

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Савченко А.С.

“___” _____ 2020 р.

ДИПЛОМНА РОБОТА **(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ
“МАГІСТРА”

ЗА СПЕЦІАЛІЗАЦІЮ “ІНФОРМАЦІЙНІ УПРАВЛЯЮЧІ СИСТЕМИ ТА
ТЕХНОЛОГІЇ (ЗА ГАЛУЗЯМИ)”

Тема: “Процедурний тренажер для підготовки екіпажів повітряних суден”

Виконавець: Джуринський Кирило Антонович

Керівник: д.т.н., професор Моржов Володимир Іванович

Нормоконтролер: _____ Райчев І.Е.

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, спеціалізація: 12 "Інформаційні технології", 122 "Комп'ютерні науки", "Інформаційні управляючі системи та технології (за галузями)"

ЗАТВЕРДЖУЮ
Завідувача кафедри
Савченко А.С.
"___" _____ 2020 р

ЗАВДАННЯ

на виконання дипломної роботи студента
Джуринського Кирила Антоновича

- 1. Тема дипломної роботи:** «Процедурний тренажер для підготовки екіпажів повітряних суден» затверджена наказом ректора 02.10.2020 р. № 1891/ст.
- 2. Термін виконання проекту:** з 5.10.2020 р. до 31.12.2020 р.
- 3. Вихідні дані до роботи :** розробити програмний продукт за допомогою апаратних та програмних VR-технологій
- 4. Зміст пояснювальної записки:**
 1. Підготовка членів льотного екіпажу. Порядок проведення тренувань на авіаційному тренажері
 2. Склад та конструкція робочих місць інструкторів авіаційного тренажеру. Вимоги до апаратних та програмних засобів програмного забезпечення
 3. Архітектура програмного забезпечення системи для тренування пілотів на авіаційному тренажері із застосуванням технологій віртуальної реальності
 4. Програмна реалізація системи для тренування пілотів на авіаційному тренажері із застосуванням технологій віртуальної реальності. Математичні розрахунки. Функціональні можливості
- 5. Перелік обов'язкового ілюстративного матеріалу:**
 - Порівняння існуючих програмних рішень проектування методики фільтрації зображення.
 - Функціональні можливості програмного продукту.
 - Графічне представлення структури та проекту системи. Схема роботи.
 - Демонстрація роботи системи.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Підпис керівника
1.	Розробка та затвердження графіка роботи..	05.10.20 – 11.10.20	
2.	Підготовка та написання 1 розділу	12.10.20 – 19.10.20	
3.	Перший нормо-контроль 1-2 розділів Підготовка та написання 3 розділу	26.10.20 - 01.11.20	
4.	Підготовка та написання 4 розділу	02.11.20 – 22.11.20	
5.	Редагування та друк пояснювальної записки, графічного матеріалу	23.11.20 - 29.11.20	
6.	Проходження нормо-контролю, перепліт пояснювальної записки. Отримання відгуку керівника. Підготовка презентації та тексту доповіді.	30.11.20 - 14.12.20	
7.	Попередній захист дипломної роботи. Підпис ПЗ завідувачем кафедри на допуск до захисту та для отримання рецензії.	14.12.20 - 15.12.20	
8.	Отримання рецензії. Для проходження контролю на плагіат здати секретарю ДЕК текст ПЗ одним файлом.	16.12.20 - 17.12.20	
9.	Здати секретарю ДЕК: ПЗ, ГМ, CD-R з електронними версіями ПЗ, ГМ, презентацію, відгук керівника, рецензію, довідку про успішність, 2 папки, 2 конверта)	17.12.20 – 18.12.20	
10.	Захист дипломної роботи перед ЕК	21.12.20 - 27.12.20	

7. Консультація з окремих розділів роботи:

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
1	Моржов В.І.		
2	Єрмачков Ю.О		

8. Дата видачі завдання:

06.10.2020 р.

Керівник дипломної роботи:

д.т.н. професор Моржов В.І.

Завдання прийняв до виконання:

Джуринський К.А.

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Процедурний тренажер для підготовки екіпажів повітряних суден» містить 94 сторінок, з них 4 сторінки додатків, 25 рисунків, 8 таблиць, 11 використаних джерел, 9 формул.

Ключові слова: ПРОЦЕДУРНИЙ АВІАЦІЙНИЙ ТРЕНАЖЕР, VR-ТЕХНОЛОГІЇ, ПРИСТРОЇ ВВОДУ, ВІРТУАЛЬНА РЕАЛЬНІСТЬ, ВІДТВОРЕННЯ КАБИНИ ЛІТАКА, КОМП'ЮТЕРНА ГРАФІКА, ПРОТОТИП ПРОГРАМНОГО ПРОДУКТУ.

Об'єкт дослідження - програмний продукт система для тренування пілотів на авіаційному тренажері із застосуванням технологій віртуальної реальності: проект, прототип, програмна документація, програмна реалізація.

Мета дипломної роботи – запропонувати новий підхід до тренувань пілотів на процедурному авіаційному тренажері, використовуючи VR-технології. Розробка програмного рішення та пропонування отриманої методики для експлуатації.

Метод дослідження – визначення задач, розробка програмної архітектури, розробка математичної моделі та реалізація VR-застосунку авіаційного тренажеру.

Результати роботи можуть бути використані при розробці, модифікації або прямої інтеграції програмного забезпечення для тренування пілотів на процедурному авіаційному тренажері.

Розробка та дослідження проводилися під управлінням ОС Windows 10. Розробка програми проводилася у середі розробки Unity, на мові програмування C++.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, термінів.....	9
ВСТУП.....	11
РОЗДІЛ 1.	18
ПІДГОТОВКА ЧЛЕНІВ ЛЬОТНОГО ЕКІПАЖУ. ПОРЯДОК ПРОВЕДЕННЯ ТРЕНУВАНЬ НА АВІАЦІЙНОМУ ТРЕНАЖЕРІ.....	18
1.1. Порядок підготовки пілотів.....	18
1.2.1. Програмний тренажер.....	22
1.2.1.1. Основні можливості:	22
1.2.1.2. Програмні модулі:	23
1.2.2. Комплексний тренажер.....	23
1.2.3. Навчально-тренувальний комплекс.....	24
1.2.4. Автоматизована система навчання.....	26
1.2.4.1. АСН дозволяє:.....	26
1.2.4.2. Основні можливості:	26
1.3. Структура і зміст навчально-методичних прийомів навчання пілотування.....	27
1.3.1. Прийоми навчання пілотування	29
1.3.2. Показ і інструктування вирішується наступними прийомами:.....	29
1.3.3. Тренування забезпечується наступними прийомами:.....	29
1.3.4. Перевірка результатів і визначення якості навчання:	29
1.4. Порядок проведення підготовки на тренажерах.....	30
1.5. Висновки розділу	31
РОЗДІЛ 2.	33
СКЛАД ТА КОНСТРУКЦІЯ РОБОЧИХ МІСЦЬ ІНСТРУКТОРІВ АВІАЦІЙНОГО ТРЕНАЖЕРУ. ВИМОГИ ДО АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	33
2.1. Структурний склад системи авіаційного тренажеру та функції її окремих елементів	33

2.2. Моделювання подачі інформації на робоче місце інструктора	41
2.3. Опис панелі приборів потоків інформації	42
2.4. Пристрої вводу	45
2.4.1. Клавіатура	46
2.4.2. Миша	46
2.4.3. Взаємодії за допомогою сенсорного вводу	47
2.5. Обов'язкові вимоги, які необхідні для взаємодії з сенсорним екраном.....	52
2.6. Висновки розділу	53
РОЗДІЛ 3.	55
АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ДЛЯ ТРЕНУВАННЯ ПІЛОТІВ НА АВІАЦІЙНОМУ ТРЕНАЖЕРІ	55
3.1. Багаторівнева архітектура	57
3.1.1. Рівень графічного інтерфейсу користувача	60
3.1.2. Рівень графічного модулю	62
3.1.3. Рівень логіки обчислень	63
3.1.4. Рівень роботи з даними	65
3.1.5. Рівень об'єктного модулю.....	66
3.2. Опис алгоритму взаємодії програмних модулів	68
3.3. Висновки розділу	69
РОЗДІЛ 4.	71
ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЛЯ ТРЕНУВАННЯ ПІЛОТІВ НА АВІАЦІЙНОМУ ТРЕНАЖЕРІ ІЗ ЗАСТОСУВАННЯМ ТЕХНОЛОГІЙ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ. МАТЕМАТИЧНІ РОЗРАХУНКИ. ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ.....	71
4.1. Технології та інструменти розробки програмної системи.....	71
4.1.1. Unity.....	72
4.1.2. DirectX 11	72
4.1.3. Visual studio.....	73
4.1.4. Rift Cat	73
4.1.5. Java.....	74
4.1.6. Steam VR	74

4.2. Розрахунок лінійних та нелінійних параметрів ПС.....	75
4.2.1. Системні координати, які використовуються у динаміці польоту ПС.....	75
4.2.2. Розрахунок швидкості польоту ПС.....	76
4.3. Основні функції та процеси програмного забезпечення.....	79
4.4. Системні вимоги. Вимоги до апаратної частини комп'ютерної системи.....	83
4.5. Висновки розділу	83
ВИСНОВКИ.....	86
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ	91
ДОДАТОК А. ТЕКСТ ПРОГРАМИ	92

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

АСН – автоматизовані системи навчання. Комплекс програмно-технічних і навчально-методичних засобів, що забезпечують активну індивідуальну пізнавальну діяльність особи, яка навчається, на основі програмного керування цією діяльністю.

АСУ – автоматизовані системи управління. Комплекс апаратних і програмних засобів, а також персоналу, призначений для управління різними процесами в рамках технологічного процесу, виробництва, підприємства. АСУ застосовуються в різних галузях промисловості.

ІНК – інтегровані навчальні комплекси, які можуть включати в різні комбінації автоматизовані навчальні системи, тренажери різного рівня складності, робочі місця наземних операторів.

КЛЕ – керівництво льотної експлуатації. Керівництво, яке включає порядок дій у звичній, особливій та аварійній ситуаціях, контрольні карти, обмеження, інформацію про льотно-технічні характеристики, відомості про безпілотний літальний апарат та кожну відповідну модель пункту дистанційного пілотування, а також інші матеріали, пов'язані з експлуатацією безпілотного авіаційного комплексу.

ПЗ – програмне забезпечення. Сукупність програм системи обробки інформації і програмних документів, необхідних для експлуатації цих програм.

ПК – персональний комп'ютер.

ПС – літальний апарат, що підтримується в атмосфері у результаті його взаємодії з повітрям, відмінної від взаємодії з повітрям, відбитим від земної поверхні.

СУ – силова установка. Енергетичний комплекс, призначений для отримання механічної енергії і складається з рухової установки і допоміжного оснащення до неї.

API - (програмний інтерфейс програми, інтерфейс прикладного програмування) (англ. Application programming interface, API) - опис способів (набір класів, процедур, функцій, структур або констант), якими одна комп'ютерна програма може взаємодіяти з іншою програмою. Зазвичай входить в опис будь-якого інтернет-протоколу (наприклад, RFC), програмного каркаса (фреймворка) або стандарту викликів функцій операційної системи). Часто реалізується окремою програмною бібліотекою або сервісом операційної системи.

GUI - Графічний інтерфейс користувача — тип інтерфейсу, який дає змогу користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки, на відміну від текстових інтерфейсів, заснованих на використанні тексту, текстовому наборі команд та текстовій навігації.

VR - Віртуальна реальність — різновид реальності в формі тотожності матеріального й ідеального, що створюється та існує завдяки іншій реальності. У вужчому розумінні — ілюзія дійсності, створювана за допомогою комп'ютерних систем, які забезпечують зорові, звукові та інші відчуття.

ВСТУП

Важко переоцінити, наскільки важливою є авіаційна галузь у всій промисловості країни. Наразі, повітряний простір – це найбільш прямолінійний і швидкий шлях від однієї точки в іншу. Україна, до недавніх часів, входила в дев'ятку країн, що мали замкнутий технологічний цикл будівництва літаків. Крім цього, мережу авіаремонтних підприємств, в тому числі і для відновлення бойових літаків. Але, останнім часом, авіаційна галузь України занепала.

Нові геополітичні умови вимагають стрімкого технологічного розвитку, у тому числі авіації. За відомих обставин, починаючи з 2014 р. ,Україна почала відродження авіації, виділяючи все більші і більші бюджети на розвиток цієї галузь.

Якісному розвитку авіаційних тренажерів для потреб Міністерства Оборони України в значній мірі перешкоджає відсутність сучасних нормативних документів, що визначають критерії їх кваліфікаційної оцінки. Якнайшвидше вирішення цієї проблеми повинно максимально прискорити і здешевити процес підготовки військового льотчика. Особливо це актуально в умовах сучасної України.

Виходячи з різноманіття завдань, які можливо сформулювати і відпрацювати на тренажері, в передових країнах-лідерах авіабудування і використання авіації створено велике різноманіття спеціалізованих програм підготовки екіпажів на симуляторах різних конфігурацій. Як цивільні, так військові льотчики передових країн світу, як мінімум, 80-100 годин на рік, проводять за штурвалами тренажерів перед тим, як сядуть за управління бойового літального апарату.

Крім самих літальних апаратів, потрібні підготовлені кадри, що будуть ними керувати. Тому існує окремий напрямок в авіації – підготовка пілотів та бортмеханіків. Сучасні авіаційні тренажери знаходять також застосування в дослідницьких цілях, наприклад, для відпрацювання дій екіпажу при виході за межі експлуатаційних обмежень (вихід на великі кути атаки, вихід зі складних просторових положень і т. п.).

Метою дипломного проекту є створення прототипу процедурного авіаційного тренажера для підготовки пілотів із застосуванням технології VR (Virtual Reality), тобто, віртуальна реальність.

Об'єктом досліджень є авіаційний тренажер та сам процес тренування. В ході досліджень було визначено основні технології та пристрої, що застосовуються під час тренування, які можна вдосконалити, тим чи іншим чином, або, замінити на нові, більш технологічно досконалі пристрої.

У цій роботі тренажери діляться на два типи: комплексні та процедурні (насправді їх набагато більше). Процедурні тренажери призначені для відпрацювання екіпажем процедур підготовки та виконання польоту. У тренажерах такого призначення пульти, прилади і органи управління зазвичай імітуються за допомогою сенсорних моніторів.

Під комплексними тренажерами розуміють авіаційні тренажери, що забезпечують підготовку екіпажів в повному обсязі їх функціональних обов'язків з льотної експлуатації повітряного судна конкретного типу.

Комплексні тренажери - це тренажери найвищого рівня. Як правило, вони мають систему рухливості. Кабіна комплексного тренажера виконується у вигляді повної репліки реальної кабіни повітряного судна. На комплексні тренажери встановлюються передові системи візуалізації.

Було проаналізовано які складові тренувального комплексу можна вдосконалити, або, замінити VR-технологією. Цю ідею намагаються реалізувати провідні країни світу, такі як США, Німеччина, Китай, Японія, Канада, фінансуючи проекти, фонди, корпорації, що займаються VR-технологіями.

Завданням даної дипломної роботи є розробка програмного забезпечення для прототипу процедурного авіаційного тренажера для підготовки пілотів, із застосуванням VR-технологій. ПЗ повинно відтворювати кабінний простір літака, мати достатній функціонал, для відтворення процедур, які виконуються під час польоту, забезпечувати почуття реальності того, що відбувається.

Для досягнення поставленої мети, розглядаються основні шляхи вирішення даної проблеми:

- проаналізувати принцип роботи існуючих авіаційних тренажерів;
- визначити основні потреби кінцевого користувача;
- на основі потреб користувача, скласти вимоги до програмного продукту;
- розробити проект, або архітектуру програмного продукту;
- обрати основні засоби та технології, які будуть застосовуватись у розробці програмного забезпечення продукту;
- розробити сам програмний продукт;
- здійснити тестування отриманого програмного забезпечення;
- представити у доступній формі програмний продукт кінцевому користувачеві;
- впродовж усього життєвого циклу програмного продукту (у тому числі після його випуску), підтримувати та вдосконалювати програмно-апаратну частину системи тренажеру.

У даній дипломній роботі, для створення прототипу програмного продукту авіаційного тренажеру для підготовки пілотів із застосування технологій VR, були застосовані такі засоби та технології:

- Unity 3D - багатоплатформене середовище розробки програмного продукту;
- SteamVR, OpenVR - це набір для розробки програмного забезпечення і інтерфейс прикладного програмування, розроблений Valve для підтримки SteamVR і інших гарнітур віртуальної реальності. Платформа SteamVR використовує його як інтерфейс прикладного програмування за замовчуванням і середу виконання.;
- RiftCat – засіб емуляції роботи VR-шолому;
- C++, C# - мови програмування загального призначення. Підтримують такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування;
- CMake - це кроссплатформенна система автоматизації збірки програмного забезпечення з вихідного коду. CMake не займається безпосередньо написаннями, а лише генерує файли управління для компіляції з файлів CMakeLists.txt;

- Visual Studio, Notepad++ - текстові редактори, що також включають у себе інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів;
- DirectX 11, Direct3D 11 - компонент інтерфейсу програмування додатків (англ. API) DirectX 11, 11-я версія Direct3D, наступник Direct3D 10 / 10.1. Direct3D 11 забезпечує функції для взаємодії операційної системи і додатків з драйверами відеокарти.

Основними конкурентними перевагами застосування VR-тренажерів замість традиційних процедурних тренажерів є:

- імітація реальності того, що відбувається;
- більш реальні взаємодії пілота-студента з об'єктами маніпуляцій (штурвали, кнопки, важелі і т.д.);
- економія простору: не потрібне велике приміщення, для того щоб розташувати робочі столи, прилади вводу та виводу інформації і т.п.;
- великий потенціал VR-технологій, стрімкий розвиток та підтримка цього напрямку від провідних технологічних компаній на країн світу.

Результатом даної роботи є програмна система на базі VR-технології, що імітує кабінний та закабінний простір; створює відчуття реальності того, що відбувається; забезпечує можливість відтворити певні процедури, які виконуються під час реального польоту; яка пропонує новий підхід до тренувань на процедурному авіаційному тренажері та до авіаційних тренувань в цілому.

У першому розділі пояснювальної записки надаються основні теоретичні відомості, проблеми, які вирішуються за допомогою авіаційних тренувань, описуються основні проблеми, пов'язані із традиційними методами авіаційних тренувань.

У другому розділі пояснювальної записки описуються основні (традиційні) методи вводу та виводу інформації, засоби взаємодії користувача з програмною та апаратною системою авіаційного тренажеру, обґрунтовуються переваги нового методу взаємодії з системою авіаційного тренажеру (технологія VR), ставляться задачі, які програмний продукт повинен виконувати.

У третьому розділі пояснювальної записки описується архітектура та надаються діаграми програмного продукту. Програма умовно розбивається на модулі, які ієрархічно залежать один від одного. Кожен модуль виконує свою область певних функцій. Результати виконання роботи нижнього за ієрархією модуля використовуються у роботі верхнього за ієрархією модуля, і так до кінця (готового програмного продукту). Також, у даному розділі надається перелік завдань, які можна виконувати в програмному забезпеченні.

У четвертому розділі пояснювальної записки описується безпосередньо реалізація. Описується застосування та роль у системі програмного продукту кожної з перелічених вище технологій та засобів (Unity 3D, SteamVR і т.д.). Підсумовується результат виконаної роботи, оцінюється якість отриманого програмного продукту, описуються проблеми, що виникли при розробці та використанні програмного продукту.

Отриманий програмний продукт є лише прототипом, а тому, повністю функціонал не реалізовано. Метою даного дипломного проекту – продемонструвати технічні можливості, при застосуванні VR-технологій у сфері авіаційного тренування.

Вимоги до авіаційного тренажеру:

1. Авіаційний тренажер, представлений для початкового оцінювання, має оцінюватися на відповідність вимогам:

- кваліфікаційних рівнів А, В, С, D для пілотажних тренажерів;
- кваліфікаційних рівнів 1, 2, 3 для тренажерів льотної підготовки;
- кваліфікаційних рівнів процедурних тренажерів типу I, II, III.

