Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

# Bakalářská práce

## Integrace softwaru pro dekompozici svalů do aplikace Blender

Plzeň 2022 Jan Hereš

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 23. června 2022

Jan Hereš

# Abstract

This work's primary goal is to create a fully functional add-on for the modelling software called Blender, which would lets users decompose muscle models into individual muscle fibres, incorporated into the Blender's user environment. Before the creation of such an add-on, the decomposition process has been quite complicated and unnecessairly time consuming, from which arose the need for this add-on. Throughout this paper, firstly is the reader introduced into the problematics when creating 3D models of muscles, followed by an introduction of the external tools, essential for the whole process. Following that, the reader is presented by the design proposal, with contrast to the concrete final implementation. Lastly comes a chapter dedicated to the testing process, practical examples and demonstrations of functionality and review of the results.



# Abstrakt

Tato práce je zaměřena na vytvoření dodatku do modelovací aplikace Blender, který usnadní uživatelům dekompozici modelů svalů přímo v rámci aplikace. Před vznikem tohoto dodatku byla tato činnost komplikovaná a zdlouhavá, z čehož vznikl požadavek na vytvoření tohoto dodatku. V rámci této práce je čtenář nejdříve uveden do problematiky modelování svalů, nadále jsou představeny externí nástroje využívané pro jejich dekompozici, a následně představen původní návrh dodatku. V kontrastu je pak představena finální implementace a demonstrace výsledků v rámci testování na praktických příkladech. Výsledný dodatek je plně funkční, nicméně je potřeba ho dále zdokonalovat, jelikož každý model svalu je odlišný.

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: 2021/2022

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

| | |
|---|---|
| Jméno a příjmení: | **Jan HEREŠ** |
| Osobní číslo: | **A19B0051P** |
| Studijní program: | **B0613A140015 Informatika a výpočetní technika** |
| Specializace: | **Informatika** |
| Téma práce: | **Integrace softwaru pro dekompozici svalů do aplikace Blender** |
| Zadávající katedra: | **Katedra informatiky a výpočetní techniky** |

## Zásady pro vypracování

1. Seznamte se s interním projektem Muscle Wrapping 2.0, zejména pak s existujícím softwarem pro dekompozici svalů.
2. Seznamte se s prostředím Blender a popište programové možnosti jeho rozšiřování o další funkcionalitu.
3. Navrhněte a naimplementujte rozšíření pro Blender umožňující dekomponovat svaly ve scéně.
4. Navrhněte a naimplementujte vhodné grafické rozhraní pro nastavování parametrů implementovaného rozšíření.
5. Vytvořte instalační balíček pro snadné nasazení rozšíření u uživatelů.
6. Proveďte uživatelské testování a dosažené výsledky zhodnoťte.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**
Rozsah grafických prací: **dle potřeby**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Doc. Ing. Josef Kohout, Ph.D.**
Nové technologie pro informační společnost

Datum zadání bakalářské práce: **4. října 2021**
Termín odevzdání bakalářské práce: **5. května 2022**

L.S.

**Doc. Ing. Miloš Železný, Ph.D.**
děkan

**Doc. Ing. Přemysl Brada, MSc., Ph.D.**
vedoucí katedry

V Plzni dne 14. října 2021

# Poděkování

Rád bych poděkoval panu Doc. Ing. Josefu Kohoutovi, Ph.D. za veškeré odborné poznatky k této práci a celkový, velice přívětivý, přístup při vedení práce.

# Obsah

# 1 Úvod

Anatomie lidského těla, konkrétně tedy lokomoční (pohybový) aparát, je součástí zkoumání lidstva již po několik staletí. Díky již získaným poznatkům jsme schopni analyzovat **některé** problémy tohoto aparátu u libovolného jedince a nabídnout jejich konkrétní řešení. Příkladem jejich využití v dnešní době je například **plán předoperačních vyšetření, rehabilitačních procedur nebo návrh protéz**. Bohužel, množství těchto poznatků je stále **nedostatečné** pro existenci všech řešení, tudíž je nutné v tomto zkoumání pokračovat. Vzhledem k tomu, že je člověk žijící organismus, není možné zkoumat klíčové vlastnosti tohoto aparátu takzvaně *in vivo*, a proto vznikl požadavek na vytvoření jeho umělého modelu. S využítím moderních technologií bylo možné vytvořit relativně přesný model kostry člověka, avšak z hlediska příslušných **kosterních** svalů je situace mnohem složitější. Popsání jejich tvarů a možných pohybů je velice komplexním problémem a doposud nebyl nalezen způsob, který by dal vzniku takovému modelu, jež je dostatečně přesný, univerzální a na základě kterého by se daly získané poznatky aplikovat v praxi.

Cílem této práce je tedy vytvořit rozšíření do aplikace Blender, které bude využívat dále zmíněných nástrojů k automatizaci a celkovému usnadnění tvorby modelů kosterních svalů. Konkrétně bude tedy úkolem vytvořit takové rozšíření, které sjednocuje následující činnosti (konkrétněji činnosti 3, 4 a 5, obrázek 2.3b).

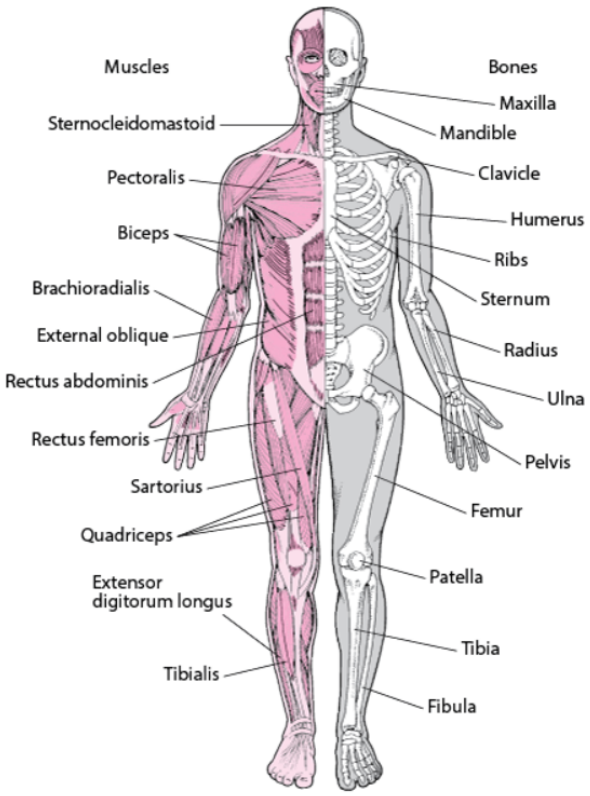
- Export modelu z modelovacího nástroje pro další použití,
- Strukturalizace modelu pro přesnější reprezentaci reálné předlohy,
- Import strukturalizovaného modelu zpět do modelovacího nástroje.

Pro zajištění komfortnosti a jednoduchosti častého používání bude navrženo grafické rozhraní s důrazem na právě zmíněné vlastnosti.

Text této práce je nadále rozdělen do kapitol. V následující kapitole bude představen pohybový aparát člověka, konkrétně struktura kosterního svalstva. Tyto svaly budou v častých případech předlohami k modelům, které budou v rámci tohoto dodatku dekomponovány. Dále jsou ukázány možnosti reprezentace a tvorby takových modelů, následně vysvětlení samotného procesu dekompozice. Poté jsou v kapitolách 3 a 4 představeny veškeré nástroje využité v rámci celého procesu, počínaje tvorbou modelu až po samotnou

dekompozici. Kapitoly 5 a 6 představují původní návrh dodatku a následnou finální implementaci. Předposlední kapitola pojednává o procesu testování dodatku na konkrétních modelech a poslední kapitola obsahuje shrnutí dosažených výsledků.

# 2 Pohybový aparát člověka




Obrázek 2.1: Znázornění jednotlivých částí pohybového aparátu člověka. Převzato z https://www.msdmanuals.com

## 2.1 Kosterní svalstvo

Pro hlubší pochopení problematiky tvorby modelu pohybového aparátu člověka je žádoucí objastnit strukturu kosterního svalstva.

Z pohledu **anatomického** označuje pojem „kosterní svalstvo" výhradně svalovou, příčně pruhovanou tkáň, tzv. *muscle belly* [7]. Svalová tkáň je souborem snopců tvořených jednotlivými svalovými vlákny. Jednotlivá vlákna se dají dále dělit dle jejich struktury, avšak pro účely této práce je lze považovat za základní stavební prvek libovolného kosterního svalstva. Úponová

oblast, taktéž označována jako *muscle tendon* [7], je měkkou a zároveň pevnou tkání představující oblast, ve které je sval „upevněn" k příslušné kosti. Stěžejním faktem je zde **odlišování** těchto částí.

**Biomechanický** pohled totiž tyto dvě části slučuje, a tedy pojmem „kosterní sval" je označován sval jako celek, respektive svalová tkáň včetně jednotlivých úpon. Vzhledem k definici úponových oblastí pro tento typ svalstva je zřejmé, že jejich mechanické vlastnosti definují limity pohybu pro obě tkáně, čímž se zjednoduší jeho biomechanická analýza. Z tohoto důvodu bude nyní kosterní svalstvo bráno z pohledu biomechanického.

## 2.2 Modelování kosterního svalstva

### 2.2.1 Reprezentace modelu svalu

V závislosti na výše zmíněné definici vyplývá, že je nutné sval a jeho úponové oblasti vhodným způsobem reprezentovat tak, aby byly dále počítačově zpracovatelné. Každý model se skládá z nějakého typu dostupných primitiv, v závislosti na možnostech použitého nástroje a uvážení autora modelu. Nejčastěji je model složen z tzv. **polygonů** (mnohoúhelníků), jež jsou tvořeny z jednotlivých vrcholů a jejich spojnic (hran). Model je pak tvořen z libovolného množství těchto primitiv, a proto je často označován jako tzv. **mesh**, neboli polygonová síť. Zde konkrétně je za primitivum považován **trojúhelník**, takže v případě celého modelu hovoříme o trojúhelníkové síti. Z pohledu úponových oblastí se jedná o tutéž situaci, jelikož ji lze definovat mnoha způsoby, například jako lomennou čáru, křivku, a další. Pro účely této práce předpokládejme reprezentaci ve formě **množiny bodů „ohraničujících" příslušnou oblast a zároveň respektující dané pořadí**.

### 2.2.2 Tvorba modelu

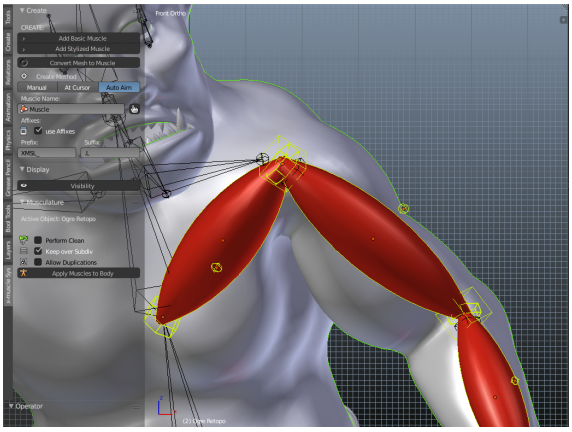
Tvorba modelu svalu na základě reálné předlohy je čistě kreativní činností a nelze ji striktně popsat. Nicméně, postup modelování lze do jisté míry zobecnit následovně.

#### Obecný postup

Jako první krok je nutné vytvořit určitý typ modelu svalu v libovolném dostupném modelovacím nástroji. Zde konkrétně se bude jednat o tzv. objemový model svalu, nebo-li model zachycující **vnější** strukturu předlohy.

Dalším krokem je tvorba modelů úponových oblastí. Vzhledem k faktu, že se kosterní sval vždy upíná k příslušné kosti, je žádoucí mít ve zvoleném nástroji dostupný její model, případně ho ručně vytvořit obdobným způsobem. Úponová oblast bude tedy samostatným objektem, nicméně by měl být každý její polygon zároveň součástí modelu kosti, ke které je sval upnut.

Posledním krokem je strukturalizace svalu. Jelikož je v tuto chvíli model celistvým objektem (viz obrázek 2.2), je nutné ho strukturalizovat (v tomto případě na jednotlivá vlákna) tak, aby odpovídal jeho reálné předloze.



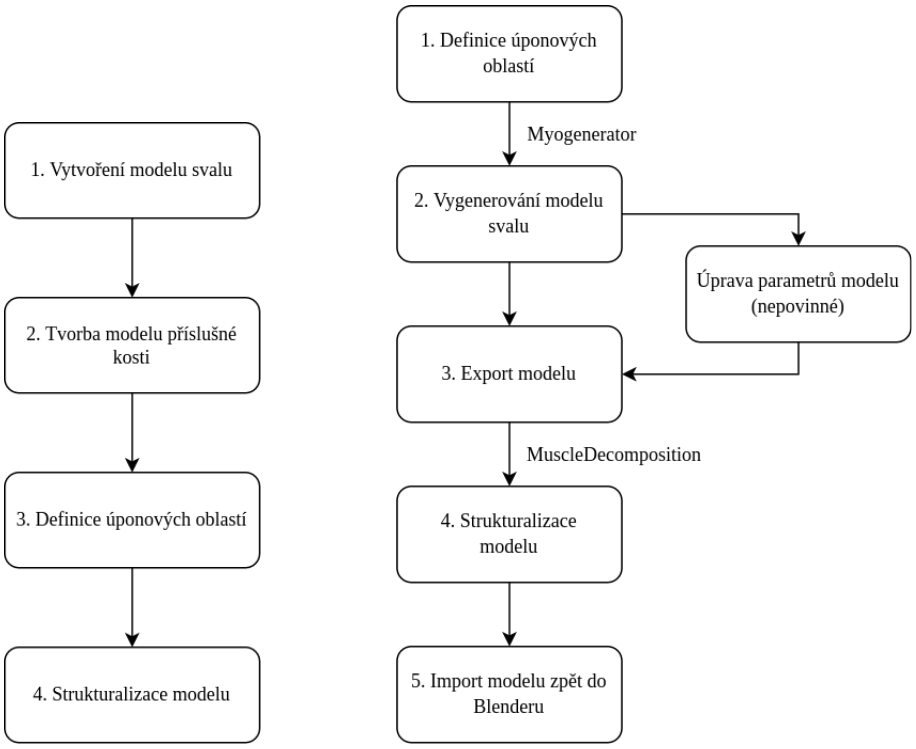

Obrázek 2.2: Příklad vytvořeného modelu svalu (převzato z https://blenderartists.org)

### Konkrétní postup

Pro účely této práce byl navržen konkrétní postup *Dr. Evou C. Herbst* ze švýcarské **University of Zürich**, se kterou je vedena úzká spolupráce, z níž přišla iniciativa vzniku rozšíření, jež je předmětem této práce. Jejím záměrem je jak vytvoření takových modelů kosterních svalů, které co nejblíže odpovídají reálným předlohám svým tvarem i vlastnostmi, tak jejich následná biomechanická analýza.

Pro tvorbu modelů svalu je zde konkrétně využíván program **Blender** (podrobněji kapitola 4.1). Pro vytvoření modelu svalu zde bude využit nástroj **MyoGenerator** (dále kapitola 3.2). Jeho výhoda spočívá v automatickém generování objemového modelu svalu v závislosti na jeho úponových oblastech. Prvním krokem zde, na rozdíl od obecného postupu, není tvorba samotného modelu svalu, nýbrž vytvoření zmíněných oblastí.

Po procesu automatického generování (konkrétnější vizualizace v kapitole 3.2) je model svalu tvarově vyhovující, avšak není nijak strukturalizován. K




(a) Obecný proces (b) Konkrétní proces

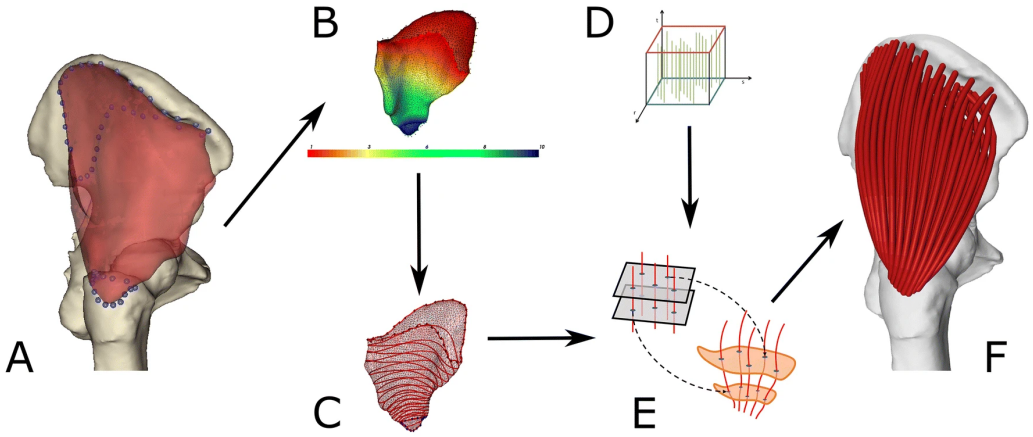
Obrázek 2.3: Znázornění jednotlivých procesů pro vytvoření modelu svalu

tomu bude využit nástroj **MuscleDecomposition**. Jeho funkčnost je detailněji popsána v kapitole 3.1.1. Ve stručnosti se jedná o nástroj, který je využíván pro dekompozici objemového modelu svalu na model jednotlivých svalových vláken. Tento nástroj je externí, čili není zakomponován přímo v modelovacím prostředí Blenderu, a proto je nutné nejdříve zmíněný vytvořený model svalu exportovat. Poté je možné provést proces dekompozice, jehož výsledek je pak nutné importovat zpět do Blenderu. Pro lepší vizualizaci je obecný a konkrétní proces tvorby modelu svalu znázorněn na obrázku 2.3.

