VIDZEMES AUGSTSKOLA
INŽENIERZINĀTŅU FAKULTĀTE

NAGU 3D MODELĒŠANA UN DIZAINU DATUBĀZE

GADA PROJEKTS

Autors: Dens Enrijs Lakučs
Stud. apl. Nr.: IT22063
Darba vadītājs: Dr.sc.ing. Kaspars Osis

VALMIERA 2024

KOPSAVILKUMS

Autors: Dens Enrijs Lakučs, Stud.apl.Nr. IT22063
Darba vadītājs: Dr.sc.ing. Kaspars Osis

Nagu 3D modelēšana un dizainu datubāze – Gada projekts, Valmiera: Vidzemes Augstskola, 2024. – 31 lpp., 5 tabulas, 10 attēli, 4 pielikumi.

Apmeklējot manikīru, var sastapties ar problēmu, ka nav skaidra sapratne par to, kā galā izskatīsies nagi. Citas skaistumkopšanas nozares jau ir atradušas analogus risinājumus šādām problēmām savās jomās un pašlaik attīsta tehnoloģijas digitāliem risinājumiem. Savukārt manikīra jomā, risinājumu klāsts priekš mākslīgu nagu vizualizācijas ir daudz mazāks.

Darba mērķis ir, izmantojot Blender 3D modelēšanas rīku, izstrādāt programmu, kas ģenerē naga modeli ar izvēlētu formu, krāsu, dizainu un tekstūru, kā arī izstrādāt SQLite datubāzi, lai uzturētu šīs naga modeļa īpašības.

Lai sasniegtu darba mērķi, autoram būs nepieciešams apgūt Blender 3D modelēšanas rīku, tā Python skriptinga iespējas caur Blender Python API, SQL un SQLite datubāžu izstrādi, kā arī Python funkcionalitātes ar SQLite.

Darba sasniegtie rezultāti liek pamatus nākotnes naga modelēšanas tehnoloģijām.

SUMMARY

Author: Dens Enrijs Lakučs, Stud. ID Nr. IT22063
Supervisor: Dr.sc.ing. Kaspars Osis

Nail 3D modelling un design database – Year project, Valmiera: Vidzeme University of Applied Sciences, 2024. – 31 pages, 5 tables, 10 images, 4 appendices.

When going to a manicurist, you may encounter the problem of not having a clear understanding of how the nails will look in the end. Other branches of the beauty industry have already found analog solutions for such problems in their own fields and are now developing technologies for digital solutions. However, in the field of manicure, the selection of solutions for nail visualization is significantly lower.

The aim of this work is to develop a program using the Blender 3D modelling tool that generates a nail model with a selected shape, color, design and texture, as well as develop an SQLite database to store such nail model properties.

To achieve the aim of the work, the author will need to learn about and to use the Blender 3D modelling tool, its Python scripting capabilities through the Blender Python API, SQL and SQLite database development, as well as Python functionalities with SQLite.

The results of this work would lay the foundations for future nail modelling technologies.

SAĪSINĀJUMI UN ATSLĒGVĀRDI

Saīsinājumi:

ACID – *Atomicity, Consistency, Isolation, Durability* – nedalamība, saskaņotība, izolācija, izturība

API – *Appilcation Programming Interface* – lietojumprogrammu programmēšanas saskarne

AR – augmentētā realitāte

CAD – *Computer Aided Designing* – datorizētā projektēšana

GUI – *Graphical User Interface* - grafiskā lietotāja saskarne

I/O – *Input/Output* – ievade/izvade

MI – mākslīgais intelekts

OOP – objektorientētā programmēšana

PK – *Primary Key* – primārā atslēga

RDBMS – *Relational Database Management System* – relāciju datubāžu vadības sistēma

RGBA – *Red, Green, Blue, Alpha* – sarkans, zaļš, zils, caurspīdīgums

SQL – *Structured Query Language* - strukturēto vaicājumu valoda

Atslēgvārdi:

Tekstūra – materiāla aprakstošās vizuālās īpašībās, piemēram, spīdīgums, raupjums, gaišums, caurspīdīgums utt.

Dizains – mākslinieciska attēls

Datorgrafikas rīks – tehnoloģija, kas rada attēlus uz datora ekrāna

Takelāža – 3D animācijas jomā, sistēma, kas sadala modeli gabalos un tos saista vienu ar otru, ļaujot veikt modeļa manipulācijas

Augmentētā realitāte – tehnoloģija, kas paplašina reālo pasauli ar virtuāliem elementiem

Spraudnis – programmas paplašinājums, kas dod iespēju veikt jaunas funkcijas vai atvieglo esošu funkciju veikšanu

Skriptings – kādas programmēšanas valodas instrukciju izpilde ar specifisku mērķi

SATURS

IEVADS

Skaistumkopšanas industrija sastāda ievērojamu daļu no pasaules ekonomikas. HelpLama ziņo, ka 2023. gadā, tā vērtība bija virs 500 miljardiem ASV dolāru. Ja pat lielu daļu no šīs industrijas sastāda e-komercija, tai starpā tiešsaistes veikali, digitāla reklamēšana un sociālie mēdiji, 46% no patērētājiem saka, ka vēlas apskatīt skaistumkopšanas produktus klātienē (HelpLama, 2024). Katrai skaistumkopšanas nozarei šī vērtība atšķiras, bet ir skaidrs, ka eksistē liela auditorija, kas vēlas augstas kvalitātes digitālos risinājumus jo pašreizējās metodes nav apmierinošas.

Matu kopšanas un kosmetoloģijas nozares tur pārsvaru mūsdienu skaistumkopšanas industrijā un ir eksistējušas kādā formā vairākus tūkstošus gadu (Matthias, 2021; NHBF, 2023). Šo iemeslu dēļ, abas nozares ir spējušas attīstīt risinājumus kā attēlot gala produktus. Matu kopšanas nozarē, parūkas ir vienmēr viegls risinājums un mūsdienu tehnoloģijas ir revolucionizējušas kosmetoloģiju ar sejas filtriem, video demonstrācijām un foto manipulēšanu. Savukārt, manikīra un pedikīra nozares to mūsdienu formā sākās tikai 20.gs., tāpēc pašlaik ir pieejami tikai daži analogie un digitālie risinājumi gala produkta attēlošanai (Phamily Enterprise, 2021).

Darba mērķis:

Gada projekts tiek izstrādāts IT nozarē nagu kopšanas uzņēmējdarbības jomā. Darba mērķis ir izstrādāt risinājumu saistītu ar naga dizainu attēlošanu. Šis risinājums ir naga 3D modeļa ģenerēšana un datubāze, kas satur manikīram pieejamas krāsas, tekstūras un dizainus.

Mērķa auditorija:

Darba mērķa auditorija ir gan 3D modelēšanas jomas dalībnieki, gan manikīra un pedikīra jomas profesionāļi, kā arī jebkurš, kas saskaras ar darbā apskatīto problēmu savā ikdienā. Programmas paredzēto funkcionalitāšu izpildei, mērķa auditorijai ir jāspēj zemā līmenī apieties ar 3D modelēšanas rīkiem, programmēšanas valodām un datubāzēm.

Darba uzdevumi:

- Līdzīgu risinājumu izpēte;
- Risinājumu kodēšana;
- Datubāzes risinājuma izstrāde;
- Risinājumu testēšana.

Darba metodes:

- Teorētiskās literatūras analīze;
- Datu apstrādes metožu analīze;
- Programmēšana;
- Testēšana;
- Modelēšana.

Darba struktūra:

- Programmas izstrādes teorētiskais pamatojums;
- PPA aprakstīšana un izstrāde;
- Testēšana;
- Ekonomiskais pamatojums;
- Secinājumi.

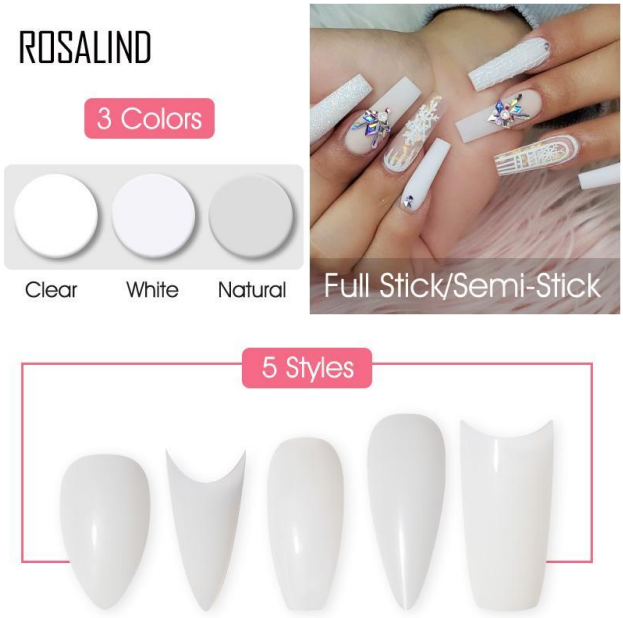
Izstrādes posmi:

- Projekta teorētiskās daļas izstrāde;
- Programmatūras prasību specifikācijas un projektējuma apraksta izstrāde;
- Naga modeļa ģenerēšanas programmas izstrāde;
- Datubāzes izstrāde;
- Testēšana.

1. LĪDZĪGU RISINĀJUMU APSKATS

1.1. ANALOGIE RISINĀJUMI

Manikīra profesijā, tāpat kā citās mākslinieciski saistītās nozarēs, ir grūti attēlot gala produktu kamēr tas vēl nav pabeigts. Tāpēc ir tapuši daži analogie risinājumi, lai manikīra klientam ir labāka sapratne par gala rezultātu. Viens no tiem ir balti nagu formu un izmēru paraugi. Šos paraugus ir iespējams iegūt lielos apjomos, tāpēc manikīri bieži nokrāso tos viņiem pieejamās krāsās.



Attēls 1. Nagu formas paraugi (Rosalind Beauty, 2024).

Krāsu opcijas demonstrē ar nokrāsotiem parauga nagiem, vai manikīrs parāda krāsu pudelīšu izvēli. Uz katras krāsas pudelītes ir arī uzrakstīts krāsas izskats jeb tekstūra tad, kad tā ir izžuvusi. Naga dizains parasti tiek atstāts manikīra ziņā pēc brīvas izvēles, vai arī klients apraksta dizainu vai parāda kāda cita manikīra veidoto dizainu un palūdz to atveidot.

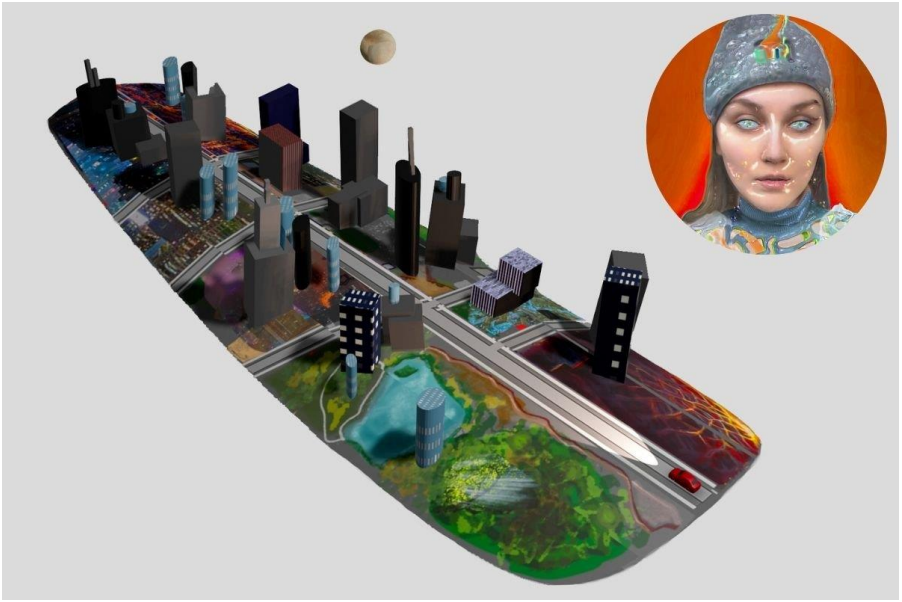
Bieži vien šie analogie risinājumi netiek izmantoti, jo reti kurš klients dodas pie manikīra bez labas sapratnes par vēlamo gala produktu.

1.2. DIGITĀLIE RISINĀJUMI

Pēdējos dažos gados, ar labākām tehnoloģijām, ir mēģinājumi digitalizēt naga attēlošanu. "AR Nails Try-On" ir viens no rīkiem, kas mēģina šo paveikt (Maliavina, 2022). Tas izmanto kameru un MI, lai atšķirtu nagu uz rokas un to 'nokrāso' jebkādā pieejamā krāsā vai dizainā. Šis rīks ir paredzēts konkrēti nagu lakas jomā, tāpēc tas nespēj pagarināt nagus vai attēlot citu nagu formu, kas nav jau esošā klienta nags.

