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

SKICÁŘ 3D MODELŮ

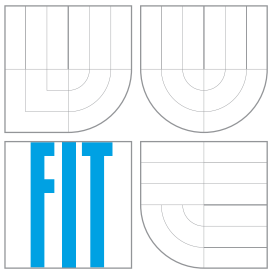
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ DAVID

BRNO 2012

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

SKICÁŘ 3D MODELŮ

SKETCHING 3D MODELS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — TOMÁŠ DAVID
AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. PŘEMYSL KRŠEK, Ph.D.
SUPERVISOR

BRNO 2012

Abstrakt

Tato práce se zabývá problematikou zpracování a reprezentace polygonálních modelů. Věnuje se metodám tvorby a úpravy 3D modelů, především pak metodě skicování. Popisuje postup při návrhu a implementaci jednoduchého zásuvného modulu do aplikace OpenFlipper. Tento modul umožňuje vytváření 3D modelu objektu pomocí nakreslení uzavřené čáry.

Abstract

This thesis deals with the processing and representation of polygonal models. It is devoted to methods for creating and editing 3D models, especially the sketching method. It describes the procedure for design and implementation of a simple plug-in for application OpenFlipper. This plug-in allows creation of 3D model of object by drawing a closed line.

Klíčová slova

skicování, 3D modelování, polygonální model, trojúhelníková síť, OpenMesh, OpenFlipper, C++

Keywords

sketching, 3D modeling, polygonal model, triangle mesh, OpenMesh, OpenFlipper, C++

Citace

Tomáš David: Skicář 3D modelů, bakalářská práce, Brno, FIT VUT v Brně, 2012

Skicář 3D modelů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Ing. Přemysla Krška, Ph.D.

.....
Tomáš David
11. května 2012

Poděkování

Na tomto místě bych rád poděkoval vedoucímu mé práce doc. Ing. Přemyslu Krškovi, Ph.D. za cenné rady, připomínky a za čas, který mi věnoval. Díky patří také mé přítelkyni za gramatickou korekci tohoto dokumentu. V neposlední řadě děkuji všem za jejich podporu a pomoc při studiu.

Obsah

Kapitola 1

Úvod

Pod pojmem skicář nebo také skicák si člověk v dnešní době asi stále ještě představí papírový blok, jehož význam by se dal popsat jako umělcova kniha pro uchovávání svých náčrtků a konceptů. Pokud se ke slovu skicář přidá ještě doplněk, o který rozměr v prostoru se jedná, většinu lidí již zřejmě napadne, že se bude jednat o jakýsi počítačový program zabývající se malováním obrázků, jinak řečeno kreslením počítačové grafiky.

Klasické skicování je proces, při kterém se z čar stávají složitější obrazce a tyto obrazce jsou dále vylepšovány, až se z nich stávají detailní návrhy určitých přestav a myšlenek kreslíře. V třídimenzionálním prostoru se jedná o tentýž proces, avšak s určitými odlišnostmi.

Třídimenzionální skicování není zdaleka tak rozšířené, jako je tomu u dvoudimenzionálního skicování. Možností jak vytvářet 3D model je velké množství a dalo by se říci, že tento obor ještě není ani zdaleka probádán. Třídimenzionální skicování je v současné době ve výzkumné fázi, stále se ještě vyvíjí a uplatnění této metody se zatím omezuje jen na oblast počítačových her.

Tato práce se zabývá skicováním využívající uzavřené a otevřené 2D čáry, které vytváří výsledný 3D model. Dále se věnuje se návrhu a implementaci jednoduchého nástroje pro skicování. Nejedná se ani tak o nástroj pro detailní zpracování modelu, ale spíše o aplikaci, která vystihuje model pomocí základních tvarů a rysů.

Tento dokument je rozdělen do několika částí. Po úvodní kapitole následuje druhá část, která se věnuje rozboru technologií pro tvorbu 3D modelů a technologií pro jejich reprezentaci. V třetí kapitole bude popsán návrh jednotlivých metod, postupů a aplikace jako celku dle sepsaných požadavků . Dále následuje kapitola implementace, v níž budou postupně a detailně vysvětleny jednotlivé části implementovaného programu a jeho testování. Pátá část představuje výsledky této práce a příklady modelů, které byly aplikací vytvořeny. Nakonec závěrečná kapitola shrnuje celou práci, její celkový přínos a pojednává o možnostech budoucího vývoje.

Kapitola 2

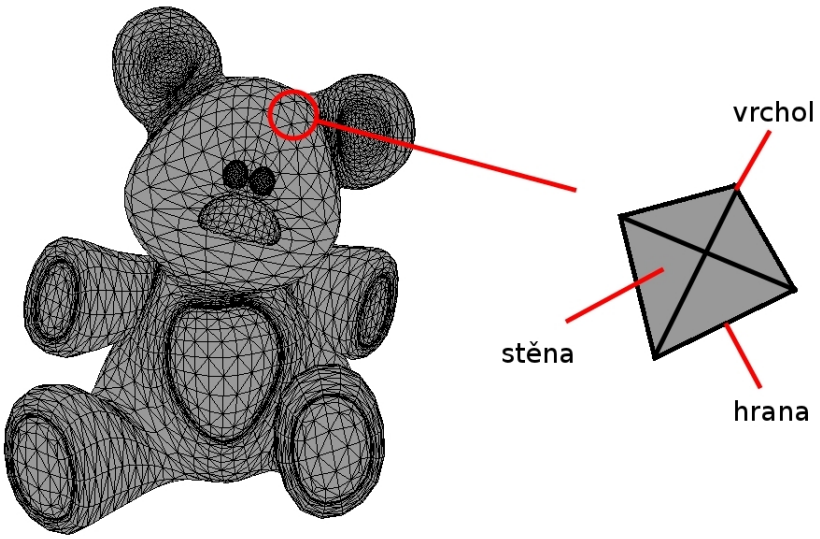
Rozbor

Tato kapitola se zabývá současnými způsoby reprezentace a vytváření polygonálních modelů. Dále jsou zde popsány nástroje, které se zabývají problematikou vytváření a úpravy těchto modelů.

2.1 Reprezentace polygonálních modelů

Polygonální model je tvořen polygonální sítí, která se skládá ze stěn, hran a vrcholů. Tato síť reprezentuje povrch modelu. Jak lze vidět na obrázku 2.1 i malý model může obsahovat velké množství zmíněných prvků a jejich rozložení může být na kterémkoliv místě velmi husté nebo i také velice řídké. Od tohoto se také odvíjí množství potřebného místa pro uchování dat modelu, přičemž platí, že čím je povrch modelu detailnější, tím je ho potřeba více.

K uložení dat popisujících polygonální síť modelu objektu je možné použít několik typů datových struktur. Mezi ty nejpoužívanější lze zařadit datové struktury, jako jsou např. okřídlená hrana nebo půl hrana.



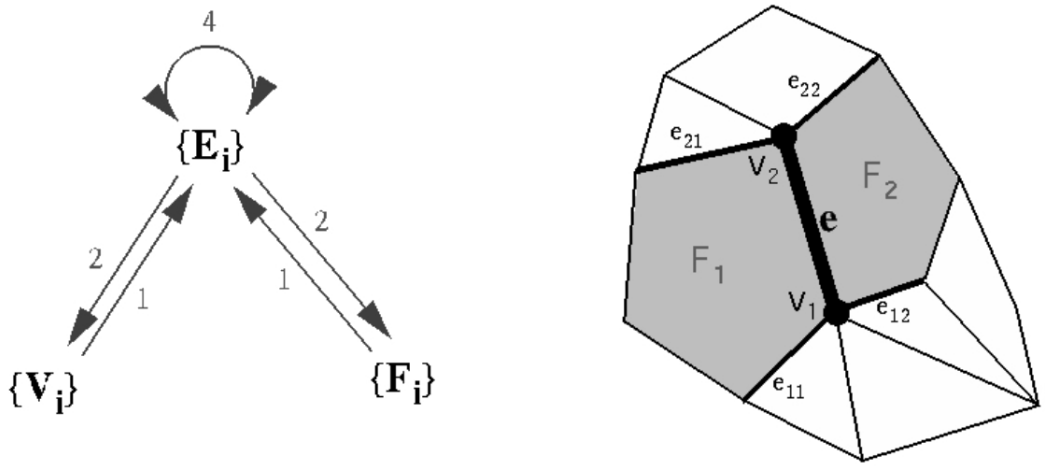
Obrázek 2.1: Polygonální model

2.1.1 Okřídlená hrana

Okřídlená hrana [15] je nejpoužívanější datová struktura pro popis polygonální sítě. Tuto reprezentaci navrhl Bruce G. Baumgart a její pojmenování souvisí s grafickým znázorněním jedné hrany a jejích sousedících prvků, které svým vzhledem připomínají křídla.

Těleso popsané touto strukturou se skládá ze tří lineárních seznamů. Jedná se o seznam hran, vrcholů a stěn. Jak lze vidět na obrázku 2.2, pro okřídlenou hranu obecně platí, že:

- každá hrana E_i obsahuje ukazatele na dva své vrcholy V_i, na dvě sousedící stěny F_i a na čtyři sousedící hrany E_i, které vycházejí z jejích vrcholů V_i
- každý vrchol V_i obsahuje ukazatel na jednu z vycházejících hran E_i
- každá stěna F_i obsahuje jeden ukazatel na jednu její hranu E_i



Obrázek 2.2: Okřídlená hrana [3]

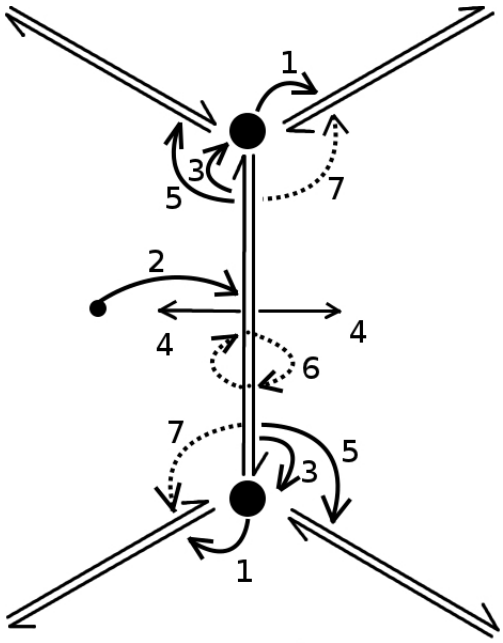
2.1.2 Půl hrana

Půl hrana [3, 14] je datová struktura, která je velmi podobná okřídlené hraně. Její výhodou je především rychlá navigace po síti pomocí rychlých iterací a cirkulací. Na druhou stranu však vyvstává nevýhoda v podobě paměťové náročnosti, která je vyšší než v případě jiných datových struktur.

Tato datová struktura obsahuje (čísla v závorkách u jednotlivých popisků označují očíslované prvky na obrázku 2.3):

- pro každý vrchol ukazatel na jednu vycházející půl hranu z tohoto vrcholu (1)
- pro každou stěnu ukazatel na jednu z půl hran, které ji ohraničují (2)
- pro každou půl hranu ukazatel na:
  - vrchol, ke kterému směřuje (3)

– stranu, které náleží (4)
– další půl hranu (řazeno proti směru hodinových ručiček) ležící uvnitř stěny (5)
– opačnou půl hranu (6)
– (volitelně) předchozí půl hranu ležící uvnitř stěny (7)



Obrázek 2.3: Půl hrana [14]

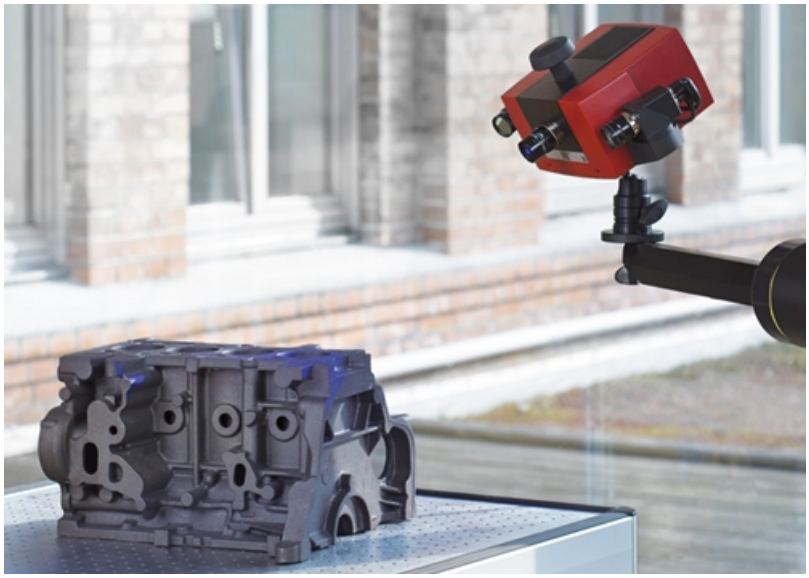
2.2 Vytváření a úprava 3D polygonálních modelů

Vytváření a úprava 3D polygonálních modelů se dá označit jedním slovem a to modelování. Modelování [10] je proces, při kterém je vytvářen a upravován model. Tento model může být vytvářen několika způsoby. Zde budou popsány pouze ty nejznámější metody (skenování reálných objektů, subdivision modelování) a metoda související s touto prací (skicování).