2. Авіаційний тренажер, представлений для початкового оцінювання, повинен пройти:

- об'єктивні випробування;
- функціональні та суб'єктивні випробування.

3. Акт кваліфікаційних випробувань, повинен включати всі дані, допоміжні матеріали та інформацію і мають бути надані в форматі, що дозволяє ефективні розгляд і оцінку до того, як може вийти на рівень кваліфікації. У випадках, де це

може бути застосовано, має ґрунтуватися на даних оцінки ПС, як визначено даними відповідності роботи.

2. Авіаційний тренажер, який представлений для повторного оцінювання, та такий, що має відповідні документи затвердженої форми, що містять результати випробувань, може бути сертифікований уповноваженою організацією відповідно до вимог даного Керівництва.

3. Авіаційний тренажер оцінюється щодо тих напрямків, що є суттєвими для підготовки, перепідготовки та перевірки членів льотного екіпажу, а саме:

- характеристик поздовжньої, поперечної та шляхової керованості;
- льотно-технічних характеристик повітряного судна на землі та в повітрі;
- необхідних особливостей пілотування;
- компонування кабіни екіпажу;
- функціонування в умовах нормальної експлуатації та у тих випадках, коли це прийнятне, в особливих умовах експлуатації;
- функціонування обладнання робочого місця інструктора.

4. Під час кваліфікаційних випробувань Авіаційний тренажер відповідно до акта кваліфікаційних випробувань повинні проводитися:

- об'єктивні випробування;
- функціональні та суб'єктивні випробування.

5. Форма подання даних випробувань Авіаційний тренажер повинна узгоджуватися з уповноваженим органом до початку кваліфікаційних випробувань.

6. Організація з експлуатації повинна представити акт кваліфікаційних випробувань, який повинен враховувати характеристики та рівні адекватності реальному ПС.

7. Після завершення початкового або повторного оцінювання, узгодження всіх невідповідностей уповноважений орган затверджує акт кваліфікаційних випробувань та засвідчує результати випробувань у Висновку з кваліфікаційних випробувань, який надалі є підґрунтям для наступних періодичних оцінювань Авіаційний тренажер.

8. Організація з експлуатації повинна:

- між щорічними оцінюваннями уповноваженою організацією виконувати повний обсяг робіт з метою підтримання технічного стану Авіаційний тренажер відповідно до встановленого кваліфікаційного рівня згідно з Висновком з кваліфікаційних випробувань, реєструвати результати поточних робіт, які повинні датуватися й зберігатися;
- для підтримання довготривалої цілісності обладнання й програмного забезпечення установити систему контролю за конфігурацією Авіаційний тренажер [1].

РОЗДІЛ 1.

ПІДГОТОВКА ЧЛЕНІВ ЛЬОТНОГО ЕКІПАЖУ. ПОРЯДОК ПРОВЕДЕННЯ ТРЕНУВАНЬ НА АВІАЦІЙНОМУ ТРЕНАЖЕРІ

1.1. Порядок підготовки пілотів

Підготовка льотного екіпажу проводиться з метою отримання і підтримки рівня знань та умінь, які необхідні для виконання правильних дій членів команди екіпажу в стандартних умовах і особливих ситуаціях, які несуть за собою високий рівень небезпеки польотів.

Підготовка проводиться відповідно до затверджених нормативними документами, програмами, інструкціями та курсами навчально-льотної підготовки та будується на принципах:

- свідомої і активної участі кожного учня в навчальному процесі;
- експлуатаційної спрямованості процесу підготовки, кінцевою метою якої є придбання (підтримка) знань і вироблення практичних умінь членів льотних екіпажів діяти безпомилково і своєчасно в очікуваних умовах і особливих ситуаціях в польоті;
- застосування навчання з використанням технічних засобів навчання, що дозволяє враховувати індивідуальні особливості учнів;
- систематичної самостійної навчання членів льотних екіпажів в процесі льотної діяльності з метою підтримки отриманих професійних знань і практичних навичок на рівні, що забезпечує високий рівень безпеки польотів;
- використання дистанційних навчальних технологій у підготовці членів льотних екіпажів.

Кафедра КІТ (47)				<i>НАУ 20 09 38 000 ПЗ</i>				
			<i>Підпис</i>	<i>Дата</i>	<i>ПІДГОТОВКА ЧЛЕНІВ ЛЬОТНОГО ЕКІПАЖУ. ПОРЯДОК ПРОВЕДЕННЯ ТРЕНУВАНЬ НА АВІАЦІЙНОМ ТРЕНАЖЕРІ</i>	<i>Літера</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Разраб</i>	<i>Джуринський К.А.</i>						18	16
<i>Керівник</i>	<i>Моржов В.І.</i>							
<i>Н. Контр.</i>	<i>Райчев І.Е.</i>						<i>УС-211М</i>	<i>122</i>

Професіоналізм льотного складу залежить, в тому числі, від ступеня відповідності теоретичної підготовки льотчика характеру його діяльності при виконанні польотів. Протягом декількох останніх десятиліть прогрес в конструкції транспортних літаків полягав, в тому числі, і в широкому впровадженні в склад ряду функціональних систем такого ПС цифрової обчислювальної техніки при забезпеченні необхідного рівня надійності цих систем. Внаслідок цього значно розширилися, по-перше, можливість автоматизації управління польотом, а по-друге, можливість автоматизації управління функціонуванням деяких літакових систем і контролю їх працездатності. Це дозволило скоротити чисельність льотного екіпажу до двох льотчиків. При цьому істотно змінилися характер і зміст їх професійної діяльності при виконанні типових польотів. Тотальне використання режиму автоматичного управління (практично на всіх етапах польоту) призвело до того, що тепер лінійний пілот в основному працює з пультами управління (ПА, FCU, MCU, CCB, MCDU, CDU), а не діє важелями, здійснюючи пілотування. Але, тим не менш, вимога вміння пілотувати літак в ручному (або штурвальному за прийнятою термінологією) режимі, як і раніше має залишатися обов'язковим компонентом його професійної підготовки. Причини цього наступні. По-перше, відповідно до нормативних документів відповідальність за благополучний результат польоту (і в цивільній, і в державній авіації) покладається персонально на льотчика (командира екіпажу). Очевидно, що це вимагає від нього і при діючому режимі автоматичного управління безперервно контролювати адекватність і якість функціонування техніки. А для цього необхідно здійснювати сприйняття і обробку пілотажно-навігаційної інформації в повному обсязі, як при ручному пілотуванні. Тільки при виконанні цієї умови забезпечується можливість своєчасного визначення порушення працездатності автоматичного режиму, його безпечне відключення і негайний перехід до штурвального режиму пілотування.

По-друге, звернемо увагу на наступні показові випадки, що мали місце при виконанні планових польотів цивільними літаками з пасажирями на борту:

- вимушена посадка літака Airbus A-320 авіакомпанії «US Airways» 15. січня 2009 року на акваторію річки Гудзон через відмову силової установки (СУ)

внаслідок зіткнення зі зграєю птахів в наборі висоти після зльоту;

- вимушена посадка літака Ту-154М авіакомпанії «Алроса-Мирний» 7 вересня 2010 року на виведеному з експлуатації аеродромі для місцевих повітряних ліній з довжиною ЗПС 1200 м через повної відмови системи електропостачання з одночасним повною відмовою інструментальної індикації просторового положення літака в умовах , коли здійснювалося пілотування за приладами.

Наведені приклади характерні тим, що льотчик для того, щоб продовжити керований політ і виконати благополучну посадку, був змушений відключити автоматичний режим і перейти на ручне пілотування. На підставі практики можна зробити висновок, що і при досягнутих в даний час рівень розвитку авіатехніки і рівні організації процесу функціонування авіаційної транспортної системи (АТС) не представляється можливим виключити певні події випадкового характеру, супутні польотів таких ВС, які зажадають подібних рішень і дій. Недостатня підготовка льотчика в цьому плані може привести до фатальних наслідків. Наочний приклад цього - катастрофа літака Боїнг 737-500 авіакомпанії «Татарстан» 17 вересня 2013 р при виконанні елементарного маневру для виконання повторного заходу на посадку (при виході на друге коло) в аеропорту міста Казань. При пілотуванні в ручному режимі через ненавмисного введення літака в складне просторове положення на малій висоті сталося зіткнення із землею в керованому польоті.

Об'єктивність вимоги вміння пілотувати літак в ручному режимі знайшла відображення в документах, що регламентують професійну підготовку льотного складу цивільної авіації. Наприклад, кваліфікація лінійного пілота має на увазі здатність льотчика «плавно і точно здійснювати ручне керування літаком в межах встановлених експлуатаційних обмежень параметрів польоту, забезпечуючи успішне виконання схеми польоту або маневру». Ну а для військових льотчиків і цивільних пілотів, які виконують польоти на легких цивільних літаках, вміння пілотувати в ручному режимі - ще більш актуальна вимога в порівнянні з лінійними пілотами. Є певні теоретичні знання, на підставі яких згодом, при проходженні курсу початкової льотної підготовки, у людини буде формуватися вміння

здійснювати пілотування літака. Вони є обов'язковими і відправними для всіх приватних методик навчання на конкретних типах ПС.

Як відомо, здатність людини здійснювати будь-який вид діяльності забезпечується наявністю у нього такої системної властивості, як психіка. Ця властивість проявляється як сукупність відповідних реакцій функціонуючого мозку на зовнішні, зумовлені впливом навколишнього середовища, і внутрішні, зумовлені станом організму, впливу. Психіка людини забезпечує йому можливість активного відображення реальності, побудови на цій основі невідчужуваною від себе суб'єктивної внутрішньої (розумової) картини і саморегуляції за рахунок цього своєї поведінки. Тобто можливість льотчика в кожен момент часу в польоті усвідомлювати фактичну ситуацію, поточну мету пілотування, приймати рішення щодо дій важелями управління і виконувати їх обумовлюється реалізованими психічними процесами і супроводжуваними ці процеси психічними станами.

1.2. Авіаційні тренажери

Авіаційні тренажери призначені для наземної підготовки пілотів. Апаратно-програмні комплекси авіаційних тренажерів імітують динаміку польоту та експлуатацію авіаційних систем за допомогою спеціального програмного забезпечення комп'ютерного комплексу тренажерів. Навчання пілотів на льотному тренажері є одним з найважливіших елементів забезпечення безпечної експлуатації літака. Він мінімізує негативний вплив так званого людського фактора, тобто мінімізує можливість помилкових дій екіпажу літака.

У військовій авіації льотні тренажери представляють особливу цінність, оскільки дозволяють майже без обмежень моделювати реальну бойову обстановку, яку дуже важко імітувати в мирний час під час тренувань.

За підрахунками, для нормального процесу навчання пілота необхідний принаймні один імітатор польоту на 25 літаків.

Авіаційні тренажери поділяються на три основні групи:

- тактичні тренажери, призначені для відпрацювання групових боїв. Вони об'єднані в єдину мережу за допомогою інтерфейсу HLA, що дозволяє поєднувати різноманітні тренажери - авіацію, танк, артилерію та інші.
- складні тренажери - це симулятори вищого рівня. Як правило, вони мають систему мобільності. Кабіна комплексного симулятора виконана у вигляді повної копії справжньої кабіни літака. На складних тренажерах встановлені вдосконалені системи візуалізації.
- навчальний пристрій для польотних процедур.

У сучасній практиці підготовки пілотів цивільної авіації найбільш поширені складні тренажери та процедурні тренажери.

1.2.1. Програмний тренажер

Перш за все, процедурний тренажер має цілком реальний інтер'єр кабіни з реальним інструментальним середовищем та органами управління, який зазвичай входить до вимог лише для найскладніших інтегрованих тренажерів.

Система візуалізації імітатора - це чотириканальна оптико-колімаційна система з комп'ютерним синтезом зображень і забезпечує безперервне та об'єднане зображення простору кабіни з кутами огляду 86° по горизонталі та 28° по вертикалі для кожного члена екіпажу.

Робочі місця інструкторів тренажера (пілот-інструктор та штурман-інструктор) забезпечують вирішення проблем управління навчальним процесом та оцінки дій учнів, включаючи використання автоматизованих систем управління. Тренажер працює за принципом генерації зображень.

1.2.1.1. Основні можливості:

- виконання польоту для бойового застосування; підготовка до польоту;
- управління бортовими системами літального апарату за інтерактивними функціональними схемами.

1.2.1.2. Програмні модулі:

- графічно салон екіпажу;
- функціональна модель літальних апаратів бортових систем;
- спрощена 3-мірна візуалізація простору кабіни;
- інтерактивні функціональні схеми бортових систем;
- система управління програмним модулем.

1.2.2. Комплексний тренажер

Тренажер призначений для підготовки та підготовки льотного екіпажу до виконання завдань, пов'язаних з пілотуванням та бойовим використанням літака МіГ-29, в умовах, максимально наближених до реальних (див. рис. 1.1). Тренажер також використовується як засіб контролю правильності дій льотного екіпажу при виконанні навчальних льотних завдань..



Рис. 1.1. Комплексний тренажер МІГ-29

Кабіна тренажера виконана на основі носової частини фюзеляжу реального літака і оснащена повним набором важелів, пультів дистанційного керування, бортів

і тренажерів. Система імітації тренажера може поставлятися в різних комплектаціях, зокрема реалізована (див. Рис. 1.2).

Триканальна оптична система колімації з комп'ютерним синтезом зображення поза кабіною, з кутами огляду FOV.

Шестиканальний комплекс проєкційного екрану з циліндричним екраном та комп'ютерним синтезом зображень в салоні, з кутами огляду FOV (120 ° Н x 60 ° V).

Допоміжний тренажер призначений для попередньої підготовки льотного екіпажу в режимі автономної роботи від головної кабіни, а також для імітації дій другого літака під час польотів "в парі".



Рис. 1.2. Система візуалізації

1.2.3. Навчально-тренувальний комплекс

Вперше на практиці запропоновано якісно новий підхід до розробки для військових літаків та вертольотів, призначених для підготовки льотного та інженерного персоналу у повному обсязі своїх функціональних обов'язків та відповідно до діючих програм бойової підготовки.

Новий підхід заснований на переході від виробництва окремих тренажерів до створення єдиних інтегрованих навчальних комплексів (ІНК), які можуть включати в різні комбінації автоматизовані навчальні системи, тренажери різного рівня складності, робочі місця наземних операторів (див. Рис. 1.3).

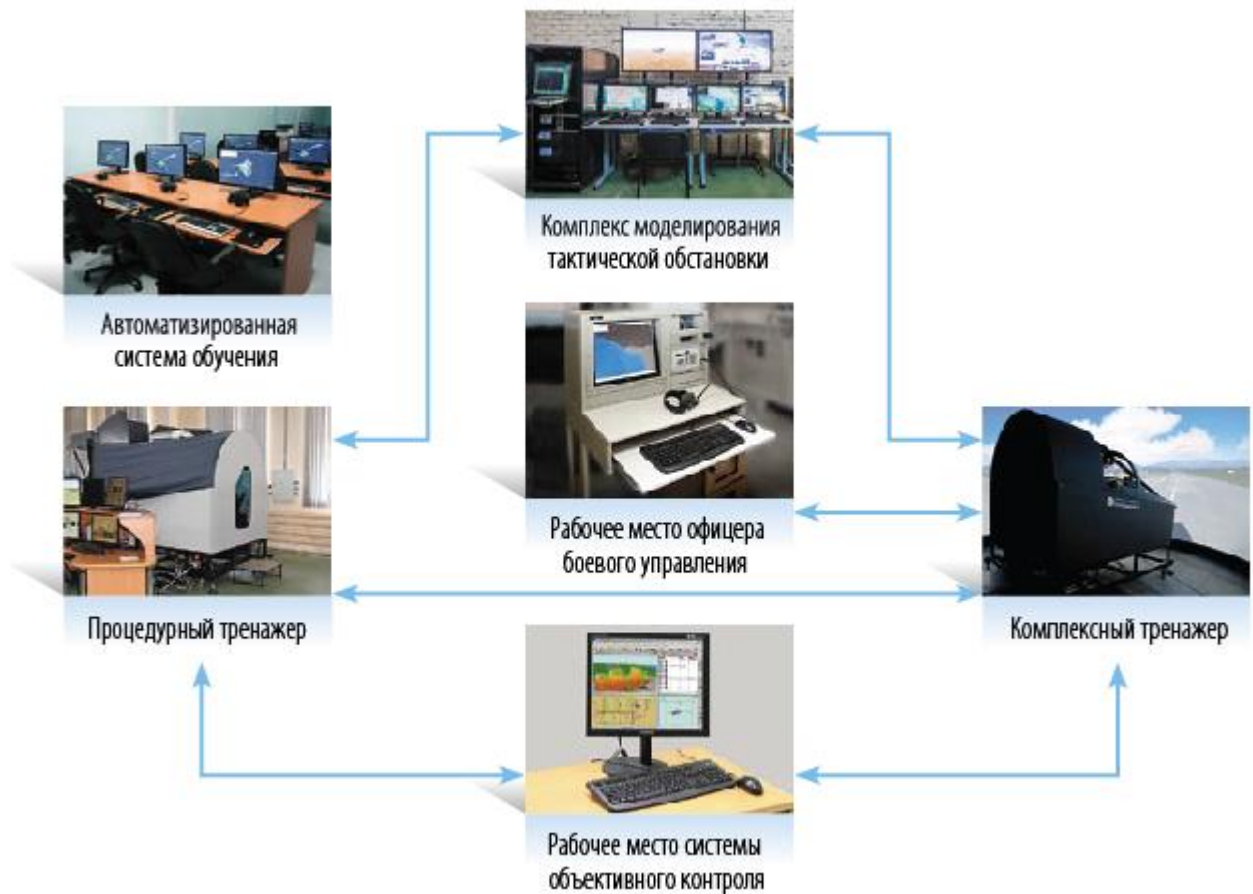


Рис. 1.3. Типова схема ІНК

Для навчання екіпажу літака, функціонально взаємопов'язаного з екіпажем літака при виконанні польотного завдання, а також набору моделювання тактичної ситуації. Наступним найскладнішим завданням є моделювання складних багатofункціональних систем тактичної підготовки, що дозволяють проводити спільну підготовку до ефективної оперативно-тактичної взаємодії різних підрозділів збройних сил, включаючи не лише індивідуальну підготовку, а й ефективну підготовку військового командування на всіх рівнях.

1.2.4. Автоматизована система навчання



Рис. 1.4. Автоматизована система навчання

У цій освітній системі реалізовані нові підходи контролю знань учня (див. Рис. 1.4). Впроваджена 3D модель кабіни, імітує роботу всіх систем літака.

1.2.4.1. АСН дозволяє:

- набуття пілотом навичок та вмінь працювати з авіаційними комплексами та системами;
- імітація льотних показників для бойового застосування на полігоні згідно із зазначеним сценарієм та вимогами КЛЕ (див. Рис. 1.5);
- розробка АСН для інженерно-технічного персоналу.

1.2.4.2. Основні можливості:

- вивчення призначення, складу, місця розташування та принципу дії всепілотного обладнання;
- проведення індивідуальних або групових занять з використанням відеопроєкційного комплексу;
- автоматизований контроль знань учнів;



Рис. 1.5. Процедурний тренажер

- здатність планувати навчальний процес викладачем та формування індивідуальних навчальних завдань;
- автоматизований дистанційний контроль за діяльністю учнів;
- формування та ведення бази даних про хід навчання та його результати;
- аналіз результатів навчання з метою прийняття інструктором рішень щодо організації навчального процесу з урахуванням індивідуальних особливостей учнів [2].

