloze

⁴Toto je jeden z příkladů, kdy obecný popis přesně neodpovídá implementaci - o tom byla řeč v úvodu kapitoly. Šířkou čepele se neroznásobuje každý vkládaný profilový řez, ale přímo výchozí profilový řez. Místo opakovaného násobení při každém vkládání nového řezu tak stačí pouze jediné provedení operace na začátku, což má pozitivní vliv na časovou náročnost vytváření čepele.



Obrázek 6.3: Náčrtek vytváření čočkovitého profilového řezu.

Obecná rovnice kružnice (upravená pro rovinu danou osami x a z):

$$(x - x_0)^2 + (z - z_0)^2 = r^2 \quad (6.3)$$

Odtud vztah pro **výpočet x -ové souřadnice generovaného vrcholu**:

$$x = r_2 - 2(z - z_0) + x_0 \quad (6.4)$$

kde x_0 a z_0 jsou souřadnice středu dané kružnice: z_0 bude nulové (protože střed kružnice se nachází vždy na ose z); x_0 bude mít dvě hodnoty: $\pm(r-D)$, tudíž dostáváme pro jednu hodnotu z -ové souřadnice vždy dvě hodnoty x -ové.

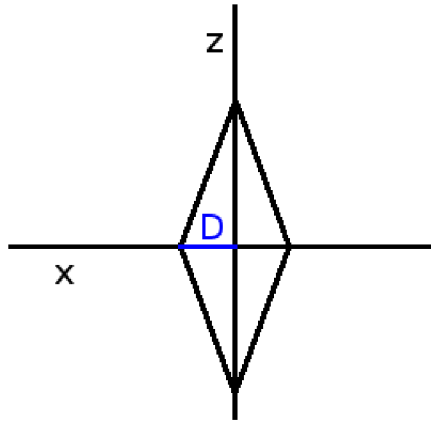
Pro vygenerování vrcholů výchozího profilového řezu tedy stačí projít ve smyčce všechny souřadnice vrcholů v z -ové ose a spočítat pro ně správné souřadnice v ose x . Hodnota y -ových souřadnic je pro výchozí profilový řez vždy nulová a zadává se až v okamžiku, kdy se z něj vytváří konkrétní profilový řez (jehož pozice na ose y je známá).

Diamantový

Princip vytvoření diamantového výchozího profilového řezu je stejný, jako v případě čočkovitého (tedy iterace přes z -ové hodnoty z intervalu $[-0,5; 0,5]$ s daným krokem a výpočet příslušných x -ových souřadnic). Jediný rozdíl je ve způsobu výpočtu páru souřadnic v ose x , neboť se jedná o (geometricky) jiný tvar (viz. náčrtek 6.4).

Jak je patrné z obrázků, stěny diamantového profilu jsou rovné. Podstata výpočtu x -ových souřadnic proto vychází z předpisu lineární funkce $y = ax$, kde sklon úsečky v grafu funkce závisí na parametru a .

Vztah daný výše uvedenou lineární funkcí musí být ovšem převeden do roviny dané osami xz , neboť i zde platí požadavek na normalizovanou šířku čepele. Zároveň se graf funkce v bodě $z = 0$ zalamuje na opačnou stranu (tzn. změní se znaménko první derivace),



Obrázek 6.4: Vytváření diamantového profilového řezu.

což je také nutné vzít v úvahu. Kladné hodnoty x -ových souřadnic lze tedy určit pomocí vztahu:

$$x_1 = D(0,5 - |z|), \quad (6.5)$$

kde D značí (stejně jako u čochovitého profilu) polovinu tloušťky čepele, x_1 je x -ová souřadnice vrcholu v kladné části osy x a z představuje z -ovou souřadnici daného vrcholu.

Souřadnice v záporné části osy x jsou osově souměrné podle osy z , pro získání x_2 stačí tedy vynásobit x_1 číslem -1 .

Vydutý

Vydutý profil je v podstatě diamantový profil se stěnami "prohnutými" dovnitř. Princip tedy zůstává, místo lineární funkce pro výpočet x -ových souřadnic je ovšem nutné najít vhodnou alternativu. Kýženeho efektu, kdy x -ové souřadnice rostou⁵ tím rychleji, čím blíže bodu 0 v ose z se nacházejí, lze dosáhnout např. jejich vynásobením hodnotou funkce 6.6.

$$1 - 2|z|. \quad (6.6)$$

Vztah pro výpočet souřadnic výchozího profilového řezu vydutého tvaru je tudíž pouhou úpravou předpisu pro výpočet diamantového profilu a vypadá následovně:

$$x_1 = D(0,5 - |z|)(1 - 2|z|), \quad (6.7)$$

kde D značí polovinu tloušťky čepele, x_1 je x -ová souřadnice vrcholu v kladné části osy x a z představuje z -ovou souřadnici daného vrcholu.

Souřadnice v záporné části osy x jsou osově souměrné podle osy z , pro získání x_2 stačí tedy vynásobit x_1 číslem -1 .

Hexagonální

Při vytváření hexagonálního profilu se dá vyjít z metody pro diamantový profil - stačí zploštit jeho strany. Výsledný algoritmus je tudíž rychlý a přímočarý, navíc staví na již připravených základech, což ulehčuje implementaci.

⁵Uvažujeme pouze x -ové souřadnice bodů v kladné části osy x .

Pro generování výchozího profilového řezu se použije vztah 6.5, pouze se dosadí dvojnásobná hodnota D . Všechny hodnoty x -ových souřadnic, které jsou větší než polovina tloušťky čepel, se "oříznou" na tuto maximální hodnotu (tím je zaručeno dodržení tloušťky čepel podle vstupního parametru). Výsledkem je požadovaný hexagonální tvar výchozího profilového řezu.

6.4.3 Drážky

Drážka v průřezu čepelí má zpravidla zaoblený tvar. Deformaci profilového řezu vlivem drážky je tedy nejvhodnější simulovat pomocí kružnice. To platí pro všechny - v současnosti do generátoru zahrnuté - typy drážek.

Při vypočtení hraniční oblast drážky se vychází z rovnice deformační kružnice, a všechny x -ové hodnoty, které padnou dovnitř deformační kružnice (tzn. budou tuto hranici přesahovat), se patřičně upraví (tedy posunou blíže k počátku osy x).

S ohledem na možnost podélného zužování čepel je navíc nutné ohlídat, aby poměr šířky drážky a šířky aktuálně deformovaného profilového řezu byl pro každý profilový řez stejný (tedy aby se šířka drážky podél osy y měnila se šířkou čepel).

Jednoduchá drážka

Jednoduchá drážka představuje nejjednodušší případ. Deformující kružnice má střed na ose x a šířka drážky odpovídá poloměru této kružnice⁶. Poloměr deformační kružnice může být teoreticky neomezený (pro extrémní případ nekonečně velkého r by z matematického hlediska byla deformačním útvarem přímka), v praxi ovšem nemají vysoké hodnoty příliš velký smysl.

Protože jednoduché drážky se u reálných exemplářů objevují v drtivé většině případů po obou stranách čepel, je i zde využito principu symetrie a výpočet provedený pro kladnou část osy x se neopakuje pro zápornou část, ale "zrcadlí" se na druhou stranu generované čepel.

Vstupní parametry deformační funkce jsou tloušťka čepel (z důvodu konzistence s předchozím výkladem budiž tloušťka čepel rovna $2D$), šířka (r) a hloubka (h) drážky. Situaci ukazuje obrázek 6.5.

