

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова**

Херсонська філія

Кафедра інформаційних технологій та фізико-математичних дисциплін

“Допущений до захисту”

Завідувач кафедри

д.т.н., доцент Гучек П.Й.

“ _____ ” _____ 2020 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

на здобуття ступеня вищої освіти “магістр”

на тему: “Дослідження моделей та технологій проектування мобільних
Android додатків для заняття спортом на «движку Unity»”

Виконав: студент VI курсу, групи 6157м
спеціальності

122 - “Комп’ютерні науки”, ОПП -

(шифр і назва спеціальності та освітньо-професійної програми)

“Інформаційні управляючі системи та технології”

Калниболотський А.О.

(прізвище та ініціали)

Керівник

к.т.н., доцент Гайдаєнко О.В.

(прізвище та ініціали)

Рецензент

В.Ф. Ажищев

(прізвище та ініціали)

30.12 - 2020 року

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова
ХЕРСОНСЬКА ФІЛІЯ

Кафедра інформаційних технологій та фізико-математичних дисциплін

Освітній ступень магістр

Галузь знань 12 - "Інформаційні технології"

(шифр і назва)

Спеціальність 122 - "Комп'ютерні науки"

(шифр і назва)

Освітньо-професійна програма "Інформаційні управляючі системи та технології"

ЗАТВЕРДЖУЮ

Гарант освітньої програми

_____ Гучек П.Й.

" 12 " жовтня 2020 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «магістр»

Студенту

Калниболотському Артему Олександровичу

(прізвище, ім'я, по батькові)

1. Тема дипломної роботи:

Керівник роботи

Гайдаєнко Оксана Володимирівна, к.ф-м.н. , доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Тема затверджена розпорядженням по філії від "12" жовтня 2020 року № 36

2. Термін подання студентом роботи 14.12.2020 р.

3. Вихідні дані до роботи:

4. Зміст та структура розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

4.1 Титульний аркуш, завдання на магістерську роботу, анотація (українською, російською, англійською), зміст, перелік умовних означень, символів, одиниць та термінів (при необхідності).

4.2 Вступ.

4.3 Дослідження технологій проектування інформаційних систем та аналіз їх особливостей.

4.4 Формування вимог до інформаційної системи, розробка її концепції і постановка задачі.

4.5 Розробка проектних рішень по інформаційній системі

4.6 Реалізація проекту інформаційної системи (технічна документація по проекту).

4.7 Розділ з економіки _____

4.8 Розділ з охорони праці та безпеки в надзвичайних ситуаціях (охорони оточуючого середовища).

4.9 Висновки.

4.10 Список використаної літератури.

4.10 Додатки (технічне завдання, загальний опис ІС, інструкція користувача, текст програми).

5. Перелік презентаційних матеріалів

5.1 Схема організаційної структури підприємства.

5.2 Архітектура автоматизованої ІС.

5.3 Організаційно-функціональна структура автоматизованої ІС.

5.4 Інформаційна модель програмного забезпечення автоматизованої ІС.

5.5 Модель переходів станів програмного забезпечення автоматизованої ІС.

5.6 Модель технологічного забезпечення автоматизованої ІС.

5.7 Інтерфейс програми

5.8 _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4.7	Ломоносов Дмитро Анатолійович		
4.8	Щедролюсєв Олександр Вікторович		

7. Дата видачі завдання _____ жовтня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів роботи	Примітка
1.	Вивчення і аналіз предметної галузі	19.10.2020	
2.	Розробка концепції ІС	22.10.2020	
3.	Розробка проектних рішень по ІС	23.10.2020	
4.	Реалізація проекту, розробка робочої документації	26.10.2020	
5.	Вирішення питань з економіки	30.10.2020	
6.	Вирішення питань охорони праці та безпеки в надзвичайних ситуаціях (охорони оточуючого середовища)	03.11.2020	
7.	Виконання графічної частини роботи	09.11.2020	
8.	Оформлення пояснювальної записки	16.11.2020	
9.	Подання роботи на попередній захист	14.12.2020	
10.	Подання роботи рецензенту	18.12.2020	
11.	Подання роботи на державний захист	23.12.2020	

Студент

_____ Підпис

Керівник роботи

_____ Підпис

Калниболотський А.О

_____ Прізвище та ініціали

Гайдаєнко О.В

_____ Прізвище та ініціали

Анотація

Дана робота присвячена дослідженню різних типів моделей в ігрових двигунах для розробки спортивного додатку. Метою роботи було дослідити моделі комп'ютерної анімації в ігровому двигуні “Unity 3D” для створення спортивного додатку на мобільні пристрої.

Робота представлена пояснювальною запискою на __167__ сторінках тексту, який включає __27__ рисунків, __3__ таблиць. Пояснювальна записка складається з вступу, шести глав, висновків, списку використаної літератури. Записка також включає п'ять додатків: технічне завдання на розробку програмного забезпечення, опис програми, керівництво користувача, програма та методика випробувань, текст програми.

Перша частина представляє вивчення моделі та технології проектування ігрових рушіїв і я в яких сферах діяльності можна викростовувати.

У рамках другої частини виконане ескізне, технічне та робоче проектування, спроектовано ІС.

У третій частині представлені логічні, фізичні діаграми та розгортання програмного забезпечення .

У четвертій частині виконані економічні розрахунки.

У п'ятій частині описана охорона праці.

У шостій частині охорона навколишнього середовища

У висновках описується актуальність проведеного дослідження та розробленого програмного забезпечення.

					MP.122.6157м.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Abstract

This work is devoted to the study of different types of models in game engines for the development of sports applications. The purpose of the work was to study the models of computer animation in the game engine "Unity 3D" to create a sports application for mobile devices.

The work is presented in an explanatory note on _167_ pages of text, which includes __27__ figures, __3__ tables. The explanatory note consists of an introduction, six chapters, conclusions, a list of references. The note also includes five Appendices: terms of reference for software development, program description, manual User description, program and test methods, program text.

The first part presents the study of the model and technology of designing IG-rov engines in which the scope of activity can be used.

Within the framework of the Second part of Execution sketch, technical and working design, software product development.

The third part presents logical, physical diagrams and software deployment.

In the fourth part economic calculations are made.

The fifth part describes labor protection.

In the sixth part, environmental protection

The conclusion describes the relevance of the study and software developed.

					MP.122.6157м.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

Зміст

Вступ	12
1.ДОСЛІДЖЕННЯ МОДЕЛЕЙ ТА РОЗРОБКА ІС ДОДАТКУ «ІГРОВОГО РУШІЯ»	13
1.1 Загальні відомості ігрового рушія.....	13
1.1.1 Історія виникнення ігрових движків.....	13
1.1.2 Загальна характеристика ігрового рушія.....	14
1.2 Технологічні властивості фізичного двигуна	15
1.2.1 Загальна характеристка фізичного двигуна	16
1.2.2 Виявлення колізій	17
1.2.3 Використання змін м'якого об'єкта	18
1.3 Можливості графічних движків	19
1.3.1 Загальний опис графічного двигуна	19
1.3.2 Візуалізація (рендерінг).....	21
1.3.3 Конвеєр рендеринга.....	22
1.3.4 Застосування, характеристика і методи візуалізації.....	23
1.3.5 Шейдерні засоби візуалізації.....	24
1.3.6 Програмування шейдерних програм.	25
1.4 Графічні API (application programming interface)	26
1.4.1 Загальний опис API.....	30
1.4.2 API Microsoft DirectX.....	32
1.4.3 API OpenGL.....	42
1.5 Використання Ігрових рушіїв в навчанні.....	44
1.6 Порівняльна характеристика ігрових рушіїв.....	45
1.7 Постановка задачі дослідження.....	47
2 ПРОЕКТУВАННЯ ІС «TRAINING GYM»»	
2.1 Загальносистемні рішення	49
2.1.1 Розробка діаграм використання ПЗ «Training gym».....	49

						Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

2.1.2. Специфікації варіантів використання	49
2.1.3 Сценарії послідовності.....	55
2.1.4 Інформаційно концептуальна модель програмного забезпечення «Traning gym».....	56
2.2 Рішення з інформаційної системи.....	58
2.2.1 Вибір мови та середовища програмування	58
2.2.2 Статична модель програми (діаграма класів).....	60
2.2.3 Специфікація класів	61
2.2.4 Динамічна модель програми	64
3.РОЗРОБКА ІС ТА ЗАГАЛЬНОСИСТЕМНІ РІШЕННЯ	67
3.1 Логічна модель	68
3.1.1 Розробка логічної моделі.....	70
3.2 Фізична модель	71
3.3 Розгортання системи	72
4 РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ ВІД ВПРОВАДЖЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ.....	74
4.1 Вступ до проблеми.....	75
4.2 Розрахунок вартості створення автоматизованої системи.....	76
4.3 Розрахунок економічної ефективності.....	77
4.4 Підсумки.....	78
5 ОХОРОНА ПРАЦІ ПРИ РОБОТІ З ПК У ВІДДІЛІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	84
5.1 Вступ до проблеми.....	85
5.2 Аналіз небезпечних та шкідливих факторів у відділі розробки програмного забезпечення.....	90
5.3 Розрахунок системи штучного освітлення приміщення.....	91
5.4 Розробка заходів щодо зменшення впливу шкідливих та небезпечних Факторів.....	92
6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА.....	93

						Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

6.1 Вступ до проблеми.....	101
6.2 Забруднення навколишнього середовища підприємством з розробки програмного забезпечення.....	103
6.3 Розробка заходів щодо зменшення забруднення.....	108
Висновки.....	111
Список використаних джерел.....	112
Додаток А.....	113
Додаток Б.....	123
Додаток В.....	134
Додаток Г.....	140
Додаток Д.....	167

						Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

Вступ

Спорт це життя. Саме так говорить велика приказка. Чим більше людина рухається, тим в кращій формі відчуває себе. Однак через глобальну комп'ютеризацію люди все менше виходять з будинків. Двори і спортивні майданчики пустують. Діти не бігають по полю і не грають м'ячем. Найчастіше вони грають в комп'ютерні ігри, дивляться серіали або проводять час іншим схожим способом. В такому випадку просто необхідно якомога частіше розминатися, адже від тривалого сидіння за комп'ютером велике навантаження лягає на хребет та очі, а це може стати причиною різних захворювань. Причинами вищесказаного можуть бути: звичайна лінь, відсутність будь-якого інтересу до спорту, велика завантаженість справами, а також небажання йти на пробіжку, якщо зовсім не хочеться займатися в самоті.

З самою лінню може боротися тільки с людина. Інтерес до спорту теж не прищепити насильно. Через щільний графік складно знайти час навіть якщо займатися спортом одному, а якщо необхідно витратити час ще й на пошук зацікавленого товариша, то це взагалі стає нездійсненним завданням. Але що ж робити, якщо лінь побороти, інтерес до спорту теж є і навіть є час для занять, але людина не розуміє як правильно займатися спортом, та котрі вправи треба винувати чи як правильно потрібно харчуватися? Особливо в наш не спокійний час потрібно вести правильний спосіб життя, добре підтримувати імунітет, займатися спортом, та правильно харчуватися.

Добре що ми живимо в досить розвиненому часі, коли є швидкий доступ до інтернету, та є можливість майже моментально знайти будь яку інформацію.

Але ж інколи не всі люди можуть шукати швидко інформацію, чи сама інформація не зовсім їм підходе. В інтернеті є величезна кількість різних спортивних статей, але в них є різні проблем наприклад: стаття може бути тільки текстова і без різних візуальних прикладів, і тому новачку може бути важко починати тренуватися. І навіть коли людина їде займатися до залу в неї не завжди є кошти на особистого тренера, тому потрібно якось вирішувати цю проблему. У наш час мобільні телефон стали не просто засобами зв'язку, а повноцінними кишеньковими комп'ютерами,

					МР.122.6157м.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

вони не тільки виконують засіб для зв'язку, але й можуть виконувати майже все те що може зробити повноцінний комп'ютер. І коли людина займається спортом смартфон може допомогти в тренуваннях, або поліпшити вже дійсні результати.

Ціль цієї роботи— це розробити спортивний додаток ‘Training gym’ на ігровому «движку» Unity з використанням програмної мови C#, на операційній мобільній системі Android. У даній роботі буде використовуватися графічні катсцени на Unity та база даних mySql в котрій будуть зберігатися різні рекомендації до вправ та додаткова інформація.

					MP.122.6157м.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

					MP.122.6157м.03.P-01					
Змн.	Арк.	№ докум.	Підпис	Дата						
Розробив		Калниболотський			ДОСЛІДЖЕННЯ МОДЕЛЕЙ ТА РОЗРОБКА ІС ДОДАТКУ «ІГРОВОГО РУШІЯ»			Лім.	Арк.	Акрушіє
Керівник		Гайдаєнко О.В..								
Консульт.		Дудченко О.М.						ХФ НУК		
Н. контр.										
Зав.кафедр		Гучек П.Й.								

1. ДОСЛІДЖЕННЯ МОДЕЛЕЙ ТА РОЗРОБКА ІС ДОДАТКУ «ІГРОВОГО РУШІЯ»

1.1 Загальні відомості ігрового рушія

1.1.1 Історія виникнення ігрових движків

Комп'ютерні ігри сьогодні є досить популярним видом розваги, як серед підлітків, так і серед дорослих людей. Тому й не дивно, що зараз існує безліч маленьких і великих компаній, що займаються розробкою ігор та програм з використанням 3D графіки або віртуальної чи доповненої реальності, і безліч великих компаній та видавництв, які займаються їх поширенням і просуванням.

Перші примітивні комп'ютерні відеоігри були розроблені в 1950-х і 1960-х роках. Вони працювали на таких платформах, як осцилографи, університетські мейнфрейми і комп'ютери EDSAC. Найпершою комп'ютерною грою, став симулятор ракети, створений в 1942 році Томасом Голдсмітом Молодшим і Істлей Рей Менном. Пізніше, в 1952, з'явилася програма, яка гра «хрестики-нулики», створена А.С. Дугласом як частина його докторської дисертації в Кембріджському Університеті. Гра працювала на великому університетському комп'ютері, відомому як EDSAC (Electronic Delay Storage Automatic Calculator)[1]. У 1958 році, Вільям Хігінботем, котрий допомагав будувати першу ядерну бомбу, в Національній Лабораторії Брукхевен (Аптон, Нью-Йорк), для розваги відвідувачів створив «Теніс для двох». У 1962 році, Стів Рассел написав «Космічна війна і Велика Пригода Джона». Гра працювала на міні-комп'ютер PDP-1 і швидко поширилася по всім університетам країни. У 1968 році, Ральф Баєр, який пізніше став відомий як «Король Відеоігор». запросив патент на ранню версію ігрової консолі «Television Gaming and Training Apparatus». У 1967 році, Баєр створив Пінгпонг гру, схожу на «Теніс для двох». Разом з Magnavox він працював над створенням першої консолі, названої Magnavox Odyssey в 1972 році. Розробка ігрових автоматів в 1970-х призвела до так званого «Золотому віці аркад». Одна з найвідоміших ігор того часу - «Pong».

З появою перших персональних комп'ютерів, ігри швидко поширилися і на цій платформі. І в даний час найпопулярнішими ігровими платформами є

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

персональні комп'ютери і спеціалізовані ігрові приставки, такі як PlayStation 4, Xbox one, Nintendo switch та смартфони.

З еволюцією ігор еволюціонували і методи їх створення. Так на початку 90-х років 20-го століття, розробники, створюючи нову гру, могли модифікувати і повторно використовувати вихідний код своїх попередніх ігрових проектів. Потім розробники зрозуміли, що повторно використаний код можна відокремити від решти гри. Так стали виділятися такі низько рівневі операції, як виведення зображення, звук, робота в мережі і деякі інші. Першо прохідцем тут можна визнати гру Doom від id Software. Подібний модульний і розширюваний дизайн дозволив гравцям і програмістам швидко видозмінювати ігрове ядро – створювати нові ігри з новими моделями, сценарієм, звуками, або змінювати існуючий матеріал.

Індустрія створення і продажу двигунів придбала дуже широкий розмах завдяки розвитку 3D-технологій. Реалізація низькорівневих операцій роботи з графікою і звуком стала дуже трудомісткою і дорогою, і багато розробників вважають за краще використовувати вже готові движки для своїх ігор. Продаж двигунів стала настільки вигідною, що деякі розробники ігор займаються в першу чергу створенням двигунів, а не ігор. Наприклад, ціна ліцензії на використання відомого движка Unity 5 від компанії Unity Technologies становить 750000 доларів на одну платформу плюс 100000 за кожен додаткову платформу (якщо ви є великою комерційною компанією), але якщо ви простий користувач котрий хоче зробити свій додаток з використанням графіки та для особистого користування то є можливість користуватися ігровим движком безкоштовно. Раз так званий ігровий рушій є основою для побудови ігор та додатків, спробуємо розібратися, що ж він із себе представляє.

1.1.2 Загальна характеристика ігрового рушія

Ігровий рушій – це центральний програмний компонент комп'ютерних відеоігор або інших інтерактивних додатків з комп'ютерною графікою, оброблюваної в реальному часі, він забезпечує основні технологічні інструменти

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

для розробки відеоігор або додатків та надає додатку можливість запускатися на декількох платформах, таких як ігрові консолі та настільні операційні системи, наприклад, Linux, Mac OS та Microsoft Windows або мобільні як IOS та Android[2].

Рушій гри, також відомий як архітектура гри, ігровий двигун або game engine – є середовищем розробки додатків призначених для розробників, для створення відеоігри або додатків. Розробники використовують ігрові рушії для створення ігор для консолей, мобільних пристроїв і персональних комп'ютерів. Основні функціональні можливості, що зазвичай надає ігровий рушій, включають в себе механізм рендеринга («рендерер») для 2D або 3D графіки, фізичний рушій або виявлення зіткнень (і реакцію на зіткнення), звук, сценарії, анімацію, штучний інтелект, роботу через інтернет мережу, потокову передачу, пам'ять, управління багатопоточність, підтримка локалізації, графічних сцени і може включати підтримку відео для кінематографії. Розробники часто економлять на процесі розробки графічних додатків, повторно використовуючи або адаптуючи, в значній мірі, один і той же ігровий рушій для створення різних відеоігор або для допомоги в перенесенні відеоігор на кілька платформ.

У багатьох випадках ігрові рушії надають набір засобів візуальної розробки додатків до багаторазового використання програмним компонентам. Ці інструменти можуть надаватися в інтегрованому середовищі розробки, щоб спростити і прискорити розробку відеоігор або програм для котрих потрібна комп'ютерна графіка на основі вже відомих даних . Розробники ігрового рушія намагаються «заздалегідь винайти колесо», розробляючи надійні програмні комплекси, які включають безліч елементів, які можуть знадобитися для розробника додатку щоб створити додаток з використанням комп'ютерної графіки або інших елементів. Більшість наборів ігрових рушіїв надають засоби, що спрощують розробку, такі як графіка, звук, фізика та функції штучного інтелекту (II).

Ці ігрові рушії іноді називають «проміжним програмним забезпеченням», тому що, як і в діловому сенсі цього терміна, вони надають гнучку і багаторазову

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

використовувану програмну платформу, яка надає всі основні функції, необхідні для розробки з самого початку, для розробки ігрової програми при одночасному зниженні витрат. Складності і час виходу на ринок – все це критичні фактори у висококонкурентній індустрії відеоігор. У 2001 році Gamebryo, JMonkeyEngine і RenderWare були широко використовуваними проміжними програмами цього типу. Як і інші типи проміжного програмного забезпечення, ігрові рушії зазвичай забезпечують абстракцію платформи, дозволяючи одній і тій же відеогрі або додатку запускатися на різних платформах (включаючи ігрові консолі і персональні комп'ютери) з невеликими змінами, якщо такі є, в вихідний код програми. Часто розробники створюють ігрові рушії з компонентної архітектурою, яка дозволяє змінювати або розширювати певні системи в рушії більш спеціалізованими (і часто більш дорогими) компонентами проміжного програмного забезпечення програми. Деякі ігрові рушії містять ряд слабо пов'язаних компонентів проміжного програмного забезпечення додатку, які можна вибірково комбінувати для створення спеціалізованого рушія замість більш поширеного підходу, що полягає в розширенні або налаштування гнучкого інтегрованого продукту. Як би там не було, розширюваність залишається пріоритетом для ігрових рушіїв через широке різноманіття їх застосувань. Незважаючи на специфіку назви «ігровий рушій», кінцеві користувачі часто перенаправляють ігрові движки для інших видів інтерактивних додатків з графічними вимогами в реальному часі, такими як маркетингові демонстрації, архітектурні візуалізації, навчальні симуляції і середовища моделювання. Деякі ігрові рушії надають тільки можливості 3D-рендеринга в реальному часі замість широкого набору функцій, необхідних додаткам. Ці двигуни покладаються на розробника відеогри, щоб реалізувати решту цієї функціональності або зібрати її з інших компонентів проміжного програмного забезпечення додатку. Ці типи рушіїв зазвичай згадуються як «графічний рушій», «рушій рендеринга» або «3D рушій» замість більш всеосяжного терміна «ігровий рушій». Ця термінологія використовується непослідовно, оскільки багато повнофункціональні двигуни 3D-ігор називаються просто «двигунами 3D». Приклади графічних движків:

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

Crystal Space, Genesis3D, Irrlicht, OGRE, RealmForge, Truevision3D і Vision Engine. Сучасні ігрові або графічні движки зазвичай надають графічних сцен – об'єктно-орієнтоване уявлення тривимірного ігрового світу, яке часто спрощує ігровий дизайн і може використовуватися для більш ефективного рендеринга великих віртуальних світів. У міру старіння технологій компоненти рушія можуть втратити свою актуальність або виявитися недостатніми для вимог конкретного проекту. Оскільки складність програмування абсолютно нового двигуна може привести до небажаних затримок (або вимагати повторного запуску проекту з самого початку), група розробників рушія може вирішити оновити свій існуючий рушій новими функціями або компонентами

Не існує єдиного, загальноприйнятого визначення ігрового движка. Кожен розробник розуміє його по-своєму, і виділяє в нього якісь певні елементи. Спробуємо побудувати найбільш широку схему ігрового движка на прикладі Unity.

Ігровий рушій надає собою гнучку і багаторазово використовувану програмну платформу з усією необхідною функціональністю для розробки ігрової програми, скорочуючи витрати, складність і час розробки – всі критичні фактори в сильно конкуруючій індустрії відеоігор чи розробників різних додатків з використанням комп'ютерної графіки.

У найбільш широкому випадку, ігровий рушій включає в себе:

- Систему ігровий логіки
- Систему введення і роботи в мережі
- Систему анімації
- Фізичний рушій або систему навігації і виявлення зіткнення
- Систему штучного інтелекту
- Звуковий рушій
- Скриптова рушій
- Базу даних тривимірних моделей і зображень
- Різноманітні редактори для роботи з движком і створення гри

Низько рівневі бібліотеки, наприклад, DirectX, SDL і OpenAL, зазвичай використовуються в іграх, так як забезпечують апаратно-незалежний доступ до апаратного забезпечення комп'ютера, такого як пристрої введення (миша, клавіатура), мережеві і звукові карти, і сильно полегшують розробку графічних додатків або відео ігор[2] .

Незважаючи на специфічність назви, ігрові движки часто використовуються в інших типах інтерактивних програм, що вимагають графіку в реальному часі, таких як рекламні демо-ролики, архітектурні візуалізації, навчальні симулятори та середовища моделювання.

1.2. Технологічні властивості фізичного двигуна

1.2.1 Загальні характеристика фізичного двигуна

Фізичний двигун є комп'ютерною програмою, який дає змогу досить вірно моделювання деяких різновидів фізичних систем, такі як переміщення твердого тіла в просторі (а також знаходження зіткнення), динаміка м'яких тіл та переміщення рідин, використовується в областях комп'ютерної графіки, відеоіграх та кіно (CGI). Найчастіше вони використовуються в відеоіграх та додатках з використанням віртуальної реальності (як правило в якості проміжного програмного забезпечення), і в цьому випадку моделювання виконується в режимі реального часу. Цей термін іноді використовується в більш загальному сенсі для опису будь-якої програмної системи для моделювання фізичних явищ, такий як високопродуктивне наукове моделювання. Останнім часом фізичний двигун є невід'ємним компонентом ігрових рушіїв.

Фізичний двигун відповідає за розв'язання рівняння руху та виявлення зіткнень. Ви можете думати про фізичний двигун як про безперервний цикл. Починається з імітації зовнішньої сили, такої як сила тяжіння. На кожному новому кроці часу він виявляє будь-які зіткнення, потім обчислює швидкість і положення об'єкта. І нарешті, відправляє дані про місцезнаходження на графічний процесор. Цей безперервний цикл створює ілюзію, що предмет падає внаслідок сили тяжіння.

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

Фізика комп'ютерної анімації включає введення законів фізики в симуляцію або ігровий рушій, особливо в тривимірну комп'ютерну графіку, з метою зробити ефекти більш реалістичними для спостерігача. Зазвичай фізика моделювання є лише близьким наближенням до реальної фізики, і обчислення виконуються з використанням дискретних значень. Крім того, відеоігри можуть навмисно відхилятися від реальної фізики для ігрових цілей; Типовий приклад - це можливість зробити подвійний стрибок, коли нема з чого стрибати, або зміна значень певних фізичних параметрів, таких як сила тяжіння. Є кілька елементів, які утворюють компоненти фізики моделювання, включаючи фізичний рушій, програмний код, який використовується для моделювання ньютонівської фізики в середовищі, і виявлення зіткнень, що використовується для вирішення проблеми визначення моменту перетину двох або більше фізичних об'єктів в середовищі. шляху один одного.

Зазвичай існує два класи фізичних движків: реального часу і високоточні. Високоточні фізичні движки вимагають більшої обчислювальної потужності для дуже точних обчислень фізики і зазвичай використовуються вченими і комп'ютерними анімаційними фільмами. Механізми фізики в реальному часі, що використовуються в відеоіграх і інших формах інтерактивних обчислень, використовують спрощені обчислення і знижену точність для обчислення часу, щоб гра або додатки котрі використовують комп'ютерну графіку могли краще та швидше реагувала з відповідністю до запитів.

Один з перших комп'ютерів загального призначення, ENIAC, використовувався як дуже простий тип фізичного движка. Він використовувався для розробки балістичних таблиць, щоб допомогти військовим США оцінити, де артилерійські снаряди різного маси приземлялися би при пострілі під різними кутами і пороховими зарядами, а також з урахуванням дрейфу, викликаного вітром. Результати були розраховані тільки один раз і занесені в роздруковані таблиці, роздані командирам артилерії. Фізичні двигуни широко використовуються на суперкомп'ютерах з 1980-х років для виконання обчислювального моделювання гідродинаміки, коли часткам присвоюються

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

вектори сил, які об'єднуються для відображення циркуляції. Через вимог до швидкості і високої точності для прискорення обчислень були розроблені спеціальні комп'ютерні процесори, відомі як векторні процесори. Ці методи можуть використовуватися для моделювання погодних умов при прогнозуванні погоди, даних в аеродинамічній трубі для проектування повітряних і водних судів або транспортних засобів, включаючи гоночні автомобілі, і теплового охолодження комп'ютерних процесорів для поліпшення теплоотводів. Як і в багатьох обчислювальних процесах, точність моделювання пов'язана з дозволом моделювання і точністю обчислень; невеликі коливання, які не змодельовані при моделюванні, можуть різко змінити прогнозовані результати. Виробники шин використовують фізичне моделювання, щоб вивчити, як нові типи протекторів шин будуть працювати в мокрих і сухих умовах, використовуючи нові матеріали шин різної гнучкості і при різних рівнях вагового навантаження.

1.2.2 Виявлення колізій

Об'єкти в додатках взаємодіють з користувачем, навколишнім середовищем і один з одним. Як правило, більшість 3D-об'єктів в додатках представлені двома окремими сітками або формами. Одна з цих сіток – дуже складна і деталізована форма, видима користувача в графічному додатку, така як ваза з елегантними вигнутими і петляють ручками. З метою швидкості використовується друга спрощена невидима сітка для представлення об'єкта фізичного механізму, так що фізичний рушій розглядає приблизну вазу як простий циліндр. Таким чином, було б неможливо вставити стрижень або вистрілити снарядом через отвори ручки у вазі, тому що модель фізичного двигуна заснована на циліндрі і не знає ручок. Спрощена сітка, яка використовується для обробки фізики, часто називається геометрією зіткнень. Це може бути що обмежує прямокутник, сфера або опукла оболонка. Механізми, які використовують, обмежують прямокутники або обмежують сфери в якості остаточної форми для виявлення зіткнень, вважаються надзвичайно простими. Зазвичай обмежує прямокутник використовується для виявлення зіткнень в

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

широкій фазі, щоб звузити кількість можливих зіткнень до того, як на вузькій фазі виявлення зіткнень буде виконано повне виявлення зіткнень з сіткою на сітці. Інший аспект точності дискретного виявлення зіткнень включає частоту кадрів або кількість моментів часу в секунду при обчисленні фізики. Кожен кадр обробляється окремо від усіх інших кадрів, і відстань між кадрами не обчислюється. Низька частота кадрів і невелика кількість рухомих об'єктів викликають ситуацію, коли об'єкт не рухається плавно в просторі, а замість цього здається, що він переміщується з однієї точки в просторі в іншу при обчисленні кожного кадру. Снаряди, що рухаються з досить високою швидкістю, не потрапляють в ціль, якщо сама ціль досить мала, щоб уміститися в проміжку між розрахунковими кадрами швидко рухомого снаряда. Різні методи використовуються для подолання цього недоліку, такі як в відеогрі «Second Life» уявлення з снарядами у вигляді стрілок з невидимими хвостами довше, ніж зазор в кадрах, зіткнення з будь – яким об'єктом, який може поміститися між обчисленими кадрами. Навпаки, безперервне виявлення зіткнень, таке як в Bullet або Navok, не страждає цією проблемою.

