

Міністерство освіти і науки України
Одеський національний технологічний університет
Кафедра комп'ютерної інженерії



**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему Розробка гри в жанрі RPG за допомогою
(назва кваліфікаційної роботи згідно наказу ОНТУ)
платформи Unity

Здобувача Костишена Д.М.
(прізвище, ініціали)
4 курсу 543б групи

Керівники: ст. викл. Слушина Н.В.
(посада, прізвище та ініціали)
д.т.н., проф. Артеменко С.В.
(посада, прізвище та ініціали)

Консультанти: _____
(посада, прізвище та ініціали)
д.е.н, проф. Басюркіна Н.Й.
(посада, прізвище та ініціали)

Кваліфікаційна робота допускається до захисту

Рішення кафедри від 10.06 2023 р., протокол № 8
Завідувач кафедри комп. інженерії _____ Сергій АРТЕМЕНКО
(назва кафедри) (підпис) (Ім'я ПРІЗВИЩЕ)

Одеса – 2023 рік

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерної інженерії, програмування та кіберзахисту
Кафедра комп'ютерної інженерії
Ступінь вищої освіти бакалавр
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма Розробка ігор та інтерактивних медіа у віртуальній реальності

ЗАТВЕРДЖУЮ

Зав. кафедри комп'ютерної інженерії

Сергій АРТЕМЕНКО

« 10 » серпня 2022 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Костишена Дмитра Миколайовича

1. Тема роботи Розробка гри в жанрі RPG за допомогою платформи Unity

Затверджена наказом університету від « 10 » квітня 2023 р., наказ № 147-03

2 Термін здачі здобувачем закінченої роботи 5 червня 2023 р.

3. Вихідні дані роботи

1. Середовище програмування «Visual Studio». 2. Двигун для розробки двовимірних та тривимірних додатків та ігор «Unity». 3. Ассети персонажів.

4. Перелік питань, які потрібно розробити

1. Вступ 2. Дослідження предметної області 3. Технічне завдання.

4. Обґрунтування засобів реалізації. 5. Опис етапів реалізації.

6. Опис програмного продукту. 7. Оцінка ефективності проекту.

8. Охорона праці. 9. Загальні висновки.

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

Слайд 2. Мета дипломної роботи. Слайд 3. Розглянуті аналоги програмного продукту.

Слайд 4. Задачі проектування. Слайд 5. Засоби реалізації. Слайд 6. Структура проекту.

Слайд 7. Головне меню гри. Слайд 8. Бойова система. Слайд 9. Ключові персонажі.

Слайд 10. Ворожі персонажі. Слайд 11. Ефективність проекту. Слайд 12. Висновок.

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Економіка	Басюркіна Н.Й., д.е.н., проф.		
Охорона праці	Артеменко С. В., д.т.н., проф.		
Нормоконтроль	Жуковецька С. Л., ст. викл.		

7. Дата видачі завдання 30.09.2022

Керівники

Сергій АРТЕМЕНКО

Наталя СЛУШНА

Завдання прийняв до виконання

Дмитро КОСТИШЕН

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Дослідження предметної області.	15.02.2023	
2.	Аналіз існуючих аналогів .	01.03.2023	
3.	Постановка технічного завдання.	11.03.2023	
4.	Обґрунтування засобів реалізації.	23.03.2023	
5.	Проектування.	04.04.2023	
6.	Реалізація.	26.04.2023	
7.	Тестування.	07.05.2023	
8.	Економічні розрахунки.	18.05.2023	
9.	Охорона праці.	25.05.2023	
10.	Оформлення пояснювальної записки.	04.06.2023	

Керівники роботи

Сергій АРТЕМЕНКО

Наталя СЛУШНА

Несу відповідальність за ідентичність електронного та друкованого варіантів кваліфікаційної роботи, даю згоду на обробку персональних даних та не заперечую проти розміщення кваліфікаційної роботи на офіційних web-ресурсах ОНТУ.

Підтверджую, що в кваліфікаційній роботі відсутні порушення норм академічної доброчесності.

Здобувач – дипломник

Дмитро КОСТИШЕН

АНОТОЦІЯ

Ця дипломна робота присвячена розробці гри в жанрі рольових ігор (*RPG*). Метою роботи є створення захопливого та неповторного ігрового всесвіту, у який гравці зможуть зануритися та випробувати захопливі пригоди.

У межах роботи буде проведено аналіз наявних *RPG*-ігор, вивчено особливості ігрового процесу, елементи ігрового дизайну та механіки, а також розглянуто вимоги та вподобання цільової аудиторії. На основі отриманих даних буде розроблено концепцію гри, що охоплює сюжетну лінію, персонажів, ігрові механіки та системи прогресії.

У процесі розробки будуть використані сучасні інструменти та технології, такі як ігрові рушії, програмування і візуалізація. При створенні гри буде приділено увагу балансу складності, естетиці ігрового світу і взаємодії ігрових елементів.

Остаточна гра буде піддана тестуванню та оптимізації для забезпечення плавного і якісного ігрового досвіду. Також буде проведено оцінку ефективності проекту.

Результати роботи буде представлено у вигляді готової гри в жанрі *RPG*, а також документації, що описує процес розробки, використані технології та отримані результати.

Ключові слова: *ігри жанру “RPG”, ігровий двигун “Unity”, ігровий всесвіт.*

ABSTRACT

This thesis is devoted to the development of a game in the genre of role-playing games (RPG). The goal of the work is to create an exciting and unique gaming universe in which players can immerse themselves and experience exciting adventures.

Within the framework of the work, an analysis of existing RPG games will be carried out, features of the gameplay, elements of game design and mechanics will be studied, as well as the requirements and preferences of the target audience will be considered. Based on the data received, a game concept will be developed, covering the storyline, characters, game mechanics and progression systems.

Modern tools and technologies, such as game engines, programming and visualization, will be used in the development process. When creating the game, attention will be paid to the balance of complexity, the aesthetics of the game world, and the interaction of game elements.

The final game will be tested and optimized to ensure a smooth and quality gaming experience. An evaluation of the project's effectiveness will also be conducted.

The results of the work will be presented in the form of a finished game in the RPG genre, as well as documentation describing the development process, the technologies used and the results obtained.

Keywords: "RPG" genre games, "Unity" game engine, game universe..

3MICT

стоп.

ВСТУП	8
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	13
1.1 Опис предметної області	13
1.2 Аналіз існуючих аналогів	14
1.2.1 <i>The Elder Scrolls</i>	14
1.2.2 <i>Fallout</i>	15
1.2.3 <i>Dark Souls</i>	16
1.3 Постановка задачі	17
РОЗДІЛ 2 ТЕХНІЧНЕ ЗАВДАННЯ.....	18
2.1 Вступ.....	18
2.2 Підстави для розробки	19
2.3 Призначення розробки.....	19
2.4 Вимоги до програми або програмного продукту	19
2.5 Вимоги до програмної документації.....	21
2.6 Техніко економічні показники	21
2.7 Стадії й етапи розробки	21
2.8 Порядок контролю й приймання	22
РОЗДІЛ 3 ОБГРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ.....	23
РОЗДІЛ 4 ОПИС ЕТАПІВ РЕАЛІЗАЦІЇ.....	26
4.1 <i>DFD</i> –діаграма.....	26
4.2 <i>SADT</i> модель.....	30
4.3 <i>IDEF3</i>	35
4.4 <i>Use-Case</i> діаграма.....	38
4.5 Проектування бази даних	42

					КРБ.КІ.1.147–03.3.6					
Змн.	Арк.	№ докум.	Підпис	Дат	Розробка гри в жанрі RPG за допомогою платформи Unity			Лім.	Арк.	Аркушів
Розроб.		Дмитро КОСТИШЕН								
Перевір.		Сергій АРТЕМЕНКО							6	92
Рецензент		Марина БУЛЮК						гр. КІ–5436, ОНТУ		
Н. контр.		Світлана ЖУКОВЕЦЬКА								
Затверд.		Сергій АРТЕМЕНКО								

РОЗДІЛ 5 ОПИС ПРОГРАМНОГО ПРОДУКТУ	44
РОЗДІЛ 6 ОЦІНКА ЕФЕКТИВНОСТІ ПРОЕКТУ	58
РОЗДІЛ 7 ОХОРОНА ПРАЦІ	69
7.1 Фактори що впливають на працю	69
7.2 Загальні вимоги безпеки праці з комп'ютерною технікою	73
7.3 Вимоги до організації робочого місця	74
ЗАГАЛЬНІ ВИСНОВКИ	76
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	77
ДОДАТКИ.....	79
Додаток А Програмний код.....	79
Додаток Б Графічний матеріал	87

					КРБ.КІ.1.440–03.3.12	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Комп'ютер – примітивна річ, без якої, мабуть, не обійдеться жодна людина. Всього за кілька десятиріч комп'ютерні технології розвинулися до неможливих меж. І навіть зараз їх розвиток не перестає дивувати. З кожним роком ми можемо побачити, як на полицях магазинів з'являються нові товари, які тісно зв'язані з комп'ютером, а й також являються їхніми невід'ємними частинами, без яких комп'ютер не зможе працювати. Такий швидкий розвиток зумовлений високою комп'ютеризацією. Складно уявити сучасну велику компанію або невелику фірму, які обходяться без комп'ютерів і комп'ютерних мереж. Адже момент винаходу комп'ютерної технології став переломним в житті людства, що спричинило великий вплив у розвиток промисловості, підвищило ефективність військової техніки, внесло безліч змін в роботу засобів масової інформації. Комп'ютери стрімко зайняли важливу роль у сучасних людей. З ними ми взаємодіємо майже постійно – на роботі, вдома, в машині чи літаку.

Разом з появою комп'ютерів з'явилися комп'ютерні ігри, які відразу ж здобули велику популярність. Розвиток і вдосконалення комп'ютерних ігор також тісно зв'язані з розвитком комп'ютерних технологій, що дало змогу багатьом людям створювати власні ігри з характерними рисами в кожній з них. Таким чином комп'ютерні ігри стали поділятися на ігрові жанри, яких на сьогоднішній день далеко за один десяток. Одним з таких жанрів є *RPG*.

RPG (RolePlayGame) – жанр комп'ютерних ігор, заснований на елементах ігрового процесу традиційних настільних рольових ігор. У рольовій грі гравець управляє одним або декількома персонажами, кожен з яких описаний набором чисельних характеристик, списком здібностей і вмінь; прикладами таких характеристик можуть бути кількість здоров'я (*hitpoints, HP*), показники сили, спритності, інтелекту, захисту, ухилення, рівень розвитку того чи іншого навичку і т.п

					КРБ.КІ.1.147–03.3.6	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дат		

Жанр рольових ігор (*RPG*) у комп'ютерних іграх відрізняється своїм фокусом на розвитку персонажа та прогресії ігрового світу. Він надає гравцеві можливість приймати рішення від імені персонажа та формувати його історію, а також взаємодіяти з віртуальним світом, виконувати завдання, боротися з ворогами та розвивати свої навички.

У *RPG*-іграх гравець зазвичай контролює одного або кількох персонажів, які можуть бути визначені розробниками гри або створені самим гравцем.

Ключовими елементами жанру є:

1. Прогресія персонажа. Гравець розвиває свого персонажа, підвищуючи його рівень, покращуючи навички, набуваючи нового обладнання та зброї. Прогресія персонажа може ґрунтуватися на накопиченні досвіду, виконанні завдань чи досягненні певних цілей.
2. Система завдань та сюжет. *RPG*-ігри зазвичай мають розгалужені сюжети та безліч завдань, які гравець може виконувати. Завдання можуть змінюватись від основних сюжетних місій до побічних квестів, які допомагають розкрити світ гри та розвинути персонажа.
3. Бойова система. *RPG*-ігри часто містять тактичну бойову систему, де гравець може керувати персонажами у боях із застосуванням різних навичок, заклинань та тактик. Бойові битви можуть бути в реальному часі або покроковими.
4. Дослідження та інтерактивність. *RPG*-ігри зазвичай пропонують великі ігрові світи, які гравець може досліджувати. У цих світах можуть бути міста, підземелля, печери, ліси та інші локації, кожна з яких може мати свої особливості та завдання. Взаємодія з ігровими персонажами (*NPC*) та іншими гравцями може бути важливою частиною ігрового процесу.

					КРБ.КІ.1.147–03.3.6	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дат		

5. Характеристики та класи. *RPG*-ігри часто мають системи характеристик та класів, які визначають спеціалізацію та роль персонажа в грі.
6. Вибір та наслідки. *RPG*-ігри часто надають гравцеві можливість приймати рішення, які впливають на хід гри та розвиток сюжету. Ці рішення можуть мати моральні, етичні чи практичні наслідки, які можуть вплинути на ігровий світ та взаємодію з іншими персонажами.
7. Спорядження та різноманітність предметів. *RPG*-ігри часто пропонують широкий асортимент обладнання, зброї, броні та предметів, які можуть бути знайдені, придбані або створені у грі. Це дозволяє гравцеві налаштовувати персонажа під свій стиль гри та підвищувати його силу та ефективність.
8. Розрахований на багато користувачів режим. Деякі *RPG*-ігри пропонують можливість грати разом з іншими гравцями в онлайн-режимі. Це дозволяє взаємодіяти з іншими гравцями, спільно виконувати завдання, битися в *PvP*-боях або створювати гільдії та спільноти.
9. Додатковий контент і тривалість. Багато *RPG*-ігор пропонують додатковий контент у вигляді завантажених доповнень (*DLC*), що дозволяють розширити ігровий світ або надати нові завдання та пригоди. Також багато *RPG*-ігор мають тривалу продовження, дозволяючи гравцям проводити в них безліч годин, розвиваючи своїх персонажів і досліджуючи світ гри.

Об'єктом дослідження даної дипломної роботи є інформаційні процеси, пов'язані зі створенням власної комп'ютерної гри

Темою даної дипломної роботи є «Розробка гри в жанрі *RPG* за допомогою платформи *Unity*».

Новизна дипломної роботи: унікальні ігрові механіки та оригінальний підхід до сюжету.

					КРБ.КІ.1.147–03.3.6	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дат		

Метою роботи є розробка гри у жанрі *RPG* на движку *Unity*, яка має реалізовувати та містити в собі такі функції:

1. Зручний інтерфейс користувача.
2. Захоплюючий сюжет.
3. Реалізація бойового процесу.
4. Отримання бойового опиту та підвищення рівня персонажа.
5. Прокачка навиків.
6. Взаємодія з іншими персонажами.
7. Можливість взаємодії з предметами на локаціях.

Дипломна робота складається з 7 розділів, кожен з яких описує розроблений програмний продукт з різних сторін.