## 2.3 Dekompozice modelu svalu

V obou výše popsaných postupech je zmíněn proces dekompozice svalu na jednotlivá vlákna, nicméně nikde není implicitně řečeno, jakým způsobem tato činnost probíhá. Konkrétní postup dekompozice by mohl být rozdělen na následující činnosti. Nejdříve jsou úponové oblasti zobrazeny na polygonovou síť samotného svalu. Poté je z této sítě vytvořeno pole skalárů, kdy se nejnižší hodnoty nacházejí v oblasti počáteční úpony a nejvyšší v

oblasti úpony koncové. Na základě těchto hodnot jsou vytvořeny izolinie, kdy každá reprezentuje jedno svalové vlákno. Pomocí šablony jsou pak tyto spojnice transformovány zpět do sítě původního modelu svalu, čímž vzniknou modely jednotlivých vláken. Pro hlubší pochopení je na obrázku 2.4 tento proces detailněji znázorněn.




Obrázek 2.4: Znázornění procesu dekompozice svalu (*gluteus medius*) na jednotlivá svalová vlákna (převzato z [8]). Sval reprezentovaný pomocí zmíněné *mesh* (A), společně s úponovými oblastmi (vyznačeny modrými body) jsou brány jako vstup pro MuscleDecomposition. Tyto oblasti jsou poté zobrazeny na samotnou síť modelu, která je převedena na pole skalárů (B), protnuté množinou izolinií reprezentující body jednotlivých vláken (C). Šablona pro samotnou strukturalizaci (D) je pak transformována opět na samotnou síť modelu (E), z důvodu převodu do standardního formátu. V tomto konkrétním případě byl sval diskretizován do 100 svalových vláken (F), přičemž se každé vlákno skládá z 15 lomenných čar.

# 3 Popis využitých nástrojů pro dekompozici modelů svalu

U konkrétního postupu Dr. Herbst jsou kromě modelovacího softwaru využívány externí nástroje, které byly vytvořeny pro usnadnění celého procesu dekompozice. Mezi tyto nástroje patří zmíněný *MyoGenerator* (samostatný nástroj) a *MuscleDecomposition*, který je součástí rozsáhlého projektu Muscle Wrapping 2.0. Struktura a princip využití těchto nástrojů jsou popsány v následující podkapitolách.

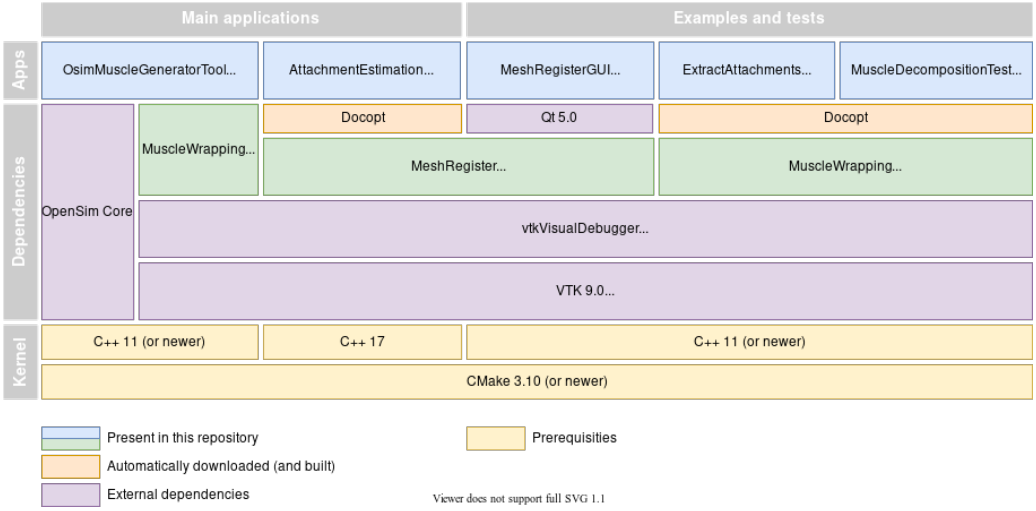
## 3.1 Projekt Muscle Wrapping 2.0

Muscle Wrapping 2.0 [1] je velmi obsáhlým projektem a na první pohled nemusí být zřejmé, jakým způsobem bude v této práci využit. Dle jeho architektury ho lze rozdělit do dvou odvětví: *dodatky do programu OpenSim* a *testovací nástroje*. V prvním odvětví se nachází aplikace vytvořené principiálně jako funkčnostní dodatky do nástroje zvaného OpenSim[1]. Vzhledem k faktu, že zmíněný nástroj není předmětem této bakalářské práce, nebude společně s jeho dodatky, vyskytujícími se v tomto projektu, dále v této práci zmiňován. Z toho tedy vyplývá, že se veškeré další nástroje, koncepčně spjaté s touto prací, nacházejí ve druhé zmíněné kategorii. Z té bude konkrétně využit nástroj **MuscleDecomposition**.

### 3.1.1 Popis nástroje MuscleDecomposition

MuscleDecomposition je nástroj vyvíjený v jazyce C++ sloužící k dekompozici existujícího modelu svalu na jednotlivá svalová vlákna se zřetelem na jeho úponové oblasti (viz obrázek 2.4). Je navržen ve formě konzolové aplikace se třemi povinnými vstupními parametry: *soubor reprezentující svalovou tkáň, soubor reprezentující počáteční úponovou oblast a soubor reprezentující koncovou úponovou oblast.* Kritérium pro všechny zmíněné parametry, jež je dáno autory tohoto nástroje, je ryze textový formát těchto souborů. Příkladem standardních formátů splňující toto kritérium, a které

[1] `https://simtk.org/projects/opensim`

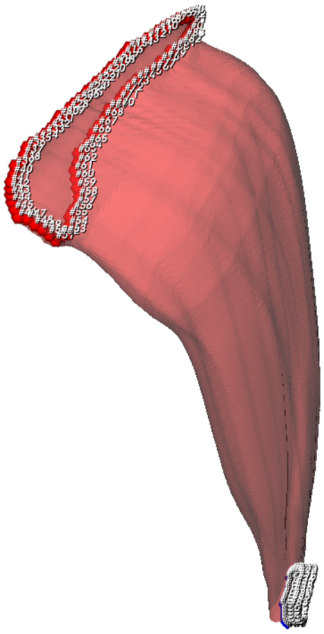



Obrázek 3.1: Struktura projektu Muscle Wrapping 2.0 (převzato z [1])

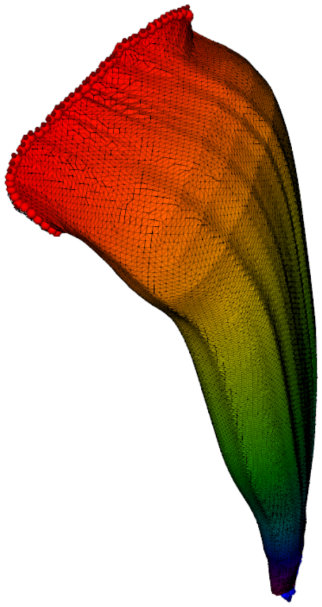
jsou podporovány tímto nástrojem, jsou například STL [2], OBJ [3] či VTK [4]. Na začátku takového souboru se musí vyskytovat hlavička, popisující konkrétní vnitřní strukturu. U standardních formátů za hlavičkou následují řádky, každý reprezentující bod jako trojici souřadnic, respektive k jednotlivým osám trojdimenzionálního prostoru. Kromě toho nabízí nástroj možnost specifikace následujících volitelných parametrů, které určují pak výslednou „kvalitu" svalového modelu.

- *fibres* - počet svalových vláken dekomponovaného modelu. Tento parametr udává, z kolika svalových vláken se výsledný model bude skládat. Jedno vlákno je následně reprezentování jakožto lomenná čára,
- *resolution* - rozlišení jednoho svalového vlákna. Tento atribut udává, jak moc členité samotné vlákno bude. Čím větší bude jeho hodnota, tím více segmentů lomenné čáry bude tvořit jedno vlákno,
- *visualization mode*:
  - 0: žádná vizualizace,
  - 1: vizualizace modelu po dokončení dekompozice pomocí VTK (Visualization toolkit),
  - 2: vizualizace jednotlivých kroků dekompozice a konečného modelu pomocí VTK (určeno pro ladění).

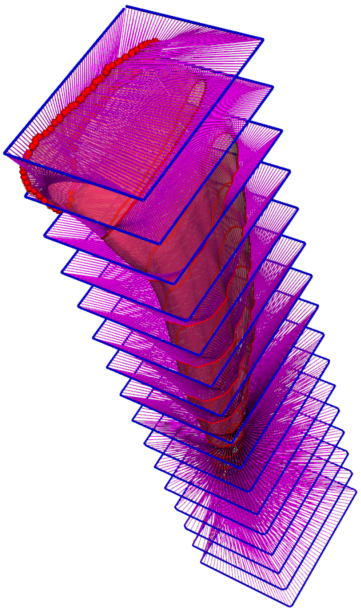
Vliv parametrů *fibres a resolution* je zřejmý, nicméně z hlediska vizualizace nemusí být hned jasné, oč se jedná. Z toho důvodu jsou na obrázku 3.2 zobrazeny jednotlivé kroky vizualizace.



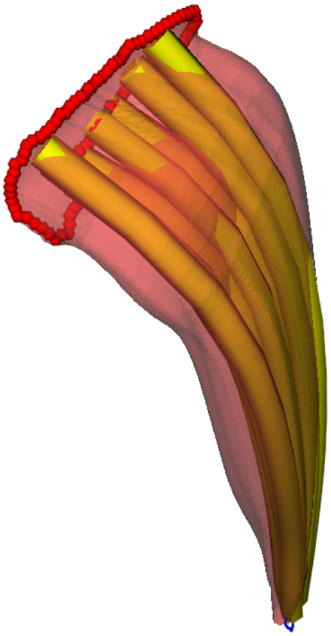
(a) Načtený model



(b) Skalárová reprezentace modelu



(c) Vizualizace šablony pro transformaci izolinií na síť svalu



(d) Dekomponovaný model svalu

Obrázek 3.2: Posloupnost vizualizace při módu ladění (2) (při základní vizualizaci (1) je zobrazen pouze výsledný model)

Konečným výstupem je opět soubor v jednom z podporovaných formátů (dle koncovky v požadovaném názvu), který reprezentuje samotný sval jakožto množinu svalových vláken. Lze si tedy povšimnout, že nastavení těchto volitelných parametrů je stěžejní, zejména při posuzování, zda-li model odpovídá jeho reálné předloze. Mohou tedy nastat případy, kdy tyto parametry **znehodnotí reprezentativnost modelu**, což je nutné mít na paměti. Příkladem takové situace je nastavení malého počtu vláken, tudíž by se v extrémním případě mohl sval skládat z pouze jednoho vlákna, což absolutně neodpovídá reálnému svalu. V situaci, kdy nejsou tyto parametry uživatelsky specifikovány, má nástroj sadu svých přednastavených tak, aby výsledný model výrazným způsobem nedeformoval.

## 3.2 Popis nástroje MyoGenerator

MyoGenerator [6] je externí skript vytvořený Dr. Herbst sloužící k automatickému vytvoření modelu svalu v závislosti na předem definovaných úponových oblastech. Pro vytvoření modelu svalu je nejprve nutné vytvořit speciální, takzvanou **NURBS** křivku. NURBS, neboli *Non-uniform rational basis spline*, jsou křivky sloužící pro popsání tvaru objektů ve 3D scéně. Na vyobrazeném konkrétním případě (obrázek 3.3) se jedná o jednu křivku, jež je spojnicí těžišť úponových oblastí protínající 5 vygenerovaných bodů. Tyto body jsou generovány na normále mezi těmito těžišti vždy v pětině jejich vzdálenosti. Podél této křivky je pak vytvořen objemový model svalu, který tvarově kopíruje počáteční úponovou oblast.

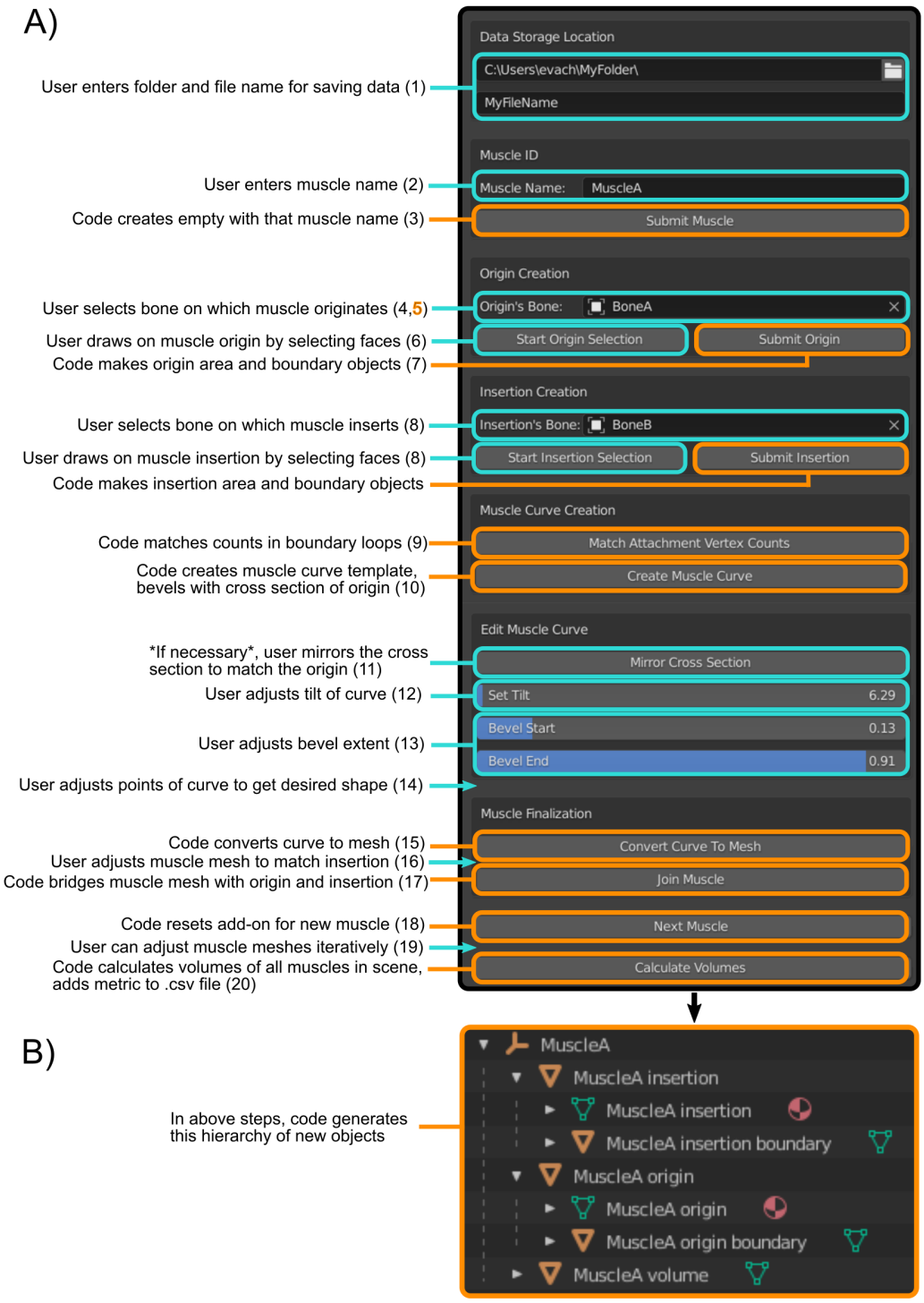
Obrázek 3.3: Ukázka automaticky vygenerovaného svalu pomocí nástroje *MyoGenerator*. Žlutá lomenná čára vyznačuje osu svalu mezi úponovými oblastmi

Před vytvořením takového modelu je ovšem nejdříve nutné specifikovat následující parametry pro nástroj (číselné označení odpovídá obrázku 3.4).

- Cesta do adresáře, kam budou uloženy datové soboury reprezentující vygenerovaný sval (1),
- Název svalu (2),
- Výběr konkrétního modelu kosti, ke kterému je upnuta počáteční úponová oblast svalu (4),
- Výběr konkrétního modelu kosti, ke kterému je upnuta koncová úponová oblast svalu (8).

Po samotném vygenerování je žádoucí model upravit, zejména kvůli koncové úponové oblasti, která se v některých případech může značně lišit od koncové úpony na modelu svalu (viz obrázek 3.3, vlevo). K tomu lze využít parametry *bevel* a *tilt* (obrázek 3.4, body 12, 13, 14).

Nutno podotknout, že se síť takto vygenerovaného modelu skládá z **polygonů**, nikoliv trojúhelníků. Ve verzi 1.1 nástroje *MuscleDecomposition* je vstupní model svalu automaticky převeden do trojúhelníkové reprezentace, avšak pro starší verzi by bylo nutné tyto modely převést manuálně.




Obrázek 3.4: A) Grafické rozhraní nástroje MyoGenerator [6], B) Hierarchie modelu svalu po zpracování

# 4 Aplikace Blender a její možnost rozšiřování o další funkcionalitu

## 4.1 Seznámení s aplikací

Jelikož je Blender primárním softwarem pro modelování svalů používaný Dr. Herbst, je žádoucí se jím zabývat detailněji. Blender je *open-source*, neboli volně dostupný, software sloužící pro modelování a tvorbu trojrozměrných počítačových modelů, případně jejich animací. Jelikož byla aplikace vytvořena pod licencí GNU/GPL[1], není striktní k jakýmkoliv rozšíření respektující dané podmínky. Pro jejich zakomponování do aplikace zde slouží tzv. *aplikační programovatelné rozhraní*, též označováno jako **API** [5]. Toto rozhraní poskytuje veškeré metody potřebné k manipulaci se základními prvky jednotlivých částí celé aplikace. Tyto metody umožňují následující interaktivitu.

- Manipulace s parametry, které může uživatel upravovat prostřednictvím vestavěného uživatelského rozhraní,
- Změna uživatelského nastavení, klávesnicových zkratek a témat,
- Spuštění a práce se všemi nástroji, které jsou pro uživatele dostupné v rámci aplikace,
- Vytvářet nové prvky uživatelské rozhraní, jako například menu, panely či dialogová okna,
- Vytvářet úplně nové nástroje s podporou interaktivity s uživatelem,
- Vytvářet nové nástroje pro vykreslování scény,
- Manipulace s parametry prvků dané scény,
- Definovat nová nastavení pro parametry či data,
- Vykreslovat a přidávat objekty do dané scény.