1.2.3 Використання змін м'якого об'єкта

Динаміка м'яких тіл – це область комп'ютерної графіки, яка фокусується на візуально реалістичних фізичних симуляції руху і властивостей деформованих об'єктів (або м'яких тіл). Додатки в основному використовуються в відеоіграх та фільмах. На відміну від моделювання твердих тіл, форма м'яких тіл може змінюватися, що означає, що відносне відстань двох точок на об'єкті не фіксована. Хоча відносна відстані між точками не фіксована, очікується, що тіло в деякій мірі збереже свою форму (на відміну від рідини). Можливості динаміки м'яких тіл досить широкі, включаючи моделювання м'яких органічних матеріалів, таких як м'язи, жир, волосся і рослинність, а також інших деформування матеріалів, таких як одяг і тканину. Як правило, ці методи забезпечують тільки візуально правдоподібне моделювання, а не точне наукове або інженерне моделювання, хоча є певні перетини з науковими методами,

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

особливо в разі моделювання методом кінцевих елементів. В даний час декілька ігрових двигунів надають програмне забезпечення для моделювання м'яких тел.

Альтернативою використанню систем фізики твердого тіла на основі деформування обмежує рамки є використання системи на основі кінцевих елементів. У такій системі тривимірний об'ємна мозаїка створюється з тривимірного об'єкту. В результаті тесселяції створюється ряд кінцевих елементів, які представляють такі аспекти фізичних властивостей об'єкта, як міцність, пластичність і збереження обсягу. Після побудови кінцеві елементи використовуються програмою для моделювання напруги в тривимірному об'єкті. Напругу можна використовувати для руйнування, деформації та інших фізичних ефектів з високим ступенем реалістичності і унікальності. У міру збільшення кількості модельованих елементів здатність двигуна моделювати фізичну поведінку збільшується. Візуальне уявлення тривимірного об'єкту змінюється системою кінцевих елементів за рахунок використання шейдера деформації, що запускається на CPU або GPU. Системи на основі кінцевих елементів були непрактичними для використання в відеоіграх через накладність витрат на продуктивність і відсутність інструментів для створення уявлень кінцевих елементів з тривимірними арт-об'єктами. Завдяки більш високопродуктивним процесорам і інструментам для швидкого створення об'ємних мозаїк системи кінцевих елементів в реальному часі стали використовуватися в іграх, починаючи з Star Wars: The Force Unleashed, в якій цифрова молекулярна матерія використовувалася для деформації і руйнування дерева, сталі та інших матеріалів. плоть і рослини з використанням алгоритму, розробленого доктором Джеймсом О'Брайеном в рамках його докторської дисертації.

Пружинні моделі це два вузла точки яких мас з'єднані паралельним контуром пружини і демпфера. У цьому підході тіло моделюється як набір точкових мас (вузлів), з'єднаних ідеальними невагомими пружними пружинами, котрі підпорядковуються деякому варіанту закону Гука. Вузли можуть починатися з країв двовимірного багатокутного сіткового представлення поверхні об'єкта, або з тривимірною мережі вузлів і ребер, що моделює

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

внутрішню структуру об'єкта (або навіть з одновимірної системи ланок, якщо, наприклад, моделюється мотузка або пасмо волосся). Можуть бути додані додаткові пружини між вузлами або змінений закон сили пружин для досягнення бажаного ефекту. Застосування другого закону Ньютона до точкових мас, включаючи сили, прикладені пружинами і будь-які зовнішні сили (через контакт, сили тяжіння, опору повітря, вітру і т. Д.), Дає систему диференціальних рівнянь для руху вузлів, яка вирішується стандартними чисельними схемами рішення ОДУ. Візуалізація тривимірної решітки маси-пружини часто виконується з використанням деформації довільної форми, при якій візуалізована сітка впроваджується в решітку і спотворюється, щоб відповідати формі решітки в міру її розвитку. Вважаючи всі точкові маси рівними нулю, можна отримати метод розтягнутої сітки, спрямований на вирішення декількох інженерних задач, пов'язаних з поведінкою пружної сітки. Іноді їх називають моделями з пружинним демпфером. У м'яких тілах під тиском модель маси пружини поєднується з силою тиску на основі закону ідеального газу

Моделювання методом кінцевих елементів – це більш фізично точний підхід, який використовує широко використовуваний метод кінцевих елементів для розв'язання диференціальних рівнянь в особистих похідних, які керують динамікою пружного матеріалу. Тіло моделюється як тривимірний пружний континуум шляхом розбиття його на велику кількість твердих елементів, які підходять один до одного, і рішення для напруг і деформацій в кожному елементі з використанням моделі матеріалу[3]. Елементи, як правило, тетраедричних, вузлів є вершинами тетраедрів (існують відносно прості методи для тетраедричного перетворення тривимірної області, обмеженою сіткою багатокутника, в тетраедри аналогічно тому, як двовимірний багатокутник може бути триангульований в трикутники). Деформація (яка вимірює локальну деформацію точок матеріалу від їх стану спокою) кількісно визначається тензором деформації. Стрес (який вимірює місцеві сили за одиницю площі в усіх напрямках, що діють на матеріалі) можна виміряти за допомогою тензора напружень Коші. З огляду на поточну локальну деформацію, локальне

напруження може бути обчислено за допомогою узагальненої форми закону Гука: де – «тензор пружності», який кодує властивості матеріалу (параметризовані лінійної пружністю для ізотропного матеріалу за допомогою коефіцієнта Пуассона і модуля Юнга). Рівняння руху вузлів елемента виходить шляхом інтегрування поля напружень по кожному елементу і співвіднесення його з допомогою другого закону Ньютона з прискореннями вузлів. Pixelux (розробники системи Digital Molecular Matter) використовують підхід на основі кінцевих елементів для своїх м'яких тіл, використовуючи тетраедричну сітку і перетворюючи тензор напружень безпосередньо в вузлові сили. Рендеринг виконується за допомогою форми довільної деформації

Моделювання тканини – відноситься до моделювання м'яких тіл у вигляді двовимірних континуальних еластичних мембран, тобто для цієї мети фактичну структуру реальної тканини на рівні пряжі можна ігнорувати. За допомогою ефектів рендеринга це може створити візуально правдоподібну імітацію текстилю та одягу, які використовуються в різних контекстах у відеоіграх, анімації і фільмах. Його також можна використовувати для моделювання двовимірних аркушів матеріалів, відмінних від текстилю, таких як деформування металевих панелей або рослинності. У відеоіграх він часто використовується для підвищення реалістичності одягнених анімованих персонажів.

1.3 Можливості графічних движків

1.3.1 Загальний опис графічного движкуна

Графічний рушій є програмне забезпечення, яке обробляє структури даних тривимірного або двовимірного світу та візуалізує ігровий світ з точки зору користувача або камери. Вирішення цього завдання може бути відносно простим, якщо ігровий рушій розробляється модульним, так що є не так багато зв'язків між графічним движком і іншими ігровими підсистемами.

Графічний рушій є «обличчям» додатку з використанням комп'ютерної графіки або відеогри. Саме за якістю одержуваної картини у користувача

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

складається перше враження про додаток.

Сучасний графічний ігровий рушій може:

- Малювати інтерфейс користувача
- Екранні меню
- Ігровий інтерфейс
- Малювати сцену
- Ландшафт
- Об'єкти
- Моделі (з анімацією)
- Оточення (небо, хмари, погода і т.д.)
- Ефекти Тіні

Даний список може бути не повним, так як все залежить від жанру відеогри чи додатку для котрого використовують комп'ютерну графіку, і від цього ж залежить, на яких частинах двигуна робиться наголос розробників. Наприклад, в стратегіях, на відміну від 3D-шутерів, не потрібні настільки деталізовані об'єкти, ландшафт, тіні.

Найчастіше 3D двигуни або системи рендеринга в ігрових движках побудовані на графічному API, такому як Direct3D або OpenGL, який забезпечує програмну абстракцію GPU або відеокарти. До появи апаратно-ускоряемой 3D графіки використовувалися програмні візуалізатори. Програмний рендеринг все ще використовується в деяких інструментах моделювання, для рендеринга зображень, для яких візуальна достовірність важливіше продуктивності (кількість кадрів в секунду) або коли апаратне забезпечення комп'ютера не задовольняє вимогам.

1.3.2 Візуалізація (рендерінг)

Візуалізація або синтез зображення - це процес створення фотореалістичного або нефотореалістичного зображення з 2D або 3D моделі за допомогою комп'ютерної програми. Отримане зображення називається рендером. Кілька моделей можуть бути визначені в файлі сцени, що містять

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

об'єкти на певній мові або в структурі даних. Файл сцен можуть містити геометрію, точку огляду, текстуру освітлення і затенення, інформація, що описує віртуальну сцену. Дані, що містяться в файлі сцени, потім передаються в програму рендеринга для обробки і виведення в файл цифрового зображення або растрового графічного зображення[4].

Термін «рендеринг» також використовується для опису процесу обчислення ефектів в програмі редагування відео для отримання остаточного виведення відео. Рендеринг – одна з основних підтем комп'ютерної 3D-графіки, і на практиці вона завжди пов'язана з іншими. Це останній важливий крок в графічному конвеєрі, що надає моделям і анімації їх остаточний вигляд.

Із зростанням складності комп'ютерної графіки з 1970-х років вона стала більш самостійною темою. Візуалізація знаходить застосування в архітектурі, відеоігор, симуляторах, візуальних ефектах кіно і телебачення, а також в візуалізації дизайну, кожен з яких використовує різний баланс функцій і методів. Доступна велика кількість рендерер. Деякі з них інтегровані в різні пакети моделювання та анімації, деякі є автономними, а деякі є безкоштовними проектами з відкритим вихідним кодом. Усередині рендерер є ретельно розробленою програмою, засновану на декількох дисциплінах, включаючи фізику світла, візуальне сприйняття, математику і розробку програмного забезпечення. Хоча технічні деталі методів рендеринга розрізняються, загальні проблеми, які необхідно подолати при створенні 2D-зображення на екрані з 3D-вистави, збереженого у файлі сцени, вирішуються графічним конвеєром в пристрої рендеринга, такому як GPU.

Графічний процесор – це спеціально розроблений пристрій, який допомагає процесору виконувати складні обчислення рендерингу. Якщо сцена повинна виглядати досить реалістичною і передбачуваною при віртуальному освітленні, програма візуалізації повинна вирішити рівняння візуалізації[15].

Рівняння рендеринга не враховує всі явища освітлення, а замість цього діє як загальна модель освітлення для комп'ютерних зображень. У разі 3D-графіки сцени можуть бути попередньо візуалізовані або згенеровані в реальному часі.

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

Попередній рендеринг – це повільний, ресурсномісткий процес, який зазвичай використовується для створення фільмів, де сцени можуть бути згенеровані заздалегідь, в той час як рендеринг в реальному часі часто виконується для 3D-відеоігор та інших додатків, які повинні динамічно створювати сцени. Апаратні 3D прискорювачі можуть поліпшити продуктивність рендеринга в реальному часі.

1.3.3 Конвеєр рендеринга

Перш ніж готова картинка виявляється на екрані монітора, виконується ряд певних дій, які можна умовно розділити на підготовку до рендерингу і сам рендеринг.

На етапі підготовки до рендерингу ігровий процесор визначає, які об'єкти зараз задіяні, які моделі вони мають, які текстури вони використовують, що з ними буде відбуватися далі і де вони знаходяться в світі. Відеогра також визначає місце розташування камери і її напрямок. Далі ігровий процесор передає інформацію візуалізатором.

Отримавши інформацію про сцену, графічний рушій повинен приступити до власного рендерингу сцени. Конвеєр рендеринга може дещо відрізнятися, в залежності від використовуваного 3D API. Але в найбільш загальному випадку його можна представити таким чином:

У разі моделей, візуалізатор повинен спочатку оцінити розмір моделей і місце розташування камери, і потім визначити, чи буде модель взагалі бути присутнім на екрані, чи потрапляє вона в поле зору камери, або вона настільки віддалена, що її взагалі не видно. Візуалізатор може навіть використовувати деякий спосіб опису світу для знаходження видимості моделі.

Система візуалізації світу визначає де в світі знаходиться камера, і які секції або полігони світу видно в полі зору камери. Все це можна здійснити різними способами оптимізації. Наприклад, за допомогою BSP дерев, або методу порталів. Можливе комбіноване застосування різних методів. Всі полігони, що

проходять через тест відсікання зайвої геометрії, передаються візуалізатором полігонів.

Для кожного полігону, передається на візуалізатор, візуалізатор здійснює трансформацію полігону відповідно до локальної математики (тобто анімацією моделі) і математикою світу (місця розташування моделі по відношенню до камери). Потім полігони досліджуються на предмет наявності нелицьових полігонів (що знаходяться на невидимій стороні об'єкта). Нелицьові полігони, знову ж таки, відкидаються. Решта полігони висвітлюються відповідно до чинних світлових джерел. Візуалізатор потім визначає, які текстури полігон використовує і засвідчується, що API чи відеокарта буде використовувати ті ж текстури для відображення. Потім полігони направляються на API рендеринга і потім на відеокарту.

В результаті виходить сформований зображення в буфері кадру відеокарти, яке потім відображується на моніторі.

Від швидкості проходження конвеєра рендеринга в великій мірі залежить продуктивність високотехнологічних відеогри. Повільна швидкість роботи відеогри надає неприємне враження на загальний ігровий процес. Тому підвищення продуктивності графічного движка є серйозною проблемою для розробників ігор, і важливо знайти правильний баланс між продуктивністю і якістю графіки. Для можливості запуску гри на різних конфігураціях персональних комп'ютерів (це не відноситься до ігрових приставок, т. К. Технічні можливості приставок визначаються при їх створенні і не змінюються протягом всього циклу їх життя) розробники повинні залишити можливість самим користувачам визначити рівень якості отримуваної картинки. Відповідно можливість таких налаштувань повинна бути передбачена при розробці графічного движка.

1.3.4 Застосування, характеристика і методи візуалізації

Коли попереднє зображення (зазвичай каркасний ескіз) завершено, використовується рендеринг, який додає растрові текстури або процедурні

					MP.122.6157м.03.Р-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

текстури, джерела світла, рельєфне відображення і відносне положення по відношенню до інших об'єктів. Результатом є закінчене зображення, яке бачить споживач або передбачуваний глядач. Для анімації фільмів необхідно візуалізувати кількох зображень (кадрів) та об'єднати їх разом в програмі, здатної створювати анімацію такого роду. Більшість програм для редагування 3D-зображень можуть це зробити

Візуалізоване зображення можна зрозуміти з точки зору ряду видимих функцій. Дослідження і розробки в галузі візуалізації в значній мірі були мотивовані пошуком способів їх ефективного моделювання. Дехто ставиться безпосередньо до конкретних алгоритмів і методам, в той час як інші створюються разом.

Затінення – як колір і яскравість поверхні змінюються в залежності від освітлення.

Текстуроване накладення – метод деталізації поверхонь.

Bump-mapping - метод імітації мелкомасштабної нерівності на поверхні.

Туманна – як тьмяніє світло при проходженні через непрозору атмосферу або повітря

Тіні – ефект загороджування світла

М'які тіні – різна темрява, викликана частково затемненими джерелами світла

Відображення – дзеркальне або глянсове відображення

Прозорість (оптика), прозорість (графіка) або непрозорість – різке пропускання світла через тверді об'єкти

Напівпрозорість – сильно розсіяне пропускання світла через тверді об'єкти

Заломлення – викривлення світла, пов'язане з прозорістю

Дифракція – вигин, поширення і інтерференція світла, що проходить повз об'єкта або апертури, яка порушує промінь.

Непряме освітлення – поверхні, освітлені світлом, відбитим від інших поверхонь, а не безпосередньо від джерела світла (також відоме як глобальне освітлення)

Каустика (форма непрямого освітлення) – відображення світла від блискучого об'єкта або фокусування світла через прозорий об'єкт для створення яскравих відблисків на іншому об'єкті.

Глибина різкості – об'єкти виглядають розмитими або не в фокусі, якщо вони знаходяться дуже далеко попереду або позаду об'єкта у фокусі.

Розмиття в русі– об'єкти виглядають розмитими через високошвидкісного руху або руху камери

Нефотореалістичніє рендеринг– рендеринг сцен в художньому стилі, який повинен виглядати як картина або малюнок

Були досліджені багато алгоритми рендеринга, і програмне забезпечення, що використовується для рендеринга, може використовувати ряд різних методів для отримання остаточного зображення. Відстеження кожної світлової частки в сцені майже завжди зовсім непрактично і вимагає колосальної кількості часу. Навіть відстеження ділянки, досить великого для створення зображення, займає непомірно багато часу, якщо вибірка не обмежена розумно.

Таким чином, з'явилося кілька різноманітних сімейств більш ефективних методів моделювання повітряного транспорту: растеризация, включаючи рендеринг рядків розгортки, геометрично проектує об'єкти сцени на площину зображення без додаткових оптичних ефектів; при моделюванні променів розглядається сцена як спостережувана з певної точки зору, обчислення спостережуваного зображення засноване тільки на геометрії і дуже простих оптичних законах інтенсивності відображення і, можливо, з використанням методів Монте-Карло для зменшення артефактів; Трасування променів аналогічна моделювання променів, але використовує більш просунуте оптичне моделювання і зазвичай використовує методи Монте-Карло для отримання більш реалістичних результатів зі швидкістю, яка часто на кілька порядків вище.

Четвертий тип техніки перенесення світла, випромінювання, зазвичай не реалізується як метод візуалізації, а замість цього розраховує проходження світла, коли він залишає джерело світла і висвітлює поверхні. Ці поверхні зазвичай відображаються на дисплеї з використанням одного з трьох інших

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

методів. Найбільш просунуте програмне забезпечення поєднує в собі два або більше методів для отримання достатньо хороших результатів за розумною ціною. Інша відмінність полягає у способах порядку зображення, які перебирають пікселі площині зображення, і алгоритмами порядку об'єктів, які перебирають об'єкти в сцені. Як правило, порядок об'єктів більш ефективний, так як зазвичай в сцені менше об'єктів, ніж пікселів.

1.3.5 Шейдерні засоби візуалізації

Шейдером в широкому сенсі називається програма для візуального визначення поверхні об'єкта. Це може бути опис освітлення, текстуровання, постобробки і т.п. Шейдери виникли з робіт Кука (Cook's shade trees) і Перлина (Perlin's pixel stream language). Програмовані шейдери були вперше представлені в RenderMan компанії Pixar, там визначені кілька типів шейдерів: light source shaders, surface shaders, displacement shaders, volume shaders, imager shaders. Ці шейдери найчастіше програмно виконуються універсальними процесорами і не мають повної апаратної реалізації. Надалі, багато дослідників описували схожі на RenderMan мови, але вони вже були призначені для апаратного прискорення. Peercy співтовариші розробили техніку для того, щоб програми з циклами і умовами виконувати на традиційних апаратних архітектурах за допомогою декількох проходів рендеринга. Шейдери RenderMan розбивалися на кілька проходів, які комбінували у буфері кадру. Пізніше з'явилися мови, які були апаратно прискореними в Direct X і OpenGL. Так шейдери були адаптовані для графічних додатків реального часу[5,6].

Відеочіпи раннього часу були програмовані і виконували тільки заздалегідь запрограмовані дії (fixed-function), наприклад, алгоритм освітлення був повністю зафіксований в апаратному забезпеченні, де нічого не можна було змінити. Потім, компанії-виробники відеочіпів поступово ввели в свої чіпи елементи програмованості, спочатку це були дуже слабкі можливості, які не отримали програмної підтримки в Microsoft DirectX API, але з часом можливості постійно розширювалися.

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

Версія Shader Model 2.0 (SM2), з'явившись в DirectX 9, серйозно розширила можливості шейдерів реального часу, запропонувавши більш довгі і складні шейдери і помітно розширився набір команд. Була додана можливість розрахунків з плаваючою комою в піксельних шейдерах, що також стало найважливішим поліпшенням. DirectX 9, а одна із найголовніших можливостей SM2, це створення мови шейдерів високого рівня – high-level shader language (HLSL), вельми схожий на мову Сі. І ефективний компілятор, що переводить HLSL програми до низькорівневого коду, «зрозумілий» для апаратних засобів. В цілому, шейдери додали до графічного конвеєра безліч нових можливостей по трансформації і висвітлення вершин і індивідуальної обробки пікселів так, як цього хочуть розробники кожного конкретного додатка.

Всі шейдери складається з двох складових: це вершинні і піксельні шейдери. Вершинні шейдери – це програми, які виконуються відео чіпами, які виробляють математичні операції з вершинами, інакше кажучи, вони надають можливість виконувати програмовані алгоритми зі зміни параметрів вершин і їх висвітлення. Кожна вершина визначається декількома змінними, наприклад, положення вершини в 3D просторі визначається координатами: x , y і z . Вершини також можуть бути прописані характеристиками кольору, текстур, координатами та іншим. Вершинні шейдери, в залежності від алгоритмів, змінюють ці дані в процесі своєї роботи, наприклад, обчислюючи і записуючи нові координати або колір. Таким чином, вхідні дані вершинного шейдера – дані про одну вершині геометричній моделі, яка в даний момент обробляється. Зазвичай це координати в просторі, нормаль, компоненти кольору і текстур координати. Результируючі дані виконуваної програми служать вхідними для подальшої частини конвеєра, растеризатор робить лінійну інтерполяцію вхідних даних для поверхні трикутника і для кожного пікселя виконує відповідний піксельний шейдер.

Піксельні шейдери – це програми, які виконуються відеочіпом під час растеризації для кожного пікселя зображення, вони виробляють вибірку з текстур та/або математичні операції над кольором і значенням глибини (Z-buffer)

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

пікселів. Всі інструкції піксельного шейдера виконуються попіксельно, після того, як операції з трансформацією і освітленням геометрії завершені. Піксельний шейдер в результаті своєї роботи видає кінцеве значення кольору пікселя і Z-значення для подальшого етапу графічного конвеєра, змішування (blending)[7].

1.3.6 Програмування шейдерних програм.

Виконання шейдерних програм відбувається в процесі візуалізації зображення, який розбитий на безліч послідовних кроків. Іншими словами в графічному процесорі використовується конвеєрна обробка даних.

На самому початку графічного конвеєра розташований блок обробки вершин. Програма, по якій відбувається обробка – є вершинним шейдером (Vertex Shader). Крім значення координат до вершин так само можуть бути прив'язані текстурні координати, колір, нормалізує обмін речовин, і.т.д. Шейдерна програма обробляє їх, видаючи трансформовані вершини, задані в логічній системі однорідних координат і лежать в проміжку від -1.0 до 1.0. Так само на даному етапі обчислюється освітленість кожної точки, що знаходиться в верховому буфері[12].

					МР.122.6157м.03.Р-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

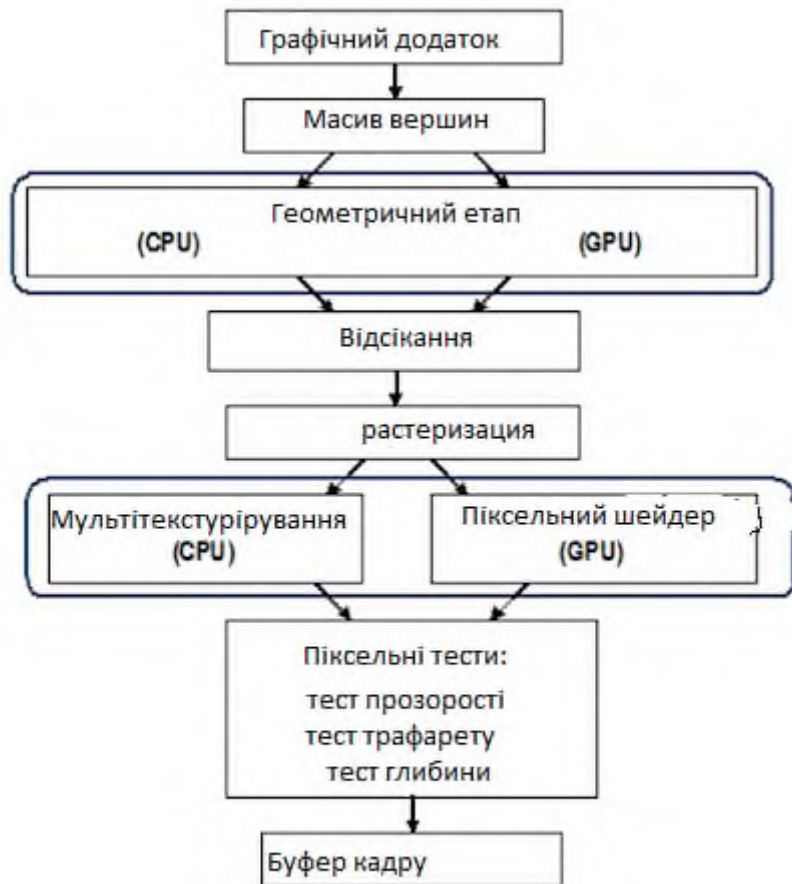


Рисунок 1.1 – Схема графічного конвеєра

На наступному етапі йде процес складання вершин в примітиви. Залежно від заданих параметрів це можуть бути трикутники (як правило, це саме вони), лінії, точки, і т.п. На цьому ж етапі йде відсікання невидимих, прихованих і повністю прозорих частин. Це робиться з метою економії ресурсів і часу, так як немає необхідності візуалізувати те, чого не видно. У підсумку після завершення даної стадії конвеєра ми вже маємо справу з потоком якихось примітивів, а не з набором ніяк не пов'язаних вершин.

Після того як з точок були зібрані графічні примітиви в справу вступає геометричний шейдер (за умови того що програма була написана на Shader Model 4.0 або вище), який перетворює існуючі примітиви і при необхідності додає нові.

Далі координати вершин трансформуються в віконну систему, в якій вони і будуть відображені на екран. Слід так само відзначити, що розробник

програмного забезпечення може з самого початку ставити усі вершини в віконних координатах. У цьому випадку необхідність в початкових етапах конвеєра відпадає адже програма самостійно їх виставляє.

Після закінчення процесу трансформації координат в віконну систему наступний етап растеризації, на якому примітиви перетворюються в різні типи або набори пікселей на екрані монітору. Параметри, прив'язані до вершин, піддаються інтерполяції, в результаті котрої кожен піксель отримує додаткові властивості, окрім його координат. Так, наприклад, задавши колір для вершин лінії, ми отримаємо обережний та градієнтний перехід за кольором протягом всієї довжини лінії. Таким чином, ми отримуємо набір потенційних пікселів, справжнє значення котрих будуть обчислені пізніше. По закінченню виконання даного етапу, ми втрачаємо будь-який зв'язок з вершинним поданням примітивів, далі ми працює виключно з колірними значеннями.

Тепер потік пікселів, отриманий в результаті інтерполяції примітивів, потрапляє на початок піксельного шейдеру (Pixel Shader), де він обробляється відповідно до заданої програми. Крім кольору пікселя, на вхід до шейдеру можуть подаватися й інші параметри. Найчастіше це значення текстурних координат, за якими відбувається вибірка пікселей з наперед визначеними текстурами, що знаходиться в пам'яті відео карти або в оперативній пам'яті комп'ютера. Незалежно від програми, піксельний шейдер зобов'язаний повернути колір підсумкового пікселя, що надходить на вхід наступного етапу графічного конвеєра.

Постобработка пікселів є занесення результатів виконання шейдерной програми в буфер кадрів, звідки він може бути виведений на екран або в збережену текстуру.

1.4 Графічні API (application programming interface,)

1.4.1 Загальний опис API

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

Розробка сучасних складних графічних програм, особливо 3D додатків, нерозривно пов'язана з використанням API-інтерфейсів (Application Programming Interface).

API – це набір бібліотек, що представляють собою готовий інтерфейс для роботи програми з 3D-акселераторами. В даний час подібних інтерфейсів існує досить багато, але всі їх можна розділити на два класи: універсальні і спеціалізовані[8].

Універсальні API є загальними для всіх 3D-акселераторів, а підтримка апаратного прискорення для цих API покладається на самі дискретні або інтегровані відеокарти. В першу чергу тут слід виділити Microsoft DirectX і OpenGL. Ці обидва API використовуються, в основному, в програмах або додатках пов'язаних з комп'ютерною анімацією.

Спеціалізовані API призначені для роботи з графічними акселераторами, побудованими на певних 3D-чіпсетах; найбільш відомими серед них є Glide API - інтерфейс для роботи з чіпами VooDoo®; Metal - для чіпів Savage3D та інше. Програми, написані з використанням спеціалізованих API, працюють тільки на тих акселераторах, під які створювалися ці API. Більшість спеціалізованих API надає тільки низькорівневий інтерфейс програмування, проте останнім часом, нові версії DirectX включають інтерфейси високорівневою підтримки, такі як DirectX for VisualBasic, який здійснює мовну підтримку мультимедіа-додатків, написаних в середовищі візуального програмування Visual Basic.

