

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки»

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Параметрична 3D-модель промислового робота-маніпулятора

Здобувача(ки) групи ІТ-91 Веретельнікова Миколи Вадимовича
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ (підпис)

Микола ВЕРЕТЕЛЬНІКОВ

(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник старший викладач кафедри «Інформаційні технології»

к.т.н., Едуард КУЗНЄЦОВ

(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В. о. зав. кафедри ІТ

_____ Світлана ВАЩЕНКО
« _____ » _____ 2023 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ
Веретельнікову Миколі Вадимовичу

1 Тема роботи Параметрична 3D-модель робота-маніпулятора

керівник роботи Кузнєцов Едуард Геннадійович, к.т.н., старший викладач,

затверджені наказом по університету від « 29 » 05 2023 р. №0588-VI

2 Строк подання студентом роботи « 13 » 06 2023 р.

3 Вхідні дані до роботи Технічне завдання на розробку

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) вступ, аналіз предметної області, огляд останніх досліджень і публікацій, аналіз програмних продуктів-аналогів, постановка задачі, моделювання та проектування, моделювання варіантів використання інформаційної системи, моделювання інформаційної системи в IDEF0, засоби реалізації та методи моделювання об'єктів, розробка 3D-моделей, моделювання в Blender, робота в русійному середовищі Unity, висновки, список використаних джерел, додатки

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Актуальність, Постановка задачі, Аналіз продуктів-аналогів, порівняння продуктів аналогів, Вимоги до реалізації, Моделювання та проектування, Діаграма варіантів використання, Засоби реалізації, Висновки

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.01.2023

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження предметної області	24.01.2023	Виконано
2	Моделювання	26.03.2023	Виконано
3	Розробка системи управління	19.05.2023	Виконано
4	Тестування додатку	10.06.2023	Виконано
5	Архівація проекту	11.06.2023	Виконано

Студент _____
(підпис)

Микола ВЕРЕТЕЛЬНИКОВ

Керівник роботи _____
(підпис)

к.т.н., ст. викл. Едуард КУЗНЄЦОВ

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Параметрична 3D-модель промислового робота-маніпулятора».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку джерел із 22 найменувань, додатків. Загальний обсяг роботи – 64 сторінки, у тому числі 42 сторінки основного тексту, 3 сторінки списку використаних джерел, 19 сторінок додатків.

В першому розділі проведено огляд останніх досліджень за темою роботи, порівняльний аналіз продуктів-аналогів, та сформульовані мета і задачі виконуваного проекту.

В другому розділі описано використані засоби та методи розробки проекту, змодельовані діаграми у нотації IDEF0 (діаграма функціонального моделювання) та Use Case (діаграма варіантів використання). Обрані програмні засоби реалізації та середовище розробки проекту (Blender та Unity3D).

В третьому розділі описана та продемонстрована розробка 3D-моделі та додатку, розроблено механізм програмування рухів моделі по користувацьких файлах з даними для руху ланок моделі, показано результат виконаного проекту, а також проведено тестування роботи додатку.

Результатом проведеної роботи є створена 3D-модель та додаток з можливістю програмованого керування моделлю робота-маніпулятора.

Ключові слова: робот-маніпулятор, візуалізація, 3D-модель, програмування рухів, Blender, Unity, додаток.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Огляд останніх досліджень і публікацій	8
1.2 Аналіз програмних продуктів-аналогів	9
1.3 Постановка задачі	13
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ	14
2.1 Моделювання інформаційної системи в IDEF0	14
2.2 Моделювання варіантів використання інформаційної системи	16
2.3 Засоби реалізації та методи моделювання об'єктів	17
3 РОЗРОБКА 3D-МОДЕЛЕЙ	19
3.1 Моделювання в Blender	19
3.2 Робота в русійному середовищі Unity	28
3.3 Тестування	36
ВИСНОВКИ	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43
ДОДАТОК А	46
ДОДАТОК Б	48
ДОДАТОК В	61

ВСТУП

Оскільки технологія прогресує з кожним днем, роботи замінюють людей для виконання повторюваних завдань. Головним чином, це винахід, у якому більше ніж дві галузі, такі як машинобудування, електротехніка та інженерія інформатики, об'єднані для виконання різних завдань однією машиною. Він може бути стаціонарним або рухомим, залежно від мети використання. Їх здатність виконувати складні та рутинні завдання з високою точністю та швидкістю робить їх важливим фактором у підвищенні продуктивності та ефективності виробничих процесів.

В останні роки параметричне моделювання стало важливим інструментом у галузі проектування та розробки роботів-маніпуляторів. Параметрична 3D-модель представляє конструкцію робота у вигляді гнучкого та настроюваного набору параметрів, які дозволять ввести керовані обмеження на положення ланок робота в просторі, нададуть можливість завдання руху ланок, спростять процес проектування чи адаптації робота до різних завдань в цілому. Однак досі існує необхідність у розробці та вдосконаленні методів параметричного моделювання промислових роботів-маніпуляторів з метою підвищення точності, надійності та продуктивності системи.

Метою даної дипломної роботи є дослідження та розробка 3D-моделі робота-маніпулятора з використанням сучасних методів та технологій у галузі 3D-моделювання, розробити систему керування, яка дозволить користувачу напряму керувати рухом робота-маніпулятора.

Для досягнення поставленої мети потрібно виконати такі задачі:

- проаналізувати предметну область 3D моделювання;
- обрати засоби та методи реалізації задачі;
- створити модель робота-маніпулятора;

– створити додаток, в якому користувач буде мати можливість керувати роботом-маніпулятором.

Практичне значення роботи полягає в тому, що отримана модель буде здатна відтворювати рухи та функції робота-маніпулятора у віртуальному середовищі, що дозволить в подальшому використовувати додаток для симуляції різних ситуацій під час роботи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Робот-маніпулятор – це промислове обладнання, яке виконує функції людської руки. Це може бути як окремий пристрій, так і складова роботизованого комплексу.

З кожним роком зростає потреба у зниженні експлуатаційних витрат на підприємствах, через що попит на промислову робототехніку постійно зростає. Це сприяє розвитку галузі. Зменшення витрат на підприємствах можна досягти за допомогою зниження кількості помилок, втрат сировини та числа нещасних випадків. Також необхідно зробити підприємства більш продуктивними та технологічно гнучкими. Усі ці питання вирішують роботи-маніпулятори. Вони здатні працювати з високою точністю навіть у небезпечних умовах.

Як зазначають дослідники у листопаді 2022 року, рівень промислової автоматизації значно зріс останніми роками через підвищення продуктивності підприємств, зниження витрат на робочу силу, підвищення ефективності виробничих процесів та покращення якості продукції. У майбутньому такі галузі, як виробництво продуктів харчування та напоїв, автомобільна та нафтогазова промисловість буде більш схильні до автоматизації через гостру конкуренцію, яка підштовхує до підвищення швидкості виробництва та покращення якості товару [1].

Застосування промислових роботів постійно розширюється: якщо раніше основним споживачем промислових роботів було автомобілебудування, то нині відбувається зростання їх застосування у харчовій, фармацевтичній, електротехнічній/електронній та інших галузях обробної промисловості, що свідчить про нові тенденції розвитку та способи застосування робототехніки.

В автомобільній промисловості маніпулятори використовуються для зварювання автомобілів. Також вони можуть виконувати завдання зі збирання та монтажу складових автомобіля, таких як двері, вікна та сидіння. Крім цього, автоматизовані пристрої використовуються для фарбування машин. Також маніпулятори можуть обробляти шибки, тобто займатися різкою і шліфуванням.

Ще одна сфера, в якій використовуються роботи-маніпулятори, – це виробництво електроніки. Висока точність дозволяє машинам легко справлятися з такими операціями, як складання та монтаж друкованих плат, встановлення та підключення мікročіпів, конденсаторів та резисторів. Вони також можуть керувати процесом паяння та наступного маркування готової продукції.

Визначивши всю користь та різноманіття галузей застосування промислових роботів-маніпуляторів, можна зробити висновок про необхідність дослідження таких інструментів автоматизації та подібних їм. Саме тому ця робота присвячена моделюванню прототипу руки робота-маніпулятора та демонстрації її можливостей за допомогою віртуальної симуляції.

1.2 Аналіз програмних продуктів-аналогів

Дуже важливою частиною кожного проекту є аналіз продуктів-аналогів. Даний аналіз допоможе зрозуміти які переваги і недоліки мають існуючі на просторах інтернету 3D-моделі та зрозуміти на що потрібно орієнтуватися при моделюванні та розробці.

Для успішної постановки задачі необхідно проаналізувати існуючі продукти-аналоги в рамках створення порівняльної таблиці та визначення конкретних характеристик на розробку власного прототипу.

Існує велика кількість різноманітних та широко розповсюджених на підприємствах моделей. Проте, для виконання даного дослідження необхідно обрати найбільш підходящу. Проаналізувавши більшість актуальних на

сьогоднішній день представлених на ринку моделей, було вирішено обрати робота-маніпулятора FANUC LR Mate 200iC. Цей робот є компактним з легкою і тонкою конструкцією. Він має 6 осей та навантаження до 5 кг і діапазон роботи 704 мм. Його компактні розміри дозволяють легко працювати в закритих або вузьких просторах, що робить його ідеальним вибором для застосування з невеликим корисним навантаженням і обмеженою площею підлоги. Незважаючи на свій розмір, LR Mate 200iC може працювати у розширеному робочому просторі з можливістю повного нахилу назад. Можливість розміщення в зворотному напрямку разом з різними варіантами монтажу, включаючи підлогові, обернення або нахилення, надають виробникам підвищену гнучкість в експлуатації. Зазвичай LR Mate 200iC використовують у лабораторіях чи чистих приміщеннях, оскільки він має закриту механічну конструкцію, що унеможливорює виходження кабелів чи шлангів. Незалежно від робочої середовища, LR Mate 200iC може виконувати ряд завдань, включаючи роботизовану збірку, автоматизовану обробку матеріалів, обслуговування машин та автоматизований вибір і розміщення [2].

3D-моделей робота-маніпулятора FANUC LR Mate 200iC на просторах інтернету не дуже багато і більшість з них по якості розробки між собою однакові, але ще можна знайти декілька моделей, які можна порівняти між собою.

Перша модель – це безкоштовний варіант, який розташований у відкритому доступі (рис. 1.1). Його головна проблема у тому, що він завантажується в нерозповсюдженому форматі. Для того, щоб користувач надалі міг з цією моделлю працювати, йому потрібно купити та встановити програму, в якій модель була зроблена, і конвертувати файл у потрібний формат для подальшого використання.

Друга модель – платний варіант, який стоїть великих грошей (рис. 1.2). Сама модель зроблена дуже якісно і завантажується вона у зручному форматі, на відміну від попередньої моделі, що дуже зручно для користувача для подальшої роботи.

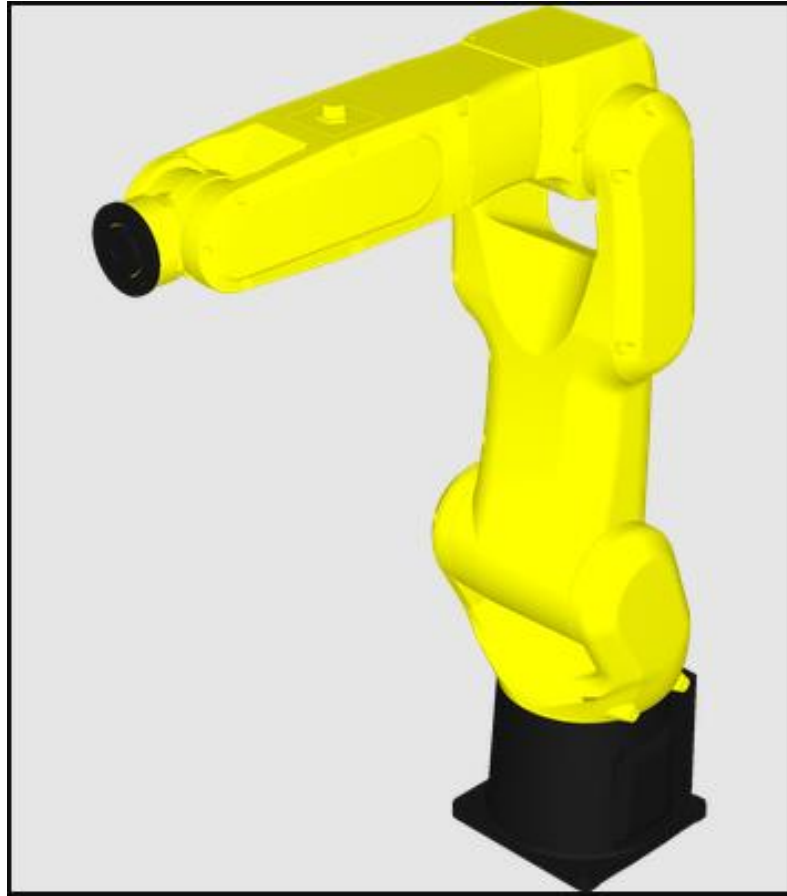


Рисунок 1.1 – Безкоштовна 3D-модель

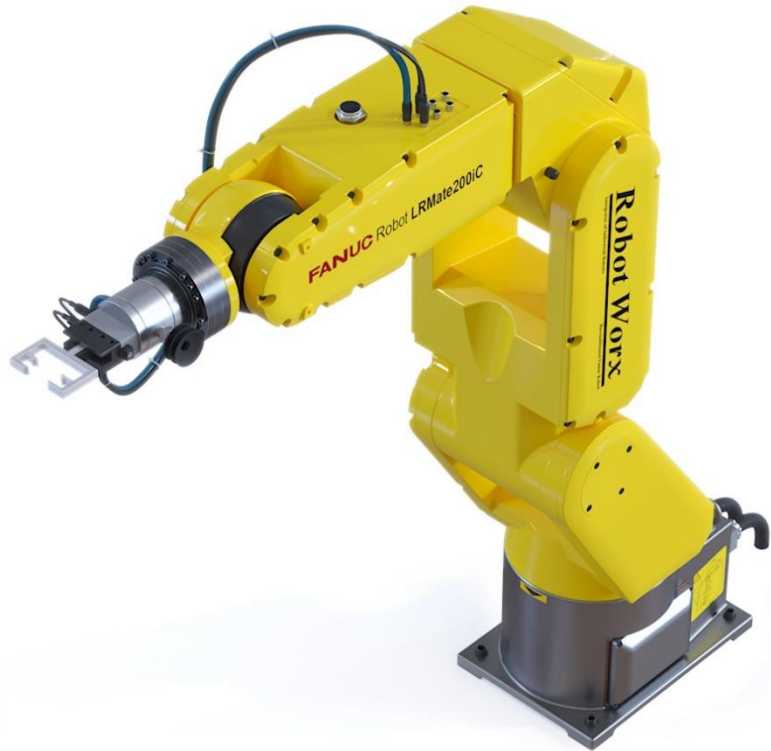


Рисунок 1.2 – Платна 3D-модель

В обох розглянутих варіантах відсутня можливість для керування роботом, тобто користувачу потрібно буде самому розробляти скрипти для маніпулювання та, при необхідності вдосконалювати модель для більш коректної її роботи.

Зробимо таблицю, де ми порівняємо обидві моделі більш детально. Таблиця буде ґрунтуватись на інформації, яка доступна на сайті для завантаження моделей. Щоб було більш зрозуміло, позначимо безкоштовну модель – «Модель1», а платну – «Модель2».

Таблиця 1 – Порівняння 3D-моделей

Характеристика	Модель1	Модель2
Роздільна здатність	+	-
Якість текстуровання	-	+
Кількість полігонів	-	+
Анімаційні можливості	+	+
Керування роботом	-	-
Формат файлу	-	+
Рівень деталізації	-	+

Проаналізувавши ці моделі, можна побачити, що платна модель, яка зображена на рисунку 2, має багато плюсів, які недостають безкоштовній моделі (рис. 1). Тому, задача буде стояти в тому, щоб зробити 3D-модель, яка по якості буде схожа на платну модель(рис.2), але щоб вона була більш доступною, завантажувалась у зручному форматі, для легкого доступу до 3D-моделі та у користувача була можливість керувати роботом.

1.3 Постановка задачі

Метою проекту є створення параметричної тривимірної моделі робота-маніпулятора FANUC LR Mate 200iC, яка в подальшому буде використовуватись як ознайомчий та навчальний інструмент для вивчення руху робота, його зон досяжності та проведення симуляцій, які допоможуть передбачити поведінку робота в різних умовах. Для забезпечення рухливості ланок моделі вона робиться параметричною – положення її ланок в просторі задається не жорстко, а за допомогою параметрів, які можна змінювати в процесі роботи з моделлю на вимогу користувача: вручну або по заданій програмі пересування та поворотів ланок через файл зі значеннями параметрів.

Для досягнення цієї мети потрібно виконати наступні задачі:

- провести аналіз предметної області та дослідити існуючі аналоги – комп'ютерні моделі робота-маніпулятора;
- обрати відповідні програмні інструменти, методи та середовище проектування для створення параметричної моделі;
- провести моделювання процесу використання моделі та взаємодії з нею користувача;
- розробити модель, розробити програмні елементи (скрипти) для завдання та відтворення руху робота;
- створити додаток для надання користувачу можливості керувати роботом-маніпулятором, протестувати його.

В результаті виконання проекту необхідно надати користувачу розуміння поведінки робота, вивчити його параметри руху та можливості застосування для певних практичних задач. Застосування розроблюваного додатку допоможе підвищити інформативність спеціальних дисциплін та мінімізувати витрати навчальних закладів на придбання наочних матеріалів для забезпечення викладання технічних дисциплін.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

2.1 Моделювання інформаційної системи в IDEF0

В першу чергу, для коректного моделювання та проектування роботи, необхідно розробити контексту діаграму IDEF0, яка надає графічне представлення функціональної моделі. Вона надає змогу більш детально розглянути і зрозуміти призначення і головну функцію інформаційної системи. IDEF0 – це методологія функціонального моделювання. Вона використовується для створення функціональної моделі, що відображає структуру і функції системи, а також потоки інформації і матеріальних об'єктів, що зв'язують ці функції.[2].

Кожний блок в діаграмі являє собою процеси, що відбуваються протягом якогось часу. Взаємозв'язки між процесами позначаються стрілками.

