

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. Горизонтальна наводка"»**

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

Виконавець роботи: студент групи ІТ-71 Кравченко Дмитро Олександрович

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2021 р.

Науковий керівник

(підпис)

к.т.н., ст. викл. Кузнецов Е. Г.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

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»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
«__» _____ 2021 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Кравченко Дмитро Олександрович

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

1 керівник роботи Кузнецов Едуард Геннадійович, к.т.н., ст. викладач,

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

2 Строк подання студентом роботи «7» червня 2021 р.

3 Вхідні дані до роботи технічне завдання на розробку віртуального тренажеру прицілу ПГ-4, горизонтальній кути наведення.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз області віртуальних тренажерів, проектування інформаційної системи віртуального тренажеру, розроблення віртуального тренажеру.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність віртуального тренажеру, аналіз аналогів додатку, мета та задача дипломного проекту, аналіз технологій, етапи розробки та тестування віртуального тренажеру.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.10.2020

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	01.10.2020- 15.10.2020	
2	Створення технічного завдання	16.10.2020- 26.10.2020	
3	Проектування віртуального тренажеру	06.11.2020- 14.01.2021	
4	Створення 3D моделі прицілу	15.01.2021- 10.03.2021	
5	Проектування математичної моделі взаємодії механізмів прицілу	11.03.2021- 25.03.2021	
6	Розроблення тренажеру	26.03.2021- 11.05.2021	
7	Оформлення документації проекту	12.05.2021- 31.05.2021	

Студент

(підпис)

Кравченко Д.О.

Керівник роботи

(підпис)

к.т.н., ст. викл. Кузнецов Е. Г.

РЕФЕРАТ

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

Пояснювальна записка складається зі вступу, трьох головних розділів, висновку, списку використаних джерел складених з 15 посилань та чотирьох додатків.

Загальний обсяг пояснювальної записки складається з 94 сторінок, 57 сторінок з яких складаються з головних розділів, 2 сторінок використаних джерел, решта складається з 37 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці віртуального тренажера горизонтальних пристроїв наведення прицілу ПГ-4.

В першому розділі наведено аналіз предметної області віртуальних тренажерів та обрано інструментів реалізації додатку.

В другому розділі описано вимоги до тренажера та спроектовано інформаційну систему тренажеру за допомогою діаграми Use Case та IDEF0, визначено завдання для реалізації додатку, складені діаграми WBS, OBS та календарний план.

В третьому розділі детально описано процес практичної реалізації проекту, описано математичні залежності механізмів прицілу, проектування моделі прицілу та інтерфейсу додатка, програмна реалізація тренажеру.

Результатом проведеної роботи є Віртуальний тренажер установки горизонтальних кутів прицілювання прицілу ПГ-4.

Практичне значення роботи полягає у використанні віртуального тренажеру в процесі навчання майбутніх офіцерів-артилеристів військової кафедри СумДУ.

Ключові слова: Віртуальний, тренажер, приціл, ПГ-4, SOLIDWOKS, Unity, C#, проектування, навчання.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Проблематика навчального процесу та причини створення додатку	7
1.2 Аналіз програмних продуктів.....	7
1.3 Вибір інструментів реалізації	18
1.4 Постановка задачі	20
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ	22
2.1 Опис вимог до тренажеру	22
2.2 Проектування інформаційної системи.....	23
3 РОЗРОБЛЕННЯ ТРЕНАЖЕРУ ПРИЦІЛУ ПГ-4	26
3.1 Опис математичних залежностей механізмів оригінальної моделі.....	26
3.2 Опис структури додатку.....	29
3.3 Дизайн тренажеру	32
3.3 Програмна реалізація тренажеру.....	41
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
Додаток А Технічне завдання.....	60
Додаток Б Планування робіт.....	73
Додаток В Лістинг коду.....	82
Додаток Г Акти впровадження та апробації проекту.....	91

ВСТУП

Все частіше для підготовки спеціалістів використовують методи віртуального навчання, використання віртуальних технологій в медицині надає можливість майбутньому доктору безліч раз проводити операцію на віртуальному пацієнті, здобуваючи необхідні знання з проведення операції, не ризкуючи життям реального пацієнта. У галузі авіації, перед тим як пілот почне керувати справжнім літаком спочатку він проходить усі нормативи віртуального тренажера літака, проте порівняно с галуззю медицини та авіації офіцер артилерист має здійснювати тренування за допомогою реальних механізмів артилерійської установки, кількість та мобільність яких обмежена, як і можливість викладача контролювати дії учня.

Підготовка офіцера артилериста вимагає багато часу та ресурсів, зазвичай тренування проходять один-два рази у баштах артилерійських установок, тому куратору складно слідкувати за діями учня та перевіряти встановлені значення, а саме від кількості практичних занять офіцера залежить майбутній процент успішно виконаних завдань.

Саме тому метою даного проєкту є створення віртуального тренажера артилейского приціла ПГ-4, яким обладнують самохідну артилерійську установку «2С3», механізм приціла має багато спільного з іншими прицілами, що подалі допоможе у вивченні інших типів прицілів студенту артилеристу. Завдяки віртуальному тренажеру не буде гострих проблем нестачі тренажерів та практичних навичок, складностей з контролем навчання майбутнього офіцера.

Таким чином створення віртуального тренажера ПГ-4 на базі військової кафедри СумДУ вирішить гостру проблему практичного навчання майбутніх офіцерів-артилеристів, підвищивши якість та ефективність навчання кафедри.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Проблематика навчального процесу та причини створення додатку

На даний момент навчання офіцера артилериста проводиться за допомогою лекційних занять, на яких вивчають теорію розрахунку кутів наведення, технічні характеристики зарядів, будову бойових машин, проте результативність практичних занять низька, кафедра має лише один тренажер прицілу ПГ-4 знятий з артилерійської установки, тому класу учнів, який зазвичай складається з тридцяти майбутніх офіцерів має витратити не більше двох хвилин на практичне навчання учня, також викладач ще повинен контролювати дії навідника та перевіряти встановлені значення, що призводить до низької якості навчання.

Схожа ситуація була з тренажерами будови самохідної артилерійської установки 2С3 «Акація» та установки детонатору артилійского заряду, після впровадження віртуальних версій тренажерів [1], якість навчання та ефективність кафедри було підвищено в декілька разів, що підтверджує необхідність впровадження віртуальних тренажерів.

1.2 Аналіз програмних продуктів

Для створення віртуального тренажера, крім поставленої мети необхідно проаналізувати наявні аналоги. Дослідивши недоліки та переваги схожих додатків, щоб створити якісний, унікальний продукт. Аналогів віртуального тренажеру ПГ-4 немає, розроблення такого тренажеру ведеться вперше, проте є схожі за метою та складом тренажери:

Тренажер заряду ЖН-546 [1] – це додаток створений на базі військової кафедри СумДУ, користувач за допомогою інтерактивної взаємодії з інструментами для заряду та самим зарядом повинен за певний час встановити підрильник у активне положення, також наводиться будова компонентів заряду, приклад тренажеру наведено на рисунку 1.1.



Рисунок 1.1 – Тренажер заряду ЖН-546

Після встановлення детонатору заряду та натискання кнопки «Закінчити тест», користувачу надається стисла інформація щодо кількості здійснених помилок та витраченого часу, інформаційне вікно зображено на рисунку 1.2

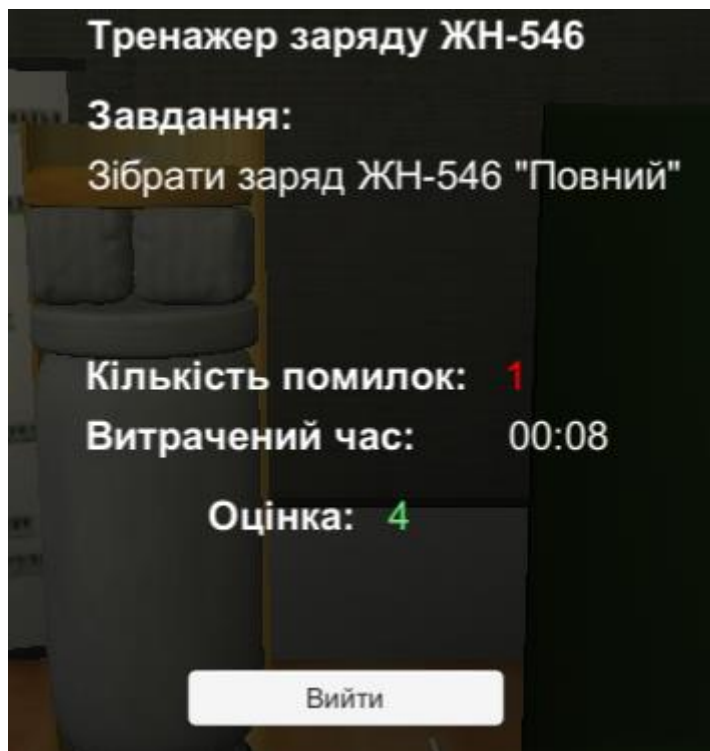


Рисунок 1.2 – Тренажер заряду ЖН-546

Зробимо оцінку тренажеру за п'ятьма критеріями, які були сформовані за побажаннями учнів кафедри та самою метою тренажера, оберемо десятибальну шкалу оцінювання, оцінювання тренажеру наведено в таблиці 1.1.

Таблиця 1.1 – Оцінювання тренажеру ЖН-546

Критерій	Оцінка	Коментар
Зручність використання	5	Тренажер не має інструкції щодо користування, головного меню, масштабованість зображення відсутня.
Мобільність	8	Тренажер доступний лише на комп'ютері з операційною системою Windows 7 або вище.

Продовження таблиці 1.1 – Оцінювання тренажеру ЖН-546

Критерій	Оцінка	Коментар
Стабільність тренажеру	9	Тренажер працює стабільно. Проте з нього неможливо вийти, лише через примусове закриття додатку.
Якість відображення навчальної моделі	4	Навчальна модель має низьке розширення, модель неможливо досліджувати (приближувати/віддаляти, обертати)
Інтерактивність моделі	5	Елементи для взаємодії не видно, взаємодія відбувається за допомогою анімації, дії користувача необхідні лише для запуску статичної анімації.
Якість оцінювання дій користувача	4	Витрачений час на завдання ніяк не впливає на оцінку, у інформаційному вікні написано лише кількість помилок, проте саме яких не зазначено.

Тренажер БТР-4Е [2] – Динамічний тренажер машини БТР-4Е, призначений для навчання водіїв, навідників та командирів. Тренажер складається з динамічною платформи на якій знаходиться кабіна БТР-4Е та контролерів всередині кабіни, усі дії відбуваються на віртуальному полігоні, приклад тренажеру наведено на рисунку 1.3.



Рисунок 1.3 – Загальний вигляд тренажеру БТР-4Е

Тренажер може навчати відразу декількох учнів за різними спеціальностями, навчання проводиться одночасно з водієм, навідником та командиром, під час занять викладач в режимі реального часу має змогу переглядати успіхи учнів за окремим столом. На теперішній час це найсучасніші військові тренажери України. Проаналізуємо даний тренажер та занесемо дані у таблицю 1.2.

Таблиця 1.2 – Оцінювання тренажеру БТР-4Е

Критерій	Оцінка	Коментар
Зручність використання	9	Тренажер є повною копією кабіни БТР з пристроями відповідно до реального БТР, проте динамічні платформи потрібно технічно обслуговувати.

Продовження таблиці 1.2 – Оцінювання тренажеру БТР-4Е

Критерій	Оцінка	Коментар
Мобільність	3	Для тренажеру необхідний спеціально обладнаний кабінет, який займає велику площу.
Стабільність тренажеру	10	Тренажер працює стабільно.
Якість відображення навчальної моделі	10	Повна фізична копія усіх пристроїв кабіни БТР.
Інтерактивність моделі	10	Через копію усіх механізмів кабіни БТР та динамічної платформи забезпечує повне занурення у практичний навчальний процес.
Якість оцінювання дій користувача	10	В інформаційному вікні міститься уся інформація про швидкість, положення та місцезнаходження БТР на полігоні, статичний аналіз пострілів гармати БТР, також дані можна переглянути саме під час виконання завдання.

Тренажер Мі-24П [3] – Тренажер льотчика гелікоптера Мі-24П за будовою схожий на тренажер БТР-4Е, складається з копії приладів справжнього літака, віртуального полігону на якому відбувається виконання завдань та пульта контролю, викладач може переглядати результати учня, загальний вигляд тренажера наведено на рисунку 1.4.



Рисунок 1.4 – Загальний вигляд тренажеру Мі-24П

Тренажер має набагато більше приладів порівняно з тренажером БТР-4Е, проте сама кабіна пілота знаходиться на статичній платформі. Зробимо оцінювання даного тренажеру та занесемо дані у таблицю 1.3.

Таблиця 1.3 – Оцінювання тренажеру Мі-24П

Критерій	Оцінка	Коментар
Зручність використання	7	Копія приладів з гелікоптера допомагає пілотам швидше звикати до справжнього Мі-24П.
Мобільність	5	Для тренажеру необхідний спеціально обладнаний кабінет та сама платформа з пристроями.

Продовження таблиці 1.3 – Оцінювання тренажеру Мі-24П

Критерій	Оцінка	Коментар
Стабільність тренажеру	10	Тренажер працює стабільно.
Якість відображення навчальної моделі	10	Повна фізична копія усіх пристроїв кабіни гелікоптера Мі-24П.
Інтерактивність моделі	8	Усі механізми інтерактивні та реагують на дії пілота
Якість оцінювання дій користувача	9	Викладач може переглядати усі дані гелікоптера під час заняття та після його завершення, інформаційне вікно містить статистичні дані щодо часу точності виконання завдання учнем.

Оцінивши найближчі аналоги, зробимо оцінку тренажера прицілу ПГ-4. Тренажер горизонтальних пристроїв наведення прицілу ПГ-4 – це пристрій, а саме панорама Герца [4] завдяки якій виконується наведення гармати артилерійської установки 2С3 «Акація» на ціль. Складається тренажер з панорами, та пристроїв для її керування (рис. 1.5).

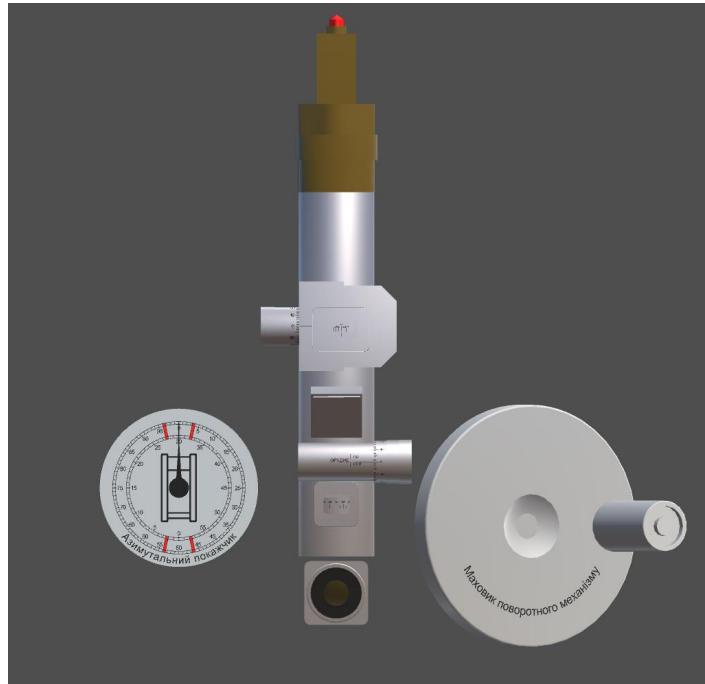


Рисунок 1.5 – Модель тренажеру ПГ-4

Після встановлення значень наведення кута заряду та натискання кнопки «Перевірка» інформаційне вікно зображено на рисунку 1.6

Час	Оцінка	Похибка
00 хв 42 сек	Незадовільно	Похибка кутоміра -31

Меню

Рисунок 1.6 – Оцінка встановлених параметрів

Оцінимо тренажер прицілу артилерійської установки за критеріями наведеними у таблиці 1.3.

Таблиця 1.3 – Оцінювання тренажеру ПГ-4

Критерій	Оцінка	Коментар
Зручність використання	9	Тренажер має головне меню з довідковою інформацією та відео інструкцію з прикладом використання тренажеру, усі елементи динамічні та реагують на дії користувача.
Мобільність	8	Тренажер доступний на комп'ютерах з операційною системою Windows 7 або вище.
Стабільність тренажеру	10	Тренажер працює стабільно.
Якість відображення навчальної моделі	9	Навчальна 3D модель спроектована відповідно до оригіналу, модель можливо обертати та досліджувати
Інтерактивність моделі	9	Маховики та їх покажчики пов'язані один з одним та залежать від дій користувача.
Якість оцінювання дій користувача	10	Витрачений час на завдання впливає на оцінку, інформаційне вікно містить дані щодо допущених помилок та оцінку учня.

Проаналізувавши тренажер та його аналоги побудуємо таблицю з загальними оцінками тренажерів та зробимо висновок щодо тренажера ПГ-4 та його аналогів, загальну оцінку наведено в таблиці 1.4.