Attēls 2. AR Nails Try-On demonstrācija (Maliavina, 2022).

Māksliniece Piper ZY, kā daļu no "100 Day of Making Augmented Reality Art" izaicinājuma, iztēlojās nagu dizainu nākotni un izveidoja 3D miniaturizētu pilsētu uz naga modeļa, ko var attēlot virsū uz reālas cilvēka rokas (Hitchon, 2022). Šādi var iegūt jebkādu formu un dizainu un pat iztēloties kaut ko jaunu, bet ņemot vērā, ka šis tika radīts mākslas projekta ietvaros manuāli veidojot 3D modeli, šim nav praktiska pielietojuma manikīra jomā.

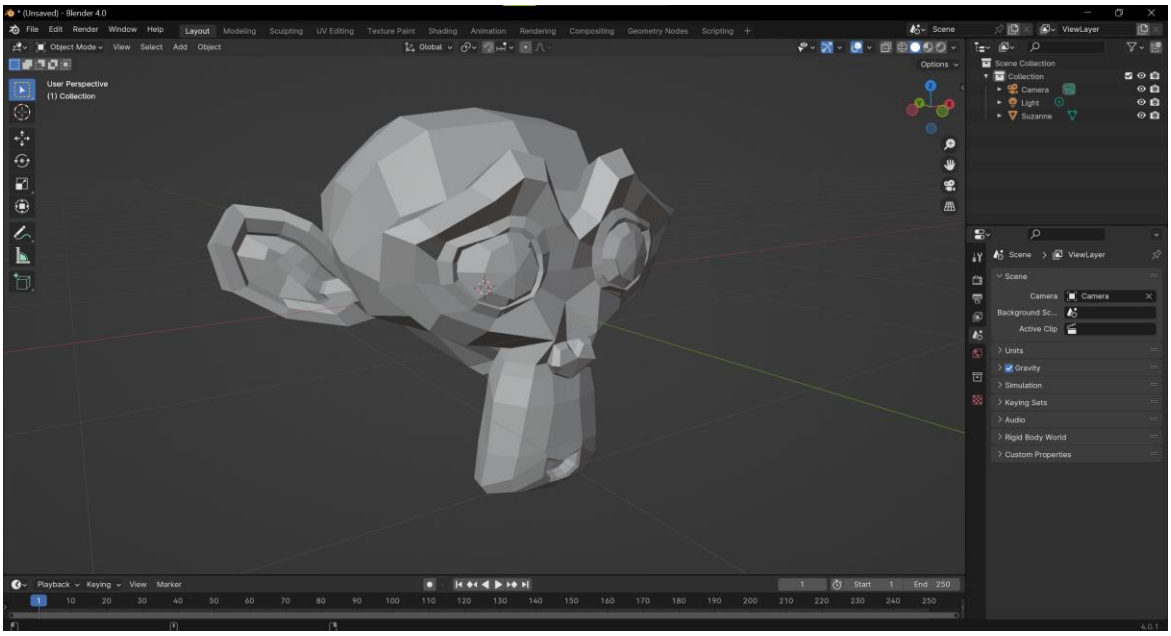


Attēls 3. Mākslinieces Piper ZY viedotais naga modelis (Hitchon, 2022).

2. TEHNOLOĢIJU APSKATS

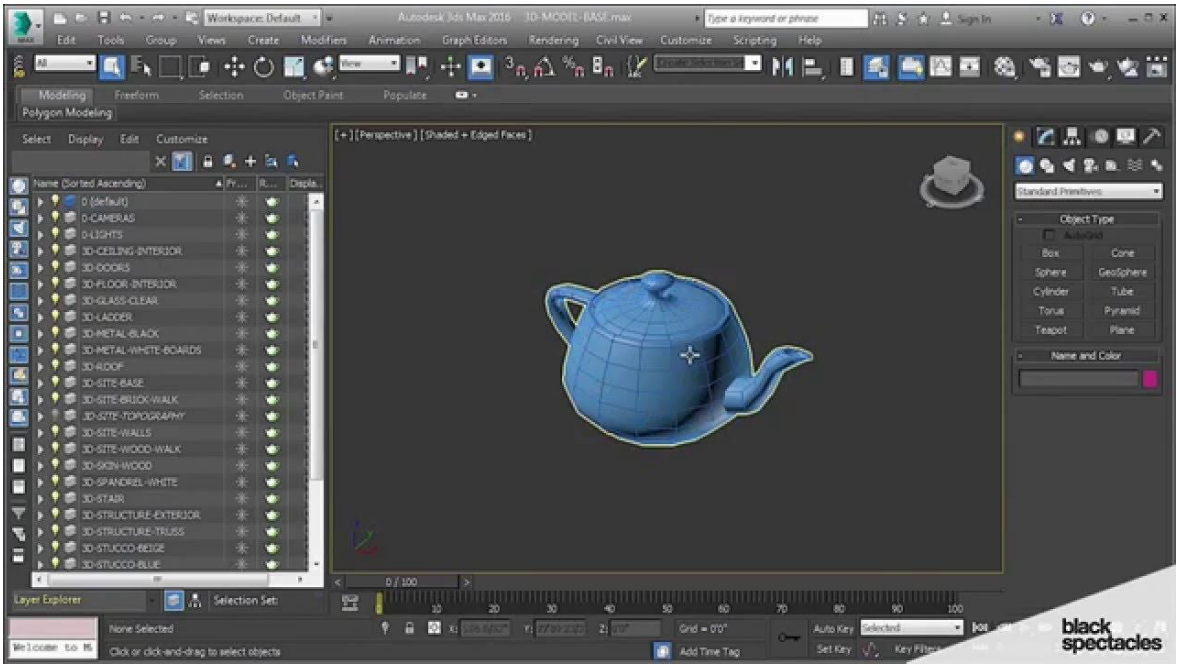
2.1. MODELĒŠANAS RĪKI

Blender ir brīvs, atvērtā koda 3D datorgrafiku rīks paredzēts 3D modelēšanai, skulpturēšanai, animāciju, video, attēlu un datorspēļu veidošanai, modeļu takelāžai, virtuālajai realitātei, dažādu fizikālu modeļu simulēšanai, kā piemēram šķidrumu un gāžu simulēšana, apgaismojuma simulēšana, daļiņu simulēšana un mīksto ķermeņu simulēšana. Blender ir balstīts uz C, C++ un Python programmēšanas valodām un ir pieejams uz galvenajām operētājsistēmām Windows, Linux un macOS. Tam ir Blender Python API, kas spēj automātiski izpildīt vairumu Blender funkcijas caur skriptingu, kā arī izveidot spraudņus caur OpenGL GUI. Kā vairums 3D modelēšanas rīku, Blender darbojas ar virsotnēm, kas ir punkti definēti 3D telpā ar trim koordinātām X, Y un Z. Starp šīm virsotnēm var izveidot līnijas un starp līnijām un virsotnēm var izveidot skaldnes. Virsotnes, līnijas un skaldnes var atlasīt un manipulēt ar matemātiskām funkcijām, manuāli ar roku, vai ar skulptūru veidošanas rīkiem. Skaldnes ir iespējams nokrāsot ar jebkādu krāsu, piedot tām kāda materiāla īpašības, tekstūras vai jebkādu attēlu (Blender Foundation, 2024).



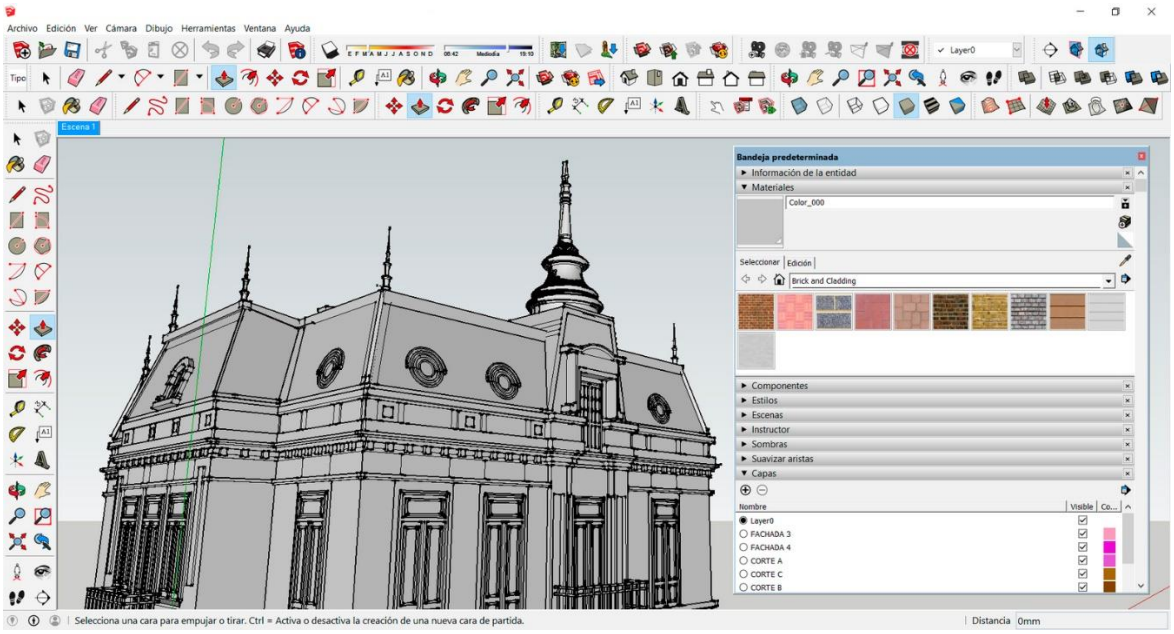
Attēls 4. Blender standarta modelis Suzanne (autora veidots attēls).

Autodesk 3ds Max ir profesionāls 3D datorgrafiku rīks priekš 3D modeļiem, animācijām un datorspēlēm. Autodesk 3ds Max ir balstīts uz C, C++ un Python programmēšanas valodām un ir pieejams uz Windows operētājsistēmas. Tajā ir iebūvēta skriptinga valoda MAXScript, kas spēj automatizēt uzdevumus un izveidot jaunas funkcionalitātes bez arējas programmēšanas valodas nepieciešamības. Autodesk 3ds Max strādā uz poligonu modelēšanas principa, kas ir pieeja objektu modelēšanai mēģinot tieši vai aptuveni attēlot to virsmu izmantojot daudzstūrus. Alternatīvi, tam ir pieejami NURBS, kas ir matemātiski aprakstītas līknes, kā rezultātā tos neietekmē ģeometriski pārveidojumi vai projekcijas, atšķirībā no galīgu punktu metodēm (Autodesk, 2011).



Attēls 5. "Jūtas tējkanna" Autodesk 3ds Max modelēšanas rīkā (Black Spectacles, 2016).

SketchUp ir 3D modelēšanas programma visbiežāk izmantota ēku, ainavu un mēbeļu modelēšanai arhitektūras un interjerdizaina jomās, kā arī 3D printēšanai. SketchUp ir tīmeklī pieejama brīva versija SketchUp Free kā arī maksas versijas SketchUp Go, SketchUp Pro un SketchUp Studio ar vairāk funkcijām paredzētas komerciālai lietošanai. Maksas versijas arī atbalsta programmatūras paplašinājumus, kas ir sarakstīti Ruby programmēšanas valodā. 3D Warehouse ir plaša datubāze ar lietotāju radītiem modeļiem, kas arī ir lietojami citās modelēšanas programmās kā AutoCAD un Revit (Grover, 2009).



Attēls 6. Ēkas modelis SketchUp modelēšanas rīkā (ArchDaily, 2017).

Tabula 1. Modelēšanas rīku salīdzinājums (autora veidota tabula)

Modelēšanas rīks	Piemērots maziem, nekantainiem modeļiiem	Maksas versijas	Skriptings/ API pa brīvu	Brīvs, iepriekš izveidotu modeļu klāsts	Aktuāla rokasgrāmata/ dokumentācija
Blender	+	–	+	+/–	+
Autodesk 3ds Max	+	+	+	–	+/–
SketchUp	–	+	–	+	+/–

Balstoties uz Tabula 1, autors izvēlas Blender kā 3D modelēšanas rīku. SketchUp neatbilst darba prasībām, galvenokārt tam trūkst spēja ērti veidot kompleksas formas modeļus. Autodesk 3ds Max atbilst darba prasībām, bet, būdama maksas programma ar tikai izmēģinājuma versiju pa brīvu, tā izmantošana sarežģītu darba izstrādi. Blender atbilst darba prasībām, būdama ērti lietojama, bet tāpat saturot pietiekami funkcionalitātes darba izstrādei. Darba autoram arī ir iepriekšēja pieredze strādājot ar Blender.

