

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА ІМЕНІ О. М. БЕКЕТОВА

Навчально-науковий інститут енергетичної,
інформаційної та транспортної інфраструктури

Кафедра комп'ютерних наук та інформаційних технологій

**Пояснювальна записка
до кваліфікаційної роботи бакалавра**

на тему:

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІГРОВОГО ЗАСТОСУНКУ

Виконав:

студент групи КН 2022-1

Семенко Станіслав Віталійович

Керівник:

Новожилова Марина Володимирівна

Рецензент:

Пахомов Юрій Васильович

Харківський національний університет міського господарства імені О. М. Бекетова
(повне найменування закладу вищої освіти)

Навчально-науковий Інститут енергетичної, інформаційної

та транспортної інфраструктури

Кафедра Комп'ютерних наук та інформаційних технологій

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КНтаІТ



Новожилова М. В.

« 23 » 06 2026 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Семенко Станіславу Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Проектування розробка ігрового застосунку

керівник роботи Новожилова Марина Володимирівна

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

затверджені наказом закладу вищої освіти від «22» травня 2026 р. № 440-03

2. Термін подання роботи здобувачем вищої освіти 23 червня 2026 року






3. Вихідні дані до роботи: Наукові та методичні джерела, матеріали дистанційного курсу, внутрішня дизайн-документація ігрового застосунку Cat Tree, структура Unity-проєкту, графічні ресурси, вимоги до функціональності гри та приклади існуючих програмних аналогів. _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): проаналізувати предметну область та наявні аналоги; визначити функціональні вимоги до ігрового застосунку; спроектувати структуру даних, архітектуру та основні програмні модулі; описати алгоритми пасивного доходу, збереження прогресу та відкриття нових зон; розробити програмне та технічне забезпечення застосунку; виконати тестування основних функціональних можливостей; розглянути питання охорони праці

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): UML-діаграма діяльності користувача; UML-діаграма варіантів використання; UML-діаграма класів; схема загальної архітектури застосунку; схема структури програмних модулів; схема структури папок Unity-проєкту; скріншоти

інтерфейсу застосунку; презентаційний матеріал.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 1	Новожилова М. В., керівник роботи		
Розділ 2	Новожилова М. В., керівник роботи		
Розділ 3	Новожилова М. В., керівник роботи		
Розділ 4	Вікторія Малишева, к. т. н., доцент кафедри безпеки життєдіяльності		

7. Дата видачі завдання 12 травня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми кваліфікаційної роботи	05.05.2026	
2	Затвердження теми, наукового керівника, завдання та календарного плану підготовки дипломної роботи	12.05.2026	
3	Написання I розділу	05.06.2026	
4	Написання II розділу	10.06.2026	
5	Написання III розділу	17.06.2026	
6	Підготовка матеріалів з охорони праці	20.06.2026	
7	Подання дипломної роботи керівнику	21.06.2026	
8	Робота по усуненню зауважень керівника, уточненню і доповненню практичного матеріалу, оформленню додатків до роботи	22.06.2026	
9	Подання доопрацьованого варіанту роботи керівнику	23.06.2026	
10	Захист матеріалів дипломної роботи на засіданні кафедри	24.06.2026	

11	Офіційний захист матеріалів дипломної роботи на засіданні Державної екзаменаційної комісії	26.06.2026	
----	--	------------	--

Здобувач вищої освіти



(підпис)

Семенко Станіслав
Віталійович

(прізвище та ініціали)

Керівник роботи



(підпис)

Новожилова Марина
Володимирівна

(прізвище та ініціали)"

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка до кваліфікаційної роботи бакалавра студента групи КН2022-1 спеціальності 122 Комп'ютерні науки Семенка Станіслава Віталійовича за темою «Проектування та розробка ігрового застосунку» складається з 4 розділів, містить 59 сторінок, 12 рисунків, 10 таблиць та 26 використаних джерел.

Кваліфікаційну роботу присвячено проектуванню та розробці ігрового застосунку Cat Tree. У роботі розглянуто предметну область казуальних idle-game, виконано аналіз існуючих аналогів, визначено основні функціональні вимоги до застосунку та описано ключові ігрові механіки: розвиток дерева, отримання і розміщення котів, накопичення ресурсів, відкриття нових зон, роботу з колекціями, декором, шкідниками та додатковими ігровими подіями.

У розділі «Загальні положення» наведено опис предметного середовища, процесу діяльності користувача, функціональної моделі застосунку та огляд наявних аналогів. У розділі «Інформаційне та математичне забезпечення» визначено вхідні й вихідні дані, спроектовано структуру даних та об'єктно-орієнтовану модель системи. У розділі «Програмне та технічне забезпечення» описано засоби розробки, вимоги до програмного й технічного забезпечення, особливості програмної реалізації, систему збереження та тестування основних функцій.

Результатом роботи є ігровий застосунок, розроблений засобами Unity та мови програмування C#, який забезпечує взаємодію користувача з ігровим середовищем, підтримує систему прогресу, збереження даних, керування ігровими об'єктами та роботу графічного інтерфейсу.

Ключові слова: UNITY, C#, ІГРОВИЙ ЗАСТОСУНОК, CAT TREE, IDLE-GAME, КОТИ, ІГРОВА ВАЛЮТА, ПРОЄКТУВАННЯ, ПРОГРЕС, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС.

ABSTRACT

Structure and scope of work. The explanatory note of the bachelor's qualification work of the student of group KN2022-1, specialty 122 Computer Science, Semenko Stanislav Vitaliiiovych, on the topic "Design and implementation of a game application" consists of 4 chapters and contains 59 pages, 12 figures, 10 tables and 26 references.

The qualification work is devoted to the design and development of the game application Cat Tree. The work considers the subject area of casual idle-games, analyzes existing analogues, defines the main functional requirements for the application and describes the key game mechanics: tree development, obtaining and placing cats, resource accumulation, unlocking new zones, working with collections, decorations, pests and additional game events.

The chapter "General provisions" describes the subject environment, the user activity process, the functional model of the application and the review of existing analogues. The chapter "Information and mathematical support" defines input and output data, designs the data structure and object-oriented model of the system. The chapter "Software and technical support" describes development tools, software and hardware requirements, software implementation, data persistence and testing of the main functions.

The result of the work is a game application developed using Unity and the C# programming language. The application provides user interaction with the game environment, supports the progression system, data persistence, game object management and graphical user interface functionality.

Keywords: UNITY, C#, GAME APPLICATION, CAT TREE, IDLE-GAME, CATS, GAME CURRENCY, DESIGN, PROGRESSION, USER INTERFACE.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ЗАДАЧІ ДОСЛІДЖЕННЯ.....	10
1.1 Опис предметного середовища.....	10
1.1.1 Опис процесу діяльності.....	13
1.1.2 Опис функціональної моделі.....	14
1.2 Огляд наявних аналогів.....	16
1.3 Постановка задачі.....	19
Висновки до розділу 1.....	20
РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	21
2.1 Аналіз предметної області.....	21
2.2 Визначення шляхів і методів розв'язання задачі дослідження...	22
2.1.1 Вхідні дані.....	22
2.1.2 Вихідні дані.....	23
2.3 Математичне та алгоритмічне забезпечення.....	31
Висновки до розділу 2.....	34
РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	35
3.1 Засоби розробки.....	35
3.2 Вимоги до технічного та програмного забезпечення.....	37
3.3 Опис програмної реалізації.....	38
3.4 Керівництво користувача.....	41
Висновки до розділу 3.....	45
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	46
4.1 Організаційно-правові основи забезпечення безпеки праці.....	46
4.2 Характеристика об'єкта та виявлення потенційних небезпек....	47

4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проектування та розробка заходів щодо їх попередження.....	52
Висновки до розділу 4.....	56
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58

ВСТУП

Комп'ютерні ігри є однією з найбільш динамічних галузей сучасної індустрії програмного забезпечення. Завдяки розвитку апаратного забезпечення, графічних технологій та засобів розробки ігор створення інтерактивних програмних продуктів стало доступним не лише великим компаніям, а й невеликим командам розробників. Сучасні ігрові застосунки поєднують у собі програмну реалізацію, графічний дизайн, користувацький інтерфейс та механіки взаємодії, що забезпечують залучення користувачів і підтримку їх інтересу протягом тривалого часу.

За даними Newzoo, у 2024 році світовий ринок ігор оцінюється у 187,7 млрд доларів США, а кількість гравців у світі досягає 3,42 млрд осіб [1]. За прогнозами Statista, кількість користувачів ринку відеоігор у світі й надалі зростатиме [2]. Це підтверджує актуальність створення нових ігрових застосунків, зокрема казуальних ігор із довготривалим прогресом користувача.

Одним із популярних напрямів сучасних казуальних ігор є жанр *idle-game*. Для нього характерні поступове накопичення ресурсів, автоматизація частини ігрового процесу та можливість прогресу без постійної активної участі користувача [4]. Такі застосунки мають низький поріг входження, зрозумілий ігровий цикл та можуть підтримувати інтерес гравця за рахунок відкриття нового контенту.

Актуальність теми полягає у необхідності створення сучасного ігрового застосунку, який поєднує простий та зрозумілий ігровий процес із системою довгострокового прогресу користувача. Такий застосунок повинен забезпечувати зручну взаємодію з інтерфейсом, підтримувати механізми накопичення ресурсів, відкриття нового контенту, колекціонування ігрових об'єктів та збереження результатів діяльності гравця.

Об'єктом дослідження є ігровий застосунок *Cat Tree* як програмна система для персонального комп'ютера.

Предметом дослідження є методи, моделі та програмні засоби проектування і розробки ігрового застосунку жанру idle-game із системою прогресу, колекціонуванням ігрових об'єктів та відкриттям нових функціональних можливостей.

Метою кваліфікаційної роботи є проектування та розробка ігрового застосунку Cat Tree, у якому користувач розвиває віртуальне дерево, відкриває нові зони, отримує та колекціонує котів, взаємодіє з допоміжними ігровими механіками та поступово розширює доступні можливості ігрового середовища.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати предметну область та існуючі аналоги;
- визначити функціональні вимоги до програмного застосунку;
- спроектувати структуру системи та основні програмні модулі;
- розробити модель взаємодії користувача з ігровим середовищем;
- реалізувати програмний застосунок засобами Unity та мови програмування C#;
- провести тестування основних функціональних можливостей системи.

Практичне значення отриманих результатів полягає у створенні працездатного ігрового застосунку, який може використовуватися кінцевими користувачами як розважальний програмний продукт та слугувати основою для подальшого розвитку і розширення функціональних можливостей.

РОЗДІЛ 1

АНАЛІЗ ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Опис предметного середовища

Предметною областю даної кваліфікаційної роботи є розробка ігрового застосунку жанру idle-game під назвою Cat Tree. Застосунок призначений для персональних комп'ютерів та орієнтований на користувачів, які надають перевагу казуальним іграм із поступовим розвитком, колекціонуванням ігрових об'єктів та довгостроковою системою прогресу.

