

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

ПРОЄКТУВАННЯ ТА РОЗРОБКА
ІНТЕРАКТИВНОЇ НАВЧАЛЬНОЇ ГРИ «ПАЗЛ»

(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-19-2

Морозов М. М.

(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Сакало Є.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

(підпис)

Кобилін О.А.

(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)Кафедра Інформатики
(повна назва)Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« ____ » _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Морозову Миколі Миколайовичу
(прізвище, ім'я, по батькові)1. Тема роботи Проектування та розробка інтерактивної навчальної гри «Пазл»

затверджена наказом університету від 15 травня 2023 року № 474 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 29 травня 2023 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, мова програмування Java, середовище розробки IntelliJ IDEA, бібліотека для роботи графічного інтерфейсу Desktop-застосунку JavaFX.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Аналіз методів розробки Desktop-застосунків.

2. Математичні моделі для зберігання та обробки даних.

3. Програмна реалізація Desktop-застосунку.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми розробки мобільних Desktop-застосунків, постановка задачі, методи розробки Desktop-застосунків, архітектурні рішення для реалізації, вибір методу роботи з даними, програмна реалізація.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	10.04.2023	
2	Аналіз завдання, підбір літератури	11.04.23-17.04.23	
3	Аналіз літератури з досліджуваної проблеми	18.04.23-20.04.23	
4	Аналіз засобів розробки Desktop-застосунків	21.04.23-30.04.23	
5	Проектування застосунку	01.05.23-14.05.23	
6	Програмна реалізація	15.05.23-23.05.23	
7	Оформлення пояснювальної записки	24.05.23-30.05.23	
8	Перевірка на плагіат	04.06.23	
9	Рецензування	05.06.23	
10	Підготовка презентації та доповіді	05.06.23	
11	Занесення роботи в електронний архів	05.06.23	
12	Попередній захист кваліфікаційної роботи	07.06.23	

Дата видачі завдання 10 квітня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Сакало Є.С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 58 с., 5 табл., 10 рис., 30 джерел.

JAVA, ВІДЕОІГРИ, НАВЧАННЯ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ, ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, РОЗРОБКА ІГОР.

Об'єктом роботи є реалізація гри «Пазл» для вивчення термінів на ІТ тематику.

Метою цієї роботи є створення застосунку з використанням бази даних для навчання. В тому числі і забезпечення простоти та легкості використання програми.

Для досягнення цієї мети будуть використані новітні методи для роботи з текстовими та графічними даними. Застосовано різноманітні бібліотеки для покращення роботи застосунку.

У результаті виконання реалізовано програмний застосунок для навчання.

JAVA, VIDEO GAMES, TRAINING, INTELLECTUAL ANALYSIS, ОБ'ЄКТНО-ORIENTED PROGRAMMING, GAME DEVELOPMENT.

The object of the work is the implementation of the game "Puzzle" to study the terms on IT topics.

The purpose of this work is to create an application using a database for training. Including ensuring the simplicity and ease of use of the application.

To achieve this, the latest methods for working with text and graphic data will be used. Various libraries are used to improve the operation of the application.

As a result of execution, a software application for learning is implemented.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ	9
1 Аналіз сучасних відеоігор.....	11
1.1 Розвиток відеоігор та вплив на сучасне суспільство	11
1.2 Жанри відеоігор, їх застосування та користь в сучасному житті ...	12
1.2.1 Екшн	12
1.2.1.1 Шутери	12
1.2.1.2 Екшн-пригоди	13
1.2.1.3 Платформери.....	13
1.2.1.4 Слешери.....	13
1.2.2 Рольові ігри (RPG).....	13
1.2.3 Пригодницькі ігри	14
1.2.4 Стратегії	15
1.2.5 Симулятори.....	16
1.2.6 Мультиплеєрні ігри	17
1.2.6.1 Кооперативні ігри	17
1.2.6.2 Змагальні ігри.....	18
1.2.6.3 ММО (Massively Multiplayer Online)	18
1.2.7 Головоломки	18
1.2.7.1 Розриваючі головоломки	19
1.2.7.2 Фізичні головоломки	19
1.2.7.3 Логічні головоломки.....	19
1.2.7.4 Механічні головоломки.....	20
1.2.8 Інді-ігри	20
1.2.9 Музичні ігри.....	21
1.2.10 VR-ігри	22
1.3 Постанова задачі.....	23
2 Алгоритми та методи створення відеоігор	24

2.1	Загальний огляд відеоігор	24
2.1.1	Графічні алгоритми	24
2.1.1.1	Алгоритм растеризації (Rasterization Algorithm).....	24
2.1.1.2	Алгоритм затінення (Shading Algorithm).....	25
2.1.1.3	Алгоритм текстуровання	25
2.1.1.4	Алгоритм антиаліасингу (Antialiasing Algorithm)	25
2.1.1.5	Алгоритм відлуння (Ray Tracing Algorithm)	26
2.1.2	Звук.....	27
2.1.3	Фізика	28
2.1.3.1	Моделювання колізій.....	28
2.1.3.2	Моделювання руху	29
2.1.3.3	Моделювання гравітації	29
2.1.3.4	Моделювання фізичних властивостей матеріалів	30
2.1.4	Штучний інтелект (ШІ)	30
2.1.4.1	Прийняття рішень	31
2.1.4.2	Пошук шляху	31
2.1.4.3	Стратегії та тактики	31
2.1.4.4	Співпраця та комунікація	32
2.1.5	Управління геймплеєм	32
2.2	Методи створення відеоігор	33
2.2.1	Використання готових ігрових двигунів	34
2.2.2	Використання спеціалізованих мов та інструментів	35
2.2.3	Графічне моделювання та анімація	36
2.2.4	Розробка інді-ігор	36
3	Програмна реалізація навчальної гри «Пазл».....	38
3.1	Обґрунтування вибору середовища, мови та платформи для програмної реалізації	38
3.1.1	Обґрунтування вибору середовища.....	38
3.1.2	Обґрунтування вибору платформи	40
3.1.3	Обґрунтування вибору мови програмування	41
3.2	Програмна реалізація	42

3.2.1	Опис задумки програми	42
3.2.2	Бібліотеки класів.....	43
3.2.3	Використання ООП	44
3.2.4	Класи у програмі.....	45
3.2.4.1	Класи з графічної бібліотеки SWING	45
3.2.4.2	Поля класу	46
3.2.4.3	Методи класу.....	47
3.3	Інструкція користувача	48
	Висновки.....	54
	Перелік джерел посилання	55

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

IT – інформаційні технології

RPG – Role-Playing Game (рольова гра)

RTS – Real Time Strategy (стратегія у реальному часі)

TBS – Turn Based Strategy (покрокова стратегія)

GS – Global Strategy (глобальна стратегія)

MMO – Massively Multiplayer Online (масова розрахована на багато користувачів онлайн-гра)

VR – Virtual Reality (віртуальна реальність)

ШІ – штучний інтелект

NPC – Non-Player Character (неігровий персонаж)

SAT – Separating Axis Theorem (теорема про розділову вісь)

ОС – операційна система

SDK – Software Development Kit (набір для розробки програмного забезпечення)

ВСТУП

Сучасний світ заснований на постійному розвитку технологій, які впливають на різні сфери нашого життя. Однією з таких сфер є освіта, яка постійно шукає нові підходи та інструменти для покращення навчання. Одним з найцікавіших та перспективних напрямків є використання комп'ютерних ігор як засобу навчання. Вступивши в епоху цифрових технологій, комп'ютерні ігри вийшли за межі розваги та стали потужним інструментом для освіти та навчання.