1.3. Структура і зміст навчально-методичних прийомів навчання пілотування

Навчання пілота спрямоване на формування у пілота належного реального іміджу польоту та стабільних навичок управління літаком у всьому спектрі завдань, для яких воно призначене. Динаміка розвитку навчально-методичних прийомів підготовки пілотів відповідає етапам розвитку авіації та методам пілотування. За структурою та змістом навчально-методичні прийоми поділяються на групи на основі складності пілотування:

- основи пілотування;
- пілотування всіх властивостей літака;
- пілотування в будь-якій місії.

Основні методи підготовки пілотів сформувались із народженням авіації, головним чином як методи самонавчання, оскільки перших пілотів навчали

пілотування тільки самостійним польотом. Пізніше оволодіння основами пілотування методом спроб і помилок та формулювання основних прийомів навчання пілотування, які пілоти знають, почали готувати початківців. Таким чином, швидко розвиваються методи пілотного навчання почали застосовуватися як універсальний засіб навчання та навчання, тобто як методи навчання.

У навчанні пілотів переважає предметно-оперативний метод, який включає інструктаж, демонстрацію, навчання та перевірку результатів. На основі дидактичних принципів та психологічної теорії діяльності управління процесом навчання здійснюється на рівні організації, аналізу, регулювання та стимулювання. Організація навчання включає формулювання мети навчання, вибір шляхів досягнення цілей з урахуванням закономірностей навчання, індивідуальних особливостей студента та конкретних умов. Підхід методів навчання спрямований на максимізацію можливостей учня та компенсацію несприятливих рис особистості, оскільки кожна людина формує свій, індивідуальний стиль роботи, завдяки типологічним особливостям нервової системи, які по-різному проявляються в конкретних умовах пілотування

Аналіз та регулювання забезпечують цілеспрямоване формування навичок пілотування, встановлення причин помилок, їх взаємозалежності, а також пошук шляхів зміни навичок у бажаному напрямку.

Функція стимулювання - це здатність генерувати фізичну та розумову активність учня, яка повинна проявлятися на етапі засвоєння інформації та формування сенсомоторних навичок. Дидактичними принципами застосування навчальних та методичних методів пілотного навчання є постійне спостереження, перевірка та оцінка професійних знань, умінь та навичок.

1.3.1. Прийоми навчання пілотування

Прийоми навчання пілотування ґрунтуються на наступних методах:

- показ і інструктування;
- тренування і самостійне виконання дій;
- перевірка результатів і визначення якості навчання.

1.3.2. Показ і інструктування вирішується наступними прийомами:

- інструктування перед виконанням польотів;
- вивчення схеми польотного завдання;
- вивчення дій по ділянках пілотування;
- показ дій при пілотуванні на різних ділянках;
- показ дій в особливих ситуаціях у польоті.

1.3.3. Тренування забезпечується наступними прийомами:

- тренування на тренажері під наглядом інструктора;
- пілотування літака під наглядом інструктора;
- самостійне пілотування літака.

1.3.4. Перевірка результатів і визначення якості навчання:

- спостереження за діями пілота на тренажері;
- спостереження за діяльністю пілота при пілотуванні;
- реєстрація параметрів при діяльності на тренажері;
- реєстрація параметрів при пілотуванні;
- аналіз результатів тренування на тренажері;
- аналіз результатів пілотування літака;
- визначення готовності виконання польотного завдання;
- визначення якості пілотування і виконання польоту;

- формування думки про можливість допуску пілота до відпрацювання наступного етапу льотної підготовки;
- формування думки про кваліфікацію пілота [3].

1.4. Порядок проведення підготовки на тренажерах

Програма періодичної підготовки пілотів на комплексному пілотажному тренажері (FSTD) передбачає дві підготовки та перевірки пілота на рік. Процедура організації та проходження періодичних тренувань на тренажері виконується за такою схемою (див. Рис. 1.6) (зазвичай):

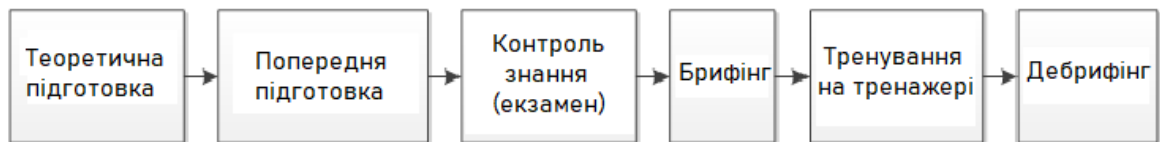


Рис. 1.6. Процедура організації та проходження періодичної підготовки на тренажері

Завершенням циклу підготовки елементів (теорія - іспит - навчання - тест) є місяць тестування на тренажері. В процесі навчання використовуються аеродроми (як правило), які використовуються оператором як пункти призначення та запасні аеродроми, включаючи запасні вздовж маршруту. У процесі теоретичної підготовки пілот самостійно вивчає матеріали для підготовки до контролю знань (іспит), перелік вправ, маневрів та відмов, які включені в сценарій навчання (навчання / випробування), аеродроми, на яких буде проводитися навчання, повторює стандартні процедури, правила експлуатації в надзвичайних ситуаціях, правила взаємодії з кабінным екіпажем. Кожному тренуванню передують попередній тренінг, який проводить інструктор.

Попередня підготовка пілота до тренувань на тренажері передбачає (зазвичай):

- вивчення навчальної програми та посібників, необхідних для навчання та тренувань пілотів на тренажері;
- вивчення аеродромів, на яких проводиться навчання;

- вивчення маршрутів польотів, на яких проводиться навчання;
- вивчення особливостей техніки пілотування, експлуатації авіаційних систем та порядку взаємодії членів екіпажу в аваріях на польоті на всіх етапах її реалізації, щодо поточного сценарію навчання (розподіл завдань);
- аналіз фактичних даних польотної інформації пілота;
- огляд характерних помилок під час навчання.

Контроль знань (іспит) здійснює інструктор (тестувальник). Контроль готовності здійснюється відповідно до теми поточного сценарію, після складання контролю знань (іспиту). Контроль готовності здійснюється інструкторами, допущеними до тренувань на тренажері, з результатами у формі контролю готовності (зазвичай).

Умови попередньої підготовки, контролю знань (іспиту) та контролю готовності встановлюються оператором-пілотом, повторно проходять попередню підготовку, контроль знань (іспит) та контроль готовності на випадок, якщо він з якихось причин, несправність тренажера, хвороба тощо) втратив право виконувати рейси, і його потрібно перенести на наступний навчальний період, встановлений оператором.

Пілот, призначений членом екіпажу підтримки для навчання та випробувань на тренажері, звільняється від проходження теоретичної підготовки, попередньої підготовки, контролю знань (іспиту) та контролю готовності, зазначених на схемі (див. Рис. 1.6) [4].

1.5. Висновки розділу

Процедурний тренажер кабіни екіпажу: моделювання відкритої кабіни екіпажу з встановленим пілотажним обладнанням, основними бортовими системами, повномасштабними копіями приладів, панелями приладів, органами управління, включаючи все обладнання та програмне забезпечення, необхідне для відтворення руху та управління літаком на земля. політ.

Завершенням циклу підготовки елементів (теорія - іспит - навчання - тест) є місяць тестування на тренажері. В процесі навчання використовуються аеродроми (як правило), які використовуються оператором як пункти призначення та запасні аеродроми, включаючи запасні вздовж маршруту.

Навчання пілота спрямоване на формування у пілота належного реального іміджу польоту та стабільних навичок управління літаком у всьому спектрі завдань, для яких воно призначене. Динаміка розвитку навчально-методичних прийомів підготовки пілотів відповідає етапам розвитку авіації та методам пілотування.

РОЗДІЛ 2.

СКЛАД ТА КОНСТРУКЦІЯ РОБОЧИХ МІСЦЬ ІНСТРУКТОРІВ АВІАЦІЙНОГО ТРЕНАЖЕРУ. ВИМОГИ ДО АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Авіаційний тренажер призначений для підготовки курсантів та пілотів до пілотування, навігації, експлуатації повітряно-десантних систем, дій у разі відмов різних повітряно-десантних систем та в особливих випадках згідно з керівництвом з експлуатації польотів повітряних суден (КЛЕ). Навчання забезпечує формування, вдосконалення та підтримку курсантів та пілотів стійких навичок управління літаками в різних умовах.

2.1. Структурний склад системи авіаційного тренажеру та функції її окремих елементів

Склад інтегрованого авіаційного тренажера (див. Рис. 2.1):

- симулятори кабіни;
- система мобільності;
- система візуалізації кабіни;
- комп'ютерний пілотажний імітатор літаків;
- автоматизоване робоче місце інструкторів;
- програмний комплекс тренажера;
- локальна мережа;

Кафедра КІТ (47)				НАУ 20 09 38 000 ПЗ			
		Підпис	Дата				
Разраб	Джуринський К.А.			СКЛАД ТА КОНСТРУКЦІЯ РОБОЧИХ МІСЦЬ ІНСТРУКТОРІВ АВІАЦІЙНОГО ТРЕНАЖЕРУ. ЗАСОБІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	Літера	Аркуш	Аркушів
Керівник	Моржов В.І.					33	20
Н. Контр.	Райчев І.Е.				УС-211М		122

- обладнання для інструктажу та аналізу польотів на тренажері;
- система електропостачання;
- експлуатаційна документація.

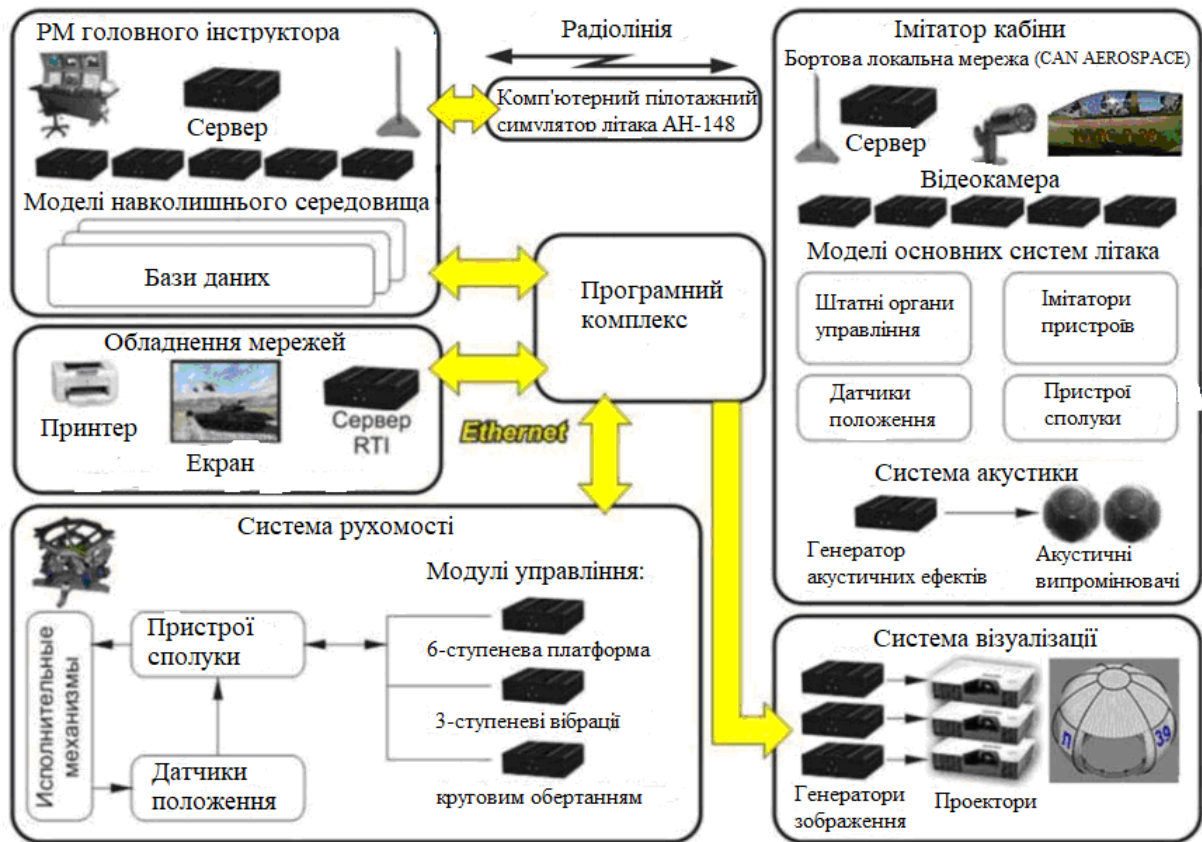


Рис. 2.1. Структурна схема елементів тренажеру та їхня взаємодія

а) Тренажер кабіни - це копія кабіни фюзеляжу літака в масштабі 1: 1 (див. рис. 2.2).

Симулятори передньої та задньої частини кабіни:

- засоби управління повітряними суднами повністю узгоджуються з елементами керування фактичним повітряним судном;
- система моделювання навантаження органів управління літаком дозволяє відтворювати нелінійні зусилля на рукоятці управління та на педалях залежно від аеродинамічних сил, що діють на ступки, кермо напрямку та висоти при всіх режимах польоту літака;

- симулятори пристроїв забезпечують відображення їх поточного стану, що визначається математичними моделями роботи відповідних бортових систем та двигуна залежно від положення органів управління та режиму польоту [5].



Рис. 2.2. Імітатор кабіни літака АН-140

б) Система рухливості:

Система рухливості призначена для імітації ефектів прискорювача, які впливають на пілотів реального літака під час дії зльоту та посадки, а також в різних повітряних умовах (див. Рис. 2.3).

Тип приводів - електромеханічний.

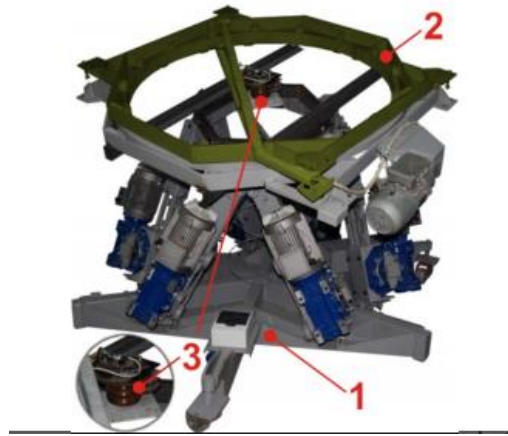


Рис. 2.3. Система рухомості

1	Шестиступенева електромеханічна динамічна платформа
2	Контактний пристрій, що обертається для передачі електроживлення
3	Триступенева вібраційна платформа
4	Проблисковий сигнал працюючої системи рухомості

в) Система візуалізації простору кабіни

Тренажер використовує проекційну систему для візуалізації простору кабіни, яка забезпечує можливість візуального спостереження за наземними та повітряними цілями, задимленням, пожежами тощо (див. Рис. 2.4).

Пристрої:

- сферичний екран;
- ширококутні проектори;
- металеві монтажні ферми з ескалаторами для кріплення.



Рис. 2.4. Система візуалізації

Характеристики:

Роздільна здатність пікселів на канал	Не менше 1024 x 768
Частота зміни кадрів, кадр / сек	Щонайменше 30
Сектор проектування (горизонт. X верт.), град	Не менш 225 x 70

г) Автоматизоване робоче місце інструкторів



Рис. 2.5. Робоче місце інструктора

1	Стіл	1 шт.
2	Клавіатура	1 шт.
3	Оптичний маніпулятор	1 шт.
4	Сенсорний монітор	6 шт.
5	Блок безперебійного живлення	1 шт.
6	Гарнітура для переговорів між інструктором та пілотом	1 шт.
7	Лазерний принтер	1 шт.
8	Системний блок	6 шт.
9	Сервер з загальним та спеціальним ПЗ	2 шт.

Екрани сучасних моніторів можуть не тільки відображати зображення, але й дозволяють взаємодіяти з пристроєм за допомогою датчиків (див. Рис. 2.5). Сьогодні сенсорні екрани широко використовуються в мобільних пристроях, програвачах, камерах та відеокамерах тощо. Таким чином, на кожному з перерахованих пристроїв може застосовуватися той чи інший тип сенсорного екрану. В даний час розроблено

кілька типів сенсорних панелей, і відповідно кожен з них має свої переваги та недоліки.

Технологія резистивних сенсорних екранів набула найбільшого поширення серед авіаційної промисловості завдяки простоті технології та низькій собівартості виробництва. Резистивний екран являє собою РК-дисплей, який накладається на дві прозорі пластини, розділені шаром діелектрика. Верхня пластина гнучка, оскільки натискається користувачем, нижня жорстко фіксується на екрані (див. Рис. 2.6). Провідники прикладаються до поверхні, зверненої один до одного.

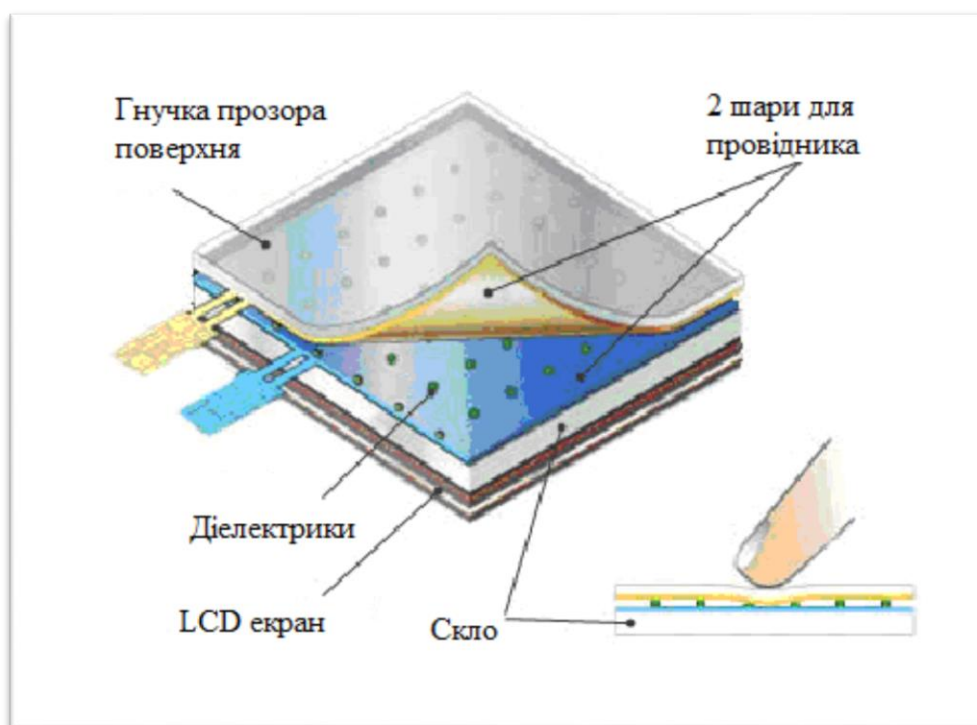


Рис. 2.6. Технологія резистивних сенсорних екранів

Мікроконтролер послідовно подає напругу на електроди верхньої та нижньої пластин. При натисканні на екран гнучкий верхній шар згинається, а його внутрішній шар проводить на поверхні нижнього провідного шару, змінюючи тим самим опір всієї системи. Зміна опору реєструється мікроконтролером і таким чином визначаються координати точки дотику.