У першому розділі наведено основні відомості з роботи ігри, приведений аналіз аналогів ігрового продукту з подібними функціями та надано їх характеристики. Визначено характеристики, що вигідно відрізняють програмний продукт, що розробляється, від проаналізованих аналогів. Описана постановка задачі, вимоги до створеного програмного продукту.

У другому розділі складене технічне завдання на створення програмного продукту. Технічне завдання описує вимоги до ПП, що розробляється; визначає склад та зміст робіт зі створення ПП. Розробка технічного завдання – один з основних етапів розробки системи, її основний документ.

У третьому розділі описані засоби реалізації програмного продукту. Використана база даних та середовище програмування.

У четвертому розділі приведено модель системи з допомогою *UML*-нотації: приведено загальні відомості, діаграму класів (структура системи), діаграму схем станів та діаграму прецедентів (поведінка програмної системи з точки зору користувача програмного продукту). Далі визначаються: мова

					КРБ.КІ.1.147–03.3.6	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дат		

програмування інтерфейсу програмного продукту та система керування базами даних, за допомогою якої буде створена база даних програмного продукту.

П'ятий розділ присвячений всебічному аналізу створеного програмного продукту: приводяться результати роботи системи, виконується порівняльна характеристика програмного продукту – наводяться переваги та недоліки в роботі системи. Приведено інструкцію користувача з описанням основних функцій та приведенням інтерфейсу системи з допомогою рисунків.

В шостому розділі приведено економічне обґрунтування дипломної роботи, виконані розрахунки основних економічних показників впровадження програмного продукту, зроблений висновок про доцільність використання програмного продукту.

Сьомий розділ присвячений основним вимогам з охорони праці та техніки безпеки для користувачів персональних комп'ютерів при використанні програмного продукту.

Описання та програмний код основних функцій програмного продукту наведено у додатку А. Додаток Б містить друковану презентацію.

					КРБ.КІ.1.147–03.3.6	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Розробка комп'ютерних ігор – процес створення комп'ютерних ігор (відеоігор). Розробкою ігор займаються розробники, яких може бути від десятка до одного. Звісно великомасштабну гру, як правило, розробляють цілі команди, що спеціалізуються на цьому процесі, яких, зазвичай фінансує інша компанія, так звана “Компанія-видавець”, яка при закінченні розробки займається виданням гри та витрати на розробку. Рідше команди розробників можуть розробляти ігри за свої кошти і поширювати їх самими (Інді-ігри).

Інді-ігри – відеоігри, створені незалежно від фінансової підтримки великих видавців. Часто ці ігри дешеві або безкоштовні, багато незалежних ігор мають невеликий розмір і тому поширюються через інтернет за допомогою цифрової дистрибуції або як *freeware*.

Розробка найбільш крупнобюджетних ігор («ААА-гри») може коштувати десятки мільйонів доларів США, причому протягом останніх десятиліть ці бюджети безперервно росли, як і чисельність команд розробників і терміни розробки. Середній бюджет ААА-проекту, що випускається найбільшими компаніями-видавцями – коливається від 18 до 24 млн. доларів.

Розробляють комп'ютерні ігри на спеціальних ігрових двигунах (Ігровий рушій) – базове програмне забезпечення, на основі якого будуються всі інші складові гри, які можна змінювати за бажанням.

					КРБ.КІ.1.147–03.3.6	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дат		

Вперше про них почали писати ще в 90-их роках з появою багатьох ігор, схожих на культовий шутер 1993 року – *Doom*. З того часу розробилось все більше ігрових двигунів, деякі з яких попали у вільний доступ, що дало змогу розробляти власні ігри звичайним користувачам.

Кожен двигун дає можливість варіації, тобто на одному двигуні можуть бути розроблені декілька різних ігор за допомогою безлічі елементів гри. Самі двигуни відрізняються один від одного ігровою логікою, фізикою об'єктів, правилами прорисовки об'єктів та геймплеєм.

Що до меж двигунів, зазвичай вони дозволяють створювати гру одного єдиного жанру. Наприклад один и той же движок не може використовуватися для створення стратегій та шутерів одночасно.

1.2 Аналіз існуючих аналогів

1.2.1 *The Elder Scrolls*

The Elder Scrolls (рис. 1.1) – фентезійна рольова серія, відома своїми величезними відкритими світами, багатими знаннями та захоплюючим геймплеєм. Гравці можуть досліджувати різноманітні ландшафти, брати участь у епічних квестах, взаємодіяти з унікальними персонажами та формувати власну історію, орієнтуючись у динамічному та захоплюючому фантастичному світі.



Рис. 1.1 – Банер гри *The Elder Scrolls*

					КРБ.КІ.1.147–03.3.6	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дат		

Плюси:

- цікавий сюжет;
- велика серія ігор;
- велика кількість локацій.

Мінуси:

- баги та технічні проблеми;
- монотонність завдань.

1.2.2 *Fallout*

Fallout (рис. 1.2) — це пост апокаліптична рольова серія, відома своєю похмурою та суворою атмосферою, захоплюючим оповіданням і вибором, керованим гравцем. Дія відбувається в ретро-футуристичному світі, спустошеному ядерною війною, гравці пересуваються через небезпечні пустки, стикаються з унікальними фракціями, беруть участь у тактичних боях і приймають рішення, які впливають на долю пустки та її мешканців.



Рис. 1.2 – Банер гри *Fallout*

Плюси:

- велика серія ігор;
- легка в освоєнні;
- велика кількість локацій.

					КРБ.КІ.1.147–03.3.6	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дат		

Мінуси:

- штучний інтелект та компаньйони;
- технічний інтерфейс.

1.2.3 *Dark Souls*

Dark Souls (рис. 1.3) — складна рольова *RPG*-гра, відома своєю надзвичайною складністю, атмосферним світом і глибокими методичними боями. Гравці вирушають у темний та взаємопов'язаний фантастичний світ, борючись із різними ворогами, розгадуючи заплутану історію та долаючи напружені битви з босами, створюючи напружений та корисний досвід для тих, хто прагне справжнього випробування майстерності та наполегливості.



Рис. 1.3 – Банер гри *Dark Souls*

Плюси:

- різноманітність ворогів;
- різновид зброї та обладнання.

Мінуси:

- неясність та обмеженість сюжету;
- складна для проходження.

					КРБ.КІ.1.147–03.3.6	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дат		

Порівняльна характеристика аналогів

<i>Характеристики</i>	<i>The Elder Scrolls</i>	<i>Fallout</i>	<i>Dark Souls</i>
Графіка	+	+	—
Стилістика	+	+	+
Інтерфейс	+	—	—
Нелінійність	—	—	+
Штучний інтелект	+	—	+
Технічне налагодження	—	+	+

1.3 Постановка задачі

Для досягнення поставленої мети необхідно виконати наступні дії:

1. Дослідити предметну область.
2. Знайти аналоги програмного продукту.
3. Вибрати наступні компоненти:
 - середу розробки гри;
 - середу програмування;
4. Розробити програмну оболонку, що міститиме:
 - можливість зберігання всієї інформації про об'єкти в файлах;
 - ігровий інтерфейс;
 - можливість зручного управління та гри;
5. Перевірити працездатність системи і при необхідності зробити налагодження.
6. Розробити інструкцію для користувача.

РОЗДІЛ 2

ТЕХНІЧНЕ ЗАВДАННЯ

Розглянувши існуючі аналоги, була поставлена наступна мета роботи – розробити комп’ютерну гру. Програмний продукт має відповідати таким вимогам:

- захоплюючий сюжет;
- зручний та інтуїтивно зрозумілий інтерфейс;
- довідкова інформація;
- проходження сюжетної лінії;
- перегляд даних пов’язаних з опонентами головного героя;
- перегляд даних пов’язаних з головним героєм;
- можливість торгівлі;
- редагування характеристик головного героя;
- вибір класу.

Користувачем гри може бути людина з елементарним рівнем користування ПК.

2.1 Вступ

1. Повне найменування системи і її умовне позначення. Розробка гри в жанрі *RPG* за допомогою платформи *Unity*.
2. Перелік об’єктів, на яких передбачається використання системи. Система може використовуватися на персональних комп’ютерах користувачів.

					КРБ.КІ.1.147–03.3.6	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дат		

2.2 Підстави для розробки

1. Розробник системи Костишен Дмитро Миколайович.
2. Перелік документів, на підставі яких створюється інформаційна система. Повне завдання дипломного керівника.
3. Порядок оформлення і пред'явлення замовникові результатів робіт із створення системи, її частин і окремих коштів. Виконання завдань та представлення результатів керівнику у встановлений термін часу для перевірки та внесення змін, при необхідності.

2.3 Призначення розробки

Розробка гри в жанрі *RPG* за допомогою платформи *Unity*.

2.4 Вимоги до програми або програмного продукту

1. Вимоги до функціональних характеристик:
 - перелік завдань, що підлягають автоматизації. Збереження даних у БД, створення персонажу користувача та розробка ігрового процесу;
 - вимоги до якості реалізації кожної функції, до форми представлення вихідної інформації, характеристики точності, достовірності інформації, що виводиться. Результати виконання запитів гри виводяться у вигляді ігрового інтерфейсу та ігрового процесу;
2. Вимоги до надійності:
 - вимоги до надійності, безпеки, ергономіки, транспортабельності, експлуатації, технічного обслуговування, ремонту, захисту інформації від несанкціонованого доступу, захисту від впливу зовнішніх дій. Технічне обслуговування майже не потрібне;

					КРБ.КІ.1.147–03.3.6	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дат		

3. Умови експлуатації:

- характеристики навколишнього середовища. Навколишнє середовище не повинно бути вологим і температура не повинна знижуватися нижче 10°C;

4. Вимоги до складу й параметрів технічних засобів:

- система працює на ПК пристроях;

5. Вимоги до інформаційної й програмної сумісності:

- інформаційне забезпечення (склад, структура і організація даних, обмін даними між компонентами системи, інформаційна сумісність з суміжними системами, класифікатори, СУБД, контроль даних і введення інформаційних масивів, процедури додання, що використовуються). Інформація вся зберігається в базі даних (у таблицях), обмін даних здійснюється за допомогою зв'язків та підключенням бази даних у системі;
- лінгвістичному (мови програмування, мови взаємодії користувачів з системою, системи кодування, мови введення-виводу). Мова програмування *C Sharp*. Мова взаємодії користувача з системою українська;
- програмне забезпечення (незалежність програмних засобів від платформи, якість програмних засобів і способи його контролю, використання фондів алгоритмів і програм). Програмний продукт працює на ПК пристроях, але для його коректної роботи рекомендується використовувати операційну систему *Windows*;
- організаційне забезпечення (структура і функції експлуатації підрозділів, захист від помилкових дій персоналу). Програмний продукт розрахований на експлуатацію одного користувача;
- методичне забезпечення (склад нормативно-технічної документації) є повне керівництво по роботі з програмою;

					КРБ.КІ.1.147–03.3.6	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дат		

- математичне забезпечення (склад і область вживання математичних моделей і методів, типових і розроблюваних алгоритмів). Використовуються математичні моделі при обчисленні статистичних показників;
- 6. Вимоги до маркування й упакуванню до даного програмного забезпечення не встановлені.
- 7. Вимоги до транспортування й зберіганню не вказані.

2.5 Вимоги до програмної документації

1. Перелік належних розробці документів. Технічне завдання, пояснювальна записка, презентація.
2. Перелік документів на машинних носіях. Код програми, пояснювальна записка, презентація.
3. Вигляд і порядок експертизи технічної документації. Документація оформлена відповідно до нормо контролю і Держстандарту.

2.6 Техніко економічні показники

1. Найменування і необхідні значення технічних, технологічних, виробничо-економічних та інших показників об'єкту, які повинні бути досягнуті при впровадженні ІС економічний ефект від впровадження.

2.7 Стадії й етапи розробки

1. Перелік стадій і етапів робіт. Аналіз, проектування, реалізація, впровадження, супровід.

					КРБ.КІ.1.147–03.3.6	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дат		

2. Склад організацій виконавців робіт на стадії аналізу, а також на всіх початкових стадіях створення системи брали участь розробник і дипломний керівник проекту, далі до розробки приєдналися консультанти по охороні праці і по економічній частині, а також керівник відділу кадрів.

2.8 Порядок контролю й приймання

1. Програма забезпечення надійності для програмного продукту використовується каскадне тестування типу білий ящик.
2. Види, склад, об'єм і методи випробувань системи α , β тестування, каскадне тестування.
3. Загальні вимоги до приймання робіт по стадіях після закінчення кожної стадії розробки програмного продукту дипломному керівнику пред'являється відповідна частина коду програми і розділ дипломної записки.

					КРБ.КІ.1.147–03.3.6	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 3

ОБГРУНТУВАННЯ ЗАСОБІВ РЕАЛІЗАЦІЇ

Для реалізації комп'ютерної гри було використано:

- багатоплатформовий інструмент для розробки двовимірних та тривимірних додатків та ігор *Unity*;
- мова програмування *C#*;
- середовище програмування *VisualStudio*.

Unity – середовище розробки комп'ютерних ігор, розроблена американською компанією *Unity Technologies*. *Unity* дозволяє створювати додатки, що працюють на більш ніж 25 різних платформах, що включають персональні комп'ютери, ігрові консолі, мобільні пристрої, інтернет-додатки та інші. Випуск *Unity* відбувся в 2005 році і з того часу йде постійний розвиток.

Основними перевагами *Unity* є наявність візуальної середовища розробки, міжплатформеній підтримки і модульної системи компонентів.

На *Unity* написані тисячі ігор, додатків, візуалізації математичних моделей, які охоплюють безліч платформ і жанрів. При цьому *Unity* використовується як великими розробниками, так і незалежними студіями.

Редактор *Unity* має інтерфейс, що складається з різних вікон, які можна розташувати на свій розсуд. Завдяки цьому можна проводити налагодження гри чи застосунка прямо в редакторі. Головні вікна – це оглядач ресурсів проєкту, інспектор поточного об'єкта, вікно попереднього перегляду, оглядач сцени та оглядач ієрархії ресурсів.

У 2D іграх *Unity* переважно використовує спрайти. В 3D іграх *Unity* здебільшого використовує тривимірні моделі (меші), на які накладаються текстури (зумовлюють вигляд поверхні об'єктів), матеріали (зумовлюють як поверхня реагуватиме на різні фактори) та шейдери.

					КРБ.КІ.1.147–03.3.6	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дат		

Unity підтримує стиснення текстур, міпмапінг і різні налаштування роздільності екрана для кожної платформи; забезпечує бамп-мапінг, мапінг відображень, паралакс-мапінг, затінення навколишнього світла у екранному просторі, динамічні тіні за картами тіней, рендер у текстуру та повноекранні ефекти обробки зображення, такі як зернистість, глибина чіткості, розмиття в русі, відблиски віртуальних лінз або ореол навколо джерел світла.

