

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ
ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Приладобудівний факультет
Кафедра виробництва приладів

«На правах рукопису»

УДК 004.946:681.2

«До захисту допущено»

Завідувач кафедри

 Михайло БЕЗУГЛИЙ
«02» 12 2022 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»
на тему: «Система доданої реальності для виконання складальних робіт»

Виконав:

Студент II курсу, групи ПБ-11мп
Чигрін Олег Вячеславович

Науковий керівник:

Доцент, кандидат технічних наук
Барандич Катерина Сергіївна

Консультант з розробки стартапу:

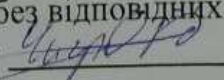
Завідувач кафедрою економічної кібернетики,
д.е.н., професор

Бояринова Катерина Олександрівна

Рецензент: Старший викладач

Кравченко Ігор Володимирович

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць інших
авторів без відповідних посилань.

Студент 

Київ – 2022 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Приладобудівний факультет
Кафедра виробництва приладів

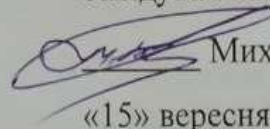
Рівень вищої освіти – другий (магістерський)

Спеціальність – 151- Автоматизація та комп'ютерно-інтегровані технології

Освітньо-професійна програма «Комп'ютерно-інтегровані системи та технології в приладобудуванні»

ЗАТВЕРДЖУЮ

Завідувач кафедри

 Михайло БЕЗУГЛИЙ

«15» вересня 2022 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

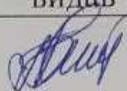
Чигріну Олегу Вячеславовичу

1. Тема дисертації «Система доданої реальності для виконання складальних робіт», науковий керівник дисертації Барандич Катерина Сергіївна, к.т.н., доцент, затверджені наказом по університету від «09» листопада 2022р. № 4110-с
2. Термін подання студентом дисертації 05 грудня 2022 року.
3. Об'єкт дослідження: процеси синтезу доданої реальності з елементами складальних робіт.
4. Вихідні дані: креслення складальних одиниць.
5. Перелік завдань, які потрібно розробити: 1. Аналіз сучасних систем AR підтримки складальних робіт. 2. Постановка задачі дослідження. 3. Розробка алгоритму роботи застосунку для технічної підтримки працівника, що виконує складальні роботи. 4. Розробка основних сценаріїв використання застосунку. 5. Вибір платформи для створення застосунку. 6. Розробка базового функціоналу застосунку. 7. Тестування ефективності розробки. 8. Збір даних за результатами тестування, їх аналіз та налагодження застосунку. 9. Виконання стартап-проекту.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: блок-схеми алгоритму роботи застосунку; графічні ілюстрації отриманих результатів дослідження.

7. Орієнтовний перелік публікацій: тези на науково-технічній конференції, стаття, авторське свідоцтво на твір (комп'ютерна програма).

8. Консультанти розділів дисертації*

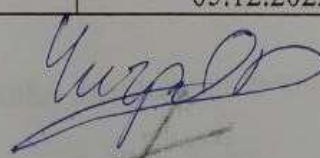
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Стартап-проекти	Завідувач кафедри економічної кібернетики, д.е.н., проф. Бояринова К.О.		

9. Дата видачі завдання 15 вересня 2022 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Аналіз сучасних систем AR підтримки складальних робіт	22.09.2022	
2.	Постановка задачі дослідження	29.09.2022	
3.	Розробка алгоритму роботи застосунку. Розробка основних сценаріїв використання	06.10.2022	
4.	Вибір платформи для створення застосунку	13.10.2022	
5.	Розробка базового функціоналу застосунку	28.10.2022	
6.	Тестування ефективності розробки.	04.11.2022	
7.	Збір даних за результатами тестування, їх аналіз та налагодження застосунку	11.11.2022	
8.	Виконання стартап-проекту	18.11.2022	
9.	Оформлення пояснювальної записки та презентації.	25.11.2022	
10.	Презентація матеріалів	05.12.2022	

Студент



Олег ЧИГРІН

Науковий керівник



Катерина БАРАНДИЧ

АНОТАЦІЯ

В магістерській дисертації було описано процес розробки застосунку для виконання складальних робіт із використанням доданої реальності. Мета розробки - зменшити час на підготовку до виконання складальної операції. Проект містить п'ять розділів, перший – оглядовий; другий – теоретична частина; третій розділ – практична частина; четвертий – розділ апробації і п'ятий – стартап проект. В першому розділі було описано технології доданої реальності, наведено приклади використання доданої реальності для зменшення часу на виконання складальних робіт. Другий розділ роботи містить у собі огляд інструментів необхідних для розробки застосунку, їх аналіз та порівняння, для створення додатку було обрано середовище розробки Unity, оскільки в ній наявний увесь необхідний функціонал для створення необхідних функцій в додатку. Було розроблено алгоритм роботи застосунку, та спроектовано архітектуру додатку. Третій розділ містить опис розробки додатку, а саме підготовку 3D моделей для використання у середовищі розробки Unity, реалізація функціоналу анімації розборки і складання складальної одиниці. Створено кнопки для керування анімацією усього складального процесу та покрокової анімації, для більш детального перегляду необхідних дій. Було реалізовано функціонал анотування 3D моделей. В четвертому розділі було описано методику використання розробки та результати апробації на виробництво, використовуючи додаток вдалось зменшити час на виконання складальної операції в середньому на 30%. Були описані можливі перспективи і подальший розвиток додатку. В п'ятому розділі було створено стартап проект, було проаналізовано ринкові можливості запуску проекту, розроблена ринкова та маркетингова стратегія стартап проекту.

ABSTRACT

The master's thesis were described the process of developing an application for assembly work using augmented reality. The purpose of the development is to reduce the time for preparing for the assembly operation. The project contains five sections, the first is overview; the second – the theoretical part; the third section is the practical part; the fourth is the approbation section and the fifth is the startup project. In the first chapter, augmented reality technologies were described, and examples of the use of augmented reality to reduce assembly time were given. The second section of the work contains an overview of the tools necessary for application development, their analysis and comparison, the Unity development environment was chosen for the application creation, as it has all the necessary functionality for creating the necessary functions in the application. The algorithm of the application was developed, and the architecture of the application was designed. The third section contains a description of the development of the application, namely the preparation of 3D models for use in the Unity development environment, the implementation of the disassembly and assembly animation functionality of the assembly unit. Created buttons to control the animation of the entire assembly process and step-by-step animation, for a more detailed view of the necessary actions. The functionality of annotation of 3D models was implemented. In the fourth chapter, the methodology of using the development and the results of approval for production were described, using the application it was possible to reduce the time for the assembly operation by an average of 30%. Possible prospects and further development of the application were described. In the fifth chapter, a startup project was created, the market opportunities for launching the project were analyzed, and the market and marketing strategy of the startup project was developed.

Зміст	
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	7
Вступ	8
1. Аналітичний розділ.....	11
1.1 Історія виникнення технології доповненої реальності	11
1.2 Використання доповненої реальності	12
1.3 Аналіз сучасних систем ДР підтримки складальних робіт	18
Висновки першого розділу.....	23
2. Теоретичний розділ.....	24
2.1 Вибір платформи для створення застосунку	24
2.2 Розробка алгоритму роботи застосунку.....	33
2.3 Розробка архітектури застосунку.	36
Висновки до другого розділу.	42
3.1 Розробка функціоналу застосунку	43
3.1.1 Підготовка моделі складальної одиниці.....	43
3.1.2 Створення проекту. Налаштування середовища розробки.....	45
3.1.3 Створення анімації зборки і розборки циклоїдального редуктора.....	51
3.1.4 Створення покрокової анімації зборки і розборки циклоїдального редуктора.....	54
3.1.5 Анотування 3D моделей складальних одиниць циклоїдального редуктора.....	63
4. Практична апробація	75
4.1 Рекомендації щодо впровадження розробки в виробництво.	75
4.2 Методика використання розробки.....	75
4.3 Результати тестувань застосунку в реальних умовах	77
Можливі перспективи розробки.	77
Висновки четвертого розділу.	78
5. Розробка стартап-проекту	79
5.1 Опис ідеї проекту	79
5.2 Аналіз ринкових можливостей запуску стартап-проекту	81
5.3 Розроблення ринкової стратегії проекту.....	89
5.4 Розробка маркетингової стратегії стартап проекту	91
Висновки до п'ятого розділу.....	94
ЗАГАЛЬНІ ВИСНОВКИ	95
Список використаних джерел	96

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

AR – Augmented reality

ДР – Додана реальність

VR – Virtual reality

ВР – Віртуальна реальність

SDK -Software Development Kit

САПР – система автоматизованого проектування

Вступ

Актуальність роботи. Застосування технології доповненої реальності (ДР) для управління процесом складання є новим підходом у вітчизняному виробництві. Доповнена реальність дозволяє інтегрувати віртуальну інформацію, таку як комп'ютерна графіка, текст, звук у фізичне середовище, щоб користувачі могли сприймати цю інформацію як існуючу в реальному часі [1]. Одною з перспектив розвитку технологій доповненої реальності є їх інтеграція у виробничий процес у якості «цифрових креслень», проте не в звичному до сьогодні форматі. Технологи на виробництвах під час розрахунку часу на виконання складальних робіт завжди враховують час, який витрачається на ознайомлення з кресленням, що значно збільшує кількість витраченого часу. А це, в свою чергу, впливає на витрати на заробітну плату та собівартість продукції. У випадку складних складальних вузлів робітник потребує постійної та поетапної перевірки виконаних дій, проте сьогодні креслення існують зазвичай лише у 2D формі на комп'ютері, планшеті/телефоні або на папері, що не є дуже зручним, оскільки робітнику доводиться чергувати свою увагу між фактичним складанням та інструкціями зі складання.

Метою роботи є створення додатку на телефон з операційною системою Android, для зменшення підготовчо-заключного часу на виконання складальної операції.

Для досягнення поставленої мети необхідно виконати наступні **задачі**:

1. Провести аналіз існуючих рішень використання доданої реальності в складальних процесах.
2. Розробити алгоритм роботи та архітектуру застосунку для відображення цифрової інформації необхідної для складального процесу.
3. Розробити застосунок для відображення цифрової інформації необхідної для складального процесу.
4. Розробити методику використання розробленого застосунку в умовах одиничного виробництва.
5. Провести апробацію розроблених рішень в умовах реального виробництва.

Об'єкт дослідження: процеси синтезу доданої реальності з елементами складальних робіт.

Предмет дослідження: методи створення системи доданої реальності для виконання складальних робіт.

Наукова новизна одержаних результатів:

Розроблено методику використання доданої реальності для виконання складальних робіт, що дозволяє зменшити час їх виконання.

Практична цінність:

1. Розроблено алгоритм роботи застосунку використання доданої реальності, що включає розпізнавання маркерів, анімацію складання та відображення анотацій деталей.

2. Розроблено архітектуру застосунку.

3. Розроблено застосунок анімації зборки і розборки циклоїдального редуктора на платформі «Unity» мовою програмування C#.

Апробація результатів дисертації.

Результати досліджень отримані в даній магістерській дисертації були оприлюднені на:

- XXI Науково-практична конференція студентів та аспірантів «Приладобудування: стан і перспективи», 17-18 травня 2022р., м. Київ, Україна.
- XVIII Науково-практична конференція студентів, аспірантів та молодих вчених «Ефективність та Автоматизація інженерних рішень у приладобудуванні», 06-07 грудня 2022 р., м. Київ, Україна.

Публікації:

1. Чигрін О.В. Використання можливостей доданої реальності при виконанні складальних робіт / О.В. Чигрін, К.С. Барандич, М.М. Гладський // Збірник тез доповідей XXI Міжнародної науково-технічної конференції «ПРИЛАДОБУДУВАННЯ: стан і перспективи», ПБФ, КПІ

ім. Ігоря Сікорського, 17-18 травня 2022 р., Київ, Україна, 2022. – с. 65-67.

2. Чигрін О.В. Використання доданої реальності при складанні циклоїдального редуктора / О.В. Чигрін, К.С. Барандич // Збірник праць XVIII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Ефективність та автоматизація інженерних рішень у приладобудуванні», 06-07 грудня 2022 р. К.: ПБФ, КПІ ім. Ігоря Сікорського. – 2022. – прийнято до опублікування.
3. Чигрін О.В., Барандич К.С., Гладський М.М. ВИКОРИСТАННЯ ДОДАНОЇ РЕАЛЬНОСТІ ПРИ СКЛАДАННІ ЦИКЛОЇДАЛЬНОГО РЕДУКТОРА, Вісник Київського політехнічного інституту. Серія приладобудування, Вип. 64(2), 2022. – прийнято до опублікування.

1. Аналітичний розділ

1.1 Історія виникнення технології доповненої реальності

Доповнена реальність була вперше певною мірою досягнута кінематографістом на ім'я Мортон Хейліг [2] у 1957 році. Він винайшов Sensorama, яка доставляла глядачеві зображення, звуки, вібрацію та запахи (рис.1.1).



Рис.1.1 Винахід Мортон Хейліга Sensorama[3]

Звичайно, комп'ютером установка не керувалась, але це був перший приклад спроби додати інші канали відтворення інформації. Потім у 1968 році Іван Сазерленд, американський комп'ютерний науковець винайшов накладний дисплей як своєрідне вікно у віртуальний світ. Застосовувана на той час технологія зробила винахід непрактичним для масового використання. У 1975 році Майрон Крюгер, американський комп'ютерний художник, розробив перший інтерфейс «віртуальної реальності» у формі «Videoplace», який дозволяв користувачам маніпулювати віртуальними об'єктами та взаємодіяти з ними в реальному часі.

Першою належним чином функціонуючою системою ДР була, ймовірно, та, що була розроблена в дослідницькій лабораторії ВПС Армстронга Луїсом Розенбергом у 1992 році (рис.1.2). Вона називалася Virtual Fixtures [4] і була неймовірно складною робототехнічною системою. Її призначенням була компенсація браку високошвидкісної процесорної потужності 3D-графіки на початку 90-х років. Це дозволяло накладати сенсорну інформацію на робочий простір для підвищення продуктивності людини.

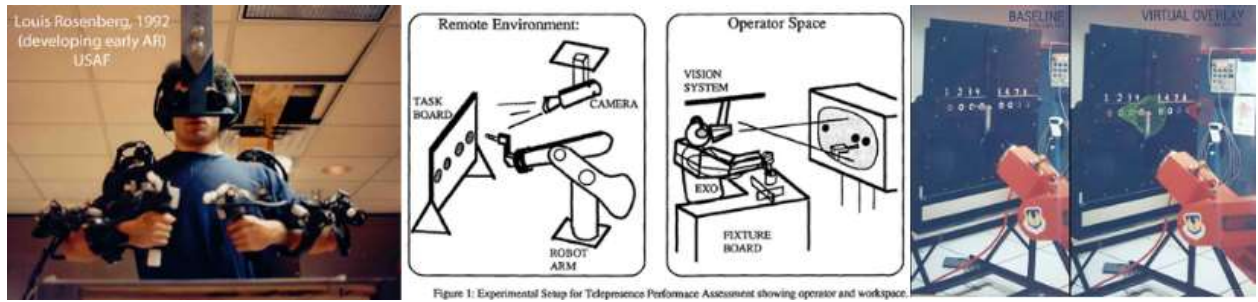


Рис.1.2 Винахід Луїса Розенберга Virtual Fixtures[5]

1.2 Використання доповненої реальності

Доповнена реальність або англійською augmented reality (AR) [6] змінює спосіб зв'язку та взаємодії на виробництві. Розширена версія навколишнього світу, доповнена реальність – це поєднання цифрових і аудіовізуальних елементів, які поєднують реальний і цифровий світи. На відміну від віртуальної реальності (VR), ДР дозволяє взаємодіяти з цифровою інформацією та навколишнім світом. VR відриває користувача від реального світу, блокуючи його відчуття справжньої реальності за допомогою гарнітур. Доповнена реальність, з іншого боку, розміщує їх в полі зору користувача у реальному світі. Незалежно від обставин дисплеї розширеної реальності інтегрують нову реальність у нашу поточну. В цілому ДР покращує виробничі можливості, роблячи її ідеальним інструментом для Індустрії 4.0.

Компанії все більше інвестують у розвиток взаємодії людини та машини (рис.1.3). Щоб залишатися конкурентоспроможними, провідні виробники продовжують запроваджувати автоматизацію для виконання простих повторюваних завдань і відстеження даних. Однак ручні процеси все ще вкрай актуальні. Виробники ставлять більш складні завдання своїм працівникам і підвищують вимоги до швидкості та якості робіт. Доповнена реальність забезпечує вирішення потреб сучасного виробництва. ДР показує наступні дії та відстежує ручні процеси, що веде до нового розуміння сучасних, складних операцій. Дослідження взаємодії людей і машин за допомогою доповненої реальності в Університеті Патри [7] виявили, що «інтерфейси доповненої реальності можуть поєднуватися з робочим середовищем і

збагачувати його, роблячи взаємодію з роботами природною та інтуїтивно зрозумілою».

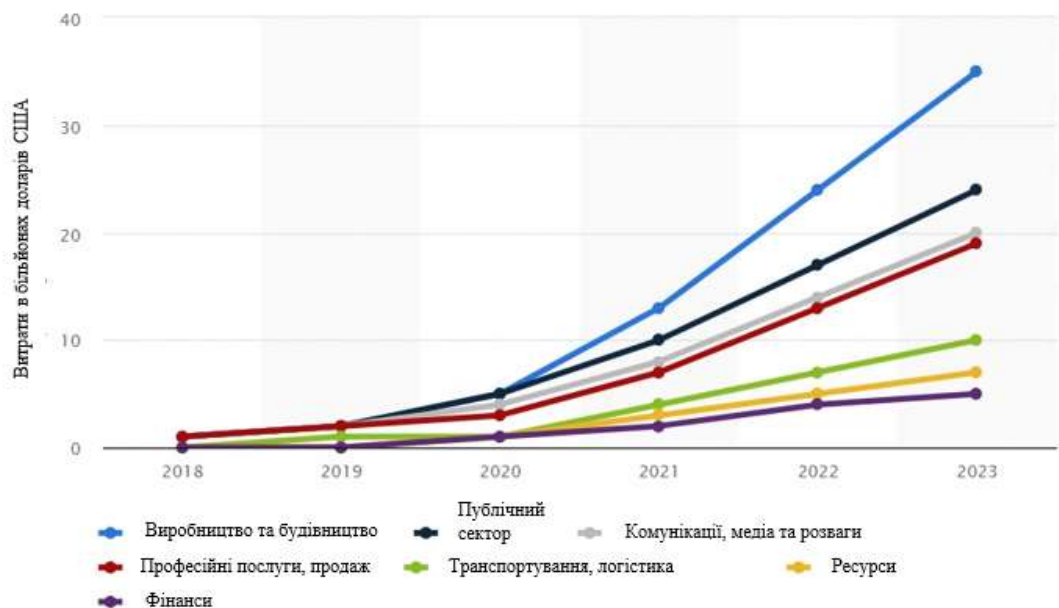


Рис. 1.3 Очікуваний зріст ринку ДР після врахування ефекту пандемії[8]

Подібні дослідження показують, що зв'язок між людиною та машиною за допомогою ДР допомагає провідним виробникам виявляти можливі дефекти в виробничому процесі. За найменшими рухами, виявленими 3D-сенсорами та камерами машинного бачення, найкращі рішення ДР для виробництва можуть ідентифікувати найдрібніші системні випадки. Агрегування такої інформації приводить виробників до точних моментів неефективності на рівні підприємства та показує їм, як оптимізувати операції для підвищення продуктивності.

Надалі наведено перелік, як сучасні компанії використовують ДР на виробництвах.

- Тренування та підвищення кваліфікації.

Використання рішень ДР для вирішення проблем виробничого навчання та підвищення кваліфікації приносить користь компаніям і працівникам на всіх рівнях. Цифрові робочі інструкції з доповненою реальністю масштабуються відповідно до бізнесів, де вони є, незалежно від того, чи потрібно працівникам навчатися на індивідуальному, фабричному чи корпоративному рівні. Одне натискання кнопки легко перемикає програми, забезпечуючи виробниче навчання для окремих осіб або

великих груп. Ці програми легко адаптуються до будь-якого рівня кваліфікації, залишаючись актуальними як для тих, хто має невеликий досвід виробництва так і для тих, хто має 30+ років досвіду. Перш за все, цифрові робочі інструкції стандартизують навчання. Усні інструкції можуть відрізнятися залежно від того, хто навчає. Паперові інструкції можуть відрізнятися залежно від часу їх останнього оновлення, на відміну від цифрових онлайн інструкцій. Робочі інструкції з доповненою реальністю оновлюються на підприємстві в один клік. Підключені системи оновлюються одночасно, що забезпечує відсутність втрати інформації. Крім того, цифрові робочі інструкції забезпечують однакову інформацію на кожній робочій станції.

- Цифрові робочі інструкції

Робочі інструкції ДР направляють працівників через стандартизовані процеси (рис.1.4). Замість того, щоб працівник читав, наприклад, про три дії одночасно лише з одним зображенням для демонстрації, ДР проектує робочі інструкції безпосередньо на робочу поверхню, направляючи працівників крок за кроком. Тому вся увага зосереджена на якості однієї дії, що значно скорочує час, кількість помилок і когнітивне навантаження на працівника.



Рис. 1.4 Відображення робочих інструкцій для робітника[9]

Крім того, ДР полегшує написання цифрових робочих інструкцій. Додаючи візуальні та звукові підказки до кроків, потрібно менше тексту. У таких системах, таких як програмне забезпечення ДР LightGuide, є додаткові сенсорні функції, такі як

звукові підказки, миготливі вогні, анімація та відео, щоб привернути увагу до інструкцій.

- Варіація технічних завдань

Після написання робочих інструкцій вони додаються до індексу програми, яку має виконати працівник. Завдяки такій системі працівники можуть швидко та зручно адаптуватися до варіацій продукту і без значної втрати часу приступити до виконання нового технічного завдання. Одне сканування штрих-коду дозволяє програмному забезпеченню AR, підключеному до корпоративної системи, розпізнавати нову деталь і автоматично викликати відповідні робочі інструкції.

- Забезпечення якості

Інспекція та перевірка надзвичайно важливі для забезпечення якості. Проте все ще часто перевірки не проводяться до завершення процесу. Завдяки керованим рішенням ДР робочі інструкції можуть включати швидкі перевірки на будь-якому етапі без обмеження часу циклу.

- Ергономіка

Оскільки доповнена реальність відображає цифрові накладки на поверхні навколо вас, вона може адаптуватися до ваших потреб. Не всі працюють в однакових умовах. Потрібні коригування відповідно до типу роботи, яка виконується, місця та кількості завдань, що виконуються за раз. В ергономіці ці коригування визначаються як фізичні, когнітивні та організаційні потреби. Разом ергономіка значною мірою впливає на те, як працівники взаємодіють із робочими завданнями. Цифрові робочі інструкції ДР можна налаштувати. Коли потрібно внести зміни, щоб пристосувати працівників, це можна зробити за лічені хвилини. Крім того, цифрові робочі інструкції можуть пропонувати текст кількома мовами. Незважаючи на те, що візуальні та звукові елементи не потребують тексту, автоматично зменшуючи мовні бар'єри, додані кнопки дозволяють миттєво перемикаати параметри мови.

- Аналітика процесів

Розвиток автоматизації також призвів до збільшення доступних даних про процеси, що виконуються працівниками. Однак ручні процеси все ще актуальні, і виробники не дуже часто звертають увагу на них. Технологія ДР дозволяє

виробникам отримувати доступ до ручних процесів і автоматизованих даних одночасно. Система ДР продовжує збирати тривалість циклів і дефекти, та додає більше дрібних точок для аналізу, наскільки ефективно проходить процес. Наприклад, розширені робочі інструкції відстежують час кроків, точні дії, які призвели до дефекту, і робочі показники в реальному часі. Загалом ці додані вручну дані надають ширшу картину заводських функцій, показуючи, коли саме та де виникають проблеми. Виробники в різних галузях по всьому світу інвестують у рішення AR. Хоча вони стикаються з різними проблемами, ДР достатньо універсальна, щоб вирішити кожную з них і надати додаткові переваги.

Зараз існує три основних типи AR, які використовуються для підтримки виробничих і монтажних операцій. Кожен із них забезпечує необхідний ефект відображення інформації, який покращує роботу. Але, виходячи з умов і потреб підприємства, знання того, який тип ДР найкраще відповідає додатку, призведе до найбільшого успіху в роботі.

Перший тип – це технологія доповненої реальності з використанням планшетів (рис.1.5). Це комбінація планшетів та інших портативних пристроїв (наприклад, смартфонів), які надають робочі інструкції через програми доповненої реальності.



Рис. 1.5 Використання планшетів для взаємодії з ДР[10]

Планшетна ДР особливо проста у використанні. Більшість людей знайомі з технологією планшетів, що спрощує навчання та скорочує час початку роботи. Розповсюдженість пристроїв та відносно невелика ціна робить технологію доволі

розповсюдженою і легко-впроваджуваною. Під час використання планшетної ДР користувач отримує додаткову інформацію про об'єкт або інструкції до виконання на ділянці, крізь інтерфейс планшету.

Наступний тип – доповнена реальність в окулярах. Технологія ДР проектує накладання графіки на навколишнє середовище за допомогою окулярів ДР (рис.1.6) або гарнітури. Більшість людей знайомі з розумними окулярами ДР, такими як Google Glass і Microsoft HoloLens. Вони діють як легкі комп'ютери для відображення та зберігання інформації, за використання яких руки лишаються вільними.



Рис. 1.6 Використання робітником ДР за допомогою окулярів[11]

Порівняно з доповненою реальністю для планшетів, доповнена реальність, що доступна крізь окуляри, є популярним вибором на виробництвах завдяки своїй мобільності та динамічній графіці.

Рішення ДР для виробництва з проекцією на поверхню. Проектована ДР використовує комбінацію проекторів і датчиків зору для відображення віртуальної покрокової інтерактивної графіки на будь-якій робочій поверхні (рис.1.7). Його можна використовувати в широкому діапазоні виробничих рішень, включаючи складання, підбір деталей і комплектацію, випробування та перевірку, навчання та технічне обслуговування.



Рис. 1.7 Розширення реальності за допомогою проекції інформації на поверхню[12]

1.3 Аналіз сучасних систем ДР підтримки складальних робіт

Які варіанти використання ДР у розумному виробництві? Застосування доповненої реальності в поєднанні зі штучним інтелектом відкрило [13] нові можливості в розробці продукції, виробництві, обслуговуванні машин, логістиці та підтримці клієнтів.

Розробка продукту: завдяки прогресу в технологіях, екологічних нормах і очікуваннях клієнтів розробка нових продуктів має вийти на новий рівень, щоб забезпечити більш короткий час виходу на ринок за менших витрат. Використання доповненої реальності на ранній стадії розробки продукту допомагає дизайнерам віртуально та більш гнучко переробляти свої проекти від ідеї до концепції, не несучи витрат на швидке створення прототипів.

Виробництво: від першої промислової революції до Індустрії 4.0 кількість компонентів у будь-якому продукті зросла в тисячу разів, що робить процес складання більш складним і виснажливим, щоб працівники могли виконувати завдання без помилок у найшвидший спосіб. Інтерактивні закадрові інструкції з доповненою реальністю використовуються, наприклад, у складних процесах складання мобільних телефонів і літаків (рис.1.8). Тоді процес може бути швидшим і безпомилковим, без надмірного тиску на працівників.

Технічне обслуговування машин: потреба в продуктах найвищої якості призвела до використання дуже складних машин у виробничих установках. Звичайно,

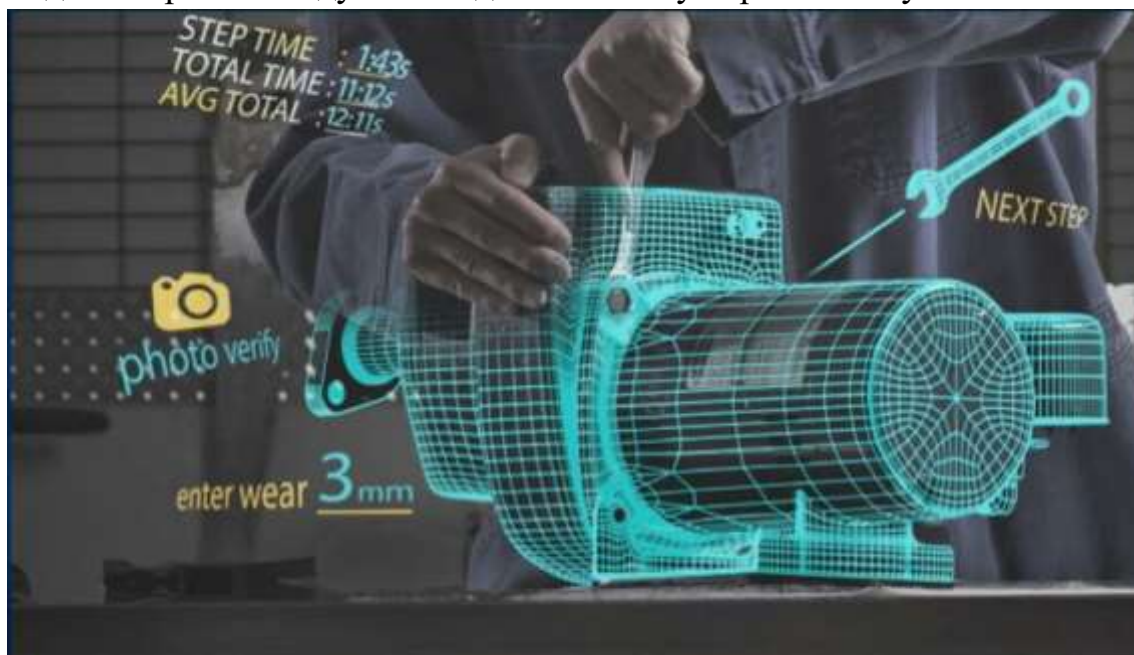


Рис. 1.8 Використання ДР для складальних операцій на виробництві[14]

раптові поломки машин завжди призводили до простоїв і втрат виробництва. Доповнена реальність може скоротити витрати на періодичне технічне обслуговування машини та втрати через поломку за допомогою щоденного моніторингу в реальному часі. Він також може допомогти передбачити потенційні тенденції збоїв, виконуючи швидку діагностику, а також швидший ремонт і відновлення. Логістика: Зменшення витрат на складські операції завжди було передбачувано через втрату часу доставки та збитки, спричинені неправильною обробкою або завантаженням вантажу. ДР може оптимізувати підбір вантажу для вантажників за допомогою допоміжних та інтерактивних голосових інструкцій, щоб знайти потрібний вантаж і визначити, де його потрібно розмістити в транспортному засобі.

Підтримка клієнтів: зазвичай служба підтримки клієнтів обчислювальна електронна техніка потребує кількох візитів техніків для проведення діагностики, замовлення запасних частин і заміни чи ремонту обладнання. Це ускладнює для виробників обладнання забезпечення кращої взаємодії з клієнтами за допомогою найшвидшого вирішення. Розумні окуляри ДР можуть запропонувати технічну

підтримку клієнтів у режимі реального часу з персоналізованими інструкціями у віддаленому робочому середовищі та можуть допомогти швидше вирішувати проблеми без необхідності фізичного візиту.

На сьогодні деякі компанії вже використовують ДР технології для вирішення подібних задач. Інженери компанії Boeing продемонстрували застосунок (рис.1.9), що використовує додану реальність для допомоги робітникам при складанні літаків [15]. Випробування з використанням доданої реальності призвело до покращення продуктивності монтажу проводки на 40%.