## 4.2 Rozšiřování funkcionality

Blender API nabízí rozšiřitelnost pomocí programovácího jazyka **Python**. Python se řadí mezi vysokoúrovňové skriptovací jazyky a dnes je jedním z

[1] `https://www.blender.org/about/license/`

nejrozšířenějších programovacích jazyků na světě. Python, stejně jako Blender, je vyvíjen jako *open-source* software. To umožňuje externím vývojářům dodávat do jazyka dodatečnou funkčnost ve formě takzvaných **modulů**, které jsou, vzhledem k licenci, taktéž volně dostupné. Moduly obsahují definice nových funkcí a datových struktur.

Je tedy zřejmé, že instance tohoto jazyka je nutnou prerikvizitou pro rožšiřování Blenderu o dodatečnou funkcionalitu. Rozšíření lze vytvářet v integrovaném vývojovém prostředí Blenderu, případně v externím vývojovém prostředí. Oba způsoby mají své výhody i nevýhody. První způsob umožňuje programátorovi přímé spuštění skriptu v aplikaci, z něhož lze v reálném čase pozorovat výsledky. Na druhou stranu, vestavěné vývojové prostředí nenabízí některé užitečné funkce těch externích, jako například automatické doplňování syntaxí, nebo jejich kontrolu. Při použití externích vývojových nástrojů je však nutné výsledné skripty spustit v integrovaném prostředí, a nebo je zakomponovat do interního adresáře aplikace, odkud jsou skripty načteny. Pro vývoj rozšíření poskytuje Blender modul **bpy**, který obsahuje řadu dalších „podmodulů", jež jsou rozděleny vzhledem k využití následovně.

- *data* - konkrétní datové záznamy právě otevřeného projektu, jako například: existující objekty, počet scén,
- *types* - základní typy objektů a primitiv,
- *context* - parametry konkrétního objektu existujícího ve scéně, jež byl uživatelem vybrán, například po kliknutí myši,
- *ops* - reprezentují množinu akcí, jež lze nad objekty provádět, například transformace (rotace, translace), skrytí, ...,
- *utils* - nástroje sloužící pro manipulaci s aplikací, nikoliv se samotnými scénami, kupříkladu načtení souboru, nastavení výchozí cesty do adresáře, ...,
- *path* - slouží k manipulaci s absolutními a relativními cestami v souborovém systému,
- *app* - obsahuje parametry aplikace, jež jsou v základu neměnné,
- *props* - možnosti přidávat dodatečné atributy ke všem základním primitivům.

## 4.3 Dědičnost v Pythonu

Python, na rozdíl například od Javy, zavádí model vícenásobného dědění. Díky tomu může libovolá třída dědit od více nezávislých tříd. Taková třída

je v Pythonu označována pojmem **podtřída**. Do verze Pythonu 2.6 byly rodičovské třídy označovány jako **bázové** a jejich potomci **nemuseli** implementovat všechny deklarované metody. Vyšších verzí se však objevil nový archetyp označován jako **abstraktní bázová třída**, která nutnost implementace **přikazuje**, stejně jako tomu je například u rozhraní v Javě. Tento koncept bude důležitý zejména dále (sekce 6.2.1)

# 5 Návrh rozšíření pro Blender umožňující dekomponovat svaly ve scéně

## 5.1 Specifikace požadavků

Dále zmíněné požadavky byly specifikovány na online schůzce s Dr. Herbst ze dne 13.12.2021. Konkrétní požadavky na rozšíření jsou následující.

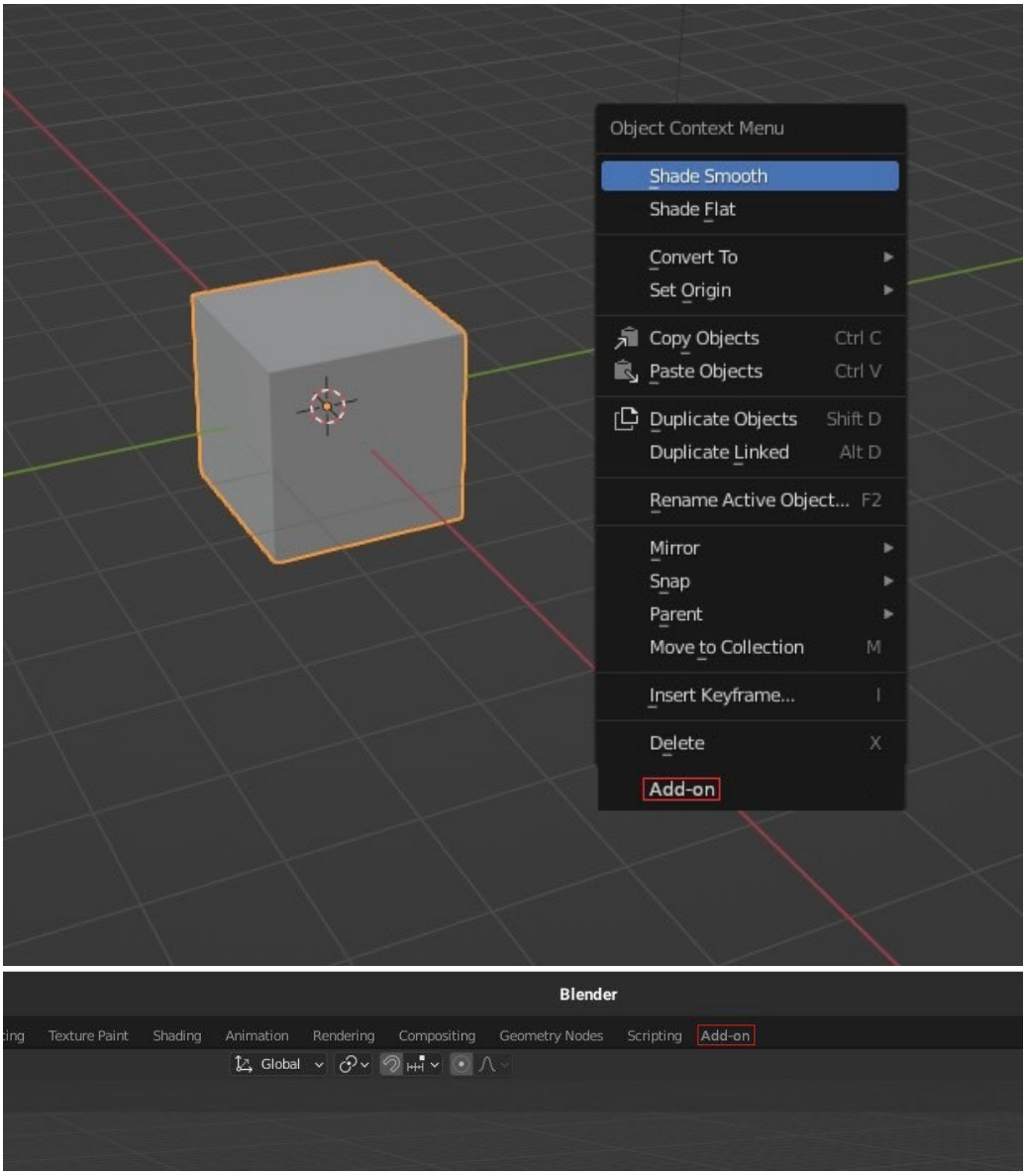
- Rozšíření musí být zakomponováno do interního grafického rozhraní Blenderu (konkrétní způsob nespecifikován),
- V rámci rozšíření musí být implementováno grafické rozhraní pro nastavení parametrů a ovládání,
- Rozšíření bude umožňovat dekompozici svalů (vytvořených nástrojem *MyoGenerator*) ve scéně (pomocí *MuscleDecompositionTest*),
- Rozšíření bude umožňovat uživatelskou instalaci obsahující veškeré nutné prerekvizity.

Následující funkce nejsou povinné k implementaci, nýbrž jsou spíše v kategorii *„nice to have”*:

- Možnost dekompozice více svalů ve scéně,
- Detekce protínajících se svalů před jejich dekompozicí.

## 5.2 Umístění rozšíření v grafickém rozhraní Blenderu

V rámci API Blenderu lze přidávat další ovládací prvky do libovolné části jeho grafického rozhraní. Prvotní návrh se bude snažit o implementace ovládání v rámci horní nástrojové lišty. Konkrétní možnosti umístění dodatku jsou vyznačeny na obrázku 5.1.

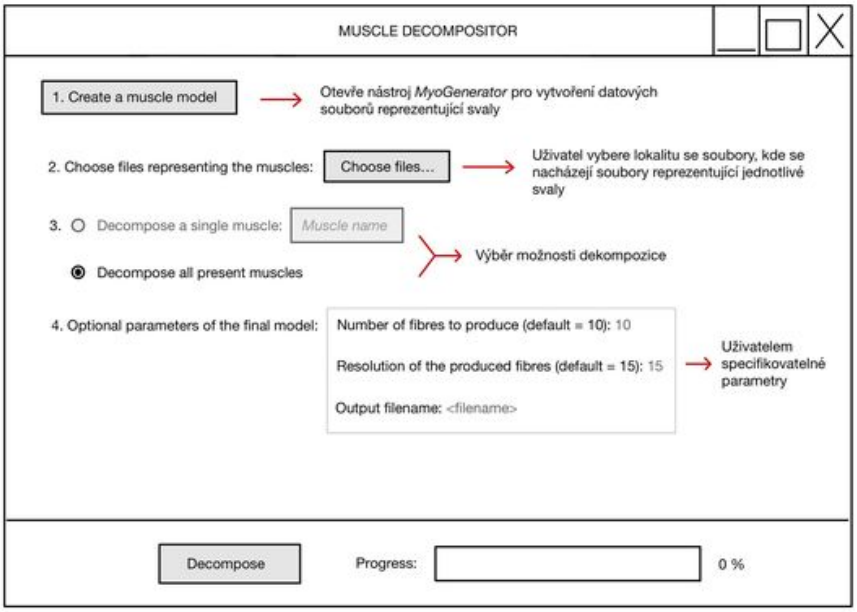



Obrázek 5.1: Návrh umístění rozšíření v interním grafickém rozhraní v Blenderu. Pozice je vždy vyznačena červeným zvýrazněním.

## 5.3 Využití externích nástrojů

Z kapitoly 3.1.1 vyplývá, že *MuscleDecomposition* je tedy samostatný nástroj, nicméně to implicitně neznamená, že pro účely této práce musí být nutně využit. Použití tohoto nástroje je samozřejmě možné, avšak jeho využití s sebou přináší jeden zásadní problém. Tento nástroj je v podstatě spustitelným souborem, což znamená, že musí být pro konkrétní platformu takzvaně **sestaven**. Z toho plyne, že pro různé platformy musí existovat příslušná sestavená verze tohoto nástroje. Tato vlastnost je absolutně nevhodná pro multiplatformní řešení, avšak možná. Nicméně se této „překážce" dá vyhnout. Jelikož je projekt *MuscleWrapping 2.0* tzv. *open-source*, jsou veškeré zdrojové kódy plně dostupné k vlastnímu použití. Po prozkoumání „hlavního" zdrojového souboru *MainApp.cpp* lze vyčíst následující chování. Samotný nástroj nejdříve „přebere" uživatelské parametry, poté je vytvořena nová instance třídy *vtkMAFMuscleDecompositionKukacka* v rámci níž jsou volány příslušné „settery" pro předání zadaných parametrů této instanci. Pomocí metod *SetOriginArea* a *SetInsertionArea* jsou „označeny" jednotlivé úponové oblasti. Poté jsou úponové oblasti promítnuty pomocí metody *addOIPoints* na síť svalu a pomocí metody *AddOrUpdateLines* vygenerovány jednotlivá svalová vlákna. Na základě této analýzy lze tedy usoudit, že není pro proces dekompozice potřebný celý sestavený nástroj. Python totiž nabízí možnost rozšíření zdrojového kódu o funkčnosti implementovaných v C/C++ knihovnách pomocí modulu **ctypes**. To tedy umožňuje provázání potřebných zdrojových souborů v rámci MuscleDecomposition nástroje a dodatku, jež je cílem této práce. Problém tohoto přístupu spočívá v potřebě mnohem detailnější analýzy interní implementace generování svalových vláken, společně s přesnou reprezentací datových struktur a prerekvizit. Vzhledem k právě řečenému bude v návrhu řešení předpokládáno využití prvního zmíněného přístupu, tedy využití již sestavené verze nástroje. Toto rozhodnutí bylo provedeno zejména na základě jednoduchosti a rychlosti řešení, aby mohl být výsledný dodatek co nejdříve dostupný pro Dr. Herbst.

Z pohledu nástroje *MyoGenerator* nejsou kladena žádná podobná omezení, zejména díky faktu, že je nástroj vyvíjen konkrétně pro Blender a jeho interního použití, a proto nevytváří žádná podobná omezení z hlediska využití. Nicméně, v rámci ryzí fuknčnosti dodatku nebude považován za přímou prerekvizitu, avšak je jeho využití doporučeno. Výsledný dodatek bude v prvotní verzi implementován nezávisle na tomto nástroji, nicméně bude vyvíjen tak, aby mohl být v pozdějších verzích přímou součástí tohoto nástroje, bude-li to žádoucí ze strany Dr. Herbst.




Obrázek 5.2: Návrh grafického rozhraní rozšíření do Blenderu

## 5.4 Funkcionalita a návrh grafického rozhraní

Vzhledem ke specifikovaným požadavkům rozšíření je vhodné, aby jej konkrétní návrh grafického rozhraní (obrázek 5.2) respektoval. Po otevření nástroje se otevře nové okno s možností nastavení veškerých prerekvizit pomocí využitých externích nástrojů (sekce 3.1.1 a 3.2). Nejdříve je nutné, aby v rámci dodatku bylo možné specifikovat umístění, se kterým bude moci uživatel pracovat, ideálně pomocí nějakého dialogu, nejlépe s možností úprav samotné cesty. Následně je žádoucí, aby existovala možnost úprav parametrů pro dekompozici (viz kapitola 3.1.1). Vzhledem k tomu, že se jedná zejména o číselné parametry, je vhodné hodnoty těchto parametrů omezit dle omezení nástroje *MuscleDecomposition*. Po specifikaci těchto parametrů bude umožněno spustit proces exportování stisknutím příslušného tlačítka, s případným informováním o chybovém hlášení pomocí dialogů (v závislosti na nabídce elementů v Blenderu). Stisknutí tohoto tlačítka na pozadí provede export vybraného modelu svalu, který následně dekomponuje na jednotlivá svalová vlákna a výsledný model importuje zpět do Blenderu. Uživatel by měl taktéž být informován o dokončeném procesu.

# 6 Konkrétní implementace návrhu

## 6.1 Obecná struktura dodatků Blenderu

Aby byl dodatek funkční pro použití v Blenderu, je nutné, aby se v jeho zdrojovém kódu vyskytovaly následující definice: atributu *bl_info*, metody *register()* a metody *unregister()*. Atribut *bl_info* je v pojetí jazyka Python tzv. *dictionary*. Tento datový typ se využívá k ukládání informací ve formátu „klíč:hodnota". V tomto atributu jsou dále definována *metadata*[1] dodatku obsahující základní informace. Konkrétní příklad vytvoření metadat pro rozšíření je uveden níže.

```
  ### Příklad vytvoření metadat k dodatku do Blenderu ###

  bl_info = {
  "name": "MyoGenerator Exporter",
  "description": "Addon used for
                  exporting muscle models
                  created by the Myogenerator,
                  decomposing them into muscle
                  fibers using 3rd party software
                  and then importing
                  them back to Blender",
  "author": "Jan Heres",
  "blender": (2, 80, 0),
  "version": (0, 0, 1),
  "category": "Export",
}
```

Metoda *register()* je volána ve chvíli, kdy je dodatek nainstalován do Blenderu, či znovu aktivován, a slouží k registraci veškerých elementů (tříd) grafického rozhraní, které jsou v dodatku využívány (například Panel, nebo Operator, podrobněji dále). Proces registrace tříd z využitých modulů je zaveden kvůli možnosti aktivace, respektive deaktivace, dodatků za běhu.

[1] `https://wiki.blender.org/wiki/Process/Addons/Guidelines/metainfo`

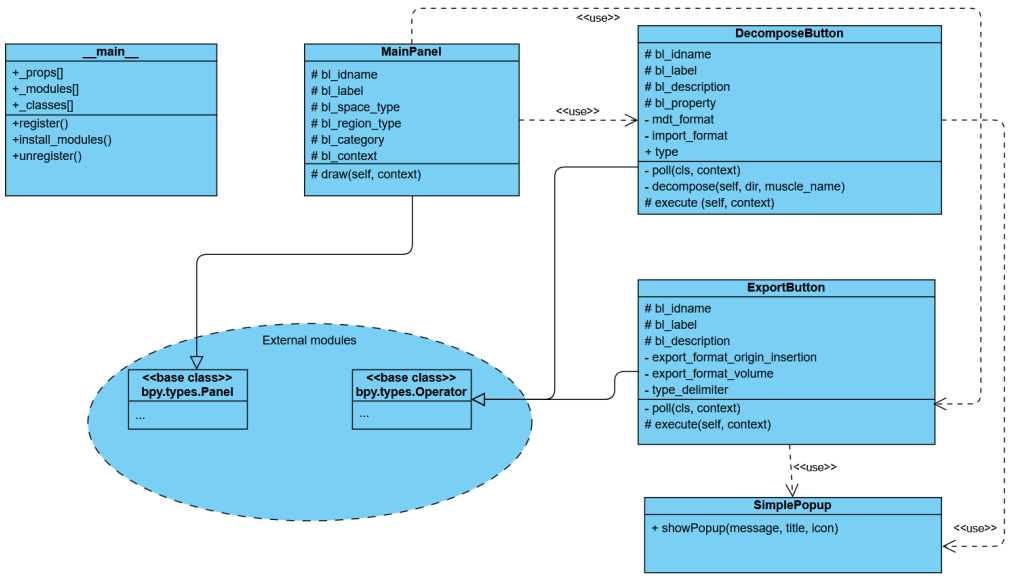
Kdyby tento proces nebyl přítomen, musely by se při každé aktivaci/deaktivaci instalovat i jednotlivé moduly obsahující právě používané třídy, což je časově náročný proces. Samotná registrace tříd se provádí voláním metody *register_class(class)* s příslušným parametrem. Mimo tříd je nutná registrace veškerých nově zavedených *properties*. *Properties* jsou nově definované globální proměnné, jež nejsou součástí Blenderu, nýbrž vytvořené v samotném kódu dodatku. Mezi tyto vlastnosti lze zařadit například nově definované cesty ke specifickým souborům, nová vlastnost právě vytvořeného modelu a další. Metoda *unregister()* je pak logickým opakem k procesu registrace. Je zřejmé, že je nutné **všechny** zaregistrované metody a nově definované vlastnosti umět „odregistrovat".

```
### Příklad metod register() a unregister() ###
_classes = [
    MainPanel,
    ExportButton,
    DecomposeButton
]

def register():
  # Class registration
  for class in _classes:
    register_class(class)

  # Property registration
  bpy.types.Scene.output_path = bpy.props.StringProperty(...)

def unregister():
  # Unregistering classes
  for class in _classes:
    unregister_class(class)

  # Unregistering Property
  del(bpy.types.Scene.output_path)

if __name__ == "__main__":
  register()
```

Pokud jsou v požadovaném skriptu zakomponovány výše zmíněné atributy a metody, jedná se o plně validní dodatek do Blenderu i přes fakt, že sám o sobě neobsahuje žádnou další funkcionalitu. Pokud je zdrojový kód dodatku dělen do více souborů, běžně se ten, který obsahuje výše zmíněné metody, pojmenovává jako **„___init___.py"**. Tento název je vnitřně rozpoznáván jakožto vstupní bod komplexnější aplikace, což platí obecně pro vývoj v Pythonu, nikoliv jen pro samotný Blender. Jedná se o jakousi analogii u souborů v Javě, kdy vstupním bodem aplikace považujeme právě ten, ve kterém se nachází metoda *„main"*.