г) Програмный комплекс тренажера

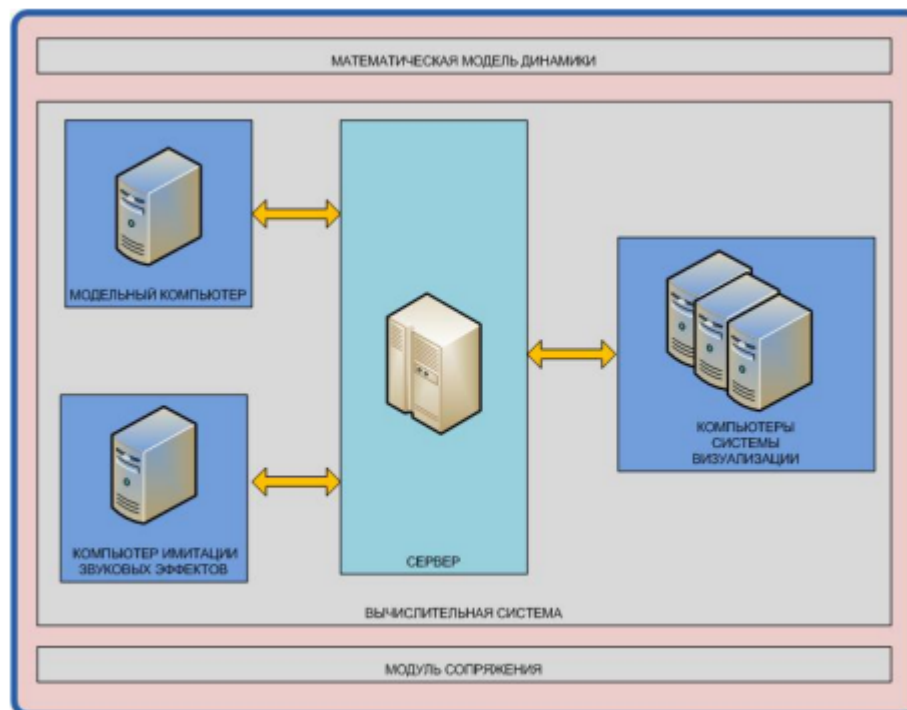


Рис. 2.7. Структура программного комплекса

Імітація динаміки літака (на землі та в повітрі) здійснюється в режимі реального часу, як в робочому діапазоні висот і швидкості польоту, так і в спеціальних режимах польоту. Аеродинаміка літального апарату поблизу землі розраховується з урахуванням особливостей рельєфу конкретної ділянки земної поверхні. Зовнішня підвіска (на пілонах) моделюється від стану реакції ПС до способу поділу вантажу та реакції на зміну ваги ПС. Зовнішні збурення вітру та стан атмосфери відповідно до прийнятого стандарту встановлені та правильно враховані. Моделювання роботи основних систем літального апарату з можливістю контролю технічних несправностей окремих вузлів, пристроїв та систем (див. Рис. 2.7).

Математична модель динаміки польоту забезпечує:

- адекватне моделювання поведінки літака в режимі реального часу як у звичайному режимі, так і в спеціальних режимах польоту (вихід з ладу

двигуна, імпульсний стрибок двигуна, несправність шасі, вихід з ладу головного генератора тощо).

- моделювання зовнішньої підвіски (на пілонах) від стану реакції ПС до методу скидання (стрільби) вантажу та реакції на зміну ваги ПС.
- правильний облік зовнішніх збурень вітру та стану атмосфери відповідно до прийнятого стандарту.
- імітаційне моделювання роботи основних систем літального апарату з можливістю контролю технічних несправностей окремих вузлів, пристроїв та систем [6-7].

2.2. Моделювання подачі інформації на робоче місце інструктора

Вертикальна передня панель в кабіні, призначена для розміщення технічних засобів відображення інформації - екранів, індикаторів, покажчиків, щитів, сигнальних ламп, а також елементів керування системами та блоками літального апарату: ручками, тумблерами, кнопками тощо. бути твердим виробом або складатися з панелей та пультів дистанційного керування. Термін "панель управління" зазвичай не застосовується до бічних, верхніх, нижніх, середніх і задніх панелей та консолей в кабіні.

Панель управління зазвичай робиться навісною, щоб забезпечити доступ до приладів та арматури для технічного обслуговування. Крім того, панелі управління часто встановлюються на амортизаторах, щоб зменшити вібраційні та ударні навантаження, які впливають на точні прилади та електроніку.

2.3. Опис панелі приборів потоків інформації



Рис. 2.8. Панель приборів

а	командно-пілотажний індикатор
б	багатофункціональний індикатор
в	комплексний індикатор систем і сигналізації
г	багатофункціональний пульт управління
г	пристрій управління курсором
д	комплексний пульт управління радіотехнічними системами
е	пілотажний прилад комбінований резервний
є	авіагоризонт
ж	навігаційно-посадковий індикатор
з	пульт управління системи автоматичного управління
и	пульт управління індикацією
і	малогабаритний годинник авіаційний

а) Командно-керуючий індикатор (ККІ), також командно-адміністративний пристрій (КАП) або пілотажно-командний пристрій (ПКП) - багатофункціональний авіаційний пристрій, що відображає дані про просторове положення (повітряний горизонт), а також команди системи управління траєкторією (STU). Відповідно до ГОСТ 22686-85 позначення «контрольно-пропускний пункт» та «контрольно-пропускний пункт» не допускаються, але вони зустрічаються в літературі, розмовній мові, а також у позначеннях пристроїв (див. Рис. 2.8).

б) Багатофункціональний індикатор / дисплей - загальний термін багаторежимного пристрою для відображення різноманітної службової графіки та телеінформації в кабіні літака, виданий бортовими системами та датчиками. Пілот (оператор) може перемикає режими відображення інформації як автоматично, так і вручну. Однорежимні пристрої зазвичай мають більш конкретні назви, наприклад, «екран тактичної ситуації».

Незважаючи на різні фізичні принципи побудови, термін "BFI" зазвичай застосовується до кольорового рідкокристалічного дисплея.

Екран BFI можна доповнити кнопками (розташованими, найчастіше, вздовж контуру цього екрану).

Висновок інформації про BFI здійснюється у такій формі, що дозволяє екіпажу виконувати льотні та наземні завдання, а також контролювати режими роботи бортових систем.

в) Інтегрована електронна система індикації та сигналізації, яка є частиною сучасних прицільних та навігаційних систем, у зручному форматі надає екіпажу ПС повну інформацію, необхідну для пілотування та навігації.

г) Багатофункціональна панель управління - це пристрій для введення інформації та взаємодії пілота з системами літальних апаратів.

ґ) Пристрій управління курсором дозволяє користувачеві безпосередньо контактувати з екраном. Важіль - це електронний пристрій, який при русі по екрану визначає відповідний рух курсору по екрану.

д) Інтегрований пульт дистанційного керування радіоприладами призначений для ручного управління радіозасобами навігації, посадки та зв'язку, а також для

прийому, обробки та передачі сигналів управління радіозасобами від комп'ютерної системи літальних апаратів.

е) Комбінований резервний пілотаж - це конструкція, що складається з окремих функціональних блоків, пов'язаних між собою механічно та електрично. Прилад виконує функцію вимірювання, обчислення, індикації параметрів висоти та швидкості та передачі їх на бортові системи.

є) Авіаційний пристрій для визначення справжнього горизонту та вимірювання поперечного та поздовжнього кренів літака. Спостерігаючи положення мініатюрного силуету літака, зображеного в масштабі повітряного горизонту, пілот визначає положення свого літального апарату щодо справжнього горизонту.

ж) Індикатор навігації та посадки призначений для індикації параметрів навігації та посадки на сигналах, що надходять від комплексу бортового обладнання.

з) Панель управління автоматичної системи управління призначена для управління автоматичними системами управління (АСУ) з кабіни пілота [8].

2.4. Пристрої вводу

Широкий клас технічних засобів, призначених для введення даних у комп'ютер. До пристроїв цього типу належать: клавіатури, графічні планшети, сенсорні екрани, пристрої координат або маніпуляторів, сканери, зчитувачі (магнітні, оптичні та електричні) міток та штрих-кодів, пристрої мовного введення та інші. Раніше пристрої також використовували для зчитування даних із перфокасеток та перфокарт. Основний набір ПК різної конфігурації включає клавіатуру та монітор із сенсорним екраном.

2.4.1. Клавіатура

Набір клавіш, розташованих у певному порядку, які використовуються для введення та редагування даних, а також управління курсором та людсько-машинних операцій:

- розширена клавіатура - клавіатура, яка містить 101 або більше букв і цифр, функціональних та керуючих клавіш.
- буквено-цифрова клавіатура - частина клавіатури, призначена для введення даних у комп'ютер у вигляді літер, цифр та інших символів.
- функціональна клавіатура - частина клавіатури, призначена для ініціювання виконання певних функцій та / або операцій системою. Елементами функціональної клавіатури є функціональні клавіші, окреме або спільне використання яких з іншими (у тому числі - буквено-цифровими клавішами) забезпечує дії, визначені активною (тобто активною в даний час) програмою.

Варіанти стандартів для клавіатури:

- QWERTY - вказує розташування англійських клавіш за замовчуванням на друкарській машинці або клавіатурі ПК. Відповідає порядку перших шести клавіш верхнього ряду алфавітної клавіатури. У не англомовних країнах може бути використана інша більш зручна розкладка клавіатури, наприклад AZERTY або QWERTZ.
- клавіатура Dvorak - альтернативна розкладка клавіш QWERTY, у якій найбільш часто використовувані клавіші центруються для прискорення операцій набору тексту.

2.4.2. Миша

Маніпуляторно-координатний пристрій ЕОМ. Є доповненням до клавіатури і необхідною частиною обладнання при користуванні графічним інтерфейсом.

- клацання [click] - дія, яка пов'язана з швидким натисканням і відпусканням клавіші (кнопки) миші для виконання відповідної функції, наприклад, - установці курсора на обраній позиції екрану.
- подвійне клацання [double click] - дія, яка пов'язана з дворазовим швидким натисканням і відпусканням клавіші (кнопки) миші для виконання відповідної функції, наприклад, - виділення на екрані слова.

2.4.3. Взаємодії за допомогою сенсорного вводу

Навчальна програма розроблена з урахуванням того, що основним методом введення буде сенсорне введення. Однак слід зазначити, що оптимізований для користувача інтерфейс користувача не завжди буде кращим за традиційний інтерфейс. І те, і інше має свої переваги та недоліки у технології та застосуванні. При переході на користувальницький інтерфейс Touch важливо розуміти основні відмінності між сенсорним екраном, сенсорною панеллю, ручкою, планшетом та введенням з клавіатури. Важливі API: UI.Xaml.Input, UI.Core, Devices.Input [9].

Багато пристроїв оснащені мультисенсорними екранами, які підтримують використання одного або декількох пальців (або сенсорних контактів) для введення даних. Сенсорні контакти та їх рухи інтерпретуються як жести дотику та маніпуляції для підтримки різних типів взаємодії з користувачем. Програма повинна включати ряд різних механізмів обробки сенсорного введення.

Для сенсорної взаємодії потрібні 3 речі:

- сенсорний екран.
- прямий дотик (або тісна взаємодія, якщо дисплей оснащений безконтактними датчиками і підтримує виявлення наведення) одним або кількома пальцями на екран.
- переміщення контактних точок.
- вхідні дані, що виводяться сенсорним датчиком, можуть:
- інтерпретуватися як фізичний жест для безпосередньої взаємодії з одним або кількома елементами користувацького інтерфейсу (наприклад, для

- зсуву, обертання, зміни розміру або переміщення); навпаки, взаємодія з елементом через його вікно властивостей, діалогове вікно або інший користувальницький інтерфейс називається непрямю маніпуляцією;
- бути визнаним альтернативним способом введення, таким як миша чи ручка;
 - використовуватись для доповнення або зміни різних аспектів інших методів введення, таких як розмиття обведення.

Сенсорне введення, як правило, передбачає пряме маніпулювання елементом на екрані. Елемент негайно реагує на будь-який торкальний контакт і відповідно реагує на будь-яке подальше переміщення контактних контактів, включаючи видалення (табл. 2.1). Нестандартні жести сенсорного введення та взаємодії слід розробляти з особливою увагою. Вони повинні бути інтуїтивно зрозумілими, мати швидку реакцію та забезпечувати впевненість користувачів у роботі з програмою.

Традиційні пристрої введення (наприклад, миша та клавіатура) знайомі та знайомі багатьом користувачам. Вони можуть забезпечити швидку роботу, точність та тактильний зворотний зв'язок, недосяжні для сенсорного управління. Забезпечення унікальних та різноманітних процедур взаємодії для всіх пристроїв введення забезпечить підтримку якнайширшого діапазону можливостей та параметрів, завдяки чому програми навчання пілотів будуть успішними в найширшому діапазоні програм.

Таблиця 2.1

Обов'язкові вимоги, які необхідні для взаємодії з сенсорним екраном

Фактор	Взаємодії за допомогою сенсорного вводу	Взаємодія за допомогою миші, клавіатури, пера	Touchpad
Точність	Контактна поверхня кінчиків пальців більше, ніж окрема пара координат X-Y, тому збільшується ймовірність випадкового виконання команд.	Миша і перо передають точні координати X-Y.	Аналогічно миші.
	Форма контактної поверхні змінюється при русі.	Переміщення миші і пера передають точні координати X-Y. Клавіатурна фокусування визначена явно.	Аналогічно миші.
	Відсутній покажчик (курсор) для націлювання.	Покажчик миші, покажчик пера і клавіатурна фокусування дозволяють виконати націлювання.	Аналогічно миші.

	<p>Рухи кінчиків пальців є неточними, так як прямолінійне переміщення одного або декількох пальців ускладнюється через вигини суглобів кисті і одночасної участі в русі декількох суглобів.</p>	<p>Мишею або пером прямолінійний рух виконується легше, так як кисть руки пересувається на відстань, яка значно коротша, ніж переміщення курсору на екрані.</p>	<p>Аналогічно миші.</p>
<p>Анатомія людини</p>	<p>Деякі ділянки на поверхні сенсорного екрану можуть бути важкодоступні при певних положеннях пальців і самого пристрою.</p>	<p>Показчик миші або пера може досягати будь-якій частини екрану, і при цьому будь-який елемент управління повинен бути доступний з клавіатури за допомогою послідовності табуляцій.</p>	<p>Проблемою може стати розташування пальців і руки, що тримає пристрій.</p>
	<p>Об'єкти можуть затулятися одним або декількома пальцями або кистю руки. Це називається загоразивание екрану.</p>	<p>Пристрої введення непрямой дії не викликають загоразивание екрану.</p>	<p>Аналогічно миші.</p>

Стан об'єкту	Для сенсорного введення характерні два стани: сенсорна поверхня дисплея може або стикатися з рукою (активна), або ні.	Для миші, пера і клавіатури існують три стану: верхнє (неактивний), нижнє (активне) і при наведенні. Стан при наведенні дозволяє вивчити елементи призначеного для користувача інтерфейсу за допомогою спливаючих підказок. Також воно допомагає зрозуміти, які об'єкти доступні для взаємодії, і виконати	Аналогічно миші.
		націлювання.	
Складна взаємодія	Підтримується мультисенсорная технологія: множинні точки введення (кінчиками пальців) на поверхні сенсорного екрану.	Підтримується єдина точка введення.	Аналогічно сенсорному введенню.
	Підтримується пряме маніпулювання об'єктами за допомогою таких операцій, як дотик, перетягування, ковзання, зміна розміру і обертання.	Пряма маніпуляція не підтримується, так як миша, перо і клавіатура є пристроями введення непрямої дії.	Аналогічно миші.

2.5. Обов'язкові вимоги, які необхідні для взаємодії з сенсорним екраном

Правильний візуальний зворотний зв'язок під час взаємодії з програмою допомагає користувачам вчитися, вчитися та адаптуватися до того, як їх взаємодія інтерпретується самим додатком та платформою, для якої програма була створена. Візуальний зворотний зв'язок може показати успішну взаємодію, визначити стан системи, покращити чутливість системи управління, зменшити кількість помилок, допомогти користувачам зрозуміти систему та пристрій введення та заохотити взаємодію з системою.

Візуальний зворотний зв'язок особливо важливий, коли сенсорний ввід використовується для операцій, які вимагають правильного та точного позиціонування. Зворотній зв'язок відображається щоразу, коли виявляються дотики, що дозволяє користувачеві зрозуміти правила взаємодії з програмою.

а) Постановка цілі

Для оптимізації націлювання ми використовуємо:

- розміри сенсорних цілей;
- форма області дотику.

Найбільш вірогідний об'єкт цілі визначається по всій зоні контакту пальця.

б) Точність

Для обробки недбалих дотиків використовуються:

- точки кріплення, які допоможуть вам залишатися в потрібному місці під час взаємодії з елементами програми;
- Направляючі рейки, що полегшують вертикальне або горизонтальне зміщення, навіть якщо рука трохи рухається по дузі.

в) Часові властивості

Безпосереднє маніпулювання забезпечує низку переваг перед взаємодіями, заснованими на часі:

- Миттєва візуалізація відгуків підвищує інтерес користувачів та зміцнює впевненість у собі;

- Прямі маніпуляції можна зупинити - це дозволяє досліджувати систему без зайвого ризику. Користувачі можуть легко скасувати свої дії. Для цього використовуються логічні та інтуїтивно зрозумілі прийоми;
- Взаємодії, що безпосередньо впливають на об'єкти та імітують те, що відбувається насправді, легше зрозуміти та запам'ятати;
- взаємодія на основі часу може бути важкою у виконанні, оскільки користувачі повинні досягти невидимих (неочевидних) порогів;
- Крім того, рекомендується дотримуватися наступних принципів:
- Маніпуляції не слід розрізняти за кількістю використаних пальців;
- Взаємодія повинна підтримувати складні маніпуляції. Наприклад, користувач може проводити пальцем, щоб змінити розмір вмісту, і одночасно проводити пальцем, щоб переміщати його;
- Одна взаємодія повинна працювати до певного результату, незалежно від того, скільки часу потрібно для її виконання [10].

2.6. Висновки розділу

Тренування забезпечують формування, вдосконалення і підтримку у курсантів і льотчиків стійких навичок управління літаком в різних умовах.

Моделювання динаміки руху літака (по землі і в повітрі) здійснюється в реальному масштабі часу, як в експлуатаційному діапазоні висот і швидкостей польоту, так і на особливих режимах польоту. Розраховується аеродинаміка літака поблизу землі з урахуванням особливостей рельєфу конкретної ділянки земної поверхні. Моделюється зовнішня підвіска (на пілонах) з умови реакції літака на спосіб відділення вантажу і реакція на зміну ваги літака. Задаються і правильно враховуються зовнішні вітрові обурення і стан атмосфери відповідно до прийнятого стандартом.

Для вводу даних у систему існує широкий клас технічних засобів, призначених для введення даних в ЕОМ. До пристроїв цього виду відносяться: клавіатура, графічні планшети, сенсорні екрани, координатні або маніпуляторні

пристрою, сканери, пристрої зчитування (магнітного, оптичного і електричного) міток і штрихового коду, пристрої мовного введення і ін. Найбільш відомі методи вводу:

- а) Клавіатура та мишка;
- б) Сенсорний дисплей.

Клавіатура - сукупність розташованих у певному порядку клавіш, які використовуються для введення і редагування даних, а також управління курсором і виконання людино-машинних операцій;

Миша - маніпуляторно-координатний пристрій ЕОМ. Є доповненням до клавіатури і необхідною частиною обладнання при користуванні графічним інтерфейсом.

Кожен з методів вводу має свої переваги та недоліки. Головне є те, що принцип взаємодії приладу вводу та операційної системи однаковий що для клавіатури та миші, що для сенсорного дисплею.

РОЗДІЛ 3.

АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ДЛЯ ТРЕНУВАННЯ ПІЛОТІВ НА АВІАЦІЙНОМУ ТРЕНАЖЕРІ ІЗ ЗАСТОСУВАННЯМ ТЕХНОЛОГІЙ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ

Визначивши основні вимоги до програмного продукту та задачі, які повинне виконувати програмне забезпечення для тренування пілотів із застосуванням VR-технологій, створимо продукт, який буде задовольняти високу ефективність при тренуванні. Перш за все, треба визначити, які конкретні задачі та сценарії будуть виконуватися за допомогою нашого ПЗ і об'єднати задачі за спільними ознаками у групу задач. На рисунку 3.1 зображено діаграму прецедентів, яка ілюструє варіанти використання ПЗ, групуючи їх за спільними ознаками.