Ця тема має все більшу актуальність, оскільки комп'ютерні ігри стають все більш доступними та популярними серед дітей, підлітків та навіть дорослих. Сучасні ігри вражають своєю реалістичністю, візуальними ефектами та захоплюючим сюжетом. Вони здатні створити іммерсивне середовище, в якому гравець може відчувати себе активним учасником подій, зайнятися вирішенням складних завдань та розв'язуванням головоломок.

Одним з основних переваг навчання через комп'ютерні ігри є те, що вони створюють сприятливу та мотивуючу навчальну атмосферу. У порівнянні з традиційними методами навчання, які часто бувають монотонними та нудними, комп'ютерні ігри дозволяють створити цікаве та захоплююче середовище, в якому студенти мають можливість вчитися, експериментувати та розвивати свої навички в непримушеній формі. Граючись, вони отримують велику кількість практичного досвіду та можуть застосовувати отримані знання в реальних ситуаціях.

Комп'ютерні ігри також допомагають розвивати ряд ключових навичок. Наприклад, вони сприяють розвитку логічного мислення, стратегічного планування, проблемного та критичного мислення. Гравець повинен аналізувати ситуацію, приймати рішення та оцінювати наслідки своїх дій. Вирішуючи складні головоломки та завдання, вони навчаються швидко думати та знаходити ефективні рішення. Крім того, ігри сприяють розвитку

комунікаційних навичок, оскільки вони можуть бути груповими, вимагати співпраці з іншими гравцями та обміну інформацією.

Однією з головних переваг навчання через комп'ютерні ігри є індивідуалізація навчання. Кожен гравець може вибрати рівень складності, тему та тип завдань, що відповідають його потребам та інтересам. Це дозволяє кожному студенту розвиватися у власному темпі та отримувати індивідуальну підтримку та зворотний зв'язок. Крім того, ігри можуть бути адаптовані до різних стилів навчання та навчальних потреб студентів.

Однак, варто зазначити, що навчання через комп'ютерні ігри також має свої виклики та обмеження. Наприклад, недостатнє контролювання годин ігор може призвести до зловживання та знехтування інших аспектів життя. Також, не всі ігри є належними засобами навчання, тому важливо обирати якісні та педагогічно обґрунтовані ігри.

У даній роботі будуть розглянуті різні аспекти навчання через комп'ютерні ігри, їх переваги та виклики, а також стратегії та підходи до ефективного створення таких ігор.

1 АНАЛІЗ СУЧАСНИХ ВІДЕОІГОР

1.1 Розвиток відеоігор та вплив на сучасне суспільство

Аналіз сучасних відеоігор є важливим аспектом вивчення цього популярного формату розваг. Відеоігри стали невід'ємною частиною нашого життя, а їх розвиток і прогрес у технологіях спричинили появу різноманітних жанрів, стилів та концепцій гри.

Один з основних аспектів аналізу сучасних відеоігор полягає у вивченні їхньої геймплейної механіки. Кожна гра має свої унікальні правила та способи взаємодії з гравцем [1-3]. Дослідження геймплею відеоігор допомагає зрозуміти, як гравці взаємодіють з віртуальним світом, які механіки спонукають їх до подальшої гри та які фактори впливають на задоволення від гри.

Галузь відеоігор досить активно розвивається у наш час і має досить великий вплив на сучасний світ. Це можна пояснити декількома факторами:

- розвиток технологій. Відеоігри виступають каталізатором для технологічного прогресу. Їх потужність та графічна якість вимагають постійного розвитку апаратних та програмних рішень. Це спонукає до нових інновацій у галузі комп'ютерної графіки, штучного інтелекту, віртуальної реальності та інших технологій;

- економічний вплив. Відеоігри є великою індустрією, яка генерує значні прибутки та створює робочі місця. Вона залучає розробників програмного забезпечення, художників, дизайнерів, музикантів, акторів і багатьох інших професіоналів. Таким чином, вона сприяє економічному зростанню та стимулює інновації в інших галузях;

- розвиток навичок. Відеоігри можуть допомагати в розвитку різних навичок у гравців. Наприклад, стратегічні ігри сприяють розвитку планування, креативного мислення та прийняття рішень. Головоломки та логічні ігри

вдосконалюють просторову уяву, логічне мислення та проблемне мислення. Спортивні ігри сприяють фізичній активності та здоров'ю.

1.2 Жанри відеоігор, їх застосування та користь в сучасному житті

1.2.1 Екшн

Жанр ігор екшн є одним з найпопулярніших і захоплюючих жанрів відеоігор. Він привертає увагу гравців завдяки своєму швидкому темпу, захоплюючим бойовим сценам та активним діям.

У іграх екшн гравці стикаються з різноманітними викликами, такими як битви з ворогами, стрілянини, виконання акробатичних рухів, гонки та інші динамічні події. Основний акцент робиться на фізичній взаємодії з оточуючим світом, що робить геймплей більш захоплюючим та захисним.

Ігри жанру екшн можуть сприяти розвитку навичок реакції та координації, розвитку стратегічного мислення та виробленню стресостійкості та керування емоціями.

У жанрі екшн існує багато піджанрів.

1.2.1.1 Шутери

Шутери (в першій або третій особі): Цей піджанр зосереджений на веденні бою з використанням різноманітної вогнепальної зброї.

Гравцю доводиться розраховувати кути обстрілу, уникати ворожих атак та реагувати швидко на зміни в грі. Шутери можуть бути як сольовими, так і мультиплеєрними, де гравці змагаються між собою або співпрацюють у команді.

1.2.1.2 Екшн-пригоди

Цей піджанр поєднує елементи екшну та пригодницьких ігор.

Гравцеві потрібно виконувати завдання, розгадувати головоломки, взаємодіяти з персонажами та досліджувати великі світи гри.

Часто в таких іграх є розвинута сюжетна лінія та можливість впливу гравця на події.

1.2.1.3 Платформери

Цей піджанр вимагає від гравця переміщення свого персонажа через різні платформи та перешкоди.

Вони часто мають складні рівні з високим рівнем виклику, де гравцю потрібно виявляти точність, швидкість та координацію рухів.

1.2.1.4 Слешери

Цей піджанр фокусується на силових боях, де гравцю доводиться використовувати різноманітні удари, комбо та реакції, щоб перемогти супротивника.

Слешери часто мають видовищні бойові системи та спеціальні ефекти.

1.2.2 Рольові ігри (RPG)

Рольові ігри (RPG) – це жанр відеоігор, в якому гравець приймає на себе роль фіктивного персонажа і взаємодіє з вигаданим світом. Основною

особливістю RPG є можливість розвитку та прокачки персонажа, вирішення завдань, взаємодія з іншими персонажами та вплив на сюжет гри.

У рольових іграх гравець зазвичай має можливість налаштувати свого персонажа за своїми уподобаннями. Він може вибрати між різними класами персонажів, збирати та вдосконалювати різну зброю, броню та предмети, розвивати навички та характеристики персонажа. Гравці також можуть вирішувати, як розподілити витрати ресурсів і які навички розвивати, що дозволяє індивідуалізувати свого персонажа.

У рольових іграх часто присутні розгалужені сюжетні лінії з численними квестами, завданнями та подіями, які гравець може впливати своїми виборами. Він може взаємодіяти з іншими персонажами, приймати рішення, які впливають на подальший розвиток історії. Це створює почуття свободи та відповідальності за власні дії та наслідки.

Ще одна важлива складова RPG – бойова система. Вона може бути заснована на ході або в реальному часі. Гравцю доводиться зустрічатися з різними ворогами та вести битви з ними, використовуючи різні вміння та стратегії.