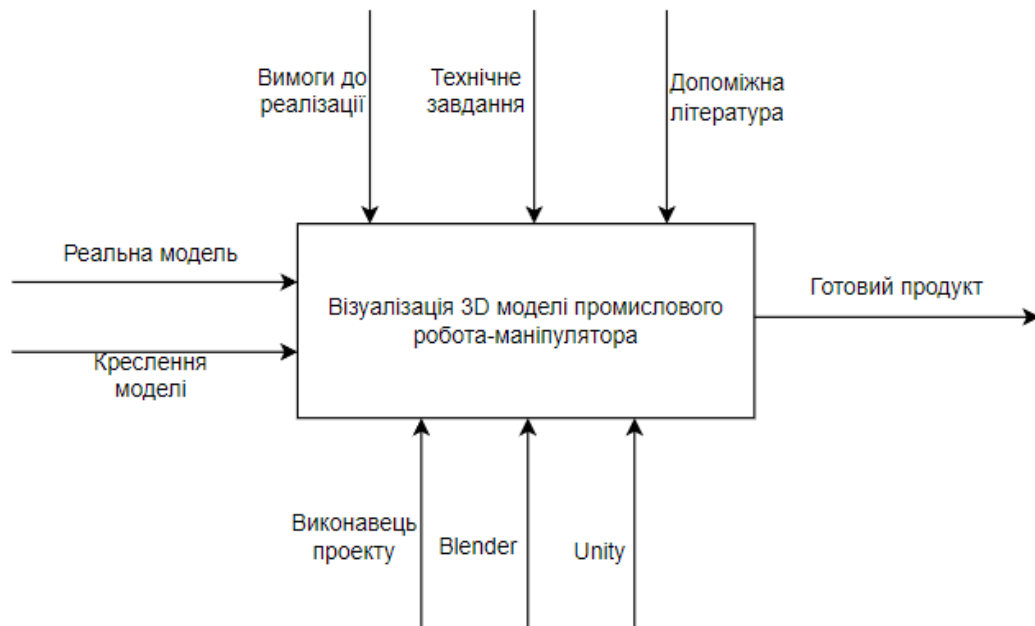


Рисунок 2.1 – Загальна модель в IDEF0

Контекстна діаграма на рисунку 2.1 надає загальне представлення яка інформація є на вході, які надаються інструменти для вирішення задачі та відображає механізми даного процесу.

Далі, слід виконати декомпозицію контекстної діаграми IDEF0 шляхом розбиття головної функції на дрібні більш оптимальні підпроцеси. Таке графічне представлення дозволяє отримати повне представлення які процеси, елементи управління та дані задіяні певних етапах роботи (рис. 2.2).

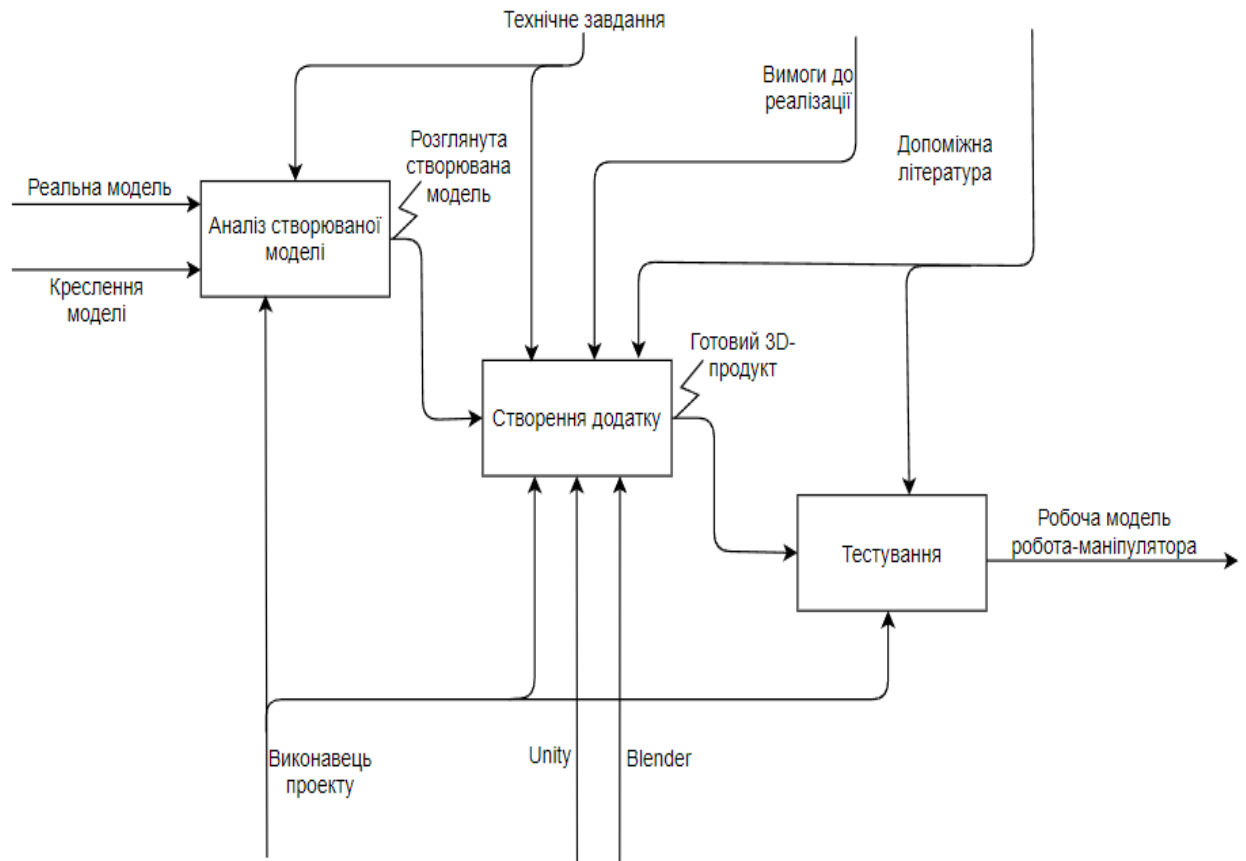


Рисунок 2.2 – Перший рівень декомпозиції

Створивши перший рівень декомпозиції на рисунку 2.3 можна спостерігати послідовні процеси після виконання котрих є вихідні дані, які потрібні для початку виконання наступного процесу.

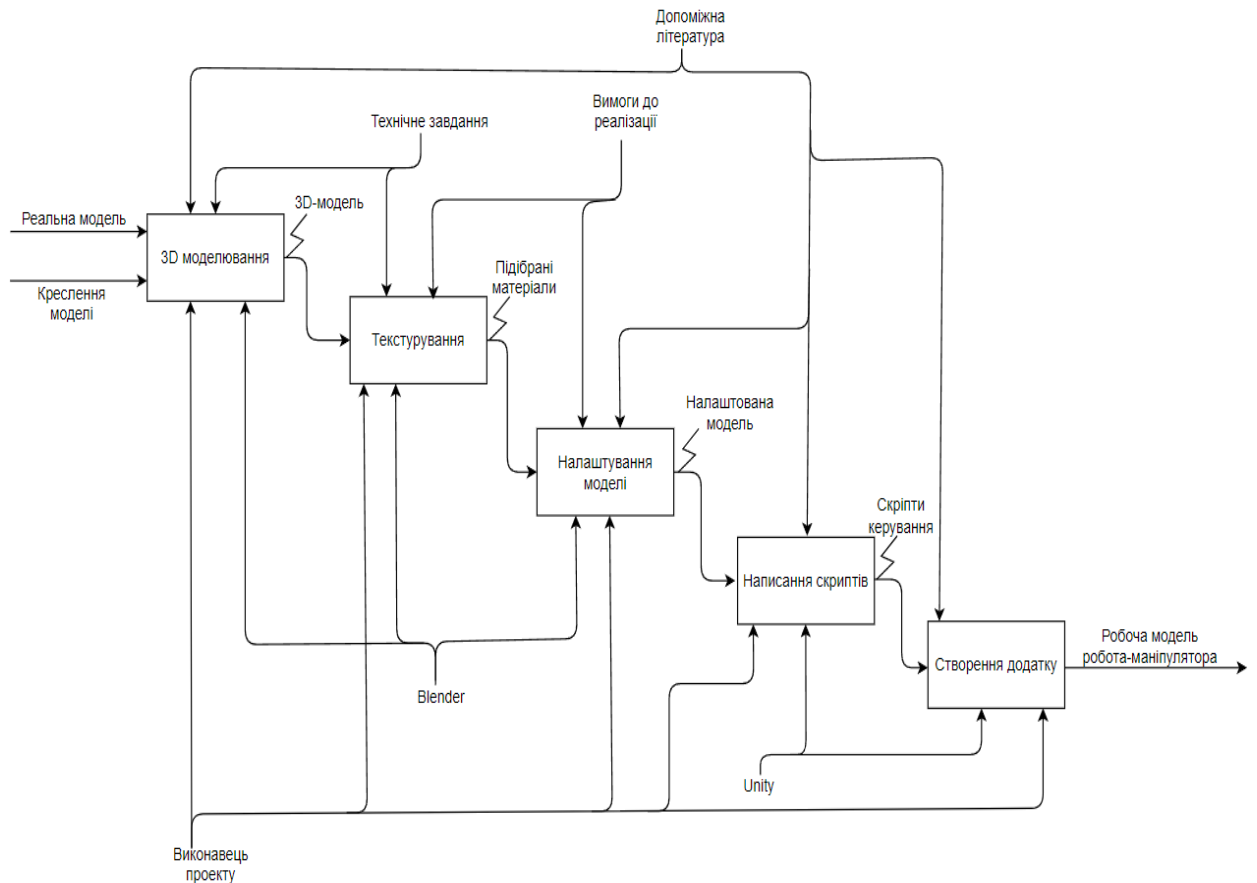


Рисунок 2.3 – Другий рівень декомпозиції

2.2 Моделювання варіантів використання інформаційної системи

Діаграми варіантів використання описують поведінку системи з точки зору користувача. Вони дозволяють визначити межі системи, а також відношення між системою та зовнішнім середовищем. Діаграми варіантів використання використовують для візуалізації вимог до системи.[1]

В діаграмі на рисунку 2.4 відображено 2 актора і показано їх зв'язок з функціями, які вони виконують.

Суцільними лініями відображено відношення асоціації, штриховими лініями показані відношення включення.

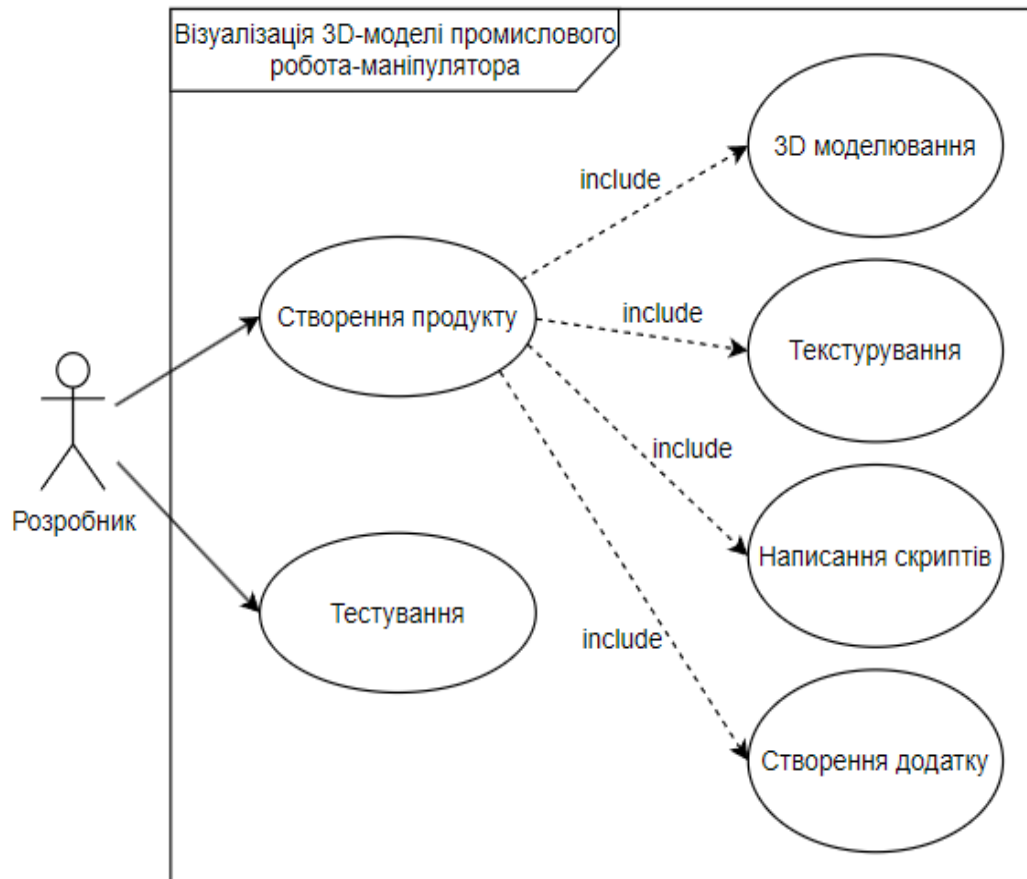


Рисунок 2.4 – Діаграма варіантів використання

2.3 Засоби реалізації та методи моделювання об'єктів

Для виконання даного проекту було обрано два програмних засоби – Blender та Unity3D.

Для створення 3D-моделей було вирішено обрати програму Blender, так як дана програма є безкоштовною, та для початківців функціонал даного середовища повністю задовольняє потреби розробника.

Для симуляції руху робота та його керування була обрана програма Unity. Основна перевага даного середовища є те, що її безкоштовно можна використовувати в навчальних цілях. Не менш важливим фактором є те, що це дуже популярне середовище для розробки, а тому при виникненні питань відповіді на них зазвичай легко можна знайти на електронних ресурсах, що значно прискорює процес розробки додатку.

При моделюванні деталей робота-маніпулятора в основному були задіяні булеві операції. Найчастіше використовувались операції об'єднання та виключення. Приклад їх робота зображено на рисунку 2.5.

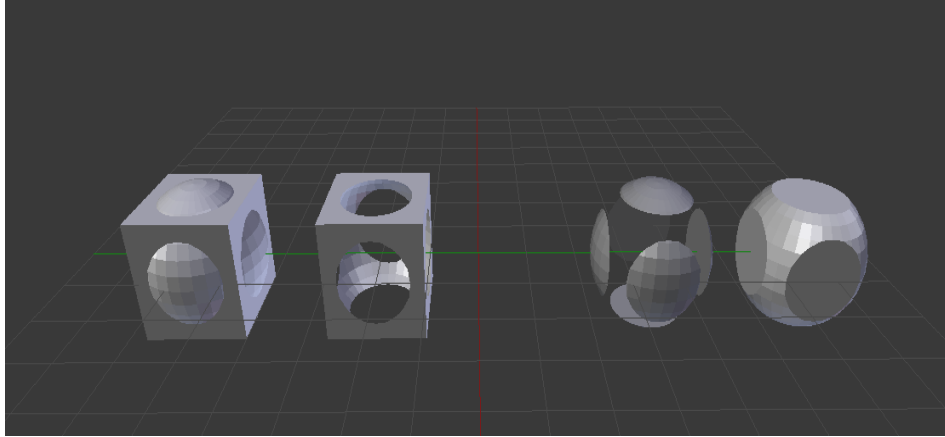


Рисунок 2.5 – Приклад роботи булевих операцій[3]

Також для підвищення якості деталей моделей були використані вбудовані модифікатори. Найбільше з них використовувались модифікатори Subdivision Surface, для згладжування поверхні моделей, та Edge Split, для створення гострих граней на загладжуваних об'єктах. Приклад роботи цих модифікаторів представлений на рисунку 2.6.

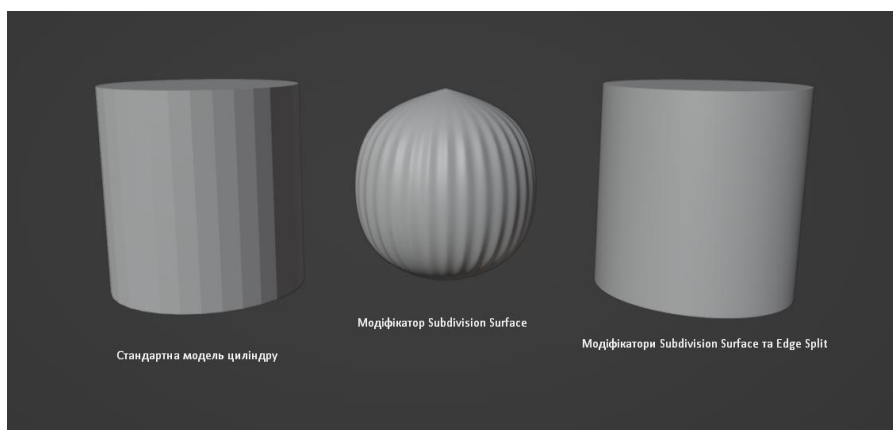


Рисунок 2.6 – Демонстрація роботи модифікаторів Subdivision Surface та Edge Split

3 РОЗРОБКА 3D-МОДЕЛЕЙ

3.1 Моделювання в Blender

Роботу було почато із проектування етапів розробки системи. Першим із визначених етапів є створення 3D-моделі промислового робота-маніпулятора у графічному редакторі Blender.

Робота починається зі створення проекту.