2.2.1 Skenování reálných objektů

Tato technika vytváření 3D modelů [2] spočívá ve skenování reálných objektů speciální technikou (kamery, skenery). Na obrázku 2.4 je možné vidět jeden z takovýchto nástrojů. Naskenovaný objekt je (v rámci viditelnosti) popsán množinou bodů, které jsou umístěny v 3D prostoru. Dále je vytvořena polygonální síť, která je uložena pro další zpracování probíhající zpravidla v profesionálních modelovacích programech. Kvalita a přesnost vytvořených polygonálních modelů závisí na použité technice a složitosti struktury povrchu objektu.

Skenování objektů však vyžaduje existující předlohu, proto není možné použít tuto techniku k vytváření polygonálních modelů dle vlastních představ. Metoda se spíše používá např. pro archivaci tvarů reálných objektů jako jsou sochy a jiná umělecká díla.

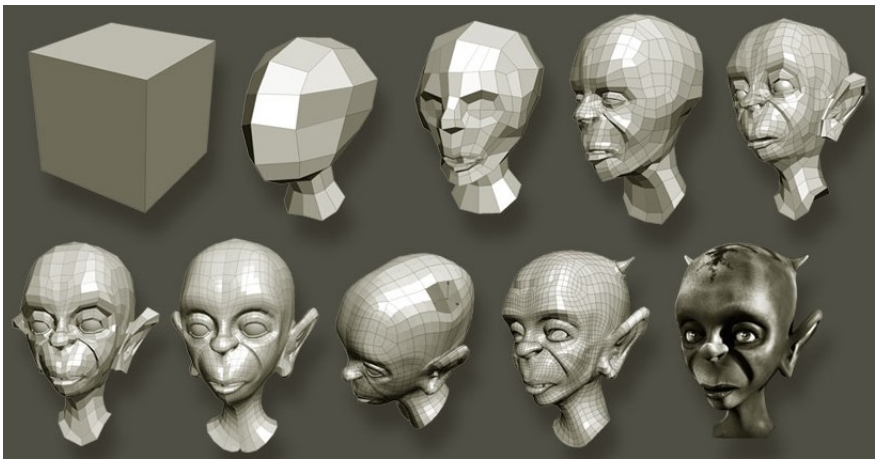


Obrázek 2.4: Skenování objektu za účelem získání polygonálního modelu

2.2.2 Subdivision modelování

Nejpoužívanější a nejznámější technika vytváření a úpravy 3D modelů. Tato technika by se dala přirovnat v reálném světě k sochařství.

Většina modelování, jak je možno vidět na obrázku 2.5, začíná s geometrickým primitivem (krychle, kvádr, koule). Počáteční objekt je rozdělen na mnoho polygonů. Jednotlivé polygony, body, hrany či jejich skupiny jsou vytahovány do prostoru, odstraňovány a otáčeny, nebo jsou s těmito prvky prováděny další operace tak, aby objekt dosáhl co možná nejpřesněji zamýšlené podoby.



Obrázek 2.5: Modelování metodou subdivision [7]

Existuje i mnoho dalších postupů jak vytvářet 3D model. Mezi další bychom mohli zařadit např. techniku, při které se nakreslí půlka tvaru objektu, otočením kolem osy se

objekt vytáhne do prostoru a vznikne polygonální model, nebo je nakreslen celý tvar objektu (např. pomocí fotografie) a vytažením do prostoru ve směru jedné z os získáme základní tvar. Následným subdivision modelováním je pak model upravován do finální podoby.

Tyto techniky pro vytváření 3D modelů jsou často využívány v jedné z nejznámějších komerčních aplikací 3D studio max (podsekce 2.3.1). Z open-source programů bychom mohli pak uvést např. Blender (podsekce 2.3.2).

2.2.3 Skicování

Skicování [6, 9] je oproti dvěma předchozím metodám poměrně odlišná technika vytváření 3D modelů. Jejím cílem není detailní zpracování modelů, ale spíše se tato technika zaměřuje na jejich rychlou a jednoduchou úpravu. Jednoduchým kreslením, za pomoci myši či pera, se vytváří objekt, který se upravuje kreslením dalších čar na jeho povrchu či mimo něj.

Nepoužívají se zde složité postupy a přesné zadávání rozměrů, ale pouze jednoduché klikání či přepínání kreslících módů. Hlavním cílem skicování je přiblížit 3D modelování širšímu spektru uživatelů a umožnit jim jednoduché a rychlé vytváření svých vlastních 3D modelů.

Skicování (obr. 2.6) v trojrozměrném prostoru se velice podobá tomu v dvojrozměrném. Tahy jsou vynášeny nepřesně a pouze s kreslířovou představivostí. Model je postupně upravován bez ohledu na detaily a vytváří se hrubý náčrt zamýšleného objektu. Ve výsledku je pak např. použit jako předloha pro pozdější detailní vymodelování v profesionálním programu pro úpravu 3D grafiky.

Jeden z prvních programů, který se zabývá skicováním 3D modelů, se nazývá Teddy. Tento program obsahuje pouze sadu základních funkcí, jako je vytvoření nového objektu, vytažení, useknutí, kreslení na model a vyhlazování, které se postupně objevily i v dalších aplikacích zabývající se skicováním. Tento program bude detailněji popsán v podsekci 2.3.3.

V návaznosti na aplikaci Teddy byl vytvořen Smooth Teddy (podsekce 2.3.4). Smooth Teddy, jak již je zřejmé podle názvu, lépe pracuje s vyhlazováním modelů a dále také přidává funkci pro vybarvení modelu.

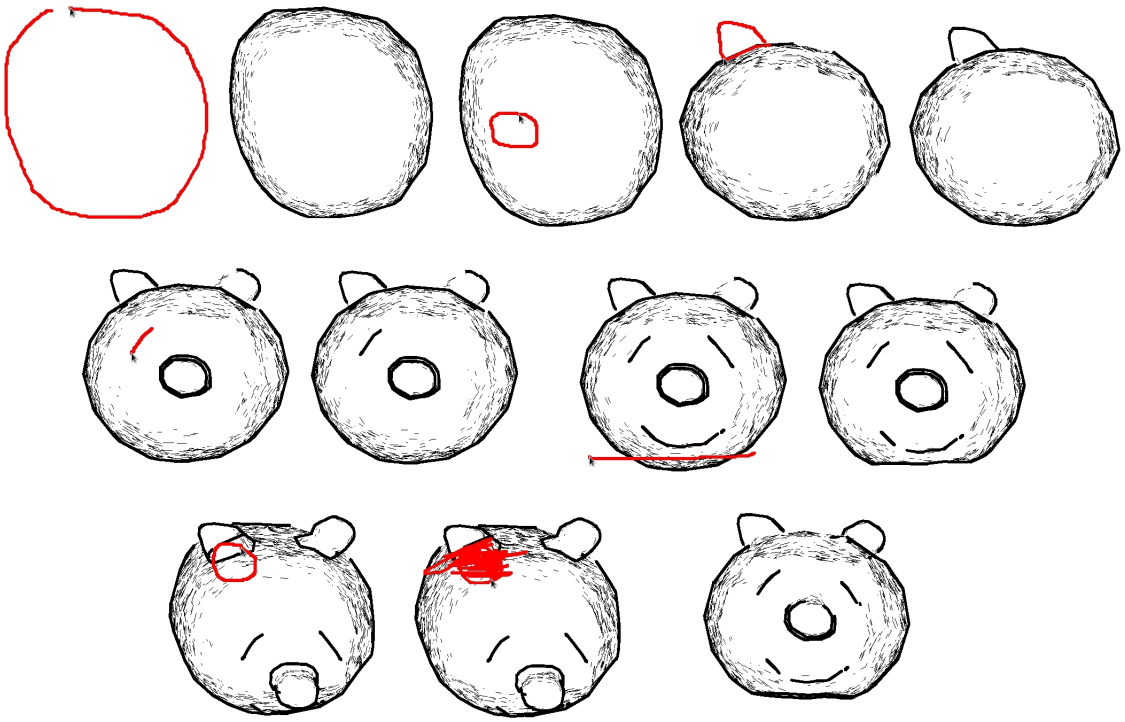
Nejnovější aplikací, která pokračuje v linii programů Teddy a Smooth Teddy je FiberMesh (podsekce 2.3.5). Tento program přináší řadu vylepšení, především více pokročilé funkce a metody zpracování modelu.

Nyní budou jednotlivě popsány již zmíněné základní operace skicování. Ke každé operaci bude také popsána implementace algoritmu, který byl použit v aplikaci Teddy.

Vytvoření nového objektu

První a nejpoužívanější operací při skicování je vytvoření nového objektu. Uživatel začne volným uzavřeným tahem. Počáteční a koncový bod je propojen a z oblasti uzavřené čárou je vytvořen 3D objekt. Pokud se však segmenty čáry někde protínají, je vyžadován nový tah. Nakreslená oblast je vždy vytažena do prostoru tak, že šířka v jednotlivých částech udává i hloubku vytažení.

Algoritmus této operace, použitý v aplikaci Teddy, začíná s vytvořením rovinného polygonu podle nakreslené čáry, která je navzorkována. Pomocí Delaunayho triangulace je oblast rozdělena na trojúhelníky a na koncích větví jsou vytvořeny trojúhelníkové vějíře. Dále jsou pospojovány středy vnitřních hran a tím vzniká hřbet, který je vytažen do prostoru. Poté jsou trojúhelníky rozděleny na menší části tak, aby vytvářely zaoblený tvar. Nakonec je celá tato trojúhelníková síť zkopírována na opačnou stranu a tím vzniká nový uzavřený objekt.



Obrázek 2.6: Příklad postupu skicování v aplikaci Teddy v pořadí zleva shora: Vytvoření nového objektu(2 kroky), vytažení (3 kroky), kreslení na objekt(2 kroky), useknutí(2 kroky), vyhlazení (2 kroky) a finální objekt (poslední krok)

Kreslení na objekt

Operace kreslení na objekt pouze vynáší čáry na povrch objektu. Používá se pro dokreslení detailů, které by bylo zbytečné složitě modelovat. Nakreslená čára musí ležet celá na povrchu objektu a její začátek a konec nesmí být spojen. Pokud je nakreslená čára začmárána, tah je odstraněn.

Implementace této operace, která byla použita v aplikaci Teddy, spočívá ve vyslání paprsků ze snímající kamery přes segmenty nakreslené čáry na model objektu. Místa, kde se paprsky protnuly s objektem, jsou pospojována a vzniká čára, která leží na povrchu modelu.

Useknutí

Pokud některý z tahů (kromě prvního) začíná a končí mimo objekt, bude se jednat o operaci useknutí. Nakreslená čára pak rozděluje objekt na dvě části. A jedna z nich je odstraněna.

Algoritmus useknutí aplikace Teddy je podobný tomu, který je použit při kreslení na objekt. Opět jsou vyslány paprsky přes segmenty čáry, které protínají objekt. Poté jsou v místech protnutí vytvořeny nové vrcholy a část objektu ležící nalevo ve směru tahu je odstraněna. Nakonec je díra, která vznikla useknutím, triangulována.

Vytažení

Při kreslení uzavřených čar na objektu je použita operace vytažení. Pro tuto operaci je zapotřebí dvou tahů. První uzavřený tah označí, která část objektu bude vytažena. Poté uživatel může natočit objekt a nastavit tím směr vytažení. Druhým tahem se provede vytažení v požadovaném směru a požadované velikosti.

Algoritmus pro vytažení aplikace Teddy nejprve trianguluje druhý tah a tím vytvoří rovinné polygony. Poté se pokračuje od objektu a vytváří se síť kopírováním prvního tahu podél těchto polygonů. Takto vzniká uzavřená oblast až k vrcholu výstupku. Nakonec jsou rovinné polygony odstraněny a nová část je připojena k objektu.

Vyhlazení

Poslední operací je vyhlazení. Pokud je část objektu označena jako při vytažení a tato část je začmárána, v této oblasti dojde k vyhlazení povrchu modelu.