Vzdálenost R_x středu deformující kružnice na ose x od jejího počátku je dána vztahem

$$R_x = r + D - h. \quad (6.8)$$

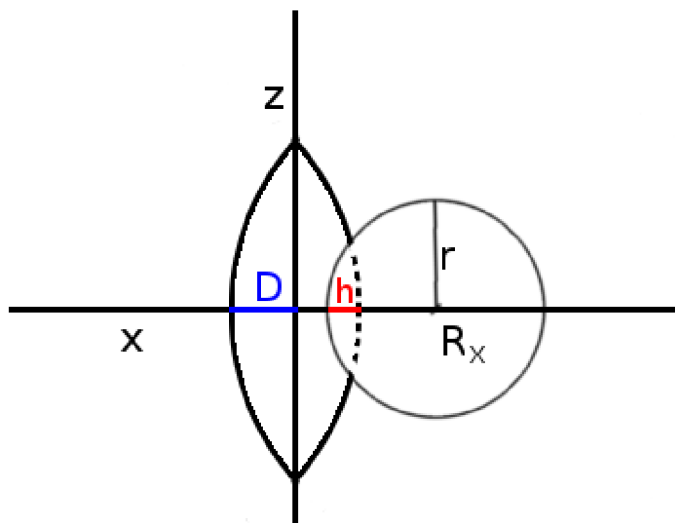
Případný posun x -ové souřadnice závisí na z -ové souřadnici uvažovaného vrcholu, poloměru r deformační kružnice a pozici R_x středu deformační kružnice na ose x . s pomocí Pythagorovy věty tedy lze odvodit vztah pro výpočet nové hodnoty x -ové souřadnice:

$$x'_1 = R_x - \sqrt{r^2 - z^2}, \quad (6.9)$$

přičemž takto vypočtená hodnota se aplikuje pouze pro vrcholy, jejichž původní x -ová souřadnice je větší, než tato nově vypočtená hodnota.⁷

⁶Není to jediná možnost, avšak během testování se tato volba z estetického hlediska osvědčila nejlépe.

⁷Při implementaci je navíc nutné dávat pozor na definiční obor funkce 6.9.



Obrázek 6.5: Deformující kružnice a vytvoření jednoduché drážky.

Dvojitá drážka

Dvojitá drážka může být vytvořena podobně jako jednoduchá. Hlavní rozdíl je v použití dvou deformačních kružnic na každé straně čepele. Poloměr r každé z deformačních kružnic tedy bude roven polovině šířky drážky.

Poloha R_x středu deformačních kružnic v ose x se spočítá stejně jako v případě jednoduché drážky podle vztahu 6.8. v ose z je pak střed posunut o r směrem nahoru či dolů tak, že se deformační kružnice dotýkají na ose x .

Výpočet posunutí x -ové souřadnice pak probíhá stejným způsobem jako u jednoduché drážky, pouze se podle kvadrantu souřadné soustavy, ve kterém se aktuální bod nachází, zvolí správná deformační kružnice.

Protilehlé drážky

Pro výpočet protilehlých drážek se dá použít stejný algoritmus, jako v případě dvojitě drážky. Jedinou úpravou je ignorování deformačních kružnic v 1. a 3. kvadrantu souřadného systému⁸.

Z estetického hlediska vypadá lépe, když jsou protilehlé drážky posunuté poněkud více od středu čepele. Z -ovou souřadnici středů deformačních funkcí je možné např. zdvojnásobit, aby bylo tohoto efektu dosaženo co nejjednodušším způsobem.

6.4.4 Podélné a profilové zúžení

Pro podélné i profilové zúžení je možné použít jednu a tu samou deformační funkci. Dále tedy bude řeč pouze o podélném zúžení, avšak ty samé principy platí i pro zúžení profilové. Základní čepel, na kterou ještě nebylo aplikováno podélné zúžení, má po celé délce konstantní šířku. Tuto šířku lze tedy pro průběh čepele podél osy y vyjádřit konstantní funkcí.

⁸Stejně dobře je ovšem možné ignorovat 2. a 4. kvadrant - je to pouze otázka volby.

Je-li potřeba dosáhnout postupného zužování čepele (tedy "zařídít", aby průběh funkce vyjadřující šířku čepele podél osy y nebyl konstantní, ale klesající⁹ - lineárně či jinak), stačí tuto funkci vynásobit jinou vhodnou funkcí.

Lineární zúžení

Nejjednodušší typ zúžení se dá (podobně jako v případě vytváření diamantového profilu) vypočítat pomocí lineární funkce. Její hodnota musí být pro y -ovou pozici na začátku čepele (tzn. u jílce) rovna 1 a pro místo u hrotu, kde čepel končí, nabývat hodnoty odpovídající míře zúžení.

Protože míra zúžení je relativní parametr, obor hodnot lineární funkce počítající zúžení pro konkrétní body na ose y se musí pohybovat mezi 0 a 1. Parametr *míra zúžení*, jehož hodnoty se v tomto rozmezí pohybují, je naneštěstí přesně opačný, než by bylo potřeba.¹⁰ Proto musí být jeho hodnoty "invertovány", což se dá snadno zařídít jeho odečtením od čísla 1.

Na základě výše uvedených úvah je možné odvodit následující vztah pro výpočet nové šířky čepele v konkrétním bodě y a tedy i šířku profilového řezu v tomto místě:

$$z' = z_0 \left(-y \frac{1-m}{l} + 1 \right), \quad (6.10)$$

kde z' je nová hodnota z -ové souřadnice (tedy nová šířka), z_0 je původní hodnota souřadnice (před zúžením), y je poloha profilového řezu na ose y , m je míra zúžení zadaná jako vstupní parametr a l je délka čepele (také zadaná jako vstupní parametr).

Konvexní zúžení

Konvexního zúžení čepele lze dosáhnout stejnou metodou jako lineárního zúžení. Klíčem je převedení lineárního průběhu funkce počítající zúžení v bodě y na průběh nelineární, odpovídající charakteru konvexního zúžení.

Tento postup je podobný postupu vytváření vydutého profilu z diamantového. Upravený předpis 6.10 tedy může vypadat například takto:

$$z' = b \cdot z_0 \left(-y \frac{1-m}{l} + 1 \right), \quad (6.11)$$

$$b = 1 - \frac{y}{l} (1-m)^2, \quad (6.12)$$

kde z' je nová hodnota z -ové souřadnice (tedy nová šířka), z_0 je původní hodnota souřadnice (před zúžením), y je poloha profilového řezu na ose y , m je míra zúžení zadaná jako vstupní parametr a l je délka čepele (také zadaná jako vstupní parametr).

Parametr b je modifikátor zajišťující právě onen požadovaný nelineární průběh, kdy zužování čepele probíhá mnohem prudčeji v místech blíže jílcí. Tím pádem závisí b na poloze na ose y - jak je ostatně z rovnice 6.12 dobře patrné.

Je samozřejmě možné použít i jiný tvar b , tento konkrétní byl zvolen na základě testování, neboť dával nejuspokojivější estetické výsledky.

⁹Uvažuje se případ souřadnic v kladné části osy z , druhá strana čepele, pro kterou by funkce musela být rostoucí, je opět osově souměrná a tím i snadno spočítatelná.

¹⁰Takto byl navržen s ohledem na uživatelskou přívětivost - počítač si hodnotu přepočítá snadněji než uživatel a proto má uživatel prioritu.

Konkávni zúžení

Konkávni zúžení lze řešit stejně jednoduše, jako předchozí případ. i zde platí základní vztah 6.11. Jediná změna oproti konvexnímu zúžení je v parametru b , jehož nový předpis je následující:

$$b = 1 + \frac{y}{l}(1 - m), \quad (6.13)$$

kde m je míra zúžení (zadaná vstupním parametrem) a l je délka čepele (také vstupní parametr).

Také v tomto případě není použitý tvar b jediný možný a výše uvedený předpis byl zvolen pro svoje "estetické" vlastnosti.

6.4.5 Ricasso

Přestože ricasso velmi často vypadá úplně jinak, než zbytek čepele, stále se v podstatě jedná o deformovaný výchozí profilový řez. Může mít různé tvary, nejčastější je však obdélníkový a ten byl tudíž zahrnut do automatického generátoru. v budoucnu lze samozřejmě přidat ještě zaoblený obdélníkový (s hladkými hranami), kulatý či elipsovité atp.