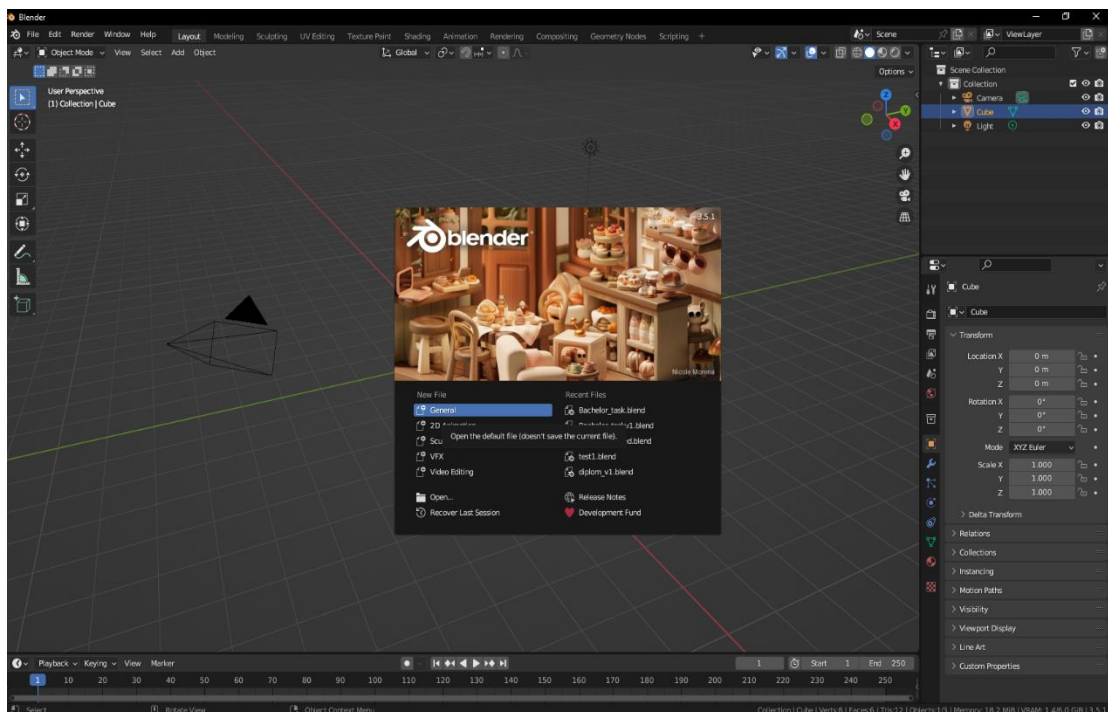


Рисунок 3.1 – Створення проекту

Наступним етапом буде почати створення базової частини робота. Для створення базової частини були використані примітивні фігури циліндричної та квадратної форми. Для придання реалістичної деталізованої форми у відповідності до референсів роботи було використано одну з найпростіших, проте ефективних функцій – булеву функцію Union. Ця функція призначена для об'єднання окремих об'єктів в один. Далі був використаний модифікатор Subdivision Surface. Він служить для збільшення кількості полігонів, тим самим згладжують грані моделі і надаючи їй більш реалістичний вигляд.

Далі, використовуючи булеву функцію Difference та примітивні фігури були створені деталі на корпусі, які потім були об'єднанні з самим корпусом вже за допомогою булевої функції Union.

В результаті ми отримуємо базову частину робота, яка представлена на рисунку 3.2.

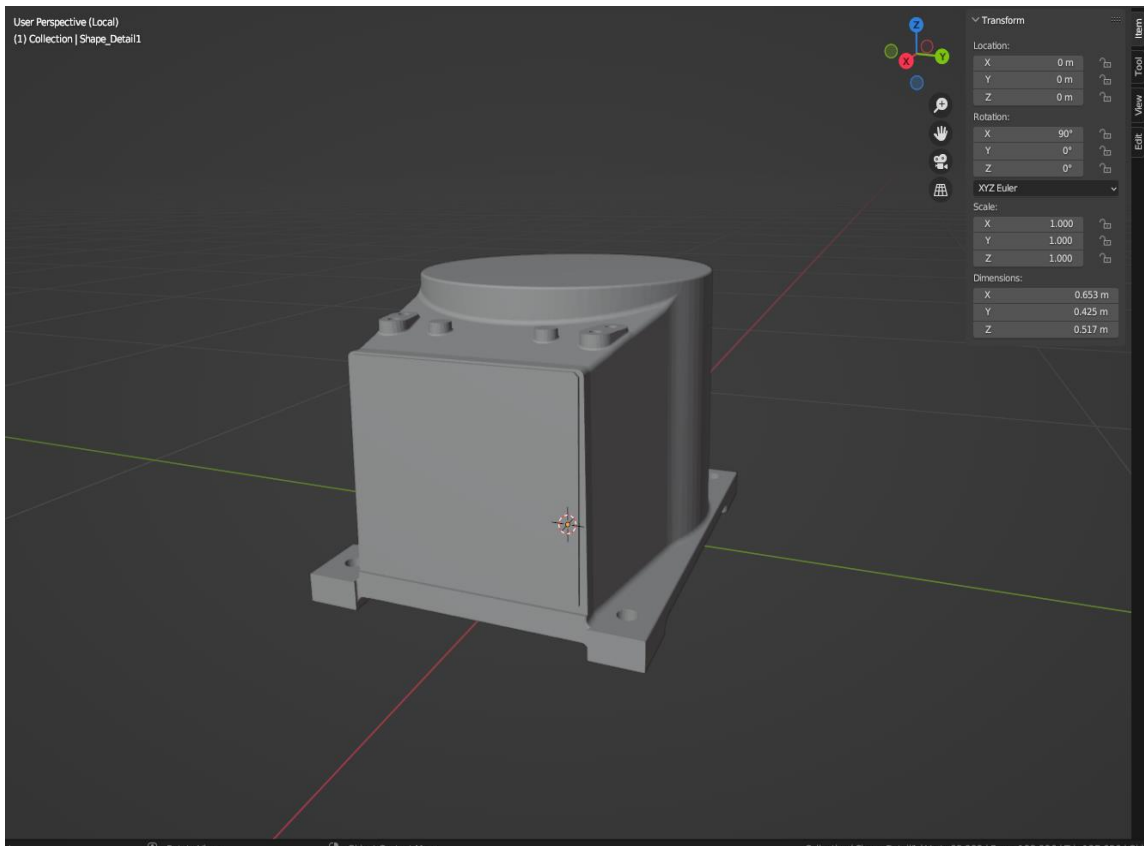


Рисунок 3.2 – Основа робота-маніпулятора

Наступна частина моделі (рис. 3.3) будувалась також з використанням булевих функцій Union і Difference.

Функція Union використовувалась для об'єднання відредагованих примітивів в одну деталь. Далі була використана функція Difference для деяких коректувань в моделі, щоб вона відповідала реальній моделі. Також, за допомогою цієї ж функції були зроблені виїмки для болтів. Для закруглення країв деталі був використаний модифікатор Bevel, за допомогою якого модель отримала більш об'ємний вигляд. Наступним етапом було застосування модифікатора Subdivision Surface для згладжування поверхонь (рис. 3.4).

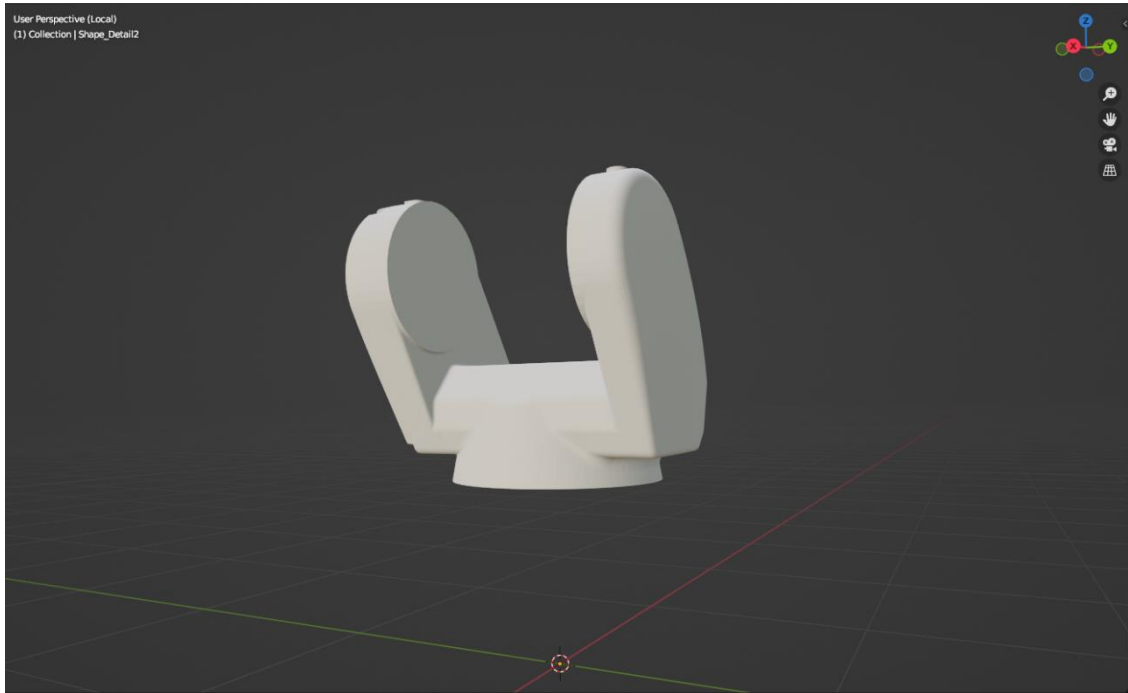


Рисунок 3.3 – Друга деталь моделі.

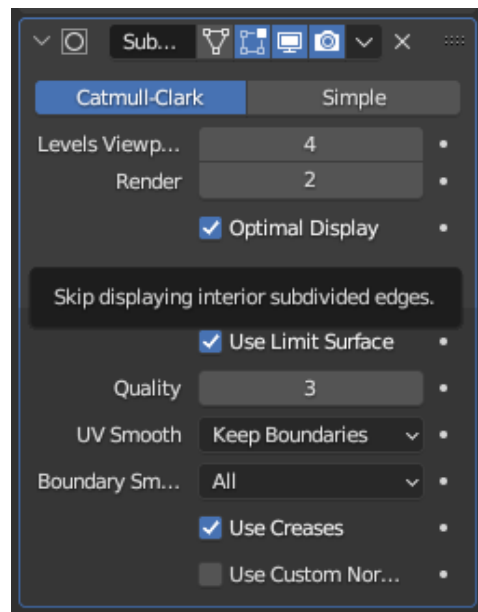


Рисунок 3.4 – Вікно налаштування Subdivision Surface

При побудові наступних деталей(рис. 3.5-3.6) використовувались ті ж самі засоби реалізації.

З початку, були взяті стандартні фігури циліндр та квадрат, яким був наданий об'єм. Далі, за допомогою булевої функції Union вони були об'єднані в один об'єкт, на якому був використаний модифікатор для згладжування

поверхні. В самому кінці була зроблена центральна виїмка (рис. 3.5) за допомогою операції Difference та фігури конус. Також за допомогою цієї операції були зроблені виїмки для болтів.

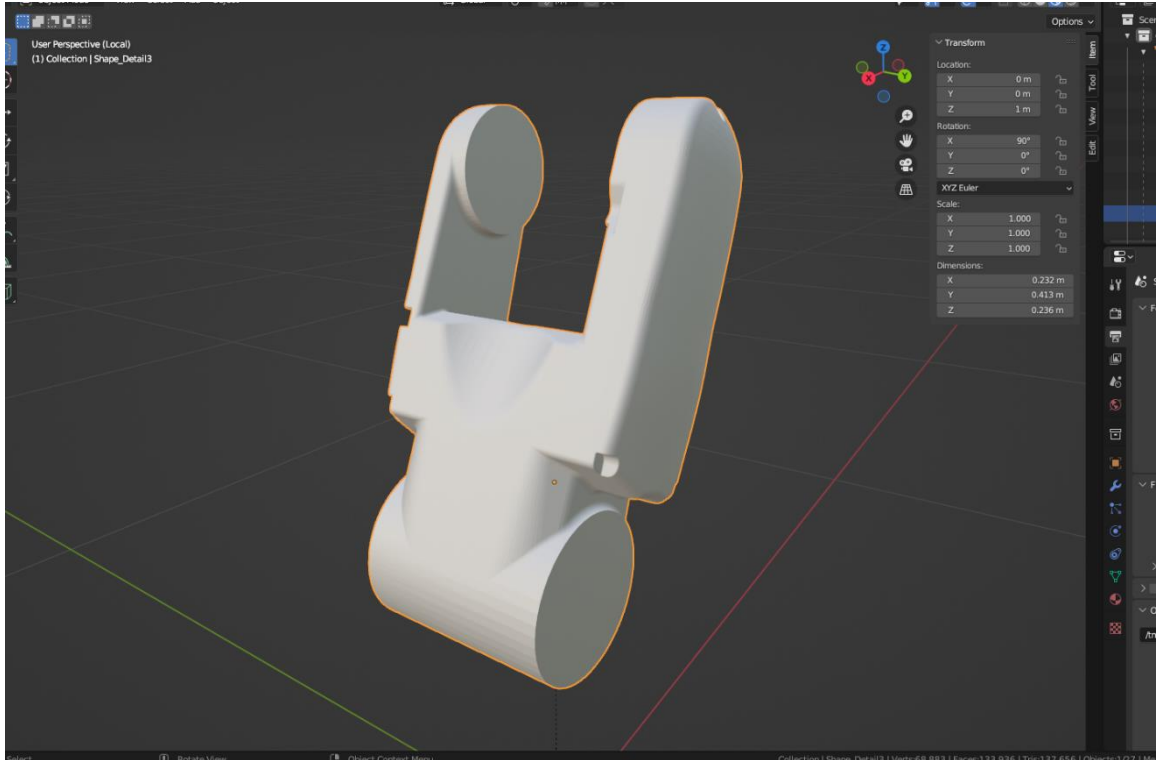


Рисунок 3.5 – Третя деталь

Для моделювання наступних деталей (рис. 3.6 – 3.7) використовувалась фігура циліндр. Заглиблення були зроблені за допомогою Union. Як і на попередніх деталях, була застосована модифікація Subdivision surface для збільшення кількості полігонів та згладжування моделі.