## 6.2 Konkrétní struktura dodatku do Blenderu

V následujících podkapitolách bude podrobně představena struktura vytvořeného dodatku. Na obrázku 6.1 je vyznačena vazba mezi jednotlivými komponentami.




Obrázek 6.1: UML diagram dodatku

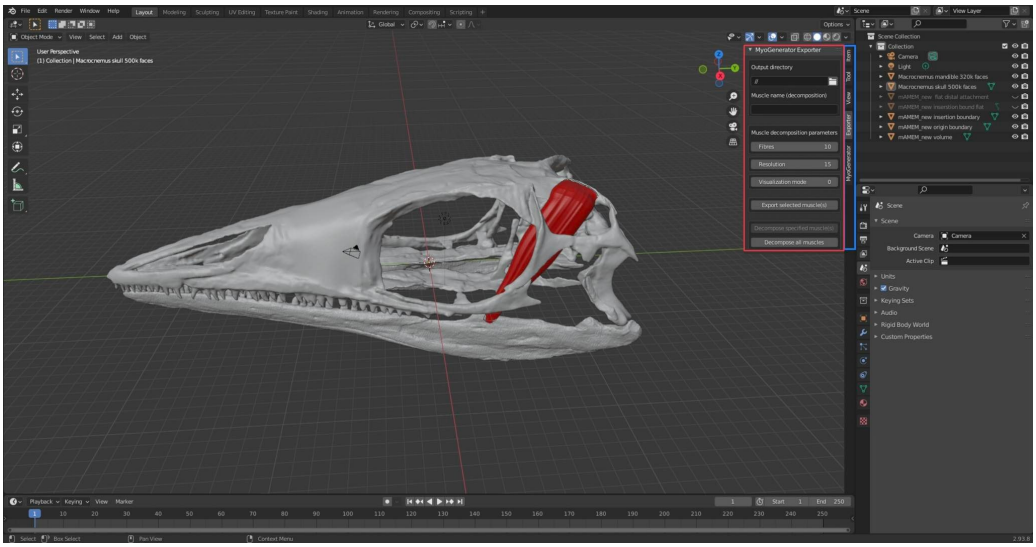
### 6.2.1 Vstupní bod programu

Vstupním bodem programu je zmíněný soubor **„___init___.py"**. V tomto souboru se nachází veškeré zmíněné prerekvizity. Nejdříve tedy proběhne standardní proces registrace všech využívaných tříd a nově definovaných

atributů. Jelikož bude dále odkazováno na právě nově definované atributy, jsou vyjmenovány v seznamu níže.

- *output_path* - *String* reprezentující cestu do adresáře, kam se mají ukládat vyexportované soubory z Blenderu,
- *muscle_name* - *String* reprezentující název, kterým má být dekomponovaný sval pojmenován,
- *export_fibres, export_resolution, export_visualize* - *Integer* reprezentující příslušný parametr dekompozice.

Jakmile je veškerá výše popsaná činnost provedena, je dodatek aktivní. V původním návrhu rozšíření bylo předpokládáno umístění dodatku do **horní** nástrojové lišty (viz obrázek 5.1). Nicméně veškeré nově přidané dodatky, není-li definováno jinak, jsou běžně umisťovány do **postranní** lišty (viz obrázek 6.2), jejichž zobrazení, respektive skrytí, se v základním nastavení Blenderu provádí klávesou „N". V této liště se, mimo jiné, vyskytuje i samotný *MyoGenerator* a z důvodu lepší provázanosti v ovládání mezi oběma dodatky bylo z původního návrhu ustoupeno a finální umístění dodatku přesunuto (viz obrázek 6.2).



Obrázek 6.2: Finální umístění v dodatku v rámci scény a rozhraní blenderu (červeně - samotný dodatek, modře - postranní lišta)

### 6.2.2 Hlavní panel dodatku

Jakýkoliv element grafického rozhraní v Blenderu je instancí libovolné třídy, která dědí od **bázové** třídy **Struct**. V ní jsou deklarovány základní

třídní atributy (identifikátor, label, ...) a s nimi spjaté metody (gettery, settery, ...). V rámci lepší čitelnosti a orientaci v API Blenderu byly vytvořeny podtřídy pro jednotlivé základní typy (Panel, Operator, ...). Nicméně i tyto podtřídy obsahují pouze **deklarace** atributů a metod, tudíž je pro jejich využití nutné vytvoření vlastních tříd.

První takovou třídou bude třída **MainPanel**. Ta dědí od třídy **Panel**[1] a slouží k vykreslení „podokna" v rámci scény, ve které se budou nacházet veškeré prvky spjaté s tímto dodatkem. Na obrázku 6.3 reprezentuje obdélníkové „okno", v němž jsou umístěny další elementy (tlačítka, text boxy, ...). V rámci této třídy je nutné inicializovat některé ze zděděných atributů (začínající předponou *bl_*), které mají vliv na chování samotného Panelu:

- *bl_idname* - identifikátor elementu v rámci scény,
- *bl_label* - popisek panelu; umístěn v záhlaví,
- *bl_space_type* - okno, ve které bude panel zobrazen (scéna s 3D objekty/image editor/console/...),
- *bl_context* - mód, ve kterém bude panel aktivní (Object mode/Select mode/...),

Po inicializaci těchto atributů je nutné deklarovat tělo metody **draw(self, context)**. V rámci této metody by mělo být popsáno jak, co a kde má být v panelu vykresleno. Parametr *self* je odkazem na konkrétní instanci („na sebe sama") a *context* je konkrétní část scény, ve které byl panel vykreslen (koresponduje s možnostmi parametru bl_space_types). Jako první krok je nutné definovat rozložení prvků v rámci celého panelu. K tomu je v rámci Blender API k dispozici několik základních rozložení (*layoutů - BorderLayout, BoxLayout, ...*, konkrétněji v [5]). V tomto konkrétním případě byl u panelu zvolen takzvaný *BoxLayout*, který umožňuje **dynamicky** rozdělovat panel na jednotlivé sloupce a řádky. Taktéž bylo toto rozvržení zvoleno kvůli jednotnosti rozložení grafických elementů mezi tímto dodatkem a využívaným *MyoGenerator* nástrojem. Uživatel, který je zvyklý na ovládání a vzhled prvků využitých ve zmíněném nástroji, nebude mít problém s porozuměním ovládání tohoto dodatku. Pro vytvoření řádku slouží volání metody *layout.box()*, která jako návratový typ vrátí „instanci" nově vytvořeného řádku. Tato instance nabízí několik různých metod pro umístění základních elementů (například *label, prop, operator, ...*). Veškeré textové popisky jsou označovány jako *labels*. Textová pole nabízející uživatelův vstup spadají do

[1] `https://docs.blender.org/api/current/bpy.types.Panel.html`

kategorie *props*. Ve valné většině případů je žádoucí si uživatelův vstup nějakým způsobem „zapamatovat", a tudíž je vždy spjatý s nějakým atributem (=*property*, viz sekce 6.2.1), která v rámci Blenderu určuje i styl a vlastnosti samotného textového pole. Pokud je tedy textové pole svázáno s atributem typu *Integer*, Blender automaticky povoluje jenom číselné vstupy. Pokud je svázáno s atributem typu *String*, je situace poněkud komplikovanější a kontrolu vstupů musí provést programátor, nebo alespoň pomocí doplňujících vlastností v rámci tvorby atributu specifikovat, jaký formát by měl být očekáván (zda-li se jedná o cestu k souboru, generický název, ...). Co se týče tlačítek, ta spadají do kategorie *operators*. U nich lze, mimo jiné, specifikovat atribut *enabled*, do kterého lze vložit logickou podmínku, kdy je tlačítko stisknutelné (například až po vyplnění určitého textového pole).

```
### Výňatek metody draw() z dodatku ###
def draw(self, context):
  layout = self.layout
  box = layout.box()

  # Output path label
  row = box.row()
  row.label(text=strings["Label_output_path_text"])

  # Output directory path
  row = box.row()
  row.prop(context.scene, "output_path", text="")

  ...

  # Parameter "number of fibres"
  row = box.row()
  row.prop(context.scene, "export_fibres", text="Fibres")

  ...

  # Export button
  row = box.row()
  col = row.column()
  col.operator(ExportButton.bl_idname,
               text=strings["ExportButton_text"])
  col.enabled = (context.scene.output_path != "")
```


Obrázek 6.3: Finální rozhraní dodatku (hlavní panel)

### 6.2.3 Operátory

Stejně tak jako tomu bylo u Panelu, i pro tlačítka existuje speciální **bázová** třída zvaná *Operator*. V rámci ní je, mimo jiné, deklarována metoda **execute(self, context)**, která je volána ve chvíli, kdy je operátor „aktivován" (v případě tlačítka se jedná o jeho stisknutí). Třídní atributy mají stejný význam, jako tomu bylo u výše zmíněného panelu a je nutné některé z nich definovat (viz výše). Oproti metodě *draw()* je zde nutné navracet hodnotu indikující, zda-li proběhla úspěšně, stále probíhá, nebo nastala chyba. Návratová hodnota by tedy měla být ve formátu **'FINISHED'**, případně **'RUNNING'**, nebo **'CANCELLED'**. Toto je nutné implementovat, jelikož v Blenderu funguje koncept vracení poslední provedené akce, anglicky *undo*, kdy se do interního žurnálu akcí zapisují **pouze** ty, které skončily hodnotou **'FINISHED'**.

Bázová třída taktéž nabízí metodu **invoke(self, context, event)** sloužící k „aktivaci" daného operátoru (vyvolání metody *execute()*) na základě nějaké specifikované události (parametr *event*).

## 6.3 Popis nastavitelných parametrů

V rámci dodatku byla, mimo jiné, implementována možnost úpravy parametrů pro nástroj *MuscleDecomposition* (viz kapitola 3.1.1). Kromě nich byly zavedeny i další upravitelné parametry, zejména pro odlišení jednotlivých modelů a uživatelské „pohodlí". Jedná se o parametry *output directory, muscle name* a *muscle decomposition parameters*.

**Output directory** První textová oblast, ve smyslu procházení panelu dodatku shora dolů, reprezentuje cestu do uživatelem vybraného adresáře. Do tohoto adresáře budou, po stisknutí tlačítka, exportovány všechny vybrané modely. Taktéž je to adresář, ze kterého budou modely importovány zpět do Blenderu. Tuto cestu lze kdykoliv měnit mezi jednotlivými činnostmi (exportování, dekompozice). Veškeré cesty **musí** být vztaženy k adresáři, v němž se nachází právě otevřený soubor se scénou (implementace Blenderu).

**Muscle name** Tento parametr je využíván pouze pro proces dekompozice (stisk tlačítek *Decompose ...*). Adresář specifikovaný parametrem „*output directory*" je poté prohledáván pro shodu jména každého souboru, a pokud je tato shoda nalezena, bude tento sval dekomponován a importován zpět do Blenderu (podrobněji dále). Tento parametr slouží pouze pro specifikaci konkrétního modelu svalu určeného k dekompozici. Pro případ, kdy chce uživatel provést dekompozici více svalu, je určeno tlačítko *Decompose all muscle(s)* (podrobněji dále).

**Muscle decomposition parameters** Parametry *fibres, resolution a visualization mode* reprezentují stejně pojmenované parametry nástroje *MuscleDecompositionTest* (viz kapitola 3.1.1). Lze je tedy libovolně měnit, nicméně mají takzvané *soft* limity. Jelikož je volba těchto parametrů realizována pomocí „posuvníku", je vhodné nastavit příslušný limit, aby byl krok jednoho posuvu vhodně rozprostřen do celé šířky tohoto elementu. Avšak mohou nastat i situace, kdy je potřeba tyto limity překročit (například více vláken nebo větší rozlišení), k čemuž slouží manuální určení parametrů pomocí dvojkliku. V závislosti na dobrém rozprostření celého intervalu <min; soft-min> do celé šířky „posuvníku" byly v rámci testování parametry *fibres a resolution* shora omezeny číslem 100.

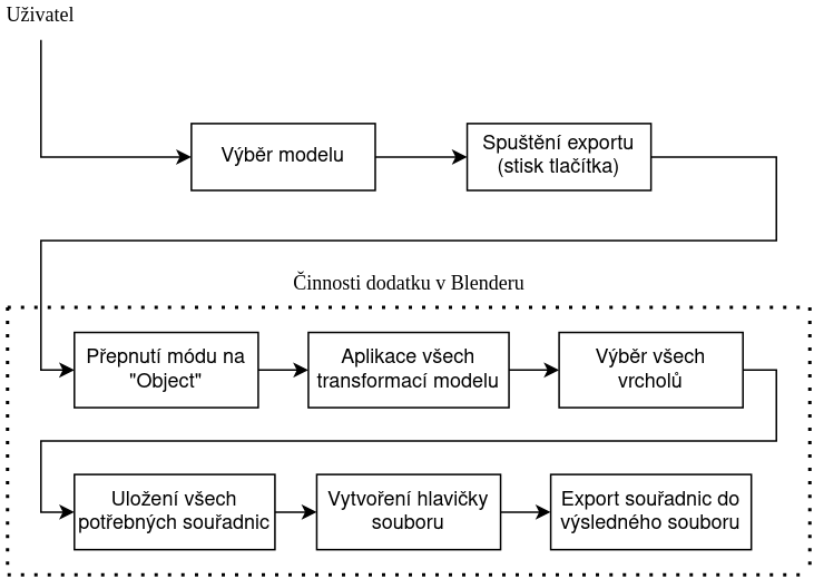
## 6.4 Proces exportování vybraných modelů

K exportování modelů mimo Blender do vnějšího adresáře (viz parametr *output_directory*) slouží tlačítko *„Export selected muscle(s)"*. Uživatel před jeho stisknutím musí vybrat **alespoň jeden** model ze všech přítomných. Pokud není vybrán žádný, tlačítko zůstává přístupné, nicméně po jeho stisknutí je zobrazeno chybové hlášení. Stejně tak může uživatel vybrat více libovolných modelů přítomných ve scéně. Aby tohle bylo možné, musí být dodrženy některé příslušné pojmenovávací konvence. Každý sval musí být v rámci scény pojmenován ve formátu **<název_svalu><mezera><origin | volume | insertion><mezera><boundary>** (část *boundary* pouze v případech, jedná-li se o úponovou oblast), aby mohly být jednotlivé části svalu odlišitelné (bude vidět dále). Pokud toto pojmenování nebude splněno, k exportování modelu dojde bez problémů, nicméně během procesu dekompozice nebude program umět spárovat správné modely svalů a jejich úponové oblasti (podrobněji dále). Jakmile uživatel vybere příslušné modely a klikne na příslušné tlačítko, vykoná se tělo metody *execute(self, context)* (viz kapitola 6.2.3) a provede se export modelu. Jednoduchý diagram stručně popisující jednotlivé probíhající činnosti je vyobrazen na obrázku 6.4.

V tomto diagramu je, mimo jiné, zmíněna činnost „Přepnutí módu na „Object"". Blender v rámci scény nabízí různé vizualizační módy modelů. Zde konkrétně budou využívány pouze „Object" a „Edit". V rámci módu *Obejct* jsou objekty vykresleny jakožto modely s příslušnou texturou, barvou, atp. V módu *Edit* je objekt vykreslen pomocí jednotlivých bodů, spojnic a sítí. Pro lepší představu je na obrázku 6.5 představen rozdíl mezi jednotlivými vizualizacemi v rámci jednoho modelu svalu.