Застосунок Cat Tree позиціонується як розслаблююча інді-гра, у якій користувач збирає котів, спостерігає за їх розвитком на дереві та відкриває нові частини дерева у процесі проходження [9]. За класифікацією комп'ютерних ігор розроблюваний застосунок можна віднести до ПК-ігор жанру idle-game з елементами колекціонування та поступового відкриття нового контенту. Узагальнену класифікацію наведено на рис. 1.1.



Рисунок 1.1 – Класифікація комп'ютерних ігор з виділенням напрямку Cat Tree

Статистичні дані щодо розвитку ігрової індустрії підтверджують доцільність створення подібних застосунків. Згідно з даними Newzoo, кількість гравців у світі зростає з 3,145 млрд у 2022 році до прогнозованих 3,759 млрд у 2027 році [1]. Динаміку кількості гравців наведено на рис. 1.2.



Рисунок 1.2 – Динаміка кількості гравців у світі за даними Newzoo

Основою ігрового процесу є розвиток великого віртуального дерева, на якому розміщуються коти. Кожен кіт є окремим ігровим об'єктом, що має власні характеристики, рівень рідкості, швидкість генерації ресурсів та зовнішній вигляд. Гравець поступово збільшує кількість доступних котів, розміщує їх на дереві та отримує ігрову валюту у вигляді листків.

Центральним елементом прогресу є система рівнів дерева. На початку гри користувачу доступний лише перший рівень. У процесі накопичення ресурсів відкриваються нові зони дерева, які надають додаткові місця для розміщення котів, відкривають нові механіки та розширюють доступні можливості застосунку. Усього в грі передбачено сім рівнів розвитку дерева.

Для урізноманітнення ігрового процесу застосунок містить додаткові механіки, серед яких система колекцій, декоративні об'єкти, мінікарта, офлайн-дохід, тематичні набори котів та система випадкових подій. Важливу

роль у взаємодії з користувачем виконує персонаж сови, яка супроводжує гравця під час проходження навчання, відкриття нових механік та переходу між рівнями розвитку дерева.

На пізніх етапах проходження користувачу стають доступними додаткові механіки, пов'язані зі шкідниками. Жуки періодично з'являються на дереві та впливають на ефективність отримання ресурсів. Для боротьби з ними передбачено використання спеціальних інструментів, зокрема клейкої стрічки, банки та спрею від шкідників.

Таким чином, предметна область поєднує механіки накопичення ресурсів, колекціонування котів, відкриття нових зон, керування ігровими об'єктами та взаємодії користувача з багаторівневим віртуальним середовищем. Загальний вигляд ігрового світу застосунку наведено на рис. 1.3.

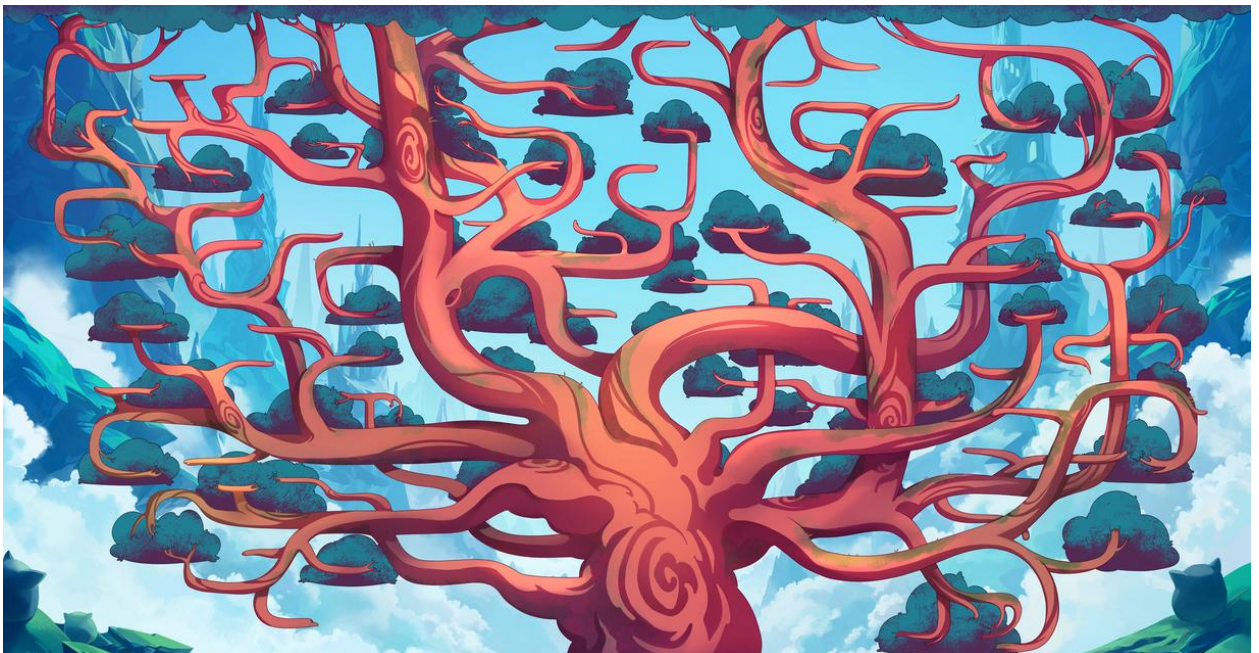


Рисунок 1.3 – Загальний вигляд ігрового світу застосунку Cat Tree

На рис. 1.3 показано багаторівневе дерево, яке є основним простором взаємодії користувача з грою. На ньому розміщуються коти, декоративні об'єкти та інші інтерактивні елементи. У процесі проходження користувач відкриває нові ділянки дерева, що розширюють доступний ігровий простір та відкривають додаткові можливості для розвитку.

1.1.1 Опис процесу діяльності

Ігровий застосунок Cat Tree належить до жанру idle-game та базується на механіці поступового розвитку ігрового середовища. Основною метою користувача є розширення дерева, отримання нових котів, збільшення доходу та відкриття додаткового контенту.

Після першого запуску гри користувач отримує доступ до початкової зони дерева та проходить коротке ознайомлення з основними елементами інтерфейсу. Важливу роль на початкових етапах виконує персонаж сови, яка супроводжує гравця та надає інформацію щодо нових можливостей ігрового процесу.

Основними ігровими об'єктами є коти. Нові коти потрапляють до гравця за допомогою спеціальних паків, які відкриваються після отримання або придбання. Кожен кіт має власні характеристики, рівень рідкості та швидкість генерації ігрової валюти. Після отримання kota користувач може розмістити його на доступному місці дерева.

Коти автоматично генерують ігрову валюту у вигляді листків. Отримані ресурси використовуються для подальшого розвитку застосунку: придбання нових паків, декоративних об'єктів, спеціальних інструментів та відкриття нових зон дерева. У міру накопичення достатньої кількості ресурсів користувач переходить на наступний рівень розвитку дерева.

Кожний новий рівень відкриває додаткові ділянки дерева та надає доступ до нових механік. Наприклад, на третьому рівні стає доступною система декору, а на четвертому рівні відкривається механіка шкідників. Завдяки цьому процес розвитку не обмежується лише накопиченням ресурсів і поступово знайомить користувача з новими функціональними можливостями застосунку.

Окремою складовою ігрового процесу є система колекцій. Після отримання та повного вирощування нового kota відповідний запис автоматично додається до колекції користувача. Колекції дозволяють

відстежувати прогрес проходження та мотивують гравця відкривати нових персонажів.

На пізніших етапах гри з'являються шкідники у вигляді жуків. Їх поява негативно впливає на ефективність отримання доходу. Для боротьби зі шкідниками користувач може застосовувати спеціальні інструменти: клейку стрічку, банку для збору жуків та спрей від шкідників. Використання цих механік дозволяє підтримувати ефективність роботи дерева та зменшувати негативний вплив випадкових подій.

Таким чином, процес діяльності користувача складається з отримання нових котів, їх розміщення на дереві, накопичення ресурсів, відкриття нових зон, взаємодії з додатковими механіками та поступового розвитку ігрового середовища. Для наочного представлення основного процесу взаємодії користувача з ігровою системою побудовано UML-діаграму діяльності, наведену на рис. 1.4.

1.1.2 Опис функціональної моделі

Функціональна модель ігрового застосунку Cat Tree описує основні можливості системи з точки зору користувача. Головним актором системи є гравець, який взаємодіє з ігровим середовищем через графічний інтерфейс.

Основною функцією застосунку є розвиток дерева. Для цього користувач отримує котів, розміщує їх на доступних місцях дерева, накопичує ігрову валюту та відкриває нові зони. Кожна нова зона розширює доступний ігровий простір і може відкривати додаткові механіки.

Однією з важливих функцій є отримання та колекціонування котів. Коти мають різну рідкість, зовнішній вигляд та показники пасивного доходу. Після отримання нового кота користувач може переглянути його характеристики, розмістити на дереві або продати. Якщо кіт був повністю вирощений, він додається до відповідної колекції.

UML Activity Diagram - Cat Tree



Рисунок 1.4 – UML-діаграма діяльності користувача в застосунку Cat Tree

Система ресурсів забезпечує економічну основу ігрового процесу. Основною валютою є листки, які автоматично генеруються котами. Листки використовуються для придбання нових паків котів, відкриття зон, купівлі декорацій та спеціальних предметів.

Окрему функціональну частину становить система декору. Вона дозволяє користувачу розміщувати декоративні об'єкти на дереві, змінювати їх положення, видаляти або повертати частину витрачених ресурсів. Така механіка розширює можливості персоналізації ігрового середовища.

Після досягнення відповідного рівня користувачу стає доступною система шкідників. Жуки з'являються на дереві та зменшують ефективність пасивного доходу. Для взаємодії з ними передбачені спеціальні предмети: клейка стрічка, банка та спрей від шкідників. Це створює додатковий рівень взаємодії користувача з ігровим світом.

Допоміжними функціями застосунку є перегляд колекцій, робота з мінікартою, взаємодія з совою, зміна налаштувань, збереження прогресу та отримання офлайн-доходу. Завдяки цим функціям система забезпечує повний цикл взаємодії користувача з ігровим застосунком. Для наочного представлення функціональних можливостей застосунку побудовано UML-діаграму варіантів використання, наведену на рис. 1.5.

1.2 Огляд наявних аналогів

Для визначення основних вимог до ігрового застосунку Cat Tree було розглянуто існуючі програмні продукти, які мають подібні механіки поступового розвитку, колекціонування об'єктів, накопичення ресурсів та взаємодії з користувачем. Аналіз аналогів дозволяє визначити сильні та слабкі сторони існуючих рішень і сформулювати вимоги до власного програмного застосунку.

Одним із найближчих аналогів є гра Chillquarium. Це казуальна idle-game, у якій користувач купує риб, вирощує їх у реальному часі, отримує прибуток від продажу та поступово розширює власну колекцію [5]. Подібність із Cat Tree полягає у механіці колекціонування, поступовому відкритті нових об'єктів та пасивному накопиченні ресурсів.