C# (*C Sharp*) – мова програмування, що поєднує об'єктно-орієнтовані і контекстно-орієнтовані концепції. Розроблено в 1998–2001 роках групою інженерів під керівництвом Андерса Хейлсберга в компанії *Microsoft* як основна мова розробки додатків для платформи *Microsoft.NET*. Компілятор з *C#* входить в стандартну установку самої *.NET*, тому програми на ньому можна створювати і компілювати навіть без інструментальних засобів по типу *VisualStudio*.

Переїнявши багато від своїх попередників мов *C++*, *Delphi*, Модула, *Smalltalk* і, особливо, *Java* – *C#*, спираючись на практику їх використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, *C#* на відміну від *C++* не підтримує множинне успадкування класів (між тим допускається множинна реалізація інтерфейсів).

C# розроблявся як мова програмування прикладного рівня для *CLR* і, як такий, залежить від можливостей самої *CLR*. Це стосується, перш за все, системи типів *C#*, яка відображає *BCL*. Присутність або відсутність тих чи інших виразних особливостей мови диктується тим, чи може конкретна мовна особливість бути трансльований в відповідні конструкції *CLR*.

Так, з розвитком *CLR* від версії 1.1 до 2.0 значно збагатився і сам *C#*; подібної взаємодії слід очікувати і в подальшому (проте, ця закономірність була порушена з виходом *C#* 3.0, що представляє собою розширення мови, що не спираються на розширення платформи *.NET*).

					КРБ.КІ.1.147–03.3.6	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дат		

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Ці продукти дають змогу розробляти як консольні програми, так і програми з графічним інтерфейсом, включно з підтримкою технології *Windows Forms*, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються *Microsoft Windows*, *Windows Mobile*, *Windows Phone*, *Windows CE*, *.NET Framework*, *.NET Compact Framework* та *Microsoft Silver light*.

Visual Studio побудована в архітектурі, що підтримує можливість використання доповнень (*Add-Ins*), – плагінів від сторонніх розробників, що дає змогу розширювати можливості середовища розробки.

					КРБ.КІ.1.147–03.3.6	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 4

ОПИС ЕТАПІВ РЕАЛІЗАЦІЇ

4.1 DFD–діаграма

Функціональні моделі, використовувані на стадії проектування ПО, призначені для опису функціональної структури проекрованої системи. Побудовані діаграми потоків *DFD* можуть уточнюватись, розширюватися і доповнюватися новими конструкціями. Окрім діаграм потоків *DFD* використовуються і інші діаграми, що відбивають системну архітектуру ПО, ієрархію екранних форм і меню, структурні схеми програм і так далі. Склад діаграм і міра їх деталізації визначаються необхідною повнотою опису системи для безпосереднього переходу до її подальшої реалізації (програмуванню).

Так, наприклад, для діаграми потоків *DFD* перехід від моделі бізнес-процесів організації до моделі системних процесів може відбуватися таким чином:

- зовнішні сутності на контекстній діаграмі замінюються або доповнюються технічними пристроями (наприклад, робочими станціями, принтерами і так далі);
- для кожного потоку даних визначається, за допомогою яких технічних пристроїв інформація робиться або передається;
- процеси на контекстній діаграмі нульового рівня замінюються відповідними процесорами, оброблювальними пристроями (процесорами можуть бути як технічні пристрої, персональні комп'ютери кінцевих користувачів, робочі станції, сервери баз даних, що так і служать, виконавці);
- визначається і зображується на діаграмі тип зв'язку між процесорами (наприклад, локальна мережа *Local Area Network*);

- визначаються завдання для кожного процесора (додатки, необхідні для роботи системи), для них будуються відповідні діаграми і визначається тип зв'язку між завданнями;
- встановлюються посилання між завданнями і процесами діаграм потоків цих наступних рівнів.

Ієрархія екранних форм моделюється за допомогою діаграм послідовностей екранних форм. Сукупність таких діаграм є абстрактною моделлю призначеного для користувача інтерфейсу системи, послідовністю появи екранних форм, що відбиває собою, в додатку. Побудова діаграм послідовностей екранних форм виконується таким чином:

- на діаграмі потоків *DFD* вибираються інтерактивні процеси нижнього рівня. Інтерактивні процеси потребують призначеного для користувача інтерфейсу, тому можна визначити екранну форму для кожного такого процесу;
- форма діаграми зображується у вигляді прямокутника для кожного інтерактивного процесу на нижньому рівні діаграми;
- визначається структура меню. Для цього інтерактивні процеси групуються в меню (або також як в моделі процесів, або іншим способом
- за функціональними ознаками або залежно від приналежності до певних об'єктів);
- форми з меню зображуються над формами, що відповідають інтерактивним процесам, і з'єднуються з ними переходами у вигляді стрілок, спрямованих від меню до форм;
- визначається верхня форма (головна форма додатка), що зв'язує усі форми з меню.

Опис предметної області

Якщо користувачем системи є користувач додатку, йому доступні наступні можливості:

- початок гри;

					КРБ.КІ.1.147–03.3.6	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дат		

- налаштування персонажу;
- вибір вже створених персонажів;
- взаємодія з об'єктами на локаціях;
- взаємодія з *NPC*.

Якщо користувачем системи є адміністратор, йому доступні наступні можливості:

- налаштування функцій гри;
- створення та додання нових ігрових елементів.

Перед побудовою контекстної діаграми потоків даних – *DFD* необхідно проаналізувати зовнішні події (зовнішні об'єкти) Зовнішні об'єкти взаємодіють з *IC* шляхом інформаційного обміну з нею. З опису предметної області виходить, що з програмною оболонкою працюють наступні групи людей: туристи, перекладачі. Ці групи є зовнішніми об'єктами. Вони не лише взаємодіють з системою, але так само визначають її межі і зображуються на початковій контекстній діаграмі потоків даних *DFD* як термінаторів (зовнішні сутності). Початкова контекстна діаграма потоків даних зображена нижче (рис. 4.1). У використовуваній нотації зовнішні сутності позначаються прямокутниками, а процеси – колами.

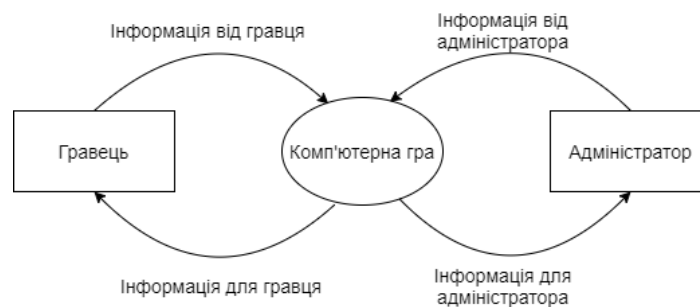


Рис. 4.1 – Початкова контекстна діаграма

Список подій будується у вигляді матриці *ELM* і описує різні дії зовнішньої сутності і реакцію системи на них. Ці дії являють собою зовнішні події що впливають на системи.

Всі дані позначаються як *ND*, тому що іони є подіями, які система сприймає безпосередньо.

Таблиця 4.1

Матриця списку подій (*ELM*)

№	Опис	Тип	Реакція
1	Адміністратор бажає додати нового ворога	<i>ND</i>	Внести нового ворога до бази даних
2	Адміністратор бажає додати новий рівень	<i>ND</i>	Додати новий рівень до бази даних
3	Адміністратор бажає внести оновлення у грі	<i>ND</i>	Внести зміни у роботі системи
4	Адміністратор бажає додати нові ігрові функції	<i>ND</i>	Додати нові функції у процес гри
5	Користувач бажає змінити конфігурацію	<i>ND</i>	Змінити стандартну конфігурацію на користувацьку
6	Користувач бажає створити нове збереження даних	<i>ND</i>	Створення комірки збереження даних у базі даних
7	Користувач бажає обрати певний рівень	<i>ND</i>	Запустити обраний рівень
8	Користувач бажає вийти до головного меню	<i>ND</i>	Вихід до головного меню
9	Користувач бажає вийти з гри	<i>ND</i>	Закриття та повний вихід з програми
10	Користувач бажає звернутися до відділу підтримки	<i>ND</i>	Відправлення даних до поштової адреси відділу підтримки

Для завершення аналізу функціонального аспекту поведінки системи будується повна контекстна діаграма – діаграма 0-го рівня, при цьому 0-вий процес декомпозується на декілька основних процесів, що представляє діяльність системи.

Існуючі абстрактні потоки даних між зовнішніми сутностями і процесами трансформуються в потоки, що представляють обмін даними на більш конкретному рівні. Список подій показує, які потоки існують на цьому рівні.



Рис. 4.2 – Повна контекстна діаграма

4.2 SADT модель

Етап проектування є критичним для створення високоякісних систем. Системне проектування – це дисципліна, що визначає підсистеми, компоненти і способи їх з'єднання, а також задає обмеження, при яких система повинна функціонувати, та вибирає найбільш ефективне поєднання людей, машин і програмного забезпечення для реалізації системи.

Під словом «система» ми розуміємо сукупність взаємодіючих компонент і взаємозв'язків між ними.

Під терміном «моделювання» ми розуміємо процес створення точного опису системи. Особливо важким виявляється опис систем середньої складності, таких, як система комутації в телефонних мережах, управління аероповітряними перевезеннями, збірка автівок тощо.

З погляду людини, ці системи описати достатньо важко, тому що вони настільки великі, що практично неможливо перелічити всі їх компоненти із своїми взаємозв'язками, і в той же час недостатньо великі для вживання загальних спрощуючих припущень.

SADT (Технологія структурного аналізу і проектування) – це графічні позначення для опису систем.

Деякі області ефективного застосування *SADT*:

- програмне забезпечення телефонних мереж;
- системна підтримка і діагностика;
- довгострокове і стратегічне планування;
- автоматизоване виробництво і проектування;
- конфігурація комп'ютерних систем;
- навчання персоналу;
- вбудоване програмне забезпечення оборонних систем;
- управління фінансами і матеріально технічним постачанням.

Широкий спектр областей указує на універсальність методології *SADT*.

Таким чином, розробники почали формалізувати процес створення системи, розбиваючи його на наступні фази:

- аналіз;
- проектування;
- реалізація;
- об'єднання;
- тестування.

SADT – це методологія, розроблена спеціально для того, щоб полегшити опис і розуміння штучних систем, які потрапляють в розряд середньої складності. В *SADT* моделях використовуються як природня так і графічна мови. Для передачі інформації про конкретну систему джерелом природньої мови служить людина, що описує систему, а джерелом графічної мови – сама методологія *SADT*.

З погляду *SADT* модель може бути зосереджена або на функціях системи, або на її об'єктах. *SADT* – моделі, орієнтовані на функції, прийнято називати функціональними моделями, а орієнтовані на об'єкти системи – моделями даних. Функціональна модель представляє з необхідним ступенем деталізації систему функцій, які у свою чергу відображають свої взаємозв'язки через об'єкти системи.

При визначені моделі необхідно визначити позицію, з якої спостерігатиметься система, і створюватиметься її модель. Якість опису системи різко знижується, якщо вона не сфокусована ні на чому, *SADT* вимагає, щоб модель розглядалась весь час з однієї і тієї ж позиції. Ця позиція називається точкою зору даної моделі.

SADT – модель об'єднує і організовує діаграми в ієрархічні структури, в яких діаграми верхніх рівнів менш деталізовані ніж діаграми нижніх рівнів. Іншими словами, *SADT* – модель можна представити у вигляді деревовидної структури діаграм, де верхня діаграма є самою загальною, а самі нижні найбільш деталізовані. Діаграма є основним робочим елементом при створенні моделі.

Кожна *SADT* – діаграма містить блоки і дуги. Блоки зображують функції системи, яка моделюється.

Дуги зв'язують блоки разом і відображають взаємодії і взаємозв'язки між ними. Функціональні блоки на діаграмах зображуються прямокутниками. Блок представляє функцію або активну частину системи, наприклад, визначити ступінь виконання завдання, вибрати інструменти, підготувати робоче місце.

На відміну від інших графічних методів структурного аналізу в *SADT* кожна сторона блоку має особливе, цілком визначене призначення. Ліва сторона блоку призначена для входів, верхня – для управління, права – для виходів, нижня – для механізмів.

					КРБ.КІ.1.147–03.3.6	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дат		

Таке позначення відображає певні системні принципи: входи перетворюються у виходи, управління обмежує або вказує умови виконання перетворень, механізми показують хто, що і як виконує функцію.

Дуги на *SADT* – діаграмі зображуються суцільними смугами із стрілками на кінцях. Для функціональних *SADT* – діаграм дуга представляє безліч об'єктів. Тут використовується загальне поняття об'єкт, оскільки дуги в *SADT* можуть представляти, наприклад, плани, дані в комп'ютерах, машини, інформацію тощо.

Методологія *SADT* є сукупністю методів, правил і процедур, призначених для побудови функціональної моделі об'єкту будь-якої предметної області.

Функціональна модель *SADT* відображає функціональну структуру об'єкту, тобто виділені нею дії і зв'язки між цими діями. Основні елементи цієї методології ґрунтуються на наступних концепціях:

1. Графічне представлення блокового моделювання. Графіка блоків і дуг. *SADT* діаграми відображає функцію у вигляді блоку, а інтерфейси. Входу/виходу представляються дугами, які відповідно входять в блок і виходять з нього. Взаємодія блоків один з одним описується за допомогою інтерфейсних дуг виражаючих «обмеження», які у свою чергу визначають, коли і ким функції виконуються і управляються.
2. Концепція строгості і точності. Виконання правил *SADT* вимагає достатньої строгості і точності, не накладаючи при тому великих обмежень на дії аналітика. Правила *SADT* включають обмеження кількості блоків на кожному рівні (3–6 блоків), зв'язаність діаграм (номери блоків), унікальність міток і найменувань (відсутність імен, що повторюються).

Методологія *SADT* може використовуватися для моделювання широкого круга систем і визначення вимог і функцій, а потім для розробки системи, яка задовольнятиме цим вимогам і реалізовуватиме ці функції.

					КРБ.КІ.1.147–03.3.6	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дат		

Вже існуючі системи *SADT* можуть використовуватися для аналізу функцій, що виконуються системою. А також для вказівки механізмів, за допомогою яких вони здійснюються.

Результатом використання методології *SADT* є модель, яка складається з діаграм та фрагментів текстів, що мають посилання один на одного.

Однією з найважливіших особливостей методології є поступове введення все більшої кількості рівнів деталізації в ході створення діаграм, які відображають модель.