Celý princip spočívá v posunu vrcholů profilového řezu směrem nahoru či dolů tak, aby na okrajích čepele (kde je normálně ostří) vznikly rovné plochy. Je třeba započítat i změnu šířky relativně k šířce čepele - což je specifikováno parametrem. pro dosažení kýženého efektu stačí tímto parametrem vynásobit výsledné z -ové souřadnice.

Jeden z parametrů také ovlivňuje relativní tloušťku ricassa vzhledem k tloušťce čepele. Tento parametr se aplikuje stejným způsobem, jaký byl popsán v předchozím odstavci. Jedinou změnou je to, že se parametrem násobí x -ové souřadnice (neboť tloušťka je rozměr v ose x).

Algoritmus je přímočarý a jeho podstatou je průchod přes všechny *vertexy* profilového řezu a provedení požadovaných úprav. v budoucnu by mohl být vylepšen ještě o generování plynulého přechodu mezi ricassem a čepelí.

6.4.6 Ztupení horního ostří

Ztupení horního ostří - podobně jako ricasso - zplošťuje hranu čepele. v tomto případě je však ovlivněna pouze hrana horní. Přestože tato úprava se při implementaci aplikuje již při vytvoření výchozího profilového řezu (v zápětí bude vysvětlen důvod), popis metody je v tomto dokumentu pro přehlednost a jednoduchost výkladu oddělen.

Kdyby byl použit stejný postup, jako v případě ricassa, zůstala by tato plocha rozdělena na více polygonů, což zbytečně zvyšuje složitost modelu.¹¹ Ztupení se na každý pád aplikuje po celé délce meče, je tedy provedení tohoto výpočtu pouze jednou (na začátku) optimalizací i z hlediska časové složitosti.

Samotný algoritmus je velmi jednoduchý. do cyklu, ve kterém se vytváří vrcholy výchozího profilového řezu se vloží podmínka, která zajistí, že pokud se má ztupit horní ostří, nebudou *vertexy* které jsou příliš "vysoko" na ose z vůbec vytvořeny. Hranici pro "vypuštění" vrcholu lze snadno určit ze vstupního parametru udávajícího míru ztupení.

¹¹V případě ricassa je to ospravedlnitelné, neboť z něj vychází standardní čepel, které tento detail potřebuje.

6.4.7 Hrot

Vytváření hrotu je v mnoha ohledech pouze speciálním případem vytváření čepele. Jedná se o "sadu" profilových řezů, na které je aplikováno maximální podélné i profilové zúžení (to znamená, že na konci se spojí v jediném bodě). Také se při jeho vytváření dá bez jakýchkoliv dodatečných úprav použít již existující deformační funkce (tedy *podélné zúžení*) a pro vygenerování tvaru podle základních rozměrů hrotu stejná metoda jako u zbytku čepele.

Hrot má však ještě jednu unikátní vlastnost - *asymetrii*. Jeho špička (poslední *vertex*, zakončující meč) může být vychýlena směrem nahoru či dolů, v závislosti na parametru *asymetrie*. Tento efekt není zcela nahraditelný pomocí deformační křivky a byla pro něj tedy vytvořena speciální deformační funkce.

Velikost posunutí d (v kladném či záporném směru osy z) se dá spočítat z následujícího vztahu:

$$d = w(-0,5 + a) \frac{y}{l_p}, \quad (6.14)$$

kde w je výchozí šířka hrotu (tzn. šířka čepele v místě, kde začíná hrot)¹², a je parametr *asymetrie*, y je poloha aktuálního profilového řezu na ose y a l_p je délka hrotu (což je také vstupní parametr).

K aplikování efektu stačí pouze posunout z -ové souřadnice všech vrcholů profilového řezu o hodnotu d . Z estetického hlediska je dále vhodné kontrolovat, aby toto posunutí nevychýlilo profilový řez výš (nebo níž), než je okraj čepele (což lze zajistit vhodnou podmínkou pro maximální velikost d).

6.4.8 Zakřivení

Předem definované křivky, ze kterých si uživatel vybírá, se ukládají normalizované na jednotnou délku. Podle parametrů čepele se před aplikováním upraví tak, aby svojí délkou přesně kopírovaly čepel (v podstatě stačí vynásobit jejich y -ový rozměr délkou čepele poměřenou ku normalizované délce křivky).

Abyste zachovaly vzorové křivky, vytvoří se před jejich přidáním do modelu kopie. Tuto kopii pak uživatel může libovolně editovat a původní vzor zůstává nedotčen.

API Blenderu obsahuje metody pro aplikování modifikátorů na objekt, takže implementace tohoto přístupu nepředstavuje žádný větší problém.

6.5 Optimalizace

Přestože existují automatické skripty a další nástroje pro redukování počtu polygonů modelů v Blenderu, je vhodné, aby meče vytvářené automatickým generátorem byly co nejlépe zoptimalizované již při svém vytvoření. ve prospěch tohoto požadavku mluví mimo jiné skutečnost, že při vytváření meče je dobře známá jeho topologie a proporcionální vlastnosti vůbec, což může významně přispět k úspěšnosti optimalizace při stoprocentním zachování detailu.

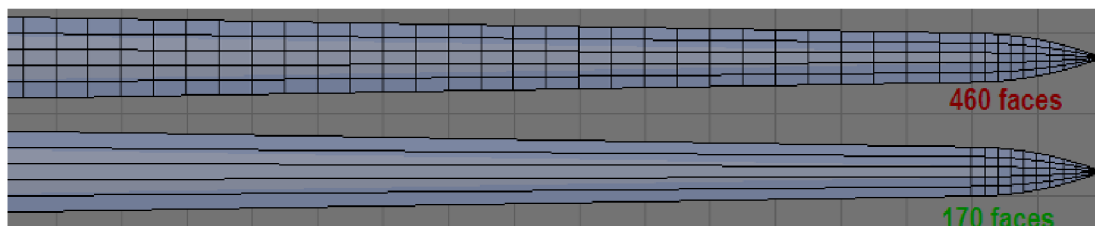
Někteří uživatelé navíc nemusí mít o této problematice dostatečné povědomí a pokud má být generátor určen i pro ně, musí se o tento problém umět "postarat" sám. v každém

¹²Pozor! Vzhledem k možnosti podélného zúžení čepele nemusí být nutně stejná jako šířka výchozího profilového řezu.

případě je to navíc známka dobrého a pečlivého zpracování, kterou jistě ocení i uživatelé zkušeni.

Metoda pro dosažení co nejmenšího počtu polygonů není založena na žádném obecném předpokladu či algoritmu. Vychází z toho, že při užití deformačních funkcí se dá určit vliv parametrů na tvar profilového řezu v daném místě podél osy y .

Protože jsou předem známé také všechny parametry, lze snadno identifikovat místa podél osy y , kde dochází ke změně tvaru čepelí a pouze do takových oblastí umístit profilové řezy. Kdyby byly řezy vkládány pravidelně se zvoleným intervalem, bylo by k dosažení stejné kvality potřeba podstatně více polygonů.



Obrázek 6.6: Optimalizace počtu polygonů - ilustrační obrázek.

Obrázek 6.5 vše demonstruje na čepeli, která je svým tvarem typická pro středověké meče. v tomto konkrétním případě má model po optimalizaci 2,7-krát méně polygonů a detail přitom zůstává zcela zachován.

Výsledek se ovšem u konkrétních čepelí může výrazně lišit. Například pokud je na meč aplikováno nějaké zakřivení, je třeba po celé délce čepelí vložit více profilových řezů, aby výsledný model nebyl "kostrbatý". Dokonce i pro rovné čepelí může být potřeba více profilových řezů - jsou nezbytné např. v místech, kde končí ricasso či na pozicích začátku a konce drážky.