2.2. PROGRAMMĒŠANAS VALODAS

Python ir augsta līmeņa, vispārīga programmēšanas valoda kas, tādēļ, ka kods tiek automātiski kompilēts un palaists, ir piemērota skriptingam, tīmekļa aplikācijām, augstas datorjaudas aprēķiniem, kā arī vienkāršiem uzdevumiem. Tā struktūras principi padara to par skaidru un lasāmu valodu visos līmeņos, tādējādi tā arī ir viena no populārākajām programmēšanas valodām pasaulē. Python ir plašs funkciju klāsts – iebūvēti augsta līmeņa datu tipi, loģisko operāciju struktūras, organizacionālās struktūras, OOP funkcijas, koda kompilācija jebkurā brīdī bez atsevišķa soļa, lietotāju radītas bibliotēkas un paplašinājumi uz C un C++, kā arī uz citām valodām. Python izmanto indentācijas, lai parādītu bloku struktūru. Tā, un citu iemeslu dēļ, Python izceļas ar koda izstrādes ātrumu, palaišanas ātrumu, skaidrību un uzkopjamību (Kuhlman, 2011).

Python kods, kas nosaka vai punkts atrodas sfēras robežās (autora veidots):

```python
import random
import math

radius = 1
x = random.uniform(0, radius)
y = random.uniform(0, radius)
z = random.uniform(0, radius)

def point_distance(x,y,z):
    return math.sqrt(x ** 2 + y ** 2 + z ** 2)

print("Random Point: {:.2f}, {:.2f}, {:.2f}".format(x,y,z))

if point_distance(x,y,z) < radius:
    print("The point is inside the sphere")
else:
    print("The point is outside the sphere")
```

Java ir programmēšanas valoda, kas ir veidota, lai tiktu galā ar aplikāciju izstrādi dažādās vidēs, operētājsistēmās un aparatūrā, kamēr vienlaicīgi patērējot minimālos sistēmas resursus un ļaujot iespēju dinamiskai paplašināšanai. Šo mērķu sasniegšanai, Java ir:

- Vienkārša, objektorientēta un pazīstama – nav nepieciešama daudzpusīga iepriekšējā pieredze ar programmēšanu, no pamatiem radīta uz objektorientētiem principiem, pieejams plašs bibliotēku klāsts ar jau izveidotiem objektiem un ir radīta, lai līdzinātos C++ vieglākai migrācijai;
- Izturīga un droša – vienkārša atmiņas pārvalde bez pointeriem, kas rada problēmas citās valodās un ar iebūvētām drošības funkcijām;

- Ar neitrālu arhitektūru un pārvietojama – spējīga darboties uz jebkādas aparatūras vai operētājsistēmas;
- Augstas veiktspējas – strādā pilnā jaudā bez vajadzības pārbaudīt vidi un ar automātisku atmiņas tīrīšanu;
- Interpretēta, vītņota un dinamiska – elastīgāka programmēšana, iespēja palaist vairākus procesus vienlaicīgi un spēja pievienot jaunus koda moduļus pēc vajadzības (Oracle, 2024).

Java kods, kas nosaka vai 3 līniju garumi veido kubu (autora veidots):

```java
import static java.lang.Math.min;

public class Main {
    public static void main(String[] args) {
        taskThree(3, 3, 3);
        taskThree(1, 2, 3);
        taskThree(0, -1, 3);
    }
    static boolean taskThree(int a, int b, int c) {
        if (a == b && b == c && a > 0) {
            System.out.println("The three lines form a cube\n");
            return true;
        } else if (a > 0 && b > 0 && c > 0) {
            System.out.println("The three lines don't form a cube\n");
            return false;
        }
        System.out.println("The length of a edge must be positive!\n");
        return false;
    }
}
```

C++ ir vispārīga programmēšanas valoda, kas uzsver vieglu, datu tipu bagātu abstrakciju dizainu un pielietojumu. Tā ir augsta līmeņa, starp-platformu un objektorientēta, ar zema līmeņa atmiņas funkcijām. C++ ir radīta priekš sistēmu programmēšanas un iebūvētām, ierobežotu resursu aplikācijām un sistēmām. Tā pamats ir klase, kas ir lietotāja definēts datu tips, kas nodrošina datu slēpšanu, inicializāciju, datu tipu pārvēršanu, lietotāja kontrolētu atmiņas vadību un mehānismus priekš pārlādētiem operatoriem. C++ un tā standarta bibliotēkas ir radītas priekš pārvietojamības, kas nodrošina tā darbību uz jebkādas iekārtas un operētājsistēmas. Tā iemesla dēļ C++ ir atradis plašu pielietojumu darbvirsmas lietojumprogrammās, serveros, operētājsistēmās un GUI (Stroustrup, 2013).

C++ kods, kas izvada apļa laukumu (Chainani, 2022):

```cpp
// C++
#include <math.h>
#include <iostream>
#include <span>
#include <vector>

struct Circle {
    float r;
};
void PrintTotalArea(std::span<Circle> circles) {
    float area = 0;
    for (const Circle& c : circles) {
        area += M_PI * c.r * c.r;
    }
    std::count << "Total area: " << area << "\n";
}
auto main(int argc, char** argv) -> int {
    std::vector<Circle> circles = {{1.0}, {2.0}};
    PrintTotalArea(circles);
    return 0;
}
```

Tabula 2. Programmēšanas valodu salīdzinājums (autora veidota tabula)

Programmēšanas valoda	Interpretēta	Izturīga un pārvietojama	Ērti lietojama	Aktuāla rokasgrāmata/ dokumentācija	3D modelēšanas rīka API
Python	+	+/–	+	+	+
Java	+	+	+	+	–
C++	–	+	–	+/–	+/–

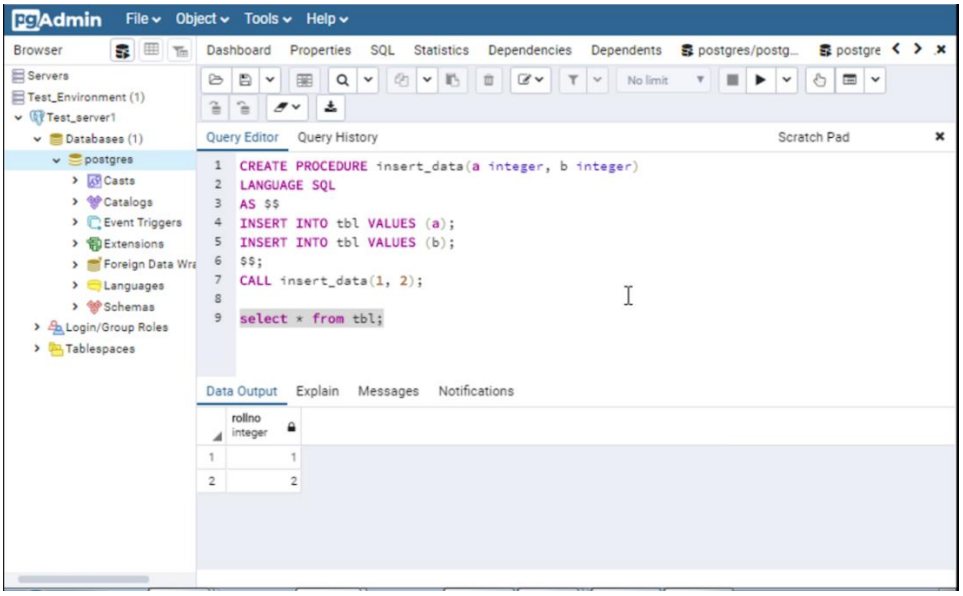
Balstoties uz Tabula 2, autors izvēlas Python kā programmēšanas valodu. Java neatbilst darba prasībām, jo tai nav integrācija ar kādu 3D modelēšanas rīku. C++ ir sarežģīta, bet spēcīga valoda, kā arī daļa 3D modelēšanas rīku ir balstīti uz C++. Šo iemeslu dēļ, tā būtu atbilstoša augstas datorjaudas modelēšanai vai darbībai ar vairākiem modelēšanas rīkiem. Savukārt, darba izstrādei šīs funkcionalitātes nav nepieciešamas. Python atbilst darba prasībām, būdama ērti lietojama, ar pietiekamu jaudu darbību veikšanai un integrāciju ar Blender caur Blender Python API. Darba autoram ir iepriekšēja pieredze strādājot ar Java un Python.

2.3. DATUBĀZES

PostgreSQL spēcīgs, atvērtā koda RDBMS, kas pielieto un paplašina SQL valodu. Tas strādā uz visām galvenajām operētājsistēmām un ir atbilstošs ACID principam. PostgreSQL ir daudzas funkcijas, kas ir paredzētas lietotņu izstrādei, datu integritātes saglabāšanai, datu un datu plūsmas kontrolei un pārvaldībai neatkarīgi no datu apjoma. Tas arī ļauj definēt pielāgotus datu tipus, funkcijas un kodu no citām programmēšanas valodām. PostgreSQL funkcijas iekļauj:

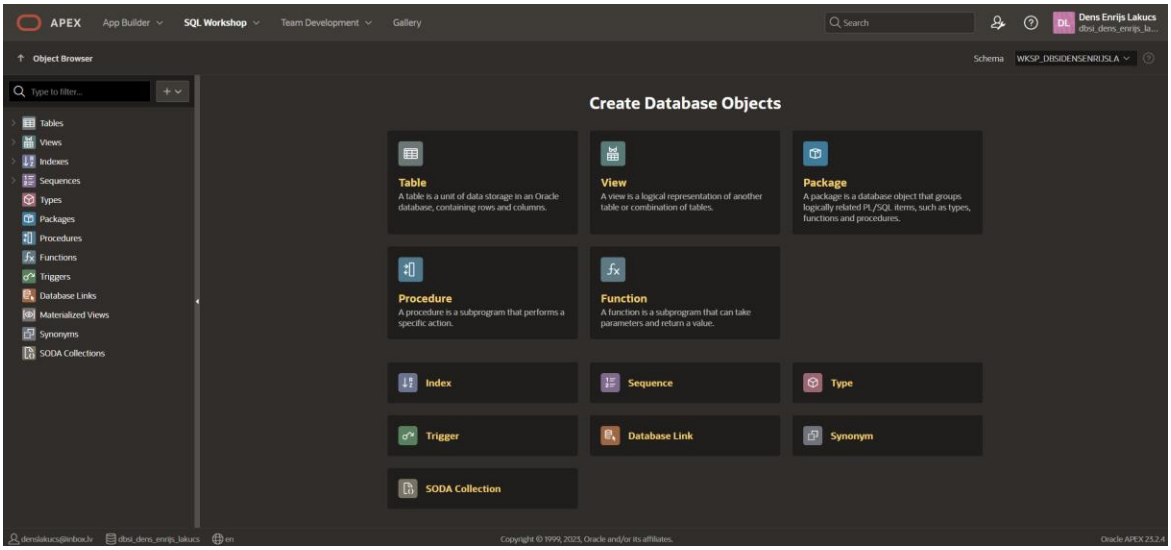
- Datu tipi – primitīvie un strukturētie datu tipi, dokumenti un faili, ģeometriskie, pielāgotie un kompozīti datu tipi;
- Datu integritāte – primārās atslēgas, ārējās atslēgas, datu ierobežojumus;
- Vienlaicīgums un veiktspēja – indeksēšana, vaicājumu plānošana, transakcijas un paralelizācija;
- Izturība un negadījumu seku novēršana;
- Drošība – autentifikācija, piekļuves kontroles sistēma;
- Paplašināmība – glabātas funkcijas un procedūras, programmēšanas valodas, paplašinājumi;
- Internacionalizācija un teksta meklēšana (PostgreSQL, 2024).

pgAdmin ir vispopulārākais vadības rīks priekš PostgreSQL. To var izmantot gan caur pārlūkprogrammu, gan darbvirsmas lietojumprogrammu uz galvenajām operētājsistēmām (pgAdmin, 2024).