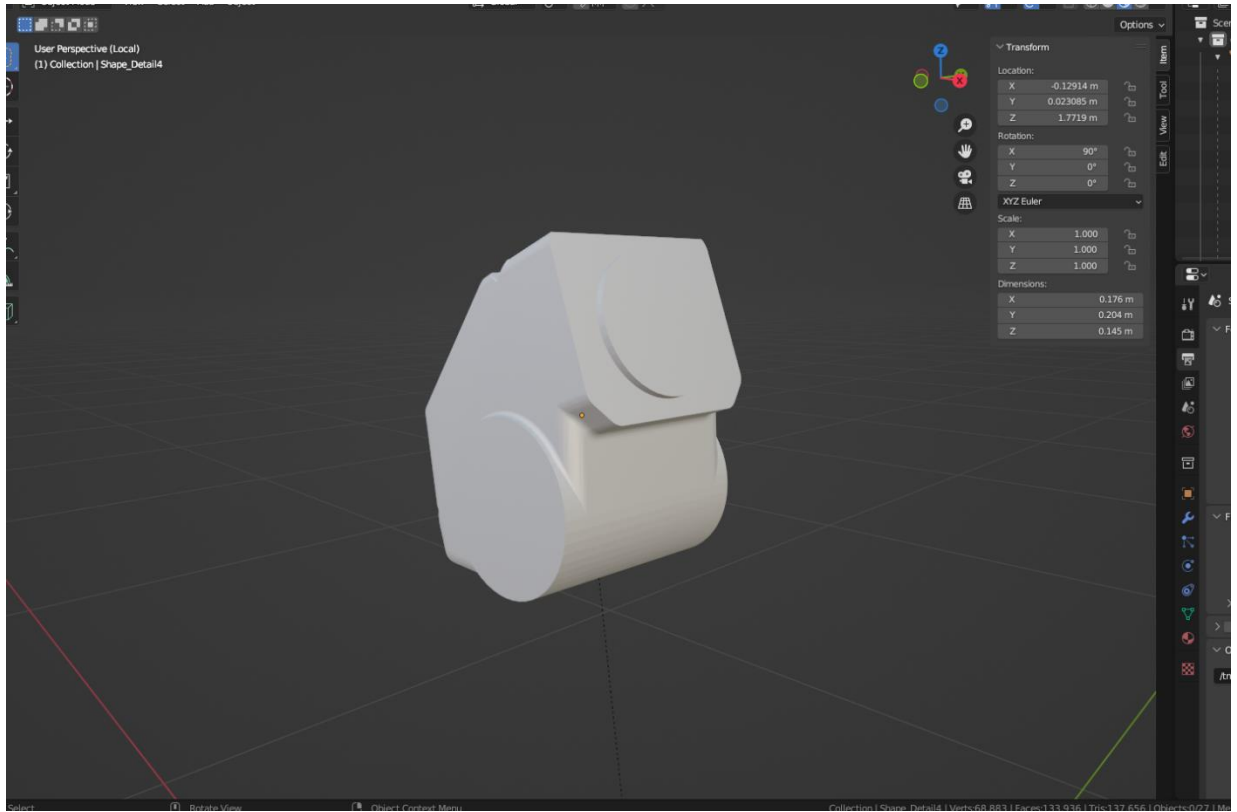


Рисунок 3.6 – Четверта деталь

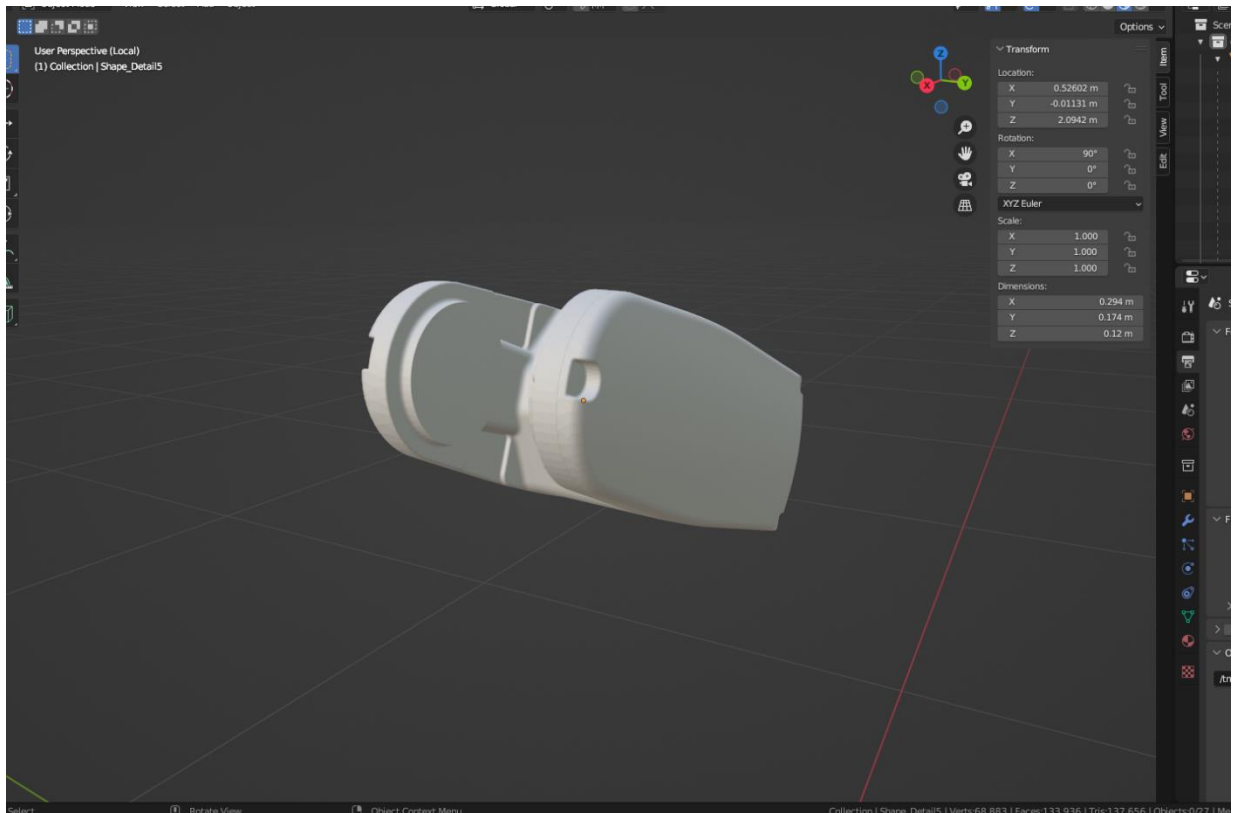


Рисунок 3.7 – П'ята деталь

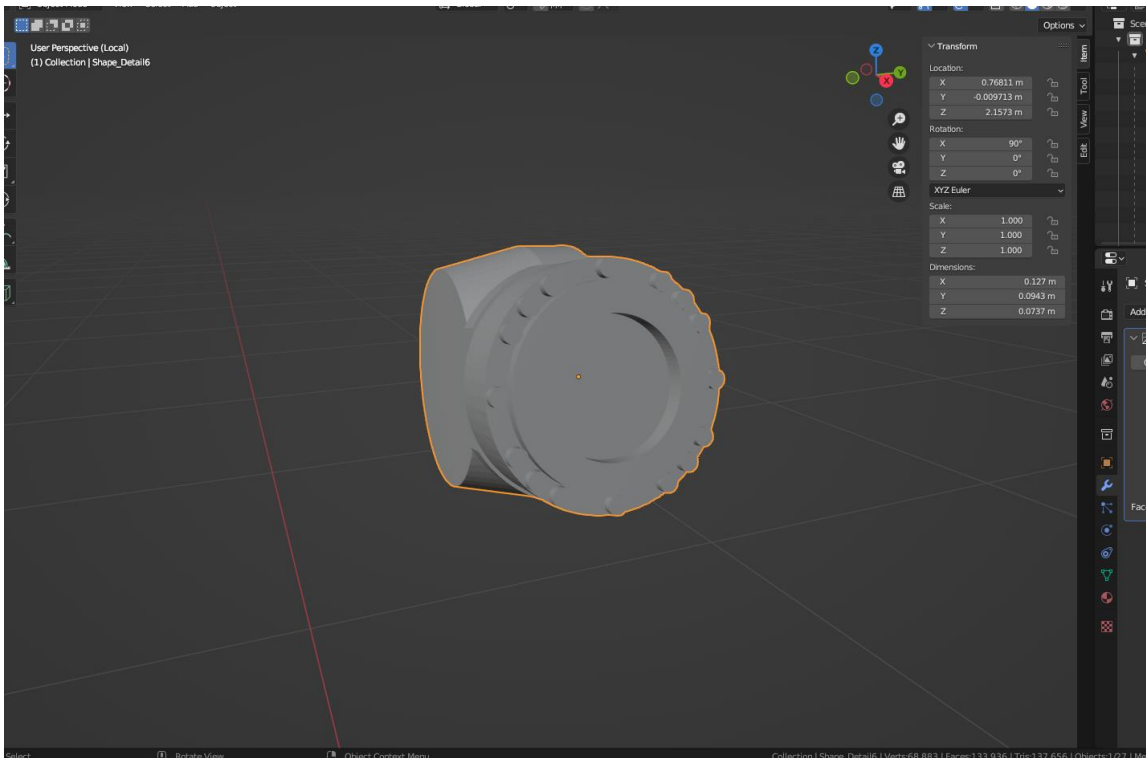


Рисунок 3.8 – Шоста деталь

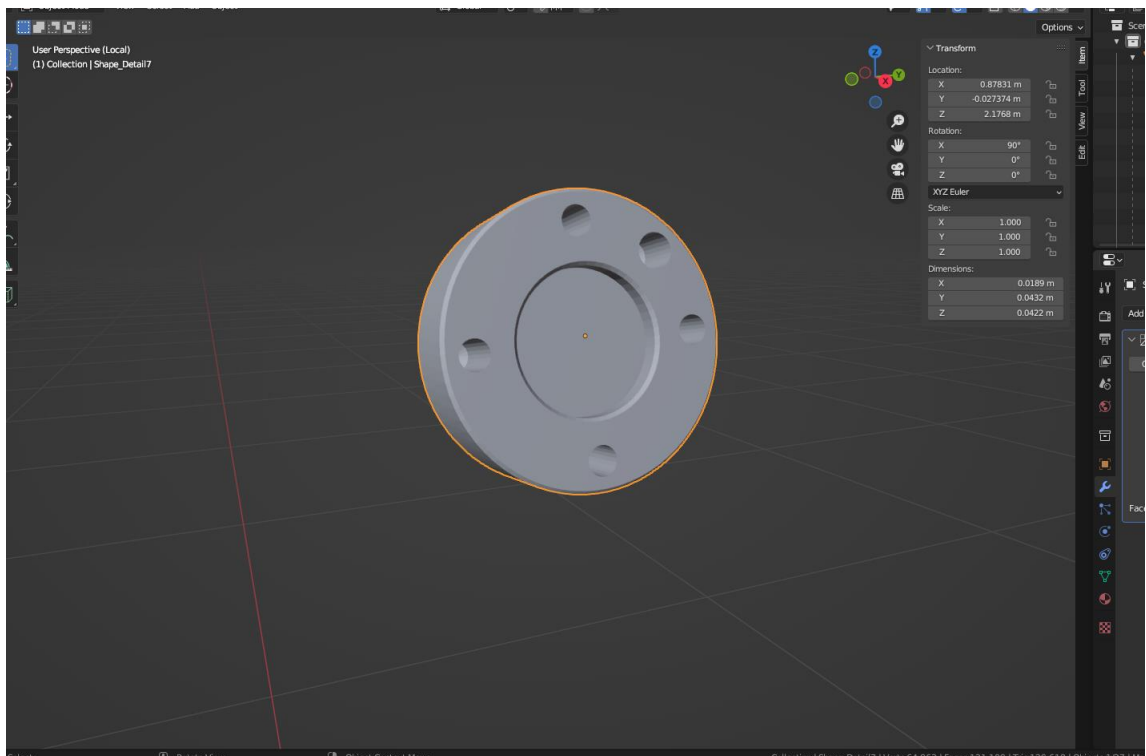


Рисунок 3.9 – Сьома деталь

Готовий вигляд моделі представлений на рисунку 3.10.



Рисунок 3.10 – Вигляд повної моделі

Наступним етапом, після процесу моделювання, було зроблене текстурування. При дослідженні моделі було виявлено, що майже весь робот зроблений з металевих складових. Саме тому, у вкладці Shading було проведено відповідне налаштування.

По перше, був обраний колір деталі (рис. 3.11). Наступним етапом було його налаштування для придання деталі металової текстури. Для цього використовувались параметри Metallic, Specular і Roughness.

Методом підбору були обрані оптимальні значення для цих параметрів (рис. 3.12).

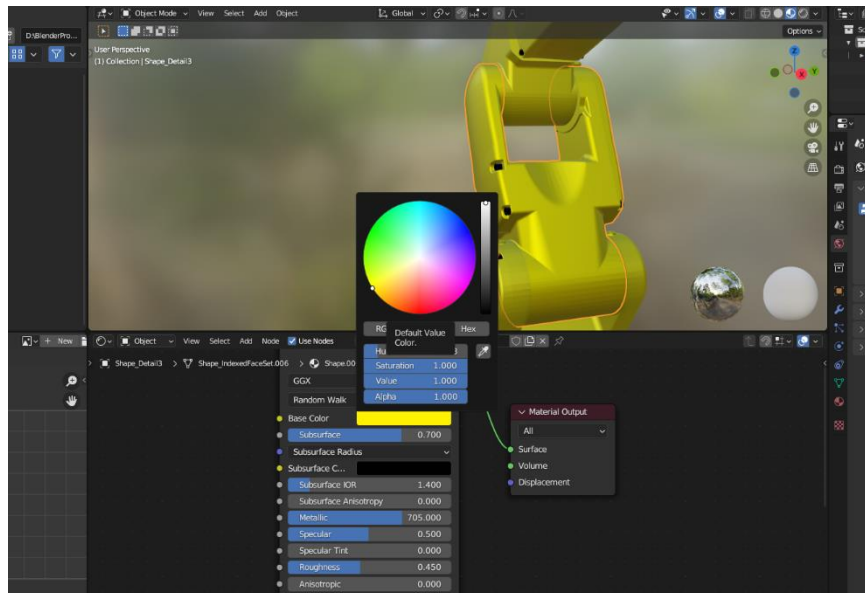


Рисунок 3.11 – Вибір кольору деталі

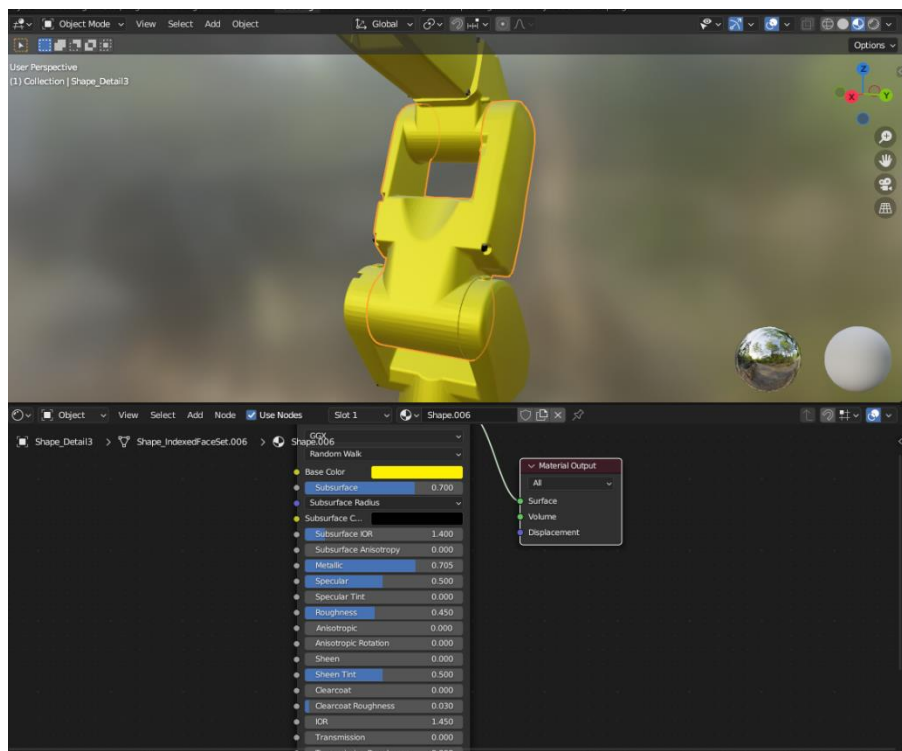


Рисунок 3.12 – Налаштування матеріалів

Використовуємо ті ж налаштування і для інших ланок моделі.

При аналізі був зроблений висновок, що базова частина і дві останні ланки механічної руки, скоріш за все, зроблені з пластику, тому для них були застосовані наступні налаштування(рис. 3.13 – 3.14).

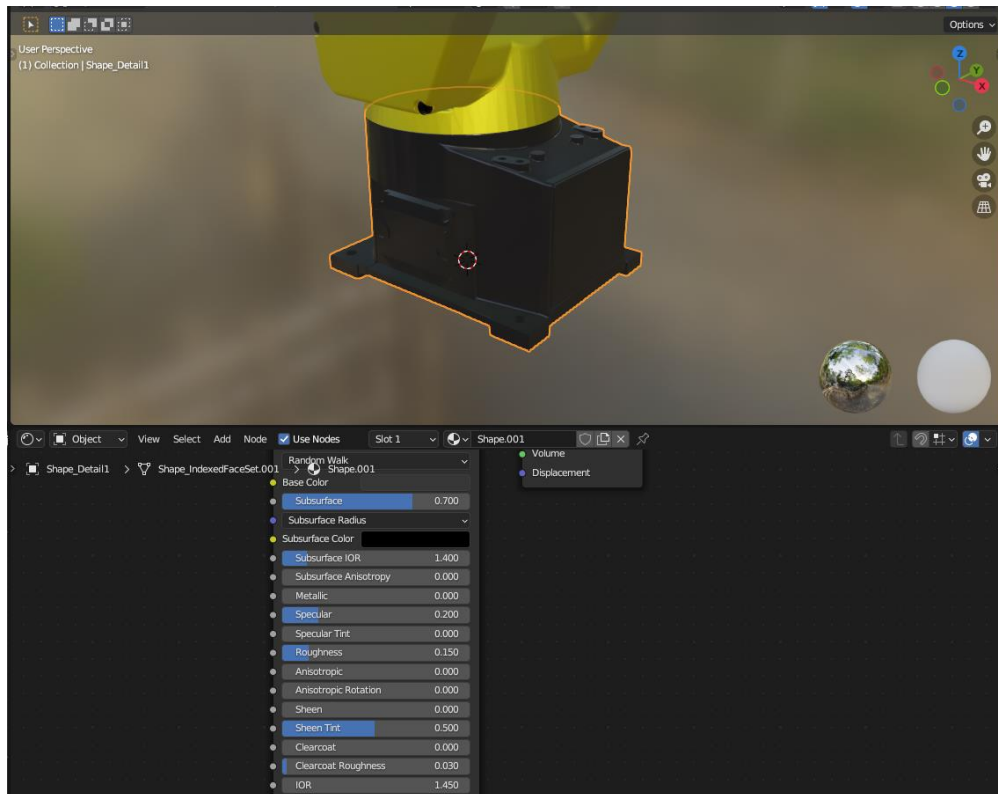


Рисунок 3.13 – Налаштування текстур базової частини

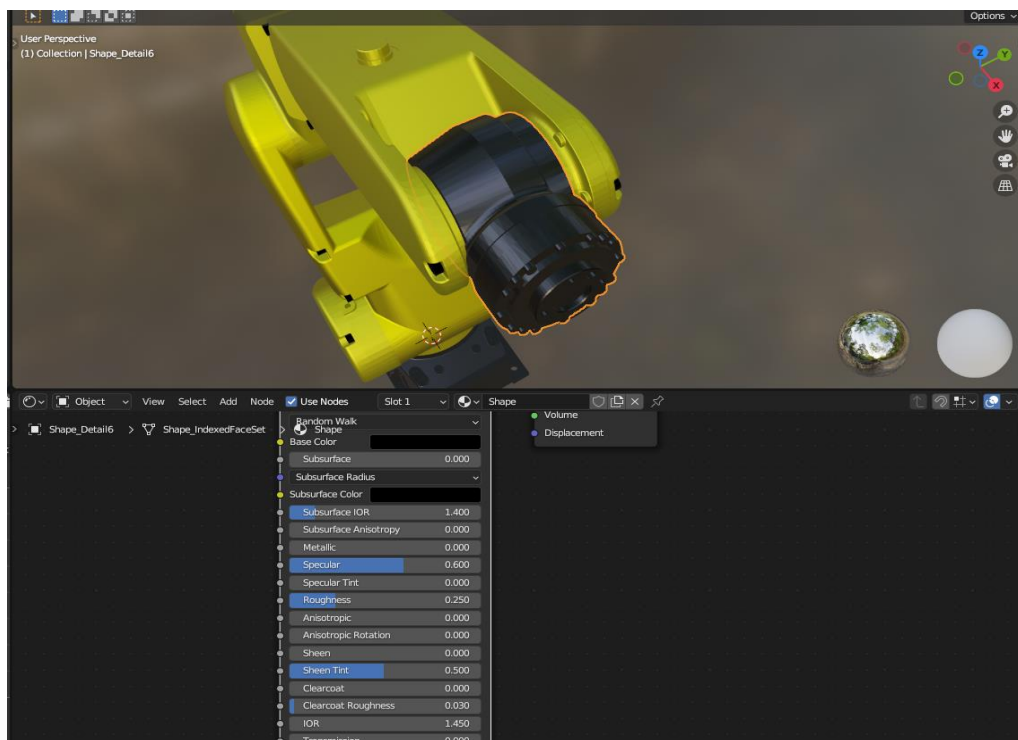


Рисунок 3.14 – Параметри текстур крайніх ланок

Перед тим, як експортувати модель в середовище Unity потрібно було налаштувати модель, щоб вона коректно відображалась в Unity.