Kapitola 7

Uživatelské rozhraní a realizace generátoru

Před samotnou realizací musí být vyjasněna otázka uživatelského rozhraní. Následující řádky popisují hlavní problémy řešené v této souvislosti. v druhé části kapitoly jsou diskutovány použité technologie, je zmíněno několik důležitých informací týkajících se implementace generátoru mečů a na závěr je stručně popsáno, jak probíhalo testování.

7.1 Výchozí požadavky na uživatelské rozhraní

Protože generátor je plánován jako volně šiřitelný a cílí na poměrně široký okruh uživatelů, musí být tento záměr podpořen dobrým uživatelským rozhraním.

Jazyk uživatelského rozhraní

Zdaleka největší část komunity okolo Blenderu se dorozumívá anglicky a protože není v silách autora vytvořit ke generátoru několik jazykových mutací, byla jako jazyk pro GUI zvolena právě angličtina. Totéž platí i pro uživatelský manuál a komentáře ve zdrojovém kódu.

Uspořádání velkého množství parametrů

Záměr poskytnout nástroj pro vytváření většiny typů mečů vedl k použití velkého množství parametrů. Aby byla zaručena požadovaná funkčnost, není možné se bez nich obejít. Ovšem z pohledu uživatele nemusí být široká paleta možností vždy výhodou. Vede totiž zpravidla ke složitému způsobu ovládání, který je třeba předem nastudovat, a veškerá práce s aplikací je zdlouhavá.

Proti sobě tedy stojí dva zdánlivě neslučitelné požadavky: zahrnutí všech podstatných parametrů vs. jednoduchost rozhraní. Je jasné, že redukováním počtu parametrů se nelze dobrat uspokojivého řešení tohoto problému.

Odpovědí je dvojitý přístup k uživateli - v první řadě dostane nabídnutou "jednoduchou" cestu systémem předvoleb, jež mu umožní vytvořit vizuálně přitažlivý model meče po pár kliknutích myši. Druhá cesta, pro náročnější použití, pak spočívá ve zpřístupnění všech parametrů. Výhodou je, že nové uživatele neodradí složitost práce s generátorem a dostanou "svůj" model snadno a rychle. ti pokročilejší a zvědavější naopak mohou prozkoumávat možnosti skriptu a experimentovat s různým nastavením

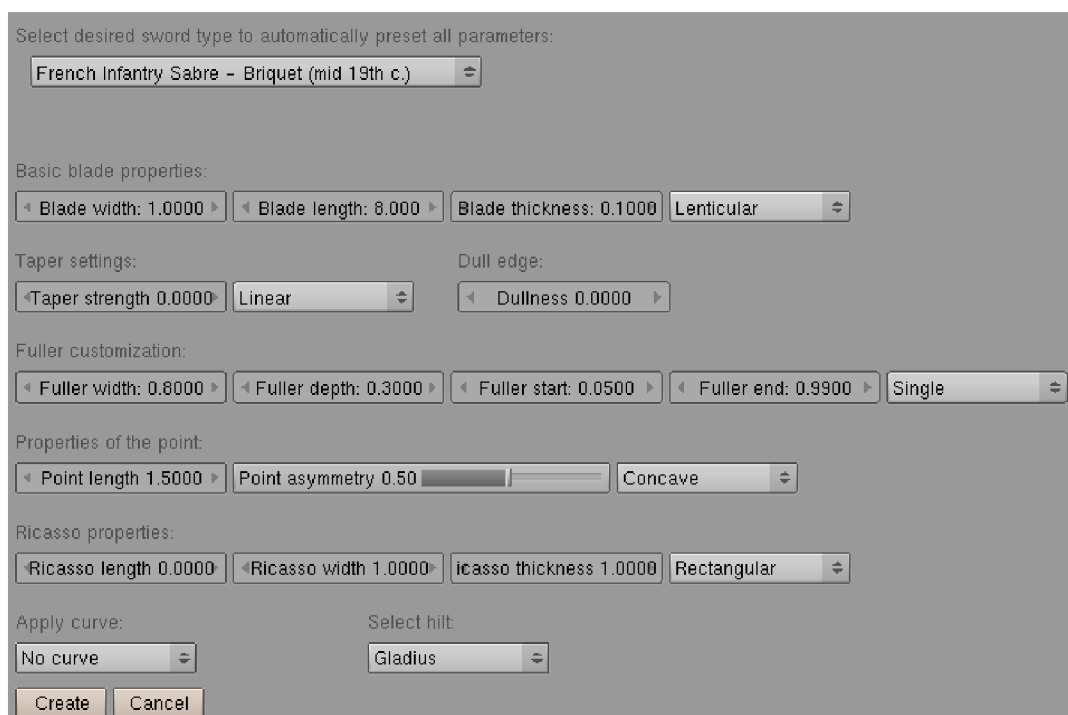
System předvoleb funguje na jednoduchém principu. Uživatel si z nabídky vybere typ meče, který by chtěl vytvořit (např. šavli, obouruční meč, gladius,...). Tím se všechny parametry nastaví na příslušné hodnoty a dále stačí jen kliknout na tlačítko a model je hotový. Pokud na výsledném meči chce uživatel provést ještě nějaké změny, může upravit požadované parametry (např. délku, drážky,...) a vytvořit jej znovu.

Způsob zadávání parametrů

Pro vstup parametrů má GUI Blenderu mnoho nástrojů. Jejich použití je intuitivní a proto nemá smysl je zde podrobně popisovat. Každý čtenář, který někdy v Blenderu pracoval, je dobře zná.

Zajímavější problematiku představuje charakter vstupních hodnot - ten je koncipován tak, aby byl z psychologického hlediska co nejbližší lidem. Mozek člověka se velmi dobře orientuje v analogiích, neustále srovnává a hledá rozdíly. Proto jsou téměř všechny parametry v generátoru vyjadřovány relativně k základním rozměrům, díky čemuž si uživatel může snadněji představit jejich vliv na výsledný model.

7.2 Výsledný vzhled



Obrázek 7.1: Uživatelské rozhraní automatického generátoru mečů - snímek obrazovky běžícího skriptu.

Výsledné rozhraní je logicky rozčleněno. Úplně nahoře se nachází menu pro výběr jednoho z přednastavených typů meče. Pod ním je přehled všech parametrů, které jsou seskupené podle příbuznosti. v dolní části jsou pak umístěna dvě tlačítka - pro vytvoření modelu a pro ukončení skriptu. Vše ukazuje obrázek 7.2.

Podrobnější popis funkčnosti GUI je možné nalézt v uživatelském manuálu (viz. přílohy).

7.3 Způsob implementace

Při implementaci automatického generátoru mečů bylo použito procedurální programovací paradigma¹. Důvodů lze uvést hned několik.

V první řadě stojí fakt, že použití procedur k aplikování deformací na profilové řezy vytváření čepele dobře modeluje matematickou podstatu problému. Úloha je navíc poměrně specifická a proto by použití OOP filozofie znamenalo především další programátorskou režii - zatímco přehlednost a organizace zdrojového kódu by se nijak významně nezlepšila.

Volbu "programovacího stylu" také nezanedbatelně ovlivnila všeobecně zažitá praxe v této oblasti - většina podobných skriptů, prostudovaných a popsáných v rámci semestrálního projektu (který diplomové práci předcházela a na nějž navazuje i tento text) byla vytvořena podobným způsobem.

Poznámka ke komentářům zdrojového kódu

Zdrojový kód je důsledně komentován, tak aby byly všechny postupy a algoritmy pokud možno pochopitelné bez jakékoliv další dokumentace. Jak již bylo zmíněno dříve, zdrojové texty včetně komentářů jsou psané v angličtině, aby byly dostupné pro co nejširší skupinu zájemců.

Na příslušných místech jsou také uvedeny poznámky a návrhy na budoucí vylepšení (uvozeny většinou klíčovým slovem *TODO*). Kde je to možné, jsou také popsány alternativní způsoby řešení či navrženy další efekty, který je možno úpravou kódu dosáhnout.

