МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему «Віртуальний тренажер "Установка кутів прицілювання прицілу ПГ-4. Вертикальна наводка"»

за спеціальністю 122 «Комп'ютерні науки»
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ-71 Чичикало Євгеній Андрійович

Кваліфікаційна робота бакалавра захищена на засіданні ЕК з оцінкою _____ «____»____**2021р.**

Науковий керівник: _____ к.т.н. Кузнєцов Е.Г.
(підпис)

Голова комісії _____ Шифрін Д.М.
(підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань

Студент _____
(підпис)

Суми-2021

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук
Секція інформаційних технологій проектування
Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

ЗАТВЕРДЖУЮ

Зав. секції ІТП

_____В.В. Шендрик

«___»_____2021 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Чичикало Євгеній Андрійович

1 Тема роботи *Віртуальний тренажер "Установка кутів прицілювання прицілу ПГ-4. Вертикальна наводка"*

керівник роботи *Кузнєцов Едуард Геннадійович, к.т.н*

затверджені наказом по університету від *« 14» квітня 2021 р. № 0181-VI*

2 Строк подання студентом роботи *« 07 » червня 2021р.*

3 Вхідні дані до роботи *технічне завдання на розробку програмного додатку віртуального тренажеру вертикальних кутів наводки прицілу техніки*

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) *аналіз предметної області, моделювання, проектування програмного додатку, реалізація додатку віртуального тренажера вертикальних кутів наводки прицілу ПГ-4*

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

_____

6. Консультанти розділів роботи:

Розділ	**Консультант**	**Підпис, дата**	
		Завдання видав	**Завдання прийняв**

7.Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Розробка концепції проекту	18.01.21-20.01.21	
2	Обговорення роботи з керівником	21.01.21-23.01.21	
3	Затвердження теми	24.01.21-25.01.21	
4	Виконання розділу «Керування проектами»	26.01.21-08.02.21	
5	Розробка додатку	09.02.21-15.04.21	
6	Тестування додатку	16.04.21-21.04.21	
7	Розробка документації	15.02.21-07.05.21	
8	Передача програмного продукту замовнику	03.06.21-17.06.21	

Студент _____ Чичикало Є.А.

Керівник роботи _____ к.т.н. Кузнєцов Е.Г.

РЕФЕРАТ

Тема роботи «Віртуальний тренажер "Установка кутів прицілювання прицілу ПГ-4. Вертикальна наводка"».

Пояснювальна записка містить вступ, 3 розділи, висновки, список використаних джерел, 4 додатки. Загальний обсяг роботи – 87 сторінок.

Кваліфікаційна робота бакалавра має на меті проведення розробки додатку віртуального тренажеру вертикальних кутів наводки прицілу ПГ-4.

У ході роботи було проведено аналіз предметної області, огляд останніх досліджень та публікацій. Було виконано пошук аналогів розроблюваного додатку та їх опис.

Було виконано структурно-функціональний аналіз розроблюваного додатку та проведено проектування програмного продукту з використанням нотації UML.

Під час реалізації додатку було описано математичні залежності між об’єктами для обробки їх взаємодії, зображено структура додатку та проведеної її опис.

Було показано розроблені тривимірні моделі частин прицілу ПГ-4, які необхідні для виконання вертикальної наводки, описано метод конвертації формату та експорту тривимірних моделей.

Проведено опис розробленого інтерфейсу додатку, його робочої області та показано розміщення всіх компонентів в середовищі Unity.

У результаті виконаної роботи було розроблено додаток віртуальний тренажер встановлення вертикальних кутів прицілювання прицілу ПГ-4.

Призначення розробленого додатку полягає у його використанні у військових навчальних закладах для покращення якості та швидкості навчання курсантів, відпрацювання практичних навиків виконання наводки прицілу ПГ-4 та контроль успішності учнів.

Ключові слова: C#, .Net, Unity, SolidWorks, тривимірна модель, тренажер.

ЗМІСТ

ВСТУП 6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 7
1.1 Огляд останніх досліджень і публікацій 7
1.2 Аналіз програмних продуктів-аналогів 8
1.3 Постановка задачі 11
2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ 13
2.1 Моделювання використання віртуального тренажеру установки вертикальних кутів прицілювання прицілу ПГ-4 13
2.2 Моделювання віртуального тренажеру установки вертикальних кутів прицілювання прицілу ПГ-4 в IDEF0 14
2.3 Модель аналізу віртуального тренажеру установки вертикальних кутів прицілювання прицілу ПГ-4 18
2.4. Модель проектування 20
2.5 Модель реалізації 22
3 РЕАЛІЗАЦІЯ 23
3.1 Математичні залежності 23
3.2 Структура додатку 25
3.3 Етапи розробки 26
3.4 Розробка тривимірних моделей 26
3.5 Програмна реалізація додатку 34
3.6 Використання розробленого додатку 38
ВИСНОВКИ 44
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 46
Додаток А. 48
Додаток Б 58
Додаток В 68
Додаток Г 83

ВСТУП

У теперішній час комп’ютеризації є можливість створення віртуальних тренажерів для вивчення функціоналу реальних пристроїв. У даному випадку проводиться розробка частини віртуального тренажера для навідника артилерійської установки з використанням прицілу ПГ-4. А саме виконується розробка комп’ютерної програми в середовищу Unity, що детально відображає будову пристрою прицілювання і має функціонал для виконання прицілювання по вертикалі. Тривимірні моделі будуть розроблені у програмі SolidWorks з використанням розмірів, що були отримані в результаті вимірів деталей прицілу.

Основною причиною для створення даного програмного забезпечення була потреба прискорення освоєння навчального матеріалу курсантами та відпрацювання практичних навичок щодо прицілювання артилерійської установки. Внаслідок даних фактів було прийнято рішення створити програмне забезпечення, що може дозволити курсантам отримувати практичні навички прицілювання.

Даний програмний продукт розроблюється паралельно з іншим студентом як комплекс для тренування та оцінювання навиків курсантів кафедри військової підготовки.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

На даний момент для навчання військових фахівців використовують лекційні матеріали та презентації. Вказані методи не надають можливості в повній мірі описати будову та принципи роботи певних механізмів. Даний факт ускладнює навчання курсантів. Найбільш ефективним можна вважати проведення навчання на реальних об’єктах, проте в цьому випадку значно зростає час навчання, також варто враховувати високу вартість.

Для вирішення даної проблеми було прийняте рішення розробити додаток для персональних комп’ютерів на базі Windows. Вікно додатку буде містити тривимірну модель прицілу ПГ-4, рухомі механізми, що необхідні для встановлення вертикальних кутів прицілювання. Користувач додатку за допомогою миші або клавіатури зможе виконувати взаємодію з рухомими частинами прицілу. Меню буде надавати можливість перегляду довідкової інформації.

Для виконання поставленої задачі були обрані три основні інструменти: SolidWorks, Unity, Microsoft Visual Studio 2019.

SolidWorks дозволяє створювати тривимірні моделі та встановлювати їм параметри дизайну (колір, матеріал). Дана САПР має широкий функціонал та достатньо простий інтерфейс. У випадку виникнення проблем при розробці моделей можна виконати ознайомлення з відповідними пунктами книги «Основы мелирования в SOLIDWORKS».

Unity дозволяє реалізувати інтерфейс додатку, розміщення тривимірних об’єктів на сцені, проте найбільш важливим варто вважати той факт, що дане середовище має функціонал обробки положення фігур на сцені та взаємодії з ними

за допомогою миші з використанням скриптів на мові програмування C#. Ознайомитися з основами роботи в Unity можна на електронних ресурсах (learn.unity.com, habr.com/ru/post/161463, learn.unity.com/project/beginner-gameplay-scripting) і можна вивчати офіційну документацію на сайті docs.unity3d.com [1].

Microsoft Visual Studio 2019 можна використовувати для написання скриптів мовою програмування C#. Використання саме цієї версії програми обумовлене можливістю зміни скриптів в режимі реального часу. Це реалізовано паралельною роботою VS та Unity. Дана властивість дуже корисна для налагодження програми. Для ознайомлення з особливостями C# можна використовувати сайт Metanit [2]. Також слід не забувати про документація мови програмування C# [3].

1.2 Аналіз програмних продуктів-аналогів

У навчальній програмі військових кафедр та ліцеїв уже не перший рік використовуються різного роду симулятори стрільбищ, траєкторії польоту снарядів та інші.

Розроблюваний додаток є унікальним у своєму роді, адже досі не було випущено тренажерів з подібним функціоналом. Проте варто звернути увагу на тренажери, що використовуються на інших спеціальностях окрім артилерійської. Ознайомлення з ними допоможе спростити процес розробки інтерфейсу розроблюваного додатку та визначення додаткових функцій.

Наприклад, розглянемо підготовку стрільців-зенітників. В даному випадку використовується тренажер, що виконаний у вигляді шолому віртуальної реальності, що містить вбудовану систему орієнтування в просторі та макет реальної пускової ПЗРК (рис. 1.1).

Рисунок 1.1 – Зображення елементів тренажеру стрільців-зенітників

Використання подібних віртуальних тренажерів сильно знижає вартість навчання операторів, адже зникає необхідність використання ракети, адже постріл ракетою з ПЗРК коштує приблизно 16 тисяч гривень.

Після виконаних дій стрілець-зенітник отримує оцінку та інформується про допущені помилки. На екрані відображається повітряна та наземна обстановка під час підготовки до стрільб та під час їх проведення, візуально відображається результат виконаного пострілу.

Також часто використовуються інтерактивні лазерні стрілецькі тренажери (рис. 1.2).

Даний вид тренажерів дозволяє виконувати стрільбу зі стрілецької зброї, гранатометів. Для віртуального тиру існують різні сценарії роботи. Так завдяки ньому можна відпрацьовувати прицільну, швидкісну та тактичну стрільби.

Для ведення стрільби доступні такі види мішеней:

- статичні;
- динамічні;
- такі, що з’являються.

Рисунок 1.2 – Інтерактивний тир

Також відомий факт використання автомобільних тренажерів на базі БМ 9А33 «Оса» (рис. 1.3).

Рисунок 1.3 – Автомобільний тренажер БМ 9А33 «Оса»

Даний тренажер містить кабіну водія, пульт інструктора. Пульт керування дозволяє інструктору відслідковувати дії курсантів, моделювати такі ситуації:

- зміна дня та ночі;
- перегрів двигуна;
- обстріл.

Тренажер дозволяє відпрацьовувати навики водіння на різних покриттях дорожнього полотна та на пересіченій місцевості. Враховуючи великий об’єм та потужність двигуна реальної бойової машини, використання віртуальної моделі дозволяє економити багато коштів на паливі та обслуговуванні техніки.

Після ознайомлення з існуючими тренажерами можна зробити висновок, що спільною рисою для них є простий та зручний у використанні інтерфейс без зайвих елементів. Адже увага курсантів під час роботи з ними повинна бути сконцентрована на основній задачі, а саме вивчення функціоналу реальних пристроїв та відпрацювання практичних навиків при роботі з ними.

1.3 Постановка задачі

Основною метою розробки програмного продукту є створення віртуального тренажеру навідників артилерійських установок на базі прицілу ПГ-4, в даному випадку проводиться розробка механізму виконання встановлення вертикальних кутів наводки. Віртуальний тренажер дозволить проводити практичні заняття спрямовані на отримання та покращення навиків наведення прицілу ПГ-4 для великої кількості курсантів, що було раніше неможливо через наявність однієї одиниці необхідної техніки або взагалі її відсутність.

Серед поставлених задач можна виділити наступні:

- виявлення актуальності навчання за допомогою тренажерів;

- розробка моделі майбутнього тренажеру згідно наявних вимог щодо навчального процесу;
- розробка тривимірних моделей елементів прицілу ПГ-4 в SolidWorks;
- розміщення тривимірних моделей в середовищі Unity;
- розробка інтерфейсу додатку та реалізація функціоналу;
- тестування.

Готовий програмний продукт повинен бути сумісний з операційними системами на базі Windows. Для його роботи потрібна установка компонентів необхідних для роботи додатків на базі Unity.

Більш детально поставлені завдання та особливості додатку описані в технічному завданні Додатку А.

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ДОДАТКУ

Перед процесом розробки додатку необхідно виконати систематизацію самого процесу. Це можливо зробити виконавши моделювання процесу розробки програмного продукту з точки зору управління проектами з використанням діаграм нотацій UML, IDEF та інших. Даний підхід дозволить точно описати кроки розробки додатку та завчасно уникнути виконання зайвих кроків, що значно оптимізує процес реалізації програмного додатку.

2.1 Моделювання використання віртуального тренажеру установки вертикальних кутів прицілювання прицілу ПГ-4

Діаграма варіантів використання в UML показана на рисунку 2.1 [4]. Вона відображає доступні користувачу функції додатку, а саме:

- можливість запуску робочого простору з автоматично сформованим завданням;
- можливість запуску робочого простору з ручними введення завдання;
- виконання змін значень прицілу;
- отримання результатів виконання поставленого завдання;
- перегляд довідкової інформації.

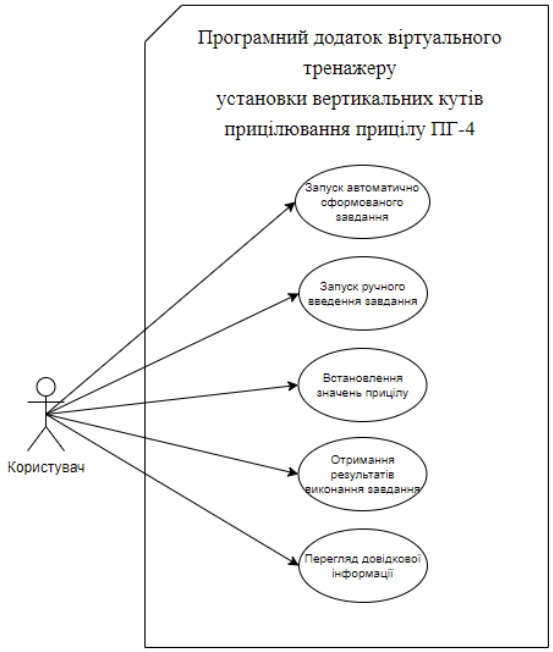

Рисунок 2.1 – Діаграма варіантів використання

2.2 Моделювання віртуального тренажеру установки вертикальних кутів прицілювання прицілу ПГ-4 в IDEF0

2.2.1 Функціональне моделювання віртуального тренажеру установки вертикальних кутів прицілювання прицілу ПГ-4

Виконаємо побудову функціональної моделі програмного продукту з використанням IDEF0 [5].

В даному випадку вхідними даними буде команда завдання наведення.

Серед обмежень будуть:

- інструкція щодо виконання наведення;
- Нормативи оцінки виконання даної вправи;
- рекомендації дипломного керівника.

В якості ресурсів зображено програмне, апаратне забезпечення та користувачі додатку.

На виході отримаємо оцінку навиків курсантів та статистичні дані щодо успішності виконання даної вправи курсантами.

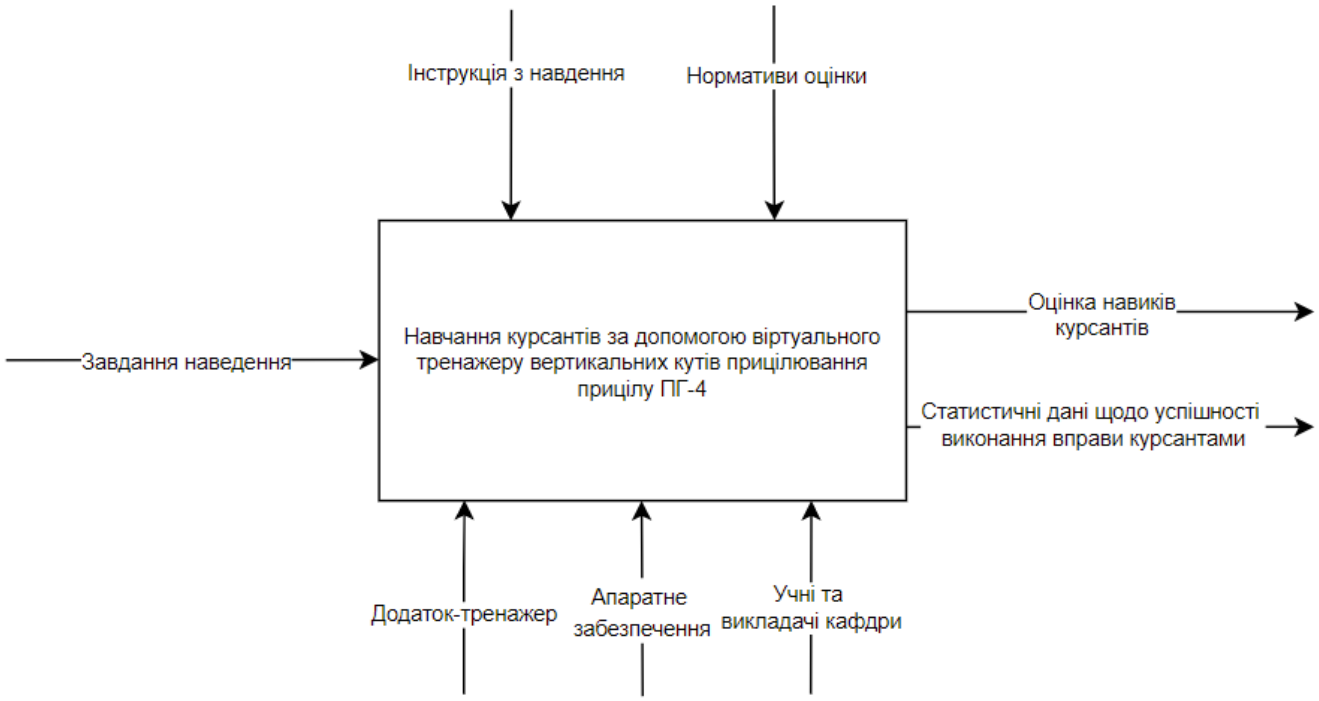

Рисунок 2.2 – IDEF0

Виконаємо побудову моделі розробки програмного продукту з використанням IDEF0 (рис. 2.3).

В даному випадку вхідними даними будуть отримані знання про предметну область у ході її дослідження.

Серед обмежень будуть:

– технічне завдання;
– вимоги замовника;
– рекомендації дипломного керівника.

В якості ресурсів зображено програмне забезпечення необхідне для розробки додатку.

На виході отримаємо готовий віртуальний тренажер установки вертикальних кутів прицілювання прицілу ПГ-4.

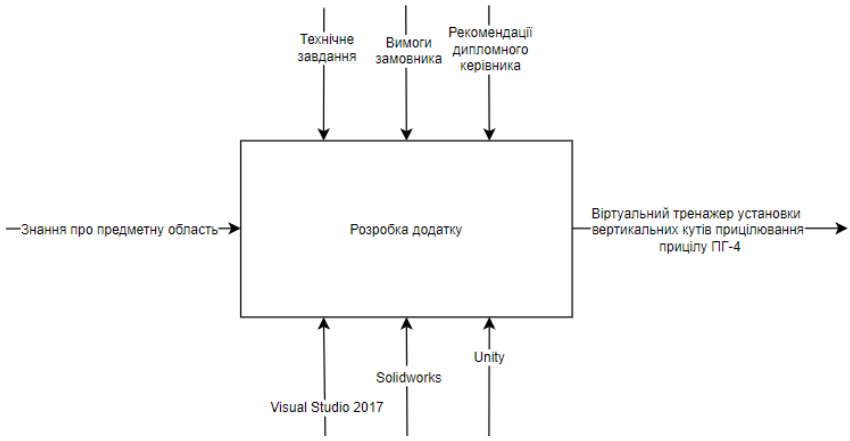

Рисунок 2.3 – IDEF0 розробки додатку

2.2.2 Функціональне моделювання додатку в IDEF0

Перший рівень декомпозиції IDEF0 дозволяє відобразити взаємозв’язки між інформаційними потоками. Дана діаграма показує послідовність роботи модулів роботи (Рис. 2.4). На рисунку 2.5 діаграма відображає послідовність процесу розробки додатку.

Як можна побачити на діаграмі після запуску додатку користувач виконує запуск робочого простору з можливими опціями вибору завдання, далі виконується встановлення значень прицілу, після завершення роботи користувач отримує результат виконаної роботи.

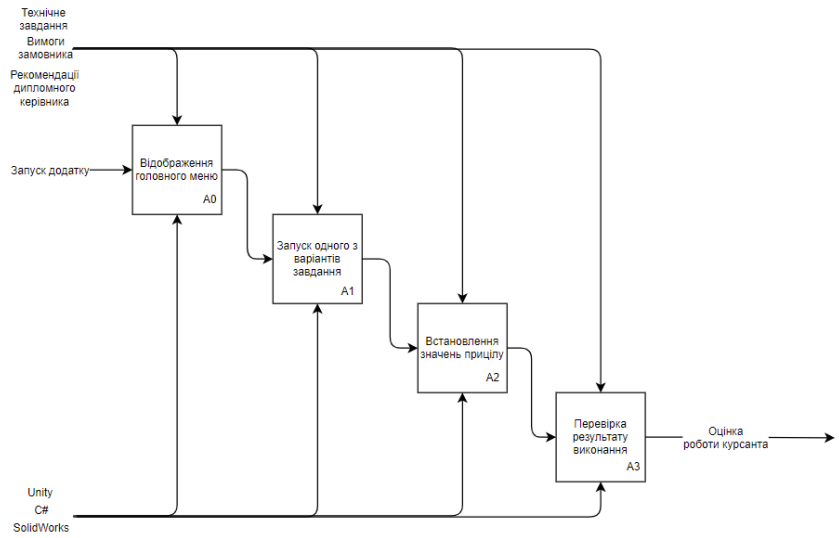

Рисунок 2.4 – Перший рівень декомпозиції IDEF0

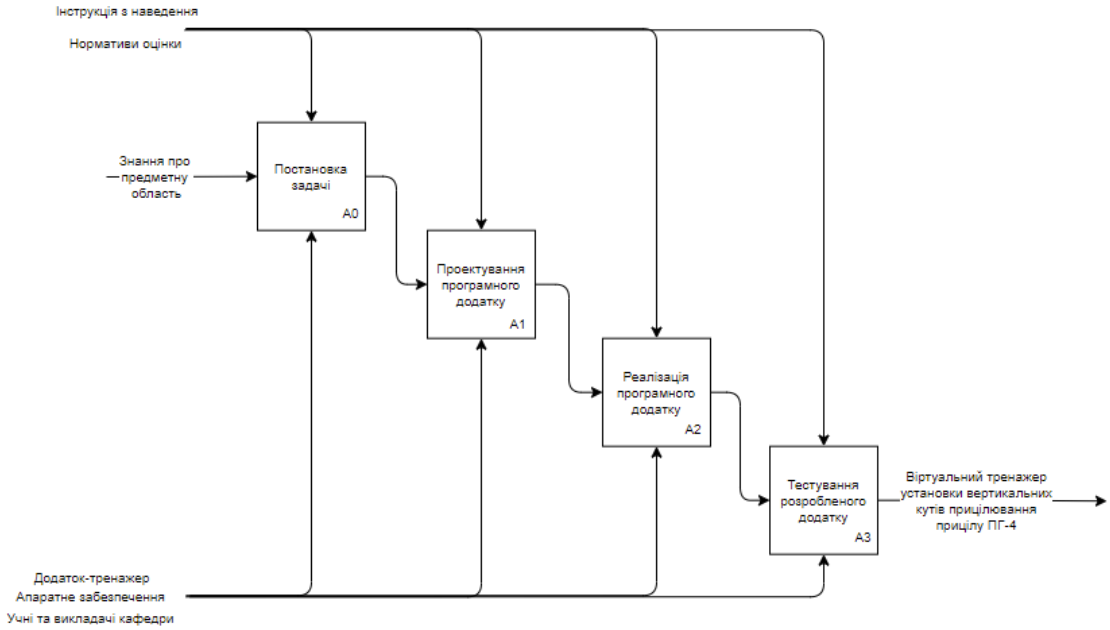

Рисунок 2.5 – Перший рівень декомпозиції IDEF0 процесу розробки

2.2.3 Структурне моделювання віртуального тренажеру установки вертикальних кутів прицілювання прицілу ПГ-4

Діаграма автоматів описує поведінку інформаційної системи у всіх можливих положеннях класів даної системи.