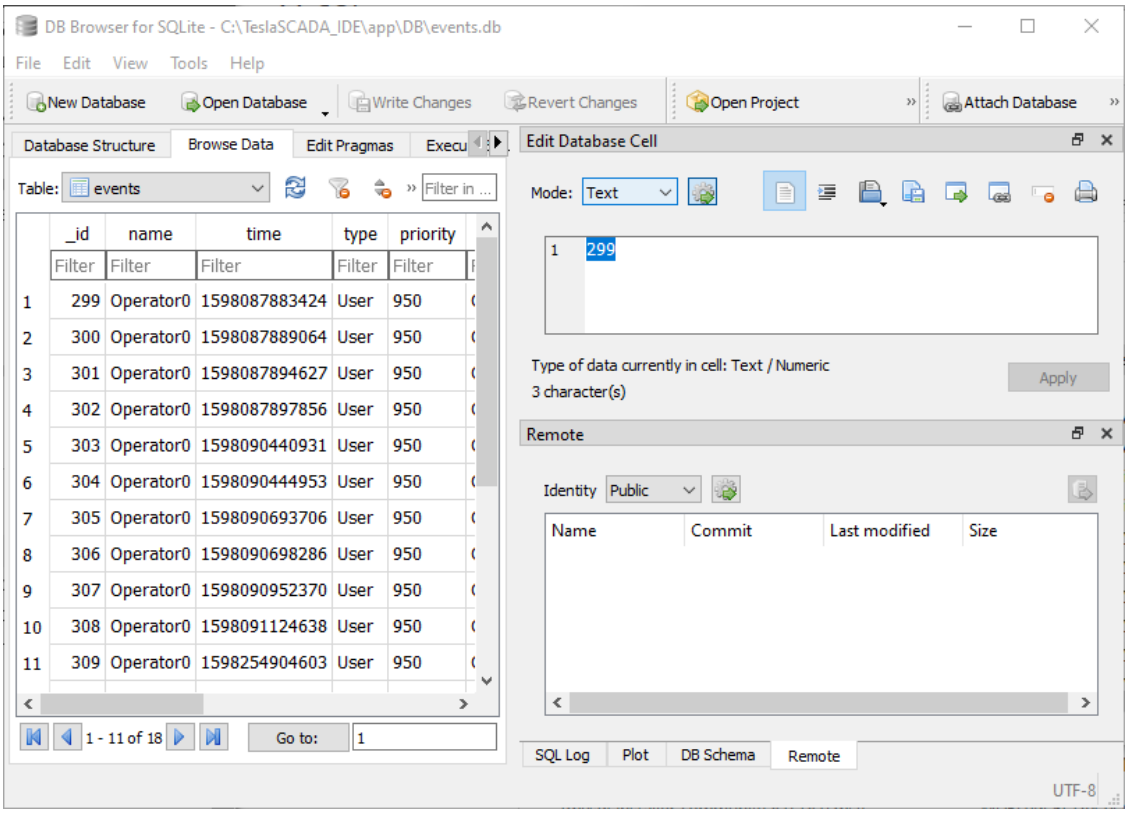
Attēls 7. pgAdmin piemērs (EnterpriseDB, 2023).

Oracle APEX ir uz datubāzēm orientēts tīmekļa lietojumprogrammu izstrādes ietvars. Tas izmanto Oracle Database funkcionalitātes, kā arī iekļauj SQL atbalstu. Tā priekšrocības ir, ka izstrādes vide ir jebkura pārlūkprogramma un Oracle APEX mājaslapa, lietojumprogrammu definīcijas ir saglabātas datubāzē kā metadati, tādējādi nav nepieciešama kompilācija vai kods un visa datu apstrāde veic PL/SQL, tieši strādājot datu shēmās kas atrodas Oracle Database. Oracle APEX plaši izmanto, lai ātri uzbūvētu steidzami nepieciešamas lietotnes, racionalizēt biznesa procesus un atvieglot datu analīzi. Oracle APEX ir paredzēts visiem biznesa darbiniekiem un struktūrvienībām, ne tikai IT departamentam (Oracle, 2015).



Attēls 8. Oracle APEX piemērs (autora veidots attēls).

SQLite ir atvērtā koda bibliotēka, kas ievieš pašietvertu, transakciju datubāzes dzinēju bez serveriem vai konfigurācijas. Tas ir iebūvēts SQL datubāzes dzinējs, kam nav servera procesi un visa datu rakstīšana un lasīšana notiek uz ierīces diska atmiņas failos, kā arī visas datubāzes tabulas, indeksi, trigeri un skati tiek glabāti diska failā. SQLite ir mazs bibliotēkas faila izmērs, pat ar visām funkcijām iespējotām, un kopā ar tā ātrdarbību, kas lielākoties ir atkarīga no sniegtās atmiņas, SQLite var pārspēt tiešos failu sistēmu I/O. Tāpēc tā ir visizplatītākā datubāze pasaulē un ir populāra izvēle kā lietotņu failu formāts. Šīs popularitātes dēļ, SQLite tiek rūpīgi pārbaudīta pirms katras relīzes ar vairākiem automatizētiem testiem, nodrošinot datubāzes darbību un apstrādi atmiņas un diska I/O kļūdu laikā, kā arī transakciju ACID princips noturas sistēmu avārijas vai strāvas padeves trūkuma situācijās (SQLite, 2023).



Attēls 9. SQLite piemērs (TeslaSCADA, 2024).

Tabula 3. Datu bāzu salīdzinājums (autora veidota tabula)

Datubāze	Mākoņa funkcijas	Nepieciešams interneta savienojums	ACID	Programmēšanas valodas	Resursu izmantošana
PostgreSQL	+	–	+	+	+
Oracle APEX	+	+	–	+/–	+/–
SQLite	–	–	+	+	+

Balstoties uz Tabula 3, autors izvēlas SQLite kā datubāzi. Oracle Database caur Oracle APEX vidi būtu atbilstoša datubāze, ja ir nepieciešama plaša pieeja datiem caur mākoni. Savukārt, šāda funkcionalitāte nav nepieciešama darba izstrādāšanai, kā arī tās implementācija caur arēju skriptingu ir sarežģīta. PostgreSQL caur pgAdmin atbilst darba prasībām un ir salīdzināma ar SQLite darba izstrādes mērķim. SQLite tiek izvēlēts ērtākas lietojamības dēļ, salīdzinot ar PostgreSQL. Darba autoram ir iepriekšēja pieredze strādājot ar Oracle APEX, PostgreSQL un SQLite.

3. PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS – NAGA MODELĒŠANA

3.1. IEVADS

Šī projekta ietvaros tiek izstrādāta programma, kas no ievada ģenerē naga modeli un pēc izvēles attēlo uz tā krāsu, dizainu un tekstūru. Programma ir paredzēta ierīcēm, uz kurām ir uzstādīta programmēšanas valoda un 3D modelēšanas rīks. Šajā dokumentā autors apraksta programmas moduļus, datu dekompozīciju un starp moduļu atkarību.

3.2. NOLŪKS

Šī dokumenta nolūks ir naga modelēšanas programmas programmatūras projektējuma apraksts. Tas ir paredzēts programmas izstrādātājam un projektā iesaistītajiem cilvēkiem ar mērķi aprakstīt programmas specifikāciju un prasības.

3.3. DARBĪBAS SFĒRA

Nagu modelēšanas programma darbojas uz vairākām ierīcēm, kurām ir uzstādīta Python programmēšanas valoda un Blender 3D modelēšanas rīks. Nagu modelēšanas programma ir paredzēta kā testa platforma uz kuras tiktu bāzētas nākotnes tehnoloģijas manikīru jomā. Programmas mērķis ir tās lietotājiem nodrošināt iespēju izveidot naga 3D modeli un attēlot jebkuru nagu krāsu, dizainu un tekstūru uz šī modeļa, kas ir pieejams datubāzē pēc lietotāja vēlmes.

Darba mērķa auditorija ir gan 3D modelēšanas jomas dalībnieki, gan manikīra un pedikīra jomas profesionāļi. Programmas paredzēto funkcionalitāšu izpildei, mērķa auditorijai ir jāspēj zemā līmenī apieties ar Blender 3D modelēšanas rīku, Python programmēšanas valodu un SQLite datubāzi.

3.4. DEFINĪCIJAS UN SAĪSINĀJUMI

- **PPA** – Programmas projektējuma apraksts;
- **LVS** – Latvijas Valsts standarts;
- **Modulis** – sistēmas komponente.

3.5. SAISTĪBAS AR CITIEM DOKUMENTIEM

PPA tika veidota vadoties pēc LVS 72:1996 standarta noteiktajām prasībām.

3.6. PROGRAMMATŪRAS DZĪVES CIKLS

Programmatūras dzīves cikls sākās 15.03.2024

3.7. MODUĻU DEKOMPOZĪCIJA

Projekts tiks sadalīts šādos moduļos:

Naga modelēšanas moduļi:

- Naga formas izvēles modulis;
- Naga krāsas izvēles modulis;
- Naga dizaina izvēles modulis;
- Naga tekstūras izvēles modulis;
- Naga modeļa ģenerēšanas un attēlošanas modulis.

Datubāzes moduļi:

- Jaunas formas ievades modulis;
- Jaunas krāsas ievades modulis;
- Jauna dizaina ievades modulis;
- Jaunas tekstūras ievades modulis;
- Formas dzēšanas modulis;
- Krāsas dzēšanas modulis;
- Dizaina dzēšanas modulis;
- Tekstūras dzēšanas modulis.

3.7.1. NAGA FORMAS IZVĒLES MODULIS

Identificējums: SelectNailShape

Nolūks: Izvēlēties naga modeļa formu. Naga forma tiek uzglabāta kā formas virsotņu saraksts, tāpēc katrai formai piešķir unikālu identifikatoru. Veicot izvēli, lietotājs ievada šo identifikatoru.

Funkcijas

- Naga formas identifikatora ievade un apstrāde.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu “Input nail shape”, kur viņš ieraksta naga formas identifikatoru. Lietotāja saskarne tiek realizēta caur Blender grafiskās izstrādes vides atbalstu.

3.7.2. NAGA KRĀSAS IZVĒLES MODULIS

Identificējums: SelectNailColor

Nolūks: Izvēlēties naga modeļa krāsu. Naga krāsa tiek uzglabāta kā RGBA vērtība un dažām krāsām arī būs tās krāsas nosaukums angliski. Veicot izvēli, lietotājs ievada krāsas identifikatoru vai nosaukumu.

Funkcijas

- Naga krāsas ievade un apstrāde.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Input nail color", kur viņš ieraksta naga krāsas nosaukumu vai krāsas identifikatoru.

3.7.3. NAGA DIZAINA IZVĒLES MODULIS

Identificējums: SelectNailDesign

Nolūks: Izvēlēties naga modeļa dizainu. Naga dizains tiek uzglabāts kā PNG attēls, tāpēc katram dizainam piešķir unikālu identifikatoru. Veicot izvēli, lietotājs ievada šo identifikatoru.

Funkcijas

- Naga dizaina identifikatora ievade un apstrāde.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Input nail design", kur viņš ieraksta naga dizaina identifikatoru.

3.7.4. NAGA TEKSTŪRAS IZVĒLES MODULIS

Identificējums: SelectNailTexture

Nolūks: Izvēlēties naga modeļa tekstūru. Naga tekstūra tiek uzglabāta kā saraksts ar vērtībām, kas atsaucas uz kādu Blender tekstūras vērtību, kā piemēram metāliskums un raupjums, tāpēc katrai tekstūrai piešķir unikālu identifikatoru. Veicot izvēli, lietotājs ievada šo identifikatoru.

Funkcijas

- Naga tekstūras identifikatora ievade un apstrāde.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Input nail texture", kur viņš ieraksta naga tekstūras identifikatoru.

3.7.5. NAGA MODEĻA ĢENERĒŠANAS UN ATTĒLOŠANAS MODULIS

Identificējums: GenerateNail

Nolūks: No izvēlētās naga formas ģenerēt modeli un izvēlētās naga krāsas, dizainus un tekstūras attēlot uz modeļa.

Funkcijas

- Naga modeļa ģenerēšana un attēlošana.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Generate model?", kur viņš ieraksta "yes" vai "no". Ja lietotājs izvēlējās "no", tad nekas nenotiek. Ja izvēlējās "yes", tad tiek ģenerēts un attēlots naga modelis ar izvēlētajām īpašībām.

3.7.6. JAUNAS FORMAS IEVADES MODULIS

Identificējums: AddNailShape

Nolūks: Pievienot jaunu naga formu datubāzē. Naga forma tiek uzglabāta kā formas virsotņu saraksts, tāpēc katrai formai piešķir unikālu identifikatoru.

Funkcijas

- Jaunas naga formas pievienošana datubāzē un apstrāde.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Input new nail shape", kur viņš ieraksta naga formu kā virsotņu sarakstu. Pēc veiksmīgas ievades, tiek attēlots jauns datubāzes skats ar pievienoto formu.

3.7.7. JAUNAS KRĀSAS IEVADES MODULIS

Identificējums: AddNailColor

Nolūks: Pievienot jaunu naga krāsu datubāzē. Naga krāsa tiek uzglabāta kā RGBA vērtība un papildus kā attiecīgs krāsas nosaukums, ja tas tiek padots.

Funkcijas

- Jaunas naga krāsas pievienošana datubāzē un apstrāde.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Input new nail color", kur viņš ieraksta naga krāsu kā RGBA vērtību un, ja vēlas, krāsas nosaukumu. Pēc veiksmīgas ievades, tiek attēlots jauns datubāzes skats ar pievienoto krāsu.