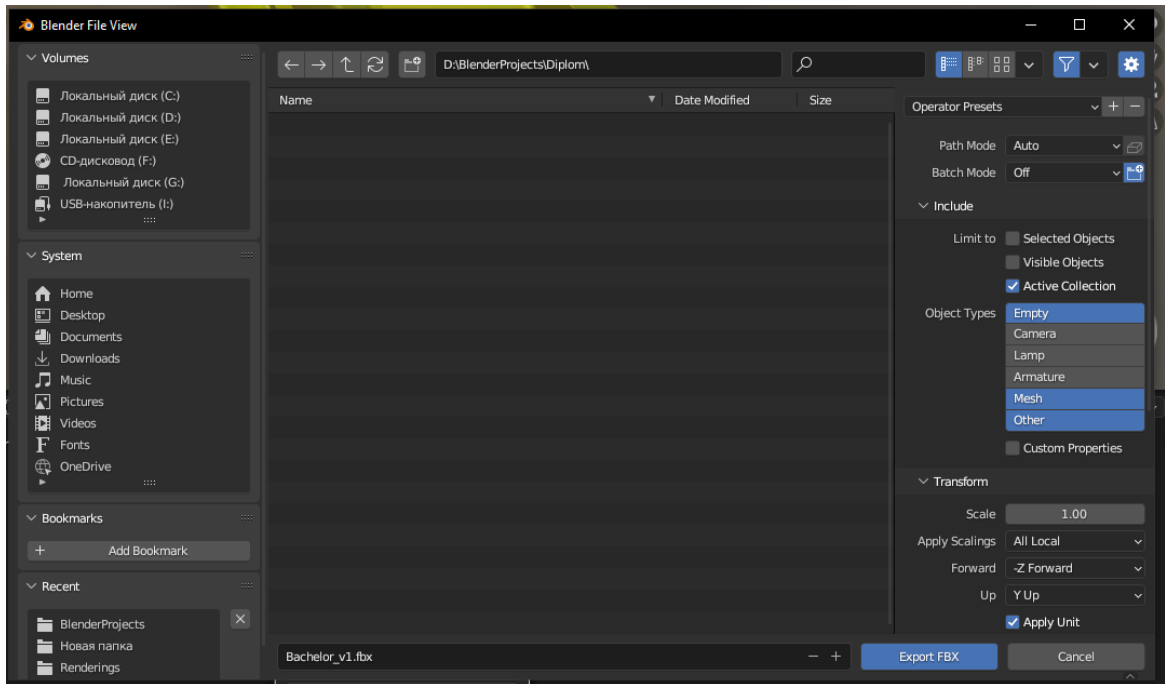


Рисунок 3.15 – Експортування моделі

3.2 Робота в русійному середовищі Unity

Наступним етапом переходимо до симуляції та демонстрації можливостей розробленої моделі.

Середовищем для симуляції було обрано ігровий русій Unity, як один із найпопулярніших та зручніших середовищ для досягнення цієї мети.

Починаємо роботу в юніті зі створення проекту за темплейтом 3д сцени та імпортування нашої 3D моделі у форматі fbx.

Після імпорту моделі слід перевірити її налаштування в інспекторі(функціональне вікно з параметрами в Editor).

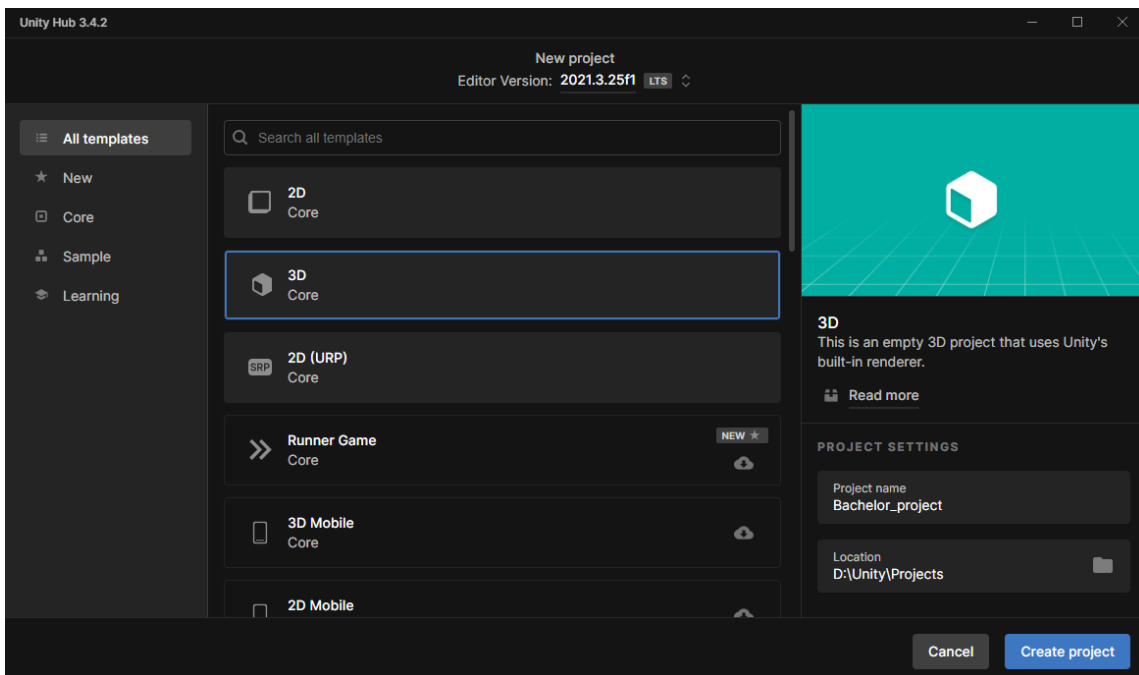


Рисунок 3.16 – Створення нового проекту

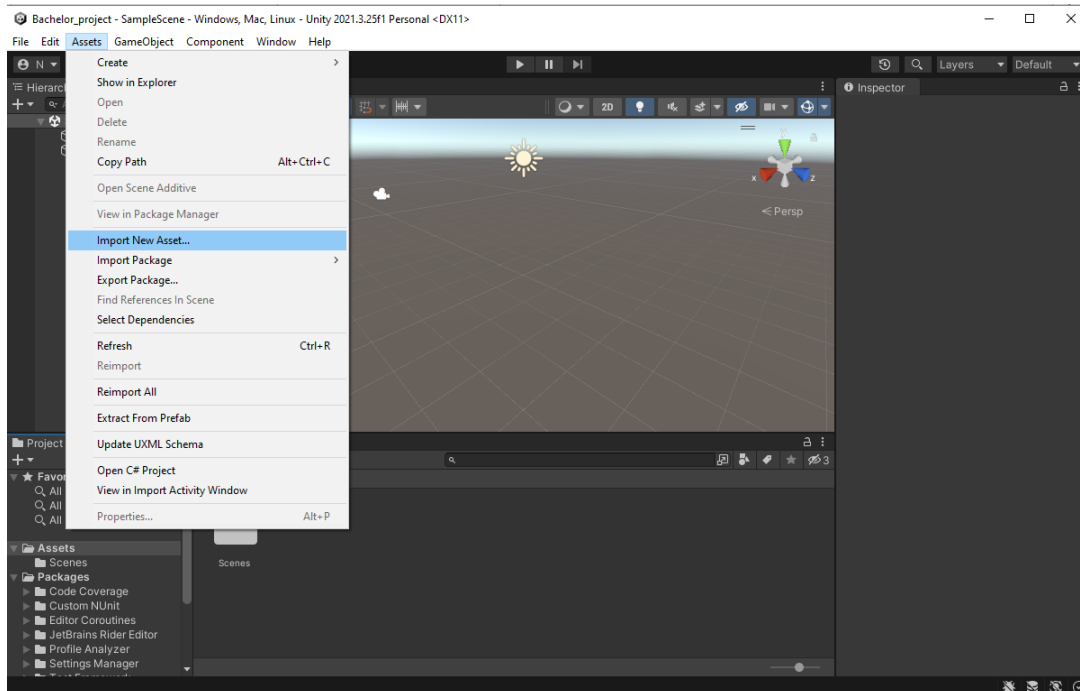


Рисунок 3.17 – Імпортування моделі

Переконавшись в коректності налаштувань, перекосямо модель на сцену та розпаковуємо Prefab. На рисунку 3.18 представлена імпортована модель робота-маніпулятора в середовище Unity.

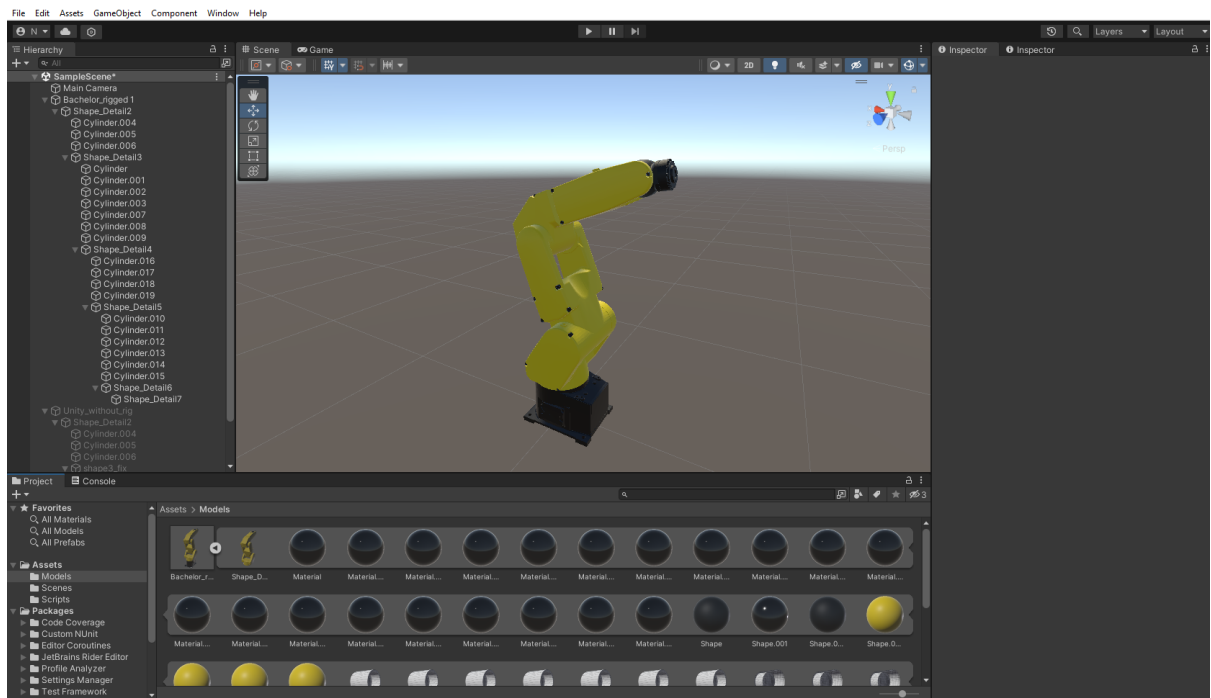


Рисунок 3.18 – Імпортована модель в середовище Unity

У вікні асетів можна побачити матеріали, що містяться на елементах моделі.

Наступним етапом було створено дві камери для того, щоб була можливість перемикатись між ними під час симуляції для більш зручного огляду моделі під час роботи. Для того, щоб перемикання між камерами було можливе був написаний скрипт, що відповідає за цей процес і доданий до компонентів основної камери (рис. 3.19)

Важливим аспектом демонстрації є надання користувачу можливості власноруч контролювати модель робота-маніпулятора за допомогою вводу із клавіатури. Існує декілька варіантів реалізації процесу симуляції рухів моделі, серед яких використання математичного алгоритму інверсивної кінематики, ввід чітких команд на рух кожної осі та безпосередній контроль за кожною ланкою моделі за їх осями.

Систему симуляції роботи моделі було розроблено наступним чином, користувач контролює кожен ланку не залежно від інших за допомогою натискання відповідних клавіш клавіатури.

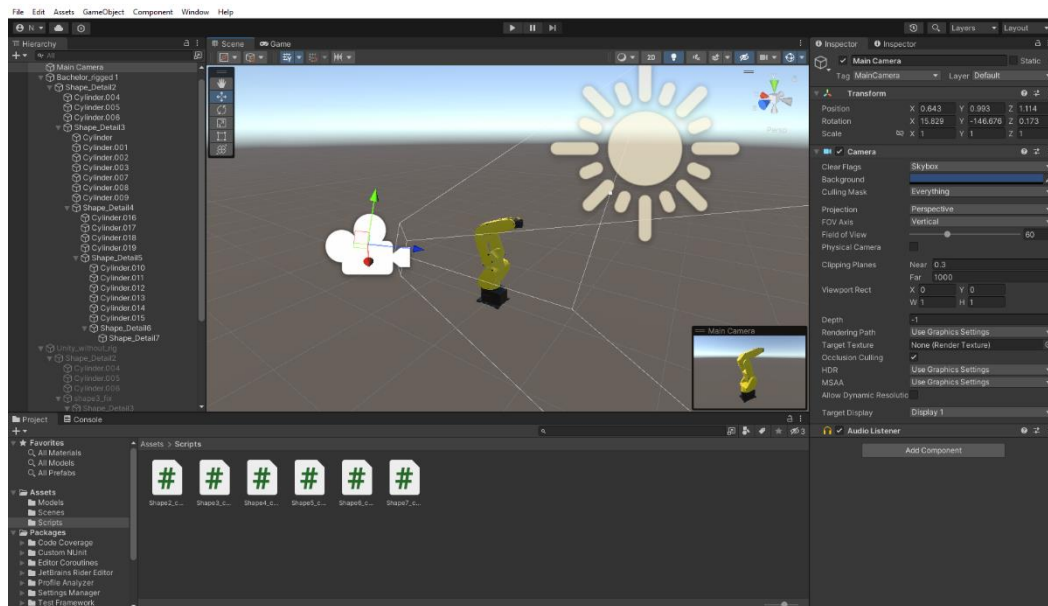


Рисунок 3.19 – Налаштування камери

Так як кожен рухомий елемент моделі має свої унікальні осі, напрям руху та ліміти, необхідно написати скрипт, що буде контролювати кожен ланку моделі. Обмеження переміщень рухомих частин здійснюється за допомогою використання функції `Mathf.Clamp()`, що обмежує значення між мінімальним та максимальним і повертає його.

На рисунках 3.20 – 3.21 відображено скрипти для маніпулювання деталями.

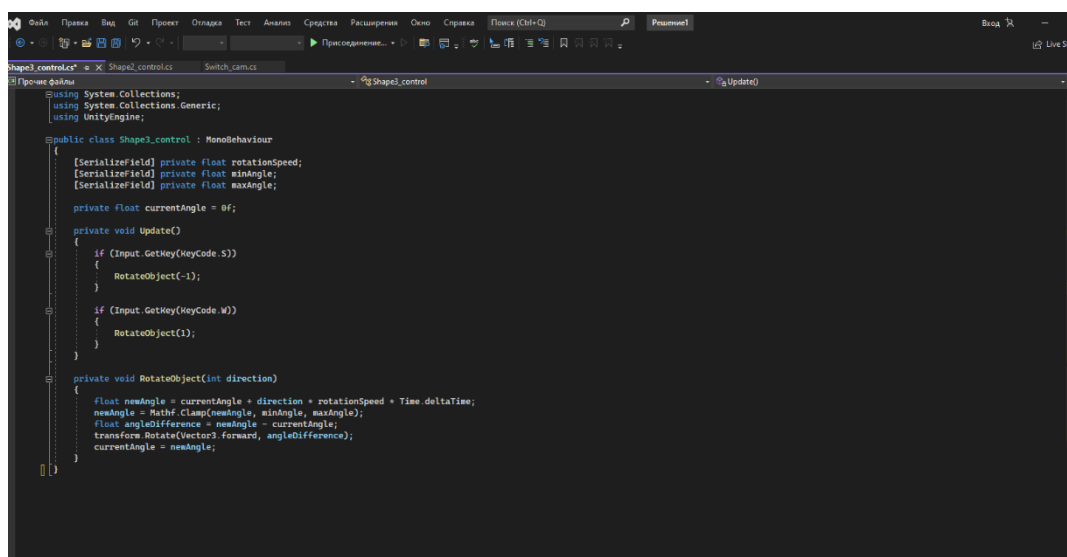


Рисунок 3.20 – C# скрипт для деталі з поступальним рухом

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UIElements;