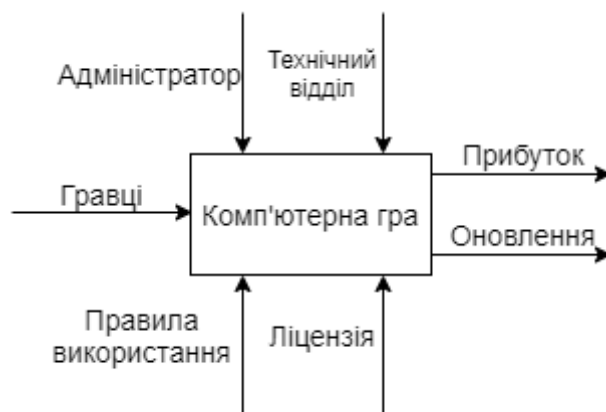


Рис. 4.3 – *SADT* модель 0 рівня

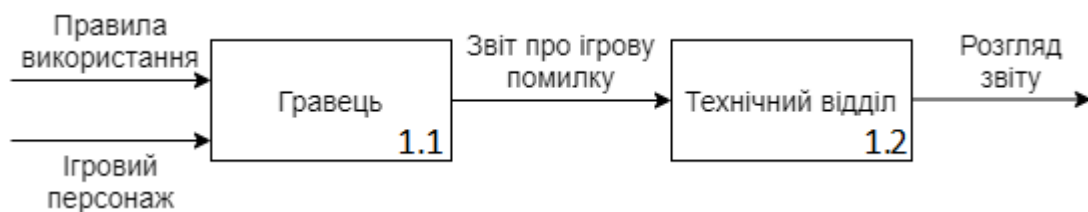


Рис. 4.4 – *SADT* модель 1 рівня



Рис. 4.5 – *SADT* модель 2 рівня

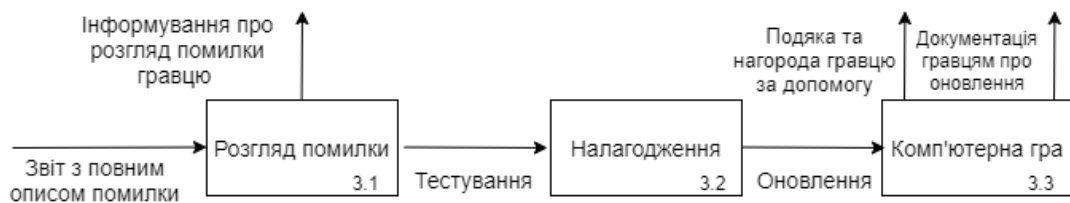


Рис. 4.6 – SADT модель 3 рівня

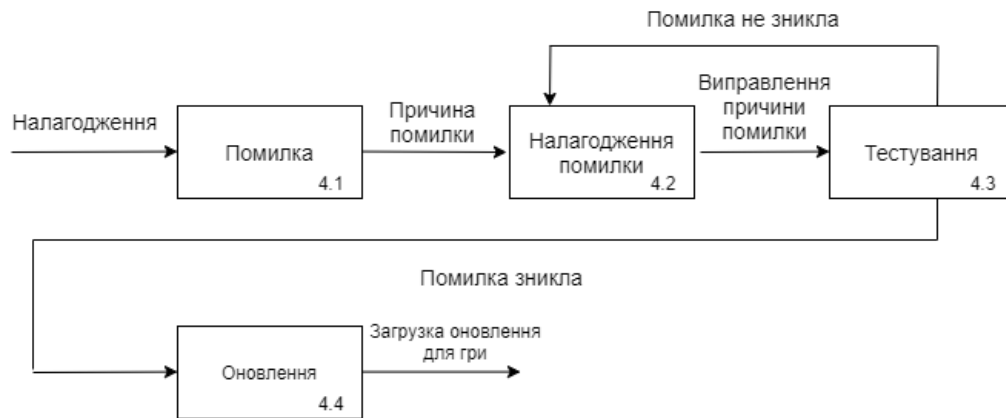


Рис. 4.7 – SADT модель 4 рівня

4.3 IDEF3

IDEF3 (Process Description Capture (Документування технологічних процесів)) – методологія документування процесів, що відбуваються в системі (наприклад, на підприємстві), описуються сценарій і послідовність операцій для кожного процесу.

IDEF3 має прямий взаємозв'язок з методологією *IDEF0*. Кожна функція (функціональний блок) може бути представлена у вигляді окремого процесу засобами *IDEF3*.

IDEF3 – це метод, що має основний метою дати можливість аналітикам описати ситуацію, коли процеси виконуються в певній послідовності, а також описати об'єкти, що брати участь спільно в одному процесі.

Техніка опису набору даних *IDEF3* є частиною структурного аналізу. На відміну від деяких методик описів процесів *IDEF3* не обмежує аналітика надмірно твердими рамками синтаксису, що може привести до створення неповних або суперечливих моделей.

IDEF3 може бути також використаний як метод створення процесів. *IDEF3* доповнює *IDEF0* і містить усе необхідне для побудови моделей, які надалі можуть бути використані для імітаційного аналізу.

Існують два типи діаграм у стандарті *IDEF3*, що представляють опис того самого сценарію технологічного процесу в різних ракурсах. Діаграми стосовні до першого типу називаються діаграмами Опису Послідовності Етапів Процесу (*Process Flow Description Diagrams, PFDD*), а до другого – діаграмами Стану Об'єкта і його Трансформацій в Процесі (*Object State Transition Network, OSTN*).

Кожна робота в *IDEF3* описує який-небудь сценарій бізнес процесу й може бути складовою іншої роботи. Оскільки сценарій описує мету й рамки моделі, важливо, щоб роботи йменувалися віддієслівним іменником, що позначають процес дії, або фразою, що містить такий іменник.

Точка зору на модель повинна бути документована. Звичайно це точка зору людини, відповідального за роботу в цілому. Також необхідно документувати мету моделі – ті питання, на які покликано відповісти модель.

Діаграма є основною одиницею опису в *IDEF3*. Важливо правильно побудувати діаграми, оскільки вони призначені для читання іншими людьми (а не тільки автором).

Одиниці роботи – *Unit of Work (UOW)*, також називані роботами (*activity*), є центральними компонентами моделі. В *IDEF3* роботи зображуються прямокутниками із прямими кутами й мають ім'я, що позначає процес дії і номер (ідентифікатор); інше ім'я відображає основний вихід (результат) роботи (наприклад, "Виготовлення виробу"). Часте ім'я міняється в процесі моделювання, оскільки модель може уточнюватися й редагуватися. Ідентифікатор роботи привласнюється при створенні й не міняється ніколи.

					КРБ.КІ.1.147–03.3.6	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дат		

Навіть якщо робота буде вилучена, її ідентифікатор не буде знову використовуватися для інших робіт.

Звичайно номер роботи складається з номера батьківської роботи й порядкового номера на поточній діаграмі.

Зв'язки показують взаємини робіт. Усі зв'язки в *IDEF3* односпрямовані й можуть бути спрямовані куди завгодно, але звичайно діаграми *IDEF3* намагаються побудувати так, щоб зв'язки були спрямовані ліворуч праворуч. В *IDEF3* розрізняють три типи стрілок, що зображують зв'язки, стиль яких устанавлюється через меню *Edit/ArrowStyle*.

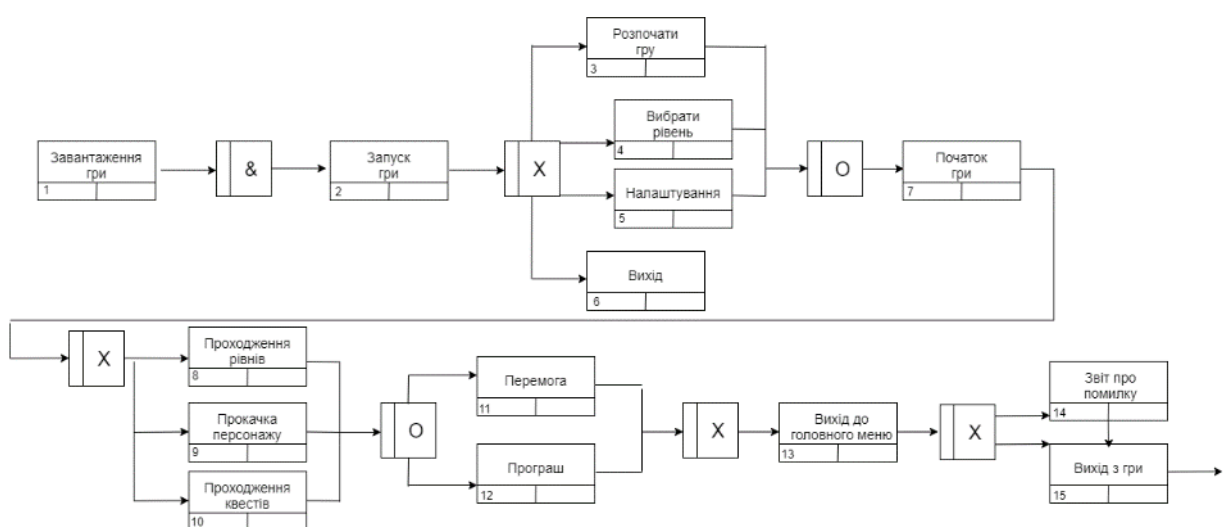


Рис. 4.8 – *IDEF3*

4.4 *Use-Case* діаграма

Діаграма *UseCase* визначає поведінку системи з погляду користувача. Вона розглядається як головний засіб для первинного моделювання динаміки системи і використовується для з'ясування вимоги до системи, що розробляється.

До складу діаграм *UseCase* входять елементи *UseCase*, актори, а так само відносини залежності, узагальнення і асоціації.

Вершинами в діаграмі *UseCase* є актори і елементи *UseCase*.

Актор це роль об'єкту поза системою, який прямо взаємодіє з її частиною, конкретним елементом *UseCase*. Розрізняють акторів і користувачів.

Користувач це фізичний об'єкт, який використовує систему. Він може грати декілька ролей і тому може моделюватися декількома акторами. Актором можуть бути різні користувачі.

Елемент *UseCase* опис послідовності дій, які виконуються системою і проводять для окремого актора видимий результат.

Один актор може використовувати декілька елементів *UseCase* і навпаки. Кожен елемент *UseCase* задає певний шлях використання системи. Набір всіх елементів *UseCase* визначає повні функціональні можливості системи.

Відносини в діаграмах *UseCase*.

Між актором і елементом *UseCase* можливий тільки один вид відносин асоціація, що відображає їх взаємодію. Вона може бути помічена ім'ям, ролями і потужністю.

Між акторами допустиме відношення узагальнення, що означає, що екземпляр нащадка може взаємодіяти з такими ж різновидами екземплярів елементів *UseCase*, що і екземпляр батька.

Між елементами *UseCase* визначені відносини узагальнення і два різновиди відношення залежності включення і розширення.

Розробка діаграми переслідує наступні дії:

- визначити загальні межі і контекст предметної області, що моделюється;
- сформулювати загальні вимоги до функціональної поведінки проектованої системи;
- розробити початкову концептуальну модель системи для її подальшої деталізації у формі логічних і фізичних моделей;
- підготувати початкову документацію для взаємодії розробників системи з її замовниками і користувачами.

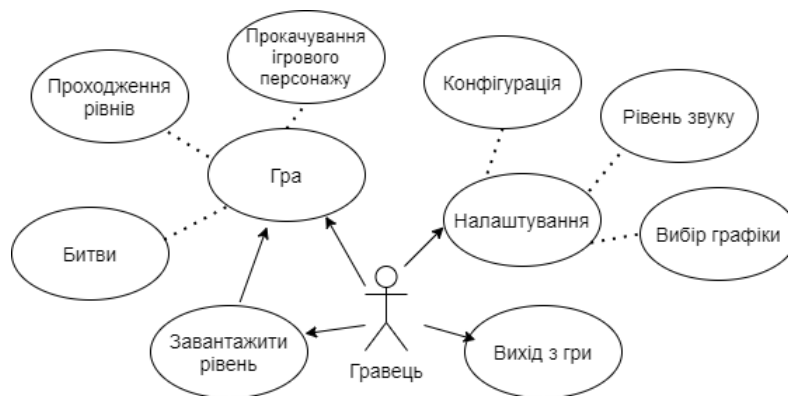


Рис. 4.9 – Use-Case діаграма з погляду гравця

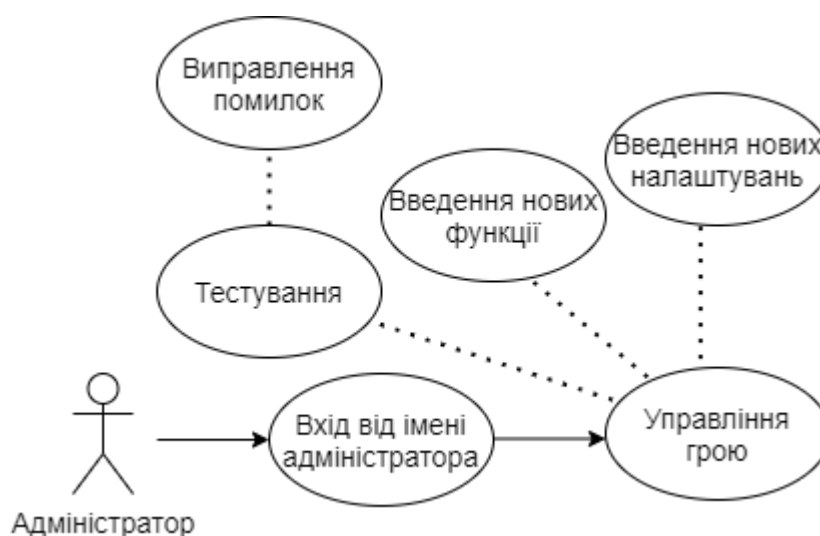


Рис. 4.10 – Use-Case діаграма з погляду адміністратора

Компонентна діаграма – різновид діаграми реалізації, що моделює фізичні аспекти систем. Вона показує організацію набору компонент і залежності між ними.

Елементами таких діаграм є компоненти і інтерфейси, а також відносини залежності і реалізації.

Компонент – фізична і замінювана частина системи, яка відповідає набору інтерфейсів і забезпечує реалізацію цього набору. Графічно він відображається як прямокутник з вкладками, що зазвичай включає ім'я.