1.4.2 API Microsoft DirectX

API Microsoft DirectX – це набір програмних інтерфейсів, що застосовуються для вирішення різних завдань: від програмного управління апаратним забезпеченням комп'ютера до розробки мультимедійних додатків, що використовують різні типи інформації, і створення віртуальних світів. Основна мета, яку переслідувала фірма Microsoft, створюючи інтерфейс DirectX - перетворити комп'ютери, що працюють під управлінням операційної системи Windows, в універсальну платформу для додатків, багатих мультимедійними

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

елементами: повно колірної графіки, відео фрагментами, тривимірною анімацією і стереозвуком. Вбудований безпосередньо в ядро ОС Windows інтерфейс DirectX є інтегрованим сервісом Windows, а також Microsoft Internet Explorer (Edge). Компоненти DirectX можуть бути також автоматично завантажені на комп'ютер при установці сучасних ігор і мультимедійних додатків, розроблених для ОС Windows . Для розробників DirectX являє набір програмних інтерфейсів, використання яких дозволяє вирішити два основні завдання.

По-перше, DirectX перетворює розроблені з його допомогою додатки в програми, сумісні з будь-якою версією Windows і працюють на будь-якому комп'ютері, де встановлена ця операційна система, незалежно від типу використовуваного програмного забезпечення. При цьому подібні додатки максимально використовують технічні можливості комп'ютера, забезпечуючи найвищу продуктивність. Це досягається за рахунок сервісу, що надається двома основними компонентами DirectX: низькорівневими інтерфейсами, що входять до складу DirectX Foundation, і високорівневими інтерфейсами, складовими DirectX Media.

По-друге, DirectX надає розробникам можливість абстрагуватися від конкретного типу дисплейного адаптера, звукової карти або відеокарти і зосередитися на логіці роботи самої програми.

DirectX Foundation надає в розпорядження розробників набір низькорівневих програмних інтерфейсів, який забезпечує ефективний доступ до всіх можливостей комп'ютера, що працює під управлінням OS Windows, реалізованим на рівні апаратного забезпечення – відео картами, звуковим картам, пристроїв введення інформації. До появи DirectX розробники, що створювали мультимедійні додатки для платформи Windows, повинні були налаштовувати свої програми на роботу з різними типами пристроїв і конфігурацій. Тепер ця проблема усунена. DirectX Foundation містить компонент, відомий як «шар апаратної абстракції» (Hardware Abstraction Layer, HAL), який використовує програмні драйвери для забезпечення взаємодії програмних та апаратних засобів. В результаті розробники можуть створювати єдину версію програми з

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

використанням інтерфейсів DirectX, не піклуючись про те, щоб воно працювало на конкретних апаратних конфігураціях. DirectX автоматично визначає технічні можливості комп'ютера і встановлює відповідні параметри. DirectX також дозволяє виконувати мультимедійні додатки, що вимагають апаратної підтримки, якої немає на даному комп'ютері. У цьому випадку вони програмно емулюються компонентом, який називається "шар апаратної емуляції" (Hardware Emulation Layer, HEL) і забезпечує програмні драйвери, що працюють як відсутні пристрої.

DirectX Media розташовується над DirectX Foundation та забезпечує високорівневі сервіси – підтримку анімації, потоковий висновок (можливість передачі і перегляду аудіо та відеоінформації у міру її завантаження в інтернеті) та інтерактивність. Автоматична інтеграція низькорівневих сервісів, реалізованих DirectX Foundation, і високорівневих, реалізованих в DirectX Media, полегшує процес створення і відтворення мультимедійних елементів, дозволяючи розробникам включати їх в свої програми та Web-сторінки і забезпечуючи тим самим недоступні раніше інтерактивний зміст. Крім того, DirectX Media допомагає вирішити задачу координації різних типів мультимедійних ефектів, полегшуючи синхронізацію їх відтворення. Крім двох зазначених основних складових Microsoft DirectX в їх склад також входять високорівневі компоненти, які забезпечують мультимедійні функції для Web додатків. До них відносяться: Windows Media Player – засіб для передачі мультимедійного змісту через інтернет. Розглянемо коротко основні компоненти DirectX Foundation. До них відносяться Microsoft DirectDraw, Direct3D (режими Immediate і Retained), DirectInput, DirectMusic, DirectSound, DirectSound 3D і DirectPlay. Ці програмні інтерфейси системного рівня забезпечують ефективний доступ до різних комп'ютерних пристроїв і забезпечують реальну апаратну незалежність додатків, знімаючи проблеми установки драйверів і несумісності апаратно-програмних платформ.

Microsoft Direct3D є інтерфейс для роботи з відеокартами. Direct3D підтримує два режими роботи – Immediate Mode і Retained Mode. У режимі

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

Immediate Mode Direct3D забезпечує розробникам апаратну підтримку ігрових і мультимедійних додатків в середовищі Microsoft Windows. Він дозволяє домогтися апаратної незалежності, підтримує перемикачі Z-буферизацію і Intel MMX-архітектуру процесорів. В цьому режимі основні графічні примітиви реалізуються безпосередньо, без використання буферів виконання (execute buffers).

Режим Retained Mode полегшує створення і анімацію тривимірних світів, підтримуючи такі наприклад функції: інтерполятора анімації зі змішанням квітів, плавними рухами об'єктів і безліччю різних видів трансформації, а також послідовне заповнення гратчастої структури 3D-об'єктів (meshes), що дозволяє здійснювати їх поступову завантаження з віддалених серверів. Це дає можливість розробникам ефективно використовувати тривимірну графіку, звільняючи їх від необхідності прямого управління структурами об'єктів на низькому рівні.

Слід зазначити, що Direct3D-додатки спілкуються з графічними пристроями однаково, незалежно від режиму. Вони можуть використовувати або не використовувати програмну емуляцію перед зверненням до HAL. Насправді Direct3D тісно інтегрований з компонентом DirectDraw.

DirectDraw – це менеджер управління пам'яттю, що забезпечує базовий набір функцій для графічних і мультимедійних додатків, що працюють на платформі Windows. На відміну від традиційної Windows-графіки DirectDraw використовує прямий доступ до дисплейної пам'яті і графічних пристроїв, забезпечуючи при цьому повну сумісність з Windows додатками або пристроїв котрі підтримують DirectDraw.

DirectInput є інтерфейс до різних пристроїв введення інформації - клавіатурі, маніпулятору типу «миша», джойстика, а також до пристроїв зі зворотним віддачею (force-feedback). У порівнянні зі звичайними, стандартними функціями даний інтерфейс підтримує більше одиниць обладнання та забезпечує швидшу реакцію на запити. Працюючи безпосередньо з драйверами пристроїв, DirectInput не використовує систему обміну повідомленнями Microsoft Windows.

					MP.122.6157м.03.Р-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

1.4.3 API OpenGL

API OpenGL (Open Graphics Language) - це інтерфейс тривимірної графіки, запропонований фірмою SGI. Він дозволяє створювати тривимірні графічні зображення незалежно від використовуваних віконних інтерфейсів, операційної системи і графічних пристроїв. В даний час розробкою стандарту OpenGL займається консорціум OpenGL Architecture Review Board (ARB), до складу якого входять фірми SGI, Microsoft, Intel, IBM і Digital Equipment. Версії OpenGL існують на графічних станціях фірми SGI (під управлінням ОС IRIX і Windows NT), робочих графічних станціях фірми DEC, комп'ютерах IBM RS / 6000 і всіх платформах, підтримуваних ОС сімейства Microsoft Windows[12].

В основі OpenGL лежить бібліотека графічних функцій, що надає розробнику різні можливості для створення і відображення двовимірних і просторових моделей і анімації. Програмісти можуть об'єднувати можливості OpenGL з іншими бібліотеками або функціями, наданими ядром конкретної ОС. Більшість графічних карт з прискорювачами і спеціалізованих 3D-карт підтримують виконання примітивів OpenGL на апаратному рівні. Нижче коротко перераховані деякі основні функції і можливості, реалізовані в бібліотеці OpenGL:

- Геометричні примітиви дозволяють задавати математичний опис об'єктів. В даний час реалізуються такі примітиви: точки, лінії, растрові зображення;
- Перетворення примітивів: масштабування, переміщення, обертання, перспективні спотворення і т.д .;
- Перегляд і моделювання дозволяють розміщувати тривимірні об'єкти на сцені, пересувати камери навколо сцени, вибрати необхідний вид (точку огляду), для якого виконується рендеринг;
- Подвійна буферизація дозволяє усунути мерехтіння при анімації. Кожен наступний кадр анімації будується в окремому буфері і відображається тільки по завершенні рендеринга;

					MP.122.6157м.03.Р-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

- Антиаліасінг згладжує "ступінчасті" краю колірних переходів для згладжування "сходового ефекту";
- Поліномне обчислення для підтримки NURBS-сплайнів. Це дозволяє малювати плавні криві через невеликий набір опорних точок, що виключає необхідність зберігати всі проміжні значення;
- Атмосферні ефекти, затінення, моделі освітлення.

1.5 Використання Ігрових рушіїв в навчанні

Інформатизація освіти і зростаючі вимоги до якості і кількості висококваліфікованих фахівців призводять до необхідності розробки та впровадження інноваційних освітніх методик і технологій, що сприяють формуванню нових форм навчання, що не обмежених просторово–часовими рамками. Цим вимогам відповідає ідея віртуального навчання, яке дозволяє отримувати якісну освіту через інтернет незалежно від географічного місця розташування учня, без відриву від роботи та з урахуванням індивідуальної освітньої траєкторії[8].

У сучасних умовах, коли настільки актуальним є впровадження інформаційно-комунікаційних технологій в систему освіти країни, зростає кількість навчальних закладів, які доповнюють традиційні форми навчання дистанційною. Створення високоякісного і високотехнологічного інформаційно – освітнього середовища розглядається в основному як досить складне технічне завдання, що дозволяє докорінно модернізувати технологічний базис системи освіти, здійснити перехід до відкритої освітньої системи, що відповідає вимогам постіндустріального суспільства. Розглянемо деякі з них: 1. Unity мультиплатформенний інструмент для розробки тривимірних додатків, що працюють зокрема під операційною системою Windows – це потужний мультиплатформенний інструмент для розробки та програмування інтерактивних браузерних і настільних додатків з двох і тривимірною графікою, яка оброблюється в реальному часі. У створенні віртуального робочого

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

середовища включає в себе двигун рендеринга, фізичний двигун, звук, систему скриптів, анімацію, штучний інтелект, мережевий код, а також управління пам'яттю і багатопоточність.

Віртуальна освіта - це процес і результат взаємодії суб'єктів і об'єктів освіти, супроводжуваний створенням ними віртуального освітнього простору, специфіку якого визначають саме дані об'єкти та суб'єкти. Середовище створюється тільки тими об'єктами та суб'єктами, які беруть участь в освітньому процесі, а не класними кімнатами, навчальними посібниками або технічними засобами. Важливо підкреслити, що навчання в новому інформаційному просторі не є антагоністичним щодо до існуючих форм навчання і не заперечує наявні освітні тенденції. Нове природним чином інтегрується в ці системи, доповнюючи і розвиваючи їх, і сприяє створенню мобільного освітнього середовища[8].

1.6 Порівняльна характеристика ігрових рушіїв

На даний момент в сфері ігрових движків існує великий вибір, що дозволяє знайти підходящий засіб розробки кожному. Для того щоб визначити найбільш підходящий саме для даної роботи, необхідно порівняти найпопулярніші ігрові движки, які представлені на ринку.

Для порівняння візьмемо чотири найбільш популярних і затребуваних движків.

Unity3D є активно розробляюваним ігровим движком. В першу чергу його популярність пов'язана з низьким рівнем входження і зручністю використання, що дозволяє вести розробку без особливих глибоких знань. Unity поширюється абсолютно безкоштовно з користувальницької ліцензією. Вона дозволяє використовувати програму до тих пір, поки щорічний дохід не перевищить 100 тис. Доларів, після чого розробники пропонують перейти на одну з трьох платних програм підписки. Unity містить величезну кількість інструментів, які допомагають в розробці. Для програмування використовується високорівнева мова C # або JavaScript. Unity дозволяє створювати гри абсолютно будь-яких

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

жанрів, а також експортувати їх на 21 платформу, серед яких: Windows, iOS, Android, macOS, Xbox, Playstation і інші . Unity також відрізняє величезне співтовариство і колосальний магазин компонентів для движка - Asset Store, в якому продаються елементи для ігор.

Unreal Engine 4 ігровий движок від студії розробників Epic Games. За останні кілька років UE кардинально змінився. На даний момент движок поширюється по безкоштовній ліцензії, стягується 5% роялті від продажу проекту, що приносить більше 3000 доларів в квартал. За допомогою движка можна розробляти як прості двомірні гри так і тривимірні ігри будь-яких жанрів. UE надає два шляхи створення проекту: за допомогою графічного скриптового мови Blueprint або C ++. У першому випадку операції зв'язуються між собою за допомогою графічного редактора. У другому випадку опис геймплея здійснюється на мові C ++. UE оснащений усіма необхідними для розробки ігор редакторами і підтримує всі популярні платформи.

Cryengine також є одним з безкоштовних ігрових движків. Він головним чином націлений на розробку ММО-гри з передової графікою. У движку добре опрацьована мережева підсистема, є підтримка широкого ряду сучасних технологій. Движок активно розвивається і часто отримує оновлення. Причиною цього є відкритий код, доступ до якого може отримати будь-який бажаючий. Також Crytek активно займається підтримкою інді розробників на движку Cryengine. Абсолютно будь-який проект, розроблений на ігровому движку, може отримати фінансову та технічну підтримку, перемігши в конкурсі.

Ігровий движок Lumberyard є відгалуженням від оригінального Cryengine. Цей движок розробляється компанією Amazon. Розробники вбудували в ігровій движок підтримку хмарних технологій Amazon Web Services і Twitch . Це дозволяє працювати і зберігати дані в хмарі Amazon. Разом з тим, ці сервіси підходять для розробки потужних мультіплеєрних проектів. Ігровий движок абсолютно безкоштовний з відкритим кодом. У Lumberyard є графічний редактор коду, схожий з редактором в UE4, він дозволяє зв'язувати суті з компонентами. Також система генерації коду дозволяє модифікувати старий код C ++ і

					MP.122.6157м.03.P-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

створювати новий.

Нижче, в таблиці 1 – представлено порівняння описаних вище ігрових движків.

Показатели	Unity3D	Unreal Engine 4	Cryengine	Lumberyard
Ціна	Особиста версія – Безкоштовно Плюс версія – 35 долл/мес з особи Про версія – 125 долл/мес с особи	Безкоштовно + 5% доходу від проекту	Безкоштовно	Безкоштовно
Поріг входження	Низький	Середній	Високий	Середній
Вихідний код	Закритий	Відкритий	Відкритий	Відкритий
Платформа для використання	Windows, macOS	Windows, macOS	Windows	Windows
Спосіб створення проекту	C#	Blueprint, C++	C++, C#	Графічний редактор, C++
Цільові платформи	iOS, Android, Windows 8 Store, Windows 10 Store, macOS, PS3, PS4, Xbox 360, Xbox One, Wii U, Oculus Rift, Gear VR, PlayStation VR, Samsung Smart TV и тд.	iOS, Android, Windows 8 Store, Windows 10 Store, macOS, PS3, PS4, Xbox 360, Xbox One, Wii U,.	Windows, Linux, PlayStation 4, Xbox One.	Windows, PlayStation 4, Xbox One.

Таблиця 1 – Порівняння ігрових рушіїв

Порівняння показує, що всі сучасні ігрові движки мають як достоїнства, так і недоліками, проте в основному надають схожі можливості. В результаті для даної роботи вибір був зроблений на користь Unity3D. Це в першу чергу пов'язано з перевагою мови програмування C#. Також ігровий движок є відносно простим для розуміння і використання, що дозволить скоротити час на розробку додатку.

					MP.122.6157м.03.Р-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

1.7 Постановка задачі дослідження

На основі проведеного аналізу предметної області, визначення існуючих проблем, наявних в розробці додатків з використанням комп'ютерної графіки та в використанні ігрового рушія в розробці програм, було вирішено створити графічний додаток з використанням ігрового рушія та технологій котрі надає двигун.

Об'єктом дослідження є ігрові двигуни котрі можна використати для розробки програмних додатків з використання комп'ютерної графіки та анімації та різних фізичних властивостей.

Предметом дослідження є моделі використання різни типів анімацій та графіки ігровому рушії Unity 3d котрий є універсальним програмним забезпеченням для розробки додатків.

Метою роботи є підвищення ефективного використання ігрового рушія при розробці додатку “Training gym” та використання графічного і фізичного двигуна в додатку, використання різних видів комп'ютерної графіки, на основі досліджених технологій ігрових двигунів.

					MP.122.6157м.03.Р-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

Висновки за розділом 1

Було досліджено технології ігрового рушія і можливість його використання при проектуванні та розробці додатків різних типів. Технології з котрих складається ігровий двигун такі як фізичний двигун, різні види комп'ютерної графіки та анімації.

Незважаючи на свою назву ігровий рушій часто використовують при розробці додатків котрі зовсім не пов'язані з відеоіграм. В наш час ігрові двигуни дозволяють нам розроблять додатки в різних сферах наприклад: при розробці симулятора літаків це може спростити розробку програми та зробити її більш дешевшу, адже майже всі технології для симуляції вже вбудовані в самий двигун і тим самим зробити її доступну та теологічну.

Майже всі ігрові рушії мають низький вхід входження і тому більшість користувачів можуть вдало користуватися програмним забезпеченням.

					МР.122.6157м.03.Р-01	Лист
Изм	Лист	№ докум	Подпись	Дата		

					MP.122.6157м.03.P-02		
Змн.	Арк.	№ докум.	Підпис	Дата	ПРОЕКТУВАННЯ ІС «TRAINING GYM»		
Розробив	Калниболотський						
Керівник	Гайдаєнко О.В.						
Консульт.	Латанська Л.О						
Н. контр.							
Зав.кафедр	Гучек П.Й.				ХФ НУК		

2. ПРОЕКТУВАННЯ ІС «TRAINING GYM»

Для функціонування додатку «Training gym» було обрано спеціалізоване програмне забезпечення ігровий рушій «Unity 3D», було використано технічний, робочий та ескізний проект. Використано об'єктно–орієнтовану методологію та CASE–методологію UML.

2.1 Загальносистемні рішення

На даному етапі зображені: діаграма варіантів використання, описані специфікації варіантів використання, діаграми діяльності та концептуальна модель, представлено проект користувацького інтерфейсу.

2.1.1 Розробка діаграм використання ІС «Training gym»

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання), обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Важливо розуміти, що діаграми варіантів використання не призначені для відображення проекту і не можуть описувати внутрішній устрій системи. Діаграми варіантів використання призначені для спрощення взаємодії з майбутніми користувачами системи, з клієнтами, і особливо знадобляться для визначення необхідних характеристик системи. Іншими словами, діаграми варіантів використання говорять про те, що система повинна робити, не вказуючи самі застосовувані методи.

На рисунку 2.1 зображена модель варіантів використання програмного забезпечення «Training gym». Актор: користувач. Прецеденти: робота з

					MP.122.6157м.03.P-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

особовою картою, формування наказів, формування договорів, формування довідників, формування звітів.

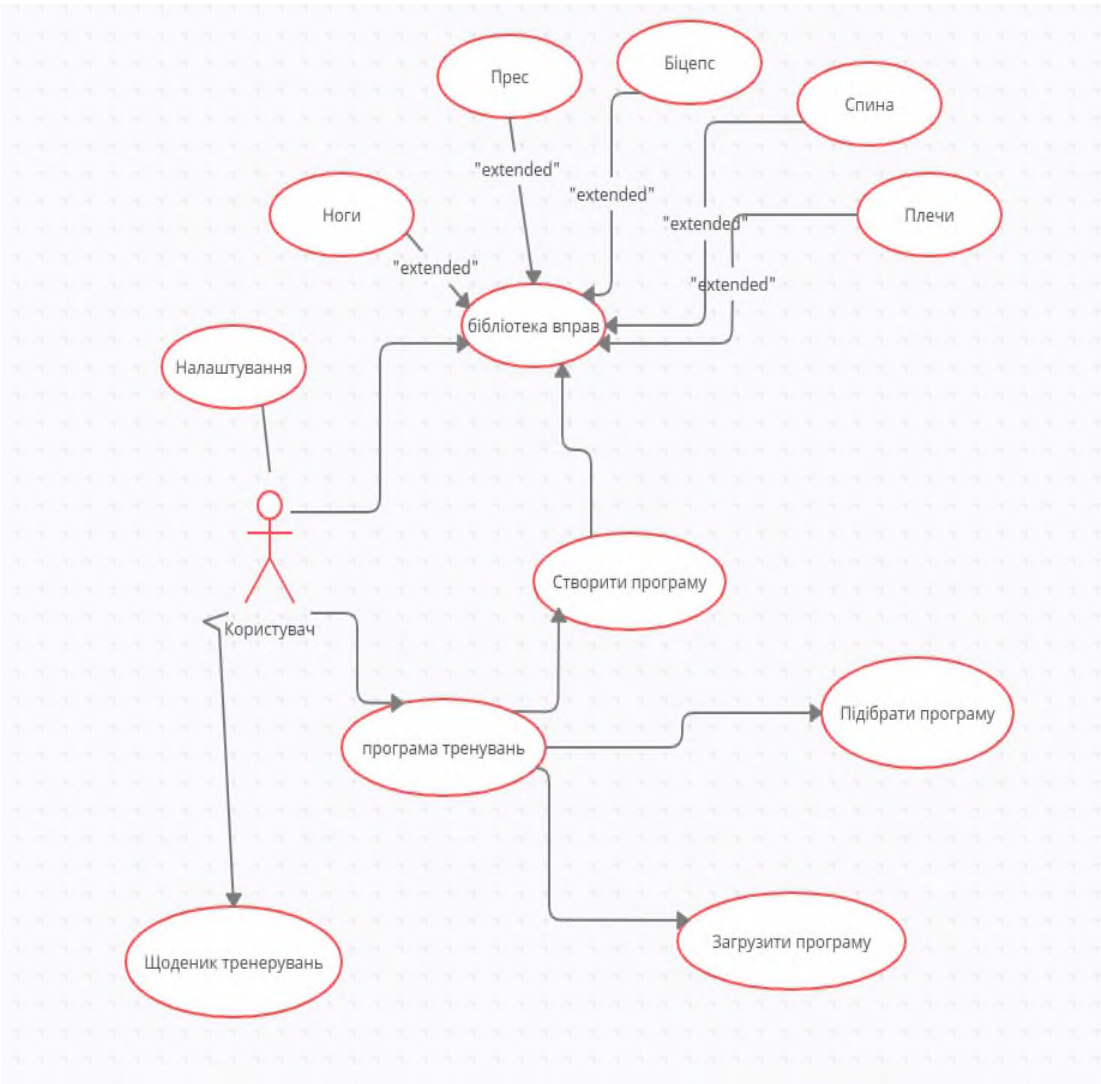


Рисунок 2.1 – Модель варіантів використання ІС «traning gym»

2.1.2 Специфікації варіантів використання

У рамках цього підрозділу розроблено специфікації варіантів використання до діаграми варіантів використання, зображеної на рисунку 2.1. Конкретизуємо варіанти використання для програмного забезпечення для заняття спортом в домашніх умовах:

а) Бібліотека вправ

Основна дійова особа: користувач.

Інші учасники прецеденту: Відсутні.

Зв'язок з іншими варіантами використання: ноги, прес, біцепс, спина, плечи, груди.

Короткий опис: Даний варіант використання потрібен для того щоб користувач міг самостійно ознайомитись з різними видами вправ

Потоки подій:

Базовий потік:

1. Користувач обирає бібліотеку вправ
2. Система показує доступні типи вправ
3. Користувач обирає потрібний тип програми
4. Система показує які вправи можна виконати
5. Користувач обирає вправу
6. Система показує вправу

Спеціальні умови: Відсутні.

Додаткові вимоги:

Відсутні

1) Передумова:

Відсутня

2) Постумова:

Відсутня.

б) Програма тренувань

Основна дійова особа: Користувач.

Інші учасники прецеденту: Відсутні.

Зв'язок з іншими варіантами використання: Відсутній.

Короткий опис: Даний варіант використання потрібен для того щоб користувач міг перейти до розділу в котрому можна створити програму тренувань, підібрати програму тренувань загрузити програму або переглянути щоденник тренувань

Потоки подій:

					MP.122.6157M.03.P-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Базовий потік: Програма тренувань

Альтернативний потік: Відсутній.

Спеціальні умови: Відсутній.

Додаткові вимоги:

1) Передумова:

Відсутня.

2) Постумова:

Відсутня.

в) Налаштування

Основна дійова особа: Користувач.

Інші учасники прецеденту: Відсутні.

Зв'язок з іншими варіантами використання: Відсутній.

Короткий опис: Даний варіант використання використовується налаштування додатку, вибору мови, одиниць вимірювання, орієнтації екрану та вибору кольору.

Потоки подій:

Базовий потік: Формування наказів.

Альтернативний потік: Відсутній.

Спеціальні умови: Відсутні.

Додаткові вимоги:

1) Передумова:

Відсутня.

2) Постумова:

Відсутня.

г) Щоденник тренувань

Основна дійова особа: користувач.

Інші учасники прецеденту: Відсутні.

Зв'язок з іншими варіантами використання: Відсутній.

					MP.122.6157M.03.P-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Короткий опис: Даний варіант використання використовується для того щоб користувач міг вносити дані про своє тренування.

Потоки подій:

Базовий потік: Щоденник тренувань.

Альтернативний потік: Відсутній.

Спеціальні умови: користувач повинен самостійно вводити свої дані.

Додаткові вимоги:

1) Передумова:

Відсутня.

2) Постумова:

Відсутня.

д) Створення програми

Основна дійова особа користувач.

Інші учасники прецеденту: Відсутні.

Зв'язок з іншими варіантами використання: бібліотека вправ.

Короткий опис: Даний варіант використання використовується для підбору систем тренування, де користувач може підібрати вправи для тренувань самостійно.

Потоки подій:

Базовий потік: програма тренувань, створення програми

Альтернативний потік: Відсутній.

Спеціальні умови: Відсутні.

Додаткові вимоги:

1) Передумова:

Відсутня.

2) Постумова:

Відсутня.

е) Підібрати програму

					MP.122.6157м.03.P-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Основна дійова особа: користувач.

Інші учасники прецеденту: Відсутні.

Зв'язок з іншими варіантами використання: Відсутні.

Короткий опис: Даний варіант використання використовується для автоматичного вибору програми тренувань, котрі були закладені раніше розробником.

Потоки подій:

Базовий потік: програма тренувань, підібрати програму

Альтернативний потік: Відсутній.

Спеціальні умови: Відсутні.

Додаткові вимоги:

1) Передумова:

Відсутня.

2) Постумова:

Відсутня.

ж) Загрузити програму

Основна дійова особа: користувач.

Інші учасники прецеденту: Відсутні.

Зв'язок з іншими варіантами використання: Відсутні.

Короткий опис: Даний варіант використання використовується для того щоб користувач міг завантажити все створене користувачем програми тренувань

Потоки подій:

Базовий потік: програма тренувань, загрузити програму

Альтернативний потік: Відсутній.

Спеціальні умови: Відсутні.

Додаткові вимоги:

1) Передумова:

Відсутня.

					MP.122.6157M.03.P-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

2) Постумова:

Відсутня.

2.1.3 Сценарії послідовності

Діаграма діяльності в UML це візуальне представлення графу діяльності. Граф діяльності є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій.

Під діяльністю (англ. Activity) розуміється специфікація виконуваного поведінки у вигляді координованого послідовного і паралельного виконання підлеглих елементів - вкладених видів діяльності і окремих дій англ. action, з'єднаних між собою потоками, які йдуть від виходів одного вузла до входів іншого.

Діаграми діяльності використовуються при моделюванні бізнес-процесів, технологічних процесів, послідовних і паралельних обчислень.

На рисунку 2.2 зображено діаграму діяльності для варіанту перегляду вправ котрі доступні в спортивному додатку.

На рисунку 2.3. зображено діаграму діяльності для варіанту використання при підборі програми тренувань.

На рисунку 2.4. зображено діаграму діяльності для варіанту створення самостійної програми тренувань.