Кафедра КІТ (47)				НАУ 20 09 38 000 ПЗ			
		Підпис	Дата		Літера	Аркуш	Аркушів
Разраб	Джуринський К.А.			АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ДЛЯ ТРЕНУВАННЯ ПІЛОТІВ НА АВІАЦІЙНОМУ ТРЕНАЖЕРІ		54	16
Керівник	Моржов В.І.						
Н. Контр.	Райчев І.Е.					УС-211М	122

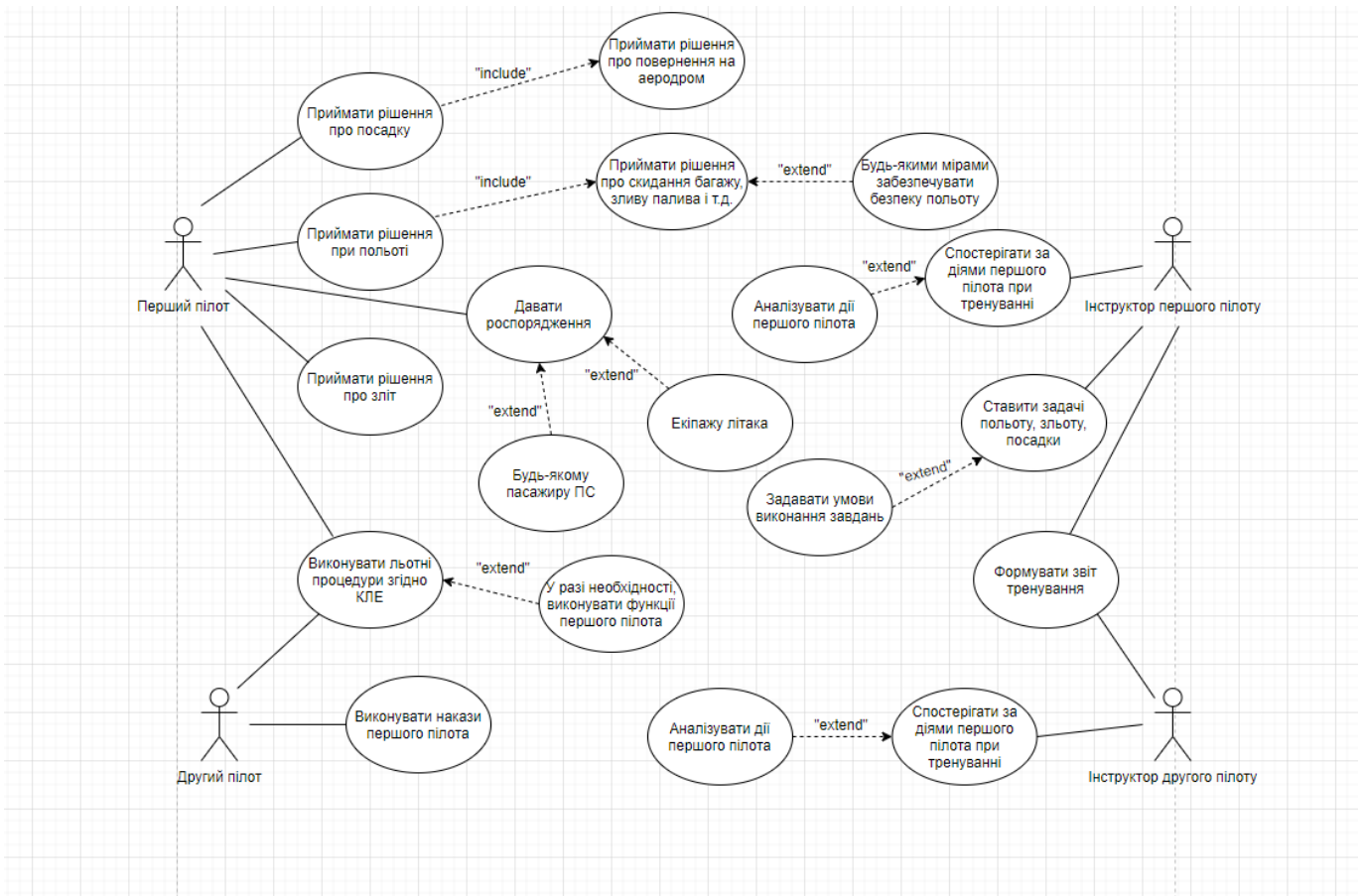


Рис. 3.1. Діаграма прецедентів тренування на тренажері

Основними діючими особами у використанні даного програмного продукту є перший пілот, другий пілот, інструктор першого та інструктор другого пілотів. Тобто, є виконавці поставлених задач (перший та другий пілоти), а є особи (інструктори) які керують, контролюють хід навчання та перевіряють результат роботи пілотів-учнів.

Кожна з діючих особин виконує виключно свій перелік функцій або контролює виконання задач і, при необхідності, виправляє помилки, які виникають у процесі навчання.

Далі буде наведений опис процесу роботи програмного модуля, як складової частини системи авіаційного тренажеру.

3.1. Багаторівнева архітектура

Програмне забезпечення для тренування пілотів на тренажері із застосуванням VR-технологій має багаторівневу архітектуру. Програмне забезпечення ділиться на підсистеми (або рівні), які розташовуються одна на одну. Кожен рівень може викликати тільки рівень на один нижче нього.

Програма складається з п'яти архітектурних рівнів: рівень об'єктної моделі, роботи з даними, рівень обчислень, графічний рівень та рівень інтерфейсу користувача, які містять в собі модулі, що розділенні за функціональною ознакою.



Рис. 3.2. Структурна схема архітектури ПЗ для тренування на тренажері

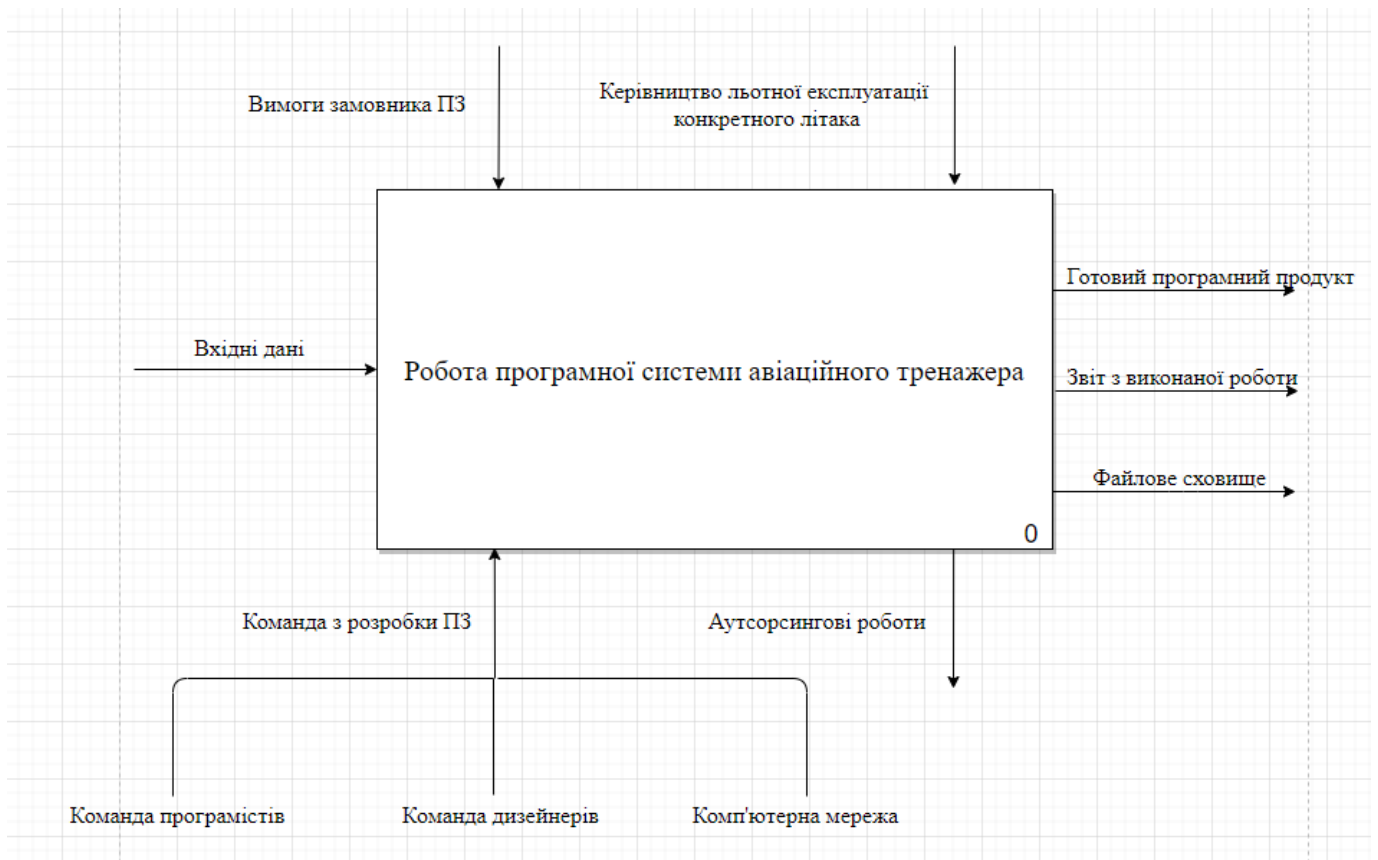


Рис. 3.3. Робота програмної системи авіаційного тренажера

Робота програмної системи авіаційного тренажера залежить від типу ПС, характеру навчання та інших факторів. Її робота керується не тільки програмним модулем, а й іншими складовими тренажера (див рис. 3.3.). У діаграмі, такі роботи були названі аутсорсинговими. Тобто ті, що виконуються поза межами програмного блоку. Але, при цьому, інші системи тренажера впливають на програмний модуль, і навпаки. Наприклад, однією з складових тренажера є система рухливості. Очевидно, її робота керується програмним модулем, і, саме від програмного модуля вона отримує інтерпретовані сигнали, що приводять її до руху. Далі опишемо взаємодії, які виконуються саме програмним модулем.

На вхід системи ми отримуємо вхідні дані (спосіб їхнього отримання опускаємо, бо він не є важливим в контексті цього дипломного проекту). Ці дані є основним об'єктом, над яким і, з яким працює програмна система. Керуючись вимогами замовника ПЗ та, звісно, керівництвом льотної експлуатації (КЛЕ) літака команда дизайнерів та програмістів реалізує завдання у коді. Код, написаний програмістами та дизайнерами – це правила, за якими об'єкти системи будуть

функціонувати та взаємодіяти між собою. Звісно, для реалізації розробникам необхідна середовище розробки – комп’ютерна мережа. Комп’ютерна мережа забезпечує збереження, цілісність, швидкий та легкий доступ і, звісно, безпеку та резервне копіювання даних.

У результаті завершення програмного циклу ми отримуємо результат роботи, який буде представлено у звіті з проведеної роботи. Звіт представлено у зрозумілій структурованій формі, для забезпечення легкості та швидкості процесу аналізу отриманих результатів навчання пілотів.

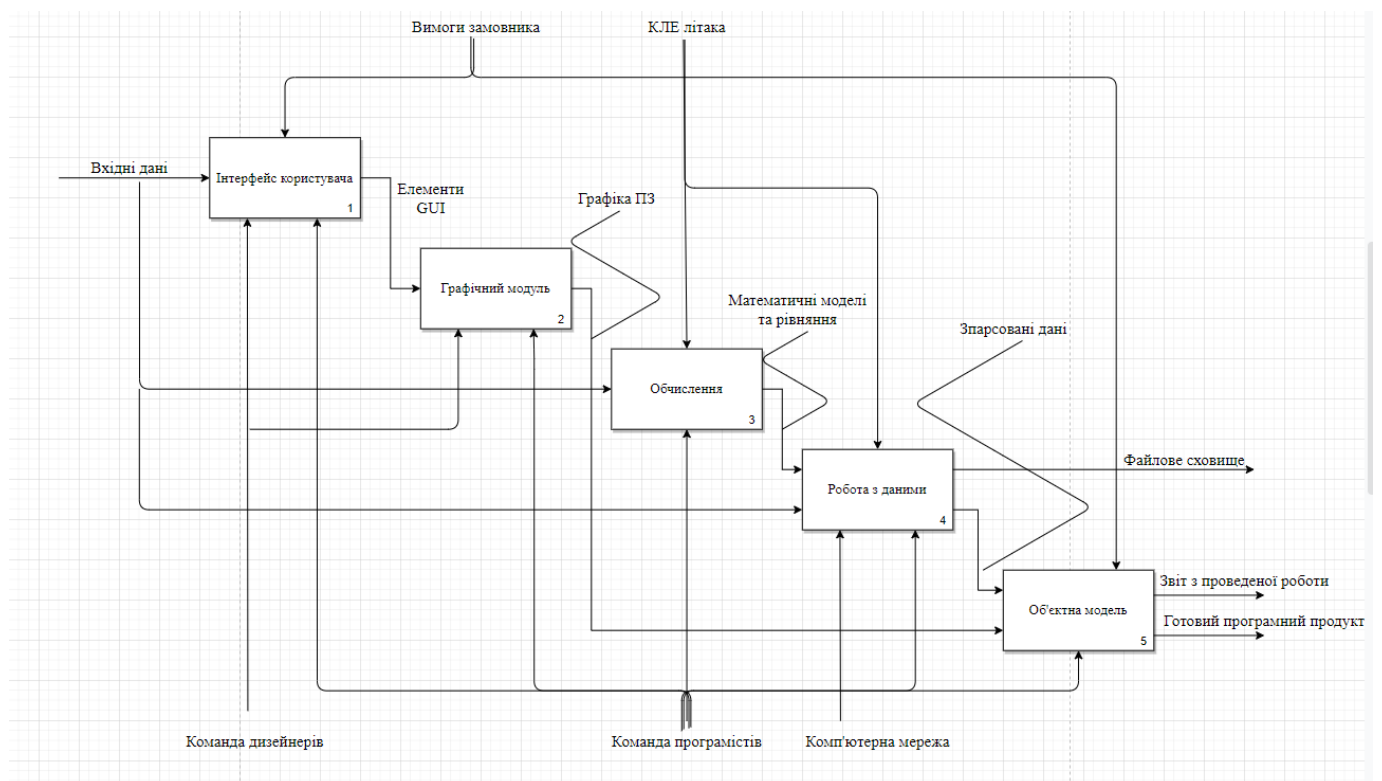


Рис. 3.4. Декомпозиція роботи програмної системи авіаційного тренажера

На рисунку показана декомпозиція роботи програмної системи авіаційного тренажера (див рис. 3.4.). Як було сказано раніше, програмна система ділиться на модулі, кожен з яких виконує свою функцію, і, певним чином взаємодіє з іншими модулями програми.

Отже, програма має такі основні модулі (опис кожного з модулів буде наданий далі):

- інтерфейс користувача

- графічний модуль
- модуль обчислення
- модуль роботи з даними
- об'єктна модель

Як показано на рисунку, виконання кожного наступного модуля (в більшості випадків) залежить від результату роботи попереднього модуля. Тобто, на виході з кожного модуля, ми отримуємо або дані, які будуть оброблюватися на наступних етапах роботи програми, або керуючі сигнали, які ініціалізують роботу наступного модуля, тобто, сигналізують про те, що свою роботу вони виконали і програмна система може продовжувати свою роботу.

3.1.1. Рівень графічного інтерфейсу користувача

Цей рівень (див рис. 3.5.) виконує декілька функцій: взаємодія із користувачем (зчитування запитів користувача, надання відповіді на запит користувача, відображення статусу системи тощо); взаємодія з рівнями, що ієрархічно знаходяться нижче (зокрема, графічний рівень та рівень обчислень). У такий спосіб рівень графічного інтерфейсу використовує конкретні можливості інших рівнів без освідомлення того, як саме буде виконаний запит, єдине, що має якесь значення для даного рівня це – коректний результат запита до нижчих рівнів на запит користувача.

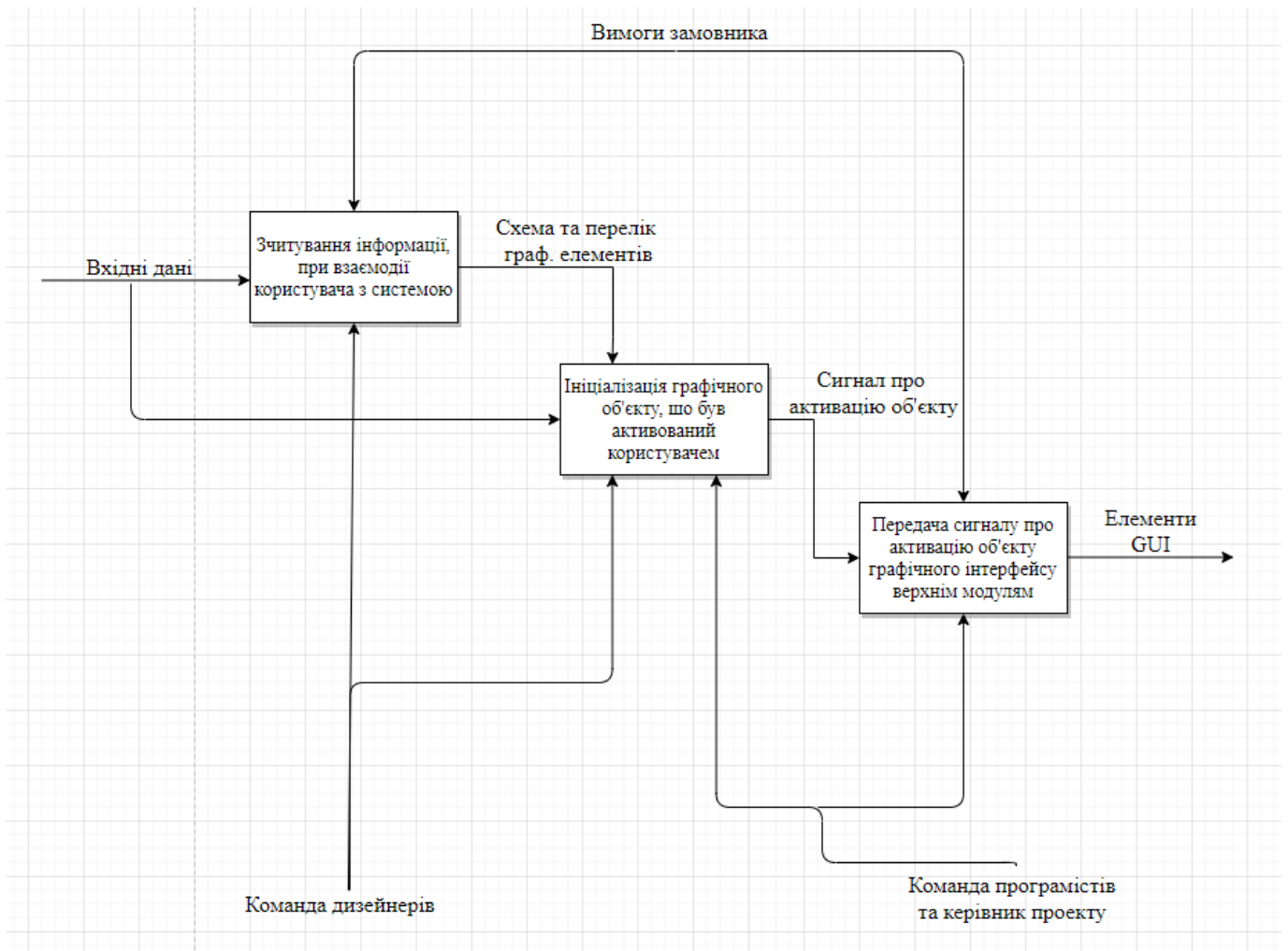


Рис. 3.5. Графічний інтерфейс користувача

Модуль графічного інтерфейсу користувача складається з таких функцій:

- зчитування інформації при взаємодії користувача з системою – функція, яка реагує на ввід інформації користувачем за допомогою пристроїв вводу таких як: миша, клавіатура, дотик на сенсорному екрані, перо, планшет, контролери і т. п.
- ініціалізація графічного об'єкту – після вводу інформації користувачем, система повинна певним відреагувати на отриманий запит. Цей процес ініціалізує об'єкт системи. Тобто, дай програмній системі сигнал до певних первинних дій (наприклад, звуковий сигнал або світловий індикатор), в залежності від характеру та об'єкту взаємодії.
- передача сигналу верхнім рівням системи – після ініціалізації первинних дій, процес передає сигнал верхнім рівням програми. Отримавши сигнал,

наступні за ієрархією модулі зчитують сигнал та виконують свої функції, які будуть описані далі.