Рис. 1.9 Використання розширеної реальності інженерами Boeing[15]

Інженери в «Iowa State University» [16] працюють над системою, що здатна розпізнавати та відстежувати деталі під час складання, накладати на них 3D моделі цих деталей та відображати наступні кроки складального процесу. Проте для того, щоб відстежувати позицію деталі у реальному часі їм необхідно використовувати прилад з датчиками глибини, який за допомогою інфрачервоного проектора створює тривимірне зображення об'єкта.

В дослідницькому інституту «Fraunhofer-Institut für Gießerei-, Composite- und Verarbeitungstechnik IGCV» розроблено застосунок [17] «MA2RA – Manual Assembly Augmented Reality Assistant» для цифрової підтримки робітника під час складальних робіт (рис.1.10).



Рис. 1.10 MA2RA інструкції до складального процесу[17]

На робочому місці працівника розміщуються штрих коди/QR коди. Скануючи їх, застосунок знає про розміщення усіх необхідних складальних елементів та зображує у 3D поетапно процес складання, що допомагає швидше виконати робоче завдання та уникнути помилок під час його виконання.

Вирішення питання супроводу працівника під час виконання складальних робіт також презентувалось на «33rd ACM Symposium on User Interface Software and Technology», інженери зі Сполучених Штатів Америки презентували [18] застосунок, для створення та візуалізації інтерактивних навчальних посібників із 3D-доповненої реальності (рис.1.11) на основі 2D-відео, яка дозволяє контролювати точку спостереження під час виконання. Використовуючи графік складання вдається виявити етапи складання, які інакше важко



Рис. 1.11 Виконання складальних процесів з використанням відео-анотації[18]

виділити з відео, що також загалом покращує виявлення об'єктів і відстеження. Щоб уникнути втрати інформації, поєднується 3D-анімація з відповідними частинами 2D-відео, це дає можливість показати детальні маніпуляції та використання інструментів, які нелегко експортувати з відео.

Оглянуті застосунки мають виражений недолік, для їх роботи, не враховуючи створення самого застосунку, необхідно виконати багато підготовчих робіт, що ускладнює процес інтеграції у виробництво. Дані розробки відображають лише технологічний процес складання, проте у багатьох випадках робітникам потрібно отримати інформацію, що яка стосується складальних одиниць, функціонал анування 3D моделей вирішив би це питання.

Проаналізувавши існуючі розробки та потреби виробництва на сьогоднішній момент, було запропоновано розробити застосунок для пришвидшення процесу складання на виробництві. В якості об'єкту дослідження запропоновано взяти циклоїдальний редуктор, що використовується як складальна одиниця в роботі маніпуляторі. Як платформа було обрано телефон на базі операційної системи Android. Розроблений мобільний застосунок має слугувати інформаційно-навчальним інструментом для працівників, що виконують складальні роботи які потребують багато людино-годинних ресурсів. Таким чином метою даної магістерської дисертації є – зменшення часу складання деталей приладів в умовах одиничного виробництва за рахунок використання технологій доданої реальності.

Для досягнення поставленої мети необхідно виконати наступні задачі:

1. Провести аналіз існуючих рішень використання доданої реальності в складальних процесах.
2. Розробити алгоритм роботи та архітектуру застосунку для відображення цифрової інформації необхідної для складального процесу.
3. Розробити застосунок для відображення цифрової інформації необхідної для складального процесу.
4. Розробити методику використання розробленого застосунку в умовах одиничного виробництва.
5. Провести апробацію розроблених рішень в умовах реального виробництва.

Висновки першого розділу.

У даному розділі було проведено огляд та аналіз сучасних систем доповненої реальності і їх використання на сучасних виробництвах. Були розглянуті можливі варіанти реалізації технологій, такі як: додана реальність з використанням планшетів або телефонів, додана реальність в окулярах та проекція додаткової реальності на робочі поверхні. Серед представлених варіантів реалізації було запропоновано обрати варіант з використанням доданої реальності за допомогою телефону. Провівши дослідження сучасних систем доповненої реальності, що використовуються на складальних ділянках, було запропоновано розробити застосунок для зменшення часу складання приладів в умовах одиничного виробництва за рахунок відображення необхідної інформації про складальний процес на телефоні робітника.

2. Теоретичний розділ

2.1 Вибір платформи для створення застосунку

Враховуючи швидкий розвиток індустрії доповненої реальності, як розробники, так і не розробники використовують деякі з найпопулярніших платформ, інструментів розробника та комплектів розробки програмного забезпечення (SDK) для створення додатків і досвіду доповненої реальності. Було проведено аналіз сучасних платформ, що працюють з технологією доповненої реальності[19].

1. ARKit

Розробники iOS використовують ARKit (рис.2.1) від Apple для створення мобільних програм та ігор ДР для iPhone, iPad та інших пристроїв Apple. ARKit використовує технологію візуальної інерційної одометрії для визначення розмірів навколишнього середовища та налаштування умов освітлення на основі розташування 3D-об'єктів. Він пропонує виявлення зображень і поверхонь, а також відстеження об'єктів і обличчя, що дозволяє створювати захоплюючі багатокористувацькі AR-ігри.

Продукт Apple, ARKit підтримує розробку лише в iOS.

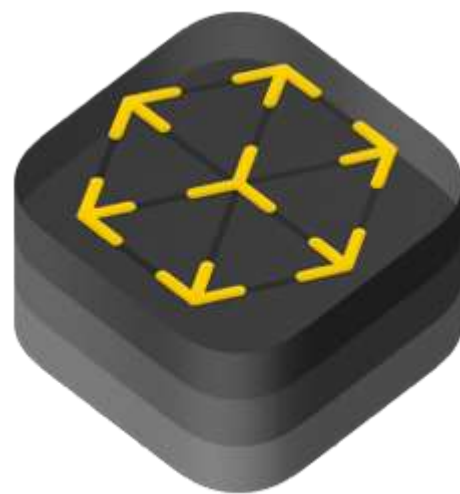


Рис.2.1 Лого платформи ARKit [20]

2. ARCore

ARCore від Google (рис.2.2) має багато функцій, які допомагають інтегрувати елементи ДР у реальне середовище, зокрема відстеження руху, виявлення поверхні та оцінку освітлення. Зокрема, це дозволяє одночасно відтворювати 3D-об'єкти на

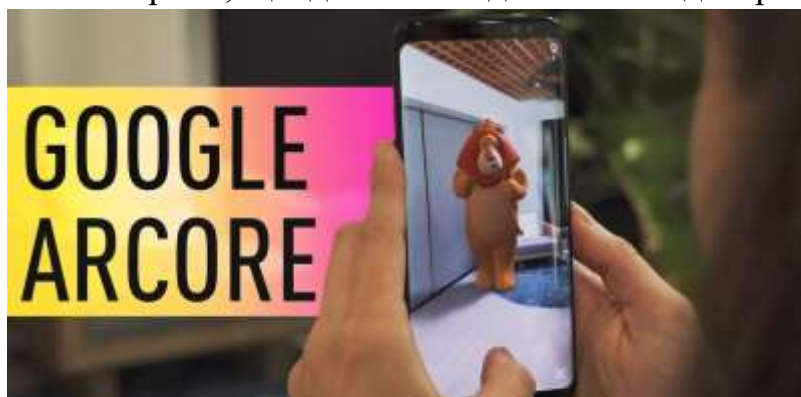


Рис.2.2 Використання платформи ARCore [21]

різних пристроях. Він підтримує розробку в Android, iOS, Unreal і Unity, а також 3D-малювання за допомогою Tilt Brush від Google.

3. echoAR

echoAR (рис.2.3) — це хмарна платформа для 3D, доповненої реальності (AR) і віртуальної реальності (VR), яка надає інструменти та інфраструктуру на стороні сервера, щоб допомогти розробникам і компаніям швидко створювати та розгортати 3D-додатки та контент. Завдяки



Рис.2.3 Використання echoAR [22]

гнучкій хмарній інфраструктурі, інтуїтивно зрозумілому управлінню вмістом і системою, швидкій мережі інтернет та іншим функціям echoAR забезпечує високоефективну 3D-розробку та процеси доставки. Він дає змогу розробникам створювати серверну програму 3D/ДР/ВР, а розробникам контенту – легко керувати та публікувати 3D-контент у своїй програмі ДР/ВР.

4. Unity

Unity є одним із найпопулярніших інструментів розробки ігор, який використовувався для створення таких популярних ігор, як Pokémon Go, HeARthstone і Rimworld. Завдяки API сценаріїв C#, вбудованій інтеграції Visual Studio та потужним інструментам анімації Unity є одним із найкращих варіантів для розробників ігор, дизайнерів і художників.

5. Vuforia

Популярний ДР SDK, Vuforia (рис 2.4) використовує технологію комп'ютерного зору для розпізнавання та відстеження 3D-об'єктів у режимі реального часу. Він дає розробникам можливість створювати AR-доповнені як на основі маркерів, так і без маркерів, а однією з його основних особливостей є підтримка постійних



Рис.2.4 Лого платформи Vuforia [23]

об'єктів. Vuforia надає API для кількох мов програмування та підтримує нативну розробку для iOS, Android, UPW та Unity.

6. Wikitude

Wikitude – це гнучкий інструмент розробки AR, за допомогою якого розробники можуть налаштовувати рішення та отримувати 3D-об'єкти зі сторонніх SDK через власну платформу. Деякі з його функцій включають підтримку навігації, масштабування на основі відстані та локалізацію. Wikitude можна використовувати для Android, iOS і Windows для планшетів і деяких розумних окулярів.

7. Kudan

Багато в чому подібно до Vuforia, Kudan використовує технологію одночасної локалізації та відображення (SLAM) для створення високоякісної 3D-графіки та розпізнавання зображень і об'єктів. Kudan має кілька чудових функцій, таких як відстеження на основі маркерів, інтеграція карт і переміщення камери, а також підтримує розробку на Android та iOS.

8. Unreal Engine

Фреймворк Unreal Engine (рис.2.5) ДР забезпечує багату уніфіковану структуру для створення додатків доповненої реальності за допомогою Unreal Engine для

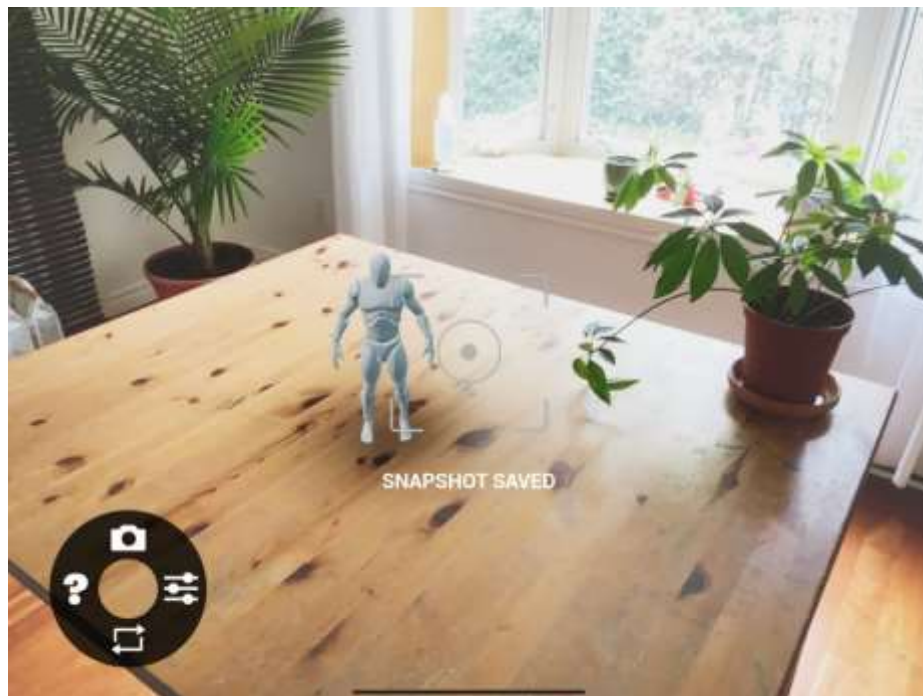


Рис.2.5 Застосунок створений в Unreal Engine 5[24]

портативних платформ iOS і Android. Уніфікована структура забезпечує єдиний шлях розробки для обох платформ, дозволяючи розробникам створювати програми доповненої реальності для обох платформ, використовуючи єдиний шлях коду. Шаблон Handheld ДР Blueprint надає повний приклад проекту, що демонструє функціональність доповненої реальності, доступну в Unreal Engine.

Наведений вище перелік платформ налічує частинку інструментів, що існують на сьогоднішній день, для створення застосунків з використанням ДР. Серед них можна виокремити повноцінні платформи, користуючись якими розробник може створити застосунок з функціоналом, який буде обмежуватися лише можливостями обраної платформи. До таких можна віднести: Unity, Unreal Engine 5, ARKit та echo3D. Технології або утиліти, що можна виокремити з цього списку, називаються SDK (software Development Kit). SDK – набір із засобів розробки, утиліт і документації, який дає програмістам змогу створювати прикладні програми за визначеною технологією або для певної платформи. Такі набори можна інстальювати в будь яку платформу, що підтримує таку можливість, та використовувати їх функціонал у своїх розробках. До SDK належать : Vuforia Engine, Wikitude, Kudan та ARCore. Необхідно зазначити, що деякі з наведених утиліт також можуть слугувати як платформа для створення застосунків, проте частіше вони доповнюють функціонал інших платформ.

Оскільки майбутній застосунок має бути програмованим, то вибір платформи звужується до двох платформ Unity та Unreal Engine (табл.2.1), для кінцевого рішення виконано аналіз та порівняння двох середовищ розробки. Ці механізми Real-Time 3D включають вбудовані та зовнішні пакети для скриптів (код і без коду), візуалізації (2D, 3D, XR), фізики, звуку, мережі, анімації, багатокористувацького користування тощо.

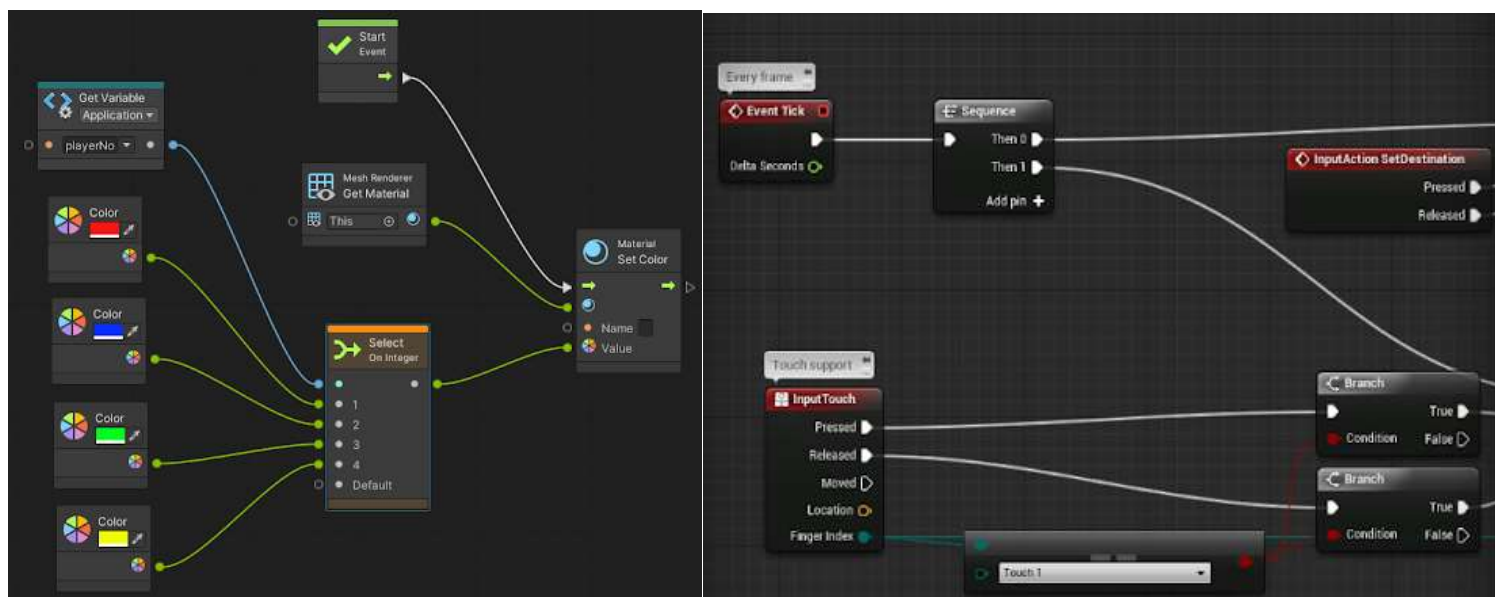
Перший сегмент порівняння є мова програмування, мова програмування якою мають бути прописані скрипти, що будуть використовуватись за потреби запрограмувати будь яку функцію застосунку.

Табл.2.1 Порівняння Unity та Unreal Engine

Unity (Bolt):	Unreal Engine:
1	2
1. Мова програмування скриптів	
C# використовується всередині редактора та з зовнішніми програмами. C# зручніший у швидкому процесі навчання, ніж C++. Крім того, C# добре підходить для загального використання та впровадження архітектури програмного забезпечення в масштабі, що потребує обслуговування, коли додаток росте.	У той час як можливо створити цілий проект в Unity з C#, те ж саме неможливо в Unreal з C++. C++ також дозволяє реалізувати архітектуру програмного забезпечення, але її важче вивчити та використовувати, ніж C#, а реалізація C++ в Unreal значною мірою покладається на специфічний для Unreal код, який не перекладається на загальне використання C++.
2. Візуальні мови програмування	
Це логіка на основі вузлів, де можна реалізувати поведінку для об'єктів у редакторі, і вона майже повністю сумісна з C#. Документація зі скриптів Unity зосереджена на C#, що робить візуальні скрипти Unity повільнішими для вивчення, ніж C#. Розширення доступних вузлів візуального сценарію не потребує спеціальних навичок. (рис 2.6, а)	Це основний спосіб зібрати логіку застосунку в Unreal, особливо для початківців. C++ забезпечує все, що не може бути виконано за допомогою Blueprint. Документація зі сценаріїв Unreal орієнтована на Blueprint, що дозволяє швидше навчатися людям, які не мають досвіду створення скриптів. Навчання використанню Blueprint не перекладається безпосередньо на сценарії на C++, що вимагає додаткового навчання та освіти. Розширення вузлів креслень вимагає ключових знань бібліотеки C++. (рис 2.6,б)

1	2
3. Кросплатформенність	
Окрім звичайних платформ, Unity має перевагу в XR: він підтримує веб-публікацію HTML5, тоді як Unreal припинив офіційну підтримку після версії 4.24. Це робить публікацію в WebXR з Unity все ще потенційно життєздатною, на відміну від Unreal.	Для розробки XR більшість платформ віртуальної та доповненої реальності підтримують Unity краще, ніж Unreal, це можна побачити у випусках набору інструментів для розробників. Unreal Engine часто випускає ці набори інструментів через кілька місяців після Unity, і функція, підтримувана Blueprints, може бути обмеженою.
4. Анімація	
Усі основні функції анімації в реальному часі підтримуються Unity. Анімація на основі кінематичних ланцюгів, анімація вершин, анімація станів об'єктів та керування послідовностями цих анімацій забезпечують необхідні будівельні блоки для побудови складних систем анімації.	Незважаючи на те, що обидві платформи забезпечують механізми анімації всередині редактора, вважається, що якість анімації та інструменти Unreal Engine більш зрілі. Серед останніх доповнень – система кріплення зі сценарієм Control Rig, яка надає більше можливостей керування механізмом без необхідності використання окремого програмного забезпечення для створення 3D-вимірів.

Візуальне програмування, аналог програмування, де розробнику не потрібно занурюватись у мову програмування, а створювати сценарії поведінки завдяки блок-схемам, що візуально доступні розробнику.



а

Рис.2.6

б

а -Візуальне програмування в Unity за допомогою інструменту Bolt [25]

б- Візуальне програмування в Unreal Engine за допомогою інструменту Blueprint [26]

Цільові платформи. Обидва механізми дозволяють крос-платформну розробку, включаючи Windows, Mac, Linux, Android, iOS, PS і Xbox. Дуже важливим аспектом у виборі є можливість створювати анімації в застосунку, оскільки переважна більшість функціоналу буде реалізована завдяки інструментам анімації.

Виконавши аналіз та порівняння двох середовищ розробки обрано для розробки додатку середовище Unity. До уваги взято велику кількість вже існуючих проектів, що реалізовані на цьому двигуні і створені в його середовищі. Можливість масштабування проекту також була важливим аспектом порівняння. Сьогоднішня тенденція великих виробничих компаній – переносити усі використовувані аплікації, застосунки в інтернет мережу, що робить бізнес більш гнучким і легшим до масштабування. Оскільки Unreal Engine зупинив підтримку WebXR, Unity отримує перевагу. Вважаючи, що функціоналу середовища Unity є більш, ніж достатньо – можна робити заключне рішення, що виконувати в цьому середовищі буде оптимальнішим для проекту.

Отже, взявши до уваги такі критерії, як можливості до програмування застосунку, можливість створювати крос-платформені застосунки, перспективи

масштабування та ціну комерційної підписки, запропоновано розробити застосунок у середовищі розробки «Unity».

Опис складальної одиниці. У рамках стартап проекту «LineBar» двоступеневий циклоїдальний редуктор (Додаток А) використовується для збільшення крутного моменту крокового двигуна на двох осях обертання робота маніпулятора [27].

Циклоїдальний привід або циклоїдальний редуктор (рис 2.7) – це механізм для зменшення частоти обертання вхідного валу на певне співвідношення. Циклоїдні редуктори швидкості мають відносно високі числа передачі (1:120) в компактних розмірах з дуже низьким люфтом (20 arcsec).

Вхідний вал (поз. 8) приводить в рух ексцентричну втулку (поз. 9), яка, у свою чергу, приводить циклоїдний диск (поз. 4) у ексцентричний, циклоїдний рух. Поверхня цього диска знаходиться у контакті з шестернею першої ступені (поз. 6) та шестернею другої ступені (поз. 5). Друга ступень приводить у рух вихідний вал (поз. 3) під час обертання циклоїдального диска (поз. 4). Радіальний рух диска (поз. 4) не передається на вихідний вал (поз.3). Специфікація складальної одиниці у Додатку Б.

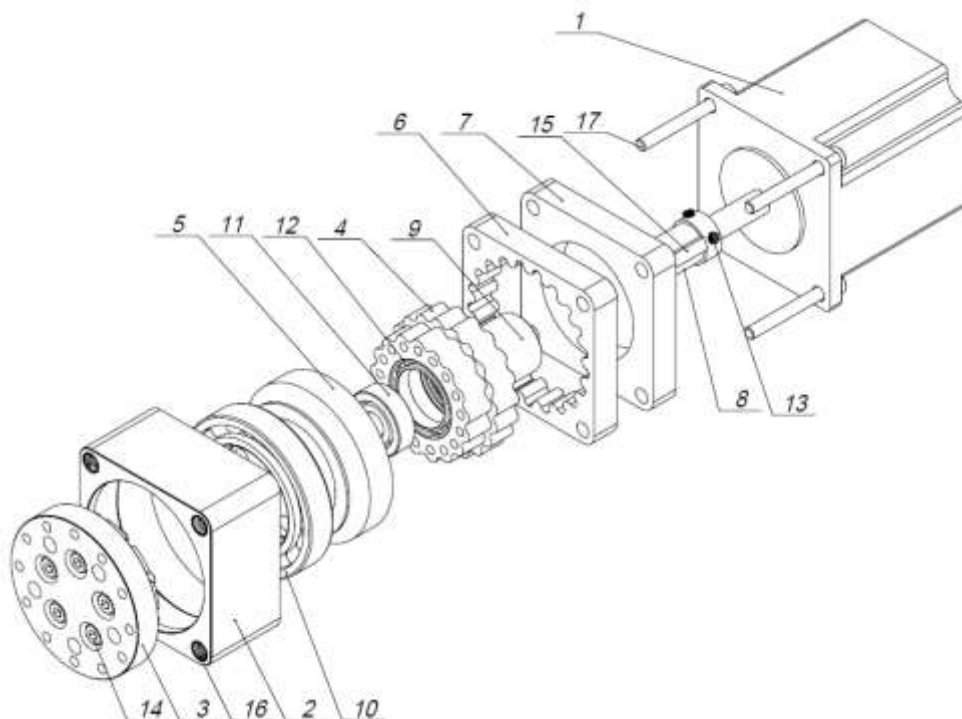


Рис.2.7 Розібрана модель циклоїдального редуктора

Циклоїдальні редуктори є високоефективними та високошвидкісними пристроями передачі руху та крутного моменту. Вони зазвичай використовуються в обладнанні, де потрібна точність позиціонування і великі корисні навантаження приводу. Останнім часом із зростаючим попитом на високоефективні та високошвидкісні пристрої для передачі крутного моменту, застосування циклоїдальних зубчастих редукторів стало популярним у сфері автоматизації, наприклад, у робототехніці, верстатах і автоматичних машинах.

Обрано наступні розмірні характеристики редуктора (рис.2.8):

$$d_1 = 48\text{мм.}, d_2 = 43\text{мм.}, D_1 = 52.5\text{мм.}, D_2 = 47.5\text{мм.}$$

$$n_1 = 20 \text{ зуб.}, n_2 = 17 \text{ зуб.}, N_1 = 21 \text{ зуб.}, N_2 = 18 \text{ зуб.}, e = 1\text{мм},$$

де d_1 та d_2 – діаметри першої та другої внутрішніх шестерень; D_1 та D_2 – діаметри внутрішніх шестерень; N_1 та N_2 – кількість зубців; e – зміщення ексцентрикової втулки.

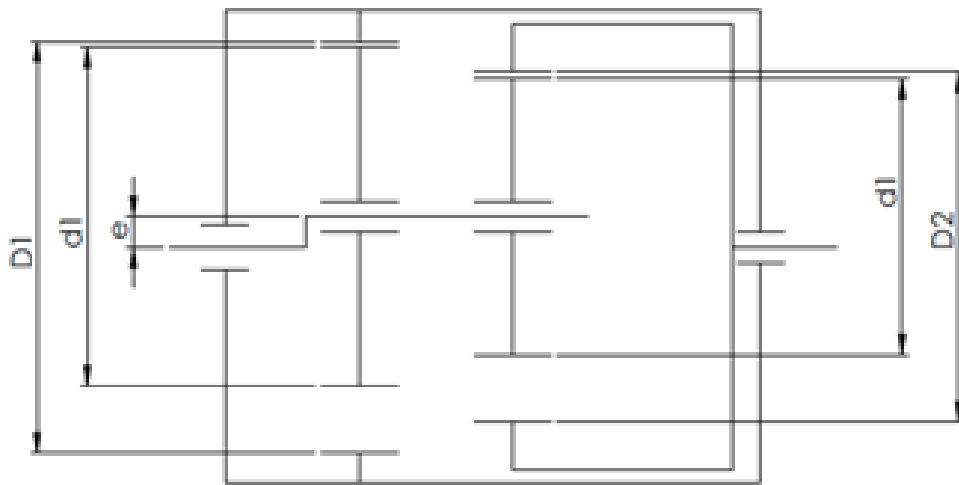


Рис.2.8 Схема двоступінчастого редуктора

Розрахунок передаточного числа двоступеневого циклоїдального редуктора розраховано за формулою 3.1. Отримане значення передаточного числа редуктора дозволило маніпулятору мати необхідну для роботи вантажопідйомність.

$$i = \frac{n_1 * n_2 + n_1}{n_1 - n_2} = \frac{20 * 17 + 20}{20 - 17} = 120 \quad (3.1)$$

Для проведення складальних робіт циклоїдального редуктора створено складальне креслення [Додаток А], структурну [Додаток В], і структурну технологічну схеми складання [Додаток Г],. А для зменшення підготовчо-заключного часу складальних робіт, запропоновано створити застосунок, що надає детальну інформацію про дії, які необхідно виконати під час складальних робіт.

2.2 Розробка алгоритму роботи застосунку.

Робота застосунку розпочинається з розпізнавання камерою маркера (рис 2.9), маркер використовується застосунком як орієнтир, що відстежує камера на ньому необхідно розмістити 3D модель. У випадку, якщо маркер вдалось розпізнати розпочинається рендеринг моделі та його відображення на маркері. Під час роботи програми положення камери відносно маркера може змінюватись і навпаки, маркер може бути переміщений. У такому випадку застосунок має прорахувати зміну положення маркера відносно камери і розпочати рендер 3D моделі та відобразити об'єкт на маркері.