					MP.122.6157м.03.Р-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

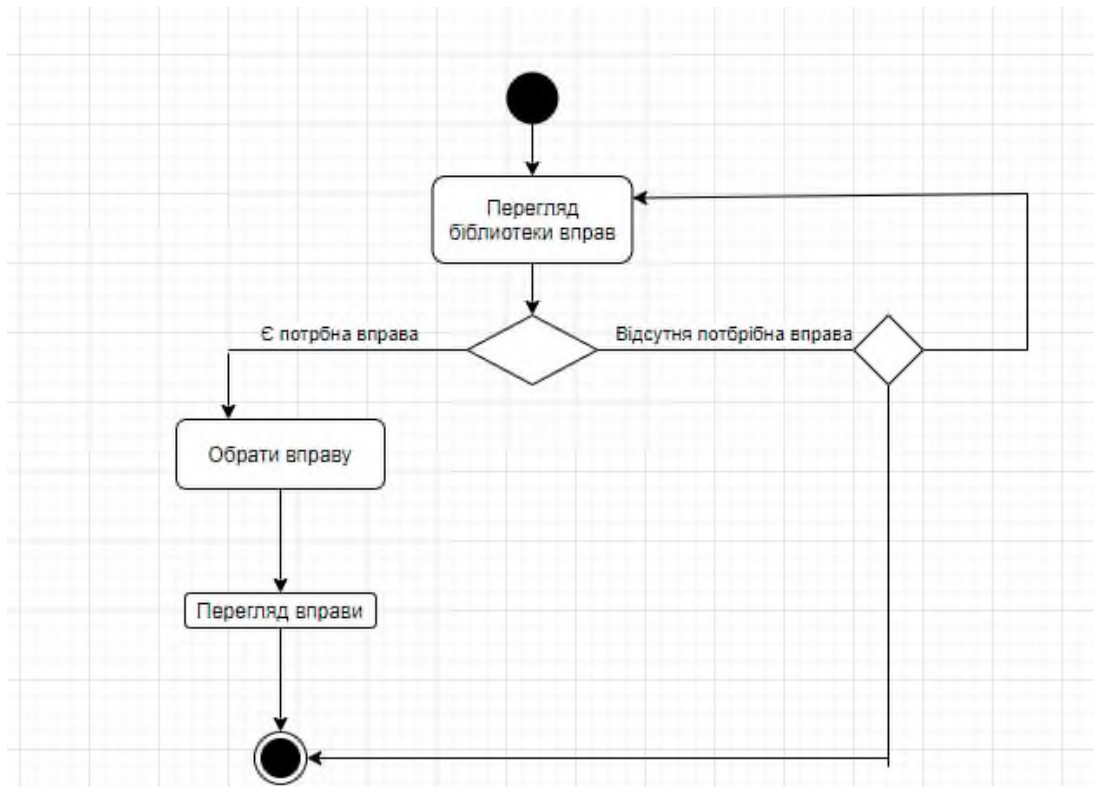


Рисунок 2.2 – Варіант перегляду вправ

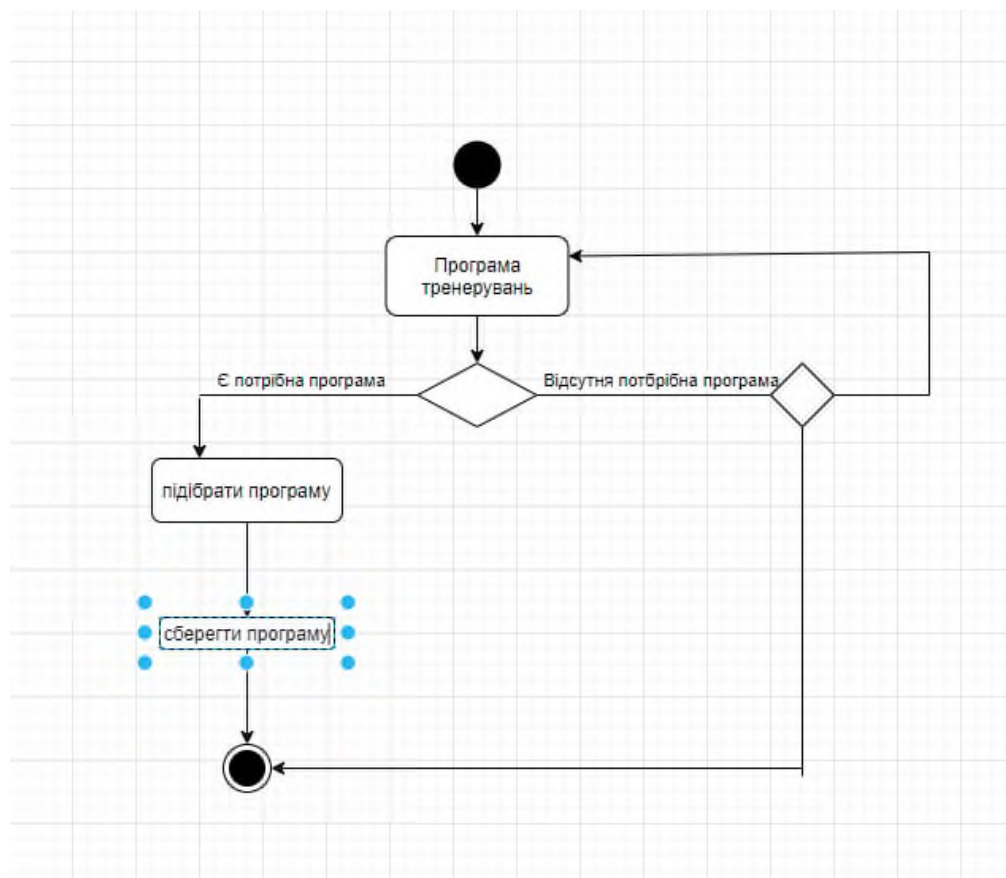


Рисунок 2.3 – Вибір готових програм тренувань

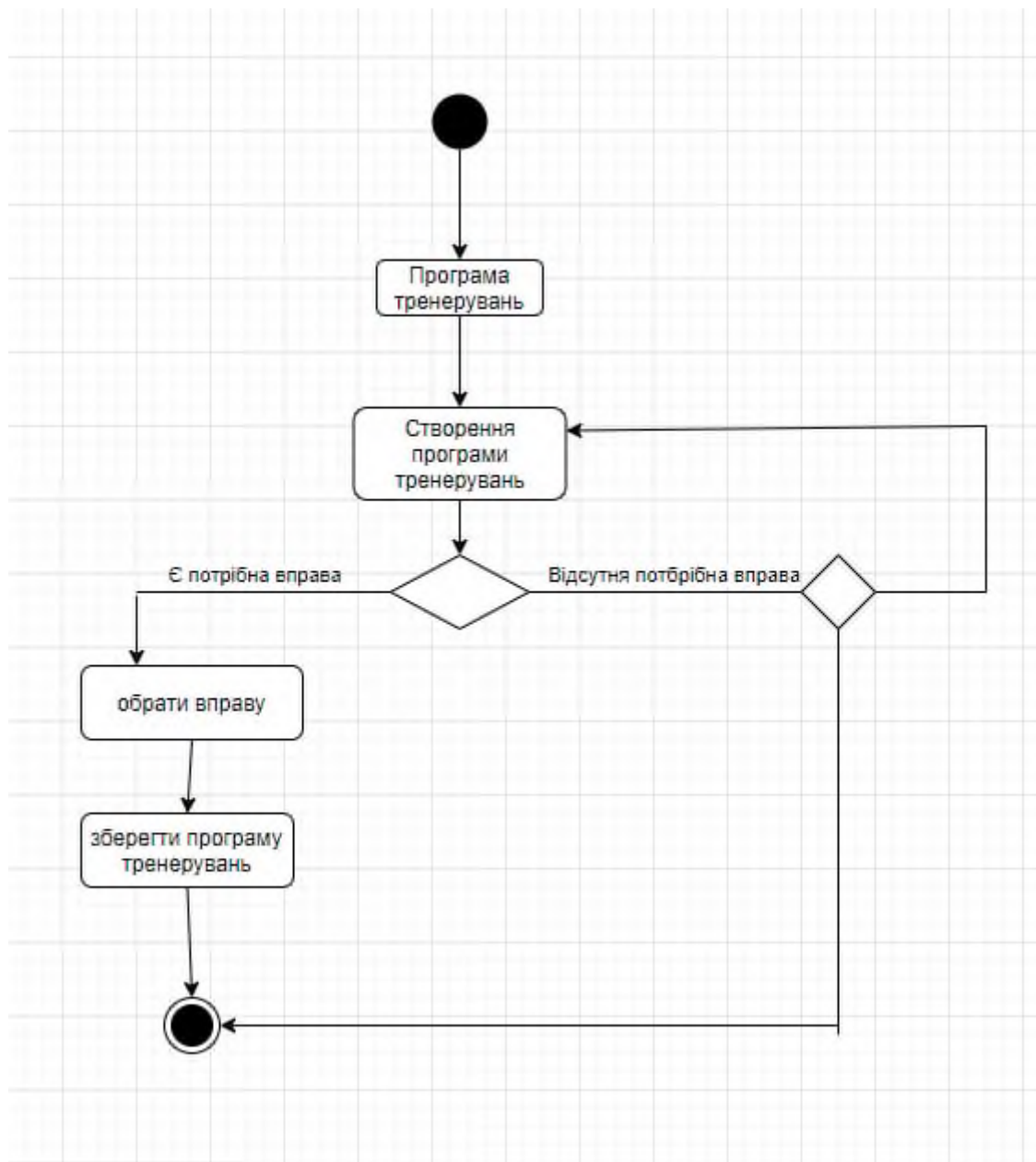


Рисунок 2.4 –Самостійне створення програми тренувань

2.1.4 Інформаційно концептуальна модель програмного забезпечення «Traning gym»

Концептуальна модель – модель предметної галузі, що складається з переліку взаємопов'язаних понять, що використовуються для опису цієї області, разом з властивостями й характеристиками, класифікацією цих понять, за типами, ситуацій, ознаками в даній галузі, і законів протікання процесів в ній. (Тлумачний словник по штучному інтелекту)

Концептуальна (змістовна) модель – це абстрактна модель, що визначає структуру модельованої системи, властивості її елементів і причинно-наслідкові зв'язки, властиві системі і суттєві для досягнення мети моделювання.

На рисунку 2.5 представлена концептуальна модель у вигляді діаграми класів для розроблюваного програмного продукту «Traning gym»

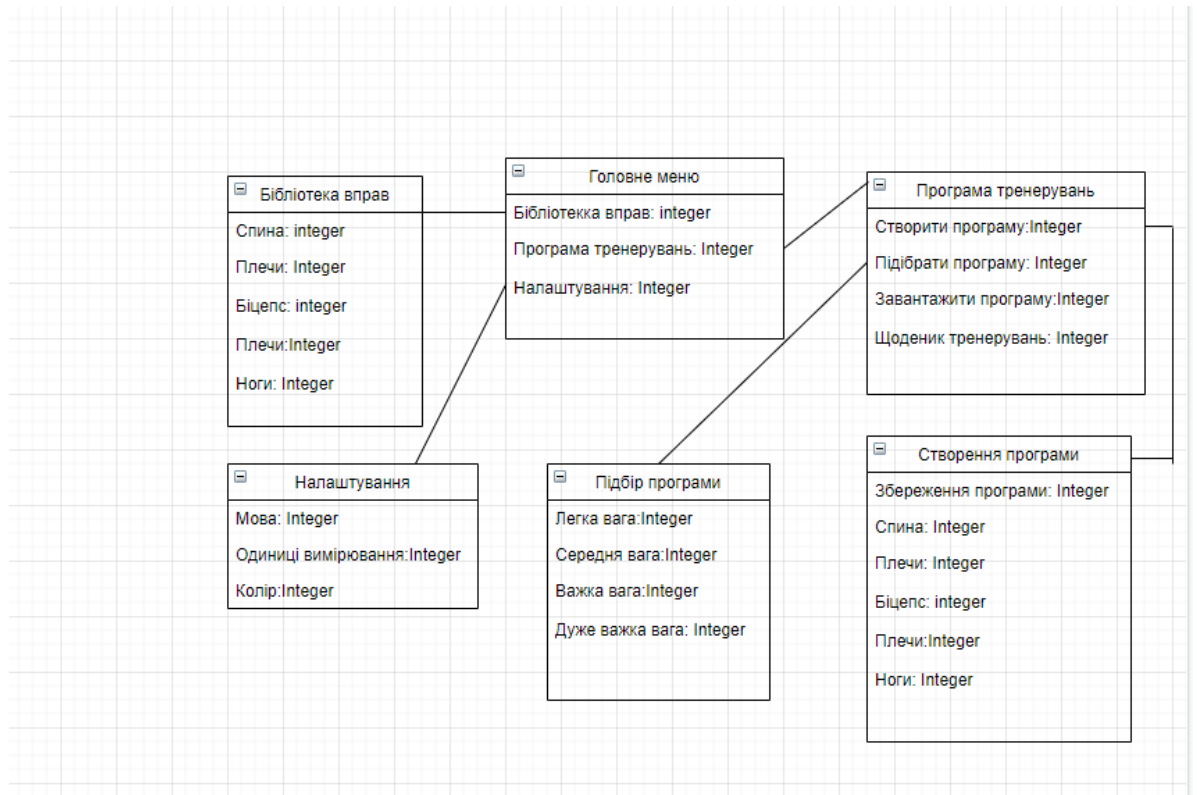


Рисунок 2.5 –Концептуальна модель програмного додатка «Traning gym»

2.2 Рішення з інформаційної системи

2.2.1 Вибір мови та середовища програмування

В якості платформи було обрано ігровий рушій Unity 3d, мова програмування C #, СУБД SQLite, SQLiteStudio для роботи з базою даних, мова програмування запитів SQL.

C # - об'єктно-орієнтована мова програмування. Розроблена в 1998-2001 роках групою інженерів компанії Microsoft під керівництвом Андерса Хейлсберг і Скотта Вільтаумота як мову розробки додатків для платформи

Microsoft .NET Framework. Згодом був стандартизований як ECMA-334 і ISO / IEC 23270 . С # відноситься до сім'ї мов з С-подібним синтаксисом, з них його синтаксис найбільш близький до С ++ і Java. Мова має статичну типізацію, підтримує поліморфізм, перевантаження операторів (в тому числі операторів явного і неявного приведення типу), делегати, атрибути, події, властивості, узагальнені типи і методи, анонімні функції з підтримкою замикань, LINQ, виключення, коментарі у форматі XML. Мова розроблена так, щоб дати програмісту максимальний контроль над усіма аспектами структури і порядку виконання програми[10].

Unity є сучасним крос-платформним движком для створення ігор і додатків, розроблений Unity Technologies. За допомогою даного движка можна розробляти не тільки додатки для комп'ютерів, але і для мобільних пристроїв (наприклад, на базі Android), ігрових приставок і інших девайсів. Unity - це інструмент для розробки двох-і тривимірних додатків та ігор, що працює під операційними системами Windows, Linux і OS X. Створені за допомогою Unity програми працюють під операційними системами Windows, OS X, Windows Phone, Android, Apple iOS, Linux , а також на ігрових приставках Wii, PlayStation 3, PlayStation 4, Xbox 360, Xbox One і MotionParallax3D дисплеях (пристрої для відтворення віртуальних голограм), наприклад, Nettlebox. Є можливість створювати додатки для запуску в браузерях за допомогою спеціального модуля Unity (Unity Web Player), а також за допомогою реалізації технології WebGL. Раніше була експериментальна підтримка реалізації проектів в рамках модуля Adobe Flash Player , але пізніше команда розробників Unity прийняла складне рішення щодо відмови від цього[8].

Додатки, створені за допомогою Unity, підтримують DirectX і OpenGL. Активно движок використовується як великими розробниками (Blizzard , EA, QuartSoft, Ubisoft), так і розробниками Indie-ігор (наприклад, ремейк Мор. Утопія (Pathologic), Kerbal Space Program, Slender: The Eight Pages, Slender : The Arrival, Surgeon Simulator 2013, Baeklyse Apps: Guess the actor і т. п.) в силу

					MP.122.6157м.03.Р-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

наявності безкоштовної версії, зручного інтерфейсу і простоти роботи з двигуном.

Редактор Unity3d має простий і зрозумілий інтерфейс, який має різні вікна і легко налаштовується під себе.

За розрахунку фізики відповідає фізичний движок PhysX від компанії NVIDIA.

Проект в Unity3d складається зі сцен – це окремі файли, які містять свої ігрові світи зі своїми об'єктами, сценаріями і настройками. Сцени складаються з об'єктів (моделей) і порожніх ігрових які не мають моделі («пустушки»). Об'єкти, мають набір компонент, з якими і взаємодіють скрипти. У об'єктів є свої імена (допускається два однакових імені). Може бути мітка і шар, на якому наш об'єкт повинен відображатися. У будь-якого об'єкта на сцені завжди присутній компонент "Transform" – зберігає координати місця розташування, повороту і розмірів об'єкта по трьох осях. Об'єкти, що мають видиму геометрію, мають компонент "Mesh Renderer", що робить модель об'єкта видимою

Unity3d підтримує фізику тканини і твердих тіл. У редакторі є система успадкування об'єктів; об'єкти дочірні будуть повторювати всі зміни, батьківського об'єкта.

Звуки, текстури, моделі і скрипти можна конвертувати в формат .unityassets і передавати іншим розробникам, або викладати у вільний доступ. Цей же формат використовується в магазині Unity Store, в якому у розробників є можливість безкоштовно і за гроші викладати в загальний доступ різні матеріали, які можуть стати в нагоді при створенні програми. Для доступу в Unity Store, потрібно мати акаунт розробника Unity3d. У Unity3d є всі компоненти для створення додатків з доступом до інтернету. У середовищі розробки Unity можна легко скомпілювати ваш додаток під різні ОС. Написання скриптів відбувається на найпопулярніших мовами програмування це C # і JavaScript, але з 2017 року залишилась підтримка Javascript завершилась.

SQLite – компактна вбудована СУБД. Сама бібліотека SQLite написана на C; існує велика кількість прив'язок до інших мов програмування, в цьому

					MP.122.6157М.03.P-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

числі Delphi, C ++, Java, C #, Python, Perl, PHP, а також до багатьох інших. SQLite підтримує динамічне типізування даних. Перевага простота і зручність вбудовування. Вбудована система управління базами даних - архітектура систем управління базами даних, коли СУБД тісно пов'язана з прикладною програмою і працює на тому ж комп'ютері, не вимагаючи професійного адміністрування. Вбудовані СУБД застосовуються в багатьох програмах, які зберігають великі масиви даних, але при цьому не потрібно доступ з багатьох комп'ютерів .

2.2.2 Статична модель програми (діаграма класів)

Діаграма класів - це тип діаграм, які частіше за все використовуються при моделюванні об'єктно-орієнтованих систем. Елементи діаграм класів пов'язані різними структурними зв'язками. Такі діаграми використовують для проектування словника системи чи кооперацій та систем. Діаграми класів мовою UML використовують для того, щоб показати параметри блоків цих діаграм та їх зв'язків.

Діаграма має вигляд символів класів – так званих іконок та зв'язків між ними. Терміном іконка позначають стандартизоване, фіксованої форми, візуальне зображення (так би мовити, ієрогліф) певного поняття, яке легко розпізнається. Іконка класу має форму прямокутника, який може поділятися на дві або три частини. Верхня його частина обов'язкова, вона містить ім'я класу. Друга й третя частини прямокутника можуть наводитися або пропускатися і містять: друга - список атрибутів класу, третя - список операцій класу.

Інакше кажучи, діаграма класу може відображати лише імена класів або імена та відповідні атрибути класів, або імена, атрибути та операції (методи) класів.

Існують різні точки зору на побудову діаграм класів в залежності від цілей їх застосування:

					MP.122.6157м.03.P-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

- Концептуальна точка зору - діаграма класів описує модель предметної області, в ній присутні тільки класи прикладних об'єктів;
- Точка зору специфікації - діаграма класів застосовується при проектуванні інформаційних систем;
- Точка зору реалізації - діаграма класів містить класи, використовувані безпосередньо в програмному коді (при використанні об'єктно-орієнтованих мов програмування).

Діаграма класів є однією з форм статичного опису системи з точки зору її проектування, показуючи її структуру. Діаграма класів не відображує динамічну поведінку об'єктів зображених на ній класів. На діаграмах класів показуються класи, інтерфейси і відносини між ними. На рисунку 2.6 представлена діаграма класів ІС «Training gym».

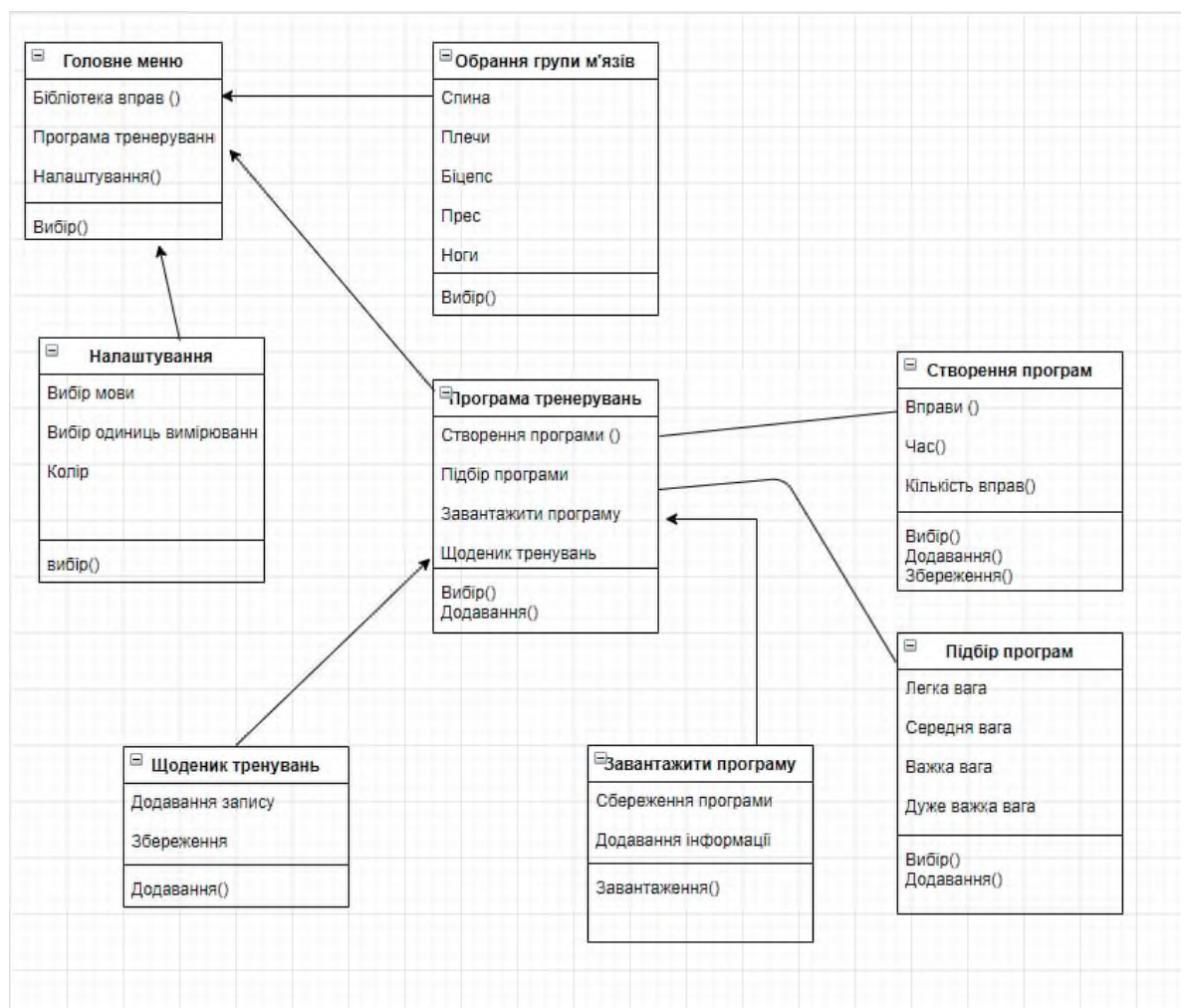


Рисунок 3.6 – Діаграма класів для ІС «Training gym»

2.2.3 Специфікація класів

Кожен із створених класів виконує свої функції. Короткий опис цих класів наведено нижче.

Назва: Головне меню.

Відповідальність: Перегляд до компонентів то форм перед переходом до наступного меню.

Атрибути: бібліотека вправ, програма тренувань, налаштування.

Операції зі специфікаціями нетривіальних операцій:

– Вибір() – служить для переходу в іншу форму.

Назва: Налаштування.

Відповідальність: даний клас призначений для налаштування додатку.

Атрибути: вибір мови, вибір одиниць вимірювання, колір.

Операції зі специфікаціями нетривіальних операцій:

– Вибір() – служить для вибору потрібного налаштування

Назва: обрання групи м'язів.

Відповідальність: даний клас призначений для вибору потрібного навантаження.

Атрибути: спина, плечі, біцепс, прес, ноги.

Операції зі специфікаціями нетривіальних операцій:

– Вибір() - служить для вибору потрібної групи м'язів.

Назва: програма тренувань.

Відповідальність: даний клас призначений для вибору програми тренувань, підбору програми та щоденник тренувань .

Атрибути: створення програми, підбір програми, щоденник тренувань.

Операції зі специфікаціями нетривіальних операцій:

– Вибір() – служить для вибору потрібної функції.

– Додавання() – служить для додавання нових тренувань

Назва: Щоденник тренувань.

					MP.122.6157M.03.P-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Відповідальність: даний клас призначений для додавання записів о тренерувані та збережені введених даних.

Атрибути: Додавання запису, збереження.

Операції зі специфікаціями нетривіальних операцій:

– Додавання() – служить для додавання записів.

Назва: Завантажити програму.

Відповідальність: даний клас призначений для завантаження вже збереженої інформації.

Атрибути: збереження програми, додавання інформації .

Операції зі специфікаціями нетривіальних операцій:

– Завантаження() – служить для завантаження вже збереженої інформації.

Назва: Підбір програм.

Відповідальність: даний клас призначений для автоматичного вибору спортивної програми.

Атрибути: легка вага, середня вага, важка вага, дуже важка вага.

Операції зі специфікаціями нетривіальних операцій:

– Вибір() - служить для вибору спортивної програми

– Додавання служить для додавання вправ в програму.

Назва: Створення програм.

Відповідальність: даний клас призначений для створення спортивної програми.

Атрибути: вправи, час, кількість вправ.

Операції зі специфікаціями нетривіальних операцій:

– Вибір() - служить для вибору вправ

– Додавання() –служить для додавання вправ

– Збереження() –служить для збереження програми

2.2.4. Динамічна модель програми

					MP.122.6157м.03.Р-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Діаграма послідовності відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень. Це відмінний засіб документування поведінки системи, деталізації логіки сценаріїв.

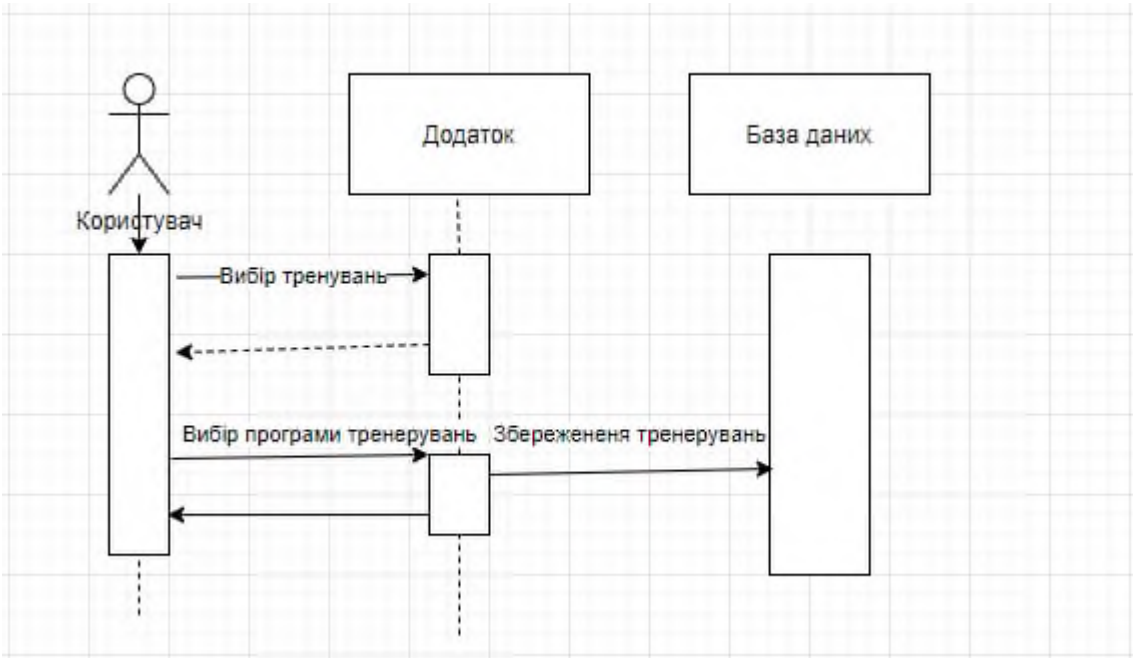


Рисунок 2.7 - Діаграма послідовності варіанту використання при вибору програми тренувань

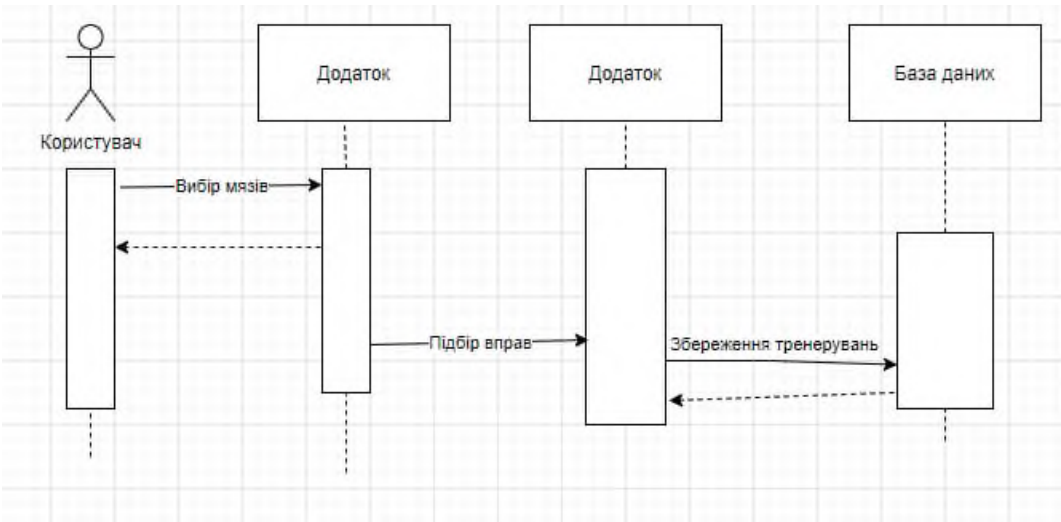


Рисунок 2.8 – Діаграма послідовності варіанту використання при створені програми тренувань

Висновки за розділом 2

В цьому розділі було спроектовано інформаційну систему в основу якого було створено програмне забезпечення «TRAINING GYM». В розділі були описано програмне забезпечення ігровий рушій Unity 3d та обрано мову програмування C# та база даних mySql.