UML Use Case Diagram - Cat Tree



Рисунок 1.5 – UML-діаграма варіантів використання застосунку Cat Tree

Іншим прикладом є Cookie Clicker. У цій грі основний процес побудований навколо автоматичного накопичення ресурсів і придбання покращень, що збільшують швидкість отримання прибутку [6]. Основна перевага Cookie Clicker полягає у простоті взаємодії, однак візуальна складова та персоналізація ігрового середовища є обмеженими.

Гра AdVenture Capitalist також належить до жанру idle-game. Її ігровий процес базується на купівлі бізнесів, автоматизації прибутку та поступовому збільшенні доходу [7]. На відміну від Cat Tree, ця гра має економічну тематику і не використовує систему колекціонування персонажів як основний елемент прогресу.

Ще одним аналогом є Rusty's Retirement. Це idle-гра, у якій ігровий процес відбувається у спокійному темпі та поєднує автоматизоване накопичення ресурсів із візуально приємним середовищем [8]. Подібність із Cat Tree полягає у казуальній атмосфері та орієнтації на довготривалий прогрес без високого навантаження на користувача.

Порівняння розглянутих аналогів наведено в табл. 1.1.

Таблиця 1.1 – Порівняльна характеристика аналогів

Назва застосунку	Пасивний дохід	Колекціонування	Відкриття нових зон	Декорування середовища	Система випадкових подій
Chillquarium	Так	Так	Ні	Ні	Ні
Cookie Clicker	Так	Ні	Ні	Ні	Ні
AdVenture Capitalist	Так	Ні	Ні	Ні	Ні
Rusty's Retirement	Так	Частково	Ні	Частково	Ні
Cat Tree	Так	Так	Так	Так	Так

У результаті аналізу було визначено, що більшість існуючих аналогів зосереджені на одній головній механіці: накопиченні ресурсів, розвитку економіки або колекціонуванні об'єктів. Особливістю Cat Tree є поєднання кількох напрямів: розвитку дерева, отримання котів, відкриття зон, декорування середовища, взаємодії з совою та механіки шкідників.

Таким чином, розроблюваний застосунок відрізняється від розглянутих аналогів більш вираженою візуальною персоналізацією, наявністю

персонажів, системою рівнів дерева та додатковими інтерактивними механіками.

1.3 Постановка задачі

Метою кваліфікаційної роботи є проектування та розробка ігрового застосунку Cat Tree, який забезпечує користувачеві можливість поступового розвитку віртуального дерева, отримання та колекціонування котів, відкриття нових зон і використання додаткових ігрових механік.

Розроблюваний застосунок належить до жанру idle-game та орієнтований на користувачів, які віддають перевагу казуальним іграм із довготривалим прогресом. Основна особливість застосунку полягає у поєднанні системи пасивного доходу, колекціонування персонажів, розвитку ігрового середовища та використання допоміжних механік, які поступово відкриваються під час проходження гри.

Основним користувачем системи є гравець. Він взаємодіє з ігровим середовищем за допомогою графічного інтерфейсу, отримує нових котів, розміщує їх на дереві, накопичує ігрову валюту, відкриває нові ділянки дерева та використовує додаткові можливості застосунку.

У межах виконання роботи необхідно вирішити такі задачі:

- виконати аналіз предметної області та існуючих аналогів;
- визначити функціональні вимоги до програмного застосунку;
- спроектувати структуру даних і програмних компонентів системи;
- реалізувати механіку отримання та розміщення котів;
- реалізувати систему пасивного накопичення ресурсів;
- реалізувати механіку відкриття нових зон дерева;
- реалізувати систему колекцій та обліку отриманих котів;
- реалізувати допоміжні механіки взаємодії з користувачем;
- реалізувати систему боротьби зі шкідниками;
- забезпечити збереження та завантаження прогресу користувача;

– провести тестування основних функціональних можливостей застосунку.

Результатом виконання роботи повинен стати програмний застосунок, який забезпечує стабільну роботу основних ігрових механік, підтримує довготривалий прогрес користувача та створює зручне середовище для взаємодії з ігровим світом.

Висновки до розділу 1

У першому розділі було проведено аналіз предметної області розробки ігрових застосунків жанру idle-game. Розглянуто особливості функціонування програмного застосунку Cat Tree, визначено основні механіки взаємодії користувача з ігровим середовищем та побудовано UML-діаграми, що відображають процес діяльності користувача і функціональні можливості системи.

Також було виконано огляд існуючих аналогів, серед яких Chillquarium, Cookie Clicker, AdVenture Capitalist та Rusty's Retirement. Проведене порівняння дозволило визначити характерні особливості розроблюваного застосунку та сформулювати перелік основних функціональних вимог.

На основі проведеного аналізу сформульовано задачі кваліфікаційної роботи, виконання яких забезпечить створення програмного застосунку з механіками колекціонування котів, накопичення ресурсів, відкриття нових зон та взаємодії з додатковими ігровими системами.

РОЗДІЛ 2

ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз предметної області

Інформаційне та математичне забезпечення ігрового застосунку Cat Tree базується на визначенні основних об'єктів системи, їх властивостей, взаємозв'язків та алгоритмів обробки ігрових подій. Оскільки застосунок належить до жанру idle-game, основна логіка системи пов'язана з накопиченням ресурсів, збереженням прогресу користувача, відкриттям нових зон дерева та керуванням ігровими об'єктами.

Предметна область застосунку включає декілька основних сутностей: гравця, котів, дерево, зони дерева, ігрову валюту, колекції, декоративні об'єкти, шкідників, інструменти боротьби зі шкідниками та налаштування користувача. Кожна з цих сутностей має власний набір параметрів і використовується в окремих підсистемах застосунку.

Гравець є основним користувачем системи. Він виконує дії через графічний інтерфейс: відкриває паки з котами, розміщує котів на дереві, накопичує валюту, купує нові зони, переглядає колекції, використовує інструменти та змінює налаштування застосунку. Такий набір дій визначає основний цикл взаємодії користувача із системою.

Кіт є основним ігровим об'єктом, який забезпечує пасивне отримання ресурсів. Для кожного kota визначаються назва, рідкість, належність до колекції, стан росту, показник доходу та можливість розміщення на дереві. Після повного вирощування kota інформація про нього додається до колекції користувача.

Дерево виступає центральним ігровим середовищем. Воно складається з окремих зон, які відкриваються поступово у процесі розвитку. Кожна зона може містити місця для розміщення котів, декоративні об'єкти та спавнери

додаткових ігрових подій. Відкриття нових зон пов'язане з рівнем дерева та кількістю накопичених ресурсів.

Основною ігровою валютою є листки. Вони автоматично генеруються котами та використовуються для придбання нових паків, відкриття зон, купівлі декору та спеціальних предметів. Така валюта забезпечує економічну основу ігрового процесу.

Додатковими сутностями предметної області є шкідники та інструменти боротьби з ними. Шкідники у вигляді жуків з'являються на відкритих зонах дерева та можуть зменшувати ефективність пасивного доходу. Для взаємодії з ними використовуються клейка стрічка, банка та спрей від шкідників.

Таким чином, предметна область застосунку Cat Tree складається з взаємопов'язаних ігрових об'єктів, які забезпечують поступовий розвиток дерева, отримання ресурсів, відкриття нового контенту та підтримку довготривалого прогресу користувача.

2.1.1 Вхідні дані

Для забезпечення роботи ігрового застосунку Cat Tree використовуються вхідні дані, які надходять від користувача або завантажуються з конфігураційних файлів системи. Дані цього типу використовуються для формування ігрового середовища, обробки дій користувача та збереження поточного стану гри.

Основною групою вхідних даних є дії користувача. Під час роботи із застосунком гравець взаємодіє з елементами інтерфейсу, відкриває паки з котами, розміщує котів на дереві, відкриває нові зони, купує декоративні об'єкти та використовує допоміжні інструменти для боротьби зі шкідниками.

Окрему категорію становлять конфігураційні дані. Вони містять параметри котів, колекцій, декоративних об'єктів, шкідників, зон дерева та інших ігрових елементів. Конфігураційні дані визначають характеристики об'єктів і використовуються системою під час запуску застосунку.

Також до вхідних даних належать дані збереження прогресу користувача. Вони містять інформацію про кількість отриманих котів, відкриті зони дерева, накопичену валюту, придбані предмети та поточний рівень розвитку гри. Основні вхідні дані системи наведено в табл. 2.1.

Таблиця 2.1 – Основні вхідні дані системи

№	Вхідні дані	Призначення
1	Дані котів	Зберігання характеристик котів, їх рідкості, доходу, стадії росту та належності до колекцій
2	Дані зон дерева	Визначення відкритих і закритих зон, вартості розблокування та доступних місць для котів
3	Дані колекцій	Формування списків отриманих котів та відображення прогресу заповнення колекції
4	Дані шкідників	Налаштування появи жуків, параметрів їх впливу та доступних способів видалення
5	Дані декору	Опис декоративних об'єктів, їх вартості та параметрів розміщення на дереві
6	Дані користувача	Інформація про дії користувача в інтерфейсі застосунку
7	Дані збереження	Поточний прогрес, валюта, відкриті зони та отримані ігрові об'єкти

Як показано в табл. 2.1, система використовує як інформацію, що надходить від користувача під час роботи із застосунком, так і службові конфігураційні дані, необхідні для коректного функціонування всіх ігрових механік.

2.1.2 Вихідні дані

Результатом роботи програмного застосунку Cat Tree є формування та відображення інформації, необхідної користувачу для взаємодії з ігровим середовищем. Вихідні дані формуються на основі поточного стану гри, виконаних дій користувача та внутрішніх розрахунків системи.

Основну частину вихідних даних становить інформація про поточний прогрес користувача. До неї належать кількість накопиченої валюти, рівень

розвитку дерева, відкриті зони, доступні механіки та загальний стан ігрового світу.

Окрему категорію становлять дані про котів. Система відображає інформацію про отриманих котів, їх рівень рідкості, поточний стан росту, належність до колекцій та величину пасивного доходу, який вони генерують.

Важливими вихідними даними є повідомлення та візуальні елементи інтерфейсу. До них належать інформаційні вікна, підказки, діалоги сови, повідомлення про відкриття нових зон, отримання нагород та інші елементи взаємодії з користувачем.

Крім того, система формує службові дані для механізму збереження прогресу, що забезпечує можливість продовження гри після повторного запуску застосунку. Основні вихідні дані системи наведено в табл. 2.2.