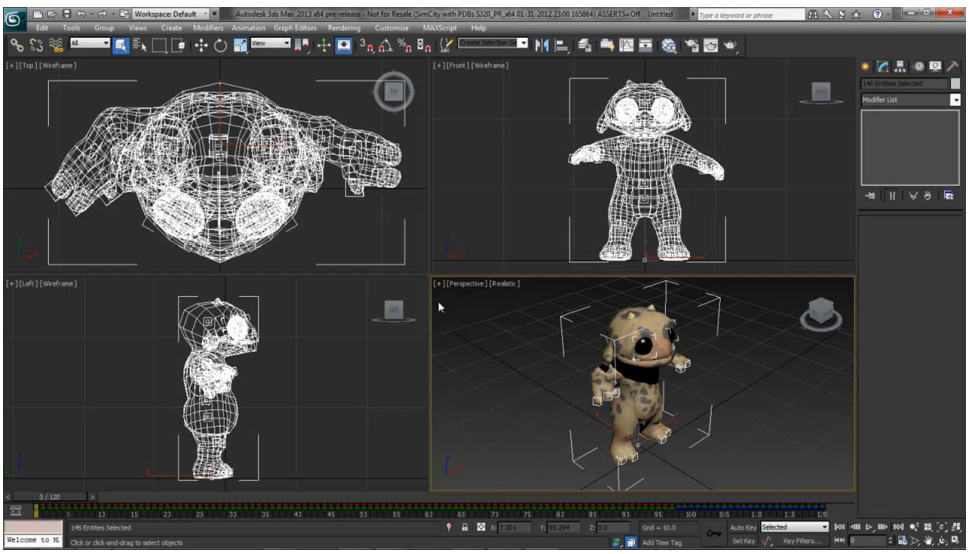
V aplikaci Teddy jsou nejprve v označené oblasti odstraněny všechny trojúhelníky. Poté jsou body vyvýšeny a pospojovány Delaunayho triangulací tak, aby vznikla nová hladší plocha.

2.3 Nástroje pro vytváření 3D modelů

V této sekci bude představeno několik aplikací pro vytváření 3D modelů. Takovýchto nástrojů existuje celá řada. Zde budou uvedeny pouze ty nejznámější a dále aplikace, které souvisí s problematikou skicování.

2.3.1 Autodesk 3D studio Max

Zřejmě jeden z nejznámějších profesionálních programů pro práci s 3D grafikou. Na obrázku 2.7 je zobrazeno uživatelské rozhraní této aplikace.

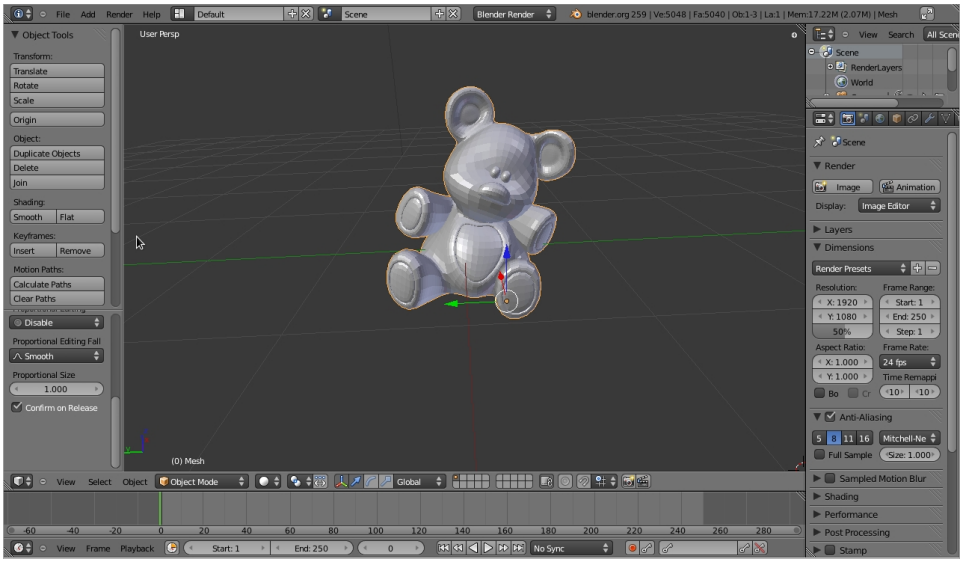


Obrázek 2.7: Uživatelské rozhraní aplikace Autodesk 3D studio Max

Tento software obsahuje nástroje pro modelování, animace, renderování a spousty dalších. Již o počátku vzniku této aplikace je Autodesk 3D studio Max [12] vyvíjen pouze pro platformu Windows. Díky své škále funkcí má vyšší požadavky na hardware počítače, na kterém je tato aplikace spuštěna. Autodesk 3D studio max se používá v mnoha odvětvích. Mezi tyto odvětví patří např. návrh architektury staveb, vytváření speciálních efektů ve filmovém odvětví, herní průmysl a další.

2.3.2 Blender

Blender [11] je profesionální multiplatformní software pro 3D modelování, texturování, osvětlení, animace a video post-processing. Rozhraní je založeno na známé grafické knihovně OpenGL. Je vyvíjen jako open-source aplikace a díky tomuto poskytuje širokou možnost rozšiřitelnosti. Jako ucelená aplikace má oproti jiným modelovacím softwarům malou velikost. Mezi uživateli má širokou podporu a existuje celá řada návodů a pomocných materiálů, které jsou ať už v knižní podobě nebo volně dostupné na internetu. Vzhled této aplikace je zobrazen na obrázku 2.8.



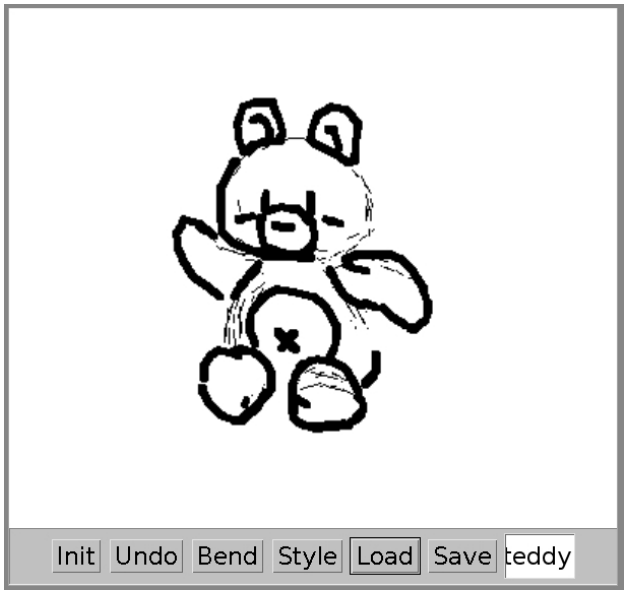
Obrázek 2.8: Uživatelské rozhraní aplikace Blender

2.3.3 Teddy

Pravděpodobně se jedná o jeden z prvních nástrojů pro skicování. Teddy [6] byl navržen pro jednoduché a rychlé vytvoření 3D modelů, např. plyšových zvířátek či jiných zaoblených objektů. Je implementován v jazyce Java. Na obrázku 2.9 je vidět, že tato aplikace obsahuje pouze několik ovládacích prvků. Většina systémových modelovacích operací (vytvoření nového objektu, vytažení, useknutí, kreslení na model a vyhlazování) je prováděna tahem myši. Uživatel vždy začíná vytvořením nového objektu a pomocí dalších operací se tento objekt už jen upravuje.

Aplikace Teddy umožnuje ukládat a nahrávat vytvořené modely ve formátu `.obj`. Díky tomu je možné s objekty dále pracovat a upravovat je i v jiných aplikacích. Další funkcí je možnost přepínání mezi pohledem zobrazující polygonální síť a pohledem zobrazující objekt

se základním stínováním. Poslední funkcí této aplikace je možnost ohnutí nakresleného modelu podle nakreslené čáry.



Obrázek 2.9: Vzhled aplikace Teddy

2.3.4 Smooth Teddy

Smooth Teddy [4] je nástupcem dvou programů: Teddy [6] a Chameleon [5]. Jak již bylo řečeno v předchozí podkapitole 2.3.3, Teddy je nástroj pro skicování. Aplikace Chameleon je nástroj pro vybarvování povrchů 3D modelů. Smooth Teddy dědí od těchto programů jejich hlavní funkce a dále je vylepšuje. Jedná se tedy o skicovací program s možností vybarvení povrchů 3D modelů (obr. 2.10).

Hlavní změny programu Smooth Teddy spočívají v úpravě polygonální sítě modelu a její vyhlazení, které se oproti předchůdci značně vylepšilo. Mezi další vylepšení patří např. možnost provádět vyhlazení, vytažení a oříznutí pouze jedním tahem, možnost umístění vytažené části či její zkopírování na druhou stranu modelu a mnoho dalších.

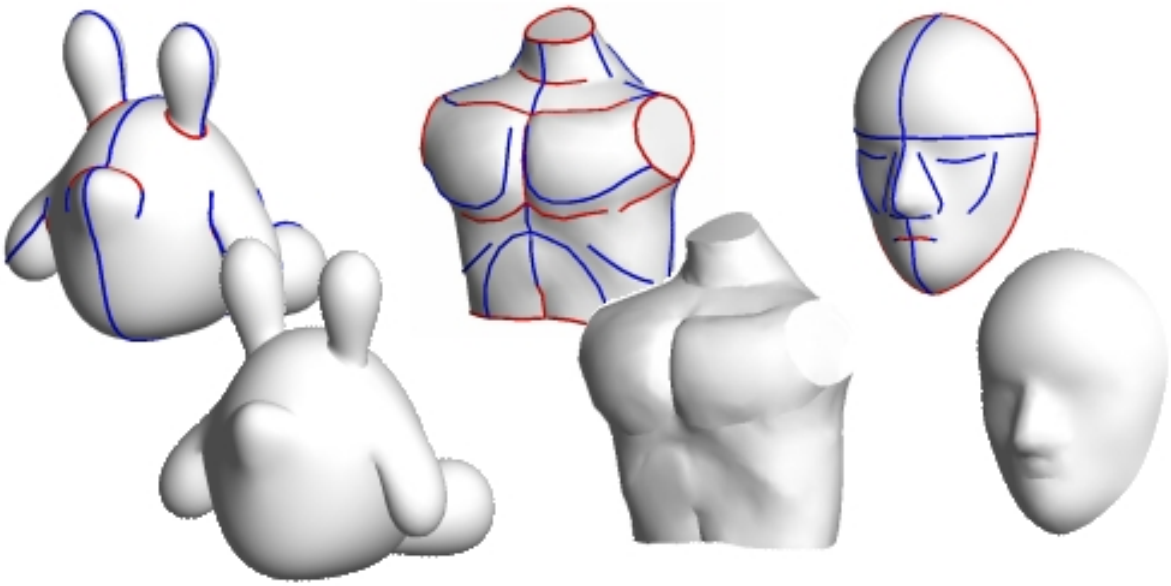
2.3.5 FiberMesh

FiberMesh [9] je pokročilý skicovací nástroj pro úpravu 3D modelů pomocí křivek. Tak jako Teddy [6] a Smooth Teddy [4] obsahuje standardní sadu nástrojů pro skicování. Kromě těchto nástrojů navíc využívá křivek, které po nakreslení zůstávají na povrchu modelu. S těmito křivkami lze i po jejich vytvoření manipulovat a tím model tvarovat a upravovat.

Nástrojů pro manipulaci s křivkami je několik. Deformační nástroj umožňuje křivku uchopit v jakémkoliv bodě, provést její posunutí a tím deformovat model. Vyhlazovací nástroj kmitavým pohybem vyhlazuje křivku a zároveň i tvar modelu. Nástroj pro mazání umožňuje křivku vymazat. Nástroj pro změnu typu mění křivku z jednoho typu na druhý.

FiberMesh obsahuje, jak lze vidět na obrázku 2.11, dva typy křivek. První typ křivky umožňuje na modelu vytvořit hladké části a naopak druhý typ vytváří části s ostrými záhyby. Tím je možné vytvořit široké spektrum tvarů objektů.

Obrázek 2.10: Vzhled aplikace Smooth Teddy [4]



Obrázek 2.11: Modely vytvořené aplikací FiberMesh [9]

Kapitola 3

Návrh

Tato kapitola pojednává o návrhu aplikace pro jednoduché skicování. Nejdříve bude popsána volba nástrojů a technologií pro implementaci, poté budou shrnuty požadavky na aplikaci a bude proveden návrh struktury aplikace. Nakonec bude popsán návrh samotných funkcí aplikace.

3.1 Volba nástrojů a technologií

Součástí každého návrhu je zvolení nástrojů a technologií, které budou později použity pro implementaci samotné aplikace. Jako všechna rozhodnutí provedená při návrhu i tato jsou velmi důležitá, protože správná rozhodnutí mohou snížit časové náklady spojené s implementací.