Основна мета рольових ігор – надати гравцеві можливість відчувати себе частиною фантастичного світу, розповісти цікаву історію та надати можливість впливати на події. Вони також сприяють розвитку стратегічного мислення, прийняттю рішень, соціальним взаємодіям (якщо вони мають мультиплеєрний режим) та можуть навіть стимулювати креативність гравця при створенні власних історій та персонажів.

1.2.3 Пригодницькі ігри

Пригодницькі ігри є одним з захоплюючих жанрів відеоігор, які пропонують гравцеві захоплюючий наратив та можливість досліджувати нові

світи та сюжети. Вони спроектовані таким чином, щоб гравці могли відчутися себе головними героями та долучитися до незабутніх пригод.

Особливість пригодницьких ігор полягає в їхньому спрямованому на сюжет геймплеї. Гравець відправляється у подорож, досліджуючи віртуальні світи, взаємодіючи з персонажами та розв'язуючи головоломки, щоб рухатися вперед у сюжетній лінії. Ці ігри зазвичай пропонують заплутані завдання, викликані головоломки та неочікувані повороти, що додають їм хвилюючості та напруженості.

У пригодницьких іграх гравці часто зустрічаються зі змінною інтерактивністю. Вони мають можливість взаємодіяти з персонажами, знаходити та використовувати предмети, розгадувати головоломки та вирішувати дилеми, що впливають на розвиток історії. Гравці можуть вибирати свої дії та напрямок розвитку сюжету, що додає імпровізацію та особистий вплив на гру.

Пригодницькі ігри також славляться за свою високу якість наративу. Вони надають гравцеві можливість погрузитися у захопливий сюжет, відчутися емоції героїв та пережити неймовірні пригоди. Часто вони включають діалоги, кінематографічні сцени та розкривають глибокий історичний або фантастичний світ, що робить гру більш цікавою та захоплюючою.

1.2.4 Стратегії

Жанр стратегій відеоігор вимагає від гравця стратегічного мислення, планування та управління ресурсами. Цей жанр пропонує гравцям можливість керувати великими територіями, розвивати міста, будувати та управляти базами, формувати армії та вести військові операції. Він може включати різні піджанри, такі як стратегії в реальному часі (Real-Time Strategy, RTS), пошагові стратегії (Turn-Based Strategy), глобальні стратегії (Grand Strategy) та інші.

У стратегічних іграх гравець зазвичай стикається зі складними вирішеннями, які впливають на подальший розвиток гри. Він повинен враховувати різні фактори, такі як економічні обмеження, географічні умови, рівень технологій та ворожі дії. Головною метою гравця може бути досягнення світової домінанти, здобуття територій, розвиток культури чи наукового прогресу.

У стратегічних іграх гравець часто змушений довго планувати свої дії, аналізувати ситуацію, приймати стратегічні рішення та використовувати тактику. Цей жанр сприяє розвитку навичок керування ресурсами, управління часом, прийняття рішень та розв'язання проблем.

У стратегічних іграх гравець може випробовувати різні стратегії, експериментувати та вивчати нові підходи. Вони часто надають велику кількість варіативності та глибину геймплею, що дозволяє гравцям повністю зануритися в світ гри та розвивати власні стратегічні навички.

1.2.5 Симулятори

Жанр симуляторів відеоігор відображає реалістичне моделювання різних аспектів реального життя або конкретних ситуацій. Він надає гравцю можливість відчувати себе в ролі реального персонажа, керувати певними процесами та вирішувати завдання, які відповідають реальному життю.

Симулятори можуть охоплювати широкий спектр тематик, включаючи авіацію, автомобілі, будівництво, господарство, риболовлю, спорт, медицину та інше. Гравці мають можливість відчувати атмосферу та взаємодіяти зі світом, який відтворює реалістичні аспекти відповідної галузі.

Однією з ключових особливостей симуляторів є реалістичність. Вони намагаються відтворити реальні фізичні, поведінкові та економічні закономірності, щоб гравці мали можливість відчувати справжність та взаємодіяти зі світом, який відтворюється в грі. В цьому жанрі гравцеві

доводиться приймати рішення, розраховувати стратегії та використовувати набуті знання та навички для досягнення поставлених цілей.

Симулятори також можуть відігравати важливу роль у навчанні та набутті досвіду. Вони дозволяють гравцям відчувати себе у реальних ситуаціях, досліджувати їх на практиці та набувати нових знань. Наприклад, симулятори авіації дозволяють пілотам тренуватися в практичних умовах без реального ризику. Такі ігри можуть поліпшити навички та реакцію гравця, що є корисним для подальшої професійної діяльності.

1.2.6 Мультиплеєрні ігри

Мультиплеєрні ігри – це жанр відеоігор, в яких декілька гравців можуть одночасно брати участь у грі та взаємодіяти між собою. Цей жанр надає можливість спільно грати з друзями або іншими гравцями з усього світу, будуючи команди, змагаючись або співпрацюючи для досягнення спільних цілей.

Мультиплеєрні ігри, або ігри зі спільним режимом гри для кількох гравців, мають багато користі для гравців. Вони сприяють реакції, координації рухів, стратегічному мисленню, швидкому прийняттю рішень та спритності. Також вони дозволяють покращити свої комунікаційні навички, так як спілкування з іншими гравцями є важливою частиною гри.

Мультиплеєрні ігри можуть бути реалізовані у різних форматах.

1.2.6.1 Кооперативні ігри

У цих іграх гравці співпрацюють, об'єднують свої зусилля та ресурси для досягнення спільних цілей. Вони можуть виконувати завдання разом,

взаємодіяти у боротьбі зі спільними ворогами або розв'язувати головоломки, використовуючи командну співпрацю.

1.2.6.2 Змагальні ігри

У таких іграх гравці змагаються один з одним за перемогу. Це можуть бути командні або індивідуальні змагання в різних форматах, наприклад, шутери, файтинги, спортивні симулятори тощо.

Гравці демонструють свою майстерність та стратегічне мислення для перемоги над супротивниками.

1.2.6.3 MMO (Massively Multiplayer Online)

Цей піджанр мультиплеєрних ігор передбачає велику кількість гравців, які можуть одночасно грати в одній віртуальній грі.

MMO-ігри часто включають великі віртуальні світи з розвинутою соціальною інтеракцією, економікою, завданнями та PvP (гравець проти гравця) битвами.

Гравці можуть спілкуватися, взаємодіяти, торгувати та змагатися між собою у віртуальному світі.

1.2.7 Головоломки

В цих іграх головною метою є розв'язування різноманітних головоломок, логічних завдань та загадок.

Головоломки можуть бути різних типів і складності, вимагати розумового напруження, кмітливості та творчого мислення гравця.

Головоломки надають гравцям можливість розвивати свої розумові навички, кмітливість, логічне та творче мислення.

Вони стимулюють гравців думати за новими шляхами, розвивати проблемне мислення та використовувати свої розумові ресурси для розв'язання завдань. Такі ігри можуть бути інтелектуально задоволеністю та викликом для гравців будь-якого віку.

Жанр головоломок можна поділити на піджанри.

1.2.7.1 Розриваючі головоломки

Ці головоломки базуються на розумових здібностях гравця, його спостережливості та здатності знаходити приховані зв'язки або рішення.

Вони можуть включати шифри, коди, криптографічні головоломки та складні логічні завдання.

1.2.7.2 Фізичні головоломки

Цей тип головоломок вимагає гравця використовувати фізичні принципи або механіку гри для розв'язання завдань.

Це можуть бути головоломки з фізикою, механічні головоломки або пазли, де гравець повинен правильно маніпулювати об'єктами або елементами оточуючого середовища.