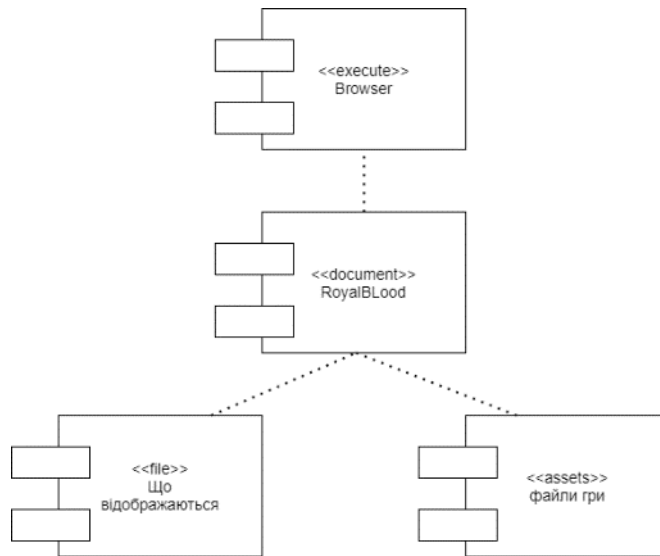


Рис. 4.11 – Діаграма компонентів

Діаграма послідовності – в *UML*, відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень.

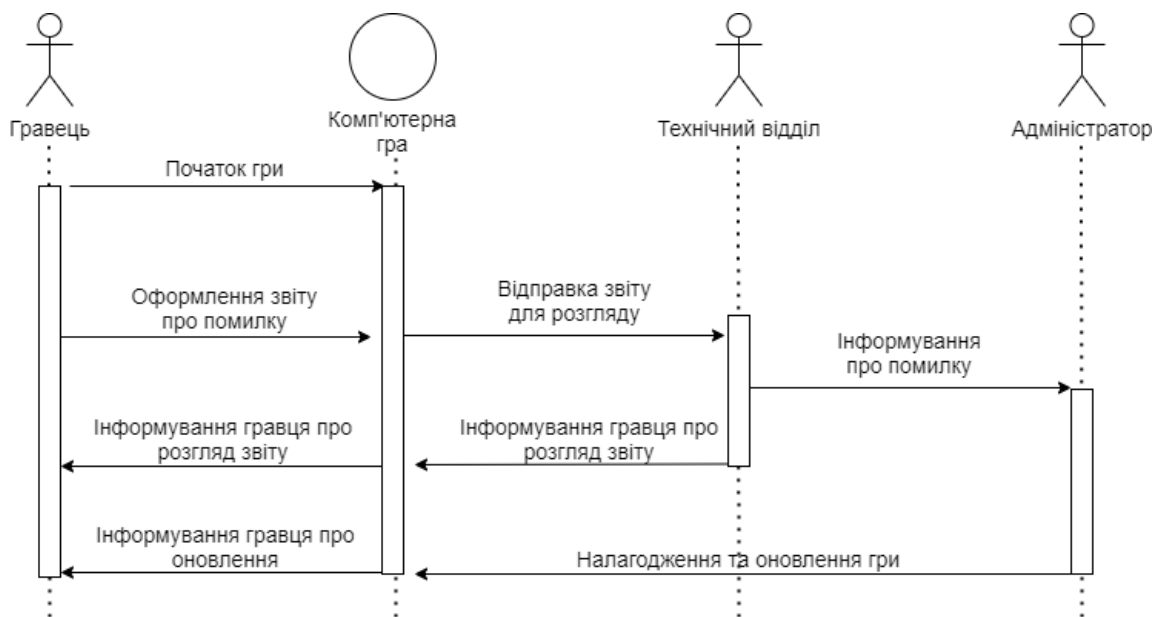


Рис. 4.12 – Діаграма Послідовності

Діаграма класів (*class diagram*) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування.

Діаграма класів може відбивати, зокрема, різні взаємозв'язки між окремими сутностями предметної області, такими як об'єкти і підсистеми, а також описує їх внутрішню структуру і типи стосунків. На цій діаграмі не вказується інформація про тимчасові аспекти функціонування системи. З цієї точки зору діаграма класів є подальшим розвитком концептуальної моделі проектованої системи.

Діаграма класів складається з безлічі елементів, які в сукупності відбивають декларативні знання про предметну область. Ці знання інтерпретуються у базових поняттях мови *UML*, таких як класи, інтерфейси і стосунки між ними і їх складовими компонентами. При цьому окремі компоненти цієї діаграми можуть утворювати пакети для представлення загальнішої моделі системи.

У загальному випадку пакет статичної структурної моделі може бути представлений у вигляді однієї або декількох діаграм класів.

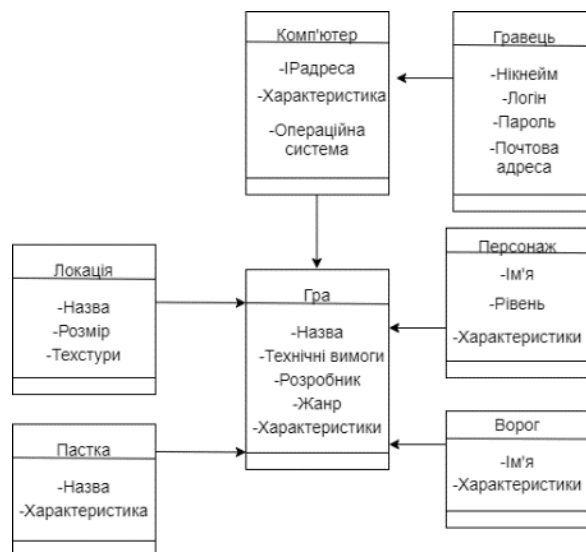


Рис. 4.13 – Діаграма класів

4.5 Проектування бази даних

Ігровий движок *Unity* має вбудовану базу даних для ігрових об'єктів. Дані з БД зберігаються у реєстрі комп'ютера з адресою *HKEY_CURRENT_USER\Software\Unity\UnityEditor\[companyname]\[projectname]*. Внесена у БД інформація має вигляд програмного коду, написаного на мові програмування *C#*.

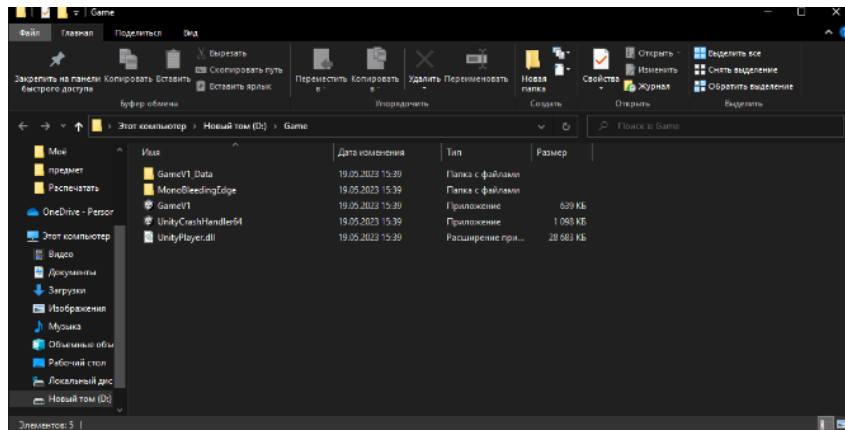


Рис. 4.14 – Папка з грою

Вбудована у ігровий двигун база даних заносить до даних реєстру усю інформацію, що зберігається у грі.

Таблиця «Гравець» (Рис. 4.15) зберігає дані про гравця.

Гравець			
IDКористувача	integer		
Логин	text		
Пароль	text		
Почта	text		
Страна проживания	text		
Страна	text		
Возраст	text		
IDПерсонажа	text		

Рис. 4.15 – Таблица “Гравець”

Таблиця «Персонаж» (Рис. 4.16) зберігає інформацію про конкретного персонажа та його характеристики.

Персонаж			
	IDПерсонажа	integer	 
	Уровень	text	 
	Здоровье	text	 
	Урон	text	 
	Прыжок	text	 
	Скорость	text	 
	Опыт	text	 
	Броня	text	 
	IDЛокации	text	 

Рис. 4.16 – Таблица “Персонаж”

РОЗДІЛ 5

ОПИС ПРОГРАМНОГО ПРОДУКТУ

Для того, щоб розпочати гру, потрібно мати саму папку з грою та запустити її за допомогою ярлика.

При запуску гри відкривається головне меню (рис. 5.1), яке пропонує гравцям такі можливості як:

- створити нового персонажа;
- обрати вже існуючого персонажа;
- вихід з гри.



Рис. 5.1 – Головне меню гри

Скориставшись кнопкою “Створити персонажа” гравець попадає до нового вікна програми (рис. 5.2), де має можливість обрати клас персонажу та його ім’я. Щоб продовжити потрібно натиснути кнопку “Створити”.



Рис. 5.2 – Створення нового персонажу

					КРБ.КІ.1.147–03.3.6	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дат		

Якщо ж натиснути на кнопку “Продовжити” – гравець попадає в вікно програми (рис. 5.3), де може обрати та продовжити гру вже на створеному персонажі, якщо такі існують.

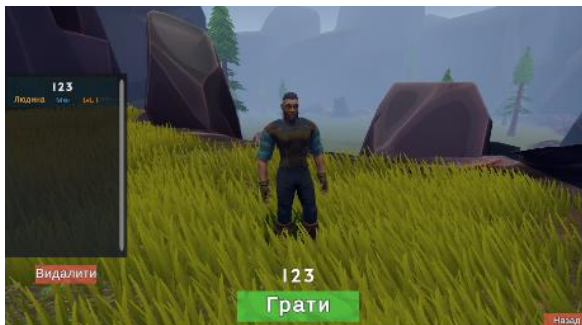


Рис. 5.3 – Вибір існуючого персонажу

Кнопка “Вихід з гри” вимикає гру та закриває її повністю.

Створивши нового чи обравши вже існуючого персонажа, запускається завантаження основного рівня (рис. 5.4).

Натиснувши “ЛКМ” гравця перекидає на основний рівень (рис. 5.5).



Рис. 5.4 – Екран завантаження рівня



Рис. 5.5 – Початкова локація рівня

Після цього гравець має змогу управляти своїм персонажем за допомогою клавіш. Щоб дізнатися які клавіші за що відповідають, змінити гучність або самі клавіші потрібно натиснути клавішу “Esc” та у спливаючому вікні “Опції” (рис. 5.6) натиснути кнопку “Налаштування”.



Рис. 5.6 – Меню опцій



Рис. 5.7 – Меню налаштування

Також у вікні “Опції” (рис. 5.6) натиснувши кнопку “Головне меню”, гравця перекидає до головного меню гри (рис. 5.1). Кнопка “Вихід з гри” повністю вимикає гру.

Розібравшись із керуванням гравцю дається повний контроль над персонажем. Він може робити все, що йому заманеться, якщо це дозволяє гра. Але для того, щоб пройти гру, гравцеві потрібно розпочати сюжетну лінію.

Для цього слід підійти до персонажа (рис. 5.8), що стоїть ліворуч від точки появи та натиснути по ньому англійською клавішею «E».



Рис. 5.8 – Персонаж “Староста”

Натиснувши кнопку відкриється діалогове вікно (рис. 5.9) з цим персонажем. Гравець може обрати один із кількох варіантів відповідей, що різняться між собою. Наприклад, відповідь «Прощавай» завершить діалог, а «Я воїн» навпаки продовжить розмову, що допоможе гравцеві більше дізнатися про світ, в якому він знаходиться і зможе просунути далі за сюжетом гри.



Рис. 5.9 – Діалогове вікно з персонажем “Староста”

					КРБ.КІ.1.147–03.3.6	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дат		

Практично кожен персонаж має своє діалогове вікно. Це додає атмосферу грі і дозволяє гравцеві все більше і більше дізнаватися про навколишній світ. Нижче наведено кілька варіантів діалогових ліній із різними персонажами.

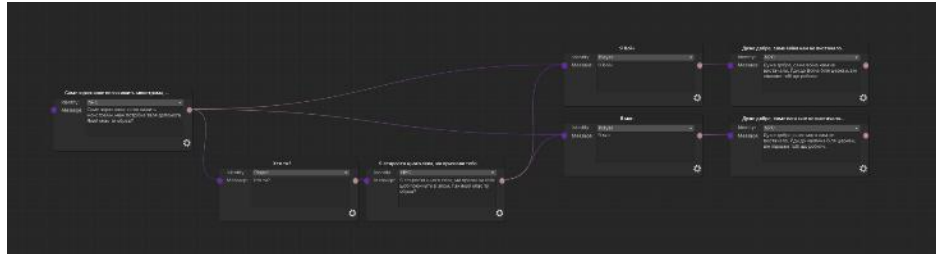


Рис. 5.10 – Діалогова лінія персонажа “Староста”

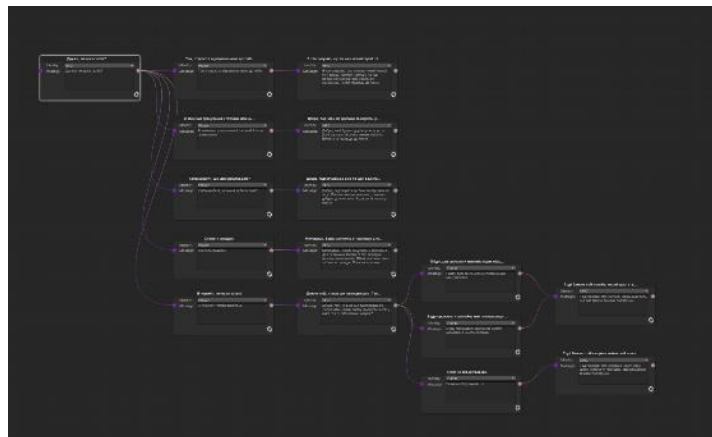


Рис. 5.11 – Діалогова лінія персонажа “Чаклун”

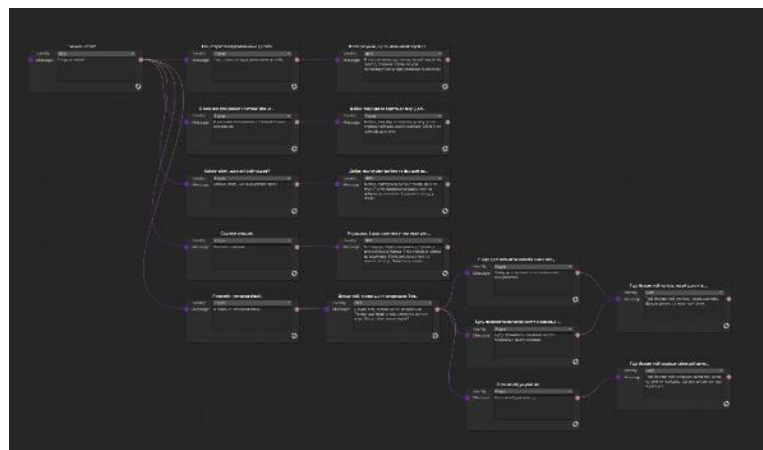


Рис. 5.12 – Діалогова лінія персонажа “Воїн”

Для того щоб пройти гру, гравцеві потрібно повністю пройти сюжетну лінію. Так після розмови зі старостою гравець має підійти до одного з двох персонажів залежно від класу, який вибрав гравець. Якщо гравець вибрав клас воїна, то треба поговорити саме з персонажем «Воїн» (рис. 5.13), а якщо клас мага, то з персонажем «Чаклун» (рис. 5.14). Інакше діалог ні до чого не спричинить.