Таблиця 1.4 – Оцінка тренажерів

	ПГ-4	ЖН-546	БТР-4Е	Мі-24П
Зручність використання	9	5	9	7
Мобільність	8	8	3	5
Стабільність тренажеру	10	9	10	10
Якість відображення навчальної моделі	9	4	10	10
Інтерактивність моделі	9	5	10	8
Якість оцінювання дій користувача	10	4	10	9
Загальна кількість балів	55	35	52	49

Найближчим схожим за реалізацією та метою додатка є тренажер ЖН-546, проте в якості реалізації, а саме моделі для тренування, оцінювання дій учня та стабільності тренажер програє.

У критеріях інтерактивності, якості відображення моделі тренажер ПГ-4 програє аналогам БТР-4Е та Мі-24П, через те що дані тренажери мають фізичну копію своїх приладів для взаємодії, проте саме завдяки цьому ПГ-4 набирає багато балів у мобільності, для нього не потрібно ніяких складних обладнань та великих приміщень лише комп'ютер.

1.3 Вибір інструментів реалізації

Наступним етапом необхідно обрати інструменти досягнення мети. Необхідні інструменти для моделювання та програмування віртуальної моделі прицілу. За допомогою обраних інструментів спроектувати модель та запрограмувати її на інтерактивну взаємодію. Серед великої кількості програм для проектування моделі виділимо основні та серед них оберемо програму для проектування моделі прицілу, використовуючи порівняльну таблицю 1.5.

Таблиця 1.5 – Порівняння програм для 3D моделювання

	Blender	3ds Max	SOLIDWOKS
Ключові особливості	Існує багато модифікацій, легкий та зрозумілий інтерфейс	Вважається найкращим інструментом для створення 3d моделей, завдяки інструментам редагування полігонів	Серед конкурентів має перевагу в проектуванні інженерних деталей
Складність використання	Легко	Середня складність	Середня складність

Найкращим варіантом для додатку буде програма «SOLIDWOKS» [5], додаток створений спеціально для проектування інженерних деталей завдяки чому створена модель буде мати ідентичні розміри та пропорції відносно оригіналу.

Наступним чином необхідно обрати платформу для створення додатку. Зазвичай використовують платформи Unity, Unreal Engine та Godot [6] порівняння наведено в таблиці 1.6.

Таблиця 1.6 – Порівняння платформ розробки

	Unity	Unreal Engine	Godot
Графіка	Гнучка в налаштуванні графіка проте, не найкраща серед можливих	Одна з найкращих платформ орієнтована на якість графічного відображення	Низька якість графіки порівняно з іншими платформами

Продовження таблиці 1.6 – Порівняння платформ розробки

Складність розробки	Легко	Середня складність	Дуже гарна платформа для початківців
Ключові особливості	Велика кількість готових рішень та документації, легкий та інтуїтивний інтерфейс	Висока якість графіки та оптимізації	Має свою унікальну мову програмування GDScript

Для розробки обрано платформу Unity, мовою програмування якої є C sharp [7] порівняно з Unreal Engine графічне відображення моделі буде нижче, проте швидкість та якість розробки буде краща.

1.4 Постановка задачі

Метою роботи є створення віртуального тренажеру артилійського приціла ПГ-4 для навчання майбутніх офіцерів військової кафедри артилерійської підготовки СумДУ. Віртуальні тренажери є сучасною методикою для підвищення якості знань, порівняно з фізичними аналогами мають переваги у мобільності, гнучкості використання та ціни реалізації.

Основними задачами для створення якісного додатку необхідно:

- Дослідити предметну область;
- Спроекувати інформаційну систему проекту, календарний план робіт реалізації;
- Створити віртуальну модель прицілу «ПГ-4»;

- Зробити віртуальну модель інтерактивною;
- Розробити інтерфейс додатку;
- Протестувати додаток.

Проаналізувати найближчі аналоги додатку виділивши їх переваги та зазначити сильні сторони аналогів під час створення вимог до реалізації.

Виконати проектування інформаційної системи розподілити пріоритетність робіт, а саме побудувати діаграми Use-Case [8], IDEF0 [9], діаграми планування структури робіт [10] та діаграму Ганта [11].

Дослідити фізичну модель прицілу та створити віртуальні копії механізмів моделі відповідно до оригіналу, використовуючи програму для моделювання «SOLIDWOKS».

Зробити модель приціла інтерактивною, для цього обрано платформу для створення інтерактивних додатків «Unity», серед інших інструментів дана платформа відрізняється великою кількістю посібників, модифікацій та крос-платформеністю. Платформа «Unity» [12] використовує мову програмування C# [13], яка є об'єктно-орієнтованою мовою програмування.

Розробити інтерфейс додатку, який має бути не перевантаженим та допомагати учню взаємодіяти з моделлю, кожен елемент інтерфейсу має бути інтуїтивно зрозумілим та описаним у довідці.

На кожному етапі розробки необхідно виконувати тестування додатку, приділяючи увагу зручності тренажеру, відповідності до оригіналу, точності встановлювальних даних та їх перевірки.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ

Для чіткої та поетапної реалізації додатку необхідно описати вимоги тренажеру, взявши до уваги мету проєкту та переваги найближчих аналогів. Спроектувати архітектуру інформаційної системи за допомогою діаграм Use Case та IDEF0, створити календарний план та діаграму розподілу робіт необхідних для реалізації проєкту.

2.1 Опис вимог до тренажеру

Викладачі військової кафедри з артилерійської підготовки сформувавши вимоги до тренажеру, в якому також мають бути реалізовані переваги найближчі аналогів, вимоги детально описані у Додатку А.

Серед головних вимог є можливість перегляду та взаємодії з елементами моделі приціла ПГ-4, тобто елементи моделі мають бути пов'язані один з одним, користувач має змогу детально переглядати модель. Були сформовані необхідні критерії оцінювання та формування завдання для учня, додаток має генерувати завдання наведення, аналізувати дії учня та рахувати оцінку автоматично. Головне меню додатку повинно містити довідку в якій має бути описаний алгоритм роботи тренажера та способи взаємодії користувача з віртуальним прицілом. Тренажер потрібен бути невибагливим та стабільним, щоб охопити велику кількість комп'ютерів кафедри. Оцінку тренажеру необхідно провести за допомогою складених тест кейсів, для підвищення якості оцінювання перевірити тренажер саме у навчальному процесі військової кафедри.

2.2 Проектування інформаційної системи

Дослідивши варіанти використання додатку визначено такі основні операції застосування:

- Перегляд довідки;
- Введення завдання прицілювання;
- Виконання завдання прицілювання;
- Отримання оцінки за виконання завдання.

За якими було складено діаграму User Case. Діаграма Use Case використовується для зображення сценаріїв використання додатку користувачами системи або акторами, які використовують функції додатку [8]. Акторами в системі є учень та викладач, діаграму User Case наведено на рисунку 2.1.



Рисунок 2.1 – Діаграма Use Case

Учень має змогу переглянути довідку з інструкцією користування прицілом ПГ-4. Почати виконувати завдання самостійно вводячи дані для прицілювання або автоматично, після виконання завдання додаток виставляє оцінку зі статичними даними учня, викладач переглядає оцінку та заносить її в журнал.

За допомогою діаграми IDEF0 [9]. Діаграма IDEF0 зображує логічні зв'язки між роботами необхідними для виконання завдання наведення. Розглянемо роботи діаграми IDEF0 на рисунку 2.2:

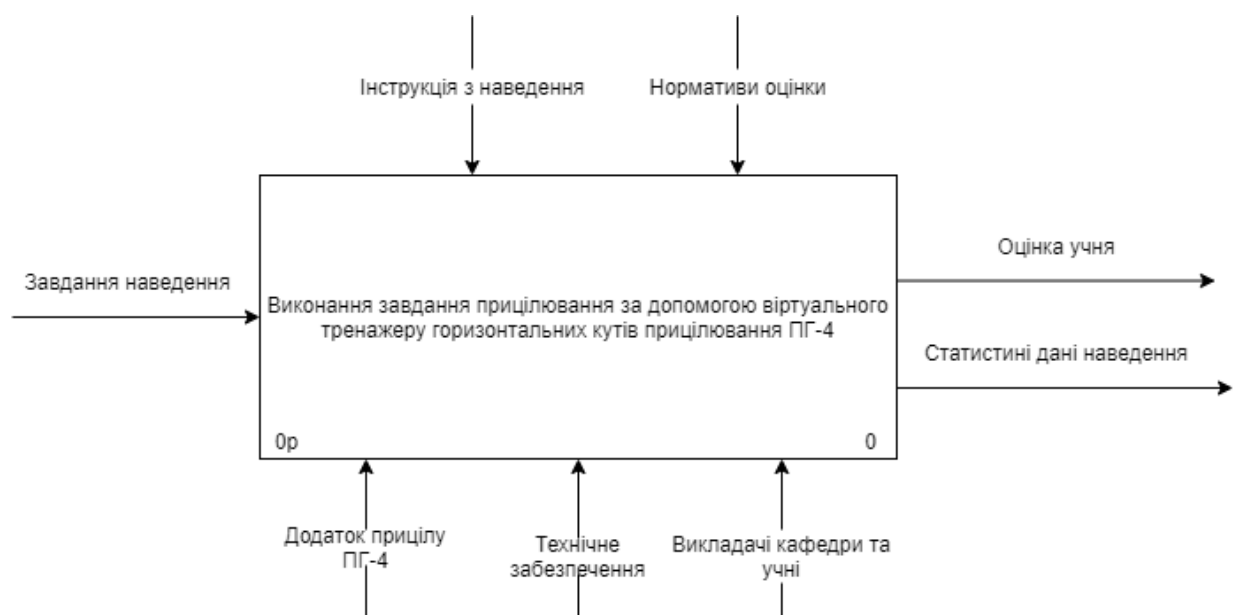


Рисунок 2.2 – Модель IDEF0

На вході завдання наведення сформоване викладачем або автоматично згенеровано тренажером. Серед керуючих механізмів інструкція з наведення та нормативи оцінювання учня, для виконання завдання використовуються учень та викладач кафедри артилерійської підготовки, технічне забезпечення, тобто комп'ютер та додаток прицілу ПГ-4.

Результатом є оцінка учня виставлена викладачем або додатком та статистичні дані учня зібрані додатком під час виконання завдання.

На основі діаграми IDEF0 побудуємо деталізовану модель головного блоку. Діаграму декомпозиції IDEF0 наведено на рисунку 2.3.

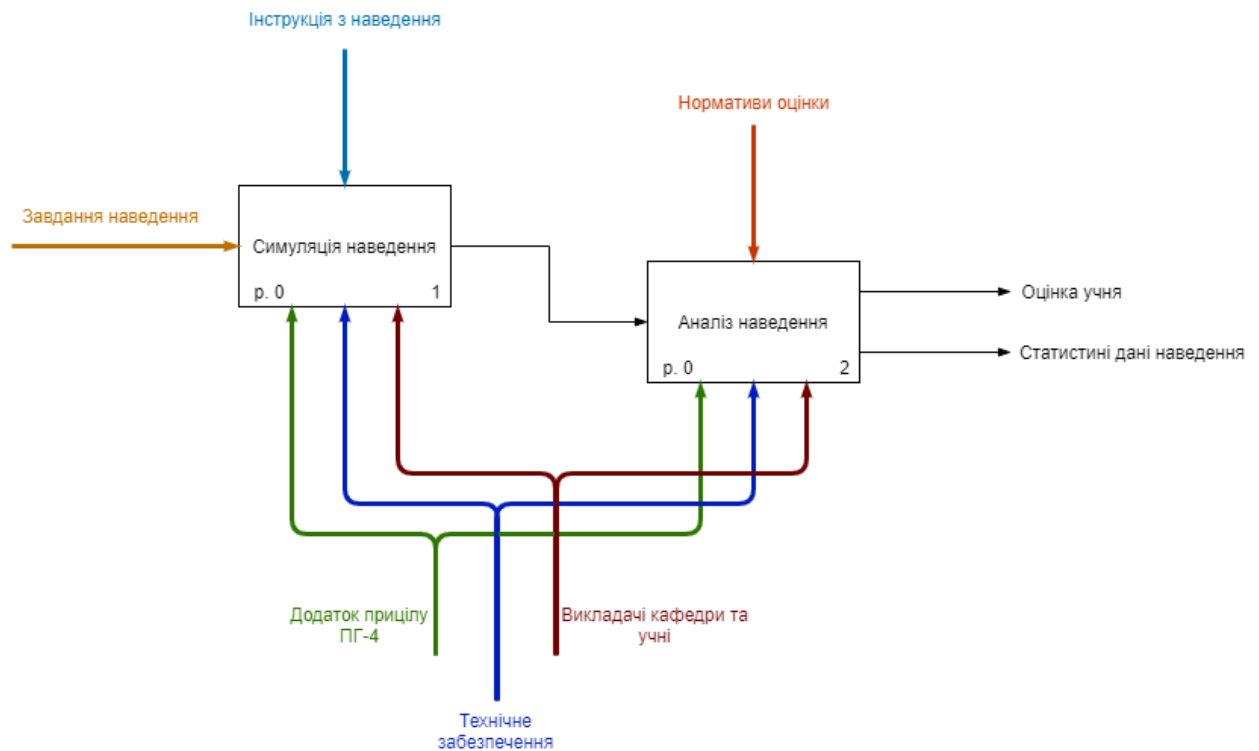


Рисунок 2.3 – Декомпозиція діаграми IDEF0

Після того як учень отримав завдання він здійснює наведення за допомогою тренажеру, попередньо прочитавши інструкцію або під наглядом викладача. Коли завдання наведення виконане учень натискає кнопку «Перевірка», додаток наводить час то точність встановлених даних, та розраховує оцінку за встановленими нормативами.

Завдяки сформованим вимогам та побудованим діаграмам була створена діаграми робіт WBS та OBS необхідними для планування пріоритетності завдань реалізації проекту, побудований календарний план робіт проекту який зображено за допомогою діаграми Ганта, деталізовано мету проекту, створено діаграму пріоритетності робіт та календарний плану проекту який наведені у додатку Б.

3 РОЗРОБЛЕННЯ ТРЕНАЖЕРУ ПРИЦІЛУ ПГ-4

Для створення додатку необхідно проаналізувати математичні залежності між елементами приціла, описати структуру додатку, спроектувати віртуальну модель приціла за допомогою програми «SOLIDWORKS» та реалізувати інтерактивність моделі за допомогою платформи для розробки додатків «Unity» та мови програмування C sharp.

3.1 Опис математичних залежностей механізмів оригінальної моделі

Одним з найголовніших критеріїв тренажеру має бути точність розрахунку значень кутів прицілу порівняно з оригіналом. Для розробки моделі необхідно дослідити алгоритм наведення, який зображено на рисунку 3.1.

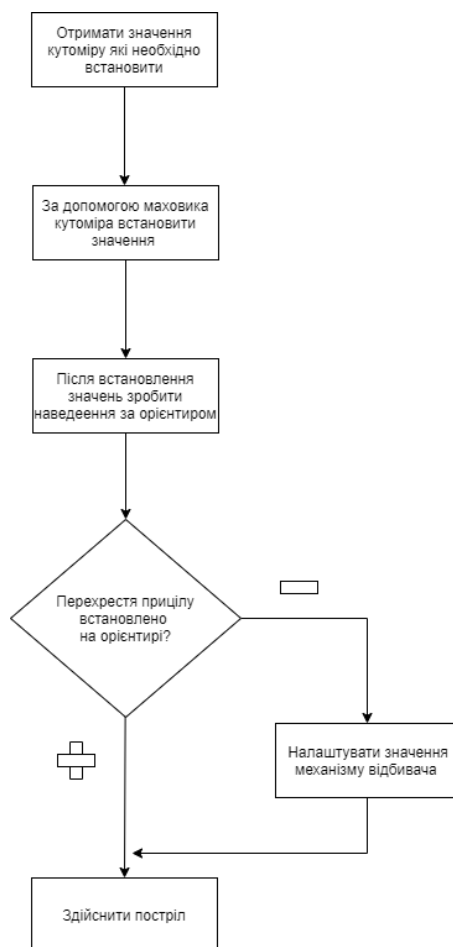


Рисунок 3.1 – Алгоритм наведення

Для початку визначимо інтервал значень кутоміру. Значення кутоміру представлено у спеціальних артилерійських вимірах для підвищення точності прицілу та зручності передання даних [14]. Найменша поділка кутоміру це тисячна, тобто $1/6000$ частина довжини кола та дорівнює 3.6 хвилини кута, груба поділка яка містить 100 найменших поділок шкали кутоміру дорівнює 6 градусів, тобто в одній грубій поділці 360 хвилин, маховика кутоміру та грубої шкали кутоміру приклад наведено на рисунку 3.2.