public class Shape2_control : MonoBehaviour
{
    [SerializeField] private float speed;
    // Start is called before the first frame update
    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKey(KeyCode.A))
        {
            //float objectY = transform.rotation.y - speed * Time.deltaTime;
            Vector3 rottarget = new Vector3(0, -speed * Time.deltaTime, 0);
            //transform.rotation = Quaternion.Euler(0,objectY,0);
            transform.Rotate(rottarget);
        }
        else if (Input.GetKey(KeyCode.D))
        {
            Vector3 rottarget = new Vector3(0, speed * Time.deltaTime, 0);
            transform.Rotate(rottarget);
        }
    }
}

```

Рисунок 3.21 – скрипт мовою C# для деталі з обертальним рухом

Також під час написання були розглянуті креслення моделі, щоб зрозуміти які конструктивні обмеження мають рухомі елементи цієї моделі, таким чином, обмеження симуляції повністю відповідають конструктивним обмеженням цієї моделі згідно її характеристик та креслень(рис. 3.22 – 3.23).

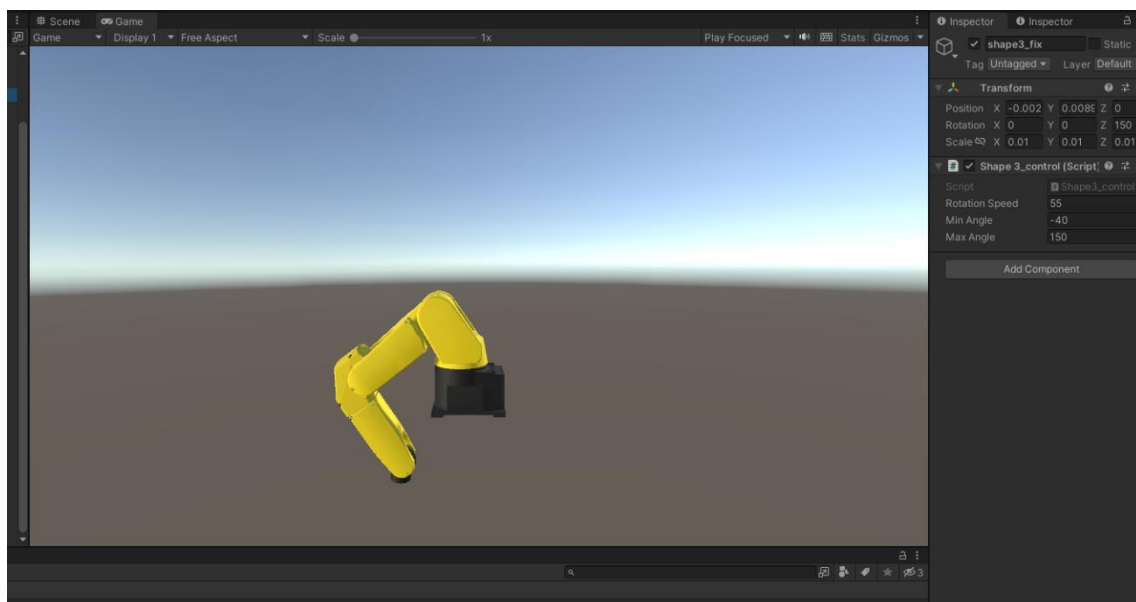


Рисунок 3.22 – Максимальний кут другої ланки

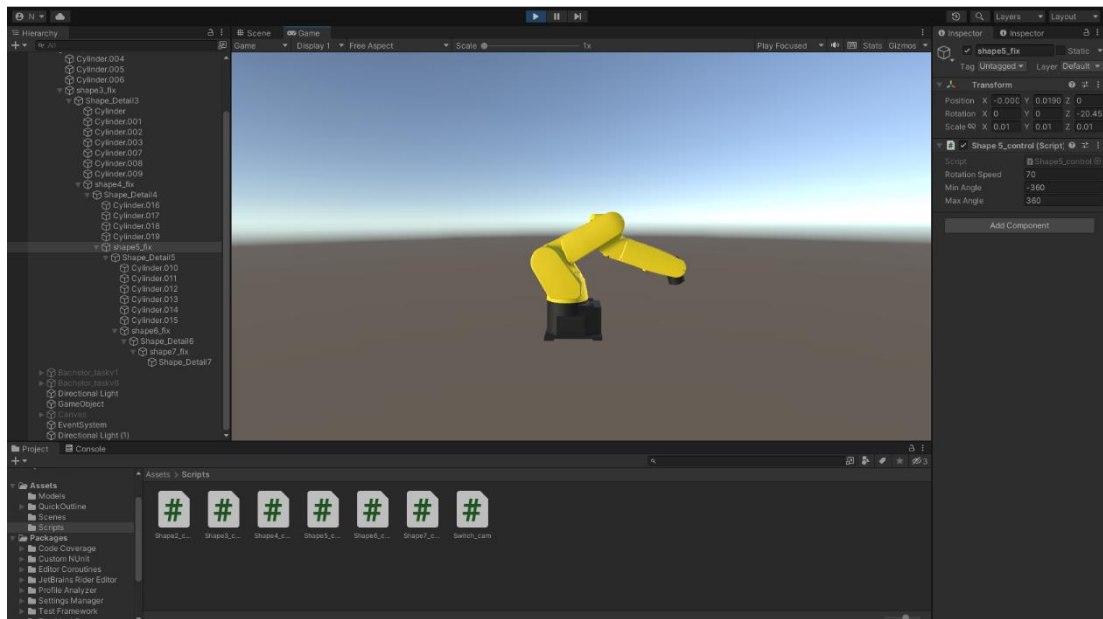


Рисунок 3.23 – Мінімальний кут другої ланки

Цільова операційна система для проведення симуляції була обрана Windows, як ода з найпоширеніших та найдоступніших.

Після того, як були написані усі скрипти, налаштовані камери та світло, заключним етапом була зроблена компіляція проекту (рис. 3.24) та запуск додатку (рис.3.25).

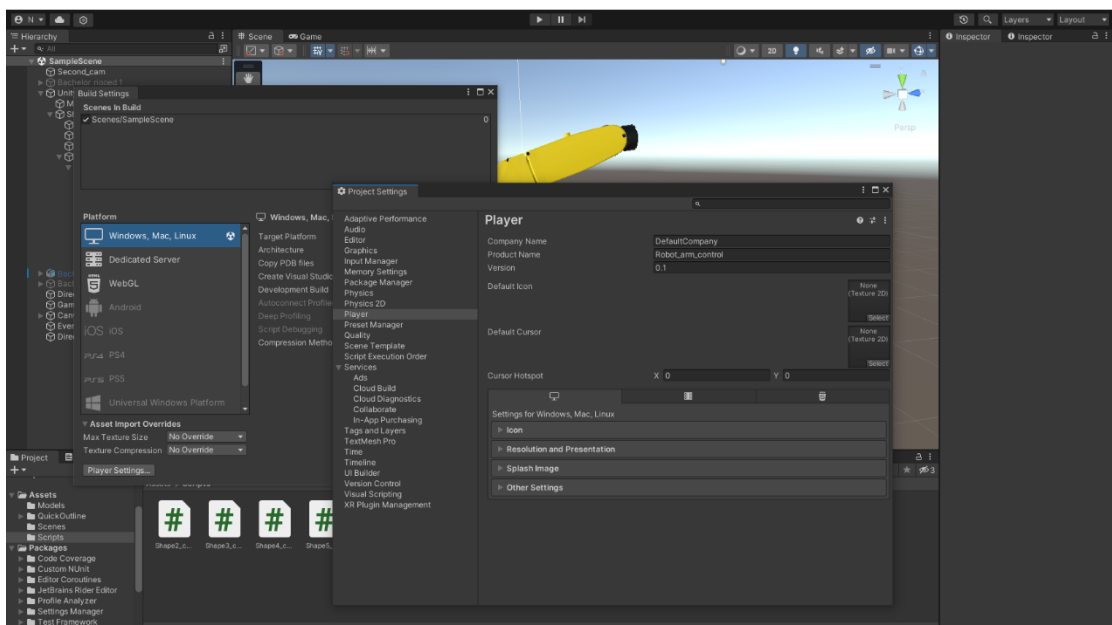


Рисунок 3.24 – Компіляція додатку



Рисунок 3.25 – Скопільований проект

Головна мета використання додатку – вивчення рухів прототипу робота-маніпулятора, його можливостей та зон досяжності. Для цього в додатку треба передбачити можливість керування моделі по даним сторонніх розрахунків кінематичних параметрів прототипу. Простіше за все передбачити можливість автоматичного виконання рухів моделі по файлах з числовими даними для переміщення ланок моделі по результатам таких розрахунків. Ці файли можна заздалегідь підготувати в сторонньому додатку користувача, який реалізує математичну модель прототипу.

В додатку зроблено можливість читати числові дані з файлів по натисканню клавіш (скрипт-обробник). Тобто, при кожному натисканні на клавішу скрипт буде зчитувати з текстового файлу значення переміщення або обертання навколо вісі певних ланок моделі робота-маніпулятора.

Користувач може обирати який текстовий файл буде використовуватися – вбудований в додаток чи власний, який отримано по результатах розрахунків кінематики. Створено інтерфейс для обирання файлів з числовими даними (рис. 3.26).

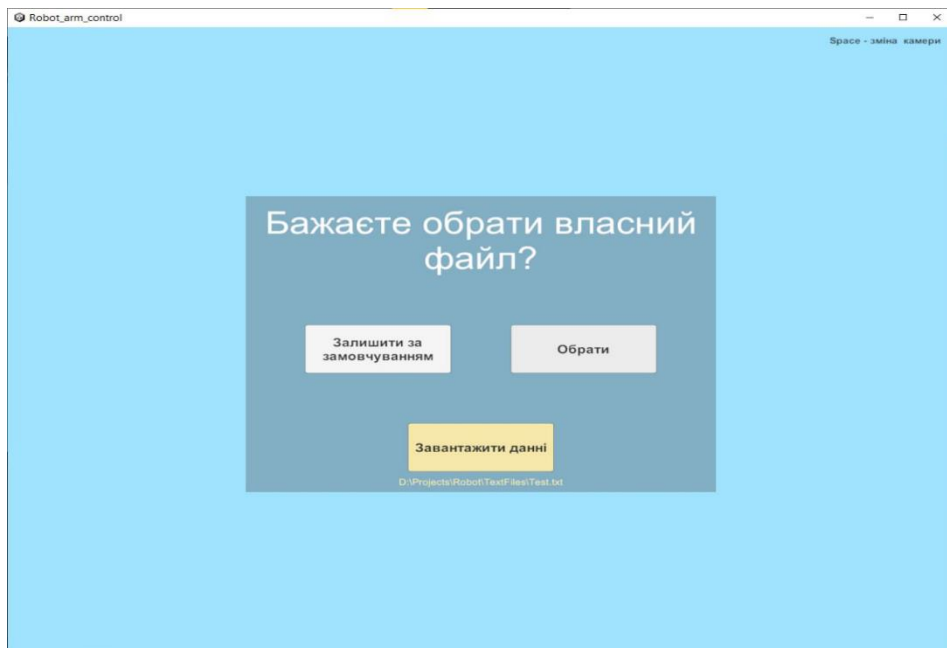


Рисунок 3.26 – Вікно вибору

Наступним кроком стала розробка скриптів.

Був розроблений скрипт LoadFile.cs за допомогою якого із текстових файлів будуть зчитуватися значення. Приклад методу зчитування дефолтного файлу показаний на рисунку 3.27.

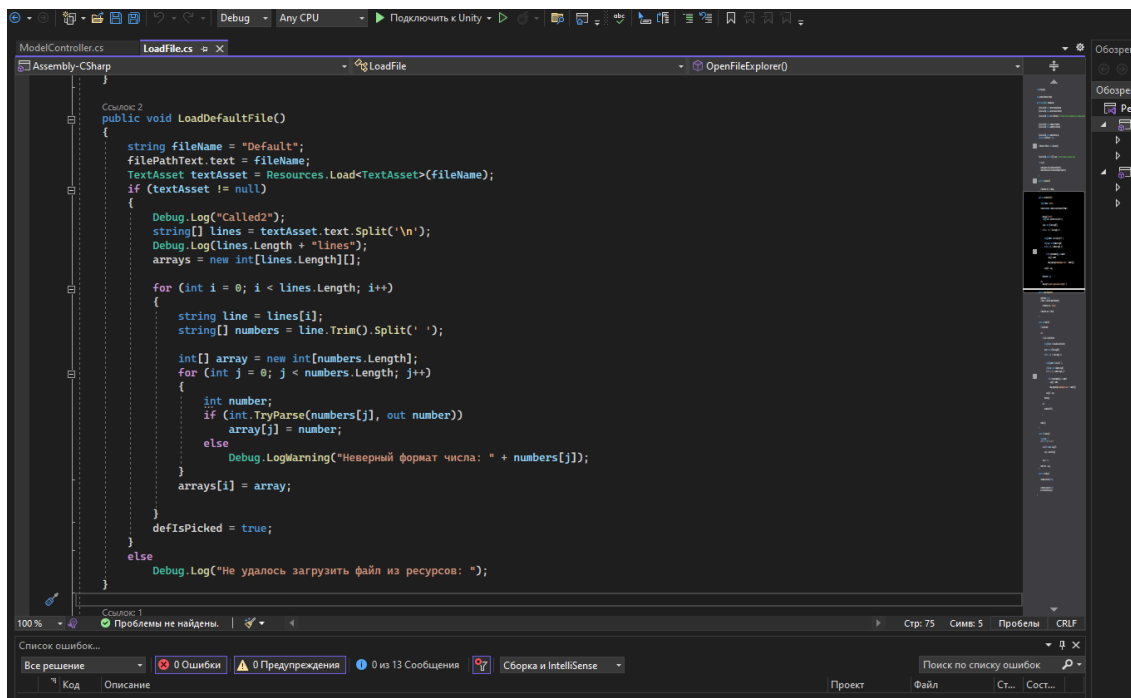


Рисунок 3.27 – Зчитування вбудованого файлу

Після цього був зроблений скрипт, який вже дозволяв керувати самим роботом при натисканні цифр на клавіатурі (рис. 3.28).

```

// Update is called once per frame
void Update()
{
    if (Input.GetKeyDown(KeyCode.Alpha1))
    {
        MoveLanka(0);
    }
    else if (Input.GetKeyDown(KeyCode.Alpha2))
    {
        MoveLanka(1);
    }
    else if (Input.GetKeyDown(KeyCode.Alpha3))
    {
        MoveLanka(2);
    }
    else if (Input.GetKeyDown(KeyCode.Alpha4))
    {
        MoveLanka(3);
    }
    else if (Input.GetKeyDown(KeyCode.Alpha5))
    {
        MoveLanka(4);
    }
    else if (Input.GetKeyDown(KeyCode.Alpha6))
    {
        MoveLanka(5);
    }
}