Рис. 5.13 – Персонаж “Воїн”



Рис. 5.14 – Персонаж “Чаклун”

Дані персонажі всю гру будуть давати гравцеві квести (завдання), які він зобов'язаний виконати. Після виконання завдання гравцеві відкриється доступ до нового варіанту діалогу з персонажем, що продовжить сюжет.

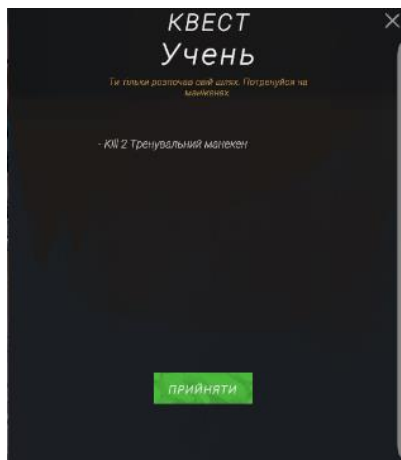


Рис. 5.15 – Перший квест

Гравець у будь-який момент може переглянути інформацію про будь-який квест, який був прийнятий до цього моменту. Щоб це зробити, потрібно натиснути англійську клавішу «Q». Після цього відкриється меню (рис. 5.16) з перерахованим списком всіх прийнятих чи закінчених квестів. Для перегляду інформації про квест потрібно на нього натиснути. Також від квесту можна відмовитись або відслідковувати його прогрес, натиснувши відповідні кнопки.

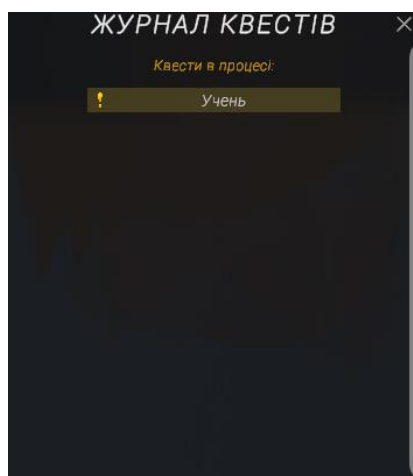


Рис. 5.16 – Меню квестів



Рис. 5.17 – Інформація про квест “Учень”

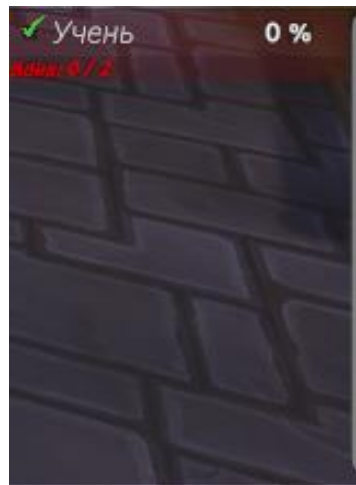


Рис. 5.18 – Панель відслідковування прогресу квесту “Учень”

Зазвичай завдання квестів пов'язані з убивством певного ворога. Тому гравцю слід розібратися з бойовою механікою та можливістю покращення свого персонажа.

Бойова механіка у цій грі полягає у застосуванні здібностей персонажа, які в кожного класу різні. На цій версії гри у кожного класу є максимум по чотири здібності, які відкриваються при досягненні певного рівня персонажа.

Перша (базова) здібність відкрита на початку гри, за допомогою якої гравець може починати свій шлях. Друга відкривається на другому рівні, третя на третьому рівні, четверта, відповідно, при досягненні четвертого рівня.

Для перегляду всіх здібностей персонажу потрібно натиснути англійську клавішу “B”, що відкриє меню Здібностей (рис. 5.19). Зеленим позначені відкриті здібності, червоним – закриті. Щоб мати змогу використовувати нову здібність (після досягнення нового рівня) потрібно затиснувши ліву кнопку миші перетягнути цю здібність у порожню комірку здібностей (рис. 5.20).

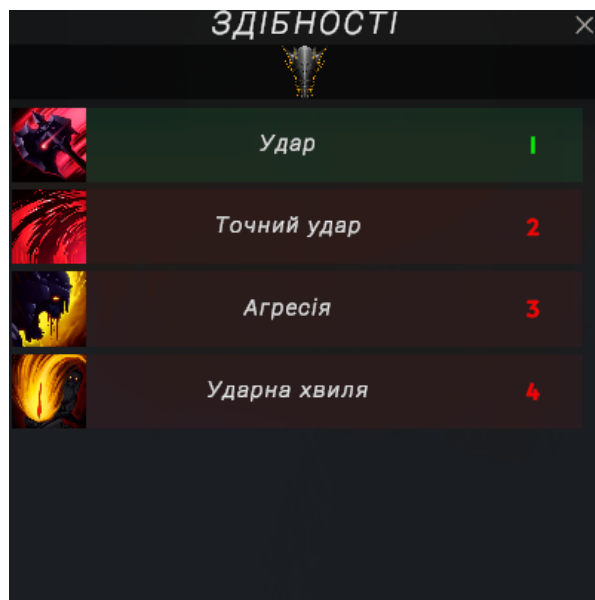


Рис. 5.19 – Меню здібностей



Рис. 5.20 – Головний інтерфейс

Кожну здібність гравець може прокачувати, покращуючи її характеристики, що дозволить швидше та легше перемогати ворогів. Прокачування здібностей відбувається у відповідному меню. Щоб його увімкнути, для початку, потрібно натиснути англійську клавішу “C”, після чого відкриється меню з ім'ям гравця та його обладнанням (рис. 5.21). Праворуч від меню можна буде побачити 4 іконки, нам потрібна третя, під ім'ям "Таланти". Натиснувши на неї, відкриється потрібне нам меню (рис. 5.22).

Так як у кожного класу здібності різні, то і меню прокачування здібностей у них також різні. Щоб прокачати здібність, потрібно натиснути на неї лівою кнопкою миші. Для прокачування використовуються бали, які гравець отримує у разі підвищення рівня персонажа. Слід зауважити, що прокачати можливо тільки ту здібність, котра вже відкрита.



Рис. 5.21 – Меню обладнання персонажу

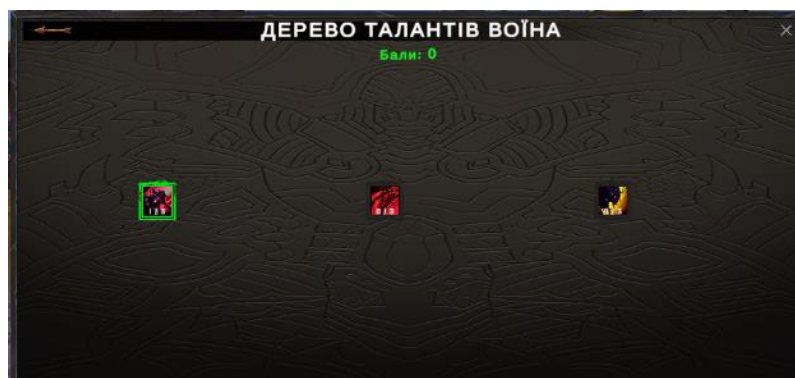


Рис. 5.22 – Меню прокачування здібностей класу “Воїн”

Щодо обладнання персонажа, у грі є різновид зброї, броні та зілля. Гравець може комбінувати їх як завгодно. Для цього йому доступний інвентар (рис. 5.23), де він може зберігати будь-яку річ і меню обладнання (рис. 5.21), в якому він може одягати броню або зброю.

Інвентар відкривається за допомогою англійської клавіші “I”. Щоб одягти необхідну річ, потрібно натиснути на неї правою кнопкою миші.



Рис. 5.23 – Інвентар персонажу

Також і в місті можна знайти торговця (рис. 5.24), у якого можна купити необхідну гравцю річ. Список товарів для продажу (рис. 5.25) також залежить від класу персонажа. Купуються товари за ігрову валюту, яку можна отримати, вбиваючи ворогів, але про це пізніше.



Рис. 5.24 – Персонаж “Торговець”

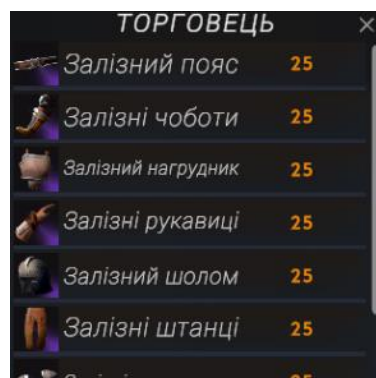


Рис. 5.25 – Список товарів торговця для класу “Воїн”

Після обмундирування персонажа та вивчення бойової механіки, настає час подолання ворогів. У грі кожен ворог по-своєму унікальний, тому гравцю слід бути обережним з ними, особливо з босом. Для прикладу розглянемо кілька ворогів, які зустрічатимуться на шляху гравця. Дізнаємося їх можливості, особливості, характеристики і як краще з ними боротися.

Першим і самим безтурботним ворогом (якщо його таким можна назвати) став тренувальний манекен (рис. 5.26). Він не атакує і служить помічником для вивчення бойової механіки на практиці та підняття перших рівнів персонажа. Дуже корисна річ для новачка. Знаходиться біля головних воріт.



Рис. 5.26 – Ворог “Тренувальний манекен”

Наступним із представників ворогів є кабан (рис. 5.27). Не дуже складний для вбивства, вважається першим для проходження мобом після манекена. Якихось особливостей крім звичайних ударів не має.

Однак не варто атакувати його близько до інших кабанів, вони мають властивості допомагати один одному.



Рис. 5.27 – Ворог “Кабан”

Наступним ворогом став скелет (рис. 5.28). Щоб здолати цього моба потрібно багато постаратися. Має велику шкоду і на додаток здібність відновлення життя. Скелети не допомагають один одному. Головна слабкість скелетів це їхня повільність. Можна скористатись перервою між атаками, щоб без ушкоджень атакувати його.



Рис. 5.28 – Ворог “Скелет”

Ведмідь – найскладніший ворог гри (рис. 5.29). Щоб його здолати потрібно добре прокачати персонажа, покращити всі здібності та надіти

найкращі обладунки зі зброєю, інакше шанси здолати його вкрай малі. Мають хорошу швидкість, швидкі та сильні удари та велику кількість здоров'я.

Так само, як і кабанам, властиві допомагати один одному. Нова особливість полягає в тому, що якщо близько до них підійти, вони першими атакують гравця. Здібністю ведмедів є шанс кровоточивих ударів, які забирають здоров'я у гравця з часом.



Рис. 5.29 – Ворог “Ведмідь”

Якщо гравця перемогли, то не варто засмучуватися. Усі речі та прогрес персонажа зберігатимуться. Також вилізе вікно смерті (рис. 5.30), де гравцеві запропонують переродитись. Натиснувши на зелену кнопку “Переродитись” гравець з'явиться на точці спавну у місті.

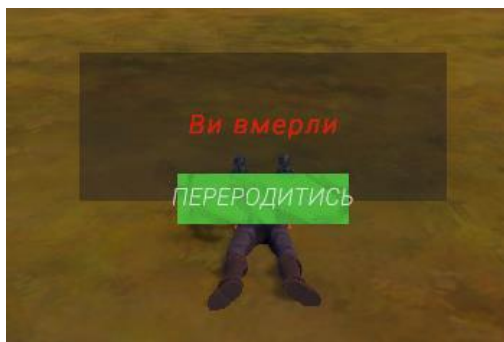


Рис. 5.30 – Вікно смерті

РОЗДІЛ 6

ОЦІНКА ЕФЕКТИВНОСТІ ПРОЕКТУ

Дана дипломна робота призначена для розробки комп'ютерної гри у жанрі *RPG* за допомогою платформи *Unity*.

Швидкий розвиток комп'ютерів призвів до неможливих на той час можливостей. Наразі навіть самим звичайним комп'ютером користуються майже всі. Він знаходиться у кожному будинку і без нього багато людей вже не зможуть обійтись. Різні сфери виробництва, розробки, проектування – всюди використовуються дані пристрої. Але ж люди використовують комп'ютери не тільки як інструмент праці, а й для розваг. Саме для цього розробляються комп'ютерні ігри, які різняться за виміром, жанром, графікою. Кожна людина зможе знайти щось цікаве для себе, так як ринок комп'ютерних ігор вже доволі величезний і різноманітний. Через високу конкуренцію команди розробників можуть працювати роками, щоб саме їхня гра досягла успіху, отримала шанувальників й заробила на цьому великі гроші, окупивши затрати на розробку.

У цьому розділі буде проведено оцінку ефективності проекту, який було реалізовано в рамках даної дипломної роботи. Оцінка ефективності є важливою складовою процесу управління проектом, оскільки дозволяє визначити ступінь досягнення поставленої мети та оцінити успішність виконання проекту. Також буде розглянуто вплив розробленої гри на життя людей, та викладено розрахунки витрат на її розробку.

“В умовах відкритої ринкової економіки сфера оцінки ефективності науково-технічного розвитку постійно розширюється, і з'являється все більше видів основних ефектів НДДКР, які потребують оцінки. Вони включають:

					КРБ.КІ.1.147–03.3.6	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дат		

1. Маркетинговий ефект, що відображає комерційну успішність на ринку, що виражається у збільшенні попиту, продажу, прибутку.
2. Науково-технічний ефект, який відображає досягнення та результати наукових та технічних досліджень, що застосовуються для розробки. Він виявляється у поліпшенні продуктивності, зниженні витрат, підвищенні якості та інноваційності, що сприяє прогресу та розвитку відповідної галузі.
3. Соціальний ефект, що описує позитивний вплив проекту на суспільство. Він може виявлятися у поліпшенні якості життя, соціальної справедливості, доступності послуг, розвитку освіти, охорони здоров'я, екологічної стійкості та інших аспектів, що сприяють благополуччю суспільства та підвищенню рівня розвитку.
4. Економічний ефект, що показує результативність та фінансову вигоду проекту. Він проявляється у збільшенні прибутку, зниженні витрат, зростанні продуктивності, створенні нових робочих місць, стимулюванні економічного зростання та розвитку, залученні інвестицій та поліпшенні конкурентоспроможності компанії чи регіону.

