

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Laas Hendrik Lumberg**  
**Trimmer – Lõikelehe töövoog rakendus Blenderile**  
**Bakalaureusetöö (9 EAP)**

Juhendaja: Jaanus Jaggo

# **Trimmer – Lõikelehe töövoog rakendus Blenderile**

## **Lühikokkuvõte:**

Lõikelehtede töövoog on tekstuurimise tehnika, kus kasutatakse mitut tekstuuri ühes pildis. Töös luuakse Blenderi tarkvarale uus pistikprogramm Trimmer, mis lõikelehtede töövoogu lihtsustab. Selle jaoks analüüsitakse kahte varasemat Blenderile loodud lahendust. Varasemate lahenduste põhjal koostatakse loodavale programmile nõuded. Valminud pistikprogramm avaldatakse Blenderi laienduste veebilehel. Seda võrreldakse varasemate lahendustega ning selle kasutatavust testitakse kunstitudengite poolt.

## **Võtmesõnad:**

3D, tekstuurimine, lõikeleht, Blender, pistikprogramm

**CERCS:** P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

# **Trimmer – A Trim-sheet Workflow Add-on for Blender**

## **Abstract:**

The trim-sheet workflow is a texturing technique that uses multiple textures (trims) in one image. In the thesis, a new add-on called Trimmer is developed for the Blender software, with the goal of simplifying trim-sheet workflow. Two previous Blender add-ons are analysed, based on which the requirements for Trimmer are established. The completed add-on is published on the Blender extensions webpage. It is compared to the previous solutions and its usability is tested by art students.

## **Keywords:**

3D, texturing, trim-sheet, Blender, add-on

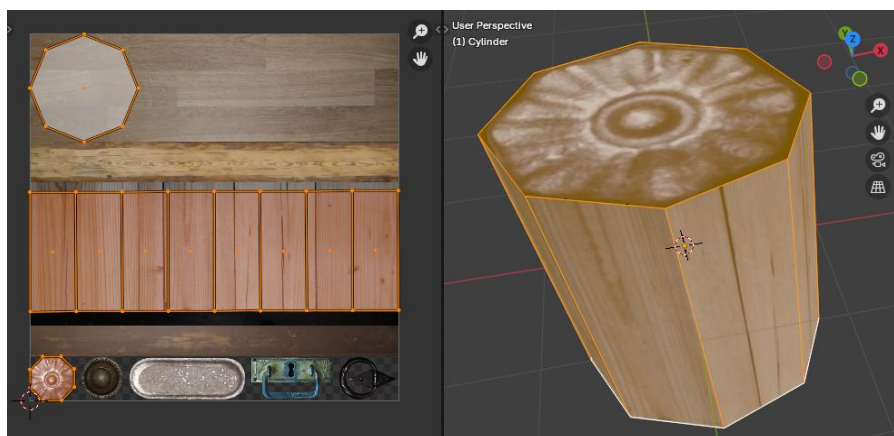
**CERCS:** P170 Computer science, numerical analysis, systems, control

# Sisukord

Sissejuhatus.....	4
Mõisted ja terminid.....	5
1. Töö taust, tekstuurimise alused.....	6
1.1 Tekstuurimine Blenderis .....	6
1.2 Lõikelehtede töövoog Blenderis .....	8
1.3 Varasemad lahendused.....	10
1.3.1 Ultimate Trim UV.....	10
1.3.2 DECALmachine.....	11
1.4 Nõuded .....	14
1.4.1 Lõikelehe defineerimine .....	14
1.4.2 Lõike asetamine .....	15
1.4.3 Lõike paigutuse muutmine.....	16
2. Implementatsioon.....	18
2.1 Metaandmed, kasutajaliides .....	18
2.2 Andmete haldus.....	19
2.3 Algoritmid.....	19
2.3.1 Tahkude komplekti lahti voltimine.....	20
2.3.2 Tahkude komplekti piirjoone leidmine.....	21
2.3.3 Piirjoone venitamise mõju komplekti koordinaatidele .....	22
2.4 Lõigete rakendamine, pööramine, peegeldamine.....	23
3. Tulemused.....	25
3.1 Testimine.....	25
3.2 Võrdlus teiste lahendustega.....	27
3.3 Edasiarendamise võimalused .....	29
Kokkuvõte.....	30
Viidatud kirjandus.....	31
Lisad.....	32
I. Lähtekood ja kasutusjuhend.....	32
II. Küsitlus .....	33

## Sissejuhatus

Et luua videomängudele virtuaalseid keskkondi, on tarvis palju tekstuuriatud 3D mudeleid. Enamasti koosneb üks keskkond paljudest sarnastest pindadest. Näiteks tellistest seinad majadel ja puidust mööbel siseruumis. Seda omadust kasutatakse lõikelehtedel põhinevas tekstuurimise töövoos, et kunstniku tööd tõhusamaks muuta. Lõikeleht on mitut tekstuuririba sisaldav pilt. Seda kasutades saab mingi mudeli või keskkonna tekstuurida ainult ühe pildiga. Joonisel 1 on *UV Editing* vaade Blenderi tarkvaras. Vasakul on lõikeleht koos tahkude asetusega, paremal on tekstuuritud mudel.



Joonis 1. Lõikeleht tahkude asetusega (vasakul) ning tekstuuritud mudel (paremal).

Töö eesmärk on luua Blenderi tarkvarale tööriist, mis hõlbustab olemasoleva lõikelehe rakendamist mudeli tahkudele. Tööriist luuakse pistikprogrammi kujul, mis avaldatakse Blenderi laienduste veebilehel<sup>1</sup>. Seda testitakse Tartu Kunstikooli tudengite poolt, kes testivad mudeli tekstuurimist nii pistikprogrammiga kui ka ilma. Hinnatakse mõlema tekstuurimise protsessi kasutatavust *UMUX* mudeli abil.

Töö esimeses peatükis kirjeldatakse töö tausta, sealhulgas 3D modelleerimise ning tekstuurimise mõisteid, lõikelehtede töövoogu ja varasemaid lahendusi. Seal seatakse ka nõuded loodavale pistikprogrammile. Teises peatükis kirjeldatakse pistikprogrammi implementatsiooni. Seal antakse ülevaade Blenderile pistikprogrammide arendamisest, loodava rakenduse struktuurist ning kasutatavatest algoritmidest. Kolmandas peatükis analüüsitakse testimise tulemusi, võrreldakse tööriista varasemate lahendustega ning pakutakse võimalusi rakenduse edasi arendamiseks.

---

<sup>1</sup> <https://extensions.blender.org/>

## **Mõisted ja terminid**

**Lõikeleht** – pilt, millesse on paigutatud mitu erinevat tekstuuri. Tekstuurid lõikelehel on enamasti esitatud horisontaalsete või vertikaalsete ribadena.

**UV** – koordinaatsüsteem, mis seostab 3D mudeli koordinaate 2D tekstuuripildiga.

**Lõige** – üks tekstuur lõikelehel.

**Murdekoht** – serv mudeli kahe tahu vahel, mida võib tahkude lahti voltimisel kaheks lüüa.

**Tekstuurikleeps** – üks tekstuur lõikelehel, mis esindab mingit visuaalset detaili (nt silt, käepide, nupp).

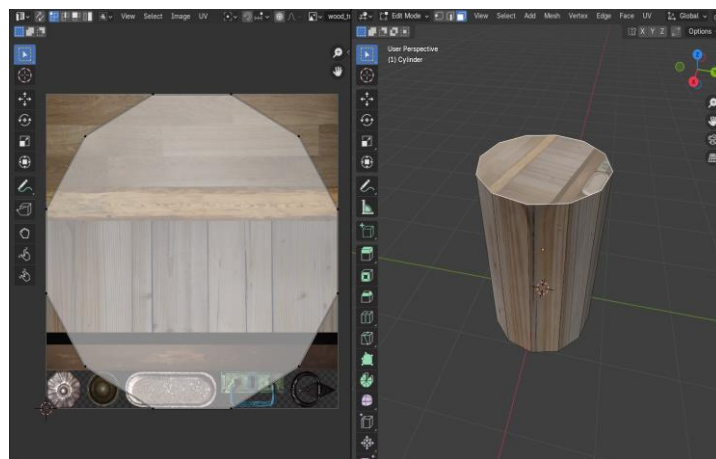
# 1. Töö taust, tekstuurimise alused

3D modelleerimiseks nimetatakse mudelite loomist 3D ruumis. See sisaldab põhilisi töövõtteid nagu mudeli või mingi mudeli osa liigutamist, pööramist või skaleerimist, aga ka täpsemaid manipuleerimisi, näiteks mudelil üksikute punktide, servade või tahkude lisamist, kustutamist ja muutmist. Sellest lähtuvalt kirjeldatakse järgnevalt tekstuurimise protsessi Blenderis, lõikelehtede töövoogu ning varasemaid lahendusi.

## 1.1 Tekstuurimine Blenderis

Blender on 1994. aastal loodud vabavaraline 3D modelleerimise tarkvara<sup>2</sup>. See on kunstnike ja teiste 3D entusiastide seas väga populaarne, sest on üks väheseid tasuta modelleerimise tööriistu, mida regulaarselt uuendatakse. Autor otsustas lahenduse realiseerida pistikprogrammina (ingl *plugin*) Blenderile, sest see on laialdaselt kasutusel ning sellele on pistikprogrammide loomine tehtud võimalikult lihtsaks.

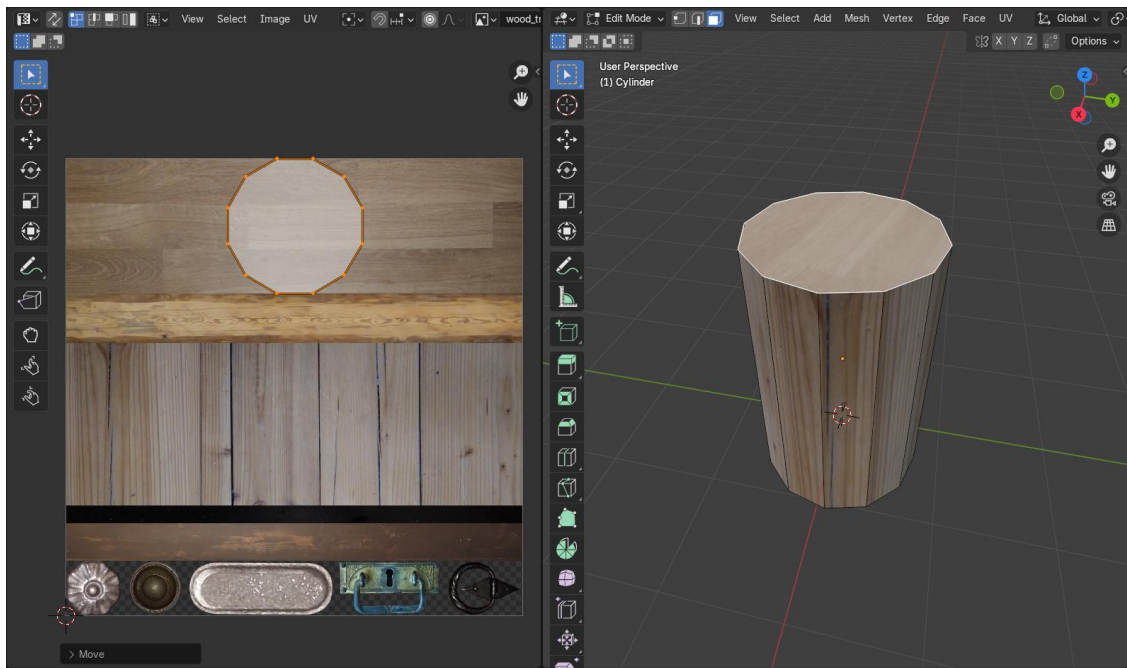
Tekstuurimine tähendab 2D pinnalaotuse tekstuuri lisamist 3D mudelile [1]. Seda pinnalaotust kasutatakse mudeli tahkude värvimiseks tekstuuri järgi. Selleks, et tekstuuri mudelile rakendada, tuleb iga mudeli tahu puhul määrata, mis ala tekstuurist see tahk katab (joonised 2-3). Blenderis toimub tekstuuride mudelile paigutamine *UV Editor* alas, mis vaikumisi asub *UV Editing* vahelehel.



Joonis 2. Algselt katab tahk *UV Editor* vaates tervet pilti. Seda on vaja liigutada, et tahul oleks näha vaid laminaatpuidu tekstuuri.