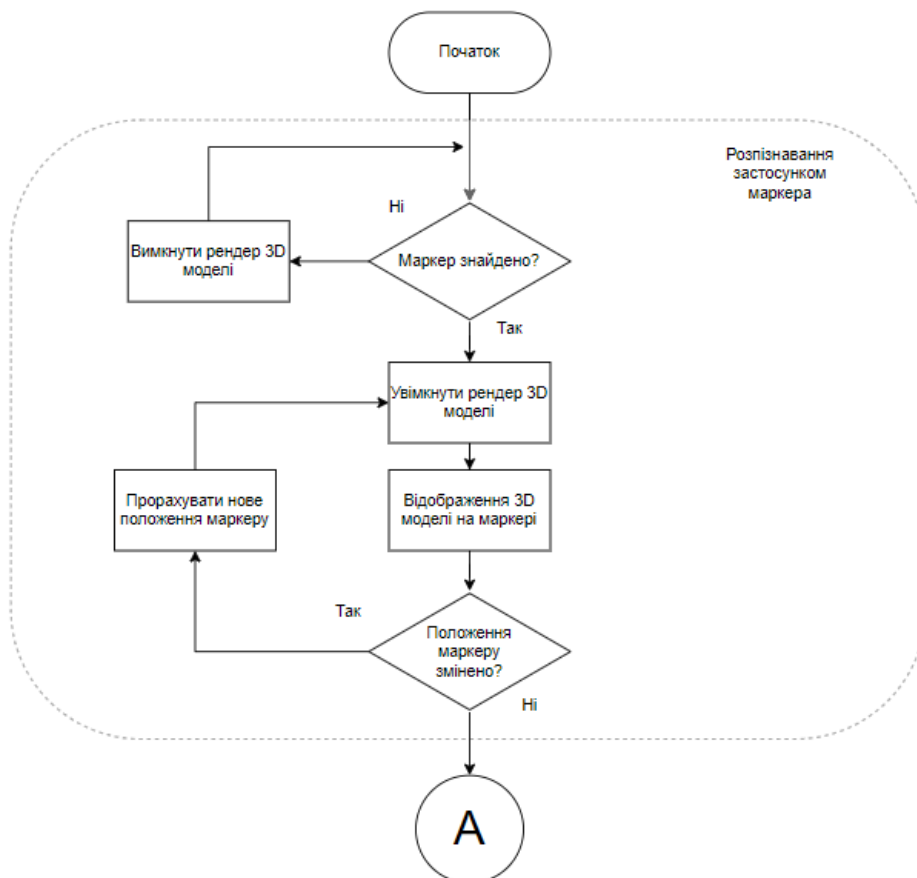


Рис.2.9 Ділянка «Розпізнавання застосунком маркера» блок схеми роботи застосунку

У випадку(рис 2.10), якщо маркер розпізнано і натиснута кнопка «Анімація»,

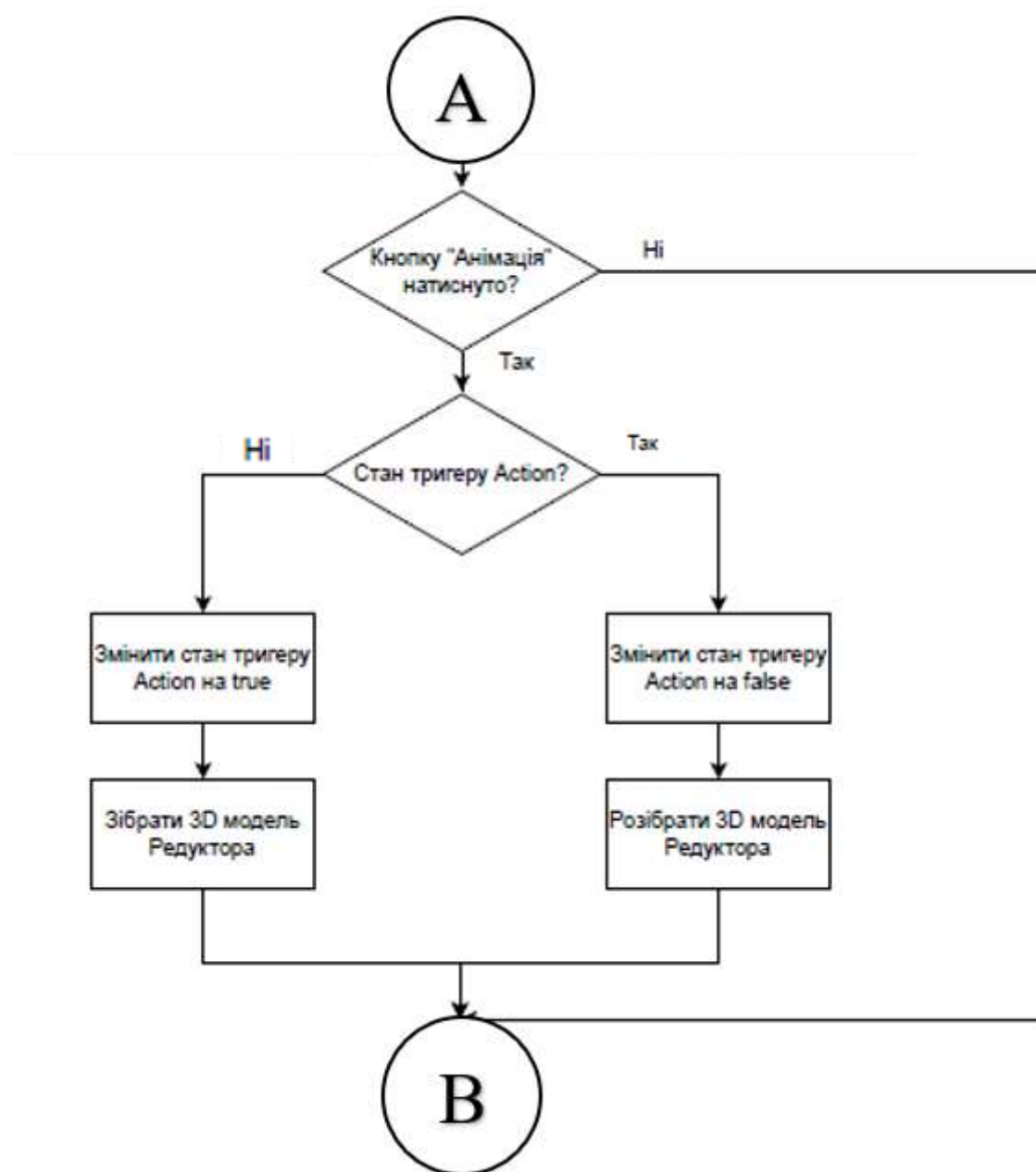


Рис.2.10 Алгоритм роботи застосунку у випадку натискання кнопки «Анімація»

застосунок має перевірити стан триггеру «Action» і змінити його на протилежний, це спричинить виклик інструменту «Animator», який розпочне анімацію зборки або розборки. Для більш детального перегляду етапів складання циклоїдального редуктору було запропоновано створити окремі кнопки для керування поетапною анімацією складання і розборки 3D моделі (рис 2.11). Для цього необхідно перевірити

чи зібрана або повністю розібрана модель циклоїдального редуктора, у такому випадку користувачеві відображається повідомлення, що редуктор повністю зібраний або повністю розібраний. У випадках, коли користувач натискає кнопку і анімація не закінчена, виконується дія, що йде наступною, згідно технічної схеми складання, а колір анімованих частин змінюється, для привернення уваги користувача.

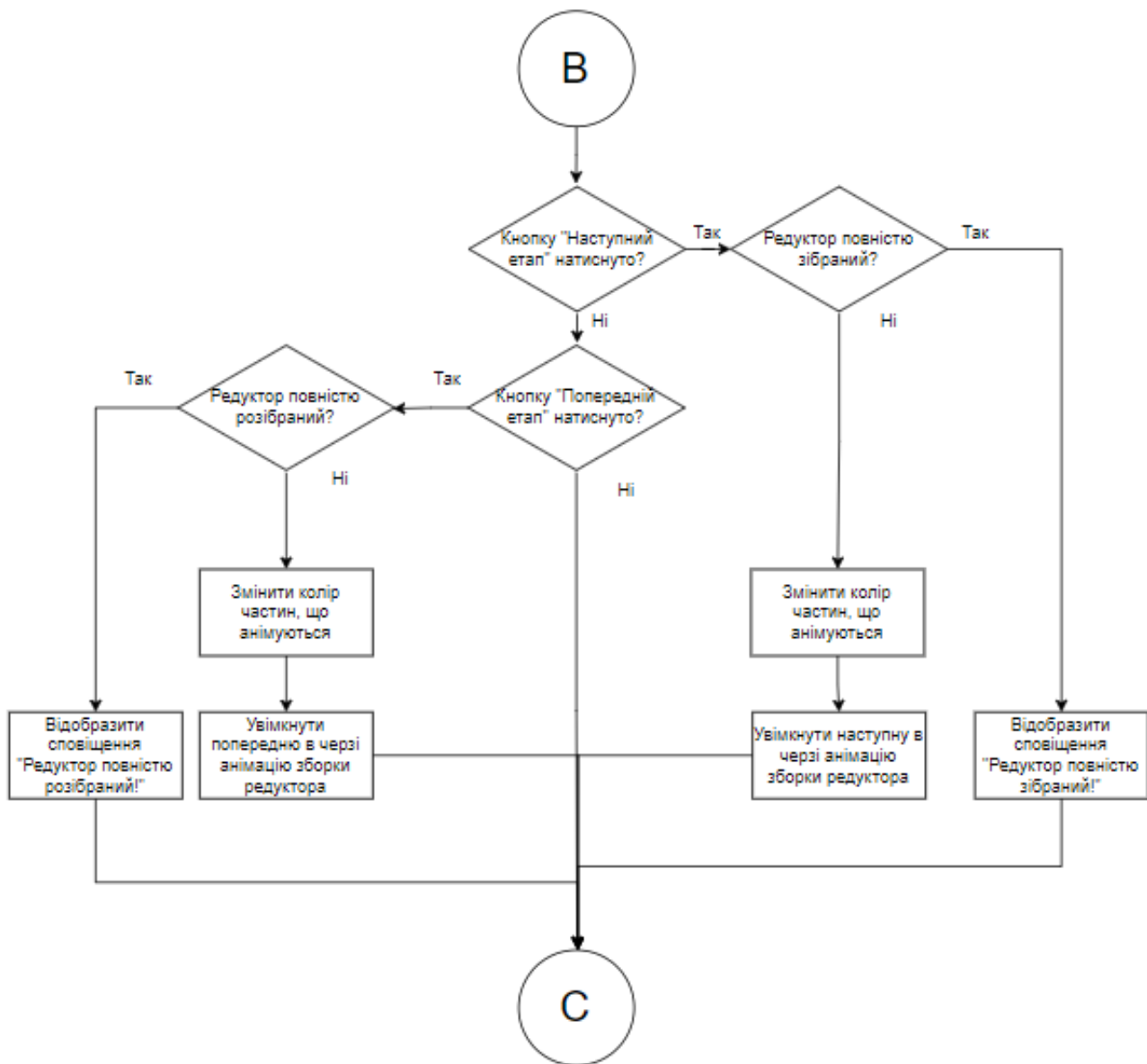


Рис.2.11 Алгоритм роботи застосунку у випадку натискання кнопок «Наступний етап» та «Попередній етап»

Алгоритм відображення анотацій 3D моделей (рис.2.12) складається з перевірки, чи розміщена деталь по центру екрану. У випадку, якщо деталь було знайдено. У випадку наявності перелічених елементів відображається анотація,

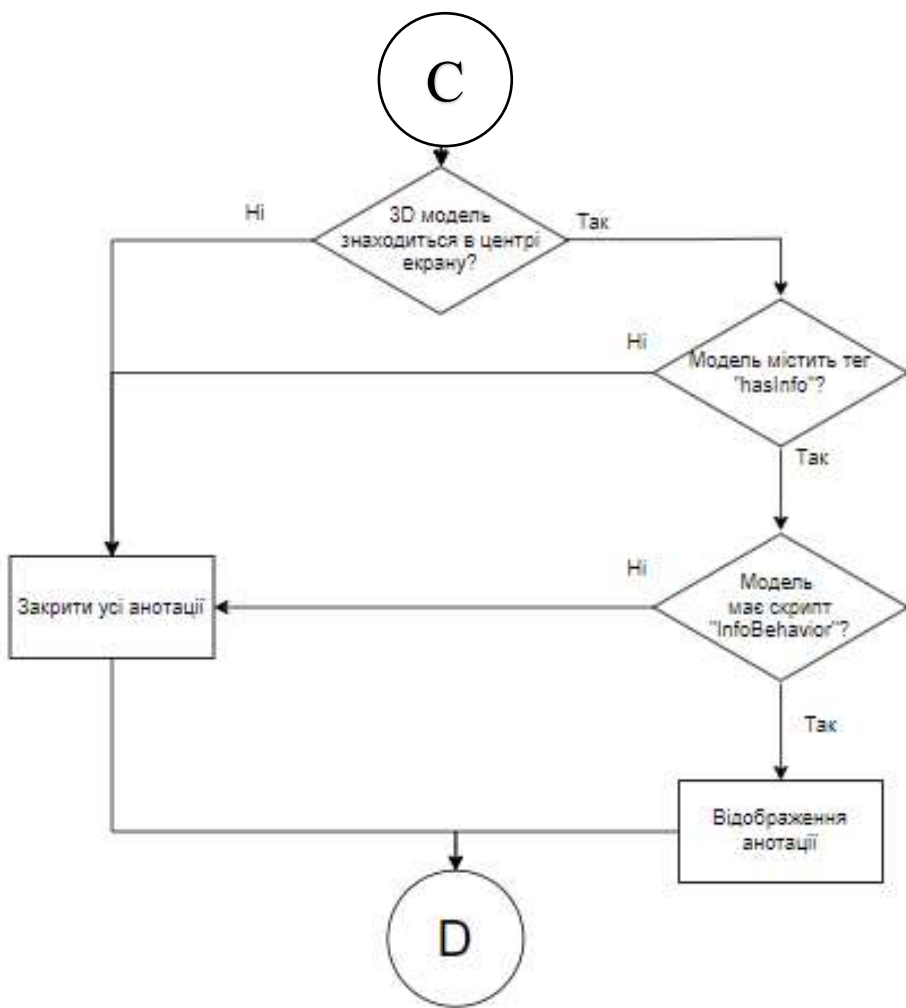


Рис.2.12 Алгоритм роботи застосунку для відображення анотацій

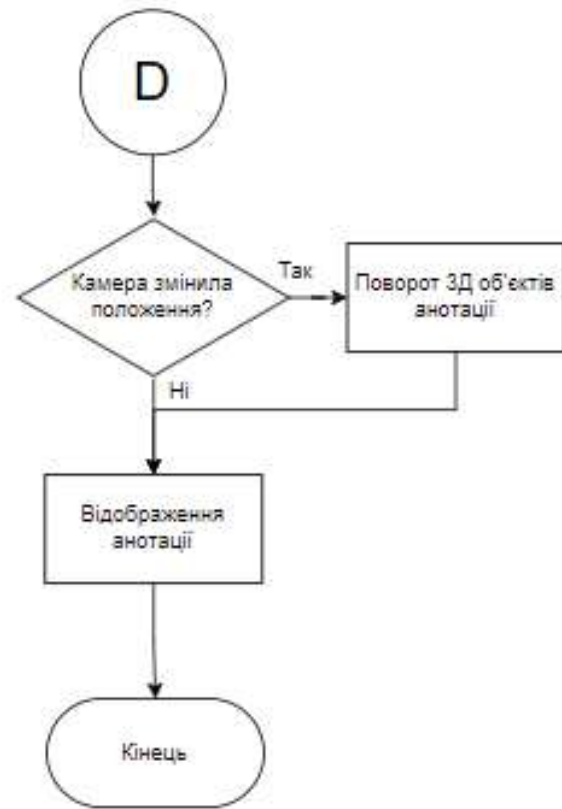


Рис.2.13 Алгоритм роботи застосунку для відображення анотацій

що відповідає 3D моделі. У випадку «False» хоча б в одній з умов, усі анотації закриваються. У випадку зміни положення камери (рис.2.13) застосунок має розвернути 3D об'єкти анотацій в сторону камери. Кожен кадр застосунку перевіряє розташування відображеної анотації відносно камери користувача, у випадку непаралельності площин відбувається поворот по осі «Z» усіх 3D об'єктів анотації.

2.3 Розробка архітектури застосунку.

Архітектура програмного забезпечення – це спосіб структурування програмної або обчислювальної системи, абстракція елементів системи на певній фазі її роботи. Система може складатись з кількох рівнів абстракції і мати багато фаз роботи, кожна

з яких може мати окрему архітектуру [28]. Терміном «Архітектура» також називають документування архітектури програмного забезпечення (ПЗ). Документування архітектури ПЗ спрощує процес комунікації між зацікавленими особами, дозволяє зафіксувати прийняті на ранніх етапах проектування рішення про високорівневий дизайн системи і дозволяє використовувати компоненти цього дизайну і шаблони проектування повторно в інших проектах.

Архітектуру застосунку запропоновано розділити на дві структури: загальна, високорівнева архітектура та атомарна структура, що описує класи і методи на рівні складальних одиниць циклоїдального редуктора.

На найвищому рівні архітектура поділяється на 4 класи (рис.2.14): «ARCamera», «Image Target», «Canvas» та «Event System», усі вони наслідують клас «GearBox Assembler», що є спільним класом усього застосунку.

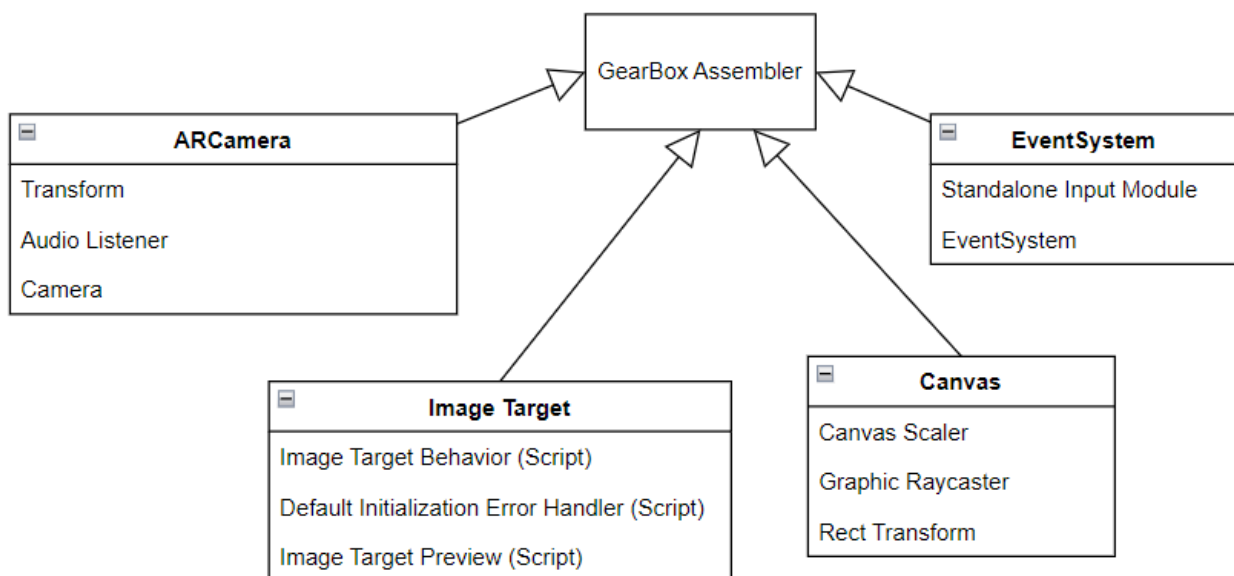


Рис.2.14 Найвищий рівень архітектури застосунку

Клас «Event System» створено для можливості динамічного керування, тобто як спосіб надсилання керуючих дій до об'єктів у програмі на основі введення, будь то клавіатура, миша, дотик або спеціальне введення. Система подій складається з кількох компонентів, які разом надсилають події [28]. Клас налічує методи «Standalone Input Module» та «Event System», що були створені автоматично. Компонент «Canvas» представляє абстрактний простір, у якому розміщується та відображається інтерфейс користувача. Усі елементи інтерфейсу користувача мають бути дочірніми до компоненту «Canvas». Автоматично створені поля у класі «Canvas» використовуються для налагодження інтерфейсу користувача. Згаданий вище клас «Image Target» містить у собі методи «Image Target Behavior» та «Image Target Preview», що являють собою скрипти, які були імпортовані до проекту разом із «SDK Vuforia». У кожному застосунку, що реалізує ДР, має бути об'єкт «ARCamera», цей клас відображає камеру користувача. Поля створені в «ARCamera» автоматично отримують інформацію під час роботи застосунку, а саме візуальне зображення надходить до «Camera» та звуковий ряд до «Audio Listener». Поле «Tag» містить інформацію про тег камери, що використовується методами, які імплементуються в класі «ARCamera» (рис.2.15).

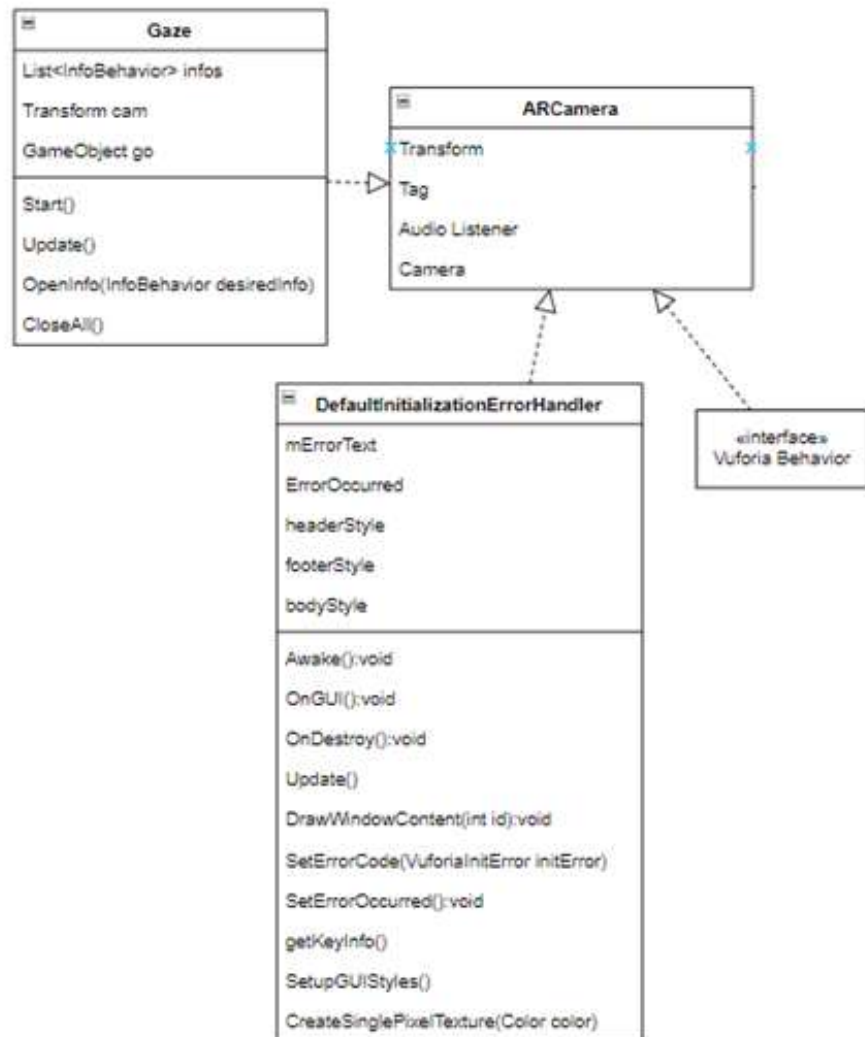


Рис.2.15 Архітектура класу «ARCamera»

Клас «ARCamera» використовує прописаний скрипт-клас «Gaze», що містить методи «Start()», «Update()», «OpenInfo()» та «CloseAll()». Скрипт «DefaultInitializationErrorHandler», що імплементує клас «ARCamera» був завантажений до застосунку, для уникнення помилок під час роботи додатку.

Автоматично, після імпортування «SDK Vuforia» створено інтерфейс «Vuforia Behavior», скрипт було імплементовано в клас «ARCamera». У класі «Image Target» імплементовано імпортований скрипт «DefaultObserverEventHandler» для уникнення програмних помилок під час використання додатку. Нащадком класу «Image Target» (рис.2.16) є об'єкт «Game Object», він імплементує написаний скрипт «Animator», що контролює усі анімації у застосунку, тому його використано в об'єкті, що поєднує у собі усі 3D моделі складальних одиниць, які містяться у класі нащадку «GearBox», що наслідує «GameObject».

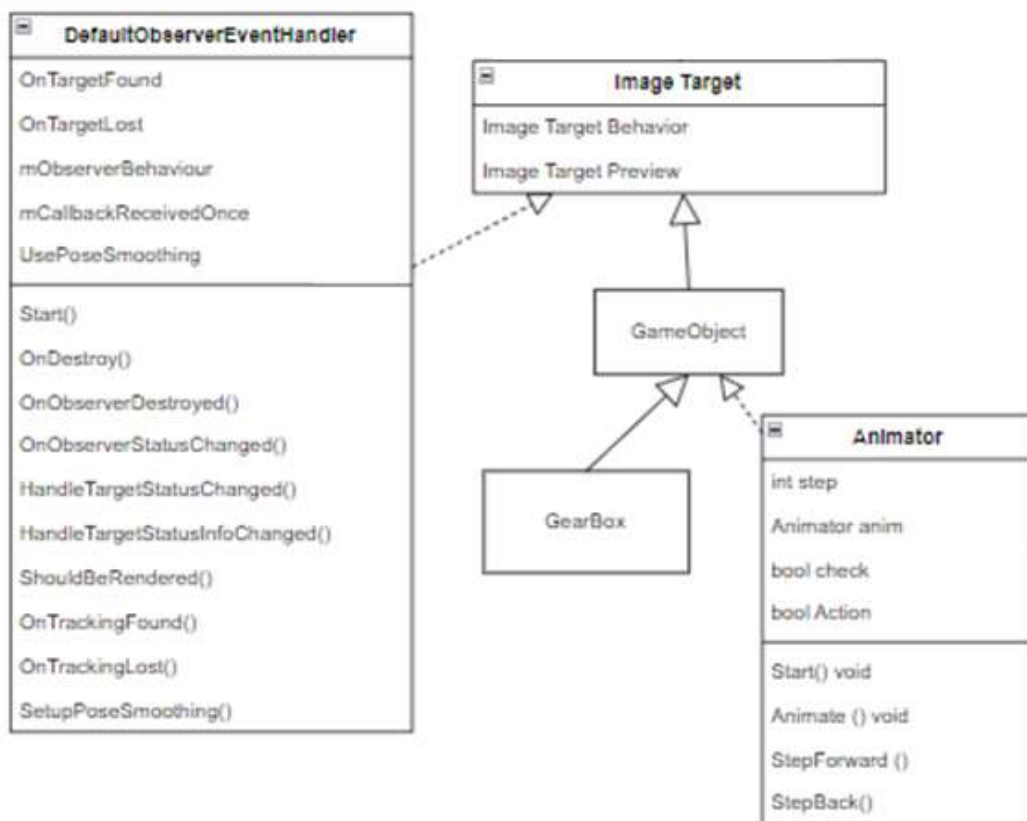


Рис.2.16 Архітектура класу «ImageTarget»

Клас «Canvas» (рис.2.17) є батьківським класом для трьох класів, «AnimateButton», «BackButton» та «ForwardButton», вони є класами кнопок, що дають можливість користувачеві керувати анімаціями, кожен з класів нащадків

використовує у собі методи «Animator», відповідно: «Animation.Animate», «Animation.StepBack», «Animation.StepForward». У кожного з класів нащадків є свій клас нащадок Text(TMP), що містить текст-назву кнопки.

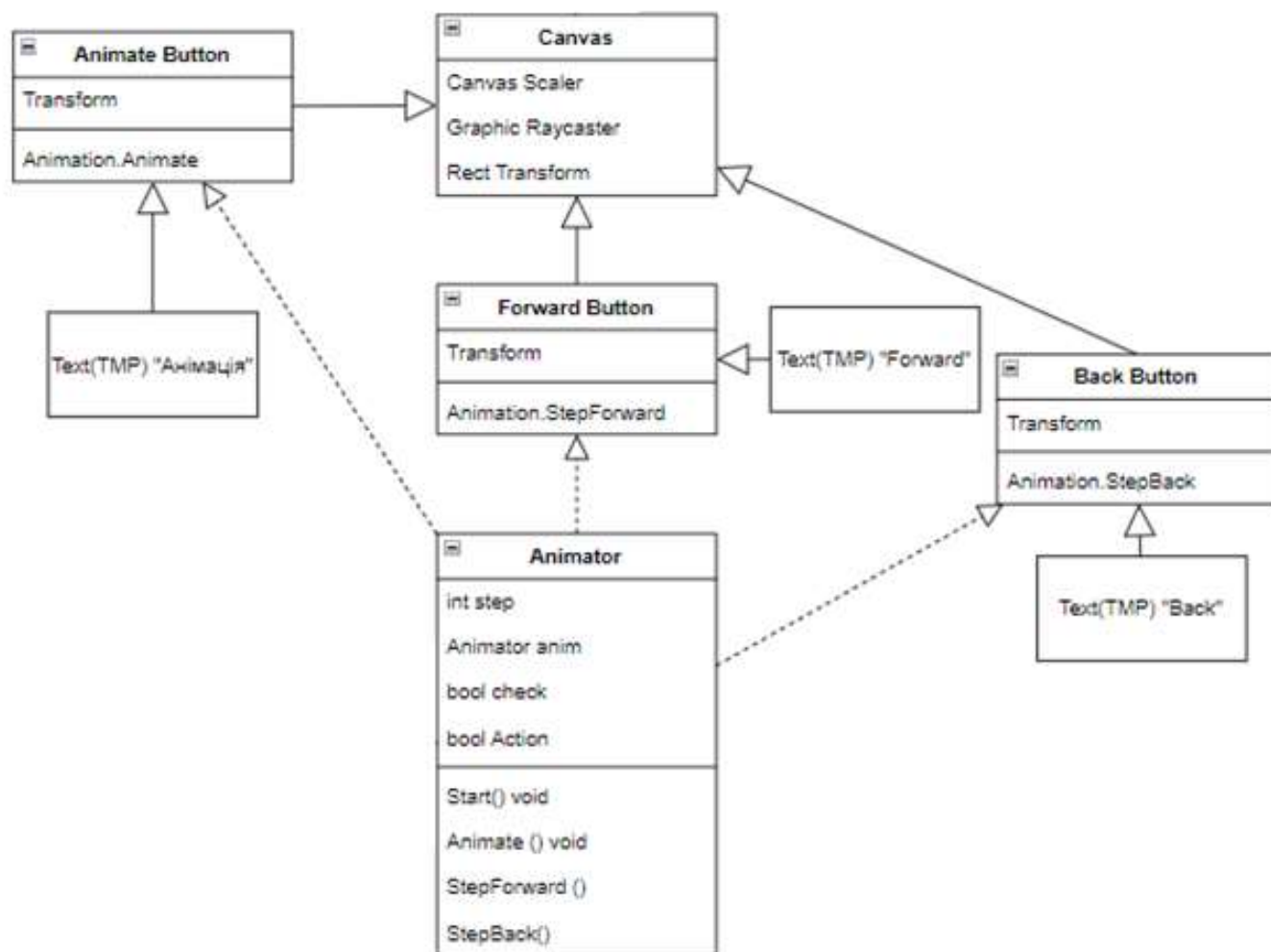


Рис.2.17 Архітектура класу «Canvas»

Другий етап архітектури це структура класу «GearBox» (рис.2.16). За замовчуванням клас містить лише об'єкти згідно специфікації, проте реалізація функціоналу анотування об'єктів вимагає розширення класу. Усі 3D об'єкти в застосунку, що мають взаємодіяти з користувачем, вимушені мати поля «Transform» – для задання координат, кутів повороту та масштабу об'єкту, «BoxCollider», «MeshRenderer» та «MeshCollider» для взаємодій з іншими об'єктами. На схемі виокремлено клас 3D моделі «Вал перехідний» і на його прикладі проілюстровано структуру кожного класу 3D моделі, в якому реалізовано функціонал анотування 3D моделей. Усі моделі, які підлягають анотуванню імплементують скрипт

«InfoBehavior». Клас «Вал перехідний» містить клас нащадок «Section Info» (рис.2.17).