Діаграма автоматів показана на рисунку 2.6.

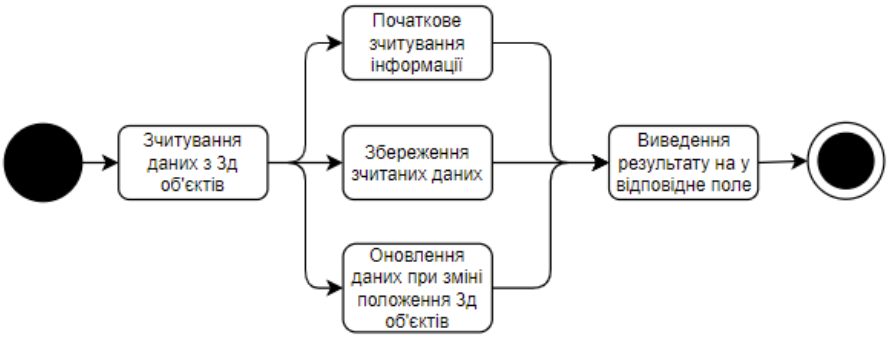

Рисунок 2.6 – Діаграма автоматів

2.3 Модель аналізу віртуального тренажеру установки вертикальних кутів прицілювання прицілу ПГ-4

2.3.1 Діаграма класів аналізу.

Діаграма класів показує класи інформаційної системи та їх зв’язки між собою. Вона є досить схожою на стандартну діаграму класів [6].

Діаграму класів аналізу показано на рисунку 2.7.

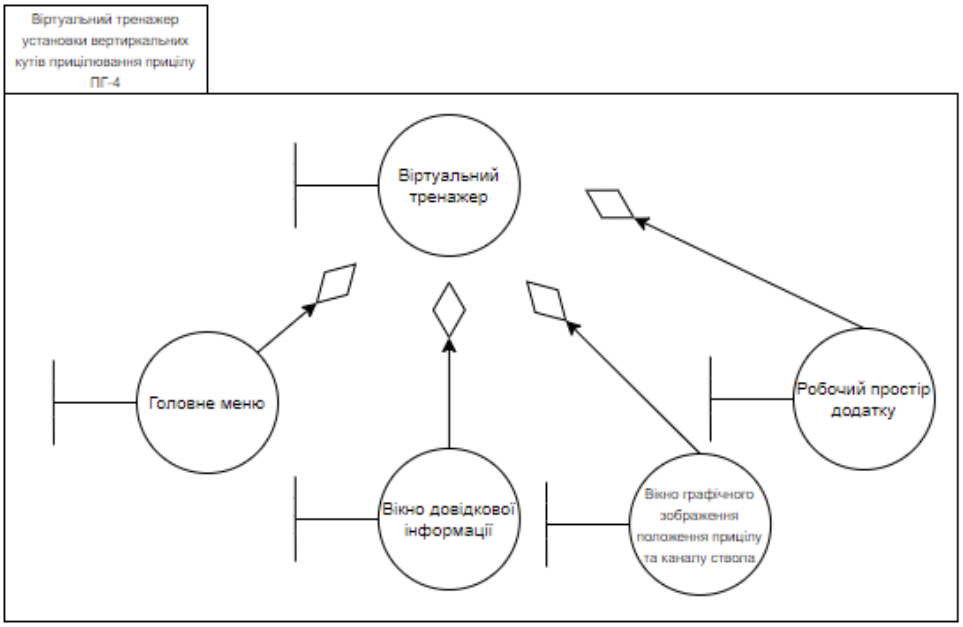

Рисунок 2.7 – Діаграма класів аналізу

2.3.2 Діаграми послідовності.

Діаграма послідовності відображає послідовність взаємодій між елементами інформаційної системи. Елементами можна вважати користувача та додаток, яким він буде користуватися. На певні дії користувача будуть надаватися певні реакції додатку і в певному порядку [7].

Діаграму послідовності показано на рисунку 2.8.

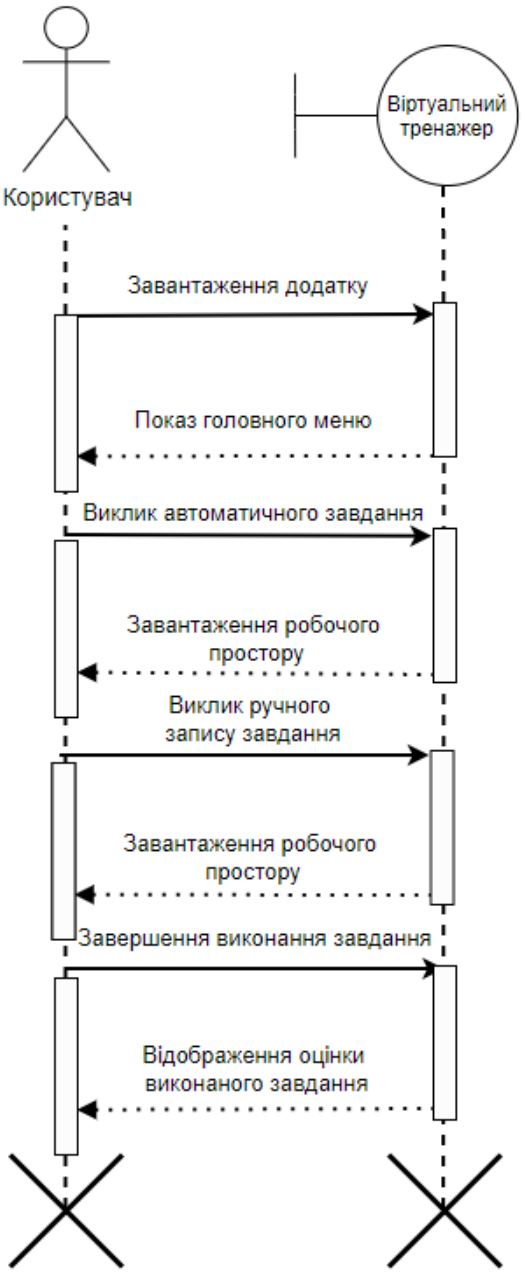

Рисунок 2.8 – Діаграма послідовності

2.3.3 Діаграми комунікації.

Діаграма комунікації відображає шлях, яким виконується взаємодія між модулями програмного продукту.

Діаграму комунікації показано на рисунку 2.9.

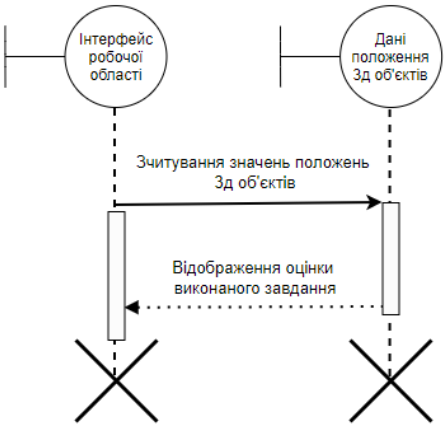

Рисунок 2.9 – Діаграма комунікації

2.4. Модель проектування

3.4.1 Діаграми класів.

Діаграма класів відображає класи, їх атрибути та функції, які виконуються у поточному класі.

Діаграму класів показано на рисунку 2.10.

Рисунок 2.10 – Діаграма класів

2.4.2 Діаграми діяльності.

Діаграма діяльності у вигляді блок-схеми описує поведінку програмного продукту. Тобто можливі варіанти використання додатку та реакції на дії користувача [8].

Діаграму діяльності показано на рисунку 2.11.

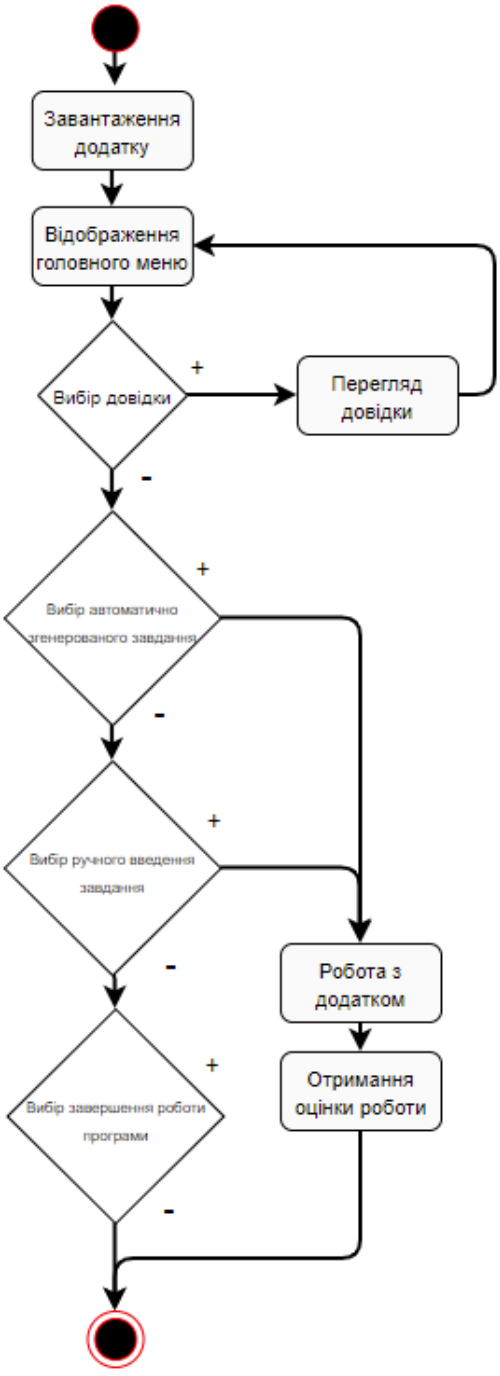

Рисунок 2.11 – Діаграма діяльності

2.5 Модель реалізації

2.5.1 Діаграма розгортання.

Діаграма розгортання відображає зв’язок між програмним та апаратним забезпеченням. Це дозволяє побудувати фізичну модель розгортання програмних компонентів на апаратних компонентах.

Діаграму розгортання показано на рисунку 2.12.

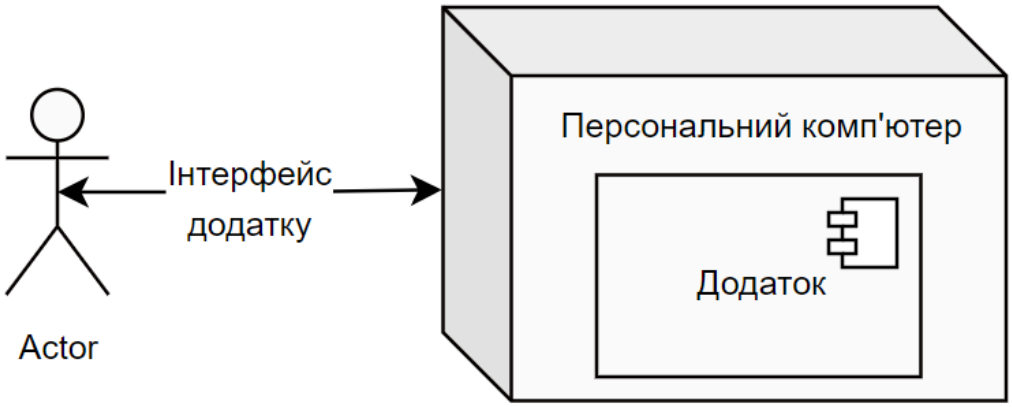

Рисунок 2.12 – Діаграма розгортання

2.5.2 Управління проектами

Управління проектами дозволяє визначити вимоги до проекту, сформувати чіткі і зрозумілі цілі, враховувати проектні обмеження, а саме ресурси, ризики, дедлайни, якість. Також виконується детальне планування, організація та контроль виконання поставлених завдань відповідальними за них особами.

Більш детально з розділом «Управління проектами» можна ознайомитися в Додатку Б.

3 РЕАЛІЗАЦІЯ

3.1 Математичні залежності

Перед програмною реалізацією керування рухомими частинами прицілу необхідно провести аналіз їх взаємодії між собою. Для цього необхідно виконати обчислення коефіцієнтів співвідношення обертання усіх пар зв’язаних маховиків.

Для отримання значення прицілу необхідно враховувати кількість поділок грубої шкали механізму вертикальної наводки. Для неї можливі значення в діапазоні від -1 до 12. Тобто всього в сумі отримаємо 14 поділок, кількість яких відповідає повним обертам маховика точної шкали. Точна шкала містить 100 поділок. В результаті отримаємо мінімальне значення $-1 \cdot 100 = -100$ та максимальне значення $12 \cdot 100 = 1200$.

Для зв’язку даних елементів прицілу необхідно встановити співвідношення між обертанням кожного з них. Для цього за допомогою моделі грубої шкали можна визначити кут між поділками. На рисунку 3.1 можна побачити, що даний кут рівний 15,47°. Оскільки один оберт маховика точної шкали рівний 360°, то на основі цих значень можна визначити коефіцієнт для співвідношення обертання. Виконаємо розрахунок $360 / 15{,}47 = 23{,}27$. Варто враховувати, що при експорті тривимірної моделі можуть виникнути невеликі зміщення положення елементів, тому отримане значення коефіцієнту співвідношення може корегуватися в діапазоні $\pm 0{,}5$ під час тестування.

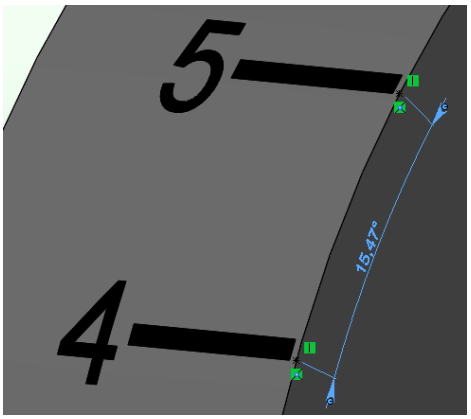

Рисунок 3.1 – Кут між поділками грубої шкали

Для забезпечення співвідношення повороту маховика каналу ствола та правої шкали панелі узгодження було використано формулу $R_{ш} = \frac{R_к}{2}$, де $R_{ш}$ кут повороту правої шкали панелі узгодження, $R_к$ кут повороту маховика каналу ствола. В реальній башті самохідної артилерійської установки для зміни положення ствола необхідно виконувати велику кількість обертів маховика, через це для зручності було вирішено обмежити кут повороту маховика до 60°.

Зі шкалами панелі узгодження пов’язані кути нахилу ліній прицілу та каналу ствола на графіку. Значення кута повороту правої шкали (вісь каналу ствола) рівна куту нахилу відповідної лінії на графіку, така сама залежність діє для лівої шкали (положення прицілу).

3.2 Структура додатку

Структура розроблюваного додатку формується на основі базової структури Unity, в якій найбільш важливими є «Components», які містять об’єкти, властивості об’єктів, засоби їх обробки та зберігають дані.

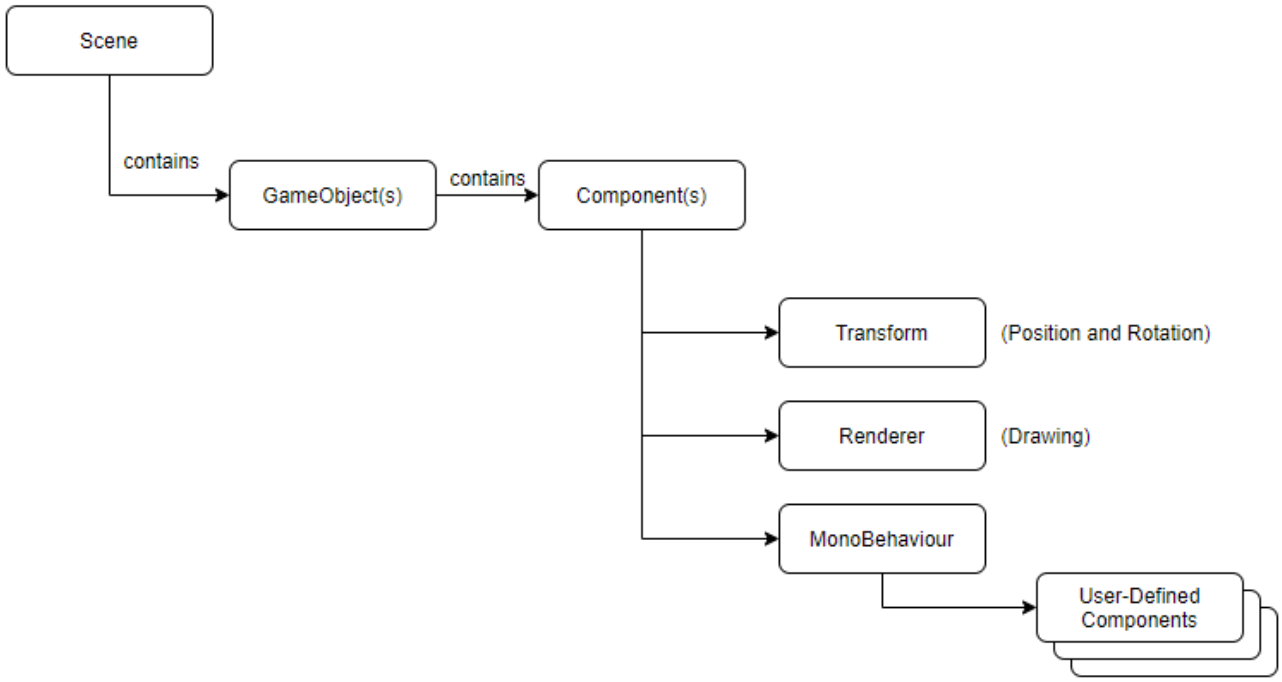

Рисунок 3.2 – Структура додатку

Схема на рисунку 3.2 показує, що Scene є колекцією GameObjects, GameObjects в свою чергу є колекцією Components, які можуть реалізовувати та включати:

- Системи (камери, фізика);
- Дані (конфігурації, анімації, текстури);
- Поведінки (ігрові механіки та сценарії подій).

MonoBehaviour є колекцією компонентів, що створюються користувачем.

3.3 Етапи розробки

Під час реалізації додатку були виконані наступні етапи:

- Ознайомлення з механізмом роботи прицілу ПГ-4;
- Зняття розмірів деталей прицілу;
- Побудова тривимірних моделей в SolidWorks;
- Експорт розроблених моделей в Blender та їх конвертація в формат FBX.
- Імпорт складових частин прицілу до середовища Unity, налаштування їх положення та матеріалів;
- Розробка інтерфейсу додатку;
- Розробка скриптів для рухомих об’єктів та їх зв’язки між собою;
- Тестування роботи додатку.

3.4 Розробка тривимірних моделей

Для розробки тривимірних моделей деталей прицілу ПГ-4 було використано САПР SolidWorks. Розміри та кольори деталей були отримані на основі проведених вимірювань та фотографій реальної моделі прицілу.

Розроблені тривимірні моделі необхідних деталей прицілу ПГ-4 показано на рисунках 3.3-3.11.