Ссылка 1
public void SetCoordinates(int[][] arrays)
{
    coordinates = arrays;
    for (int i = 0; i < arrays.Length; i++)
    {
        lankaCurrentCoordIndex[i] = 0;
    }
}

```

Рисунок 3.28 – Скрипт для руху ланок

Заключним етапом було проведення тестування додатку на наявність помилок в русі робота-маніпулятора та некоректного відображення деталей та тексту.

3.3 Тестування

Завершальним етапом проекту є тестування збірки додатку та перевірка його працездатності. Для цього запускаємо застосунок з .exe файлу на комп'ютері з операційною системою Windows. Застосунок завантажився без помилок. Далі тестуємо систему керування промисловим роботом-маніпулятором.

Підказки для керування роботом відображаються у верхньому лівому куті екрану. Для керування першою ланкою моделі відбувається за допомогою натискань клавіш A,D, для обертання навколо осі У вправо та вліво відповідно.

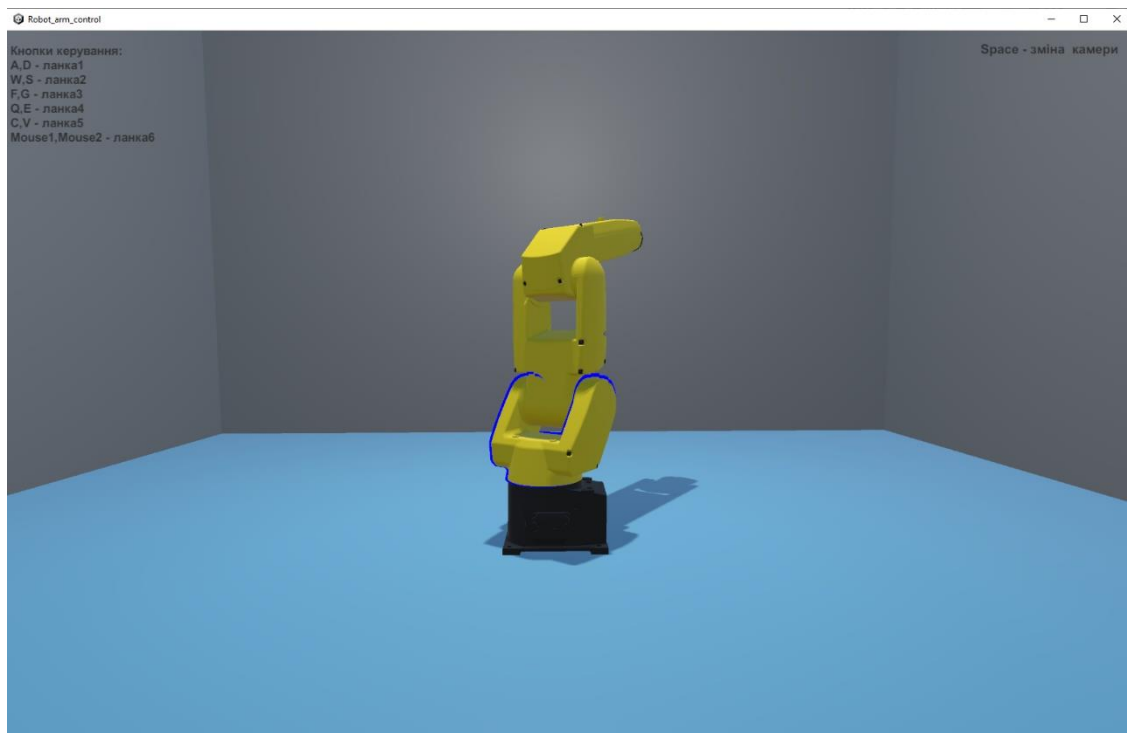


Рисунок 3.24 – Демонстрація обертання першої ланки

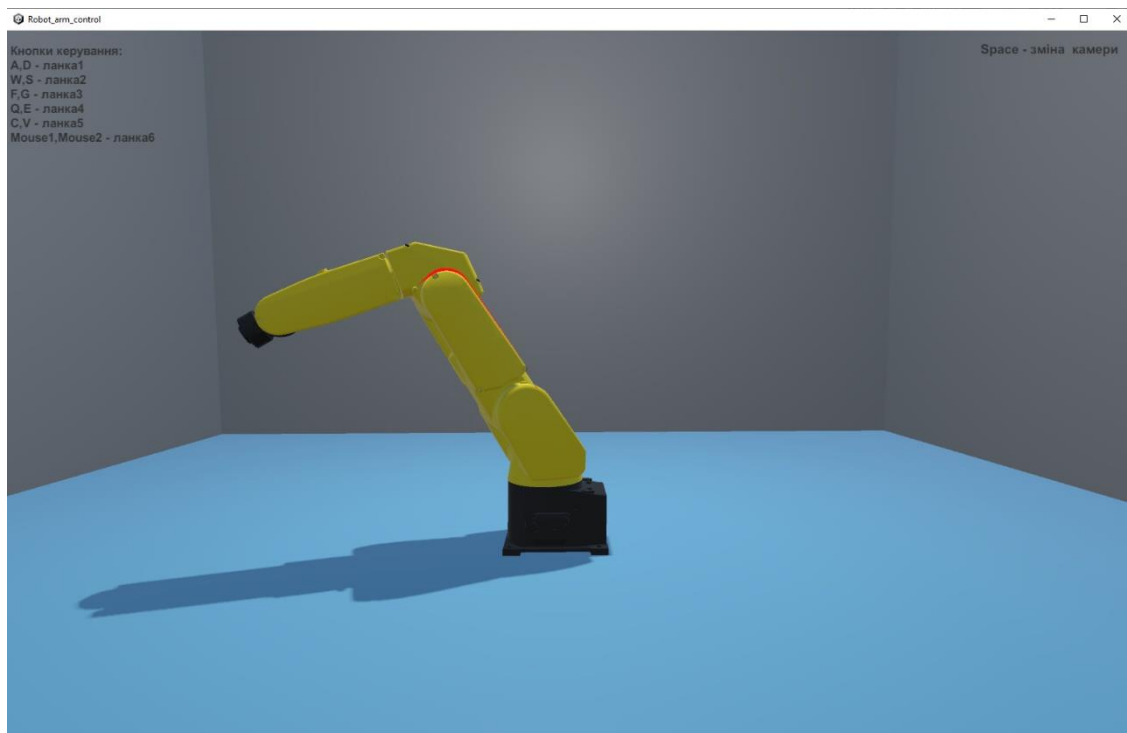


Рисунок 3.25 – Демонстрація роботи другої ланки

Наступна ланка маніпулятора відповідає за згинання руки робота за віссю Z. Згинання та розгинання виконується за клавішами W,S відповідно. Ліміти були встановлені на $\min = -40$ та $\max = 150$.

Наступною іде додаткова ланка згинання для більшого радіусу робочого простору моделі. Вона керується за допомогою клавіш F,G відповідно, та має ліміти $\min = -140$, $\max = 4$.



Рисунок 3.26 – Демонстрація роботи третьої ланки



Рисунок 3.27 – Демонстрація роботи четвертої ланки

Для виконання захоплення об'єктів під різними кутами служить четверта ланка. Її керування відбувається з використанням клавіш Q та E відповідно. Лімітів ця ланка не має.



Рисунок 3.28 – Демонстрація роботи п'ятої ланки



Рисунок 3.29 – Демонстрація роботи шостої ланки

Наступна п'ята ланка дозволяє досягати об'єкти для збірки під різними кутами. Керування виконується клавішами C V. Має ліміти $\min = -100$, $\max = 100$.

Остання частина моделі відповідає за безпосередньо захват деталей та розміщення їх під необхідним кутом. Контроль обертання відбувається за допомогою клавіш миші. Лімітів для обертання немає.

Після того, як проект був успішно протестований потрібно завантажити готовий білд на репозиторій git.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було проведено дослідження актуальності розробки (проекту), виконане вивчення предметної області. В якості прототипу для розробки тривимірної параметричної моделі обрано зразок промислового робота-маніпулятора Fanuc Lr mate 200iC. Розглянуті існуючі тривимірні моделі цього робота-маніпулятора були проаналізовані та порівняні між собою. Були знайдені риси та можливості, які треба реалізувати в розроблюваній моделі. Таким чином, були встановлені вимоги до продукту. На основі цього була визначена мета проекту і сформульовані задачі для її досягнення.

Розроблений календарний план робіт над проектом, здійснений розподіл обов'язків та визначені зони відповідальності виконавців. Проведений аналіз ризиків на різних етапах виконання.

Під час моделювання розробки проекту побудована діаграма варіантів використання (Use Case) і здійснене декомпозиційне моделювання інформаційної системи в нотації IDEF0. Отримані діаграми допомогли формалізувати всі процеси проекту та визначити користувацькі та функціональні вимоги.

Для створення програмного продукту обрано новітні програмні інструменти, мова написання скриптів та середовище розробки моделі. Отримана параметрична модель повністю відповідає вимогам до проекту та має можливість бути керованою за допомогою програмних команд. Використання скриптів допомогло організувати обмеження на рух окремих ланок моделі, виявити зони досяжності маніпулятора.

На заключному етапі розробки додаток оснащений можливістю програмувати рухи моделі за допомогою використання зовнішніх файлів з числовими значеннями для рухів окремих ланок. Натискаючи клавіші в додатку користувач вивчає рухливість ланок моделі та кінематику робота-прототипа як такого.

Тестування отриманої моделі показало повну роботоспроможність моделі, керованість та стійкість до помилкових введень користувача.

Результатом роботи над даним проектом є функціональна тривимірна модель робота-маніпулятора, яка виконана у вигляді самостійного застосунку для симуляції різних виробничих умов, вивчення керування промисловими роботами в цілому.

СПИСОК ВИКОРИСТАННИХ ДЖЕРЕЛ

1. Манипулятор [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://studfile.net/preview/16471652/>.
2. Comparing the LR Mate 200ic and 200id [Електронний ресурс] – Режим доступу до ресурсу: <https://robotsdoneright.com/Articles/comparison-lr-mate-200ic-and-200id.html>.
3. Використання уніфікованої мови моделювання (UML) [Електронний ресурс] – Режим доступу до ресурсу: http://iwanoff.inf.ua/oop_kn/LabTraining05.html.
4. Нотація IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%9A%D0%BE%D0%BD%D0%B4%D1%96%D1%83%D1%81%20%20%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D0%B2%D0%B0/page9.html.
5. Промислові роботи-маніпулятори [Електронний ресурс] – Режим доступу до ресурсу: <https://prorobotov.org/blog/stati/promyshlennye-roboty-manipulyatory/>.
6. Промислові роботи. Класифікація промислових роботів [Електронний ресурс] – Режим доступу до ресурсу: https://kemppi.in.ua/articles/promochlenie_roboti/
7. 3D-моделювання і візуалізація [Електронний ресурс] – Режим доступу до ресурсу: <https://koloro.ua/3d-modelirovanie-i-vizualizaciya.html>.
8. Діаграми IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/1741999/page:4/>.
9. What is robotic manipulator? [Електронний ресурс] – Режим доступу до ресурсу: <https://reachrobotics.com/blog/what-is-a-robotic-manipulator/>.
10. FANUC LR Mate 200iC, kinematic diagram [Електронний ресурс] – Режим доступу до ресурсу: <https://www.researchgate.net/figure/FANUC-LR->

Mate-200iC-robot-kinematic-diagram-24-TABLE-I-D-H-PARAMETER-OF-THE-FANUC-LR_fig1_269270372.

11. Robot Manipulators. Position, Orientation and Coordinate Transformations [Електронний ресурс] – Режим доступу до ресурсу: https://www.maplesoft.com/content/EngineeringFundamentals/13/MapleDocument_13/Position,%20Orientation%20and%20Coordinate%20Transformations.pdf.

12. Melek W. W. Robot Manipulators: Kinematics, Dynamics, and Control" [Електронний ресурс] / Режим доступу до ресурсу: https://www.maplesoft.com/content/EngineeringFundamentals/14/MapleDocument_14/Forward%20Kinematics.pdf.

13. Fanuc Robotic Arm 3D model [Електронний ресурс] – Режим доступу до ресурсу: <https://www.turbosquid.com/ru/3d-models/3d-model-fanuc-robot-arm-1247948#>.

14. Налесная Я. А. Розробка промислового робота-маніпулятора для спеціальних цілей : дис. канд. ек. наук / Налесная Я. А., 2017. – 8 с.

15. DIY Robotic Arm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.instructables.com/DIY-Robotic-Arm/>.

16. Перспективи використання промислових роботів-маніпуляторів [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://vektorus.ru/blog/promyshlennye-roboty-manipulyatory.html>.

17. FANUC LR MATE 200iC. Diagrams [Електронний ресурс] – Режим доступу до ресурсу: <https://www.eurobots.com.ua/fanuc-robots-lr-mate-200ic-p179-uk.html>.

18. Обзоры роботів-маніпуляторів [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/companies/top3dshop/articles/373649/>.

19. Robotic Manipulation [Електронний ресурс] – Режим доступу до ресурсу: <https://robotics.leeds.ac.uk/research/ai-for-robotics/robotic-manipulation/>.

20. Benedict O. What are manipulator robots? Overview of types and applications [Електронний ресурс] / O'Neill Benedict. – 2022. – Режим доступу до ресурсу: <https://www.wevolver.com/article/what-are-manipulator-robots-overview-of-types-and-applications>.

21. WORKING OF ROBOTIC ARM [Електронний ресурс] – Режим доступу до ресурсу: <https://www.universal-robots.com/in/blog/working-of-robotic-arm/>.

22. Github репозітарій [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/1f0rward1/Diplom>

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ **на розробку 3D-моделі** **«Параметрична 3D-модель промислового робота-маніпулятора»**

ПОГОДЖЕНО:

Старший викладач кафедри
«Інформаційні технології»

_____ Кузнецов Е. Г.

Студент групи ІТ-91

_____ Веретельніков М.В.

Суми 2023

1 Призначення й мета створення додатку

1.1 Призначення додатку

Додаток призначений для демонстрації рушійних можливостей та можливості керування роботом-маніпулятором.

1.2 Мета створення додатку

Мета проекту – створення продукту 3D-моделі робота-маніпулятора, параметрично візуалізувати цю модель за допомогою програми Blender та зробити додаток, в якому користувач зможе керувати роботом.

1.3 Цільова аудиторія

Обмеження цільової аудиторії відсутні.

2 Вимоги до інформаційної системи

2.1 Вимоги до інформаційної системи в цілому

2.1.1 Вимоги до структури й функціонування

Результат проекту повинен бути представлений у вигляді файлу додатку на платформі Unity, де користувач зможе ознайомитись з моделлю робота-маніпулятора та матиме змогу керувати ним.

2.1.2 Вимоги до персоналу

Для поліпшення роботи, додавання нових функцій до додатку тощо, обслуговуючий персонал повинен мати навички роботи з будь-яким додатком для розробки 3D-моделей(бажано Blender) та графічною платформою Unity.

2.1.3 Вимоги до збереження інформації

Уся інформація повина зберігатися на персональному комп'ютері.

2.1.4 Вимоги до лінгвістичного забезпечення

Весь текст у додатку має бути виконаний українською або англійською мовами.

2.1.5 Вимоги до розмежування доступу

Розроблювана модель має бути загальнодоступною у мережі Інтернет. Права доступу розмежовані за групами користувачів: розробники (адміністратори), користувачі. Адміністратори мають необмежений доступ до даних з правами перегляду контенту та користування. Доступ до матеріалів надається за спеціальним репозиторієм Git-Hub.

Користувач додатку може тільки використовувати готову версію продукту та надсилати повідомлення за допомогою форми зворотного зв'язку.

3 Склад і зміст робіт зі створення додатку

Докладний опис етапів роботи зі створення додатку для робота-маніпулятора наведено в таблиці А.3.

Таблиця А.3 – Етапи створення робота-маніпулятора

№	Склад і зміст робіт	Строк розробки
1	Моделювання 3D моделі	20 днів
4	Налаштування освітлення 3D моделі	3 дні
5	Рендерінг 3D моделі	1 дні
6	Створення базових механік руху	15 днів
8	Створення інтерактивного меню управління додатком	2 дні
9	Налаштування матеріалів всередині процесу додатку	1 дні
10	Перевірка працездатності	2 дні
11	Написання супровідної документації	6 днів
	Загальна тривалість робіт	50 днів

ДОДАТОК Б

Деталізація мети проекту методом SMART. Продуктом дипломного проекту є параметрична 3D-модель промислового робота-маніпулятора.

Результати деталізації методом SMART розміщені у табл. Б.1.

Таблиця Б.1 – Деталізація мети за технологією SMART

Specific (конкретна)	Моделювання функціонуючого робота-маніпулятора FANUC та систему керування.
Measurable (вимірювана)	Результат допоможе скоротити час та ресурси на виробництві.
Achievable (досяжна)	Проект розробляється за допомогою програмного забезпечення 3ds Max та Unity 3D.
Relevant (реалістична)	Отримання досвіду з роботою з додатком для моделювання.
Time-framed (обмежена у часі)	Проект виконується враховуючи встановлені часові обмеження.

Планування змісту структури робіт. Для планування змісту структури робіт буде використовуватись WBS (Work Breakdown Structure – Ієрархічна структура робіт) – це ієрархічна діаграма, яку ілюструють як багаторівневу схему завдань, де масштабна мета розбивається на етапи, а потім на дрібніші задачі, які є більш зрозумілими та досяжними. Поділ здійснюється до моменту, поки всі роботи для досягнення мети не буде враховано.

Ієрархічна структура робіт має три основні цілі:

- Опис складу робота у завданнях
- Планування роботи
- Оцінка вартості кожного завдання

На рисунку Б.1 представлено WBS проекту щодо візуалізації 3D моделі.

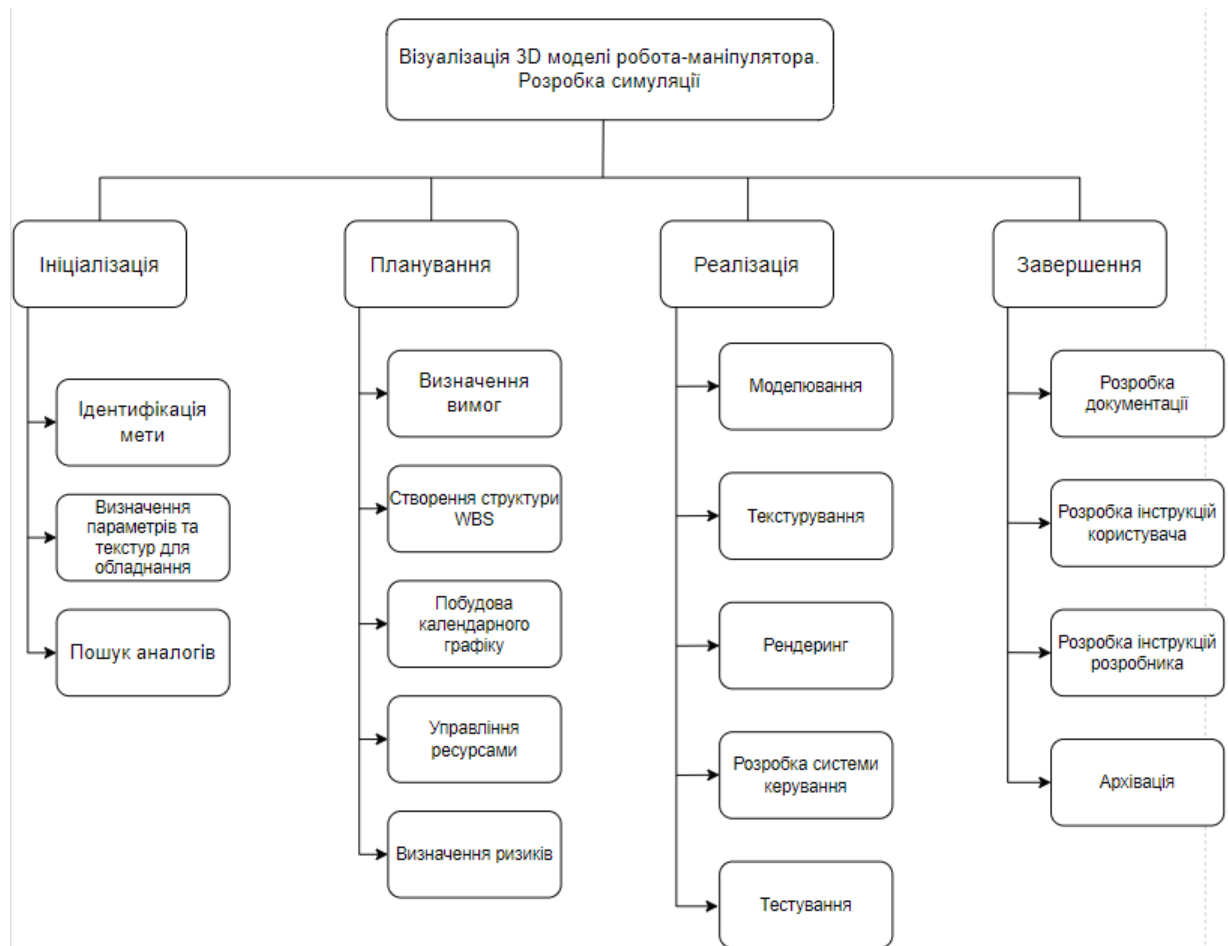


Рисунок Б.1 – WBS-структура робіт проекту

Планування структури організації, для впровадження готового проекту (OBS). Після декомпозиції процесів наступним етапом виконується OBS(Organizational Breakdown Structure – організаційна структура виконавців) – це графічна структура, які відображає учасників або відповідальних осіб, які приймають участь у виконанні елементарної роботи, що зазначена у ієрархічній структурі робіт, для реалізації проекту. Кожна елементарна робота може розглядатися як окремий проект.

На рисунку Б.2 представлено організаційну структуру планування проекту. Список виконавців, що функціонують в проекті описано в таблиці Б.2.

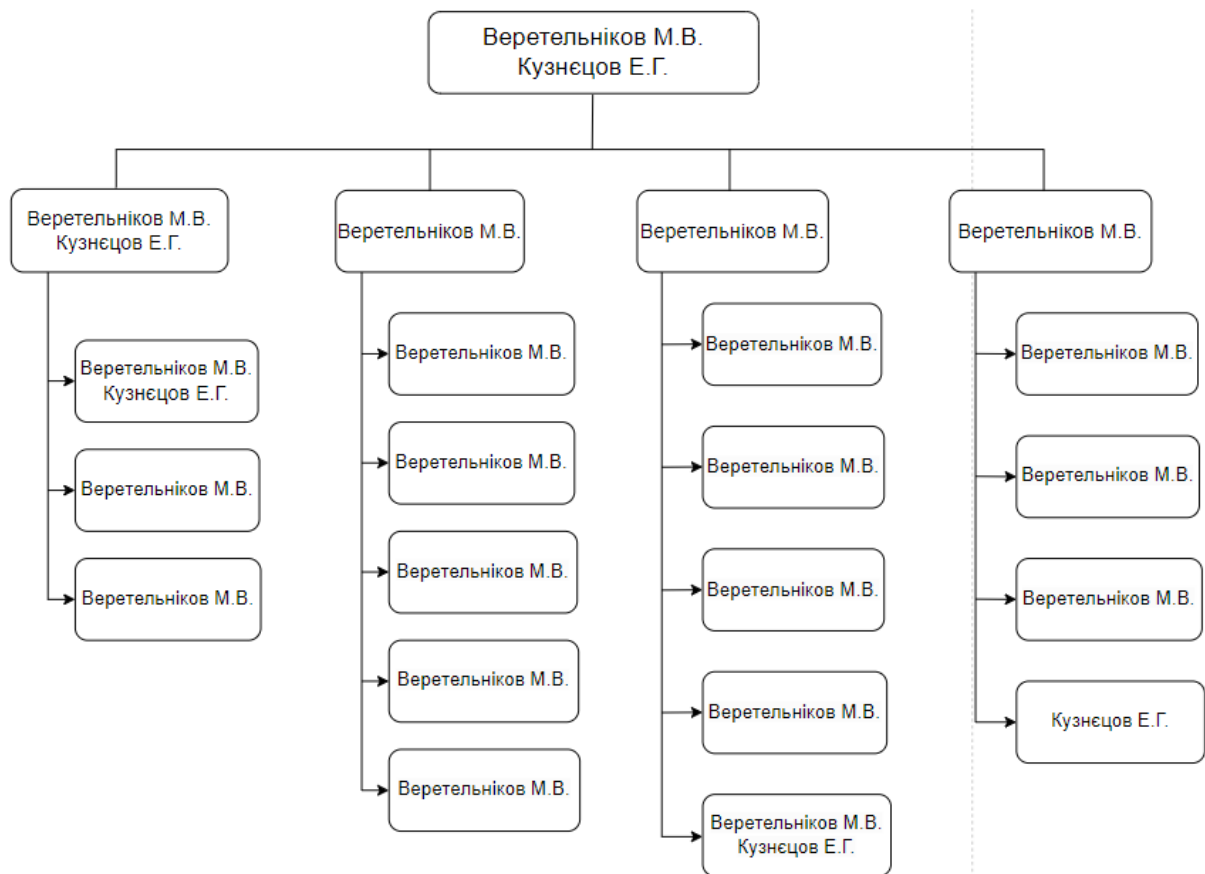


Рисунок Б.2 – OBS-структура робіт проект

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Керівник проекту	Кузнецов Е.Г.	Відповідає за виконання термінів, розподіл ресурсів, та завдань між учасниками. Виконує збір та аналіз даних.
3Д-моделлер	Веретельніков М.В.	Створення 3Д моделі робота-маніпулятора
Програміст	Веретельніков М.В.	Створення симуляції керування роботом
Тестувальник	Веретельніков М.В. Кузнецов Е.Г.	Знаходження помилок

Діаграма Ганта

Побудова календарного графіку (діаграми Ганта) є одним з важливих етапів планування проекту, що виглядає як розклад виконання робіт з реальним розподілом дат. Завдяки йому можна отримати достовірне уявлення про тривалість процесів з обмеженнями у ресурсах, урахуванням вихідних днів та свят.

Календарний графік проекту представлено на рисунках Б.3-Б.4.

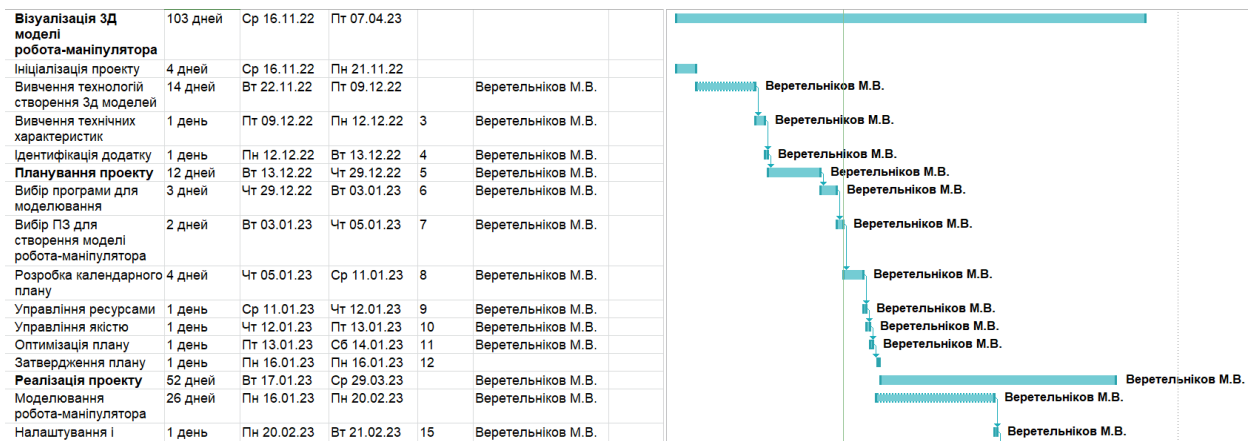


Рисунок Б.3 – Діаграма Ганта. Частина 1

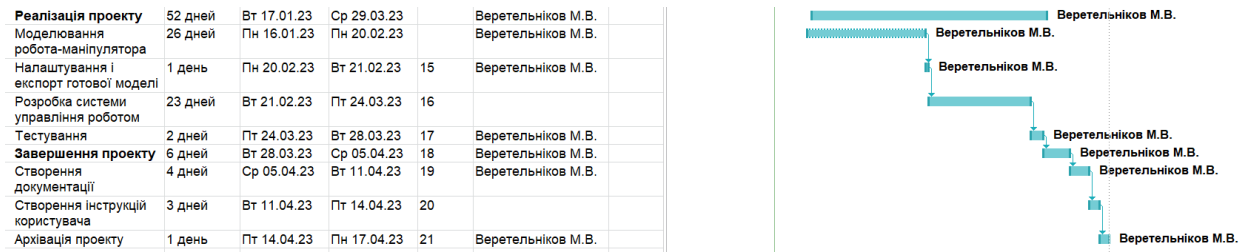


Рисунок Б.4 – Діаграма Ганта. Частина 2

Аналіз ризиків

Аналіз ризиків є невід'ємною частиною будь-якого проекту. Він дозволяє ідентифікувати потенційні загрози, які можуть вплинути на успішне виконання проекту, а також розробляти стратегії та плани дій для їх управління. Під час аналізу ризиків звертається увага на можливі проблеми, помилки або несприятливі обставини, які можуть виникнути протягом проекту. Це можуть бути технічні проблеми, зміни вимог, недоступність ресурсів, затримки у графіку, фінансові ризики тощо.

Головною метою аналізу ризиків є зниження впливу негативних подій на проект та забезпечення успішного завершення. Шляхом ретельного виявлення й категоризації ризиків можна розробити план дій, який дозволить передбачити можливі наслідки та підготуватися до них заздалегідь.

Аналіз ризиків також надає можливість визначити потенційні можливості, які можуть позитивно вплинути на проект. Це можуть бути нові технології, сприятлива ринкова ситуація, партнерські угоди тощо. Виявлення таких можливостей дозволяє використовувати їх на користь проекту та забезпечити його успішний розвиток.

Аналіз ризиків є важливим етапом управління проектом, який допомагає зменшити невизначеність та підвищити шанси на досягнення поставлених цілей. Він вимагає систематичного підходу, уваги до деталей та активної участі всіх учасників проекту.

У таблиці Б.3 представлено шкалу для класифікації ризиків за величиною впливу на проект та ймовірністю виникнення.

Таблиця Б.3 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу.

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Планування реагування на ризики є важливою складовою ефективного управління проектом. Його мета полягає в зниженні негативного впливу ризиків на проект шляхом активної підготовки та прийняття відповідних заходів.

Перш за все, необхідно ідентифікувати всі можливі ризики, які можуть виникнути протягом проекту. Це включає аналіз потенційних загроз, оцінку їх впливу та ймовірності виникнення. На основі цієї інформації складається список ризиків, які потребують уваги. Після ідентифікації ризиків, необхідно оцінити їх важливість та пріоритетність. Для цього використовуються різні методи, такі як матриця ймовірності-впливу або метод аналізу чутливості. Це допомагає визначити ті ризики, які мають найбільший потенціал негативного впливу на проект. Після оцінки ризиків важливо розробити конкретні стратегії та плани дій для кожного ризику. Це означає визначення конкретних кроків, які будуть вжиті для зменшення впливу ризику або навіть його усунення. Це можуть бути запобіжні заходи, плани резервування ресурсів, переговори зі зацікавленими сторонами або зміни в графіку проекту. Після розробки планів реагування на ризики важливо впровадити їх у практику. Це означає, що весь проектний колектив повинен бути ознайомлений з планами реагування, а відповідальність за виконання відповідних заходів повинна бути чітко визначена. Крім того, важливо встановити моніторинг та контроль за ризиками, щоб вчасно виявляти зміни в ситуації та вживати необхідних заходів.

Планування реагування на ризики є проактивним підходом до управління проектом. Воно дозволяє знизити негативний вплив ризиків на

проект, забезпечити більш ефективно виконання завдань та досягнення поставлених цілей.

Оцінювання виконується за показниками, що описані в таблиці Б.3. У результаті планування реагування було отримано матрицю ймовірності виникнення ризиків та впливу ризику, що зображена на рисунку Б.4. Зеленим кольором на матриці позначають прийнятні ризики, жовтим – виправдані, а червоним – недопустимі.

		Probability		
		1	2	3
Impact	1			RS_2
	2	RS_8	RS_1, RS_6, RS_9	
	3		RS_5, RS_10	RS_3, RS_4, RS_7

Рисунок Б.4. – Матриця ймовірності

Класифікація ризиків за рівнем, відповідно до отриманого значення індексу, представлена у таблиці Б.4. У таблиці Б.5 описано ризики та стратегії реагування на кожен з них.

Таблиця Б.4 – Шкала оцінювання за рівнем ризику.

№	Назва	Межі	Ризики, які входять
1	Прийнятні	$1 < R < 2$	8,5,10
2	Виправдані	$3 < R < 4$	1,3,4,6,7,9
3	Недопустимі	$6 < R < 9$	2

Таблиця Б.5 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Нечітке завдання на розробку	Середня	Високий	6	<p>1.Ясно і однозначно обговорити із замовником усі види вимог.</p> <p>2.Скласти глосарій для запобігання розбіжностей у розумінні слів та термінів.</p> <p>3.Періодичний контроль замовником етапів роботи.</p>	Попередження	При виявленні невідповідностей деяких характеристик продукту заявленим вимогам потрібно уважно та чітко окреслити те, що було виконано невірно та зробити правки.

Продовження таблиці Б.5 – Ризики та стратегії реагування

RS_2	Відкритий	Неоптимальний розподіл часу	Висока	Високий	9	Провести аналіз актуальності процесів та робіт. Звернути особливу увагу на розподіл часу. Правильно визначити пріоритети виконання робіт. Дотримуватись календарного плану.	Пом'якшення	Змінити порядок пріоритетів робіт. Знайти способи оптимізації роботи з вже існуючою розстановкою. Обговорити варіанти внесення поправок до термінів реалізації із замовником.
------	-----------	-----------------------------	--------	---------	---	---	-------------	---

Продовження таблиці Б.5 – Ризики та стратегії реагування

RS_3	Відкритий	Вибір не ефективної технології розробки 3д-додатку	Середня	Середній	4	1.Проаналізувати методи та засоби для виконання завдання. 2.Обрати зрозумілу та легку в використанні технологію розробки.	Пом'якшення	Виділити час та ресурси на пошук покращення обраної технології. Застосувати допоміжні ресурси.
RS_4	Відкритий	Неправильна оцінка в масштабі додатку	Низька	Середній	2	1.Провести детальний аналіз проекту. 2.Визначити основні етапи проекту, розподілити час на її виконання. 3.Проаналізувати масштаби проекту на основі додаткових джерел.	Пом'якшення	Переоцінка масштабів проекту. Перебудова стратегії реалізації проекту.

Продовження таблиці Б.5 – Ризики та стратегії реагування

RS_5	Відкритий	Помилки 3Д-моделювання	Середня	Середній	4	На етапі моделювання тісно співпрацювати із замовником та на певних етапах демонструвати поточні результати.	Пом'якшення	Здійснювати проміжний контроль результатів в ході виконання проекту.
RS_6	Відкритий	Збої в роботі програмного забезпечення	Низька	Середній	3	1. Підготувати резерв програмних засобів. 2. Залучити спеціаліста для усунення збоїв.	Попередження	Замінити програмне забезпечення.
RS_7	Відкритий	Відсутність резервних копій	Низька	Середній	2	1. Налаштувати автоматичне збереження даних. 2. Зберігати дані на різних носіях інформації.	Попередження	Робити копію даних після кожного виконаного етапу.

Продовження таблиці Б.5 – Ризики та стратегії реагування

RS_8	Відкритий	Невикористання моніторингу проекту	Середня	Низький	2	Здійснювати проміжний контроль результатів в ході виконання проекту.	Перенесення	Здійснювати моніторинг проєкту замовником. Надання проміжних результатів виконання проекту після кожного етапу.
RS_9	Відкритий	Виникнення проблем із програмним забезпеченням користувачів	Середня	Середній	4	Розробка проекту з урахуванням вимог до ПЗ користувачів проекту.	Прийняття	Модифікація проекту з урахуванням різних версій ПЗ, яке буде застосовуватись.
RS_10	Відкритий	Зміна вимог замовника в процесу розробки	Низька	Середній	3	Узгодити всі питання на початкових етапах, щоб мінімізувати кількість змін під час розробки.	Пом'якшення	Переоцінка проекту кожного разу, коли вимоги змінюються.

ДОДАТОК В

Програмний код

Shape2_control.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UIElements;

public class Shape2_control : MonoBehaviour
{
    [SerializeField] private float speed;
    private Outline outline;
    void Start()
    {
        outline = GetComponent<Outline>();
    }

    void Update()
    {
        if (Input.GetKey(KeyCode.A))
        {
            outline.enabled = true;
            Vector3 rottarget = new Vector3(0, -speed * Time.deltaTime, 0);
            transform.Rotate(rottarget);
        }
        else if (Input.GetKey(KeyCode.D))
        {
            outline.enabled = true;
            Vector3 rottarget = new Vector3(0, speed * Time.deltaTime, 0);
            transform.Rotate(rottarget);
        }
        else
        {
            if(outline.enabled)
                outline.enabled = false;
        }
    }
}