Рисунок 3.2 – Шкали кутоміру

У фізичному прицілі конвертація значень у артилерійських виміри відбувався за допомогою складного електротехнічного механізму, проте для віртуального тренажеру необхідно розробити формулу конвертації значень руху миші у звичайні кутові значення, а потім у артилерійські позначки. Сформуємо задачу, умову задачі наведено нижче в таблиці 3.1

Таблиця 3.1 – Вхідні та вихідні дані кутів наведення

		Інтервал
Дано:	Маховик прицілу	[0, 100)
	Значення руху миші	[-1, 1]
Знайти:	Формулу оберту шкали прицілу залежно від оберту маховика.	[0, 360)
	Формулу знаходження значення кутоміра.	[0, 6000)

Знайдемо коефіцієнт між кутом повороту маховика та грубою шкалою, маємо 60 обертів маховика дорівнює повному оберту шкали прицілу, тоді 1 оберт маховика відносно шкали буде $\frac{60}{6000} = 0.01$, якщо перевести значення в кути один оберт маховика це $0.01 * 360^\circ = 3.6^\circ$ градусів оберту грубої шкали.

Нехай x це значення руху миші (від -1 до 1), тоді за кожні x значень кута повороту маховика кутоміра груба шкала прицілу має обертатися на $\frac{x}{3.6}$ градусів, де від'ємне значення це обертання проти годинникової стрілки, а додатне це обертання за годинниковою стрілкою згідно з оригіналом прицілу.

Далі знайдемо формулу для визначення значення шкали прицілу. Нехай x це значення повороту маховика (від 0 до 360 градусів), а k кількість повних обертів маховика, якщо один повний оберт маховика це одна велика позначка шкали кутоміра та коефіцієнт між кутом повороту маховика та значенням шкали дорівнює 3.6° градусів.

Тоді значення малої позначки шкали буде $\frac{x}{3.6^\circ}$, а значення великої кількості k обертів маховика, повна формула матиме вигляд:

$$100 * k + \frac{x}{3.6^\circ}$$

Виходячи з цього для розрахунку куту обертання грубої шкали використовуємо формулу $\frac{x}{3.6^\circ}$, а для визначення встановлених значень кутоміра $100 * k + \frac{x}{3.6^\circ}$, де x це значення кута повороту маховика кутоміра, а k кількість обертів.

3.2 Опис структури додатку

Згідно вимог поставлених у другому розділі, додаток має містити головне меню з довідковою інформацією та можливістю генерації завдання автоматично або за даними користувача, учень повинен мати змогу досліджувати модель та взаємодіяти з маховиками приціла, систему оцінювання учня, а саме витрачений час та точність наведення.

Алгоритм роботи з додатком:

- Переглянути довідку;
- Створити завдання наведення самостійно або згенерувати завдання випадково;
- За допомогою маховика кутоміра встановити значення;
- Після встановлення значень зробити наведення за орієнтиром;
- Якщо перехрестя прицілу не встановлено на орієнтирі, налаштувати значення механізму відбивача;
- Натиснути на кнопку «Перевірка»;
- Переглянути інформаційне вікно з похибками при встановленні значення витраченим часом та оцінкою.

Додаток можна поділити на 3 частини між якими передаються дані, головне меню, головна сцена, віртуальний полігон, за допомогою діаграми на рисунку 3.3 зображено взаємодію між ними.



Рисунок 3.3 – Взаємодія між частинами тренажера

У головному меню формується завдання наведення, дані передаються до головної сцени. Головна сцена містить маховики для наведення, проте для того щоб націлитись на орієнтир необхідно перейти до віртуального полігону та зробити наведення, під час переходу з однієї частини до іншої дані та положення маховиків зберігається. Після того як наведення здійснено у головній сцені необхідно натиснути на кнопку перевірки, з'явиться вікно з оцінкою, похибками кутоміру та кнопкою повернення у головне меню.

Детальні вимоги роботи тренажера, виставлення пріоритетів вимог, ілюстрації роботи додатку наведені у додатку А.

3.3 Дизайн тренажера

Моделі розроблено за допомогою «SOLIDWORKS» та фотографій оригіналу. Спочатку були зняті розміри з оригінально приціла, приклад наведено на рисунку 3.4



Рисунок 3.4 – Зняття розмірів панорами приціла

Після чого деталь за розмірами оригіналу було відтворено у «SOLIDWORKS» (рис. 3.5).

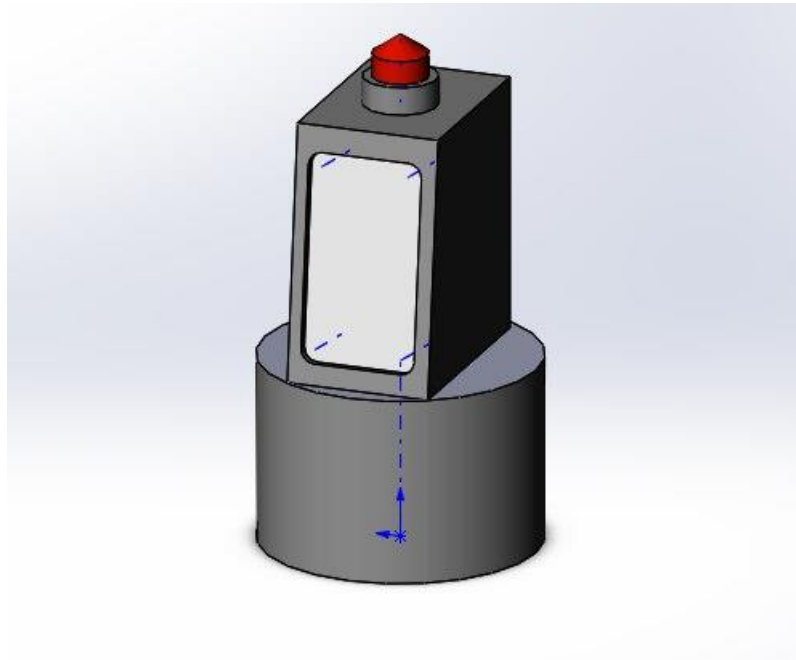


Рисунок 3.5 – Панорама прицілу ПГ-4

Наступним чином відтворено деталі усього прицілу, вони зображені та перераховані на рисунку 3.6.

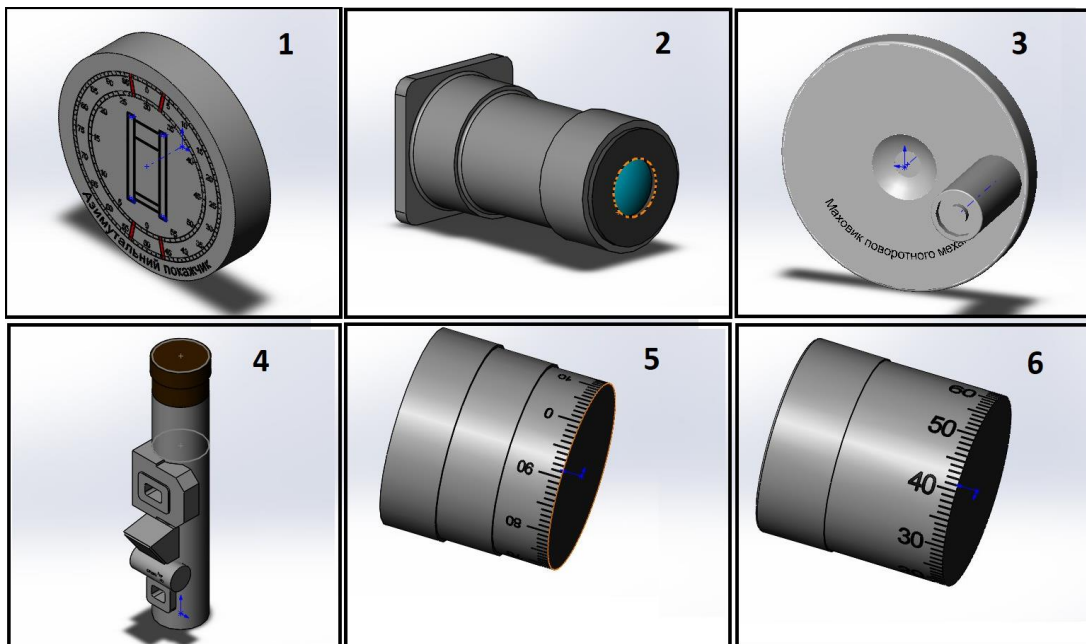


Рисунок 3.6 – Деталі прицілу ПГ-4

На рисунку в наступному порядку зображені:

1. Азимутальний показчик;
2. Монокуляр;
3. Маховик поворотного механізму башти;
4. Корпус прицілу;
5. Маховик кутоміра;
6. Маховик механізму відбивача панорами.

Після проектування деталей вони були пофарбовані в програмі «Unity», матеріали були підібрані згідно моделі та мають металевий блиск (рис.3.7).

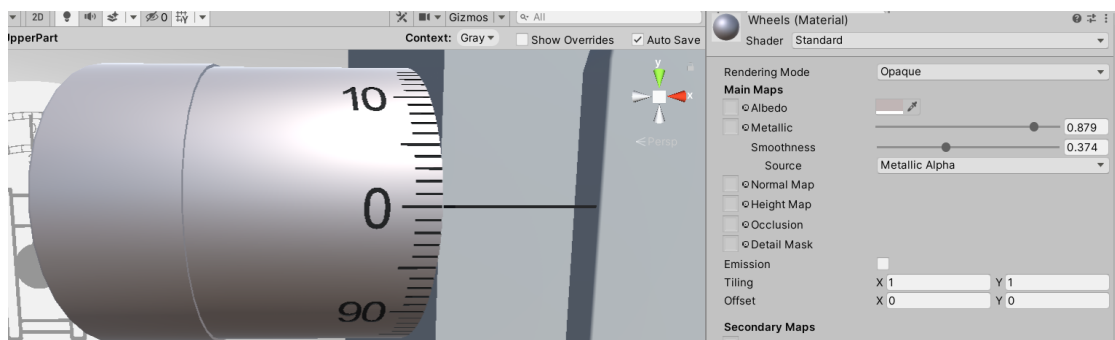


Рисунок 3.7 – Приклад фарбування деталі тренажера

Зібравши усі деталі воєдино та розмістивши їх положення згідно з оригіналом отримано віртуальну модель прицілу ПГ-4 механізмів горизонтальних кутів прицілювання, приклад наведено на рисунку 3.8

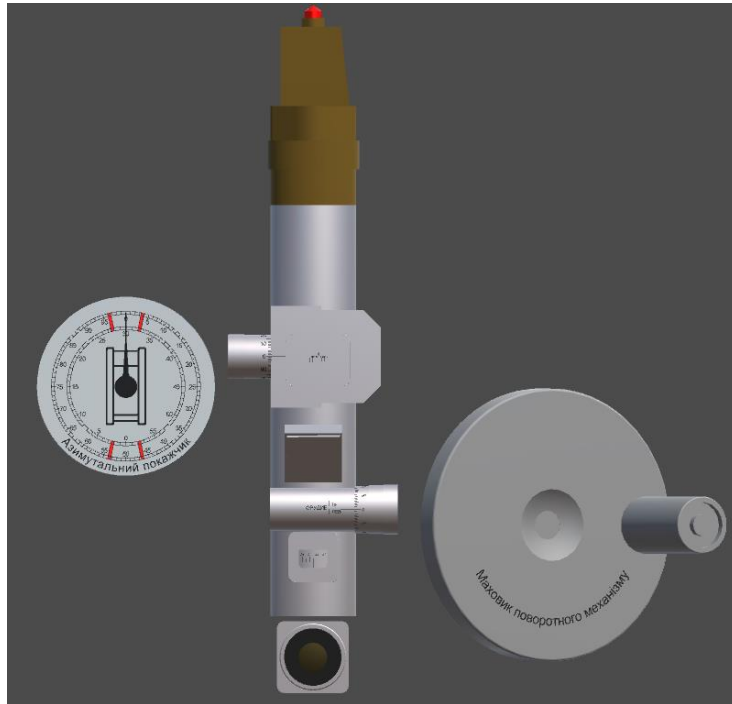


Рисунок 3.8 – Віртуальна модель прицілу ПГ-4

Інтерфейс тренажера відіграє велику роль під час ознайомлення користувача з додатком. Додаток складається з трьох частин, головне меню, головна сцена, віртуальний полігон. Наведемо демонстрацію інтерфейсу кожної частини, на рисунку 3.9 зображено головне меню додатку.

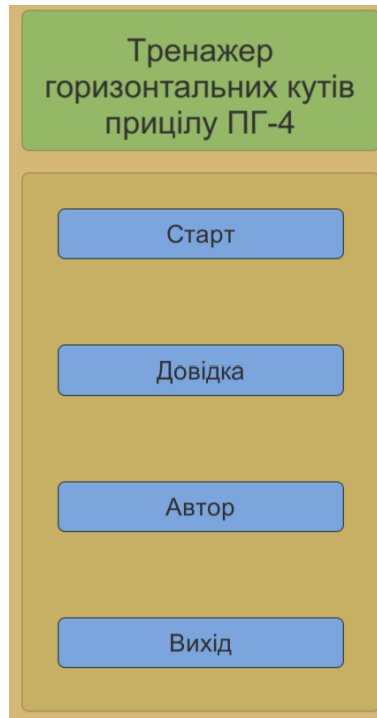


Рисунок 3.9 – Головне меню тренажеру

Головне меню складається з довідки, списку авторів та кнопки старт. На рисунку 3.10 наведено довідку.

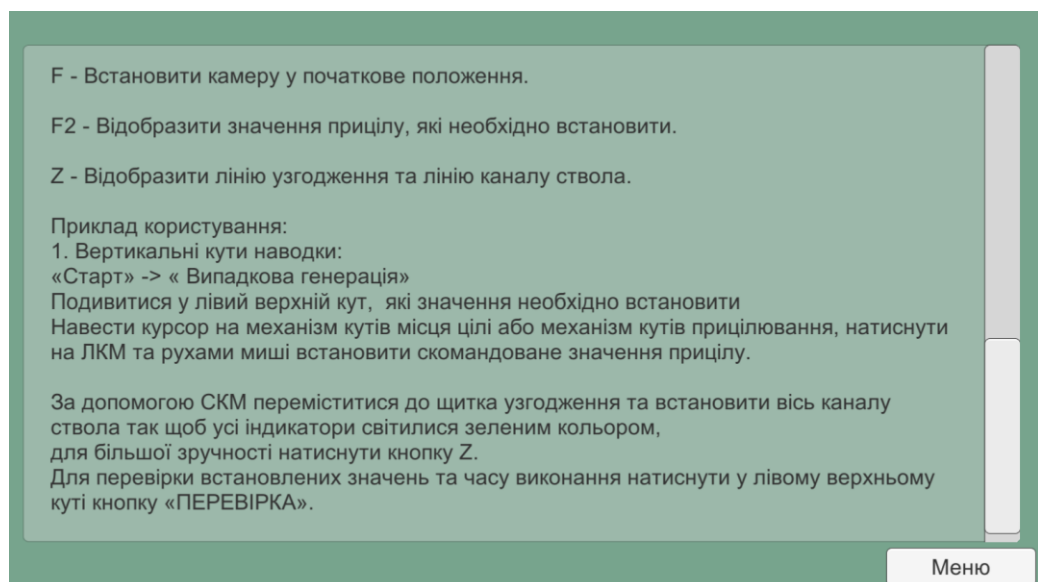


Рисунок 3.10 – Вікно довідки

Після натискання кнопки старт користувачу пропонується обрати спосіб генерації завдання, приклад наведено на рисунку 3.11.

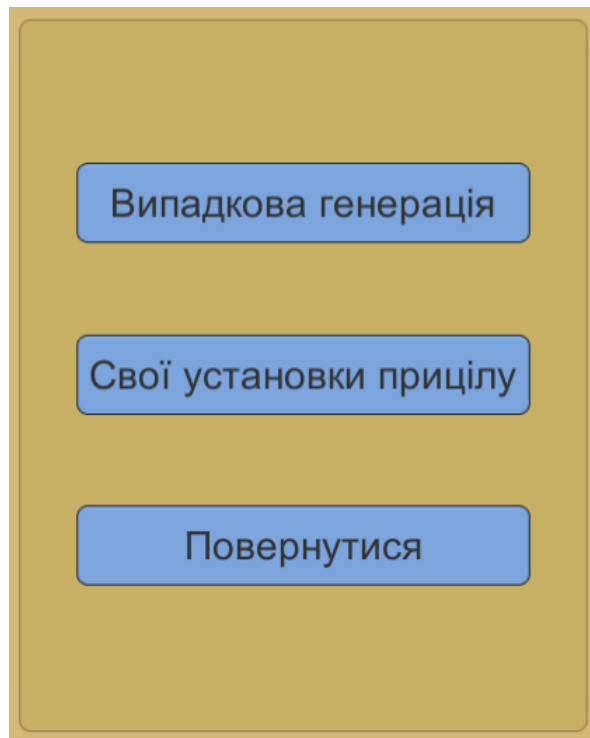


Рисунок 3.11 – Способи генерації завдання

Якщо користувач обирає «Випадкова генерація» він відразу переходить з випадково згенерованим завданням до головної сцени тренажера, натиснувши «Свої установки прицілу», з'являється вікно для заповнення даних завдання прицілювання, що зображено на рисунку 3.12.