7.4 Použité nástroje a technologie

V době vytváření této diplomové práce je aktuální stabilní verzí Blender 2.49b. Ten potřebuje ke spouštění skriptů Python verze 2.6. Skript využívá základní funkce matematické knihovny Pythonu (např. odmocnina, absolutní hodnota,...) a také knihovnu *time* (pouze pro uvedení data a času u výpisu parametrů).

Co se týká API Blenderu, jsou používány knihovny pro práci s *meshem*, objekty a s modifikátory. pro vytvoření uživatelského rozhraní je knihovna *Draw*. Všechny tyto knihovny jsou zahrnuty ve standardní instalaci Blenderu (případně Pythonu) a žádné dodatečné či nestandardní knihovny ani moduly nejsou potřeba.

7.5 Přejít na další stabilní verzi Blenderu (2.6)

V současné době (rok 2010) probíhá intenzivní vývoj nové verze Blenderu. v první polovině roku byla uveřejněna verze 2.5 alpha, která naznačuje výrazné změny v uživatelském rozhraní i skriptování. Významnou inovací je např. včlenění Pythonu přímo do instalace Blenderu, v budoucnu tedy již nebude potřeba explicitně instalovat ještě správnou verzi Pythonu. Současný bouřlivý vývoj by podle [5] měl postupně vyústit v další stabilní verzi (s označením 2.6) ještě v roce 2010.

¹Python je ve své podstatě dynamický objektový jazyk, takže toto vyjádření je poněkud nepřesné, nicméně v dalším textu je snad dostatečně objasněno

Na domovských stránkách Blenderu (viz. [9]) je tomuto tématu věnováno několik článků a další informace jsou dostupné i v online dokumentaci.

Dá se očekávat, že způsob vytváření a práce s uživatelským rozhraním ze skriptu se radikálně změní. Protože tato informace byla známa již v počátcích vytváření automatického generátoru mečů, přicházely v úvahu dvě možnosti. Buďto se pustit do experimentálního vývoje přímo pro novou verzi, nebo použít poslední stabilní verzi (2.49b). Nakonec zvítězilo druhé řešení, neboť u nových, vývojových verzí hrozí nečekaná a rozsáhlá změna funkčnosti, takže by mohla nastat situace, že skript přestane ze dne na den pracovat.

Aby byl přechod na novou verzi maximálně usnadněn, byla při psaní skriptu důsledně oddělena část obsluhující uživatelské rozhraní a samotné funkce generující model meče. Protože při generování seznamu vrcholů a polygonů jsou veškeré hodnoty ukládány do standardních datových struktur Pythonu, může při přechodu na novou verzi Blenderu nastat konflikt pouze v uživatelském rozhraní, ve způsobu vytvoření *meshe* nebo při aplikaci křivky pomocí modifikátoru.

Poslední dvě oblasti by neměly představovat větší problém (samozřejmě záleží na rozsahu změn, který stále ještě není definitivně známý). Co se týká uživatelského rozhraní, bude s největší pravděpodobností nutné vytvořit jej od základu znovu, protože změny v GUI Blenderu jsou ve verzi 2.5 opravdu revoluční.²

7.6 Testování, zpětná vazba

Základní testování skriptu probíhalo průběžně s jeho implementací. Jak byly inkrementálně dodávány nové funkce, okamžitě se testovala funkčnost (tzn. zda výstup odpovídá záměru) a ladily se případné chyby.

V pozdějším stádiu bylo hlavní úsilí věnováno správnému omezení hodnot parametrů. Jak již bylo v tomto dokumentu na několika místech upozorňováno, přestože teoreticky (s ohledem na matematickou podstatu) mnoho parametrů nemusí být nijak zvlášť omezeno, v praxi je to z estetických a jiných důvodů vhodné. Kromě toho také byly na základě pokusů upravovány některé deformační funkce, aby podávaly co nejlepší vizuální výsledek.

Poté se pozornost přesunula na ověření schopností generátoru, tedy na experimentální důkaz tvrzení, že generátor zvládne většinu základních typů mečů, jak bylo deklarováno v podkapitole 5.1. Autor tedy zkoušel vytvořit různé druhy středověkých mečů, šavle, rapíry a kordy, ... Některé takové ukázky lze najít v příloze.

Předposledním krokem bylo stanovení vhodných parametrů pro jednotlivé typy mečů. k tomu dostal skript jako přídavek funkci, která vypisuje do příkazové řádky hodnoty všech parametrů použitých pro vytvoření modelu. Díky tomu se povedené pokusné modely daly snadno uložit pro pozdější zahrnutí do nabídky typů mečů.

Na závěr proběhlo krátké, neoficiální uživatelské testování. Několik dobrovolníků si skript vyzkoušelo, potvrdilo zamýšlenou funkčnost a uspokojivou formu uživatelského rozhraní. Největší "stížnosti" pak byly na malý počet předem vytvořených jílců. Uživatelé také požadovali více přednastavených typů mečů. Tato přání budou postupně splněna, přednost však zatím měla dokumentace projektu.

Nejcennější zpětnou vazbou bylo pro autora několik návrhů na další zajímavé typy mečů, které by mohly být v generátoru zahrnuty. Za zmínku stojí nejzajímavější z nich, tzv. Flamberge (v překladu "ohnivá čepel"). Jedná se o meč s vlnitou čepelí, který byl

²Podle dosavadních ohlasů na diskuzních fórech, např. [?] a v komunitě obecně jsou ale tyto změny přijímány velice kladně. Stejný názor sdílí i autor tohoto textu.

používán především v 16. století v Německu. Díky (již hotové) komplexní parametrizaci a pečlivému návrhu principu vytváření čepele bylo jeho zapracování do generátoru otázkou několika minut, neboť stačilo pouze dodat správnou křivku, pomocí které se čepel podélně deformuje.

Skutečné praktické otestování však generátor teprve čeká, protože v průběhu června bude tento generátor zveřejněn v komunitě Blenderu. Reakce uživatelů a jejich případné návrhy pak určí další směr vývoje automatického generátoru, který se bude dále rozvíjet i po odevzdání této diplomové práce.

Kapitola 8

Závěr

Splnit hlavní cíl - tedy vytvořit pro Blender automatický generátor mečů, který zvládne v přijatelné kvalitě produkovat modely většiny běžných mečů - se jistě podařilo. Důkazem toho jsou v první řadě výstupy v podobě obrázků (viz. přílohy).

Další vývoj automatického generátoru by měl směřovat především k parametrizaci a automatickému generování jílce, rozšíření palety nástrojů o atypické prvky (např. zuby, drápky, výřezy, atd.) a možná také k zavedení hierarchie přednastavených typů mečů, aby se uživateli usnadnila orientace v nich.

Oblast 3D modelování se díky této práci opět trochu obohatila: i přes důkladné hledání se autorovi nepodařilo nalézt žádný jiný automatický generátor mečů. Nyní jsou ovšem položeny základy a na nich se dá (a určitě také bude) stavět v budoucnu. Podněty a myšlenky pro další rozšíření, které byly uvedeny v tomto dokumentu, tvoří jen zlomek toho, co lze ještě zrealizovat.

Nezanedbatelným pozitivem je rozšíření možností Blenderu, který je velmi často oceňován pro svoji jednoduchost a jako takový se často používá např. k prototypování.

Samotný princip vytváření, popsáný v kapitole 6, se dobře osvědčil a některá teoretická východiska a konkrétní matematická řešení mohou být jistě s výhodou použita i v jiných aplikacích. Stejně metody vytváření čepele by například bylo možné nasadit i v případě vývoje nástroje pro průmyslový design mečů (čímž není myšleno pouze *Computer Aided Manufacturing*). Se základy položenými v této práci by se vývojáři mohli plně soustředit na řešení výrobních aspektů a fyzikálních kvalit.

V neposlední řadě je pak důkazem dobrého zpracování fakt, že funkcionalitu a rozmanitost výstupu generátoru lze poměrně snadno a rychle rozšiřovat "na přání". Tato vlastnost byla podrobně komentována v závěru kapitoly 7.