```

Shape3_control.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Shape3_control : MonoBehaviour
{
    [SerializeField] private Outline outline;
    [SerializeField] private float rotationSpeed;
    [SerializeField] private float minAngle;
    [SerializeField] private float maxAngle;

    private float currentAngle = 0f;

    void Start()
    {
    }

    private void Update()
    {
        if (Input.GetKey(KeyCode.S))
        {
            RotateObject(-1);
        }
        else if (Input.GetKey(KeyCode.W))
        {
            RotateObject(1);
        }
        else
        {
        }
    }
}

```

```

        if (outline.enabled)
            outline.enabled = false;
    }
}

private void RotateObject(int direction)
{
    outline.enabled = true;
    float newAngle = currentAngle + direction * rotationSpeed * Time.deltaTime;
    newAngle = Mathf.Clamp(newAngle, minAngle, maxAngle);
    float angleDifference = newAngle - currentAngle;
    transform.Rotate(Vector3.forward, angleDifference);
    currentAngle = newAngle;
}
}

```

Shape4_control.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Shape4_control : MonoBehaviour
{
    [SerializeField] private Outline outline;
    [SerializeField] private float rotationSpeed;
    [SerializeField] private float minAngle;
    [SerializeField] private float maxAngle;

    private float currentAngle = 0f;

    private void Update()
    {
        if (Input.GetKey(KeyCode.F))
        {
            RotateObject(-1);
        }

        else if (Input.GetKey(KeyCode.G))
        {
            RotateObject(1);
        }
        else
        {
            if (outline.enabled)
                outline.enabled = false;
        }
    }

    private void RotateObject(int direction)
    {
        outline.enabled = true;
        float newAngle = currentAngle + direction * rotationSpeed * Time.deltaTime;
        newAngle = Mathf.Clamp(newAngle, minAngle, maxAngle);
        float angleDifference = newAngle - currentAngle;
        transform.Rotate(Vector3.forward, angleDifference);
        currentAngle = newAngle;
    }
}

```

Shape5_control.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Shape5_control : MonoBehaviour
{
    [SerializeField] private Outline outline;
    [SerializeField] private float rotationSpeed;
    [SerializeField] private float minAngle;
    [SerializeField] private float maxAngle;

    private float currentAngle = 0f;

    private void Update()

```

```

{
    if (Input.GetKey(KeyCode.Q))
    {
        RotateObject(-1);
    }

    else if (Input.GetKey(KeyCode.E))
    {
        RotateObject(1);
    }
    else
    {
        if (outline.enabled)
            outline.enabled = false;
    }
}

private void RotateObject(int direction)
{
    outline.enabled = true;
    float newAngle = currentAngle + direction * rotationSpeed * Time.deltaTime;

    newAngle = Mathf.Clamp(newAngle, minAngle, maxAngle);

    float angleDifference = newAngle - currentAngle;

    transform.Rotate(Vector3.right, angleDifference);

    currentAngle = newAngle;
}
}

```

Shape6_control.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Shape6_control : MonoBehaviour
{
    [SerializeField] private Outline outline;
    [SerializeField] private float rotationSpeed;
    [SerializeField] private float minAngle;
    [SerializeField] private float maxAngle;

    private float currentAngle = 0f;

    private void Update()
    {
        if (Input.GetKey(KeyCode.C))
        {
            RotateObject(-1);
        }

        else if (Input.GetKey(KeyCode.V))
        {
            RotateObject(1);
        }
        else
        {
            if (outline.enabled)
                outline.enabled = false;
        }
    }

    private void RotateObject(int direction)
    {
        outline.enabled = true;
        float newAngle = currentAngle + direction * rotationSpeed * Time.deltaTime;
        newAngle = Mathf.Clamp(newAngle, minAngle, maxAngle);
        float angleDifference = newAngle - currentAngle;
        transform.Rotate(Vector3.forward, angleDifference);
        currentAngle = newAngle;
    }
}

```

Shape7_control.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Shape7_control : MonoBehaviour
{
    [SerializeField] private Outline outline;
    [SerializeField] private float rotationSpeed;
    private float currentAngle = 0f;

    private void Update()
    {
        if (Input.GetKey(KeyCode.Mouse0))
        {
            RotateObject(-1);
        }

        else if (Input.GetKey(KeyCode.Mouse1))
        {
            RotateObject(1);
        }
        else
        {
            if (outline.enabled)
                outline.enabled = false;
        }
    }

    private void RotateObject(int direction)
    {
        outline.enabled = true;
        float newAngle = currentAngle + direction * rotationSpeed * Time.deltaTime;
        float angleDifference = newAngle - currentAngle;
        transform.Rotate(Vector3.right, angleDifference);
        currentAngle = newAngle;
    }
}

```

Switch_cam.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Switch_cam : MonoBehaviour
{
    private Camera TheCamera;
    public Camera SecondCam;
    void Start()
    {
        TheCamera = GetComponent<Camera>();
        TheCamera = Camera.main;
    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            TheCamera.enabled = !TheCamera.enabled;
            SecondCam.enabled = !SecondCam.enabled;
        }
    }
}

```