1.2.7.3 Логічні головоломки

Цей тип головоломок базується на логічному мисленні гравця. Вони можуть включати логічні загадки, математичні завдання, логічні гри та різні

комбінаторні головоломки. Гравець повинен розуміти логіку та залежності в грі, аналізувати інформацію та знаходити оптимальні рішення.

1.2.7.4 Механічні головоломки

Ці головоломки базуються на використанні різних механічних пристроїв або механізмів для розв'язання завдань. Вони можуть включати рухомі платформи, різноманітні механізми, мікроконтролери та інші елементи, які гравець повинен вміло використовувати для подолання перешкод або досягнення мети.

1.2.8 Інді-ігри

Це незалежно розроблені відеоігри, створені невеликими студіями або навіть окремими розробниками без фінансової підтримки великих видавництв. Цей жанр відомий своєю творчістю, нестандартністю та оригінальністю.

Інді-ігри часто вирізняються своїм унікальним стилем графіки, музикою та неповторним геймплеєм. Розробники інді-ігор мають волю експериментувати зі своїми ідеями та необмежену творчу свободу, оскільки вони не залежать від комерційного успіху інвесторів.

Одна з особливостей інді-ігор полягає у тому, що вони можуть пропонувати унікальні інтерактивні досвіди, які не були розроблені великими видавництвами. Це можуть бути незвичайні ідеї, неочікувані сюжетні повороти або нестандартні механіки гри. Інді-розробники також часто використовують групову спільноту гравців для тестування, отримання фідбеку та підтримки, що сприяє подальшому вдосконаленню гри.

Інді-ігри стають все популярнішими серед гравців, оскільки вони часто пропонують неочікувані та незвичайні ігрові досвіди, які можуть захопити

уяву та викликати емоції. Вони є показником творчого потенціалу розробників та важливим джерелом інновацій в ігровій індустрії.

Багато інді-ігор виявилися дуже успішними та отримали визнання критиків та гравців. Деякі з них стали культовими та отримали широкий комерційний успіх. Інді-ігри створюють підвищений інтерес до творчості молодих розробників і сприяють розширенню границь того, що можна зробити в ігровій сфері.

Інді-ігри створюють нові можливості для розробників зробити внесок в ігрову індустрію та надихають нашу уяву, відкриваючи нам унікальні світи та досвіди, які іноді неможливо знайти в іграх великих видавництв.

1.2.9 Музичні ігри

Жанр музичних ігор є особливим і захоплюючим для любителів музики та ритму. Вони поєднують в собі музику, геймплей та виконавську вправність гравця. Основна мета музичних ігор полягає в тому, щоб гравець синхронізував свої дії з ритмом музичних композицій і виконував певні завдання.

У музичних іграх гравець зазвичай має доступ до певного набору пісень або музичних треків, які він може відтворювати або інтерактивно взаємодіяти з ними. Це може включати натискання правильних кнопок або нот відповідно до ритму, виконання певних рухів або танців під музику, або навіть створення власної музики за допомогою спеціальних інструментів.

Один з популярних типів музичних ігор – це ритм-гри. У таких іграх гравець повинен відповідати ритму, натискати правильні кнопки або виконувати рухи відповідно до показаного на екрані. Гравці можуть відчувати себе частиною музичного виконавця, виконуючи музику та рухи разом з віртуальними персонажами або виконавцями на екрані.

Інші музичні ігри можуть пропонувати інтерактивний підхід до створення музики. Вони надають гравцеві можливість створювати власні музичні композиції, використовуючи різні інструменти, звукові ефекти та семпли. Це дозволяє гравцеві бути творчим і виразити свою музичну обдарованість.

Музичні ігри не тільки розважають гравців, але і можуть поліпшити координацію рухів, ритмічність та концентрацію. Вони також можуть розвивати музичний смак та виховувати інтерес до музичного мистецтва.

1.2.10 VR-ігри

Жанр віртуальної реальності відкриває нові можливості для ігрової індустрії та дозволяє гравцям зануритися в цілком інший віртуальний світ. VR-ігри створюються з використанням спеціального обладнання, такого як віртуальні навушники, контролери руху та інші пристрої, які забезпечують іммерсивний геймплей.

Однією з основних переваг VR-ігор є їхня здатність поглинати гравця в ігровий світ. Гравець може відчувати себе на місці головного героя гри, бачити світ через очі персонажа та відчувати його рухи і дії. Це створює унікальний рівень іммерсії та взаємодії, який не можна досягти в звичайних іграх.

VR-ігри дозволяють гравцям взаємодіяти з віртуальним середовищем. За допомогою контролерів руху або інших пристроїв, гравець може виконувати різноманітні дії, такі як стрибки, біг, стрільба, маніпуляція об'єктами тощо. Це надає більш реалістичний та відчутний досвід гри.

Окрім ігрового аспекту, VR-ігри також мають потенціал для використання в різних сферах, таких як освіта, медицина, віртуальний туризм та інше. Вони можуть надати можливість віртуального навчання, тренувань або створення імітаційних середовищ для досліджень та експериментів.

1.3 Постанова задачі

Таким чином, розробка навчальних ігор є актуальним завданням розвитку технологій та більш ефективного навчання. Тому ставиться завдання розробки алгоритмів для створення таких ігор.

Об'єктом роботи є реалізація гри «Пазл» для вивчення термінів на ІТ тематику.

Метою цієї роботи є створення застосунку з використанням бази даних для навчання. В тому числі і забезпечення простоти та легкості використання програми.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- розробити концепцію гри, визначити основні правила та механіку гри;
- створити дизайн інтерфейсу гри, включаючи графічні елементи, зображення та анімацію;
- протестувати гру на різних цільових аудиторіях, збираючи й враховуючи їхні відгуки та пропозиції для поліпшення гри.

В результаті успішної реалізації проєкту очікується створення інтерактивної навчальної гри «Пазл», яка буде сприяти розвитку креативного та логічного мислення.

2 АЛГОРИТМИ ТА МЕТОДИ СТВОРЕННЯ ВІДЕОІГОР

2.1 Загальний огляд відеоігор

Відеоігри є однією з найпопулярніших форм розваги та розвитку сучасного світу. Вони представляють собою інтерактивні програми, що дозволяють користувачеві взаємодіяти з віртуальним середовищем та контролювати персонажів або об'єкти в цьому світі. Відеоігри здатні передати різні жанри та концепції, включаючи екшн, пригоди, стратегії, головоломки, гонки та багато інших.

Відеоігри володіють кількома основними складовими, що додають їм унікальність та привабливість для гравців.

2.1.1 Графічні алгоритми

Графічні алгоритми відіграють важливу роль у створенні відеоігор, оскільки вони визначають процес відображення графічних об'єктів та створення візуальних ефектів [4-9].

Нижче наведено докладний опис деяких ключових графічних алгоритмів:

2.1.1.1 Алгоритм растеризації (Rasterization Algorithm)

Алгоритм растеризації є основою для відображення тривимірних об'єктів на двовимірному екрані. Він використовується для перетворення тривимірних геометричних об'єктів (таких як полігони) на пікселі на екрані.

Алгоритм растеризації включає кроки, такі як визначення видимих полігонів, обчислення інтерполяції кольорів та текстур, а також заповнення растру.

2.1.1.2 Алгоритм затінення (Shading Algorithm)

Алгоритм затінення використовується для обчислення освітлення та кольорів тривимірних об'єктів на основі їх поверхневих властивостей та властивостей джерел світла.