3.1.1 Programovací jazyk

Jako první z mnoha rozhodnutí, které bylo potřeba udělat, byla volba programovacího jazyka. Největší zkušenosti jsem měl s programováním v jazyce C/C++, ale lákala mě i volba programování v čistě objektovém jazyce Java. S tímto jazykem jsem však neměl žádné zkušenosti a samotné seznámení s jeho syntaxí a s jeho vlastnostmi by zabralo nemalou dobu. Navíc pro jazyk Java neexistuje příliš velké množství nástrojů pro práci s polygonálními sítěmi a geometrickými daty. Nakonec jsem tedy zvolil jazyk C++, jehož použití je asi ještě stále pro práci s 3D grafikou nejrozšířenější.

Jazyk C++ není čistě objektový, což se mnohdy považuje za jeho nevýhodu. V mém případě jsem však tuto vlastnost ocenil. Hlavně z důvodu, že není potřeba psát zdlouhavé zápisy, které leckdy zpomalují vývoj samotné aplikace.

3.1.2 Typ aplikace

Aplikace, které se zabývají zpracováváním 3D geometrických dat, používají jako vstupní zařízení klávesnici nebo myš. Pro výstup pak používají obrazovku počítače, jinak řečeno, jejich výstup je grafický.

V tomto případě se bylo potřeba rozhodnout zda tento projekt bude vytvořen zcela samostatně, či jako zásuvný modul pro některou již existující aplikaci.

Mezi výhody samostatné aplikace spadá např. možnost vlastního návrhu dle vlastních potřeb či fakt, že není potřeba nastudovat někdy až příliš složitou dokumentaci. Na druhou stranu však převažuje jedna z největších nevýhod – zdlouhavý návrh a implementace

samotného základu aplikace, která zabírá mnoho času a samotné řešení jádra problému se pak stále odsouvá.

Zásuvné moduly mají právě opačnou výhodu, a to takovou, že se programátor může soustředit čistě na podstatu svého řešeného problému. Mezi hlavní nevýhody lze pak zařadit některá omezení pro zásuvné moduly aplikace.

Pro samostatnou aplikaci by bylo možné použít technologii OpenGL, nástavbu Orge3D postavenou na této technologii nebo nástroj OpenSceneGraf. Jelikož jsem se však rozhodl aplikaci vytvořit jako zásuvný modul, rozhodoval jsem se mezi aplikacemi Blender, MeshLab a OpenFlipper. V nástroji pro 3D modelování Blender se však zásuvné modely programují v jazyce Python, u kterého mám jen základní zkušenosti, a proto jsem tuto volbu zavrhl. U dvou zbývajících nakonec rozhodlo to, že OpenFlipper měl lépe zpracovanou dokumentaci a navíc jeho použité technologie jsou novější než u aplikace MeshLab, a tudíž pro mě i zajímavější.

3.2 Požadavky na aplikaci

Základem každého návrhu jsou požadavky na aplikaci. Tyto parametry aplikace se během vývoje mohou změnit, ale neměly by se měnit příliš často, aby nedocházelo k velkým úpravám celého systému. Pro tuto aplikaci byly na začátku sepsány tyto požadavky:

- Aplikace se bude zabývat skicováním 3D modelů.
- Aplikace bude vytvořena jako zásuvný modul do programu OpenFlipper.
- Aplikace bude využívat knihovnu OpenMesh pro práci s geometrickými daty.
- Aplikace bude implementována v jazyce C++.
- Aplikace bude multiplatformní.
- Z nakreslené uzavřené smyčky, která sama sebe neprotíná, aplikace vytvoří 3D model.
- Model může být dále upraven vytažením nebo useknutím určité části.

3.3 Struktura Aplikace

Pro funkčnost zásuvného modulu v aplikaci OpenFlipper je potřeba si udělat určitý návrh a představu, jak spolu tyto dvě věci budou spolupracovat. Především se jedná o to, zda zásuvný modul bude automaticky spuštěn nebo se bude spouštět až po zadání příkazu či po stisknutí tlačítka. Dále si je potřeba ujasnit zda po spuštění zásuvného modulu bude uživatel moci používat funkce ostatních zásuvných modulů, nebo zda tyto funkce budou omezeny či zcela zakázány.

3.3.1 Zakomponování v aplikaci OpenFlipper

Návrh na zakomponování v aplikaci OpenFlipper byl zvolen tak, že zásuvný modul zcela spolupracuje s aplikací. Po spuštění aplikace zásuvný modul není automaticky spuštěn a je možné provádět jiné operace. Zapnutím se zpřístupní operace vytváření nového modelu. Pokud se operace vytvoření nového modelu zdařila, vzniká 3D model dle zadaného tvaru.

Na druhou stranu, pokud došlo k některé z výjimek, operace se nezdařila a čeká se na zopakování akce. Po prvotním vytvoření modelu je možné zásuvný modul vypnout a pracovat s jinými funkcemi aplikace OpenFlipper, nebo upravovat model dalšími skicovacími funkcemi (vytažení, useknutí). Ve finální fázi modelovacího procesu může být výsledný model zahozen či uložen díky vestavěné funkčnosti aplikace OpenFlipper.

3.3.2 Ovládací prvky zásuvného modulu

Ovládacích prvků aplikace OpenFlipper je velké množství. Samotný skicovací zásuvný modul bude mít pouze dva ovládací prvky.

Prvním z nich je tlačítko na zapínání a vypínání. Během modelovacího procesu je možné jej kdykoliv stisknout. Jeho funkce je však pouze v rozsahu vypnutí a zapnutí kreslení skicovací čáry, kterou se modeluje výsledný 3D model.

Druhým ovládacím prvkem je kurzor myši, kterým se kreslí skicovací čáry na obrazovku.

3.4 Funkce aplikace

Zde bude popsán návrh hlavních funkcí aplikace, jejich vstupy, výstupy a omezení. Jedná se o návrh jejich fungování, samotná implementace a přesné vlastnosti jenž byly implementovány budou uvedeny v kapitole 4.

Na obrázku 3.1 je možno vidět návrh modelovacího procesu z pohledu uživatele. Vždy se začíná nakreslením první čáry, pomocí které se vytvoří nový objekt. Poté může být objekt vždy natočen a opakovaně upravován.

3.4.1 Nakreslení čáry

Čára vzniká tahem kurzoru po zobrazovací části programu OpenFlipper. Nakreslená čára se skládá z velkého množství bodů libovolně od sebe vzdálených. Není potřeba uchovávat souřadnice všech bodů, ale pouze některých. Tohoto se dosáhne segmentací výše zmíněné čáry. Čára musí mít uzavřený tvar, a proto pro dosažení této podmínky bude poslední a první bod propojen. S tímto souvisí jedno omezení, které zakazuje, aby kterákoliv úsečka tvořená dvěma body protínala jakoukoliv jinou úsečku. Jinak řečeno, čára nesmí sama sebe protínat v jakémkoliv místě.

3.4.2 Vytvoření nového 3D modelu

Vytvoření nového 3D modelu spočívá ve vytažení uzavřené oblasti, která je tvořena čárou nakreslenou na obrazovku, do prostoru. Čára však definuje pouze šířku modelu ve dvou rozměrech. Třetí rozměr může být dopočítán pomocí těchto dvou rozměrů. Jedním z možných řešení je, že uzavřený tvar bude vytažen do prostoru jako by byl koule či jí podobný zaoblený objekt. Jinak řečeno, můžeme si přestavit uzavřený tvar vyskládaný koulemi s tím předpokladem, že tyto koule budou pospojovány tak, aby z nich vznikl jeden hladký objekt.

3.4.3 Kreslení čar na objekt

Stejně jako tomu bylo u kreslení první čáry, vznikne čára, která musí být navzorkována. Body této čáry budou posunuty tak, aby byly namapovány na povrchu objektu. Pokud některé body leží mimo objekt, mohou být zneviditelněny nebo odstraněny. Nakonec takto

Obrázek 3.1: Návrh modelovacího procesu zásuvného modulu z pohledu uživatele (červeně jsou označeny uživatelské akce, šedě pak akce prováděné aplikací)

nakreslená čára bude ležet na povrchu modelu zcela, částečně, a nebo na něm nebude ležet vůbec a v tom případě se s ní nebude pracovat.

3.4.4 Useknutí

Useknutí je dvoufázová operace. V první fázi je nakreslena čára, která začíná a končí mimo objekt. Pokud tomu tak není, nemůže se jednat o operaci useknutí. V místě, kde čára objekt protíná, bude provedeno rozdělení objektu na dvě části. Ve druhé fázi se označí, která část bude odstraněna. Toto bude provedeno kliknutím kurzoru na tu část, které se chceme zbavit. Na závěr se musí zacelit díra, která vznikla po useknutí.

3.4.5 Vytažení

Vytažení je třífázová operace, přičemž ve dvou fázích je kreslena pomocná čára.

V první fázi se na objekt nakreslí uzavřený tvar. Je zcela nezbytné, aby všechny body čáry byly na objektu, jinak tato operace nemůže být provedena. Ve druhé fázi se musí natočit objekt do takového směru, kterým chceme vytažení vést. Nakonec v poslední fázi se provede tah, který určuje jak má vytažení vypadat. Zde však platí, že tento tah musí být veden od čáry kreslené na povrchu objektu, tudíž nesmí být veden ve směru, v kterém by protínal aktuální objekt. Po třech úspěšných krocích je (dle jejich parametrů) povrch objektu v místě akce vytažen.

Kapitola 4

Implementace

Následující kapitola představuje implementační část této práce. Je rozdělena na čtyři sekce, přičemž v první sekci budou popsány použité nástroje a technologie. Druhá část se zabývá implementací, která se odehrává v dvojrozměrném prostoru a v třetí části bude popsána implementace, která souvisí s třídimenzionálním prostorem. Nakonec v poslední části této kapitoly bude zmíněno testování aplikace.

4.1 Použité nástroje a technologie

V této části jsou popsány použité nástroje. Jedná se o aplikaci OpenFlipper a knihovnu OpenMesh, která je tímto programem využívána.

4.1.1 OpenFlipper

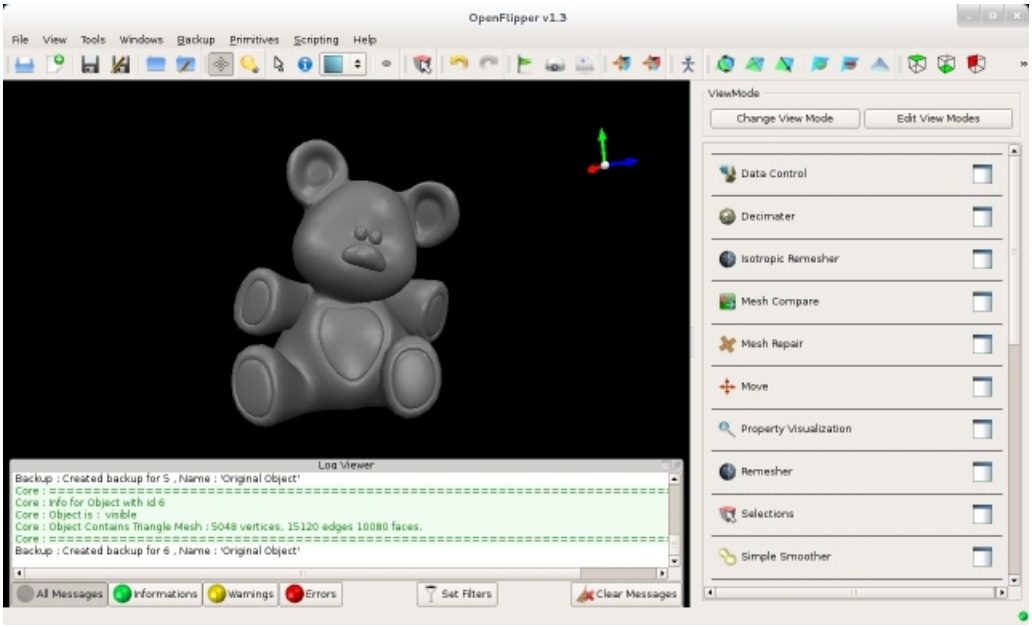
OpenFlipper [13] je open-source multiplatformní aplikace a programovací framework pro zpracování, úpravu a renderování geometrických dat. OpenFlipper je v současné verzi podporován operačními systémy Windows, Linux a MacOS X. OpenFlipper je založen na knihovně Qt, a proto umožňuje libovolně implementovat algoritmy a rozšíření dle požadavků programátora. Standardně obsahuje již několik základních zásuvných modulů pro menší operace s geometrickými daty, například zásuvný modul pro vyhlazování, pro optimalizaci sítě modelu a další. Pro uložení geometrických dat využívá knihovnu OpenMesh, a právě díky této knihovně je možné nahrávat a ukládat 3D modely do mnoha typů souborů.