Таблиця 2.2 – Основні вихідні дані системи

№	Вихідні дані	Призначення
1	Поточний баланс валюти	Відображення кількості накопичених листків
2	Дані про котів	Виведення характеристик, стану росту та доходу котів
3	Дані про дерево	Відображення рівня дерева, відкритих зон і доступних місць розміщення
4	Дані колекцій	Показ прогресу отримання та вирощування котів
5	Інформаційні повідомлення	Підказки, нагороди, діалоги сови та системні повідомлення
6	Дані про шкідників	Відображення кількості жуків, їх стану та впливу на дохід
7	Дані інтерфейсу	Оновлення меню, кнопок, мінікарти, поп-апів та інших UI-елементів
8	Дані збереження	Формування інформації для запису та подальшого завантаження прогресу

Таким чином, вихідні дані забезпечують відображення поточного стану ігрового середовища та надають користувачу всю необхідну інформацію для подальшої взаємодії із застосунком. Формування вихідних даних відбувається автоматично в процесі роботи системи та безпосередньо залежить від дій користувача і внутрішніх механізмів гри.

2.2 Проектування системи

Після визначення основних вхідних та вихідних даних необхідно спроектувати структуру програмного застосунку. Проектування системи дозволяє визначити основні програмні компоненти, їх призначення та взаємозв'язки між ними.

Оскільки застосунок Cat Tree реалізується засобами Unity, його архітектура базується на об'єктно-орієнтованому підході. Кожна ігрова сутність представлена окремим програмним об'єктом із власними властивостями та методами взаємодії. Такий підхід спрощує підтримку проєкту, розширення функціональності та повторне використання програмного коду [3].

До основних програмних компонентів системи належать модулі керування котами, зонами дерева, колекціями, економікою гри, декоративними об'єктами, шкідниками, інтерфейсом користувача та механізмом збереження прогресу. Кожен із зазначених модулів виконує окремий набір функцій і взаємодіє з іншими частинами системи через спільні структури даних.

Для забезпечення цілісності системи всі компоненти працюють у межах єдиного ігрового середовища та використовують спільні дані про поточний стан гри. Такий підхід дозволяє підтримувати актуальність інформації про ресурси, відкриті зони, отриманих котів та інші об'єкти ігрового світу.

Загальну архітектуру програмного застосунку наведено на рис. 2.1.

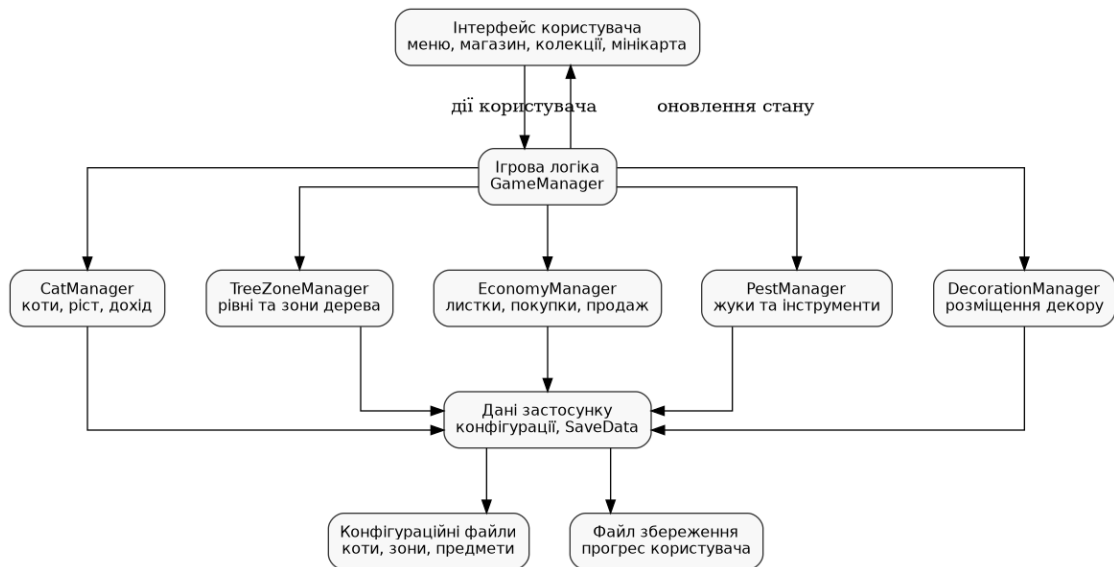


Рисунок 2.1 – Загальна архітектура програмного застосунку Cat Tree

На рис. 2.1 показано, що взаємодія користувача відбувається через інтерфейсний рівень, який передає дії до ігрової логіки. Ігрова логіка координує роботу окремих менеджерів, а дані застосунку використовуються для завантаження конфігурацій та збереження прогресу користувача.

2.2.1 Проєктування структури даних та локального сховища

У застосунку Cat Tree не використовується класична реляційна база даних серверного типу, оскільки гра працює локально та орієнтована на одного користувача. З цієї причини зберігання даних реалізовано у вигляді локального сховища стану гри та конфігураційних файлів, що містять параметри ігрових об'єктів.

Конфігураційні дані містять сталі параметри ігрових об'єктів: характеристики котів, ціни паків, умови відкриття зон, параметри шкідників, декоративних об'єктів та спеціальних інструментів. Дані збереження містять змінний стан гри: баланс валюти, відкриті зони, отриманих котів, стан колекцій, налаштування користувача та час останнього входу до гри.

Такий підхід відповідає особливостям розробки невеликих і середніх ігрових застосунків у Unity, де для зберігання параметрів часто використовуються локальні структури даних, серіалізація та конфігураційні

об'єкти. Розділення сталих конфігураційних даних і змінного прогресу користувача дає змогу розширювати контент без переробки основної логіки гри.

Структуру основних сутностей програмного застосунку наведено в табл.

2.3.

Таблиця 2.3 – Основні сутності системи

№	Сутність	Призначення
1	PlayerData	Зберігає загальний прогрес користувача, баланс валюти та рівень дерева
2	CatData	Описує kota, його рідкість, дохід, стан росту та належність до колекції
3	TreeData	Містить інформацію про рівень дерева та відкриті зони
4	ZoneData	Описує окрему зону дерева, її стан і умови відкриття
5	CollectionData	Зберігає інформацію про отриманих котів та прогрес колекції
6	DecorationData	Описує декоративні об'єкти, їх вартість та позицію розміщення
7	PestData	Містить параметри шкідників, їх кількість і вплив на пасивний дохід
8	SaveData	Використовується для збереження та завантаження стану гри

Як видно з табл. 2.3, структура даних побудована навколо основних ігрових сутностей. Це дає можливість окремо зберігати конфігураційні параметри та поточний стан користувача.

Для опису зв'язків між сутностями доцільно виділити два типи даних: сталі конфігураційні дані та змінні дані прогресу. Конфігураційні дані визначають правила роботи системи, а дані прогресу змінюються під час проходження гри. Таке розділення наведено в табл. 2.4.

Таблиця 2.4 – Розподіл даних за типом використання

№	Тип даних	Приклади даних
1	Конфігураційні дані	Параметри котів, ціни предметів, умови відкриття зон, характеристики декору
2	Дані прогресу	Баланс листків, відкриті зони, отримані коти, заповнені колекції
3	Службові дані	Час останнього збереження, параметри офлайн-доходу, стан інтерфейсу

Поділ даних на конфігураційні, прогресивні та службові спрощує підтримку застосунку. У разі додавання нових котів, зон або предметів розробнику достатньо змінити конфігураційні параметри без переробки основної логіки гри.

2.2.2 Побудова об'єктно-орієнтованої моделі

Об'єктно-орієнтована модель застосунку Cat Tree відображає основні класи системи та зв'язки між ними. Її побудова необхідна для формалізації структури програмного забезпечення та подальшого переходу до програмної реалізації.

Основним класом системи є Player, який зберігає інформацію про поточний прогрес користувача, кількість накопичених листків, рівень дерева та відкритий контент. Клас Player взаємодіє з іншими класами через операції відкриття паків, купівлі зон і збереження прогресу.

Клас Tree відповідає за стан дерева, рівень розвитку та кількість відкритих зон. З ним пов'язані класи Zone, Cat, Decoration і Pest, оскільки саме дерево є основним середовищем розміщення ігрових об'єктів.

Клас Cat описує основні характеристики кота: назву, рідкість, швидкість генерації доходу та стадію росту. Для обліку отриманих котів використовується клас Collection, який містить записи про відкритих персонажів та дозволяє відображати прогрес колекціонування.

Клас Decoration використовується для реалізації декоративних об'єктів, які користувач може розміщувати на дереві. Клас Pest відповідає за появу шкідників і їх вплив на пасивний дохід. Клас SaveData забезпечує збереження і завантаження прогресу користувача.

Для відображення основних класів системи та взаємозв'язків між ними побудовано UML-діаграму класів, наведену на рис. 2.2.

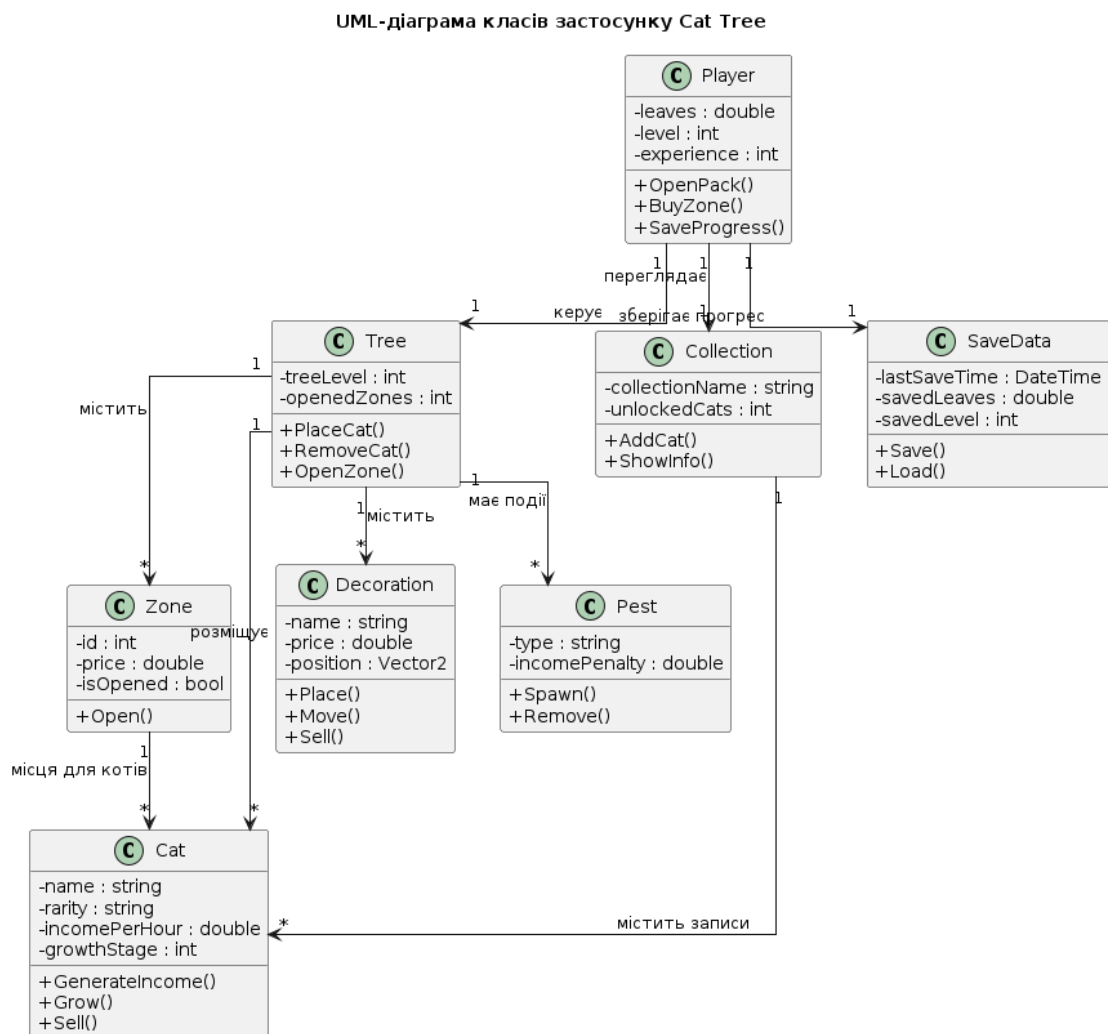


Рисунок 2.2 – UML-діаграма класів застосунку Cat Tree

Побудована UML-діаграма класів показує, що система має модульну структуру. Основні ігрові об'єкти описані окремими класами, а взаємодія між ними забезпечує роботу ключових механік: отримання котів, накопичення валюти, відкриття зон, роботу колекцій, декорування та збереження прогресу.