Проаналізуючи ці ефекти, можна обґрунтувати та визначити оцінку ефективності розробленого продукту, дивлячись на його вплив на життя суспільства.”[26]

Маркетинговий ефект диплома може виявитися у кількох аспектах.

По-перше, дипломна робота може привернути увагу потенційних інвесторів чи спонсорів, зацікавлених у розробці подібних ігор, що може призвести до фінансової підтримки чи партнерства.

По-друге, публікація результатів дослідження та розробки у наукових чи професійних журналах чи конференціях може підвищити впізнаваність та авторитет автора, а також привернути увагу ігрових спільнот та розробників, що може відкрити нові можливості для подальшої співпраці.

					КРБ.КІ.1.147–03.3.6	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дат		

По-третє, проект може позначитися репутації навчального закладу, де було виконано роботу. Успішна розробка може стати візитною карткою університету, привертаючи увагу студентів, абітурієнтів та партнерів. Це може підвищити престиж та рейтинг навчального закладу, а також залучити більше студентів та ресурсів для подальшого розвитку освітніх програм у галузі ігрової розробки.

Ринок розробників ігор у всьому світі є динамічним та конкурентним. В даний час ігрова індустрія переживає значне зростання, і розробники ігор відіграють ключову роль у створенні захоплюючих та інноваційних ігрових продуктів. Ринок розробників ігор включає як великі студії та видавців, так і незалежних розробників, що працюють у різних країнах.

Країни, такі як США, Японія та Великобританія, історично були великими ігровими ринками і мають довгу і успішну історію розробки ігор. Однак останніми роками низка інших країн також заявила про себе в ігровій індустрії, включаючи Канаду, Китай, Південну Корею, Австралію, Німеччину та інші.

Сучасний ринок розробників характеризується різноманітністю жанрів, платформ та моделей поширення ігор. Розробники створюють ігри для комп'ютерів, консолей, мобільних пристроїв, віртуальної реальності та інших платформ. Вони працюють над різними жанрами, включаючи екшн, пригоди, стратегії, рольові ігри, спортивні ігри та інші, щоб задовольнити різноманітні уподобання гравців.

У сучасній ситуації ринок розробників ігор став ще більш конкурентоспроможним. Розробники прагнуть привернути увагу гравців та досягти комерційного успіху, пропонуючи високоякісні ігрові продукти, інноваційні механіки, унікальний контент та захоплюючі історії. Разом з тим, розробники також стикаються з викликами, пов'язаними зі швидкими технологічними змінами, зміною смаків та уподобань гравців, а також необхідністю адаптуватися до нових платформ та ринкових трендів.

					КРБ.КІ.1.147–03.3.6	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дат		

В цілому, ринок розробників ігор є динамічним та різноманітним, пропонуючи безліч можливостей та викликів для тих, хто прагне створювати захоплюючі та успішні ігрові продукти.

У даному проекті, розроблений продукт – повноцінна гра у жанрі RPG від третьої особи, що коштує 100 грн, яка пропонує чудову графіку, гарну оптимізацію та різноманітні можливості. Порівнюючи її з іншими аналогами, можна виділити кілька переваг:

1. Ціна: гра відрізняється серед конкурентів привабливою та низькою, як для ігри такого жанру, ціною.
2. Оптимізація: Хороша оптимізація дозволяє грати з високою продуктивністю та дозволяє грі працювати навіть на старих пристроях.
3. Унікальний ігровий світ: Гра має свій унікальний ігровий світ з унікальними характеристиками, лором та стилем.
4. Система бою та геймплей: Гра має свою унікальну бойову систему та цікавий геймплей. Були розроблені різноманітні механіки, такі як торгівля, прокачка бойових навиків та інші.
5. Складність: Гра пропонує зростаючу складність, що дасть гравцям мотивацію ставати все сильнішими, щоб перемогти останнього боса гри.

Науково-технічний ефект включає дослідження і розробку, проектування інтерфейсу, алгоритми та штучний інтелект, оптимізацію продуктивності та тестування якості гри. Дипломний проект зробив внесок у сферу розробки ігор, представляючи нові технічні рішення та інноваційні підходи. Воно може містити розробку нової ігрової механіки, алгоритмів штучного інтелекту, графічних движків або інших технічних компонентів, які сприяють поліпшенню процесу створення та ігрового досвіду в жанрі *RPG*. Також гра може бути науковим дослідженням, яке узагальнює існуючі знання та пропонує нові підходи в галузі розробки ігор у жанрі *RPG*.

					КРБ.КІ.1.147–03.3.6	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дат		

Це може включати аналіз існуючих ігрових механік, дослідження впливу сюжету та персонажів на ігровий досвід, оцінку впливу аудіовізуальних ефектів на ігрову атмосферу та інші наукові аспекти, які сприяють глибшому розумінню та розвитку даного жанру ігор.

Соціальний ефект гри у жанрі *RPG* (рольових ігор), проявляється у наступних аспектах:

1. Ігрові ролі та самовираження: гра дозволяє гравцям приймати на себе різні ігрові ролі та розвивати своїх персонажів. Це дає можливість для самовираження, створення унікальної історії та формування віртуального "Я". Гравці можуть розвивати навички, прокачувати персонажів та приймати рішення, які відображають їх переваги та цінності.
2. Ігрова терапія та соціальна інтеграція: гра може бути використана як форма ігрової терапії, особливо для людей із соціальними чи емоційними труднощами. Ігровий світ та інтерактивність гри може допомогти покращити самооцінку, розвинути навички комунікації та допомогти у соціальній інтеграції.

Таким чином, гра має значний соціальний ефект, сприяючи розвитку мислення, самовираження та ігрової терапії. Вони надають гравцям можливість взаємодії, створення соціальних зв'язків та розвитку навичок, які можуть застосовуватись і в реальному світі.

Економічний ефект програми розглядається з точки зору потенційного ринку, потенційного доходу, розвитку навичок, залучення інвестицій та потенціалу співпраці. Створення цієї гри буде прибутковим на ринку, приносячи дохід за рахунок продажів і додаткових функцій. Ви також можете отримати важливі навички та шанс залучити фінансування або співпрацювати з іншими експертами у сфері створення ігор.

За проведеним орієнтовним розрахункам витрати на розробку складатимуть 75 000 грн (або 2031 доларів США). Детальніші статті витрат наведені в таблиці 6.1.

Таблиця 6.1

Статті витрат на розробку

Статті витрат	Кількість	Одиниця виміру	Ціна (грн)	Вартість (грн)
Зарплата студента	200 годин	година	80	16 000
Зарплата керівника	50 годин	година	150	7500
Оренда офісу	2 місяці	місяць	3 000	6 000
Електроенергія	2 місяці	місяць	1 500	3 000
Опалення	2 місяці	місяць	800	1 600
Інтернет	2 місяці	місяць	500	1 000
Комп'ютерна техніка	1	штука	20 000	20 000
Програмне забезпечення	1	штука	3 000	3 000
Реклама	—	—	—	400
Резервний фонд	—	—	—	16 500
Всього:	—	—	—	75 000

“Науково-технічну ефективність (НТЕ) результатів прикладних робіт визначають на основі показників науково-технічного рівня. Оцінка науково-технічної ефективності НДДКР відбувається на основі показника (ОНТЕ), який представляє собою ступінь досягнення максимально можливого рівня, значення якого дорівнює 1 (одиниці):

$$O_{\text{НТЕ}} = \frac{K^{\Phi}_{\text{НТЕ}}}{K^{\Pi}_{\text{НТЕ}}} \quad (6.1)$$

де $K^{\Phi}_{\text{НТЕ}}$ — показник (коефіцієнт) фактичного рівня науково-технічної ефективності;

$K^{\Pi}_{НТЕ}$ — показник (коефіцієнт) потенціально можливого рівня науково-технічної ефективності (дорівнює одиниці).

Значення показника $K^{\Phi}_{НТЕ}$ визначають на основі шкали експертних оцінок (табл. 6.2).”[26]

Таблиця 6.2

Шкала експертних оцінок для виміру рівня науково–технічної ефективності проектів

№	Групи показників	Характеристика показників	Інтервал рейтингового числа	Коефіцієнт значущості показників
1	Науково–технічний рівень	Перевищує кращі світові аналоги	10	0,35
		Відповідає світовому рівню	7 – 9	
		Нижче кращих світових аналогів	5 – 6	
		Перевищує кращі вітчизняні аналоги	3 – 4	
		Відповідає вітчизняному рівню	1 – 2	
		Нижче вітчизняного рівня	0	
2	Перспективність	Першочергова значущість	8 – 10	0,35
		Значущий	5 – 7	
		Корисний	1 – 4	
3	Потенційний масштаб практичного використання	Світовий ринок	10	0,20
		Галузі національної економіки	7 – 9	
		Галузь (регіон)	3 – 6	
		Окремі підприємства (об’єднання)	1 – 2	
4	Ступінь вірогідності досягнення позитивних результатів	Великий	10	0,10
		Середній	5–9	
		Малий	1–4	

“Примітка: об’єкт оцінки і аналог(и), які порівнюють за однаковими показниками, наведеними у співставленому вигляді відхилення в значеннях кожного з показників, мають бути однаковими для варіантів, що порівнюються.

Проведення оцінки

Визначають K^{Φ}_{HTE} на основі експертної оцінки науково-технічного рівня розробки.

З цією метою:

- розроблюють перелік специфічних показників, необхідних для виміру науково-технічного рівня розробки;
- формують групу аналогів, які реалізовані на світовому і вітчизняному ринках;
- здійснюють відповідні розрахунки для співставлення показників і визначення балів по табл. 6.2.

До числа специфічних показників відносять:

- **для нової техніки:** продуктивність, споживання інженерних ресурсів на виробітку одиниці продукції, потреба в робочих, які обслуговують обладнання, експлуатаційні витрати на одиницю продукції;
- **для нових матеріалів і речовин:** вміст корисних речовин для виробітки готової продукції, питома вага відходів у загальному обсязі переробленої сировини, вартість одиниці ... нового матеріалу;
- **для нових технологій:** якість виробленої продукції, енергоємність і трудомісткість продукції, собівартість одиниці продукції.
- з метою спрощення визначення K^{Φ}_{HTE} у табл. 6.3 не введено показника витрат на одиницю продукції.

					КРБ.КІ.1.147–03.3.6	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дат		

- для нових матеріалів і речовин: вміст корисних речовин для виробітки готової продукції, питома вага відходів у загальному обсязі переробленої сировини, вартість одиниці нового матеріалу;
- для нових технологій: якість виробленої продукції, енергоємність і трудомісткість продукції, собівартість одиниці продукції.

З метою спрощення визначення K^{Φ}_{HTE} у табл. 6.3 не введено показника витрат на одиницю продукції.”[26]

Таблиця 6.3

Порівняльні показники для виконання оцінки НТЕ

ПОКАЗНИКИ	Варіанти технології	
	розробленої	TheElderScrolls
Рівень новизни	регіональний	світовий
Якість продукції	вища	вища
Графіка	висока	висока
Геймплей	традиційний	традиційний
Механіка	середня	середня
Дизайн	середній	високий
Звук	середній	високоякісний

“На основі співставлення даних таблиці встановлюють бали по характеристиках чотирьох груп і на цій основі розраховують значення інтегрального показника НТЕ:

$$HTE = \sum B_i \times K_i^3 \quad (6.2)$$

де $i = 1 \div 4$,

B_i – бали (рейтингове число),

K – коефіцієнт значущості показників.

Рівень науково–технічної ефективності НДДКР розраховано на основі наведених даних прикладу (табл. 6.4).”[26]

Таблиця 6.4

Експертна оцінка і розрахунок величини інтегрального показника НТЕ

№	Групи показників	Рейтинг експертів			Середня за експертними оцінками	НТЕ
		1	2	3		
1	Науково–технічний рівень	8	7	8	7,66	2,68 (7,66 x 0,35)
2	Перспективність	7	7	6	6,66	2,33 (6,66 x 0,35)
3	Потенційний масштаб практичного використання	4	4	5	4,33	0,866 (4,33 x 0,20)
4	Ступінь вірогідності досягнення позитивних результатів	9	7	8	8	0,8 (8 x 0,10)
В С Ь О Г О						6,676

$$\text{“НТЕ} = 7,66 \cdot 0,35 + 6,66 \cdot 0,35 + 4,33 \cdot 0,2 + 8 \cdot 0,1 = 2,68 + 2,33 + 0,866 + 0,8 = 6,676$$

Отриманий результат слід порівняти з максимально можливим значенням, яке дорівнює 10 балам ($10 \cdot 0,35 + 10 \cdot 0,35 + 10 \cdot 0,2 + 10 \cdot 0,1$).

Отже, оцінка рівня НТЕ може бути зроблена за допомогою інтегрального коефіцієнта оцінки НТЕ ($K_{НТЕ}$):

$$K_{НТЕ} = \frac{НТЕ}{10} \times 100 \% \quad (6.3)$$

На основі даних табл. 3.3 можна дійти до висновку, що $K_{НТЕ}$ відповідає 66,7%, тобто:

$$\frac{6,676}{10} \times 100\% = 66,7 \%$$

					КРБ.КІ.1.147–03.3.6	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дат		

В тому випадку, коли значення $K_{НТЕ}$ перевищує середнє значення, яке дорівнює 5,0, має бути зроблено висновок про достатній рівень НТЕ:

1. Цілком достатній 5,0 – 6,0;
2. Достатній 6,1 – 8,0;
3. Достатньо високий 8,1 – 9,0;
4. Високий 9,1 – 10.

Таким чином, рівень НТЕ технології можна визнати достатнім. Отже, розроблену технологію пропонується впроваджувати у виробництво.”[26]

					КРБ.КІ.1.147–03.3.6	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дат		

РОЗДІЛ 7

ОХОРОНА ПРАЦІ

7.1 Фактори що впливають на працю

Сучасний розвиток технічного та технологічного стану навчальних закладах передбачає постійну автоматизацію та оптимізацію виробничих процесів.. Через масовий характер робіт, що виконуються студентами за допомогою комп'ютера, законодавством України чітко врегульовано норми та вимоги до використання комп'ютерної техніки, безпосередньо й охорона праці при роботі з комп'ютером.

Працюючи з комп'ютером, розробник потрапляє під вплив різноманітних факторів виробничого середовища та трудового процесу. Розглянемо ці фактори та їх дію на організм людини.

Хімічні речовини у повітрі робочої зони. Хімічний фактор виробничого середовища не є характерним для розробника, який використовує у своїй роботі ПК. Проте, за певних умов він може мати місце. Так, внаслідок інтенсивного використання копіювальної техніки безпосередньо у робочій зоні програмістів може бути присутнім озон (хімічна речовина 1 класу небезпеки, гостроспрямованої дії на організм людини, граничнодопустима концентрація (ГДК) становить 0,1 мг/м³). За результатами проведених гігієнічних досліджень вміст озону, як правило, не перевищує ГДК.