Uživatelské rozhraní (obr. 4.1) této aplikace se skládá z několika částí. První z nich je zobrazovací část, takzvaný viewport, ve kterém jsou vykreslovány geometrická data. Viewport umožňuje přepnout do několika zobrazovacích módů (např. zobrazení bodů, polygonů, půl hran a další). Dále je možné nastavit ortogonální či perspektivní projekci. Zobrazovaný model lze pohybem myši libovolně natáčet, přibližovat a oddalovat. Velice užitečnou funkcí může být i změna barvy pozadí např. pro případnou prezentaci na projektoru či tisk. Také je možné vytvořit snímek zobrazovací oblasti.

Druhou částí uživatelského rozhraní je ovládací prvek pro zobrazení nahraných zásuvných modulů. Každý zásuvný modul zde může mít ovládací prvky pro manipulaci s geometrickými daty či ovládací prvky pro jejich úpravu.

Další částí je prvek pro výpis záznamů. Mezi záznamy, které zde programátor může vypisovat, patří chybová hlášení, varování a informační zprávy. Tento nástroj také umožňuje nastavení filtrů zpráv.

Posledním ovládacím prvkem je panel nástrojů obsahující ikony pro rychlý přístup k základním funkcím a funkcím zásuvných modulů.



Obrázek 4.1: Vzhled aplikace OpenFlipper

4.1.2 OpenMesh

OpenMesh [14] je knihovna pro reprezentaci a manipulaci s polygonálními sítěmi. Knihovna umožňuje pracovat jak s trojúhelníkovými sítěmi, tak i se sítěmi čtyřúhelníkovými a typově se řadí mezi datové struktury využívající půl hrany.

OpenMesh umožňuje rychlé procházení a iterování mezi vrcholy, hranami, půl hranami či stěnami modelu. Stejně jako bylo popsáno v podsekci 2.1.2, tato knihovna obsahuje nástroje pro cirkulace kolem vrcholu, stěny či půl hrany, díky kterým můžeme zjistit sousední elementy sítě. Na obrázku 2.3 lze vidět, jak jsou jednotlivé elementy sítě propojeny.

Mezi další nástroje a funkce patří například kontrola zda prvek sítě leží na okraji sítě, možnost obohacení každého prvku sítě o vlastní atribut či možnost výběru vhodného typu pro reprezentaci skalárních hodnot prvků či prostorových souřadnic.

OpenMesh je implementován v programovacím jazyce C++ a je vyvíjen již od roku 2002.

4.2 Práce v 2D prostoru

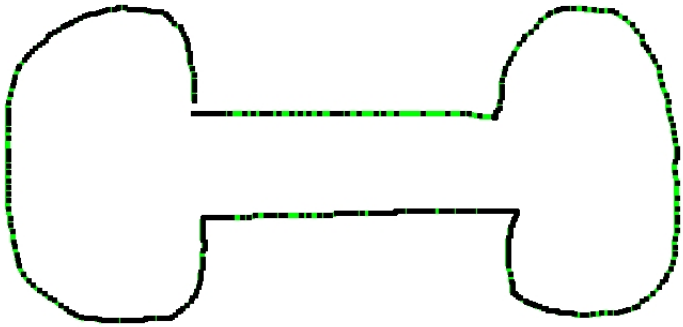
V této sekci se s veškerými prováděnými operacemi pracuje pouze na úrovni dvou rozměrů, tj. na osách x a y. Na osu z nemusí být prozatím brán zřetel, protože se s touto osou nepracuje. Následující operace se provádějí pouze na ploše obrazovky v zobrazovací části okna aplikace OpenFlipper.

4.2.1 Kreslení čáry

Prvním krokem skicování je nakreslení čáry. Čára je kreslena na zobrazovací část aplikace OpenFlipper. Může být libovolně velká, může se libovolně zatáčet, ohýbat se a měnit směr. Nakreslená čára má svůj počátek a svůj konec. Po dokončení tahu jsou tyto dva hraničící body spojeny, aby bylo docíleno uzavření oblasti, kterou tato čára obklopuje. Kreslení čáry má však také jedno omezení, a to že čára nesmí sama sebe v jakémkoli místě protínat. To znamená, že se nesmí střetnout s již nakreslenou částí. Toto omezení platí i při spojování začátku a konce.

Tato oblast bude v dalších sekcích tvorby udávat tvar výsledného objektu. Je zcela nezbytné, aby byl tah správně uzavřen, protože pokud tomu tak nebude, program nemůže vytvořit objekt kvůli nedostatečným či chybným datům. Pokud je však tvar objektu nakreslen správně, může být vytvořeno velké množství objektů, které svým vzezřením budou připomínat reálné předměty.

K nakreslení čáry je využita knihovna ACG, která je součásti aplikace OpenFlipper, a která poskytuje třídu pro kreslení čar `LineNode`. Z této třídy vytvoříme objekt, kterému jsou nastaveny parametry tloušťky a barvy. Tloušťka je zvolena takovým způsobem, aby byla čára dobře viditelná, stejně jako její barva, která by měla být dobře rozeznatelná jak na standardní černé barvě pozadí, tak i na často používané barvě bílé. Ideální je některá z RGB barev. V tomto případě byla použita barva zelená (obr. 4.2).



Obrázek 4.2: Nakreslená čára (zeleně) se zvýrazněnými body(černě)

4.2.2 Vzorkování čáry

Nakreslená čára je množina libovolně od sebe vzdálených bodů. Při rychlém tahu jsou body od sebe vzdáleny více, při pomalém tahu jsou body nahuštěny těsně vedle sebe (obr. 4.2). Pro vytváření trojúhelníkové sítě mezi těmito body není tato vlastnost příliš vhodná. Abychom ošetřili vzdálenosti sousedních bodů, je potřeba čáru navzorkovat pomocí konstanty, která bude udávat novou vzdálenost.

V tomto postupu se bude vytvářet navzorkovaná čára z čáry originální. Jako první operace se provede přiřazení počátečního bodu originální čáry do navzorkované čáry. V dalších krocích se vždy vypočte vzdálenost v mezi posledním bodem navzorkované čáry P a následujícím bodem N čáry originální podle vzorce:

$$v = \sqrt{(N_x - P_x)^2 + (N_y - P_y)^2}.$$

Pokud je vzdálenost v menší než vzorkovací konstanta k, bod je ignorován a sekvence se opakuje pro následující bod originální čáry. Při vzdálenosti, která je stejná jako vzorkovací konstanta se originální bod přiřadí do navzorkované čáry. V posledním případě, kdy je vzdálenost v větší, se vypočtou souřadnice nového bodu A.

Nejprve se vypočte velikost úhlu α ve směru originálního bodu podle vzorce:

$$\alpha = \cos^{-1}\left(\frac{N_x - P_x}{v}\right)$$

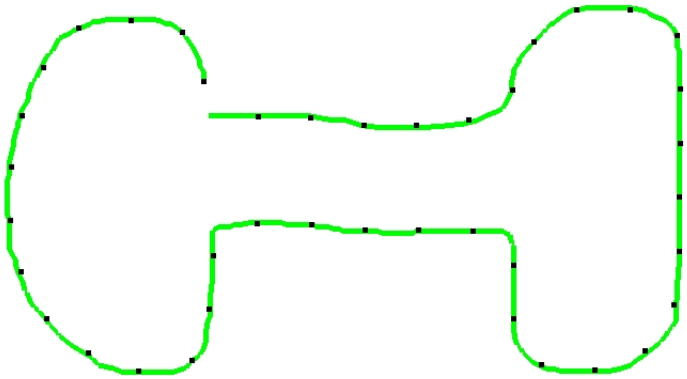
a v tomto směru vypočteme souřadnice nového bodu A, který je vzdálen o velikost vzorkovací konstanty k, podle vzorce:

$$A = \sin(\alpha) \cdot k \cdot P$$

a tento nový bod je přidán do navzorkováné čáry.

Další body jsou přidávány dokud není vzdálenost mezi posledním bodem nově navzorkované čáry a následujícím bodem originální čáry menší nebo rovna vzorkovací vzdálenosti.

Tento celý cyklus je prováděn dokud se nenarazí na poslední bod originální čáry. V poslední fázi výpočtu se nakonec navzorkuje i vzdálenost mezi prvním a posledním bodem. Navzorkovanou čáru je možné vidět na obrázku 4.3.



Obrázek 4.3: Nakreslená čára (zeleně) se zvýrazněnými navzorkovanými body(černě)

4.2.3 Delaunayho triangulace

Dalším krokem vytvoření 3D modelu bude pospojování bodů tak, aby mezi těmito body vznikla trojúhelníková síť. Této operaci se říká triangulace. Nejpoužívanější triangulací, která se v praxi používá je Delaunayho triangulace. Existuje mnoho variací této metody. V tomto případě byl použit základní přístup metodou inkrementální konstrukce [1], kterou je možné vidět na obrázku 4.4.

Nyní je vytvořena navzorkovaná čára, která se skládá z bodů od sebe vzdálených o konstantní vzdálenost. Nejprve je potřeba nalézt bod s extrémní souřadnicí x nebo y. V tomto případě bylo zvoleno, že bude nalezen bod s minimální souřadnicí na ose x. K tomuto bodu je potřeba nalézt nejbližší bod. Jelikož, jak již bylo zmíněno, je čára navzorkována a body jsou od sebe vzdáleny stejně daleko, může být zvolen bod nalevo či napravo od tohoto bodu. Tímto dostáváme hranu prvního trojúhelníku Delaunayho triangulace.

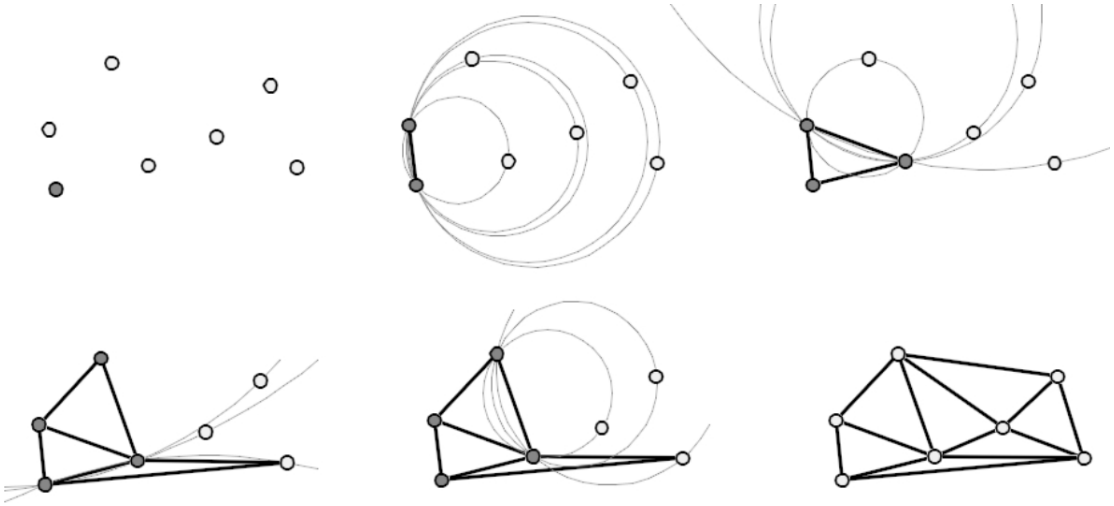
V dalším kroku hledáme třetí bod, pro který má kružnice opsaná ke dvěma bodům hrany minimální poloměr. Tím zaručíme, že žádný jiný bod neleží uvnitř trojúhelníku. Abychom zjistili, o které tyto tři body se jedná, je potřeba dosadit každý bod do rovnice kružnice:

$$(x - x_0)^2 + (y - y_0)^2 = r^2.$$

Pro tři body trojúhelníku vzniknou tři rovnice o třech neznámých [8]. Z těchto rovnic pak vypočítáme:

- souřadnici středu kružnice x_0
- souřadnici středu kružnice y_0
- poloměr kružnice r.

Po nalezení trojúhelníku vzniknou tři hrany. Pro dvě nově vytvořené hrany opět hledáme nový bod, pro který má kružnice opsaná nejmenší poloměr. Proces se opakuje dokud nejsou prohledány všechny hrany. Nakonec vznikne plošná trojúhelníková síť (obr. 4.5), která slouží jako základ pro další postup při vytváření 3D modelu.