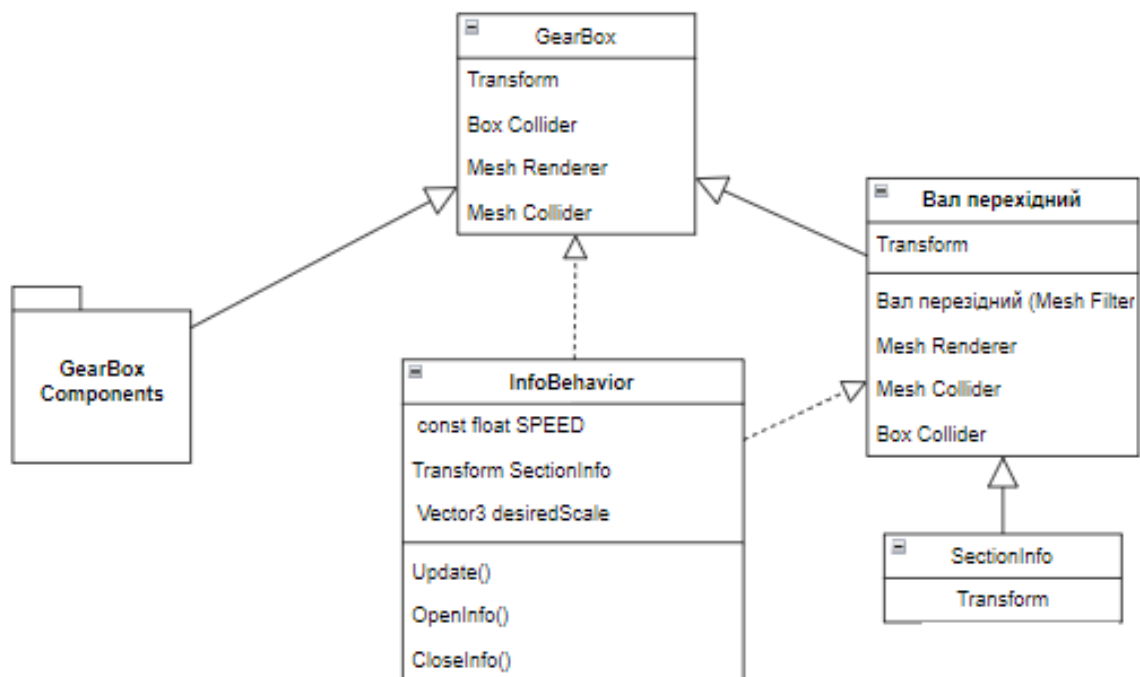


Рис.2.16 Архітектура класу «GearBox»

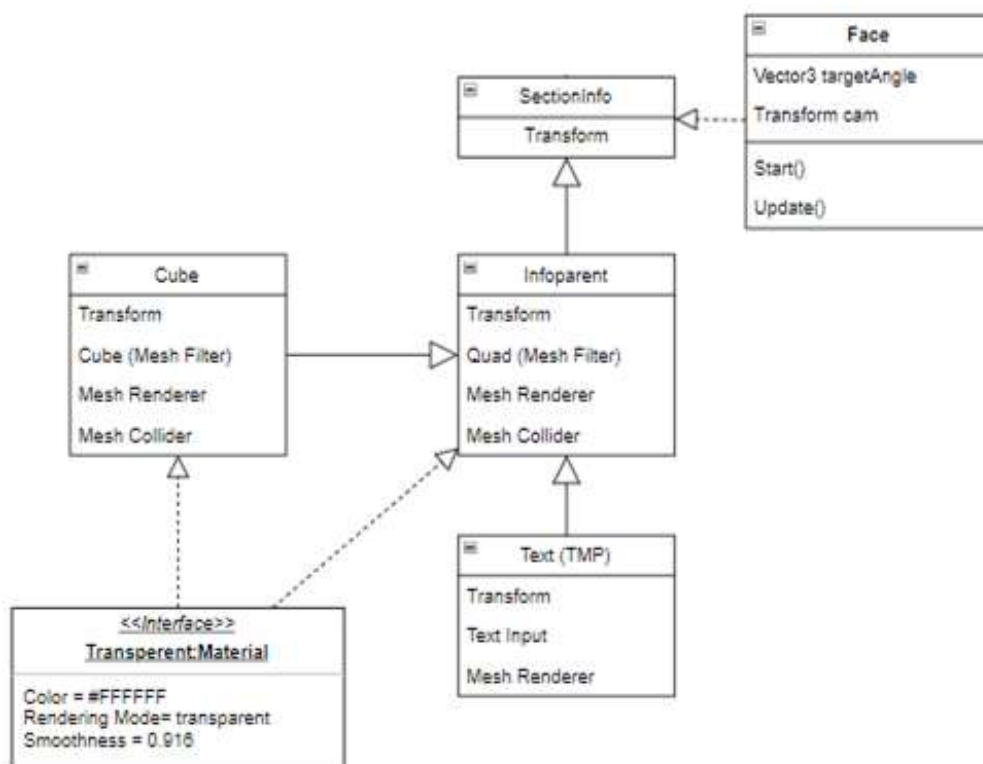


Рис.2.17 Архітектура класу «SectionInfo»

Клас «Section Info» імплементує скрипт «Face» для постійного відображення анотації «обличчям» до користувача та має клас нащадок «InfoParent», що являє собою блок анотацій. Класи «Cube» та «Text(TMP)» наслідують клас «InfoParent».

Для відображення кольору на об'єктах анотації створено інтерфейс матеріал і імплементовано його в класи «InfoParent» та «Cube», поля «Color», «Rendering Mode» та «Smoothness» в інтерфейсі матеріалу змінено для задання темного, прозорого кольору.

Висновки до другого розділу.

У другому розділі розглянуто середовища розробки, які можна використати для створення незалежного застосунку для зменшення часу на виконання складальних операцій. Проведено порівняння двох варіантів: Unity та Unreal Engine. Оглянувши та порівнявши функції та сегменти для кожного з середовищ розробки, а саме мова програмування, можливості візуального програмування, платформи розробки та можливості плагінів для анімації об'єктів обрано систему Unity для подальшого використання. Розроблено алгоритм роботи додатку, побудовано блоки схем для функцій повної анімації складання і розборки двоступеневого циклоїдального редуктора по натисканню кнопки, покрокова анімація для огляду кожної дії при складанні окремо. Створено архітектуру побудови для майбутнього застосунку, яка складається із ... Для відображення додаткової інформації про кожну деталь під час процесу складання, запропоновано додати функціонал анотації 3D моделей.

3.1 Розробка функціоналу застосунку

3.1.1 Підготовка моделі складальної одиниці

Перед початком роботи в застосунку «Unity» необхідно підготувати модель циклоїдального редуктора. Така необхідність виникає через те, що середовище розробки не підтримує форматів файлів, що використовує програма «Solidworks», в якій створено модель циклоїдального редуктора. «Unity» працює з полігональними 3D моделями. У тривимірній комп'ютерній графіці багатокутне моделювання (полігональне) (рис.3.1) – це підхід для моделювання об'єктів шляхом подання або апроксимації їх поверхонь з використанням багатокутників. Багатокутне моделювання добре підходить для рендерингу і тому є методом, який найчастіше використовується для комп'ютерної графіки в реальному часі.

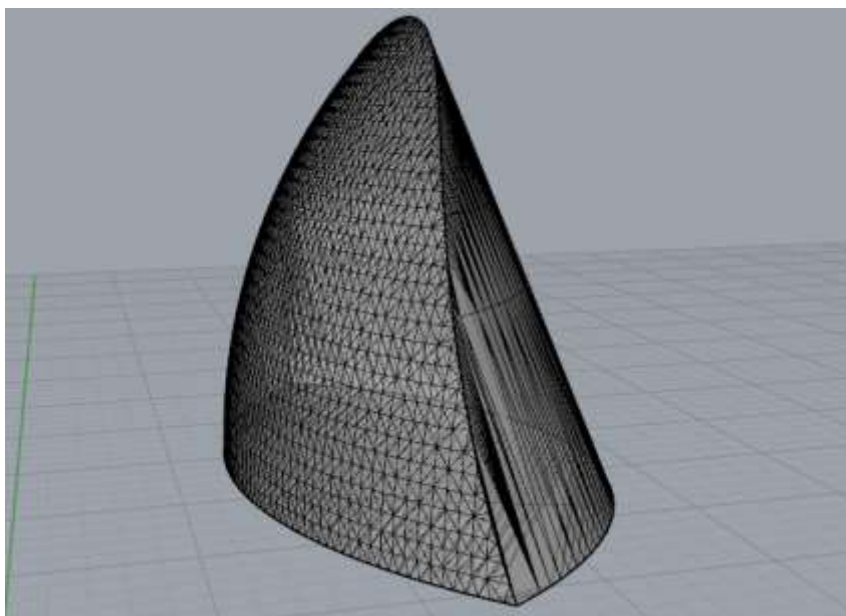


Рис.3.1 Модель побудова полігональним моделюванням[30]

Основним об'єктом, що використовується в моделюванні сітки, є вершина, точка в тривимірному просторі. Дві вершини, з'єднані прямою, стають ребром. Три вершини, пов'язані одна з одною трьома ребрами, визначають трикутник, який є найпростішим багатокутником в евклідовому просторі. Більш складні багатокутники можуть бути створені з кількох трикутників або як один об'єкт з більш ніж трьома вершинами. Чотиристоронні багатокутники і трикутники є найбільш поширеними формами, які використовуються при багатокутному моделюванні.

Група багатокутників, що пов'язані один з одним спільними вершинами, зазвичай називається елементом. Кожен з багатокутників, що становлять елементи, називаються межею. Для зберігання даних 3D-багатокутників є безліч форматів, обрано формат файлу .fbx. Для того щоб зробити процес конвертування моделей у полігональний формат максимально швидким, використано програму Analysis Situs (рис.3.4) з функціоналом, що дозволяє конвертувати формат файлу .stp, який може зберегти файли 3D моделей деталей з «SolidWorks» в форматі .stp, в формат файлу .fbx.

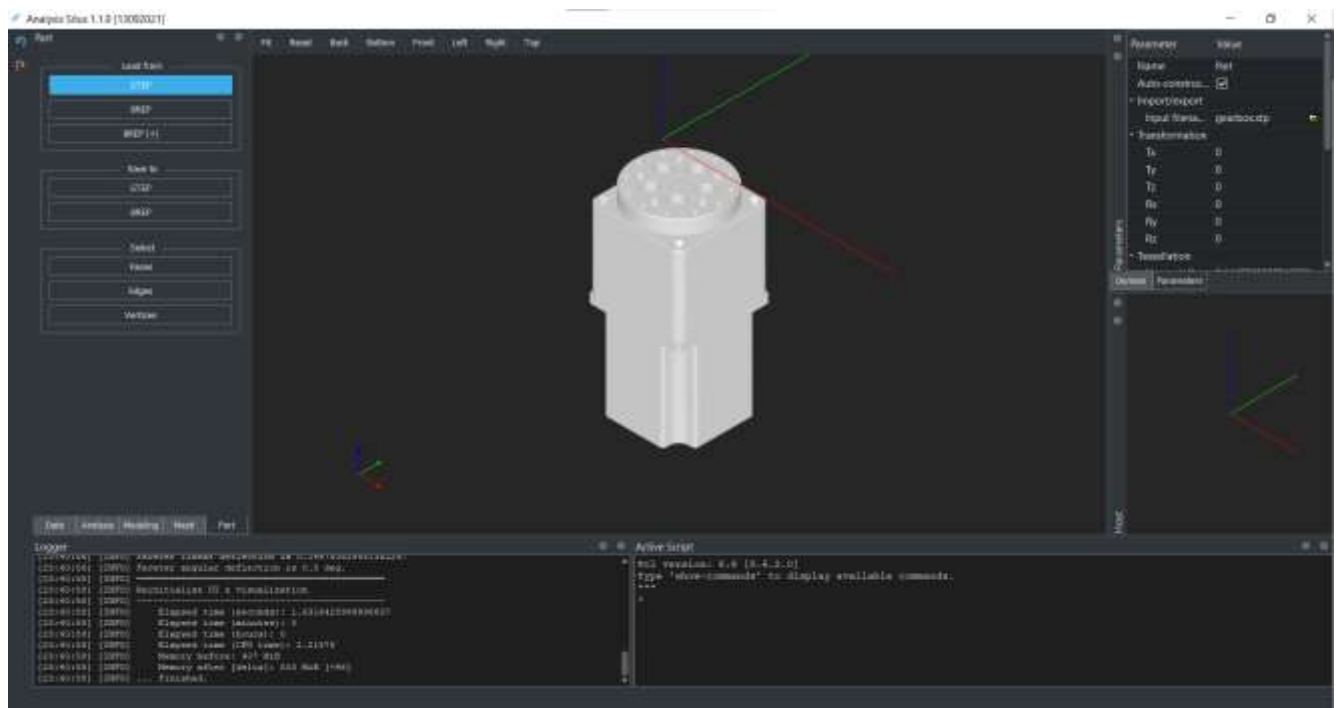


Рис.3.4 Модель з розширенням .stp завантажена в застосунок Analysis Situs

Для використання даного інструменту необхідно використано скрипт для виклику методів та функцій програми. Скрипт прописано у вікні «Active Script» (рис.3.5).

```
Active Script
Tcl version: 8.6 [8.6.2.0]
Type 'show-commands' to display available commands.
***
> asm-xde-load -model GearBox -filename "D:/test/gearbox.stp"
> asm-xde-browse -model GearBox
> asm-xde-generate-facets -model GearBox -normal|
> asm-xde-save-fbx -model GearBox -filename D:/test/MyNewFbxModel.fbx
>
```

Рис.3.5 Прописаний скрипт для конвертації формату моделі з .stp в .fbx

Спершу необхідно відкрити модель, для цього було викликано команду «load», створено об'єкт під назвою «GeARBox» та вказаний шлях до файлу:

```
> asm-xde-load -model GeARBox -filename "D:/test/gearbox.stp"
```

Наступним кроком виконано перевірку чи правильно було імпортовано модель, чи зберіглась структура зборки, оскільки нам важлива наявність окремих складальних одиниць в циклоїдальному редукторі для подальших дій з файлом. Для перевірки викликано команду «browse» та вказано назву об'єкту «GearBox», до якого звертається програма:

```
> asm-xde-browse -model GearBox
```

Далі було конвертовано геометрію моделі у полігональну 3D модель, тобто сітку утворену елементарними геометричними формами – трикутниками. Для перетворення використано команду «facets» та вказано назву об'єкту «GeARBox» до якого звертається скрипт. Також задано критерій щільності трикутників на моделі:

```
> asm-xde-generate-facets -model GearBox -normal
```

Останнім кроком викликано команду «save», вказано назву об'єкту «GearBox», до якого звертається скрипт та заданий шлях, де має бути збережена конвертована 3D модель зборки циклоїдального редуктора.

```
>asm-xde-save-fbx -model GeARBox -filename D:/test/MyNewFbxModel.fbx
```

Результатом роботи програми є файл з розширенням .fbx, який може бути імпортований в середовище розробки «Unity». У файлі збережено дерево побудови і усі складальні одиниці зборки, що дає можливість взаємодіяти з ними в наступних етапах роботи.

3.1.2 Створення проекту. Налаштування середовища розробки.

Для створення застосунку, що має відображати 3D модель циклоїдального редуктора, як зазначалося в розділі 2 даної роботи, обрано платформу «Unity».

Unity – багатоплатформовий інструмент для розробки відеоігор і застосунків, і двигун, на якому вони працюють. Створені за допомогою «Unity» програми працюють на настільних комп'ютерних системах, мобільних пристроях та гральних

консолях у дво- та тривимірній графіці, та на пристроях віртуальної чи доповненої реальності.

Для створення нового проекту створено аккаунт користувача та завантажено платформу «Unity Hub», де можна створити та редагувати проект. Вікно проекту середовища розробки (рис.3.6).налічує декілька вкладок з різним функціоналом для керування проектом.

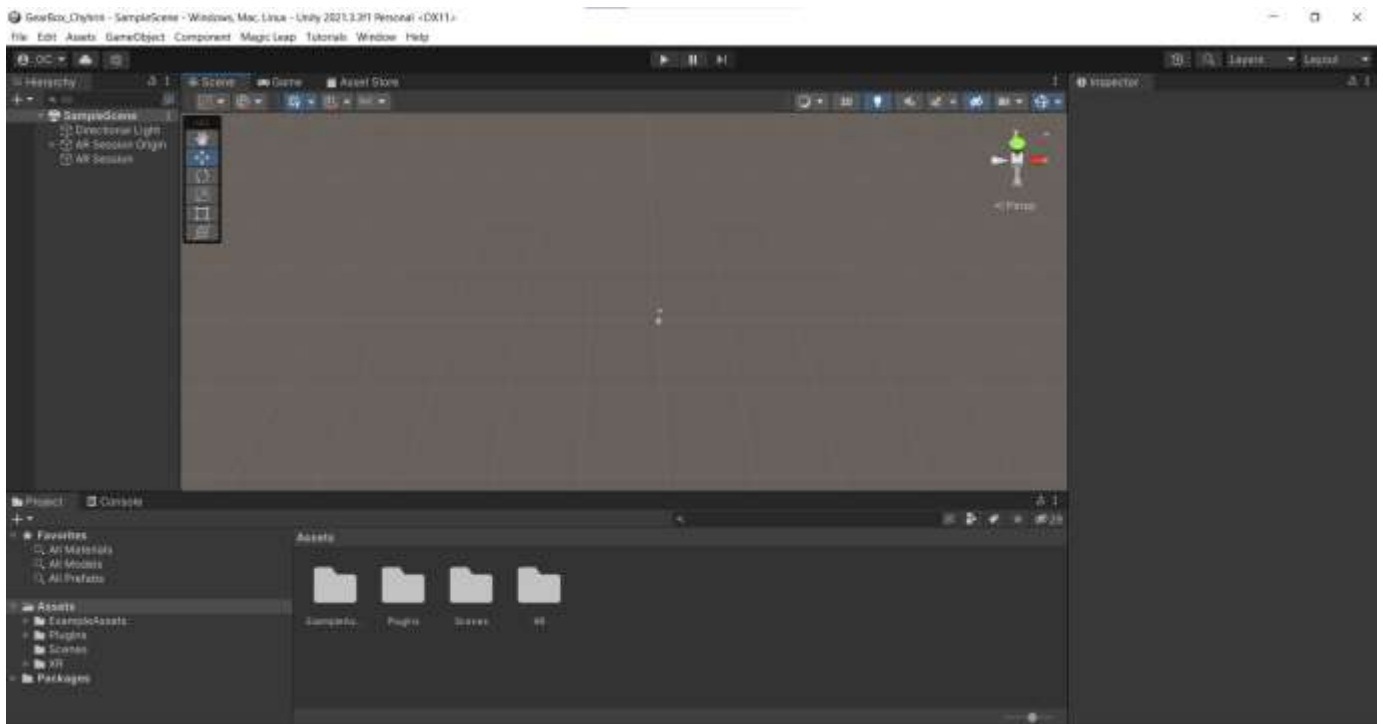


Рис.3.6 Вікно проекту у середовищі розробки «Unity»

Вкладка «Scene» є основним середовищем розробки, роботи в застосунку, у цьому просторі розміщуються усі об'єкти, з якими взаємодіє користувач застосунку. Як простір розробки воно має координатне поле. Зліва розміщена вкладка «Hierarchy», вона слугує текстовим відображенням усіх створених або імпортованих у програму об'єктів. Вкладки «Project» та «Console» розміщені знизу. Вікно «Project» слугує папкою усього проекту. Вкладка «Console» надає розробнику інформацію про процеси, що виконуються під час роботи/компіляції проекту. Розміщене справа вікно «Inspector» представляє інформацію про кожен об'єкт: його координати, можливі налаштування, імпортовані скрипти, що задають функціонал для об'єкту.

Операційну систему для смартфонів та планшетів – Android обрано, як платформу, на якій буде використовуватись програма, оскільки розповсюдженість

телефонів надає можливість масштабувати майбутній застосунок та зменшить час впровадження нової технології на виробництво.

Доповнена реальність додає цифрові елементи [31] до реального середовища, але спосіб розміщення елементів залежить від мети застосунку і середовища його використання. Два найпоширеніші типи доповненої реальності без маркерів і на основі маркерів. Доповнена реальність на основі маркерів використовує призначений маркер для активації застосунку. До популярних маркерів належать QR-коди доповненої реальності, логотипи або упаковка продукту. Форми або зображення мають бути чіткими та впізнаваними, щоб камера могла правильно ідентифікувати їх у різноманітному оточенні. Є ще один важливий фактор доповненої реальності на основі маркерів. Досвід доповненої реальності на основі маркерів прив'язаний до маркера. Це означає, що розміщення цифрових елементів залежить від розташування маркера. У більшості випадків об'єкт відображатиметься поверх маркера та рухатиметься разом із маркером, коли його повертатимуть або обертатимуть.

Доповнена реальність без маркерів не використовує маркер. Замість маркера роботу застосунку розпочинає користувач, вибравши функцію ДР у програмі чи на веб-сайті. Доповнена реальність без маркерів сканує реальне оточення та розміщує цифрові елементи на впізнаваному об'єкті, як-от плоскій поверхні. Отже, замість прив'язки до маркера цифрові елементи розміщуються на основі геометрії.

У розробленому застосунку обрано використовувати доповнену реальність на основі маркерів. Рішення обґрунтовано тим, що мета застосунку надати робітнику інформацію, щодо його завдання, у максимально швидкому форматі анімації у зручному методі використання розширеної реальності. На виробництві робітники постійно працюють з кресленнями, що зараз використовуються як джерело усієї необхідної інформації, тому доцільно було б використати креслення як маркер для застосунку. Розпізнавши такий, програма отримувала б інформацію щодо змінного завдання робітника і відображала наступні дії, які потрібно виконати працівнику. Питання безпеки також враховується, оскільки маркер на кресленні унеможливить використання застосунку окрім як на робочому місці за столом, де працівнику нічого не загрожує і він може зосередити свою увагу на екрані телефону або планшету.

Застосунок без використання маркерів міг би використовуватись будь де, тобто в потенційно небезпечних зонах.

Для створення застосунку з використанням доданої реальності вирішено використовувати інструмент під назвою «Vuforia».

«Vuforia» – це платформа доповненої реальності та інструментарій для розробки програмного забезпечення доповненої реальності SDK для мобільних пристроїв. «Vuforia» підтримує різноманітні типи цілей, включаючи цільове зображення без маркерів, тривимірні багатоцільові цілі та маркери посилення, які виділяють об'єкти у просторі для розпізнавання. Додаткові можливості дозволяють уникнути ефекту маскування (оклюзії) об'єктів за допомогою так званих «Віртуальних кнопок», забезпечують виділення об'єктів і можливість програмного створення і переналагоджування їх в рамках самостійної модифікації. «Vuforia» надає інтерфейси прикладного програмування на C++, Java, Objective-C і .Net через інтеграцію з ігровим рушієм «Unity» (рис.3.5).

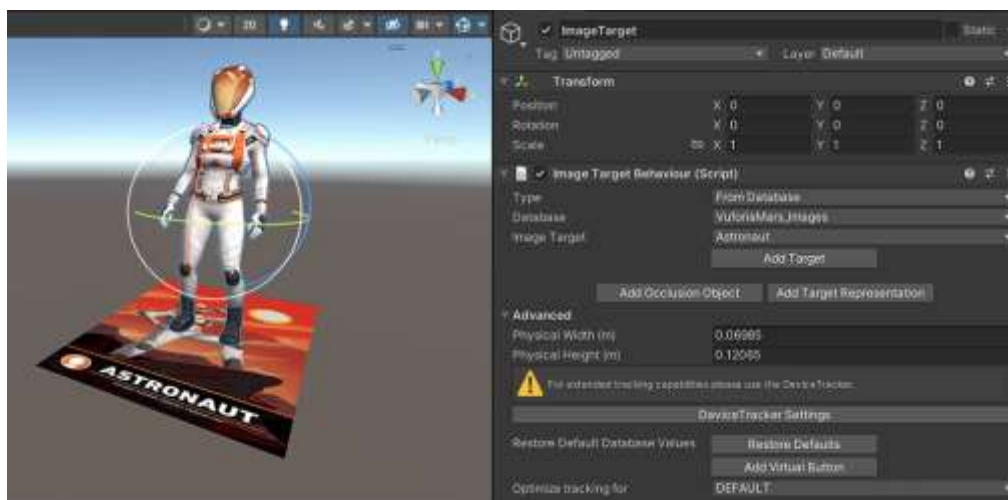


Рис.3.5 Стандартний об'єкт «Image target» створений у «Unity» за допомогою Vuforia

Таким чином, SDK підтримує розробку нативних ДР-додатків для iOS і Android, одночасно припускаючи розробку в «Unity», результати якої можна легко перенести на обидві платформи. Додатки доповненої реальності, які побудовані на платформі «Vuforia», сумісні з широким спектром пристроїв, включаючи iPhone, iPad, Android

смартфони і планшети з версією 2.2 і процесором, починаючи з архітектури ARMv6 або 7 з можливістю виконання обчислень з плаваючою комою.

Для того, щоб імпортувати інструмент в середовище розробки «Unity» завантажено необхідний пакет даних та верифіковано ліцензійний ключ.

Наступним кроком створено об'єкт під назвою «Image Target». «Image Targets» представляють зображення, які «Vuforia Engine» може виявляти та відстежувати. Механізм виявляє та відстежує зображення, порівнюючи витягнуті природні особливості із зображення камери з відомою цільовою базою даних ресурсів. Після виявлення цільового зображення «Vuforia Engine» відстежуватиме зображення та безперебійно доповнюватиме вміст, використовуючи відстеження зображень. Розпізнавши «Image Target» об'єкт, програма відображає запрограмовані об'єкти або 3D моделі з можливістю виконувати дії над цими об'єктами.

Для того, щоб можна було завантажити зображення (рис. 3.6), яке в майбутньому буде використовуватись як маркер у застосунку, його необхідно додати у базу даних інструменту «Vuforia». Необхідно завантажити зображення у форматі .jpg та вказати ширину зображення у реальному світі, оскільки в подальшому застосунок буде «шукати» камерою обране зображення. «Target Manager» обробляє зображення, щоб створити як дані, так і візуальні представлення – «target features» (рис.3.7). Це також дозволяє переглядати очікуваний рейтинг виявлення та

Gearbox

Edit Name Remove



Рис.3.6 Завантажене в Target Manager зображення

Gearbox

Edit Name Remove



Рис.3.7 Feature points завантаженого зображення

відстеження цілі. Після цього «Image Targets» можна завантажити як пакет, придатний для інтеграції в «Unity». Інструмент деякий час обробляє фотографію і додає її в базу даних.

Базу даних, що містить інформацію про деталь імпортовано в середовище розробки «Unity». Наступним кроком створено об'єкт «Image Target» в робочому просторі (рис 3.8).

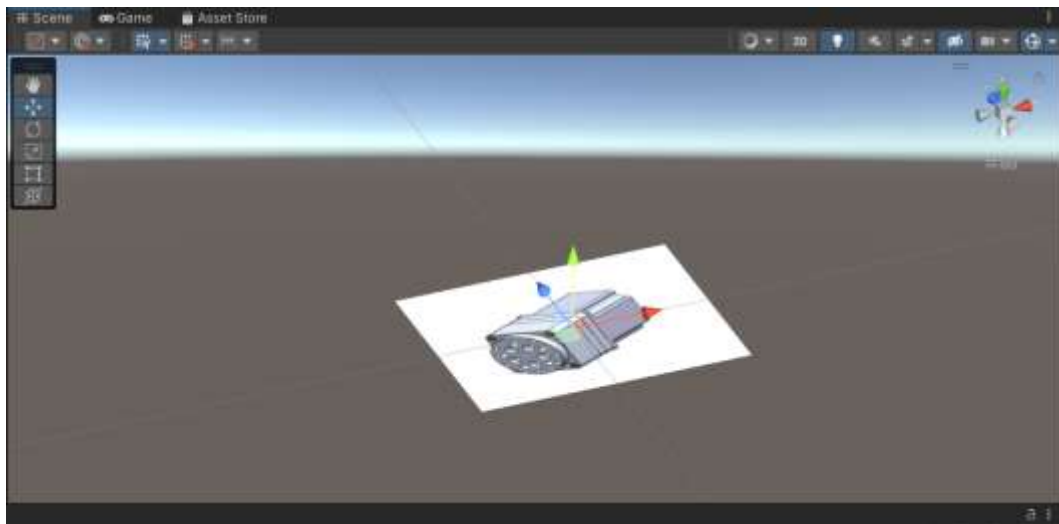


Рис.3.8 Зображення-маркер у робочому просторі

Для того, щоб застосунок мав можливість відобразити модель саме на маркері, додано 3D модель (рис.3.9), як об'єкт нащадок до об'єкту «Image Target».

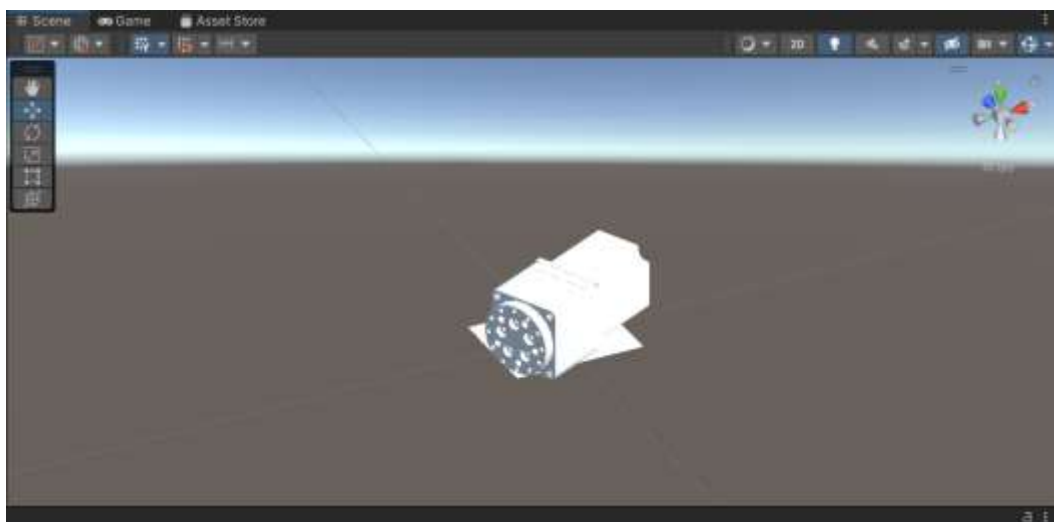


Рис.3.9 3D модель розміщена на маркері в робочому просторі

На даному етапі проекту отримано можливість позиціонувати 3D моделі циклоїдального редуктора у просторі за допомогою створеного маркера.

3.1.3 Створення анімації зборки і розборки циклоїдального редуктора

Для спрощення взаємодії з застосунком запропоновано створити можливість збирати і розбирати 3D модель натисканням однієї кнопки. Для цього створено цикл з анімацій, що виконуються з запрограмованим порядком і швидкістю та виконуються за умови активності додатково створеного тригера, що контролює користувач.

Для створення анімацій в середовищі розробки «Unity» створено «Animator» (рис. 3.10). Інструмент «Animator» потрібен для застосування анімації до об'єктів, що

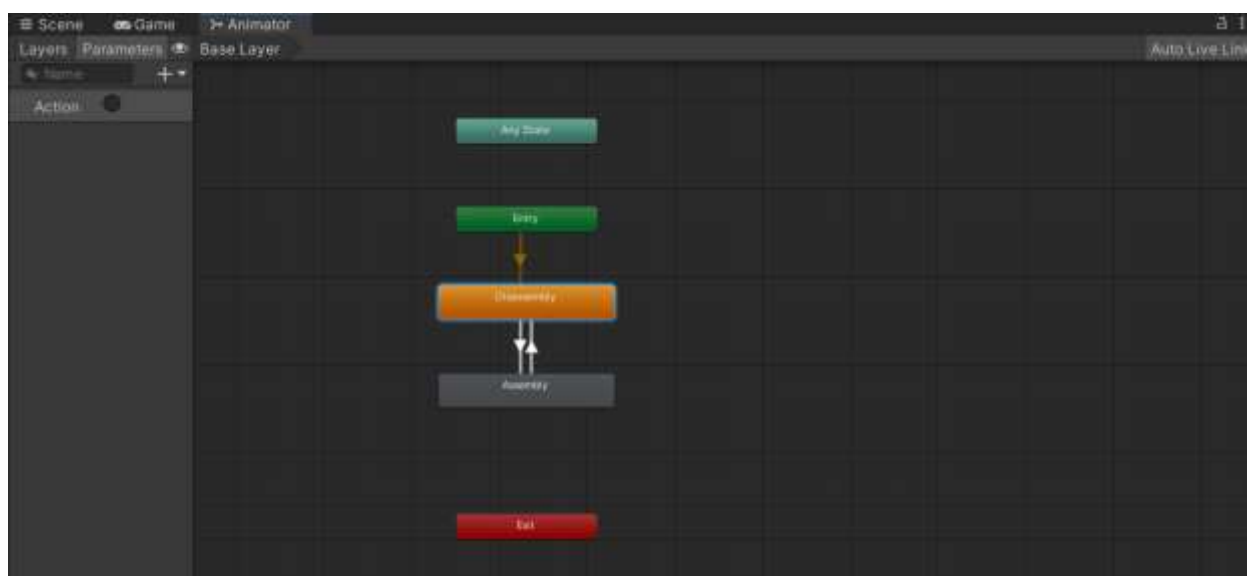


Рис.3.10 Блок схема циклу роботи базового функціоналу застосунку

знаходяться у проекті. Контролер аніматора дозволяє керувати набором анімацій для об'єкту і перемикатися між ними при виконанні умови. Контролер контролює переходи між анімаціями, використовуючи так звану «State machine», свого роду програму, написану візуальною мовою програмування в «Unity».

Як базовий функціонал запропоновано виконати анімацію розборки та складання 3D моделі циклоїдального редуктора (рис.3.11). Під базовим функціоналом застосунку розуміється можливість відобразити 3D модель циклоїдального редуктора у доданій реальності, а також можливість переглянути анімацію складання виробу.