3.7.8. JAUNA DIZAINA IEVADES MODULIS

Identificējums: AddNailDesign

Nolūks: Pievienot jaunu naga dizainu datubāzē. Naga dizains tiek uzglabāts kā PNG attēls, tāpēc katram dizainam piešķir unikālu identifikatoru.

Funkcijas

- Jauna naga dizaina pievienošana datubāzē un apstrāde.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Input new nail design", kur viņš ieraksta naga dizainu kā attēla faila ceļu lietotāja ierīcē. Pēc veiksmīgas ievades, tiek attēlots jauns datubāzes skats ar pievienoto dizainu.

3.7.9. JAUNAS TEKSTŪRAS IEVADES MODULIS

Identificējums: AddNailTexture

Nolūks: Pievienot jaunu naga tekstūru datubāzē. Naga tekstūra tiek uzglabāta kā saraksts ar vērtībām, kas atsaucas uz kādu Blender tekstūras vērtību, kā piemēram metāliskums un gludums, tāpēc katrai tekstūrai piešķir unikālu identifikatoru.

Funkcijas

- Jaunas naga tekstūras pievienošana datubāzē un apstrāde.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Input new nail texture", kur viņš ieraksta naga tekstūru kā sarakstu ar vērtībām. Pēc veiksmīgas ievades, tiek attēlots jauns datubāzes skats ar pievienoto tekstūru.

3.7.10. FORMAS DZĒŠANAS MODULIS

Identificējums: DeleteNailShape

Nolūks: Izvēlēties naga modeļa formu un to izdzēst no datubāzes.

Funkcijas

- Naga formas identifikatora ievade un attiecīgās formas dzēšana.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Input nail shape", kur viņš ieraksta naga formas identifikatoru. Pēc veiksmīgas ievades, tiek attēlots jauns datubāzes skats bez izdzēstās formas.

3.7.11. KRĀSAS DZĒŠANAS MODULIS

Identificējums: DeleteNailColor

Nolūks: Izvēlēties naga modeļa krāsu un to izdzēst no datubāzes.

Funkcijas

- Naga krāsas ievade un dzēšana.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Input nail color", kur viņš ieraksta naga krāsas RGBA vērtību vai nosaukumu. Pēc veiksmīgas ievades, tiek attēlots jauns datubāzes skats bez izdzēstās krāsas.

3.7.12. DIZAINA DZĒŠANAS MODULIS

Identificējums: DeleteNailDesign

Nolūks: Izvēlēties naga modeļa dizainu un to izdzēst no datubāzes.

Funkcijas

- Naga dizaina identifikatora ievade un attiecīgā dizaina dzēšana.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Input nail design", kur viņš ieraksta naga dizaina identifikatoru. Pēc veiksmīgas ievades, tiek attēlots jauns datubāzes skats bez izdzēstā dizaina.

3.7.13. TEKSTŪRAS DZĒŠANAS MODULIS

Identificējums: DeleteNailTexture

Nolūks: Izvēlēties naga modeļa tekstūru un to izdzēst no datubāzes.

Funkcijas

- Naga tekstūras identifikatora ievade un attiecīgās tekstūras dzēšana.

Lietotāja saskarne: Lietotājam ir redzams ievades lauks ar tekstu "Input nail texture", kur viņš ieraksta naga tekstūras identifikatoru. Pēc veiksmīgas ievades, tiek attēlots jauns datubāzes skats bez izdzēstās tekstūras.

3.8. DATU DEKOMPOZĪCIJA

Liela daļa no sistēmas funkcionalitātes ir datu plūsma no datubāzes uz programmu. Šajā nodaļā tiek aprakstītas datu plūsmas un to saturs.

3.8.1. DATUBĀZES STRUKTŪRA

Visas nagu formas, krāsas, dizaini un tekstūras tiek glabātas SQLite datubāzē. SQLite datubāze tiek izveidota un ar to strādā izmantojot DB Browser for SQLite. Datubāze satur četras tabulas – shapes, colors, designs, textures.

Tabula "shapes"

Satur informāciju par pieejamajām nagu formām. Tabula sastāv no 3 kolonnām – shape_id, shape_vertex un shape_face. shape_id ir tabulas PK, shape_vertex satur sarakstu ar nagu formu virsotņu X, Y, Z koordinātām un shape_face satur sarakstu ar skaldnēm, kas sastāv no 4 punktiem.

Tabula "colors"

Satur informāciju par pieejamajām nagu krāsām. Tabula sastāv no 3 kolonnām – color_id, color un color_name. color_id ir tabulas PK, color satur sarakstu ar krāsas RGBA vērtībām un color_name satur krāsas nosaukumu.

Tabula "designs"

Satur informāciju par pieejamajām nagu dizainiem. Tabula sastāv no 3 kolonnām – design_id, design un design_name. design_id ir tabulas PK, design satur faila ceļu lietotāja ierīcē uz dizaina PNG attēlu un design_name satur dizaina nosaukumu. Datubāzēs uzglabāt attēlus tiešā veidā nav ērti un labāka prakse ir uzglabāt failu ceļus.

Tabula "textures"

Satur informāciju par pieejamajām nagu tekstūrām. Tabula sastāv no 3 kolonnām – texture_id, texture un texture_name. texture_id ir tabulas PK, texture satur sarakstu ar tekstūras metāliskumu un raupjumu, kas ir Blender materiālu īpašības un nosaka tā izskatu, un texture_name satur tekstūras nosaukumu.

shapes	
PK	shape_id INTEGER NOT NULL UNIQUE
	shape_vertex TEXT NOT NULL UNIQUE
	shape_face TEXT NOT NULL

colors	
PK	color_id INTEGER NOT NULL UNIQUE
	color TEXT NOT NULL UNIQUE
	color_name TEXT

designs	
PK	design_id INTEGER NOT NULL UNIQUE
	design TEXT NOT NULL UNIQUE
	design_name TEXT

textures	
PK	texture_id INTEGER NOT NULL UNIQUE
	texture TEXT NOT NULL UNIQUE
	texture_name TEXT

Attēls 10. Naga īpašību datubāzes struktūra (autora veidots attēls).

4. DARBA PRAKTISKĀ IZSTRĀDE

4.1. PROGRAMMAS IZSTRĀDE

Balstoties uz I Pielikums pieejamo PPS tiek izveidots darba izstrādes PPA. Vadoties no PPA, tiek izstrādāta naga modelēšanas programma. Darba autors apraksta izstrādāto programmu, sadalītu pa PPA noteiktajiem moduļiem un izskata programmas atbilstību PPA un PPS prasībām. Programmas pirmkoda fragmenti ir pieejami II Pielikums.

- Naga īpašību izvēles moduļi – selectNailShape, selectNailColor, selectNailDesign, selectNailTexture

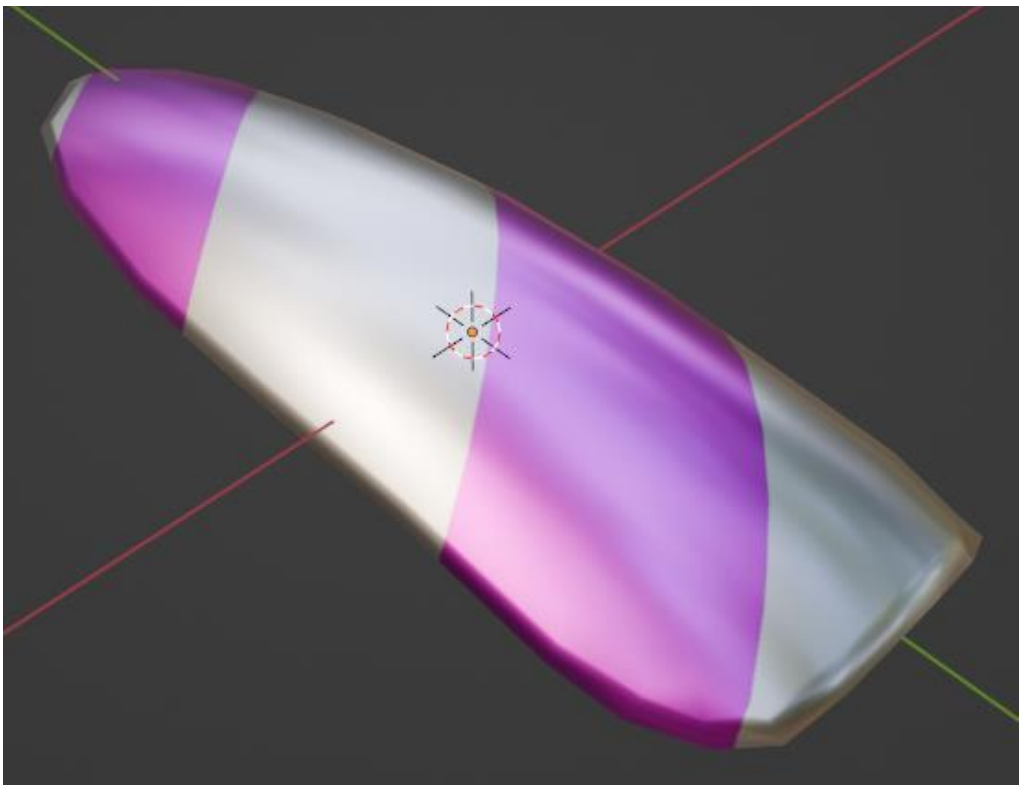
Naga īpašību izvēles moduļi ir funkcionāli ļoti līdzīgi, tāpēc tie tiek aprakstīti kopā. Izsaucot īpašības izvēles funkciju, tai ir jāpievada vērtība, kas identificē naga īpašību datubāzē. Šai vērtībai ir jābūt int tipa mainīgajam vai String tipa mainīgajam, kas satur tikai skaitli vai satur krāsas nosaukumu. Funkcija pārvērš ievadīto vērtību uz String tipa mainīgo, lai būtu iespējams ievadīt gan int, gan String tipa mainīgos un lai izvairītos no problēmām ar datu tipiem izsaucot SQL vaicājumu. Vaicājumam tiek padota ievadītā vērtība, kas nosaka īpašības tabulas PK un tiek veikti SQL vaicājumi, lai iegūtu naga īpašību vērtības.

Apskatot PPA un PPS, šiem moduļiem, kā arī visiem turpmāk apskatītajiem moduļiem, nav ieviesta teksta ievade, kad lietotājs izvēlas naga īpašību. Darba izstrādes laikā tika konstatēts, ka Blender Python API nav iespējams izveidot ievades lauku, kā tas ir iespējams Python. Šim nolūkam ir paredzēts izveidot Blender spraudni un OpenGL GUI priekš lietotāja ievades.

PPS tiek noteikts, ka krāsas tiek apstrādātas HEX formātā, bet PPA un programmā krāsas tiek uzdotas RGBA formātā. Blender Python funkcijas pieņem RGBA krāsas un, ņemot vērā, ka pāriet uz RGBA no HEX ir vienkārši, darba izstrādē tiek izmantotas RGBA krāsu vērtības. Ir jāmin, ka Blender RGBA krāsas ir vērtībās no 0 līdz 1, kamēr parasti RGBA ir no 0 līdz 255.

- Naga modeļa ģenerēšanas un attēlošanas modulis – generateNail

Izsaucot generateNail funkciju, tai ir jāievada naga forma, krāsa, tekstūra un dizains. Šo visērtāk var panākt, izsaucot iepriekš minētās selectNail funkcijas, to rezultātus piešķirot mainīgajam un šos mainīgos ievadot generateNail funkcijā. Funkcija izveido naga modeli no ievadītās naga formas, izveido un piešķir modelim jaunu Blender materiālu ar ievadīto krāsu un tekstūru un dublicētam modelim piešķir dizaina attēlu. Darbā izmantotie modeļi ņemti no Freepik.



Attēls 11. Ģenerēta naga modeļa piemērs (autora veidots attēls).

- Naga īpašību pievienošanas datubāzē moduļi – addNailShape, addNailColor, addNailDesign, addNailTexture

Naga īpašību pievienošanas datubāzē moduļi ir funkcionāli ļoti līdzīgi, tāpēc tie tiek aprakstīti kopā. Izsaucot īpašības pievienošanas funkciju, tai ir jāievada attiecīgā īpašība pareizā formātā. PPA un PPS tiek noteikts, ka ievade tiek pārbaudīta, vai tā atbilst vajadzīgajam formātam. addNailColor un addNailTexture šo funkcionalitāti var ieviest, jo to ievadītās vērtības ir ar noteiktu garumu un formātu. addNailDesign un addNailShape šo ir daudz grūtāk panākt, jo abu funkciju ievades var būt patvaļīga garuma un addNaiDesign var būt pat ar atšķirīgiem faila ceļu formātiem. Šo iemeslu dēļ, šīm funkcijām netiek ievesta ievades pārbaudes funkcionalitāte.