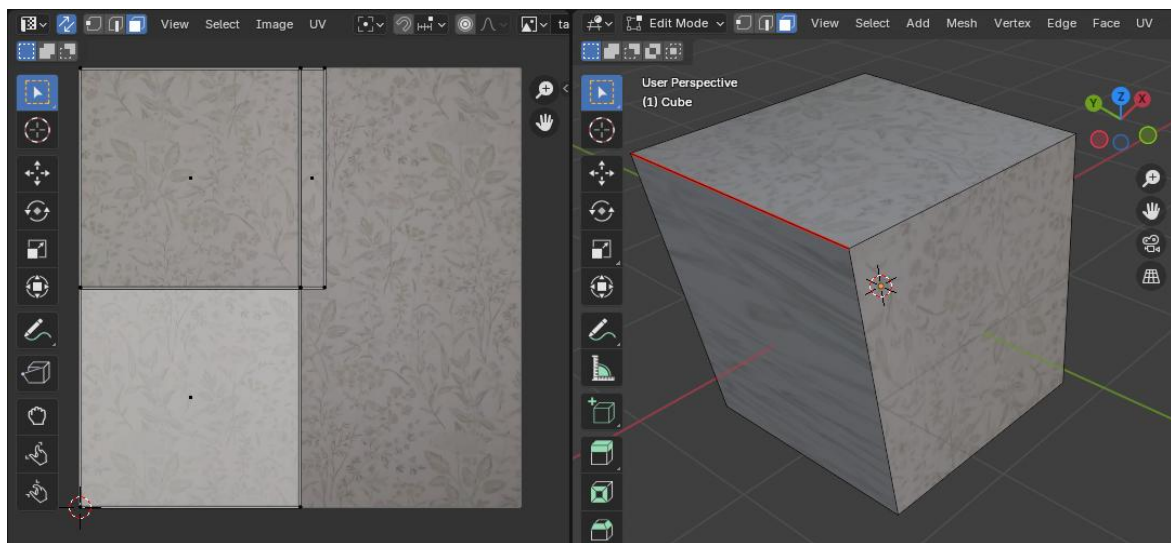
---

<sup>2</sup> <https://www.blender.org/>



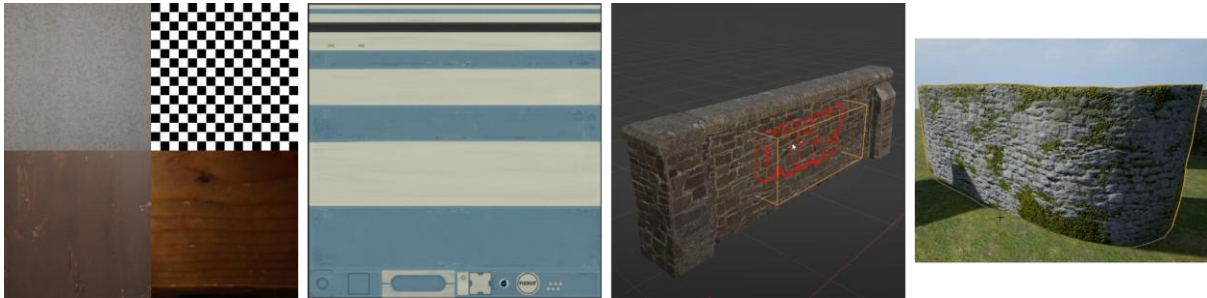
Joonis 3. Tahu UV koordinaate on skaleeritud ja liigutatud, et need sobiks laminaatpuidu tekstuuri sisse.

Täpne UV paigutus ei pruugi olla oluline, kui tekstuur on näiteks täisvärv, kuid on väga oluline mustrite, materjalitekstuuride ning piltide puhul, et tahul nähtav tekstuur poleks vales asendis, vale suurusega või venitatud. Samuti selleks, et kõrvuti tahkudel olevatel tekstuuridel oleks sujuv üleminek. Joonisel 4 on näha kuubi mudelit, kus üks külg on märgatavalt moonutatud, sest tahk on UV vaates kitsamaks tehtud.



Joonis 4. Ühe moonutatud küljega UV paigutus (vasakul) ja sellele vastav mudel (paremal).

Videomängutööstuses on tekkinud mudelite tekstuurimiseks palju erinevaid tehnikaid, mille eesmärk on töövoogu kiirendada ning tulevast mängu optimeerida. Aaltonen [2] toob erinevate tehnikatena välja näiteks tekstuuriatlased, lõikelehed, tekstuurikleepsud (ingl *decals*) ja tekstuuride segamise (ingl *texture blending*) (joonis 5).

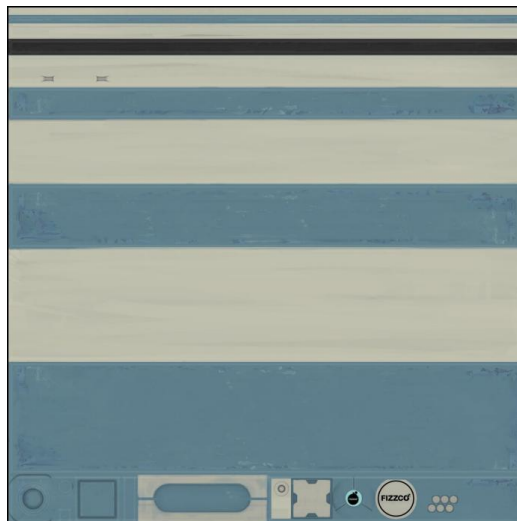


Joonis 5. Vasakult paremale: tekstuuriatlas, lõikeleht [3], tekstuurikleepsu asetamine kiviseinale [4], tekstuuride segamine kiviseinal [5].

Selles töös keskendutakse lõikelehtede tehnikale.

## 1.2 Lõikelehtede töövoog Blenderis

Radsby [6] ja Aaltonen [2] kirjeldavad lõikelehte pildina, millesse on paigutatud mitu erinevat horisontaalselt korduvat tekstuuri – lõikelehel paiknevaid tekstuure nimetatakse lõigeteks. Lisaks korduvatele tekstuuridele võib osa pildist olla jäetud ka üksikutele detailidele (joonis 6). Nõnda saab ühte pilti kasutades värvida mudeli mitme erineva tekstuuriga.



Joonis 6. Lõikeleht erinevat värvi lõigete ning muude detailidega alumises servas. [3]

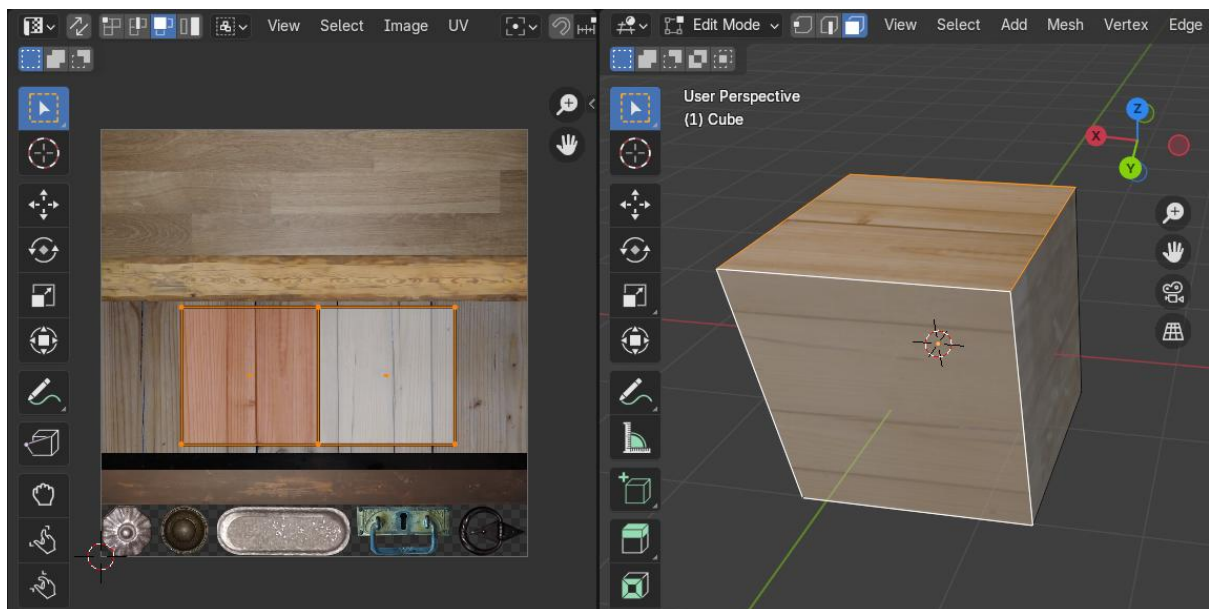
Üheks lõikelehtede eeliseks on see, et sama mudelit saab kasutada mitme erineva värviskeemiga, kui vahetada mudeli lõikeleht välja. Kui mudelile on rakendatud lõikeleht, vahetub lõikelehe pilti vahetades ka kogu mudeli lõigete komplekt (joonis 7).



Joonis 7. Üleval on maja mudel tekstuuritud ühe lõikelehega. All on sama maja, mille lõikeleht on vahetatud. [3]

Töövoo alustamiseks on vaja, et tekstuuritav mudel ning lõikelehe tekstuur oleks eelnevalt loodud. Seejärel tuleb Blenderis minna *UV Editing* vahelehele, kus aktiivne objekt on redigeerimisrežiimis (ingl *Edit Mode*). Tuleb avada lõikelehe pilt ning üksikhaaval paigutada mudeli tahud lõikelehele, neid vajadusel pöörates või skaleerides. Seda protsessi tuleb korrata mitu korda, et kõik tahud tekstuuriga katta. Samuti pole tahu paigutamine õige lõike peale kasutajasõbralik, sest liigutamise ning skaleerimise süsteem, mida ka siin kasutatakse, on loodud modelleerimiseks.

Joonisel 8 on kuubi mudel, mille kaks valitud tahku on *UV Editor* alal liigutatud keskmise lõike peale. Selleks, et soovitud tahku või tahke kindlale lõikele asetada, on vaja vastav tahkude hulk mudelil valida ning see *UV Editor* alal liigutada lõikele.



Joonis 8. Vasakul UV Editor alas on valitud kaks tahku ning liigutatud soovitud lõikele.

Kui eesmärk on tahud täpselt lõike peale venitada, tuleb neid käsitsi skaleerida nii vertikaalselt kui horisontaalselt. Vahel on vaja ka tahkude UV koordinaate pöörata, et tekstuur tahkudel õiget pidi oleks.

## 1.3 Varasemad lahendused

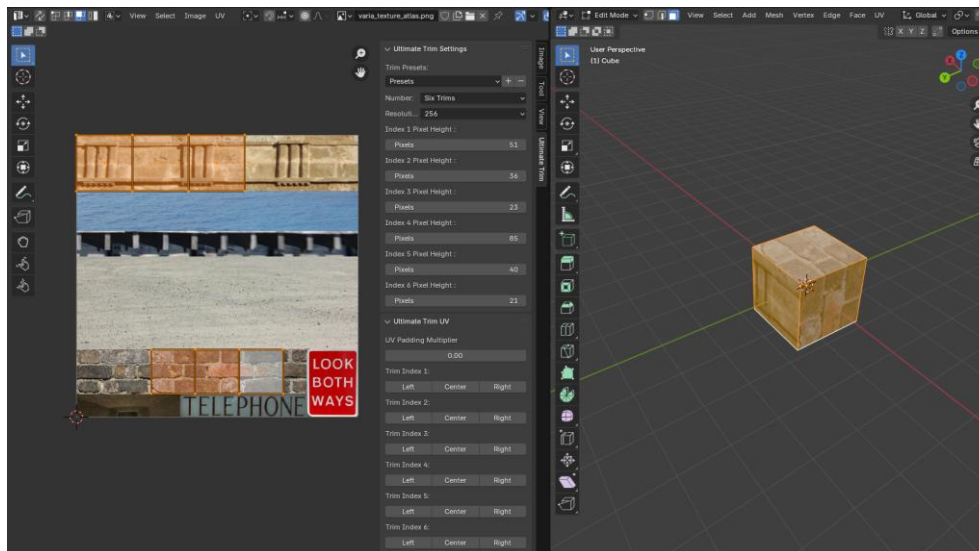
Et saada ettekujutust, millise funktsionaalsuse ja kasutajaliidesega pistikprogrammi arendada, uuriti kahte olemasolevat lahendust, mis lõikelehtede loomist ja mudelile asetamist lihtsustavad. Nendeks on Ultimate Trim UV ja DECALmachine.