Також були спроектовані: динамічна модель, діаграма класів, специфікація класів, діаграм використання, сценарії послідовності.

					MP.122.6157м.03.Р-02	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

					MP.122.6157м.03.P-01						
Змн.	Арк.	№ докум.	Підпис	Дата	РОЗРОБКА ІС ТА ЗАГАЛЬНОСИСТЕМНІ РІШЕННЯ				Лім.	Арк.	Акрушів
Розробив	Калниболотські										
Керівник	Гайдаєнко О.В..										
Консульт.	Латанська Л.О..								ХФ НУК		
Н. контр.											
Зав.кафедр	Гучек П.Й.										

3. РОЗРОБКА ІС ТА ЗАГАЛЬНОСИСТЕМНІ РІШЕННЯ

3.1 Логічна модель

3.1.1 Розробка логічної моделі

Жодна ІС не існує без бази даних в яку вноситься, зберігається, обробляється, редагується та представляється користувачеві інформація. Для цього потрібно розробити логічну та фізичну модель даних.

Логічна модель – це загальний погляд на дані, модель даних в певній предметній області інколи її називають концептуальною моделлю даних. Такою моделлю і є модель типу сутність-зв'язок.

Програмний пакет ERwin дозволяє створити логічну модель даних сутність-зв'язок у стандарті IDEF1x і на її основі побудувати фізичну базу даних практично для всякої СКБД. Більше того, він дозволяє зробити зворотне перетворення і з фізичної моделі побудувати логічну. Це досить цінна якість. Бази даних у наш час розповсюджені доволі широко і розроблялись вони в різних програмних середовищах. Використовуючи вказану властивість, можна переробити існуючі бази даних, розроблені в середовищі однієї СКБД в логічну модель, а пізніше зробити зворотне перетворення в середовище зовсім іншої СКБД з тих, які підтримує пакет ERwin. Тобто існуючі бази даних можна практично автоматично, не застосовуючи мови програмування, перекодувати в нове програмне середовище.

На логічній моделі можна побачити наступне: сутність, їх атрибути, типи даних, ключові атрибути і зв'язки між сутностями.

Нижче наведена логічна модель даних ІС «Training gym» на рисунку 3.1

Для побудови логічної моделі необхідно проаналізувати діаграму сутність-зв'язок та з'ясувати які таблиці та атрибути повинні бути реалізовані в ній. Логічна модель даних ІС «Training gym» зображена на рисунку 3.1. Та представлена у вигляді набору таблиці 2.1 .

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

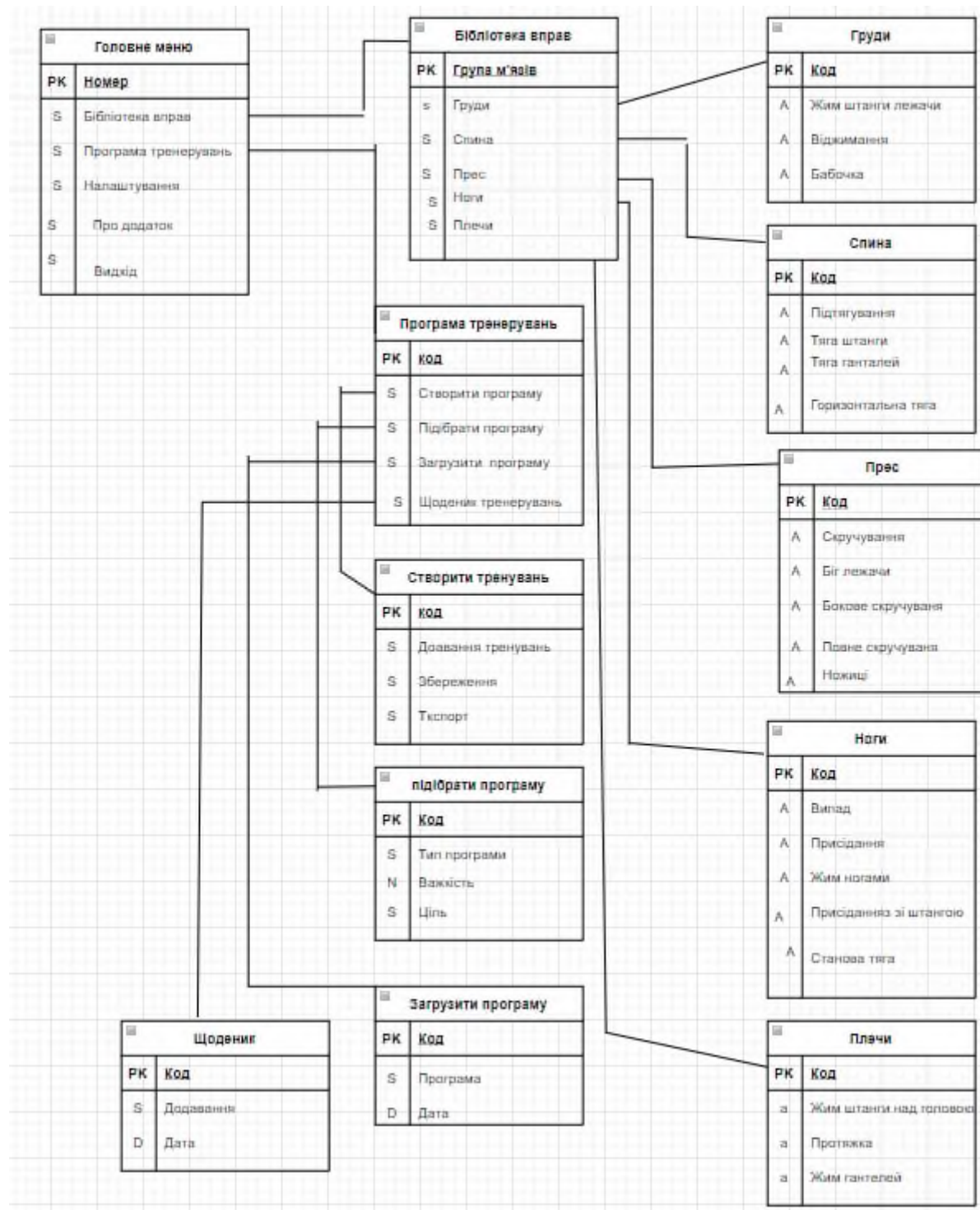


Рисунок 3.1 – Логічна модель ІС «Traning gym»

Таблиця 3.1 - Таблиці логічної моделі даних ІС «Traning gym»

Назва атрибуту	Тип даних	Ключ	Допустимість
1	2	3	4
Головне меню			
Номер	Number	PK	not null
Продовження таблиці 3.1			

1	2	3	4
Бібліотека вправ	String		not null
Програма тренувань	String		not null
Налаштування	String		not null
Про додаток	String		not null
Вихід	String		not null
Бібліотека вправ			
Група м'язів	Number	PK	not null
Груди	String		not null
Спина	String		not null
Прес	String		not null
Ноги	String		not null
Плечи	String		not null
Груди			
Код	Number	PK	not null
Жим штанги лежачи	Animation		not null
Віджимання	Animation		not null
Бабочка	Animation		not null
Спина			
Код	Number	Pk	not null
Підтягування	Animation		not null
Тяга штанги	Animation		not null
Тяга ганталей	Animation		not null
Горизонтальна тяга	Animation		not null
Прес			
Код	Number	PK	not null
Скручування	Animation		not null
Біг лежачи	Animation		not null
Бокове скручування	Animation		not null
Повне скручування	Animation		not null
Ножиці	Animation		not null
Ноги			
Код	Number	PK	not null
Випад	Animation		not null
Присідання	Animation		not null
Жим ногами	Animation		not null
Присідання зі штангою	Animation		not null
Станова тяга	Animation		not null
Плечи			
Жим над головою	Animation		not null
Протяжка	Animation		not null
Жим ганталей	Animation		not null
Продовження таблиці 3.1			

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

1	2	3	4
Програма тренувань			
Код	Number	PK	not null
Створити програму	String		not null
Підібрати програму	String		not null
Загрузит програму	String		not null
Щоденик тренувань	String		not null
Створити програму			
Код	Number	PK	not null
Додавання тренувань	String		not null
Збереження	String		not null
Експорт	String		not null
Підібрати програму			
код	Number	PK	not null
Тип програми	String		null
важкість	Number		null
ціль	String		null
Завантажити програму			
код	number	PK	not null
програма	String		not null
дата	datetime		not null
Щоденик тренувань			
код	number	PK	not null
Додавання	String		null
Дата	datetime		null

3.2 Фізична модель

Фізичні моделі даних служать для відображення моделей даних. Основними поняттями моделі даних є поле, логічна запис, логічний файл. Слово «логічний» введено, щоб відрізнити поняття, які стосуються логічної моделі даних, від понять, що відносяться до фізичної моделі даних.

Основними поняттями фізичної моделі даних, що використовуються для представлення логічної моделі даних, є поле, фізичний запис, фізичний файл. Зокрема, логічна запис, що складається з полів, може бути представлена у вигляді фізичного запису (з тих же полів), логічний файл - у вигляді фізичного файлу.

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Способи фізичної організації даних в різних СУБД, як правило, різняться визначаються типом використовуваного комп'ютера, інструментальними засобами розробки СУБД, а також критеріями, якими керуються розробники СУБД при виборі методів розміщення даних і способів доступу до цих даних. Зауважимо, що найбільш поширеним критерієм служить час доступу до даних, однак в якості критерію може вибиратися, наприклад, трудомісткість реалізації відповідних методів.

На рисунку 3.2 – зображену фізичну модель ІС «Training gym»

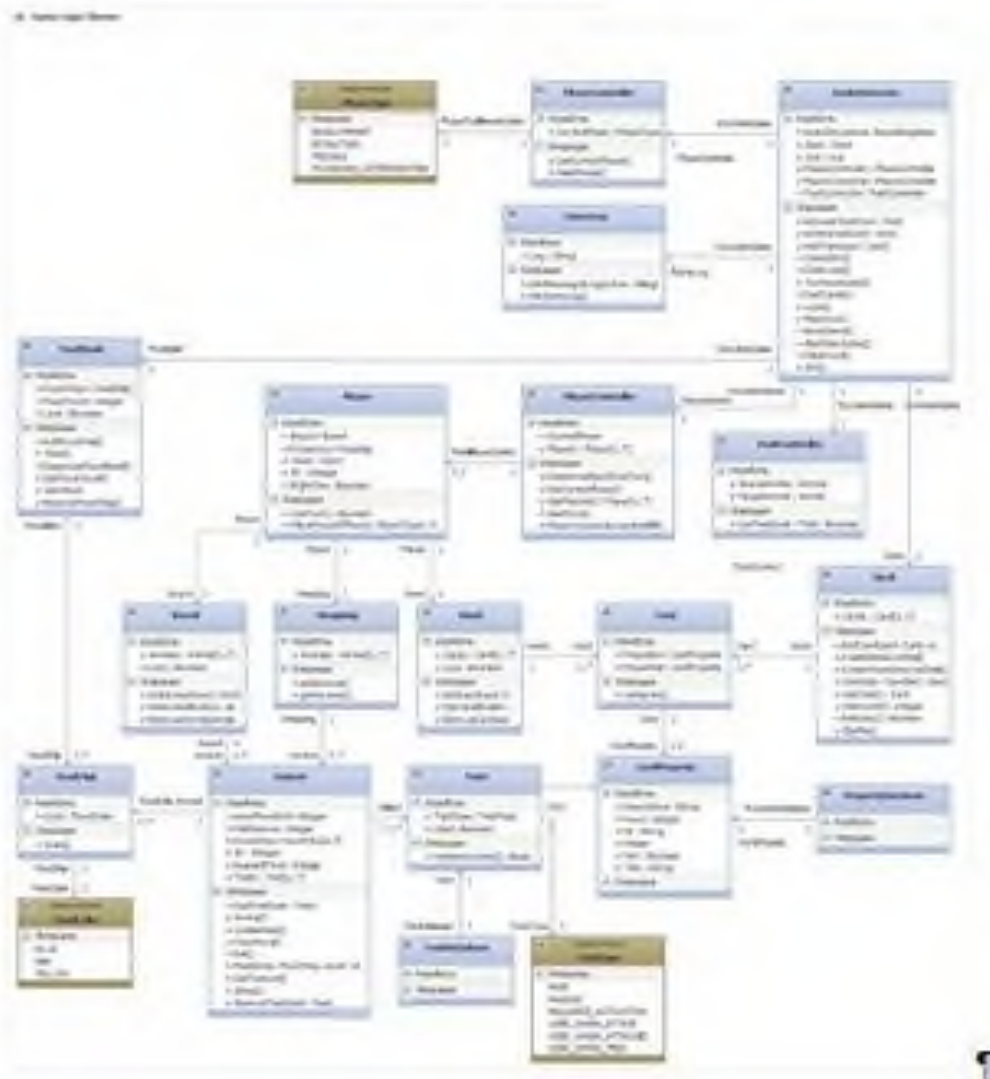


Рисунок 3.2 – Фізична модель ІС «Training gym»

3.3 Розгортання системи

Розгортання програмного забезпечення - всі дії, які роблять систему програмного забезпечення доступною для використання.

Загальний процес розгортання складається з декількох взаємопов'язаних дій з можливими переходами між ними. Ці дії можуть відбутися в стороні виробника або в споживчій стороні або обох. Оскільки кожна система програмного забезпечення унікальна, точні процеси або процедури в рамках кожної діяльності можуть ледь бути визначені. Тому, «розгортання» має інтерпретуватися як загальний процес, який повинен бути налаштований згідно певним вимогам або особливостям. Короткий опис кожної діяльності буде представлено пізніше.

На рисунку 3.3 – показано більше детальне розгортання інформаційної системи “training gym”

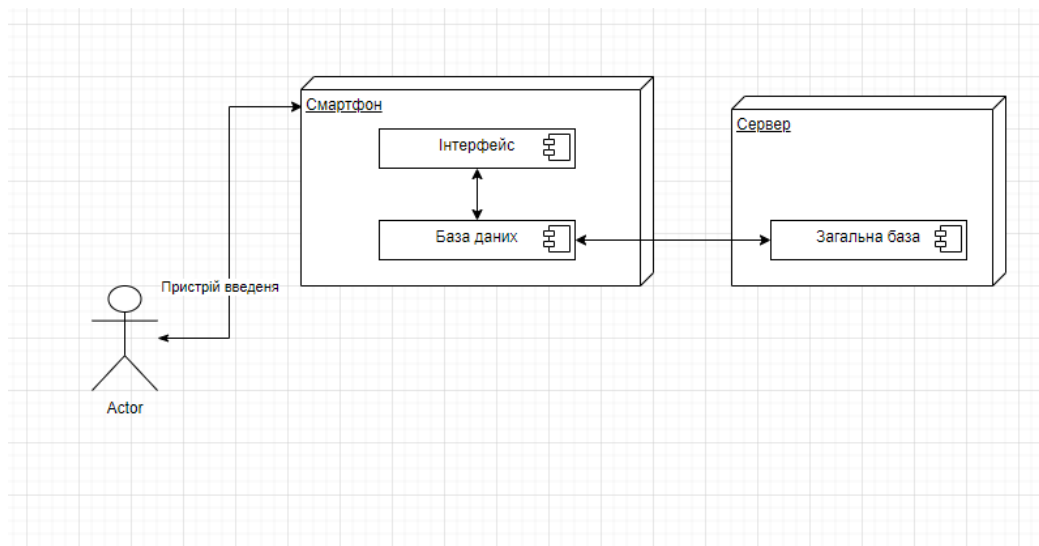


Рисунок 3.3 – Розгортання ІС “training gym”

Висновки за розділом 3

В розділі розроблені логічна та фізична модель додатку, а також приведено модель розгортання додатку «Training gym».

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

					MP.122.6157м.03.P-04						
Змн.	Арк.	№ докум.	Підпис	Дата	РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ ВІД ВПРОВАДЖЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ				Лім.	Арк.	Акрушіє
Розробив	Калниболотські										
Керівник	Гайдаєнко О.В..										
Консульт.	Ломоносов А.В..								ХФ НУК		
Н. контр.											
Зав.кафедр	Гучек П.Й.										

4 РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ ВІД ВПРОВАДЖЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ

4.1 Вступ до проблеми Інформаційні системи широко використовуються багатьма організаціями України різної форми власності. Ці системи розповсюджуються все більше і більше, змінюючись відповідно до вимог замовників та модернізуючись з плином часу.

Створення інформаційної системи вимагає одноразових витрат на її розробку та придбання необхідних технічних засобів, а також поточних витрат, пов'язаних з функціонуванням підсистеми. Економічний ефект від експлуатації системи визначається з урахуванням витрат на її експлуатацію. Відношення економічного ефекту до витрат на створення системи характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються за діючими на момент розрахунку цінами, тарифами та ставками заробітної плати.

4.2 Розрахунок вартості створення автоматизованої системи з волатильною бізнес-логікою

Загальні витрати (на проектування, реалізацію та впровадження системи)

$$З = С_{п-р} + С_{тз} + С_{пз} + С_{впр},$$

де $C_{п-р}$ – витрати на проектування і реалізацію;

$C_{тз}$ – витрати на придбання або модернізацію обчислювальної техніки, периферійних пристроїв, засобів зв'язку, допоміжного устаткування та оргтехніки (з урахуванням витрат на транспортування, монтаж, налагодження і запуск), виробничо-господарського інвентарю;

$C_{пз}$ – витрати на придбання та інсталяцію програмного забезпечення (ПЗ), необхідного для розробки;

$C_{впр}$ – витрати на впровадження системи (навчання персоналу тощо).

Витрати на проектування та реалізацію розраховуються за формулою $C_{п-р} = C_3 + C_{від} + C_m + C_n$,

де C_3 – заробітна плата розробників;

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

ВІД

С

– відрахування на соціальні заходи (відповідно до чинного законодавства ВІД С

становить 47,5% від фонду оплати праці);

М

С - експлуатація та амортизація, $C_m = Z_{\text{експл}} + Z_{\text{амор}}$;

Н

С – накладні витрати у розмірі 50-150% від витрат на оплату праці (охоплюють оплату праці управлінського персоналу з нарахуваннями; оплату службових відряджень; оплату консультаційно-інформаційних послуг; оплату ремонту і техобслуговування основних фондів, крім персональних комп'ютерів (ПК)).

1) Розробка підсистеми вимагає 6 місяців (експертна оцінка, яка базується на досвіді розробки систем такого класу). Заробітна платня розробників за місяць дорівнює 12000 грн (відповідно до сучасних розцінок та попиту на спеціалістів такого профілю). Кількість розробників – одна людина.

$C_z = 1 * 12000 * 6 = 72000$ грн.

$C_{\text{від}} = 72000 * 0,475 = 34200$ грн. 2) витрати на амортизацію ЕОМ, на якій виконується розробка: $З (Ц / T) (T / 12)$

амор еом сл разр

де $C_{\text{еом}}$ 6000 грн. – балансова вартість ЕОМ; T 5

сл

років – термін служби ЕОМ; T 0,5

разр

року – час, необхідний для розробки системи, виражений в місяцях, і тому ділиться на 12, щоб одержати число років.

Замор $(6000/5)0,5//600$ (грн.); 3) витрати на експлуатацію ЕОМ, на якій виконується розробка, полягають в оплаті споживаної нею електричної

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

енергії:

експл ЭВМ разр рд рч ст

З Р Т N N Э ,

де РЭВМ 0,4 кВт/год – потужність ЕОМ;

раТ= 6 міс. – тривалість розробки;

N 21

рд день – число робочих днів у місяці; N 8

рч годин – число годин у робочому дні;

ст

Э = 0,3073 грн – вартість 1 кВт електроенергії.

Зекспл 0,46.21.8.0,3073.123,90 (грн); С_м=600+123,90=723,90 (грн.)

С_н=72000*0,3=21600 (грн.) С_{п-р}=72000+34200+723,90+21600= 128 523,90

(грн.) Витрати на програмне забезпечення не закладаються у даний кошторис, оскільки в роботі системи планується використовувати безкоштовне ПЗ з відкритими вихідними кодами.

Витрати на апаратне забезпечення складають 19981 грн (придбання серверу та комутаційного обладнання і матеріалів).

До витрат на впровадження системи можна віднести витрати вартість навчання кадрів, витрати на монтаж і налаштування мережі, вартість устаткування, вартість програмного забезпечення.

Витрати на впровадження системи приведені в таблиці 4.1.

Таблиця 4.1 – Витрати на провадження системи

№

Найменування інвестицій Сума, грн. 1 Вартість устаткування 19981 2

Вартість програмного забезпечення - 3 Вартість підготовки кадрів 500 4

Монтаж і налаштування локальної мережі 300 Разом на впровадження

системи 20781 Таким чином, загальні витрати на проектування, реалізацію та

впровадження системи складають $З = С_{п-р} + С_{тз} + С_{пз} + С_{впр} = 128\,523,90 +$

$20781 = 149304,90$ (грн).

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Витрати на створення та експлуатацію системи в перший рік експлуатації $Z_{се} = Z + Z_{ер}$, де $Z_{ер}$ – експлуатаційні витрати на технічне забезпечення за рік.

$Z_{ер} = АОБ + Z_{м} + Z_{е}$, де АОБ – амортизаційні відрахування на устаткування; $Z_{м}$ – витрати на основні і допоміжні матеріали; $Z_{е}$ – витрати на електроенергію.

Амортизаційні відрахування на устаткування складають 60% балансової вартості в рік (за чинним законодавством) $Аоб = 20781 * 0,6 = 12468,6$ грн.

В масштабі підприємства річні витрати на основні та допоміжні матеріали (гнучкі диски, папір тощо) визначаються в розмірі 3% вартості основного устаткування

$$Z_{м} = 20781 * 0,03 = 623,43 \text{ грн.}$$

Річний обсяг робіт ПК у нормо-годинах $ФМ = 253,3 * ТС$,

де ТС – середнє місячне завантаження устаткування (близько 5 годин); 253,3 – середня кількість робочих днів у році.

$$Фм = 253,3 * 5 = 1266,5 \text{ годин.}$$

Витрати на електроенергію під час 1266,5 годин роботи устаткування в рік при потужності ЕОМ 0,4 кВт/год та ціні 0,3073 грн за 1 кВт електроенергії складуть

$$Z_{е} = 1266,5 * 0,3073 * 0,4 = 155,68 \text{ грн.}$$

Експлуатаційні витрати на технічне забезпечення за рік $Z_{ер} = 12468,6 + 623,43 + 155,68 = 13247,71$ грн.

Отже, у перший рік витрати на створення та експлуатацію системи складуть

$$Z_{се} = 149304,90 + 13247,71 = 162552,61 \text{ грн.}$$

4.3 Розрахунок економічної ефективності Економічна ефективність використання розподіленої інформаційної системи на базі технології хмарних обчислень полягає у наступному: зменшення часу на обслуговування системи за рахунок її відмовостійкості та автоматичного масштабування; можливість надання користувачам обчислювальних потужностей в оренду.

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Це зменшення відбудеться за рахунок зменшення складності користування системою, збільшення надійності, стабільності та рівня безпеки. Визначимо, що це зменшення складе 50% від сучасного.

Однак, не піддається прямій грошовій оцінці зменшення кількості помилкових та необережних рішень, підвищення оперативності керування, поліпшення організації роботи та своєчасне отримання необхідної інформації про пріоритетність варіантів рішень тощо.

Використання даної системи дозволяє вивільнити 0,2 робочого часу. Так як з системою працює 10 робітників, то система умовно вивільнить 2 робітників.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження системи вивільнить 2 робітників.

Річна платня за рахунок вивільнення робочого часу складе

п

$$C = 12000 \cdot 2 \cdot 12 - 13247,71 = 274752,29 \text{ грн.}$$

Річний економічний ефект розраховується за наступною формулою: $\Delta C E_{kp}$ год п де E_p – нормативний коефіцієнт економічної ефективності капітальних вкладень у галузь і для обчислювальної техніки приймається 0,5; k – додаткові капітальні вкладення з урахуванням витрат на проектування, створення і функціонування системи.

$$\Delta C E_{kp} = 274\,752,29 - 0,5 \cdot 162\,552,61 = 193\,475,98 \text{ (грн.)}$$

Строк окупності підсистеми

$$T = k / \Delta C E_{kp},$$

тобто

$$T = 162\,552,61 / 274\,752,29 = 0,6 \text{ (року).}$$

Таким чином, витрати на створення розподіленої інформаційної системи окупляться протягом 0,6 року.

4.4 Підсумки

В результаті розрахунків витрати на створення системи обліку робочого часу

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

(проектування, реалізація і впровадження) складуть 149304,90 грн; експлуатаційні витрати щорічно складатимуть 13247,71 грн. Загалом в перший рік витрати на модернізацію й експлуатацію складуть 162552,61 грн.

Вивільнюється 2 умовних фахівці.

Вивільнена річна зарплатня працівників складає 274752,29 грн.

Витрати на створення системи окупляться протягом 0,6 року.

За отриманими значеннями річного економічного ефекту та строку окупності можливо зробити висновок про економічну вигідність і обґрунтованість створення розподіленої інформаційної системи. Слід зазначити, що підвищити економічну ефективність цієї системи можна за рахунок запропонування стороннім користувачам обчислень як послуги. Таким чином можна буде скоротити строк, за який окупляться витрати на створення системи.

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

					MP.122.6157м.03.P-05						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розробив	Калниболотські				ОХОРОНА ПРАЦІ				Лім.	Арк.	Акрушіє
Керівник	Гайдаєнко О.В..										
Консульт.	Щедролосев А.В.								ХФ НУК		
Н. контр.											
Зав.кафедр	Гучек П.Й.										

5 ОХОРОНА ПРАЦІ

Питання охорони праці та здоров'я наших громадян у процесі їх трудової та будь-якої іншої діяльності повинні стати пріоритетними та увійти до розряду питань найвищого державного рівня, оскільки саме люди, їх життя та здоров'я є найбільшим багатством будь-якої держави.

Із вищезазначеного стає зрозумілим, що охорона праці відіграє важливу роль як соціальний чинник, оскільки, якими б вагомими не були трудові здобутки, вони не можуть компенсувати людині втраченого здоров'я, а тим більше життя — те й інше дається лише один раз. Необхідно пам'ятати, що внаслідок нещасних випадків та аварій гинуть на виробництві не просто робітники та службовці, на підготовку яких держава та підприємства витратили значні кошти, а перш за все люди — годувальники сімей, батьки та матері дітей.

Окрім соціального, охорона праці має, безперечно, важливе економічне значення — це й висока продуктивність праці, зниження витрат на оплату лікарняних, компенсацій за важкі та шкідливі умови праці тощо.

Охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних та лікувально-профілактичних заходів і засобів, спрямованих на збереження здоров'я та працездатності людини в процесі праці.

Законодавчими актами, що визначають основні положення про охорону праці, є загальні закони України, а також спеціальні законодавчі акти. До загальних законів, що визначають основні положення про охорону праці належать: Конституція України, Закони України «Про охорону праці», «Про охорону здоров'я», «Про пожежну безпеку», «Про використання ядерної енергії та радіаційний захист», «Про забезпечення санітарного та епідемічного благополуччя населення», «Про загальнообов'язкове державне соціальне страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності», Кодекс законів про працю України (КЗпП). Спеціальними законодавчими актами в галузі охорони праці є

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Державні нормативні акти про охорону праці, Державні стандарти Системи стандартів безпеки праці, Будівельні норми та правила, Санітарні норми, Правила технічної експлуатації електроустановок споживачів та інші нормативні документи.

В статті 43 Конституції України записано: «Кожен має право на працю, що включає можливість заробляти собі на життя працею, яку він вільно обирає, або на яку вільно погоджується», «Кожен має право на належні, безпечні і здорові умови праці, на заробітну плату, не нижчу від визначеної законом», «Використання праці жінок і неповнолітніх на небезпечних для їхнього здоров'я роботах забороняється».

Основоположним законодавчим документом у галузі охорони праці є Закон України «Про охорону праці», дія якого поширюється на всі підприємства, установи і організації незалежно від форм власності та видів їх діяльності, на усіх громадян, які працюють, а також залучені до праці на цих підприємствах.

5.1 Аналіз потенційно-небезпечних і шкідливих факторів у приміщенні обчислювального центру

У приміщенні по розрахунках з побутовими і промисловими споживачами в обчислювальному центрі виробничого управління теплової енергії можуть мати місце такі фізичні і психологічні шкідливі фактори, як - порушення стану мікроклімату, недостатня освітленість робочої зони, забруднення повітря на робочих місцях, виробничий шум та вібрація, електромагнітні випромінювання, електростатичні поля, іонний склад повітря, відсутність чи недолік природного світла, поразка електричним струмом, загоряння, монотонність праці, перенапруга очей, емоційні перевантаження.