### 6.4.1 Exportování úponových oblastí

K exportování úponových oblastí slouží metoda, která je definovaná v rámci samotného *operátoru*, *export_origin_insertion(self, obj, type)*. Parametr *self* je pouze odkazem na konkrétní instanci tlačítka, není pro účel samotné registrace důležitý, nicméně tam musí být, aby bylo možné odlišit více různých instancí jedné třídy. Parametr *obj* je konkrétní model příslušné úponové oblasti a *type* jen specifikuje, zda-li se jedná o počáteční či koncovou úponovou oblast. V rámci této metody je nejdříve nutné relativní cestu do externího adresáře (viz sekce 6.3) převést na cestu absolutní, jelikož Blender vztahuje veškeré atributy typu *path* k lokaci svých interních souborů. K tomu slouží převodní metoda *abspath()* v modulu **bpy.path**. Poté se **v rámci objektu, nikoliv samotné scény (!)** aplikují akce typu *OBJECT_MODE,*

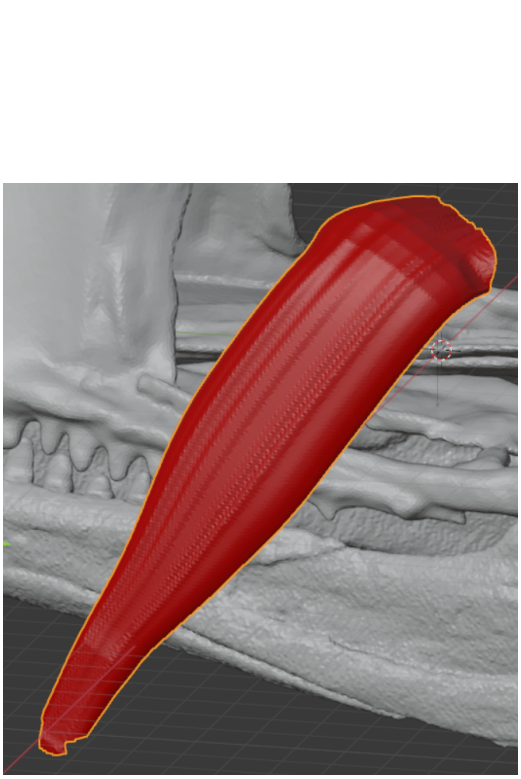



Obrázek 6.4: Diagram přehledu operací při procesu exportování

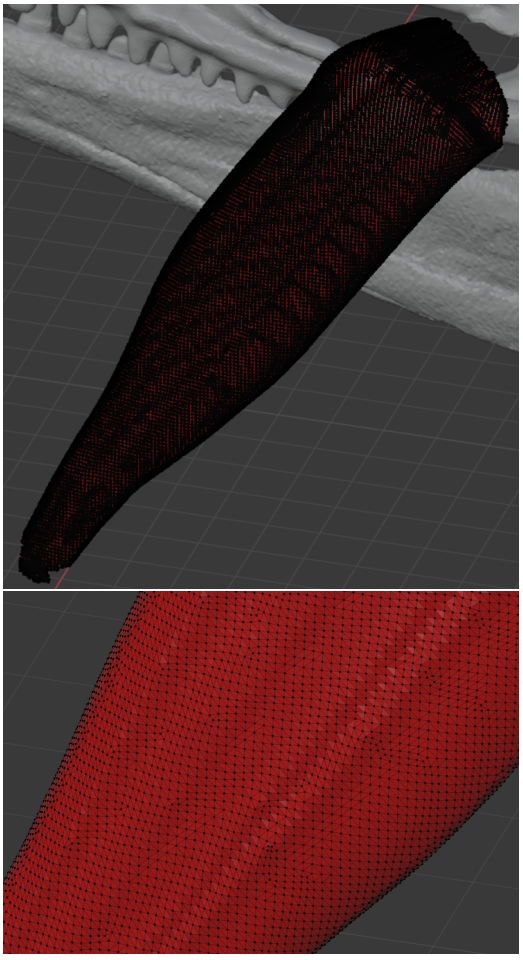
*ACTION_DESELECT, ACTION_SELECT_ALL*. Akce *OBJECT_MODE* reprezentuje přepnutí libovolného módu, ve kterém se uživatel nachází, zpět do zmíněného *Object* módu.

V rámci procesu by nadále mohla nastat situace, kdy by nedopatřením uživatele byla vybrána pouze část příslušné oblasti (jen některé vrcholy) a došlo by k exportu pouze těch vybraných. Proto jsou nejdřív všechny vrcholy daného modelu „odvybrány" a následně vybrány všechny příslušné vrcholy úponové oblasti. Akce *ACTION_SELECT* tedy reprezentuje výběr celého objektu (respektive všech jeho součástí), akorát na rozdíl od výběru objektu myší uživatele je tento výběr vykonán programově v rámci Blenderu. Akce *ACTION_DESELECT* je pak jejím logickým opakem.

Transformace modelu jsou další operací, která musí být zajištěna. Blender veškeré modely ve scéně vykresluje jakožto „kopii". Surová data (souřadnice jednotlivých vrcholů, hrany, ...) zůstávají nezměněna, dokud nejsou aplikovány všechny transformace provedé uživatelem. Pokud je scéna v Blenderu hojně zastoupená relativně komplexními modely, zpracování úprav modelů může být časově náročné, zejména z hlediska I/O operací, a proto jsou tyto operace zpracovávány postupně. V případě, že by se před dokončením operace uživatel snažil o export ještě nezpracovaného modelu, došlo by k exportu „chybných" dat. Pro eliminaci této situace je nutné na tato data aplikovat veškeré transformace samotného modelu před samotným exportem, k čemuž slouží volání metody *transform_apply()* v modulu **bpy.ops.object**. Volání



(a) „Object mode”



(b) „Edit mode”

Obrázek 6.5: Vizualizace modelu svalu v různých módech v rámci scény v Blenderu

této metody upřednostní specifikovaný objekt, jelikož se předpokládá, že s ním bude nadále pracováno.

```
### Výňatek metody export_origin_insertion() ###
...
# Apply all transforms before exporting
bpy.ops.object.transform_apply(location=True,
                               rotation=True,
                               scale=True)
...
```

Ve výše uvedeném výňatku je uveden praktický příklad aplikace zmíněných transformací objektu. V rámci parametrů metody si lze všimnout, že je v rámci API možnost aplikovat pouze některé typy transformací (translace, rotace či škálování).

Jakmile jsou veškeré výše zmíněné činnosti provedeny, nastává čas samotného exportu. V rámci vybraného modelu proběhne jednoduchý cyklus skrz všechny vrcholy, kde jsou vzaty jednotlivé souřadnice a zapsány do výsledného souboru. Příklad takto popsaného cyklu je uveden níže.

```
### Příklad cyklu pro exportování souřadnic vrcholů modelu ###
pointsList = []

for v in bm.verts:
if v.select:
    coords = str(tuple(v.co))
    coords = re.sub('[(),]', '', coords)
    pointsList.append(coords)
pointsCount = len(pointsList)

with open(output_file, "w") as of:
            header = str('# vtk DataFile Version 3.0\n
                          vtk output\n
                          ASCII\n
                          DATASET POLYDATA\n
                          POINTS %d float\n'
                          %pointsCount)
            of.write(header)
            for point in pointsList:
                of.write(point +'\n')
```

### 6.4.2 Exportování modelu svalu

Pro model svalu je proces exportování velice podobný, avšak mírně odlišný, kvůli následujícímu důvodu. Jelikož byl Dr. Herbst zvolen formát **STL** (z dále nespecifikovaných důvodů) pro export modelu svalu, dochází zde k odlišnosti ve zdrojovém kódu metody. Blender pro tento formát nabízí samostatnou metodu, nicméně je zde jeden důležitý aspekt. Jelikož má Blender pro formát STL natočenou soustavu souřadnic **odlišně** vůči ostatním formátům, je nutné v rámci parametru specifikovat, v jakém směru jsou osy natočeny. Praktický příklad této metody je uveden níže.

```
### Příklad exportu do formátu STL ###

# Export the volume as an stl mesh
  bpy.ops.export_mesh.stl(filepath=output_file,
                          check_existing=True,
                          filter_glob='*.stl',
                          use_selection=True,
                          global_scale=1.0,
                          use_scene_unit=False,
                          ascii=False,
                          use_mesh_modifiers=True,
                          batch_mode='OFF',
                          axis_forward='Y',
                          axis_up='Z')
```

## 6.5 Proces dekompozice modelu

V kapitole 5.3 byly uvedeny dva přístupy k možnosti dekompozice modelu svalu. Pro připomenutí - v konkrétním návrhu byl zvolen přístup pomocí využití již sestavené verze nástroje *MuscleDecomposition*. Možnosti dekompozice modelů pro uživatele jsou popsány v následujících kapitolách.

### 6.5.1 Dekompozice konkrétního modelu

K dekompozici konkrétního modelu je nutné specifikovat následující parametry(viz kapitola ): jméno modelu svalu (pomocí parametru *muscle name*) a adresář, ve kterém se soubor nachází (parametr *output directory*). Pokud není vybrán adresář, nebo specifikované jméno modelu, je tlačítko „*Decompose specified muscle*" znepřístupněno. Před dekompozicí je samozřejmě možné upravit příslušné parametry pro dekompozici.

Po uvedení parametrů a stisknutí tlačítka je nejdříve prohledán přislušný adresář. V rámci programu jsou hledány soubory „*nazev_modelu* origin.vtk", „*nazev_modelu* insertion.vtk" a „*nazev_modelu* volume.stl". Všechny zmíněné soubory jsou automaticky vytvářeny při procesu exportu v rámci dodatku. Pokud nebylo možné najít jeden z vyjmenovaných souborů, nemůže dekompozice proběhnout, proces je zastaven a uživatel informován chybovým hlášením. V tento moment je zřejmá nutnost dodržování jednotné konvence při pojmenovávání modelů v rámci celého procesu (viz 6.4). Pokud byly soubory nalezeny, je v rámci kódu dodatku spuštěn nástroj *MuscleDecomposition* přebírající všechny potřebné parametry. Příklad spuštění externího nástroje uveden níže.

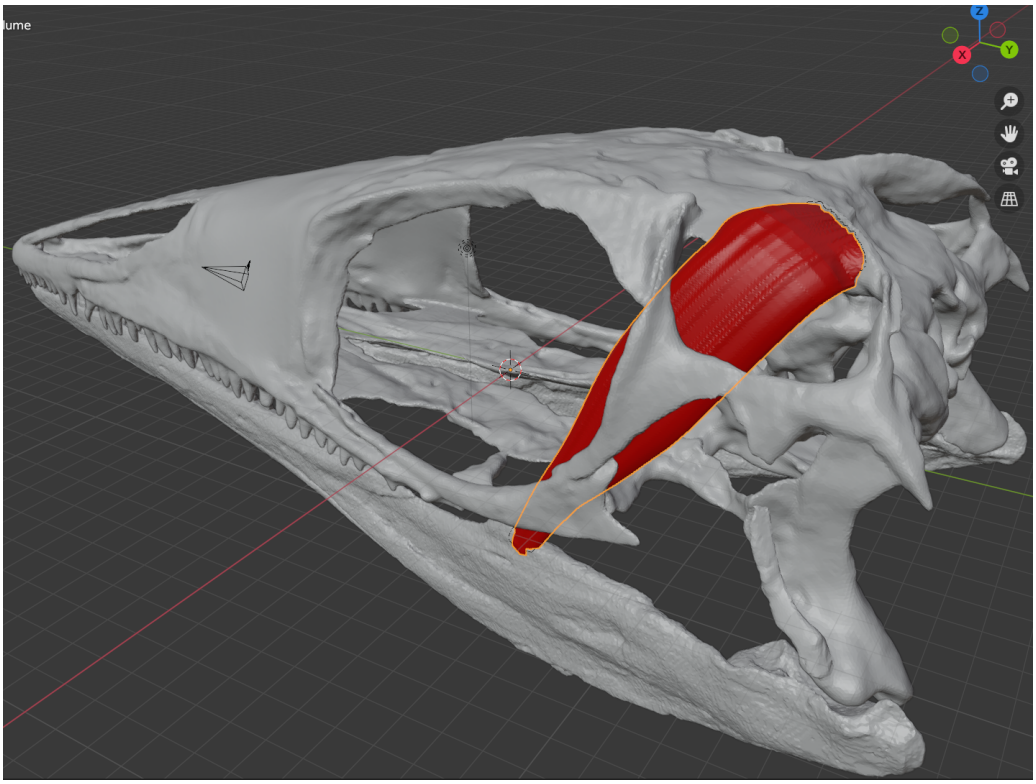
```
mdt = "./assets/MuscleDecompositionTest" # Executable
    if(platform == "win32"):
        mdt += ".exe"

cmd = [
        os.path.join(os.path.realpath(os.path.dirname(__file__)), mdt)
        volume,
        "-i",
        insertion,
        "-o",
        origin,
        "-n",
        str(bpy.context.scene.export_fibres),
        "-r",
        str(bpy.context.scene.export_resolution),
        "-v",
        str(bpy.context.scene.export_visualize),
        "-f",
        output
      ]
result = subprocess.run(cmd, stdout=subprocess.PIPE)
```

Zde je vhodné upozornit na jeden důležitý aspekt. Hlavní prerekvizitou je přítomnost sestavené verze nástroje *MuscleDecomposition*. To znamená, že uživatel musí mít dostupnou sestavenou verzi pro svou konkrétní platformu, bez které nemůže proces dekompozice proběhnout. Výhodou tohoto přístupu je nezávislost dodatku na konkrétní verzi nástroje, tudíž je potřeba pouze sestavit novou verzi pro danou platformu a nahradit konkrétní spus-

titelný soubor ve složce s dodatkem (konkrétní instalace a další poznámky jsou uvedeny v uživatelské příručce, kapitola 9.1). Zároveň je opět nutné dodržet konvenci pojmenovávání - spustitelný soubor **musí** být pojmenován *MuscleDecompositionTest*.

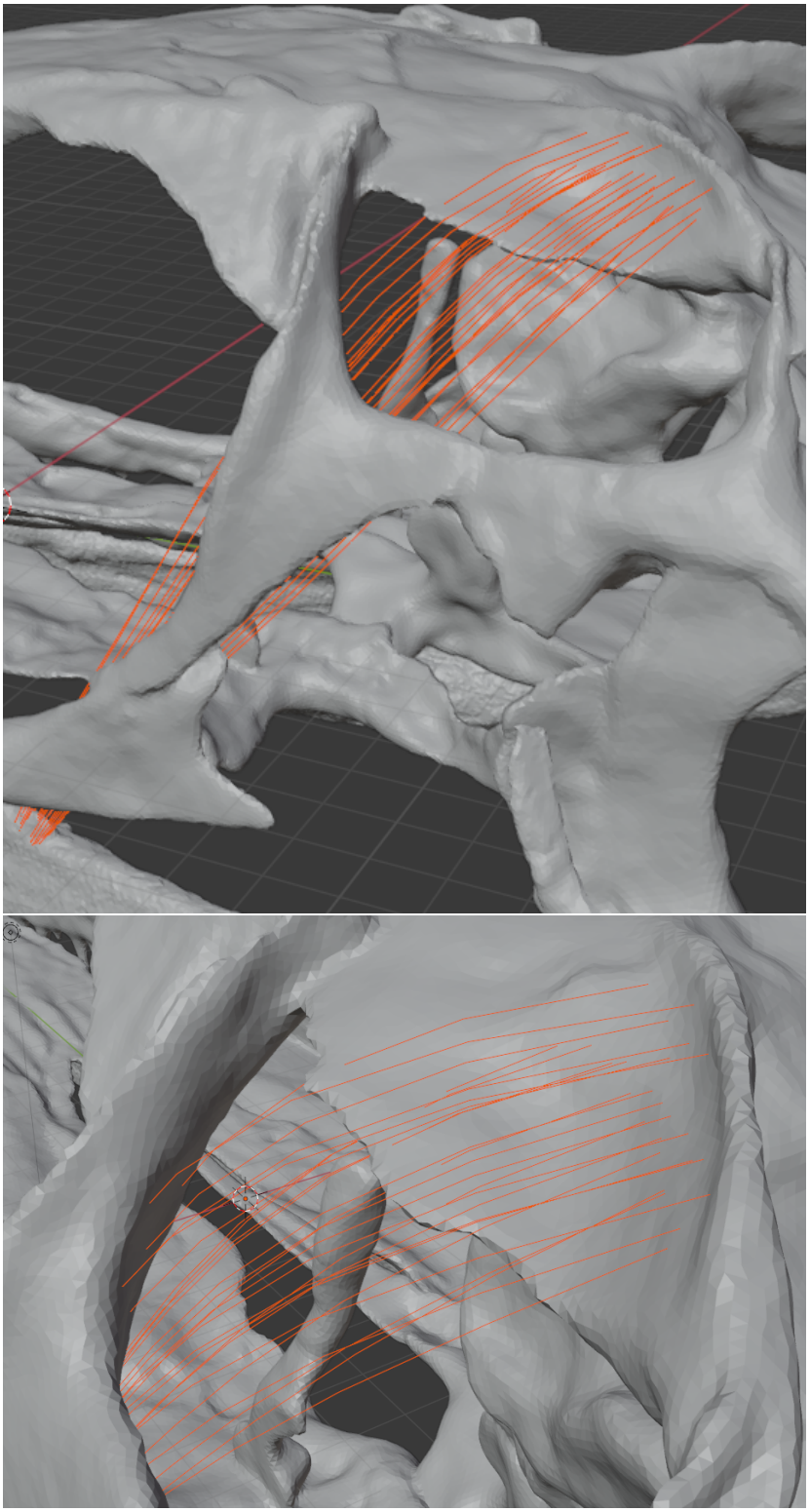
Pokud je spustitelný soubor **kompatibilní** pro danou platformu, proběhne proces dekompozice interně v rámci nástroje. Jestliže proběhl proces bez chyb, vznikne nový soubor dle pojmenovávacích konvencí „*název_svalu* decomposed.obj". Pokud nastala chyba během procesu dekompozice, je uživatel informován o chybě pomocí chybového hlášení.

Výsledkem celého procesu dekompozice je model strukturovaný do jednotlivých svalových vláken. Ukázka výsledku dekompozice vyobrazena na obrázcích 6.6, 6.7 a 6.8.