### 1.3.1 Ultimate Trim UV

Ultimate Trim UV<sup>3</sup> on pistikprogramm, mis lihtsustab tahkude UV koordinaatide joendamist lõikelehele, avaldatud Justin Lazarro poolt 2018. aastal. Selles on UV laotus võimalik jaotada kuni kahekümneks horisontaalseks ribaks ehk lõikeks. Kui UV vaates valida üks tahkude saar ehk katkematu tahkude kogum, saab seda jaotatud ribadele sobitada ja joondada kas vasakule, keskele või paremale.

Lõigete kõrgust saab määrata, defineerides terve lõikelehe resolutsiooni valikmenüüst, seejärel lõike kõrguse kogu resolutsiooni suhtes (joonis 9).

<sup>3</sup> <https://jlazzaro.artstation.com/projects/V5k34>



Joonis 9. Pistikprogrammi sätetes (keskel) saab määrata lõikelehe resolutsiooni ning iga lõike kõrguse resolutsiooni suhtes. UV vaates (vasakul) on kaks pistikprogrammi abil joondatud nägude komplekti.

See tähendab, et kasutaja peab oma lõikelehe lõigete kõrguseid täpselt teadma ning ei saa neid Blenderis visuaalselt paika panna.

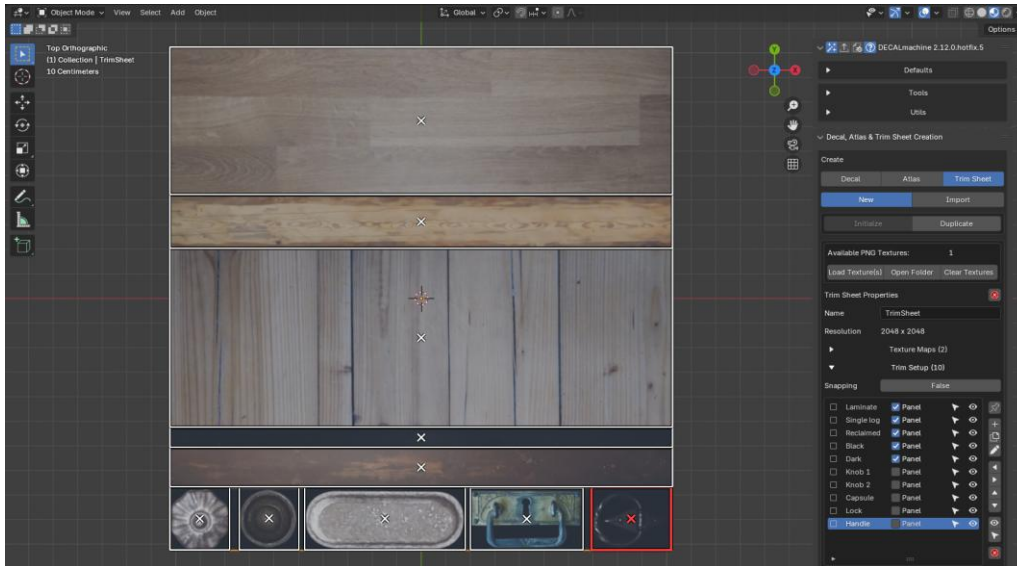
Lisatud on lõigete sisemise ääre (ingl *padding*) märkimise võimalus, kuid selle väärtus pole seotud kasutaja seatud resolutsiooni väärtusega, vaid iga siseäärise ühik moodustab 1/2000 tervest pildist.

### 1.3.2 DECALmachine

Võrreldes Ultimate Trim UV pistikprogrammiga on DECALmachine<sup>4</sup> (avaldatud 2017. aastal) võimsam, aga ka tasuline ning keskendub rohkem tekstuuriklepsudele kui lõikelehtedele. Siiski on rakendusega võimalik lõikelehti nii luua kui ka mudelile rakendada.

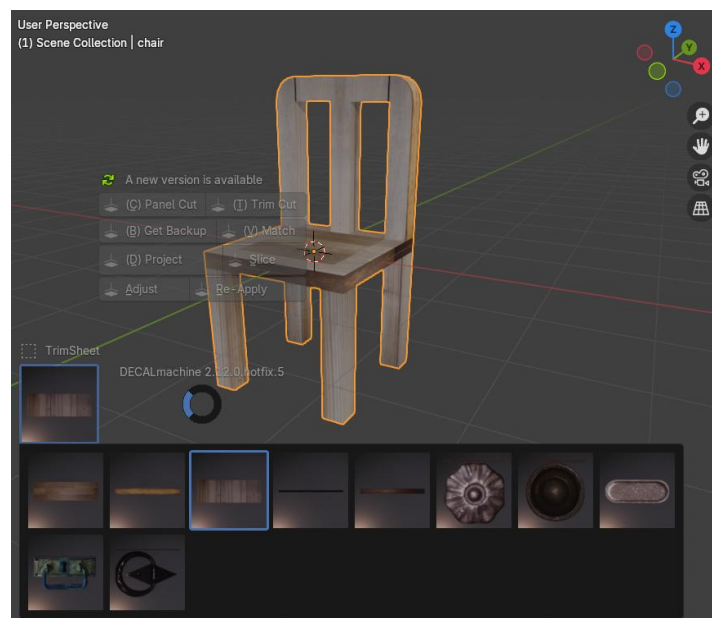
Lõikelehtede loomiseks antakse ette tasapind, millele on vaja lisada lõikelehe pilt. Joonisel 10 on näha, kuidas saab oma äranägemise järgi lisada lõikeid ning neid DECALmachine'i kasutajaliidesega liigutada ja skaleerida. All paremal menüüs saab muuta lõigete nimesid ning määrata nähtavust ja lukustatust. Iga lõike puhul, mis horisontaalselt kordub, peab menüüs märkima märkeruudu *Panel*. Loodud lõikeleht on võimalik ka salvestada ning hiljem pistikprogrammi otse sisse laadida.

<sup>4</sup> <https://machin3.io/DECALmachine/docs/>

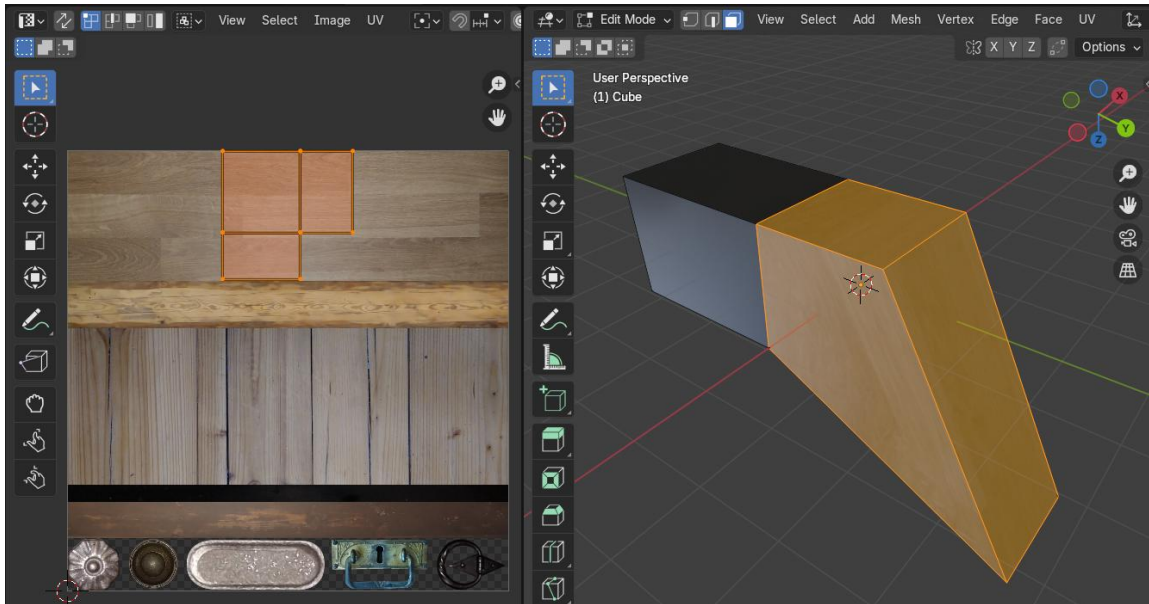


Joonis 10. DECALmachine'i kasutajaliides.

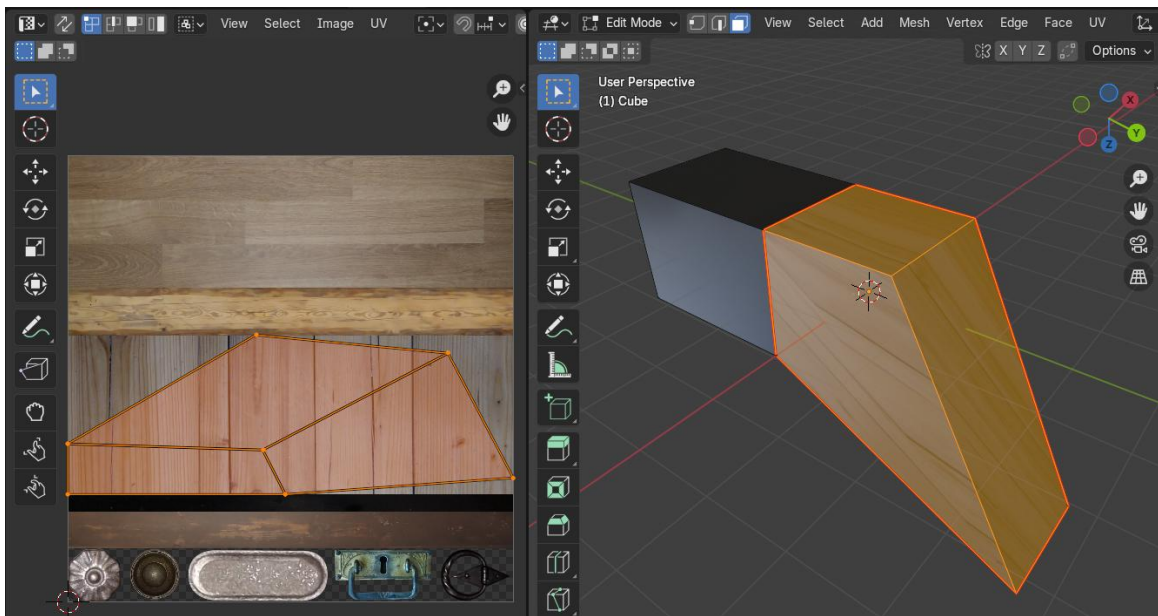
Peale lõikelehe loomist on võimalik pistikprogrammi kiirvalikust (joonis 11) rakendada lõiget aktiivsele tahule või tahkude kombinatsioonile. Tahkude kombinatsiooni korral võidib DECALmachine tahud lahti. Tahkude lahti voltimiseks on kaks võimalust. Kui rakendataval lõikel on märgitud *Panel*, siis eelistatakse murdekohtade tegemist ning moonutamise vältimist. Kui lõikel pole märgitud *Panel*, siis eelistatakse moonutamist (joonised 12-13).



Joonis 11. Kiirklahvi D vajutades on näha DECALmachine'i salvestatud lõikeid.



Joonis 12. Tahkude lahti voltimine lõikele tüübiga “Panel”.



Joonis 13. Tahkude lahti voltimine lõikele, mis pole tüübiga “Panel”.

Kui tahud volditakse lahti lõikele tüübiga *Panel*, siis pärast tahkude lahti voltimist venitatakse need tahud ristkülikuteks.

See tähendab, et kui ühel ja samal lõikel soovitakse tahkude komplekte lahti voltida nii moonutustega kui ka murdekohtadega, tuleb lõige luua kaks korda: üks koos *Panel* märkega ja teine ilma. Kogenenud kasutajale, kelle lõikelehtedel leidub ka palju tekstuurikleepse, on DECALmachine hea lahendus. Algajale kasutajale on DECALmachine pigem keeruline kasutada ning talle sobiks paremini just lõikelehtedele fokuseeritud tööriist.