У вікні інструменту «Animator» створено два об'єкта анімації під назвами «Assembly» та «Disassembly» для роботи програми запрограмовано цикл, в якому

модель змінює стан з розібраного в зібраний. Створено «trigger» під назвою «Action», за якої має відбуватись перехід від одного стану моделі до іншого. Для можливості в майбутньому керувати анімаціями в налаштуваннях циклу додано умову переходу, а саме активний/пасивний стан триггеру «Action».

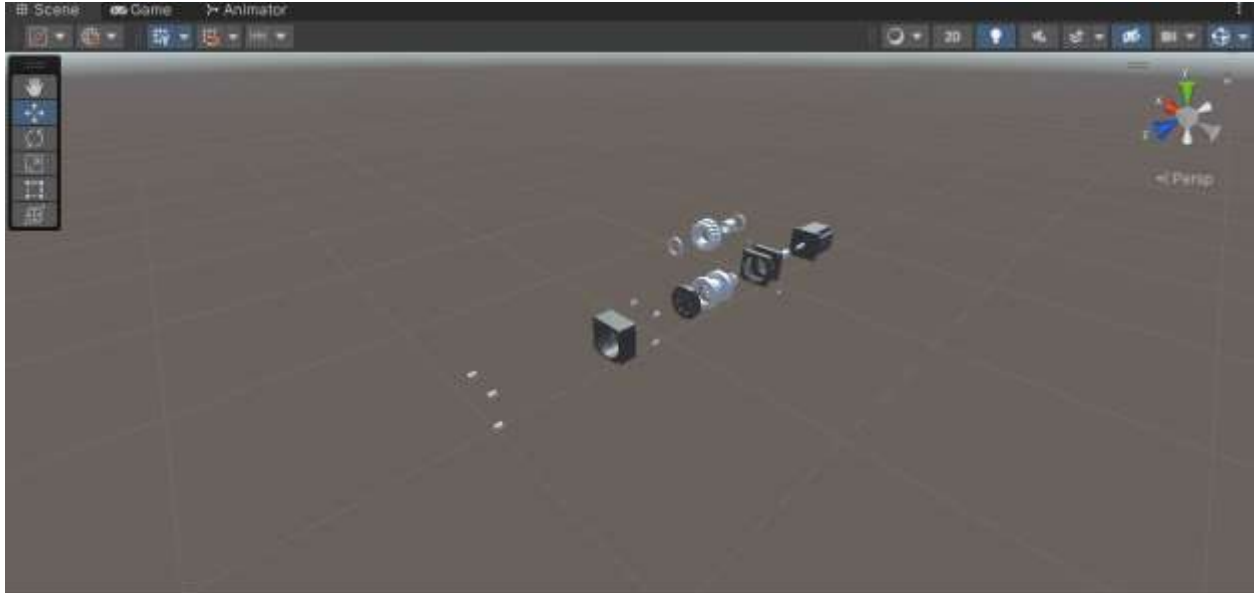


Рис.3.11 Розібрана 3D модель циклоїдального редуктора

Для керування анімацією створено інструмент «Animation». Для кожної складальної одиниці створено «Key». Елемент «Key» – це відображення, дублікат кожного об’єкту, який буде брати участь в анімації в застосунку. Переміщуючи «Key» у тайм-лайн є можливість задавати необхідні координати, кути повороту, зміни кольорів та матеріал об’єктів в певний момент часу.

Для анімації розборки циклоїдального редуктора виконано переміщення усіх відображень об’єктів у рядку їх тайм-лайну (рис.3.12) та задані положення

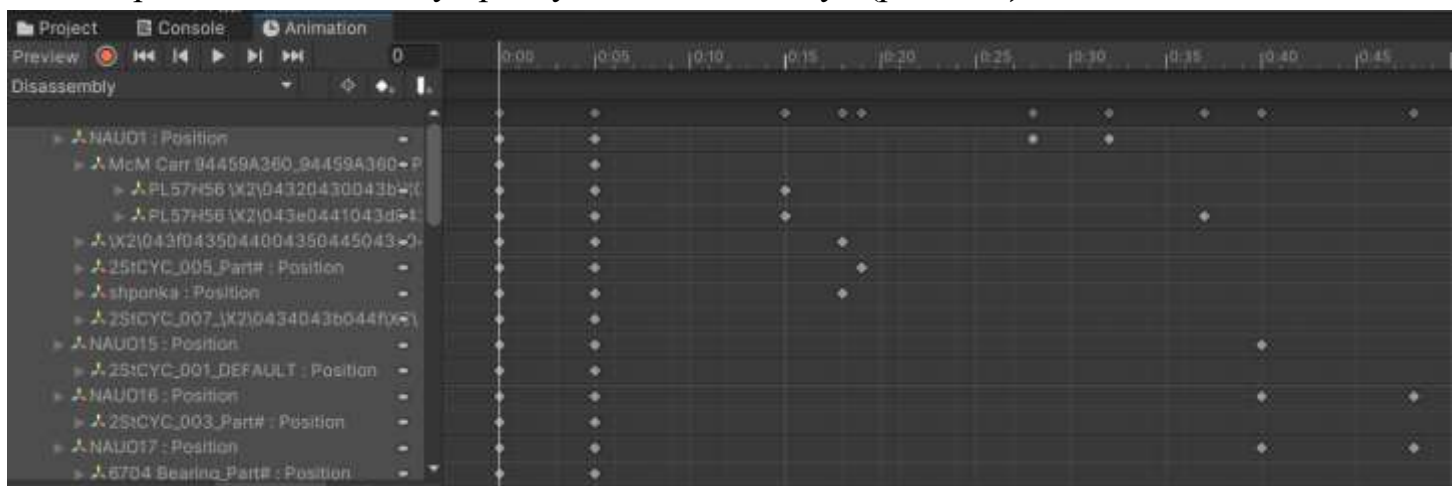


Рис.3.12 Тайм-лайн анімації зборки 3D моделі циклоїдального редуктора

(координати) об'єктів у відносній системі координат, де точка O ($x = 0, y = 0, z = 0$) – це маркер, на якому з'являється 3D модель циклоїдального редуктора.

Для анімації складання необхідно розмістити усі елементи «Key» кожного з об'єктів в іншому порядку, розпочинаючи з тих самих положень, на яких розташовувались. Розташувавши усі 3D моделі складальних одиниць циклоїдального редуктора в необхідних початкових координатах та надавши кожному елементу «Key» логічне місце в тайм-лайні проекту, отримано анімацію складання (рис.3.13).

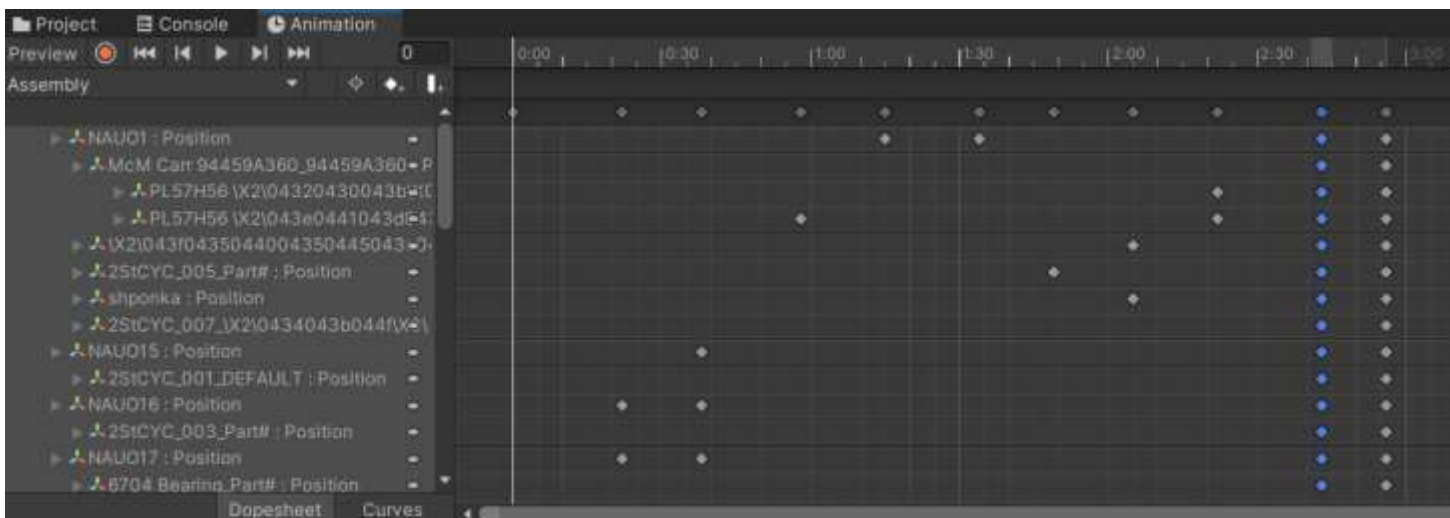


Рис.3. 13 Тайм-лайн анімації зборки 3D моделі циклоїдального редуктора

Для керування анімаціями складання і розборки 3D моделі циклоїдального редуктора, у робочому просторі створено елемент «Button». У папці проекту створено елемент – скрипт. Скрипти є важливою складовою всіх програм, які створює розробник в «Unity». Для більшості програм скрипти можна використовувати для створення графічних ефектів, керування фізичною поведінкою об'єктів. Скрипти в середовищі розробки «Unity» прописувалися мовою програмування C#. Для редагування коду обрано середовище розробки Visual Studio. У випадку розглядуваного застосунку, кнопка буде запускати скрипт, що буде змінювати стан триггеру «Action», що був створений під час побудови блок схеми анімації.

Скрипт Animation [Додаток Д], скрипт, що керує анімаціями в додатку
`public class Animation : MonoBehaviour`
{

```

private Animator anim;
public bool check;
void Start()
{
    anim = GetComponent<Animator>();
}

public void Animate ()
{
    anim = GetComponent<Animator>();
    if (!check)
    {
        anim.SetBool("Action", true);
        check = true;
    }
    else
    {
        anim.SetBool("Action", false);
        check = false;
    }
}
}

```

У скрипті створено клас «Animation», що є нащадком базового класу «MonoBehavior», що створений програмою автоматично. Створено булеву змінну «check», для подальшого використання у перевірці стану триггеру. У функції Animate створено екземпляр класу Animator, завдяки цьому отримано можливість звертатись до раніше створеного об'єкту триггеру «Action». Наступним кроком виконується перевірка стану змінної, що дублює стан триггеру, за умови значення триггеру false, значення триггеру разом зі змінною check змінюється на true, в іншому випадку навпаки, змінні набувають значення false.

3.1.4 Створення покрокової анімації зборки і розборки циклоїдального редуктора.

Отримавши анімацію розборки і зборки циклоїдального редуктора запропоновано розробити покрокову версію, де користувач має можливість переміщуватись вперед і назад по етапах складання, що дасть змогу більш детально

оглянути кожну необхідну дію, яку необхідно виконати під час складання кожного з вузлів.

Для розширення можливостей застосунку виконано розгалуження алгоритму роботи застосунку(рис.3.14), а саме додано можливість змінювати стан 3D моделі зібраного в розібраний іншим, покроковим шляхом. Для цього створено певну кількість окремих анімацій. При цьому необхідно звернути увагу не те, що положення 3D об'єктів у кінці кожної анімації мають співпадати із положеннями у наступній анімації. Це зробить анімації легшими для розуміння і робітникам не потрібно буде зосереджувати свою увагу на пошуку необхідних йому деталей на екрані.

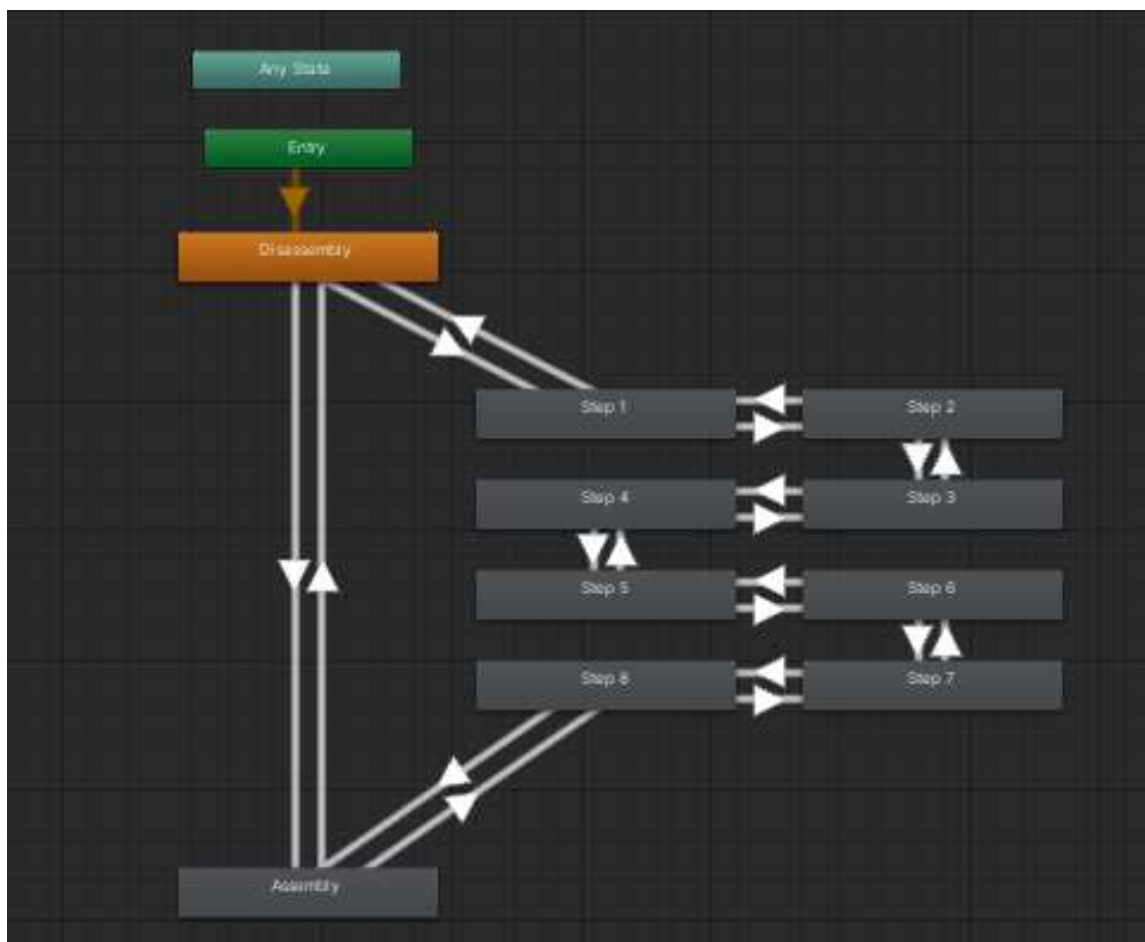


Рис.3. 14 Блок схема порядку анімацій у застосунку

Отримані блоки кроків анімації пов'язані між собою вузлами, у які закладено умову, за якої буде відбуватись перехід від одного блоку до іншого, тобто анімація конкретного кроку складання. Раніше використаний функціонал «Trigger» не доречний у цьому випадку. Необхідно контролювати перехід кроків у будь якому порядку, оскільки користувач повинен мати можливість передивитись один крок

декілька разів, за такої умови використання функціоналу об'єктів тригерів значно ускладнює програму і її логіку.

Для реалізації безперешкодної зміни анімацій у довільному порядку створено змінну типу `int` під назвою «step», а як умову для зміни блоків анімацій вирішено зробити порівняння значення цієї змінної із номером кроку, який має відбутись по натисканню кнопки. Тобто для переходу до кроку номер 2 змінна має набутися значення 2, відповідно якщо користувачу необхідно повернутись на крок (рис.3.15) назад зміна набуває значення на один менше, а саме 1. Таким чином для реалізації цього функціоналу достатньо створити лише одну змінну і запрограмувати умови для переходів.

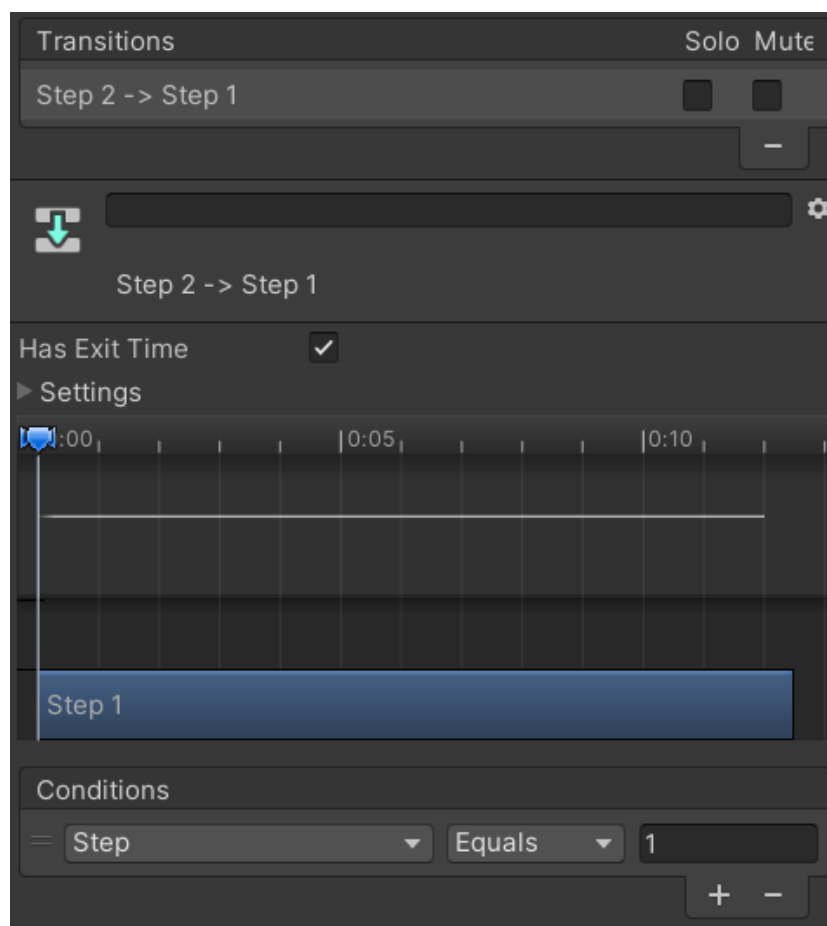


Рис.3. 15 Налаштування умови переходу від кроку №2 до кроку №1

Оскільки створені переходи автоматично налаштовані відігравати другу половину попередньої анімації і першу половину наступної (рис.3.16), зміщено

повзунок часу програвання цих анімацій у точку 0.00 (рис.3.17). Таким чином анімація відіграється лише один раз, як запрограмовано у майбутніх кроках.

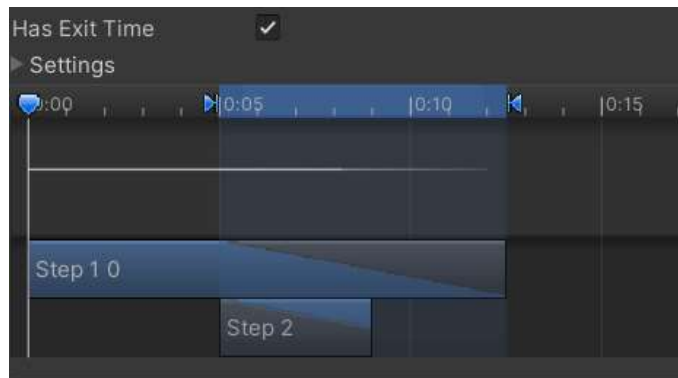


Рис.3. 16 Неналаштований таймлайн анімації

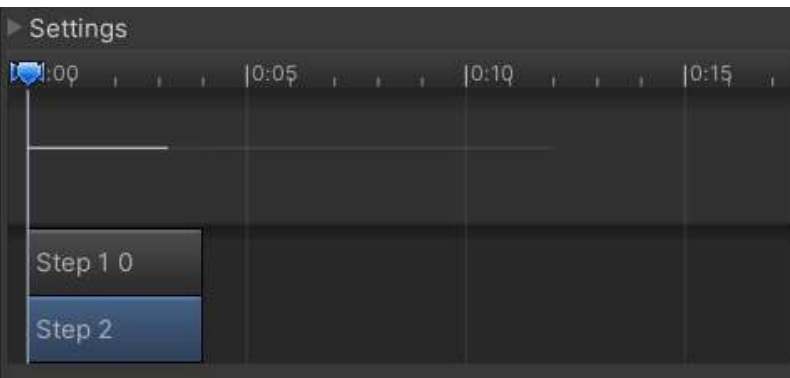


Рис.3. 17 Налаштований таймлайн анімації

Для керування анімаціями створено дві кнопки: попередня і, відповідно, наступна анімація (рис.3.18).

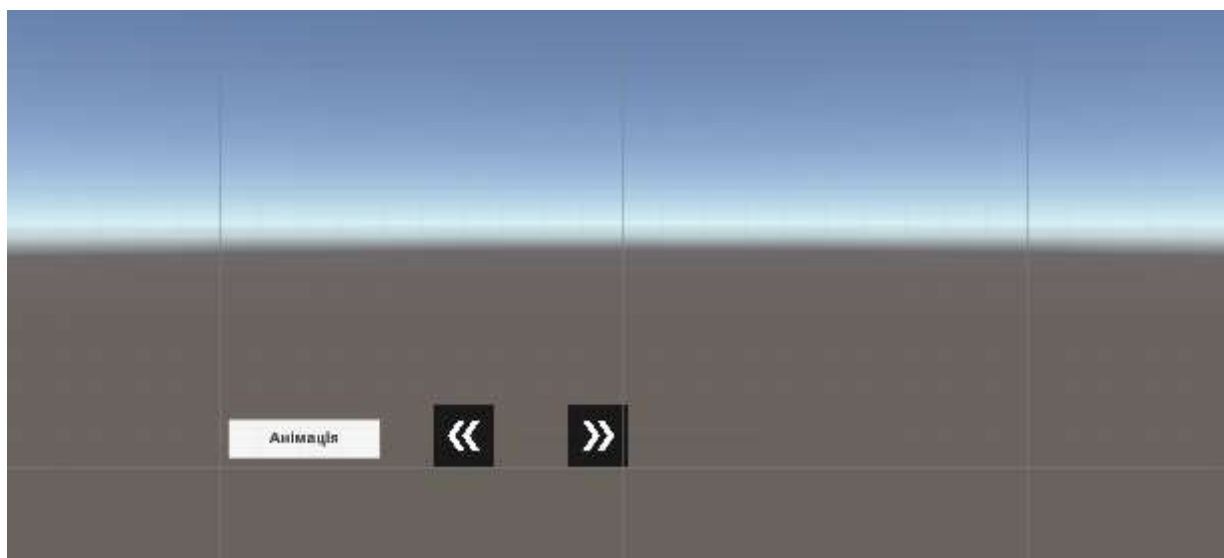


Рис.3. 18 Кнопки застосунку для керування анімаціями

Керуючий скрипт, який виконує усі зміни, для переходу блоків анімацій можна імплементувати у вже існуючий скрипт, а саме код, що керує базовими анімаціями складання і розборки циклоїдального редуктора. Для програмування кнопок прописано дві функції, що змінюють змінну «step», яка порівнюється з вище описаними умовами на кожному кроці.

Функція «StepForward()» викликається кожен раз, коли буде натиснута кнопка «Наступний крок». У функції створюється екземпляр класу «Animator», який дає доступ до змінних, які раніше було створено у вікні «Animator». У коді додатково реалізована перевірка значення змінної «step», для уникнення ситуацій, коли значення змінної стає більшим за кількість існуючих кроків анімації.

Функція StepForward (). Скрипт Animation [Додаток Д], скрипт, що керує кнопкою «Наступна анімація»

```
public void StepForward ()
{
    anim = GetComponent<Animator>();
    if (anim.GetInteger("Step") < 9)
    {
        anim.SetInteger("Step", anim.GetInteger("Step") + 1);
    }
}
```

Відповідно у випадку, якщо змінна не більша за кількість прописаних кроків, тобто 9, до неї додається 1, що надалі зумовлює перехід до наступної анімації. У коді використовуються функції «GetInteger()» та «SetInteger()», завдяки яким є можливість отримати доступ до змінної «step» та відповідно змінити її значення.

Для реалізації переходу до попереднього блоку анімації, прописано функцію «StepBack()», що має аналогічну логіку, тільки у випадку виконання усіх умов, значення змінної «step» зменшується на 1, тобто відбувається перехід до попередньої анімації.

Функція StepBack(). Скрипт Animation [Додаток Д], скрипт, що керує кнопкою «Попередня анімація»

```
public void StepBack()
{
    anim = GetComponent<Animator>();
    if (anim.GetInteger("Step") > 0)
    {
        anim.SetInteger("Step", anim.GetInteger("Step") - 1);
    }
}
```

```
}  
}
```

На даному етапі, у випадку, якщо користувач натисне кнопку анімації знаходячись на будь якому кроці лічильник «step» не змінює свого значення, що робить алгоритм роботи нелогічним. У такому випадку користувачеві необхідно повернути це значення до 0 або 9 залежно від складеного чи розібраного стану 3D моделі циклоїдального редуктора. Для вирішення цього недоліку необхідно додати декілька строчок коду у минулі скрипти анімації.

Функції Start() та Animate (). Скрипт Animation [Додаток Д], скрипт, що керує анімаціями в додатку.

```
void Start()
```

```
{  
    anim = GetComponent<Animator>();  
    anim.SetInteger("Step", 0);  
}
```

```
public void Animate ()
```

```
{  
    anim = GetComponent<Animator>();  
    if (!check)  
    {  
        anim.SetBool("Action", true);  
        check = true;  
        anim.SetInteger("Step", 0);  
    }  
    else  
    {  
        anim.SetBool("Action", false);  
        check = false;  
        anim.SetInteger("Step", 9);  
    }  
}
```

}

}

Функція «Start()» виконується автоматично один раз під час запуску застосунку. Для задання логічного початку роботи застосунку у функції «Start()» змінній «step» надане значення 0, інакше програма не має інформації, щодо цієї змінної і не може бути скомпільована. У функції «Animate ()» прописані процеси для анімацій складання і розборки 3D моделі, в блоці коду «if» описується поведінка змінних у розібраному стані, тому значення змінної «step» тут набуває 0, тобто початок складання. У блоці «else» описано поведінку змінних у зібраному стані, тому змінній «step» було надано значення 9, тобто повністю зібраний циклоїдальний редуктор.

Після отримання робочого функціоналу зміни блоків анімацій, розроблено анімацію для кожного кроку зборки та розборки циклоїдального редуктора. Для зручності запропоновано розробляти кроки анімацій починаючи із зібраного стану редуктора, тобто з 8-го кроку. Процес складання поділено на 8 кроків, оскільки 8 етапами можна чітко розділити процес складання одночасно не ускладнюючи його сприйняття і не створюючи забагато окремих блоків для етапів складання, що ускладнило б роботу з застосунком [Додаток Г].

Першим етапом згідно з технологічною схемою складання є фіксація перехідного (рис.3.19) валу (поз.8) з шпонкою 15x4x4 (поз.15) на валу двигуна Nema 23 двома установочними гвинтами M3x8 DIN 913 (поз.13).

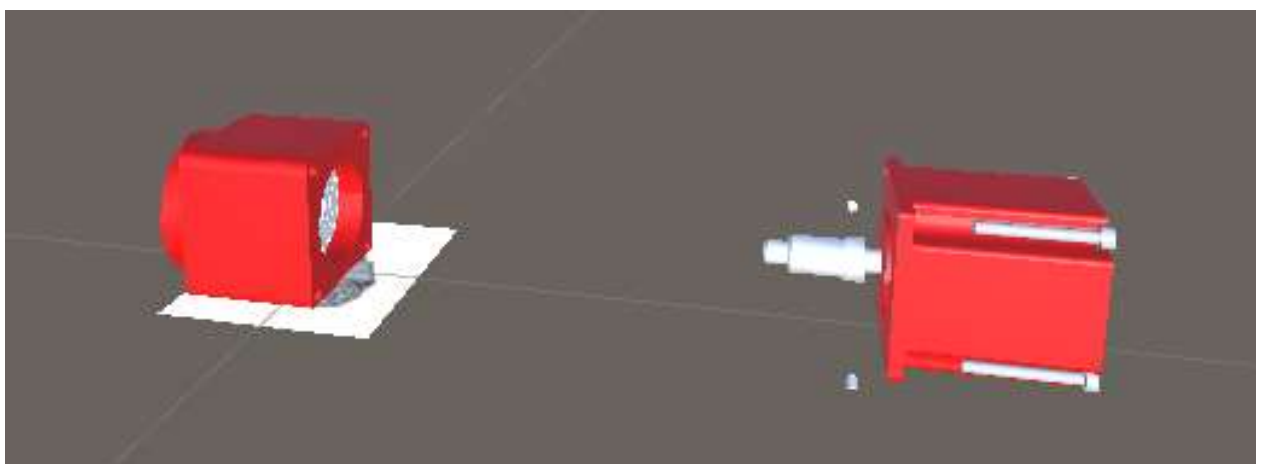


Рис.3.19 Установка перехідного валу на двигуні

Особливістю послідовних анімацій є збереження позицій об'єктів між суміжними кроками. Для збереження позиції кожного об'єкту на прямій часу (рис.3.20) створено об'єкт «Key», що зберігає інформацію про положення об'єкту в точці часу де він створений.

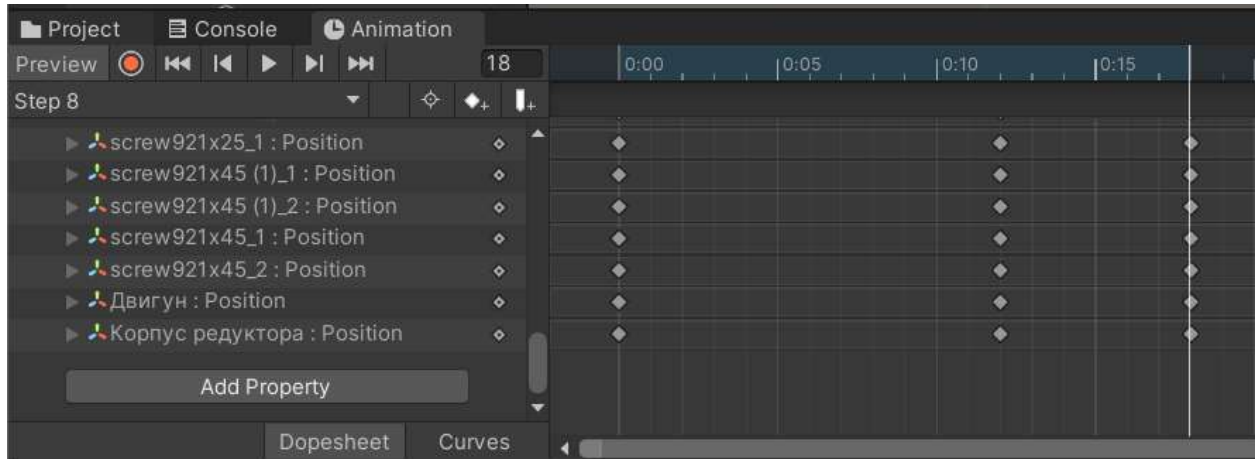


Рис.3.20 Пряма часу створена для 8 кроку

Наступний крок (рис.3.21) розборки редуктора відображає зняття перехідного валу з валу двигуна і зняття шпонки з перехідного валу.