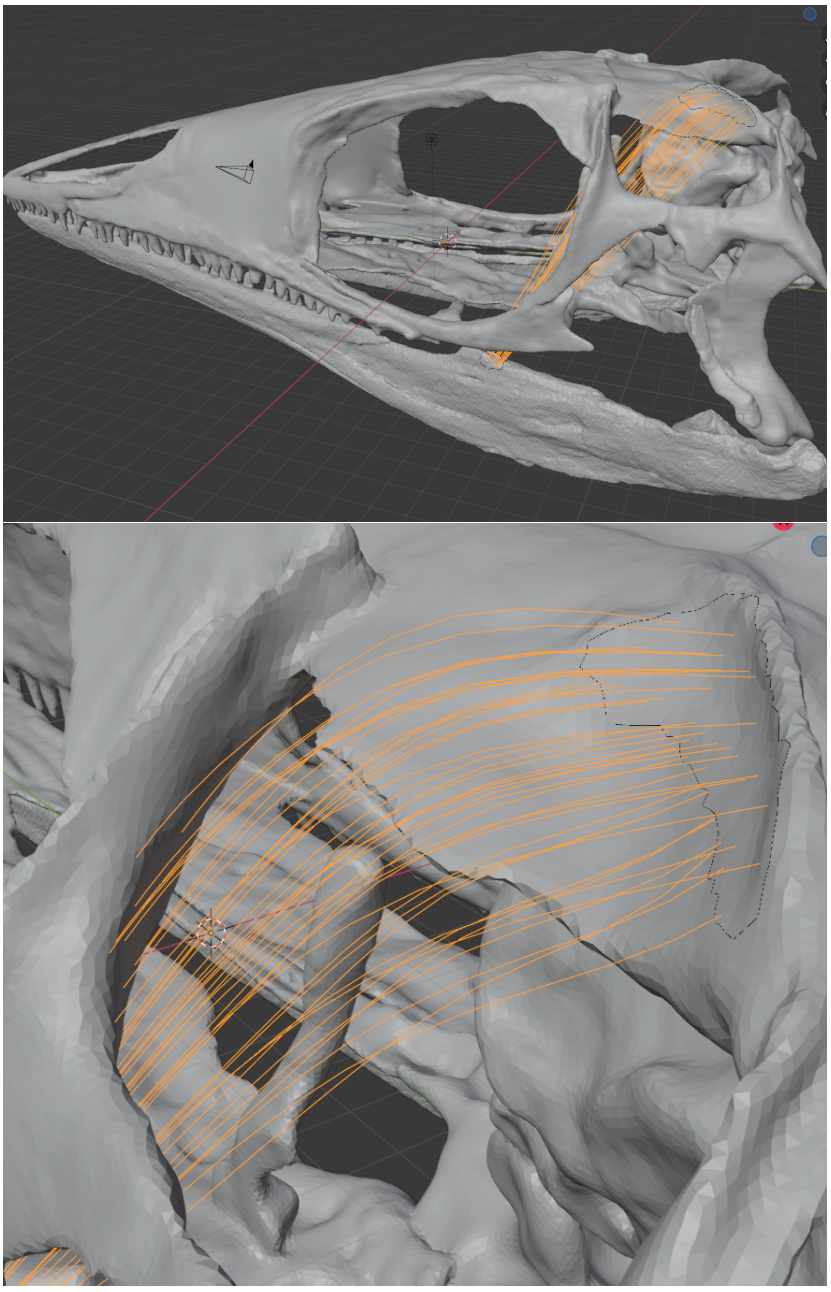


Obrázek 6.6: Model svalu před dekompozicí

Na obrázku 6.7 si lze všimnout, že některá vygenerovaná vlákna protínají jednu z kostí modelu. Toto je praktická ukázka důležitosti volby parametrů samotné dekompozice, kdy je nutné zvýšit zejména parametr *resolution*, který představuje „segmenty" každého vlákna. Při zvýšení tohoto parametru výsledný model mnohem lépe reprezentuje předlohu, zejména tvarem vláken (viz obrázek 6.8).



Obrázek 6.7: Model svalu po dekompozici. Parametr *fibers* = 25, *resolution* = 15



Obrázek 6.8: Model svalu po dekompozici. Parametr *fibers* = 30, *resolution* = 45

### 6.5.2 Dekompozice více modelů

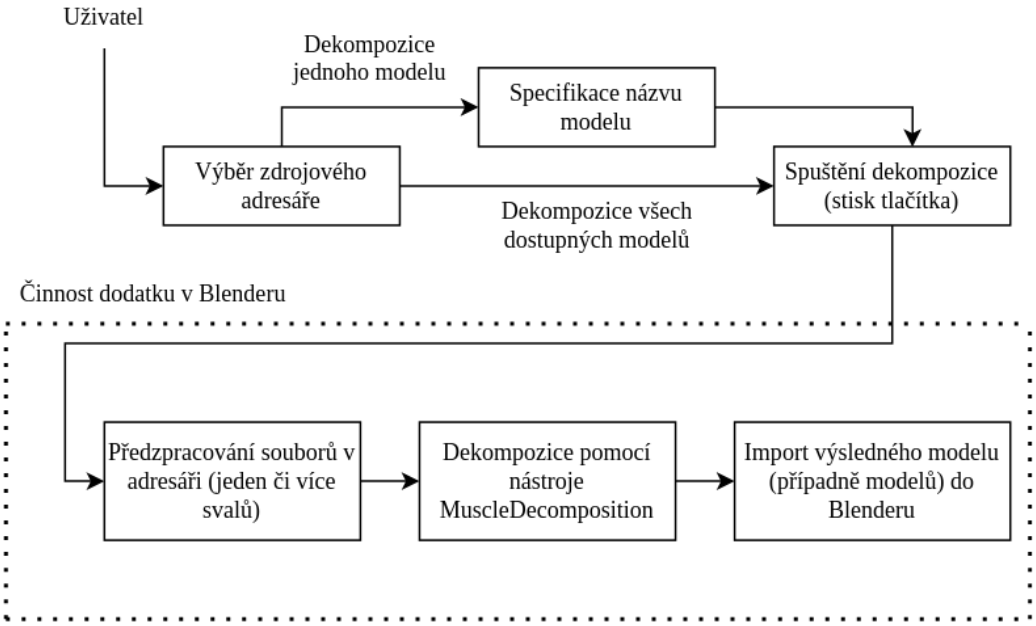
Druhou možností dekompozice je stisknutí tlačítka „*Decompose all muscles*". Oproti dekompozici jednoho svalu je potřeba specifikace pouze adresáře, kde se nachází modely určené pro dekompozici. Pokud není adresář specifikován, je toto tlačítko znepřístupněno. V opačném případě je prozkoumáván celý adresář, soubor po souboru. Nejdříve se názvy souborů oddělí a jsou uloženy jen názvy příslušných svalů. Například, nachází-li se v adresáři následující soubory:

- *muscle1 origin.vtk*,
- *muscle1 volume.stl*,
- *muscle1 insertion.vtk*,
- *muscle2 origin.vtk*,
- *muscle2 volume.stl*,
- *muscle2 insertion.vtk*,

jsou po celém projetí adresáře načteny pouze názvy *muscle1* a *muscle2*. Pokud by adresář nebyl předzpracován a bral by se každý soubor, pro každý sval by proběhly tři procesy dekompozice, což je zbytečné a nežádoucí. Kvůli tomu bylo nutné zajistit eliminaci duplicitních procesů a jsou vybíráný pouze modely s odlišnými názvy. Pro lepší pochopení je uveden příklad předzpracování adresáře níže. Poté už jsou rozpoznané modely v cyklu dekomponovány, tudíž ve výše zmíněném případě vzniknou dva soubory, vždy jeden pro daný sval (soubor typu OBJ). Dekomponované modely jsou všechny importovány zpět do Blenderu.

```
### Příklad předzpracování adresáře pro dekompozici ###

to_decompose = [] # Muscle names to decompose
dir = bpy.path.abspath(bpy.context.scene.output_path)

for entry in os.listdir(dir):
    if(entry.endswith(ExportButton.export_format_origin_insertion)
        or entry.endswith(ExportButton.export_format_volume)):

        name = entry.split(ExportButton.type_delimiter)[0]

        if(("decomposed" not in name) and
           (name not in to_decompose)):
              to_decompose.append(name)
```

Pro lepší vizualizaci celého procesu dekompozice je na obrázku 6.9 zobrazen diagram znázorňující posloupnost jednotlivých činností.




Obrázek 6.9: Znázornění procesu dekompozice

# 7 Testování vytvořeného dodatku

V rámci testování všech aspektů dodatku byl vytvořen následující postup:

- Správné načtení a vykreslení dodatku v rámci scény v Blenderu,
- Testování každého separátního elementu (lze editovat, aktivní pouze v situacích, kdy má, je správně parametrově omezen, ...),
- Testování procesu exportování,
- Testování procesu dekompozice a importování.

Testování probíhalo na následujících platformách:

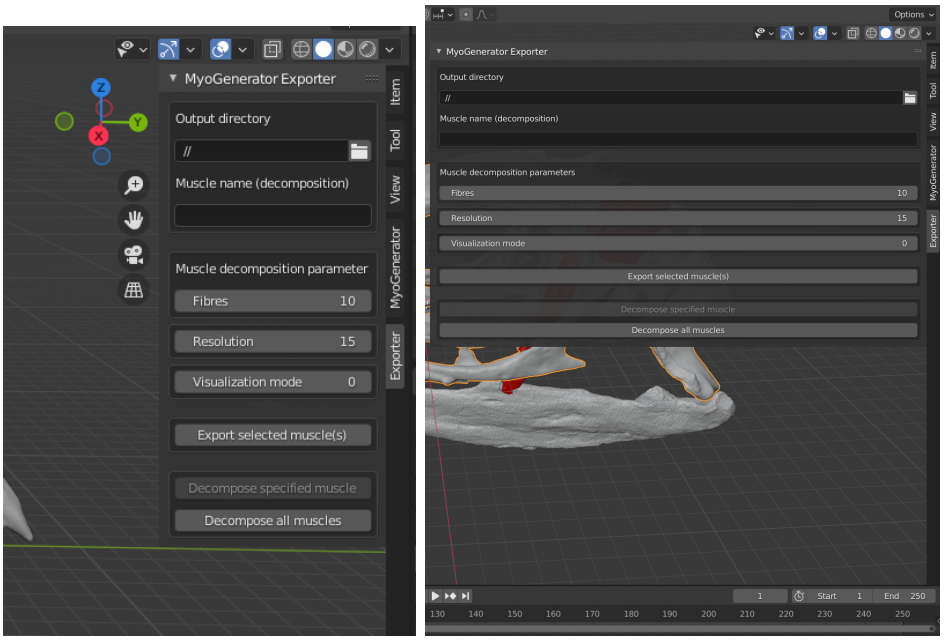
- Platforma Windows NT: Windows 10 Home,
- Platforma GNU/Unix: distribuce Fedora Workstation 36, GNOME 42.1 (Wayland i X.Org sessions).

## 7.1 Testování grafického rozhraní

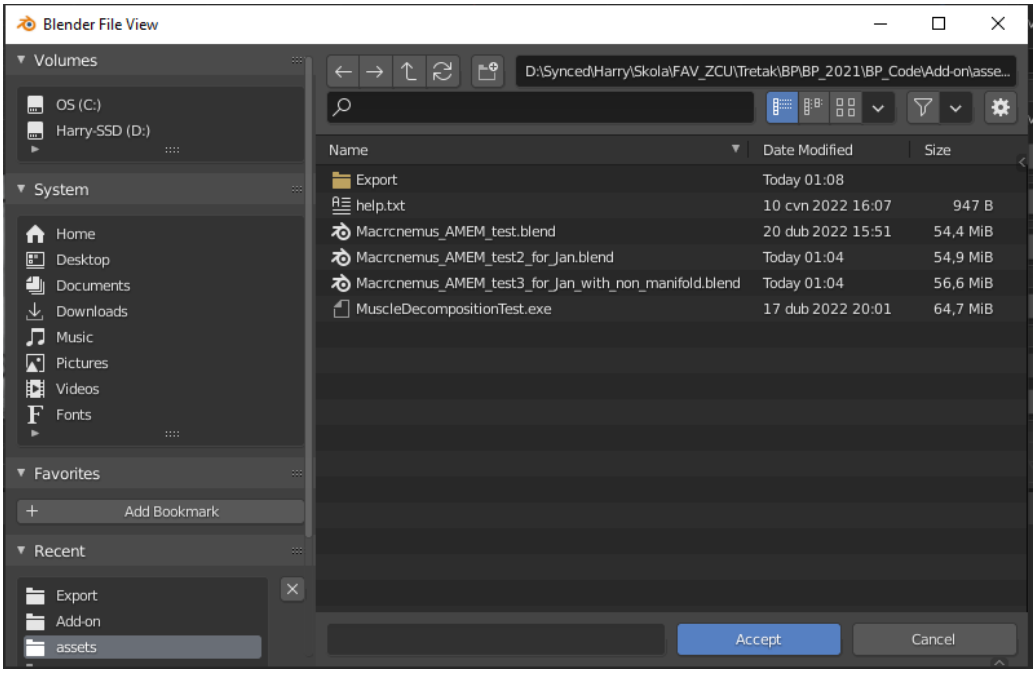
Podrobný proces instalace dodatku je k dispozici v příloze „Uživatelská příručka". Po instalaci byl okamžitě dodatek dostupný v pravé nástrojové liště, která se zobrazí stisknutím klávesy „N", což koresponduje s popisem implementace. U dodatku je taktéž možné upravovat šířku (výšku nikoliv, ta je pevně dána v rámci layoutu tak, aby pokrývala veškeré elementy), načež uspořádání elementů zůstává nezměněno (obrázek 7.1).

Po instalaci si lze všimnout, že jsou v parametru *output directiory* uvedeny dvě lomítka. To je předvyplněno samotným Blenderem a značí umístění, ve kterém se nachází právě otevřený soubor se scénou (koncovka *.blend*). Parametr lze speficikovat ručně (vypsání cesty musí být vztaženo k adresáři, kde se nachází právě otevřený soubor se scénou). Nedodržení tohoto úzu neomezí samotnou funkčnost dodatku, nicméně není garantováno správné umístění výsledných souborů. V pravé části textového pole se nachází tlačítko, po jehož stisknutí se otevře stromová struktura úložiště pro vybrání adresáře pro uložení (viz obrázek 7.2).

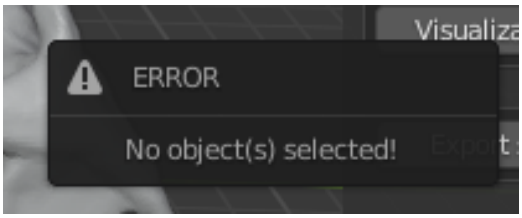
Parametr *muscle name* je v tuto chvíli prázdný, v závislosti na němž je správně nedostupné tlačítko „*Decompose specified muscle*". Opět je textové pole plně a volně editovatelné.




Obrázek 7.1: Vykreslení dodatku v rámci scény




Obrázek 7.2: Dialogové okono pro výběr adresáře




Obrázek 7.3: Příklad zobrazeného dialogu v případě chyby

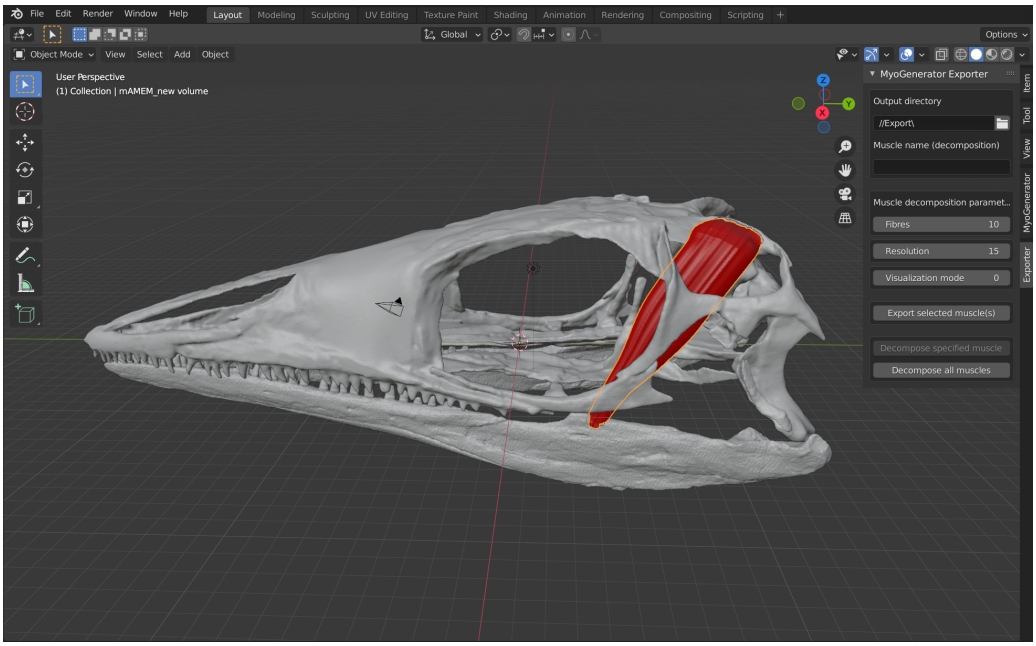
Po zadání libovolného řetězce a potvrzením (klávesou ENTER, případně kliknutí myši mimo textové pole) se správně aktivuje výše zmíněné tlačítko. Parametry dekompozice jsou v základním nastavení nastaveny na hodnoty *fibres = 10, resolution = 15, visualization mode = 0*. Tyto hodnoty jsou nastaveny jako „defaultní" v samotném nástroji *MuscleDecomposition*, tudíž byly pouze převzaty. V rámci „posuvníku" jsou aplikovány zmíněné soft limity, avšak po dvojím kliknutí na „posuvník" lze manuálně zadat libovolnou hodnotu (pokud uživatel zadá hodnotu menší jak 1, je automaticky přepsána na hodnotu minimální (=1)). Jelikož neexistuje žádné pravidlo omezující tyto parametry, **nejsou shora omezeny (!)**, nicméně je nutné předpokládat rapidní nárůst běhu procesu samotné dekompozice se zvyšujícími se hodnotami. Vizualizační mód je omezen pouze na hodnoty 0, 1, 2 (viz 3.1.1), při zadání hodnoty mimo interval je vybrána nejbližší validní.