Один з популярних алгоритмів затінення – алгоритм Фонга (Phong shading) – використовує модель Фонга для обчислення рівня затінення на основі нормалей об'єктів та векторів джерел світла.

2.1.1.3 Алгоритм текстурування

Алгоритм текстурування використовується для накладання текстур на поверхні тривимірних об'єктів.

Він включає кроки, такі як відображення текстурних координат на поверхню об'єкта, обчислення кольорів пікселів на основі значень текстури та інтерполяція кольорів між текстурними координатами.

2.1.1.4 Алгоритм антиаліасингу (Antialiasing Algorithm)

Алгоритм антиаліасингу використовується для зменшення аліасингу або «скачку» країв геометричних об'єктів на растровому зображенні. Він включає методи згладжування країв, такі як суперсемплінг та мультисемплінг, що дозволяють обчислювати кольори пікселів з використанням більш високої роздільної здатності, щоб отримати більш плавні та згладжені краї.

2.1.1.5 Алгоритм відлуння (Ray Tracing Algorithm)

Алгоритм відлуння є більш складним графічним алгоритмом, який дозволяє моделювати реалістичне освітлення та відображення світла на поверхнях об'єктів. Він використовує променеве відстеження для визначення шляху променів світла від джерел світла до камери, а також взаємодію цих променів з об'єктами сцени.

У таблиці нижче наведено декілька формул, які використовуються в графічних алгоритмах (табл. 2.1).

Таблиця 2.1 – Приклади формули для графічних алгоритмів

Алгоритм	Формула
1	2
Алгоритм растеризації	<p>Формула прямої $y = m * x + b$, де y, x, b – на площині в алгоритмі растеризації. y та x використовуються для позначення горизонтальних і вертикальних координат пікселя на екрані, а b відповідає зсуву прямої; m – нахил прямої. Він вказує, як швидко пряма змінюється по вертикалі відносно зміни по горизонталі.</p> <p>Формула інтерполяції: $value = value_1 + (value_2 - value_1) * t$ $value$, $value_1$, $value_2$, – $value$ представляє значення, яке інтерполюється або обчислюється на основі двох значень $value_1$ та $value_2$; t – є відсотковим значенням, що вказує на положення між $value_1$ та $value_2$ для виконання інтерполяції.</p>
Алгоритм затінення	<p>Модель Фонга: $I = I_{ambient} + I_{diffuse} + I_{specular}$ I – відповідає загальній інтенсивності; $I_{ambient}$ – інтенсивності амбієнтного (фонового) освітлення; $I_{diffuse}$ – інтенсивності розсіяного (дифузного) освітлення; $I_{specular}$ – інтенсивності відбитого (відблисків) освітлення.</p>

Продовження таблиці 2.1

1	2
Алгоритм текстурування	<p>Інтерполяція текстурних координат: $u = u_1 + (u_2 - u_1) * t$</p> <p>Знаходження кольору пікселя на основі текстури:</p> $color = texture(u, v)$ <p>u, v – позначення текстурних координат. Текстурні координати вказують на позицію пікселя на текстурі, яка буде накладена на поверхню об'єкта;</p> <p>$color$ – значення кольору пікселя, який обчислюється на основі значень текстури.</p>

2.1.2 Звук

Звук є важливою складовою відеоігор, яка доповнює графічний аспект і створює іммерсивний геймплей для гравців. Він включає звукові ефекти, музику, голосові актори та інші аудіо-елементи, які сприяють створенню атмосфери, підсилюють дію та емоційну привабливість гри.

Звукові ефекти відтворюються під час різних подій у грі, таких як вибухи, постріли, удари, пересування об'єктів, падіння та багато інших. Вони допомагають створити реалістичне відчуття взаємодії з ігровим світом. Звукові ефекти можуть бути створені шляхом запису реальних звуків або генерування їх шляхом цифрової синтезу.

Музика відеоігор може бути оригінальною композицією, створеною спеціально для гри, або використовувати ліцензовані треки. Музика може варіюватися від епічних симфонічних мелодій до електронної музики або рок-композицій, залежно від жанру та настрою гри. Музика допомагає створити атмосферу, підсилює настрій гравця та додає емоційну глибину до геймплею.

Голосові актори відіграють важливу роль у відеоіграх, особливо у іграх з великою кількістю діалогів та сюжетних моментів. Голосове акторство

включає запис голосу для персонажів, коментарів, наративу та інших аудіо-елементів. Голосове акторство дозволяє надати персонажам унікальні голосові характеристики та додає глибину до їхньої персоналізації.

Реалізація звуку відеоігор вимагає використання спеціалізованих алгоритмів та інструментів. Один з найпоширеніших методів – це використання аудіо-двигуна, який забезпечує відтворення, змішування та обробку звукових ефектів та музики. Аудіо-двигуни дозволяють програмістам легко інтегрувати звук в гру, керувати його параметрами, такими як гучність, панорамування, ефекти звуку та інші.

Важливою частиною розробки звуку відеоігор є аудіодизайн, який включає в себе процес створення та вибір звукових ефектів, композицій та голосового акторства. Аудіодизайнери використовують спеціалізовані програми для створення та редагування звуків, семплів та музичних композицій, а також співпрацюють з розробниками гри для досягнення бажаного звукового досвіду.

2.1.3 Фізика

Фізичні алгоритми відповідають за моделювання реалістичної фізики в відеоіграх [10-14]. Ці алгоритми дозволяють об'єктам у грі взаємодіяти між собою, рухатися відповідно до прикладених сил та демонструвати реалістичну поведінку.

2.1.3.1 Моделювання колізій

Моделювання колізій є важливим аспектом фізичних алгоритмів відеоігор. Воно відповідає за виявлення стику об'єктів та їхню реакцію на зіткнення. Один з популярних методів для моделювання колізій – це алгоритм

роздільних вісей (Separating Axis Theorem, SAT). Він базується на перевірці перетину осями відокремлення для визначення того, чи стикаються два об'єкти. Якщо перетин виявлений, то застосовуються подальші розрахунки, такі як розрахунок точки зіткнення та відбиття.

2.1.3.2 Моделювання руху

Для створення реалістичного руху об'єктів у відеоіграх використовуються фізичні алгоритми, такі як закони Ньютона та інші рухові рівняння. Наприклад, для моделювання руху об'єкта під впливом сили гравітації можна використовувати другий закон Ньютона.

$$F = m * g,$$

де F – сила;

m – маса об'єкта;

g – прискорення вільного падіння.

2.1.3.3 Моделювання гравітації

Моделювання гравітації є важливою частиною фізичних алгоритмів відеоігор. Для цього використовуються рівняння, що описують закон всесвітнього тяжіння. Одним з таких рівнянь є закон всесвітнього тяжіння Ньютона.

$$F = \frac{G*(m_1*m_2)}{r^2},$$

де F – сила гравітації;

G – гравітаційна константа;

m_1 та m_2 – маси двох об'єктів;

r – відстань між ними.

2.1.3.4 Моделювання фізичних властивостей матеріалів

Деякі відеоігри включають моделювання фізичних властивостей матеріалів, таких як пружність, жорсткість, деформація тощо. Для цього використовуються різні алгоритми, такі як алгоритм кінцевих елементів, який дозволяє розрахувати поведінку матеріалу під дією сил. Формули та розрахунки в цьому випадку можуть бути досить складними і залежать від конкретних властивостей матеріалу, які потрібно моделювати.

Це лише кілька прикладів фізичних алгоритмів, які використовуються в створенні відеоігор. У цій галузі постійно відбуваються дослідження та розробки нових методів для покращення реалістичності та іммерсивності геймплею.