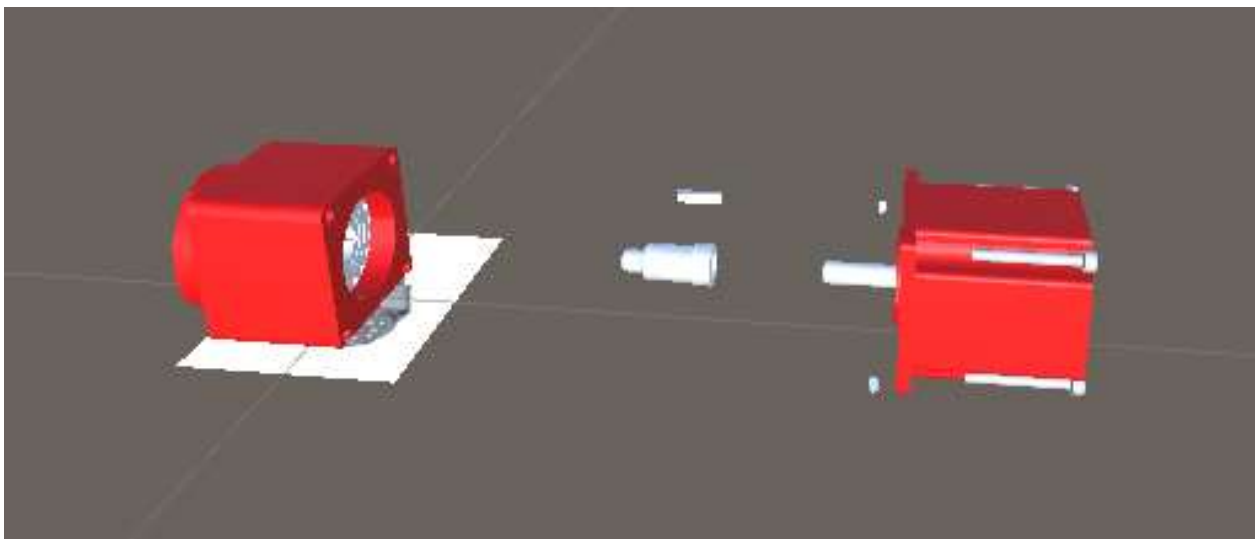


Рис.3.21 Сьомий крок зборки редуктора

Шостий крок відображає наступний етап складання. Для зручності відображення деталі дистанційна пластина (поз.7) і перша ступінь редуктора (поз.7) переміщено на задній план, оскільки з ними ніяких дій виконувати не потрібно. По завершенню переміщення зайвих об'єктів запускається анімація розборки Циклоїди

СК (рис.3.22), а саме ексцентрична втулка (поз.9) і два підшипника 6704ZZ виймаються з циклоїди (поз.4).

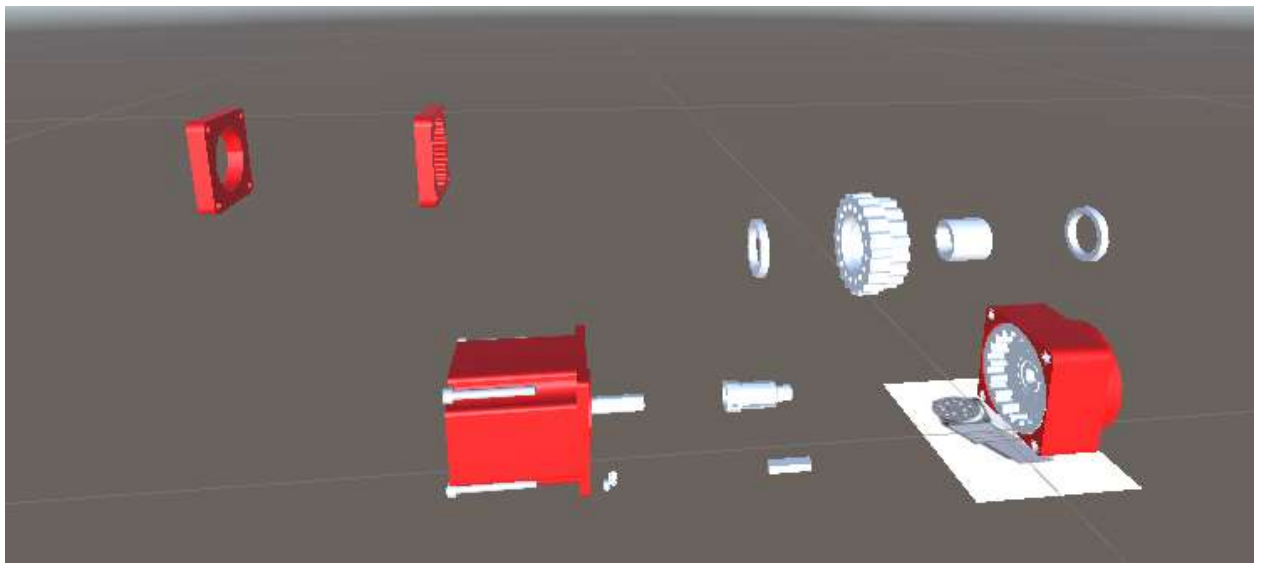


Рис.3.22 Розборка Циклоїди СК

Наступним, п'ятим (рис.3.23), блоком анімації є відокремлення ротора редуктора (поз.3) від Корпусу СК. Ротор фіксується 5 гвинтами M4x30 DIN 912 (поз.17), що закручуються у запаяні різьбові втулки (поз.16).

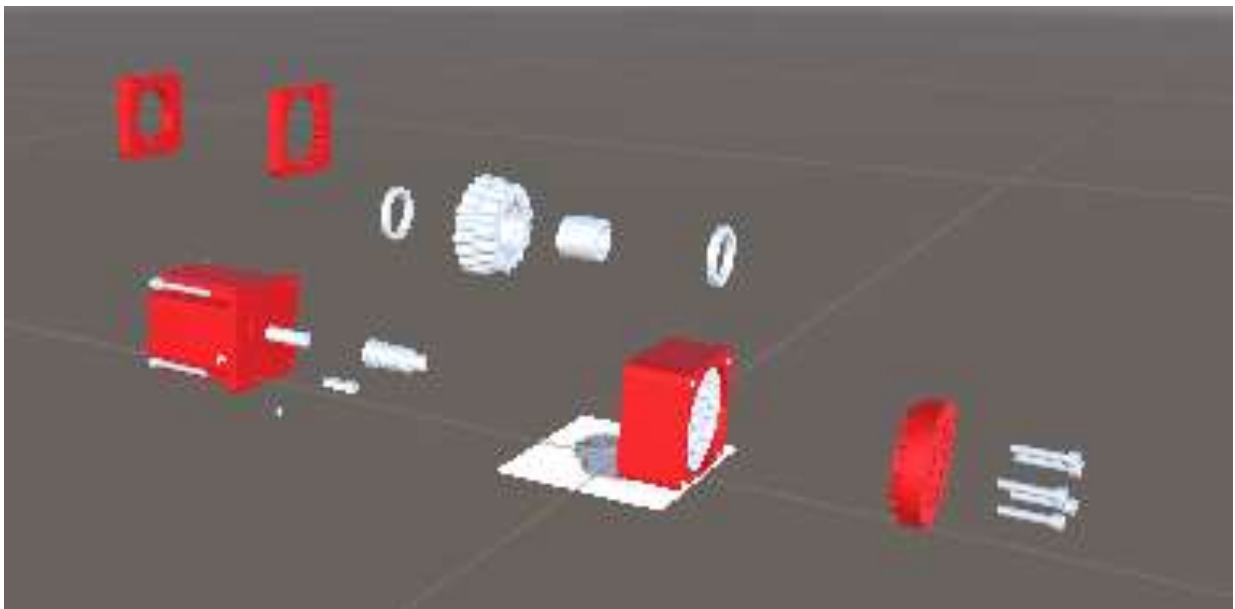


Рис.3.23 Відокремлення ротора редуктора

Четвертим блоком анімації є початок складання Корпус редуктора СК, а саме візуалізація впаювання різьбових втулок (поз.16) у корпус редуктора (поз.2). Для

виконання цієї операції необхідно використати додатковий інструмент – паяльник. 3D моделі корпусу і втулок винесено на передній план для наближення розташування як у реальному просторі цифрових 3D об'єктів.

Третій та другий блоки анімації відображають установку підшипників: підшипник 608ZZ (поз.11) та підшипник 61907 2Z (поз.10) установлюються на другу ступінь редуктора (поз.5). Останній етап анімації розборки редуктора – це запаювання 5-ти різьбових втулок у другу ступінь редуктора. Надалі у різьбові втулки будуть вкручуватись гвинти для фіксації ротора редуктора (рис.3.24).

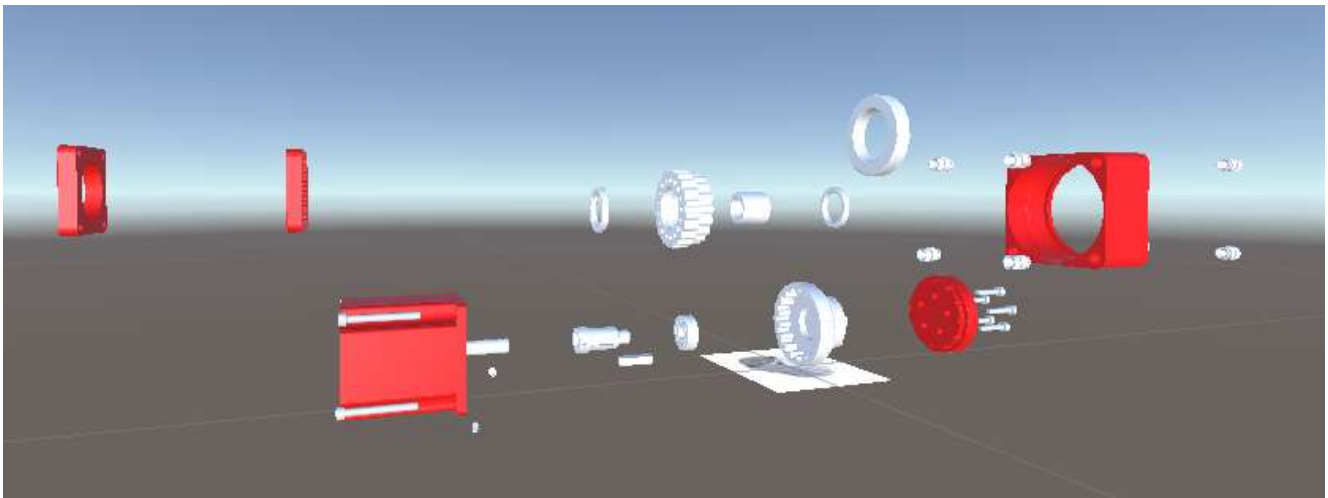


Рис.3.24 Останні три етапи анімацій

3.1.5 Анотування 3D моделей складальних одиниць циклоїдального редуктора

Під час виконання складальних робіт робітникам, що виконує змінне завдання, часто необхідно додатково дослідити інформацію про складальні одиниці, габаритні розміри деталей, квалітети точності або будь яка інша інформація, що буде корисною у складальному процесі. Було запропоновано розробити функціонал, що дозволив би за певної умови відображати раніше створені анотації 3D об'єктів.

Відтворення функціоналу анотування 3D моделей складається з чотирьох частин:

- Створення об'єктів, на яких буде відображена інформація про 3D модель;

- Написання програмного коду для анімації відображення раніше створених об'єктів анотації та функціоналу, що дозволить об'єктам блоку анотації завжди розташовуватися перпендикулярно до користувача;
- Написання програмного коду для забезпечення відображення анотацій тільки за певної умови, що залежить від дій користувача.
- Створення сітки для кожної 3D моделі для реалізації умови відображення анотації.

Перший пункт розробки функціоналу анотування починався зі створення пустого об'єкту «GameObject», цей елемент включає написаний скрипт для відображення блоку анотації. Новостворений об'єкт перейменовано на «SectionInfo», у ньому створено об'єкт нащадок «Quad» – площина для відображення тексту. Об'єкт було названо «InfoParent», він слугує батьківським класом для майбутньої анотації 3D моделі. Габарити створеної площини відредаговано для суміжних з 3D моделлю розмірів. Для відображення зв'язку між анотацією та 3D моделлю деталі, створено об'єкт нащадок «Cube», розміри кубу було змінено до прямокутника, що слугуватиме умовною підставкою до блоку з анотацією. Для кожного з об'єктів створено матеріал і задано сіру прозору структуру кольору, що надає ергономічності функціоналу (рис.3.25).

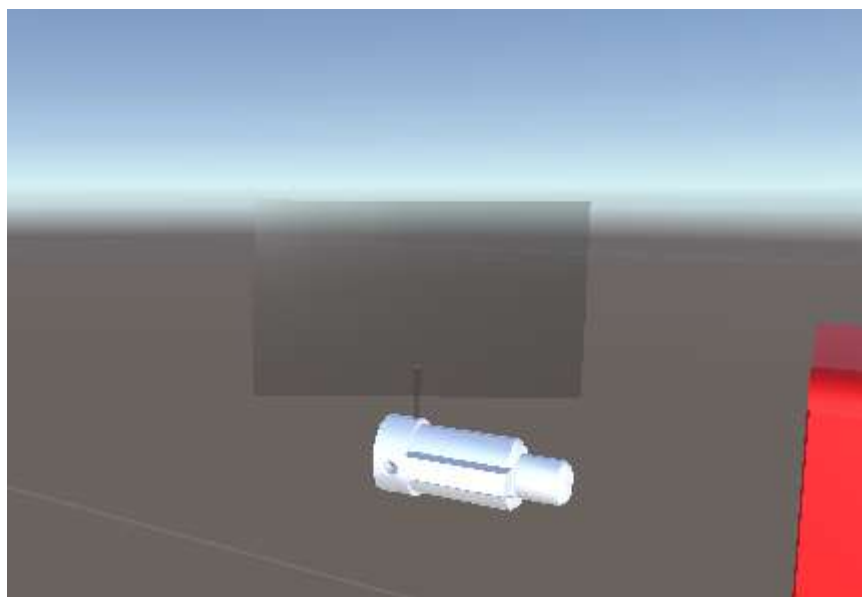


Рис.3.25 3D об'єкти для відображення тексту анотації

Для відображення тексту у блоку «Quad», створено текстовий блок «Text (TextMeshPro)». Для коректного відображення інформації у блоку увімкнено відображення «Gismos», цей режим використовуються для візуального налагодження або допомоги в налаштуваннях в робочому просторі. Поле відображення тексту було масштабовано до рамок створеного раніше об'єкту «Quad» (рис.3.26). Розмір тексту можливо автоматично підібрати інструментом «Autosize», завдяки такому інструменту надалі текст буде масштабуватись відповідно раніше вказаних полів.

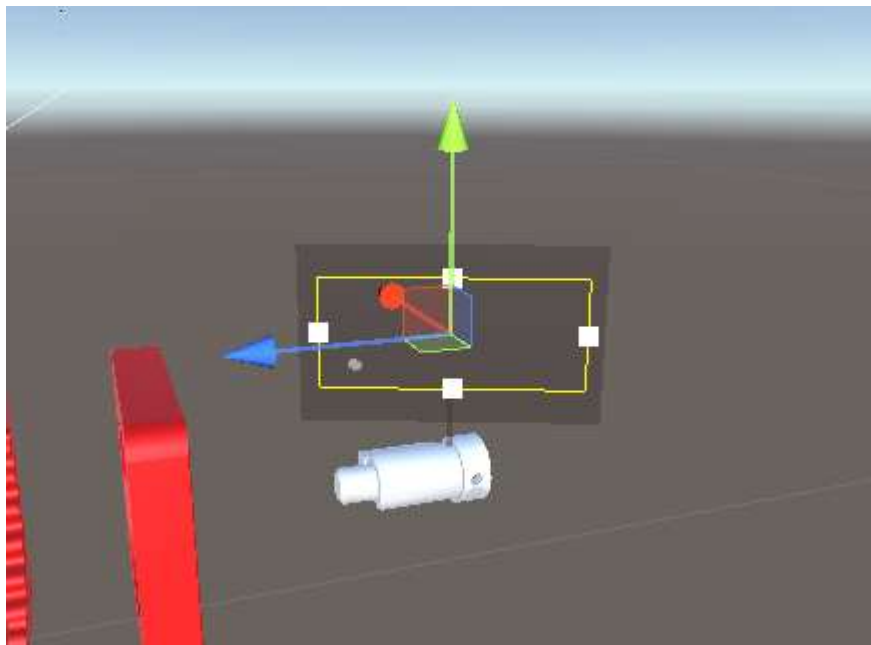


Рис.3.26 Масштабування полів відображення тексту

У тексті анотації вказується необхідна інформація для складального процесу: квалітет точності поверхонь, граничні відхилення, посадки, допуски на відхилення форми (рис.3.27).

На цьому перший етап створення анотації, а саме створення 3D об'єктів для відображення було завершено. Для реалізації функціоналу, який дозволить анотації завжди бути повернутою до камери, було створено скрипт під назвою «Face». В скрипті використано об'єкт «Vector 3».

В «Unity» вектори поділяються на основі розмірів «Vector2» для 2D виміру та «Vector3» для 3D виміру. «Vector3» містить інформацію про величину та напрямок для всіх трьох вимірів. У розробці застосунків «Vector 3» здебільшого використовується для визначення положення об'єкта та відстані між об'єктами. Для користування функціоналом під час роботи в середовищі розробки була прописана команда: [ExecuteInEditMode]

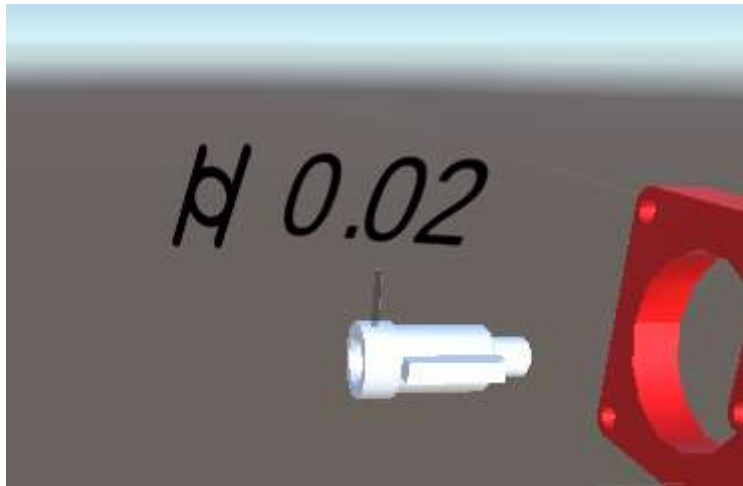


Рис.3.27 Анотована інформація перехідного валу

Для взаємодії з камерою створена змінна «cam» та для відображення напряму, в яку сторону має бути повернутий блок анотацій, створено змінну «targetAngle» класу «Vector3», для уникнення помилок при компіляції вектору надано нульові значення. У функції «Start()» змінній «cam» надано посилання на головну камеру у застосунку, тобто камеру користувача.

Функція Start(). Скрипт Face [Додаток Д], скрипт, що обертає 3D об'єкти анотацій до користувача.

```
public class Face : MonoBehaviour
{
    Transform cam;
    Vector3 targetAngle = Vector3.zero;
    void Start()
    {
        cam = Camera.main.transform;
```

```
}
```

У функції «Update()» у першому рядку функція «LookAt()» обертає об'єкт, над яким скрипт виконує дію, до об'єкту, який вказаний дужках, тобто камеру користувача. На наступному кроці кути, на які повернувся об'єкт зберігаються у раніше створену змінну «targetAngle» класу «Vector3» і надалі для забезпечення обертання тільки по осі «Y» значення кутів обертання по осям «X» та «Z» прирівнюються до нуля.

Функція Update(). Скрипт Face [Додаток Д], скрипт, що обертає 3D об'єкти анотацій до користувача.

```
void Update()
{
    transform.LookAt(cam);
    targetAngle = transform.localEulerAngles;
    targetAngle.x = 0;
    targetAngle.z = 0;
    transform.localEulerAngles = targetAngle;
}
```

Написаний скрипт зв'язано з об'єктом, який має обертатись до камери, а саме об'єкт «SectionInfo», що є батьківським класом для усього блоку анотацій (рис.3.28).

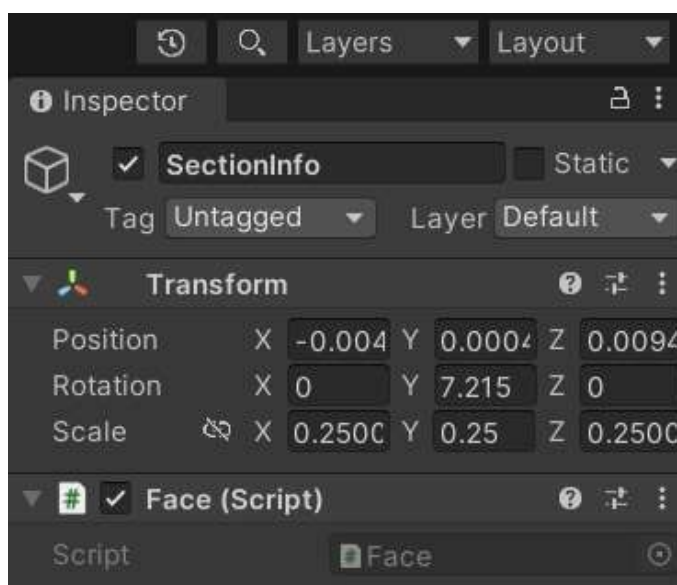


Рис.3.28 Підв'язаний скрипт до об'єкту SectionInfo

Третім етапом реалізації функціоналу анотацій є відображення інформації тільки за певної умови. Для цього запропоновано розробити систему, в якій анотація відображається за умови наведення центру камери на об'єкт. Створено два скрипти «InfoBehavior» та «Gaze». В скрипті «InfoBehavior» реалізовано функціонал відображення анотації завдяки зміні масштабу 3D об'єктів. Створено об'єкт «SectionInfo», що буде масштабуватись відносно своєї точки 0.

Функція Update(). Скрипт InfoBehavior [Додаток Д], скрипт, що відображає анотацію

```
public class InfoBehavior : MonoBehaviour
```

```
{
```

```
    const float SPEED = 6f;
```

```
    [SerializeField]
```

```
    Transform SectionInfo;
```

```
    Vector3 desiredScale = Vector3.zero;
```

```
    // Update is called once per frame
```

```
    void Update()
```

```
    {
```

```
        SectionInfo.localScale = Vector3.Lerp(SectionInfo.localScale, desiredScale,  
Time.deltaTime * SPEED);
```

```
    }
```

В функції «Update()» об'єкти масштабуються з заданою швидкістю. У функціях «OpenInfo()» та «CloseInfo()» прописані команди для відображення та відповідно закриття блоку анотацій.

Функція OpenInfo() CloseInfo(). Скрипт InfoBehavior [Додаток Д], скрипт, що відображає анотацію

```
public void OpenInfo()
```

```
{
```

```
    desiredScale = Vector3.one;
```

```
}
```

```
public void CloseInfo()
```

```
{
```

```

        desiredScale = Vector3.zero;
    }
}

```

Скрипт «Gaze» створено для контролю відображення кожного з блоків анотації. На початку роботи програми, під час запуску, зібрано в один список усі об'єкти, що містять у собі раніше написаний скрипт «InfoBehavior». Для цього попередньо створюється змінна типу «List<InfoBehavior>» під назвою «infos».

Функція Start(). Скрипт Gaze [Додаток Д], скрипт, що перевіряє напрям погляду користувача

```

public class Gaze : MonoBehaviour {

```

```

    List<InfoBehavior> infos = new List<InfoBehavior>();

```

```

    void Start()
    {

```

```

        infos = FindObjectsOfType<InfoBehavior>().ToList();
    }

```

У функції «Update()», що виконується з кожним кадром застосунку спочатку перевіряється умова, чи створений «промінь» перетинає будь який з об'єктів, тобто чи знаходиться 3D модель складальної одиниці в центрі екрану. Для цього застосовано функцію Raycast. Вона створює «промінь» від початку, у даному випадку це центр екрану користувача, в перпендикулярному напрямку і взаємодіє з усіма «колайдерами» 3D об'єктів в додатку. За умови виконання попередньої умови, застосунку необхідно перевірити чи має 3D модель інформацію, яку необхідно відобразити. Для цього створено змінну «go» типу «GameObject», якій надано значення об'єкту, що в даному випадку знаходиться в центрі екрану. Функцією «CompareTag» виконано перевірку, чи має обраний об'єкт додаткову інформацію для відображення. Усі об'єкти, для яких створюється анотація необхідно позначити тегом «hasInfo» (рис.3.29). За наявності потрібного тегу викликається функція «OpenInfo»,

в неї передається компонент «<InfoBehavior>()» об'єкту, який обробляється на даний момент. У випадку, якщо теги відрізняються викликається функція «CloseAll()».

Функція Update(). Скрипт Gaze [Додаток Д], скрипт, що перевіряє напрям погляду користувача

`void Update()`

```
{  
    if (Physics.Raycast(transform.position, transform.forward, out RaycastHit hit)){  
        GameObject go = hit.collider.gameObject;  
        if (go.CompareTag("hasInfo"))  
        {  
            OpenInfo(go.GetComponent<InfoBehavior>());  
        }  
    else  
    {  
        CloseAll();  
    }  
}
```

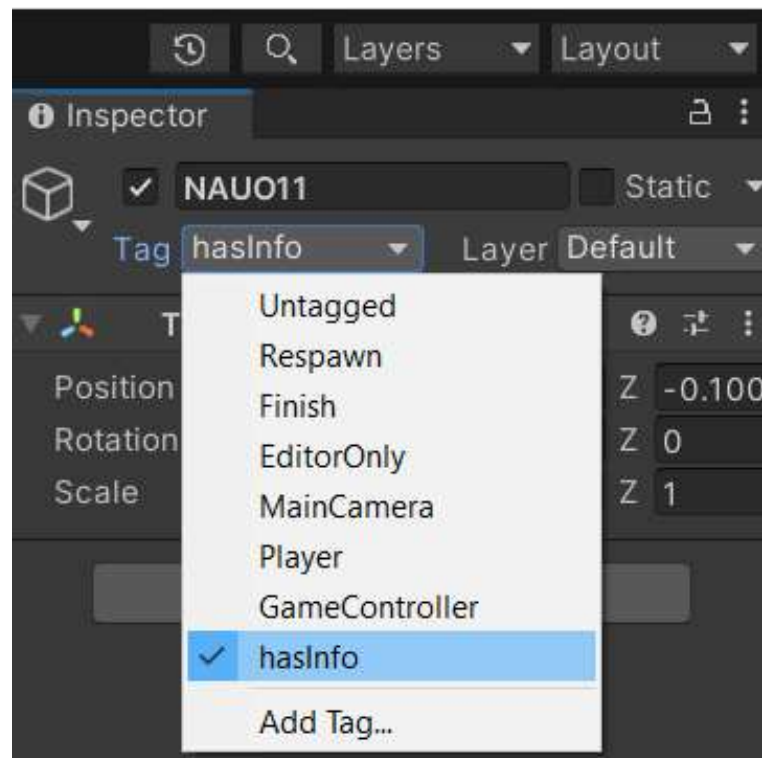


Рис.3.29 Маркування об'єкту з анотацією тегом «hasInfo»

Далі прописано раніше використані функції «OpenInfo» та «CloseAll». У функцію «OpenInfo» передаються значення об'єкту типу «InfoBehavior», які необхідно відобразити. У циклі програма проходить по усьому раніше створеному списку, таким чином перевіряється чи є об'єкт типу «InfoBehavior» в нащадках у 3D моделі, яка знаходиться по центру екрана. У випадку виконання умови викликається стандартна функція «OpenInfo()» класу «info». У всіх інших випадках, а саме коли розміщений у центрі об'єкт не містить об'єктів нащадків типу «InfoBehavior», викликається функція «CloseInfo()» класу «info».

Функція OpenInfo(). Скрипт Gaze [Додаток Д], скрипт, що перевіряє напрям погляду користувача

```
void OpenInfo(InfoBehavior desiredInfo)
{
    foreach(InfoBehavior info in infos)
    {
        if (info == desiredInfo)
        {
            info.OpenInfo();
        }
        else { info.CloseInfo();
        }
    }
}
```

У випадку виклику функції «CloseAll()», програма проходить по усьому раніше створеному списку і вимикає відображення анотацій функцією «CloseInfo()» класу info.

Функція CloseAll(). Скрипт Gaze [Додаток Д], скрипт, що перевіряє напрям погляду користувача

```
void CloseAll()
{
    foreach(InfoBehavior info in infos)
    {
```

```
info.CloseInfo();
}}}
```

Описаний скрипт додано до об'єкту «ARCamera», таким чином програмний код «знає», з зображенням якої камери має виконуватись його програма. Додатково промарковано у середовищі розробки, що раніше створена «ARCamera» має головний пріоритет у застосунку, тобто її зображення буде показуватись користувачеві під час роботи застосунку.

Останнім етапом реалізації функціоналу анотованих 3D моделей було створено віртуальні габарити об'єкту, його сітку. У випадку відсутності цих об'єктів, раніше написаний скрипт «Gaze», не зможе розпізнати, чи розміщений 3D об'єкт по центру зображення. Об'єкти, які створено для реалізації необхідного функціоналу: «Mesh», «Mesh Renderer», «MeshCollider» та «BoxCollider». Об'єкт «Mesh» містить у собі

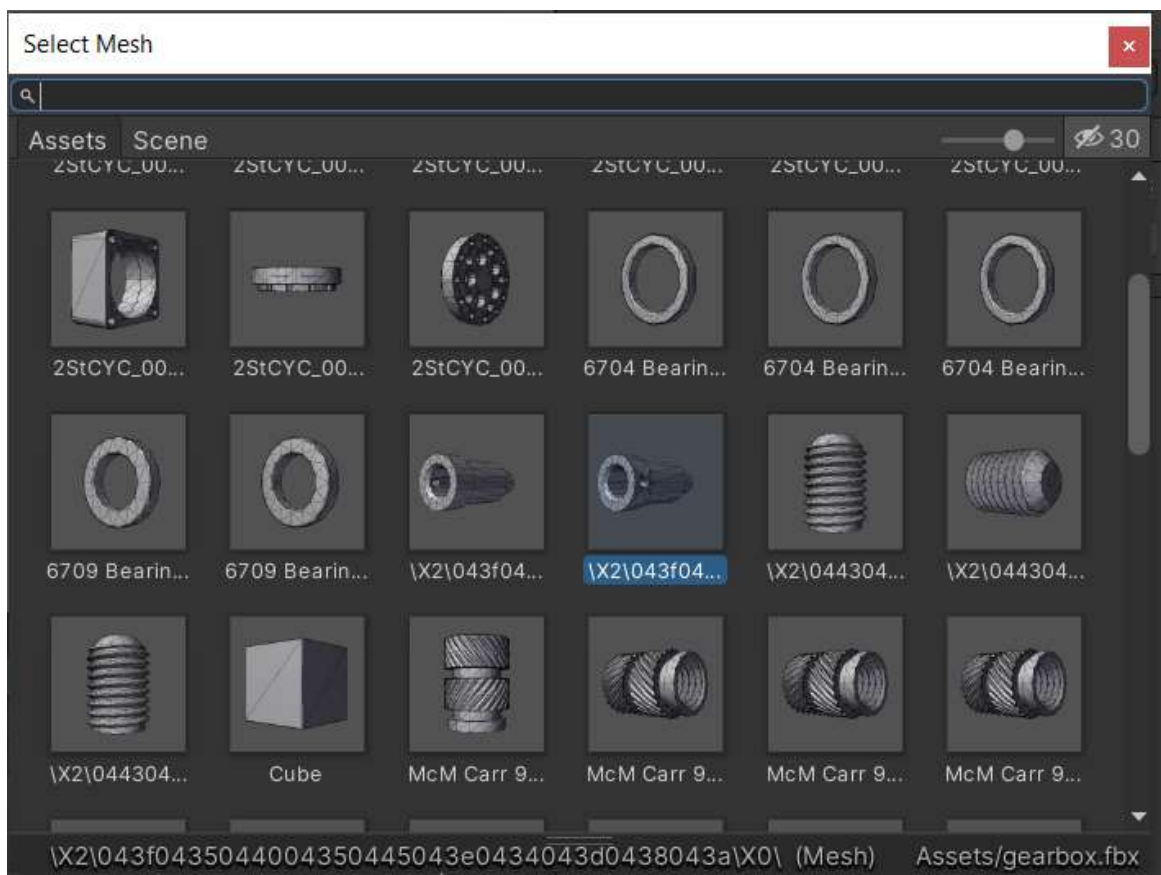


Рис.3.30 Вибір сітки 3D моделі

інформацію про форму 3D моделі, сітки кожної складальної одиниці отримано у пункті «3.1.1 Підготовка моделі складальної одиниці», для додавання сітки до об'єкту складальної одиниці необхідно вибрати її серед усіх доступних (рис.3.30).