Skvělým úspěchem byla účast této diplomové práce v soutěži EEICT (bližší informace viz. [11]), kde autor skončil na 3. místě v kategorii Grafika a multimédia. Toto uznání bylo obrovskou inspirací a impulzem k další práci.

Literatura

- [1] Ebert, David S., a kol.: *Texturing & Modeling - A Procedural Approach*. Morgan Kaufman Publishers, 2003, iISBN 1-55860-848-6.
- [2] Hess, R.: *The essential Blender*. Blender Foundation, 2008, iISBN 978-1-59327-166-4.
- [3] Kršek, P.: Přednášky k předmětu Vizualizace a CAD: Repräsentace dat v CAD systémech. https://www.fit.vutbr.cz/study/courses/VIZ/private/lecture/viz_slide_cad_reprezentace_print.pdf.
- [4] Plch, P.: *Historický šerm*. Státní pedagogické nakladatelství Praha, 1981, <http://jentak.sandbox.cz/books/plch/>.
- [5] WWW stránky: Blender 2.52 release log. <http://www.blender.org/development/release-logs/blender-250/>.
- [6] WWW stránky: Blender 3D Architect. <http://www.blender3darchitect.com/>.
- [7] WWW stránky: Blender Art Gallery. <http://www.blender.org/features-gallery/gallery/art-gallery/>.
- [8] WWW stránky: Blender Artists Forum. <http://blenderartists.org/forum/index.php>.
- [9] WWW stránky: Blender Foundation homepage. <http://www.blender.org>.
- [10] WWW stránky: Blender Wiki: Doc:Manual/Textures/Maps/Bump and Normal Maps. http://wiki.blender.org/index.php/Doc:Manual/Textures/Maps/Bump_and_Normal_Maps.
- [11] WWW stránky: EEICT 2010. <http://www.feec.vutbr.cz/EEICT/>.
- [12] WWW stránky: L-System Building - Blender Artists Forum. <http://blenderartists.org/forum/showthread.php?t=47116>.
- [13] WWW stránky: L-System documentation. http://mafief.soler.free.fr/Monsite/lssystem_en_css.htm.
- [14] WWW stránky: Make Human. <http://www.makehuman.org/blog/index.php>.
- [15] WWW stránky: Making of Grapes - Lucas Veber. <http://www.lucky3d.fr/MakingOf/1.html>.
- [16] WWW stránky: Manufacturing Potatoids. http://jmsoler.free.fr/didacticiel/blender/tutor/python_script05_en.htm.

- [17] WWW stránky: myArmoury.com: Understanding Blade Properties.
http://www.myarmoury.com/feature_properties.html.
- [18] WWW stránky: Scripts Catalog - The comprehensive repository of scripts.
<http://wiki.blender.org/index.php/Extensions:Py/Scripts/Catalog>.
- [19] WWW stránky: Sword - Wikipedia, the free encyclopedia.
<http://en.wikipedia.org/wiki/Sword>.
- [20] WWW stránky: Výzbroj bojovníků v 11.-12. stol.
<http://curiavitkov.cz/valka3.html>.
- [21] WWW stránky: Wikimedia Commons: File:Rapiere-*img* 0100.jpg.
[http://en.wikipedia.org/wiki/File:Rapiere-*img*.0100.jpg](http://en.wikipedia.org/wiki/File:Rapiere-<i>img</i>.0100.jpg).

Příloha A

Obsah CD

1. Zdrojové texty tohoto dokumentu a makefile pro překlad.
2. Elektronická verze tohoto dokumentu ve formátu PDF.
3. Uživatelská příručka (anglicky)
4. Zdrojový kód skriptu generátoru
5. Soubor s kompletním generátorem (.blend formát)
6. Obrazová příloha

Příloha B

Meče - Slovníček pojmů

Zde je uveden přehled odborných termínů používaných v textu práce pro popis mečů a jejich částí:

Čepel (Blade) - Část meče určená k sekání, příp. bodání. Díly, které nejsou součástí čepele tvoří jílec.

Drápky (Finger Rings) - Kruhové pruty ležící v rovině čepele, připevněné ke kořeni quillonu; částečně či zcela se dotýkají hran čepele. Jsou určeny k ochraně prstů spočívajících na ricassu.

Drážka (Fuller) - Prohlubeň, která se táhne podél celé čepele nebo její části, na obou stranách či pouze na jedné. Slouží k odlehčení zbraně.

Falešné ostří(False Edge) - U zbraní s jedním ostřím představuje naostřenou část tupé hrany čepele (protilehlé k ostří) a nachází se typicky u hrotu.

Hlavice (Pommel) - Kulovité, často bohatě zdobené zakončení jílce. Pomáhá jej upevnit k čepeli a vyvažuje meč. Ve starší literatuře označovaná také jako "jablko".

Hrot (Point) - Termín označující ostré zakončení čepele na straně protilehlé jílcí.

Hrubí (Strong = Forte) - Spodní třetina čepele, část blíže k jílcí. Nejdolnější část zbraně.

Hřbetní prstenec (Side Ring = Port) - Část záštity určená k ochraně ruky při boji. Nachází se uprostřed křížové záštity, kolmo na čepel.

Jablko - Viz. "hlavice".

Jílec (Hilt) - Rukojeť vč. záštity, typicky složená z příčky, sloupku a hlavice.

Křížová záštita - Viz. "příčka".

Ochranný koš (Shell Guard) - Typ záštity, často kulatého či oválného tvaru. S postupem času dostal typický hemisférický tvar.

Ostří (Edge) - Nabroušená část čepele určená k sekání či řezání.

Palcový kroužek (Counter Guard) - Nachází se na opačné straně křížové záštity, než hřbetní prstenec, a je určený k ochraně palce a vnitřní strany ruky. Někdy může jít o celý systém kroužků, oblouků a prutů.

Pěstní oblouk (Knuckle Guard = Knuckle Bow) - Součást jílce, mající tvar oblouku. Vede z křížové záštity často až k hlavici.

Profil čepele (Cross-section) - Tvar řezu vedeného kolmo na směr čepele (tvar čepele při pohledu ve směru od hrotu k hlavici).

Příčka (Cross-Guard) - Základní ochranný prvek. Je součástí jílce, umístěna křížem na konci čepele.

Quillon - Rozšíření křížové záštity, k němuž došlo během procesu vývoje účinnějších ochranných prvků. Slouží také k zachycení soupeřovy čepele.

Quillion - Viz. "quillon".

Rameno (Shoulder) - Označení té části meče, kde se nachází příčka.

Ricasso - Nenaostřená část čepele blízko jílice, před quillonem, obvykle chráněná nějakou formou záštity. Dovoluje umístit prst (či prsty) ještě před quillon a tak zlepšit ovladatelnost hrotu.

Rukojeť - Viz. "sloupek".

Sloupek (Grip) - Část jílice sloužící k uchopení zbraně rukou.

Tenčí (Weak = Foible) - Přibližně horní třetina čepele, končící u hrotu.

Trn (Tang) - Výstupek, který je součástí čepele; nachází se na opačné straně než hrot. Upevňuje se na něj jílec.

Záštita (Guard) - Souhrnné označení pro části meče určené k ochraně bojovníkovy ruky.

Žlábek - Viz. "drážka".

Následující část popisuje termíny zavedené pro potřeby této práce:

Profilový řez - Skupina vrcholů (*vertexů*), která reprezentuje profil čepele v konkrétním místě čepele. Profilové řezy se při vytváření modelu vkládají na místa, kde se tvar čepele jakkoliv mění.

Výchozí profilový řez - Šablona, pro všechny profilové řezy. Je to vzorek, ze kterého se při vytváření čepele vychází. Významově je tento termín totožný s *profilem čepele* (viz. výše), ale *výchozí profilový řez* označuje konkrétní seznam *vertexů* při vytváření modelu a proto bylo vhodné jej tímto způsobem odlišit.