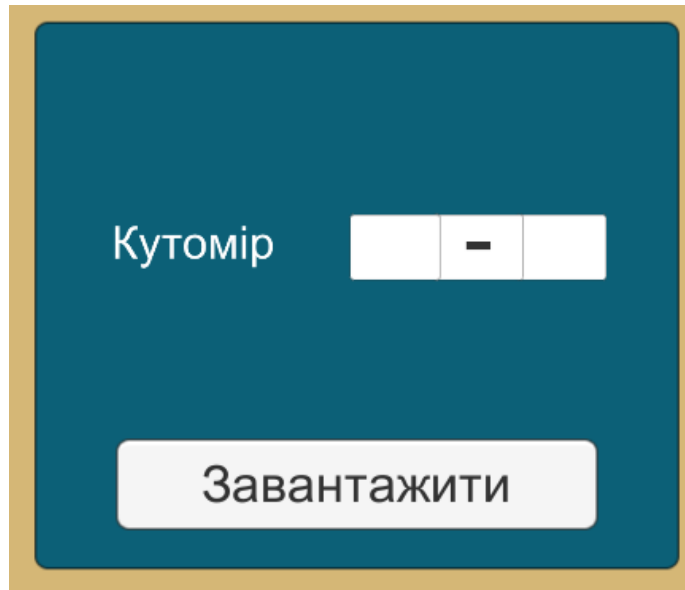


Рисунок 3.12 – Встановлення значення кутоміра

Усі введені дані перевіряються та надають коментар у разі неправильного формату введення, вікно помилки наведено на рисунку 3.13

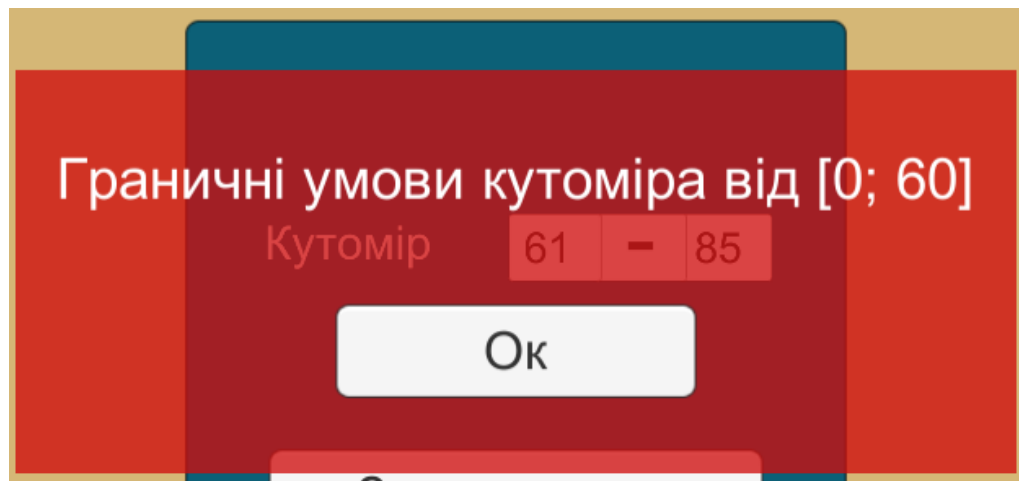


Рисунок 3.13 – Помилка при введенні значень кутоміра

Наступна частина додатку «Головна сцена» містить модель прицілу для взаємодії (рис. 3.14).

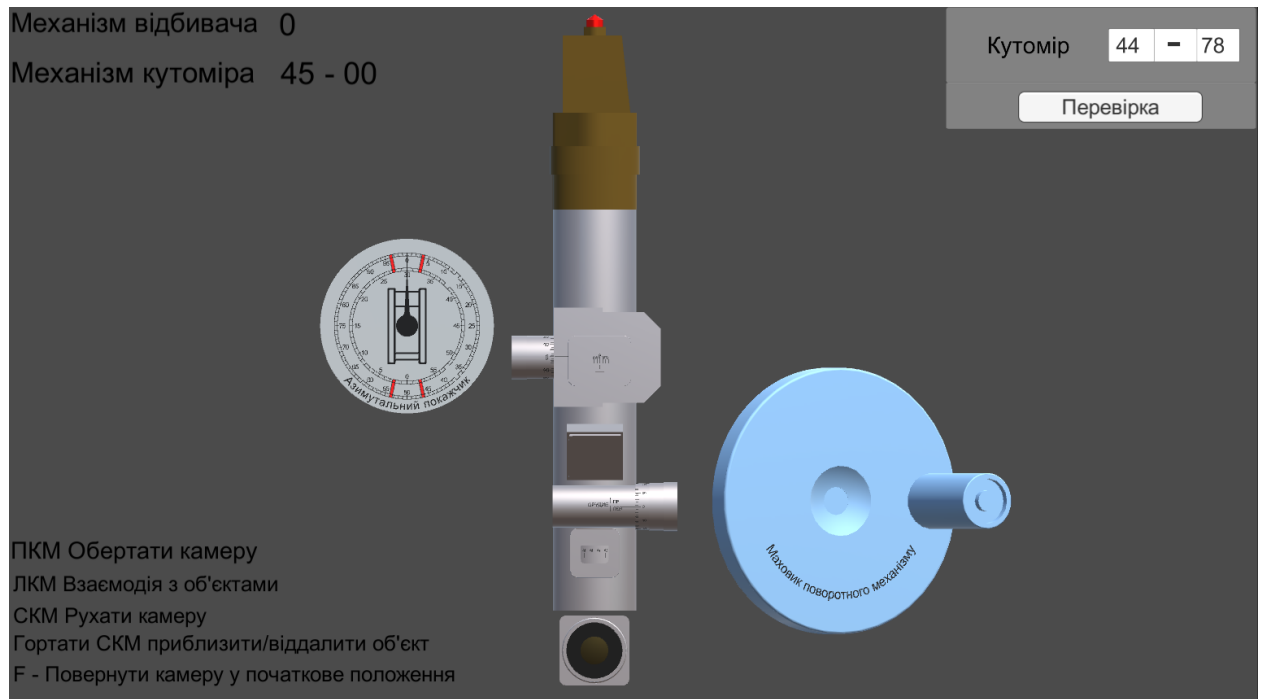


Рисунок 3.14 – Головна сцена додатку

Усі маховики прицілу інтерактивні та відокремлюються синім кольором при взаємодії. Натиснувши клавішу «F1» з'явиться коротка інструкція щодо взаємодії з моделлю та встановленими значеннями. У правому верхньому куті наведено завдання яке було сформоване у головному меню. Після встановлення значень кутоміра користувач натиснувши на монокуляр переходить до віртуального полігона, який зображено на рисунку 3.15.

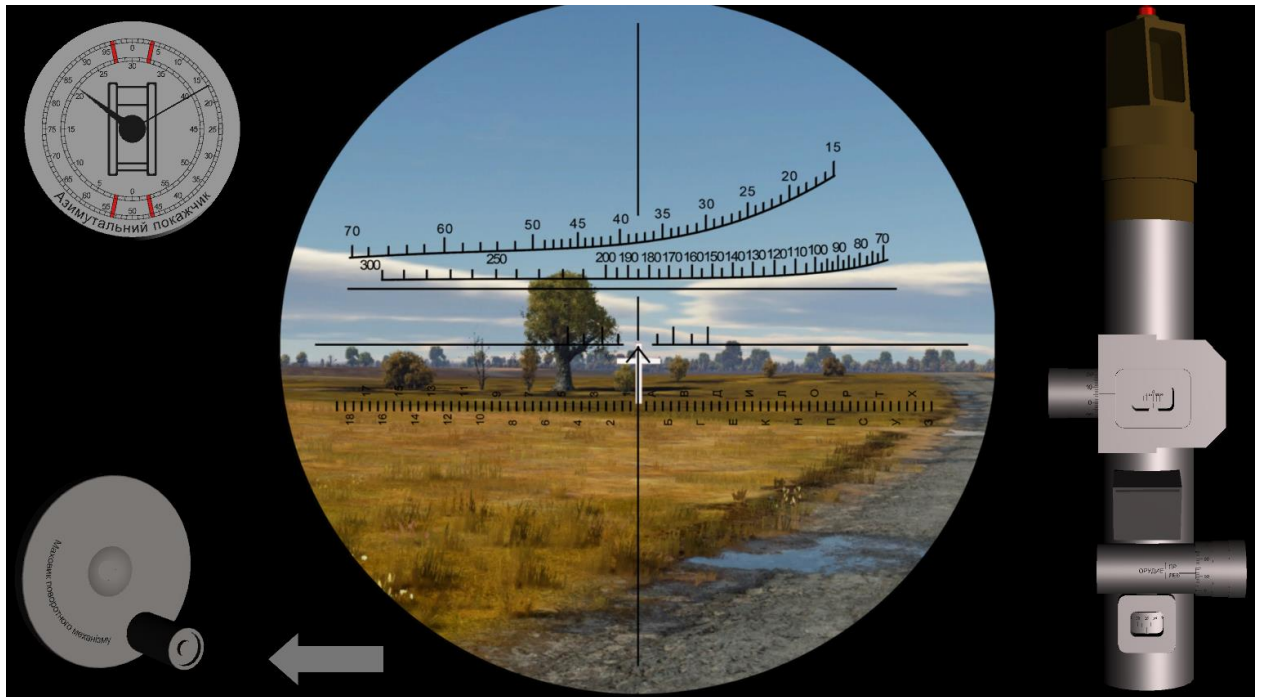


Рисунок 3.15 – Віртуальний полігон

Усі елементи приціла, крім монокуляра переходять до віртуального полігону, зберігається їх положення та значення, а кут панорами відповідає положенню перехрестя прицілу на полігоні, після наведення за орієнтиром, дані переносяться до головної сцени. Виконавши завдання наведення учень натискає кнопку перевірки та переглядає свою оцінку, інформаційне вікно з оцінкою наведено на рисунку 3.16.

Час	Оцінка	Похибка
04 хв 10 сек	Незадовільно	Похибка кутоміра 850

Меню

Рисунок 3.16 – Оцінка встановлених значень

Після перегляду результатів користувач має змогу повернутися у головне меню додатку натиснувши кнопку «Меню».

3.3 Програмна реалізація тренажера

Платформою для створення тренажера було обрано «Unity», серед особливостей платформи є інтерактивні вікна, опишемо головне вікно програми, що наведено на рисунку 3.17.

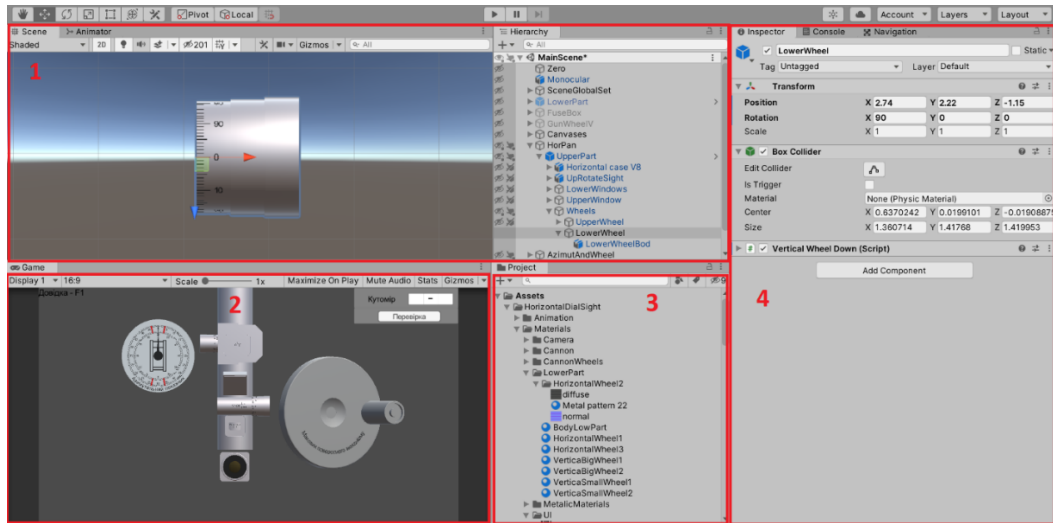


Рисунок 3.17 – Поля головного вікна «Unity»

На першому полі зображуються об'єкти на сцені та їх список, тобто як вони розташовуються у просторі під час проектування додатку, завдяки списку можливо обрати об'єкт. Друге вікно відображає камеру користувача, тобто як під час компіляції буде виглядати додаток, наступне вікно містить список компонентів, що містяться у головній директорії проекту. Останнє четверте вікно складається з елементів для відстежування та зміни параметрів об'єкту, вікно інспектора завдяки ньому можна переглянути які властивості має об'єкт, його координати та додати нові особливості об'єкту, наприклад завдяки `BoxCollider` визначається область в якій користувач може взаємодіяти з маховиком.

Мовою програмування платформи є `C#`, серед головних ключових особливостей мова програмування має:

- Автоматичне збирання сміття – Розробнику немає необхідності контролювати пам'ять для об'єктів, проте також він може викликати методи для звільнення перемінних;
- Система виключень – Завдяки системі виключень розробник може контролювати обробку виключень динамічно, тобто при створенні виключення розробник може його відразу обробити, також розробник має змогу створити свої виключення;

- Нульові типи даних – Якщо змінна не ініціалізована вона має значення за умовчанням, змінна типу `int` буде 0, проте для контролю даних її можна зробити за замовченням `null`, тоді коли розробник забуде задати значення змінної замість математичної операції з додаванням нулю, буде викликано виключення через використання не ініціалізованої змінної;
- Підтримка асинхронні операцій – Зазвичай при роботі програми, усі дії відбуваються поступово одна за одною, проте завдяки асинхронним операціями можливо робити багатозадачні додатки, обчислювати одночасно декілька масивів даних, що значно пришвидшує роботу додатку.

Мовою C# для роботи додатку створено наступні класи, які наведено в таблиці 3.2, лістинг коду класів наведено у Додатку В.

Таблиця 3.2 – Класи додатку та їх призначення

Назва класу	Призначення
Menu	Містить методи які викликаються при натисканні пунктів меню, перевіряє введені дані завдання наведення.
SceneSwitch	Має методи для переходу між сценами додатку.
SceneDataPasser	Статичний клас, який зберігає дані завдання наведення при переході між сценами додатку.

Продовження таблиці 3.2 – Класи додатку та їх призначення

Назва класу	Призначення
CameraControl	Клас який містить методи для керування камери: руху, обертання та переміщенню її навколо моделі. Повертає камеру у початкове положення.
InfPanel	Виводить дані про маховики у текст на екрані.
InputFields	Порівнює встановлені дані отримані з класу InfPanel та дані завдання наведення з класу SceneDataPasser, виводить оцінку та похибку на екран.
Wheel	Абстрактний клас містить абстрактні методи які мають бути реалізовані в маховиках, методи розрахунку кута маховика, кольор для відображення маховика при взаємодії з ним користувача.
VerticalWheelDown	Відповідає за маховик кутоміра в головній сцені, містить методи для взаємодії з користувачем та розрахунку встановлених даних

Продовження таблиці 3.2 – Класи додатку та їх призначення

Назва класу	Призначення
VerticalWheelDownCanvas	Керує маховиком кутоміра у віртуальному полігоні, містить методи для взаємодії з користувачем, а саме обертання панорами на полігоні
VerticalWheelDownUp	Відповідає за маховик відбивача в головній сцені, оброблює натискання користувача та розраховує встановлені дані
VerticalWheelDownUpCanvas	Керує маховиком відбивача у віртуальному полігоні, для взаємодії з користувачем, а саме обертання панорами на полігоні.
OnDragCannonWheelH	Відповідає за маховик поворотного механізму башти, містить методи для взаємодії з користувачем та розрахунку встановлених даних та методи повертання башти артилерійської установки на віртуальному полігоні

3.4 Приклади використання та тестування додатку

При старті тренажеру з'явиться головне меню додатку, вигляд меню наведено на рисунку 3.18.

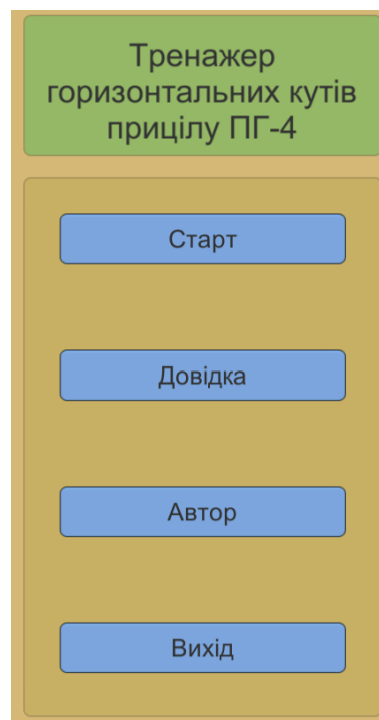


Рисунок 3.18 – Головне меню тренажеру

Головне меню складається з кнопки старт, списку авторів та довідки. Користувач натиснувши кнопку старт обирає спосіб генерації завдання, випадково або вводить свої значення (рис.3.19).