3.1.2. Рівень графічного модулю

Графічний модуль - це програмне та/або апаратне забезпечення комп'ютера, яке формує графічне зображення, що виводиться на дисплей, використовуючи для цього апаратні ресурси відео-карти у поєднанні з драйвером пристрою для створення зображень на дисплеї.

Цей рівень (див рис. 3.6.) дозволяє програмі відображати на екрані той чи інший об'єкт, спосіб відображення на екрані та графічне зображення якого, було розраховано графічним модулем, використовуючи результати обробки вхідних даних на попередніх рівнях програмної системи. Спосіб відображення об'єкту залежить від багатьох факторів, наприклад, положення голови (фактично VR-шолому) у просторі, зовнішніх факторів, таких як погодні, швидкість та висота польоту та інші.

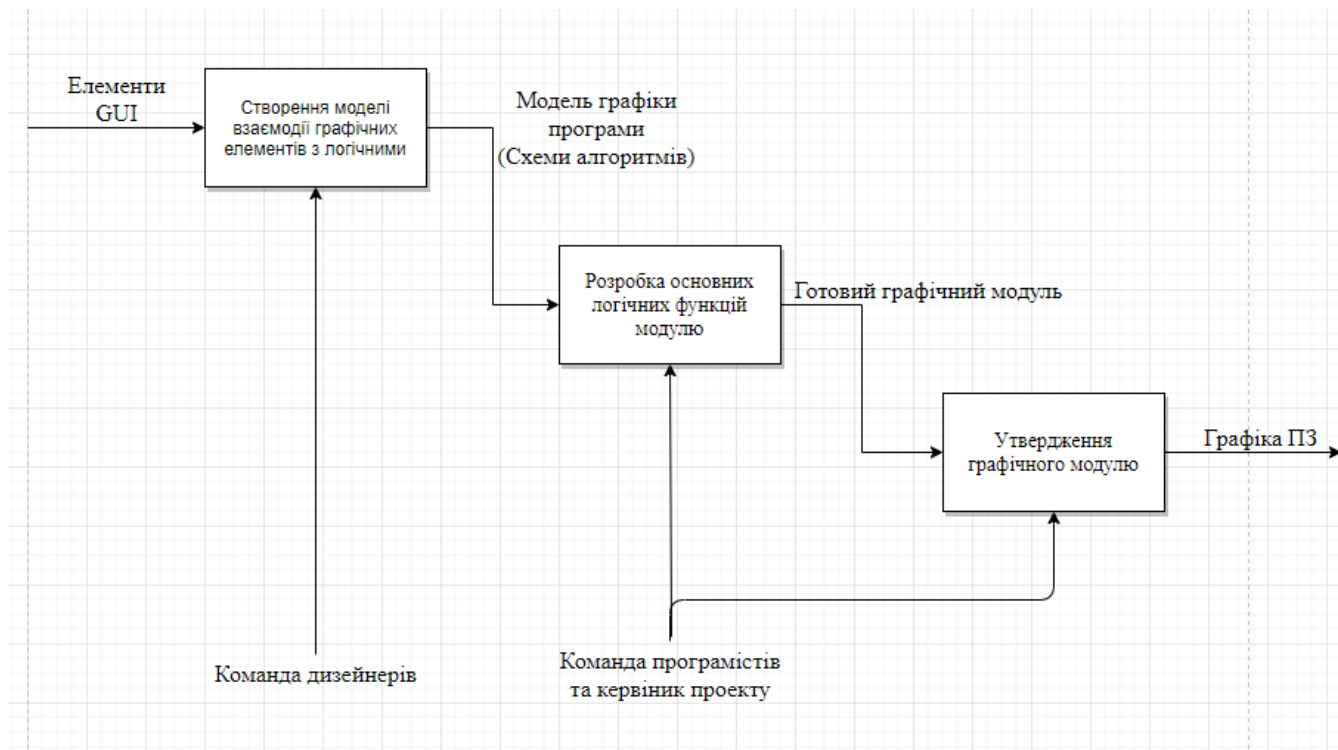


Рис. 3.6. Графічний модуль

Графічний модуль складається з таких функцій:

- створення моделі взаємодії графічних елементів з логічними – графікою можна назвати всі ті операції та події, які ми спостерігаємо наочно при виконанні програми. Зображення складається з текстур, моделей і т.д. Для того щоб користувач міг взаємодіяти з графічними об'єктами, об'єкти взаємодії треба наділити певними властивостями, щоб вони стали інтерактивними. Цей процес забезпечує інтерактивність програми.
- виконання основних логічних операцій модулю – після взаємодії з певним об'єктом, повинна відбутися певна подія або змінитися стан об'єкту. Ця функція забезпечує відповідну реакцію програмної системи на дії користувача. Наприклад, при натисканні кнопки А, відкриються двері Б.
- утвердження графічного модулю – результуючий процес, який представляє собою результат виконання двох попередніх функцій.

3.1.3. Рівень логіки обчислень

Рівень обчислень визначає призначення програмної системи і саме на цьому рівні відбувається реалізація логіки обчислень бізнес-задач, для яких створюється система. Крім того, рівень обчислень виконує зв'язування інтерфейсу, який використовує користувач, та рівні, що знаходяться нижче за ієрархією рівнів програмної архітектури (тобто, робота з даними та об'єктна модель).

Саме на цьому рівні (див рис. 3.7.) розташовується логіка роботи з даними, а саме: обробка даних про положення повітряного судна у просторі, його швидкості та висоті польоту, та інші змінні, які характеризують рух літака в повітрі і на землі.

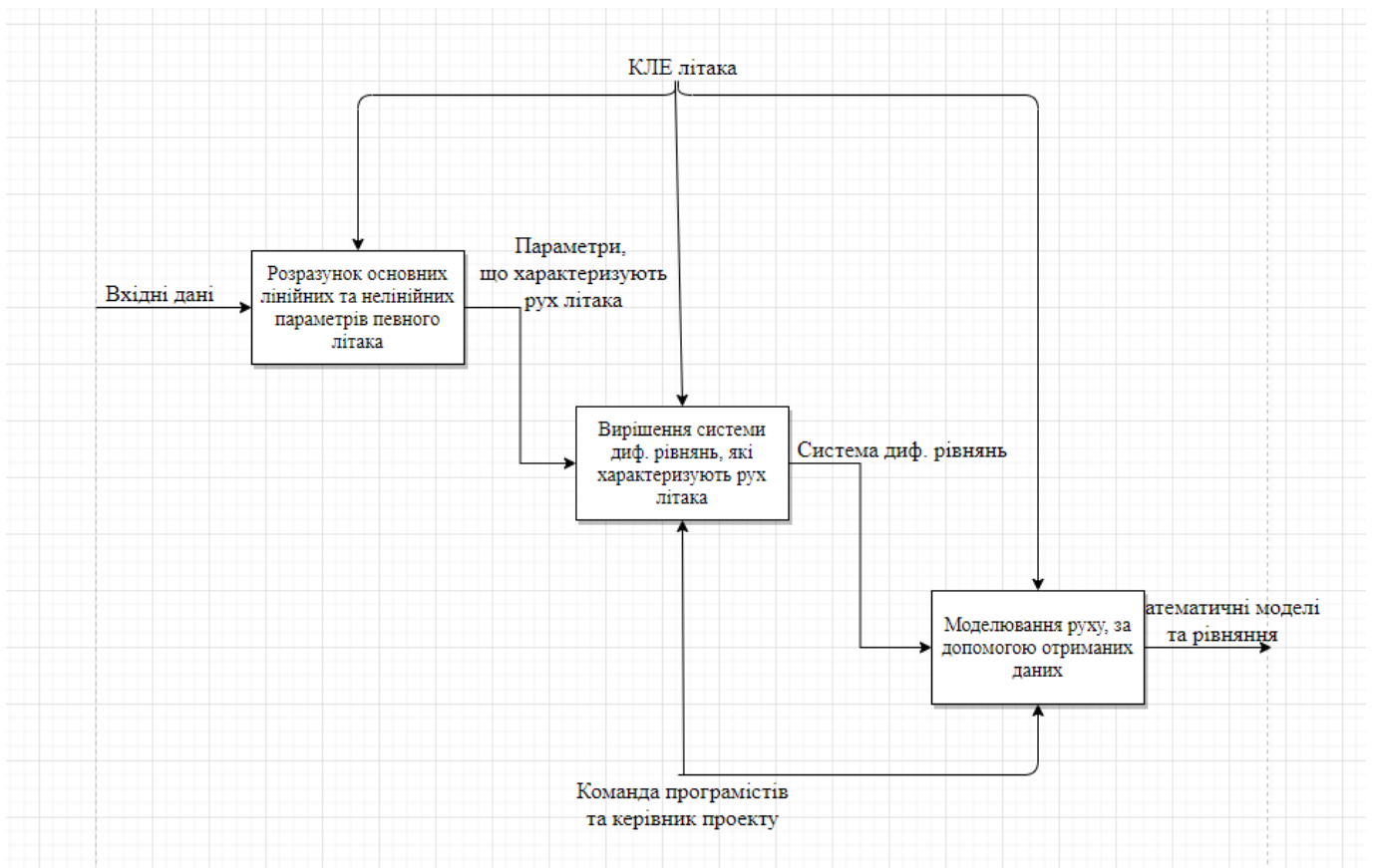


Рис. 3.7. Модуль обчислення

Модуль обчислення складається з таких функцій:

- розрахунок основних лінійних та нелінійних параметрів – рух літака і літак, як об'єкт, характеризуються певними лінійними та нелінійними параметрами. Деякі ми отримуємо у вигляді вхідних даних, а, деякі, потрібно розрахувати, виходячи з вхідних даних. Саме це завдання виконує даних процес.
- вирішення системи диференціальних рівнянь, які характеризують рух літака – очевидним є той факт, що рух будь чого можна описати математично. І, навіть, рух такої складної системи, як літак, можна описати системою диференціальних рівнянь. Основним завданням цієї функції є вирішення системи диференціальних рівнянь.
- моделювання руху на основі отриманих даних – це процес перевірки даних, які були отримані у попередньому процесі. Тобто, ця функція перевіряє

коректність отриманих результатів вирішення диференціальних рівнянь з попереднього процесу.

3.1.4. Рівень роботи з даними

Рівень доступу до даних у системі комп'ютера, або у комп'ютерній мережі - це рівень програми, який забезпечує спрощений доступ до даних, що зберігаються в постійному стані, наприклад, реляційній базі даних.

Наприклад, замість використання таких команд, як вставка, видалення та оновлення для доступу до певної таблиці в базі даних, у базі даних можна створити клас та кілька збережених процедур. Процедури будуть працювати в межах класу, де їх викликають, процедура повертає об'єкт як результат її роботи, що містить запитувані значення. Команди вставки, видалення та оновлення можуть виконуватися за допомогою простих функцій, таких як *CREATE*, *INSERT*, *UPDATE*, *DELETE* і т.д., які зберігаються на рівні доступу до даних (див рис. 3.8.).

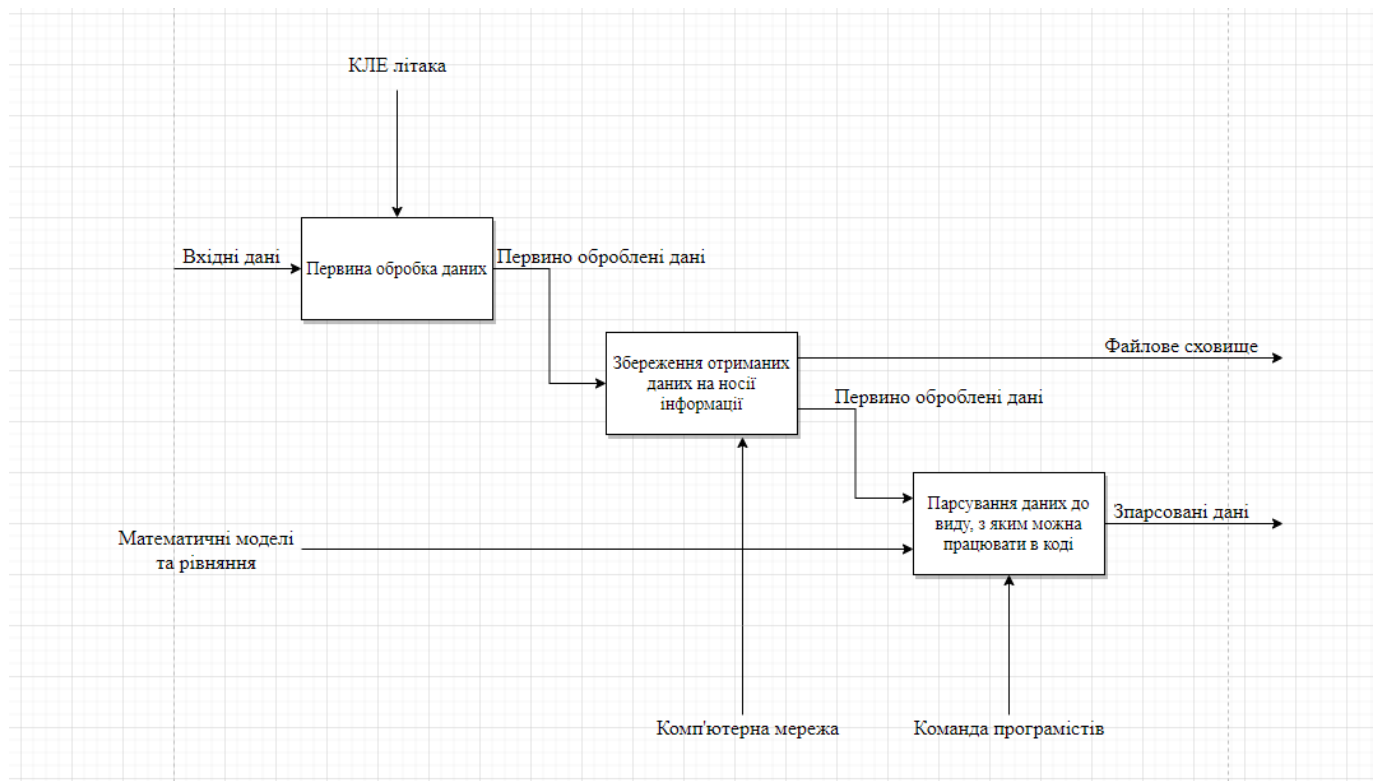


Рис. 3.8. Модуль роботи з даними

Модуль роботи з даними складається з таких функцій:

- первина обробка даних – приведення даних до формату, необхідного для коректного функціонування програми. До таких обробок можна віднести: валідацію (процес перевірки коректності даних), зміна кодування, приведення даних до певної структури (масиви, списки, словники).
- збереження отриманих даних на носії інформації – процес, який забезпечує і гарантує збереження даних на файловому сховищі (сервер, або кластер серверів). Крім збереження, дані шифруються та робиться резервна копія найважливіших даних.
- парсування даних до виду, з яким можна працювати у кодї – в залежності від функції або методу, дані повинні мати свою спеціальну форму, або представлення. Наприклад, певні функції працюють лише з числовим типом, а на вхід поступають строкові дані. Для роботи певного методу необхідно певним чином обробити ці дані, таким чином метод буде відпрацьовувати свої дії коректно.

3.1.5. Рівень об'єктного модулю

Об'єктний модуль - колекція об'єктів або класів, за допомогою яких програма може досліджувати та маніпулювати деякими конкретними частинами своєї архітектури. Іншими словами, об'єктно-орієнтований інтерфейс до якоїсь послуги чи системи (див рис. 3.9.).

Наприклад, об'єктна модель документа - це сукупність об'єктів, що представляють сторінку у веб-браузері, що використовується програмами сценаріїв для вивчення та динамічної зміни сторінки. Існує об'єктна модель Microsoft Excel для управління Microsoft Excel з іншої програми.

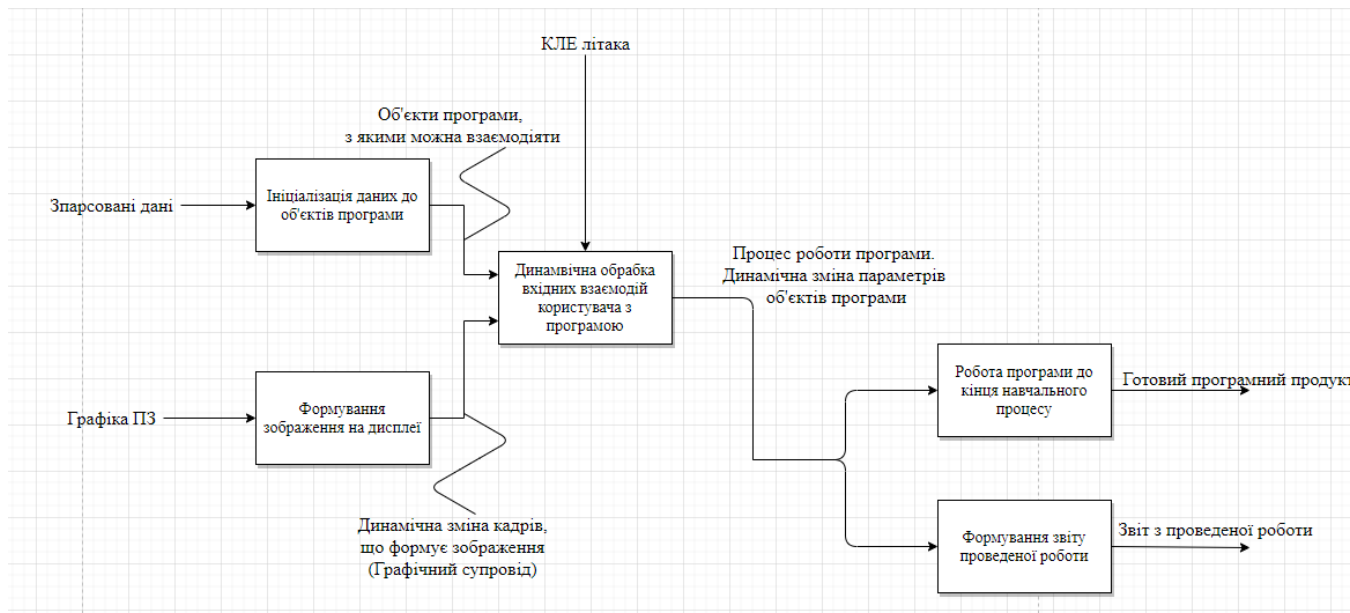


Рис. 3.9. Об'єктний модуль

Об'єктний модуль формують такі важливі функції:

- формування зображення на дисплеї – це графічні об'єкти, що були попередньо сформовані у графічному модулі. Графічними об'єктами є моделі, анімації, об'єкти взаємодії, текстури, графічні спрайти і т. д.
- ініціалізація даних до об'єктів програми – це процес ініціалізації об'єктів. Тобто, надання об'єктам певних властивостей, параметрів, які характеризують цей об'єкт. Отримані, сухі дані застосовуються до об'єктів.
- динамічна обробка вхідних взаємодій користувача з програмою – цей процес є найважливішим, адже, він виконує саму програму, як її бачить користувач у кінцевому вигляді. Результати работ усіх модулів програми об'єднуються у цьому процесі і ми отримуємо кінцевий продукт.
- формування звіту проведеної роботи – процес формування звітності проведеної роботи. У звіті показується які параметри мали певні показники у певний момент часу.
- закінчення робочого циклу програми – процес виходу с програми. Цей процес ініціалізує завершення роботи усього програмного циклу.