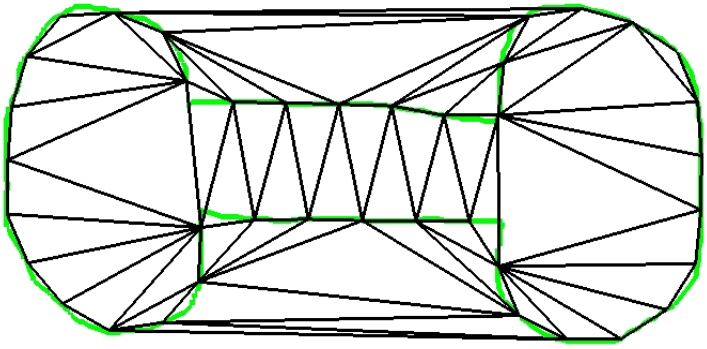


Obrázek 4.4: Delaunayho triangulace metodou inkrementální konstrukce [1]

4.2.4 Odstranění vnějších trojúhelníků

Vytvořená síť (obr. 4.5) se nyní dá rozdělit na dvě části. První vnitřní oblast obsahuje trojúhelníky, které jsou obklopeny body nakreslené čáry. Naopak druhá část obsahuje trojúhelníky, které jsou mimo zmíněnou oblast. Tyto trojúhelníky jsou nežádoucí a musí být odstraněny, protože body nakreslené čáry vymezují tvar budoucího 3D objektu a bez tohoto odstranění by nebylo možné vytvořit objekt složitějších tvarů.

K odstranění těchto trojúhelníku by bylo možné využít vlastností půl hran. Každá hrana má dvě opačně směřující půl hrany a u každé hrany musí existovat alespoň jedna stěna, přičemž tato stěna přiléhá k jedné z půl hran. V tomto případě je zřejmé, že pokud existuje pouze jedna stěna, bude se jednat o stěnu uvnitř oblasti ohraničené čárou. Při postupném

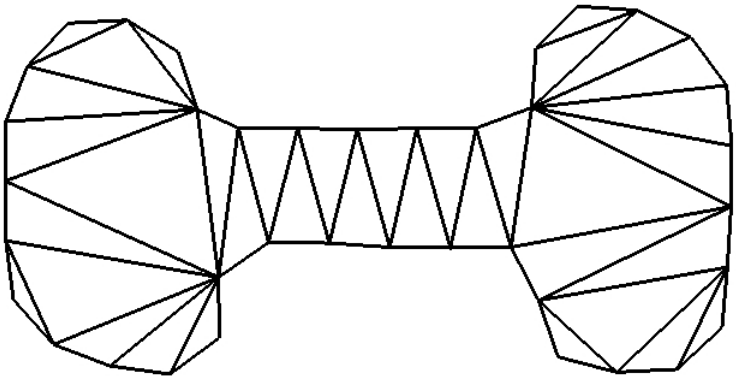


Obrázek 4.5: Síť vytvořená Delaunayho triangulací (černě) a původní nakreslená čára (zeleně)

průchodu po hranách tvořených sousedními body hraniční čáry budeme odstraňovat stěny ležící na vnější (opačné) půl hraně.

Tento postup však odstraní pouze trojúhelníky, které se dotýkají oblasti jen hranou, nikoli však trojúhelníky, které se dotýkají pouze pomocí bodů hraničící čáry. Takovéto stěny by bylo možné odstranit, a to tak, že po předchozím postupu budou odstraněny všechny trojúhelníky, které nemají žádnou společnou hranu. I přes toto opatření ještě nemusí být odstraněny všechny vnější stěny. Posledními neodstraněnými trojúhelníky mohou zůstat ty, které se dotýkají pouze body hraničící čáry a zároveň mají společnou hranu s jiným takovýmto trojúhelníkem.

V tomto zásuvném modulu byla použita metoda vycházející z již nastíněných postupů. Při postupném průchodu po hranách, které jsou tvořeny sousedními body hraniční čáry, jsou (pokud existují) odstraněny stěny ležící u vnější (opačné) půl hrany a rekurzivně i stěny sdílející hranu s touto stěnou. Odstraňovaní je vždy zastaveno v případě, že se narazí na hranu tvořenou dvěma sousedními body hraniční čáry. Na obrázku 4.6 je možné vidět síť, která již neobsahuje vnější trojúhelníky.



Obrázek 4.6: Síť bez externích trojúhelníků

4.2.5 Trojúhelníkové vějíře

Trojúhelníkový vějíř se dá popsat jako několik trojúhelníků spojených do jednoho společného vrcholu. V tomto zásuvném modulu jsou použity pro rozdělení konečných větví sítě. Využívá se jich z důvodu možnosti snadného zaoblení konečných oblastí a eliminaci prudkého zkosení.

Abychom mohli vytvořit na koncích větví trojúhelníkové vějíře, je potřeba nalézt všechny trojúhelníky s dvěma externími hranami. To jsou stěny, které mají s ostatními stěnami společnou pouze jednu hranu. Tohoto dosáhneme opět za pomocí vlastností půl hran. Pokud u opačných půl hran trojúhelníku leží jen jedna strana, jedná se o hledaný trojúhelník.

Nyní pro každý trojúhelník s dvěma externími hranami vytvoříme půlkruh se středem v polovině vnitřní hrany a ve směru zmíněných externích hran. Nyní otestujeme, zda leží třetí bod trojúhelníku mimo půlkruh. Pokud ne, trojúhelník odstraníme a postupujeme na sousední trojúhelník. V případě, že se jedná o stěnu s jednou externí hranou postupujeme podobně jako v prvním případě. Na nové vnitřní hraně vytvoříme půlkruh a testujeme, zda některý z již testovaných bodů či některý z vrcholů aktuálního trojúhelníku, neleží mimo tento půlkruh. Takto pokračujeme dokud nenarazíme na trojúhelník bez externí hrany (trojúhelník, který je obklopen jinými stěnami) nebo dokud některý z bodů neleží mimo aktuální půlkruh.

V posledním kroku tvorby trojúhelníkových vějířů vytvoříme nový bod. V případě, že jsme narazili na trojúhelník bez externích hran, vytvoříme těžiště tohoto trojúhelníku a do tohoto bodu zkonstruujeme vějíř z bodů odstraněných trojúhelníků. Pokud předchozí průchod skončil tak, že některý z bodů ležel mimo půlkruh, vytvoříme střed aktuální vnitřní hrany a vějíř trojúhelníku zkonstruujeme do tohoto bodu.

Na obrázku 4.7 je možno vidět vytvořené trojúhelníkové vějíře na koncích větví oblasti.

4.2.6 Vytvoření hřbetu

Další operací, která rozděluje síť tak, aby bylo možné v dalších krocích vytvořit zaoblený objekt, je vytvoření hřbetu, který rozdělí oblast a její větve na polovinu.

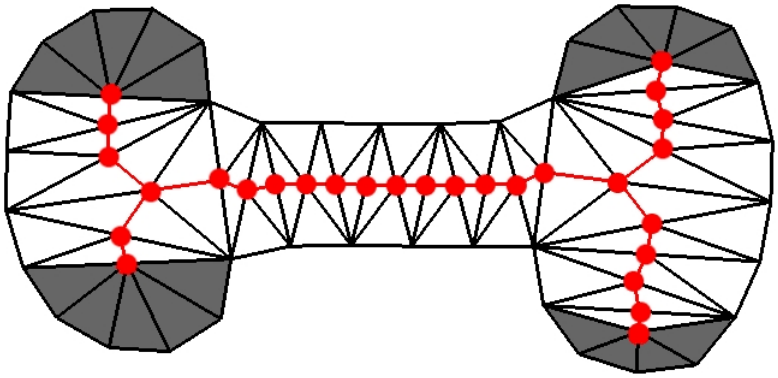
Každá vnitřní hrana sítě bude rozdělena (pokud ještě nebyla v podsekci 4.2.5) bodem v jejím středu. Tyto body rozdělí trojúhelníky s jednou vnější hranou na tři menší.

Do trojúhelníků bez vnějších hran bude přidán nový bod. Tímto novým bodem bude těžiště, které rozdělí trojúhelník na menší části. Pokud však již některá z částí byla upravena trojúhelníkovým vějířem, nebude se dále upravovat. Všechny ostatní části budou rozděleny na dva trojúhelníky. První bod těchto trojúhelníků bude těžiště, druhý bod střed vnitřní hrany a třetí bod bude jeden z původních bodů této hrany.

Nově vytvořený hřbet se tedy bude, jak je možno vidět na obrázku 4.7, skládat z těžišť trojúhelníků bez externí hrany a ze středů rozdělující vnitřní hrany.

4.2.7 Rozdělení trojúhelníků

Nyní již máme oblast, která je ohraničená body nakreslené čáry, a její větve jsou rozdělené hřbetem a trojúhelníkovými vějíři na jejich koncích. To však nestačí pro vytažení do prostoru. Výsledný objekt by tvarem připomínal jehlan, který by byl tažen po hřbetu. Dalším krokem tedy bude rozdělení všech trojúhelníků ještě na menší části, aby po vytažení do prostoru tvar objektu více připomínal zaoblený předmět. Na obrázku 4.8 je zobrazena takto rozdělená síť.



Obrázek 4.7: Síť s vyznačeným hřbetem (červeně) a s trojúhelníkovými vějíři (šedě)

V této části můžeme rozdělit trojúhelníky na dvě skupiny. První skupinu budou tvořit trojúhelníky, které mají alespoň jednu vnější hranu. Tyto trojúhelníky se zužují směrem ke hřbetu. Druhou skupinou jsou trojúhelníky bez vnější hrany. Ty se naopak zužují směrem od hřbetu.

V případě první skupiny budeme rozdělovat trojúhelníky směrem od vnější hrany. Postupně budeme přičítat na obou vnitřních hranách konstantu. Na těchto místech budeme vytvářet nové body a zároveň s nimi i nové trojúhelníky. Abychom věděli, jak máme trojúhelníky vytvořit, porovnáme vzdálenost posledního bodu levé vnitřní hrany a nového bodu na pravé straně vnitřní hrany se vzdáleností posledního bodu pravé vnitřní hrany a nového bodu levé vnitřní hrany. Nová hrana vznikne tam, kde bude tato vzdálenost kratší. Nakonec vznikne poslední trojúhelník, který bude mezi posledními body vnitřních hran a bodem který tyto hrany spojuje.

U druhé skupiny budeme při rozdělovaní na menší trojúhelníky, postupovat obdobně. Rozdíl bude spočívat v tom, že rozdělení bude probíhat směrem od vrcholu, který leží na okraji oblasti. První trojúhelník tedy bude vytvořen s tímto vrcholem a se dvěma novými. Poslední dva trojúhelníky vzniknou mezi posledními přidanými body ležícími na vnitřních hranách a dvěma body ležícími na hřbetu.

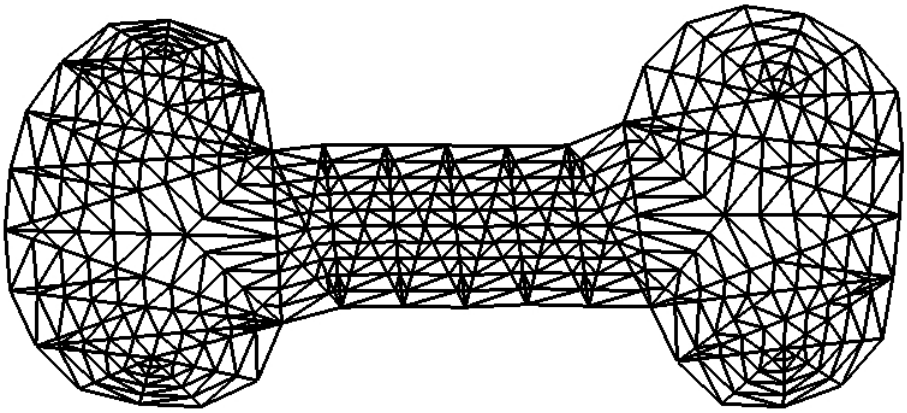
V každém rozdělení původních trojúhelníku je potřeba kontrolovat existenci nových bodů, protože vnitřní hrany jsou sdíleny s jinými trojúhelníky a mohlo by docházet k nekonzistentnosti nově vytvořené sítě.

4.3 Práce v 3D prostoru

Tato sekce rozšiřuje práci i do třetího prostoru. Do této doby nebylo nutné pracovat s osou z. Pro další operace je potřeba přidat souřadnice této osy každému bodu, a tím převést všechny body do trojrozměrného prostoru.

4.3.1 Převod do 3D prostoru

Všechny souřadnice bodů byly do této doby na pouze na osách x a y. Neuvažovalo se natočení kamery snímající objekt, natočení samotného objektu či jiné parametry měnící zobrazení v aplikaci. Prozatím tedy byly body trojúhelníkové sítě rozloženy pouze na ploše obrazovky v zobrazovací části okna aplikace OpenFlipper.



Obrázek 4.8: Síť s rozdělenými trojúhelníky

Pro přidaní souřadnice z není možné pouhé její nastavení na hodnotu 0. Plošné souřadnice bodu je potřeba přepočítat. K samotnému přepočítání byla použita funkce `unproject` z knihovny ACG, která převádí plošné souřadnice na souřadnice prostorové. Jako její parametry je potřeba vložit souřadnice bodu x, y a konstantní hodnotu 0.5, která udává umístění bodu na ploše obrazovky v zobrazovací části aplikace.