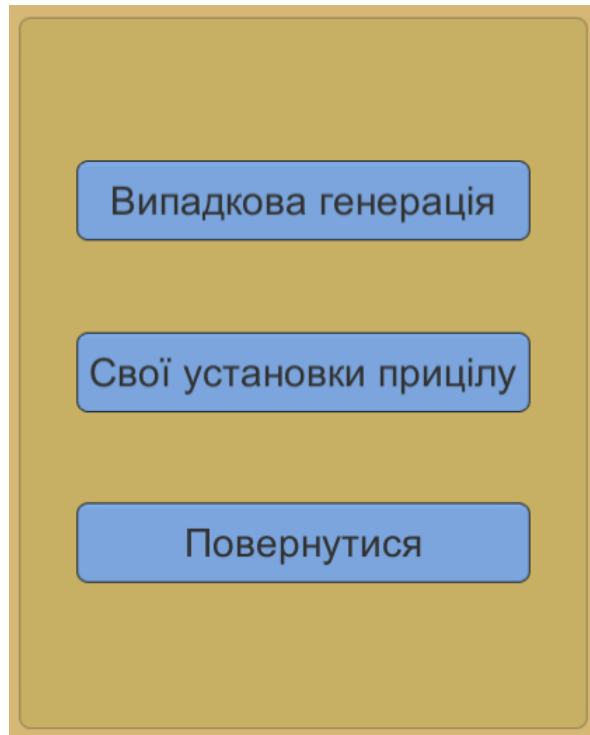


Рисунок 3.19 – Способи створення завдання наведення

Було згенеровано завдання наведення 44-48, перші дві цифри це позначки грубої шкали кутоміру, наступні це значення точної шкали кутоміру. На початку завдання всі значення виставлені за замовченням, значення кутоміру 45-00, механізму відбивача панорами 0, башта знаходиться у похідному положенні, тобто паралельно шасі артилерійської установки (рис.3.20).

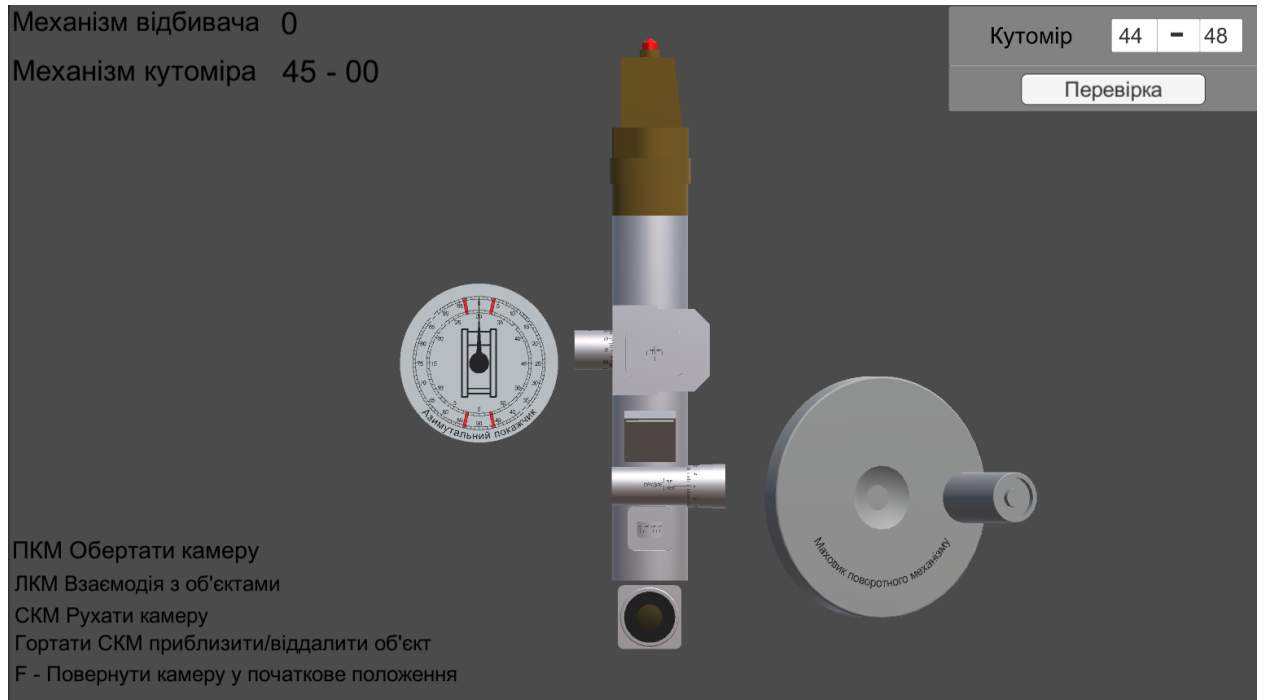


Рисунок 3.20 – Початкові значення

Встановимо значення кутоміра обертаючи маховик кутоміра вгору, при цьому груба шкала кутоміра буде обертатися праворуч, значення грубої шкали 44 та точної шкали 48 встановлено на рисунку 3.21.

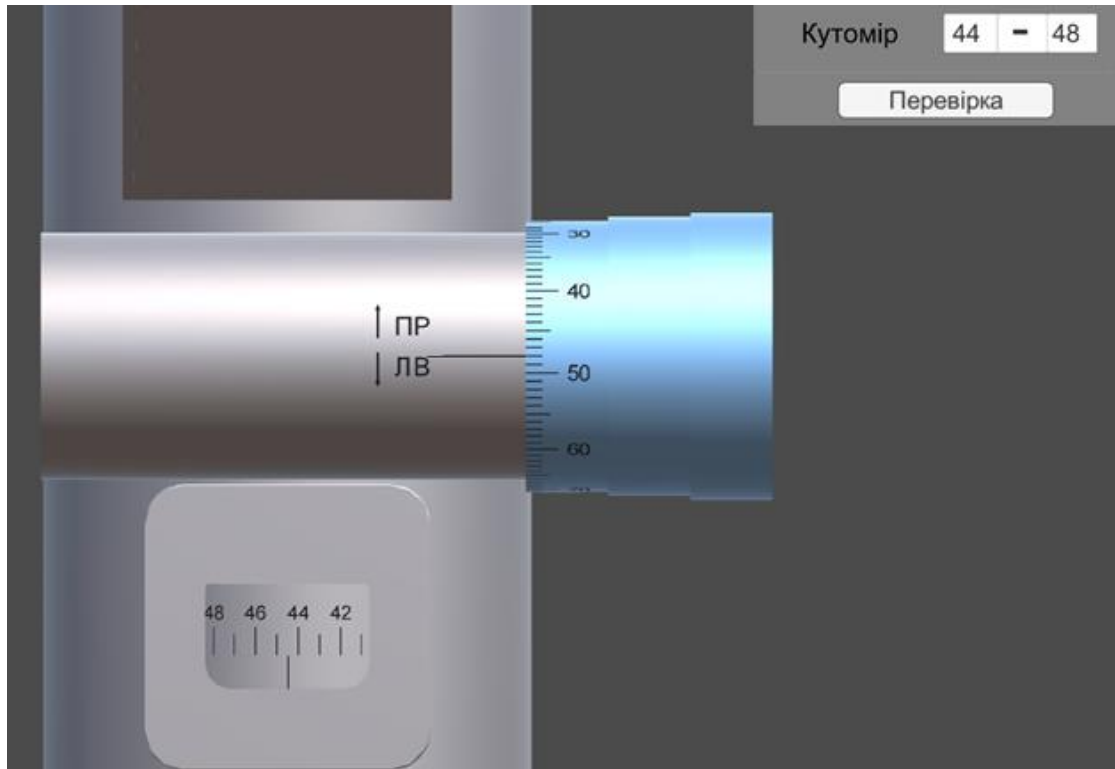


Рисунок 3.21 – Значення кутоміру 44-48

Після встановлення значення користувач взаємодіючі з монокуляром переходить до віртуального полігону, після того як ми встановили значення кутоміру перехрестя прицілу змістилося праворуч відносно до орієнтиру (рис.3.22).



Рисунок 3.22 – Віртуальний полігон

Наступним етапом необхідно встановити перехрестя на орієнтир, обертаючи вже не панораму прицілу, а башту самохідної артилерійської установки використовуючи маховик поворотного механізму (рис.3.23).

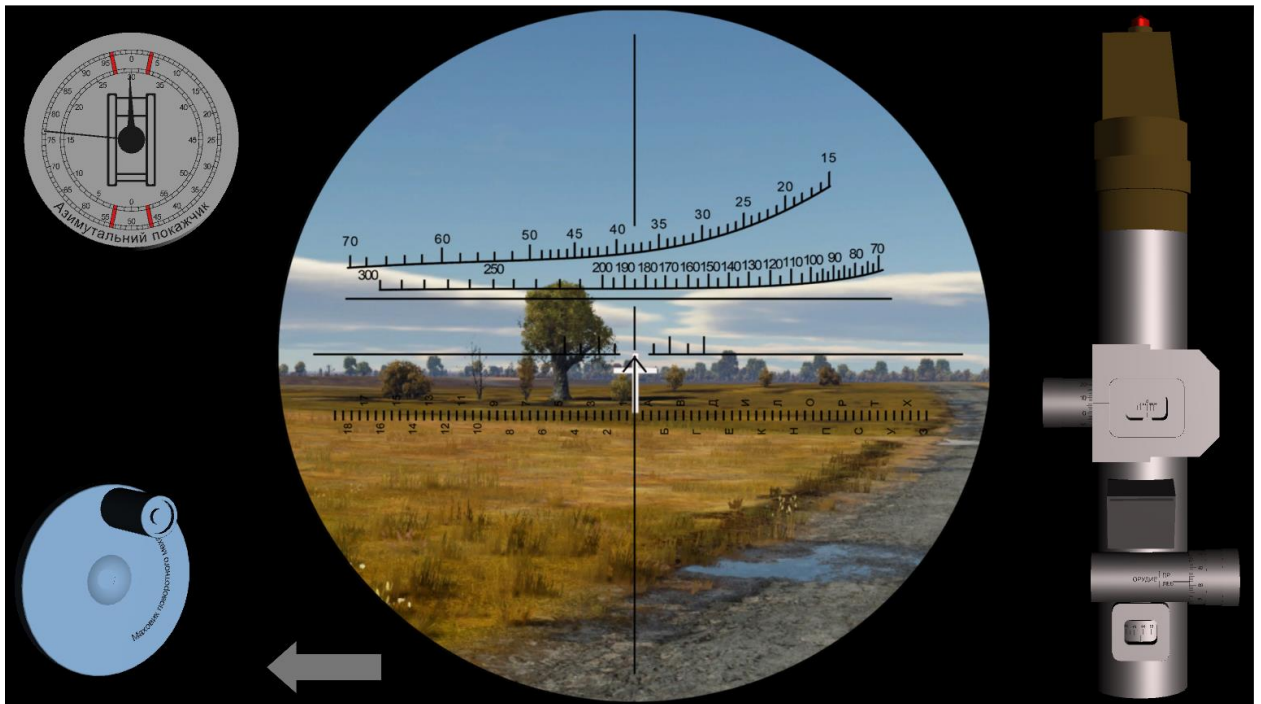


Рисунок 3.23 – Обертання башти на орієнтир.

Встановивши значення учень натискає кнопку «Перевірка» та отримує оцінку, на рисунку 3.24 наведено витрачений час оцінка та похибка.



Рисунок 3.24 – Оцінка встановлених значень

При розробці тренажера були складені тест кейси методом еквівалентного розбиття [15], результати тестування наведені в таблиці 3.3

Таблиця 3.3 – Тест кейси додатку та їх результат

№	Вид класу	Клас еквівалентності	Очікуваний результат	Фактичний результат	0/1
1	Неправильний	Обертання маховика кутоміра $x > 60000$	При спробі вийти за максимальне допустиме значення кутоміру, буде встановлено мінімальне значення	При встановлені значення кутоміру 60-00 (60000 у точній шкалі) наступний оберт встановить значення 0	1
2	Неправильний	Обертання маховика кутоміра $x < 0$	При спробі вийти за мінімальне допустиме значення кутоміру, буде встановлено максимальне значення	При встановлені значення кутоміру 0-00 наступний оберт встановить значення 60000	1
3	Неправильний	Обертання маховика механізму відбивача $y > 300$	Значення механізму відбивача не може бути встановлене більше 300 поділок шкали	При встановлені значення відбивача 300, маховик далі не обертається, тільки у протилежну сторону	1

Продовження таблиці 3.3 – Тест кейси додатку та їх результат

№	Вид класу	Клас еквівалентності	Очікуваний результат	Фактичний результат	0/1
4	Неправильний	Обертання маховика механізму відбивача у <-300	Значення механізму відбивача не може бути встановлене менше -300 поділок шкали	При встановленні значення відбивача -300, маховик далі не обертається, тільки у протилежну сторону далі не обертається	1
5	Правильний	Обертати маховик кутоміра вгору або вниз	Перехрестя панорами рухається праворуч при оберту вгору та ліворуч при оберту вниз	При обертанні маховика вниз перехрестя рухається ліворуч при оберту вгору - праворуч	1
6	Правильний	Взаємозв'язок маховика обертального механізму башти з панорамою	При обертанні маховика перехрестя панорами рухається, проте значення кутоміра залишаються незмінними	Башта обертається разом панорамою, тому перехрестя рухається, а значення кутоміра не змінюється	1
7	Правильний	Взаємодія маховиком механізму відбивача з	При обертанні маховика перехрестя прицілу рухається вгору або вниз залежно від напрямку руху маховика	Перехрестя прицілу змінює положення вертикально, залежно від обертання маховика механізму відбивача	1

Продовження таблиці 3.3 – Тест кейси додатку та їх результат

№	Вид класу	Клас еквівалентності	Очікуваний результат	Фактичний результат	0/1
8	Правильний	Встановити значення кутоміру за 65 секунд або менше без похибки кутоміра. $t \in [0, 65]$	Отримано оцінку «Відмінно!»	Отримано оцінку «Відмінно!»	1
9	Правильний	Встановити значення кутоміру за 66-75 секунд, без похибки кутоміра $t \in [66, 75]$	Отримано оцінку «Добре»	Отримано оцінку «Добре»	1
10	Правильний	Встановити значення кутоміру за 76-85 секунд, без похибки кутоміра $t \in [76, 85]$	Отримано оцінку «Задовільно»	Отримано оцінку «Задовільно»	1
11	Правильний	Встановити значення кутоміру за 86 секунд або більше, без похибки кутоміра $t \in [86, +\infty]$	Отримано оцінку «Незадовільно, занадто довго»	Отримано оцінку «Незадовільно, занадто довго»	1
12	Неправильний	Зробити завдання за 86 секунд або менше з похибкою кутоміра $z > 2$	Наведена похибка кутоміра та оцінка «Незадовільно»	Похибка кутоміра z , оцінка «Незадовільно»	1

Тренажер неодноразово протестовано викладачами та учнями кафедри (рис.3.25).

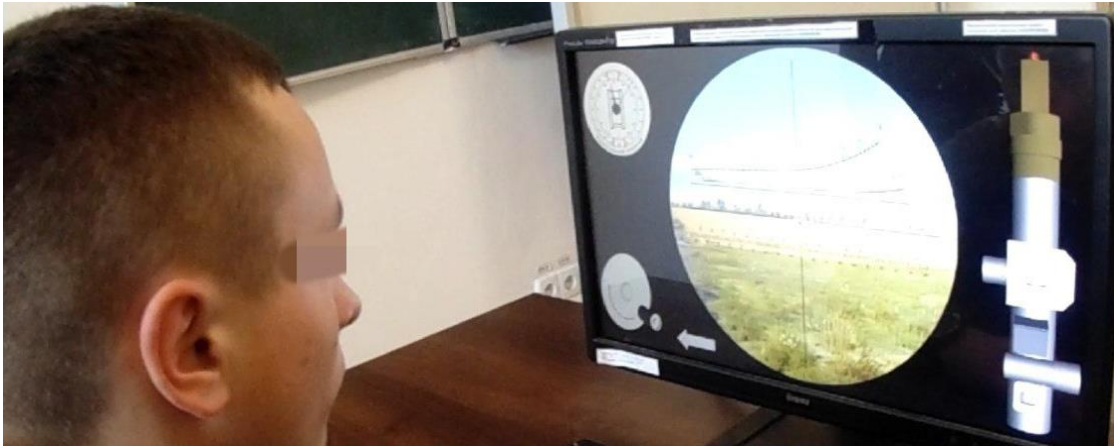


Рисунок 3.25 – Використання тренажеру на базі військової кафедри

Викладачі та учні кафедри були задоволені якістю моделі приціла, зручності використання додатку та високим рівнем відповідності роботи з прицілом згідно з оригіналом.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра досліджено предметну область військових віртуальних тренажерів та методи віртуального навчання, віртуальні тренажери активно впроваджуються, що говорить про необхідність та ефективність віртуальних тренажерів.

Були виконанні задачі реалізації тренажера:

- Досліджено предметну область та аналоги додатку;
- Спроектовано інформаційну систему проекту та план робіт реалізації додатку;
- Створено віртуальну модель прицілу «ПГ-4»;
- Реалізовано логіку взаємодії з віртуальною моделлю;
- Розроблено інтерфейс додатку;
- Додаток протестовано.