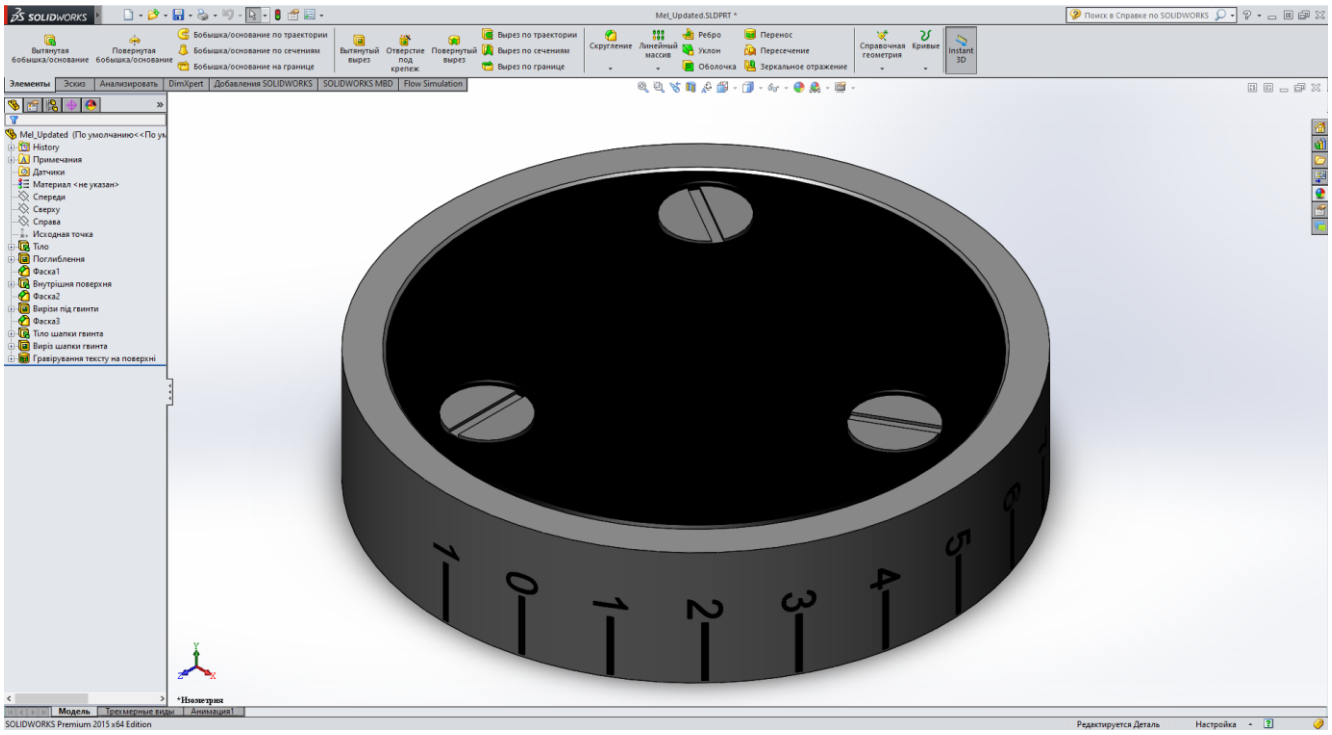

Рисунок 3.3 – Груба шкала прицілу

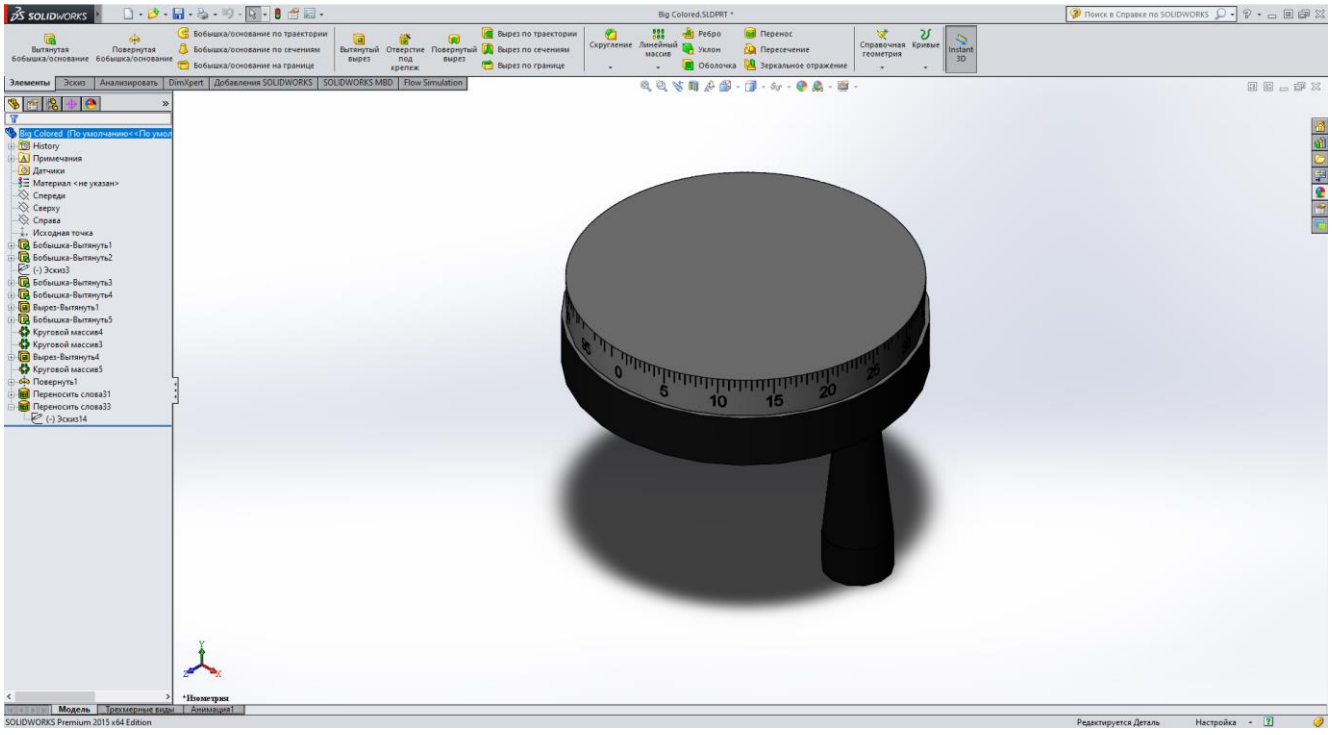

Рисунок 3.4 – Маховик точної шкали прицілу

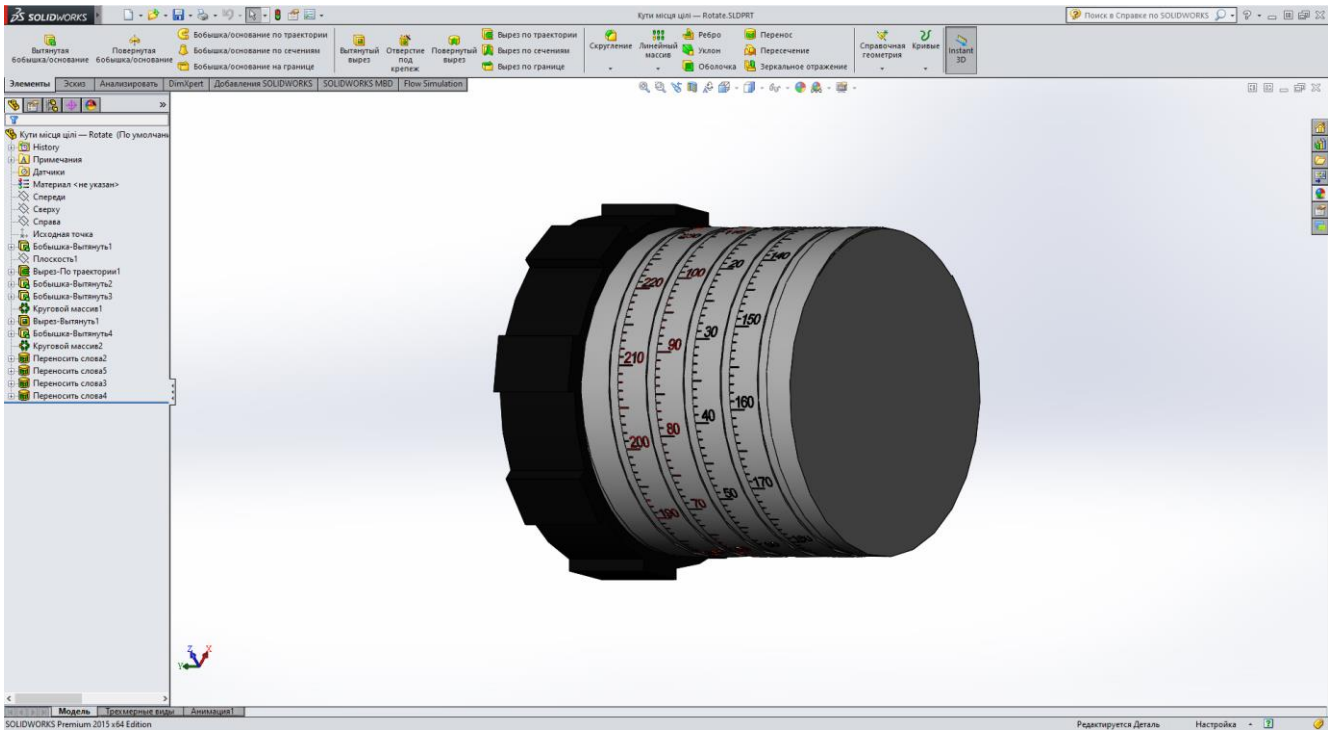

Рисунок 3.5 – Маховик місця кутів цілі

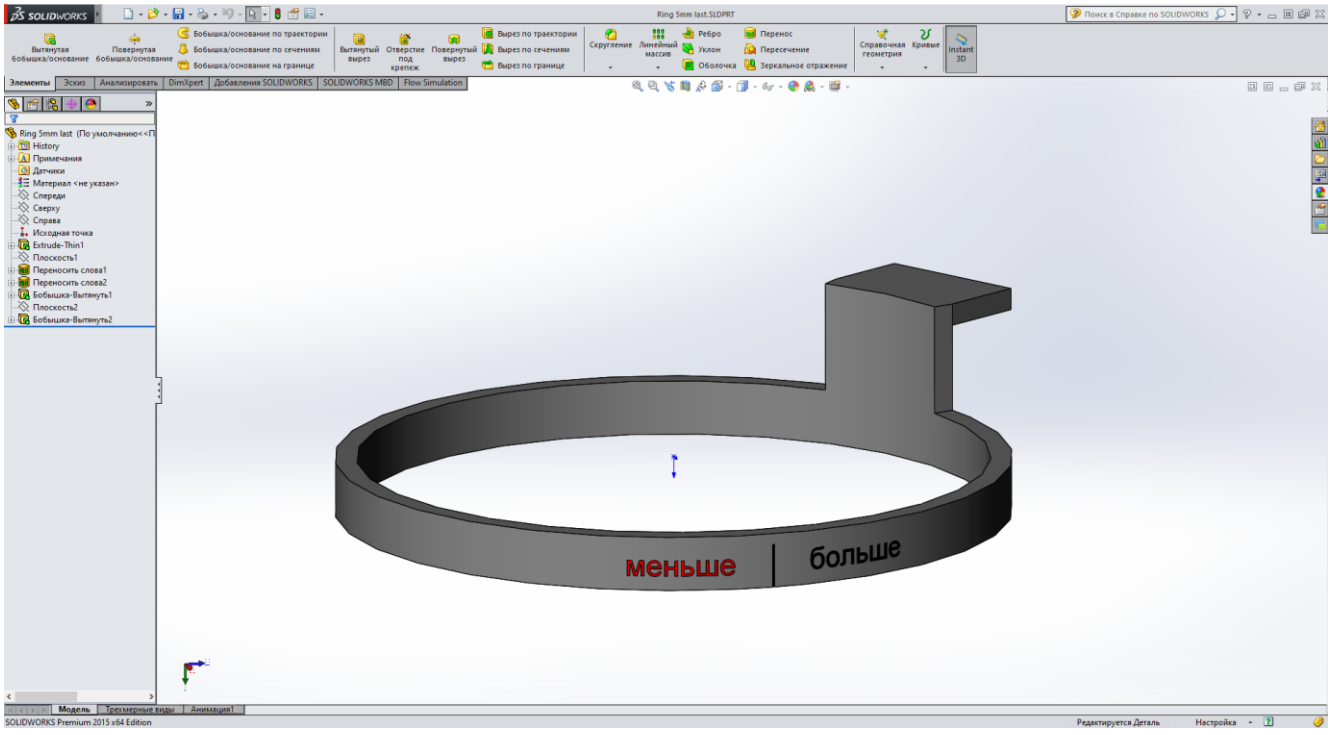

Рисунок 3.6 – Кільце-показник значення кутів місця цілі

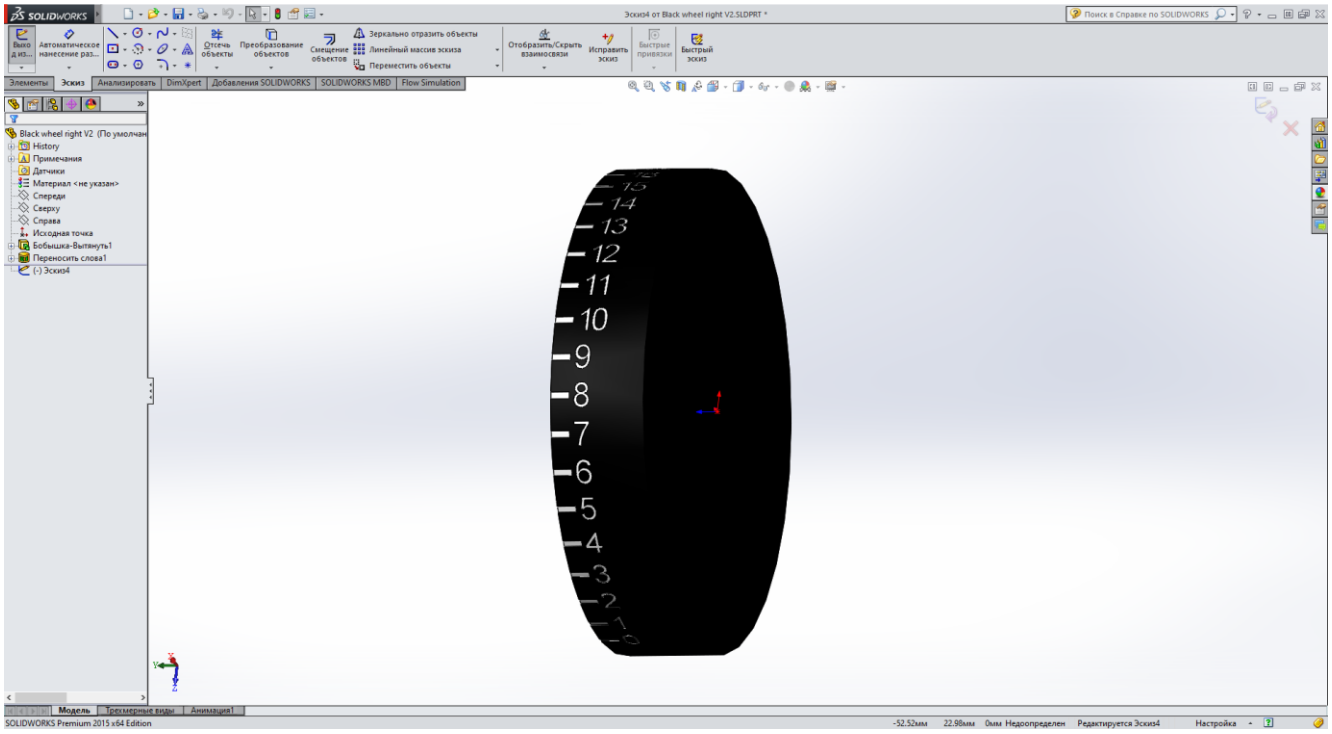

Рисунок 3.7 – Права шкала панелі узгодження

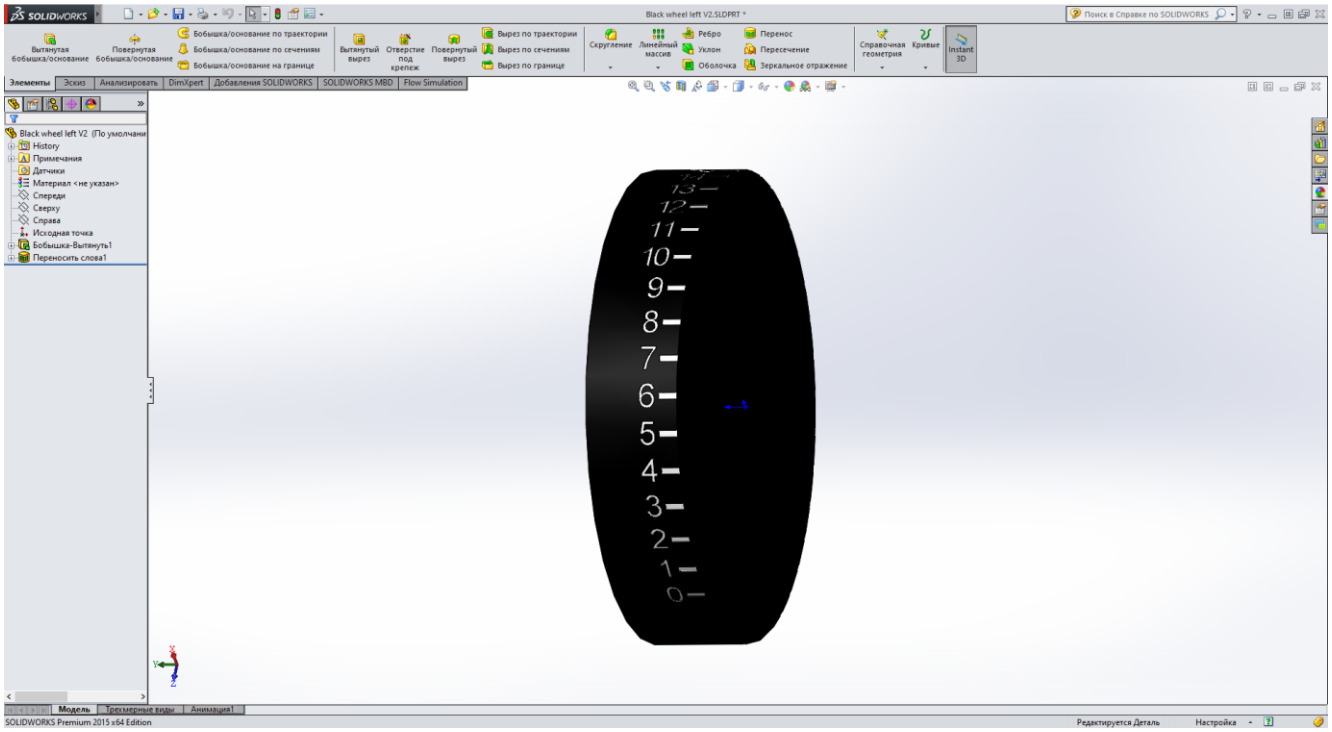

Рисунок 3.8 – Ліва шкала панелі узгодження

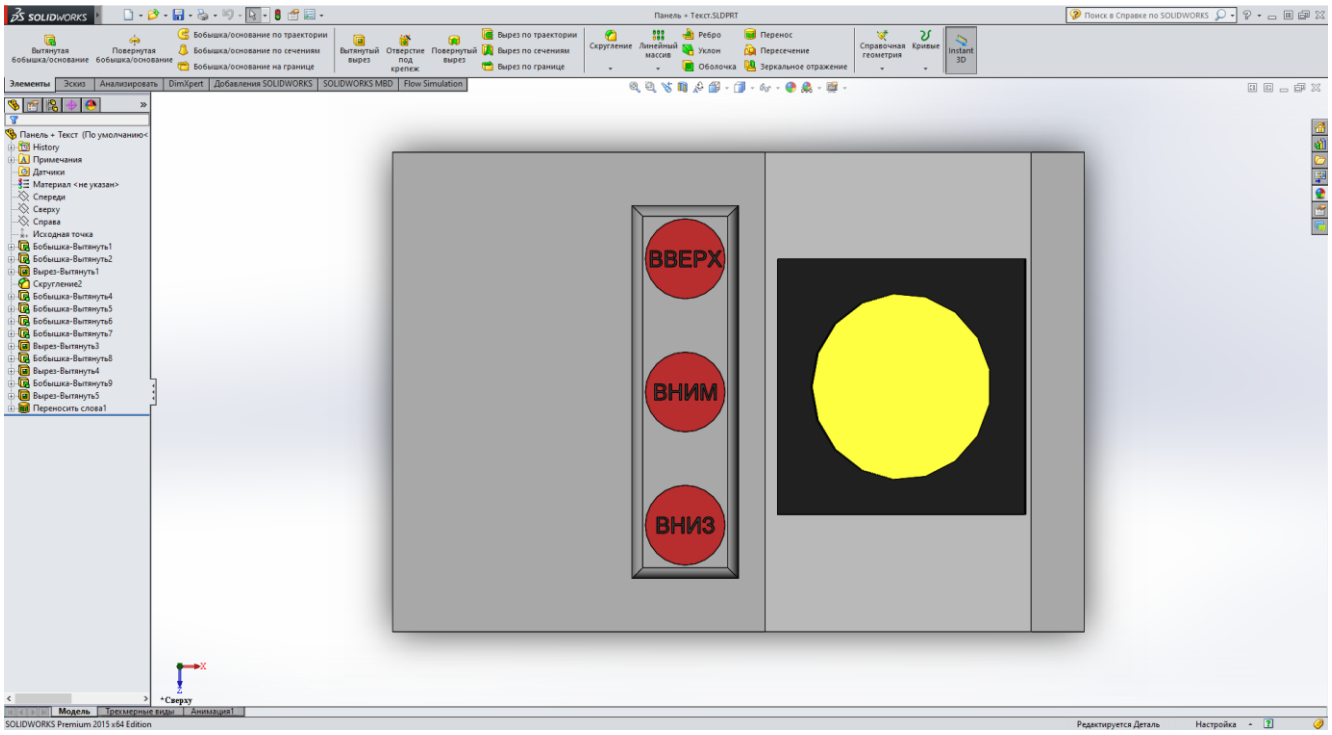

Рисунок 3.9 – Корпус панелі узгодження

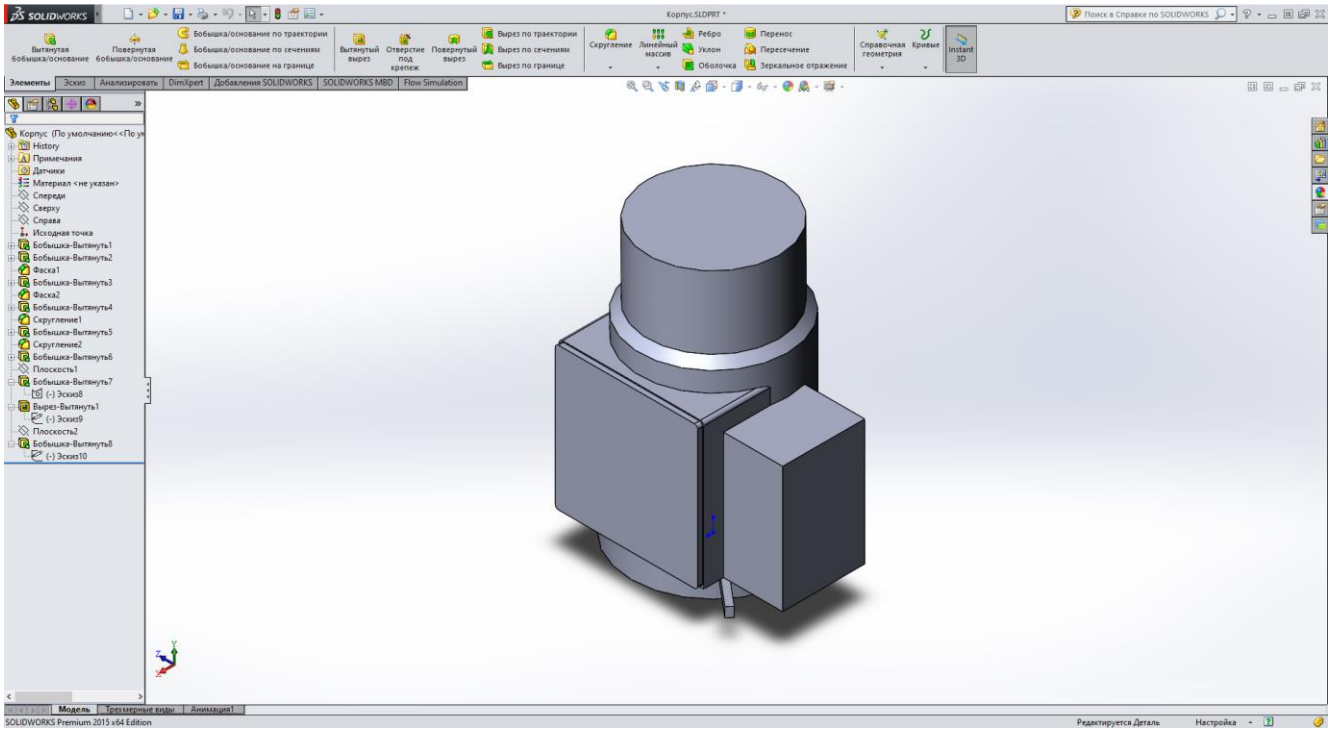

Рисунок 3.10 – Частина корпусу прицілу ПГ-4

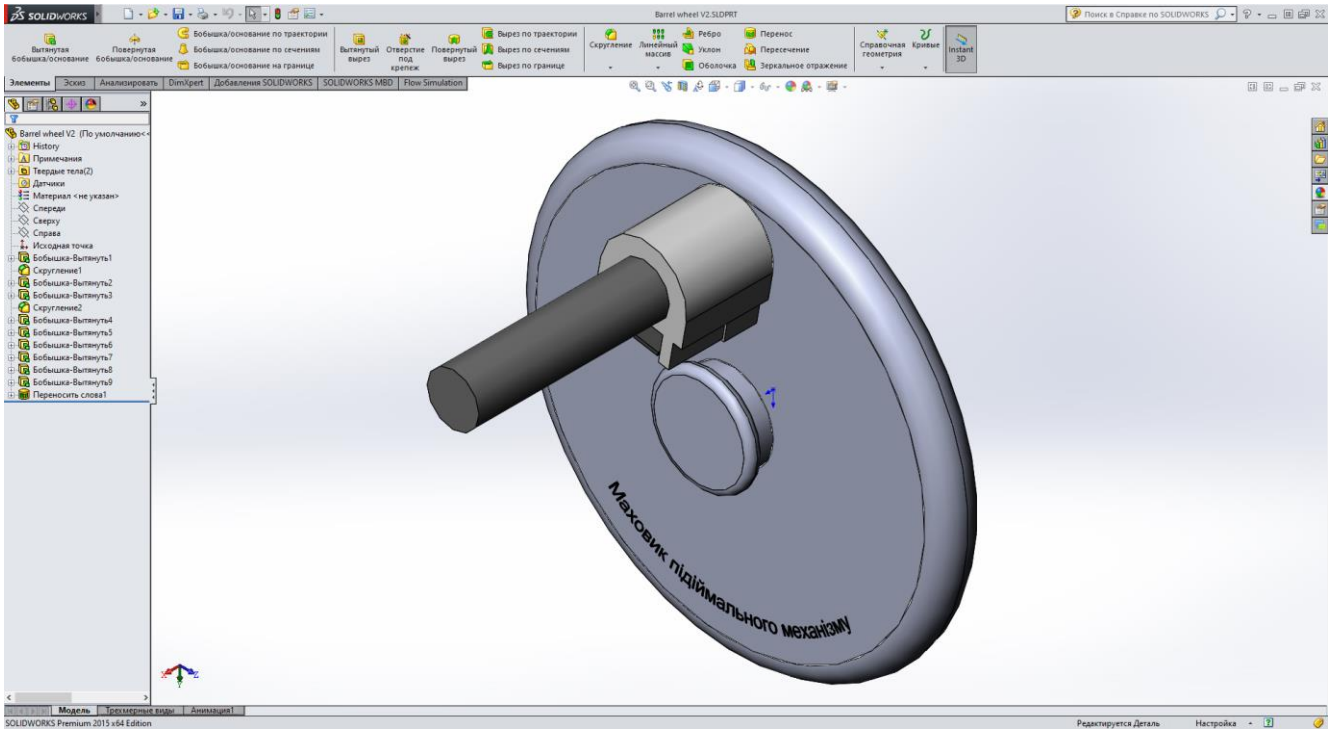

Рисунок 3.11 – Маховик підіймального механізму ствола

Налаштування матеріалів або кольорів частин моделі можна проводити в розділі налаштування зовнішнього вигляду (рис. 3.12).

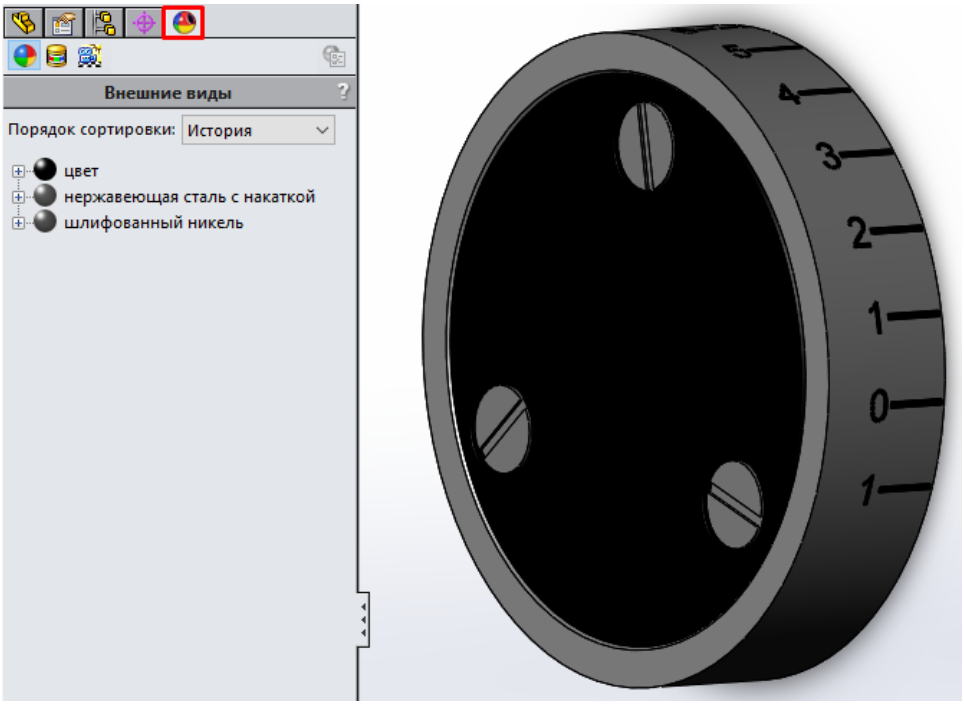

Рисунок 3.12 – Параметри зовнішнього вигляду

Для переведення моделі у формат FBX використаємо 3ds Max 2022. Після відкриття файлу обираємо параметри імпорту (рис. 3.13).

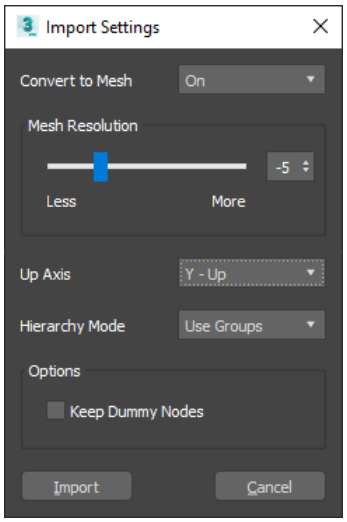

Рисунок 3.13 – Налаштування імпорту моделі в 3ds Max

Після імпорту деталі перевіряємо коректність відображення матеріалів (рис. 3.14).

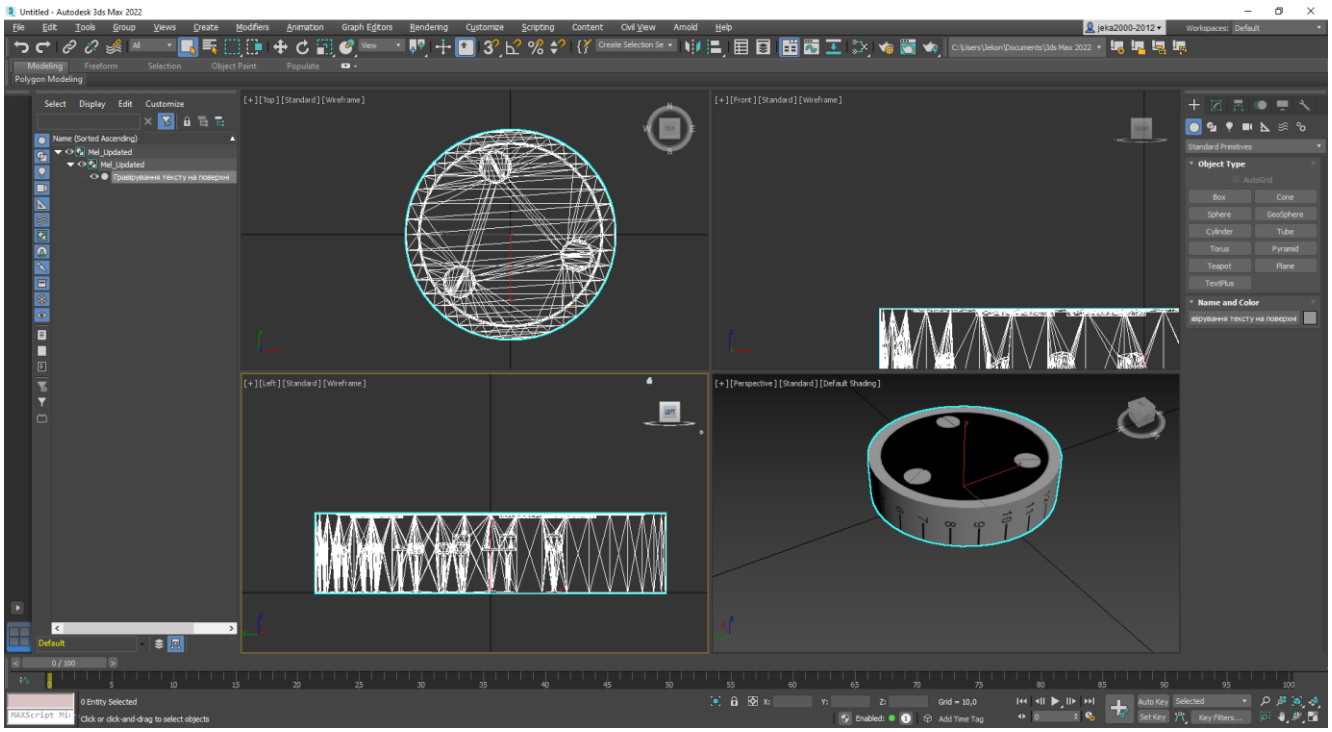

Рисунок 3.14 – Вигляд деталі в 3ds Max

Далі виконаємо експорт моделі (рис. 3.15).

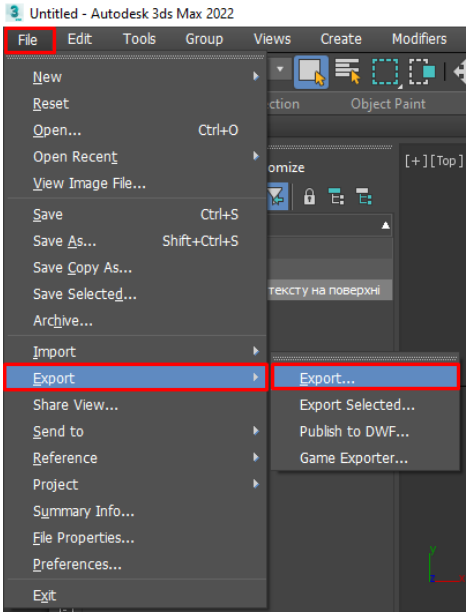

Рисунок 3.15 – Виконання експорту моделі

У вікні експорту обираємо місце збереження файлу, його назву, формат FBX та натискаємо кнопку збереження (рис. 3.16).