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

5.1.1 Електробезпека

У приміщенні по розрахунках з побутовими і промисловими споживачами в обчислювальному центрі виробничого управління теплової енергії на кожному робочому місці використовується персональний комп'ютер та інші електричні прилади, які повинні бути встановлені дотримуючись вимог Правил влаштування електроустановок (ПВЕ - 86). Вимоги електробезпеки встановлені електронно-обчислюванні машини і персональні комп'ютери (ПК) відображені у ДНАОП 0.00-1.31-99. Відповідно до цього нормативного документу під час проектування систем електропостачання, монтажу основного електрообладнання та електричного освітлення будівель та приміщень для ПК необхідно дотримуватись вимог Правил влаштування електроустановок (ПВЕ), ГОСТ 12.1.006-84, ГОСТ 12.1.019-79, ГОСТ 12.1.030-81, ПТЕ, ПВЕ, ВСН 59-88 «Электрооборудование жилых и общественных зданий. Нормы проектирования», СН 357-77 «Инструкция по проектированию силового осветительного оборудования промышленных предприятий», Правил пожежної безпеки в Україні та інших нормативних документів, що стосуються штучного освітлення і електротехнічних пристроїв, а також вимог нормативно-технічної експлуатаційної документації заводу виробника. ПК, периферійні пристрої ПК та устаткування для обслуговування, ремонту та налагодження ПК, інше устаткування, електропроводи та кабелі за виконанням та ступенем захисту мають відповідати класу зони за ПВЕ, мати апаратуру захисту від струму короткого замикання та інших аварійних режимів.

Під час монтажу та експлуатації ліній електромережі необхідно повністю унеможливити виникнення електричного джерела згорання внаслідок короткого замикання та перевантаження проводів, обмежувати застосування проводів з легкозаймистою ізоляцією і, за можливістю, перейти на негорючу ізоляцію.

Лінія електромережі для живлення ПК, периферійних пристроїв ПК та устаткування для обслуговування, ремонту та налагодження ПК виконується як окрема групова три провідна мережа, шляхом прокладання фазового, нульового

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів і прокладається від стійки групового розподільчого щита, розподільчого пункту до розеток живлення.

Усі провідники повинні відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу провідників, температурному режиму та типам апаратури захисту, вимогам ПВЕ.

У приміщенні, де одночасно експлуатується або обслуговується більше п'яти персональних ПК, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

ПК, периферійні пристрої ПК та устаткування для обслуговування, ремонту та налагодження ПК повинні підключатися до електромережі тільки з допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників повинні мати спеціальні контакти для підключення нульового захисного провідника. Необхідно унеможливити з'єднання контактів фазових провідників з контактами нульового захисного провідника.

Неприпустимим є підключення ПК, периферійних пристроїв ПК та устаткування для обслуговування, ремонту та налагодження ПК до звичайної двопровідної електромережі, в тому числі - з використанням перехідних пристроїв.

Штепсельні з'єднання та електророзетки для напруги 12 В та 36 В за своєю конструкцією повинні відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В. Окрім того вони мають бути пофарбовані в колір, який візуально значно відрізняється від кольору штепсельних з'єднань, розрахованих на

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

напругу 127 В та 220 В.

Індивідуальні та групові штепсельні з'єднання та електророзетки необхідно монтувати на негорючих або важкогорючих пластинах з урахуванням вимог ПВЕ та Правил пожежної безпеки в Україні.

Електромережу штепсельних розеток для живлення персональних ПК, периферійних пристроїв ГІК та устаткування для обслуговування, ремонту та налагодження ПК при розташуванні їх уздовж стін приміщення прокладають по підлозі поряд зі стінами приміщення, як правило, в металевих трубах і гнучких металевих рукавах з відводами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання.

При розташуванні в приміщенні за його периметром до 5 персональних ПК, використанні трипровідникового захищеного проводу або кабелю в оболонці з негорючого або важкогорючого матеріалу дозволяється прокладання їх без металевих труб та гнучких металевих рукавів.

Електромережу штепсельних розеток для живлення персональних ПК, периферійних пристроїв ПК та устаткування для обслуговування, ремонту та налагодження ПК при розташуванні їх у центрі приміщення, прокладають у каналах або під знімною підлогою в металевих трубах або гнучких металевих рукавах. При цьому не дозволяється застосовувати провід і кабель в ізоляції з вулканізованої гуми та інші матеріали, що містять сірку. Відкрита прокладка кабелів під підлогою забороняється.

Металеві труби та гнучкі металеві рукави повинні бути заземлені. Заземлення повинно відповідати вимогам ДНАОП 0.00-1.21-98 "Правила безпечної експлуатації електроустановок споживачів".

Отвори в плитах для прокладання кабелів електроживлення виконуються безпосередньо в місцях встановлення устаткування відповідно до затвердженого технологічного плану розміщення устаткування та його технічних характеристик.

Для підключення переносної електроапаратури застосовують гнучкі

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

проводи в надійній ізоляції. Тимчасова електропроводка від переносних приладів до джерел живлення виконується найкоротшим шляхом без заплутування проводів у конструкціях машин, приладів та меблях. Доточувати проводи можна тільки шляхом паяння з наступним старанним ізолюванням місць з'єднання.

Є неприпустимими:

- експлуатація кабелів та проводів з пошкодженою або такою, що втратила захисні властивості за час експлуатації, ізоляцією;
- залишення під напругою кабелів та проводів з неізольованими провідниками;
- застосування саморобних подовжувачів, які не відповідають вимогам ПВЕ до переносних електропроводок;
- застосування для опалення приміщення нестандартного (саморобного) електронагрівального обладнання або ламп розжарювання;
- користування пошкодженими розетками, розгалужувальними та з'єднувальними коробками, вимикачами та іншими електроприладами, а також лампами, скло яких має сліди затемнення або випинання;
- підвішування світильників безпосередньо на струмопровідних проводах, обгортання електроламп і світильників папером, тканиною та іншими горючими матеріалами, експлуатація їх зі знятими ковпаками (розсіювачами);
- використання електроапаратури та приладів в умовах, що не відповідають вказівкам (рекомендаціям) підприємств-виготовлювачів.

5.1.2 Пожежна безпека

Пожежна небезпека — можливість виникнення та (або) розвитку пожежі в будь-якій речовині, процесі, стані. Слід зазначити, що пожеж безпечних не буває. Якщо вони і не створюють прямої загрози життю та здоров'ю людини, то завдають збитків довкіллю, призводять до значних матеріальних втрат. Коли людина перебуває в зоні впливу пожежі, то вона може потрапити під дію

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

наступних небезпечних та шкідливих факторів: токсичні продукти згорання; вогонь; підвищена температура середовища; дим; недостатність кисню; руйнування будівельних конструкцій; вибухи; витікання небезпечних речовин, що відбуваються внаслідок пожежі; паніка.

Залежно від особливостей виробничого процесу, крім загальних вимог пожежної безпеки, здійснюються спеціальні протипожежні заходи для окремих видів виробництв, технологічних процесів та промислових об'єктів. Для споруд та приміщень, в яких експлуатуються відео термінали та ПК такі заходи визначені «Правилами пожежної безпеки в Україні, ДНАОП 0.00-1.31.99» та іншими нормативними документами.

Для приміщень повинна бути визначена категорія з вибухопожежної і пожежної безпеки відповідно до ОНТП 24-86 "Определение категорий помещений и зданий по взрывопожарной и пожарной опасности", та клас зони згідно з ПВЕ. Відповідні позначення повинні бути нанесені на входні двері приміщення.

Будівлі і ті їх частини, в яких розташовуються ПК, повинні мати не нижче II ступеня вогнестійкості. Якщо відповідно до СНиП 2.09.02-85 ці приміщення повинні бути відокремленими від приміщень іншого призначення протипожежними стінами, то межа їх вогнестійкості визначається відповідно до СНиП 2.01.02-85.

Неприпустимим є розташування приміщень категорій А і Б (ОНТП 24-86), а також виробництв з мокрими технологічними процесами поряд з приміщеннями, де розташовуються ПК.

Приміщення для розрахунків зі споживачами за характеристикою речовин та матеріалів відноситься до категорії Д, тобто там знаходяться негорючі речовини та матеріали у холодному стані.

Сховища інформації слід розміщати у відокремлених приміщеннях, обладнаних негорючими стелажамі і шафами. Фальшпідлога повинна бути виготовлена з негорючих матеріалів. Міжпідлоговий простір під знімною

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

підлогою має бути оснащений системою автоматичної пожежної сигналізації та засобами пожежогасіння відповідно до вимог.

Звукопоглинальне облицювання стін та стель слід виготовляти з негорючих або важкогорючих матеріалів.

Приміщення повинно бути оснащене системою автоматичної пожежної сигналізації з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 м² площі приміщення.

Не рідше одного разу на квартал необхідно очищати від пилу агрегати та вузли, кабельні канали та простір між підлогами.

Для запобігання виникнення пожежі необхідно передбачити міри пожежної профілактики: дотримання протипожежних вимог при проектуванні й експлуатації систем вентиляції згідно СНиП 1.01.02-84; дотримання умов пожежної безпеки електроустановок згідно ПУЕ-84; наявність засобів оповіщення:

- пожежні повідомлювачі (ЛІПНУВ-1, РІП-105 2/1 і т.д.);
- установки пожежегасіння (АУП);
- інструкції з мір протипожежної безпеки, план евакуації людей і технічних засобів.

Для поліпшення умов пожежної безпеки в приміщенні по розрахунках з побутовими і промисловими споживачами обчислювального центру Виробничого управління водопровідно-каналізаційного господарства повинна бути встановлена підлога з непальних матеріалів, технологічно знімний; папір зберігається в металевій шафі; у наявності два вуглекислотних вогнегасники типу ОУ-5, а також два димових датчики; у машинному залі систематично проводиться вологе збирання і вентилявання приміщення.

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

5.1.3 Освітлення

Освітлення у приміщенні по розрахунках з побутовими і промисловими споживачами в обчислювальному центрі виробничого управління теплової енергії, використовується персональний комп'ютер, може здійснюватися природним і штучним освітленням. Приміщення з ПК повинні мати природне і штучне освітлення відповідно до СНиП 11-4-79 "Естественное и искусственное освещение". Освітлення може здійснюватися природним і штучним світлом. При недостатності природного освітлення використовується змішане освітлення.

Останнє є освітленням, при якому в світлий час доби використовується одночасно природне і штучне світло.

Штучне освітлення даного приміщення має бути обладнане системою загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, де переважають роботи з документами, допускається вживати систему комбінованого освітлення (додатково до загального освітлення встановлюються світильники місцевого освітлення).

Загальне освітлення має бути виконане у вигляді суцільних або переривчатих ліній світильників, що розміщуються збоку від робочих місць (переважно зліва) паралельно лінії зору працівників. При розташуванні відеотерміналів ПК за периметром приміщення лінії світильників штучного освітлення повинні розміщуватися локально над робочими місцями. Для загального освітлення необхідно застосовувати світильники із розсіювачами та дзеркальними екранними сітками або віддзеркалювачами, укомплектовані високочастотними пускорегулювальними апаратами (ВЧ ПРА). Застосування світильників без розсіювачів та екранних сіток забороняється.

Як джерело світла при штучному освітленні повинні застосовуватися, як правило, люмінесцентні лампи типу ЛБ. Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300-500 лк. У разі неможливості забезпечити даний рівень освітленості системою загального

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

освітлення допускається застосування світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрану та збільшення освітленості екрану більше ніж 300 лк.

Для штучного освітлення приміщень варто використовувати люмінесцентні лампи, тому що в них висока світлова віддача (до 75 лм/Вт і більш), тривалий термін служби (до 10000 годин), мала яскравість світної поверхні, близький до природного спектральний склад випромінюваного світла, що забезпечує гарну передачу кольору. Найбільш прийнятними для дисплейних приміщень є люмінесцентні лампи ЛБ (білого світла) і ЛТБ (тепло-білого світла) потужністю 40, 80 Вт.

Для виключення засліплювання екранів дисплеїв прямими світловими потоками світильники загального освітлення розташовують збоку від робочого місця, паралельно лінії зору оператора і стіні з вікнами. Таке розміщення світильників дозволяє робити їхнє послідовне включення в залежності від величини природної освітленості і виключає роздратування очей смугами світла, що чергуються, і тіні, що виникають при поперечному розташуванні світильників.

5.1.4 Виробничий шум та вібрація

Шум — одна з форм фізичної дії середовища, безпосередньо впливає на працездатність.

Джерелами шуму є: транспорт, насоси, двигуни, пневматичні та електричні інструменти, верстати, будівельна техніка. У нашому випадку також додається шум, який утворює людський натовп, оскільки торгівельні організації підприємства розміщені на ринках.

Для забезпечення нормованих рівнів шуму у виробничих приміщеннях та на робочих місцях застосовуються шумопоглинальні засоби, вибір яких обґрунтовується спеціальними інженерно-акустичними розрахунками.

Як засоби шумопоглинання повинні застосовуватися негорючі або

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

важкогорючі спеціальні перфоровані плити, панелі, мінеральна вата з максимальним коефіцієнтом звукопоглинання в межах частот 31,5 - 8000 Гц, або інші матеріали аналогічного призначення, дозволені для оздоблення приміщень органами державного санітарно-епідеміологічного нагляду. Крім того, необхідно застосовувати підвісні стелі з аналогічними властивостями.

Рівні вібрації під час виконання робіт з ПК у виробничих приміщеннях не повинні перевищувати допустимих значень, визначених в СН 3044-84 та ГОСТ 12.1.012-90 “Вибрационная безопасность. Общие требования”.

Для зниження вібрації обладнання, пристрої, пристосування необхідно встановлювати на спеціальні амортизуючі прокладки, передбачені нормативними документами.

5.1.5 Регулювання параметрів мікроклімату

Суттєвий вплив на стан організму працівника, його працездатність здійснює мікроклімат у приміщенні, що визначається діючою на організм людини сукупністю температури, вологості, руху повітря та теплового випромінювання нагрітих поверхонь.

Приміщення по розрахунках з побутовими і промисловими споживачами в обчислювальному центрі виробничого управління теплової енергії повинно бути обладнане системами опалення, кондиціювання повітря або припливно-витяжною вентиляцією відповідно до СНиП 2.04.05-91 “Вентиляция производственных помещений”.

Параметри мікроклімату, іонного складу повітря, вміст шкідливих речовин на робочих місцях, оснащених відеотерміналами, повинні відповідати вимогам пункту 2.4 СН 4088-86 “Санитарные нормы микроклимат производственных помещений”, ГОСТ 12.1.005-88 “ССБТ Общие санитарно-гигиенические требования к воздуху рабочей зоны” (таблиця 4.1), СН 2152-80 “Санитарно-гигиенические нормы допустимых уровней ионизации воздуха производственных и общественных помещений” (таблиця 4.2).

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 5.1 - Нормовані параметри мікроклімату для приміщень з ВДТ та ППК

Пора року	Категорія робіт згідно з ГОСТ 12.1-005-88	Температура повітря, °С оптимальна	Відносна вологість повітря, % оптимальна	Швидкість руху повітря, м/с оптимальна
Холодна	Легка - 1 а	22-25	50-60	0,1
	Легка - 1б	21-23	40-60	0,1
Тепла	Легка - 1 а	23-25	40-60	0,1
	Легка - 1 б	22-24	40-60	0,2

Таблиця 4.2 - Рівні іонізації повітря приміщень при роботі на ВДТ та 111К (відповідно до СН 2152-80)

Рівні	Кількість іонів в 1 см ³ повітря	
	n ⁺	n ⁻
Мінімально необхідні	400	600
Оптимальні	1500 - 3000	3000 - 5000
Максимально допустимі	50000	50000

Для підтримки допустимих значень мікроклімату та концентрації позитивних та негативних іонів необхідно передбачати установки або прилади зволоження та/або штучної іонізації, кондиціонування повітря.

4.1.6 Організація робочого місця

Робоче місце - це місце постійного або тимчасового перебування працівника в процесі трудової діяльності. Організація

робочого місця користувача приміщення по розрахунках з побутовими і промисловими споживачами в обчислювальному центрі Виробничого управління теплової енергії повинна забезпечувати відповідність усіх елементів

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

робочого місця та їх розташування ергономічним вимогам ГОСТ 12.2.032-78 “ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования”.

Площа, виділена для одного робочого місця з відео терміналом або персональною ПК, повинна складати не менше 6 м², а обсяг - не менше 20 м³.

Робочі місця з відео терміналами відносно світлових прорізів повинні розміщуватися так, щоб природне світло падало збоку, переважно зліва.

При розміщенні робочих місць необхідно дотримуватись таких вимог:

робочі місця розміщуються на відстані не менше 1 м від стін зі світловими прорізами;

відстань між бічними поверхнями відео терміналів має бути не меншою за 1,2 м;

відстань між тильною поверхнею одного відео термінала та екраном іншого не повинна бути меншою 2,5 м;

прохід між рядами робочих місць має бути не меншим 1 м.

Вимоги щодо відстані між бічними поверхнями відео терміналів та відстані між тильною поверхнею одного відео термінала та екраном іншого враховуються також при розміщенні робочих місць з відео терміналами та персональними комп'ютерів в суміжних приміщеннях, з урахуванням конструктивних особливостей стін та перегородок.

Висота робочої поверхні столу для відео термінала має бути в межах 680-800 мм, а ширина - забезпечувати можливість виконання операцій в зоні досяжності моторного поля.

Робочий стіл для відео термінала, як правило, має бути обладнаний підставкою для ніг шириною не менше 300 мм та глибиною не менше 400 мм, з можливістю регулювання по висоті в межах 150 мм та кута нахилу опорної поверхні - в межах 20°. Підставка повинна мати рифлену поверхню та бортик на передньому краї заввишки 10 мм.

Робоче сидіння (стілець, крісло) користувача відео термінала та

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

персональної ПК повинно мати такі основні елементи: сидіння, спинку та стаціонарні або знімні підлокітники, також повинно бути підйомно-поворотним, таким, що регулюється за висотою, кутом нахилу сидіння та спинки, за відстанню спинки до переднього краю сидіння, висотою підлокітників. Поверхня сидіння має бути плоскою, передній край - заокругленим. Висота спинки сидіння має становити 300 ± 20 мм, ширина - не менше 380 мм, радіус кривизни в горизонтальній площині - 400 мм. Кут нахилу спинки повинен регулюватися в межах $0-30^\circ$ відносно вертикального положення. Відстань від спинки до переднього краю сидіння повинна регулюватись у межах 260-400 мм.

Екран відео термінала та клавіатура мають розташовуватися на оптимальній відстані від очей користувача, але не ближче 600 мм, з урахуванням розміру алфавітно-цифрових знаків та символів.

Клавіатуру слід розміщувати на поверхні столу або на спеціальній, регульовуваній за висотою, робочій поверхні окремо від столу на відстані 100-300 мм від краю, ближчого до працівника. Кут нахилу клавіатури має бути в межах $5-15^\circ$.

4.2 Розрахунок освітлення

4.2.1 Розрахунок природного освітлення

Зробити розрахунок КПО попереднім методом для виробничої лабораторії з боковим одностороннім освітленням. Приміщення розташоване в 4 поясі світлового клімату. Довжина приміщення: 13м, глибина: 8м. Розрахункова точка А знаходиться від зовнішньої стіни на відстані: 7 м. Відстань від робочої поверхні до верха вікон: 4.4 м.

Нормоване значення коефіцієнту бокового природного освітлення для III розряду зорової роботи III світлового поясу $e_n^{III} - 1.5\%$. У зв'язку з тим, що приміщення знаходиться в IV світловому поясі, КПО буде визначено за формулою:

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

$$e_n^{IV} = e_n^{III} m c \quad (5.2.1)$$

де $m = 0,9$ —коефіцієнт світлового клімату;

$c = 0,7$ - коефіцієнт сонячності клімату.

$$e_n^{IV} = 1.5 \times 0.9 \times 0.7 = 0.945 \%$$

Загальний коефіцієнт світло пропускання визначається за формулою:

$$\tau_0 = \tau_1 \tau_2 \tau_3 \tau_4 \tau_5 \quad (4.2.2)$$

де $\tau_1 = 0,8$ - засклення подвійне;

$\tau_2 = 0,65$ - Плетення подвійне дерев'яне;

$\tau_3 = 1$ - Несущі конструкції відсутні;

$\tau_4 = 1$ - Сонце захисні конструкції відсутні

$\tau_5 = 1$ - Захисних сіток немає

$$\tau_0 = 0.8 \times 0.65 \times 1 \times 1 \times 1 = 0,52$$

Площа світлових прорізів при боковому освітленні визначається за формулою:

$$S_{\sigma} = \frac{S_n e_n K_3 \eta_{\sigma} K_6}{100 r_1 \tau_0} \quad (4.2.3)$$

де S_n - площа підлоги приміщення,

$$S_n = 13 \times 8 = 104 \text{ м}^2;$$

$K_3 = 1,2$ - коефіцієнт запасу;

$\eta_{\sigma} = 13$ - світлова характеристика вікон;

$K_6 = 1$ - коефіцієнт, що враховує затінення вікон будинками, що стоять навпроти;

$r_1 = 2,4$ - коефіцієнт, що враховує підвищення КПО при боковому освітленні завдяки світлу, відбитому від поверхонь приміщення і підстильного шару, що прилягає до будинку.

$$S_{\sigma} = \frac{104 \times 1,26 \times 1,2 \times 13 \times 1}{100 \times 2,4 \times 0,52} = 16,38 \text{ м}^2$$

Загальна ширина вікон знаходиться за допомогою наступної формули:

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

$$b_{\epsilon} = \frac{S_{\epsilon}}{h_{\epsilon}} \quad (4.2.4)$$

де $h_B = 4$ м - висота вікна

$$b_{\epsilon} = \frac{8,24}{4} = 2,06 \text{ м}$$

Приймається 4 вікна, які мають ширину 1 м.

Висновок

У даному розділі проведено аналіз потенційно - небезпечних та шкідливих факторів у приміщенні по розрахунках зі споживачами в обчислювальному центрі інтернет-провайдера. Розроблені заходи щодо усунення або зниження впливу цих факторів на робітників до нормативних вимог. Дотримання умов, що визначають оптимальну організацію робочого місця оператора, дозволить зберегти гарну працездатність протягом усього робочого дня, що у свою чергу підвищить як у кількісному, так і в якісному відношеннях продуктивність праці.

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Перелік використаної літератури

1. Жидецький В.Ц. Основи охорони праці. Підручник. – Львів: Афіша, 2002. – 320 с.
2. Жидецький В.Ц. Охорона праці користувачів комп'ютерів . Навчальний посібник. – Вид.2-ге доп. – Львів: Афіша, 2000. – 176 с.
3. ДНАОП 0.00 – 1.31 – 99. Правила охорони праці під час експлуатації електронно-обчислювальних машин. – К., 1999. – 111 с.
4. Охорона праці. Лабораторний практикум / М.П. Купчик, М.П. Гандзюк, І.Ф. Гандзюк, І.Ф. Степанєв, В.Н Вендиханський, А.М. Литвиненко, О.В Іваненко. – К.: Основа, 1998. - 224 с.
5. Тубальцев А.М. Виробниче освітлення та його розрахунок: навчальний посібник. – Миколаїв: УДМТУ, 2001. – 84 с.
6. Щедролосєв О.В., Моїсеєнко Л.Л., Яглицький Ю.К. Основи охорони праці. Лабораторний практикум. – Херсон: Айлант, 2005. – 84.

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

					MP.122.6157м.03.P-06						
Змн.	Арк.	№ докум.	Підпис	Дата	ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА				Лім.	Арк.	Акрушів
Розробив	Калниболотськи										
Керівник	Гайдаєнко О.В..										
Консульт.	Щедролосев А.В.								ХФ НУК		
Н. контр.											
Зав.кафедр	Гучек П.Й.										

6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вступ до проблеми Охорона навколишнього середовища здійснюється різними, у тому числі й правовими, засобами. При цьому в правових формах захищаються переважно всі компоненти, які утворюють природне середовище.

Сучасними головними нормативно-правовими актами що регулюють основи організації охорони навколишнього природного середовища, є Закони України «Про охорону навколишнього природного середовища» від 25 червня 1991 р., «Про охорону атмосферного повітря» від 16 жовтня 1992 р., «Про природно-заповідний фонд України» від 16 червня 1992 р., «Про тваринний світ» від 3 березня 1993 р., «Про карантин рослин» від 30 червня 1993 р та інші.

До того ж деякі відносини у сфері використання і охорони навколишнього природного середовища врегульовані кодексами (земельним, водним, лісовим, про надра), а також Законами України «Про плату за землю» від 3 липня 1992 р., «Про ветеринарну медицину» від 25 червня 1992 р. Важливе значення у вирішенні цього питання має затверджений Постановою Верховної Ради «Порядок обмеження, тимчасової заборони (зупинення) чи припинення діяльності підприємств, установ, організацій і об'єктів у разі порушення ними законодавства про охорону навколишнього природного середовища».

Різновидами права природокористування є: право землекористування, право водокористування, право лісокористування, право користуватися надрами, право користуватися тваринним світом, право користування природно-заповідним фондом. Право природокористування це процес раціонального використання людиною природних ресурсів для задоволення різних потреб та інтересів. Найважливішими принципами природокористування є його цільовий характер, плановість і тривалість, ліцензування, врахування надзвичайного значення у житті суспільства тощо. При цьому виділяють такі групи природокористування, як право загального і спеціального використання землі, вод, лісів, надр, тваринного світу та інших природних ресурсів. Суб'єктами права загального користування природними ресурсами можуть бути,

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

згідно з Законом України «Про охорону навколишнього природного середовища», усі громадяни для задоволення найрізноманітніших потреб і інтересів. Воно здійснюється громадянами безкоштовно і безліцензійно, тобто для цього не потрібен відповідний дозвіл уповноважених органів і осіб. Загальним є, наприклад, використання парків, скверів, водойм, лісів, збір дикорослих ягід, грибів, горіхів і т. ін. Право загального природокористування закріплене у ст. 13 Конституції України: «Кожний громадянин має право користуватися природними об'єктами права власності народу відповідно до закону». Похідним від загального природокористування є спеціальне використання природних ресурсів. На відміну від першого, це використання конкретних природних ресурсів, що здійснюється громадянами, підприємствами, установами і організаціями у випадках, коли відповідна, визначена у законодавстві частина природних ресурсів передається їм для використання. Як правило, така передача має вартість і визначена в часі. Надання природних ресурсів відбувається на основі спеціальних дозволів державних актів на право постійного користування.

Крім прав суб'єктів, як природокористувачів, сучасною юридичною наукою сформовані й інтенсивно розвиваються екологічні права і обов'язки. Так, у Конституції України записано, що «кожен має право на безпечне для життя і здоров'я довкілля та на відшкодування завданої порушенням цього права шкоди.

Кожному гарантується право вільного доступу до інформації про стан довкілля, про якість харчових продуктів і предметів побуту, а також право на її поширення». Аналогічні формулювання є й у Законі України «Про охорону навколишнього природного середовища», бо це право одне з основних прав людини. Цьому праву відповідає обов'язок держави забезпечувати здійснення санітарно-гігієнічних заходів, спрямованих на поліпшення та оздоровлення навколишнього природного середовища. Усі екологічні права громадян захищаються і відновлюються у судовому порядку. Поряд з правами Закон

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

України «Про охорону навколишнього природного середовища» передбачає і певні обов'язки громадян. Так, незалежно від того, є громадяни природокористувачами, чи ні, вони зобов'язані берегти природу, раціонально використовувати її запаси, не завдавати шкоди. Крім того, Закон України «Про охорону навколишнього природного середовища» покладає на громадян і підприємства, установи й організації, як суб'єктів спеціального використання природних ресурсів, спеціальні обов'язки. Так, плата за спеціальне природокористування встановлюється на основі нормативів плати і лімітів використання природних ресурсів. Ці нормативи визначаються з урахуванням розповсюдження природних ресурсів, їх якості, можливості використання, місцезнаходження, можливості переробки і зберігання відходів. До того ж суб'єкти спеціального природокористування зобов'язані сплачувати певні кошти за забруднення навколишнього природного середовища, що встановлюються за викиди у атмосферу забруднюючих речовин; скидання забруднюючих речовин на поверхню води, у територіальні і морські води, а також під землю.

Контроль у сфері природовикористання і охорони навколишнього природного середовища здійснюється шляхом перевірки, нагляду, обстеження, інвентаризації та експертиз. Він може здійснюватись як уповноваженими державними органами, так і громадськими формуваннями. Державний контроль покладається на Ради народних депутатів, державні адміністрації та Міністерство охорони навколишнього природного середовища і його органи на місцях.

До основних пріоритетів охорони довкілля та раціонального використання природних ресурсів належать: гарантування екологічної безпеки ядерних об'єктів і радіаційного захисту населення та довкілля, зведення до мінімуму шкідливого впливу наслідків аварії на Чорнобильській АЕС; поліпшення екологічного стану басейнів рік України та якості питної води; стабілізація та поліпшення екологічного стану в містах і промислових центрах Донецько-Придніпровського регіону; будівництво нових та реконструкція діючих

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

потужностей комунальних очисних каналізаційних споруд; запобігання забрудненню Чорного та Азовського морів і поліпшення їх екологічного стану;

- формування збалансованої системи природокористування та адекватна структурна перебудова виробничого потенціалу економіки, екологізація технологій у промисловості, енергетиці, будівництві, сільському господарстві, на транспорті;

- збереження біологічного та ландшафтного різноманіття, заповідна справа.