3.2. Опис алгоритму взаємодії програмних модулів

Згідно опису основних правил роботи програмних модулів та взаємодій між ними, необхідно також описати яким саме чином будуть взаємодіяти модулі між собою з урахуванням послідовностей та залежності подій кожного модулю системи в контексті роботи програми. На рисунку 3.10 зображена послідовність взаємодії програмних модулів системи для обробки запиту користувача.

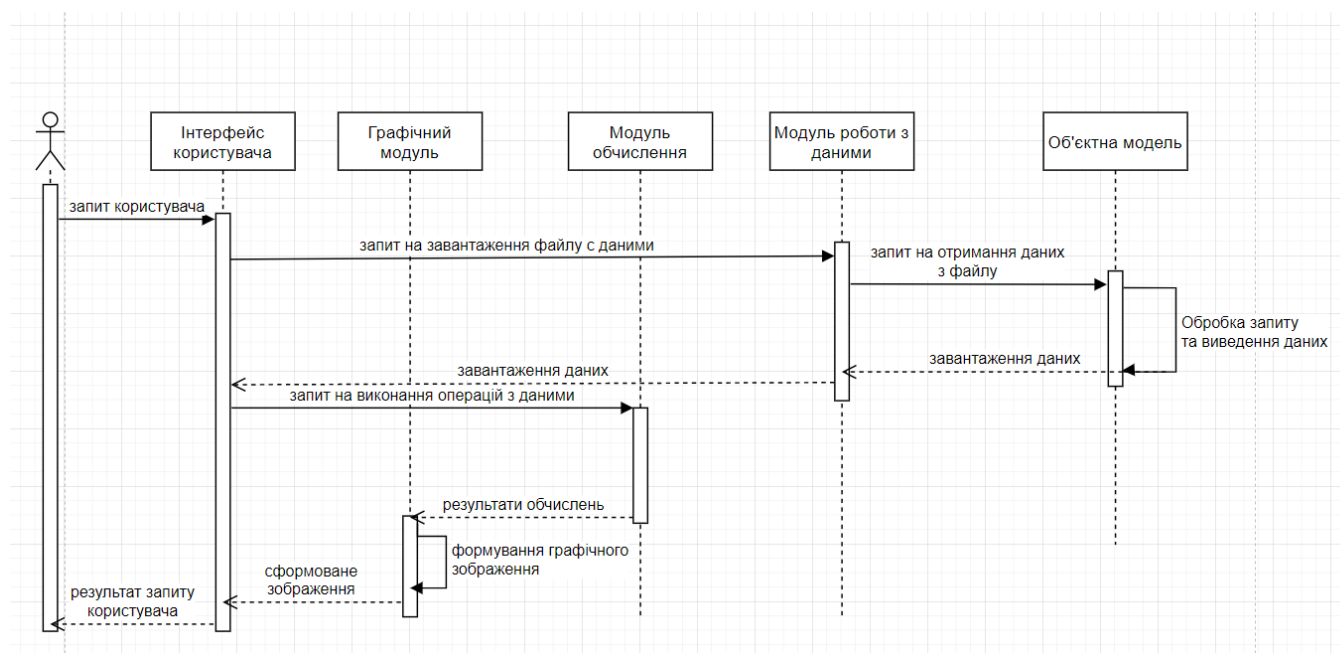


Рис. 3.10. Діаграма послідовності впливу дій користувача на програмний модуль системи для тренування пілотів

Користувач, використовуючи можливості графічного інтерфейсу користувача, складає запит, який певний чином оброблюється модулем графічного інтерфейсу та передає запит на рівень роботи з даними програмної системи. Після чого, при необхідності, відбувається завантаження файлу та її форматування у структурованому вигляді або просте повторне обчислення даних для створення необхідного для програми представлення даних. Потім відбувається передача завантажених даних у модуль обчислення, який виконує необхідні перетворення, які запросив користувач. Далі результат роботи модуля обчислення надсилає

інформація про стан виконаної роботи на графічний інтерфейс користувача, а дані перенаправляються у графічний модуль, який виконує графічне відображення даних, після чого оброблені дані виводяться на дисплей комп'ютера у вигляді графічного зображення на екрані.

3.3. Висновки розділу

Отже, результатом проектування є цілісна, автономна, зручна програмна система авіаційного тренажеру.

Було побудовано діаграму прецедентів тренування на авіаційному тренажері, яка описує задачі виконувані дійовими особами. Тобто, користуючись даною діаграмою, стає зрозумілим які самі функції та задачі повинна виконувати програмна система.

Програмна система умовно ділиться на складові, за областю функцій і задач, які вони виконують. Цими складовими є програмні модулі: модуль графічного інтерфейсу користувача (GUI); графічний модуль; модуль, який виконує обчислення; модуль роботи з даними та об'єктний модуль. Модульне представлення – один із способів розробки програмного забезпечення. За цим способом програма представляє собою ієрархічну структуру, де кожен з рівнів цієї структури – модуль. Модулем називають область виконання функцій програми, які об'єднується за принципом їх виконання, апаратними та програмними ресурсами, які необхідні для їхнього виконання та областю їхнього застосування (наприклад, графічний модуль займається лише графічною складовою програми). Зазвичай, початок роботи наступного, за ієрархією, модуля залежить від результатів роботи попереднього. При завершенні роботи усіх модулів (усієї ієрархічної структури) ми отримуємо результат роботи усієї програмної системи (у зручному форматі).

Далі надається опис взаємодії програмних модулів. Дійова особа (користувач) створює запит (здійснює введення інформації, за допомогою пристроїв вводу), який оброблюється модулем графічного інтерфейсу користувача.

Модуль GUI передає запит на дані модулю роботи з даними, який, в свою чергу, передає цей запит далі, об'єктному модулю, який є центром виконання логіки програми.

Дані певним чином оброблюються та вираховуються необхідні параметри в модулі обчислення. Графічний модуль генерує зображення і передає його на екран комп'ютера.

Таким чином, користувач бачить результат виконання програмної системи.

РОЗДІЛ 4.

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЛЯ ТРЕНУВАННЯ ПІЛОТІВ НА АВІАЦІЙНОМУ ТРЕНАЖЕРІ ІЗ ЗАСТОСУВАННЯМ ТЕХНОЛОГІЙ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ. МАТЕМАТИЧНІ РОЗРАХУНКИ. ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ

Даний розділ містить опис програмної реалізації системи, що була описана у попередньому розділі. Буде наданий перелік засобів та технологій, які використовувалися при розробці даного програмного забезпечення, надаються математичні формули, за якими були розраховані лінійні та нелінійні параметри системи, перелік умов, необхідних для запуску програмного продукту, описані основні процеси та функціональні можливості програмного забезпечення, які були реалізовані в даному дипломному проекті.

4.1. Технології та інструменти розробки програмної системи

У цьому підрозділі будуть описані основні засоби, технології та фреймворки, які були використані при написанні програми.

Кафедра КІТ (47)				НАУ 20 09 38 000 ПЗ				
			Підпис	Дата		Літера	Аркуш	Аркушів
Разраб	Джуринський К.А.				ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЛЯ ТРЕНУВАННЯ ПІЛОТІВ НА АВІАЦІЙНОМУ ТРЕНАЖЕРІ		70	16
Керівник	Моржов В.І.							
Н. Контр.	Райчев І.Е.					УС-211М	122	

4.1.1. Unity

Unity – середовище розробки програмного продукту, яка дозволяє створювати застосунки на більш ніж 25 різних платформах, таких як персональні комп'ютери, мобільні пристрої, інтернет-додатки, консолі і т.д. Головною особливістю Unity є наявність динамічної візуалізації результату кодування. Тобто, ми маємо можливість у реальному часі спостерігати динамічні зміни у нашому застосунку, не витрачаючи час на мануальну компіляцію.

У даній роботі, Unity використовувалась як основна середовище розробки програмного забезпечення. Unity має широку, відкриту (будь-хто може її доповнювати) бібліотеку програмних модулів. Це зводить написання застосунку до пошуку необхідної для конкретного завдання бібліотеки, яка містить в собі різноманітні методи та функції (потребується лише незначне редагування).

Для коректної роботи будь-якого застосунку, або програми, операційній системі потрібно розуміти як саме застосунок буде взаємодіяти з іншими застосунками, операційною системою та апаратними ресурсами комп'ютера. Для цього існує інтерфейс програмування додатків, або англ. Application Programming Interface (API). Тобто, API – це перелік правил та їхній опис, за якими програма може взаємодіяти з іншими програмами системи. У даному дипломному проєкті застосовується DirectX 11, як API.

4.1.2. DirectX 11

DirectX – це набір API (див. рис. 4.1), який був розроблений для вирішення задач, пов'язаних з програмуванням застосунків для операційних систем Microsoft Windows. Робота, яку проводить Direct X, не є наочною, а її виконання відбувається у фоновому режимі. Тобто, Direct X виступає в ролі сервісу – програмні застосунки, які відрізняються від користувацьких програмних застосунків лише тим, що роботу сервісу ми не бачимо на екрані монітору. Сервіси не мають графічного інтерфейсу

користувача, а тому керувати ними (у звичайному для нас сенсі) ми не можемо. Вони працюють у фоновому режимі.

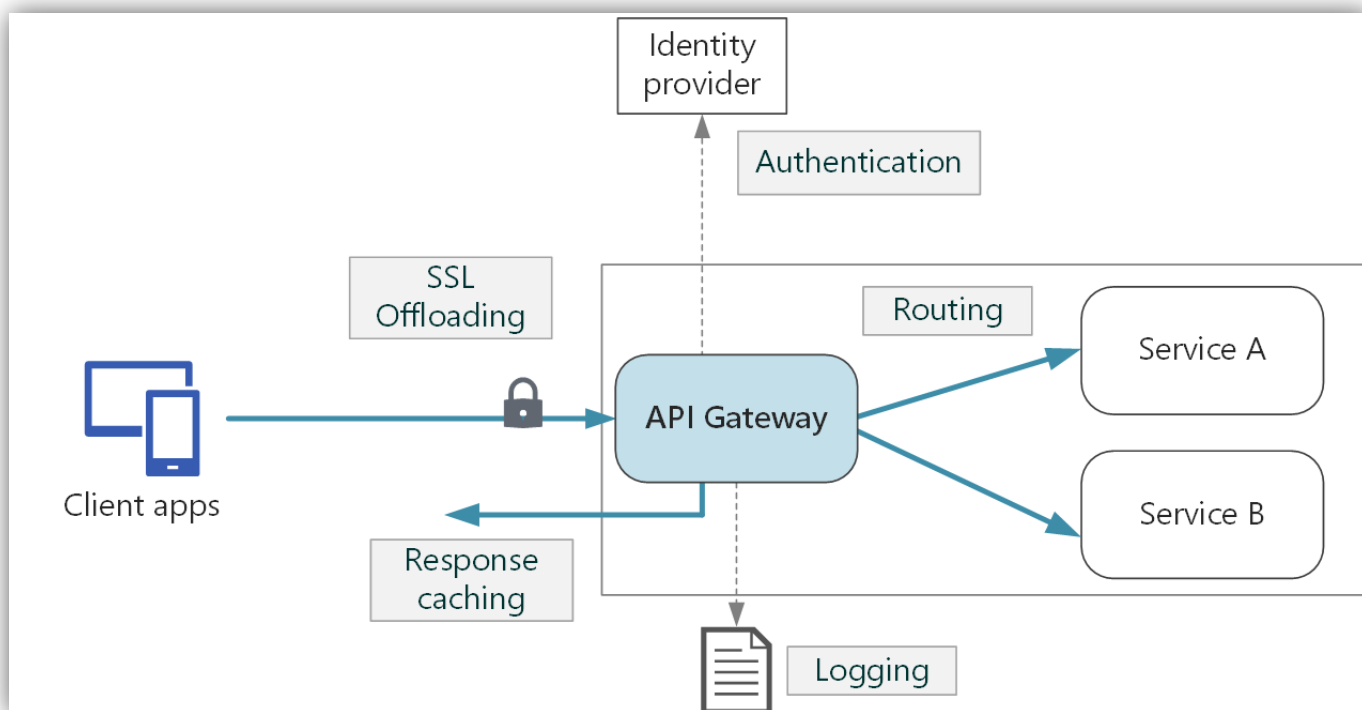


Рис. 4.1. Логічна діаграма інтерфейсу програмування додатків (API)

4.1.3. Visual studio

Visual studio – це продукт компанії Microsoft, який включає в себе інтегровану середу розробки програмного забезпечення і ряд інших інструментальних засобів. У даному проекті Visual studio використовувався як редактор вихідного коду.

4.1.4. Rift Cat

VR-системи, такі як HTC Vive та Oculus Rift, дійсно вражають, але при цьому вони дуже дорогі. Тому для розробки та налагоджування програмного забезпечення даного дипломного проекту був використаний програмний засіб Rift Cat. Rift Cat дозволяє відтворювати вміст VR-застосунків на основі ПК на невеликих мобільних

пристроях, таких як Google Cardboard, Gear VR та, навіть, мобільний телефон, що дає можливість спробувати VR-застосунки безкоштовно.

Використання цього застосунку в даному дипломному проекті є необхідністю і, звісно, краще використовувати високотехнологічні системи VR, які оснащені датчиками руху та повороту голови, гарною системою візуалізації, стереосистемою, забезпечують звукоізоляцію, наприклад, такі системи як Oculus Rift.

4.1.5. Java

Java представляє собою набір технологій та шаблонів, які дозволяють застосункам, які були написані на Java, працювати на будь-якій комп'ютерній архітектурі, для якої існує реалізація віртуальної Java-машини.

4.1.6. Steam VR

Американська компанія Valve розробила та випустила у відкритому доступі програмний набір Open VR SDK з необхідною до неї документацією. Це означає, що кожен бажаючий може використовувати їхній продукт і розвивати напрямок VR. Провідні технологічні компанії світу дуже зацікавлені у розвитку цього напрямку, адже, це новий метод керування та вводу інформації у програмну систему. Область застосування цієї технології обмежується лише фантазією. Можливо, у недалекому майбутньому, використовуючи VR-технології, будуть не тільки навчатися пілотуванню, а й керувати ПС прямо з землі.

Роль цього застосунку в даній дипломній роботі – використання Steam VR, як платформи для запуску VR-застосунку, який був розроблений в даному дипломному проекті.

4.2. Розрахунок лінійних та нелінійних параметрів ПС

Як вже було сказано, основною метою даного дипломного проекту – створення прототипу програмного забезпечення для тренування пілотів на авіаційному тренажері. Тобто, завданням є не повністю реалізувати увесь функціонал, враховувати усі погодні умови та інші зовнішні фактори. Завданням було показати яким може бути кінцевий програмний продукт під зрізом узагальнення та спрощення.

Даний програмний продукт не є кінцевою, а, лише, демо-версією. Тому, надалі деякі математичні формули матимуть спрощений вигляд, а певні внутрішні і зовнішні фактори будуть опущені. У даній програмі рух буде описуватися системою скалярних рівнянь.

4.2.1. Системні координати, які використовуються у динаміці польоту ПС

Системи координат, що застосовуються в динаміці польоту літака, являють собою прямокутні праві системи. Ці системи координат визначаються згідно з ГОСТ 20058-80.

Розглянемо системи координат, які використовуються при справжніх розрахунках.

Нормальна система координат $Ox_g Y_g Z_g$ (рис.4.2). система містить вісь OY_g , спрямовану вгору по місцевій вертикалі. Під місцевої вертикаллю розуміють пряму, що збігається з напрямком в даній точці сили тяжіння $\vec{G} = m\vec{g}$, де \vec{g} – прискорення вільного падіння.

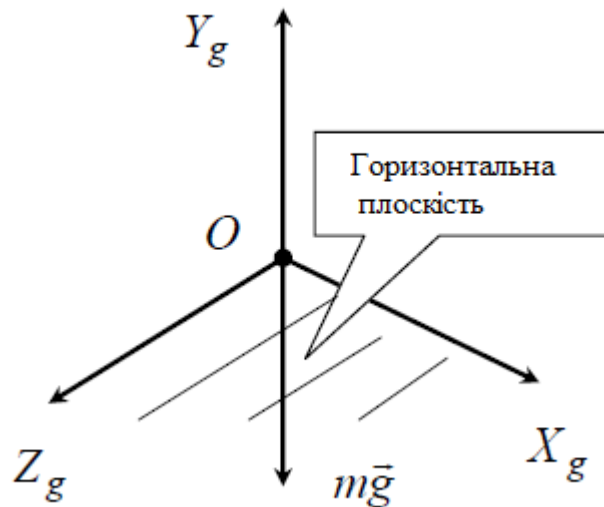


Рис. 4.2. Нормальна система координат

Початок координат O зазвичай збігається з центром мас літака. Напрямок осей OX_g і OZ_g вибирають відповідно до завдання. Для визначеності прийmemo, що вісь OX_g спрямована з півдня на північ по дотичній до географічного меридіану. Ось OZ_g розташована паралельно дотичній до паралелі в напрямку із заходу на схід.

4.2.2. Розрахунок швидкості польоту ПС

Реактивні двигуни входять до складу з'єднання та розділення деталей. Внутрішнє використання корисних сил енергетичного розвороту впливає із відносного руху маси серед твердої оболонки тіла під час її повернення у більшості проблем динаміки польоту літака цими силами можна нехтувати. Варіаційні сили зумовлені нестационарним рухом маси серед оболонок тіла.

Прийнято об'єднувати разом реактивну силу, статичні сили від різноманітності атмосферного тиску та тиску газу у вхідній секції, повітрозбірника і на виході сопла двигуна та коливальні сили, які спричинені нестационарним рухом маси всередині оболонки тіла. Отриману таким чином силу називають силою тяги двигуна \bar{P} . Момент цієї сили щодо центру мас літака позначають через \bar{M} .

Зовнішні сили, не пов'язані з роботою рухової установки і не включені в силу тяги \bar{P} , і момент цих сил позначимо відповідно як $\bar{F}_{вн}$ і $\bar{M}_{вн}$.

Таким чином, можна записати:

$$\bar{F} = \bar{F}_{вн} + \bar{P}, \quad \bar{M} = \bar{M}_{вн} + \bar{M}_p, \quad (4.1)$$

де \bar{P} – сила тяги літака;

$\bar{F}_{вн}$ – зовнішні сили сили тяги \bar{P} ;

$\bar{M}_{вн}$ – момент зовнішніх сил сили тяги \bar{P} ;

\bar{M}_p – момент сили тяги літака.

Підставивши значення формули (4.1), отримаємо остаточне рівняння руху літака у відносній (неінерційній) системі відліку:

$$d\bar{Q}_s \text{ отн} / dt = \bar{F}_{вн} + \bar{P}, \quad d\bar{K}_s \text{ отн} / dt = \bar{M}_{вн} + \bar{M}_p, \quad (4.2)$$

де $\bar{Q}_s \text{ отн}$ – головний вектор кількості руху;

t – певний проміжок часу;

$\bar{F}_{вн}$ – зовнішні сили сили тяги \bar{P} ;

\bar{P} – сила тяги літака;

$\bar{K}_s \text{ отн}$ – момент кількості руху;

$\bar{M}_{вн}$ – момент зовнішніх сил сили тяги \bar{P} ;

\bar{M}_p – момент сили тяги літака.

Головний вектор і головний момент кількості руху з формули (4.2) визначаються наступним чином:

$$\bar{Q}_s \text{ отн} = m\bar{V}, \quad \bar{K}_s \text{ отн} = I \bar{\omega}, \quad (4.3)$$

де m - маса літака;

I – його момент інерції;

\vec{V} – вектор швидкості літака;

$\vec{\omega}$ – вектор кутової швидкості обертання літака.

Для отримання скалярних рівнянь руху, придатних для виконання числових розрахунків, векторні рівняння руху замінюються еквівалентною системою диференціальних рівнянь, які є проекціями векторних рівнянь на осі обраної прямокутної системи координат.