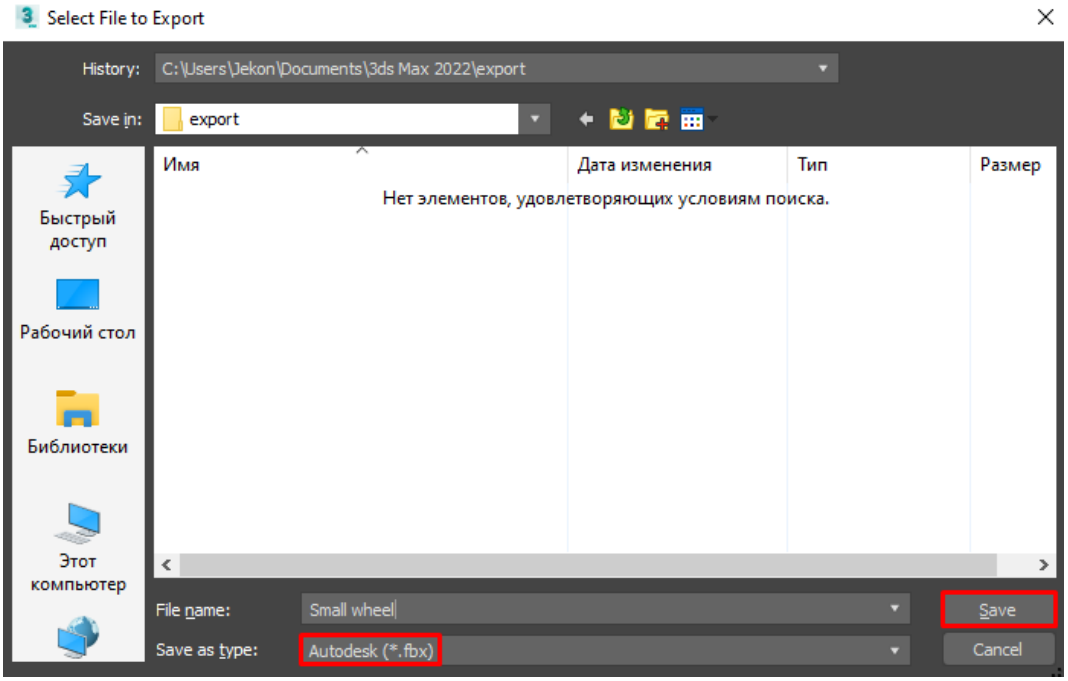

Рисунок 3.16 – Встановлення fbx формату деталі

Далі погоджуємо з параметрами експорту натисканням кнопки «OK» (рис. 3.17).

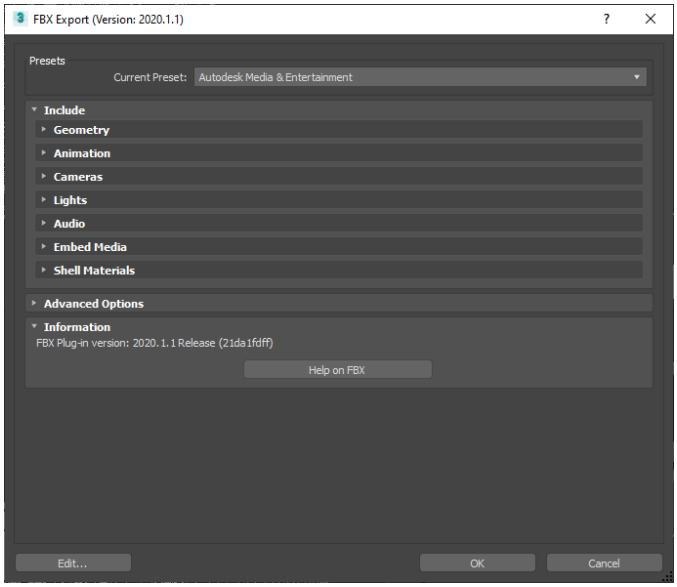

Рисунок 3.17 – Параметри експорту моделі

В результаті виконаних дій було отримано тривимірну модель, що збережена у форматі fbx (рис. 3.18).

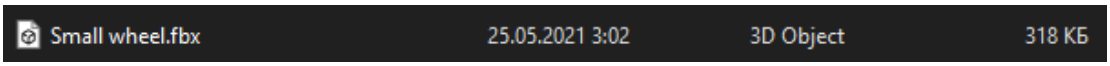

Рисунок 3.18 – Готовий файл моделі в форматі fbx

3.5 Програмна реалізація додатку

Для розроблюваного додатку було обрано розробити простий та функціональний дизайн. Навігація реалізована за допомогою кнопок меню з написом, що відповідає їх дії. Головне меню (рис. 3.19) містить відтінки жовтого, зеленого та блакитного кольорів. Для їх реалізації були використані властивості «Background» для загального фону та «Color» для елементів меню.

Рисунок 3.19 – Головне меню

Для сцени робочого простору додатку обрано темно-сірий колір фону адже він добре поєднується зі світло-сірим кольором деталей прицілу.

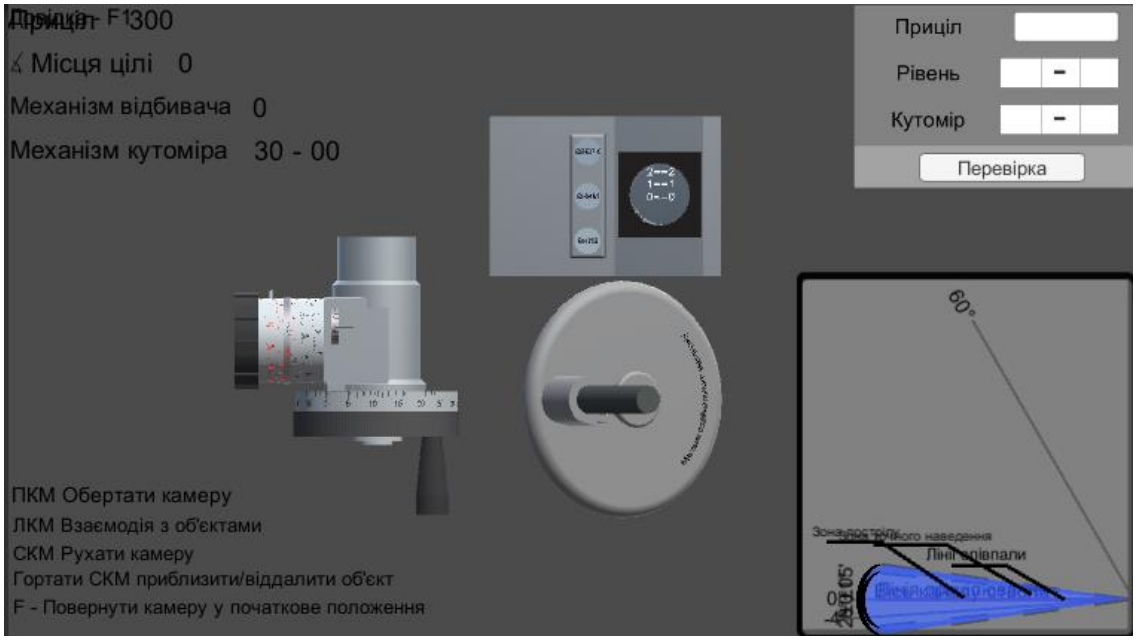

Рисунок 3.20 – Сцена робочого простору додатку

Елементи прицілу розміщені у відповідності до їх реального положення, маховик підіймального механізму ствола розміщено під панеллю узгодження для покращення ергономічності додатку (рис. 3.21).

При наведенні курсора миші на рухомі деталі, вони підсвічуються більш світлим кольором.

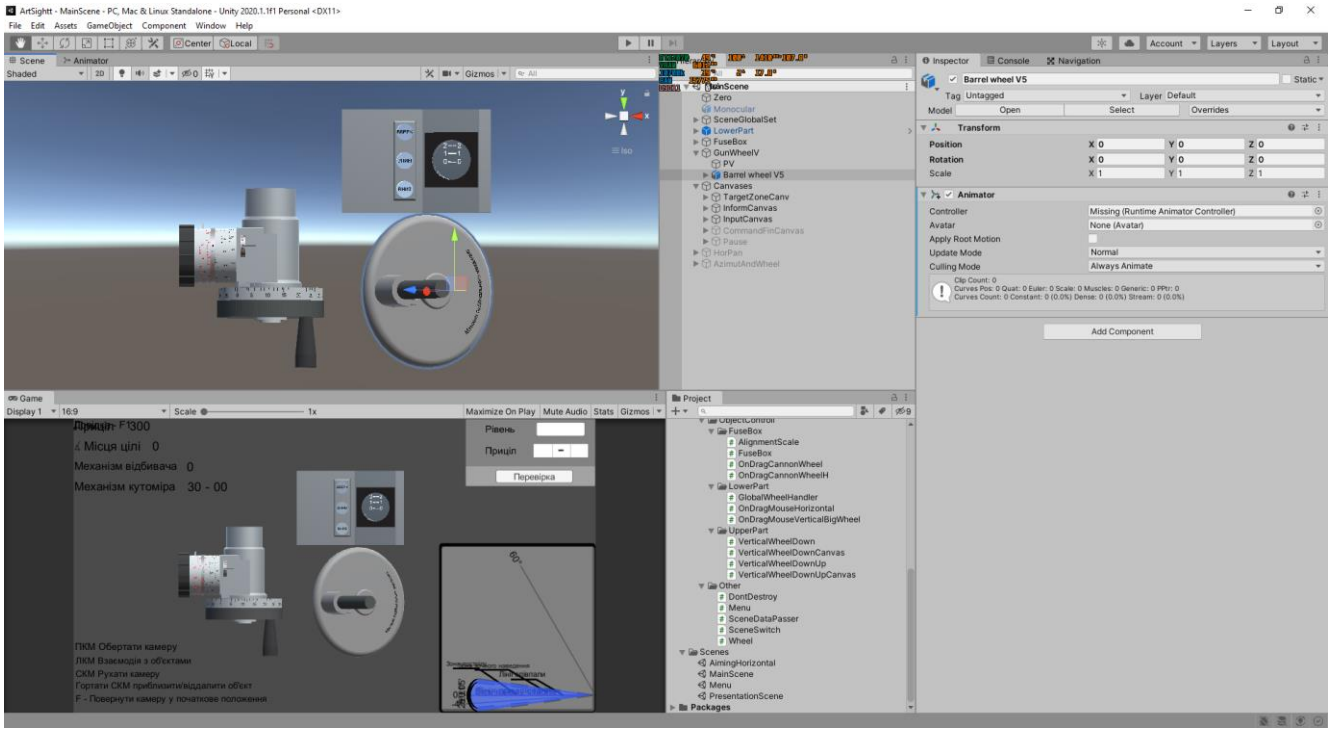

Рисунок 3.21 – Розміщення тривимірних моделей в середовищі Unity

В таблиці 3.1 наведено опис основних компонентів додатку.