Джерелом хімічних речовин у повітрі робочої зони можуть бути матеріали, які використані для ремонту та оздоблення приміщень. Серед них синтетичні матеріали для підлоги (ворсоніт, ковролін, лінолеум), полімерні матеріали для оздоблення стін, тощо. Найпоширенішою речовиною, яка може виділятися з полімерних матеріалів, є формальдегід. Вміст формальдегіду у повітрі робочої зони може бути на рівні 0,5 ГДК.

					КРБ.КІ.1.147–03.3.6	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дат		

Шум. Нормативним значенням еквівалентного рівня звуку для програмістів є 50 дБ.

Шум погіршує умови праці здійснюючи шкідливу дію на організм людини. Працюючі в умовах тривалої шумової дії випробовують дратівливість, головні болі, запаморочення, зниження пам'яті, підвищену стомлюваність, зниження апетиту, біль у вухах і т. д. Такі порушення в роботі ряду органів і систем організму людини можуть викликати негативні зміни в емоційному стані людини аж до стресових. Під впливом шуму знижується концентрація уваги, порушуються фізіологічні функції, з'являється втома у зв'язку з підвищеними енергетичними витратами і нервово-психічним напруженням, погіршується мовна комутація. Все це знижує працездатність людини і його продуктивність, якість і безпеку праці.

Неіонізуюче випромінювання. Як правило, роботу програміста пов'язують зі шкідливим впливом електромагнітних полів. Це мало місце під час використання моніторів з електронно-променевою трубкою. У сучасних комп'ютерах використовуються рідкокристалічні монітори, тому вплив електромагнітного поля від монітора практично відсутній. Наявним залишається електростатичне поле на поверхні монітора та поверхні клавіатури.

Нормативне значення напруженості електростатичного поля становить 150 В/см. Напруженість електростатичного поля, визначена на поверхні монітора та поверхні клавіатури, як правило, не перевищує нормативного значення.

Рівень напруженості електростатичного поля залежить від вологості повітря, регулярного прибирання робочого місця (усунення запиленості).

Мікроклімат у приміщенні. У теплий період року температура повітря має бути у межах 22–25 °С, швидкість руху повітря – до 0,1 м/с, відносна вологість повітря – 40-60%.

					КРБ.КІ.1.147–03.3.6	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дат		

У холодний період року температура повітря може коливатися у межах 21-24 °С, швидкість руху повітря – до 0,1 м с, вологість повітря – 40-60%.

Для робочої зони розробника, який працює в приміщенні з централізованим повітрозабезпеченням, як влітку, так і взимку характерні значення відносної вологості повітря на рівні 20-40%, знижена або підвищена швидкість руху (рухливість) повітря робочої зони.

Як правило, температура повітря перевищує нормативні значення протягом року через постійне нагрівання деталей ПК, знижену вологість.

Саме температура та вологість повітря впливають на загальне самопочуття, стан слизових оболонок очей, верхніх дихальних шляхів та шкіри користувачів ПК. Низька вологість збільшує час «зависання» пилу в повітрі приміщення.

До характеристик повітря робочої зони розробників віднесені такі, як вміст позитивних та негативних іонів у повітрі.

Як правило, у приміщенні з розвиненою системою припливно-витяжної вентиляції та/або з кондиціонерами виявляються порушення норм аероіонного складу повітря, при цьому концентрація корисних для організму негативно заряджених аероіонів може бути в 10-50 разів нижче норми, а концентрація шкідливих позитивних аероіонів значно перевищувати норму. Ось чому необхідно, щоб до робочого місця з комп'ютером надходило свіже повітря. Рекомендується робити вологе прибирання не рідше одного разу на день. Обов'язково слід протирати поверхню столу серветкою з антистатиком, оскільки позитивні іони мають властивість з'єднуватися з частинкам пилу.

У сучасних спорудах з центральною вентиляцією і відсутністю можливості провітрювання приміщень забезпечення відповідної аероіонізації є досить проблематичним.

					КРБ.КІ.1.147–03.3.6	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дат		

Слід додати, що серед програмістів відмічається високий рівень захворюваності з тимчасовою втратою працездатності, з них захворювання органів дихання складають 45-60%. За даними медичних установ, серед користувачів ПК значно поширені захворювання ЛОР-органів (до 25% користувачів), при цьому переважають ураження верхніх дихальних шляхів, а саме: хронічні катаральні фарингіти (15,7%). Слід зазначити високу частоту розвитку алергічних ринітів (1,3%), що у декілька разів вища за середню по Україні (0,28%).

Робоча поза. Цей фактор трудового процесу залежить від правильної організації робочого місця для забезпечення зручної робочої пози, адже «закам'яніле» положення шкідливо впливає на хребтово-м'язову систему.

Стіл має бути просторим, із спеціальною підставкою для ніг, а робочий стілець мати відрегульовану висоту, певний кут нахилу сидіння і спинки.

Під час роботи за комп'ютером людина сидить кілька годин поспіль у незручному положенні. Це не тільки викликає загальну втому, а й може призвести до розвитку остеохондрозу різних ділянок хребта – шийного, грудного, попереково-крижового.

Неправильне положення рук під час друкування на клавіатурі призводить до хронічних захворювань кисті. Тому клавіатура має розташовуватися на відстані 10-15 см від краю столу.

Щоб робота за комп'ютером не шкодила здоров'ю, необхідно постійно стежити за своєю поставою. Правильна постава максимально розвантажує м'язи і дозволяє працювати довше, менше втомлюючись.

Напруженість праці. Напруженість праці працюючого з ПК пов'язана, перш за все, з напруженістю очей, тривалістю зосередження уваги, яка може становити понад 75% часу. Тому саме очі найбільш страждають під час роботи з комп'ютером.

					КРБ.КІ.1.147–03.3.6	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дат		

Велике значення при роботі за комп'ютером мають такі речі як: відстань до екрана, шрифт, розмір тексту на моніторі, наявність або відсутність мерехтіння, яскравість екрана, освітлення робочого місця, наявність перерв у роботі. Саме ігнорування таких простих на перший погляд речей у значній мірі призводить до погіршення зору та хвороб очей.

7.2 Загальні умови безпеки праці з комп'ютерною технікою

Умови праці користувачів ПК мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, затверджених постановою Головного державного санітарного лікаря України від 10.12.1998 № 7 (далі — ДСанПіН 3.3.2.007-98), та Правил № 65.

Сьогодні ж у більшості приміщень встановлено нові сучасні комп'ютери з рідкокристалічними моніторами, для яких не прописано гігієнічні вимоги у ДСанПіН 3.3.2.007-98. Водночас вимоги ДСанПіН 3.3.2.007-98 до виробничих приміщень, параметрів виробничого середовища приміщень, організації і обладнання робочих місць, режимів праці та відпочинку є актуальними під час робіт з будь-якими комп'ютерами та моніторами.

Відповідно до пунктів 4.1, 4.2 Правил № 65 ПК під час експлуатації мають відповідати вимогам стандартів, вказаних Правил і пройти державну санітарно-епідеміологічну експертизу згідно з Порядком проведення державної санітарно-епідеміологічної експертизи, затвердженим наказом МОЗ України від 09.10.2000 № 247 (у редакції наказу МОЗ України від 14.03.2006 № 120).

ПК закордонного виробництва мають відповідати вимогам національних стандартів держав-виробників і мати відповідну позначку на корпусі, у паспорті або в іншій експлуатаційній документації.

					КРБ.КІ.1.147-03.3.6	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дат		

Зазначені вище нормативні документи – обов’язкові для виконання Вони визначають критерії безпечного використання комп’ютерної техніки та призначені для запобігання несприятливої дії на користувача шкідливих факторів, що супроводжують роботу з візуальними дисплейними терміналами.

Керуючись зазначеними вище нормативними документами можна дійти висновку, що для забезпечення комфортної роботи розробника необхідно звернути увагу на такі основні вимоги:

- стан виробничих приміщень;
- характеристики та стан техніки (ПК, монітори, принтери, наявність заземлення тощо);
- організація робочого місця;
- дотримання режимів праці та відпочинку;
- моніторинг стану здоров’я розробника (стан органів зору, показники захворювань з тимчасовою втратою працездатності, ін.);
- профілактичні заходи.

7.3 Вимоги до організації робочого місця

- при розміщенні декількох робочих місць в одному приміщенні мінімальна площа для одного робочого місця складає 6 м²;
- екран монітора має бути розміщений на оптимальній відстані від очей розробника (60–70 см), але не ближче 50 см;
- екран має бути розташований для забезпечення комфортного зорового спостереження у вертикальній площині під кутом + 30° до нормальної лінії погляду розробника;
- під час роботи необхідно робити перерви для розвантаження очей.

					КРБ.КІ.1.147–03.3.6	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дат		

Режим праці та відпочинку:

У процесі роботи з комп'ютером необхідно дотримуватися правильного режиму праці та відпочинку. Інакше у програміста спостерігається незадоволеність роботою, головний біль, роздратування, порушення сну, втома і больові відчуття в очах, попереку, у ділянці шиї та рук. Для збереження здоров'я розробника, запобігання професійним захворюванням і підтримки працездатності слід передбачати внутрішньо змінні регламентовані перерви для відпочинку, для розробника програми – 15 хв. через кожну годину роботи за комп'ютером.

Висновок:

Під час роботи з ПК на людину мають вплив безліч небезпечних чинників. Щоб попередити або зменшити цей вплив, потрібно дотримуватися правил використання ПК. Це є дуже важливим питанням, тому що правильне використання комп'ютерної техніки дозволяє зменшити шкідливий вплив на людину, запобігти погіршенню її самопочуття та уникнути нещасних випадків.

					КРБ.КІ.1.147–03.3.6	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дат		

ЗАГАЛЬНІ ВИСНОВКИ

В даній дипломній роботі було створена комп'ютерна гра за допомогою двигуна *Unity*. Гра написана на ігровому двигуні *Unity* та середовищі програмування *VisualStudio*.

Описаний в даній роботі програмний продукт включає наступні можливості:

- початок гри;
- налаштування персонажу;
- бойовий процес;
- вибір вже створених персонажів;
- взаємодія з об'єктами на локаціях;
- взаємодія з *NPC*.

Для досягнення поставлених цілей, а також вирішення задачі, що склалася була зроблена наступна робота :

1. Ознайомлення з існуючими аналогами та вивчення предметної області.
2. Зібрання необхідного матеріалу – знань, що потрібні для створення подібного програмного забезпечення.
3. Вибір програмних засобів.
4. Створення програмного додатку зі зручним інтерфейсом користувача.

В подальшому програмний продукт можна поширити – а саме випустити нові квести, ворогів або локації для проходження гри.

З економічної точки зору впровадження товару є вигідним.

У розділі, присвяченому охороні праці, даної дипломної роботи були викладені вимоги до робочого місця користувача. Створені відповідно до цих вимог умови праці повинні забезпечити комфортну і безпечну роботу. Дотримання умов, що визначають оптимальну організацію робочого місця дозволяє зберегти хорошу працездатність в перебігу всього робочого дня.

					КРБ.КІ.1.147–03.3.6	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дат		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Unity* – Вікіпедія [Електронний ресурс]. – Режим доступу:
<https://uk.wikipedia.org/wiki/Unity>
2. *Visual Studio* – Вікіпедія [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio
3. *The Elder Scrolls* – Вікіпедія [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/The_Elder_Scrolls
4. *Fallout* – Вікіпедія [Електронний ресурс]. – Режим доступу:
<https://uk.wikipedia.org/wiki/Fallout>
5. *Dark Souls* – Вікіпедія [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/Dark_Souls
6. *RPG* – Вікіпедія [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/Рольова_відеогра
7. *DFD* діаграма – Вікіпедія [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/Діаграма_потоків_даних
8. *SADT* модель – Вікіпедія [Електронний ресурс]. – Режим доступу:
<https://uk.wikipedia.org/wiki/SADT>
9. *IDEF3* – Вікіпедія [Електронний ресурс]. – Режим доступу:
<https://uk.wikipedia.org/wiki/IDEF>
10. *Use-Case* діаграма – Вікіпедія [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/Сценарій_використання
11. *Jesse Schell. The Art of Game Design: A Book of Lenses.* – 1-е изд. CRC Press, 2008. – 520 с.
12. *Richard Bartle. Designing Virtual Worlds.* – 1-е изд. New Riders, 2003. – 768с.
13. *Sebastian Deterding., José Zagal. Role-Playing Game Studies: Transmedia Foundations.* – 1-е изд. Routledge, 2018. – 494 с.

					КРБ.КІ.1.147–03.3.6	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дат		

14. Don Norman. *Role-Playing Game Studies: Transmedia Foundations*. Basic Books, 2013. – 368 с.
15. Flint Dille., John Platten. *The Ultimate Guide to Video Game Writing and Design*. Lone Eagle, 2008. – 272 с.
16. Robert Nystrom. *Game Programming Patterns*. Genever Benning, 2014. – 354с.
17. Tynan Sylvester. *Designing Games: A Guide to Engineering Experiences*. – 1-е изд. O'Reilly Media, 2013. – 413 с.
18. Jacob Habgood., Mark Overmars., Phil Wilson. *The Game Maker's Apprentice: Game Development for Beginners*. – 1-е изд. Apress, 2006. – 336 с.
19. Jane Frank *Role-Playing Game and Collectible Card Game Artists: A Biographical Dictionary*. McFarland, 2012. – 284 с.
20. Morgan McGuire., Odest Jenkins. *Creating Games: Mechanics, Content, and Technology*. – 1-е изд. A K Peters/CRC Press, 2008. – 500 с.
21. Tracy Fullerton. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. – 2-е изд. Morgan Kaufmann, 2008. – 496 с.
22. Kevin Saunders., Jeannie Novak. *Game Development Essentials: Game Interface Design*. – 2-е изд. Cengage Learning, 2012. – 320 с.
23. Joe Hocking. *Unity in Action: Multiplatform Game Development in C# with Unity 5*. – 1-е изд. Manning Publications, 2015. – 352 с.
24. Jeannie Novak. *Game Development Essentials: An Introduction*. – 3-е изд. Cengage Learning, 2011. – 510 с.
25. Scott Rogers. *Level Up! The Guide to Great Video Game Design*. – 2-е изд. Wiley, 2014. – 560 с.
26. Басюркіна Н.Й., Свистун Т.В. Методичні вказівки до оцінки науково-технічної ефективності розробки нової технології, нового обладнання та інших інновацій. Для студентів всіх спеціальностей СВО «бакалавр» і «магістр» денної і заочної форм навчання.. – Одеса.: ОНТУ, 2022. – 18 с.