2.1.4 Штучний інтелект (ШІ)

Штучний інтелект (ШІ) відіграє важливу роль у створенні відеоігор, оскільки він відповідає за поведінку ворогів, NPC (некерованих персонажів) та інших ігрових об'єктів, надаючи їм інтелектуальну поведінку та здатність реагувати на дії гравця.

ШІ алгоритми використовуються для створення різних аспектів геймплею, включаючи ворожі атаки, розташування ворогів, збирання ресурсів, уникання перешкод, співпрацю та багато іншого. Вони дозволяють ігровим об'єктам взаємодіяти з навколишнім світом та гравцем у реалістичний та прогресивний спосіб [15].

2.1.4.1 Прийняття рішень

Алгоритми прийняття рішень визначають, як ІІІ об'єкти реагують на ситуації та події у грі. Вони включають в себе логіку, яка визначає, які дії повинні бути виконані залежно від поточного стану гри, оточення та цілей об'єкта.

Алгоритми можуть використовувати різні підходи, такі як правила, дерева прийняття рішень, машинне навчання та інші.

2.1.4.2 Пошук шляху

Алгоритми пошуку шляху визначають оптимальний шлях, яким об'єкт може пройти від однієї точки до іншої в грі. Це може включати уникнення перешкод, обходження складних теренів та знаходження найкоротшого шляху до цілі.

Розробники використовують різні алгоритми пошуку шляху, такі як алгоритм A*, Dijkstra, та інші, для забезпечення ефективного та реалістичного руху ІІІ об'єктів.

2.1.4.3 Стратегії та тактики

Алгоритми стратегій та тактик визначають, як ІІІ об'єкти взаємодіють у грі та виконують свої цілі. Вони включають в себе розробку планів, урахування різних факторів та вибір найкращих дій для досягнення поставлених завдань. Алгоритми можуть використовувати стратегії, які підкріплюють атаку, оборону, витривалість, імітацію поведінки гравця та інші аспекти геймплею.

2.1.4.4 Співпраця та комунікація

Деякі відеоігри мають багатокористувацький режим або включають командну гру, де гравці можуть співпрацювати або взаємодіяти з ШІ об'єктами. Алгоритми співпраці та комунікації визначають, як ШІ об'єкти можуть спілкуватися між собою, координувати дії та досягати спільних цілей. Це може включати передачу повідомлень, спільне планування та кооперативні стратегії.

Штучний інтелект у відеоіграх може варіюватися від простих алгоритмів до більш складних систем, що використовують машинне навчання та нейронні мережі. Розробники використовують різні підходи та алгоритми ШІ для створення реалістичного та цікавого ігрового досвіду для гравців.

2.1.5 Управління геймплеєм

Управління геймплеєм є одним з найважливіших аспектів створення відеоігор. Воно визначає правила та механіку гри, а також способи взаємодії гравця з ігровим світом. Алгоритми управління геймплеєм виконуються в реальному часі та відповідають за різноманітні аспекти геймплею, включаючи обробку вхідних даних, реагування на події, управління станами гри та керування переходами між ними.

Обробка вхідних даних: Алгоритми управління геймплеєм отримують вхідні дані від гравця, такі як натискання кнопок на геймпаді або клавіатурі, рухи миші чи торкання екрану на мобільному пристрої. Вони аналізують ці дані та визначають, які дії повинні бути виконані відповідно до правил гри.

Реагування на події: Управління геймплеєм включає алгоритми, які відповідають на різні події в грі. Це можуть бути події, такі як зіткнення об'єктів, досягнення певного рівня або виконання певної умови. Алгоритми

реагування на події виконують відповідні дії, наприклад, відтворення анімації, зміна стану гри або надання винагороди гравцю.

Управління станами гри: Відеоігри можуть мати різні стани, такі як меню, інтро, головний режим гри, пауза тощо. Алгоритми управління геймплеєм керують переходами між цими станами та забезпечують правильну логіку гри в кожному стані. Вони визначають, які дії може виконувати гравець у кожному стані та як взаємодіяти з інтерфейсом гри.

Керування переходами: Відеоігри можуть мати складну структуру з різними рівнями, сценами або місіями. Алгоритми управління геймплеєм визначають правила для переходу між цими рівнями та забезпечують безперервну грати. Вони контролюють, коли гравець завершив певний рівень та які дії повинні бути виконані для переходу до наступного.

Для реалізації управління геймплеєм розробники використовують різні алгоритми та методи, включаючи кінцеві автомати, дерева рішень, стекові машини та інші. Вони також використовують мови програмування та спеціальні інструменти розробки, які допомагають вирішувати завдання з управління геймплеєм ефективно та елегантно.

Управління геймплеєм є важливою складовою успішної відеогри, оскільки воно визначає, як гравець взаємодіє з грою та який досвід отримує від неї. Грамотно реалізоване управління геймплеєм робить гру цікавою, захоплюючою та задовольняючою для гравців.

2.2 Методи створення відеоігор

Методи створення відеоігор можна поділити на кілька підкатегорій, які включають в себе використання готових ігрових двигунів, спеціалізованих мов та інструментів, графічного моделювання та анімації. Давайте розглянемо кожну з цих підкатегорій детальніше, використовуючи формули та таблиці, де це можливо [16-18].

2.2.1 Використання готових ігрових двигунів

Використання готових ігрових двигунів є широко поширеним методом створення відеоігор. Ігрові двигуни надають розробникам потужний набір інструментів та редакторів, що спрощують процес розробки гри. Для порівняння різних ігрових двигунів можна використовувати різні критерії (табл. 2.2).

Таблиця 2.2 – Критерії порівнянь

Критерій	Unity	Unreal Engine	Godot
Мова програмування	C#, UnityScript, C++	C++, Blueprint	GScript, C#
Графічний редактор	Unity Editor	Unreal Editor	Godot Editor
Підтримувані платформи	Багато (PC, консолі, мобільні пристрої і т.д.)	Багато (PC, консолі, мобільні пристрої і т.д.)	Багато (PC, консолі, мобільні пристрої і т.д.)
Штучний інтелект	Вбудована підтримка AI, можливість розширення	Вбудована підтримка AI, можливість розширення	Вбудована підтримка AI, можливість розширення
Ресурсоємність	Залежить від проекту	Залежить від проекту	Залежить від проекту

2.2.2 Використання спеціалізованих мов та інструментів

Для створення відеоігор також можна використовувати спеціалізовані мови програмування та інструменти. Наприклад:

– мова програмування C++ використовується для розробки швидкодіючих ігрових двигунів. Для порівняння продуктивності декількох ігрових двигунів можна використовувати формулу

$$P = C_c * C_b * C_p,$$

де P – продуктивність;

C_c – кількість кадрів за секунду;

C_b – кількість об'єктів;

C_p – кількість операцій;

– мова програмування Python широко використовується для скриптів та автоматизації відеоігор.

Для порівняння продуктивності різних скриптових мов можна використовувати таблицю (табл. 2.3).

Таблиця 2.3 – Порівняння продуктивності мов

Мова програмування	Швидкодія (операцій в секунду)	Пам'ять (використання пам'яті)
Python	1000	100 MB
Lua	2000	50 MB
JavaScript	1500	80 MB

2.2.3 Графічне моделювання та анімація

Для створення графічного вмісту відеоігор часто використовуються спеціалізовані програми для моделювання та анімації. Одна з найпопулярніших програм для цих цілей – Blender. Для порівняння різних програм для графічного моделювання можна використати таблицю (табл. 2.4).