## 7.2 Testování procesu exportování

Pokud není vybrán žádný model a stisknuto tlačítko pro export, je zobrazen příslušný dialog (obrázek 7.3). Pokud je vybrán alespoň jeden objekt v rámci scény, je nadále testováno, zda-li se v názvu vyskytuje pojem *volume*/*origin*/*insertion*, vzhledem k zavedeným konvencím, následně dle kterých se rozlišuje konkrétní proces exportizace (viz kapitola 6.4). Pokud tomu tak není, je o tom uživatel informován následujícím upozorněním a proces exportizace zastaven.

```
Wrong muscle name format!
Model should be named using following pattern:
<muscle_name><space><origin|volume|insertion>
<space><boundary>!
```

V případě, že model splňuje zavedené konvence a zároveň jsou v rámci scény vybrány modely svalu a úponových oblastí, je po kliknutí tlačítka tento model exportován a uživatel je informován dialogem o úspěšném procesu.

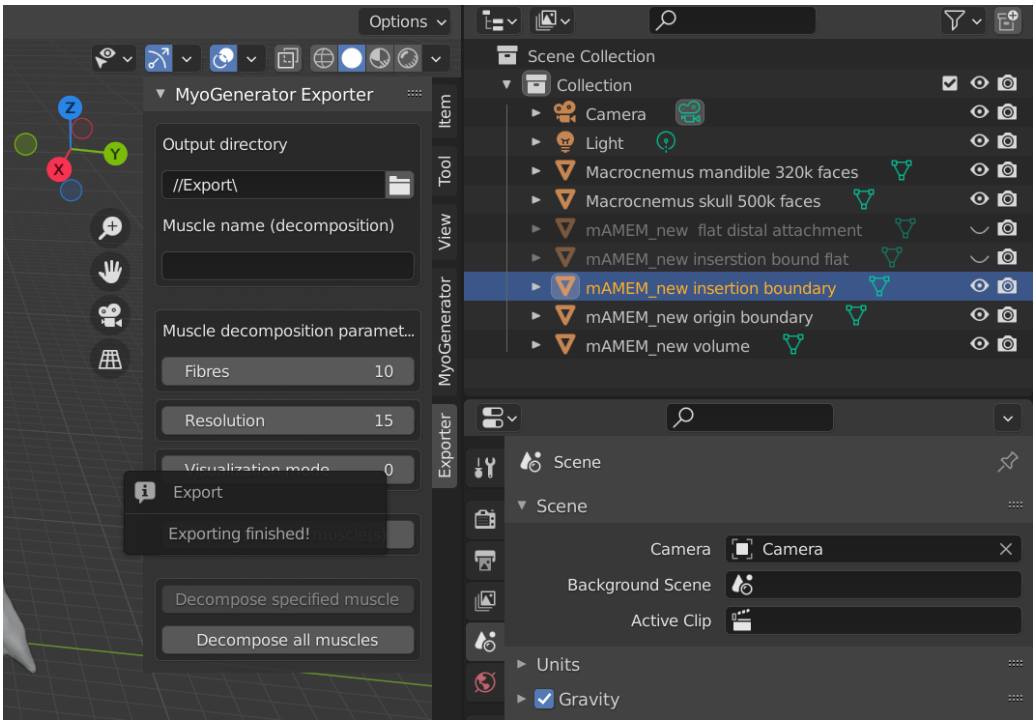



Obrázek 7.4: Model „mAMEM_new”

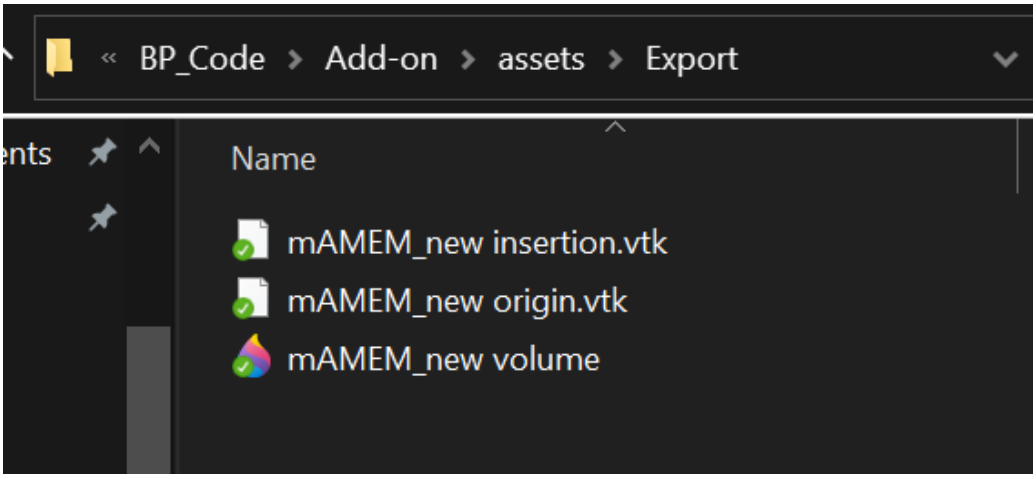
Pro účely testování exportu a dekompozice vytvořila Dr. Herbst tři separátní soubory reprezentující různé scény v rámci Blenderu. Tyto soubory (typu *.blend*) jsou součástí příloh k této práci. V rámci příloh k této práci jsou taktéž dostupné soubory pro porovnání exportovaných dat pro jednotlivé modely svalů, které byly zkontrolované a schválené Dr. Herbst. V rámci prvního souboru bude testování prováděno na modelu svalu pojmenovaném „mAMEM_new” (viz obrázek 7.4). Nejdříve je vybrána jedna část svalu (například jedna z úponových oblastí) a poté je kliknuto na tlačítko „*Export selected muscle(s)*”. Výsledek této operace je nově vytvořený soubor v dříve specifikovaném adresáři pod názvem „*mAMEM_new insertion.vtk*” (viz obrázek 7.5). Následně je tento proces zopakován pro zbylé části modelu. Jelikož se exportované soubory shodují s požadovanými výsledky, lze se nyní přesunout k testování procesu dekompozice.

## 7.3 Testování procesu dekompozice

Pro proces dekopozice jednoho specifického modelu je nutné specifikovat jak adresář, ze kterého má být model „načten”, tak i název modelu svalu (pouze ve tvaru „*název_svalu*”, ve výše uvedeném příkladě „mAMEM_new”). V případě, že je kliknuto na toto tlačítko a ve zvoleném adresáři **neexistuje** žádný soubor vyhovující zadanému názvu, je uživatel




(a) Exportizace úponové oblasti modelu „mAMEM_new”




(b) Výsledek exportizace všech částí modelu „mAMEM_new”

Obrázek 7.5: Ukázka exportu modelu „mAMEM_new”

upozorněn následujícím hlášením:

```
File has not been found!
Please, check the filename and directory
(File: <cesta_k_souboru>)
```

Pokud příslušný model existuje, proběhne kontrola, zda-li jsou dostupné všechny části modelu (tj. počáteční a koncová úponová oblast, model samotného svalu). Pokud ne, je uživatel upozorněn, která konkrétní část nebyla nalezena (popis chyby identický k výše uvedenému). Jakmile jsou veškeré potřebné soubory dostupné, je nutné otestovat, zda-li je dostupný sestavený nástroj *MuscleDecompositionTest* pro danou platformu. Pokud není sestavená verze zmíněného nástroje nalezena, je uživatel varován následujícím chybovým hlášením:

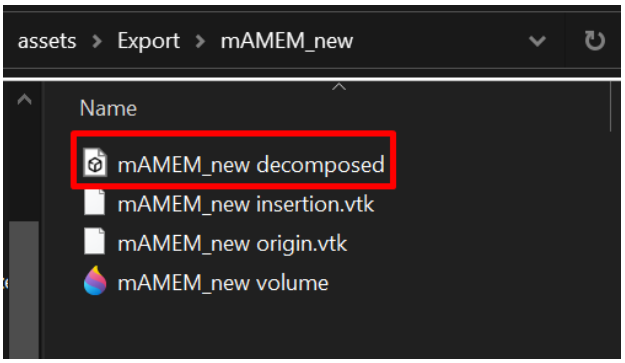
```
MuscleDecomposition executable
has not been found in the add-on folder!
Please, proceed to user manual for further information.
```

Tato chyba může být relativně běžná, zejména pro uživatele na linuxových distribucích, jelikož tento dodatek **nebude** poskytován se sestavenou verzí nástroje *MuscleDecomposition* pro tuto platformu. Toto rozhodnutí vzniklo kvůli rozmanitosti v nadstavbách, v kombinaci se samotným počtem různých distribucí. Jelikož vyžaduje řešení tohoto problému lehce delší posloupnost kroků, bude uvedeno v rámci přílohy „*Uživatelská příručka*". Pokud bylo řešení ze strany uživatele úspěšné, proběhne proces dekompozice a zobrazí se následující hlášení:

```
MuscleDecomposition output: Successful!
```

a v rámci scény je importován dekomponovaný model svalu. Pokud proces dekompozice z nějakého důvodu úspěšný nebyl, objeví se identický dialog, avšak část „*Successfull*" bude nahrazena výpisem konkrétního problému. Výsledek dekompozice uvedeného modelu svalu (obrázek 7.4) je zobrazen na obrázcích 6.6, 6.7 a 6.8. Pokud byl proces úspěšný, vznikne ve vybraném adresáři nový soubor (v rámci konkrétního modelu uveden na obrázku 7.6).

Vzniká zde však nový problém. Jelikož neexistuje žádná předloha či pravidlo, jakým způsobem má konečný dekomponovaný sval vypadat, není možné stoprocentně ověřit, zda-li dekomponovaný sval odpovídá jeho předloze. Nicméně, lze alespoň dle několika aspektů rozhodnout, zda-li proces dekompozice proběhl dle předpokladů. Pokud je nově vzniklý model vyhovující, budou všechny úponové body jednotlivých vláken uvnitř příslušné




Obrázek 7.6: Nově vzniklý soubor po úspěšném procesu dekompozice

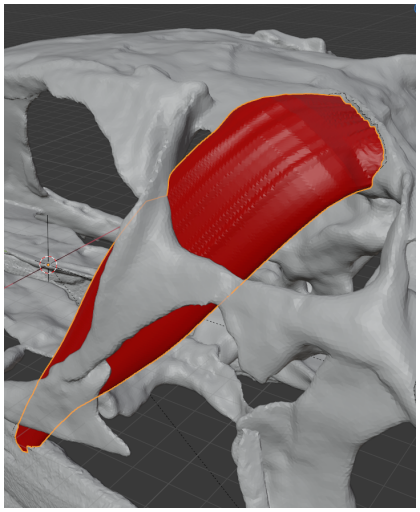
úponové oblasti svalu. Taktéž je při nízkém počtu vláken vhodné zkontrolovat, zda-li opravdu odpovídají zvoleným parametrům.

Veškerý výše uvedený postup byl v identické posloupnosti otestován na další dvou poskytnutých souborech se svaly. Ve scéně druhého dostupného souboru se nachází tři modely svalů: *mAMEM_new* (identický k předchozímu), *mAMEM_two* (identický k *mAMEM_new*, nicméně zde byla lehce upravena síť modelu, aby lépe lícovala s příslušnými kostmi) a nový sval *DM*. Pro všechny dostupné modely byl proveden výše popsaný postup testování, jejichž výsledky jsou vyobrazeny na obrázcích 7.7 a 7.8.

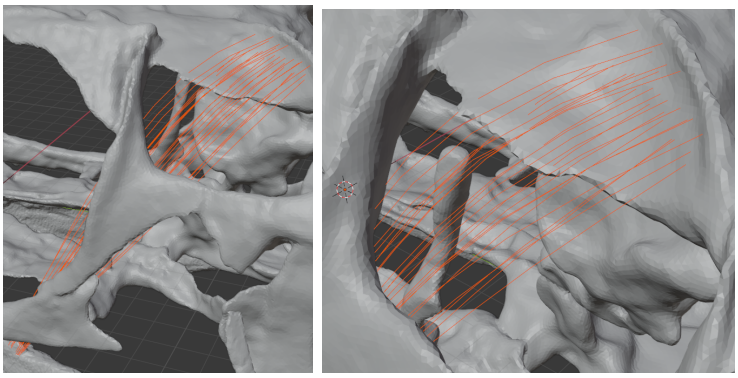
V rámci posledního souboru se nachází identické modely svalů v porovnání s druhým souborem, nicméně u modelu *DM* byla upravena jeho trojúhelníková síť, do které byly zakomponovány potencionální chyby, ke kterým by mohlo dojít během tvorby modelu (chybějící hrana, řidší struktura atp.). Výsledky dekompozice dopadly stejným způsobem, viz obrázek 7.8.

**Poznámka:** U různých scén je nutné rozkliknout příslušné modely (viz obrázek 7.9) pro zobrazení jejich jednotlivých částí. Při exportizaci úponových oblastí je nutné zvolit nejnižší úroveň modelu (tj. pouze ohraničení těchto oblastí). Pokud budou exportovány celé modely úponových oblastí, dojde k chybě během dekompozice, jelikož tyto modely **nebudou** splňovat jejich definici, viz kapitola 2.2.1 (pro připomenutí: *„Pro účely této práce předpokládejme reprezentaci ve formě množiny bodů „ohraničujících" příslušnou oblast a zároveň respektující dané pořadí."*).

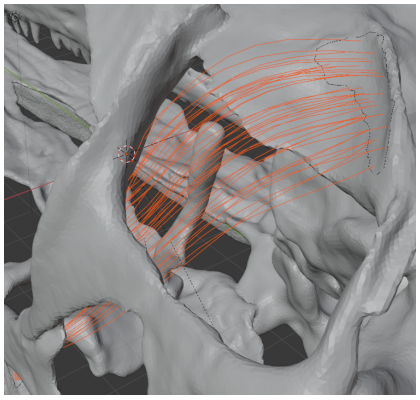
Pro testování procesu dekompozice **všech** dostupných svalů byly soubory exportovaných svalů *mAMEM_two a DM* přesunuty do jednoho adresáře.



(a) Původní model svalu

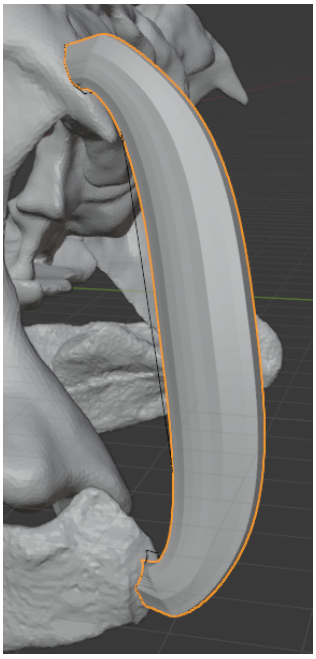


(b) Dekompozice s parametry: *fibres* = 25, *resolution* = 15

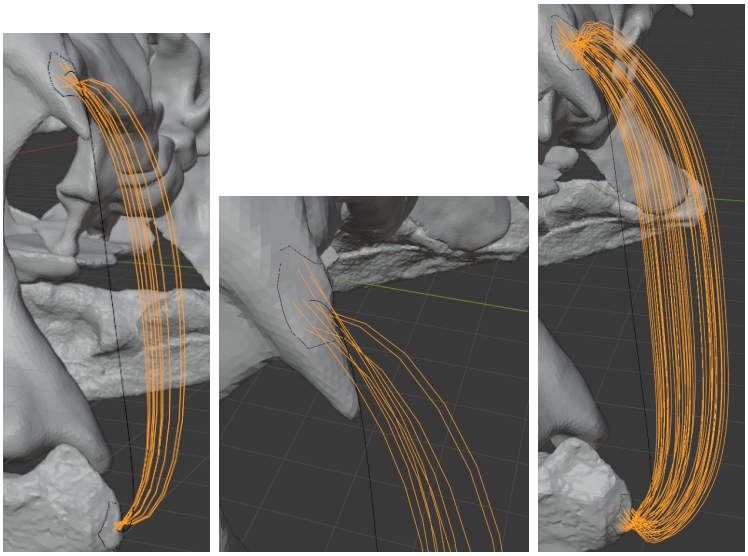


(c) Dekompozice s parametry: *fibres* = 40, *resolution* = 40

Obrázek 7.7: Výsledky dekompozice modelu „*mAMEM_two*"

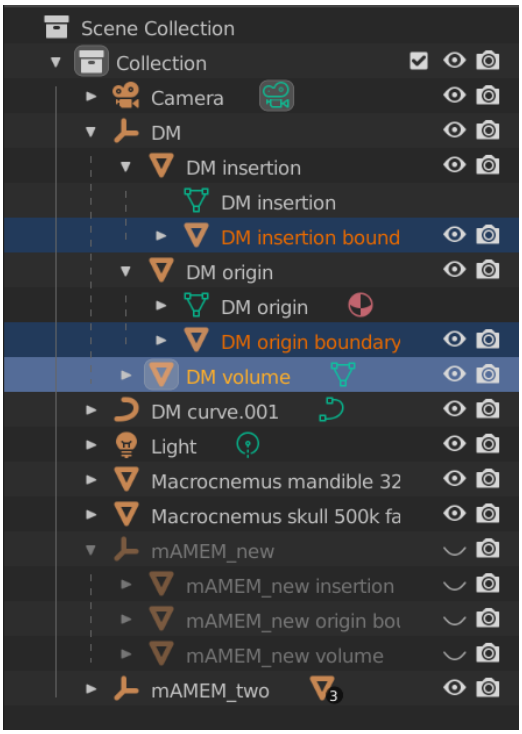


(a) Původní model svalu



(b) Dekompozice s parametry: *fibres* = 10, *resolution* = 15 a *fibres* = 45, *resolution* = 55

Obrázek 7.8: Výsledky dekompozice modelu „*DM*"




Obrázek 7.9: Struktura modelu svalů v testovacích souborech (nutné vybrat hranice s označením „boundary")

Tento adresář byl následně vybrán v rámci parametru *output directory* a stisknuto tlačítko *Decompose all muscles*. Výsledkem byly oba dekomponované modely (viz obrázky 7.7 a 7.8), parametry byly ponechány v základním nastavení.

# 8 Závěr

Dodatek, jež byl předmětem této práce, byl úspěšně vytvořen se všemi povinnými požadavky (viz kapitola 5.1). V rámci kategorie „*nice to have*" byla implementována funkčnost možnosti dekompozice více svalů najednou. Grafické rozhraní bylo implementováno tak, aby korespondovalo s rozhraním nástroje *MyoGenerator*, tudíž uživatel využívající tento nástroj může očekávat exaktní chování společných elementů. Vytvořený dodatek úspešně umožňuje uživateli exportovat libovolný sval ze scény společně s jeho úponovými oblastmi, jehož výsledné soubory jsou mu plně k dispozici. Dodatek taktéž implementuje funkční verzi dekompozice za využití nástroje *MuscleDecompositionTest*, pomocí kterého je model svalu dekomponován na jednotlivá svalová vlákna.