Досліджено найближчі аналоги тренажера, щоб використати переваги аналогів у проекті. Визначено мету та поставлено задачу тренажеру, згідно з цим обрано інструменти створення додатку, програми для моделювання «SOLIDWORKS» та платформу для розробки інтерактивних додатків «Unity» мовою C#, завдяки обраним інструментам готовий додаток легко інтегрувати у навчальний процес та модифікувати.

Побудована математична модель взаємодії механізмів прицілу, а саме знайдені коефіцієнти відношення маховика кутоміру до грубої шкали та формулу розрахунку значень кутоміра. Розроблена діаграма Use Case, за допомогою якої зображено взаємодія користувача з додатком. Спроектовано діаграму IDEF0 в якій визначено зв'язки між роботами виконання завдання наведення прицілу. На основі діаграми IDEF0 створено діаграму декомпозиції головного блоку IDEF0, тобто деталізовану діаграму IDEF0, в якій наведено алгоритм використання віртуального тренажера.

Сформоване технічне завдання, що містить вимоги до проекту та розроблений план виконання робіт для створення додатку.

Спроектовано віртуальну модель горизонтальних приладів наведення прицілу ПГ-4, завдяки програмі для моделювання «SOLIDWOKS».

Модель інтегровано в платформу для створення додатків «Unity» та пофарбовано, розроблено дизайн інтерфейсу взаємодії з користувачем, використовуючи мову програмування C# реалізовано логіку взаємодії компонентів приціла.

Під час формування вимог та розробки були сформовані тест кейси, додаток пройшов усі випробування, додатково був протестований учнями військової кафедри СумДУ.

Результатом став тренажер горизонтальних пристроїв наведення прицілу ПГ-4, тренажер введено у процес навчання офіцерів-артилеристів кафедри військової підготовки.

Проект апробовано на конференції «ІМА 2021» та на міжнародній науково-практичній конференції Одеської військової академії «Спільні дії військових формувань і правоохоронних органів держави», впроваджено два наукових твори, здобутки проекту наведені у додатку Г.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інформаційні технології у підготовці військових спеціалістів РВ і А в особливий період [Електронний ресурс] – Режим доступу до ресурсу: https://essuir.sumdu.edu.ua/bitstream-download/123456789/79179/1/Derevianchuk_1554.pdf.
2. Бородин С. В. Особливості підготовки екіпажів бронетранспортерів бтр4е з використанням сучасних тренажерних комплексів / С. В. Бородин, Ю. В. Самсонов, О. В. Корнієнко. // Системи обробки інформації. – 2017. – С. 174–176.
3. Українське підприємство розробило новий тренажер льотчика вертольота Мі-24П [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mil.gov.ua/news/2019/07/13/ukrainske-pidpriemstvo-rozrobilo-novij-trenazher-lotchika-vertolota-mi-24p/>.
4. Призначення та будова панорами ПГ-4. Дія механізмів. Шкали панорами та користування ними. Коліматор К-1 [Електронний ресурс] – Режим доступу до ресурсу: http://bookwu.net/book_artilerijske-ozbroyennya-ta-pripasi_1038/55_6.7.3-priznachennya-ta-budova-panorami-pg-4.-diya-mehanizmiv.-shkali-panorami-ta-koristuvannya-nimi.-kolimator-k-1.
5. SOLIDWORKS [Електронний ресурс] – Режим доступу до ресурсу: <https://www.solidworks.com/>.
6. Best Game Engines of 2021 [Електронний ресурс] – Режим доступу до ресурсу: <https://gamedevacademy.org/best-game-engines/>.
7. Unity : Developing Your First Game with Unity and C# [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/archive/msdn-magazine/2014/august/unity-developing-your-first-game-with-unity-and-csharp>.

8. What is Use Case Diagram? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
9. IDEF0 - Part 1 (understanding it) [Електронний ресурс] – Режим доступу до ресурсу: http://www.syque.com/quality_tools/tools/Tools19.htm.
10. Work Breakdown Structure (WBS) [Електронний ресурс] – Режим доступу до ресурсу: <https://www.projectmanager.com/work-breakdown-structure..>
11. What is a Gantt Chart? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.gantt.com/>.
12. Working in Unity [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.unity3d.com/Manual/UnityOverview.html>.
13. A tour of the C# language [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>.
14. Способи визначення дальності стрільби і застосування формули тисячної [Електронний ресурс] – Режим доступу до ресурсу: <https://works.doklad.ru/view/M30fdcU43vA.html>.
15. Особливості вимог програмного забезпечення. Методи тестування. [Електронний ресурс] – Режим доступу до ресурсу: <https://qlearning.com.ua/theory/lectures/material/requirements-testing-methods-equivalence/>.

Додаток А

Технічне завдання

1 ПРИЗНАЧЕННЯ ТА МЕТА СТВОРЕННЯ ДОДАТКУ

Призначення віртуального тренажеру це навчання на практиці майбутнього офіцера артилериста, додаток має відповідати прицілу ПГ-4, а саме його механізмам горизонтального наведення, генерувати завдання для наведення прицілу та перевіряти встановлені дані. Завдяки тренажеру студенти зможуть закріпити знання на практиці, без необхідності використовувати фізичний тренажер.

1.1 Мета створення віртуального тренажера

Метою даного проєкту є створення додатку симуляції горизонтального наведення прицілу ПГ-4 на базі операційної системи Windows. Застосунок буде призначений для учнів кафедри артилерійської підготовки, сам приціл ПГ-4 який обладнують самохідну артилерійську установку «2С3 має багато спільних рис з іншими прицілами, що подалі допоможе у вивченні інших типів прицілів студенту.

1.2 Цільова аудиторія

До цільової аудиторії віртуального тренажера відносяться усі студенти військової артилерійської підготовки кафедри СумДУ.

2. ВИМОГИ ДО ТРЕНАЖЕРУ

2.1 Вимоги до структури й функціонування додатку

Програмний продукт має бути сумісним з операційною системою Windows, для роботи додатку необхідна буде операційна система Windows 7 або Windows 10.

Даний програмний продукт має охоплювати найбільш велику кількість пристроїв тому мінімальні технічні характеристики має бути невибагливими. Мінімальні технічні характеристики пристроя для запуску додатку:

- Центральний процесор з токової частотою від 2.6 ГГц та двома ядрами.
- Відеопам'ять об'ємом 2 ГБ.
- Оперативна пам'ять об'ємом 4 ГБ.
- Місце на жорсткому диску 2 ГБ (разом з програмним забезпеченням).

2.2 Вимоги до персоналу

Персонал має бути ознайомлений з базовими знаннями з користуванням ОС Windows 7 або Windows 10

2.3 Вимоги до збереження інформації

Додаток не генерує ніяких даних на жорсткому диску, усі дані ініціалізуються та зберігаються в оперативній пам'яті комп'ютера під час роботи додатку та звільняються під час закриття додатку.

2.4 Вимоги до розмежування доступу

Додаток створений для студентів артилерійської підготовки кафедри СумДУ, додаток передається викладачу кафедри, який і виконує безкоштовне поширення додатку на базі військової кафедри в навчальних цілях.

2.5 Структура додатку

Віртуальний тренажер складається з меню, головної сцени та допоміжної сцени, вони виконують такі функції:

Меню – виконує функцію навігації, містить пункти меню: «Старт», «Довідка», «Вихід».

Головна сцена – містить маховики для горизонтальної наводки прицілу.

Допоміжна сцена – полігон в якому відбувається наведення на ціль.

Довідка – містить інструкції з користування додатку.

2.5.1 Навігація

Навігація відбувається за допомогою головного меню яке завантажуються при запуску додатку, діаграму переходу з пунктів меню зображено на рисунку А 2.1:

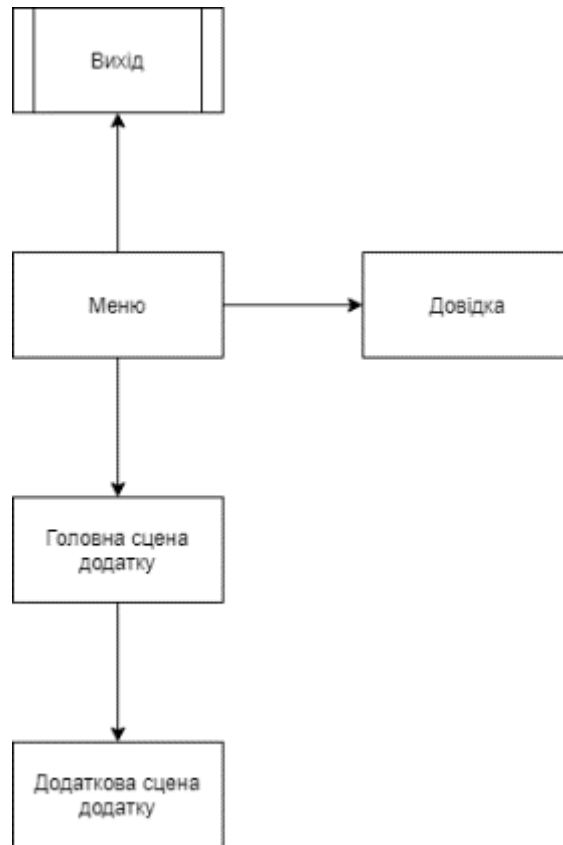


Рисунок А 2.1– Меню додатку

2.5.2 Наповнення додатку

Додаток буде містити модель горизонтальних пристроїв наведення прицілу ПГ-4, інформативні вікна які демонструють встановленні значення прицілу та віртуальний полігон з орієнтиром за допомогою якого і відбувається наведення.

2.5.3 Дизайн та структура додатку

Додаток має відповідати прицілу «ПГ-4», мати зручне управління встановлення прицілу та інформативні вікна які допоможуть користувачу перевірити правильність встановлених значень. Нижче наведено макет головного меню на рисунку А 2.2:



Рисунок А 2.2 – Макет головного меню додатку

Для встановлення горизонтальних кутів прицілювання буде використано модель горизонтальних пристроїв наведення прицілу ПГ-4 макет пристроїв наведено на рисунку А 2.3

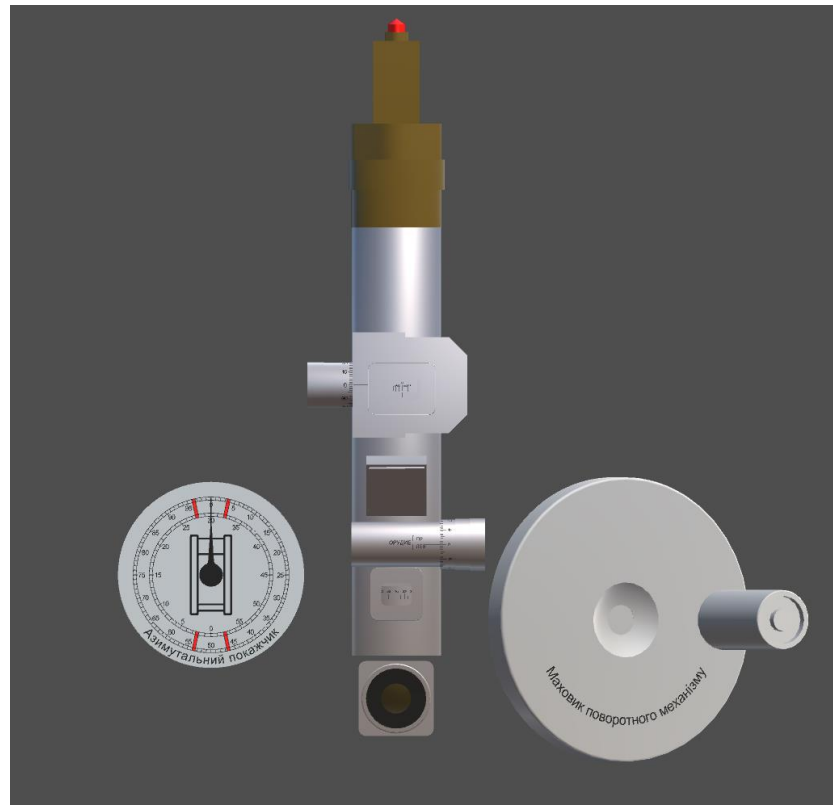


Рисунок А 2.3 – Макет пристроїв наведення додатку

Для точної стрільби з закритих вогневих позицій необхідно використовувати орієнтир який буде встановлено на віртуальному полігоні, макет зображено на рисунку А 2.4.

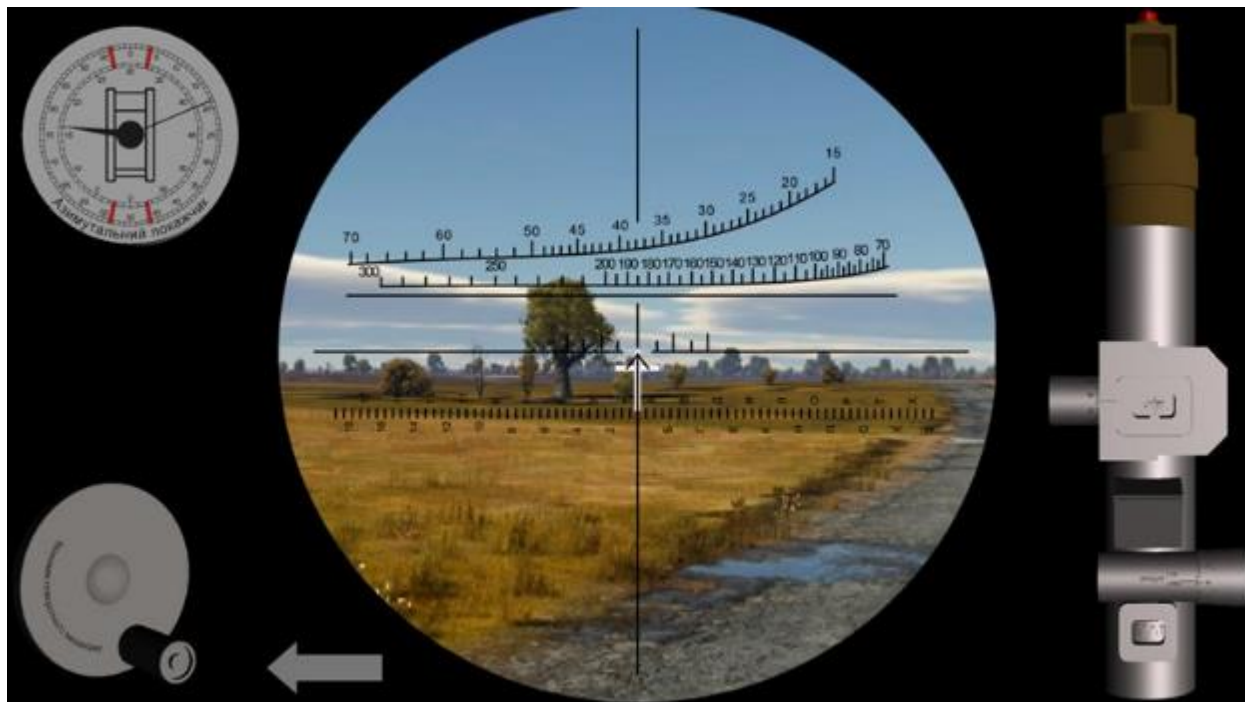


Рисунок А 2.4 – Макет віртуального полігону з орієнтиром.

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача, складаються на основі мети, цільової аудиторії та структури додатку які зображено в таблиці А 2.1.

Таблиця А 2.1 – Перелік вимог до віртуального тренажера прицілу ПГ-4

ID	Потреби користувача	Джерело
PN-01	Головне меню	Клієнт
PN-02	Інформаційна довідка в головному меню	Клієнт
PN-04	Інформація про час прицілювання та похибку встановлених значень	Клієнт

Продовження таблиці А 2.1 – Перелік вимог до віртуального тренажера прицілу

ID	Потреби користувача	Джерело
PN-05	Генерація завдання для встановлення кутів має відбуватися випадково або за значеннями користувача, залежно за його бажаннями	Клієнт
PN-06	Встановлення значень маховиків прицілу за допомогою миші та клавіатури	Клієнт
PN-07	Альтернативні методи встановлення значень маховиків прицілу за допомогою клавіатури та миші клавіатури	Клієнт
PN-08	Відповідні елементи моделі прицілу мають рухатися під час обертання маховиків як його реальна копія.	Клієнт
PN-09	Віртуальний полігон з орієнтиром для практики наведення з використанням орієнтиру	Клієнт

2.3.2 Функціональні вимоги

Проаналізувавши потреби користувача, були визначені такі функціональні вимоги:

- Головне меню з довідковою інформацією;
- Інформативні вікна встановлених значень та оцінка часу та точності встановлених даних;
- Маховики прицілу та його шкали мають бути інтерактивними та взаємодіяти один з одним;
- Віртуальний полігон з встановленим орієнтиром.

2.3.3 Системні вимоги

Системні вимоги були сформовані на побажаннях учнів та викладачів кафедри, системні вимоги наведено нижче в таблиці А 2.2.

Таблиця А 2.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
CR-01	Довідкова інформація	Має бути	Інструкція для користувача має ознайомити користувача з алгоритмом наведення прицілу.
CR-02	Моніторинг встановлених значень	Має бути	Дає оцінку користувачу базуючись з похибки встановлених значень та часу наведення.
CR-03	Взаємодія користувача з маховиками	Має бути	Інтерактивна взаємодія користувача з маховиками прицілу за допомогою миші або клавіатури.