Додатково для використання сітки під час роботи застосунку створено об'єкт «MeshRenderer». Задля взаємодії сітки з іншими об'єктами застосунку, а саме створеним в скрипті «промінем», що має перевіряти наявність об'єкту у центрі, створено об'єкт «MeshCollider». В тих випадках, коли об'єкт занадто малий, або має у своїй сітці порожнини, що може завадити коректній роботі застосунку, створювався елемент «BoxCollider», це дозволило збільшити зону, перетинання якої дозволить застосунку відобразити інформацію. Автоматично створений елемент «BoxCollider» має занадто великий об'єм, для коректного використання застосунку необхідно зменшити його границі (рис.3.31).

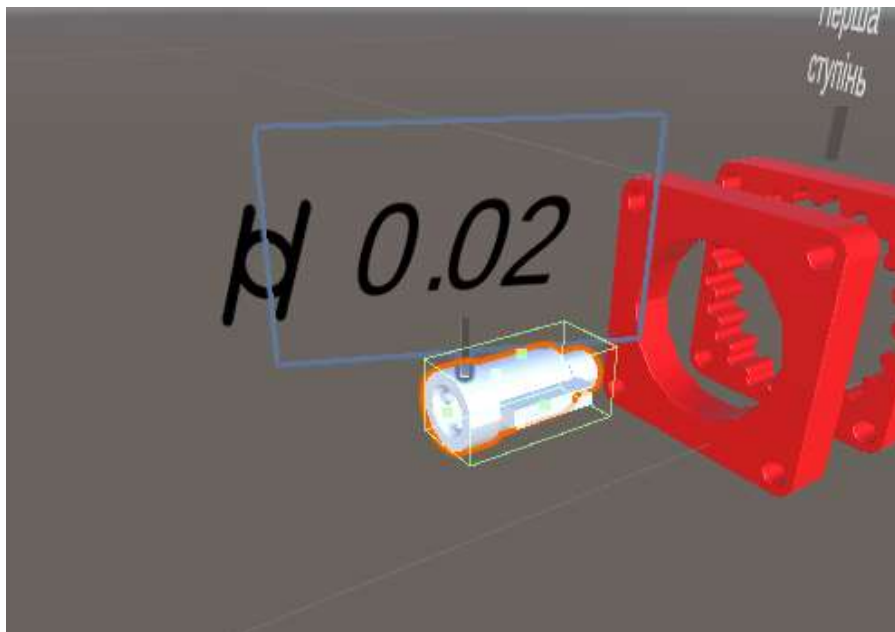


Рис.3.31 Редаговані границі елементу «BoxCollider»

Висновки третього розділу.

В процесі виконання третього розділу розроблено практичну програмну частину магістерської дисертації. Конвертовано 3D модель циклоїдального редуктора у полігональний формат моделі для подальшої взаємодії з нею поза САПР. Для застосунку створено зображення-маркер, на якому відображається 3D модель циклоїдального редуктора. У середовищі розробки «Unity» створено проект, налагоджено середовище розробки. Як базовий функціонал застосунку реалізовано анімацію складання і розборки 3D моделі, для цього застосовано плагін «Animator» та написано керуючі скрипти, що дають можливість користувачеві керувати

застосунком. Як розширений функціонал додано можливість перемикати анімації покроково. Для цього модифіковано попередньо написані скрипти та створено додаткові блоки в плагіні «Animator». Запропоновано реалізувати анотацію 3D моделей, задля отримання додаткової інформації під час складального процесу. Під час реалізації вище згаданого функціоналу додано сітку до 3D моделей деталей, які необхідно анотувати. Створено 3D об'єкти, що будуть відображатись у якості анотації. Прописані керуючі скрипти для головної камери проекту, створених об'єктів анотації та 3D моделей, що містять анотації.

4. Практична апробація

4.1 Рекомендації щодо впровадження розробки в виробництво.

Використання застосунку передбачається для відображення додаткової цифрової інформації під час складального процесу при дрібносерійному або одиничному виробництві. Розробка дозволяє зменшити час виконання підготовчо-заклучних робіт, цифрові підказки допомагають працівникові швидше зорієнтуватись у нестандартному завданні, яке йому необхідно виконати. У випадку крупносерійного виробництва із малою кількістю номенклатури також можна застосувати аналогічну розробку, проте тільки під час впровадження нової продукції, технологія складання до якої ще недостатньо відпрацьована.

4.2 Методика використання розробки.

Розроблений застосунок працівник має використовувати у робочий час, під час виконання робочих обов'язків. Розпочинаючи свою зміну, працівник вмикає застосунок і перевіряє, чи створено для нього змінне завдання. Тобто чи розміщена інформація, на якій ділянці і що саме він має складати, яка кількість виробів, скільки часу для цього виділено і які інструменти для цього він має використати.

Отримавши інформацію, щодо завдання, роздрукувавши креслення, робітник направляється на своє робоче місце. Застосунок створено для зменшення часу на підготовчо-заклучні процеси, тому ознайомившись з кресленням, яке застосунок буде використовувати у якості маркеру, і розташувавши його у безпечному місці, працівник може його відкрити і почати ознайомлюватись з 3D моделлю об'єкту, який необхідно скласти.

Перед початком складальних робіт, робітник може використати інструменти анімації в застосунку. Для загального, швидкого ознайомлення працівника з процесом зборки, було реалізовано процес анімації повної розборки циклоїдального редуктора та його складання. Для більш детального ознайомлення та опрацювання кожного кроку, реалізовано покрокову анімацію складання і розборки 3D моделі. Використовуючи кнопки «Back» та «Forward» (рис.3.32) користувач може переміщуватись по анімації від розібраного циклоїдального редуктора до зібраного.

Під час виконання складальних робіт, у випадку коли, робітникам необхідна



Рис.3.32 Кнопки керування покроковою анімацією; а-попередня анімація,
б-наступна анімація

певна інформація щодо конкретних складальних одиниць, йому потрібно розмістити деталь, що його цікавить, по центру екрану. Якщо застосунок має інформацію для відображення, анотація з'явиться над 3D моделлю деталі, як це показано на рис.3.33.

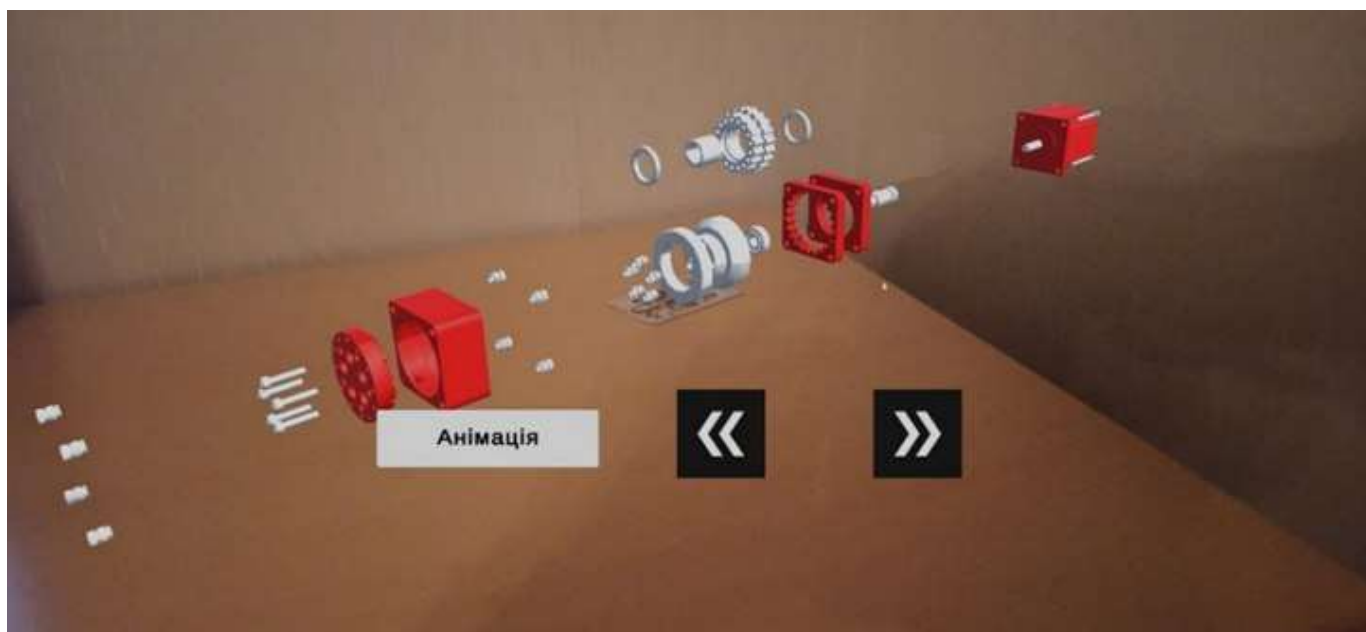


Рис.3.33 Анотована 3D модель у робочому застосунку

4.3 Результати тестувань застосунку в реальних умовах

Експериментальне використання застосунку відбувалося на базі стартап проекту LineBar при складанні циклоїдального редуктора. З використанням застосунку робітниками, що ніколи не виконували складальні операції циклоїдального редуктора, час на складання в середньому скоротився в середньому на 30% (табл.4.1) у порівнянні з використанням тільки складального креслення.

Табл.4.1 Порівняння часу складання циклоїдального редуктора з використання розробленого застосунку та без

	Робітник 1	Робітник 2	Робітник 3	Робітник 4
Складання з застосунком, хв	5,4	5,8	5,1	4,8
Складання без застосунку, хв	6,9	7,0	6,7	6,8

Під час складання робітниками (табл.1) було виявлено деякі недоліки і надано пропозиції щодо покращення застосунку. А саме запропоновано в анотаціях до 3D моделей прописувати номер позиції деталі та її назву. Не зважаючи на складність конструкції, кожен робітник зібрав циклоїдальний редуктор без помилок.

Про актуальність розробки і її відповідність потребам реального виробництва свідчить акт впровадження на виробництво «ТОВ ЗАВОД ЄВРОФОРМАТ», застосунок буде використаний на дільниці складання ліфтового обладнання. [Додаток Е].

Можливі перспективи розробки.

Перспективами розвитку розробленої системи є автоматизація процесу підготовки моделей складальних одиниць, процесу їх анотування та анімації. Зменшивши час на підготовку цифрових інструкцій збільшується кількість випадків, в яких доречно буде використовувати ДР на складальних ділянках виробництва.

Перспективами впровадження доданої реальності у вигляді цифрових інструкцій є ряд частин виробничого циклу, які можна пришвидшити або полегшити

певний процес. Впровадження у масове виробництво з дуже малою номенклатурою. Зменшення собівартості деталі за рахунок зменшення часу на її виробництво збільшить прибуток виробництва. Необхідність у постійних настановах працівників на конвеєрній лінії, або на будь якій дільниці зменшить відсоток помилок, причиною яких є людський фактор, що призведе до зменшення кількості заводського браку.

Масштабування застосунку до мережі, де працівники постійно можуть переглядати необхідну для них інформацію та залишати звіти щодо виконаних робіт або перешкоди, що заважають їм виконувати їх роботу. Сьогодні на великих виробництвах працівники взаємодіють з системою керування підприємством через комп'ютер або у паперовому вигляді. У разі виявлення браку, або недостачі деталей працівникові необхідно подати звіт з фото-підтвердженням, такий протокол зупиняє виробництво проблемної серії не менш ніж на добу. Пропонується в подальших розробках створити дисплей у вигляді окуляр, що мають можливість відображати цифрову інформацію та знімати фото та відеоматеріали. Використовуючи подібний девайс на рівні усього виробництва, що надає змогу працівникам мати прямий контакт з мережею, час оформлення звіту про проблему зменшується до хвилини, а вирішення питання починається у той самий день.

Висновки до четвертого розділу.

У розділі практичної апробації описано процес інтеграції застосунку у реальне виробництво. Розроблений застосунок дозволив зменшити час складання циклоїдального редуктора на 30%. Розроблено методику використання застосунку на виробництві, у методиці описані дії працівника від отримання змінного завдання до зібраного циклоїдального редуктора. Описано перспективи розробленого додатку, запропоновано автоматизувати підготовку усіх необхідних файлів для імпортування 3D моделей в середовище «Unity». Другим аспектом автоматизації створення додатку є пришвидшення процесу створення анімацій і додання анотацій.

5. Розробка стартап-проекту

5.1 Опис ідеї проекту

У попередніх розділах було описано розроблений застосунок і користь його інтеграції у робочий процес. У даному розділі було виконано аналіз стартап проекту на базі розробленого застосунку. Проведений огляд необхідний для розуміння сильних та слабких сторін майбутнього продукту, що дає більшої ясності при формуванні стратегії виходу на ринок.

Запропонована ідея стартап проекту це використання доданої реальності у вигляді цифрових інструкцій під час виконання складальних робіт. Суть полягає у використанні додатку на смартфон робітника, скануючи складальне креслення робітник отримує можливість працювати з 3D моделями у доданій реальності. Для подання цілісного уявлення про зміст ідеї, було розроблено таблицю (табл. 5.1) в ній було розглянуто можливі ринки у полі яких необхідно шукати групи потенційних клієнтів.

Таблиця 5.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Відображення цифрових інструкцій складання за допомогою доданої реальності	Скорочення підготовчо-заклучного часу складальної операції	Зменшення собівартості виробу
	Проведення навчання робітників	Зменшення кількості браку продукції

Пропонується розробити застосунок з можливістю відображення цифрової інформації за допомогою екранів телефонів та планшетів, з метою зменшення часу на виконання складальних операцій і зменшення браку продукції.

Наступним кроком було проведено аналіз потенційних техніко-економічних переваг ідеї порівняно із, доступними на сьогоднішній день, аналогами:

- визначено перелік техніко-економічних властивостей та характеристик ідеї;
- визначено попереднє коло конкурентів або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проведено збір інформації щодо значень

техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

– проведено порівняльний аналіз показників (табл. 5.2): для власної ідеї визначено показники, що мають : гірші значення (W, слабкі); аналогічні (N, нейтральні) значення; кращі значення (S, сильні)

Таблиця 5.2. Позначення нейтральних, слабких та сильних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проєкт	Конкурент1 MA ² RA	Конкурент2 Radkowski work[16]	Конкурент3 ARAT[18]			
1.	Фінанси	Власні	Університетське фінансування	Університетське фінансування	Власні	+		
2.	Швидкість інтеграції	Швидка	Середня	Повільна	Середня			+
3.	Складність апробації	Легко	Середнє	Середнє	Важко			+
4.	Технічний потенціал	Середній	Високий	Високий	Середній		+	
5.	Перспективи удосконалення	Наявні	Наявні	Відсутні	Відсутні			+
6.	Вартість інтегрування	Дешево	Середнє	Коштовно	Середнє			+

За результатом формування таблиці було зроблено висновок, що можливість технологічної реалізації проекту більш ніж можлива і доцільна, беручи до уваги наявні переваги поміж конкурентами у таких техніко-економічних характеристиках як швидкість інтеграції, складність апробації, перспективи удосконалення та вартості інтегрування.

Наступним кроком було виконано огляд існуючих технологій, якими доцільно було б використати для технічної реалізації створеного проекту. Визначення здійсненності технологічної ідеї проекту передбачає аналіз складових в таблиці 5.3.

Таблиця 5.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технології
1.	Програмне забезпечення за допомогою якого можливо створити описаний застосунок	Мова програмування C#	Технологія наявна	Доступна
		Середовище розробки Unity	Технологія наявна	Доступна
		SDK Vuforia Engine	Технологія наявна	Доступна
		Програма для конвертації файлів Analysis Situs	Технологія наявна	Доступна
2.	Програмне забезпечення оперативною системою	Windows	Технологія наявна	Доступна
		Android	Технологія наявна	Доступна
Обрана технологія реалізації ідеї проекту: Розробка мобільного додатку з програмним забезпеченням Android в середовищі розробки Unity за допомогою SDK Vuforia Engine та мови програмування C#, попередньо підготувавши 3D моделі в системі Analysis Situs				

Результатом аналізу таблиці було зроблено висновок, що для реалізації проекту мобільного додатку для відображення цифрових інструкцій з метою зменшити час на виконання складальних операцій на сьогоднішній момент усі технології наявні і доступні. Обрано середовище розробки Unity, на базі мови програмування C# та операційної системи Windows. Застосунок має функціонувати на телефоні з операційною системою Android.

5.2 Аналіз ринкових можливостей запуску стартап-проекту

Було визначено ринкові можливості, які можна використати під час ринкового впровадження проекту, та ринкові загрози, які можуть перешкодити його реалізації. Аналіз ринкових можливостей сьогодні дозволить спланувати напрямок розвитку

проекту з врахуванням попиту, характеристики потенційних клієнтів, факторів загроз і можливостей та аналіз конкуренції на ринку.

Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 5.4).

Таблиця 5.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	13 білльйонів доларів[8]
3	Динаміка ринку (якісна оцінка)	зростає
4	Наявність обмежень для входу (вказати характер обмежень)	немає
5	Специфічні вимоги до стандартизації та сертифікації	Обов'язкова сертифікація безпеки використання застосунку на виробництві
6	Середня норма рентабельності в галузі (або по ринку), %	60%

Провівши аналіз було виявлено що динаміка ринку є зростаючою. Вимогами є обов'язкова сертифікація безпеки використання на виробництві. Середня норма рентабельності в галузі (або по ринку) має значення вище середнього, отже можна зробити висновок за результатами аналізу, що ринок є привабливим для входження.

Надалі було визначено потенційні групи клієнтів, їх характеристики, та сформовано орієнтовний перелік вимог до товару для кожної групи (табл. 5.5).

Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Необхідність зменшити час на виконання складних складальних операцій	Виробництва з комплексною продукцією – експериментальні виробництва	Різні виробництва використовують різні CAD системи	Зручність у використанні, ефективність
2	Зменшити кількість браку виробничої продукції	Масові виробництва з малою кількістю номенклатури	Різні системи керування підприємством	Швидкість і простота інтеграції на виробництва
3	Пришвидшити процес навчання нових працівників	Крупносерійні виробництва	Фінансові можливості виробництва	Швидкість створення файлів анімації для нової продукції

Було розглянуто потенційних клієнтів стартап-проекту, їх потреби, які формують ринок, цільові сегменти ринку. Було освітлені відмінності різних потенційних цільових груп клієнтів стартап проекту. Стартап проект буде створено беручи до уваги вимоги споживачів до товару, а саме зручність у використанні, швидкість і простоту інтеграції на виробництва, увагу буде приділено ефективності використання створеного застосунку.

Необхідно взяти до уваги наявні загрози, що можуть стати на заваді під час реалізації стартап проекту. Було розроблено таблицю з факторами та змістом загрози і опрацьована можлива реакція на описаний фактор, що загрожує компанії.

Таблиця 5.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
-------	--------	---------------	--------------------------

1.	Конкуренти з подібним сервісом	Розвиток компаній, що розробляють схожий продукт	Конкурентно-спроможний продукт у поєднанні із якісним сервісом гарантує збереження прибутку від клієнтів
2.	Компанії, що самостійно розробляють цифрові інструкції	Компанії самостійними потужностями будуть розробляти цифрові інструкції для робітників	Надання сервісних послуг компаніям, підтримка під час інтеграції у виробничих процес зробіть пропозицію привабливішою
3.	Зменшення актуальності методу	Швидкість розвитку технологій лишить позаду розроблену методику використання доповненої реальності	Ефективна робота відділу R&D забезпечує орієнтацію у нових технологіях, що зробіть компанію лідером ринку
4.	Глобалізація рішення	Популяризація розробки і доступність технологій дають передумови збільшенню кількості проектів із повторенням розробленого функціоналу	Захисту патентованих технологій юридичним відділом

Було проаналізовано та опрацьовано кожну з передбачених загроз, що можуть стати перешкодою для стартап проекту під час виходу і існуванні на ринку. Усвідомлюючи ризики, було запропоновано рішення індивідуально до кожної з проблем, а саме постійний моніторинг ринку інноваційних технологій, захист технології патентуванням у правовому полі, сервісна підтримка клієнтів.

Було запропоновано розглянути фактори можливостей(таблиця 5.7) усвідомлюючи переваги стартап проекту буде правильним сконцентруватись на сильних сторонах під час запуску проекту.

Таблиця 5.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Зростання ринку доповненої реальності	Витрати виробничих компаній на технології доданої реальності збільшуються	Необхідно пришвидшити темпи розвитку проекту, залучувати інвестиції, відвідувати міжнародні виставки
2.	Простота інтеграції	Оскільки застосунок реалізовано на телефоні, інтеграція у виробництво не забере багато часу	Швидкість і простота інтеграції надає можливість швидкого масштабування проекту на велику кількість заводів

3.	Потенціал технологій доповненої реальності	Розвиток технологій доданої реальності робить їх легшими і кориснішими у застосуванні	Розвиток технологій дозволить покращити якість продукту, збільшити спектр послуг, що надає компанія
4.	Ліміт можливостей автоматизації процесів	Не усі техн. процеси можна автоматизувати, тому завжди будуть складні процеси для виконання людиною, яка може використовувати технологію стартап проекту	Необхідно зосередити увагу на пошуку галузей, де тільки людина може виконувати технічне завдання

Було виокремлено декілька факторів, що можуть сприяти розвитку проекту, маючи на увазі усі з перерахованих пунктів, проект гарантовано почне свій шлях у правильному напрямку, а саме: участь в міжнародних конференціях і виставках, залучення інвестицій, зосередження уваги на масштабуванні проекту на виробництвах, постійний аналіз сучасних технологій і аналіз можливостей застосування у ширший спектр виробництв.

Було проведено аналіз конкуренції на ринку, це дало змогу визначити загальні риси конкуренції на ринку (табл.5.8).

Таблиця 5.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1.Тип конкуренції - монополістична	Монополістичну конкуренцію характеризують тим, що кожна фірма в умовах диференціації продукції володіє деякою монополією над своїм товаром: вона може підвищувати або ж знижувати ціну на нього незалежно від дій конкурентів.	Поступовий набір бази клієнтів, розширення впливу на ринок завдяки коопераціям з вже відомими брендами
2. За рівнем конкурентної боротьби - національна	Національний рівень конкурентної боротьби характеризується конкуренцією в межах однієї країни	Вихід на міжнародні ринки виробництв буде правильним рішенням в умовах стагнації виробництва в Україні
3. За галузевою ознакою – внутрішньогалузева	Внутрішньогалузева ознака визначається впровадженням нової техніки, нових	Використання нових технологій, пошук

	технологій з метою зниження витрат	перспективних розробок для впровадження
4. Конкуренція за видами товарів – товарно-родова	Товарно-родова конкуренція на рівні технології задоволення потреб.	Зменшення часу на виконання технологічної операції складання у порівнянні із звичайними методами
5. За характером конкурентних переваг - нецінова	Нецінова характеризується вдосконаленню якості продукції, технологій виробництва і умов її продажу	Полегшення процесу імплементації у виробництво, зменшення вартості інтеграції
6. За інтенсивністю – не марочна	Роль торгової марки незначна	Популяризація в медіа, колаборації з відомими виробниками

Було виконано ступеневий аналіз конкуренції на ринку, було визначено усі особливості конкурентного середовища, висновком аналізу є необхідність популяризації проекту у медіа і колаборація з вже відомими брендами виробників, поступовий набір бази клієнтів на вітчизняному ринку, а потім вихід на міжнародний ринок з перевіреним і якісним продуктом. Після аналізу конкуренції було проведено більш детальний аналіз умов конкуренції в галузі за моделлю 5 сил М. Портера.

Таблиця 5.9. Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналіз у	Ma ² RA Radkowski work ARAT	Потенційним і конкурентами є розробники застосунків з відображенням доданої реальності для складальних робіт	Постачальники відсутні	Клієнтами є виробництва, які потребують зменшення часу на виконання складальних операцій	2D креслення, відео інструкції
Висновки:	Конкурентна боротьба є середньою	Можливості для входу на ринок наявні Потенційні конкуренти є Строки виходу на ринок 6 місяців	Постачальники не диктують умови роботи на ринку	Клієнти диктують умови вартості розробки і платформи на якій реалізуються проект	Товари замінники спричиняють зменшення притоку клієнтів

Було виконано аналіз таблиці конкуренції в галузі за М. Портером. Як висновок було виявлено прямих і потенційних конкурентів. Було констатовано факт, що постачальники не диктують умови на ринку у той час як через вартість розробки платформи, клієнти можуть диктувати умови. Було виявлено, що товари замітники спричиняють зменшення притоку клієнтів.

На основі аналізу конкуренції, проведеного в таблиці 5.9, а також із урахуванням характеристик ідеї проекту в таблиці 5.2, вимог споживачів до товару таблиця 4.5 та факторів маркетингового середовища таблиця 5.6 та 5.7 визначимо та обґрунтуємо перелік факторів конкурентоспроможності. Аналіз оформлюється за табл. 5.10

Таблиця 5.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Ефективність	Зменшення часу на виконання операції є ключовим фактором, що є найвагомішим серед усіх
2	Швидкість інтеграції	Завдяки реалізації технології на телефонах, її впровадження не вимагає багато часу
3	Ціна	Легке досягнення рентабельності робить ціну нижче
4	Гнучкість	Використовувати розроблений застосунок можливо для багатьох цілей у декількох напрямках
5.	Підтримка і розвиток	Систему легко модифікувати в залежності від вимог виробництва

За визначеними факторами конкурентоспроможності таблиця 5.10, було проведено аналіз сильних та слабких сторін стартап-проекту таблиця 5.11.

Таблиця 5.11. Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з розробленим додатком						
			-3	-2	-1	0	+1	+2	+3
1	Ефективність	19				+			
2	Швидкість інтеграції	16		+					
3	Ціна	12		+					
4	Гнучкість	15			+				
5.	Підтримка і розвиток	18						+	

З таблиць 5.10 та 5.11 видно, що загальними переважаючими факторами є швидкість інтеграції та ціна у порівнянні з конкурентами, проте ефективність є на рівні з аналогами.

Таблиця 5.12. SWOT- аналіз стартап-проєкту

Сильні сторони: 1. Швидкість інтеграції 2. Ціна 3. Ефективність	Слабкі сторони: 1. Складність розробки 2. Відсутність медійності проєкту
Можливості: 1. Зростання ринку доповненої реальності 2. Проста інтеграція розробки 3. Потенціал технологій доповненої реальності 4. Ліміт можливостей автоматизації процесів	Загрози: 1. Конкуренти з подібним сервісом 2. Компанії, що самостійно розробляють цифрові інструкції 3. Глобалізація рішення 4. Зменшення актуальності методу

В таблиці 5.12 було складено SWOT- аналіз, було підсумовані сильні і слабкі сторони проєкту, проаналізовано можливості стартапу які необхідно використати і загрози, які необхідно врахувати.

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки для того, щоб вивести стартап-проєкт на ринок та визначення орієнтовного оптимального часу їх ринкової реалізації з огляду на потенційні проєкти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи було проаналізовано з точки зору строків та ймовірності отримання ресурсів.

Таблиця 5.13. Альтернативи ринкового впровадження стартап-проєкту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Вирішення проблеми відсутності медійності проєкту шляхом швидкого розповсюдження проєкту завдяки швидкій інтеграції на виробництва	Висока	1-5 місяців
2	Завдяки ефективності розробки і невеликій ціні впровадження,	Висока	5-8 місяців

	рентабельність проекту вирішує проблему складності розробки		
3	Стратегія виходу з ринку	Низька	

З описаних альтернатив було обрано ринкову поведінку №1, а саме вирішення проблеми відсутності медійності проекту шляхом швидкого розповсюдження проекту завдяки швидкій інтеграції на виробництво, оскільки ймовірність отримання ресурсів є високою з відносно невеликими термінами.

5.3 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 5.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Виробництва з комплексною продукцією – експериментальні виробництва	Готові	Високий	Середня	Проста
2	Масові виробництва з малою кількістю номенклатури			Низька	Проста
3	Крупносерійні виробництва			Середня	Середня
Які цільові групи обрано: Під час аналізу потенційних груп споживачів було прийнято рішення що компанія буде працювати із виробництвами з комплексною продукцією, експериментальними виробництвами					

За результатами аналізу потенційних груп споживачів було обрано цільові групи виробництвами з комплексною продукцією, експериментальними виробництвами. Було обрано стратегію диференційованого маркетингу

Для роботи в обраному сегменті ринку необхідно сформулювати базову стратегію розвитку.

Таблиця 5.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
	Вирішення проблеми відсутності медійності проекту шляхом швидкого розповсюдження проекту завдяки швидкій інтеграції на виробництва	Диференційований маркетинг	Швидко інтегрований, ефективний застосунок, що має низьку ціну	Стратегія диференціації

Було обрано, в результаті аналізу визначення базової стратегії розвитку, стратегію диференціації, вона надає товару дуже важливого для користувача позитивних відмінностей. Перевагою стратегії диференціації полягає у тому, що знижується ступінь можливості замінити товар, підвищується поріг входу на ринок, підвищення рентабельності.

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 5.16).

Таблиця 5.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні	Так	Базовий функціонал застосунку суміжний з товарами конкурентів. Розширений функціонал відрізняється	Стратегія зайняття конкурентної ніші

Було проаналізовано базову стратегію конкурентної поведінки проекту в таблиці 5.16, що визначає активність та напрям дій по відношенню до лідера відповідного ринку. Обрана стратегія – зайняття конкурентної ніші.

На основі вимог споживачів з обраного сегменту до постачальника і продукту, а також в залежності від стратегії розвитку та стратегії конкурентної поведінки було розроблено стратегію позиціонування, яка визначається у формування ринкової позиції, за яким споживачі мають ідентифікувати проект.

Таблиця 5.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Зручність і простота у використанні	Стратегія диференціації	Швидкість інтеграції у виробничий процес, ефективність використання застосунку.	Ефективність Швидкість Зручність

У таблиці 5.17 визначення стратегії позиціонування було сформовано асоціацій, що мають сформувати комплексну позицію описаного проекту, а саме Ефективність, швидкість, зручність.

5.4 Розробка маркетингової стратегії стартап проекту

Під час розроблення маркетингової програми першим кроком є розробка маркетингової концепції товару, який отримає споживач. У таблиці 5.18 було підсумовано результати аналізу конкурентоспроможності товару.

Таблиця 5.18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Ефективність	Скорочення часу на виконання операції	Зменшення часу на виконання операції веде до зменшення собівартості, що в кінці веде до збільшення прибутку
2	Швидкість	Швидкість інтеграції у виробничий процес	Відсутність глобальних змін необхідних до впровадження перед інтеграцією в процес зменшує час на інтеграцію, отже ефект буде відчутній одразу.
3	Зручність	Зручність у користуванні застосунком	Зручність у користуванні не викличе відторгнення нової технології

Визначивши ключові переваги перед конкурентами, було отримано висновок, що вагомими перевагами проекту є швидка інтеграція ефективного застосунку на виробництво, що миттєво буде прийнятий робітниками і ефект зменшення часу на

виконання складальних операцій буде проглядатись вже на перших тижнях інтегрування.