Z pohledu Dr. Herbst dodatek splňuje veškerou požadovanou funkčnost. Bohužel, jelikož byla velice časově vytížena, neměla možnost dodatek řádně otestovat. Nicméně, s Dr. Herbst bylo v průběhu vývoje uspořádáno několik online schůzek, kde byla představena prozatimní funkčnost dodatku a probrány další postupy. Poslední schůzka proběhla během fáze testování, kdy došlo ke konzultaci některých chyb, které byly již opraveny. V rámci poslední schůzky byla Dr. Herbst velice spokojena s výsledkem vývoje, taktéž byly probrány další možnosti rozšíření dodatku, jako podpora novějších verzí Blenderu, nebo implementace dodatku bez požadavku na sestavený nástroj *MuscleDecompositionTest*. Taktéž bylo oceněno ovládání dodatku, jež koresponduje s ovládáním v rámci nástroje *MyoGenerator*, tudíž je velice intuitivní přecházet mezi těmito nástroji.

Nicméně, jak bylo řečeno výše, je nutné ve vývoji dodatku a souvisejících nástrojů dále pokračovat. Mohou se, kvůli absolutní různorodosti svalů, vyskytnout případy, kdy nebude dekompozice úplně odpovídat své předloze, přičemž je vhodné tyto případy dále zkoumat a přizpůsobovat dodatek tak, aby byl vhodný pro co nejuniverzálnější použití. Proto je předpokládáno, že spolupráce mezi autorem tohoto dodatku a Dr. Herbst bude nadále vedena i mimo rámec této práce.

# Literatura


[1] *Projekt Muscle Wrapping (verze 2.0) - repozitář* [online]. 2018. [cit. 10/10/2021]. Dostupné z: `https://gitlab.com/besoft/muscle-wrapping-2.0`.

[2] *Dokumentace k souborovému formátu STL* [online]. 2019. [cit. 2022/01/08]. Dostupné z: `https://www.loc.gov/preservation/digital/formats/fdd/fdd000504.shtml`.

[3] *Dokumentace k souborovému formátu OBJ* [online]. Fileformat.com, unknown. [cit. 2021/12/13]. Dostupné z: `https://docs.fileformat.com/3d/obj/`.

[4] *Dokumentace k souborovému formátu VTK* [online]. Kitware, 2015. [cit. 2021/12/13]. Dostupné z: `https://vtk.org/wp-content/uploads/2015/04/file-formats.pdf`.

[5] *Dokumentace k API Blenderu* [online]. blender.org. Dostupné z: `https://docs.blender.org/api/current/index.html`.

[6] Herbst, D. E. C. *Skript do aplikace Blender pro vytvoření svalového primitiva - repozitář* [online]. Dostupné z: `https://github.com/evaherbst/MyoGenerator`.

[7] Jaroslav Sklípka, Z. T. *Základy Histologie*. Univerzita Karlova, 2019.

[8] Luca Modense, J. K. *Automated Generation of Three-Dimensional Complex Muscle Geometries for Use in Personalised Musculoskeletal Models* [online]. Dostupné z: `https://link.springer.com/article/10.1007/s10439-020-02490-4`.

# 9 Přílohy

## 9.1 Uživatelská dokumentace

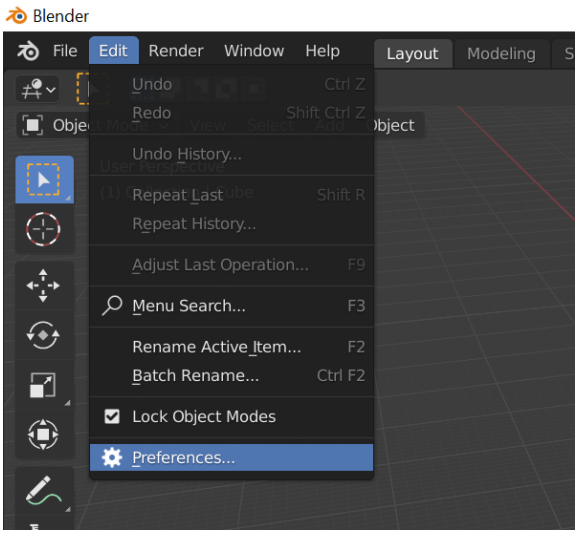
### 9.1.1 Prerekvizity

- Zip archiv obsahující soubory dodatku
- Blender - oficiálně testovaná verze 2.93.X
- Sestavená verze nástroje MuscleDecompositionTest
  - V rámci dodatku je dostupná sestavená verze pro platformu Windows
  - Pro platformu Linux je nutné sestavit vlastní verzi (kvůli různorodosti distribucí)

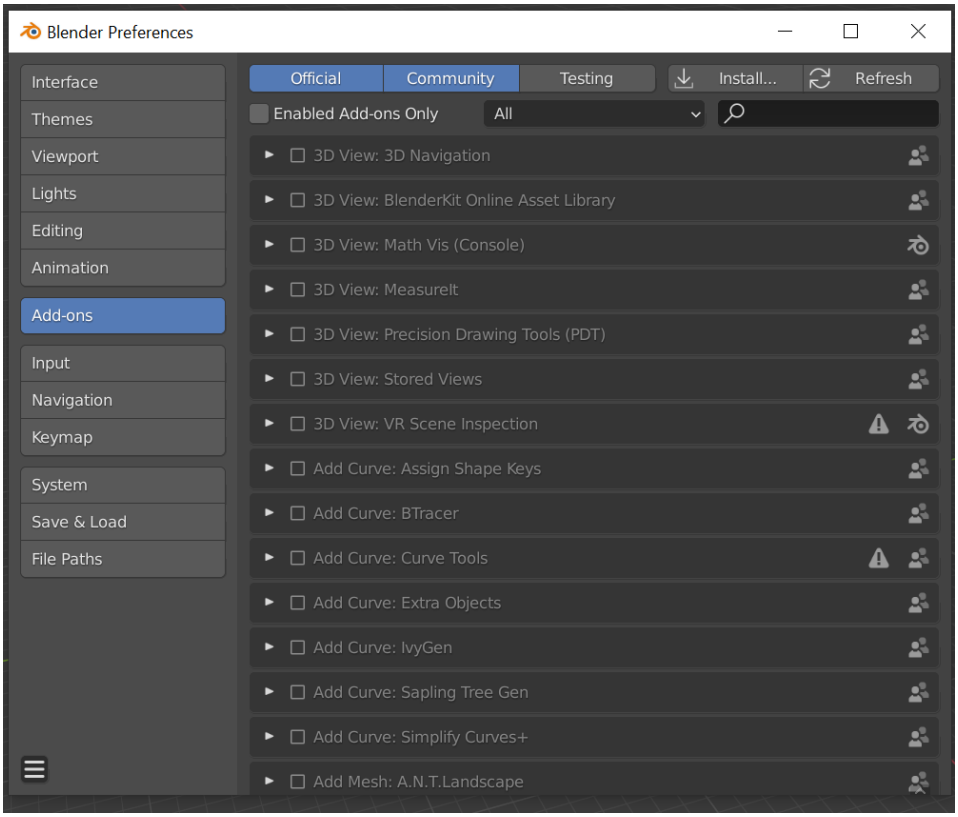
### 9.1.2 Postup instalace

#### Windows

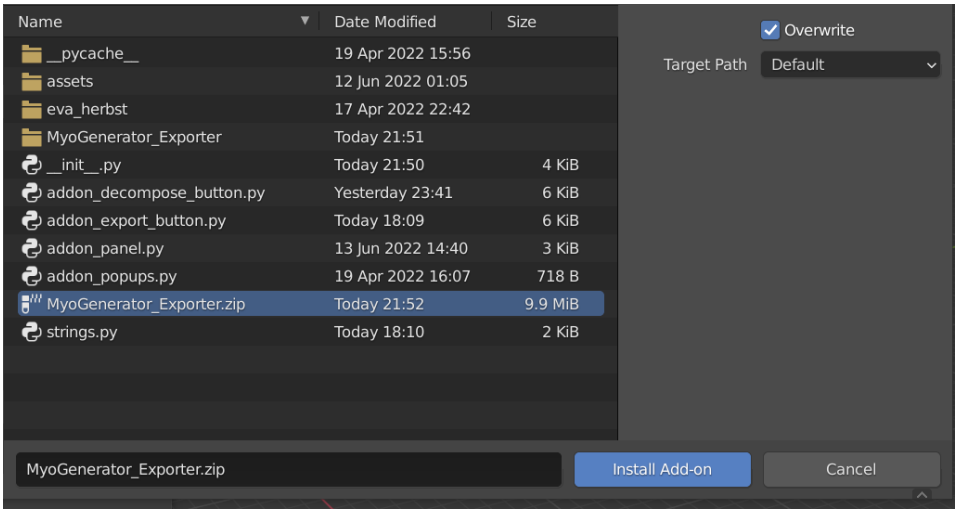
1. Stáhněte a nainstalujte Blender běžným způsobem (`https://www.blender.org/`), starší verze na `https://download.blender.org/release/` (preferovaná verze 2.93.X)
2. Nainstalovaný Blender otevřete, zobrazí se Vám hlavní okno programu.
3. Na této obrazovce klikněte v levém horním rohu na tlačítko **Edit -> Preferences**



4. Zobrazí se vám nové okno se nastavením, zde v levé liště klikněte na možnost **Add-ons**

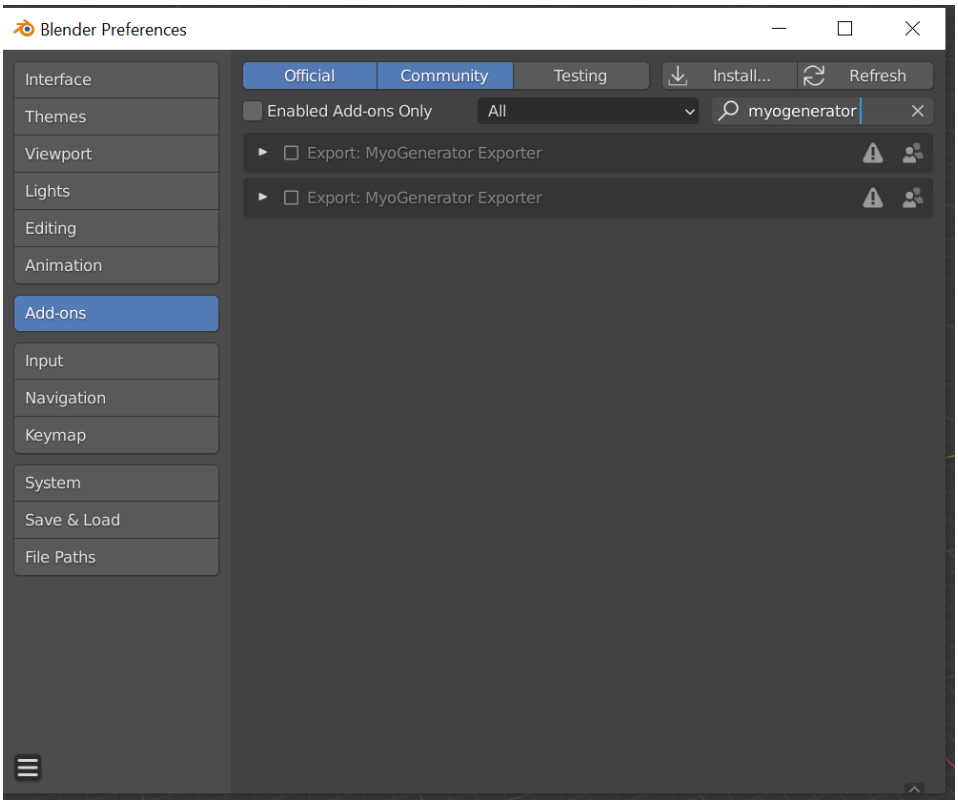



5. Nyní jsou zobrazeny všechny dostupné dodatky, v právém horním rohu klikněte na tlačítko **Install** a v rámci dialogu vyberte *zip* archiv s dodatkem a opět potvrďte tlačítkem **Install add-on**

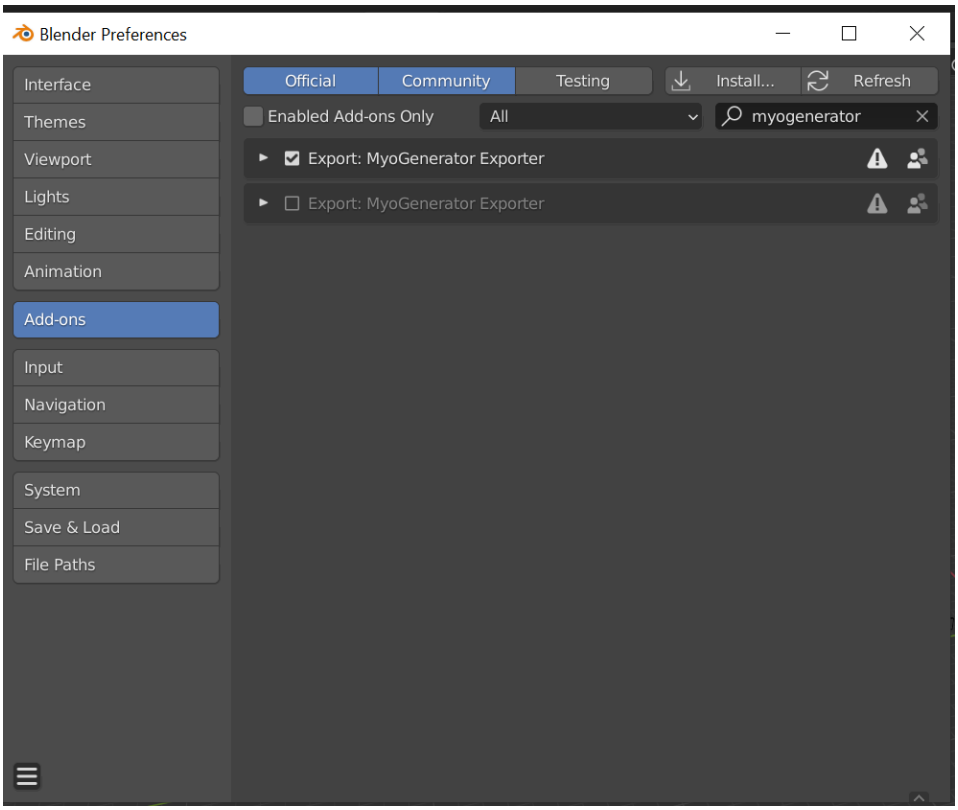



6. Dodatek je nyní nainstalován, nicméně je potřeba ho aktivovat v okně se všemi dostupnými dodatky.

7. Využijte vyhledávacího pole v pravém horním rohu a pomocí „myogenerator exporter” najdete dodatek.




8. Může se stát, že bude dodatek zobrazen dvakrát, jedná se o neduh Windows, kdy vytvoří dva symbolické linky na dodatek (nejedná se o nijak závažnou chybu)
9. Je potřeba jeden ze záznamů zaktivovat pomocí zaškrtnutí „checkboxu”




10. V tuto chvíli je dodatek připraven k použití. V módu „Object" je po zmáčknutí klávesy „N" dostupná lišta, kde je dodatek umístěn

### Linux

Postup pro instalaci dodatku na platformu Linux je identický, nicméně je potřeba dodatečných kroků, jelikož v rámci dodaného *.zip* archivu není přiložen sestavený nástroj MuscleDecompositionTest. Po provedení kroků výše je nutný dodatečný postup:

1. Sestavte verzi nástroje MuscleDecompositionTest pro Vaši konkrétní platformu: `https://gitlab.com/besoft/muscle-wrapping-2.0`
2. Sestavenou spustitelnou verzi nástroje pojmenujte jako „**MuscleDecompositionTest**"
3. Přesuňte tento nástroj do složky s instalovaným dodatkem, běžně v následujícím adresáři: *~/.config/blender/<instalovaná verze>/scripts/addons/MyoGenerator_Exporter/assets*
4. V rámci této složky poté otevřete *terminál* a vykonejte následující příkaz, který umožní v rámci Blenderu používat tento nástroj:

   *sudo chmod u+x MuscleDecompositionTest*

5. Nyní je dodatek plně funkční a připraven k použití

### 9.1.3 Volitelné instalace

V rámci dodatku je doporučeno využívání nástroje *MyoGenerator* pro tvorbu modelů svalů, nicméně jeho využití a instalace **nejsou** povinné. Nicméně postup instalace je následující:

1. Z odkazu `https://github.com/evaherbst/MyoGenerator` stáhněte složku „Addon Folder", vytvořte nový *.zip* archiv a tuto složku do něj vložte.
2. Dále postupujte identickým postupem, jako při instalaci tohoto dodatku

### 9.1.4 Chybová hlášení

**„Wrong muscle name format! Model should be named. . . "** Jedná se o chybu, kdy nebyly dodrženy pojmenovávací konvence modelů ve scéně. Je nutné dodržet následující pojmenování:

- Počáteční úponová oblast - „*název_svalu* origin boundary",
- Koncová úponová oblast - „*název_svalu* insertion boundary",
- Model svalu - *název_svalu* volume".

**„MuscleDecomposition executable has not been found. . . "** Jedná se o chybu, kdy není v rámci dodatku nalezen sestavený nástroj *MuscleDecompositionTest*. V tuto chvíli je nutné zkontrolovat, zda-li se nástroj „*MuscleDecompositionTest*" (.exe na Windows) vyskytuje ve složce s instalovaným dodatkem:

- Běžné umístění na platformě Windows: C:/Users/<jméno uživatele>/ AppData/Roaming/Blender Foundation/Blender/<instalovaná verze>/ scripts/addons/MyoGenerator_Exporter/assets/

- Běžné umístění na platformě Linux: ~/.config/blender/<instalovaná verze>/scripts/addons/MyoGenerator_Exporter/assets