Продовження таблиці А 2.2 – Системні вимоги

CR-04	Взаємозв'язок між маховиками та їх шкалами	Має бути	Приближує віртуальну модель до реальної, навчаючи користувача орієнтуватися за шкалами прицілу
CR-05	Віртуальний полігон	Середній	Віртуальний полігон завдяки якому користувач робить розрахунки координат приціла за орієнтиром
CR-06	Альтернативні способи введення значень	Низький	Завдяки альтернативним способам введення даних користувач зможе обирати швидкість та точність наведення прицілу.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Віртуальний тренажер повинен охоплювати найбільшу кількість комп'ютерів, тому необхідно зробити акцент не на графічному дизайні, а на функціональності та

оптимізації системних вимог до комп'ютера. Платформа Unity 3D має можливість завантажувати інструменти створеними користувачами, що значно збільшує кількість методів реалізації, що дає велику гнучкість для створення додатку, також порівняно з Godot платформа спеціалізується на 3D графіці, проте порівняно з Unreal Engine графіка має більш низьку якість, проте гнучкість та можливість реалізації певних механізмів додатку обмежена.

2.4.2 Вимоги до лінгвістичного забезпечення

Додаток створюється для учнів військової кафедри артилерійської підготовки СумДУ тому має бути локалізований українською мовою.

2.4.3 Вимоги до програмного забезпечення

Для роботи додатку необхідна операційна система Windows 7 або вище.

3 СКЛАД ТА ЗМІСТ РОБІТ ЗІ СТВОРЕННЯ ВІРТУАЛЬНОГО ДОДАТКУ

Докладний опис етапів роботи зі створення віртуального додатку наведено в таблиці А 3.1.

Таблиця А 3.1 – Етапи створення віртуального додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Постановка цілей необхідних для досягнення певного результату	6 днів
2	Складання технічного завдання	12 днів
3	Підготовка прототипу	15 днів
4	Створення макету дизайну	8 днів
5	Перевірка працездатності віртуального тренажеру	12 днів
6	Завершення роботи	1 днів
	Загальна тривалість робіт	54 днів

4 ВИМОГИ ДО СКЛАДУ ТА ЗМІСТУ ДОДАТКУ

Для введення додатку в експлуатацію необхідно надати доступ до додатку викладачам кафедри, які подалі будуть виконувати навчання за допомогою додатку на базі військової кафедри.

Щоб запустити додаток необхідна операційна система Windows 7 або вище. Для старту програми необхідно запустити файл «Віртуальний_Тренажер_прицілу_ПГ_4» з розширенням «exe», що знаходиться в головній папці програми.

Додаток Б

Планування робіт

1 РОЗРОБЛЕННЯ КОНЦЕПЦІЇ ПРОЕКТУ

1. Ідентифікація ідеї проєкту

Віртуальні тренажери полегшують отримання практичних навичок. Функціональність та можливості реалізації віртуальних тренажерів безмежна.

Метою даного проєкту є створення додатку на базі операційної системи Windows. Цей застосунок буде призначений для прицілу ПГ-4, який обладнують самохідну артилерійську установку «2С3» механізм приціла має багато спільних рис з іншими прицілами, що подалі допоможе у вивченні інших типів прицілів студенту.

1.2 Деталізація мети методом SMART

Згідно з метою проєкту було побудовано деталізацію цілі додатку методом SMART (табл. Б 1.1.)

Таблиця Б 1.1 – Деталізація мети методом SMART

S.M.A.R.T	Коментар
Specific (конкретна)	Розробити додаток на базі Windows ОС для практичних занять установки горизонтальних кутів прицілювання для прицілу ПГ-4

Продовження таблиці Б 1.1 – Деталізація мети методом SMART

S.M.A.R.T	Коментар
Measurable (вимірювання)	Оскільки даний проєкт не є комерційним, то результатом роботи є оцінка замовника.
Achievable (досяжна, узгоджена)	Ціль даного проєкту вважається досяжною, оскільки розробник володіє необхідними навичками у створенні додатків мовою C# на платформі Unity. Мета була узгоджена з вимогами та потребами замовника.
Relevant (реалістична)	Для реалізації продукту проєкту є всі необхідні програмні засоби (середовище розробки Visual Studio та платформа розробки додатків Unity), доступ до мережі Інтернет. Розробник досить кваліфікований для виконання поставлених задач.
Time-framed (обмежена в часі)	Додаток розроблюється з обмеженням у часі на основі сформованого календарного плану.

2 ОПИС ІТ-ПРОЄКТУ НА ФАЗІ РОЗРОБКИ

2.1 Планування змісту структури робіт ІТ-проєкту

За допомогою діаграми WBS сплановано структуру робіт необхідних для створення додатку. Діаграмою зручно відображується ієрархічна декомпозиція робіт. Діаграму наведено на рисунку Б 2.1

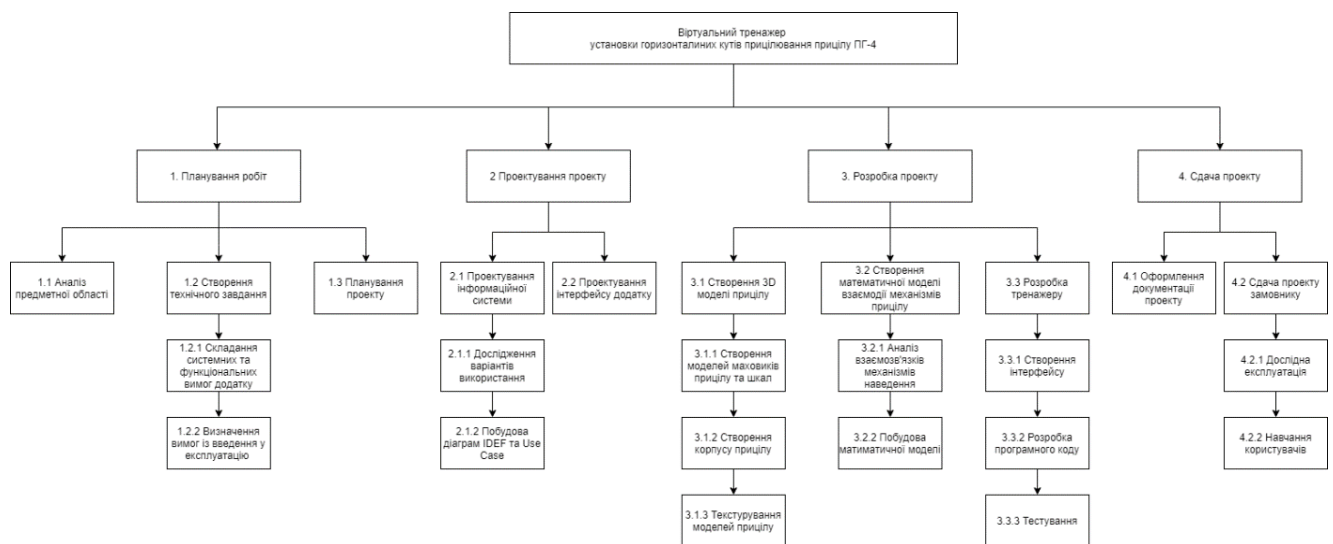


Рисунок Б 2.1 – WBS-структура проєкту

2.2 Планування структури організації

Виконавці проекту:

- Менеджер проекту (Керівник дипломної роботи);
- Виконавець проекту (Студент).

На основі даної структури була складена таблиця виконавців робіт реалізації проекту Б 2.1.

Таблиця Б 2.1 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник -тестувальник	Кравченко Д.О.	Створює додаток. Перевіряє функціональні вимоги проекту.
Менеджер проекту (дипломної роботи)	Кузнецов Е.Г.	Відповідає за виконання термінів, підтримує розробника – тестувальника проекту в питаннях виконання продукту проекту.

Використовуючи таблицю Б 2.1 побудуємо OBS структуру виконавців робіт проекту (рис. Б 2.2)

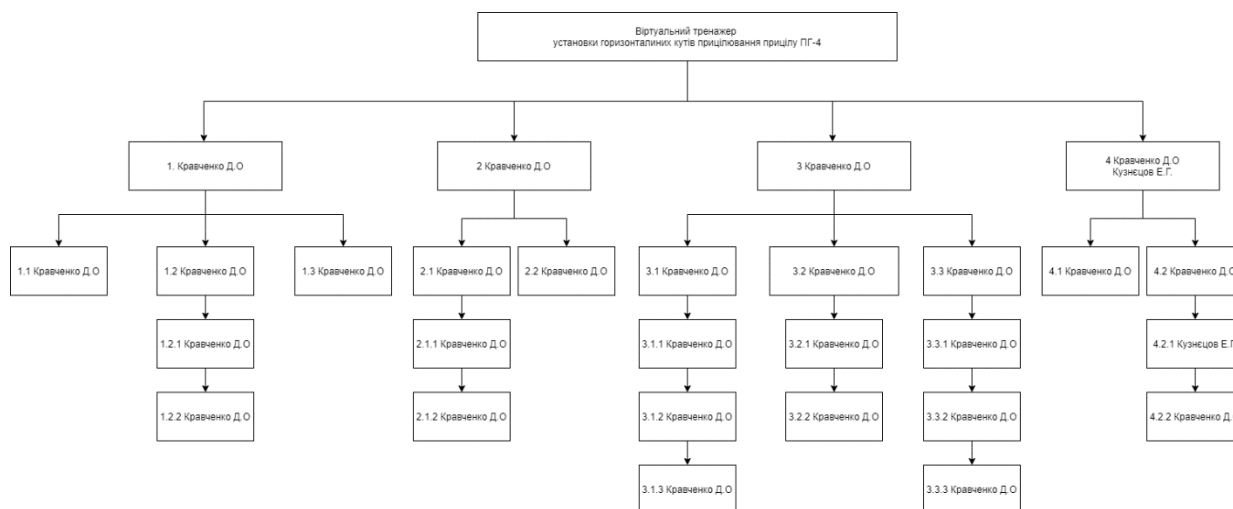


Рисунок Б 2.2 – OBS-структура проекту

2.3 Побудова діаграми Ганта

Наступним етапом спроектуємо календарний план дипломного проекту. Для цього скористаємося діаграмою Ганта діаграмою легко зобразити графіки часу та взаємозв'язок між задачами проекту. Побудуємо діаграму Ганта, врахувавши вихідні дні та встановивши тривалість робочого дня 8 годин, діаграму зображено на рисунку Б.2.3.

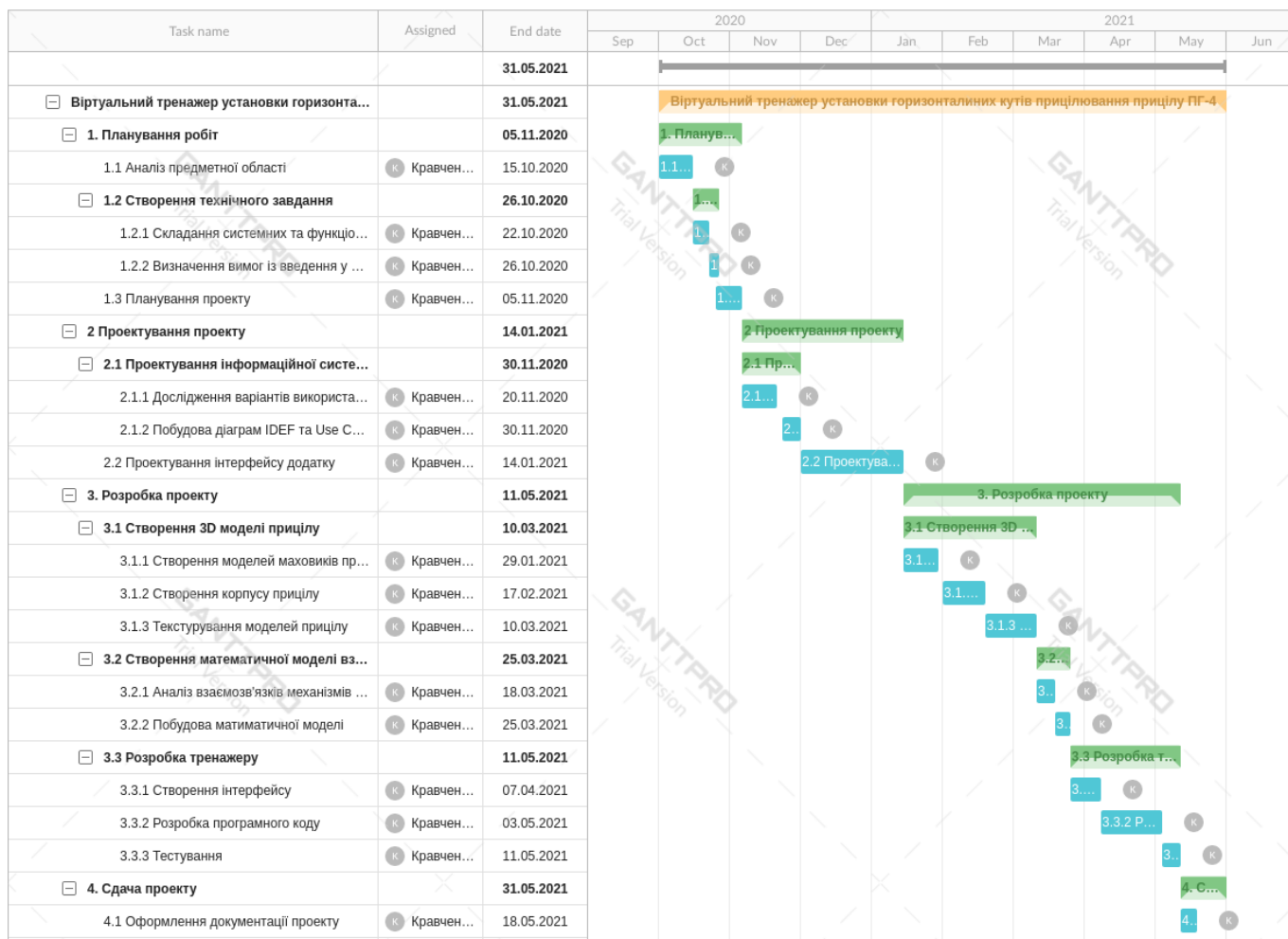


Рисунок Б 2.3 – Діаграма Ганта проекту

2.3 Аналіз ризиків

При розробці програмного забезпечення, можливі певні ризики у ході реалізації проєкту. 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 проєкту таблицю з оцінювання величин можливих втрат наведено в таблиці Б 2.2 .

Таблиця Б 2.2 – Оцінювання величини можливих втрат

Величина можливих втрат, бали		Стан виконання етапів проєкту (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 події та оцінювання величини можливих втрат (табл. Б 2.3).

Таблиця Б 2.3 – Класифікація ризиків

№	Назва ризику	Ймовірність	Вплив ризику
RC_1	Недостатня детальність ТЗ	6	5
RC_2	Порушення дедлайнів для певного етапу розробки проєкту	2	2
RC_3	Нестабільна робота ПЗ	5	5
RC_4	Нестабільна робота АЗ	1	2

Продовження таблиці Б 2.3 – Класифікація ризиків

№	Назва ризику	Ймовірність	Вплив ризику
RC_5	Сезонні хвороби учасників проекту	1	1
RC_6	Недостатньо якісне виконання тестування	4	6

За таблицею Б 2.3 було сформовано матрицю ризиків зі встановленими значеннями ймовірності та величини втрат на рисунку Б 2.4.

Ймовірність	1	RC_5	RC_4				
	2		RC_2				
	3						
	4						RC_6
	5					RC_3	
	6					RC_1	
		1	2	3	4	5	6
	Вплив ризику						

Рисунок Б 2.4 – Матриця ризиків

Після оцінювання ризиків було складено план реагування, який наведено в таблиці Б 2.3.