Лістинг програмного коду наведено в додатку В.

Таблиця 3.1 – Опис компонентів додатку

Компонент	Призначення
CameraControll	Виконує контроль положення камери за допомогою комп’ютерної миші.
InfPanel	Панель, що використовується для виведення графіку положення направлення прицілу та вісі каналу ствола
InputFields	Панель, що використовується для введення завдання користувачем
AlignmentScale	Виконує обчислення положення шкал панелі узгодження відносно зв’язаних з ними маховиками
FuseBox	Описує логіку роботи індикаторів панелі узгодження
OnDragCannonWheel	Виконує реалізацію повороту маховика підіймального механізму ствола за допомогою миші та клавіатури
GlobalWheelHandler	Виконує отримання значення положення рухомих об’єктів
OnDragMouseHorizontal	Виконує обробку значення прицілу при обертанні маховика точної шкали
OnDragMouseVerticalBigWheel	Виконує обробку значення кутів місця цілі при обертанні відповідного маховика
DontDestroy	Реалізує збереження даних про положення об’єктів сцени
Menu	Реалізує роботу головного меню, а саме обробку натискання кнопок
Wheel	Базовий клас для усіх рухомих об’єктів

3.6 Використання розробленого додатку

Після завантаження додатку користувач потрапляє до головного меню.

Рисунок 3.22 – Головне меню

Натиснувши кнопку «Довідка» можна відкрити вікно довідки для ознайомлення з теоретичною інформацією.

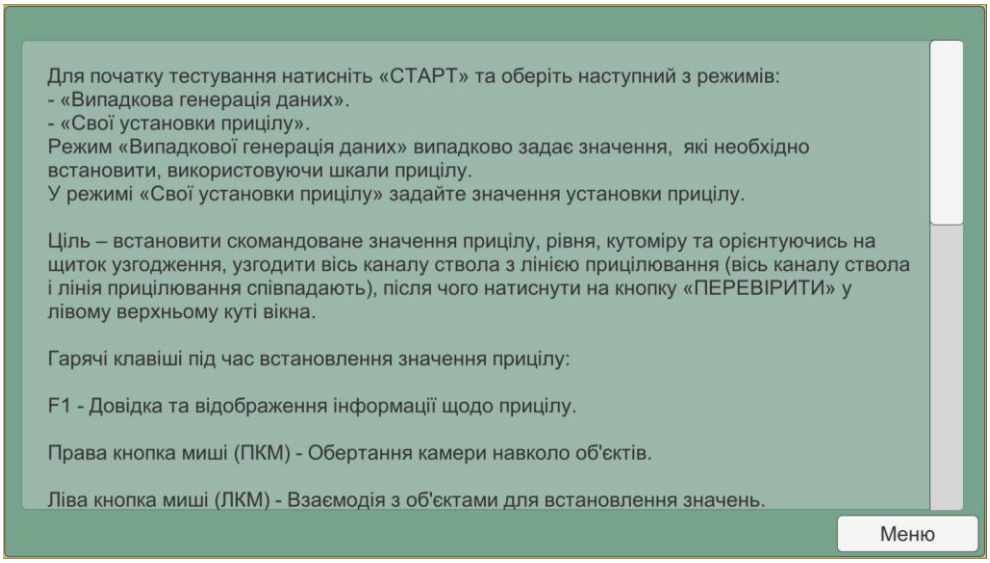

Рисунок 3.23 – Вікно довідки

Після натискання кнопки «Старт» з’являється можливість вибору способу формування завдання.

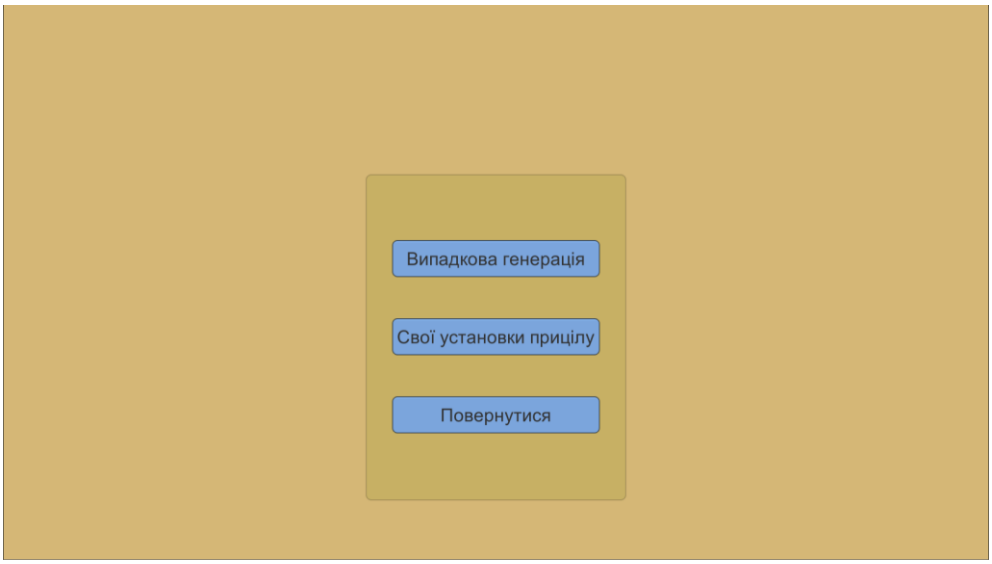

Рисунок 3.24 – Меню вибору способу формування завдання

Після вибору другого способу формування завдання з’являються поля введення значень.

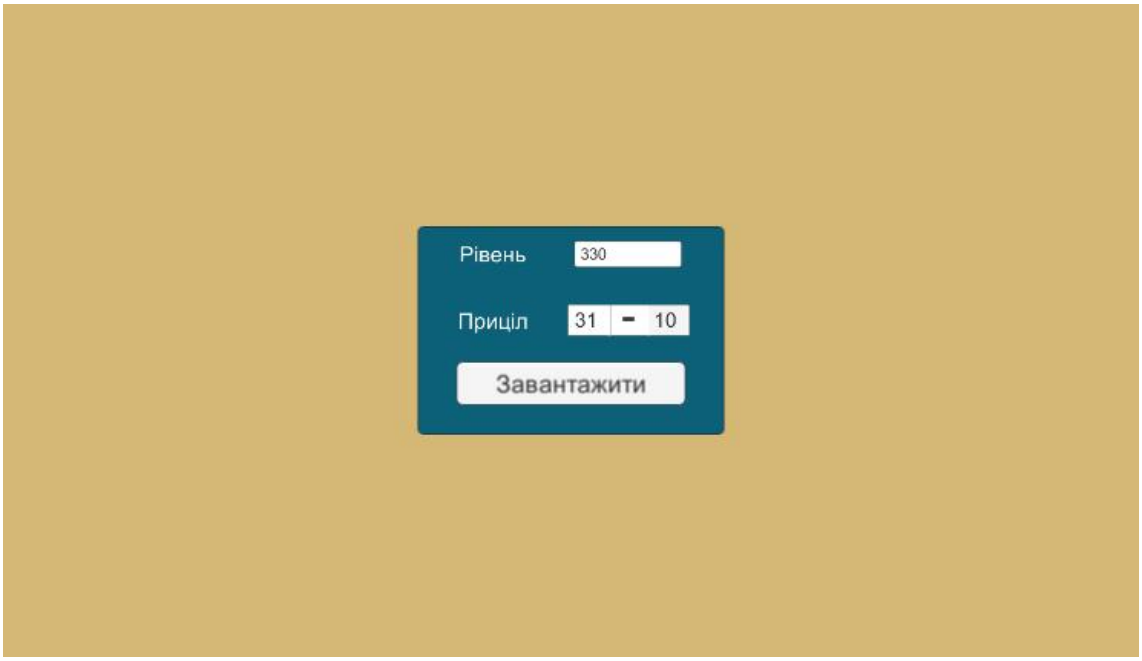

Рисунок 3.25 – Поля введення завдання

Після введення значень натискання кнопки «Завантажити» дозволяє розпочати роботу з додатком, кнопка «Випадкова генерація» також виконує завантаження робочої області з тією відмінністю, що завдання генерується випадково.

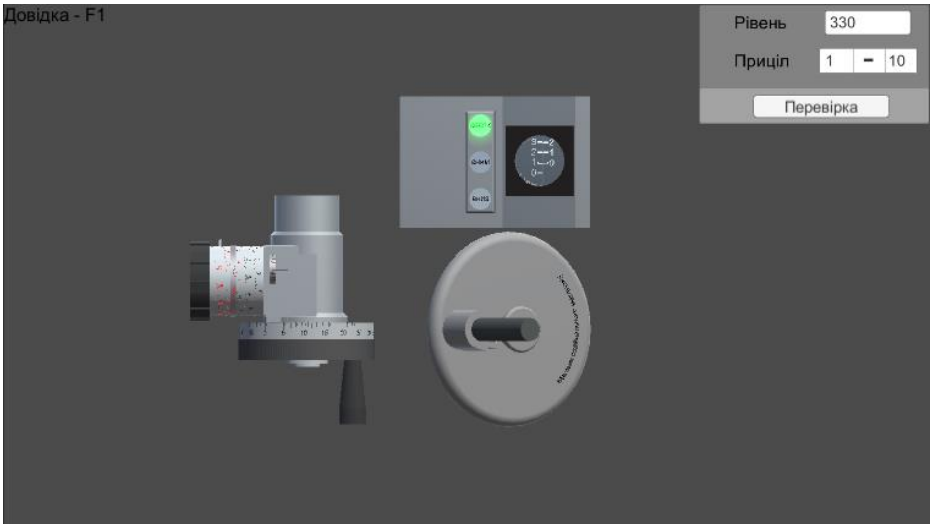

Рисунок 3.26 – Вид робочої області додатку

Натискання клавіші «F1» викликає додаткову інформацію в лівій частині вікна додатку, натискання клавіші «Z» активує відображення графіку положення ліній прицілювання та каналу ствола в лівому нижньому куті.

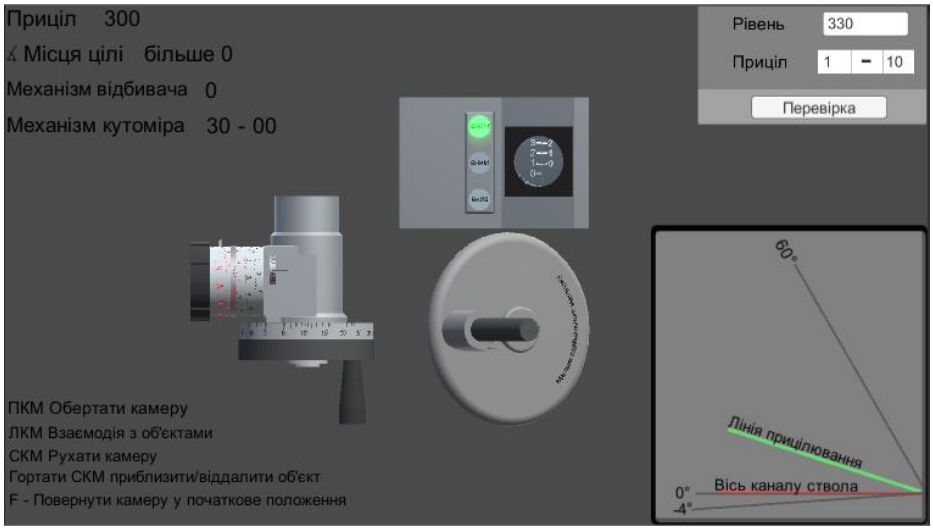

Рисунок 3.27 – Виклик меню довідки та графіку

Виконуємо встановлення положення прицілу за допомогою обертання маховика точної шкали прицілу.

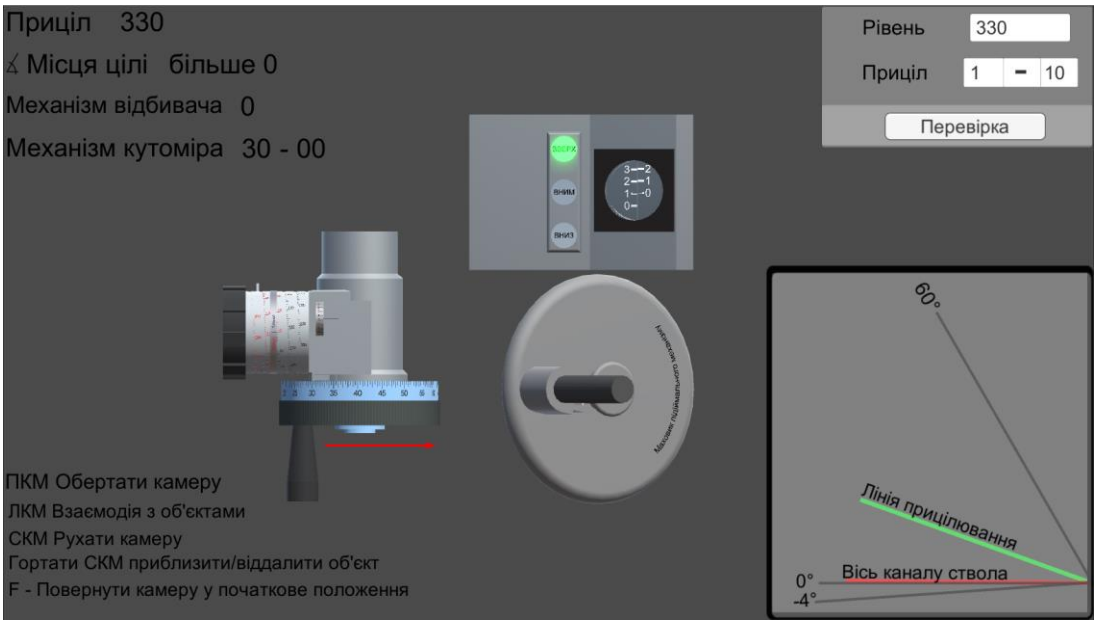

Рисунок 3.28 – Встановлення положення прицілу

За допомогою обертання маховика кутів місця цілі встановлюємо необхідне значення.

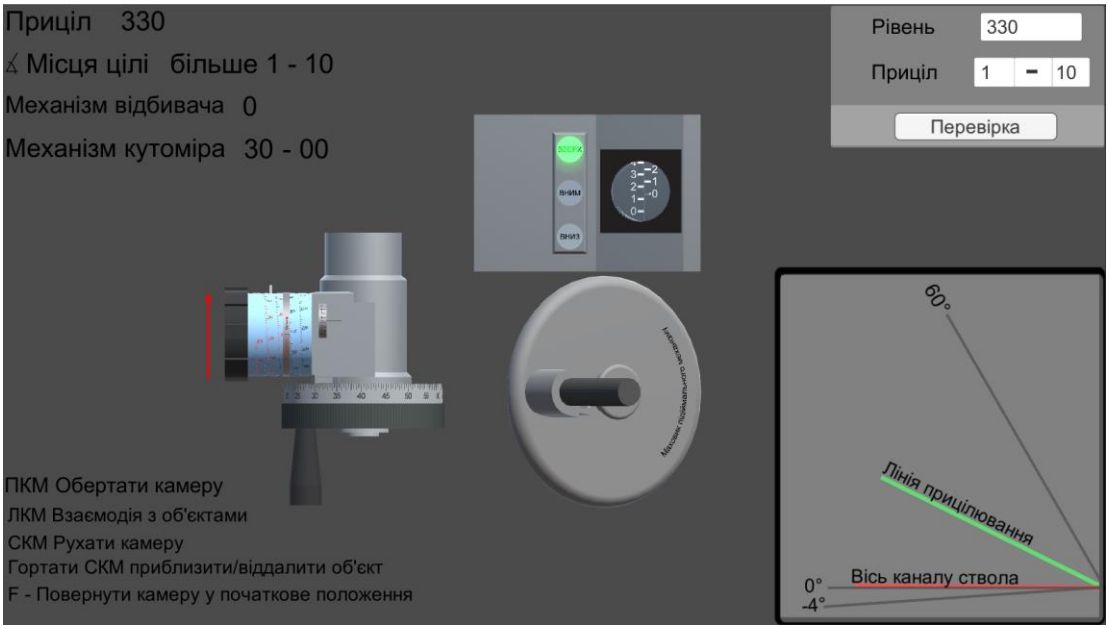

Рисунок 3.29 – Встановлення положення кутів місця цілі

Фінальним кроком є узгодження положення прицілу та каналу ствола за допомогою обертання маховика підіймального механізму до моменту включення усіх індикаторів на панелі узгодження.

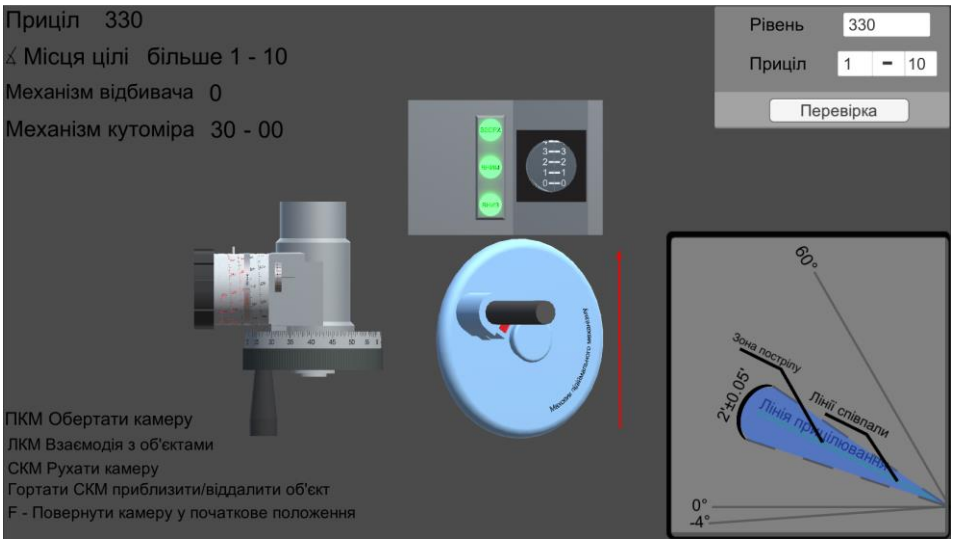

Рисунок 3.30 – Встановлення положення каналу ствола

Для завершення встановлення значень прицілу та отримання оцінки необхідно натиснути кнопку «Перевірка».

Рисунок 3.31 – Отримання результатів

Розроблений тренажер був використаний під час занять у військовій кафедрі Сумського державного університету (рис. 3.32). Додаток отримав позитивні відгуки як від курсантів так і від викладачів.

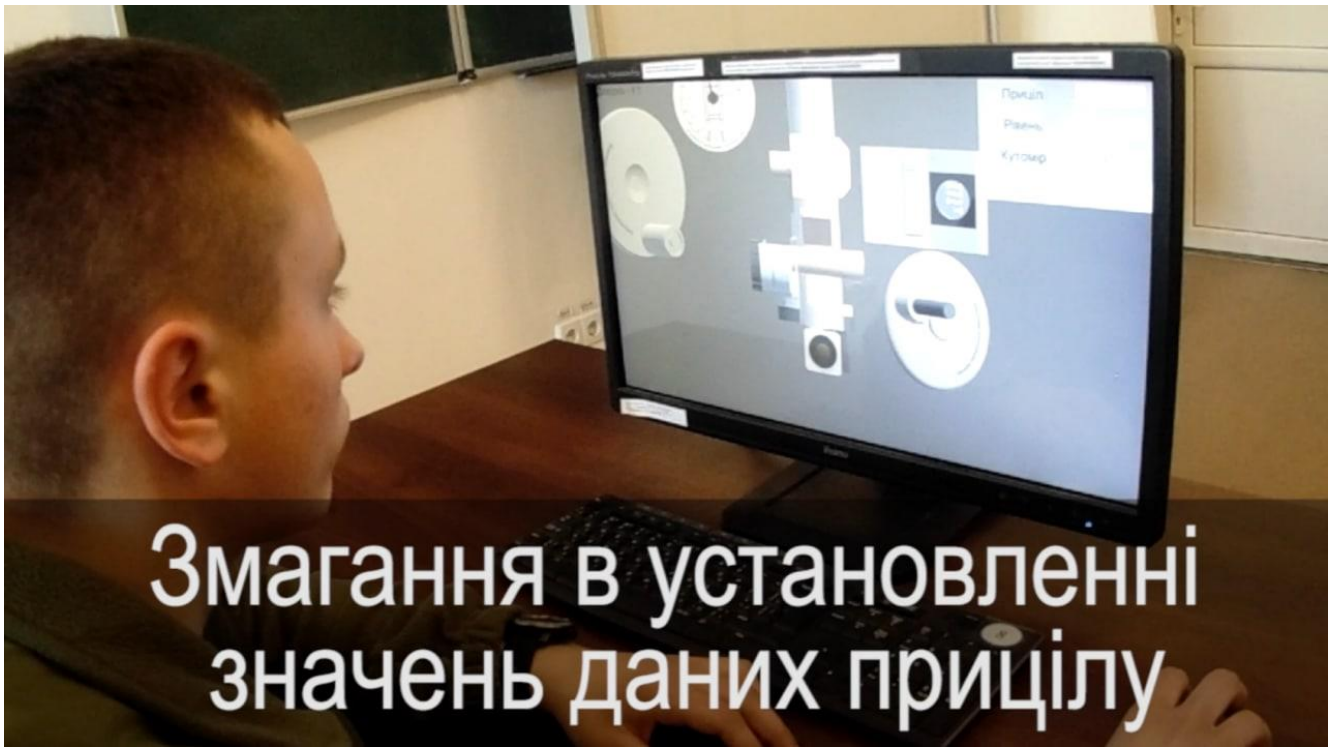

Рисунок 3.32 – Отримання результатів

ВИСНОВКИ

У ході виконання переддипломної практики була проведена робота над оглядом останніх досліджень і публікацій та їх вибором для отримання теоретичної інформації, аналізом програмних продуктів-аналогів, їх описом. Була виконана постановка задачі.