Příloha C

User Manual

C.1 Introduction

This python script for Blender is intended to help you with designing and creating of swords and sword-like weapons. You are only few clicks away from your own hi-poly 3D model of sword. The only thing you need to run this generator is Blender 2.49 (may work with older versions as well, however this cannot be guaranteed) and Python 2.6.

Script comes integrated in a *.blend* file and without it, complete functionality won't be available (furthermore, the script will probably fail to run as it will miss several key components). See section [C.4](#) for more information.

C.2 First steps

Load the *.blend* file and run the script (ALT+P). To create a sword, all you need to do is select a preset from the dropdown menu labeled "Select desired sword type...". Then click the "Create" button to make the sword. Script can be closed with the "Cancel" button.

C.3 Advanced settings

You will probably want to do more than just create predefined shapes. There are many parameters you can change to customize your sword. If you hover your mouse over the desired control, a hint will appear explaining its basic functionality.

Try to experiment with various settings to get a clue on how things work. To help you begin, a brief explanation of all available parameters (sorted from top to the bottom of the GUI) is listed below.

Basic blade properties

These include length, width, thickness and profile shape. For comparison with real values, 1 generator unit can be considered approximately equal to 10cm (or 4 inches). Blade profile determines, how will shape of the blade look like. For now, you can select from lenticular, diamond, hollow-ground or hexagonal shape.

Taper settings

Taper defines, how much smaller will the blade get towards the point. You can set strength (where 0 means no taper at all and 1 is maximal narrowing towards the point) with any value between 0 and 1. A shape of taper can be selected from the dropdown menu (*linear* - meaning straight, *concave* - which is more acute towards the point and *convex* - which looks kind of "hyperbolic").

Dull edge

Not every sword has sharp edges on both sides. To make upper edge dull, set the *dullness* parameter to non-zero value (the bigger the number, the flatter the edge).

Fullers

You can add fullers to the sides of the blade. Various parameters allow you to define where fuller starts and ends (relative to the blade length), how deep and wide it is and what is the type of fuller (currently single, double or opposing).

Point

Properties of the point can be set independently of the blade. You can select its length, shape (linear, convex or concave) and specify whether the point should be displaced upwards or downwards (use the slider to do this).

Ricasso

Unsharpened (and often rectangular) part of the blade, right near the hilt, is called ricasso. You can set its length, width and thickness relative to the blade properties.

Curving

To make the blade curved, simply select predefined curve from the dropdown menu labeled "Apply curve". After you create the sword, curve remains visible in the model and curving modifier is not yet applied. This allows you to manually change the blade curving to your satisfaction... or simply apply the modifier (if you like the outcome already) to make the curving permanent.

Hilt selection

Choose preferred type of the hilt from the dropdown menu labeled "Select hilt". There will be more choices in the future.

C.4 Using the source code, extending functionality

Unfortunately, there is no developer's handbook yet. But the source code is richly commented, so you shouldn't have problem using or modifying it.

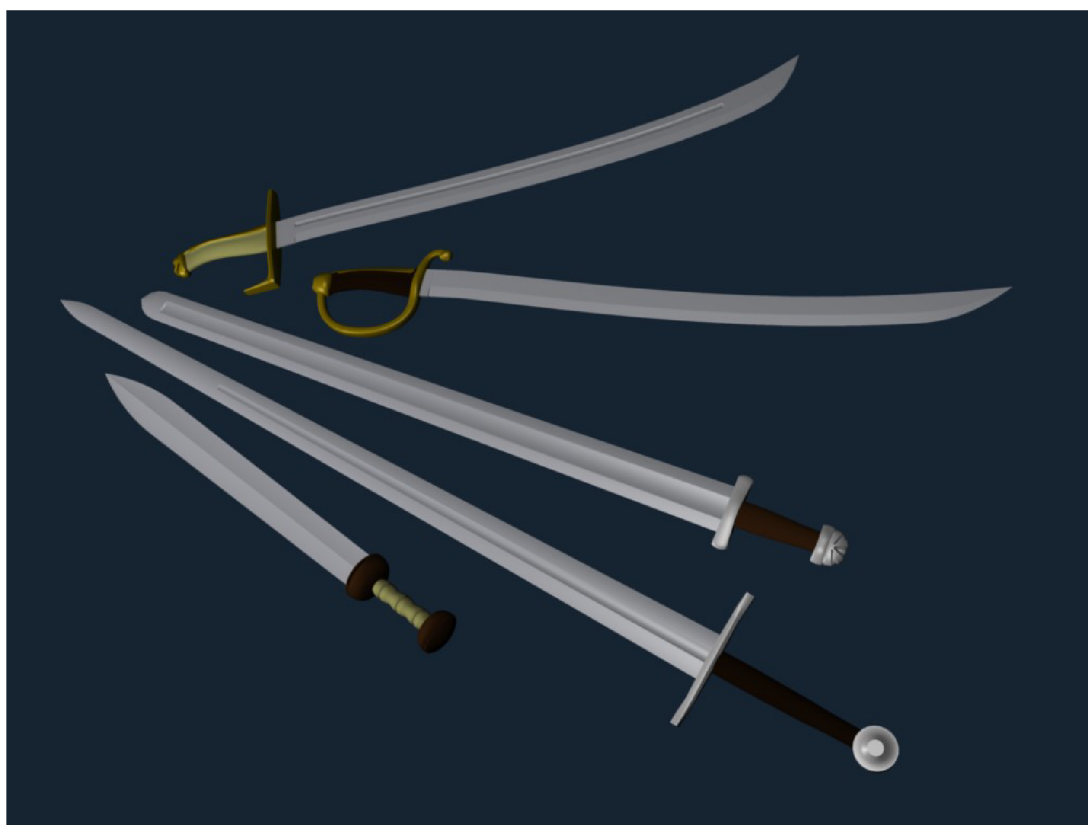
To add new model of the hilt or a curve, add the object to the *.blend* file. You will also need to add the new item to the GUI components and update section of the code, where it

gets applied to the model. The ability to add new models will be automatized in the future, but for now, the only way is the hard way..

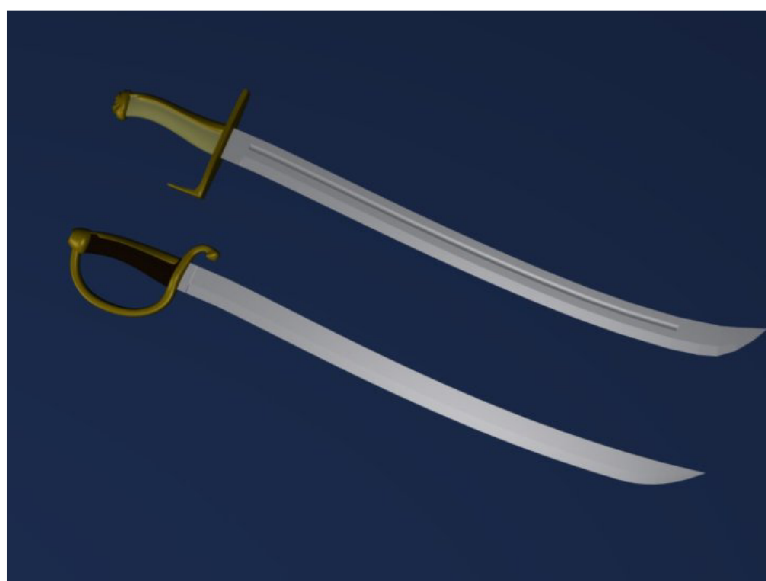
Příloha D

Ukázky renderovaného výstupu

Následující obrázky jsou rendery mečů vytvořených s pomocí generátoru. Jedná se tedy o skutečný výstup z aplikace, nikoliv ruční modelování.