Таблиця 5.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Застосунок для зменшення часу на виконання складних та комплексних складальних операцій		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1.Довговічність (немає строку давності)	Нм	Тх
	2. Ефективність	Нм	Тх
	3. Невелика ціна застосунку	М	Е
	4. Зручність користування	Нм	Вр
	5. Рентабельність	М	Е
	6.Швидкість інтеграції	Нм	Ор
	Якість: відповідає нормам ISO 12207:2016		
	Пакування: програмне забезпечення встановлюється на операційну систему Android		
	Марка: AR Assembler		
III. Товар із підкріпленням	До продажу: Використання 2D креслень для складання		
	Після продажу: Використання цифрових інструкцій, 3D моделей для складання		

У таблиці 5.19 було виконано опис трьох рівнів моделі товару-розробки, а саме товар за задумом, товар у реальному виконання, товар із підкріпленням,

Наступним кроком було визначено цінові межі, якими необхідно керуватися при встановленні ціни на потенційний товар, це передбачає аналіз цін товарів конкурентів, та доходів споживачів продукту (табл. 5.20).

Таблиця 5.20. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
	4000-6000 грн	7000-8000 грн	500000 грн	4000-8000 грн

У таблиці було визначено межі встановлення ціни, для цього було використано рівні цін на товари заміники, аналоги, було звернуто увагу на рівень доходів цільової

групи споживачів. Як результат було запропоновано верхню та нижню межі ціни на послугу.

Таблиця 5.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Необхідно обрати ефективний продукт, який легко інтегрувати	Постійна підтримка клієнтів у «Secondary development» продукції	Пряма	Напряму до клієнта
2	Економія виділених коштів	Зменшувати порог технологічного розвитку виробництва для можливості використання технології	Пряма	Напряму до клієнта

У таблиці 5.21 було сформовано систему збуту, обрано формат збуту напряму до клієнта, оскільки у посередництві немає сенсу на даному етапі.

Таблиця 5.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Бажання одним рішенням вирішити декілька проблем	Міжнародні конференції, експозиції, промислові форуми	Багатогранність, ефективність	Охопити у повідомленні увесь спектр вирішених проблем за допомогою застосунку	«Одним застосунком на усі стільці»
2	Складне рішення не зацікавить клієнта на відміну від зрозумілого і простого	Міжнародні конференції, експозиції, промислові форуми	Простота у використанні	Надання прикладу простого використання розробки	«Якщо щось виглядає легко, хтось дуже постарався, щоб це не виглядало складно»

У таблиці 5.22 було опрацьовано концепцію маркетингових комунікацій. Результатом отримано маркетингову програму стартап проекту, популяризацію та визначено межі ціни, сформовано системи збуту.

Висновки до п'ятого розділу

На виробництвах сьогодення актуально автоматизувати усі процеси, проте не кожен процес можна та треба автоматизувати, проте є інші шляхи скорочення часу та підвищення якості, в цих причин було запропоновано реалізувати застосунок, що зменшував би час ручної праці і підвищував би її якість шляхом відображення цифрових інструкцій для користувача.

Під час розробки стартап проекту були визначенні сильні та слабкі сторони проекту, було визначено цільове коло клієнтів, під потреби яких підходить реалізований функціонал додатку. Базовою стратегією було обрано стратегію диференціації, конкурентна поведінка буде реалізована на базі зайняття конкурентної ніші. Припускається, що зацікавленим типом груп будуть клієнти - виробництва з комплексною продукцією, експериментальними виробництвами. До асоціацій, що мають характеризувати проект було віднесено ефективність, швидкість, зручність.

Створений SWOT-аналіз демонструє слабкості стартап проекту, проте і показує його переваги. Було відображено існуючі загрози стартап проекту, на противагу ним перераховано можливості, які команді необхідно використати. Ступеневий аналіз конкуренції проекту надає розуміння конкурентоспроможності проекту на галузевому ринку.

Потреба сьогоденних виробництв у ефективному і швидко-інтегрованому рішенні дає гарний поштовх до виходу на ринок, проте через вартість розробки вони можуть диктувати в ціновому діапазоні.

Ідея стартап проекту вважається прибутковою і проект має бути успішним, оскільки потреба виробництв у пришвидшенні виробничого процесу була є і буде, швидкість інтеграції і функціонал робить проект лише привабливішим.

ЗАГАЛЬНІ ВИСНОВКИ

1. У ході виконання магістерської дисертації виконано аналіз існуючих рішень використання доданої реальності в складальних процесах, що дало розуміння концепту майбутнього застосунку.
2. Розроблено алгоритм та архітектуру застосунку, що дозволило упорядкувати роботу функціоналу застосунку з використання доданої реальності.
3. Розроблено мобільний застосунок на операційній системі Android, що дозволяє виконувати анімацію зборки та розборки циклоїдального редуктора та відображати анотації його деталей.
4. Створено методику для використання застосунку на виробництві, застосування якої дозволило зменшити час складання циклоїдального редуктора на 30 %.
5. Виконано апробацію результатів дисертаційних досліджень на базі «ТОВ ЗАВОД ЄВРОФОРМАТ», що підтвердило дієздатність розробленого застосунку та методики його використання, а також дозволило окреслити перспективи майбутнього розвитку даної розробки.

Список використаних джерел

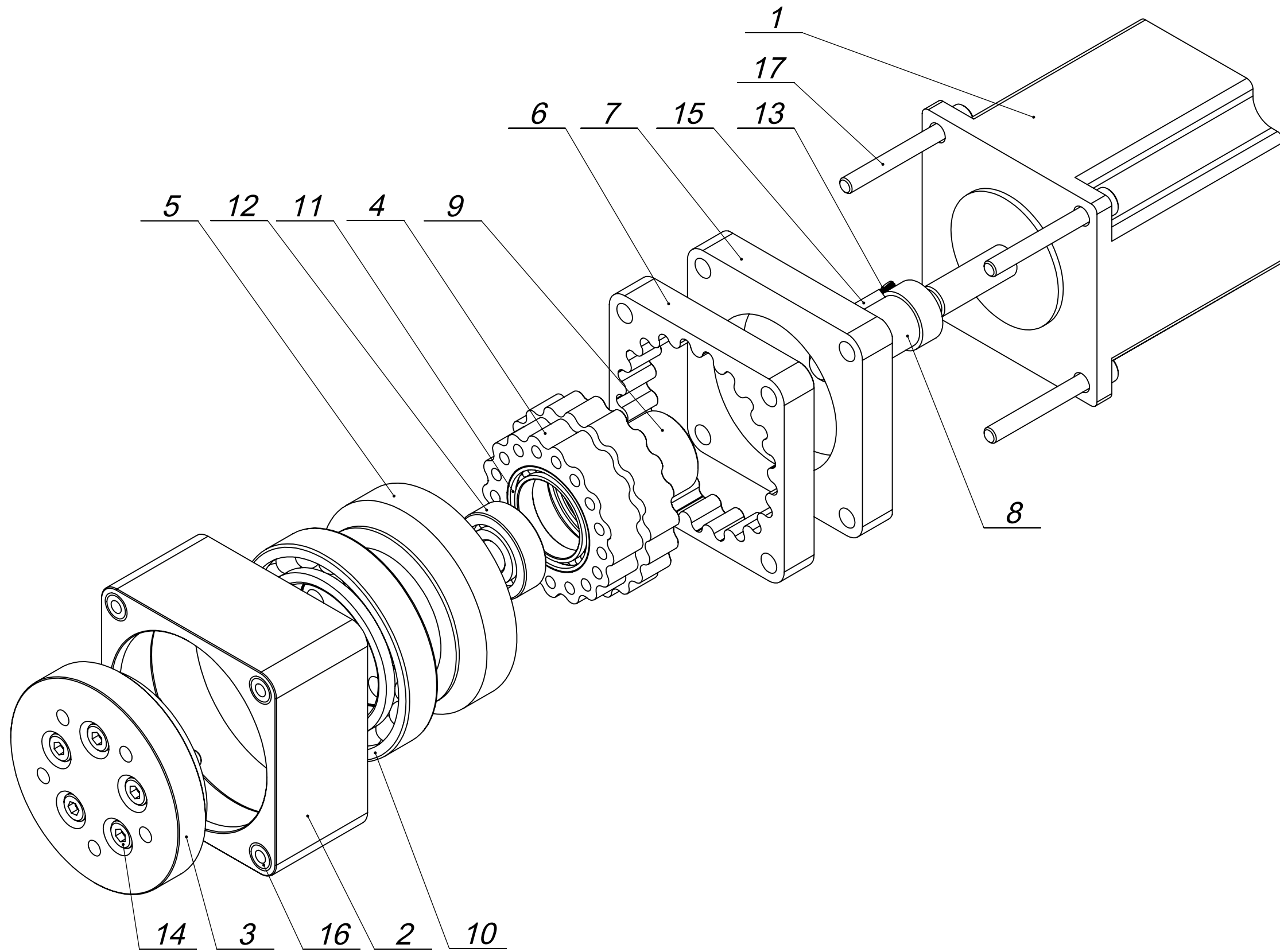
1. M. L. Yuan, S. K. Ong & A. Y. C. Nee (2008) Augmented reality for assembly guidance using a virtual interactive tool, International Journal of Production Research, 46:7, 1745-1767, DOI: 10.1080/00207540600972935
2. M. Loisier, “A brief history of Augmented Reality - Numerized,” <https://numerized.com/.https://numerized.com/augmented-reality/historyaugmented-reality/>
3. H. Brockwell, “Forgotten genius: the man who made a working VR machine in 1957,” TechRadar, Apr. 03, 2016. <https://www.techradar.com/news/wearables/forgotten-genius-the-man-who-made-a-working-vr-machine-in-1957-1318253>
4. B. Poetker, «A Brief History of Augmented Reality (+Future Trends & Impact)».- G2,2019-[Електронний ресурс]. URL:<https://www.g2.com/articles/history-of-augmented-reality>
5. WCS100, “English: Louis Rosenberg testing Virtual Fixtures, one of the first Augmented Reality systems ever developed (1992).,” Wikimedia Commons, May 04, 2014.https://commons.wikimedia.org/wiki/File:Louis_Rosenberg_Augmented_Reality_Rig.png (accessed Nov. 28, 2022).
6. 6 Uses of Augmented Reality for Manufacturing In Every Industry [Електронний ресурс] // LightGuide. – 2022. – Режим доступу до ресурсу: <https://www.lightguidesys.com/resource-center/blog/6-uses-of-augmented-reality-for-manufacturing-in-every-industry/>.
7. Augmented Reality (AR) based framework for supporting human workers in flexible manufacturing : ISSN 2212-8271 / . – Procedia CIRP, 2021. – 96 с.
8. Market spending on extended reality (XR) technologies worldwide from 2018 to 2023, by industry [Електронний ресурс] // Statista. – 2021. – Режим доступу до ресурсу: <https://www.statista.com/statistics/1096765/global-market-spend-on-xr-technologies-by-industry/>.
9. “Absolutely real: Virtual and augmented reality open new avenues in the BMW Group Production System,” www.press.bmwgroup.com.

- <https://www.press.bmwgroup.com/global/article/detail/T0294345EN/absolutely-real:-virtual-and-augmented-reality-open-new-avenues-in-the-bmw-group-production-system?language=en>
10. “Augmented Reality,” VDI Wissensforum. <https://www.vdi-wissensforum.de/news/augmented-reality-fuer-die-digitale-zukunft-der-industrie/> (accessed Nov. 28, 2022).
 11. “thyssenkrupp: HoloLens und Skype- kürzere Wartungszeiten bei Aufzügen,” News Center Microsoft Deutschland, Sep. 15, 2016. <https://news.microsoft.com/de-de/thyssenkrupp-hololens-und-skype-sorgen-fur-bis-zu-viermal-kurzere-wartungszeiten-bei-aufzugen-und-ermoglichen-sichere-komfortable-fortbewegung/> (accessed Nov. 28, 2022).
 12. J. Ciampa, “6 Uses of Augmented Reality for Manufacturing In Every Industry,” LightGuide, Feb. 23, 2022. <https://www.lightguidesys.com/resource-center/blog/6-uses-of-augmented-reality-for-manufacturing-in-every-industry/>
 13. S. Kumar, “Augmented Reality in Smart Manufacturing,” [blog.isa.org](https://blog.isa.org/augmented-reality-in-smart-manufacturing). <https://blog.isa.org/augmented-reality-in-smart-manufacturing>
 14. VIAR, “HOW TO USE AUGMENTED REALITY FOR TRAINING?,” Medium, May 20, 2018. <https://medium.com/@viARbox/how-to-use-augmented-reality-for-training-25ede8b6be08> (accessed Nov. 28, 2022).
 15. Boeing Tests Augmented Reality in the Factory — Режим доступа: [www.URL:https://www.boeing.com/features/2018/01/augmented-reality-01-18.page](https://www.boeing.com/features/2018/01/augmented-reality-01-18.page)— 19.01.2018 p.
 16. Radkowski, R. Object tracking with a range camera for augmented reality assembly assistance. J. Comput. Inf. Sci. Eng. 2016, 16, 011004.
 17. M. König et al., "MA2RA - Manual Assembly Augmented Reality Assistant," 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2019, pp. 501-505, doi: 10.1109/IEEM44572.2019.8978844.

18. Yamaguchi, Masahiro & Mori, Shohei & Mohr, Peter & Tatzgern, Markus & Stanescu, Ana & Saito, Hideo & Kalkofen, Denis. (2020). Video-Annotated Augmented Reality Assembly Tutorials. 1010-1022. 10.1145/3379337.3415819.
19. echo3D, "Top 10 AR Developer Tools Every Developer Should Know," Medium, Feb. 11, 2021. <https://medium.com/echo3d/top-10-AR-developer-tools-every-developer-should-know-32fef1471883> (accessed Nov. 28, 2022).
20. "Augmented reality updates now available - Latest News - Apple Developer," developer.apple.com. <https://developer.apple.com/news/?id=lvr5n745> (accessed Nov. 28, 2022).
21. A. Robertson, "Google ARCore gives Android users augmented reality without Tango," The Verge, Aug. 29, 2017. <https://www.theverge.com/2017/8/29/16219696/google-arcore-augmented-reality-platform-announce-release-pixel-samsung> (accessed Nov. 28, 2022).
22. "Echo3D raises \$4M for cloud authoring of 3D, AR, and VR content," VentureBeat, Oct. 05, 2021. <https://venturebeat.com/arvr/echo3d-raises-4m-for-cloud-authoring-of-3d-ar-and-vr-content/> (accessed Nov. 28, 2022).
23. Unity Technologies, "Unity - Manual: Vuforia," Unity3d.com, 2018. <https://docs.unity3d.com/2018.4/Documentation/Manual/vuforia-sdk-overview.html> (accessed Nov. 28, 2022).
24. "Handheld AR Template Quickstart," docs.unrealengine.com. <https://docs.unrealengine.com/4.27/enUS/SharingAndReleasing/XRDevelopment/AR/ARHowTos/HandheldAR/ARQuickStart/> (accessed Nov. 28, 2022).
25. "Control units | Visual Scripting | 1.5.2," docs.unity3d.com. <https://docs.unity3d.com/Packages/com.unity.visualscripting@1.5/manual/vs-control.html> (accessed Nov. 28, 2022).
26. "Blueprints Visual Scripting Overview," docs.unrealengine.com. <https://docs.unrealengine.com/5.0/en-US/overview-of-blueprints-visual-scripting-in-unreal-engine/> (accessed Nov. 28, 2022).
27. Чигрін О.В., «Автоматизований 3D принтер LineBar. Блок маніпулятора», диплом бакалавра, НТУУ КПІ ім. Ігоря Сікорського», Київ, 2021

28. Цибульник, С. О. Технології розроблення програмного забезпечення. Частина 1. Життєвий цикл програмного забезпечення [Електронний ресурс] : підручник для здобувачів ступеня бакалавра за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології» / Цибульник С. О., Барандич К. С. ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 3,43 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 270 с. – Назва з екрана.
<https://ela.kpi.ua/handle/123456789/50623>
29. “Event System | Unity UI | 1.0.0,” docs.unity3d.com.
<https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/EventSystem.html>
30. Falcón, Raúl. (2015). Modelización matemática de sistemas CAD en Edificación. Modelling in Science Education and Learning. 10.4995/msel.2015.3258
31. “Markerless vs. Marker-based AR with Examples - Aircards,” www.aircards.co.
<https://www.aircards.co/blog/markerless-vs-marker-based-ar-with-examples>
(accessed Dec. 07, 2021).

Додаток А



Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подп.	Дата

МД.ПБ1111.1702.001.01

Копировал

Формат А3

Пист

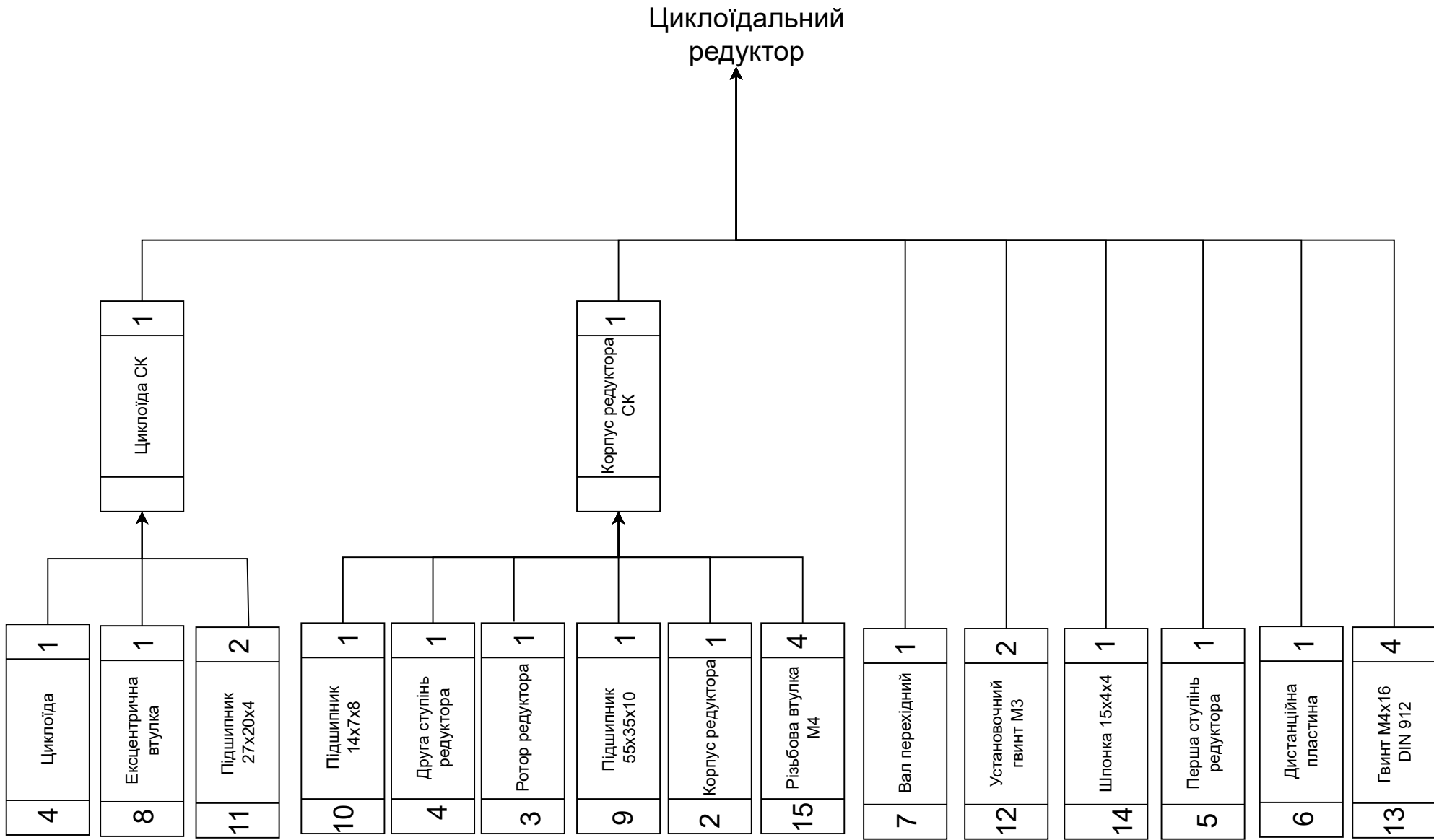
Файл: Редуктор СК2

Додаток Б

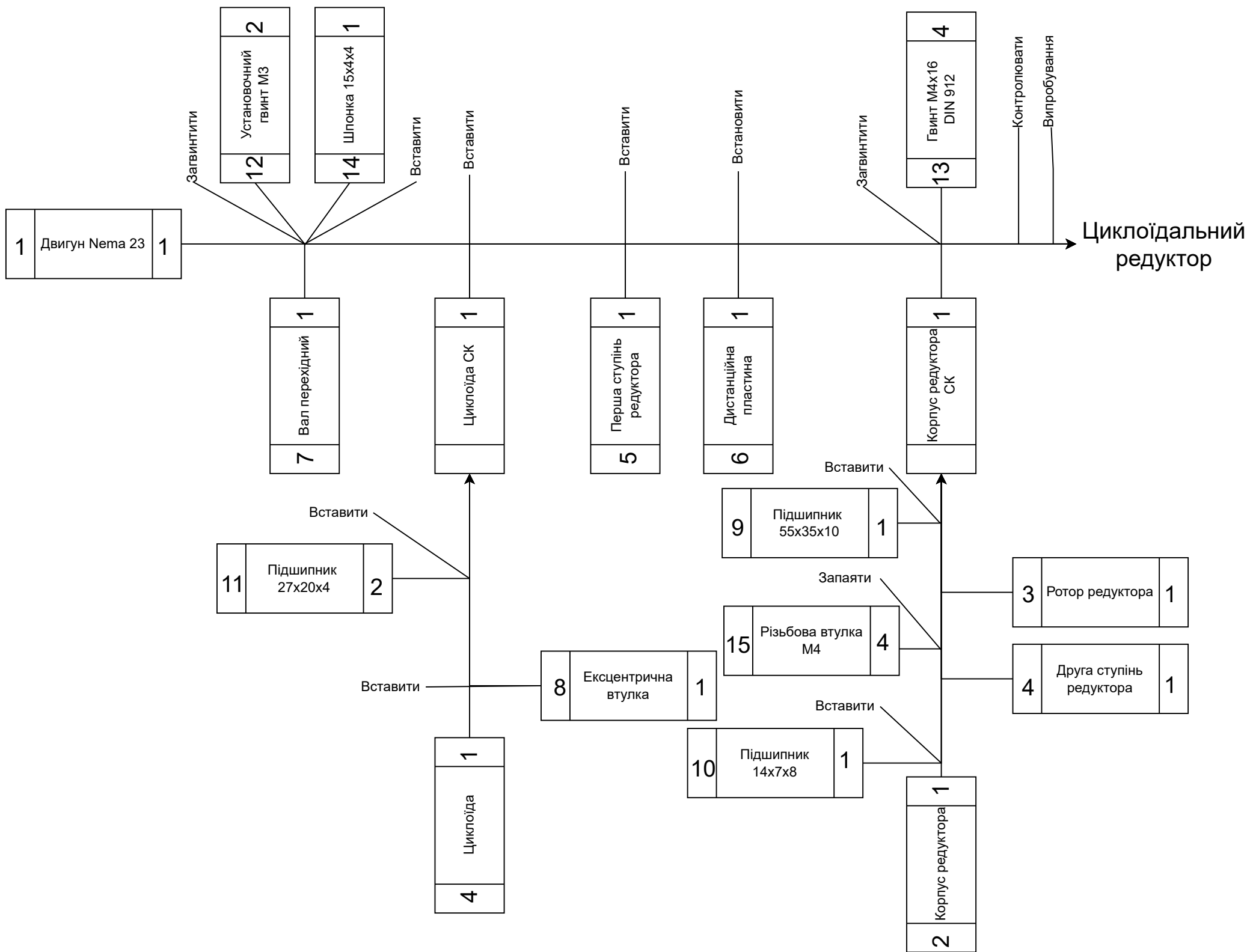
Форм.	Зона	Позн.	Позначення	Назва	Кіл.	Прим.
				Документація		
A3			МД.ПБ1111.1702.001.01	Циклоїдальний редуктор СК	1	
				Складальні одиниці		
				Деталі		
A3	1		МД.ПБ1111.1702.001.01.01	Двигун Nema 23	1	
A3	2		МД.ПБ1111.1702.001.01.02	Корпус редуктора	1	
A4	3		МД.ПБ1111.1702.001.01.03	Ротор редуктора	1	
A4	4		МД.ПБ1111.1702.001.01.04	Циклоїда	1	
A4	5		МД.ПБ1111.1702.001.01.05	Друга ступінь редуктора	1	
A4	6		МД.ПБ1111.1702.001.01.06	Перша ступінь редуктора	1	
A4	7		МД.ПБ1111.1702.001.01.07	Дистанційна пластина	1	
A4	8		МД.ПБ1111.1702.001.01.08	Вал перехідний	1	
A4 4	9		МД.ПБ1111.1702.001.01.09	Ексцентрична втулка	1	
				Покупні вироби		
		10		Підшипник 61907 2Z	1	
		11		Підшипник 608ZZ	1	
		12		Підшипник 6704ZZ	2	
				ДП.ПБ7117.1702.001.01		
Изм	Лист	№ докум.	Підп.	Дата		
Розроб.	Чигрін О.В.				Циклоїдальний редуктор СК Лит. Лист Листів 1 2 ПБФ, ПБ-11мп 2 курс	
Перев..	Барандич КС.					
Н.контр.						
Затв.						

[illegible]

Додаток В



Додаток Г



Додаток Д

Скрипт InfoBehavior, скрипт, що відображає анотацію

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Collections.Specialized;
using System.Security.Cryptography;
using UnityEngine;

public class InfoBehavior : MonoBehaviour
{
    const float SPEED = 6f;
    [SerializeField]
    Transform SectionInfo;

    Vector3 desiredScale = Vector3.zero;

    // Update is called once per frame
    void Update()
    {
        SectionInfo.localScale = Vector3.Lerp(SectionInfo.localScale, desiredScale,
Time.deltaTime * SPEED);
    }
    public void OpenInfo()
    {
        desiredScale = Vector3.one;
    }
    public void CloseInfo()
    {

```

```
        desiredScale = Vector3.zero;
    }
}
```

Скрипт Gaze, скрипт, що перевіряє напрям погляду користувача

```
using System.Collections;
using System.Linq;
using System.Collections.Generic;
using System.Net.NetworkInformation;
using System.Security.AccessControl;
using UnityEngine;
public class Gaze : MonoBehaviour {

    List<InfoBehavior> infos = new List<InfoBehavior>();

    void Start()
    {
        infos = FindObjectsOfType<InfoBehavior>().ToList();
    }

    // Update is called once per frame
    void Update()
    {
        if (Physics.Raycast(transform.position, transform.forward, out RaycastHit hit)){
            GameObject go = hit.collider.gameObject;
            if (go.CompareTag("hasInfo"))
            {
                OpenInfo(go.GetComponent<InfoBehavior>());
            }
        }
    }
}
```

```

        else
        {
            CloseAll();
        }
    }
}

void OpenInfo(InfoBehavior desiredInfo)
{
    foreach(InfoBehavior info in infos)
    {
        if (info == desiredInfo)
        {
            info.OpenInfo();
        }
        else { info.CloseInfo();
        }
    }
}

void CloseAll()
{
    foreach(InfoBehavior info in infos)
    {
        info.CloseInfo();
    }
}

}

```

Скрипт Face, скрипт, що обертає 3D об'єкти анотацій до користувача.

```
using System.Collections;
using System.Collections.Generic;
using System.Collections.Specialized;
using UnityEngine;

[ExecuteInEditMode]
public class Face : MonoBehaviour
{
    Transform cam;
    Vector3 targetAngle = Vector3.zero;
    void Start()
    {
        cam = Camera.main.transform;
    }

    // Update is called once per frame
    void Update()
    {
        transform.LookAt(cam);
        targetAngle = transform.localEulerAngles;
        targetAngle.x = 0;
        targetAngle.z = 0;
        transform.localEulerAngles = targetAngle;
    }
}
```

Скрипт Animation , скрипт, що керує анімаціями в додатку.

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

public class Animation : MonoBehaviour
{

    private Animator anim;
    public bool check;
    // Start is called before the first frame update
    void Start()
    {
        anim = GetComponent<Animator>();
        anim.SetInteger("Step", 0);
    }

    public void Animate ()
    {
        anim = GetComponent<Animator>();
        if (!check)
        {
            anim.SetBool("Action", true);
            check = true;
            anim.SetInteger("Step", 0);
        }
        else
        {
            {
```

```
        anim.SetBool("Action", false);
        check = false;
        anim.SetInteger("Step", 9);
    }

}

public void StepForward ()
{
    anim = GetComponent<Animator>();
    if (anim.GetInteger("Step") < 9)
    {
        anim.SetInteger("Step", anim.GetInteger("Step") + 1);
    }
}

public void StepBack()
{
    anim = GetComponent<Animator>();
    if (anim.GetInteger("Step") > 0)
    {
        anim.SetInteger("Step", anim.GetInteger("Step") - 1);
    }

}
}
```

Додаток Е

«02» 12 2022 р.

ЗАТВЕРДЖУЮ

Головний інженер-технолог
ТОВ «ЗАВОД ЄВРОФОРМАТ»
КУЦЕНКО Д.О.

АКТ

Виробничої апробації результатів магістерської дисертаційної роботи
Чигріна Олега Вячеславовича
«Використання доданої реальності при виконанні складальних робіт»

Дійсний акт констатує, що в умовах ТОВ «ЗАВОД ЄВРОФОРМАТ»,
проведено апробацію розробленого застосунку для виконання складальних
робіт із використанням доданої реальності.

Розроблений додаток відповідає потребам ТОВ «ЗАВОД ЄВРОФОРМАТ»
та буде використаний на ділянці складання ліфтового обладнання з метою
зменшення часу на підготовку до виконання складальної операції.

ЗАТВЕРДЖУЮ

Головний інженер-технолог
ТОВ «Завод Євроформат»,



КУЦЕНКО Д.О.

Додаток Ж

СПИСОК

наукових та навчально-методичних праць

Чигріна Олега Вячеславовича

№ п/п	Назва (мовою оригіналу)	Вихідні дані	Обсяг друк. стор.	Співавтори
1	2	3	4	5
1	Використання можливостей доданої реальності при виконанні складальних робіт	Збірник тез доповідей ХХІ Міжнародної науково-технічної конференції «ПРИЛАДОБУДУВАННЯ: стан і перспективи», ПБФ, КПІ ім. Ігоря Сікорського, 17-18 травня 2022 р., Київ, Україна, 2022. – с. 65-67.	1	Барандич К.С., Гладський М.М.
2	Використання доданої реальності при складанні циклоїдального редуктора	Вісник НТУУ «КПІ» «Приладобудування 2022», ПБФ м. Київ, Вип. 64(2), 2022. – прийнято до опублікування.	1	Барандич К.С., Гладський М.М.
3	Використання доданої реальності при складанні циклоїдального редуктора	Збірник праць ХVІІІ Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Ефективність та автоматизація інженерних рішень у приладобудуванні», 06-07 грудня 2022 р. К.: ПБФ, КПІ ім. Ігоря Сікорського. – 2022. – прийнято до опублікування.	1	Барандич К.С.