Далі було виконано проектування додатку. Завдяки моделюванню вдалося визначити об’єкти, функції та процеси, які відбуваються у програмному додатку. В даному розділі були виконані наступні етапи:

- моделювання використання додатка;
- моделювання в IDEF0;
- модель проектування;
- модель реалізації.

У ході виконання розділу «Управління проектами» було розроблено такі етапи:

- ініціалізація;
- планування;
- розробка;
- завершення.

На етапі ініціалізації була проведена розробка концепції проекту, техніко-економічне дослідження, підготовка оціночного висновку.

На етапі планування були пройдені такі етапи:

- планування змісту структури IT-проекту;
- планування структури організації;
- побудова матриці відповідальності;
- побудова PDM-мережі;
- побудова календарного графіку;

– управління ризиками проекту;

Також під час виконання робіт були використані наступні програмні продукти та веб-сервіси:

– Microsoft Word;
– Microsoft Excel;
– Microsoft Project;
– Draw.io;
– Google-диск.

Для реалізації додатку були виконані наступні етапи:

– визначено факт актуальності розробки віртуальних тренажерів для навчання курсантів на основі аналізу представлених аналогів та факту появи все більшої кількості додатків даного типу;
– проведено моделювання розроблюваного додатку для визначення об’єму робіт і систематизації підходу по проектування;
– розроблено тривимірні моделі деталей прицілу ПГ-4 у SolidWorks на основі його реальної моделі;
– проведено опис математичних залежностей взаємодії рухомих об’єктів;
– розміщено тривимірні моделі у середовищі Unity;
– розроблено інтерфейс додатку та скрипти для реалізації функціоналу додатку мовою програмування C#;
– проведено тестування роботи додатку.

У результаті виконаної роботи було розроблено віртуальний тренажер встановлення кутів прицілювання прицілу ПГ-4, а саме вертикальної наводки.

Також було отримано авторське право на частину тренажера представлену в даній роботі і на загальний комплекс тренажера (скан-копії – в додатку Г). Основні результати роботи пройшли апробацію на конференції IMA:2021 та Шостій Всеукраїнській курсантсько-студентській конференції (скан-копії тез в додатку Г).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Unity User Manual 2020.3 (LTS) [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://docs.unity3d.com/Manual/index.html.
2. Материалы общей тематики по языку C# и фреймворку .NET [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: https://metanit.com/sharp/general.php.
3. Документация по C# [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://docs.microsoft.com/ru-ru/dotnet/csharp/.
4. ГЛАВА 4 Диаграмма вариантов использования (use case diagram) [Електронний ресурс] – Режим доступу до ресурсу: http://khpi-iip.mipk.kharkiv.edu/library/case/leon/gl4/gl4.html#1 (дата звернення: 10.12.2019).
5. Методология IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: https://studme.org/87184/ekonomika/metodologiya_idef0 (дата звернення: 10.12.2019).
6. Анисимов В. В. ДИАГРАММЫ КЛАССОВ АНАЛИЗА [Електронний ресурс] / Владимир Викторович Анисимов – Режим доступу до ресурсу: https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema13/tema13_2 (дата звернення: 21.12.2020).
7. Рекомендация: Диаграмма последовательности [Електронний ресурс] – Режим доступу до ресурсу: http://dit.isuct.ru/Publish_RUP/core.base_rup/guidances/guidelines/sequence_diagram_AFA76EBB.html (дата звернення: 18.12.2020).
8. ОМПС. Діаграми діяльності [Електронний ресурс] – Режим доступу до ресурсу: https://studopedia.su/16_5481_omps-diagrami-diyalnosti.html (дата звернення: 19.12.2020).

9. Уроки SOLIDWORKS [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://autocad-lessons.ru/solidworks/.
10. INTRODUCING SOLIDWORKS [Електронний ресурс]. – 2014. – Режим доступу до ресурсу: https://files.solidworks.com/pdf/introsw.pdf.
11. 3ds Max Learning Center [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://help.autodesk.com/view/3DSMAX/2022/ENU/.
12. Unity's architecture [Електронний ресурс] – Режим доступу до ресурсу: https://subscription.packtpub.com/book/game_development/9781789349337/1/ch01lvl1sec11/unity-s-architecture.
13. Visual Studio documentation [Електронний ресурс]. – 2016. – Режим доступу до ресурсу: https://docs.microsoft.com/en-us/visualstudio/windows/?view=vs-2017&preserve-view=true.
14. Unity Hub [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://docs.unity3d.com/Manual/GettingStartedUnityHub.html.
15. Scripting [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://docs.unity3d.com/Manual/ScriptingSection.html.

Додаток А.

Технічне завдання

ТЕХНІЧНЕ ЗАВДАННЯ

на розробку інформаційної системи
«Віртуальний тренажер "Установка кутів
прицілювання прицілу ПГ-4. Вертикальна наводка"»

ПОГОДЖЕНО:

Старший викладач кафедри комп’ютерних наук

_____ Кузнєцов Е.Г.

Студент групи ІТ-71

_____ Чичикало Є.А.

Суми 2021

1. Призначення й мета створення додатку

1.1 Призначення додатку

Додаток має надавати довідкову інформацію користувачам про предметну область, мати можливість роботи користувача в режимі симулятора виставлення кутів місця цілі з автоматичним та ручним режимом встановлення завдання.

1.2 Мета створення додатку

Основною метою розробки програмного продукту є створення віртуального тренажеру навідників артилерійських установок на базі прицілу ПГ-4. Віртуальний тренажер дозволить проводити практичні заняття для великої кількості курсантів, що було раніше неможливо через наявність однієї одиниці необхідної техніки або взагалі її відсутність.

1.3 Цільова аудиторія

До цільової аудиторії додатку можна віднести курсантів, що навчаються у військових навчальних закладах та військових кафедрах університетів України.

2 Вимоги до додатку

2.1 Вимоги до додатку в цілому

2.1.1 Вимоги до структури й функціонування додатку

Готовий програмний продукт повинен бути сумісний з операційними системами на базі Windows. Для його роботи потрібна установка компонентів необхідних для роботи додатків на базі Unity.

При імпорті тривимірних моделей у середовище розробки Unity використовувати найменше допустиме розширення моделей (кількість полігонів). Даний процес необхідний для забезпечення можливості запуску готового

програмного продукту на комп’ютерах з мінімальними вимогами обчислювальної потужності. А саме:

- чотирьох потоковий центральний процесор;
- 4-8 ГБ оперативної пам’яті;
- 1-2 ГБ відеопам’яті на відеокарті для можливості коректного відображення текстур тривимірних моделей;
- 5 ГБ вільного місця на ПЗУ (постійний запам’ятовувальний пристрій) для виконання установки програмного забезпечення;

Реалізувати точну передачу кольору розроблених моделей на основі наявних фото реального прицілу ПГ-4. Текст на деталях фарбувати в чорний колір, в окремих випадках використовувати червоний колір.

2.1.2 Вимоги до персоналу

Від персоналу вимагаються базові навички роботи з системою Windows.

2.1.3 Вимоги до збереження інформації

У віконному додатку не використовуються додаткові засоби збереження, оскільки в цьому немає необхідності. Потік даних відбуваються тільки під час роботи додатку, внаслідок цього важлива інформація зберігається в оперативній пам’яті персонального комп’ютера.

2.1.4 Вимоги до розмежування доступу

Розроблюваний додаток не має обмеження доступу. Проте він повинен розповсюджуватися в якості програмного забезпечення для навчальних закладів військового напрямку.

2.2 Структура додатку

2.2.1 Загальна інформація про структуру додатку

Структура додатку містить наступні елементи:

- головне меню;
- коротка інформаційна довідка;
- вибір випадкового формування завдання;
- вибір ручного формування завдання;
- вікно для відображення завдання;
- вікно для відображення поточних значень прицілу;
- взаємодія з рухомими частинами прицілу за допомогою миші.

2.2.2 Навігація

За бажанням замовника необхідно реалізувати вікно з головним меню. Воно дозволить за допомогою відповідних кнопок завантажувати певний функціонал додатку.

2.2.3 Наповнення додатку (контент)

Для наповнення контентом додатку буде використана платформа Unity.

Всю інформацію для наповнення додатку буде взято з довідників.

2.2.4 Дизайн та структура додатку

Стиль додатку має бути простим, легким в опануванні, шрифти повинні бути зручними для читання, кольорову гаму обрати нейтральну.

Розташування елементів на головному вікні додатку схематично показано на рисунку 2.1.

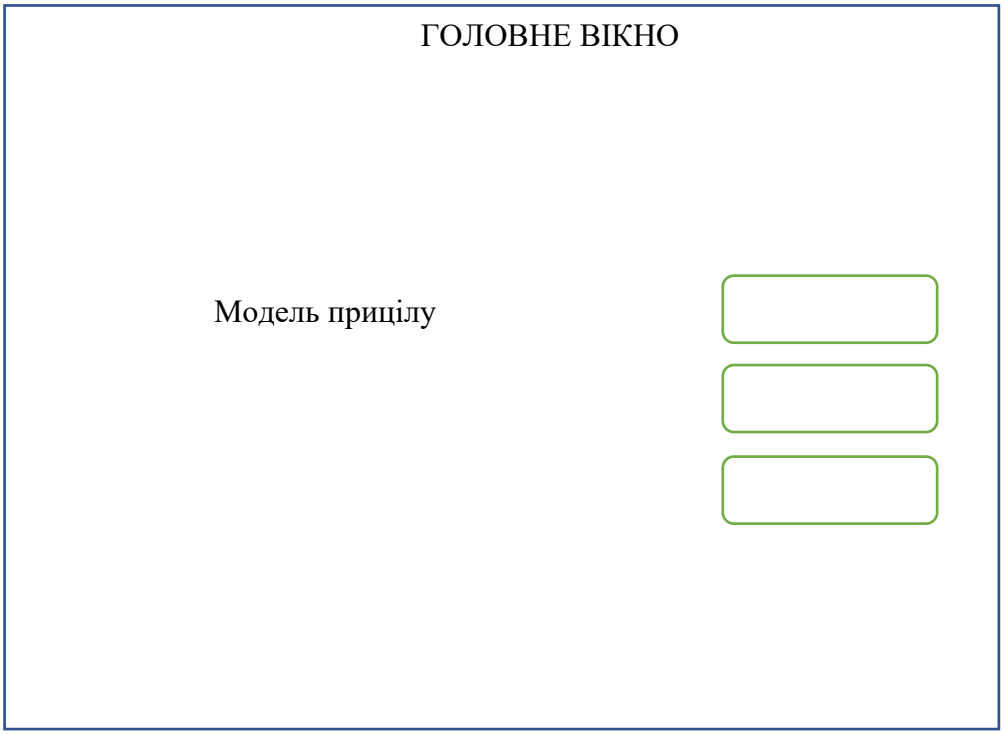

Рисунок 2.1 – Схема головного вікна

2.2.5 Система навігації (карта додатку)

Карта додатку зображена на рисунку 2.2.

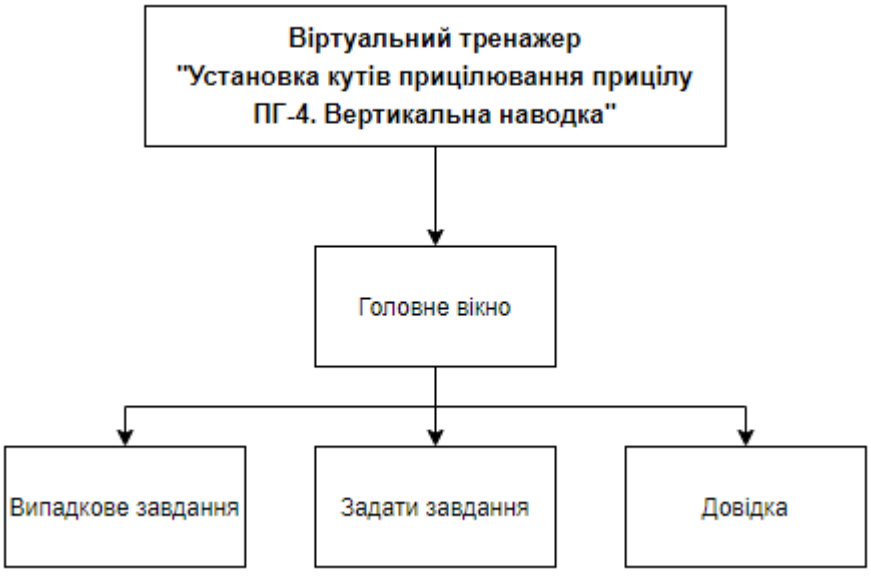

Рисунок 2.2 – Карта додатку

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці 2.1.

Таблиця 2.1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Головне меню із зображення анімованої моделі прицілу	Клієнт
UN-02	Інформаційна довідка	Клієнт
UN-03	Вибір випадкового формування завдання	Клієнт
UN-04	Вибір ручного формування завдання	Клієнт
UN-05	Вікно для відображення поставленого завдання	Клієнт
UN-06	Вікно для відображення поточних значень прицілу	Клієнт
UN-07	Взаємодія з рухомими елементами прицілу за допомогою комп’ютерної миші	Клієнт
UN-08	Взаємодія з рухомими елементами прицілу за допомогою клавіатури	Клієнт

2.3.2 Функціональні вимоги

На основі потреб користувача були визначені такі функціональні вимоги:

-автоматичне формування завдання;

-ручне формування завдання;

-можливість перегляду довідкового матеріалу;

-можливість імітації керування положенням рухомих деталей за допомогою комп’ютерної миші;

-можливість імітації керування положенням рухомих деталей за допомогою клавіатури;

-відображення на робочому просторі поточних показників прицілу;

-відображення на робочому просторі поставленого завдання.

2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці 2.2.

Таблиця 2.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Наявність вікна довідки	M	Надає можливість користувачу ознайомитися з довідковою інформацією
SR-02	Наявність анімованої моделі прицілу на головній сторінці	M	Реалізує обертання частини прицілу в головному меню
SR-03	Наявність функції запуску робочого простору з автоматично заданим завданням	S	Надає можливість запуску робочого простору з автоматично заданим завданням
SR-04	Наявність функції запуску робочого простору з ручним введенням завдання	M	Надає можливість запуску робочого простору з ручним введенням завдання

Умовні позначення в таблиці 2.2:

- Must have (M) – вимоги, які повинні бути реалізовані в системі;
- Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;
- Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація додатку відбувається з використанням:

- SolidWorks
- Unity
- Microsoft Visual Studio 2017

2.4.2 Вимоги до лінгвістичного забезпечення

Інтерфейс додатку має бути виконаний українською мовою.

2.4.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Операційна система Windows;
- Unity.

3 Склад і зміст робіт зі створення додатку

Докладний опис етапів роботи зі створення додатку наведено на рисунку А.3.

	Название задачи	Длительн	Начало	Окончание
1	**Дипломне проєктування**	**118 дней**	**18.01.21 9:00**	**30.06.21 18:00**
2	**1 Ініціалізація**	**6 дней**	**18.01.21 9:00**	**25.01.21 18:00**
3	1.1 Розробка концепції проєкту	2 дней	18.01.21 9:00	19.01.21 18:00
4	1.2 Техніко-економічне долідження	2 дней	20.01.21 9:00	21.01.21 18:00
5	1.3 Підготовка оціночного висновку	2 дней	22.01.21 9:00	25.01.21 18:00
6	**2 Планування**	**10 дней**	**26.01.21 9:00**	**08.02.21 18:00**
7	2.1 Планування змісту структури робіт IT-проєкту	1 день	26.01.21 9:00	26.01.21 18:00
8	2.2 Планування структури організації	1 день	27.01.21 9:00	27.01.21 18:00
9	2.3 Побудова матриці відповідальності	2 дней	28.01.21 9:00	29.01.21 18:00
10	2.4 Побудова PDM мережі	2 дней	01.02.21 9:00	02.02.21 18:00
11	2.5 Побудова календарного графіку	1 день	03.02.21 9:00	03.02.21 18:00
12	2.6 Управління ризиками проєкту	2 дней	04.02.21 9:00	05.02.21 18:00
13	2.7 Формування бюджету проєкту	1 день	08.02.21 9:00	08.02.21 18:00
14	**3 Розробка**	**52 дней**	**09.02.21 9:00**	**21.04.21 18:00**
15	3.1 Розробка моделі додатку	4 дней	09.02.21 9:00	12.02.21 18:00
16	3.2 Розробка прототипу з мінімальним набором функцій	14 дней	15.02.21 9:00	04.03.21 18:00
17	3.3 Розробка фінальної версії додатку	30 дней	05.03.21 9:00	15.04.21 18:00
18	**3.4 Тестування**	**4 дней**	**16.04.21 9:00**	**21.04.21 18:00**
19	3.4.1 Альфа тестування	2 дней	16.04.21 9:00	19.04.21 18:00
20	3.4.2 Бета тестування	2 дней	20.04.21 9:00	21.04.21 18:00
21	**4 Завершення**	**72 дней**	**15.02.21 9:00**	**25.05.21 18:00**
22	4.1 Розробка документації	60 дней	15.02.21 9:00	07.05.21 18:00
23	4.2 Архівація проєкту	7 дней	10.05.21 9:00	18.05.21 18:00
24	4.3 Передача ПП замовнику	5 дней	19.05.21 9:00	25.05.21 18:00
25	**5 Завершення дипломного проєктування**	**26 дней**	**26.05.21 9:00**	**30.06.21 18:00**
26	5.1 Перевірка плагіату	7 дней	26.05.21 9:00	03.06.21 18:00
27	5.2 Рецензування	7 дней	04.06.21 9:00	14.06.21 18:00
28	5.3 Перевірка керівником	7 дней	15.06.21 9:00	23.06.21 18:00
29	5.4 Усунення недоліків	5 дней	24.06.21 9:00	30.06.21 18:00

Рисунок 3.1 – Етапи створення додатку

4 Вимоги до складу й змісту робіт із введення додатку в експлуатацію

Для того, щоб додатком могли користуватися готовий програмний продукт повинен бути встановлений на персональному комп’ютері з операційною системою на базі Windows. Для його роботи потрібна установка компонентів необхідних для роботи додатків на базі Unity. Тобто необхідно з офіційного сайту Unity встановити програмне забезпечення, яке дозволить завантажувати додатки, що були розроблені з використанням даної платформи.

Додаток Б

Планування робіт

Деталізація мети проекту методом SMART.

Продуктом дипломного проекту є додаток тренажер встановлення вертикальних кутів наводки прицілу ПГ-4.

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

Таблиця Б.1 − Деталізація мети методом SMART

Specific (конкретна)	Розробити додаток на базі ОС Windows для відточення практичних навичок навідників артилерійських установок на базі прицілу ПГ-4..
Measurable (вимірювання)	Ціль даного проекту – створити додаток, який буде надавати можливість підвищувати рівень кваліфікації навідників, знижувати витрати часу під час проведення занять та підвищити їх якість через контроль дій навідника.
Achievable (досяжна, узгоджена)	Ціль даного проекту можна вважати досяжною, оскільки розробник ознайомлений з роботою в таких середовищах як SolidWorks, Unity (використання мови програмування C#). Більшість питань узгоджена з замовником у вигляді ТЗ.

Relevant (реалістична)	Для реалізації програмного продукту є необхідні технічні та програмні засоби, в даному випадку комп'ютер з достатньою потужністю для роботи з SolidWorks, Visual Studio та Unity, доступ до мережі Інтернет. Розробник має досвід у роботі з вказаним програмним забезпеченням.
Time-framed (обмежена в часі)	Додаток розроблюється з обмеженням у часі на основі сформованого календарного плану та матриці відповідальності.

Планування змісту структури робіт IT-проекту

Планування змісту робіт було виконане у формі WBS-діаграми. Даний вид діаграм дозволяє відобразити ієрархічну декомпозицію робіт, що будуть виконані командою проекту для досягнення цілей проекту і отримання бажаних результатів. Дана діаграма допомагає структурувати та визначити весь зміст проекту.

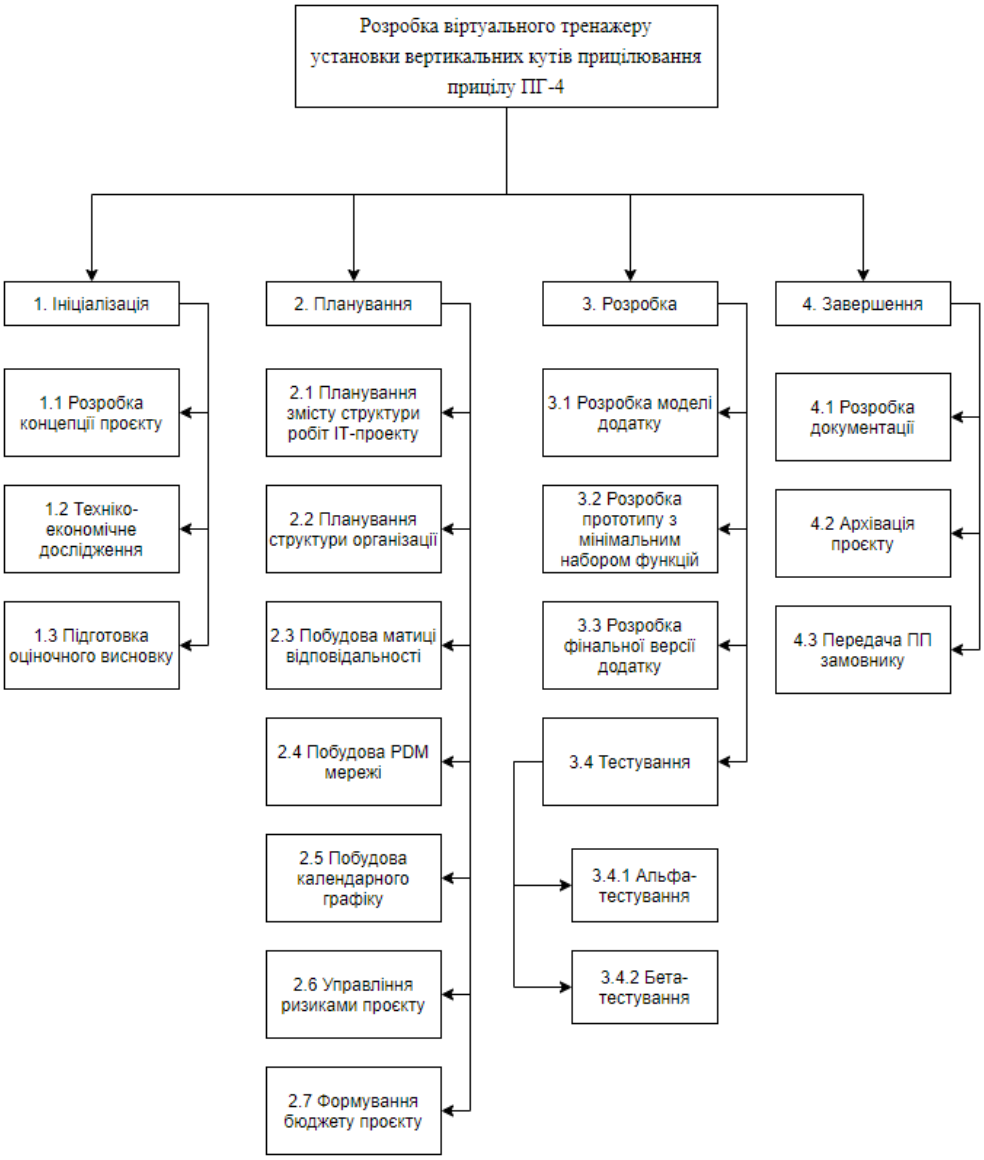

Рисунок Б.1 – WBS-структура проекту

Планування структури організації

Структура організації виконання дипломного проекту буде проста. Вона складається з двох учасників:

– студент (розробник-тестувальник);

– керівник дипломної роботи (менеджер проекту).