- Naga īpašību dzēšanas no datubāzes moduļi – deleteNailShape, deleteNailColor, deleteNailDesign, deleteNailTexture

Naga īpašību dzēšanas no datubāzēs moduļi ir funkcionāli ļoti līdzīgi, tāpēc tie tiek aprakstīti kopā. Izsaucot īpašības dzēšanas funkciju, tai ir jāievada attiecīgās īpašības tabulas PK kolonnas ID vai, izņemot deleteNailShape funkciju, attiecīgās īpašības nosaukumu. PPS tiek noteikts, ka dzēšanas funkcijām ir jāpārbauda ievade un tā jāaptur, ja nav derīga vai īpašība eksistē tabulā. Darba izstrādes laikā, tiek konstatēts, ka šāda funkcionalitāte nav nepieciešama, jo tās trūkums neizraisa kļūdas. PPA arī tiek noteikts, ka pēc dzēšanas funkcijas, programmai jāattēlo jauns datubāzes izskats bez dzēstās vērtības. Šāda funkcija nav ievesta, bet to pašu var panākt atjaunināšanas pogu datubāzē.

4.2. TESTĒŠANA

Programmas testēšana tika nemitīgi veikta darba izstrādes laikā. Programmas moduļu funkcijas tika testētas ar vairākiem ievadiem, dokumentējot rezultātus.

Tabula 4. Testēšanas testi un rezultāti (autora veidota tabula)

Tests	Rezultāts
Mēģina pievienot dizainu kā hipersaiti uz ārēju avotu	Programma nespēj ielādēt attēlu
Izvēlas naga īpašības ID kas ir ārpus datu robežām	Kļūda: List index out of range
Mēģina ievadīt nepareizi noformētu krāsu pie addNailColor	Blender System Console ir redzams teksts "Invalid color"
Mēģina ievadīt nepareizi noformētu tekstūru pie addNailTexture	Blender System Console ir redzams teksts "Invalid texture"
Mēģina dzēst neeksistējošu īpašību	Nekas nenotiek
Mēģina pievienot datubāzei pareizi noformētu īpašību	Pēc datubāzes atjaunināšanas, ir redzams jauns ieraksts tabulā
Mēģina dzēst no datubāzei pareizi noformētu īpašības ID vai nosaukumu	Pēc datubāzes atjaunināšanas, ir redzams, ka īpašība ir dzēsta

Veiktā testēšana ir sekmīgi līdzējuši programmas darbības pārbaudē un tās izstrādē, salīdzinot iegūtos rezultātus ar gaidāmajiem rezultātiem no koda loģikas un autora intuīcijas.

5. SOCIOEKONOMISKAIS APRAKSTS

Autors ir apkopojis programmas izstrādes izdevumus, nepieciešamos resursus, novērtējis ekonomiskos ieguvumus un ekonomiskās izmaksas lietotājam.

Programmas izstrādei autors izmantoja tikai savu tehniku. Visas saistītās programmatūras darba izstrādei bija bezmaksas. Izstrādes laikā, autors izmantoja interneta resursus problēmu risināšanai.

Tabula 5. Programmas izstrādes izdevumi (autora veidota tabula)

Resursi	**Autora izmaksas**	**Tehnikas izmaksas**	**Pielietojums**
Portatīvais dators	Bezmaksas	1500.00 EUR	Galvenais izstrādes rīks visos izstrādes posmos
Blender	Bezmaksas	Bezmaksas	Modelēšana un programmas izstrāde
SQLite	Bezmaksas	Bezmaksas	Datu glabāšana un apmaiņa
Internets	Bezmaksas	22.49 EUR/mēnesī	Programmas izstrāde
Cilvēkresursi	Bezmaksas	~20.00 EUR/stundā	Programmas izstrāde
Kopā:	**-**	**2112.45 EUR**	

Programma tika izstrādāta autora personīgiem mērķiem un ir pieejama jebkuram bez maksas, tāpēc autors no tā nesaņem nekādu materiālu ieguvumu. Attīstot programmu tālāk kā reālu produktu, būtu jāapsver šāda finanšu modeļa praktiskums.

Programmas lietotājam, kas vēlētos pielietot izstrādāto programmu saviem mērķiem, nebūtu jāveic lielas izmaksas, lai šo panāktu. Lielākās izmaksas varētu sastādīt dators, ja tāds jau nav, bet programmas darbībai nav nepieciešama izteikti liela datorjauda, tāpēc var lietot lētāku, vājāku datoru.

SECINĀJUMI UN PRIEKŠLIKUMI

Manikīra joma, būdama viena no jaunākajām skaistumkopšanas nozarēm, nav spējusi attīstīt augstas kvalitātes digitālos risinājumus jomas izaicinājumiem, konkrēti, naga izskata attēlošanai. Tā vēl joprojām paļaujas un paraugiem un citām analogām metodēm, sapratni starp klientu un manikīru un manikīra prasmēm, lai klients būtu pēc iespējas apmierinātāks ar gala produktu. Ja pirms vairākiem gadiem bija tehnoloģiskas grūtības izstrādāt risinājumus, tad mūsdienās ar modernām tehnoloģijām un pieejām, būtu jābūt plašam iespēju klāstam.

Darbā uzstādītais mērķis ir sasniegts. Tiek izstrādāts naga dizainu attēlošanas risinājums, kas sastāv no naga 3D modeļu ģenerēšanas un naga īpašību datubāzes.

Izstrādātais darbs atbilst PPS un PPA prasībām. Dažas moduļu funkcionalitātes nav tikušas ieviestas vai arī ir pārveidotas, bet autors uzskata, ka PPS un PPA galvenā būtība ir sasniegta.

Izstrādātā programma liek pamatus nākotnes tehnoloģijām, kas attēlotu naga modeli. Tālākai attīstībai veicamie soļi ir:

- Iegūt vairākus un augstākas kvalitātes naga modeļus, vairāk krāsas, dizainus un tekstūras, it īpaši tādas, kas reāli ir pieejamas manikīram;
- Izstrādāt GUI priekš ērtākas programmas lietošanas;
- Pāriet uz datubāzi ar mākoņa funkcionalitātēm priekš plašāk pieejamas piekļuves naga īpašībām, kā arī papildināt datubāzi ar iepriekš izveidotām īpašību kombinācijām, lai manikīrs varētu demonstrēt pieejamos dizainus;
- Papildināt Blender ainu, ieviešot reālus fiziskus apstākļus kā apgaismojumu, ēnas, atstarojumu utt. reālākai naga modeļa attēlošanai;
- Izstrādāt AR lietotni, kas attēlotu naga modeļus uz klienta rokas reālajā laikā.

Autoram ir ieceres attīstīt šo darba ideju, apskatīt un ieviest iepriekš minētās darba attīstības pieejas, izstrādājot bakalaura darbu. Zināšanu pamati jau ir likti, atliek tikai izstrādāt risinājumus.

IZMANTOTIE INFORMĀCIJAS AVOTI

1. ArchDaily. (2017. gada 24 Maijs). *How To Improve Your SketchUp Skills.* Ielādēts 2024. gada 15. Marts no https://www.archdaily.com/871893/how-to-improve-your-sketchup-skills
2. Autodesk. (2011). *Autodesk 3ds Max – Detailed Features.* Ielādēts no https://web.archive.org/web/20110219110238/http://usa.autodesk.com/adsk/servlet/pc/index?siteID=123112&id=13567426
3. Black Spectacles. (2016. gada 17. Janvāris). *Teapot Example - 3D Rendering with Vray 3.2 for 3ds Max.* Ielādēts 2024. gada 15. Marts no https://www.youtube.com/watch?v=atAQOOMxAOc
4. Blender Foundation. (2024). *Blender 4.0 Reference Manual.* Ielādēts 2024. gada 5. Februāris no https://docs.blender.org/manual/en/latest/index.html
5. Chainani, V. (2022. gada 24. Jūlijs). *Is Carbon really the next C++.* Ielādēts 2024. gada 15. Marts no https://dev.to/envoy_/is-carbon-really-the-next-c-2mcn
6. EnterpriseDB. (2023. gada 19. Janvāris). *pgAdmin, a comparable tool to PL/SQL Developer for PostgreSQL.* Ielādēts 2024. gada 21. Aprīlis no https://www.enterprisedb.com/postgres-tutorials/pgadmin-comparable-tool-plsql-developer-postgresql
7. Freepik. *Acrylic-nail 3D models.* Ielādēts 2024. gada 20. Aprīlis no https://www.freepik.com/3d-models/acrylic-nail
8. Grover, C. (2009. gada 22. Maijs). *Google SketchUp: The Missing Manual.* ISBN: 0596555768, 9780596555764
9. HelpLama. (2024). *Beauty Industry Revenue and Usage Statistics 2023.* Ielādēts 2024. gada 4. Februāris no https://helplama.com/beauty-industry-revenue-usage-statistics/
10. Hitchon, R. (2022. gada 24. Janvāris). *3D augmented reality nails: The future of nail styling?* Ielādēts 2024. gada 4. Februāris no https://www.scratchmagazine.co.uk/feature/3d-augmented-reality-nails-the-future-of-nail-styling/
11. Kuhlman, D. (2011. gada 1. Septembris). *A Python Book: Beginning Python, Advanced Python, and Python Exercises.* ISBN: 0984221239, 9780984221233
12. Maliavina, A. (2022. gada 11. Augusts). *What Are AR Nails? How & Why Sally Hansen Creates Them.* Ielādēts 2024. gada 4. Februāris no https://www.perfectcorp.com/business/blog/commerce/brands-can-now-enable-online-shoppers-to-virtually-try-on-nail-color
13. Matthias, M. (2021. gada 18. Marts). *Why Did We Start Wearing Makeup?* Ielādēts 2024. gada 20. Maijs no https://www.britannica.com/story/why-did-we-start-wearing-makeup
14. NHBF. (2023. gada 14. Septembris) *The Evolution Of Hairstyles: A Journey Through Time.* Ielādēts 2024. gada 20. Maijs https://www.nhbf.co.uk/news-and-blogs/blog/the-evolution-of-hairstyles-a-journey-through-time/
15. Oracle. (2015). *Oracle Application Express 5 – Overview.* Ielādēts 2024. gada 16. Marts no http://www.oracle.com/technetwork/developer-tools/apex/overview/apex-overview-157752.ppt
16. Oracle. (2024). *The Java Language Environment.* Ielādēts 2024. gada 5. Februāris no https://www.oracle.com/java/technologies/introduction-to-java.html
17. pgAdmin. (2024). *FAQ.* Ielādēts 2024. gada 16. Marts no https://www.pgadmin.org/faq/
18. Phamily Enterprise (2021. gada 28. Decembris) *The History Of Acrylic Nails.* Ielādēts 2024. gada 20. Maijs https://www.phamilyenterprise.com/the-history-of-acrylic-nails/
19. PostgreSQL. (2024). *About PostgreSQL.* Ielādēts 2024. gada 3. Marts no https://www.postgresql.org/about/
20. Rosalind Beauty. (2024). Ielādēts 2024. gada 15. Marts no https://www.rosalindbeauty.com/products/false-nails?variant=42733816250586
21. SQLite. (2023. gada 10. Oktobris). *About SQLite.* Ielādēts 2024. gada 1. Marts no https://www.sqlite.org/about.html
22. Stroustrup, B. (2013. gada 9. Maijs). *The C++ Programming Language 4th Edition.* ISBN: 978-0-321-56384-2
23. TeslaSCADA. (2024). *SQLite Database.* Ielādēts 2024. gada 16. Marts no https://teslascada.com/HTML/sqllite.html

PIELIKUMI

I. PIELIKUMS. PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA PPS

I.1. Ievads

I.1.1. Nolūks

Šis dokuments ir naga modelēšanas programmas programmatūras prasību specifikācija (PPS). Dokumentā aprakstītas PPS prasības, kas tiek izvirzītas naga modelēšanas programmai. Dokuments ir paredzēts programmas izstrādātājam un lietotājam kā pamatinformācija programmas izmantošanā un darbībā.

I.1.2. Darbības sfēra

Nagu modelēšanas programma darbojas uz vairākām ierīcēm, kurām ir uzstādīta pārlūkprogramma, interneta savienojums, daudzpusīga programmēšanas valoda, mākonī bāzēta datubāze un 3D modelēšanas programma. Nagu modelēšanas programma ir paredzēta kā testa platforma uz kuras tiktu bāzētas nākotnes tehnoloģijas manikīru jomā. Programmas mērķis ir tās lietotājiem nodrošināt iespēju izveidot naga 3D modeli un attēlot jebkādu nagu krāsu, dizainu un tekstūru uz šī modeļa pēc lietotāja vēlmes.

I.1.3. Saistība ar citiem dokumentiem

PPS tika veidota vadoties pēc LVS 68:1996 – "Programmatūras prasību specifikācijas ceļvedis." standarta noteiktajām prasībām.