## 1.4 Nõuded

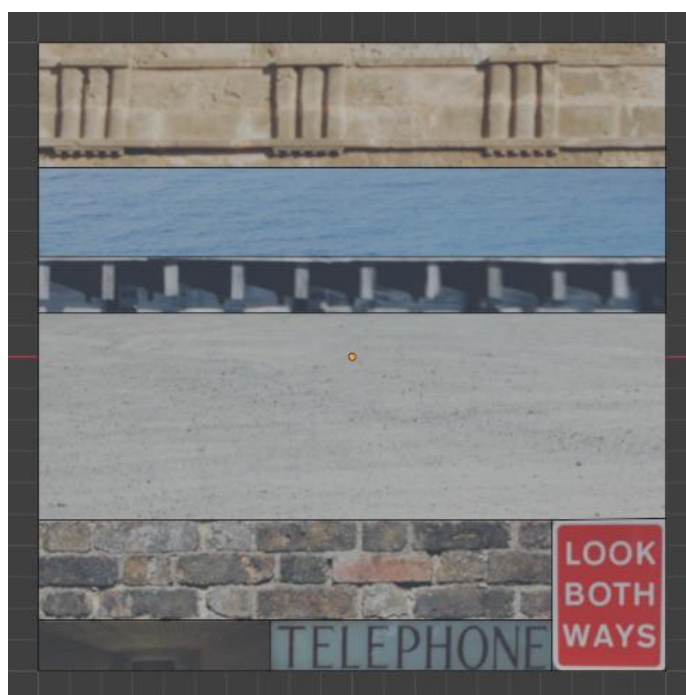
Varasemate lahenduste analüüsi põhjal said töös loodavale rakendusele seatud kolm funktsionaalset nõuet. Rakendus peab võimaldama:

- defineerida lõikeleht, mis koosneb erinevate paigutustega ristkülikukujulistest lõigetest;
- asetada lõige ühele või mitmele valitud tahule;
- muuta lõike paigutust pärast selle asetamist.

Järgnevalt on iga punkti lähemalt selgitatud.

### 1.4.1 Lõikelehe defineerimine

Sarnaselt DECALmachine pistikprogrammile defineeritakse lõikeleht, märkides iga ristkülikulise lõike asukoht lõikelehel. DECALmachine'is käib lõigete asukoha määramine läbi kohandatud kasutajaliidese. Loodavas rakenduses tehakse seda, jagades lõikelehte esindav tasapind tahkudeks, kus iga tahk esindab ühte lõiget (joonis 14).



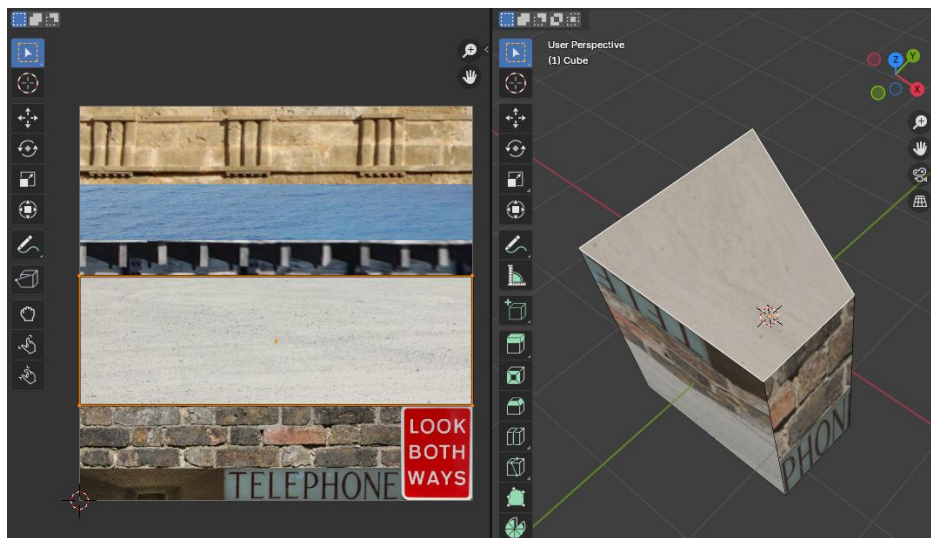
Joonis 14. Lõikeleht on paigutatud ruudule, mis on jagatud tahkudeks.

Selle lahenduse plussiks on see, et defineeritud lõikelehel algab järgmine tahk täpselt sealt, kust eelmine lõpeb – DECALmachine'i kasutades peab kasutaja ise püüdma lõike piirid võimalikult täpselt tekstuuride ümber joonistama.

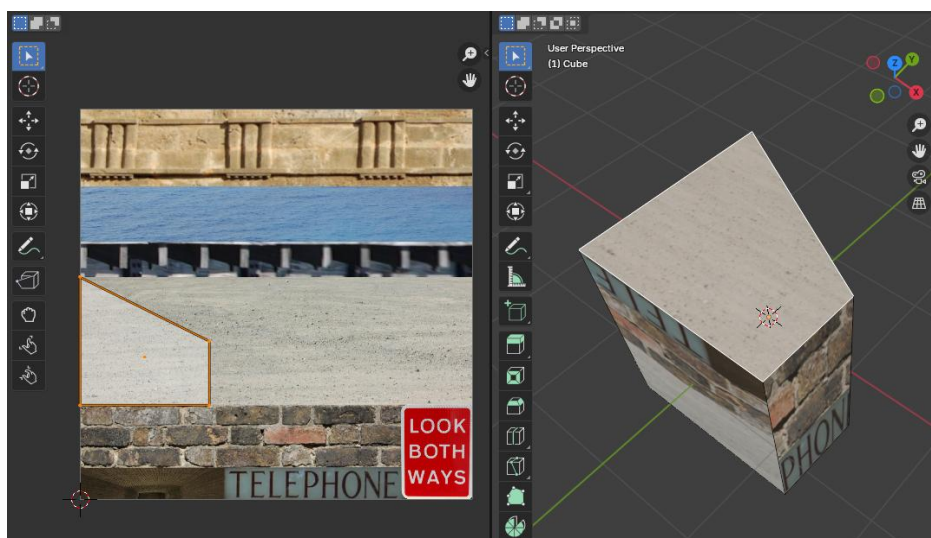
## 1.4.2 Lõike asetamine

Peamine funktsionaalsus on lõike rakendamine valitud tahule nii, et visuaalselt kataks lõige tahku võimalikult täpselt. Selleks saab rakendada kahte meetodit:

- Täitmine (ingl *fill*) – tahk venitatakse lõike peale nii, et see katab kogu lõike pinna (joonis 15).
- Sobitamine (ingl *fit*) – tahku skaleeritakse ilma proportsioone muutmata. Tahk mahub täpselt lõike sisse (joonis 16).

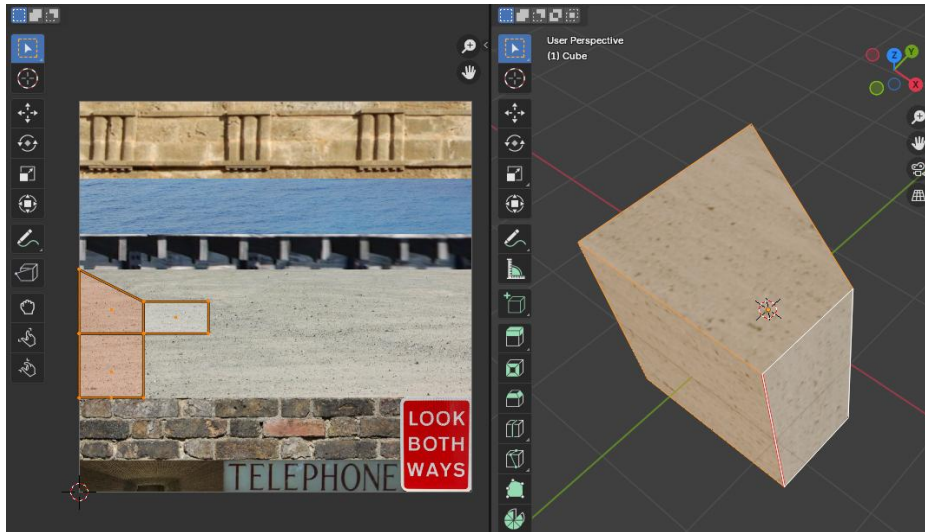


Joonis 15. Paremäl asuval mudelil on valitud ebakorrapärane tahk. Vasakul Blenderi UV Editor vaates on tahk venitatud terve lõike peale.



Joonis 16. Vasakul Blenderi UV Editor vaates on valitud tahk sobitatud lõike sisse ilma, et tahu proportsioonid muutuksid.

Samuti on oluline, et lõike rakendamiseks saaks valida mitu tahku korraga. Sel juhul peab esmalt tahkude komplekti tasapinnale lahti voltima, peale mida saab lõike valitud tahkudele sobitada (joonis 17).



Joonis 17. Vasakul Blenderi UV Editor vaates on valitud tahud esmalt lahti volditud ning seejärel sobitatud lõike sisse.

Mitme tahu komplektile lõike rakendamiseks on vaja implementeerida kahte algoritmi:

- 3D ruumis asetseva tahkude komplekti 2D tasandile lahti voltimine;
- lahti volditud kujundi sobitamine lõikele.

Lahti voltimise algoritm on vajalik, et 3D ruumis olevaid tahke ilma moonutuse või ülekatteta UV tasandile tõsta. Peale tahkude komplekti lahti voltimist on vaja need sobitada lõike kujule. Kui komplekti välimised punktid on paigutatud, on vaja ümber tõsta ka sisemised punktid. Selle ümber tõstmise jaoks sobib barütsentriliste koordinaatide süsteem, mis annab igale punktile unikaalse kaalukomplekti ümbritsevate punktide suhtes.

Kui lõiget pole mingil põhjusel võimalik tahkude komplektile rakendada (näiteks komplekt pole lahti volditav), kuvatakse kasutajale veateade.

### 1.4.3 Lõike paigutuse muutmine

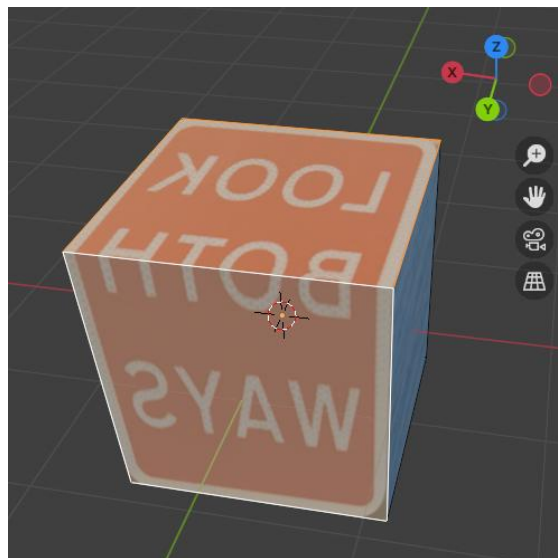
Peale tahule lõike rakendamist on aeg-ajalt vaja tekstuuri pöörata või peegeldada. Vahel on see vajalik, et tahk parem välja näeks ja muude tahkudega visuaalselt kokku sobiks. Joonisel 18 on kujutatud kaks pilti tooli mudelist. Vasakpoolsel pildil on vaja loomuliku välimuse

saavutamiseks ühe jala tekstuuri pöörata peale seda, kui see on mudeli tahule asetatud. Parempoolsel pildil on sama mudel, millel on tekstuur pööratud.



Joonis 18. Tool halvasti asetatud tekstuuriga (vasakul) ja tool pööratud tekstuuriga (paremal).

Vahel on tarvis ka lõiget peegeldada. Näiteks siis, kui lõikel on tekst, kuid see on tagurpidi lahti volditud (joonis 19).



Joonis 19. Kuubi mudel, millel on peegelpildis tekstiga silt.

Nii lõike paigutuse pööramist kui peegeldamist on võimalik teha Blenderis ka käsitsi. Siiski on need funktsioonid loodava lahenduse nõueteks, et lõike lõpliku paigutuse saavutamist kiirendada.