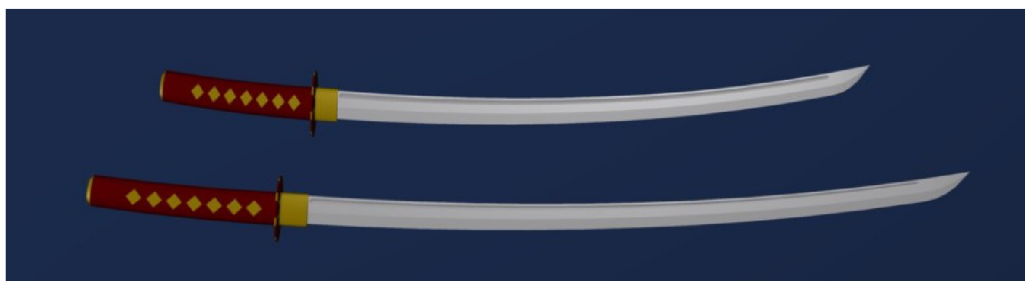
Obrázek D.1: Různé typy mečů.



Obrázek D.2: Shora: Variace jezdecké šavle (žádná konkrétní předloha); Francouzská pěchotní šavle (Briquet), pol. 19. století



Obrázek D.3: Shora: Variace jezdecké šavle (žádná konkrétní předloha); Francouzská pěchotní šavle (Briquet), pol. 19. století



Obrázek D.4: Shora: Wakizaši (žádná konkrétní předloha); Katana (žádná konkrétní předloha) - typy mečů používané v Japonsku od 15. století.



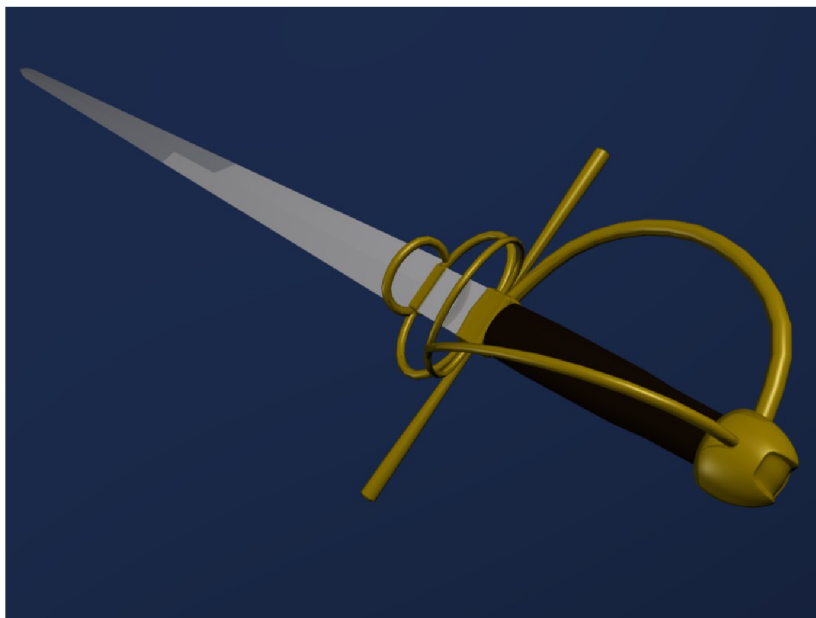
Obrázek D.5: Středověký německý válečný meč, Oakshott typ XIIa.



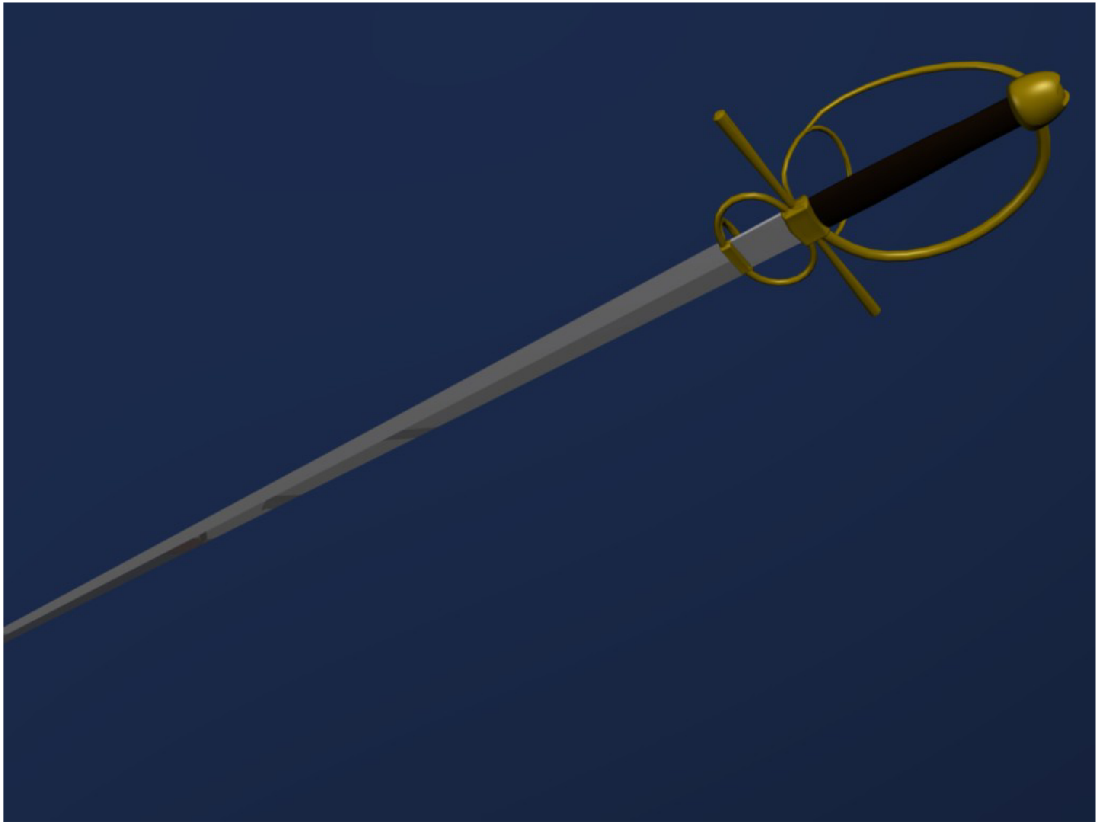
Obrázek D.6: Starověký Gladius (krátký meč).



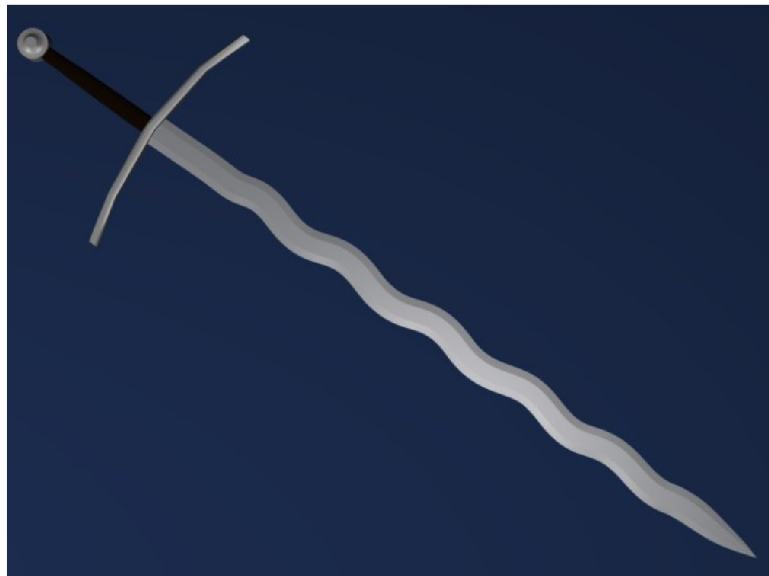
Obrázek D.7: Milánský rapír, Itálie, konec 16. století.



Obrázek D.8: Milánský rapír, Itálie, konec 16. století.



Obrázek D.9: Milánský rapír, Itálie, konec 16. století.



Obrázek D.10: Flamberge, užívaný převážně během 16. století.