На основі даної структури була складена таблиця Б.2.

Таблиця Б.2 – Виконавці проекту

Роль	Ім’я	Проектна роль
Розробник -тестувальник	Чичикало Є.А.	Виконує розробку основного функціоналу проекту. Також відповідає за тестування функціоналу проекту.
Менеджер проекту (дипломної роботи)	Кузнєцов Е.Г.	Відповідає за виконання термінів, виконує збір та аналіз даних.

На основі WBS-структури проекту та таблиці виконавців проекту була розроблена OBS-діаграма. Вона виконує ієрархічне відображення організації проекту таким чином, що виконується суміщення пакету робіт з виконуючими особами.

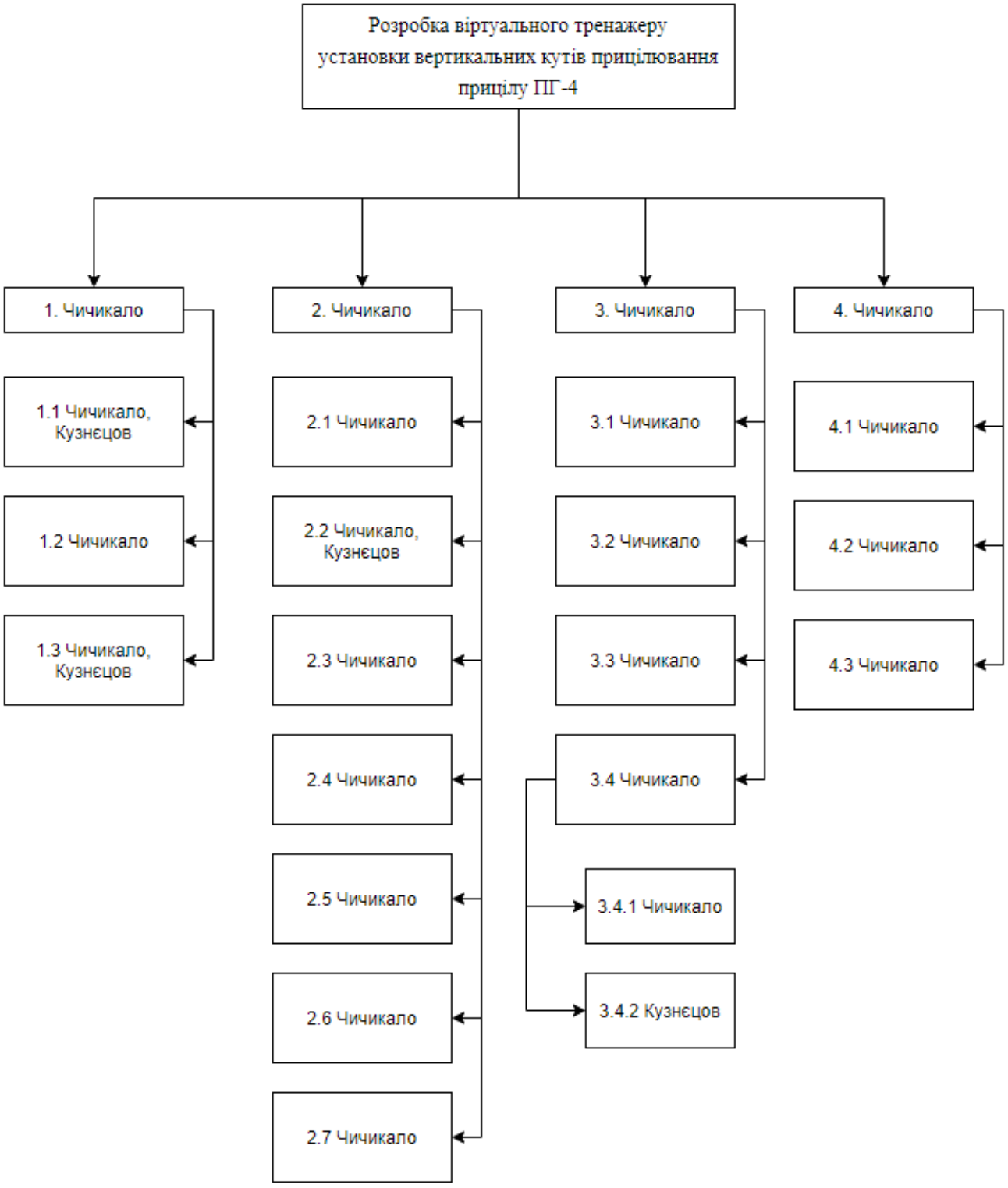

Рисунок Б.2 – OBS-структура проекту

Побудова матриці відповідальності

На основі WBS та OBS структур проекту була побудована матриця відповідальності. Вона дозволяє відобразити організаційну структуру проекту у відповідності до ієрархічної структури робіт проекту, що забезпечує призначення відповідальної особи чи осіб для відповідних робіт.

Члени команди проєкту / Задачі проєкту	Керівник проєкту	Інженер проєкту
Ініціалізація		О
Розробка концепції проєкту	У	О
Техніко-економічне дослідження		О
Підготовка оціночного висновку		О
Планування		О
Планування змісту структури робіт ІТ-проекту	У	О
Планування структури організації	У	О
Побудова матиці відповідальності	У	О
Побудова PDM мережі	У	О
Побудова календарного графіку	У	О
Управління ризиками проєкту	У	О
Формування бюджету проєкту	У	О
Розробка		О
Розробка моделі додатку		О
Розробка прототипу з мінімальним набором функцій		О
Розробка фінальної версії додатку		О
Тестування		О
Альфа-тестування		О
Бета-тестування	У	О
Завершення		О
Розробка документації		О
Архівація проєкту		О
Передача ПП замовнику		О
О - виконує основну роботу; У - приймає участь у виконанні		

Рисунок Б.3 – Матриця відповідальності

Розробка PDM мережі

PDM мережа – це метод побудови мережевих діаграм, які відображають заплановані задачі у вигляді вузлів (прямокутників). Заплановані задачі графічно зв’язані стрілками, які відображають послідовність виконання задач.

Для побудови даної мережі був використаний Microsoft Project, який на основі календарного графіку будує мережеву діаграму.

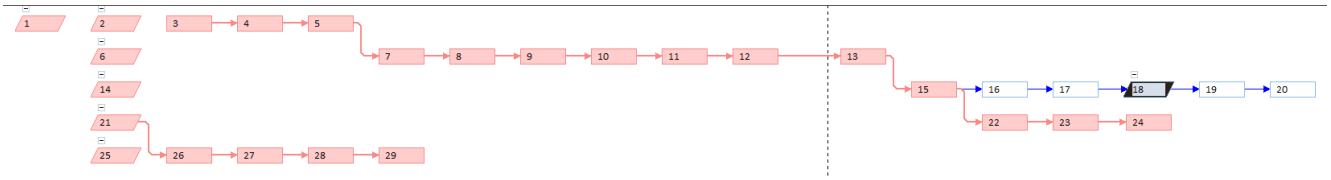

Рисунок Б.4 – PDM мережа

Побудова календарного графіку

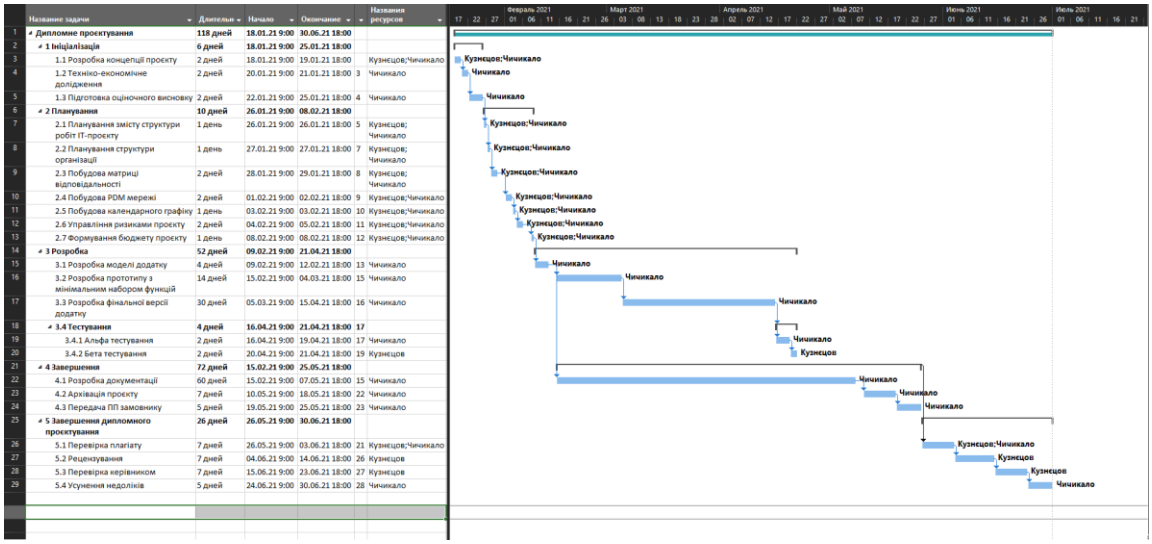

Рисунок Б.5 – Діаграма Ганта проекту

Управління ризиками проекту

При розробці програмного забезпечення, як і будь-якій іншій виробничій діяльності можливі певні ризики у ході реалізації проекту.

Під поняттям ризику мають на увазі умову або умови, які можуть спричинити втрату очікуваного прибутку, або події, які можуть поставити під загрозу успішну реалізацію запланованого проекту.

Ймовірність була обрана за даними таблиці Б.3.

Таблиця Б.3 – Оцінювання ймовірності появи ризикових подій

Ймовірність появи ризикових подій, бали		Стан виконання етапів проекту (x), %
Слабо-ймовірна	1	$0 < x \leq 10$
Малоймовірна	2	$10 < x \leq 30$
Ймовірна	3	$30 < x \leq 60$
Вельми ймовірна	4	$60 < x \leq 90$
Майже можлива	5	$90 < x \leq 100$

Величина можливих втрат була обрана за даними таблиці Б.4.

Таблиця Б.4 – Оцінювання величини можливих втрат

Величина можливих втрат, бали		Стан виконання етапів проекту (y), %
Мінімальна	1	$0 < y \leq 10$
Низька	2	$10 < y \leq 30$
Середня	3	$30 < y \leq 60$
Висока	4	$60 < y \leq 90$
Максимальна	5	$90 < y \leq 100$

Створимо класифікацію ризиків на основі назви ризику, ймовірності появи ризикової поді та оцінювання величини можливих втрат від наслідків виникнення ризикових водій.

Таблиця Б.5 – Класифікація ризиків

№	Назва ризику	Ймовірність	Величина втрат
1	Недостатня детальність ТЗ	2	3
2	Порушення дедлайнів для певного етапу розробки проекту	3	4
3	Нестабільна робота ПЗ	4	4
4	Нестабільна робота АЗ	2	3
5	Сезонні хвороби учасників проекту	3	2
6	Недостатньо якісне виконання тестування	3	1

Взявши за основу таблицю Б.5 була сформована матриця ризиків з використанням встановлених значень ймовірності та величини втрат.

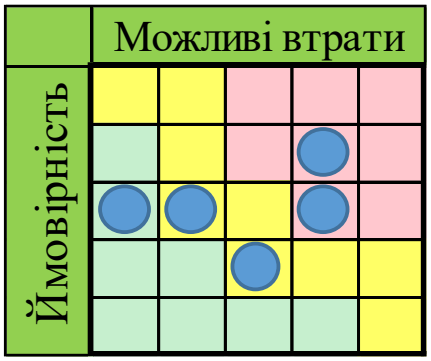

Рисунок Б.6 – Матриця ризиків

Після оцінки можливих ризиків був складений план реагування на ризики у вигляді таблиці Б.6.

Таблиця Б.6 – Реагування на ризики

Ризики проекту	Реакція на ризик
Недостатня детальність ТЗ	Обговорити з замовником або керівником проекту ті завдання, які поставлені не неоднозначно або потребують додаткових ресурсів.
Порушення дедлайнів для певного етапу розробки проекту	На етапі планування слід врахувати запас часу на обставини, що можуть викликати затримку у виконанні деяких етапів проекту.
Нестабільна робота ПЗ	Найшвидшим рішення даної проблеми можна вважати переустановку некоректно працюючого програмного забезпечення. У разі повторного виникнення помилок у роботі ПЗ варто звернутися у службу підтримки даного додатку.
Нестабільна робота АЗ	Виконати діагностику обладнання за найпростішим сценарієм. У випадку неможливості виявлення проблеми варто звернутися до кваліфікованого сервісного центру по ремонту комп’ютерного обладнання.
Сезонні хвороби учасників проекту	У бюджет проекту необхідно включити певну суму, уточнену з замовником, для медичного страхування виконавців проекту. Забезпечити запас часу на основних етапах проекту, який зможе компенсувати витрати часу на період хвороби учасників.
Недостатньо якісне виконання тестування	Варто повторити повне тестування функціоналу проекту. У разі повторного виникнення багів слід звернутися до кваліфікованого тестувальника з приводу тестування програмного продукту

Додаток В

Лістинг коду

AlignmentScale.cs

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

public class AlignmentScale : MonoBehaviour
{
    public Transform leftWheel;
    public Transform rightWheel;
    public OnDragCannonWheel cannonWheelAngle;
    public InfPanel phiAngle;

    float alignmentL;
    float alignmentR=0.0f;

    void Update()
    {
        alignmentL = phiAngle.GetAngleZone() / 2;
        leftWheel.eulerAngles= new Vector3(-alignmentL,0,0);

        alignmentR = cannonWheelAngle.GetValueAngle() / 2;
        rightWheel.eulerAngles = new Vector3(-alignmentR, 0, 0);
    }
}
```

FuseBox

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

public class FuseBox : MonoBehaviour
{
    public GameObject lamp1;
    public GameObject lamp2;
    public GameObject lamp3;

  public void AllLightOn()
```

```
    {
        lamp1.SetActive(true);
        lamp2.SetActive(true);
        lamp3.SetActive(true);
    }

    public void LightOnlyDown()
    {
        lamp1.SetActive(false);
        lamp2.SetActive(false);
        lamp3.SetActive(true);
    }

    public void LightDown()
    {
        lamp1.SetActive(false);
        lamp2.SetActive(true);
        lamp3.SetActive(true);
    }

    public void LightOnlyUp()
    {
        lamp1.SetActive(true);
        lamp2.SetActive(false);
        lamp3.SetActive(false);
    }

    public void LightUp()
    {
        lamp1.SetActive(true);
        lamp2.SetActive(true);
        lamp3.SetActive(false);
    }
}
```

OnDragCannonWheel

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

public class OnDragCannonWheel : Wheel
{
    public GlobalWheelHandler calNumC;

    public Transform pivotPoint;
    public int curRotation;
```

```
    public float stepDegCost;

    public float rotateStepM = 4f; //Mouse Sensetivity
    public float rotateStepB = 0.1f;//Keyboard Sensetivity
    public float rotateStepS=0.5f;//Scroll Sensetivity
    public float combScale = 80f;

    bool isRotateU = true;
    bool isRotateD = true;

    Color back;
    public Renderer rend;
    bool selected;
    bool blockMouseDrag;
    public CameraControll scroll;
    void Awake()//Remember original color
    {
        back = rend.materials[0].color;
        if (SceneSwitch.getCurrentScene() ==
SceneSwitch.Scene.MainScene)
            scroll = (CameraControll)GameObject.Find("Main
Camera").GetComponent("CameraControll");
    }

    void OnMouseEnter()//Change material Color if Enter
    {
        rend.materials[0].color = selectColor;
        selected = true;
    }

    void OnMouseExit()//Reset material color back if Exit
    {
        rend.materials[0].color = back;
        selected = false;
    }
    void OnMouseDrag()
    {
        if(!blockMouseDrag)
            MouseDrag(-rotateStepM * Input.GetAxis("Mouse Y"));

    }

    protected override void MouseDrag(float rotationSpeed)
    {

        if (isRotateU && isRotateD)// (MinVal, MaxVal)
        {
            transform.Rotate(Vector3.right, rotationSpeed);
        }
        else if (isRotateU && rotationSpeed > 0)// (MinVal++)
```

```
        {
            transform.Rotate(Vector3.right, rotationSpeed);
        }
        else if (isRotateD && rotationSpeed < 0)//  (MaxVal--)
        {
            transform.Rotate(Vector3.right, rotationSpeed);
        }
    }
    void Update()
    {

        if (selected && Input.GetKey(KeyCode.LeftAlt) &&
SceneSwitch.getCurrentScene() == SceneSwitch.Scene.MainScene)
        {
            MouseDrag(-rotateStepS * Input.mouseScrollDelta.y);
            blockMouseDrag = true;
            scroll.blockScroll = true;
        }
        else if (selected)
        {
            KeyboardControll(KeyCode.UpArrow, KeyCode.DownArrow);
            blockMouseDrag = false;
            scroll.blockScroll = false;
        }
    }
    protected override void KeyboardControll(KeyCode firstButton,
KeyCode secondButton)
    {
        float rotateStepBIncr = rotateStepB * combScale;

        if (Input.GetKey(firstButton) && isRotateD)
        {
            transform.Rotate(Vector3.left, rotateStepB);
        }
        else if (Input.GetKey(secondButton)&& isRotateU)
        {
            transform.Rotate(Vector3.right, rotateStepB);
        }

        if (Input.GetKey(firstButton) &&
Input.GetKey(KeyCode.LeftShift) && isRotateD)
        {
            transform.Rotate(Vector3.left, rotateStepBIncr);
        }
        else if (Input.GetKey(secondButton) &&
Input.GetKey(KeyCode.LeftShift) && isRotateU)
        {
            transform.Rotate(Vector3.right, rotateStepBIncr);
        }
    }
```

```
    protected override float GetAngle()
    {
        Vector3 forwardVector = pivotPoint.rotation * Vector3.up;
        float curDeg = Mathf.Atan2(forwardVector.y, forwardVector.z)
* Mathf.Rad2Deg;

        if (curDeg < 0) curDeg += 360f;

        return curDeg;
    }
    protected override float GetValue()
    {
        float num;
        num = ((curRotation - 1) * 0.5f) +
                    (Mathf.Round(((GetAngle() /
stepDegCost)*100)))/100;
        num -= 180;
        if (num < -60)
        {
            isRotateD = false;
            isRotateU = true;
            num = -60;
        }
        else if (num > 4)
        {

            num = 4;
            isRotateD = true;
            isRotateU = false;
        }
        else
        {
            isRotateD = true;
            isRotateU = true;
        }

        return num;
    }
    public float GetValueAngle()
    {
        return -GetValue();
    }
}
```

GlobalWheelHandler

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