Для досягнення цього передбачається вирішення таких завдань:

1. зменшення до мінімуму рівня радіаційного забруднення;
2. захист повітряного басейну від забруднення, насамперед у великих містах і промислових центрах;
3. захист і збереження земельних ресурсів від забруднення, виснаження і нераціонального використання;
4. збереження і розширення територій з природним станом ландшафту, посилення природоохоронної діяльності на заповідних і рекреаційних територіях;
5. підвищення стійкості та екологічних функцій лісів;
6. знешкодження, утилізація та захоронення промислових та побутових відходів;
7. запобігання забрудненню морських і внутрішніх вод, зменшення та припинення скиду забруднених стічних вод у водні об'єкти, захист підземних вод від забруднення;
8. збереження та відродження малих річок, здійснення управління водними ресурсами на основі басейнового принципу;
9. завершення створення державної системи моніторингу навколишнього природного середовища;
10. створення системи прогнозування, запобігання та оперативних дій у разі надзвичайних ситуацій природного і природно-техногенного походження;

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

11.забезпечення екологічного супроводу процесу конверсії військово-промислового комплексу; 12.здійснення заходів щодо екологічного контролю за діяльністю Збройних Сил України; 13.розробка механізмів реалізації схем природокористування;

14.впровадження дійових економічних складових впливу на систему природокористування;

15.створення системи екологічної освіти, виховання та інформування.

6.2 Забруднення навколишнього середовища підприємством з розробки програмного забезпечення

Основними чинниками забруднення навколишнього середовища фірмою є лише використання автотранспорту. Бо підприємство окрім забруднення повітряного середовища не забруднює ні водне, ні ґрунтове.

В сучасних умовах автомобільний транспорт стає найбільш значним джерелом забруднення атмосферного повітря, особливо великих міст.

Відомо, що атмосферне повітря міст постійно забруднюється і за всіма параметрами докорінно відрізняється від повноцінного природного повітря.

Міські поселення характеризуються найвищими рівнями антропогенних навантажень на навколишнє середовище, в результаті чого воно деформується, набуває якісно нових рис, аж до зміни мікрокліматичних факторів і фізико-хімічних властивостей середовища, зокрема повітряного басейну.

Метеорологічні спостереження свідчать, що температура повітря в межах міських територій у середньому на декілька градусів вища, ніж у приміських районах. Над містами, особливо великими, частіше випадають атмосферні опади, бувають густі тумани, а також смоги – густі тумани, змішані з димом, кіптявою та вихлопними газами. Прозорість атмосфери в містах менша, ніж за її межами і в сільській місцевості. Тумани, а також запиленість повітря помітно зменшують проникнення до земної поверхні сонячних променів.

Однак за рахунок автомобільних викидів якість атмосферного повітря в містах погіршилась.

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Прогресуючому забрудненню атмосфери в містах сприяє висока питома вага автомобілів, адже зростання кількості автотранспортних засобів супроводжується збільшенням об'ємів забруднюючих речовин із вихлопних труб. Останнім часом у міському повітрі виріс об'єм оксидів вуглецю, вуглеводнів, оксидів азоту, сажі. Але найбільшу небезпеку, окрім оксидів азоту, складають сірчані та свинцеві сполуки. Їх вміст у міському повітрі значною мірою виріс. Місто не пристосоване до такої кількості автотранспорту. Довжина пробігу без зупинок між світлофорами становить лише 400–600 м., внаслідок цього середня швидкість руху вдень у центрі міста і на великих автошляхах знижується до 12–20 км/год, а це збільшує витрати палива в 3–4 рази.

Відповідно збільшуються й викиди забруднюючих речовин. Автотранспорт також призводить до специфічних форм забруднення повітря. При русі стираються шини і тисячі тонн гуми у вигляді пилу потрапляє в повітря.

Автомобільний транспорт не тільки забруднює повітря продуктами згорання палива, він сприяє зростаючому надходженню свинцю в навколишнє середовище. В Україні поки ще використовують бензин із вмістом свинцю 0,36 г/л, тоді як в Англії, Німеччині та США – 0,013–0,15.

6.3 Розробка заходів щодо зменшення забруднення

Природоохоронною є будь-яка діяльність, спрямована на збереження якості навколишнього середовища на рівні, що забезпечує стійкість біосфери. До неї відноситься як великомасштабна, здійснювана на загальнодержавному рівні діяльність по збереженню еталонних зразків недоторканої природи і збереженню розмаїтості видів на Землі, організації наукових досліджень, підготовці фахівців-екологів і вихованню населення, так і діяльність окремих підприємств по очищенню від шкідливих речовин стічних вод і відхідних газів, зниженню норм використання природних ресурсів і т.д. Така діяльність здійснюється в основному інженерними методами.

Існують два основних напрямки природоохоронної діяльності

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

підприємств.

Перший – очищення шкідливих викидів. Цей шлях «у чистому вигляді» малоефективний, тому що з його допомогою далеко не завжди вдається цілком припинити надходження шкідливих речовин у біосферу. До того ж скорочення рівня забруднення одного компонента навколишнього середовища веде до посилення рівня забруднення іншого .

І, наприклад, встановлення вологих фільтрів при газоочищенні дозволяє скоротити забруднення повітря, але веде до ще більшого забруднення води.

Уловлені з відхідних газів і зливальних вод речовини часто отруюють значні земельні площі.

Використання очисних споруд, навіть найефективніших, різко скорочує рівень забруднення навколишнього середовища, однак не вирішує цієї проблеми цілком, оскільки в процесі функціонування цих установок теж виробляються відходи, хоча й у меншому обсязі, але, як правило, з підвищеною концентрацією шкідливих речовин. Нарешті, робота більшої частини очисних споруджень вимагає значних енергетичних витрат, що, у свою чергу, теж небезпечно для навколишнього середовища.

Крім того, забруднювачі, на знешкодження яких йдуть величезні засоби, являють собою речовини, на які уже витрачена праця і які за рідкісним винятком можна було б використовувати в народному господарстві.

Для досягнення високих еколого-економічних результатів необхідно процес очищення шкідливих викидів сполучити з процесом утилізації уловлених речовин, що уможливить об'єднання першого напрямку з другим.

Другий напрямок – усунення самих причин забруднення, що вимагає розробки маловідходних, а в перспективі і безвідходних технологій виробництва, які дозволяли б комплексно використовувати вихідну сировину й утилізувати максимум шкідливих для біосфери речовин [8].

Однак далеко не для усіх виробництв знайдені прийнятні техніко-

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

економічні рішення по різкому скороченню кількості відходів, що утворюються, і їх утилізації, тому на даний час доводиться працювати в обох зазначених напрямках.

Піклуючись про удосконалювання інженерної охорони навколишньої природного середовища, треба пам'ятати, що ніякі очисні спорудження і безвідхідні технології не зможуть відновити стійкість біосфери, якщо будуть перевищені припустимі (граничні) значення скорочення природних, не перетворених людиною природних систем, у чому виявляється чинність закону незамінності біосфери.

Таким порогом може виявитися використання більш 1% біосфери і глибоке перетворення більш 10% природних територій (правила одного і десяти відсотків). Тому технічні досягнення не знімають необхідності рішення проблем зміни пріоритетів суспільного розвитку, стабілізації народонаселення, створення достатнього числа заповідних територій і інших, розглянутих раніше. Основними напрямками робіт в галузі захисту атмосфери від забруднення викидами автотранспорту є:

- а) створення і розширення виробництва автомобілів з високоекономічними і малотоксичними двигунами, у тому числі подальша дизелізація автомобілів;
- б) розвиток робіт зі створення і впровадження ефективних систем нейтралізації відпрацьованих газів;
- в) зниження токсичності моторних палив;
- г) розвиток робіт з раціональної організації руху автотранспорту в містах, удосконаленню дорожнього будівництва з метою забезпечення невинного руху на автомагістралях.

					МР.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В результаті виконання дипломної роботи було проведено наукове дослідження моделей ігрових рушіїв та використання їх при розробці ПЗ. Було розроблено спортивний додаток на ігровому двигуні “UNITY 3D”

В рамках дипломної роботи був проведений аналіз технологій проектування та використання ігрових рушіїв, була побудована модель предметної області, виділені основні концептуальні класи. Був проведений аналіз вимог, побудована діаграма варіантів використання і визначена функціональність системи, проектував інтуїтивний призначений для користувача інтерфейс. На основі аналізу предметної області та діаграми варіантів використання спроектована структура класів системи.

В ході роботи були вивчені всі необхідні принципи моделювання та створення комп’ютерних анімацій задля наглядного перегляду різних вправ.. Готовий додаток відповідає всім вимогам і виконує всі поставлені завдання.

Даний додаток дозволяє виробляти переглядати вже готові спортивні вправи створювати комплекси своїх вправ, та використовувати вже готові системи тренувань.

Цілі розробки даної системи виконані: дослідження моделей проектування та використання ігрового рушія, аналіз та використання комп’ютерних моделей. Створено спортивний додаток завдяки котрому можна займатись спортом самостійно.

					MP.122.6157м.№3	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРАСТАНИХ ДЖЕРЕЛ

1. Історія комп'ютерних ігор [Електронний ресурс] Режим доступу https://ru.wikipedia.org/wiki/История_компьютерных_игр (дата звернення 10.11.2020)
2. Ігровий рушій [Електронний ресурс] Режим доступу https://uk.wikipedia.org/wiki/Ігровий_рушій (дата звернення 11.11.2020)
3. Метод кінцевих елементів: теорія, алгоритми, реалізація / В.А.Толок [и др.]. К .: Наук. думка, 2003. 316 с.
4. Pharr, Matt; Humphreys, Greg (2004). Physically based rendering from theory to implementation. Amsterdam: Elsevier/Morgan Kaufmann.
5. Олексій Боресков. Розробка та налагодження шейдерів. - БХВ-Петербург, 2006
6. Боресков А. В. Расширения OpenGL. — БХВ-Петербург, 2005.
7. Шейдер [Електронний ресурс] <https://uk.wikipedia.org/wiki/Шейдер> (дата звернення 04.12.2020)
8. Webservice In Unity [Електронний ресурс] // URL: http://wiki.unity3d.com/index.php?title=Webservice_In_Unity (дата звернення: 09.12.2020)
9. Калниболотський А.О. Використання віртуальної реальності в дистанційному навчанні [Текст] / А.О. Калниболотський, І.В. Янчеглов, О.В. Гайдаєнко //Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2020): Всеукраїнська науково-практична інтернет конференція (26-28 жовтня 2020 р.) – Миколаїв: НУК ім. адмірала Макарова, 2020 – Режим доступу: <http://itconf.nuos.edu.ua/2020/proceedings/>
10. Обозний А. А. Структура концептуальної моделі у людини-оператора // Актуальні проблеми психології праці, інженерної психології та ергономіки. Вип. 1 / Под ред. В. А. Бодрова, А. Л. Журавльова. - М .: Изд-во "Институт психологии РАН", 2009. - С. 403-413. - 615 с.
11. Limits In SQLite [Електронний ресурс] Режим доступу https://sqlite.org/limits.html#max_length%C2%A0 (дата звернення 01.11.2020)

					МР.122.6157м.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

12. «Orange Book» — OpenGL Shading Language by Randi J. Rost, Bill M. Licea-Kane, Dan Ginsburg and John M. Kessenich

13. *Elisha Riedlinger*. Fixes compatibility issues with older games running on Windows 10 by wrapping DirectX dlls. Also allows loading custom libraries with the file extension .asi into game processes:

14. Maria Bielikova, Marko Divecy, Rudolf Kajan, Peter Jurnecka, L'ubos Omelina (2008) «Automatic generation of adaptive, educational and multimedia computer games» - Springer-Verlag London Limited 2008, pp. 371-384

					MP.122.6157м.03.ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

					MP.122.6157м.03.						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розробив	Калнбилотський				Додатки			Лім.	Арк.	Акрушів	
Керівник	Гайдаєнко О.В										
Консульт.	Дудченко О.М.							ХФ НУК			
Н. контр.											
Зав.кафедр	Гучек П.Й.										

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ «TRAINING GYM»

Вступ

Повне найменування програмного забезпечення: Розробка програмного забезпечення «training gym» .

Коротка характеристика області застосування: програмне забезпечення планується використовувати для самостійного заняття спортом в спортзалі або в дому. Також у даному ПЗ передбачено, автоматичного підбора систем тренувань, щоденник тренувань, самостійне створення різних програм тренувань та рекомендованого підбору типу їжі для тренувань. Дане ПЗ має функцію автоматичного оновлення бази даних через інтернет і тому потрібно скачувати таблиці з комплектуючими зі спеціального сайту.

Робоча назва проекту: «»training gym»

1 Підстави для розробки

Підставою для розробки є завдання на виконання магістра роботи за напрямом «Комп'ютерні науки».

2 Призначення розробки

2.1 Функціональне призначення

Програмне забезпечення повинно поліпшити процес заняття спортом.

2.2 Експлуатаційне призначення

Програмне забезпечення призначене для перегляду спортивних вправ та тренувань.

3 Вимоги до програмного забезпечення

3.1 Вимоги до функціональних характеристик

3.1.1 Вимоги до складу виконуваних функцій

					MP.122.6157м.03.Р-01	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Програмне забезпечення повинне виконувати наступні функції:

– Перегляд вправ

Вхідна інформація: перегляд вправ.

Вихідна інформація: перегляд вправ в анімації.

– Програма тренувань:

Вхідна інформація: підбір тренувань.

Вихідна інформація: показ потрібних тренувань .

– Створення програми тренувань

Вхідна інформація: створення програми.

Вихідна інформація: створена база тренувань.

– Щоденник тренувань

Вхідна інформація: Щоденник.

Вихідна інформація: база тренувань.

– Підбір харчування:

Вхідна інформація: обрання харчування.

Вихідна інформація: база даних харчування.

3.1.2 Вимоги до організації вхідних та вихідних даних

Вхідною інформацією для даної системи є база даних, в котрій зберігаються комплектуючі до персонального комп'ютера. Сам користувач може додавати до цієї бази даних потрібну інформацію.

При запуску програми користувач починає обирати комплектуючі, які програма перевіряє на сумісність .

Вихідною інформацією є відповідь на заданий користувачем запит у виді файла в котрому збережена база даних тренувань.

3.1.3 Вимоги до часових характеристик роботи програми

Час реакції програмного продукту на збереження та підбору потрібних комплектуючих 1 с.

					MP.122.6157м.03.Р-01	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Об'єм оперативної пам'яті, необхідний для роботи програми повинен бути не менше 1 Гбайт.

Час запуску програми не повинен перевищувати більше 6 секунд. Дисківий простір, необхідний для збереження програми 50 Мбайт. В архівному вигляді файли початкового коду програми не повинні перевищувати 4 Мбайт.

3.2 Вимоги до надійності

3.2.1 Вимоги до надійного функціонування

При виникненні збою в роботі апаратури, відновлення нормальної роботи програми має виконуватися після перезавантаження операційної системи; повторного виконання дій, втрачених до останнього збереження інформації на магнітному диску. Програма повинна забезпечувати коректну обробку виняткових ситуацій.

Для забезпечення надійної роботи програмного забезпечення також необхідно передбачити можливість резервного копіювання інформації та випробування програмних засобів на наявність комп'ютерних вірусів. Щоб створити резервну копію повинні бути використані вбудовані можливості операційної системи.

3.2.2 Вимоги до контролю вхідної і вихідної інформації

Програма повинна забезпечувати правильне введення інформації за рахунок використання, шаблонів введення, процедурного блокування введення некоректної інформації, списків та авто підстановки. Обробка виняткових ситуацій, пов'язаних з доступом до дисків, пристроїв введення-виведення інформації, повинна оброблятися програмно з виведенням відповідних інформаційних повідомлень, не призводити до блокування роботи програми.

					MP.122.6157м.03.Р-01	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

3.2.3 Вимоги до часу відновлення після відмови

Час відновлення після відмови, не пов'язаної з роботою програмного продукту, повинен складатися з: часу перезапуску операційної системи; часу повторного запуску самої програми.

3.3 Умови експлуатації

Програмний продукт призначений для експлуатації в операційній системі Android. Програмне забезпечення не потребуватиме спеціальних навичок для роботи із ним. Мінімальна кількість персоналу, необхідного для роботи програми: один користувач.

3.4 Вимоги до складу й параметрів технічних засобів

Для роботи програмного забезпечення необхідна наявність у користувача персонального комп'ютера, що має такі системні характеристики:

- МТК 6673 / Snapdragon 400 або пізнішої версії з підтримкою x64.
- обсяг оперативної пам'яті не нижче 1 Гб;
- не менше 200 МБ вільного дискового простору жорсткого диска;
- Телефон с сенсорним екраном;

3.5 Вимоги до інформаційної і програмної сумісності

Для роботи з програмним забезпеченням необхідно мати смартфон на операційній системі Android 5.0

- операційну систему Android 5.0.
- Бібліотеку net framework 3.0

3.6 Вимоги до маркування й упакування

Програмний продукт повинен розміщатися на USB носіях, або на жорсткому диску у вигляді каталогу з файлами.

					MP.122.6157м.03.Р-01	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

3.7 Вимоги до транспортування і збереження

Програмний продукт повинен зберігатися USB носіях, або на жорсткому диску які знаходяться у спеціальних футлярах. В приміщенні у якому зберігають носії температура повинна бути на позначці не нижче -5 градусів за Цельсієм та не вище +45 градусів за Цельсієм при вологості від 15% до 65%.

4 Вимоги до програмної документації

Документація до програмного засобу повинна містити наступні документи:

- технічне завдання;
- опис програми;
- текст програми;
- програма і методика випробувань;
- інструкція користувача.

5 Стадії та етапи розробки

Стадії та етапи розробки програми та терміни їх виконання повинні відповідати затвердженому графіку дипломного проектування та зводитись до таких, що наведені в таблиці 1.

Таблиця 1 – Стадії, етапи розробки програми, та терміни їх виконання

Стадії розробки	Етапи робіт	Зміст робіт	Терміни виконання
1	2	3	4
Технічне завдання	Розробка технічного завдання.	Обґрунтування необхідності розробки програми	12.04.19 – 19.04.19
	Затвердження технічного завдання.	Розробка пояснювальної записки. Узгодження і затвердження технічного завдання.	15.04.19 – 15.04.19

					MP.122.6157м.03.P-01	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Ескізний проект	Розробка ескізного проекту	Попередня розробка структури вхідних та вихідних даних, уточнення методів рішення завдань, загальний опис алгоритму.	16.04.19 – 20.04.19
	Затвердження ескізного проекту	Розробка пояснювальної записки. Узгодження і затвердження ескізного проекту.	18.04.19 – 21.04.19
Технічний проект	Розробка технічного проекту	Уточнення структури вхідних та вихідних даних, розробка алгоритму рішення задачі, розробка структури програми.	20.04.19 – 22.04.19
	Затвердження технічного проекту	Розробка пояснювальної записки. Узгодження і затвердження технічного проекту.	22.04.19 – 23.04.19

					MP.122.6157м.03.Р-01	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

Продовження таблиці 1.

1	2	3	4
Робочий проект	Розробка програми	Програмування та налагодження програми	23.04.19 — 11.05.19
	Розробка програмної документації	Розробка програмної документації відповідно до вимог	11.05.19 — 20.05.19
	Випробування програми	Розробити, узгодити та затвердити програму і методику випробувань, коригування програми і програмної документації за результатами випробувань.	21.05.19 — 27.05.19

7. Порядок прийому та контролю

Для контролю й прийому необхідно надати інструкцію користувача та опис програми, а також програму та методику випробування.

Порядок контролю і прийому даної розробки здійснюється представником замовника у присутності представників розробника згідно з програмою та методикою випробувань.

За результатами прийому складається акт, який підписується представником замовника та представником розробника та утверджується керівниками організації-замовника і організації-розробника.

У випадку виявлення помилок під час прийому програмного виробу складається акт про виявлені помилки, який підписується представниками замовника та розробника й стверджується керівниками організації - замовника й організації-розробника. Розробник повинен в продовж не більш як 1-го місяця видалити вказані зауваження, й повідомити замовника про повторне проведення перевірки, не пізніше як за 2 тижні.

					MP.122.6157м.03.Р-01	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК Г

ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ «КОНФІГУРАТОР ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА»

1 Загальні відомості

Назва: Проект програмного забезпечення “Розробка програмного забезпечення для заняття спортом”.

Область застосування – як предметна галузь розглядається заняття спортом.

Для проектування програмного забезпечення було обрано SQLiteStudio, Програма написана з використанням мови програмування C# і реалізована в програмному продукті Unity 3d.

Для роботи програми необхідна операційна система Android 5/0i вище. Файли, що зберігаються й завантажуються програмою, є таблицями mySQL і можуть бути відкриті, змінені в інших, сумісних з ним редакторах.

2 Функціональне призначення розробки

Програмне забезпечення планується використовувати для того, для заняття спортом. ПЗ дозволяє швидко та зручно обрати потрібні вам вправи для заняття спортом та перегляду різних типів вправ. Також в даному ПЗ передбачене скачування бази даних через інтернет, а також дозволяє вам обрати потрібну програму тренувань та систему харчування. Дане ПЗ має функцію автоматичного оновлення бази даних через інтернет.

Вимоги до складу виконуваних функцій:

- Вибір комплектуючих
- Перевірка сумісності комплектуючих
- Обчислення вартості
- Обчислення продуктивності
- Збереження конфігурації персонального комп'ютера

3 Опис логічної структури

					MP.122.6157м.03.P-01	Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

Програмна логіка

Після запуску програмного забезпечення, з'являється головне вікно ПЗ «trainig gym». В котрому користувач може обирати потрібні поля програм, користувач може яе переглядати вправи, автоматично створювати програми тренувань, та самостійно створювати системи тренувань.

4 Технічні засоби

Для роботи програмного забезпечення необхідна наявність у користувача персонального комп'ютера, що має такі системні характеристики:

- МТК 6673 / Snapdragon 400 або пізнішої версії з підтримкою x64.
- обсяг оперативної пам'яті не нижче 1 Гб;
- не менше 200 МБ вільного дискового простору жорсткого диска;
- Телефон с сенсорним екраном;

5 Виклик і завантаження

Для запуску програми необхідно запустити файл TG.exe.

6 Вхідні дані

Вхідною інформацією для даної системи є таблиці в котрих зберігаються дані про вправи та програми спорту.

7 Вихідні дані

Вихідною інформацією є готова програма тренувань завдяки котрим користувач може займатися спортивними вправами

					MP.122.6157м.03.Р-01	Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

ДОДАТОК Б

ПРОГРАМА І МЕТОДИКА ВИПРОБУВАНЬ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ “TRAINING GYM”

1. Об’єкт випробувань

Об’єктом випробувань є програмне забезпечення для “training gym”.

Смартфони застосовуються в різних областях людської діяльності, і в наш час використовується вже досить велика кількість смартфонів. Заняття спортом – це дуже важливий і складний процес. При цьому слід враховувати те, для яких цілей буде проводиться тренування. На даний момент на ринку представлена величезна кількість комплектуючих. Вибір стає ще складніше тому, що не всі комплектуючі сумісні один з одним. Конфігуратор ПК дозволяє полегшити процес підбору. За допомогою даної програми навіть недосвідчений користувач може впоратися з комплектуванням потрібного ПК.

2. Мета випробувань

Метою випробування є перевірка програмного забезпечення на наявність помилок і відповідність реалізованих функцій функціям, зазначеним у технічному завданні, яке наведене у додатку А. Якщо прийнятий рівень відповідності програмного забезпечення вимогам досягнутий, а також відсутні помилки та грубі недоліки, то програмне забезпечення приймається в експлуатацію.

					MP.122.6157м.03.Р-01	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

3. Вимоги до програми

При проведенні випробувань функціональні характеристики програми підлягають перевірці на відповідність вимогам, викладеним у пункті «Вимоги до функціональних характеристик» технічного завдання.

Для проведення випробування будемо розглядати три функції програмного забезпечення для конфігурації персонального комп'ютера:

- Можливість перегляду вправ
- Створення програм тренувань
- Вести щоденник тренувань
- Автоматичний підбір програм тренувань.
- Підбір харчування

4. Вимоги до програмної документації

Документація, що пропонується до випробування програмного забезпечення «Training gym», повинна включати наступні документи:

- технічне завдання;
- опис програмного забезпечення;
- текст програмного забезпечення;
- програму та методику випробувань;
- інструкцію користувача.

5. Засоби та порядок випробувань

Засоби випробувань

Для випробувань даної програми необхідно, щоб на смартфоні де буде встановлене програмне забезпечення, стояла операційна система android 5.0 та вище.

Для роботи програмного забезпечення необхідна наявність у користувача смартфона, що має характеристики, не нижче, ніж:

					MP.122.6157м.03.Р-01	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

– ARM-сумісний смартфон з тактовою частотою процесора 2 МГц і вище;

– обсяг оперативної пам'яті не нижче 1 ГБ;

– не менше 200 МБ вільного дискового простору жорсткого диска;

– сенсорний монітор;

Порядок проведення випробувань

Випробування програмного забезпечення будуть проводитися протягом 3 (трьох) днів. Замовник самостійно проводить тестування усіх функцій програми у присутності розробника. При наявності відповідної можливості у процесі проведення випробувань можуть додатково брати участь програмісти. Попередньо до початку випробувань визначається кінцевий склад групи, що буде проводити випробування. Кожен член групи одержує необхідну документацію до програми. При випробуванні проводиться:

- контроль складу виконуємих функцій;
- контроль коректності виконання окремих функцій;
- випробування на правильність роботи окремих модулів;
- комплексне випробування.

При виявленні помилок у роботі програмного забезпечення складається перелік цих помилок і обговорюється термін їхнього виправлення розробником.

Після цього замовник проводить повторне випробування у повному обсязі.

					MP.122.6157м.03.Р-01	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

6. Методи випробувань

Випробування будемо проводити за стратегією «чорного ящика».

Складемо програму випробування:

Випробування функції «Створення програм тренувань»

На головному вікні додатку користувачу потрібно обрати «програму тренувань». Після цього користувач обирає для чого саме він буде тренуватися та для якої цілі для похудання або для особих тренувань, після цього користувач додає свої фізичне дані тоді програма автоматично підбирає програму

Випробування функції «Можливість перегляду вправ»

На головному вікні додатку користувачу потрібно обрати поле «Види тренувань». Після цього користувач переходить в вікно з видами вправ для різних видів м'язів, після цього користувач переходить в вікно з видами тренувань, а потім користувач може переглянути вправи в 3д графіці та прочитати більше про вправи.

Випробування функції «Самостійне створення програми»

На головному вікні додатку користувачу потрібно обрати поле «програму тренувань». Після цього користувач обирає поле «створення програми», а потім відкривається вікно з обрання типів вправ котрі можна додати в програму тренувань, після цього зберегти.

Випробування функції «Щоденник тренувань»

На головному вікні додатку користувачу потрібно обрати поле «програму тренувань». Після цього користувач обирає поле «щоденник тренувань», а потім відкривається вікно з введення даних котрі зберігаються.

					MP.122.6157м.03.P-01	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК В ІНСТРУКЦІЯ КОРИСТУВАЧА

Для установки програми необхідно запустити Tg.exe і слідувати інструкціям інсталятора. Також необхідно підключення до мережі інтернет для оновлення даних в базі.

При занятті спортом потрібно обережно обирати вправи для заняття спортом якщо ви вже досвідчений спортсмен то може самостійно переглядати вправи, та займатися ними. Якщо ви раніше не займалися спортом то краще обрати вже підготовлену систему тренувань та займатися за рекомендаціями котрі вже є в програмі. Також ви можете не тільки переглядати вже готові вправи, та раніше створенні програми тренувань а і створювати нові. Ви можете самостійно створювати свої системі тренувань, та детально слідкувати за своїми тренуваннями. Також користувач може підібрати програму харчування та вести щоденник тренувань, додаток автоматично зможе підібрати систему харчування, також ви можете вести щоденник тренувань в котрому ви можете записувати свої тренування та коли ви тренувались.

					MP.122.6157м.03.Р-01	Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

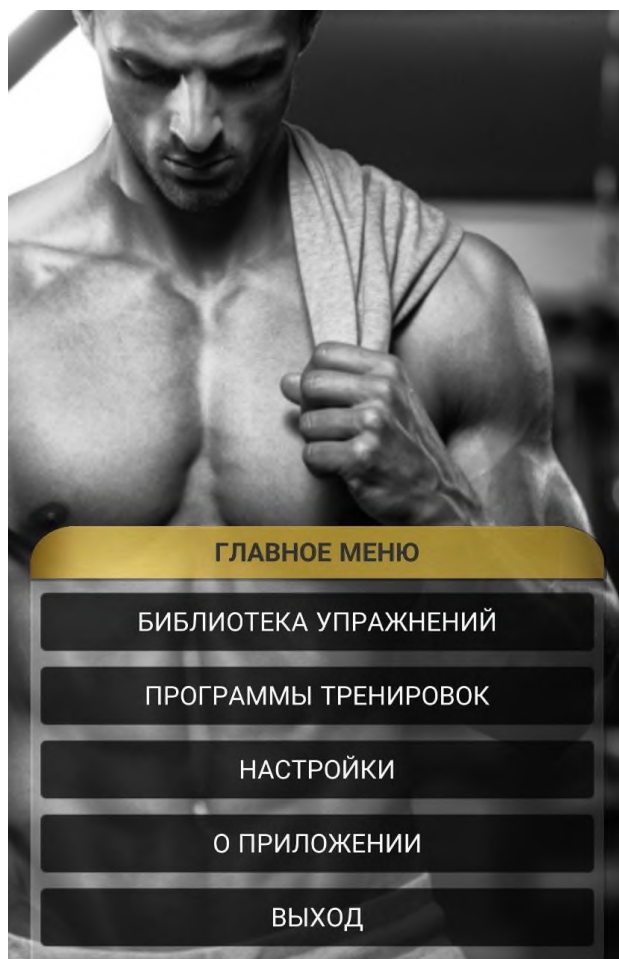


Рисунок 1 – Головне меню

На головну меню ви можете обрати потрібні вам функції такі як «бібліотека вправ» або «програми тренувань» чи «настройки» в котрій ви можете працювати с потрібними вам функціями, після обрання бібліотеки вправ на рисунке 2 можна переглянути типи м'язів для вправ .