4.3.2 Vytažení do prostoru

Vytažení do prostoru probíhá kvůli zjednodušení operací zároveň s rozdělením trojúhelníků popsaném v 4.2.7. V tomto dokumentu však tyto dvě části byly odděleny pro lepší orientaci v postupech.

Tento proces má za cíl zformovat jednu stranu objektu do přibližného elipsovitého tvaru. Nejedná se o dokonalý výpočet tvaru objektu, ale tato operace by měla vytáhnout body na hřbetu přibližně o polovinu šířky tvaru objektu v aktuálním místě. Ostatní body se budou vytahovat v závislosti na vzdálenosti od okraje.

Výpočet vektoru určující směr vytažení

Nejprve je potřeba zjistit směr, ve kterém budou body vytahovány do prostoru. Tento směr bude udávat jednotkový vektor $\vec{e}$ kolmý ke všem bodům. Jelikož všechny body leží, bez ohledu na natočení kamery, na jedné rovině, vybereme tři libovolné body např. A, B a C. Z těchto tří bodů vypočítáme vektor $\vec{a}$ a vektor $\vec{b}$, přičemž:

$$\vec{a} = A - B \qquad \vec{b} = C - B.$$

Z těchto dvou vektorů vypočteme vektorový součin, jehož výsledkem je vektor $\vec{c}$, který bude kolmý k oběma vektorům, tudíž i k ostatním bodů a rovině, na které se nacházejí. Vzorec pro výpočet tedy bude:

$$\vec{c} = \vec{a} \times \vec{b}.$$

Nakonec vypočítáme délku tohoto vektoru a posléze i jednotkový vektor, který se rovná:

$$\vec{e} = \frac{1}{|c| \cdot \vec{c}}\,.$$

Díky tomuto jednotkovému vektoru snadno určíme souřadnice všech bodů vzdálených od roviny, která je tvořena dosavadními vrcholy. Souřadnice bodu K, který se nachází ve vzdálenosti v ve směru kolmém na rovinu od bodu L ležícího na zmíněné rovině, se pak budou rovnat:

$$K_x = v \cdot \vec{e}_x \cdot L_x, \quad K_y = v \cdot \vec{e}_y \cdot L_y, \quad K_z = v \cdot \vec{e}_z \cdot L_z.$$

Výpočet vytažení do prostoru

Samotné vytažení jednotlivých bodů pak probíhá postupně po každé vnitřní hraně, která začíná bodem na okraji a končí bodem na hřbetu. Pro bod na hřbetu je vypočítán průměr délek všech hran, se kterými je tento vrchol spojen. Tato délka se rovná maximu vytažení do prostoru. O ni bude poté bod na hřbetu vytažen. Pro zbývající body na hraně se bude postupovat tak, že každý bod bude vytažen do prostoru podle rovnice elipsy:

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1.$$

Jelikož však nepočítáme s absolutními souřadnicemi, ale s relativními vzdálenostmi, můžeme rovnici přepsat na:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

Do rovnice dosadíme tyto známe hodnoty:

- a – délka aktuální vnitřní hrany,
- b – průměr délek všech hran, které jsou spojeny s aktuálním bodem na hřbetu,
- x – vzdálenost vrcholu od aktuálního bodu na hřbetu

a nakonec vypočítáme vzdálenost od roviny pomocí vzorce:

$$y = \sqrt{b^2 \cdot \left(1 - \frac{x^2}{a^2}\right)}$$

Takto postupujeme pro každý bod, až nakonec vznikne jedna strana objektu vytažená do prostoru (obr. 4.9).

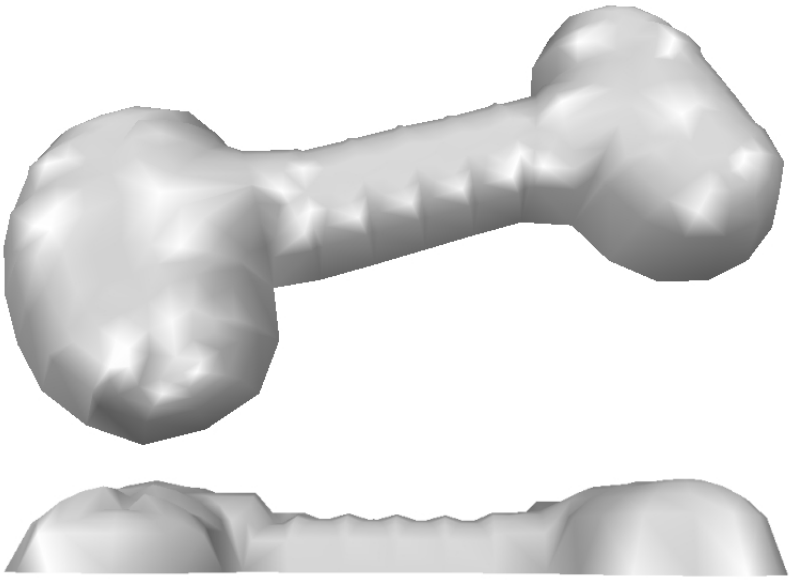
4.3.3 Vytvoření druhé strany objektu

Předposledním krokem vytvoření 3D modelu objektu je vytvoření jeho druhé strany.

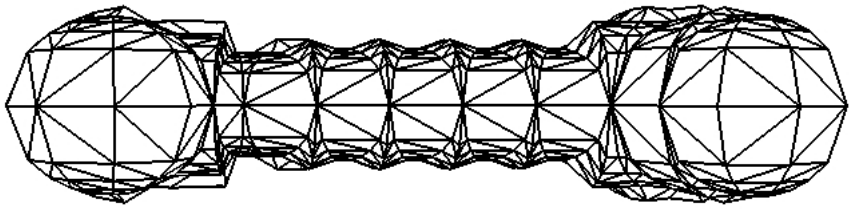
Nyní již máme, jak můžeme vidět na obrázku 4.9, jednu stranu objektu vytaženou do prostoru. Tuto stranu je potřeba zkopírovat, převrátit a propojit s první stranou tak, aby výsledný objekt byl symetrický. Na obrázku 4.10 je zobrazena síť vytvořené objektu po zkopírování strany.

Nejprve vybereme všechny vrcholy dosavadní sítě. Podle jejich propojení bude vytvořena identická kopie. Kopírovány budou však pouze body, které neleží na okraji. Pokud je součástí nového trojúhelníku krajní bod, bude použit bod originální. Tím dosáhneme propojení obou stran objektu.

Nakonec zbývá vytáhnou nové body do opačného směru se stejnou vzdáleností od roviny podle původních vrcholů. Pro každý nový bod tedy vypočteme vzdálenost a směr původního bodu od roviny a k němu vypočítáme směr opačný. Poté posuneme body podle vypočtených parametrů a tím docílíme vytvoření opačné strany objektu.



Obrázek 4.9: Jedna strana objektu vytažená do prostoru



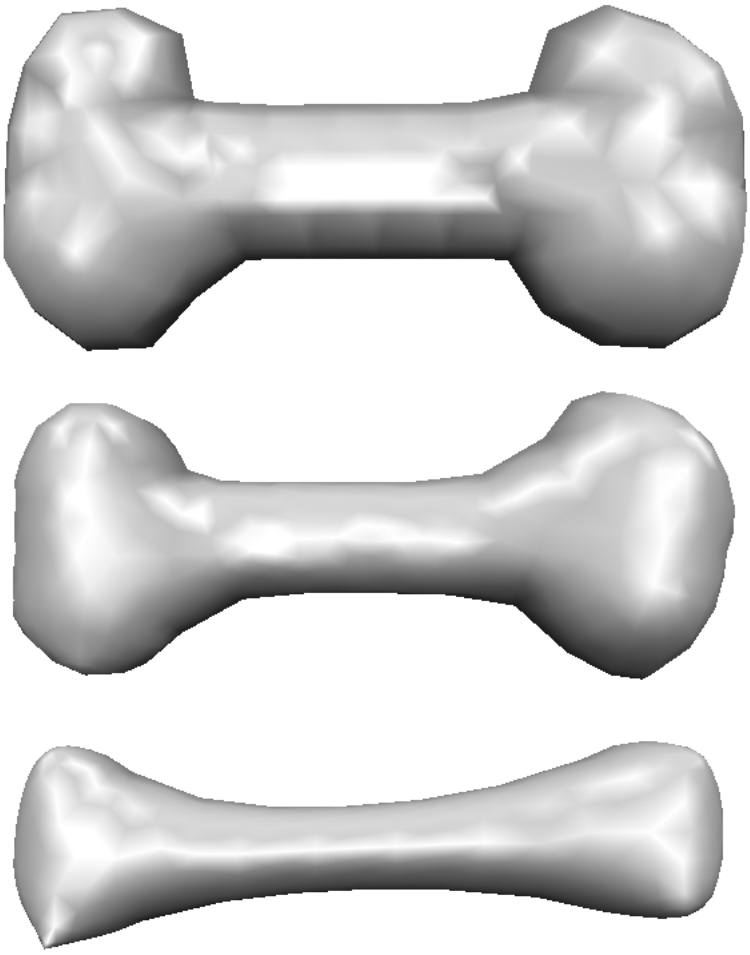
Obrázek 4.10: Síť vytvořeného symetrického objektu

4.3.4 Vyhlazení

Vytvořený objekt, jak lze vidět na obrázku 4.10, má řadu nedostatků v podobě výčnělků, špiček a ostrých hran. Takovýto objekt není esteticky přijatelný a je potřeba jej vyhladit. Přílišné vyhlazení jej však může výrazně zdeformovat a jeho tvar by se pak mohl značně lišit od tvaru, který je udáván nakreslenou čárou. Proto je nutné zvolit vhodnou míru vyhlazení. Jak lze vidět na obrázku 4.11, objekt bez vyhlazení obsahuje příliš mnoho ostrých hran a objekt, který je mnohokrát vyhlazován již neodpovídá původnímu tvaru.

V tomto zásuvném modulu byl použit základní přístup pro vyhlazování modelů. Pro každý bod modelu je zjištěna množina okolních bodů M přičemž platí, že tyto body jsou spojeny s aktuálním bodem libovolnou hranou. Z této množiny a z aktuálního bodu jsou vypočítány průměry souřadnic x, y a z a těmito průměry jsou nastaveny nové souřadnice. Tento postup je několikrát opakován k získání hladšího modelu. Vzorec pro vyhlazení se tedy rovná:

$$X_i = \frac{X_i + \sum_{i=0}^{|M|} M_i}{1 + |M|}$$



Obrázek 4.11: (shora): Objekt bez vyhlazení, vyhlazení 2x a vyhlazení 20x

4.4 Testování

Nedílnou součástí vývoje každé aplikace je testování její funkčnosti. Testování může probíhat v různých variantách a s různým počtem fází. Zde bude popsáno testování tohoto zásuvného modulu.

4.4.1 Testování aplikace

Testování této aplikace bylo prováděno zároveň s její implementací. Každý implementovaný krok byl záhy po dokončení otestován a následně byly opraveny chyby. Pokud se nalezená chyba neobjevovala či žádná další nebyla nalezena, pokračovalo se v implementaci dalších kroků.

Po dokončení poslední implementované části programu byla aplikace znovu testována, a zároveň byl zredukován zdrojový kód o opakující se bloky a byly opraveny části, které nebyly komentovány či v komentářích obsahovaly chyby.

Testování aplikace bylo prováděno především manuálně (nikoli automatickými testy). Za účelem testování byl využit standardní textový výstup příkazového řádku, rozhraní aplikace

OpenFlipper pro výpis záznamů a dále samotný grafický výstup aplikace.

Díky textovým výstupům byly zkontrolovány především části, které vyžadovaly přesný matematický výpočet, či části, které pracovaly s některou z datových struktur trojúhelníkových sítí objektů. Pro kontrolu konzistence trojúhelníkové sítě a správnost procesů pracující s touto sítí byl naopak použit výstup grafický.

4.4.2 Příklady nalezených chyb

V této podsekci budou uvedeny některé z nalezených chyb a bude vysvětlen způsob jejich opravy. Jedná se o podstatné chyby, které byly nalezeny během implementace a poté záhy opraveny.

Kreslená čára

Při kreslení čáry (podsekce 4.2.1) bylo zjištěno, že při pomalejším tahu vzniká velké množství bodů a některé body jsou i duplicitní. Oprava spočívala v okamžitém odstranění takovýchto bodů. Segmentací čáry se odstranily oblasti, kde byly body příliš nahuštěny či jejich rozložení bylo příliš řídké.