Таблиця 2.4 – Порівняння програм для графічного моделювання

Програма	Blender	3ds Max	Maya
Інтерфейс	Відкритий джерела	Комерційна	Комерційна
Функціональність	Широкий набір інструментів	Широкий набір інструментів	Широкий набір інструментів
Підтримка форматів файлів	Багато (OBJ, FBX, і т.д.)	Багато (OBJ, FBX, і т.д.)	Багато (OBJ, FBX, і т.д.)

2.2.4 Розробка інді-ігор

Розробка інді-ігор відбувається за допомогою різних методів та інструментів, які часто використовуються незалежними розробниками для створення унікальних та інноваційних ігрових проєктів. Для цього їм потрібно розуміти як реалізувати такі фактори (табл. 2.5):

- концепція гри;
- проектування рівнів;
- розробка персонажів;
- графічне оформлення;
- звукове оформлення.

Таблиця 2.5 – Програми для розробки різних етапів інді-гри

Етап розробки	Інструменти та програми
Концепція гри	Мінімальне скетчування, Google Docs, Trello
Проектування рівнів	Tiled, Unity Tilemap, Unreal Engine 4
Розробка персонажів	Blender, Maya, Photoshop, Spine, DragonBones
Графічне оформлення	Photoshop, Illustrator, GIMP, Aseprite, Pixelmator
Звукове оформлення	Audacity, FL Studio, Reaper, Wwise, FMOD

3 ПРОГРАМНА РЕАЛІЗАЦІЯ НАВЧАЛЬНОЇ ГРИ «ПАЗЛ»

3.1 Обґрунтування вибору середовища, мови та платформи для програмної реалізації

3.1.1 Обґрунтування вибору середовища

У рамках кваліфікаційної роботи була розроблена гра «Пазл». Для реалізації було обране середовище IntelliJ IDEA. Воно надає розширену підтримку для розробки програм на Java, включаючи підказки коду, автоматичне завершення, перевірку синтаксису та інші корисні функції [19-21].

– IntelliJ IDEA має дуже зручний та інтуїтивно зрозумілий інтерфейс (рис. 3.1), який полегшує процес розробки. Він надає розширені можливості для організації проекту, редагування коду, налагодження та інших розробницьких завдань;

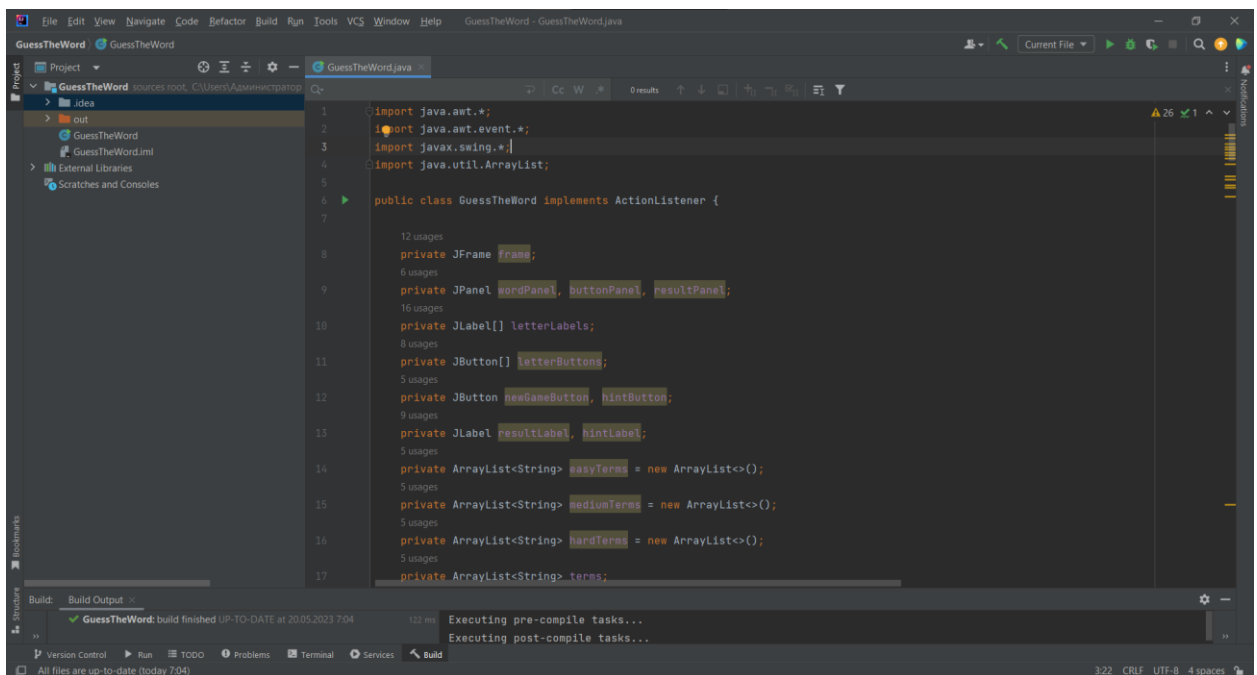


Рисунок 3.1 – Інтерфейс IntelliJ IDEA

– IntelliJ IDEA має багато вбудованих інструментів для покращення продуктивності розробки. Наприклад, воно має потужний систему керування версіями, інтегровану систему збирання та розгортання проєктів, інструменти для аналізу коду, відладки та тестування;

– IntelliJ IDEA підтримує розширення функціональності за допомогою плагінів. Це означає, що ви можете легко встановити додаткові плагіни для розширення можливостей середовища розробки або навіть створити власні плагіни для вирішення специфічних задач;

– IntelliJ IDEA має вбудовану підтримку для командної розробки, що дозволяє кільком розробникам працювати над проєктом одночасно та спільно використовувати ресурси проєкту, такі як контроль версій і обмін кодом;

– розширені можливості рефакторингу: IntelliJ IDEA надає широкий набір функцій рефакторингу, таких як перейменування, виділення методу, витягування інтерфейсу та багато інших. Це дозволяє ефективно організувати та змінювати структуру коду гри, полегшуючи його розуміння та підтримку;

– інтеграція з системами контролю версій: IntelliJ IDEA має підтримку популярних систем контролю версій, таких як Git, SVN, Mercurial і багатьох інших. Це дозволяє зручно вести версіонування та спільно працювати з кодом гри, спрощуючи процес спільної розробки;

– інтелектуальна підказка та автодоповнення: IntelliJ IDEA має потужну систему інтелектуальної підказки та автодоповнення коду, що допомагає швидко та точно вводити необхідний код. Вона розпізнає контекст та пропонує доступні методи, класи та інші елементи, що спрощує роботу з API та бібліотеками гри;

– підтримка реалізації модульної структури: IntelliJ IDEA дозволяє створювати проєкти з модульною структурою, що сприяє розділенню функціональності гри на незалежні компоненти. Це спрощує управління та розширення кодової бази, а також полегшує співпрацю розробників;

– інтеграція зі засобами тестування: IntelliJ IDEA має вбудовану підтримку для тестування, включаючи фреймворки, такі як JUnit та TestNG. Це дозволяє легко створювати та запускати тести для перевірки функціональності гри та забезпечення її якості.

Ці переваги IntelliJ IDEA в поєднанні з вибором мови програмування Java роблять його відмінним середовищем для розробки гри «Пазл» [22-27].