Перевагою такої моделі є можливість подальшого розширення застосунку. Наприклад, додавання нового типу декору або нового виду шкідників не потребує зміни всієї структури системи, оскільки відповідні механіки ізольовані в окремих класах.

Основні програмні модулі, що відповідають класам та ігровим підсистемам, наведено в табл. 2.5.

Таблиця 2.5 – Основні програмні модулі застосунку

№	Модуль	Основні функції
1	CatManager	Створення, розміщення, вирощування котів та розрахунок їх доходу
2	TreeZoneManager	Керування рівнями дерева, відкритими зонами та доступними місцями
3	EconomyManager	Облік листків, купівля паків, зон, декору та спеціальних предметів
4	CollectionManager	Оновлення колекцій і відображення прогресу отримання котів
5	DecorationManager	Розміщення, переміщення та видалення декоративних об'єктів
6	PestManager	Поява жуків, облік їх кількості та застосування штрафів до доходу
7	SaveManager	Збереження та завантаження прогресу користувача
8	UIManager	Оновлення графічного інтерфейсу та обробка взаємодії з меню

Наведені модулі забезпечують логічне розділення відповідальності між окремими частинами застосунку. Завдяки цьому кожна підсистема може розроблятися та тестуватися окремо, а загальна система залишається зрозумілою для подальшого супроводу.

2.3 Математичне та алгоритмічне забезпечення

Математичне та алгоритмічне забезпечення застосунку Cat Tree використовується для реалізації механік пасивного доходу, прогресу росту котів, відкриття нових зон дерева, роботи офлайн-доходу та визначення впливу додаткових ігрових подій на стан системи.

Основою економічної системи є механізм пасивного доходу. Кожен кіт генерує певну кількість ігрової валюти за одиницю часу. Загальний дохід дерева визначається як сума доходів усіх активних котів, які розміщені на відкритих зонах дерева.

Для розрахунку базового доходу використовується формула

$$D = \sum d_i, i = 1..n, \quad (2.1)$$

де D – загальний пасивний дохід за одиницю часу; d_i – дохід окремого кота; n – кількість активних котів на дереві.

У разі появи шкідників пасивний дохід може зменшуватися. Для цього використовується коефіцієнт штрафу k , який залежить від кількості жуків на дереві. Фактичний дохід з урахуванням шкідників визначається за формулою (2.2).

$$D_{\text{fact}} = D \times (1 - k), \quad (2.2)$$

де D_{fact} – фактичний дохід після врахування впливу шкідників; D – базовий дохід; k – коефіцієнт зменшення доходу.

Якщо активовано позитивний ефект, наприклад бонус від статуї сови, дохід може бути збільшений на відповідний коефіцієнт. У такому випадку використовується формула

$$D_{\text{bonus}} = D_{\text{fact}} \times (1 + b), \quad (2.3)$$

де D_{bonus} – дохід після застосування бонусу; b – коефіцієнт збільшення доходу.

Окремим алгоритмом є розрахунок офлайн-доходу. Якщо користувач не перебуває у грі певний час, система обчислює кількість ресурсів, яку могли згенерувати коти за цей період. Для запобігання надмірному накопиченню ресурсів використовується обмеження максимального часу нарахування.

Офлайн-дохід визначається за формулою

$$D_{\text{offline}} = D_{\text{bonus}} \times \min(t, t_{\text{max}}), \quad (2.4)$$

де D_{offline} – кількість ресурсів, отриманих під час відсутності користувача; t – фактичний час відсутності; t_{max} – максимальний час, за який може бути нарахований офлайн-дохід.

Для відкриття нових зон використовується алгоритм перевірки достатності ресурсів. Якщо поточний баланс користувача не менший за встановлену вартість зони, система списує необхідну кількість листків, змінює стан зони на відкритий та оновлює інтерфейс користувача.

Послідовність розрахунку пасивного доходу та оновлення стану гри наведено на рис. 2.3.



Рисунок 2.3 – Алгоритм розрахунку пасивного доходу в застосунку Cat Tree

Алгоритм, наведений на рис. 2.3, показує загальний порядок оновлення економічної частини гри. Спочатку система отримує список активних котів, після чого розраховує базовий дохід. Далі перевіряється наявність шкідників і бонусних ефектів, після чого фінальне значення доходу додається до балансу користувача.

Алгоритм відкриття нової зони дерева складається з таких етапів:

- перевірка поточного балансу користувача;
- порівняння балансу з вартістю зони;
- списання необхідної кількості листків;
- зміна стану зони на відкритий;
- оновлення мінікарти та доступних місць для котів;
- збереження нового стану гри.

Алгоритм роботи зі шкідниками включає появу жука на доступному спавнері, фіксацію його стану, застосування штрафу до пасивного доходу та видалення шкідника після взаємодії користувача з ним. Якщо користувач застосовує спеціальні предмети, система додатково перевіряє стан клейкої стрічки, банки або спрею від шкідників.

Таким чином, математичне та алгоритмічне забезпечення застосунку Cat Tree описує основні правила роботи ігрової економіки, оновлення прогресу користувача та взаємодії з ключовими ігровими об'єктами. Ці моделі використовуються як основа для подальшої програмної реалізації застосунку.

Висновки до розділу 2

У другому розділі було проведено аналіз інформаційного забезпечення застосунку Cat Tree. Визначено основні вхідні та вихідні дані системи, описано ключові сутності предметної області та встановлено їх призначення у межах ігрового процесу.

Виконано проектування системи, визначено структуру даних, побудовано загальну архітектуру програмного застосунку та об'єктно-орієнтовану модель. Розглянуто основні класи й програмні модулі, що забезпечують роботу котів, зон дерева, колекцій, економіки, декору, шкідників та механізму збереження прогресу.

Також описано математичне та алгоритмічне забезпечення, яке використовується для розрахунку пасивного доходу, обліку штрафів і бонусів, нарахування офлайн-доходу та відкриття нових зон дерева. Отримані результати є основою для програмної реалізації застосунку та подальшого опису програмного і технічного забезпечення.

РОЗДІЛ 3

ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Розробка ігрового застосунку Cat Tree виконувалася з використанням програмних засобів, призначених для створення інтерактивних застосунків та комп'ютерних ігор. Вибір інструментів розробки здійснювався з урахуванням особливостей жанру idle-game, необхідності роботи з двовимірною графікою, підтримки об'єктно-орієнтованого програмування та можливості подальшого розширення функціональності проекту.

Основним середовищем розробки було обрано Unity. Unity є рушієм для створення комп'ютерних ігор та інтерактивних застосунків, що підтримує роботу з двовимірною і тривимірною графікою, системою сцен, анімацією, інтерфейсом користувача та компонентною моделлю побудови об'єктів [3].

Програмна логіка застосунку реалізована мовою програмування C#. Використання C# забезпечує підтримку об'єктно-орієнтованого підходу, модульність програмного коду та зручну інтеграцію зі стандартними засобами Unity [10].

Для написання та налагодження програмного коду використовувалося інтегроване середовище розробки Microsoft Visual Studio. Воно забезпечує підсвічування синтаксису, автоматичне доповнення коду, засоби налагодження та інструменти аналізу помилок [11].

Для зберігання конфігураційних параметрів ігрових об'єктів використовувалися ScriptableObject. Такий підхід дозволяє відокремити дані від програмної логіки та спрощує налаштування параметрів котів, зон дерева, декоративних об'єктів і спеціальних предметів [12]. Для збереження прогресу користувача застосовується локальна серіалізація даних [13].

Під час розробки також використовувалися засоби контролю версій Git, що дають змогу зберігати історію змін проекту та повертатися до попередніх

версій у разі необхідності [17]. Основні засоби розробки застосунку наведено в табл. 3.1.

Таблиця 3.1 – Засоби розробки застосунку

Засіб	Призначення
Unity	Розробка ігрового застосунку, налаштування сцен, об'єктів, анімацій та інтерфейсу
C#	Реалізація програмної логіки та взаємодії між модулями
Visual Studio	Написання, редагування та налагодження програмного коду
ScriptableObject	Зберігання конфігураційних параметрів ігрових об'єктів
Git	Контроль версій та збереження історії змін проекту
Графічні редактори	Підготовка та редагування графічних ресурсів застосунку

Використання зазначених інструментів дозволило реалізувати основні функціональні можливості застосунку та забезпечити зручну підтримку проекту під час подальшого розвитку.

3.2 Вимоги до технічного та програмного забезпечення

Для коректної роботи ігрового застосунку Cat Tree необхідна наявність персонального комп'ютера з характеристиками, достатніми для виконання двовимірних графічних застосунків, розроблених на базі рушія Unity. Оскільки застосунок не використовує складних тривимірних сцен та високонавантажених обчислень, системні вимоги залишаються помірними.

Мінімальні технічні вимоги наведено в табл. 3.2. Вони визначають конфігурацію, достатню для запуску застосунку та виконання основних ігрових дій.

Таблиця 3.2 – Мінімальні системні вимоги

Компонент	Мінімальні характеристики
Операційна система	Windows 10 (64-bit)
Процесор	Intel Core i3 або аналогічний
Оперативна пам'ять	4 ГБ
Відеоадаптер	DirectX 11 Compatible
Вільне місце на диску	1 ГБ
Роздільна здатність екрана	1280×720

Для комфортної роботи застосунку доцільно використовувати характеристики, наведені в табл. 3.3.