					MP.122.6157м.03.Р-01	Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		



Рисунок 2 – Типи м'язів

На рисунке 2 показано які типи м'язів ви можете обрати для перегляду тренувань, після обрання потрібного виду м'язів ви переходите до частини обрання самої вправи на рисунке 3.



Рисунок 3 – Вправи

Після обрання потрібного виду м'язів, ви обираєте потрібну вправу, та можете переглянути саму вправу в 3д анімації котра показана на рисунку 4 .

					MP.122.6157м.03.P-01	Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		



Рисунок 4 – 3Д вправа

Після того як ми обрали потрібну вправу користувач пеглядає вправу, та більш детально переглянути саму вправу, а також прочитати що робити підчас її вправи так як краще її вивувати.

					MP.122.6157м.03.P-01	Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

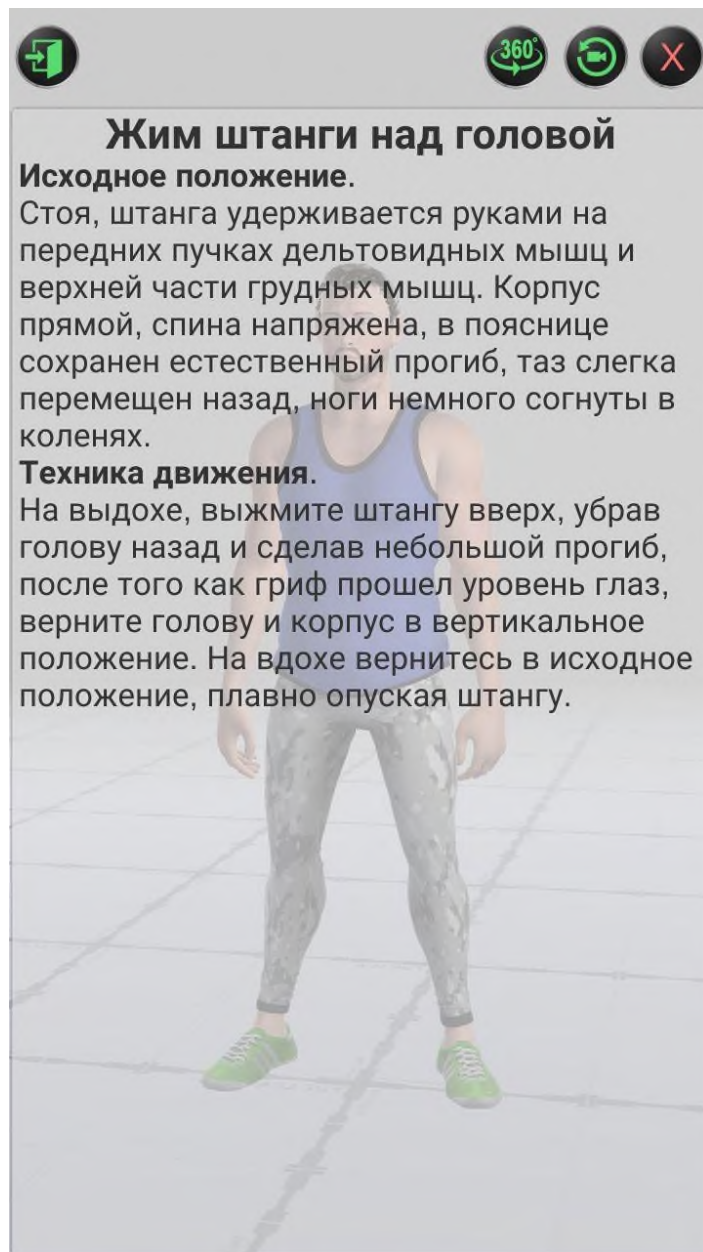


Рисунок 5 – Інформація про вправу

На рисунке 5 користувач може більш детально прочитати про вправу так як краще займатися вправою.

					MP.122.6157м.03.P-01	Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

ДАННЫЕ ДЛЯ ПОДБОРА ПРОГРАММЫ

Тип программы: i
 Для тренажёрного зала ▾

Сложность:
 Новичок ▾

Количество тренировок в неделю:
 1 в неделю ▾

Цель тренировок:
 Набор массы ▾

Закрыть Подобрать

ЗАГРУЗИТЬ ПРОГРАММУ

ДНЕВНИК ТРЕНИРОВОК

РЕКОМЕНДАЦИИ ПО ПИТАНИЮ

НАЗАД

Рисунок 6 – Підбір програми

На рисунке 6 показано автоматичний підбір програми тренувань в котрих сам користувач може отримати рекомендації від додатка, та краще займатися спортом.

					MP.122.6157м.03.P-01	Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

ПРОГРАММА ТРЕНИРОВОК

1

Разгибание голени в тренажёре

Редакт.

Подходы: 3

Повторы: 10

Нагрузка: 100%

Отдых: 01:30

2

Сгибание голени в тренажёре

Редакт.

Подходы: 3

Повторы: 10

Нагрузка: 100%

Отдых: 01:30

3

Отжимания от пола

Редакт.

Подходы: 3

Повторы: 12

Нагрузка: 100%

Отдых: 01:30

4

Горизонтальная тяга

Редакт.

Подходы: 3

Повторы: 10

Нагрузка: 100%

Отдых: 01:30

5

Отведение плеча с гантелями

Редакт.

Подходы: 3

Повторы: 10

Нагрузка: 100%

Отдых: 01:30

6

Скручивания на скамье

Редакт.

Выход

Сохранить

Экспорт

Питание

Редакт.

Рисунок 7 – Програма тренувань

На рисунке 7 показана вже автоматично створена програма тренувань завдяки котрої користувач може починати займатися спортом..



Рисунок 8 – ручне створення програми

На рисунку 8 показано вікно самостійного створення програми тренувань в котрому користувач самостійно підбирає вправи для тренувань, а також обирати харчування рекомендує для вже досвідчених спортсменів.

					MP.122.6157М.03.P-01	Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

Рисунок 9 – Програма харчування

На рисунку 9 показано як саме потрібно обрати програму харчування для недосвідчених користувачів.

					MP.122.6157м.03.P-01	Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

РЕКОМЕНДАЦИИ ПО ПИТАНИЮ

Расчёты

Ваша суточная норма калорий: 1907 ккал.

Из них:

- Белки: 572 ккал (143 гр);
- Жиры: 572 ккал (63 гр);
- Углеводы: 763 ккал (190 гр).

Расчёты проведены по следующим данным:

- Пол: Мужчина;
- Возраст: 22 лет;
- Вес: 95 кг;
- Рост: 189 см;
- Образ жизни: Сидячий;
- Цель занятий: Похудение.



Редактировать данные

Рекомендации для похудения

1. Калорийность вашего рациона не должна отличаться более чем на 35 ккал от высчитанных для вас значений.
2. Рацион насчитывает 5-6 приемов пищи в день.
3. Пропорции рациона по соотношению белков, жиров и углеводов приближены к 30%-30%-40%.
4. Завтрак по калорийности выше, чем обед, а обед выше чем ужин.
5. Первый перекус выше по калорийности, чем второй, а второй выше



Выход



Справка

Рисунок –10 Харчування

На рисунке 10 вже показано створена програма харчування для недосвідчених користувачів

					MP.122.6157м.03.P-01	Арк.
Змін.	Арк.	№ докум.	Підпись	Дата		

					ДР.6.050103.4157.03.ПЗ			
Змн.	Арк.	№ докум.	Підп.	Дата	ПРОЕКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ «КОНФІГУРАТОР ПК»	Літ.	Арк.	Арк.
Розробив		Калниболотський А.О				н		
Керівник		Латанська Л.А.				ХФ НУК ім. адм. Макарова		
Консультант								
Зав. Кафедр.		Гучик П.Й.						

ДОДАТОК Д

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections.Generic;
public class PlayerBot : MonoBehaviour {
public delegate void AnimalAddedEvent();
public delegate void TraitAddedEvent();
public delegate void PassTurnEvent();
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```
using System.IO;
using System.Data;
using System.Data.Common;
using System.Data.OleDb;
```

```
namespace WinFormDataBaseTestAppl
```

```
{
```

```
    public class DataBaseOperations
```

```
    {
```

```
        private string dbFilePath;
```

```
        private OleDbConnection myConnection;
```

```
        /
```

```
        public DataBaseOperations()
```

```
        {
```

```
            dbFilePath = string.Format("{0}{1}", AppDomain.CurrentDomain.BaseDirectory,
            "DataBase.mdb");
```

```
            if (!File.Exists(dbFilePath))
```

```
                throw new FileNotFoundException("Файл базы не найден!");
```

```
            else
```

```
            {
```

```
                string cnStr = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=DataBase.mdb";
```

```
                myConnection = new OleDbConnection(cnStr);
```

Арк.

ДР 6-050103.4157.03.ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата	

```

        myConnection.Open();
    }
}

public void CloseConnection()
{
    myConnection.Close();
}

public List<byte> GetConfigData(string key)
{
    List<byte> temp = new List<byte>();
    OleDbCommand selectCmd = new OleDbCommand("SELECT data FROM Arcadia
WHERE product LIKE " +
        "" + key + "");
    selectCmd.Connection = myConnection;
    using (OleDbDataReader dr = selectCmd.ExecuteReader())
    {
        while (dr.Read())
        {
            temp.Clear();
            byte[] data = (byte[])dr["data"];
            temp.AddRange(data);
        }
    }
    if (temp.Count == 0)
        throw new ArgumentException("Данные не найдены!");
    return temp;
}

public DataTable GetDataTable()
{
    DataTable dt = new DataTable("Arcadia");
    dt.Columns.AddRange(new DataColumn[] { new DataColumn("model", typeof(string)),
        new DataColumn("product", typeof(string)), new DataColumn("serial", typeof(string)) });
    OleDbCommand selectCmd = new OleDbCommand("SELECT model,product,serial
FROM Arcadia");

```

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	ДР.6.050103.4157.03.ПЗ	

```

selectCmd.Connection = myConnection;

using (OleDbDataReader dr = selectCmd.ExecuteReader())
{
    while (dr.Read())
    {
        DataRow row = dt.NewRow();
        row["model"] = dr["model"];
        row["product"] = dr["product"];
        row["serial"] = dr["serial"];
        dt.Rows.Add(row);
    }
}
return dt;
}

public void InsertConfigData(byte[] data)
{
    GetInfo info = new GetInfo();
    if (!info.ParseBuffer(data.ToList()))
        throw new ArgumentException("Файл не соответствует параметрам!");

    OleDbCommand insertCmd = new OleDbCommand();
    OleDbParameter param = new OleDbParameter("@data",SqlDbType.Binary);
    param.Value=data;
    insertCmd.CommandText = "INSERT INTO Arcadia (model,product,serial,data)
VALUES('" +
        info.Model + "','" + info.ProductNumber + "','" + info.SerialNumber +
        "','"@data)";
    insertCmd.Parameters.Add(param);
    insertCmd.Connection = myConnection;

    try
    {
        insertCmd.ExecuteNonQuery();
    }
}

```

						ДР.6.050103.4157.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата			

```

        catch (OleDbException)
        {
            ArgumentException ex = new ArgumentException
            ("Файл с такими данными уже содержится в базе!");
            throw ex;
        }
    }
}

}using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

using System.IO;

namespace WinFormDataBaseTestAppl
{
    public partial class Form1 : Form
    {
        DataBaseOperations ops;

        public Form1()
        {
            InitializeComponent();

            this.Load += delegate { LoadApplication(); };
            this.FormClosing += delegate { ExitApplication(); };
            btnSelectData.Click += delegate { GetDataFromDataBase(); };
            btnInsertData.Click += delegate { InsertDataToDataBase(); };
        }
    }
}

```

//Обработчик загрузки приложения

private void LoadApplication()

ДР.6.050103.4157.03.ПЗ

					Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	

```

{
    try
    {
        ops = new DataBaseOperations();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + "\n" + "Программа будет завершена!", "Ошибка
базы данных.");
        this.Close();
        return;
    }
    DataTable dt = ops.GetDataTable();
    arcadiaGridView.DataSource = dt;
}

```

```
private void GetDataFromDataBase()
```

```

{
    string model = arcadiaGridView.SelectedRows[0].Cells[0].Value.ToString();
    string key = arcadiaGridView.SelectedRows[0].Cells[1].Value.ToString();
    List<byte> data;
    if (ops != null)
    {
        try
        {
            data = ops.GetConfigData(key);

            SaveFileDialog dialog = new SaveFileDialog()
            { Filter = "eep files (*.eep)|*.eep|bin files (*.bin)|*.bin",
              FileName = string.Format("{0}_{1}", model, key )};
            if (dialog.ShowDialog() == DialogResult.OK)
            {
                File.WriteAllBytes(dialog.FileName, data.ToArray());
            }
        }
    }
}

```

```
catch (Exception ex)
```

```
{
```

					ДР.6.050103.4157.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        MessageBox.Show(ex.Message, "Ошибка!");
    }
}

//
private void InsertDataToDataBase()
{
    OpenFileDialog dialog = new OpenFileDialog()
    { Filter = "eep files (*.eep)|*.eep|bin files (*.bin)|*.bin", FileName = string.Empty };
    if (dialog.ShowDialog() == DialogResult.OK)
    {
        try
        {
            byte[] data = File.ReadAllBytes(dialog.FileName);
            ops.InsertConfigData(data);

            arcadiaGridView.DataSource = ops.GetDataTable();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Ошибка");
        }
    }
}

private void ExitApplication()
{
    if (ops != null)
        ops.CloseConnection();
}

}
namespace WinFormDataBaseTestAppl
{
    partial class Form1
    {

```

					ДР.6.050103.4157.03.ПЗ	Арк.
		/// <summary>				
Змн.	Арк.	№ докум.	Підпис	Дата		

```

///
/// </summary>
private System.ComponentModel.IContainer components = null;

/// <summary>
/// Освободить все используемые ресурсы.
/// </summary>
/// <param name="disposing">истинно, если управляемый ресурс должен быть удален;
иначе ложно.</param>
protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

#region Код, автоматически созданный конструктором форм Windows

/// <summary>

///
/// </summary>
private void InitializeComponent()
{
    System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form1));
    this.btnSelectData = new System.Windows.Forms.Button();
    this.arcadiaGridView = new System.Windows.Forms.DataGridView();
    this.btnInsertData = new System.Windows.Forms.Button();
    ((System.ComponentModel.ISupportInitialize)(this.arcadiaGridView)).BeginInit();
    this.SuspendLayout();
    //
    // btnSelectData
    //
    this.btnSelectData.Location = new System.Drawing.Point(415, 71);
}

```

					ДР.6.050103.4157.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

this.btnSelectData.Name = "btnSelectData";
this.btnSelectData.Size = new System.Drawing.Size(133, 23);
this.btnSelectData.TabIndex = 0;
this.btnSelectData.Text = "Извлечь файл";
this.btnSelectData.UseVisualStyleBackColor = true;
//
// arcadiaGridView
//
this.arcadiaGridView.AllowUserToAddRows = false;
this.arcadiaGridView.AllowUserToDeleteRows = false;
this.arcadiaGridView.AllowUserToResizeColumns = false;
this.arcadiaGridView.AllowUserToResizeRows = false;
this.arcadiaGridView.BackgroundColor = System.Drawing.SystemColors.Window;
this.arcadiaGridView.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.arcadiaGridView.Location = new System.Drawing.Point(12, 12);
this.arcadiaGridView.Name = "arcadiaGridView";
this.arcadiaGridView.ReadOnly = true;
this.arcadiaGridView.RowHeadersVisible = false;
this.arcadiaGridView.RowHeadersWidthSizeMode =
System.Windows.Forms.DataGridViewRowHeadersWidthSizeMode.DisableResizing;
this.arcadiaGridView.SelectionMode =
System.Windows.Forms.DataGridViewSelectionMode.FullRowSelect;
this.arcadiaGridView.Size = new System.Drawing.Size(321, 213);
this.arcadiaGridView.TabIndex = 1;
//
// btnInsertData
//
this.btnInsertData.Location = new System.Drawing.Point(200, 231);
this.btnInsertData.Name = "btnInsertData";
this.btnInsertData.Size = new System.Drawing.Size(133, 23);
this.btnInsertData.TabIndex = 2;
this.btnInsertData.Text = "Добавить файл";
this.btnInsertData.UseVisualStyleBackColor = true;
//

```

// Form1

//

					ДР.6.050103.4157.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.ClientSize = new System.Drawing.Size(347, 266);
        this.Controls.Add(this.btnInsertData);
        this.Controls.Add(this.arcadiaGridView);
        this.Controls.Add(this.btnSelectData);
        this.Icon = ((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
        this.MaximizeBox = false;
        this.MinimizeBox = false;
        this.Name = "Form1";
        this.Text = "Arcadia DataBase v1.0.0.0";
        ((System.ComponentModel.ISupportInitialize)(this.arcadiaGridView)).EndInit();
        this.ResumeLayout(false);

    }

```

#endregion

```

        private System.Windows.Forms.Button btnSelectData;
        private System.Windows.Forms.DataGridView arcadiaGridView;
        private System.Windows.Forms.Button btnInsertData;
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

namespace WinFormDataBaseTestAppl
{
    public class GetInfo
    {
        //
        public bool ParseBuffer(List<byte> list)
        {

```

```

            if (list.Count == 0 || list.Count > 0x4000)

```

```

                return false;

```

```

            }
        }
    }

```

					ДР.6.050103.4157.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

//
int startAddress = ((list[0] * 0x100) + list[1]) * 2 + 2;

//
if (startAddress >= list.Count - 25)
    return false;

//Получение маркера
string marker = BitConverter.ToString(list.ToArray(), startAddress, 6).Replace("-", "");

//Проверка на соответствие маркера
if (!Enum.IsDefined(typeof(Markers), long.Parse(marker)))
    return false;

//Получение серийного номера
SerialNumber = BitConverter.ToString(list.GetRange(startAddress + 6, 6).ToArray())
    .Replace("-", "")
    .Substring(0, 9);

//
ProductNumber = BitConverter.ToString(list.GetRange(startAddress + 12, 6).ToArray())
    .Replace("-", "")
    .Substring(0, 11);

//Получение модели
Model = Encoding.ASCII.GetString(list.GetRange(startAddress + 18, 14)
    .ToArray()).Replace("/", "").Trim().Replace("\0", "");

return true;
}
//Свойство для серийного номера
public string SerialNumber
{
    get;
    private set;
}

```

		private set;			ДР.6.050103.4157.03.ПЗ	Арк.
		}				
Змн.	Арк.	№ докум.	Підпис	Дата		

```

//
public string ProductNumber
{
    get;
    private set;
}
//Свойство для модели
public string Model
{
    get;
    private set;
}

enum Markers : long
{
    WM_Arcadia_1 = 030609101211,
    WM_Arcadia_2 = 030609101212,
    WM_Arcadia_3 = 030609101213,
    WM_Arcadia_4 = 030609101214
}

enum DataLength : byte
{
    Serial_Number = 9,
    Product_Number = 11,
    Model = 14
}
}
namespace SII_lr3
{
    public partial class Form1 : Form
    {
        LogicalConclusion LC;

        public Form1()
        {

```

```
InitializeComponent();
```

					ДР.6.050103.4157.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        LC = new LogicalConclusion(country, background, consequence, X, Y, Ax, A, By, B, R);

        setData();
    }

    string background = "Цена на нефть"; //предпосылка
    string consequence = "

    string[][] R = new string[][] { new string[] { "
        new string[] { " },

    private void setData()
    {
        for (int i = 0; i < Ax.Length; i++)
            oilPriceComboBox.Items.Add(Ax[i]);
        oilPriceComboBox.SelectedIndex = 5;

        MamdaniRadioButton.Checked = true;
        mechanism1RadioButton.Checked = true;
    }

    private void showRulesButton_Click(object sender, EventArgs e)
    {
        rulesListBox.Visible = true;
        hideRulesButton.Visible = true;

        this.Height = 676;

        rulesListBox.Items.Clear();

        foreach(string[] r in R)
        {
            string s = "
            rulesListBox.Items.Add(s);
        }
    }

```

		string s = "							Арк.
		rulesListBox.Items.Add(s);							
Змн.	Арк.	№ докум.	Підпис	Дата					

ДР.6.050103.4157.03.ПЗ

```

    }
}

private void hideRulesButton_Click(object sender, EventArgs e)
{
    rulesListBox.Visible = false;
    hideRulesButton.Visible = false;

    this.Height = 337;
}

private void calculateButton_Click(object sender, EventArgs e)
{
    string backgr = oilPriceComboBox.SelectedItem.ToString();
    string implicationType;

    if (MamdaniRadioButton.Checked == true)
        implicationType = "Mamdani";
    else
        implicationType = "Larsen";

    if (mechanism1RadioButton.Checked == true)
        ResultTextBox.Text = LC.Algorithm1(backgr, implicationType);
    else
        if (mechanism2RadioButton.Checked == true)
            ResultTextBox.Text = LC.Algorithm2(backgr, implicationType);
        else
            if (mechanism3RadioButton.Checked == true)
                ResultTextBox.Text = LC.Algorithm3(backgr, implicationType);
    }
}

namespace SII_lr3
{
    class LogicalConclusion
    {
        string countryName; //назв

```

					ДР.6.050103.4157.03.ПЗ	Арк.
		string[] X; //				
Змн.	Арк.	№ докум.	Підпис	Дата		

```
string[] Y; //
```

```
string[] Ax; //
```

```
double[][] A;
```

```
string[] By; /
```

```
double[][] B;
```

```
string[][] R; //
```

```
public LogicalConclusion(string countryName, string background, string consequence,  
string[] X, s
```

```
    this.countryName = countryName;
```

```
    this.background = background;
```

```
    this.consequence = consequence;
```

```
    this.X = X;
```

```
    this.Y = Y;
```

```
    this.Ax = Ax;
```

```
    this.A = A;
```

```
    this.By = By;
```

```
    this.B = B;
```

```
    this.R = R;
```

```
}
```

```
//
```

```
{
```

```
    double[][][] R_i = new double[R.GetLength(0)][][];
```

```
    for (int k = 0; k < R.GetLength(0); k++)
```

```
    {
```

```
        R_i[k] = new double[X.Length][];
```

```
        for (int i = 0; i < X.Length; i++)
```

```
            R_i[k][i] = new double[Y.Length];
```

```
    }
```

					ДП.6.050103.4157.03.ПЗ	Арк.
		for (int k = 0; k < R.GetLength(0); k++)				
Змн.	Арк.	№ докум.	Підпис	Дата		

```

{
    int a_index = Array.IndexOf(Ax, R[k][0]);
    int b_index = Array.IndexOf(By, R[k][1]);

    if (implicationType == "Mamdani")
        Mamdani(A[a_index], B[b_index], R_i[k]);
    else
        if (implicationType == "Larsen")
            Larsen(A[a_index], B[b_index], R_i[k]);
}

double[][] B_i = new double[R.GetLength(0)][];
for (int i = 0; i < R.GetLength(0); i++)
    B_i[i] = new double[Y.Length];

int a__index = Array.IndexOf(Ax, backgr); //a__index (

for (int i = 0; i < R.GetLength(0); i++)
    Max_Min(A[a__index], R_i[i], B_i[i]);

double[] B_ = new double[Y.Length];

for (int j = 0; j < Y.Length; j++)
{
    double max = B_i[0][j];
    for (int i = 1; i < R.GetLength(0); i++)
        if (B_i[i][j] > max)
            max = B_i[i][j];
    B_[j] = max;
}

string result = "";

for (int j = 0; j < Y.Length; j++)
    result += "(" + Y[j] + ")" + "/" + B_[j] + "+";

```

					ДР.6.050103.4157.03.ПЗ	Арк.
		if (result.Length > 0)				
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        result = result.Remove(result.Length - 1);

    return result;
}

public string Algorithm2(string backgr, string implicationType)
{
    double[][][] R_i = new double[R.GetLength(0)][][];
    for (int k = 0; k < R.GetLength(0); k++)
    {
        R_i[k] = new double[X.Length][];
        for (int i = 0; i < X.Length; i++)
            R_i[k][i] = new double[Y.Length];
    }

    for (int k = 0; k < R.GetLength(0); k++)
    {
        int a_index = Array.IndexOf(Ax, R[k][0]);
        int b_index = Array.IndexOf(By, R[k][1]);

        if (implicationType == "Mamdani")
            Mamdani(A[a_index], B[b_index], R_i[k]);
        else
            if (implicationType == "Larsen")
                Larsen(A[a_index], B[b_index], R_i[k]);
    }

    double[][] R_agg = new double[X.Length][];
    for (int i = 0; i < X.Length; i++)
        R_agg[i] = new double[Y.Length];

    for (int i = 0; i < X.Length; i++)
        for (int j = 0; j < Y.Length; j++)
        {
            double max = R_i[0][i][j];
            for (int k = 1; k < R.GetLength(0); k++)
                if (R_i[k][i][j] > max)
                    max = R_i[k][i][j];
            R_agg[i][j] = max;
        }
}

```

									Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	ДП-6.050103.4157.03.ПЗ				

```

        if (R_i[k][i][j] > max) max = R_i[k][i][j];
        R_agg[i][j] = max;
    }

    double[] B_ = new double[Y.Length];

    int a__index = Array.IndexOf(Ax, backgr);

    Max_Min(A[a__index], R_agg, B_);

    string result = "";

    for (int j = 0; j < Y.Length; j++)
        result += "(" + Y[j] + ")" + "/" + B_[j] + "+";

    if (result.Length > 0)
        result = result.Remove(result.Length - 1);

    return result;
}

public string Algorithm3(string backgr, string implicationType)
{
    double[] alpha = new double[R.GetLength(0)];

    int a__index = Array.IndexOf(Ax, backgr);

    List<double> tmpMin = new List<double>();

    for (int i = 0; i < R.GetLength(0); i++)
    {
        tmpMin.Clear();

        int a_index = Array.IndexOf(Ax, R[i][0]);

```

					ДР.6.050103.4157.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        tmpMin.Add(Min(A[a__index][j], A[a_index][j]));

        alpha[i] = tmpMin.Max();
    }

    double[][][] B_i = new double[R.GetLength(0)][][]; //B_i[кол-во Правил][1][кол-во Y]
    for (int k = 0; k < R.GetLength(0); k++)
    {
        B_i[k] = new double[1][];
        B_i[k][0] = new double[Y.Length];
    }

    for (int i = 0; i < R.GetLength(0); i++)
    {
        int b_index = Array.IndexOf(By, R[i][1]);

        if (implicationType == "Mamdani")
            Mamdani(new double[] { alpha[i] }, B[b_index], B_i[i]);
        else
            if (implicationType == "Larsen")
                Larsen(new double[] { alpha[i] }, B[b_index], B_i[i]);
    }

    //Agg
    double[] B_ = new double[Y.Length];

    for (int j = 0; j < Y.Length; j++)
    {
        double max = B_i[0][0][j];
        for (int i = 1; i < R.GetLength(0); i++)
            if (B_i[i][0][j] > max)
                max = B_i[i][0][j];
        B_[j] = max;
    }

```

```
string result = "";
```

					ДР.6.050103.4157.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        for (int j = 0; j < Y.Length; j++)
            result += "(" + Y[j] + ")" + "/" + B_[j] + "+";

        if (result.Length > 0)
            result = result.Remove(result.Length - 1);

        return result;
    }

    private void Mamdani(double[] A, double[] B, double[][] R)
    {
        for (int i = 0; i < A.Length; i++)
            for (int j = 0; j < B.Length; j++)
                R[i][j] = Min(A[i], B[j]);
    }

    private void Larsen(double[] A, double[] B, double[][] R)
    {
        for (int i = 0; i < A.Length; i++)
            for (int j = 0; j < B.Length; j++)
                R[i][j] = A[i] * B[j];
    }

    private double Min(double a, double b)
    {
        if (a <= b)
            return a;
        else
            return b;
    }

    private void Max_Min(double[] A, double[][] R, double[] B)
    {
        List<double> tmpMAX = new List<double>();

        for (int j = 0; j < R[0].GetLength(0); j++)
            {
                double max = 0;
                for (int i = 0; i < R.GetLength(0); i++)
                    if (R[i][j] > max)
                        max = R[i][j];
                tmpMAX.Add(max * B[j]);
            }
    }

```

		for (int j = 0; j < R[0].GetLength(0); j++)			ДР.6.050103.4157.03.ПЗ	Арк.
		{				
Змн.	Арк.	№ докум.	Підпис	Дата		

```
tmpMAX.Clear();
```

```
for (int k = 0; k < R.GetLength(0); k++)
```

```
    tmpMAX.Add(Min(A[k], R[k][j]));
```

```
B[j] = tmpMAX.Max();
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

					ДР.6.050103.4157.03.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		