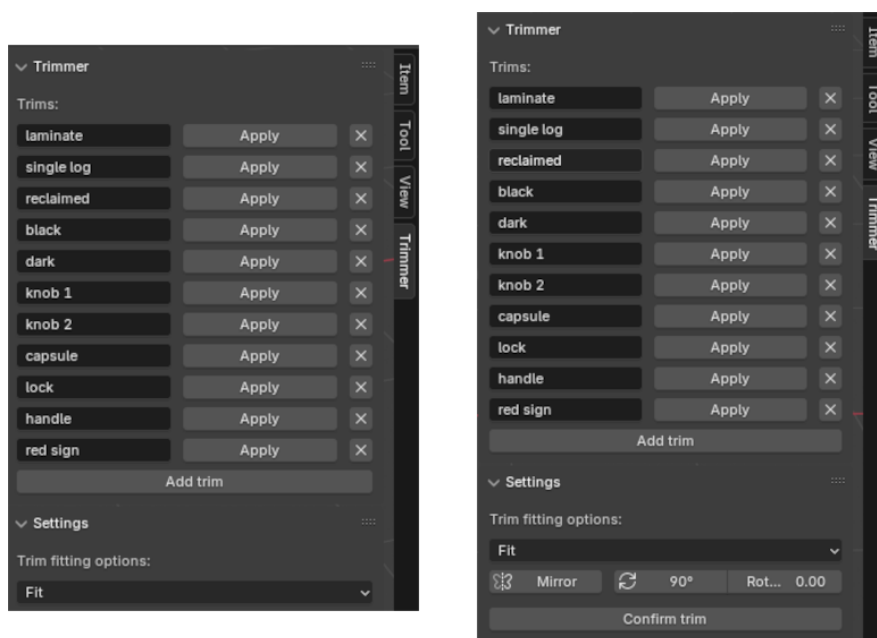
## 2. Implementatsioon

Blenderi mootor on kirjutatud C ning C++ keeles, kuid sellele laienduste loomine toimub keeles Python. Blenderi pistikprogramm võib sisaldada nii kohandatud algoritme kui ka kasutajaliidese elemente. Antud lahendus koosneb neljast peamisest osast: metaandmed, kasutajaliides, andmete haldus ning algoritmide implementatsioonid.

### 2.1 Metaandmed, kasutajaliides

Metaandmete failis (`__init__.py`) on info pistikprogrammi kohta (nt programmi nimi, autor, versioon, lühikirjeldus) ning meetodid programmi klasside registreerimiseks ning eemaldamiseks.

Kasutajaliidese failis (`ui.py`) olevad klassid defineerivad Blenderis nähtavate paneelide välimuse (joonis 20) ning interaktiivsete elementide (nt nupud, liugurid, valikmenüüd) käitumise.



Joonis 20. Paneelide vaikimisi välimuse (vasakul) ja paneelide välimuse peale lõike rakendamist (paremal).

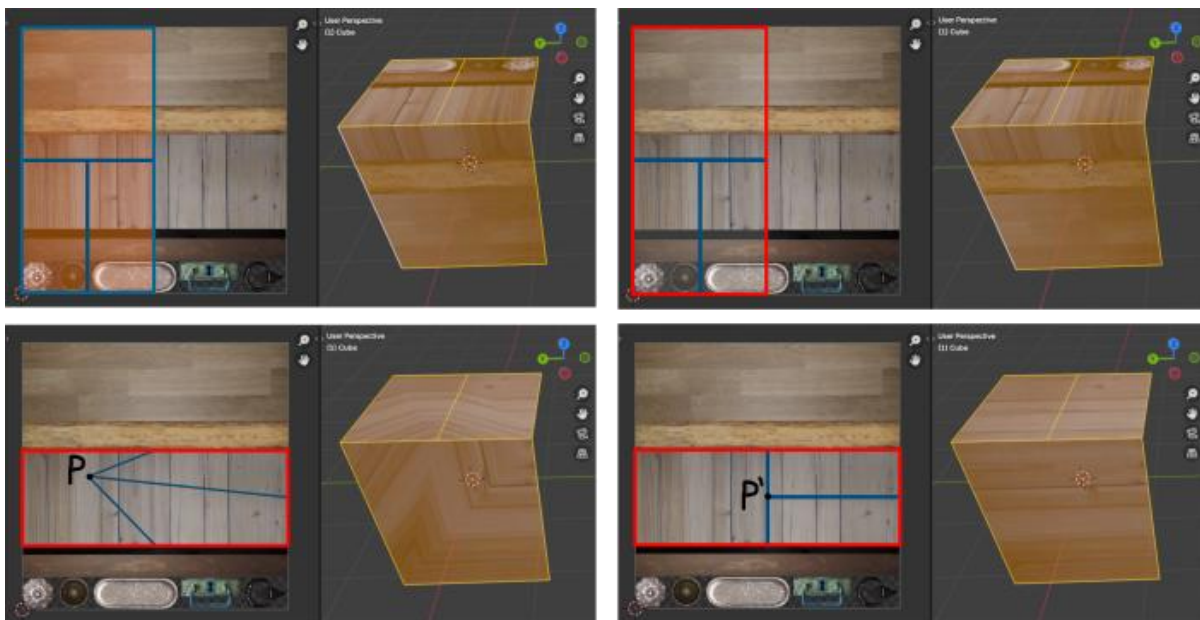
Vasakul on näha paneelide vaikimisi välimust. Ülemisel paneelil *Trimmer* on seadistatud lõiked. Alumisel paneelil *Settings* on ainult lõike paigutusviisi valikmenüü. Paremal on näha kasutajaliides peale lõike tahule rakendamist. *Settings* paneelil on nüüd ka lõike peegeldamise, pööramise ning kinnitamise nupud.

## 2.2 Andmete haldus

Andmete halduse failis (*trimmer.py*) toimub Blenderi objektide (sh UV koordinaatide) manipuleerimine. Siin on klassid *Trimmer* ja *Trim*, mille peamised ülesanded on Blenderi andmetüüpide teisendamine koordinaatideks ja vastupidi. Massiivina esitatud koordinaate kasutatakse pistikprogrammi algoritmides.

## 2.3 Algoritmid

Kui kasutaja soovib rakendada lõiget tahkude komplektile, on vaja see sobivalt lahti voltida ning seejärel asetada UV koordinaadid lõikele. Kui soov on tahkude komplektiga mingit lõiget täita ehk kasutada *fill* valikut, peab leidma komplekti piirjoone ning panema selle vastavusse lõike piirjoonega. Seejärel tuleb leida uued tahkude koordinaadid ka punktidele, mis ei asunud komplekti piirjoonel. Seda lõike rakendamise protsessi visualiseerib joonis 21. Üleval vasakul on kolmest tahust koosnev mudel, mis on *UV Editor* vaates lahti volditud. Üleval paremal on leitud tahkude komplekti piirjoon, märgitud punaselt. All vasakul on piirjoon viidud vastavusse lõike piirjoonega. Üks sisemine punkt P on endiselt oma kohal, mistõttu tahkudel olev tekstuur on moonutatud. All paremal on ka punkt P viidud oma uuele koordinaadile P'.

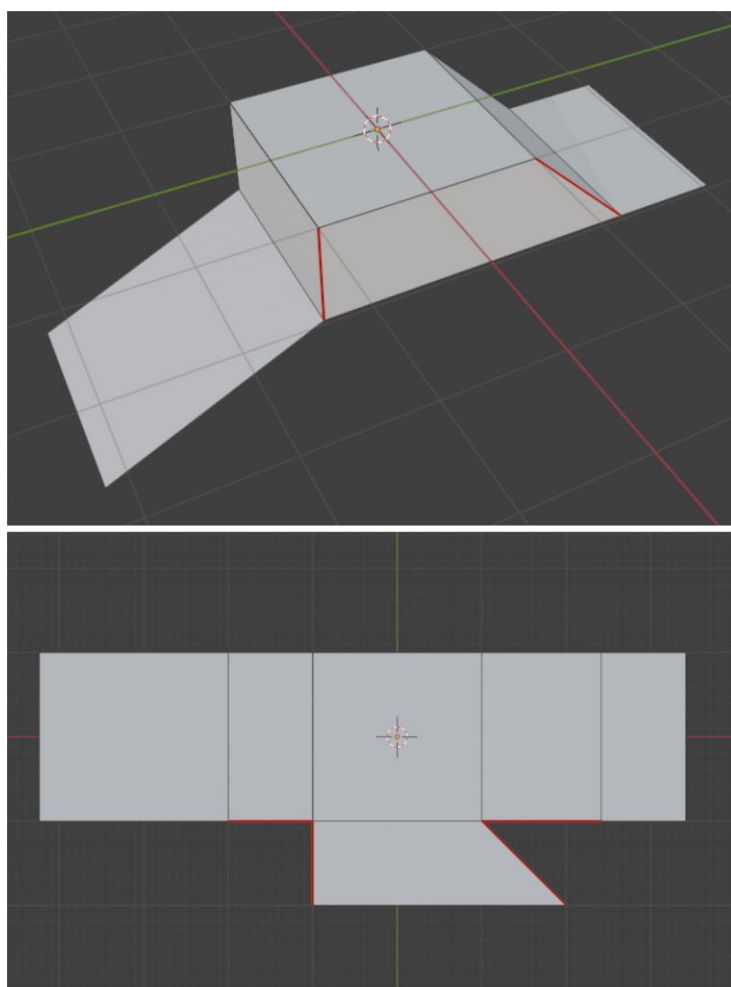


Joonis 21. Tahkude komplektile lõike täitmise protsess.

Kõikide algoritmide arendamisel kasutati automaatsete, et leida kiiremini üles vigu ning viia läbi regressioontestimist.

### 2.3.1 Tahkude komplekti lahti voltimine

Üheks lahenduse nõudeks on mitmele tahule tekstuuri rakendamine. Et valitud tahud ei pruugi olla samatasandilised, on vaja algoritmi nende lahti voltimiseks. Autor otsustas arendada algoritmi ise, sest ei leidnud olemasolevat algoritmi, mis väldiks tahkude moonutamist ning võimaldaks märkida servi murdekohtadena. Joonis 22 illustreerib lahti voltimist. Üleval on näha algset tahkude komplekti. Kaks serva (punased) on märgitud murdekohtadena. All on tahkude komplekt lahti volditud ilma, et ükski tahk oleks moonutatud.



Joonis 22. Üleval on tahkude komplekt 3D ruumis, all on lahti volditud tahud 2D tasandil.

Arendatud algoritm saab sisendina tahkude massiivi, kus iga tahku esindab selle otspunktide koordinaatide massiiv. Esmalt luuakse graaf, kus tipud esindavad tahke ning kaared tahkude vahelisi servi. Kui mingi serv on märgitud murdekohtaks, siis vastavat kaart graafi ei lisata.

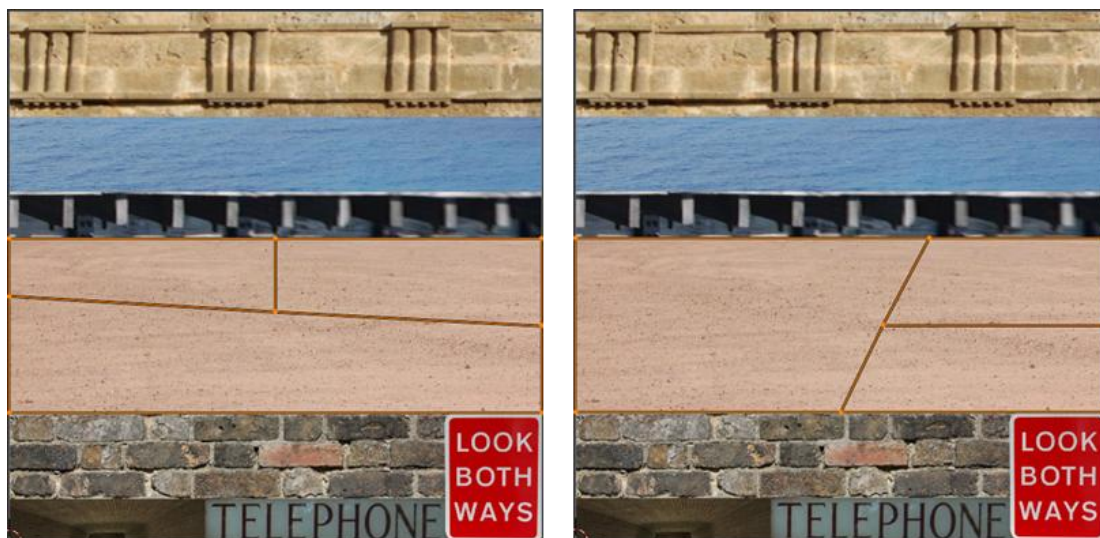
Seejärel alustatakse tahkude kaardistamist järgnevalt. Massiivi esimene tahk paigutatakse 2D tasapinnale. Tahkude graafist valitakse sellega külgnevad tahud, mis lisatakse järjekorda koos viitega esialgsele tahule. Kui järjekord pole tühi, tehakse järgnevat.