Таблиця 3.3 – Рекомендовані системні вимоги

Компонент	Рекомендовані характеристики
Операційна система	Windows 10/11 (64-bit)
Процесор	Intel Core i5 або вище
Оперативна пам'ять	8 ГБ
Відеоадаптер	DirectX 11 Compatible
Вільне місце на диску	2 ГБ
Роздільна здатність екрана	1920×1080

Програмне забезпечення користувача повинно підтримувати запуск застосунків, створених на Unity. Додаткового встановлення серверних компонентів або зовнішніх баз даних не потрібно, оскільки гра є однокористувацькою та працює локально.

Для збереження прогресу використовується файлова система комп'ютера користувача. Збереження даних виконується автоматично під час ігрового процесу та при завершенні роботи застосунку. Такий підхід спрощує запуск гри та не вимагає підключення до мережі Інтернет для основних функцій.

3.3 Опис програмної реалізації

Програмна реалізація застосунку Cat Tree побудована на компонентній архітектурі Unity. Основні функціональні можливості системи реалізовано у вигляді окремих програмних модулів, кожен з яких відповідає за певну частину ігрової логіки.

Загальну структуру програмних модулів застосунку наведено на рис. 3.1.

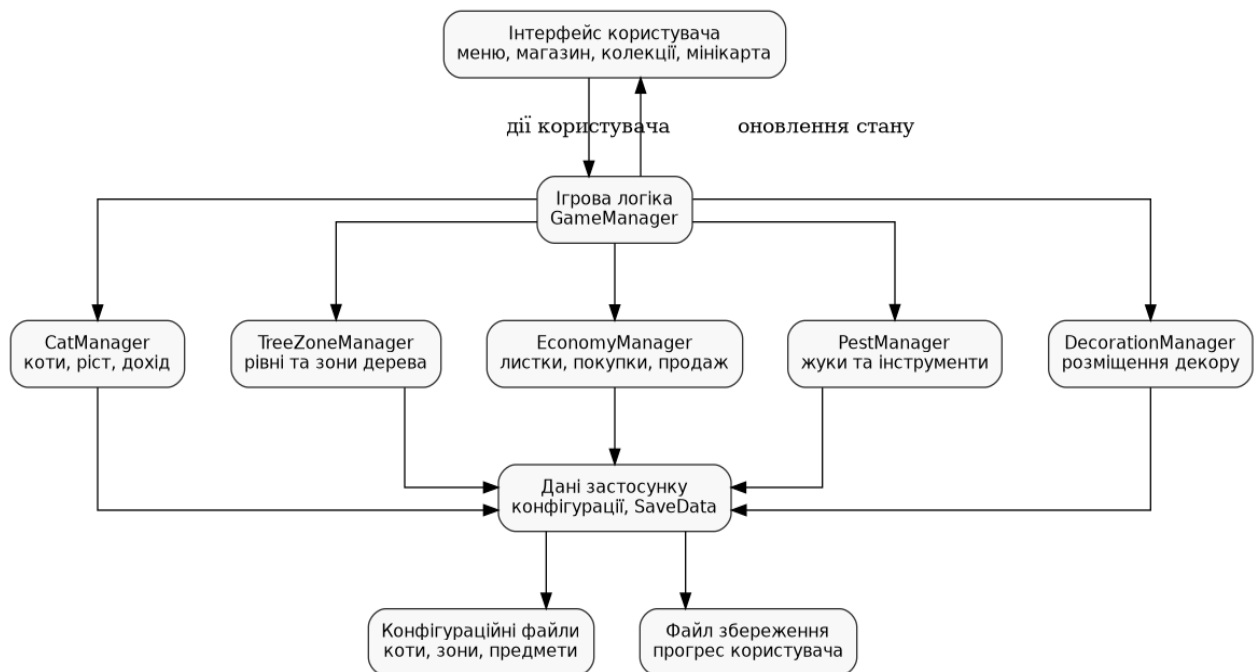


Рисунок 3.1 – Структура програмних модулів застосунку Cat Tree

Як показано на рис. 3.1, центральним елементом системи є GameManager, який координує роботу основних ігрових підсистем. Він відповідає за загальне керування станом гри, обробку ключових подій та взаємодію між менеджерами.

Модуль CatManager відповідає за створення, вирощування та розміщення котів на дереві. Після відкриття нового пака система створює нового ігрового персонажа та додає його до списку доступних котів користувача.

Модуль EconomyManager забезпечує облік ігрової валюти, нарахування пасивного доходу, виконання покупок і перевірку доступності ресурсів для відкриття нового контенту. Саме цей модуль використовується під час придбання зон, декору та спеціальних предметів.

Модуль TreeZoneManager керує рівнями дерева та відкритими зонами. Він визначає доступні місця для розміщення котів, умови відкриття нових ділянок дерева та оновлення стану ігрового середовища після переходу на новий рівень.

Для роботи з колекціями використовується CollectionManager. Після отримання або повного вирощування нового kota система автоматично оновлює відповідну колекцію та перераховує прогрес її заповнення.

Механіка декорування реалізована модулем DecorationManager. Він забезпечує розміщення декоративних об'єктів, зміну їх положення, видалення та збереження стану декору на дереві.

Особливістю застосунку є наявність системи шкідників. Для її реалізації використовується модуль PestManager, який контролює появу жуків, їх вплив на пасивний дохід та взаємодію користувача зі спеціальними інструментами боротьби зі шкідниками.

Для забезпечення збереження прогресу використовується SaveManager. Модуль виконує серіалізацію даних користувача та записує інформацію до локального сховища. Під час запуску застосунку цей модуль завантажує збережений стан гри та передає його іншим підсистемам.

Модуль SaveManager формує об'єкт SaveData, до складу якого входять баланс ігрової валюти, рівень дерева, відкриті зони, відомості про отриманих котів, стан колекцій, розміщені декоративні об'єкти, параметри шкідників та службовий час останнього збереження. Збереження виконується після ключових ігрових подій, зокрема після відкриття пака, купівлі об'єкта, розблокування нової зони, зміни налаштувань і завершення сеансу роботи користувача.

Під час запуску застосунку SaveManager перевіряє наявність файлу збереження. Якщо файл існує, дані завантажуються та передаються відповідним модулям: EconomyManager отримує баланс валюти, TreeZoneManager відновлює відкриті зони, CollectionManager оновлює колекції, а CatManager відновлює інформацію про котів. Якщо збереження відсутнє, система створює початковий стан гри.

Взаємодія користувача з грою здійснюється через UIManager, який відповідає за оновлення графічного інтерфейсу, обробку натискань кнопок, відображення інформаційних повідомлень, меню, поп-апів та інших елементів інтерфейсу [18].

Структуру каталогів проєкту Unity наведено на рис. 3.2.

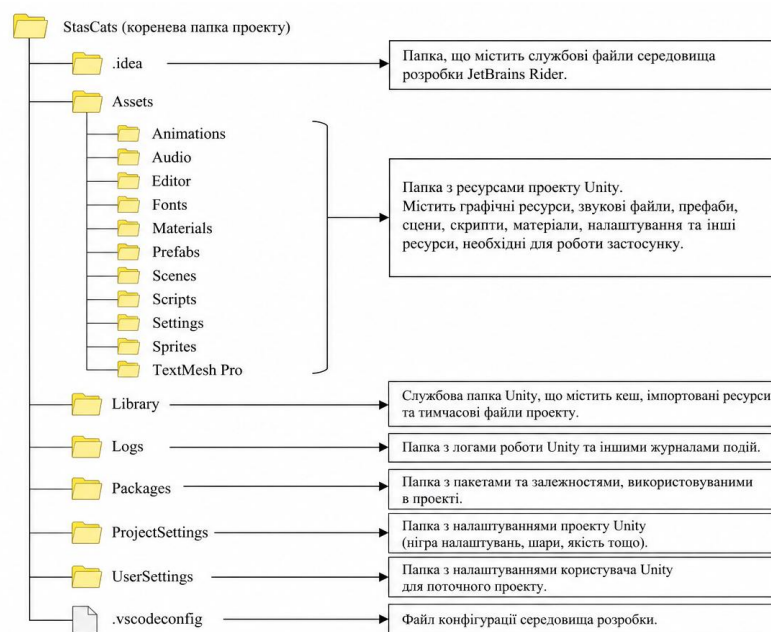


Рисунок 3.2 – Структура папок проєкту Unity

На рис. 3.2 наведено основні папки проєкту. У кореневій папці містяться службові каталоги Unity, зокрема Assets, Library, Logs, Packages, ProjectSettings та UserSettings. Основна частина ресурсів, що безпосередньо використовується у грі, розташована в папці Assets.

До основних каталогів проєкту належать Scripts, Prefabs, Sprites, ScriptableObjects, UI, Scenes та Audio. Scripts містить програмний код, Prefabs

— шаблони ігрових об'єктів, Sprites — графічні ресурси, ScriptableObjects — конфігураційні дані, UI — елементи інтерфейсу користувача, Scenes — ігрові сцени, Audio — звукові ресурси.

Завдяки такій структурі кожна підсистема може змінюватися незалежно від інших частин застосунку. Це спрощує підтримку, тестування та подальше розширення проєкту.

3.4 Керівництво користувача

Інтерфейс користувача застосунку Cat Tree розроблено з урахуванням простоти сприйняття та зручності взаємодії. Основні екрани застосунку призначені для запуску гри, перегляду ігрового середовища, керування котами, колекціями, декором та налаштуваннями.

Після запуску застосунку користувач потрапляє до початкового екрана. На ньому розміщено назву гри та кнопку початку гри. Початковий екран застосунку наведено на рис. 3.3.



Рисунок 3.3 – Початковий екран застосунку Cat Tree

Після переходу до гри користувач отримує доступ до основного ігрового середовища. Центральним елементом є дерево, на якому розміщуються коти,

декоративні об'єкти та додаткові інтерактивні елементи. У верхній частині інтерфейсу відображається поточна кількість ігрової валюти, а в нижній або боковій частині розташовані кнопки переходу до основних меню.

Для отримання нового kota користувач відкриває меню паків та вибирає доступний пак. Після відкриття система створює нового kota та додає його до доступних ігрових об'єктів. Отриманого kota можна розмістити на відкритій зоні дерева.

Для перегляду отриманих персонажів використовується меню колекцій. У ньому відображаються відкриті коти, їх рідкість та ступінь заповнення колекції. Вікно колекції котів наведено на рис. 3.4.



Рисунок 3.4 – Вікно колекції котів

Як видно з рис. 3.4, у вікні колекції відображаються комірки з котами, інформація про відкритих персонажів та загальний прогрес колекціонування. Такий інтерфейс дозволяє користувачу швидко оцінити поточний стан проходження та визначити, які коти ще не були відкриті.

У процесі гри коти автоматично генерують ігрову валюту у вигляді листків. Накопичені ресурси використовуються для відкриття нових зон дерева, придбання декору та розвитку ігрового середовища.

Доступ до декоративних елементів здійснюється через меню декору. Користувач може обрати об'єкт, встановити його на дерево або перемістити у нове місце. Після досягнення відповідного рівня стає доступною система шкідників, у межах якої користувач може застосовувати спеціальні інструменти для усунення жуків.