Рух центру маси літака можна описати першим рівнянням системи (4.2), яке після підстановки $\vec{F}_{\text{вн}} = \vec{R}_a + \vec{G}$ матиме наступний вигляд:

$$m \frac{d\vec{V}_k}{dt} = \vec{R}_a + \vec{P} + \vec{G} \quad (4.4)$$

де m - маса літака;

\vec{V}_k – проекція швидкості на горизонтальну площину;

t – певний проміжок часу;

\vec{R}_a – головний вектор аеродинамічних сил;

\vec{P} – сила тяги літака;

\vec{G} – сила притягіння.

4.2.3. Розрахунок висоти та швидкості її набору

При наборі висоти, тобто, під час польоту з позитивним кутом нахилу траєкторії $\theta > 0$, вертикальна складова швидкості визначається за формулою:

$$V_y = V \sin \theta \quad (4.5)$$

де V_y – вертикальна складова швидкості;

V – швидкість польоту літака;

θ – кут нахилу траєкторії.

Зазвичай робиться припущення, що на пологих траєкторіях ($\theta = 0$) величина сили лобового опору приблизно дорівнює величині сили лобового опору в горизонтальному польоті з тієї ж швидкістю і, отже, величиною потрібної тяги [11].

Тоді можна записати:

$$V_y^* = \Delta P V / mg, \quad (4.6)$$

де V_y^* – вертикальна швидкість;

ΔP – надлишок тяги;

V – швидкість польоту літака;

mg – сила притяжіння.

Кінематичне рівняння для визначення висоти польоту:

$$dH / dt = V \sin \theta, \quad (4.7)$$

де H – поточна висота польоту;

t – певний проміжок часу;

V – швидкість польоту літака;

θ – кут нахилу траєкторії.

4.3. Основні функції та процеси програмного забезпечення

У даному підрозділі наданий опис основних функціональних можливостей програмного забезпечення. Головним завданням програмного забезпечення, що

було розроблене в даному дипломному проєкті – показати основні унікальні можливості, які має програма написана для платформи на базі VR-технологій.

Таблиця 4.1

Функціональна можливість FR1

<i>Ідентифікатор</i>	FR1
<i>Назва</i>	Режим вільного польоту камери
<i>Рівень частоти використання</i>	Середній
<i>Опис</i>	Дана функція надає можливість користувачеві контролювати процес симуляції польоту, знаходячись, наприклад, поза кабіною пілотів, та звітувати про помилки оператору-пілота, що виконує пілотування у симуляції.
<i>Залежності</i>	Відсутні

Таблиця 4.2

Функціональна можливість FR2

<i>Ідентифікатор</i>	FR2
<i>Назва</i>	Завантаження/збереження сцени симуляції
<i>Рівень частоти використання</i>	Високий
<i>Опис</i>	Дана можливість завантаження/збереження стану сцени симуляції дозволяє користувачеві

	відпрацьовувати навички пілотування у попередньо визначених специфічних умовах.
<i>Залежності</i>	Відсутні

Таблиця 4.3

Функціональна можливість FR3

<i>Ідентифікатор</i>	FR3
<i>Назва</i>	Налаштування умов симуляції
<i>Рівень частоти використання</i>	Високий
<i>Опис</i>	Ця функція робить можливим маніпулювання користувачем специфічними умовами сцени симуляції. Наприклад, налаштування погодних умов, сили вітру, висоти польоту, технічний стан літака (наприклад, можна симулювати відмову одного двигуна).
<i>Залежності</i>	FR2

Таблиця 4.4

Функціональна можливість FR4

<i>Ідентифікатор</i>	FR4
<i>Назва</i>	Маніпулювання елементами літака
<i>Рівень частоти використання</i>	Високий

<i>Опис</i>	Можливість користувачем виконувати безпосередні маніпулювання із елементами літака, роблячи процес симуляції польоту більш повним. Наприклад, до таких елементів можна віднести шасі, які можна відкрити або закрити за допомогою тумблера, що знаходиться на панелі пілота.
<i>Залежності</i>	Відсутні

Таблиця 4.5

Функціональна можливість FR5

<i>Ідентифікатор</i>	FR5
<i>Назва</i>	Збереження журналу подій
<i>Рівень частоти використання</i>	Середній
<i>Опис</i>	Користувач має можливість переглянути журнал подій для пошуку можливих помилок, що були допущені під час процесу симуляції польоту. Крім того, даний журнал дозволяє відтворити стан сцени, щоб відпрацювати певний протокол дій у певних умовах.
<i>Залежності</i>	FR2

Таблиця 4.6

Функціональна можливість FR6

<i>Ідентифікатор</i>	FR6
----------------------	-----

<i>Назва</i>	Моделювання стохастичних явищ
<i>Рівень частоти використання</i>	Низький
<i>Опис</i>	У певний стохастичний спосіб симуляція може створювати ті, чи інші умови, що можуть привести до турбулентності, поганої видимості, пролітаючого поряд необізнаного літаючого об'єкту тощо, для того щоб, оператор-пілот виконував дії, спрямовані на уникнення аварії або інших пошкоджень.
<i>Залежності</i>	FR2

4.4. Системні вимоги. Вимоги до апаратної частини комп'ютерної системи

Мінімальні вимоги до апаратного забезпечення користувача:

- частота обробки процесору не менше 2.4ГГц;
- наявність 512 Мб вільної оперативної пам'яті під час роботи програми
- наявність 1Гб вільної пам'яті на запам'ятовуючому пристрої (HDD, SSD);
- наявність 512 Мб відео пам'яті на відео карті.

Мінімальні вимоги до програмного забезпечення користувача:

- операційна система Windows 10;
- підтримка графічних специфікацій OpenVR, DirectX 11 та вище;
- версія програмних драйверів від виробника відео карти користувача від 12/04/2015.

4.5. Висновки розділу

У даному розділі була описана реалізація проекту розробки прототипу авіаційного тренажера для пілотування із застосуванням VR-технологій. Був приведений перелік програмних засобів та технологій, які були використані при написанні програмного забезпечення, а саме:

- Unity;
- DirectX;
- Visual Studio;
- Rift Cat;
- Java;
- Steam VR.

Розділ містить опис процесу обчислень лінійних та нелінійних параметрів руху літака із наведеними формулами та системами рівнянь, що використовувалися при здійсненні розрахунків.

При цьому, системи рівнянь, які описують рух літака у просторі, були спрощені, деякі зовнішні фактори знехтувані. Це було зроблено з метою спростити сприйняття програмного забезпечення, таким чином, зробивши акцент саме на технічні можливості VR-додатку.

Тому, функціональні можливості сконцентровані на використанні VR (див. рис. 4.3):



Рис. 4.3. Основні функціональні можливості програмного забезпечення

Реалізована програма є прототипом до програмного забезпечення авіаційного тренажеру для тренування пілотів із застосуванням технологій VR.

ВИСНОВКИ

Підготовка льотного екіпажу проводиться з метою отримання і підтримки рівня знань та умінь, які необхідні для виконання правильних дій членів команди екіпажу в стандартних умовах і особливих ситуаціях, які несуть за собою високий рівень небезпеки польотів.

При підготовці використовують авіаційні тренажери для наземної підготовки пілотів. Апаратно-програмні комплекси авіаційних тренажерів імітують динаміку польоту та експлуатацію авіаційних систем за допомогою спеціального програмного забезпечення комп'ютерного комплексу тренажерів. Навчання пілотів на льотному тренажері є одним з найважливіших елементів забезпечення безпечної експлуатації літака. Він мінімізує негативний вплив так званого людського фактора, тобто мінімізує можливість помилкових дій екіпажу літака.

Один з видів тренажеру – процедурний тренажер має цілком реальний інтер'єр кабіни з реальним інструментальним середовищем та органами управління, який зазвичай входить до вимог лише для найскладніших інтегрованих тренажерів.

Система візуалізації імітатора - це чотириканальна оптико-колімаційна система з комп'ютерним синтезом зображень і забезпечує безперервне та об'єднане зображення простору кабіни з кутами огляду 86° по горизонталі та 28° по вертикалі для кожного члена екіпажу.

Робочі місця інструкторів тренажера (пілот-інструктор та штурман-інструктор) забезпечують вирішення проблем управління навчальним процесом та оцінки дій учнів, включаючи використання автоматизованих систем управління. Тренажер працює за принципом генерації зображень.

Завершенням циклу підготовки елементів (теорія - іспит - навчання - тест) є обов'язковий місяць тестування на тренажері. В процесі навчання використовуються аеродроми (як правило), які використовуються оператором як пункти призначення та запасні аеродроми, включаючи запасні вздовж маршруту.

До складу авіаційного тренажера входить програмний комплекс. Програмний комплекс, у найпростішому випадку, реалізована по клієнт-серверному принципу. Тобто, є сервер, який містить різного роду дані, в том числі і службові, і є клієнтська частина, яка оброблює та взаємодіє з цими даними. Частина розрахункової роботи виконує сервер, іншу частину – клієнт. Таким чином, розподіляється навантаження на апаратні ресурси програмного комплексу.

Поняття віртуальної реальності походить від «віртуальність», що початково позначало в схоластиці зв'язок між реальностями, що належать до різних рівнів, переважно між реальністю божественною і земною. У другій половині ХХ століття під віртуальністю стали розуміти можливе, потенційне, умоглядне. З 1980-х поширився термін «віртуальна реальність» для позначення комп'ютерних систем, що дають користувачеві інтерактивне стереоскопічне зображення. Зрештою його головним значенням стала імітація дійсності з допомогою комп'ютера і різних пристроїв, що діють на органи чуття.

Існують різні підходи до розуміння віртуальної реальності. Різні дослідники описують її як справжнє буття, небуття, неповне буття. Н. А. Носов визначав такі характеристики віртуальної реальності, що вважаються в сучасних дослідженнях основними:

- *породженість* — віртуальна реальність створюється активністю іншої реальності, зовнішньої щодо неї (константної);
- *актуальність* — віртуальна реальність існує тільки «тут і тепер», поки активна константна реальність;
- *автономність* — у віртуальній реальності свій час, простір і закони існування, що можуть бути відмінними від таких у константній реальності;
- *інтерактивність* — віртуальна реальність може взаємодіяти з іншими реальностями, в тому числі і з тією, що її створює.

До них часом додаються такі, як перетвореність (віртуальна реальність є перетворена інша реальність), спостережуваність (вона є реальністю, якщо її можна сприймати органами чуття), потенційність (вона є тим, що можливе, але не втілене), символічність, відмінність від справжньої реальності, ірраціональність.

Сама програма здійснює імітацію динаміки польоту літака (на землі та в повітрі), яка здійснюється в режимі реального часу, як в робочому діапазоні висот і швидкості польоту, так і в спеціальних режимах польоту. Аеродинаміка літального апарату розраховується за математичними формулами і рівняннями. Під час виконання програми параметри, які впливають на рух літака, обчислюються в динамічному режимі. При цьому, враховується можливий вплив значення одного параметру на значення іншого, зовнішні умови та інші фактори.

Взаємодія пілота-студента з програмним комплексом відбувається за допомогою пристроїв вводу. Для інструктора – пристрій вводу неважливий, для пілотів-студентів – VR-шолом та VR-контролери.

VR-шолом та VR-контролери повинні відповідати певним вимогам, для того щоб забезпечити ефективне навчання, і, повинні володіти наступними властивостями:

- зручність використання;
- якісний звук та звукоізоляція;
- якісне зображення (Full HD дисплеї);
- чутливі датчики руху та повороту голови на шоломі;
- чутливі датчики руху на контролерах;
- забезпечувати відчуття реальності того, що відбувається.

Була сформульована мета дипломного проекту і побудований план його реалізації. Перш за все, було визначено, які саме задачі повинно виконувати дане програмне забезпечення і, які сценарії подій повинні виконуватися. Усе це було схематично зображено на діаграмі прецедентів (див. рис. 3.1) у 3 розділі дипломної записки.

Проаналізувавши основні задачі, які програмне забезпечення повинно виконувати, була обрана модульна модель для побудови структури програми.

Програмне забезпечення для тренування пілотів на тренажері із застосуванням VR-технологій має багаторівневу архітектуру. Програмне забезпечення ділиться на підсистеми (або рівні), які розташовуються одна на одну. Кожен рівень може викликати тільки рівень на один нижче нього.

Програма складається з п'яти архітектурних рівнів:

- Рівень графічного інтерфейсу користувача;
- Графічний рівень;
- Рівень обчислень;
- Рівень роботи з даними;
- Об'єктний рівень.

Об'єктна модель - колекція об'єктів або класів, за допомогою яких програма може досліджувати та маніпулювати деякими конкретними частинами своєї архітектури. Іншими словами, об'єктно-орієнтований інтерфейс до якоїсь послуги чи системи (див рис. 3.9.).

Рівень доступу до даних у системі комп'ютера, або у комп'ютерній мережі - це рівень програми, який забезпечує спрощений доступ до даних, що зберігаються в постійному стані, наприклад, реляційній базі даних.

Рівень обчислень визначає призначення програмної системи і саме на цьому рівні відбувається реалізація логіки обчислень бізнес-задач, для яких створюється система. Крім того, рівень обчислень виконує зв'язування інтерфейсу, який використовує користувач, та рівні, що знаходяться нижче за ієрархією рівнів програмної архітектури (тобто, робота з даними та об'єктна модель).

Графічний модуль - це програмне та/або апаратне забезпечення комп'ютера, яке формує графічне зображення, що виводиться на дисплей, використовуючи для цього апаратні ресурси відео-карти у поєднанні з драйвером пристрою для створення зображень на дисплеї.

Рівень графічного інтерфейсу користувача (див рис. 3.5.) виконує декілька функцій: взаємодія із користувачем (зчитування запитів користувача, надання відповіді на запит користувача, відображення статусу системи тощо); взаємодія з рівнями, що ієрархічно знаходяться нижче (зокрема, графічний рівень та рівень обчислень). У такий спосіб рівень графічного інтерфейсу використовує конкретні можливості інших рівнів без освідомлення того, як саме буде виконаний запит, єдине, що має якесь значення для даного рівня це – коректний результат запита до нижчих рівнів на запит користувача.

Далі, була описана взаємодія програмних модулів один з одним, при взаємодії користувача з програмним комплексом. Цей процес був схематично описаний на діаграмі послідовностей у 3 розділі (див. рис. 3.10).

Останній розділ містить опис програмної реалізації системи. Був наданий перелік засобів та технологій, які використовувалися при розробці даного програмного забезпечення, надаються математичні формули, за якими були розраховані лінійні та нелінійні параметри системи, перелік умов, необхідних для запуску програмного продукту, описані основні процеси та функціональні можливості програмного забезпечення, які були реалізовані в даному дипломному проекті.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

1. Олександрів В. В. та ін. Математичні задачі динамічної імітації польоту / Під заг. ред. В.А.Садовниченко. — М.: Из-во Моск. ун-та, 1986.
2. Desdemona: The next generation in movement simulation (англ.). TNO Defence, Security and Safety. [Електронний ресурс] // режим доступу: <https://bte.by/katalog/aviatsionnaya-tekhnika/samolety/kompleksnyy-aviatsionnyy-trenazher-samoleta-vertoleta.html>
3. Dr. Sunil Murthy. Motion Control: Electrifying the feel of flight. Machine Design. — липень 2014.
4. CAE True™ Electric Motion System. [Електронний ресурс] // режим доступу: <https://www.chayka.aero/trainer-boeing-737ng>
5. Kay, Steven M. (1993). *Fundamentals of Statistical Signal Processing*. Upper Saddle River, New Jersey: Prentice Hall. ISBN 0-13-345711-7. OCLC 26504848.
6. Burningham, Norman; Pizlo, Zygmunt; Allebach, Jan P. (2002). "Image Quality Metrics". In Hornak, Joseph P. (ed.). *Encyclopedia of imaging science and technology*. New York: Wiley. doi:10.1002/0471443395.img038. ISBN 978-0-471-33276-3.
7. Thales eM2K: 6-DOF Motion System. Рулевой привод // Авиация: Энциклопедия / Под ред. Г. П. Свищева. — М.: Большая Российская Энциклопедия, 1994.
8. Носов, Н.А. (2000). Словарь виртуальных терминов / Труды лаб. виртуалистики. Выпуск 7. Труды Центра профориентации. Москва. с. 16.
9. [Електронний ресурс] // режим доступу: https://ssau.ru/files/struct/deps/dinpol/umo/DPS_raschtraek.pdf
10. Коринна Лэйтан, Эндрю Мэйнард. Дополненная реальность повсюду // В мире науки. — 2019. — № 1/2. — С. 6—7.
11. Фореман Н., Коралло Л.// Прошлое и будущее 3-D технологий виртуальной реальности. — Стаття. — УДК 612.84 004.9 004.946. — журнал Научно-технический вестник ИТМО. — листопад - грудень 2014.

ДОДАТОК А. ТЕКСТ ПРОГРАМИ

Даний документ наводить основні частини вихідного коду програмної системи.

```
{
  "name": "SteamVR_Actions",
  "references": [
    "SteamVR"
  ],
  "optionalUnityReferences": [],
  "includePlatforms": [],
  "excludePlatforms": [
    "Android"
  ],
  "allowUnsafeCode": false,
  "overrideReferences": false,
  "precompiledReferences": [],
  "autoReferenced": true,
  "defineConstraints": []
}
//-----
// <auto-generated>
//   This code was generated by a tool.
//   Runtime Version:4.0.30319.42000
//
//   Changes to this file may cause incorrect behavior and will be lost if
//   the code is regenerated.
// </auto-generated>
```

```
//-----
```

```
namespace Valve.VR
```

```
{
```

```
    using System;
```

```
    using UnityEngine;
```

```
    public partial class SteamVR_Actions
```

```
    {
```

```
        private static SteamVR_Input_ActionSet_default p__default;
```

```
        private static SteamVR_Input_ActionSet_platformer p_platformer;
```

```
        private static SteamVR_Input_ActionSet_buggy p_buggy;
```

```
        private static SteamVR_Input_ActionSet_mixedreality p_mixedreality;
```

```
        public static SteamVR_Input_ActionSet_default _default
```

```
        {
```

```
            get
```

```
            {
```

```
                return
```

```
                SteamVR_Actions.p__default.GetCopy<SteamVR_Input_ActionSet_default>();
```

```
            }
```

```
        }
```

```
        public static SteamVR_Input_ActionSet_platformer platformer
```

```
{
    get
    {
        return
SteamVR_Actions.p_platformer.GetCopy<SteamVR_Input_ActionSet_platformer>();
    }
}

public static SteamVR_Input_ActionSet_buggy buggy
{
    get
    {
        return
SteamVR_Actions.p_buggy.GetCopy<SteamVR_Input_ActionSet_buggy>();
    }
}

public static SteamVR_Input_ActionSet_mixedreality mixedreality
{
    get
    {
        return
SteamVR_Actions.p_mixedreality.GetCopy<SteamVR_Input_ActionSet_mixedreality>();
    }
}

private static void StartPreInitActionSets()
{
```

```
SteamVR_Actions.p__default =
((SteamVR_Input_ActionSet_default)(SteamVR_ActionSet.Create<SteamVR_Input_Acti
onSet_default>("/actions/default")));

SteamVR_Actions.p_platformer =
((SteamVR_Input_ActionSet_platformer)(SteamVR_ActionSet.Create<SteamVR_Input_
ActionSet_platformer>("/actions/platformer")));

SteamVR_Actions.p_buggy =
((SteamVR_Input_ActionSet_buggy)(SteamVR_ActionSet.Create<SteamVR_Input_Actio
nSet_buggy>("/actions/buggy")));

SteamVR_Actions.p_mixedreality =
((SteamVR_Input_ActionSet_mixedreality)(SteamVR_ActionSet.Create<SteamVR_Input_
_ActionSet_mixedreality>("/actions/mixedreality")));

Valve.VR.SteamVR_Input.actionSets = new Valve.VR.SteamVR_ActionSet[] {
    SteamVR_Actions._default,
    SteamVR_Actions.platformer,
    SteamVR_Actions.buggy,
    SteamVR_Actions.mixedreality };
}
}
}
```