I.1.4. Pārskats

Ievads sniedz vispārēju ieskatu PPS dokumentācijā un iepazīstina ar tā saturu. Tiek aprakstītas naga modelēšanas programmas funkcionālās, veiktspējas, standartu un aparatūras īpašības, prasības un prasību definēšana.

I.2. Vispārējs apraksts

I.3. Produkta perspektīva

I.4. Produkta funkcijas

- Izvēlēties naga formu un ģenerēt attiecīgu 3D modeli;
- Izvēlēties krāsu un attēlot to uz modeļa;

- Izvēlēties dizainu un attēlot to uz modeļa;
- Izvēlēties tekstūru un attēlot to uz modeļa;
- Attēlot modeli;
- Pievienot jaunu krāsu datubāzei;
- Pievienot jaunu dizainu datubāzei;
- Pievienot jaunu tekstūru datubāzei;
- Dzēst esošu krāsu datubāzē;
- Dzēst esošu dizainu datubāzē;
- Dzēst esošu tekstūru datubāzē.

I.5. Lietotāja raksturiezīmes

Programma tiek izstrādāta ar funkcionalitāti kā prioritāti. Programmas lietotājiem ir nepieciešamas pamatzināšanas programmas koda palaišanā, datubāžu manipulācijā un 3D modelēšanas rīku izmantošanā pilnai programmas funkcionalitātei. Galvenokārt, programmu lietos programmētāji un grafikas dizaineri.

I.6. Vispārējie ierobežojumi

Lai panāktu, ka visas programmas funkcijas darbotos pilnvērtīgi, lietotājam ir jāizmanto ierīce, kas ir aprīkota ar operētājsistēmu, funkcionējošu ekrānu, ievades ierīcēm un ieinstalētu pārlūkprogrammu, programmēšanas valodu un 3D modelēšanas rīku.

I.7. Pieņēmumi un atkarības

- Lietotājam ir strādājoša ierīce;
- Ierīce ir savienota ar ievades ierīcēm;
- Uz ierīces ir uzstādīta operētājsistēma;
- Uz ierīces ir uzstādīta programmēšanas valoda;
- Uz ierīces ir uzstādīts 3D modelēšanas rīks;
- Programmas funkcijas un veiktspēja ir atkarīga no ierīces veiktspējas;
- Uz ierīces ir uzstādīta pārlūkprogramma;
- Uz ierīces ir nodrošināts interneta savienojums.

I.8. Konkrētās prasības

I.9. Funkcionālās prasības

I.9.1. Naga formas izvēle

Mērķis

Izvēle ir nepieciešama, lai noteiktu naga gala modeļa formu.

Ievade

Lietotājs ievada vēlamās naga formas unikālu identificējošu vērtību.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizi noformētu vērtību.

Izvade

Ievade tiek bloķēta, ja netiek ievadīta pareizi noformēta vērtība. Ja ievadītā vērtība izturēja pārbaudi, tad tiek izvadīts izvēlētās naga formas identifikators.

I.9.2. Naga krāsas izvēle

Mērķis

Izvēle ir nepieciešama, lai noteiktu naga gala modeļa krāsu.

Ievade

Lietotājs ievada vēlamās naga krāsas HEX vērtību vai dažām krāsām piemītošu pseidonīmu.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizu HEX vērtību vai krāsas nosaukumu.

Izvade

Ievade tiek bloķēta, ja netiek ievadīta pareizi noformēta vērtība. Ja ievadītā vērtība izturēja pārbaudi, tad tiek izvadīts izvēlētās naga krāsas HEX vērtība un krāsas nosaukums ja tāds ir pieejams.

I.9.3. Naga dizaina izvēle

Mērķis

Izvēle ir nepieciešama, lai noteiktu naga gala modeļa dizainu.

Ievade

Lietotājs ievada vēlamās naga dizaina unikālu identificējošu vērtību.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizi noformētu vērtību.

Izvade

Ievade tiek bloķēta, ja netiek ievadīta pareizi noformēta vērtība. Ja ievadītā vērtība izturēja pārbaudi, tad tiek izvadīts izvēlētās naga dizaina identifikators.

I.9.4. Naga tekstūras izvēle

Mērķis

Izvēle ir nepieciešama, lai noteiktu naga gala modeļa tekstūru.

Ievade

Lietotājs ievada vēlamās naga tekstūras unikālu identificējošu vērtību.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizi noformētu vērtību.

Izvade

Ievade tiek bloķēta, ja netiek ievadīta pareizi noformēta vērtība. Ja ievadītā vērtība izturēja pārbaudi, tad tiek izvadīts izvēlētās naga tekstūras identifikators.

I.9.5. Naga modeļa ģenerēšana un attēlošana

Mērķis

Ģenerēt un attēlot gala naga modeli.

Ievade

Naga modeļa ģenerācijas un attēlošanas inicializācija

Apstrāde

Iegūst izvēlēto naga formu un no tā ģenerē modeli. Iegūst izvēlēto naga krāsu, dizainu un tekstūru un tās attēlo uz modeļa.

Izvade

Tiek attēlots naga modelis ar krāsu, dizainu un tekstūru.

I.9.6. Jaunas naga formas pievienošana datubāzē

Mērķis

Pieejamo naga formu sarakstam pievienot jaunu formu.

Ievade

Jaunas formas pievienošanas funkcijas inicializācija un jaunās formas ievade.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizi noformētu naga formu un vai šāda forma jau eksistē.

Izvade

Ievade tiek bloķēta, ja netiek ievadīta pareizi noformēta forma vai forma jau eksistē. Ja ievadītā forma izturēja pārbaudi, tad tiek izvadīta jaunā pievienotā forma un tiek attēlots jauns skats ar datubāzē pievienoto formu.

I.9.7. Jaunas naga krāsas pievienošana datubāzē

Mērķis

Pieejamo naga krāsu sarakstam pievienot jaunu krāsu.

Ievade

Jaunas krāsas pievienošanas funkcijas inicializācija un jaunās krāsas HEX vērtības ievade.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizi noformētu HEX vērtību un vai šāda vērtība jau eksistē.

Izvade

Ievade tiek bloķēta, ja netiek ievadīta pareizi noformēta vērtība vai krāsa jau eksistē. Ja ievadītā vērtība izturēja pārbaudi, tad tiek izvadīta jaunās pievienotās krāsas HEX vērtība un tiek attēlots jauns skats ar datubāzē pievienoto krāsu.

I.9.8. Jaunas dizaina pievienošana datubāzē

Mērķis

Pieejamo naga dizainu sarakstam pievienot jaunu dizainu.

Ievade

Jauna dizaina pievienošanas funkcijas inicializācija un jaunā dizaina ievade.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizi noformētu dizainu un vai šāds dizains jau eksistē.

Izvade

Ievade tiek bloķēta, ja netiek ievadīts pareizi noformēts dizains vai dizains jau eksistē. Ja ievadītais dizains izturēja pārbaudi, tad tiek izvadīts jaunais dizains un tiek attēlots jauns skats ar datubāzē pievienoto dizainu.

I.9.9. Jaunas tekstūras pievienošana datubāzē

Mērķis

Pieejamo naga tekstūru sarakstam pievienot jaunu tekstūru.

Ievade

Jaunas tekstūras pievienošanas funkcijas inicializācija un jaunās tekstūras ievade.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizi noformētu tekstūru un vai šāda tekstūra jau eksistē.

Izvade

Ievade tiek bloķēta, ja netiek ievadīta pareizi noformēta tekstūra vai tekstūra jau eksistē. Ja ievadītā tekstūra izturēja pārbaudi, tad tiek izvadīta jaunās pievienotā tekstūra un tiek attēlots jauns skats ar datubāzē pievienoto tekstūru.

I.9.10. Esošas formas dzēšana no datubāzēs

Mērķis

Pieejamo naga formu sarakstā dzēst esošu formu.

Ievade

Esošas formas dzēšanas inicializācija un dzēšamās formas vērtības ievade.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizi noformētu naga formu un mēģina dzēst esošu formu.

Izvade

Ievade tiek bloķēta, ja netiek ievadīta pareizi noformēta forma vai forma neeksistē. Ja ievadītā forma izturēja pārbaudi, tad forma tiek dzēsta un tiek attēlots jauns skats bez datubāzē dzēstās formas.

I.9.11. Esošas krāsas dzēšana no datubāzēs

Mērķis

Pieejamo naga krāsu sarakstā dzēst esošu krāsu.

Ievade

Esošas krāsas dzēšanas inicializācija un dzēšamās krāsas HEX vērtības ievade.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizi noformētu HEX vērtību un mēģina dzēst esošu krāsu.

Izvade

Ievade tiek bloķēta, ja netiek ievadīta pareizi noformēta krāsa vai krāsa neeksistē. Ja ievadītā krāsa izturēja pārbaudi, tad krāsa tiek dzēsta un tiek attēlots jauns skats bez datubāzē dzēstās krāsas.

I.9.12. Esoša dizaina dzēšana no datubāzēs

Mērķis

Pieejamo naga dizainu sarakstā dzēst esošu dizainu.

Ievade

Esoša dizaina dzēšanas inicializācija un dzēšamā dizaina vērtības ievade.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizi noformētu dizaina vērtību un mēģina dzēst esošu dizainu.

Izvade

Ievade tiek bloķēta, ja netiek ievadīts pareizi noformēts dizains vai dizains neeksistē. Ja ievadītais dizains izturēja pārbaudi, tad dizains tiek dzēsts un tiek attēlots jauns skats bez datubāzē dzēstā dizaina.

I.9.13. Esošas tekstūras dzēšana no datubāzēs

Mērķis

Pieejamo naga tekstūru sarakstā dzēst esošu tekstūru.

Ievade

Esošas tekstūras dzēšanas inicializācija un dzēšamās tekstūras vērtības ievade.

Apstrāde

Pārbauda vai lietotājs mēģina ievadīt pareizi noformētu tekstūras vērtību un mēģina dzēst esošu tekstūru.

Izvade

Ievade tiek bloķēta, ja netiek ievadīta pareizi noformēta tekstūra vai tekstūra neeksistē. Ja ievadītā tekstūra izturēja pārbaudi, tad tekstūra tiek dzēsta un tiek attēlots jauns skats bez datubāzē dzēstās tekstūras.

I.10. Veiktspējas prasības

Lai nodrošinātu programmas rezultātu vizuālu kvalitāti, ierīces ekrānam vajadzētu būt vismaz 6 collu lielam ar vidēju līdz augstu izšķirtspēju. Programmas funkcionalitāte ir visvairāk atkarīga no ierīces veiktspējas. Zemas veiktspējas ierīcēm būs nepieciešams ilgs laiks priekš rezultātiem vai arī pati programma nebūs lietojama.

Programmas funkcionalitāte ir atkarīga no interneta savienojuma. Ierīcei vismaz vienreiz ir jābūt savienotai ar internetu, lai iegūtu datus no datubāzes un tos varētu saglabāt atmiņā.

I.11. Aparatūras ierobežojumi

Programma ir paredzēta jebkurai ierīcei ar uzstādītu operētājsistēmu, pārlūkprogrammu, interneta savienojumu, programmēšanas valodu un 3D modelēšanas rīku.

I.12. Ārējās saskarnes prasības

I.12.1. Lietotāja saskarnes prasības

- Ierīces ievades un izvades teksta izmēram ir jābūt pietiekami lielam, lai lietotājs var ērti to salasīt;
- Ja lietotājs ir nepareizi ievadījis izvēli, tad programmai ir jābrīdina lietotājs par to, jādod padomi par kļūdas novēršanu un programmas palaišanai ir jābūt atspējotai;
- Ja ierīcei pārtūkst interneta savienojums, tad programmai ir jābrīdina lietotājs par to un programmas palaišanai ir jābūt atspējotai.

I.12.2. Ierīču saskarnes prasības

- Strādājošas ievada ierīces;
- Ierīces ekrānam jābūt vismaz 6 collu lielam ar vidēju līdz augstu izšķirtspēju;
- Ierīces veiktspējai jābūt adekvātai, lai palaistu programmu;
- Ierīcei ir jābūt nodrošinātam interneta savienojumam.