1. Järjekorrast võetakse järgmine tahk T ning sellega külgnev tahk KT. Kui tahku T on tahu KT suhtes juba lahti volditud, liigutakse järjekorra järgmise elemendi juurde.
2. Järjekorda lisatakse tahuga T külgnevad tahud koos viitega, et nendega külgnev tahk on T.
3. Tahk T paigutatakse 2D tasapinnale ning liigutatakse ja pööratakse nii, et see külgneks tahuga KT.
4. Kui tahku T on juba eelnevalt lahti volditud, kontrollitakse, kas selle uus asukoht on võrdne eelneva asukohaga. Kui ei, siis ei saa mudelit ilma moonutuseta lahti voldida ja kasutajale antakse vastav veateade. Kui on, siis märgitakse tahk T lahti voldituks ning liigutakse tagasi sammu 1.

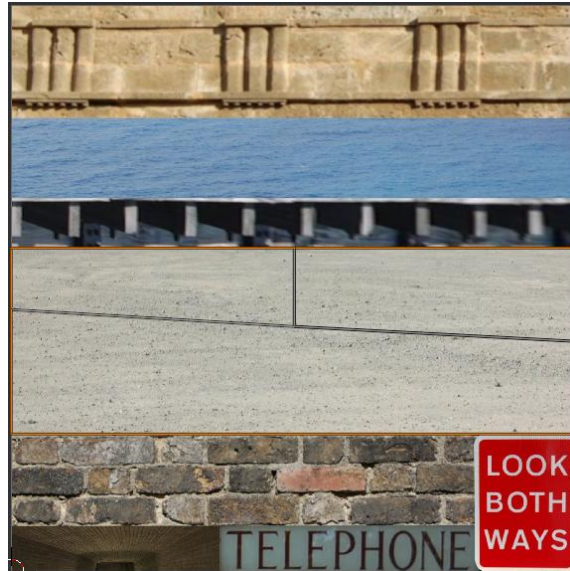
Peale järjekorra lõppemist tagastatakse lahti volditud tahud.

### 2.3.2 Tahkude komplekti piirjoone leidmine

Tahkude komplekti puhul on nii lõikele venitamiseks kui UV servakaupa pööramiseks (joonis 23) vaja algoritmi tahkude komplekti piirjoone leidmiseks. Algoritmi sisendiks on tahkude komplekt 2D koordinaatidena. Selle põhjal on vaja leida tahkude piirjoon (joonis 24).



Joonis 23. Vasakul on näidatud lahti volditud kolme tahu komplekti. Paremal on komplekti ühe serva võrra pööratud.



Joonis 24. Oranži värviga on märgitud kolme tahu komplekti piirjoon.

Esmalt leitakse esimesed punktid koordinaatide massiivis, mida ühendav serv puutub kokku vaid ühe tahuga. Sel juhul asub serv piirjoonel. Serva mööda liigutakse piki piirjoont edasi, kuni kõik piirjoone punktid on läbi käidud. Selle käigus jäetakse alles mittekollineaarsed punktid ehk need punktid, mis ei asu täpselt eelmise ja järgmise punkti vahel. Tagastatakse piirjoone mittekollineaarsete punktide komplekt.

### 2.3.3 Piirjoone venitamise mõju komplekti koordinaatidele

Lahti volditud tahkude komplekti piirjoon venitatakse lõike piirjoonele. Peale seda on vaja arvutada komplekti sisemiste ehk mitte piirjoonel asuvate punktide uus asukoht. Seda probleemi saab lahendada, andes igale komplekti punktile uue koordinaadi, mis on defineeritud piirjoone suhtes.

Autor valis lahenduseks keskvärtuskoordinaatide süsteemi (ingl *Mean Value Coordinates*), mis on loodud 2003. aastal Floateri poolt [7]. Olgu suvaline kujund, mille piirjoon koosneb punktidest  $v_1, \dots, v_k$ . Iga  $i \in \{1, \dots, k\}$  puhul  $v_i$  ja  $v_{i+1}$  vahel on serv, lisaks on serv ka  $v_k$  ja  $v_1$  vahel. Keskvärtuskoordinaatide süsteemiga on võimalik määrata igale kujundi punktile kaalude komplekt  $\lambda_1, \dots, \lambda_k$ , kus iga kaal on vastavuses ühe piirjoone punktiga ning kehtivad valemid (1) ja (2).

$$\sum_{i=1}^k \lambda_i v_i = v_0 \quad (1)$$

$$\sum_{i=1}^k \lambda_i = 1 \quad (2)$$

Olgu  $v_0$  suvaline punkt kujundil. Valemi (1) kohaselt on iga  $v_0$  punkti koordinaadid väljendatavad selle punkti kaalude ning vastavate piirjoone punktide korrutiste summana – see omadus kehtib ka siis, kui piirjoone punkte  $v_1, \dots, v_k$  on liigutatud. Valemi (2) kohaselt on kaalud normaliseeritud ehk nende summa võrdub ühega.

Keskvärtuskoordinaatide süsteemi puhul leitakse punkti  $v_0$  kaalude komplektis  $\lambda$  iga kaal  $\lambda_i$  valemitega (3) ja (4).

$$w_i = \frac{\tan(\frac{\alpha_{i-1}}{2}) + \tan(\frac{\alpha_i}{2})}{\|v_i - v_0\|} \quad (3)$$

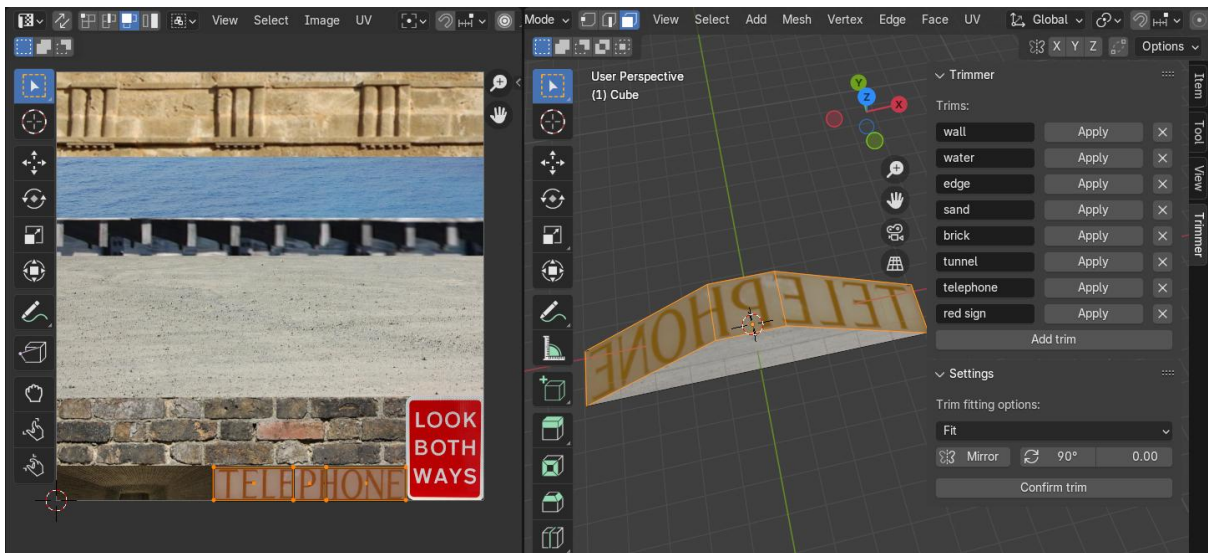
$$\lambda_i = \frac{w_i}{\sum_{j=1}^k w_j} \quad (4)$$

Valemis (3) on  $\alpha_i$  vektorite  $v_0 v_i$  ja  $v_0 v_{i+1}$  vaheline nurk. Esialgse kaalu arvutamiseks leitakse nurgad  $\alpha_{i-1}$  ning  $\alpha_i$ , mis jagatakse kahega. Saadud nurkade tangensite summa jagatakse  $v_i$  ja  $v_0$  vahelise distantsiga. Valemis (4) kaalud normaliseeritakse, et saada lõplikud kaalud.

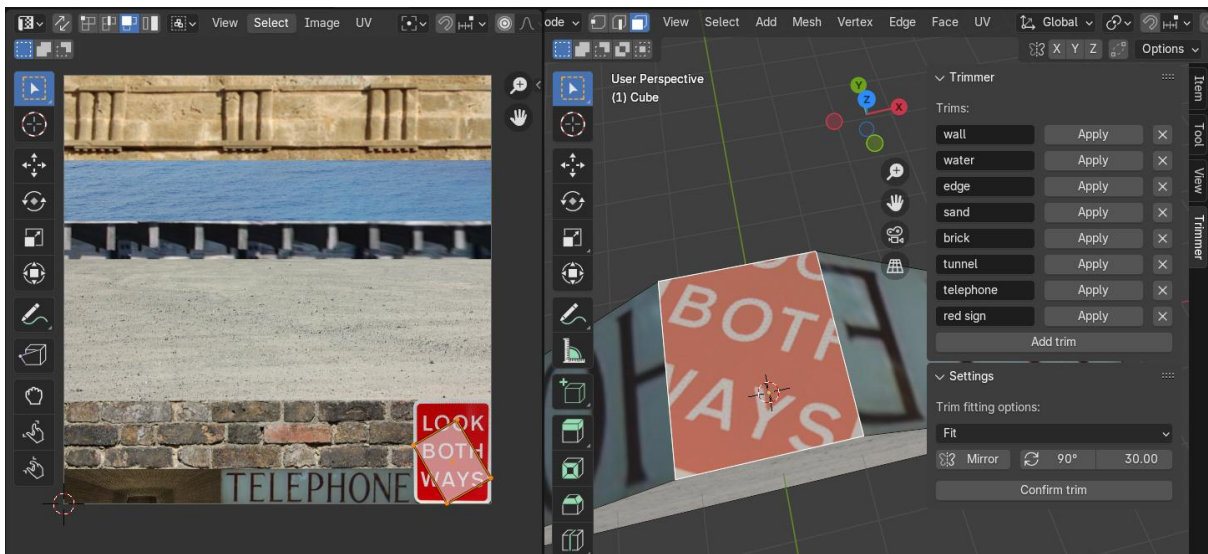
## 2.4 Lõigete rakendamine, pööramine, peegeldamine

Lõigete rakendamine on implementeeritud järgnevalt. Peale lõigete pistikprogrammi lisamist valib kasutaja tahu või tahkude komplekti ning vajutab nupule *Apply*. Sõltuvalt rakendusvalikust (*fit* või *fill*) viiakse tahkude komplekti UV koordinaadid lõike peale (joonised 15-16, alapeatükk 1.4.2).

Peale *Apply* vajutamist on võimalus lõiget tahkudel pöörata või peegeldada. Peegeldamine toimub vastu vertikaaltelge (joonis 25). Pööramine teostatakse *fill* valiku korral servapidi (joonis 23, alapeatükk 2.3.2), *fit* valiku korral pööratakse tahkude komplekti määratud kraadide võrra (joonis 26).



Joonis 25. Mudeli tahkudele rakendatud “telephone” lõiget on peegeldatud.



Joonis 26. Mudeli tahule rakendatud “red sign” lõiget on 30 kraadi pööratud.

Kuna enamasti on lõikelehtedes kasutatavad lõiked ristküliku kujulised, on neid üldjuhul vaja pöörata 90 kraadi kaupa. Seetõttu on vastav võimalus *fit* valiku puhul eraldi nupuna lisatud.

## 3. Tulemused

Loodud rakendust hinnati kahel viisil. Esiteks testiti loodud pistikprogrammi Tartu Kunstikooli tudengite poolt, kes võrdlesid lahendust Blenderi vaikimisi võimekusega ning hindasid lahenduse kasutatavust. Teiseks võrdles autor lahenduse funktsionaalsust varasemate lahendustega. Lahendus lisati Blenderi pistikprogrammide kogumisse, kus seda on tasuta võimalik alla laadida ning kasutajad saavad sellele tagasisidet anda.