Delaunayho triangulace

Během Delaunayho triangulace, která je popsána v sekci 4.2.3, při výpočtech středu kružnice byl zaznamenán v několika případech pád aplikace. Tento pád byl způsoben dělením nulou, které nastávalo v případech, kdy by se vypočítaný poloměr kružnice blížil nekonečnu (tři body na přímce). V těchto případech není možné vytvořit kružnici. Problém byl opraven zamezením vytvoření kružnice při nulovém jmenovateli ve výpočtu.

Odstranění vnějších trojúhelníků

V situacích, kdy se odstraňovaly nepotřebné vnější trojúhelníky, nebylo zvolenou metodou dosaženo stoprocentní úspěšnosti odstranění. Opravení problému proběhlo stejně jako je popsáno v podsekci 4.2.4.

Převod do 3D prostoru

Pokud bylo se scénou před nakreslením čáry jakkoli manipulováno, objekt nebyl správně vytažen do prostoru. Příčinnou tohoto problému bylo špatné převedení plošných souřadnic na souřadnice prostorové. Tento problém byl zjištěn až v poslední fázi testování celé aplikace, jelikož do této doby nebylo se scénou manipulováno. Opravení této chyby bylo zajištěno pomocí výpočtu směrového vektoru, stejně jako je popsáno v podsekci 4.3.2.

Přidání bodů do sítě

Při vkládání bodů do sítě, které již byly předtím přidány, nebyla správně rozpoznána jejich existence. To bylo způsobeno porovnáním souřadnic, jehož výsledek se lišil v řádu sto tisícin. Zaokrouhlením hodnot souřadnic na deseti tisíciny bylo dosaženo opravení této chyby.

Kapitola 5

Výsledky

Následující kapitola se zaměřuje na implementovaný zásuvný modul a výsledky dosažené jeho prostřednictvím.

5.1 Aplikace

Tato práce byla implementována jako zásuvný modul do aplikace OpenFlipper (obr. 4.1). Spouští se a vypíná se po stisknutí tlačítka v pravém panelu této aplikace, který je určen pro zásuvné moduly.

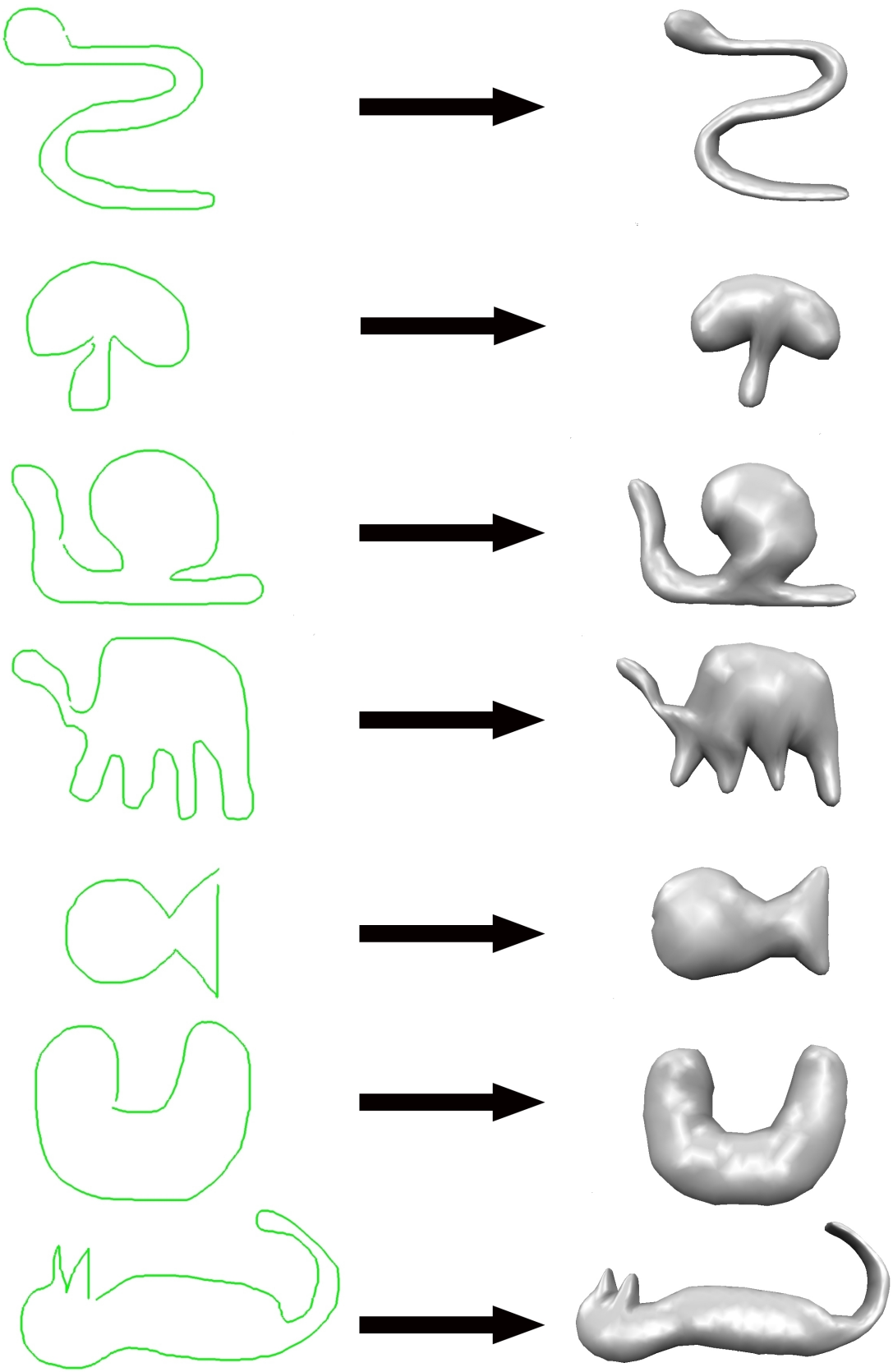
Pomocí čáry nakreslené jedním tahem je tato aplikace schopná vytvářet jednoduché 3D modely. Čára nesmí sama sebe protínat a její velikost musí být větší než vzdálenost mezi několika body. Jinak pro tuto čáru nejsou žádná omezení.

Pokud model nevyhovuje uživatelově představě, může nakreslit nový tvar, díky kterému se vytvoří nový objekt a původní je odstraněn. Takto se mohou objekty vytvářet, dokud není zásuvný modul či aplikace vypnuta.

Výsledný model je vytvořen v řádu několika sekund a je možné s ním dále pracovat díky ostatním zásuvným modulům aplikace OpenFlipper. Po ukončení práce je možné objekt zahodit, či jej uložit do jednoho z mnoha formátů, které aplikace podporuje.

5.2 Dosažené výsledky

Na obrázku 5.1 je možné vidět příklady modelů vytvořené touto aplikací. Jedná se o modely, které byly vytvořeny finální verzí aplikace, a které zobrazují tvary zamýšlených předmětů či zvířat. Na levé straně jsou zobrazeny obrysy nakreslené skicovací čarou a na straně pravé jsou vytvořeny 3D modely, které vznikly z těchto tvarů.



Obrázek 5.1: Příklady vytvořených modelů

Kapitola 6

Závěr

Cílem mé bakalářské práce bylo navržení a implementace jednoduchého nástroje umožňujícího modelování 3D objektů pomocí jednoduchých skic a náčrtků.

Výsledkem této práce je zásuvný modul do aplikace OpenFlipper. Zmíněný modul umožňuje vytvoření jednoduchých 3D modelů pomocí různě nakreslených, zakřivených a uzavřených čár, které udávájí jejich tvar. Výsledné objekty jsou reprezentovány trojúhelníkovými polygonálními sítěmi. Tyto modely je možné uložit či je dále upravovat pomocí funkcí ostatních zásuvných modulů aplikace OpenFlipper. Tímto bylo splněno zadání této práce zabývající se skicováním 3D modelů.

Skicování 3D modelů stále ještě málo probádanou oblastí, která se může ubírat různými směry. Možností jak tento zásuvný modul rozšířit je opravdu mnoho. Především by se jednalo o implementaci navržených metod (vytažení a useknutí) pro úpravu nově vzniklého objektu. Vyladěním jednotlivých funkcí či jejich náhradou by bylo možné dosáhnout zrychlení celého procesu vytváření 3D modelu. Přidáním nových funkcí, jako jsou modelovací křivky, vybarvování či texturování objektu, by posunulo aplikaci o nemalý kus dále.

Dle mého názoru by bylo však užitečnější, aby se tento skicovací modul posunul jiným směrem než dosavadní skicovací programy a neomezoval se pouze na aktuální čáru, ale aby kráčel stejnou cestou jako klasické skicování, které umožňuje nakreslit i mistrovská díla na obyčejný papír. Tím chci říci, že např. pomocí většího počtu nakreslených čar a potvrzení těchto tahů by mohl výše zmíněný program vygenerovat i velice kvalitní 3D scénu.

Na závěr je nutno podotknout, že díky této práci jsem více porozuměl problematice reprezentace a zpracování 3D modelů. Prohloubil jsem své znalosti v oblasti programování v jazyce C++, psaní dokumentů pomocí systému LaTeX a díky tomuto prvnímu většímu projektu jsem získal velmi bohaté zkušenosti. Pro můj studijní vývoj byla tato práce důležitým mezníkem.

Literatura

[1] Bayer, T.: Triangulační algoritmy. [online], 2008-11-04 [cit. 2012-04-07].
URL http://web.natur.cuni.cz/~bayertom/IM/idm4.pdf

[2] Fořt, P.: Zpracováváme polygonální 3D data pro vizualizaci. [online], [cit. 2012-04-28].
URL http://www.cad.cz/grafika-design/81-grafika-design/3149-zpracovavame-polygonalni-3d-data-pro-vizualizaci.html

[3] Funkhouser, T.: Polygonal Meshes. [online], 2006 [cit. 2012-04-22].
URL http://www.cs.princeton.edu/courses/archive/fall06/cos526/lectures/meshes.pdf

[4] Igarashi, T.: SmoothTeddy: Quick 3D Modeling and Painting. [online], 2003 [cit. 2012-04-14].
URL
http://www-ui.is.s.u-tokyo.ac.jp/~takeo/java/smoothteddy/index.html

[5] Igarashi, T.; Cosgrove, D.: Chameleon : 3D Paint for Teddy. [online], 2001 [cit. 2012-04-15].
URL http://www-ui.is.s.u-tokyo.ac.jp/~takeo/chameleon/chameleon.htm

[6] Igarashi, T.; Matsuoka, S.; Tanaka, H.: Teddy: a sketching interface for 3D freeform design. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, 1999.
URL http://www-ui.is.s.u-tokyo.ac.jp/~takeo/papers/siggraph99.pdf

[7] Kršek, P.; Španěl, M.: Nové trendy počítačové grafiky. [online], 2012 [cit. 2012-05-02].
URL https://www.fit.vutbr.cz/study/courses/IZG/private/lecture/izg_slide_new_trends_print.pdf

[8] McRae, G.: Equation of Circle given 3 points. [online], 2009-06-16 [cit. 2012-04-21].
URL http://2000clicks.com/mathhelp/GeometryConicSectionCircleEquationGivenThreePoints.aspx

[9] Nealen, A.; Igarashi, T.; Sorkine, O.; aj.: FiberMesh: designing freeform surfaces with 3D curves. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, 2007.
URL http://www-ui.is.s.u-tokyo.ac.jp/~takeo/papers/fibermesh.pdf

[10] WWW: 3D modeling. [online], 2012-03-29 [cit. 2012-04-21].
URL http://en.wikipedia.org/wiki/3D_modeling

[11] WWW: Blender. [online], 2012 [cit. 2012-04-08].
URL http://www.blender.org/

[12] WWW: Autodesk 3ds max. [online], 2012 [cit. 2012-04-09].
URL http://usa.autodesk.com/3ds-max/

[13] WWW: OpenFlipper 3D Modeling Toolkit Documentation. [online], 2012 [cit. 2012-04-29].
URL http://openflipper.org/Documentation/latest/index.html

[14] WWW: OpenMesh Documentation. [online], 2012 [cit. 2012-04-29].
URL http://openmesh.org/Documentation/OpenMesh-Doc-Latest/index.html

[15] Žára, J.: *Moderní počítačová grafika*. Brno: Computer Press, první vydání, 2004, ISBN 80-251-0454-0, 609 s.

Dodatek A

Obsah DVD

Disk přiložený k této práci obsahuje následující složky:

- **app/** – Přeložená spustitelná aplikace OpenFlipper s implementovaným zásuvným modulem
- **doc/** – Vygenerovaná technická zpráva ve formátu PDF
- **app-src/** – Zdrojové soubory aplikace
- **doc-src/** – Zdrojové soubory technické zprávy
- **poster/** – Plakát prezentující tuto práci

Dodatek B

Plakát