Таблиця Б 2.3 – Реагування на ризики

ID	Ризики проекту	Реакція на ризик
RC_1	Недостатня детальність ТЗ	Постійне обговорення з керівником ПЗ проекту, після кожного етапу

Продовження таблиці Б 2.3 – Реагування на ризики

ID	Ризики проекту	Реакція на ризик
RC_2	Порушення дедлайнів для певного етапу розробки проекту	На етапі планування слід врахувати запас часу на обставини, що можуть викликати під час реалізації проекту.
RC_3	Нестабільна робота ПЗ	Необхідна локалізація причини нестабільної роботи ПЗ, та подальше її вирішення.
RC_4	Нестабільна робота АЗ	Провести діагностику системи у разі необхідності відправити пристрій на ремонт, щоб не втрачати час продовжити роботу на запасному ПК.
RC_5	Сезонні хвороби учасників проекту	Забезпечити запас часу на основних етапах проекту, який зможе компенсувати витрати часу на період хвороби учасників.
RC_6	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

Файл «Menu.cs»:

```
using UnityEngine;
using UnityEngine.UI;
using Assets.Resources;

public class Menu : MonoBehaviour
{
    public Transform Aim;
    public Camera mainC;

    public GameObject canvasPlaySelect;
    public GameObject canvasMenu;
    public GameObject canvasInputPanel;
    public GameObject WarningPanel;
    public GameObject GuidePanelObj;
    public GameObject AutorsPanel;
    public GameObject ProjName;

    public InputField inputFieldMenuAim;
    public InputField inputFieldMenuLev1;
    public InputField inputFieldMenuLev2;
    public InputField inputFieldMenuLev1H;
    public InputField inputFieldMenuLev2H;
    public Text warningText;

    void Start()
    {
        canvasPlaySelect.SetActive(false);
        canvasInputPanel.SetActive(false);
        WarningPanel.SetActive(false);
        GuidePanelObj.SetActive(false);
    }

    public void PlaySelect()
    {
        canvasPlaySelect.SetActive(true);
        canvasMenu.SetActive(false);
        ProjName.SetActive(false);
    }

    public void MenuS()
    {
        canvasMenu.SetActive(true);
        canvasPlaySelect.SetActive(false);
        ProjName.SetActive(true);
    }

    public void PlayRand()
    {
        SceneDataPasser.LvlOneH = (int)Random.Range(44, 46);
        SceneDataPasser.LvlTwoH = (int)Random.Range(0, 99);
    }
}
```

```

SceneDataPasser.TimeInMenu = Time.realtimeSinceStartup;
SceneSwitch.loadScene(SceneSwitch.Scene.MainScene);
}

public void PlayInput()
{
    int lvlOneH, lvlTwoH;
    bool canStart = true;

    // HorLevel.
    if (!int.TryParse(inputFieldMenuLev1H.text, out lvlOneH) || lvlOneH < 0 || lvlOneH > 60)
    {
        WarningPanel.SetActive(true);
        warningText.text = UIText.LVL_ONE_LIMIT_WARNING;
        canStart = false;
    }
    if (lvlOneH >= 0 && lvlOneH < 10 && !inputFieldMenuLev1H.text.StartsWith("0"))
    {
        WarningPanel.SetActive(true);
        warningText.text = UIText.LVL_ONE_FORMAT_WARNING;
        canStart = false;
    }

    if (!int.TryParse(inputFieldMenuLev2H.text, out lvlTwoH) || lvlTwoH < 0 || lvlTwoH > 99)
    {
        warningText.text = UIText.LVL_TWO_LIMIT_WARNING;
        WarningPanel.SetActive(true);
        canStart = false;
    }
    if (lvlTwoH >= 0 && lvlTwoH < 10 && !inputFieldMenuLev2H.text.StartsWith("0"))
    {
        WarningPanel.SetActive(true);
        warningText.text = UIText.LVL_TWO_FORMAT_WARNING;
        canStart = false;
    }

    if (lvlOneH == 60 && lvlTwoH > 0)
    {
        warningText.text = UIText.LVL_ONE_TWO_FORMAT_WARNING;
        WarningPanel.SetActive(true);
        canStart = false;
    }
    if (canStart)
    {
        SceneDataPasser.LvlOneH = lvlOneH;
        SceneDataPasser.LvlTwoH = lvlTwoH;
        SceneDataPasser.TimeInMenu = Time.realtimeSinceStartup;
        SceneSwitch.loadScene(SceneSwitch.Scene.MainScene);
    }
}

public void WarningPanelOk()
{
    WarningPanel.SetActive(false);
}

public void PlayInputPanel()
{
    canvasPlaySelect.SetActive(false);
    canvasMenu.SetActive(false);
    canvasInputPanel.SetActive(true);
}

```

```

}

public void GuidePanelShow()
{
    GuidePanelObj.SetActive(true);
    ProjName.SetActive(false);
}

public void GuidePanelClose()
{
    GuidePanelObj.SetActive(false);
    ProjName.SetActive(true);
}

public void Exit()
{
    Application.Quit();
}

public void Authors()
{
    canvasMenu.SetActive(false);
    AutorsPanel.SetActive(true);
}

public void AuthorsBack()
{
    canvasMenu.SetActive(true);
    AutorsPanel.SetActive(false);
}

public void Presentation()
{
    SceneSwitch.LoadScene(SceneSwitch.Scene.PresentationScene);
}
}

```

Файл «SceneDataPasser.cs»:

```

public static class SceneDataPasser
{
    private static int lvlOneH, lvlTwoH;
    private static float time;
    public static float TimeInMenu { get => time; set => time = value; }
    public static int LvlOneH { get => lvlOneH; set => lvlOneH = value; }
    public static int LvlTwoH { get => lvlTwoH; set => lvlTwoH = value; }
}

```

Файл «SceneSwitch.cs»:

```

using UnityEngine.SceneManagement;

public class SceneSwitch
{
    public enum Scene
    {
        Menu,
        MainScene,
        AimingHorizontal,

```

```

        PresentationScene
    }
    public static void loadScene(Scene scene)
    {
        SceneManager.LoadScene(scene.ToString());
    }
    public static Scene getCurrentScene()
    {
        return (Scene)SceneManager.GetActiveScene().buildIndex;
    }
}

```

Файл «InputFields.cs»:

```

using UnityEngine;
using UnityEngine.UI;
using Assets.Resources;

public class InputFields : MonoBehaviour
{
    public InputField inputFieldAim;
    public InputField inputFieldLev1;
    public InputField inputFieldLev2;
    public InputField inputFieldLev1H;
    public InputField inputFieldLev2H;

    public GameObject canvasCheckPanel;
    public Text Calculation;
    public Text timeT;
    public Text PhiAccu;
    public Text GunAcc;
    public Text GunAccH;
    public OnDragCannonWheel cannonWheelAngle;
    public InfPanel phiAngle;

    public VerticalWheelDown getValH;

    void Awake()
    {
        InputParam();
    }

    public void InputParam()
    {
        if (SceneDataPasser.LvlOneH >= 0 && SceneDataPasser.LvlOneH < 10)
        {
            inputFieldLev1H.text = "0" + SceneDataPasser.LvlOneH.ToString();
        }
        else
        {
            inputFieldLev1H.text = SceneDataPasser.LvlOneH.ToString();
        }

        if (SceneDataPasser.LvlTwoH >= 0 && SceneDataPasser.LvlTwoH < 10)
        {
            inputFieldLev2H.text = "0" + SceneDataPasser.LvlTwoH.ToString();
        }
        else
        {
            inputFieldLev2H.text = SceneDataPasser.LvlTwoH.ToString();
        }
    }
}

```

```

    }

    inputFieldAim.readOnly = true;
    inputFieldLev1.readOnly = true;
    inputFieldLev2.readOnly = true;
    inputFieldLev1H.readOnly = true;
    inputFieldLev2H.readOnly = true;
}

public void CheckParam()
{
    canvasCheckPanel.SetActive(true);
    bool isPassed = false;

    float valHorizontal = SceneDataPasser.LvlOneH * 100;
    valHorizontal += SceneDataPasser.LvlTwoH;

    float gunAccuracyH = valHorizontal - getValH.num;
    float timers = Time.realtimeSinceStartup - SceneDataPasser.TimeInMenu;
    timeT.text = Mathf.Floor(timers / 60).ToString("00") + " xB " + (timers %
60).ToString("00") + " сек";

    if (isPassed && timers <= 65f)
    {
        Calculation.text = UIText.Result_A;
    }
    else if (isPassed && timers <= 75f)
    {
        Calculation.text = UIText.Result_B;
    }
    else if (isPassed && timers <= 85f)
    {
        Calculation.text = UIText.Result_C;
    }
    else if (isPassed)
    {
        Calculation.text = UIText.Result_D;
    }
    else
    {
        Calculation.text = UIText.Result_Time;
    }

    GunAccH.text = UIText.ACCURACY_H + gunAccuracyH.ToString();
}

public void GoToMenu()
{
    SceneSwitch.loadScene(SceneSwitch.Scene.Menu);
}
}

```

Файл «Wheel.cs»:

```

using UnityEngine;

public abstract class Wheel : MonoBehaviour
{
    protected Color selectColor = new Color(0.6f, 0.8f, 1f, 0.4f);
}

```

```

protected abstract void MouseDrag(float rotationSpeed);
protected abstract void KeyboardControl(KeyCode firstButton, KeyCode secondButton);

protected abstract float GetAngle();
protected abstract float GetValue();
}

```

Файл «OnDragCannonWheelH.cs»:

```

using UnityEngine;

public class OnDragCannonWheelH : Wheel
{
    //Rotation speed
    public float rotateStepM = 4f; //Mouse Sensetivity
    public float rotateStepB = 0.1f; //Keyboard Sensetivity
    public float rotateStepS; //Scroll Sensetivity
    public float combScale = 80f;

    //Wheel weight
    float stepDegCost = 3.6f;
    float maxVDeg = 360;
    float minVDeg = 0;
    int toZeroRot = 0;
    float rotationCost = 100f;

    public Transform pivotPoint; //Because
    public Transform rotateS;

    public int curRotation = 0;
    public int minRotatations = 0;
    public int maxRotatations = 59;

    float num = 0;
    public float Num { get => GetValue(); set => num = value; }

    [HideInInspector]
    public Transform AzPresPointer, AzRoughtPointer;
    [HideInInspector]
    public Camera cam;
    public Transform mainSceneDup;

    float synRatePresPointer = 1f;
    float synRoughtPointer = 60f;
    float synWheelRate = 60f;
    float syncDegRotatM;

    public Color back;
    public Renderer rend;
    bool selected;
    bool blockMouseDrag;
    CameraControl scroll;

    void Awake()
    {
        back = rend.materials[0].color;
    }

    void RotateTopSightCam(float wheelRotation, float syncDegRotatM)
    {
        cam.transform.Rotate(Vector3.up, syncDegRotatM);
    }
}

```

```

        cam.transform.rotation = Quaternion.Euler(cam.transform.eulerAngles.x,
cam.transform.eulerAngles.y, 0);
    }
    void OnMouseEnter()
    {
        rend.materials[0].color = selectColor;//Change material Color if Enter
        selected = true;
    }

    void OnMouseExit()
    {
        rend.materials[0].color = back;//Reset material color back if Exit
        selected = false;
    }

    void OnMouseDown()
    {
        if (!blockMouseDown)
        {
            MouseDrag(rotateStepM * Input.GetAxis("Mouse Y"));
        }
    }

    void Update()
    {
        if (selected && Input.GetKey(KeyCode.LeftAlt))
        {
            MouseDrag(-rotateStepS * Input.mouseScrollDelta.y);
            if(SceneSwitch.getCurrentScene() == SceneSwitch.Scene.MainScene)
            {
                blockMouseDown = true;
                scroll.blockScroll = true;
            }
        }
        }else if (selected)
        {
            KeyboardControl(KeyCode.UpArrow, KeyCode.DownArrow);
            if (SceneSwitch.getCurrentScene() == SceneSwitch.Scene.MainScene)
            {
                blockMouseDown = false;
                scroll.blockScroll = false;
            }
        }
    }

    private void RotateObjects(Vector3 dir, float speed)
    {
        if (GetAngle() + speed < minVDEg)
        {
            curRotation--;
        }
        else if (GetAngle() + speed > maxVDEg)
        {
            curRotation++;
        }
    }

    AzPresPointer.Rotate(dir, speed / synRatePresPointer);
    AzRoughtPointer.Rotate(dir, speed / synRoughtPointer);
    mainSceneDup.Rotate(Vector3.up, speed / synRatePresPointer);
    mainSceneDup.transform.rotation = Quaternion.Euler(mainSceneDup.transform.eulerAngles.x,
mainSceneDup.transform.eulerAngles.y, 0);

```



```

    rotateS.Rotate(dir, speed);
}

protected override void KeyboardControl1(KeyCode firstButton, KeyCode secondButton)
{
    float rotateStepBIncr = rotateStepB * combScale;

    if (Input.GetKey(firstButton))
    {
        RotateObjects(Vector3.back, rotateStepB);
    }
    else if (Input.GetKey(secondButton))
    {
        RotateObjects(Vector3.forward, rotateStepB);
    }

    if (Input.GetKey(firstButton) && Input.GetKey(KeyCode.LeftShift))
    {
        RotateObjects(Vector3.back, rotateStepBIncr);
    }
    else if (Input.GetKey(secondButton) && Input.GetKey(KeyCode.LeftShift))
    {
        RotateObjects(Vector3.forward, rotateStepBIncr);
    }
    LogicOfValues();
}

protected override void MouseDrag(float rotationSpeed)
{
    RotateObjects(Vector3.back, rotationSpeed);

    syncDegRotatM = rotationSpeed / synWheelRate;

    if (SceneSwitch.Scene.AimingHorizontal == SceneSwitch.getCurrentScene())
    {
        RotateTopSightCam(rotationSpeed, syncDegRotatM);
    }
}

protected override float GetAngle()
{
    Vector3 forwardVector = pivotPoint.rotation * Vector3.right;

    float CurDeg = Mathf.Atan2(forwardVector.y, forwardVector.x) * Mathf.Rad2Deg;

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

    return CurDeg;
}

protected override float GetValue()
{
    return ((curRotation - toZeroRot) * rotationCost) +
        (Mathf.Round((GetAngle() / stepDegCost)));
}

void LogicOfValues()
{
    if (curRotation > maxRotatations)//Full circle in a clockwise direction, set value MinVal
    {
        curRotation = minRotatations;
    }
}

```

```
    }  
    else if (curRotation < minRotatations)// Full circle in a counterclockwise direction, set  
maxVal  
    {  
        curRotation = maxRotatations;  
    }  
    }  
}
```

Додаток Г
Акти впровадження та апробації проекту

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 28 вересня 2020 р. № 243 про передачу (відступлення) майнових прав і прийнято рішення зареєструвати авторський договір, відповідно до якого майнові права на твір

Комп'ютерна програма «Віртуальний тренажер установки горизонтальних кутів прицілювання прицілу ПП-4»; Дерев'яничук Анатолій Іосифович, Шелест Микола Борисович, Кравченко Дмитро Олександрович, Чичикало Євгеній Андрійович
(ім'я, прізвище, по батьку (за наявності) нова твору, повне ім'я, походження (за наявності) автора(ів))

передаються(відступуються)

Дерев'яничук Анатолій Іосифович, 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. лікарів І. Дерелянка, 6, кв. 57, м. Суми, 40022; Чичикало Євгеній Андрійович, вул. Перша Замостинська, 7, кв. 148, м. Суми, 40007
(повне ім'я фізичної або повне офіційне найменування юридичної особи(ів), яка(і) передає(ють, відступують)(ся) 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 (відступується) право на твір, адреса)

Повністю

Реєстраційний номер	5857
Дата реєстрації	16 листопада 2020 р.

Генеральний директор
Державного підприємства
«Український інститут
інтелектуальної власності»

 Андрій КУДІН



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 майнової права на твір

Комп'ютерна програма "Віртуальний тренажер установки вертикальних кутів прицілювання прицілу ПГ-4"; Дерев'яничук Анатолій Йосипович, Шелест Микола Борисович, Кравченко Дмитро Олександрович, Чичикало Євгеній Андрійович
(ім'я, повне, скорочена (за наявності) назва твору, повне ім'я, псевдонім (за наявності) автора(ів))

передаються(відчужуються)

Дерев'яничук 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. Деревлянка, 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(ов), який передається(відчужується) 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**

**Virtual simulator of horizontal aiming angles
of the PG-4 artillery sight**

D.O. Kravchenko, Student; E.G. Kuznetsov, Senior Lecturer
Sumy State University, Sumy, Ukraine

The leading place in the system of formation of the future artillery officer is occupied by practical military-technical disciplines. The biggest problem in the training of future artillerymen is the consolidation of the studied knowledge in practice, namely: the lack of full-fledged practice of working with an artillery sight during the training of gunners of self-propelled artillery pieces. This problem is expressed in the fact that the sights themselves are behind the armor and only one person can exercise training, which reduces the control over the actions of the student from the mentor and hides the typical mistakes of the student from his peers. Thanks to the developed computer learning environment – a virtual simulator – it is possible to set the consolidation of action skills with an eye to the level of mass classes in the classroom under the supervision of a single mentor, to carry out a collective study of the main patterns of response to regular situations, to discuss typical mistakes in the actions of lagging students. Thus, the virtual simulator being developed provides the opportunity to either choose an individual training mode, or use the mode of collective and competitive practical exercises. The purpose of the application development is to create a simulator of interaction with an artillery sight by interactive simulation of horizontal guidance. This simulator is designed for use in the Windows operating system. Purpose-training of students of cadets of the Department of artillery training. The object of simulation in the simulator is the PG-4M sight – the main standard sight of the 2S3M self-propelled artillery unit. Since this product has many common features with the design of sights of similar artillery systems, the developed simulator can be useful for students studying other types of artillery weapons. The Unity platform was chosen as a tool for implementing the project. This environment allows you to use tools, models, and algorithms created in third-party modeling tools by unaffiliated developers, and also has its own fairly extensive arsenal of modeling objects and situations. Models of individual elements, as well as the sight itself, are developed in the SolidWorks environment.

Дерев'янчук А.Й.,
Шелест М.Б.,
Кравченко Д.О.,
Чичикало Є.А.

Сумський державний університет, Україна

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 унеможлиблює 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.

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

В сучасних умовах розвитку 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.

Наша система пропонує декілька різновидів симуляційних тренажерів різних за своєю сутністю та складністю.

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 інструменту 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.