### 3.1 Testimine

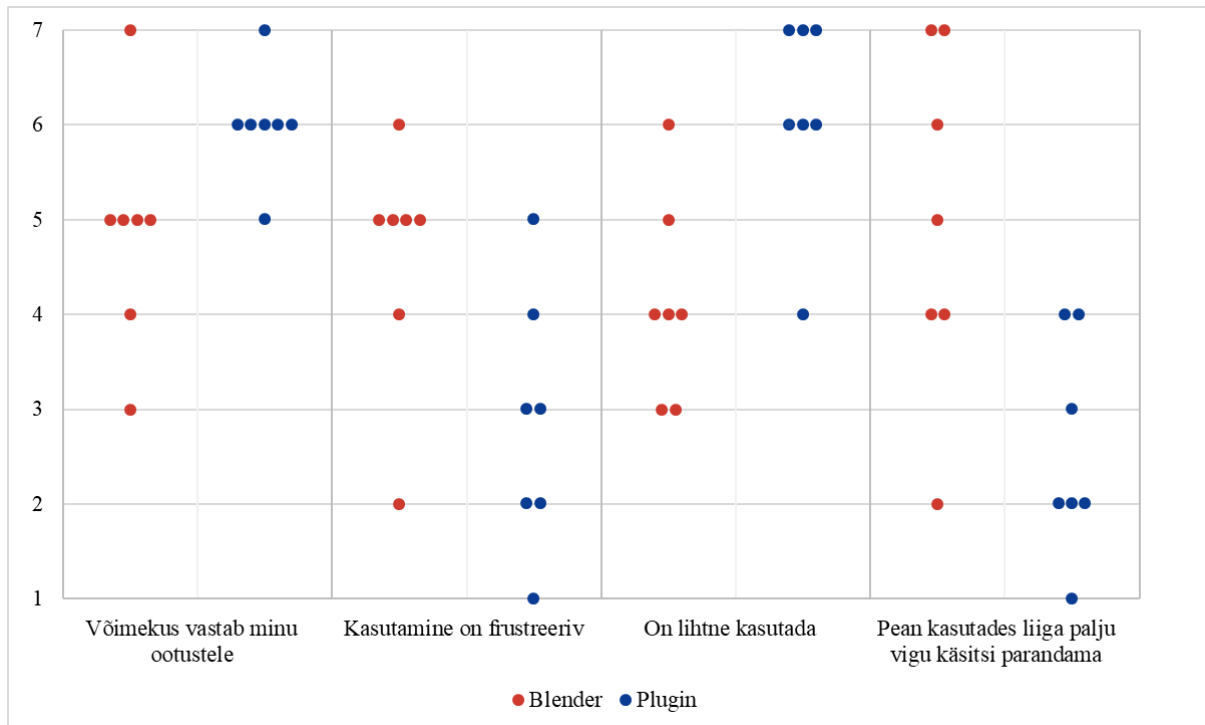
Testimine viidi läbi 2025. aasta aprillis Tartu Kunstikoolis. Testijateks oli 10 kooli õpilast, kes osalesid lõikelehtede töövoogu käsitlevas tunnis, mille viis läbi töö autor. Õpilastele tutvustati lõikelehtede töövoogu ning loodud pistikprogrammi. Seejärel tehti sama mudeli tekstuurimine läbi kaks korda – esmalt ainult Blenderi tööriistu kasutades ning seejärel koos pistikprogrammiga. Testimise eesmärk oli hinnata kasutusjuhendi arusaadavust, leida pistikprogrammis vigu ning kvalitatiivselt mõõta selle kasutatavust. Mõlema tekstuurimise protsessi jooksul küsisid testijad suuliselt abi, kui midagi segaseks jäi. Samuti täideti mõlema protsessi kohta küsimustik. Küsimustiku peamine eesmärk oli analüüsida pistikprogrammi kasutatavust, lisaks korjati sellega sisendit edasiarenduste kohta. Küsimustiku kasutatavust analüüsiv osa koostati kasutajakogemuse kasutatavuse mõõdiku (ingl *Usability Metric for User Experience* ehk *UMUX*) põhjal [8].

Küsimused olid väidete kujul, millele sai vastata numbriga skaalal 1-7, kus 1 tähendab “Ei nõustu üldse” ning 7 tähendab “Nõustun täielikult”. Lisaks oli iga numbrilise vastusega küsimuse järel võimalus oma vastust pikemalt põhjendada. Väited olid:

- “... võimekus vastab minu ootustele”;
- “... kasutamine on frustreeriv”;
- “... on lihtne kasutada”;
- “Peaaegu ... kasutades liiga palju vigu käsitsi parandama”.

Analoogsele küsimuste komplektile tuli vastata nii Blenderi vaikimisi võimekuse kui ka Trimmeri kohta. Selline lähenemine võimaldas võrrelda kasutajate töövoogu ja näha, millised probleemid neil tekivad nii pistikprogrammi kasutades kui ka ilma selleta.

Küsitlus sai 7 vastust, mida on näitatud joonisel 27. Horisontaalteljel on väited, vertikaalteljel on väitega nõustumise hinnang skaalal 1-7. Punased täpid esindavad vastust Blenderi (ilma pluginata) kohta käinud väitele, sinised täpid käivad Trimmeri kohta. Võrreldes Blenderile ja Trimmerile antud hinnanguid, on igas kategoorias Trimmeri hinded veidi paremad. Kõige suuremad vahed Blenderi (ilma pluginata) ning Trimmeri vahel oli väidetel “... on lihtne kasutada” ja “Peaaegu ... kasutades liiga palju vigu käsitsi parandama”. Seda toetavad ka kirjalikud kommentaarid.



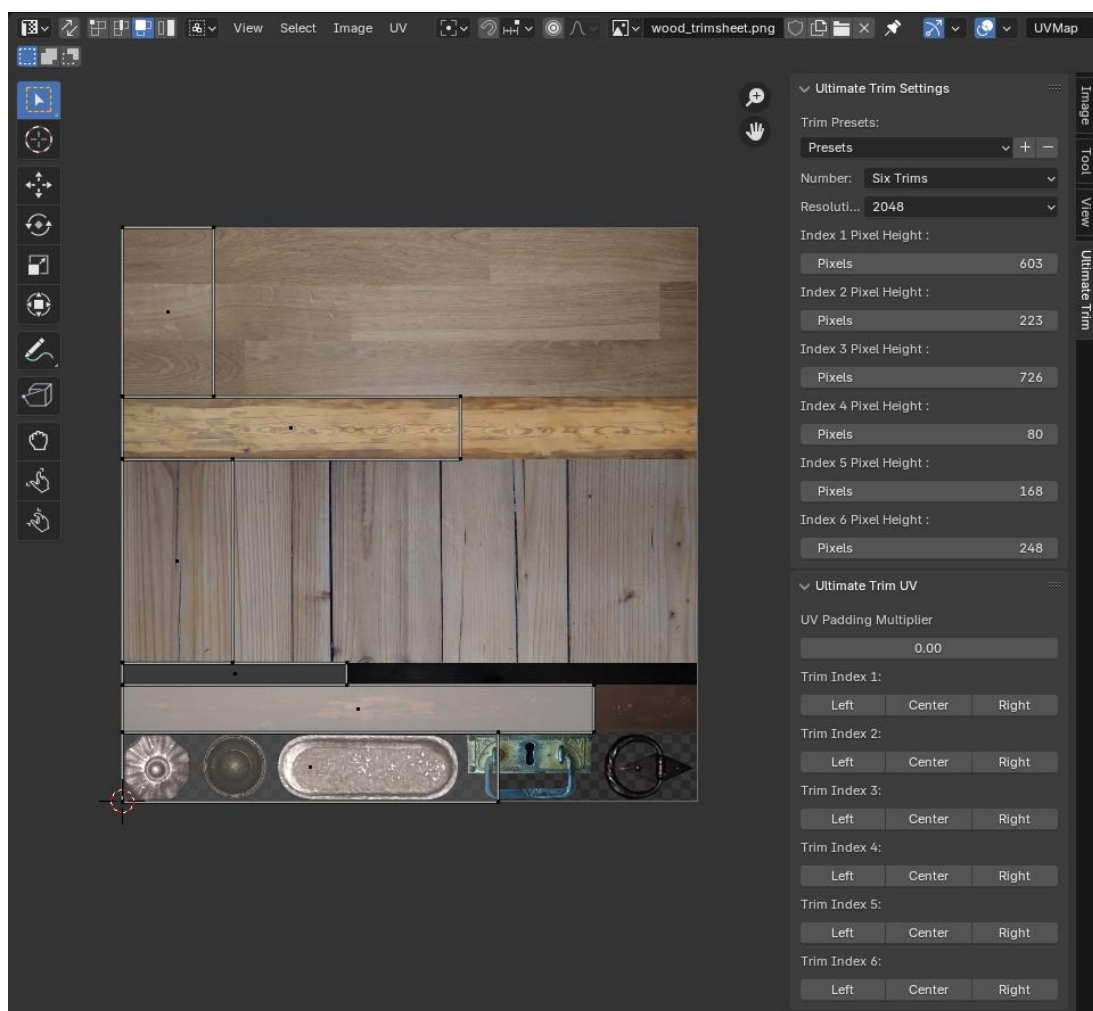
Joonis 27. Testimise küsitluse tulemused.

Peamised kommentaarid, mis testijatel lahendust kasutades tekkisid, olid seotud edasiarendustega. Näiteks mainiti, et võiks olla võimalus valida mitu tahku korraga ning *Apply* vajutamisel käsitleda neid mitte tahkude komplektina, vaid kui eraldi tahke, millele ükshaaval lõiget rakendada. Samuti sooviti võimalust kohale sobitatud (*fit*) lõiget vasaku ääre asemel keskele joondada. Viimasena oli frustreeriv, et tahkude komplekte polnud võimalik moonutustega lahti voltida ja pidi lisama käsitsi palju murdekohti. Samu mõtteid oli vastatud ka “Mis edasiarendusi soovitaksite pluginale?” küsimusele.

## 3.2 Võrdlus teiste lahendustega

Loodud pistikprogrammi võrreldi varasemate lahendustega – Ultimate Trim UV ja DECALmachine pistikprogrammidega. Siin võeti arvesse peamiselt funktsionaalset võimekust, mitte kasutajamugavust või välimust.

Ultimate Trim UV peamine funktsionaalsus on lahti volditud tahukomplektide joondamine soovitud lõikele. Selle eeltöoks on lõigete asukohtade määramine. Rakendus eeldab, et lõikeleht koosneb ainult kuni kahekümnest vertikaalselt jaotatud lõikest. See tähendab, et tekstuurikleepsudega lõikelehel pole võimalik määrata horisontaalset tekstuurikleepsu asukohta (joonis 28).



Joonis 28. “Ultimate Trim Settings” paneelis (paremal) saab lõigete asukohti määrata vaid kõrguse järgi. Lõikelehel ei saa tahku joondada täpselt ühegi tekstuurikleepsu (lõikelehe alumine osa) peale.

Sellega võrreldes on Trimmer paindlikum, sest ükskõik millist ristküliku kujulist ala saab määrata lõikeks. Lisaks pole lõigete hulk piiratud. Samuti on Ultimate Trim UV'd kasutades võimalik joondada tahke ainult ilma moonutamata. See-eest Trimmeriga saab peale *fit* valiku kasutada ka *fill* valikut, mis venitades terve lõike täidab. Peale lõike rakendamist saab seda ka pöörata või peegeldada.

Ultimate Trim UV'1 on ka plusse võrreldes Trimmeriga. Selles on võimalik eeltööna tehtud lõikelehe jaotust salvestada ning salvestatud jaotuseid sisse laadida. Samuti saab tahul joondada lõiget nii vasakule, keskele kui paremale. Trimmer joondab *fit* valiku puhul tahu alati vasakule, ning selle asukohta lõike sees peaks muutma käsitsi.

DECALmachine on küll loodud peamiselt tekstuuriklepsudega töötamiseks, kuid selle funktsionaalsus lihtsustab oluliselt ka lõikelehtede töövoogu. Peale lõigete pistikprogrammi lisamist on kasutajal palju paindlikkust, kuidas neid mudelile rakendada. Kasutajal on võimalik täpset sobitusmeetodit valida (näiteks, kas lõiget võib venitada), lõiget pöörata ning ka peale rakendamist lõigete vahel valida (joonis 29).



Joonis 29. DECALmachine kasutajaliides laseb klahvikombinatsioone kasutades täpsustada lõike sobitusmeetodit, pööramist ning lõiget ennast.