Для зміни параметрів застосунку використовується меню налаштувань. У ньому можна змінити параметри відображення, гучність звуку та інші допоміжні параметри. Усі зміни прогресу автоматично зберігаються системою та можуть бути відновлені після повторного запуску гри.

Після реалізації основних функціональних можливостей застосунку було проведено функціональне тестування. Метою тестування була перевірка коректності роботи ключових сценаріїв користувача: запуск гри, отримання котів, розміщення об'єктів на дереві, нарахування пасивного доходу, відкриття нових зон, збереження та завантаження прогресу.

Перевірка виконувалася за сценаріями, які відповідають основному ігровому циклу застосунку. Окремо перевірялися дії користувача в інтерфейсі, реакція системи на зміну ігрового стану та коректність оновлення даних після повторного запуску гри. Результати тестування наведено в табл. 3.4.

Таблиця 3.4 – Результати функціонального тестування застосунку

№	Сценарій тестування	Очікуваний результат	Результат
1	Запуск застосунку	Відкривається початковий екран гри	Успішно
2	Відкриття пака з котом	Система створює нового kota та додає його до списку доступних об'єктів	Успішно
3	Розміщення kota на дереві	Кіт розміщується на доступному	Успішно

		місці відкритої зони	
4	Нарахування пасивного доходу	Баланс листків збільшується відповідно до доходу активних котів	Успішно
5	Відкриття нової зони дерева	Після списання ресурсів зона стає доступною для користувача	Успішно
6	Оновлення колекції	Після отримання нового kota інформація відображається у вікні колекції	Успішно
7	Збереження прогресу	Дані про валюту, котів і відкриті зони записуються до локального сховища	Успішно
8	Завантаження прогресу	Після повторного запуску відновлюється попередній стан гри	Успішно
9	Поява шкідників	Жуки з'являються на відкритих зонах і впливають на пасивний дохід	Успішно

За результатами тестування встановлено, що основні функції застосунку працюють відповідно до очікуваної логіки. Критичних помилок, які унеможливають проходження основного ігрового циклу, не виявлено. Виявлені недоліки можуть бути усунуті в межах подальшого розвитку проекту та не впливають на працездатність базової версії застосунку.

Висновки до розділу 3

У третьому розділі було розглянуто програмне та технічне забезпечення застосунку Cat Tree. Визначено основні засоби розробки, серед яких Unity, мова програмування C#, Visual Studio, ScriptableObject та Git.

Проаналізовано вимоги до технічного й програмного забезпечення користувача та встановлено мінімальні й рекомендовані характеристики для запуску гри. Описано структуру програмної реалізації застосунку, основні програмні модулі, організацію Unity-проєкту та механізм локального збереження прогресу користувача.

Наведено керівництво користувача та виконано функціональне тестування основних сценаріїв роботи застосунку. Результати тестування підтверджують працездатність основних механік: запуску гри, отримання котів, розміщення об'єктів, нарахування пасивного доходу, відкриття зон, роботи колекцій, шкідників і системи збереження.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Організаційно-правові основи забезпечення безпеки праці

У сучасних умовах розвитку економіки одним із найважливіших ресурсів будь-якої держави та підприємства є людський капітал. Під цим поняттям розуміють знання, навички, досвід, професійну підготовку та працездатність працівників, які забезпечують виконання виробничих завдань і сприяють розвитку організації. Саме тому питання збереження здоров'я та працездатності персоналу набувають особливого значення.

Важливу роль у вирішенні цього завдання відіграє законодавство з охорони праці, зокрема, [1]. Воно визначає основні вимоги до організації безпечних умов праці, встановлює права працівників та обов'язки роботодавців, а також регламентує порядок управління безпекою праці на підприємствах. Наявність чітких нормативно-правових вимог дозволяє створити єдині правила, спрямовані на захист життя та здоров'я працівників незалежно від виду діяльності підприємства.

Одним із головних завдань законодавства у сфері охорони праці є запобігання виробничому травматизму та професійним захворюванням [1]. Для цього встановлюються вимоги до організації робочих місць, безпечної експлуатації обладнання, проведення навчання працівників, забезпечення належних санітарно-гігієнічних умов та здійснення контролю за станом безпеки праці. Виконання цих вимог сприяє зменшенню впливу небезпечних та шкідливих факторів виробничого середовища на працівників.

Законодавче регулювання охорони праці також спрямоване на підтримання працездатності персоналу протягом тривалого часу. Безпечні та комфортні умови праці дозволяють працівникам ефективніше виконувати свої обов'язки, зменшують рівень втоми та знижують ризик погіршення здоров'я [1, 2]. У результаті підприємство отримує можливість зберігати

кваліфікованих працівників та забезпечувати стабільність виробничих процесів.

Особливого значення питання охорони праці набувають у сферах діяльності, пов'язаних з інтенсивною розумовою працею та використанням комп'ютерної техніки. У таких умовах основними загрозами можуть бути надмірне навантаження на зір, тривале перебування у сидячому положенні, інформаційне перевантаження та психоемоційне напруження. Законодавство встановлює вимоги щодо організації робочих місць користувачів комп'ютерної техніки, режимів праці та відпочинку, освітлення, мікроклімату та інших факторів, які впливають на здоров'я працівників.

Не менш важливим є те, що законодавство створює умови для постійного вдосконалення системи охорони праці. Сучасні нормативні документи все більше орієнтуються на управління професійними ризиками, профілактику небезпечних подій та впровадження сучасних підходів до забезпечення безпеки праці. Це дозволяє роботодавцям своєчасно виявляти потенційні небезпеки та впроваджувати ефективні заходи щодо їх усунення.

4.2 Характеристика об'єкта та виявлення потенційних небезпек

Оскільки основною темою дипломної роботи є проектування та розробка ігрового застосунку, доцільно у розділі з охорони праці розглянути робоче місце розробника такого застосунку.

Розробник ігрового застосунку займається створенням програмного забезпечення, призначеного для організації ігрового процесу та взаємодії користувача з ігровим середовищем. Його робота охоплює всі основні етапи створення гри – від аналізу вимог та проектування до тестування готового продукту.

На початковому етапі розробник вивчає вимоги до майбутнього застосунку, визначає основні функції гри, продумує структуру програмного забезпечення та обирає програмні засоби для реалізації проєкту. Значна увага

приділяється проектуванню інтерфейсу користувача, логіки взаємодії між елементами гри та організації збереження даних.

Під час розробки працівник створює програмний код, який забезпечує роботу ігрових механік, управління персонажами, взаємодію об'єктів, обробку подій та виконання інших функцій. Також він інтегрує графічні елементи, анімацію, звукові ефекти та інші компоненти, необхідні для функціонування гри.

Важливою складовою роботи є перевірка працездатності застосунку. Для цього розробник багаторазово запускає програму, перевіряє правильність роботи окремих функцій, виявляє помилки та виконує їх усунення. У разі потреби здійснюється оптимізація програмного коду для покращення швидкодії та стабільності роботи гри.

Протягом реалізації цих етапів та складових робочого процесу працівник постійно використовує комп'ютерну техніку, працює з програмними середовищами розробки, графічними редакторами, засобами тестування та технічною документацією. Значна частина робочого часу пов'язана з аналізом інформації, прийняттям технічних рішень та тривалою концентрацією уваги.

Таблиця ..1 – Характеристика робочого місця розробника ігрового застосунку

Характеристика	Опис
Розташування робочого місця	Офісне приміщення або спеціалізований кабінет для роботи розробників програмного забезпечення
Тип приміщення	Закрите приміщення з постійним перебуванням працівників
Основне обладнання	Персональний комп'ютер або робоча станція з підвищеною продуктивністю
Засоби відображення інформації	Один або декілька моніторів для роботи з програмним кодом та графічними елементами

Засоби введення інформації	Клавіатура, комп'ютерна миша, графічний планшет (за потреби)
Програмне забезпечення	Середовище розробки, ігровий рушій, графічні редактори, засоби тестування
Засоби зв'язку	Локальна мережа, доступ до мережі Інтернет, корпоративні засоби комунікації
Робочі меблі	Письмовий стіл, офісне крісло, допоміжні полиці або тумби
Освітлення	Природне та штучне освітлення робочої зони
Мікроклімат	Опалення, вентиляція або кондиціонування повітря
Основний характер роботи	Інтелектуальна праця з використанням комп'ютерної техніки
Робоча поза	Переважно сидячи протягом більшої частини робочого дня
Об'єкти праці	Програмний код, графічні ресурси, технічна документація та результати тестування
Результат роботи	Створення, налагодження та тестування ігрового застосунку
Особливості трудової діяльності	Висока концентрація уваги, значне інформаційне навантаження та тривала робота за монітором

Враховуючи особливості організації робочого місця, можна визначити основні небезпеки, що впливають на розробника під час виконання завдань.

Таблиця ...2 – Основні небезпеки на робочому місці розробника та їх наслідки [3-5]

Потенційна небезпека	Джерело небезпеки	Можливі наслідки
----------------------	-------------------	------------------

Надмірне навантаження на зоровий апарат	Тривала робота з монітором, програмним кодом, графікою та інтерфейсами	Втома очей, погіршення зорового сприйняття інформації, головний біль
Тривале перебування у сидячому положенні	Виконання більшості робіт за комп'ютером без зміни робочої пози	Дискомфорт у спині, шиї та плечах, погіршення постави
Перевантаження кистей рук	Постійне використання клавіатури та комп'ютерної миші	Втома м'язів рук, неприємні відчуття у кистях та зап'ястях
Значне інформаційне навантаження	Аналіз програмного коду, пошук помилок, робота з документацією	Розумова перевтома, зниження уважності, збільшення кількості помилок
Психоемоційне напруження	Необхідність дотримання термінів розробки та усунення програмних помилок	Стрес, нервові виснаження, зниження працездатності
Недостатня освітленість робочого місця	Невдала організація системи освітлення або недостатня кількість світильників	Швидка втомлюваність, додаткове навантаження на органи зору
Відблиски на екрані монітора	Неправильне розташування монітора відносно вікон або джерел світла	Погіршення умов роботи та напруження зору

Незадовільний мікроклімат	Підвищена температура, сухе повітря або недостатня вентиляція приміщення	Погіршення самопочуття, сонливість, зниження концентрації уваги
Шумове навантаження	Робота комп'ютерної техніки, систем охолодження та офісного обладнання	Втома, дратівливість, відволікання від роботи
Ураження електричним струмом	Комп'ютерна техніка, периферійні пристрої та електромережа	Травмування працівника різного ступеня тяжкості
Виникнення пожежонебезпечної ситуації	Несправність електрообладнання, перевантаження мережі або коротке замикання	Загоряння обладнання, матеріальні збитки, небезпека для персоналу
Втрата даних унаслідок технічних збоїв	Відмова обладнання, програмні помилки або перебої електроживлення	Втрата результатів роботи, необхідність повторного виконання завдань
Спотикання та падіння	Кабелі живлення та комунікаційні дроти в зоні пересування	Забої, незначні механічні травми
Загальна перевтома організму	Тривала робота без достатніх перерв та відпочинку	Зниження продуктивності праці, погіршення самопочуття

Висока концентрація уваги протягом тривалого часу	Пошук та виправлення помилок у програмному коді, тестування гри	Психічна втома, зниження швидкості прийняття рішень
---	---	---

4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проектування та розробка заходів щодо їх попередження

Наявність значної кількості небезпек на робочому місці розробника потребує реагування та впровадження заходів із підвищення рівня безпеки та охорони праці. Проте реагування одразу на всі небезпеки потребує значних капіталовкладень від організації, в якій це робоче місце розміщено. Тому потрібно виділити ті небезпеки та загрози, які потребують реагування у першу чергу. Для цього потрібно проводити оцінювання ризиків ідентифікованих небезпек [6].