```
using UnityEngine;

public class GlobalWheelHandler : MonoBehaviour
{
    public float getAngleXZHorWheel(Transform pivotPoint)//Hor
    {
        Vector3 forwardVector = pivotPoint.rotation *
Vector3.forward;
        float CurDeg = Mathf.Atan2(forwardVector.x, forwardVector.z)
* Mathf.Rad2Deg;

        if (CurDeg < 0) CurDeg += 360f;

        return CurDeg;
    }

    public float getAngleYZVertWheel(Transform pivotPoint)
    {
        Vector3 forwardVector = pivotPoint.rotation * Vector3.up;
        float CurDeg = Mathf.Atan2(forwardVector.y, forwardVector.z)
* Mathf.Rad2Deg;

        if (CurDeg < 0) CurDeg += 360f;

        return CurDeg;
    }

    public float getAngleYZVertWheelReverse(Transform pivotPoint)
    {
        Vector3 forwardVector = pivotPoint.rotation * Vector3.up;
        float CurDeg = Mathf.Atan2(forwardVector.y, forwardVector.z)
* Mathf.Rad2Deg;

        if (CurDeg < 0) CurDeg += 360f;

        return CurDeg;
    }

    public float getValue(int curRotation, int toZeroRot,
       Transform pivotPoint, float stepDegCost, bool
isVerticalWheel, int roundType, float rotationCost = 100)
    {
        float num=0;

        if (isVerticalWheel)
        {
            if (roundType == 0)
            {

            }
```

```
            else if (roundType == 1)
            {
                num = ((curRotation - toZeroRot) * rotationCost) +
                    (Mathf.Round((getAngleYZVertWheel(pivotPoint) /
stepDegCost)));
            }
            else
            {
                num = ((curRotation - toZeroRot) * rotationCost) +
                   (Mathf.Ceil((getAngleYZVertWheel(pivotPoint) /
stepDegCost)));
            }
        }
        else
        {
            if (roundType == 0)
            {
                num = ((curRotation - toZeroRot) * rotationCost) +
                    (Mathf.Floor((getAngleXZHorWheel(pivotPoint) /
stepDegCost)));
            }
            else if (roundType == 1)
            {
                num = ((curRotation - toZeroRot) * rotationCost) +
                    (Mathf.Round((getAngleXZHorWheel(pivotPoint) /
stepDegCost)));
            }
            else
            {
                num = ((curRotation - toZeroRot) * rotationCost) +
                   (Mathf.Ceil((getAngleXZHorWheel(pivotPoint) /
stepDegCost)));
            }
        }
        return num;
    }
}
```

OnDragMouseHorizontal

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

public class OnDragMouseHorizontal : MonoBehaviour
{
    //Rotation speed
```

```
    public float rotateStepM = 4f;
    public float rotateStepB = 36f;//1 step in deg

    //Output
    public Text informationT1;

    //Wheel weight
    public float stepDegCost;
    public float minValue;
    public float maxValue;
    public float maxVDeg;
    public float minVDeg;

    public Renderer rend;
    Color back;
    Vector3 screenPoint;
    Vector3 offset;
    public Transform pivotPoint;

    bool selected = false;
    public int RenderMatNum;

    float setDeg;
    public int curRotation = 0;
    public int minRotatations;
    public int maxRotatations;

    public Transform synWheelT;

    Color blue = new Color(153 / 255f, 204 / 255f, 255 / 255f,
0.4f);
    float syncDegRotatM;
    float syncDegRotatK;
    float synWheelRate = 23.2258064516f;
    public float combScale;
    float rotateStepBIncr;

    public GlobalWheelHandler calNum;

    float num = 300;
    public float Num { get => num; set => num = value; }

    void Start()
    {
        back = rend.materials[RenderMatNum].color;
    }

    void OnMouseEnter()//Change material Color if Enter
    {
        rend.materials[RenderMatNum].color = blue;//Blue
```

```
        selected = true;
    }

    void OnMouseExit()//Change material Color back if Exit
    {
        rend.materials[RenderMatNum].color = back;
        selected = false;
    }

    void Update()
    {

        if (selected)
        {
            KeyboardControll();
        }
    }

    void KeyboardControll()
    {
        syncDegRotatK = rotateStepB / 23.225806451f;
        rotateStepBIncr = rotateStepB * combScale;

        if (Input.GetKey(KeyCode.LeftArrow) && curRotation >=
minRotatations)
        {
            if (calNum.getAngleXZHorWheel(pivotPoint) - rotateStepB
< minVDeg)
            {
                curRotation--;
            }
            transform.Rotate(Vector3.up, -rotateStepB);
            synWheelT.Rotate(Vector3.up, -syncDegRotatK);//Rotate
small Wheel
            LogicOfValues();
        }
        else if (Input.GetKey(KeyCode.RightArrow) && curRotation <=
maxRotatations)
        {
            if (calNum.getAngleXZHorWheel(pivotPoint) + rotateStepB
> maxVDeg)
            {
                curRotation++;
            }
            transform.Rotate(Vector3.up, rotateStepB);
            synWheelT.Rotate(Vector3.up, syncDegRotatK);
            LogicOfValues();
        }
```

```
        if (Input.GetKey(KeyCode.LeftArrow) &&
Input.GetKey(KeyCode.LeftShift) && curRotation >= minRotatations)
        {
            if (calNum.getAngleXZHorWheel(pivotPoint) -
rotateStepBIncr < minVDeg)
            {
                curRotation--;
            }
            transform.Rotate(Vector3.up, -rotateStepBIncr);
            synWheelT.Rotate(Vector3.up, -rotateStepBIncr /
23.225806451f);
            LogicOfValues();

        }
        else if (Input.GetKey(KeyCode.RightArrow) &&
Input.GetKey(KeyCode.LeftShift) && curRotation <= maxRotatations)
        {
            if (calNum.getAngleXZHorWheel(pivotPoint) +
rotateStepBIncr > maxVDeg)
            {
                curRotation++;
            }
            transform.Rotate(Vector3.up, rotateStepBIncr);
            synWheelT.Rotate(Vector3.up, syncDegRotatK);
            LogicOfValues();
        }
    }

    void OnMouseDrag()
    {

        float wheelRotation = rotateStepM * Input.GetAxis("Mouse
X");//Get Mouse rotation angle
        float lim = calNum.getAngleXZHorWheel(pivotPoint) +
wheelRotation;//Calculate angle rotation

        MouseControll(wheelRotation, lim);

    }
    void MouseControll(float wheelRotation, float lim)
    {
        syncDegRotatM = wheelRotation / synWheelRate; // Accuracy
0.002
        if (curRotation >= minRotatations && curRotation <=
maxRotatations)
        {
            transform.Rotate(Vector3.up, wheelRotation);
            synWheelT.Rotate(Vector3.up, syncDegRotatM);
            LogicOfValues();
        }
```

```
        else if (curRotation > maxRotatations)
        {
            if (wheelRotation < 0)
            {
                transform.Rotate(Vector3.up, wheelRotation);
                synWheelT.Rotate(Vector3.up, syncDegRotatM);
                LogicOfValues();
            }
        }
        else if (curRotation < minRotatations)
        {
            if (wheelRotation > 0)
            {
                transform.Rotate(Vector3.up, wheelRotation);
                synWheelT.Rotate(Vector3.up, syncDegRotatM);
                LogicOfValues();
            }

        }

        //Update number of full rotation
        if (lim > maxVDeg)
        {
            curRotation++;
            LogicOfValues();

        }
        else if (lim < minVDeg)
        {
            curRotation--;
            LogicOfValues();
        }
    }

    void LogicOfValues()//Wheel number Output
    {
        num = calNum.getValue(curRotation, 1, pivotPoint,
stepDegCost, false, 1);

        informationT1.text = num.ToString();

        if (curRotation > maxRotatations)
        {
            num = maxValue;
            informationT1.text = num.ToString();
        }
        else if (curRotation < minRotatations)
        {
            num = minValue;
            informationT1.text = num.ToString();
```

```
        }
    }
}
```

OnDragMouseVerticalBigWheel

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

public class OnDragMouseVerticalBigWheel : MonoBehaviour
{

    //Rotation speed
    public float RotateStepM = 4f;
    public float RotateStepK = 36f;//1 step in deg
    public float combScale = 4;

    //Output
    public Text informationT1;

    //Wheel weight
    public float stepValCost;
    public float stepPosCost;
    public float minValue;
    public float maxValue;

    public float maxPos;
    public float minPos;

    public float perRotateStep = 1100f;
    private float ToZeroCord;

    Renderer rend;
    Color back;
    public Transform pivotPoint;//Becouse

    bool selected = false;
    public int RenderMatNum;

    public Transform Ring;

    float curPosM, curPosK;
    string numText;
    float num = 0;
    public float Num { get => num; set => num = value; }
```

```
    void Start()
    {
        ToZeroCord = -Ring.position.x;
    }
    void Awake()
    {
        rend = GetComponent<Renderer>();//Save start material
        back = rend.materials[RenderMatNum].color;
    }

    void OnMouseEnter()//Change material Color if Enter
    {
        rend.materials[RenderMatNum].color = new Color(153 / 255f,
204 / 255f, 255 / 255f, 0.4f);//Blue
        selected = true;
    }

    void OnMouseExit()//Change material Color back if Exit
    {
        rend.materials[RenderMatNum].color = back;
        selected = false;
    }

    void Update()
    {
        LogicOfValues();
        if (selected)
        {
            curPosK = Ring.transform.position.x + ToZeroCord +
RotateStepK / perRotateStep;
            if (Input.GetKey(KeyCode.DownArrow) && curPosK <=
maxPos)
            {
                transform.Rotate(Vector3.up, -RotateStepK);
                Ring.transform.Translate(Vector3.up * RotateStepK /
perRotateStep);

            }
            else if (Input.GetKey(KeyCode.UpArrow) && curPosK >=
minPos)
            {
                transform.Rotate(Vector3.up, RotateStepK);
                Ring.transform.Translate(-Vector3.up * RotateStepK /
perRotateStep);

            }

            if (Input.GetKey(KeyCode.DownArrow) &&
Input.GetKey(KeyCode.LeftShift) && curPosK <= maxPos)
            {
```

```
                transform.Rotate(Vector3.up, -RotateStepK *
combScale);
                Ring.transform.Translate((Vector3.up * RotateStepK *
combScale) / perRotateStep);

            }
            else if (Input.GetKey(KeyCode.UpArrow) &&
Input.GetKey(KeyCode.LeftShift) && curPosK >= minPos)
            {
                transform.Rotate(Vector3.up, combScale *
RotateStepK);
                Ring.transform.Translate((combScale * (-Vector3.up *
RotateStepK)) / perRotateStep);
            }
        }

    }
    void OnMouseDrag()
    {
        float h = RotateStepM * Input.GetAxis("Mouse Y");//Get Mouse
rotation angle

        curPosM = Ring.transform.position.x + ToZeroCord + h /
perRotateStep;

        if (curPosM >= minPos && curPosM <= maxPos)//Check limits
        {

            transform.Rotate(Vector3.up, -h);
            Ring.transform.Translate(Vector3.up * h /
perRotateStep);
        }
        else if (curPosM > maxPos)
        {
            if (h < 0)
            {
                transform.Rotate(Vector3.up, -h);
                Ring.transform.Translate(Vector3.up * h /
perRotateStep);
            }

        }
        else if (curPosM < minPos)
        {
            if (h > 0)
            {
                transform.Rotate(Vector3.up, -h);
                Ring.transform.Translate(Vector3.up * h /
perRotateStep);
            }
```

```
        }


    }

    void LogicOfValues()//Wheel number Output
    {
        num = (Mathf.Floor(((Ring.position.x + ToZeroCord) /
stepPosCost) * stepValCost));

        // limit Values
        if (num >= maxValue)
        {
            num = maxValue;
        }
        else if (num <= minValue)
        {
            num = minValue;
        }
        string temp = num.ToString();
        if (num < 0)
        {
            temp = temp.Replace('-', ' ');
            if (num <= -100)
            {
                temp = temp.Insert(2, " - ");
            }
            temp = temp.Insert(0, "менше ");
            informationT1.text = temp;
        }
        else
        {
            if (num >= 100)
            {
                temp = temp.Insert(1, " - ");
            }
            temp = temp.Insert(0, "більше ");
            informationT1.text = temp;
        }

    }
}
```

Додаток Г

Скан-копії

УКРАЇНА

РІШЕННЯ

про реєстрацію договору, який стосується права автора на твір

Міністерство розвитку економіки, торгівлі та сільського господарства України розглянуло заяву

Сумський державний університет, вул. Римського-Корсакова, 2, м. Суми, 40007

(повне ім'я фізичної або повне офіційне найменування юридичної особи, адреса)

про реєстрацію авторського договору від **15 червня 2020 р. № 214** про передачу (відчуження) майнових прав і прийняло рішення зареєструвати авторський договір, відповідно до якого майнові права на твір

Комп'ютерна програма "Віртуальний тренажер установки вертикальних кутів прицілювання прицілу ПГ-4"; Дерев'янчук Анатолій Йосипович, Шелест Микола Борисович, Кравченко Дмитро Олександрович, Чичикало Євгеній Андрійович

(вид, повна, скорочена (за наявності) назва твору, повне ім'я, псевдонім (за наявності) автора(ів))

передаються(відчужуються)

Дерев'янчук Анатолій Йосипович, вул. Герасима Кондратьєва, 165/135, кв. 55; Шелест Микола Борисович, пров. Суджанський, 20, кв. 33, м. Суми, 40004, 0542-687869; Кравченко Дмитро Олександрович, пров. лікаря І. Деревянка, 6, кв. 57, м. Суми, 40022; Чичикало Євгеній Андрійович, вул. Перша Замостянська, 7, кв. 148, м. Суми, 40007

(повне ім'я фізичної(их) або повне офіційне найменування юридичної(их) особи(осіб), яка(і) передає(ють)(відчужує(ють)) право на твір, адреса)

Сумський державний університет, вул. Римського-Корсакова, 2, м. Суми, 40007, 0542-687869

(повне ім'я фізичної або повне офіційне найменування юридичної особи, якій передається (відчужується) право на твір, адреса)

Повністю

Реєстраційний номер **5247**

Дата реєстрації **13 серпня 2020 р.**

Заступник Міністра розвитку економіки, торгівлі та сільського господарства України Д. О. Романович

Рисунок Г.1 – Авторське право на тренажер вертикальної наводки прицілу ПГ-4

УКРАЇНА

РІШЕННЯ

про реєстрацію договору, який стосується права автора на твір

Державне підприємство «Український інститут інтелектуальної власності» розглянуло заяву

Сумський державний університет, вул. Римського-Корсакова, 2, м. Суми, 40007
(повне ім'я фізичної або повне офіційне найменування юридичної особи, адреса)

про реєстрацію авторського договору від **27 жовтня 2020 р. № 253** про передачу (відчуження) майнових прав і прийняло рішення зареєструвати авторський договір, відповідно до якого майнові права на твір

Комп'ютерна програма «Віртуальний тренажер прицільних пристроїв самохідних гаубиць»; Дерев'янчук Анатолій Йосипович, Шелест Микола Борисович, Кравченко Дмитро Олександрович, Чичикало Євгеній Андрійович
(вид, повна, скорочена (за наявності) назва твору, повне ім'я, псевдонім (за наявності) автора(ів))

передаються(відчужуються)

Дерев'янчук Анатолій Йосипович, вул. Герасима Кондратьєва, 165/135, кв. 55, м. Суми, 40021; Шелест Микола Борисович, пров. Суджанський, 20, кв. 33, м. Суми, 40004; Кравченко Дмитро Олександрович, пров. Лікаря І. Деревянка, 6, кв. 57, м. Суми, 40022; Чичикало Євгеній Андрійович, вул. Перша Замостянська, 7, кв. 148, м. Суми, 40007
(повне ім'я фізичної(их) або повне офіційне найменування юридичної(их) особи(осіб), яка(і) передає(ють)(відчужує(ють)) право на твір, адреса)

Сумський державний університет, вул. Римського-Корсакова, 2, м. Суми, 40007
(повне ім'я фізичної або повне офіційне найменування юридичної особи, якій передається (відчужується) право на твір, адреса)

Повністю

Реєстраційний номер **5932**

Дата реєстрації **10 грудня 2020 р.**

Т.в.о. Генерального директора Державного підприємства «Український інститут інтелектуальної власності» **Петро ІВАНЕНКО**

«УКРАЇНСЬКИЙ ІНСТИТУТ ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ» ідентифікаційний код 31032378

ПК «Україна». Зам. 20-2003. 2020 р. 10 екз.

Рисунок Г.2 – Авторське право на комплексний тренажер наводки прицілу ПГ-4

Virtual simulator
«Setting angles aiming the PG-4 sight. Vertical aiming»

Ye.A. Chichikalo, *Student*; E.G. Kuznetsov, *Senior Lecturer*
Sumy State University, Sumy, Ukraine

Currently, both traditional and computer-based methods are used by military specialists to gain knowledge and skills. However, in most cases, static images or presentations created in Microsoft PowerPoint are used. These tools at the present stage are no longer sufficient to fully show the structure and sequence of actions of individual mechanisms and the processes occurring in them.

This is a significant disadvantage when teaching students (cadets). It is also worth noting that real objects have a high learning efficiency and at the same time a high cost of maintenance and are not always available for practical use.

The goal of the project is to create an application development that will run on personal computers running the Windows operating system. The application will contain a three-dimensional model of the PG-4 sight and the movable mechanisms used to set the vertical aiming angles.

The user will be able to «interact» with the moving parts of the sight using a computer mouse or keyboard. The application will also contain reference information, which can be found by selecting the appropriate menu item.

As the main tool for creating three-dimensional models, we chose the SolidWorks application, which has the functionality to create parts of any complexity. To implement the program, the C# programming language was used in the Unity environment. Unity allows you to make scene settings (placement of three-dimensional models, texts, auxiliary graphics), make model settings (scalable size, color), and process user interaction with moving elements of the scene by using the C#programming language.

The developed simulator is already undergoing testing at the Department of Military Training of our university and has received positive feedback from the teaching staff.

Рисунок Г.3 – Тези з конференції IMA:2021

Вакал А.О., к.т.н, с.н.с.,
Кравченко Д.О.,
Чичикало Є.А.
Сумський державний університет

ІНТЕРАКТИВНИЙ ТРЕНАЖЕР ПРИЦІЛІВ САМОХІДНИХ ГАРМАТ

Вагоме місце в системі формування майбутнього офіцера артилериста, в тому числі офіцера запасу, посідають військово-технічні дисципліни, які ми виводимо як базову провідну якість фахівців, що експлуатують озброєння і військову техніку.

Стратегічним напрямом удосконалення технологій навчання у військових закладах, де здійснюється підготовка військових фахівців, є користування інформаційними технологіями навчання, які, розвиваючи ідеї програмованого навчання, відкривають нові, ще недосліджені технологічні варіанти навчання, пов'язані з унікальними можливостями сучасних комп'ютерних систем навчання.

Зазначимо, що комп'ютерні технології в системі підготовки військових фахівців для Збройних Сил України мають надзвичайну актуальність, що зумовлено низкою причин.

Одним із аспектів невирішеної проблеми успішного навчання з військово-технічних дисциплін є відсутність можливості зорового сприйняття навчального матеріалу слухачами при розміщенні вузлів і агрегатів за бронею самохідних гармат. Особливо гостро ця проблема стоїть під час вивченні будови та дії прицілів, користування їх шкалами й механізмами. Ця проблема стоїть і під час тренування навідників гармат: приціли знаходяться за бронею; тренування може здійснювати тільки одна особа; майже унеможливлюється контроль за діями навідника. Практичні заняття допомагають відпрацьовувати отримані навички у безпечному віртуальному середовищі. Тренажери надають можливість обрати не тільки індивідуальний підхід до навчання, але й зручний і гнучкий режим заняття.

Отже, метою і завданням доповіді є дослідження можливостей інтерактивних засобів навчання (мультимедійних тренажерів-самовчителів, віртуальних тренажерів, тренажерів віртуальної і доповненої реальності тощо).

Під час розробки віртуальних тренажерів необхідно враховувати наступне: інтерфейс повинен бути максимально наближений до реального зразка; 3D модель повинна враховувати основні реальні процеси взаємодії вузлів і механізмів прицілу; інструктору (керівнику заняття) надана можливість змінювати команди шляхом введення нештатних ситуацій.

Тренажер повинен робити аналіз і оцінку дій фахівця і використовуватися у наступних режимах: читання і огляд матеріалу; пошук потрібних матеріалів; друкування матеріалів (за бажанням замовника); перевірка знань; перегляд довідкового матеріалу.

У режимі читання користувач переглядає текстову частину обраного питання. Доступ до інших матеріалів стосовно іншого питання реалізується у вигляді гіперпосилання на графічні елементи або інші текстові розділи. Після натискання на гіперпосилання в окремому вікні відкривається відповідний текст, графічний або відео матеріал.

В доповіді наведена схема принципу роботи віртуального тренажера, демонструються фрагменти порядку установки кутів прицілювання, кутів місця цілі, кутоміра, оцінювання слухача як за часом, так і правильністю дій.

Отже, персонал як військових підрозділів, так і ремонтних органів зможе підвищити кваліфікацію, зменшити витрати на закупівлю дорогих тренажерів та ЗІП до них, а обслуговуючий персонал зможе отримувати навики поводження у нештатних ситуаціях тощо.

Науковий керівник: Дерев'янчук А.Й., к.т.н., професор.

Рисунок Г.4 – Тези з Шостої Всеукраїнської конференції

Шелест М.Б.,
Кравченко Д.О.,
Чичикало Є.А.
Сумський державний університет

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ для інтерактивних тренажерів ПРИЦІЛІВ САМОХІДНИХ ГАРМАТ

Однією з проблем, що стоять перед Збройними Силами України, є підвищення рівня кваліфікації фахівців-артилеристів під час виконання ними вогневих завдань як під час навчань з бойовою стрільбою, так і в зоні ООС. Недостатність часу для практичних занять (тренувань) на механізмах прицілу (встановлення прицілів, користування шкалами), тренування в башті САУ унеможливлює контроль за діями навідника. Відсутність практичних навиків навідника в процесі стрільби впливає на час і точність ведення вогню.

Зазначені обставини вказують на необхідність створення комп'ютерних тренажерів для відпрацювання операцій навідником щодо встановлення правильних прицілів і наведення гармати на ціль, враховуючи при цьому доведення його дій до автоматизму, що і визначає актуальність проблеми.

Використання комп'ютерних симуляторів та мультимедійних технологій дозволяє створювати та надавати розширені можливості для відпрацювання практичних навичок у системі підготовки артилерійських підрозділів, зокрема навідниками самохідних гармат.

Вирішення цього завдання передбачає створення віртуального симулятора прицілу як засобу підвищення фахових здібностей навідника. В свою чергу, розроблення віртуального симулятора вимагає належного програмного забезпечення.

Отже, метою доповіді є розроблення програмного забезпечення для віртуального інтерактивного симулятора прицілу самохідних гармат на основі платформи Unity.

Наша система пропонує декілька різновидів симуляційних тренажерів різних за своєю сутністю та складністю, але всі вони працюють за єдиним алгоритмом.

Проведений аналіз мов програмування показав, що найбільш раціональним рішенням є мова програмування «C sharp», яка має зручні інструменти для роботи з цільовою операційною системою проекту Windows та інструмент для розроблення 3D симуляторів на основі платформи Unity.

Подальші дослідження виявили необхідність застосування названої вище платформи Unity для розроблення 3D симулятора. Перевагами платформи є можливість розподілення ресурсів під час обробки даних, підтримка імпортування великої кількості форматів файлів; можливість тестування додатку відразу в редакторі, велика кількість документації та можливість інтегрувати кінцевий продукт в різноманітні операційні системи.

Доповідь супроводжується презентацією, що пояснює сутність обраного програмного забезпечення і роботу самої програми, демонстрацією роботи віртуального прицілу.

Простота алгоритму і оптимізація програми дозволяє встановлювати додаток на слабких комп'ютерах задля подальшої інтеграції у віртуальну реальність.

Таким чином, застосування розробленого програмного забезпечення для віртуального прицілу надає можливість підвищувати рівень кваліфікації навідників, зменшити витрати часу під час проведення занять та підвищити їх якість через контроль дій навідника.

Науковий керівник: Дерев'янчук А.Й., к.т.н, професор

Рисунок Г.5 – Тези з Шостої Всеукраїнської конференції