Lisaks saab DECALmachine'is märkida lõiget horisontaalselt korduvaks (*Panel* märkeruut). Seda võimalust Trimmeris pole, selles käsitletakse kõiki lõikeid samasuguselt. DECALmachine'is aga ei saa lõikeid lisaks pööramisele peegeldada, nagu Trimmeris. Sellele vaatamata on DECALmachine'il rohkem funktsionaalsusi lõikelehtede töövoo jaoks kui Trimmeril.

### 3.3 Edasiarendamise võimalused

Lahendus on saadaval GitHub projektina ning Blenderi pistikprogrammide lehel <https://extensions.blender.org/add-ons/trimmer/>. Tulevikus on autoril plaanis lahendust edasi arendada, alustades järgmiste funktsionaalsuste lisamisega:

- *fit* valiku korral peale peegeldamise ja pööramise ka võimalus lõiget keskele joondada või liuguriga liigutada;
- mitme tahu valimisel lasta kasutajal märkida, kas tahke käsitleda komplektina või rakendada lõige igale tahule eraldi;
- lõikelehe jaotus salvestada ning salvestatud jaotus rakendusse sisse laadida;
- tahkude moonutamisega lahti voltimise võimekus.

Peale funktsionaalsuste lisamise on plaanis täiendada kasutajaliidest. Tulevikus peaks visuaalselt eristama mitteaktiivsed nupud aktiivsetest. Samuti võiks saada kindlate funktsionaalsuste jaoks kasutada kiirklahve, nagu DECALmachine'is. Lisaks peaks koodi ümber struktureerima nii, et vältida ujukomaarvude ebatäpsusest tulenevaid vigu.

## Kokkuvõte

Töö eesmärk oli luua pistikprogramm Trimmer, mis lihtsustab lõikelehtede töövoogu Blenderi tarkvaras. Analüüsi varasemaid lahendusi Ultimate Trim UV ja DECALmachine ning nende põhjal koostati nõuded loodavale pistikprogrammile. Töö teoreetilises osas kirjeldati tekstuurimise protsessi 3D modelleerimises, kus 3D mudel kaetakse kahetasandilise tekstuuriga. Täpsemalt pöörati tähelepanu lõikelehtede töövoole, kus kasutatakse mitut tekstuuri ühe pildi sees.

Pistikprogramm arendati programmeerimiskeeles Python. Programmi arendamine sisaldas Blenderis kasutajaliidese loomist, Blenderi andmetüüpide ja 3D koordinaatide teisendamist ning vajalike algoritmide implementeerimist. Tähtsamad algoritmid olid tahkude lahti voltimine, keskväärtuskoordinaatide süsteem ja tahkude komplekti piirjoone leidmine.

Lahendust testiti Tartu Kunstikooli õpilastega. Alguses tutvustati õpilastele lõikelehtede töövoogu, seejärel tehti töövoogu kasutades Blenderis läbi kaks tekstuurimist. Sama mudelit tekstuuriti alguses Blenderi vaikimisi võimekusega, seejärel kasutades pistikprogrammi Trimmer. Eesmärk oli hinnata pistikprogrammi kasutatavust ning selle kasutusjuhendi selgust. Testimise põhjal tegi Trimmeri olemasolu tekstuurimise protsessi lihtsamaks, samuti anti soovitusi edasiarendusteks.

Lahendust võrreldi teiste Blenderi pistikprogrammidega, milleks olid Ultimate Trim UV ja DECALmachine. Võrreldes Ultimate Trim UV'ga on Trimmeri võimekus suurem, kuid väiksem DECALmachine'i võimekusest. Võrdluse tulemusena tehti kindlaks olulisemad kitsaskohad ning pakuti funktsionaalsusi, mida on autoril plaanis edasi arendada. Pistikprogramm on avaldatud Blenderi laienduste lehel, kus see on kõigile tasuta kättesaadav.

## Viidatud kirjandus

- [1] Dischler JM, Ghazanfarpour D. A survey of 3D texturing. *Computers & Graphics* 2001; 25:135–51. [https://doi.org/10.1016/S0097-8493\(00\)00113-8](https://doi.org/10.1016/S0097-8493(00)00113-8)
- [2] Aaltonen E. Efficient texturing techniques for game environment art. Tampere University of Applied Sciences, 2022. <https://urn.fi/URN:NBN:fi:amk-202205118536>.
- [3] Olsen M. The Ultimate Trim: Texturing Techniques of Sunset Overdrive 2015 (kasutatud 23.04.2025). <https://gdcvault.com/play/1022324/The-Ultimate-Trim-Texturing-Techniques>.
- [4] The correct way to make decals in Blender (NO SHRINKWRAP). YouTube; 2024 (kasutatud 09.05.2025). <https://www.youtube.com/watch?v=rUP8aAG4AQ8>.
- [5] UE4: Detailed Texture Blending with HeightLerp and Vertex Painting. YouTube; 2016 (kasutatud 09.05.2025). <https://www.youtube.com/watch?v=dghCetkArJI>.
- [6] Radsby C. Let's talk about Trim-Sheets Techniques 2023 (kasutatud 27.02.2025). <https://www.patreon.com/posts/lets-talk-about-94999921>.
- [7] Floater MS. Mean value coordinates. *Computer Aided Geometric Design* 2003; 20:19–27. [https://doi.org/10.1016/S0167-8396\(03\)00002-5](https://doi.org/10.1016/S0167-8396(03)00002-5).
- [8] Finstad K. The Usability Metric for User Experience. *Interacting with Computers* 2010; 22:323–7. <https://doi.org/10.1016/j.intcom.2010.04.004>.

# Lisad

## I. Lähtekood ja kasutusjuhend

Pistikprogrammi lähtekoodi ning väljalasked on leitavad GitHub'i repositooriumist:

<https://github.com/LaXHeXLuX/Trimmer/>

Repositooriumis olevas README.md failis on inglisekeelne kasutusjuhend:

<https://github.com/LaXHeXLuX/Trimmer/blob/main/README.md>

Pistikprogramm on avaldatud ka Blenderi laienduste lehel:

<https://extensions.blender.org/add-ons/trimmer/>

Lõputööga on kaasas repositooriumi kokkupaktitud kaust, mis on mõeldud Blenderi versioonile 4.2.

## II. Küsitlus

### Trimmeri tagasiside

Esimeses osas laete alla Blenderi faili ning installite Trimmer pistikprogrammi. Kui siin osas küsimusi või probleeme tekib, andke julgelt märku.

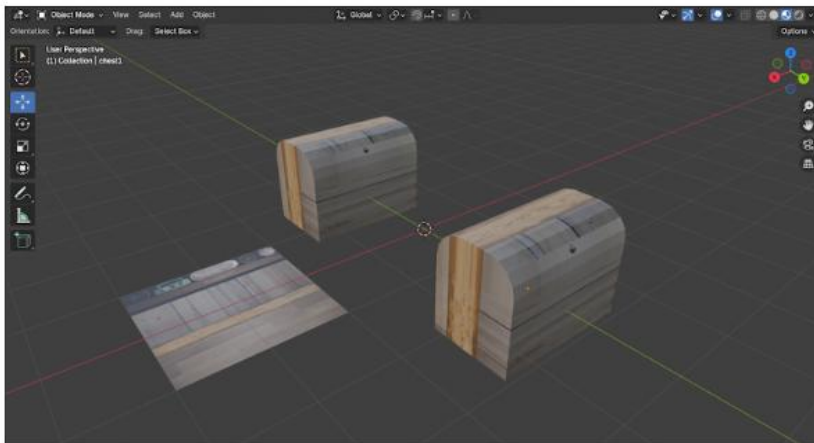
laas.lumberg@gmail.com [Switch accounts](#)



The name, email address and photo associated with your Google Account will be recorded when you upload files and submit this form

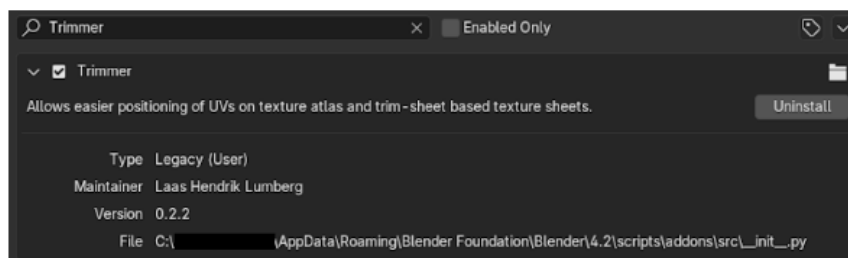
\* Indicates required question

Laadige alla Blenderi fail ning lõikelehe pilt [siit](#). Blenderi faili avades peaks 3D Viewport vaateaknas ees olema selline pilt: \*



Tehtud

Laadige alla Trimmer\_0.2.3.zip [siit](#). Installige Blenderisse Trimmer vastavalt [README juhendile](#) (Installation). Peale plugina installimist peaks Blenderi Add-ons aknas olema allolev pilt. Jätke esialgu plugin mitteaktiveerituks (märkeruut pealkirja kõrval) \*



Tehtud

Next

Clear form

## Lõikelehe töövoog - ilma pluginata tekstuurimine

Järgmiseks ülesandeks on tekstuurida Blenderi failis olevad kaks kirstu, kasutades lõikelehte. Mõlema kirstu mudelil on materjaliks määratud sama lõikeleht. Vaja on mudeli UV koordinaadid sobitada lõikelehe trimmidele.

Tekstuurige mudel *chest1* ilma Trimmer pluginat kasutamata.

Mudel *chest2* jätkke esialgu puutumata.

Täpne lõpptulemus on teie enda otsustada - tekstuurige nii, kuidas endale hea tundub.

Peale tekstuurimise lõpetamist vastake allolevatele küsimustele.

Blenderi (ilma pluginata) võimekus vastab minu ootustele \*

	1	2	3	4	5	6	7	
Ei nõustu üldse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Nõustun täielikult

Soovi korral põhjendage

Your answer

---

Blenderi (ilma pluginata) kasutamine on frustreriv \*

	1	2	3	4	5	6	7	
Ei nõustu üldse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Nõustun täielikult

Soovi korral põhjendage

Your answer

---

Blenderit (ilma pluginata) on lihtne kasutada \*

1 2 3 4 5 6 7

Ei nõustu üldse        Nõustun täielikult

Soovi korral põhjendage

Your answer

---

Pean Blenderit (ilma pluginata) kasutades liiga palju vigu käsitsi parandama \*

1 2 3 4 5 6 7

Ei nõustu üldse        Nõustun täielikult

Soovi korral põhjendage

Your answer

---

Back

Next

Clear form

## Pluginaga tekstuurimine

Mudeli *chest2* tekstuurimiseks aktiveerige plugin *Add-ons* aknast ning kasutage [README failis](#) olevat kasutusjuhendit (*Preliminary work* ja *Manual*).

Üritage *chest2* tekstuurida samasuguseks, nagu *chest1*.

Peale tekstuurimise lõpetamist vastake allolevatele küsimustele.

Pluginaga võimekus vastab minu ootustele \*

	1	2	3	4	5	6	7	
Ei nõustu üldse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Nõustun täielikult

Soovi korral põhjendage

Your answer

---

Pluginaga kasutamine on frustreeriv \*

	1	2	3	4	5	6	7	
Ei nõustu üldse	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Nõustun täielikult

Soovi korral põhjendage

Your answer

---

Pluginat on lihtne kasutada \*

1 2 3 4 5 6 7

Ei nõustu üldse

Nõustun täielikult

Soovi korral põhjendage

Your answer

Pean pluginat kasutades liiga palju vigu käsitsi parandama \*

1 2 3 4 5 6 7

Ei nõustu üldse

Nõustun täielikult

Soovi korral põhjendage

Your answer

Mis edasiarendusi soovitaksite pluginale?

Your answer

Veel mõtteid?

Your answer

Kui soovid, saad ekraanipilte lisada

Upload up to 5 supported files. Max 10 MB per file.

[Add File](#)

Back

Submit

Clear form

# Litsents

## Lihlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Laas Hendrik Lumberg,

1. annan Tartu Ülikoolile tasuta loa (lihlitsentsi) minu loodud teose **Trimmer – Lõikelehe töövoo rakendus Blenderile**, mille juhendaja on Jaanus Jaggo, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;
2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni;
3. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile;
4. kinnitan, et lihlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Laas Hendrik Lumberg

**13.05.2025**