Під ризиком у сфері охорони праці зазвичай розуміють поєднання ймовірності виникнення небезпечної події та тяжкості її можливих наслідків. Чим вищою є ймовірність виникнення небезпеки та серйознішими можуть бути її наслідки, тим більшим вважається рівень ризику.

Процедура оцінювання ризиків [6] починається з аналізу робочого місця та виконуваних працівником обов'язків. На цьому етапі визначаються всі небезпечні та шкідливі фактори, які можуть впливати на людину під час роботи.

Після виявлення небезпек необхідно визначити можливі наслідки їх впливу. Для кожного фактора оцінюється, до чого він може призвести за певних умов. Наслідками можуть бути погіршення самопочуття, втома, зниження працездатності, травмування працівника або виникнення аварійної ситуації.

Наступним етапом є визначення ймовірності виникнення небезпечної події. При цьому враховується, наскільки часто працівник контактує з

небезпечним фактором, як довго триває його вплив та за яких умов він може проявитися.

На основі аналізу можливих наслідків та ймовірності виникнення небезпеки визначається рівень ризику. Отримані результати дозволяють встановити, які небезпеки потребують першочергової уваги та впровадження додаткових заходів безпеки.

Після завершення оцінювання розробляються заходи щодо зниження ризиків. Вони можуть передбачати покращення організації робочого місця, удосконалення умов праці, використання більш безпечного обладнання, проведення навчання працівників або впровадження додаткових організаційних заходів.

В якості прикладу реалізації процедури оцінювання ризиків проведемо оцінювання деяких із ідентифікованих небезпек за допомогою матриці – простого та широко розповсюдженого методу, який передбачає оцінювання ризиків за двома критеріями – вірогідність реалізації небезпеки та тяжкість наслідків [6].

Таблиця ...3 – Приклад реалізації матриці оцінювання ризиків

Небезпека	Категорія серйозності	Характеристика наслідків	Рівень ймовірності	Характеристика ймовірності	Індекс ризику	Рівень ризику
Надмірне навантаження на зоровий апарат	III – критична	Втома очей, головний біль, зниження працездатності	B – імовірна	Виникає регулярно під час роботи за монітором	3B	Небажаний

Тривале перебування у сидячому положенні	III – критична	Дискомфорт у спині та шиї, погіршення постави	B – імовірна	Спостерігається протягом більшої частини робочого дня	3В	Небажаний
Значне інформаційне навантаження	III – критична	Розумова перевтомленість, зниження концентрації уваги	B – імовірна	Супроводжує виконання основних трудових обов'язків	3В	Небажаний
Ураження електричним струмом	II – значна	Травмування працівників різного ступеня тяжкості	D – малоімовірна	Можливе у разі несправності обладнання або електромережі	2Д	Допустимий
Виникнення пожежонебезпечної ситуації	I – катастрофічна	Травмування людей, пошкодження майна та обладнання	D – малоімовірна	Може виникнути внаслідок короткого замикання або несправності	1Д	Допустимий

				електрообла днання		
--	--	--	--	-----------------------	--	--

На основі отриманих оцінок ризиків планується черговість розробки та впровадження рекомендацій із покращення умов праці та підвищення рівня безпеки. В загальному вигляді рекомендації щодо аналізованих небезпек представляють собою наступне [7, 8]:

- для зменшення навантаження на очі необхідно забезпечити достатнє освітлення робочого місця та правильно налаштувати яскравість монітора. Під час роботи доцільно регулярно робити короткі перерви та періодично переводити погляд на віддалені предмети;

- робоче місце слід обладнати зручним кріслом, яке забезпечує підтримку спини. Рекомендується періодично вставати з робочого місця, змінювати положення тіла та виконувати нескладні вправи для розминки;

- робочі завдання бажано рівномірно розподіляти протягом дня, чергуючи складні та менш складні види робіт. Для підтримання працездатності необхідно дотримуватися режиму праці та відпочинку й уникати тривалої безперервної роботи;

- необхідно використовувати лише справне комп'ютерне обладнання та своєчасно перевіряти стан кабелів і електричних з'єднань. У разі виявлення пошкоджень експлуатацію обладнання слід припинити до усунення несправності.

- слід контролювати технічний стан електромережі та не допускати перевантаження розеток і подовжувачів. Також необхідно підтримувати вільний доступ до первинних засобів пожежогасіння та дотримуватися встановлених правил пожежної безпеки.

Висновки до розділу 4

У розділі розглянуто питання організації охорони праці на робочому місці розробника ігрового застосунку. Проаналізовано роль законодавства з охорони праці у збереженні здоров'я працівників та підтриманні належного рівня безпеки під час виконання професійних обов'язків.

Виконано характеристику трудової діяльності розробника ігрового застосунку та робочого місця. Визначено особливості умов праці, пов'язані з тривалою роботою за комп'ютером, використанням спеціалізованого програмного забезпечення та значним інформаційним навантаженням.

На основі аналізу робочого середовища було виявлено основні небезпечні та шкідливі фактори, які можуть впливати на працівника під час виконання роботи. Проведено оцінювання професійних ризиків для найбільш характерних небезпек, визначено рівень їх серйозності та ймовірність виникнення.

Для зниження рівня професійних ризиків запропоновано комплекс організаційних та технічних заходів, спрямованих на покращення умов праці, зменшення навантаження на працівника, підвищення рівня електробезпеки та пожежної безпеки.

ВИСНОВКИ

У кваліфікаційній роботі виконано проектування та розробку ігрового застосунку Cat Tree жанру idle-game. У процесі виконання роботи було проаналізовано предметну область, розглянуто наявні аналоги, визначено функціональні вимоги до системи, спроектовано структуру даних та програмні модулі, а також описано основні ігрові механіки: отримання і розміщення котів, накопичення листків, відкриття нових зон, роботу колекцій, декору та системи шкідників.

У роботі побудовано UML-діаграми діяльності користувача, варіантів використання та класів, визначено вхідні й вихідні дані, розроблено інформаційну модель предметної області та описано математичні залежності для розрахунку пасивного доходу, бонусів, штрафів і офлайн-нарахування ресурсів.

Програмний застосунок реалізовано в середовищі Unity мовою програмування C#. У межах реалізації використано модульну структуру з окремими менеджерами для роботи з котами, економікою, зонами дерева, колекціями, декором, шкідниками, інтерфейсом та збереженням прогресу. Такий підхід забезпечує зрозумілу організацію проекту та створює можливість подальшого розширення функціональності.

Проведене функціональне тестування підтвердило коректну роботу основних сценаріїв використання застосунку. Отримані результати свідчать про досягнення поставленої мети та практичну придатність розробленого застосунку як основи для подальшого розвитку ігрового проекту Cat Tree.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Newzoo. Global Games Market Report 2024 [Електронний ресурс]. – Режим доступу: <https://newzoo.com/resources/trend-reports/newzoos-global-games-market-report-2024-free-version>
2. Statista. Games – Worldwide: market forecast [Електронний ресурс]. – Режим доступу: <https://www.statista.com/outlook/amo/media/games/worldwide/>
3. Unity Technologies. Unity Engine: 2D & 3D development platform [Електронний ресурс]. – Режим доступу: <https://unity.com/products/unity-engine>
4. Incremental game [Електронний ресурс] // Wikipedia. – Режим доступу: https://en.wikipedia.org/wiki/Incremental_game
5. Chillquarium on Steam [Електронний ресурс]. – Режим доступу: <https://store.steampowered.com/app/2276930/Chillquarium/>
6. Cookie Clicker on Steam [Електронний ресурс]. – Режим доступу: https://store.steampowered.com/app/1454400/Cookie_Clicker/
7. AdVenture Capitalist on Steam [Електронний ресурс]. – Режим доступу: https://store.steampowered.com/app/346900/AdVenture_Capitalist/
8. Rusty's Retirement on Steam [Електронний ресурс]. – Режим доступу: https://store.steampowered.com/app/2666510/Rustys_Retirement/
9. Cat Tree on Steam [Електронний ресурс]. – Режим доступу: https://store.steampowered.com/app/3198400/Cat_Tree/
10. Microsoft. C# documentation [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/>
11. Microsoft. Visual Studio documentation [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/visualstudio/>
12. Unity Technologies. ScriptableObject manual [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/class-ScriptableObject.html>
13. Unity Technologies. Serialization manual [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/script-Serialization.html>

14. PlantUML. Language reference guide [Електронний ресурс]. – Режим доступу: <https://plantuml.com/guide>
17. Git Documentation [Електронний ресурс]. – Режим доступу: <https://git-scm.com/doc>
18. Unity Technologies. User interface documentation [Електронний ресурс]. – Режим доступу: <https://docs.unity3d.com/Manual/UIToolkits.html>
19. Закон України «Про охорону праці». URL: <https://zakon.rada.gov.ua/laws/show/2694-12>.
20. Safety and health at work. URL: <https://www.ilo.org/topics-and-sectors/safety-and-health-work>.
21. OSHA Computer Workstations eTool. URL: <https://www.osha.gov/etools/computer-workstations>.
22. Ergonomics in Office Work (EU-OSHA). URL: <https://oshwiki.osha.europa.eu/en/themes/ergonomics-office-work>.
23. Робота за комп'ютером: вимоги охорони праці у 2026 році. URL: <https://7eminar.ua/news/18368-robota-za-kompyuterom-vimogi-oxoroni-praci-u-2026-roci>.
24. EU-OSHA. Occupational Safety and Health Risk Assessment Methodologies. URL: <https://oshwiki.osha.europa.eu/en/themes/occupational-safety-and-health-risk-assessment-methodologies>.
25. Правила роботи за комп'ютерами в Україні: основні обов'язки роботодавця. URL: <https://oppb.com.ua/news/pravyla-roboty-za-komp-yuteramy-v-ukrayini-osnovni-obov-yazky-robotodavtsya>.
26. 6 правил роботи за комп'ютером без шкоди для здоров'я. – Режим доступу: <https://pon.org.ua/novyny/5812-6-pravil-roboti-za-kompyuterom-bez-shkodi-dlya-zdorovya.html>.