II. PIELIKUMS. PIRMKODA PARAUGA FRAGMENTI

```
#Libraries
import bpy      # Blender Python
import math     # Math Functions
import sqlite3  # Python SQLite
import ast      # Abstract Syntax
import os       # Operating System
import re       # Regular Expressions

# Clears the scene of all objects, meshes, materials and images
def clearFile(collection_name):
    collection = bpy.data.collections[collection_name]
    meshes = set()
    for obj in collection.objects:
        bpy.data.objects.remove(obj)
    for mesh in bpy.data.meshes:
        bpy.data.meshes.remove(mesh)
    for mat in bpy.data.materials:
        bpy.data.materials.remove(mat)
    for image in bpy.data.images:
        bpy.data.images.remove(image)

# Selects and returns the nail shape data from the database.
# Argument format: int or String of an int.
def selectNailShape(shape_data):
    shape_data = str(shape_data)
    shape_verts = ast.literal_eval(cur.execute("SELECT shape_vertex FROM shapes
WHERE shape_id = ?", [shape_data]).fetchall()[0][0])
    shape_faces = ast.literal_eval(cur.execute("SELECT shape_face FROM shapes
WHERE shape_id = ?", [shape_data]).fetchall()[0][0])
    return [shape_verts, shape_faces]

# Selects and returns the nail color from the database.
# Argument format: int, String or String of an int.
def selectNailColor(color_data):
    color_data = str(color_data)
    if color_data.isnumeric():
        color = ast.literal_eval(cur.execute("SELECT color FROM colors WHERE
color_id = ?",[color_data]).fetchall()[0][0])
    else:
        color = ast.literal_eval(cur.execute("SELECT color FROM colors WHERE
LOWER(color_name) LIKE LOWER(?)",[color_data]).fetchall()[0][0])
    return color

# Selects and returns the nail design from the database.
# Argument format: int, String or String of an int.
def selectNailDesign(design_data):
    design_data = str(design_data)
    if design_data.isnumeric():
        design = cur.execute("SELECT design FROM designs WHERE design_id =
?",[design_data]).fetchall()[0][0]
    else:
        design = cur.execute("SELECT design FROM designs WHERE LOWER(design_name)
LIKE LOWER(?)",[design_data]).fetchall()[0][0]
    return design
```

```
# Selects and returns the nail texture from the database.
# Argument format: int, String or String of an int.
def selectNailTexture(texture_data):
    texture_data = str(texture_data)
    if texture_data.isnumeric():
        texture = ast.literal_eval(cur.execute("SELECT texture FROM textures
WHERE texture_id = ?",[texture_data]).fetchall()[0][0])
    else:
        texture = ast.literal_eval(cur.execute("SELECT texture FROM textures
WHERE LOWER(texture_name) LIKE LOWER(?)",[texture_data]).fetchall()[0][0])
    return texture

# Generates a nail model from input data.
# Arguments must be in a specific format, use previous selectNail functions.
def generateNail(shape, color, texture, design):
    scene = bpy.context.scene
    mesh = bpy.data.meshes.new("Mesh")
    mesh.from_pydata(shape[0], [], shape[1])
    object = bpy.data.objects.new("Nail", mesh)
    scene.collection.objects.link(object)

    C = bpy.context

    bpy.ops.object.select_all(action='SELECT')

    objs = C.selected_objects
    obj = objs[0]
    obj.rotation_euler.x += math.radians(90)
    collection = bpy.data.collections['Nails']

    if collection and objs:
        for ob in objs:
            for coll in ob.users_collection:
                coll.objects.unlink(ob)
            collection.objects.link(ob)

    mat = bpy.data.materials.get("Material")
    mat = bpy.data.materials.new(name="Material")

    if obj.data.materials:
        obj.data.materials[0] = mat
    else:
        obj.data.materials.append(mat)

    mat = bpy.data.materials["Material"]
    mat.use_nodes = True
    mat.node_tree.nodes["Principled BSDF"].inputs[0].default_value = color
    mat.node_tree.nodes["Principled BSDF"].inputs[1].default_value = texture[0]
    mat.node_tree.nodes["Principled BSDF"].inputs[2].default_value = texture[1]

    for poly in obj.data.polygons:
        poly.use_smooth = True

    bpy.ops.object.duplicate(linked=False)
    design_ob = bpy.data.objects.get("Nail.001")
    design_ob.name = "Nail_Design"
```

```
    design_mat = bpy.data.materials.get("Design")
    if design_mat is None:
        design_mat = bpy.data.materials.new(name="Design")
    if design_ob.data.materials:
        design_ob.data.materials[0] = design_mat
    else:
        design_ob.data.materials.append(mat)

    design_mat.use_nodes = True
    design_mat.node_tree.nodes["Principled BSDF"].inputs[1].default_value =
texture[0]
    design_mat.node_tree.nodes["Principled BSDF"].inputs[2].default_value =
texture[1]

    design_mat = bpy.data.materials.get("Design")
    if design_mat is None:
        design_mat = bpy.data.materials.new(name="Design")

    if design_ob.data.materials:
        design_ob.data.materials[0] = design_mat
    else:
        design_ob.data.materials.append(mat)

    design_mat.blend_method = 'BLEND'

    if design_mat.use_nodes:
        trans_node = design_mat.node_tree.nodes.get("Transparent BSDF", None)
        if trans_node is None:
            design_mat.node_tree.nodes.new("ShaderNodeBsdfTransparent")
        mix_node = design_mat.node_tree.nodes.get("Mix Shader", None)
        if mix_node is None:
            design_mat.node_tree.nodes.new("ShaderNodeMixShader")
        image_node = design_mat.node_tree.nodes.get("Image Texture", None)
        if image_node is None:
            design_mat.node_tree.nodes.new("ShaderNodeTexImage")

    node_tree = design_mat.node_tree

    trans_node = design_mat.node_tree.nodes.get("Transparent BSDF", None)
    mix_node = design_mat.node_tree.nodes.get("Mix Shader", None)
    principled_node = design_mat.node_tree.nodes.get("Principled BSDF", None)
    material_node = design_mat.node_tree.nodes.get("Material Output", None)
    image_node = design_mat.node_tree.nodes.get("Image Texture", None)

    image_node.image = bpy.data.images.load(os.getcwd() + "\\" + design)

    node_tree.links.new(trans_node.outputs["BSDF"], mix_node.inputs[1])
    node_tree.links.remove(principled_node.outputs[0].links[0])
    node_tree.links.new(principled_node.outputs["BSDF"], mix_node.inputs[2])
    node_tree.links.new(mix_node.outputs["Shader"], material_node.inputs[0])
    node_tree.links.new(image_node.outputs["Color"], principled_node.inputs[0])
    node_tree.links.new(image_node.outputs["Alpha"], mix_node.inputs[0])

    bpy.ops.object.select_all(action='DESELECT')
    bpy.context.view_layer.objects.active = design_ob
    bpy.ops.object.mode_set(mode='EDIT')
    bpy.ops.mesh.select_all(action='SELECT')
    bpy.ops.uv.smart_project(angle_limit=1.5707)
    bpy.ops.object.mode_set(mode='OBJECT')
    obj.select_set(False)
```

```
# Adds a nail shape into the database.
# Argument formats: String of [(lists of vertices)], String of [(lists of faces)]
def addNailShape(shape_vertex, shape_face):
    cur.execute("INSERT INTO shapes (shape_vertex, shape_face)
VALUES(?,?)",(shape_vertex,shape_face))
    con.commit()

# Adds a nail color into the database.
# Argument formats: String of (RGBA values), String
def addNailColor(color_rgba, color_name):
    pattern =
r'^\(\s*(0(\.\d+)?|1(\.0+)?)\s*,\s*(0(\.\d+)?|1(\.0+)?)\s*,\s*(0(\.\d+)?|1(\.0+)?
)\s*,\s*(0(\.\d+)?|1(\.0+)?)\s*\)$'
    if bool(re.fullmatch(pattern, color_rgba)):
        cur.execute("INSERT INTO colors (color, color_name)
VALUES(?,?)",(color_rgba, color_name))
        con.commit()
    else:
        print("Invalid color")

# Adds a nail design into the database.
# Argument formats: String of filepath, String
def addNailDesign(design_link, design_name):
    cur.execute("INSERT INTO designs (design, design_name)
VALUES(?,?)",(design_link, design_name))
    con.commit()

# Adds a nail texture into the database.
# Argument formats: list of 2 float(0,1) values, String
def addNailTexture(texture_Metal_Rough, texture_name):
    pattern = r'^\(\s*(0(\.\d+)?|1(\.0+)?)\s*,\s*(0(\.\d+)?|1(\.0+)?)\s*\)$'
    if bool(re.fullmatch(pattern, texture_Metal_Rough)):
        cur.execute("INSERT INTO textures (texture_name, texture)
VALUES(?,?)",(texture_Metal_Rough, texture_name))
        con.commit()
    else:
        print("Invalid texture")

# Deletes a nail shape from the database.
# Argument format: int or String of an int.
def deleteNailShape(shape_id):
    cur.execute("DELETE FROM shapes WHERE shape_id = (?)",[shape_id])
    con.commit()

# Deletes a nail color from the database.
# Argument format: int, String or String of an int.
def deleteNailColor(color_data):
    color_data = str(color_data)
    if color_data.isnumeric():
        cur.execute("DELETE FROM colors WHERE color_id = (?)",[int(color_data)])
    else:
        cur.execute("DELETE FROM colors WHERE LOWER(color_name) LIKE
LOWER(?)",[color_data])
    con.commit()

# Deletes a nail design from the database.
# Argument format: int, String or String of an int.
def deleteNailDesign(design_data):
    design_data = str(design_data)
    if design_data.isnumeric():
        cur.execute("DELETE FROM designs WHERE design_id =
(?)",[int(design_data)])
    else:
        cur.execute("DELETE FROM designs WHERE LOWER(design_name) LIKE
LOWER(?)",[design_data])
    con.commit()
```

```
# Deletes a nail texture from the database.
# Argument format: int, String or String of an int.
def deleteNailTexture(texture_data):
    texture_data = str(texture_data)
    if texture_data.isnumeric():
        cur.execute("DELETE FROM textures WHERE texture_id =
(?)",[int(texture_data)])
    else:
        cur.execute("DELETE FROM textures WHERE LOWER(texture_name) LIKE
LOWER(?)",[texture_data])
    con.commit()
```

III. PIELIKUMS. APLIECINĀJUMS PAR AUTORA MANTISKO TIESĪBU NODOŠANU

APSTIPRINĀTS
ar Vidzemes Augstskolas rektora
2017.gada 17.maija rīkojumu Nr.7-r

APLIECINĀJUMS
par autora mantisko tiesību nodošanu

Gada Projekts (turpmāk – Darbs)
kvalifikācijas darbs/bakalaura darbs/maģistra darbs/gada projekts

"NAGU 3D MODELĒŠANA UN DIZAINU DATUBĀZE",
darba nosaukums

izstrādāts Vidzemes Augstskolas Inženierzinātņu fakultātē

Pamatojoties uz Autortiesību likuma 15. pantā noteiktajām mantiskajām tiesībām, kuras darba autors var nodot trešajām personām,

piekrītu, ka mans Darbs tiek padarīts sabiedrībai pieejams bez maksas pilnā apjomā

nepiekrītu, ka manu Darbu padara sabiedrībai pieejamu
Norādīt pamatotu iemeslu:

Darba autors: _____, Dens Enrijs Lakučs , ___.___.20__.
paraksts *vārds, uzvārds* *datums*

IV. PIELIKUMS. APLIECINĀJUMS PAR DARBA ATBILSTĪBU

APLIECINĀJUMS
par darba atbilstību

Gada projekts

"NAGU 3D MODELĒŠANA UN DIZAINU DATUBĀZE "

izstrādāts Vidzemes Augstskolas Inženierzinātņu fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi un tajā ir atsauces uz visām izmantotajām citu autoru atziņām un datiem. Darbs izstrādāts saskaņā ar ViA ētikas pamatprincipiem, Studējošo akadēmiskās ētikas nolikumu un fakultātes metodiskajiem norādījumiem. Apzinos, ka plaģiāta konstatēšanas gadījumā darbs tiks noraidīts.

Iesniedzot darbu, uzņemos atbildību par jebkuras konfidenciālas informācijas, kas iegūta darba izstrādes gaitā, neizplatīšanu.

Darba autors:	Dens Enrijs Lakučs *autora vārds un uzvārds*	/ *paraksts*	/ *datums*
Darbs iesniegts fakultātē	/ *fakultātes vecākā speciālista vārds un uzvārds*	/ *paraksts*	/ *datums*
Rekomendēju darbu aizstāvēšanai *(aizpildīt, ja fakultātē noteikts)*	Dr.sc.ing. Kaspars Osis *darba vadītāja zinātniskais grāds, vārds un uzvārds*	/ *paraksts*	/ *datums*
Darbs aizstāvēts 202_.gada ___. _____	ar vērtējumu: *vērtējums cipariem*		(*vērtējums vārdiem*)
Valsts pārbaudījumu komisijas priekšsēdētājs *(bakalaura un maģistra darbam*) vai studiju programmas direktors *(gada projektam)* (vajadzīgo atstāt)	/ *valsts pārbaudījumu komisijas priekšsēdētāja vai studiju programmas direktora vārds, uzvārds*	/ *paraksts*	/ *datums*