3.1.2 Обґрунтування вибору платформи

Для реалізації було обрано платформу Windows. Це пов'язано з тим, що Windows має великий ряд переваг для розробки гри:

– популярність системи. Windows є однією з найпоширеніших операційних систем у світі. Вибір цієї платформи дозволяє досягти широкого кола користувачів і забезпечити високий рівень доступності гри;

– інтеграція зі засобами розробки. Багато популярних середовищ розробки, таких як IntelliJ IDEA, підтримують Windows як основну платформу. Це забезпечує зручну інтеграцію та підтримку для розробки програм на даній платформі;

– Windows має велику кількість розробницьких інструментів, бібліотек та фреймворків, які спрощують розробку і підтримку гри. Це може включати інструменти для графічної реалізації, звукових ефектів, мережевої взаємодії та багато іншого;

– Windows надає доступ до спеціалізованих ресурсів для розробки ігор, таких як документація, форуми, бібліотеки, SDK (Software Development Kit) та інструменти для оптимізації та вдосконалення продуктивності гри;

– оскільки Windows є однією з найпоширеніших платформ, тестування гри на цій операційній системі може бути більш зручним і ефективним. Розробка гри на платформі Windows дає доступ до широкого спільноти

розробників, які діляться знаннями, досвідом і рішеннями з іншими членами спільноти;

- Windows підтримує широкий спектр апаратного забезпечення, що дозволяє використовувати його для розробки гри з високою деталізацією графіки, захоплюючими звуковими ефектами та іншими вимогами до продуктивності;

- Windows має широкий ринок цифрових майданчиків та інших каналів розповсюдження ігор, які дозволяють вам ефективно поширювати вашу гру серед широкої аудиторії. Наприклад, можна розповсюджувати гру через Steam, Microsoft Store або інші популярні платформи;

- Windows має велику та активну ігрову спільноту. Це створює можливості для спілкування з іншими розробниками, отримання зворотного зв'язку від гравців і спільного вдосконалення вашої гри.

3.1.3 Обґрунтування вибору мови програмування

Вибір мови програмування Java, об'єктно-орієнтованої мови програмування, для розробки гри «Пазл» обґрунтован досить великою кількістю переваг:

- Java є однією з найпопулярніших мов програмування з великою спільнотою розробників. Це означає, що можна легко знайти документацію, підручники, форуми та інші ресурси, які допоможуть вам вирішити проблеми, з якими ви можете зіткнутися під час розробки гри;

- багатий набір бібліотек та фреймворків. В екосистемі Java існує велика кількість сторонніх бібліотек і фреймворків, які полегшують розробку гри. Наприклад, ви можете використовувати бібліотеку JavaFX для графічного інтерфейсу користувача або фреймворк LibGDX для розробки ігор для різних платформ;

– Java є мовою програмування, що базується на об'єктно-орієнтованому підході. Це дозволяє організувати код гри в логічні об'єкти та класи, що полегшує розуміння, підтримку та розширення вашого проєкту;

– Java має вбудовані механізми для обробки помилок та винятків, що робить її більш надійною та стійкою до збоїв. Крім того, вона має вбудовану систему керування пам'яттю, що допомагає уникнути багатьох типових помилок, пов'язаних з управлінням пам'яттю [28-30].

3.2 Програмна реалізація

3.2.1 Опис задумки програми

Навчальна гра має на меті надати користувачам можливість покращити свої навички в області ІТ. Гра зосереджена на стимулюванні інтерактивного навчання шляхом використання цікавих завдань та випробувань.

У грі присутні різноманітні рівні складності, завдання та питання, які розширюються по мірі просування гравця. Гравці виконують різні дії, які вимагають застосування знань та навичок відповідної області.

Гра пропонує користувачам оцінку їхнього виконання завдання для стимулювання покращення результатів. Можливо, гра використовує систему нагород та досягнень, яка мотивує гравців досягати кращих результатів та розблокувати нові рівні або функції.

Основною метою цієї навчальної гри є забезпечення ефективного навчання через інтерактивний досвід. Гравці отримують можливість не тільки освоїти нові знання, але й застосовувати їх на практиці, вирішуючи завдання та проблеми. Гра сприяє активному залученню користувачів та створює сприятливу атмосферу для їхнього навчання та розвитку.

3.2.2 Бібліотеки класів

Бібліотеки класів використовуються в програмі для підключення готового коду та функціоналу, які надаються цими бібліотеками. Вони містять набір класів, інтерфейсів і методів, які можна використовувати для розширення можливостей програми без необхідності написання всього коду з нуля. Основна мета використання бібліотек полегшити розробку, забезпечити готові рішення для певних завдань та прискорити процес програмування.

Бібліотека «`java.awt.*`» надає класи і методи для роботи з графічним інтерфейсом користувача, такі як створення вікон, панелей, міток та кнопок. Вона дозволяє легко створювати та керувати елементами графічного інтерфейсу, відображати зображення, керувати подіями миші та клавіатури.

Бібліотека «`java.awt.event.*`» надає класи та інтерфейси для обробки подій. Вона дозволяє визначити реакцію програми на дії користувача, такі як натискання кнопки, переміщення миші або натискання клавіші. Це дозволяє забезпечити взаємодію програми з користувачем та виконувати певні дії при виникненні певних подій.

Бібліотека «`javax.swing.*`» також використовується для створення графічного інтерфейсу, але надає більш розширений набір компонентів, таких як таблиці, списки, текстові поля тощо. Вона також має більш сучасний вигляд та можливості налаштування стилів, шрифтів та інших атрибутів.

Бібліотека «`java.util.ArrayList`» використовується для роботи з колекціями. Вона надає гнучку структуру даних для зберігання та керування динамічними списками об'єктів. За допомогою «`ArrayList`» можна додавати, видаляти та змінювати об'єкти в списку, а також виконувати різноманітні операції з ними, такі як пошук, сортування тощо.

3.2.3 Використання ООП

У грі «GuessTheWord» використовується об'єктно-орієнтоване програмування для створення класів, об'єктів та взаємодії між ними. Нижче описано деякі аспекти використання об'єктно-орієнтованого програмування в грі:

- клас «GuessTheWord». Цей клас є головним класом гри і містить метод «main», який є точкою входу до програми. Він використовується для створення екземпляру гри та запуску головного циклу програми;

- класи-компоненти графічного інтерфейсу. У грі використовуються класи, які представляють графічні компоненти, такі як «JFrame», «JPanel», «JLabel», «JButton» тощо. Ці класи дозволяють створювати та відображати графічні елементи на екрані;

- класи гри та логіки. У грі використовуються класи, які відповідають за логіку гри та взаємодію з користувачем. Наприклад, клас «GuessTheWord» містить методи для створення нової гри, перевірки відгадок гравця, відображення результатів гри та інші операції пов'язані з логікою гри;

- класи-контейнери. У грі використовуються класи-контейнери, які дозволяють організувати ієрархію компонентів і керувати їх розміщенням на екрані. Наприклад, «JPanel» використовується для створення контейнеру для графічних елементів, які відображаються у певній частині інтерфейсу користувача;

- події та обробники подій. У грі використовуються обробники подій, які реагують на дії користувача, такі як натискання кнопок або клавіш. Ці обробники подій реалізовані у вигляді окремих класів, які реалізують відповідні інтерфейси, наприклад «ActionListener», та містять код, який виконується при виникненні певної події.

3.2.4 Класи у програмі

Клас «GuessTheWord» є основним класом гри «Guess the Word». В ньому міститься багато підкласів та методів.

3.2.4.1 Класи з графічної бібліотеки SWING

Класи з графічної бібліотеки свінг допомагають зробити інтерфейс програми. Ось приклади класів, які були використані в програмі:

- «JFrame». Цей клас представляє головне вікно програми. Ви використовуєте його для створення вікна гри і налаштування його вигляду та поведінки;

- «JPanel». Цей клас є контейнером для компонентів і використовується для організації компонентів в панелі. У вашій програмі використовується «wordPanel», «buttonPanel» та «resultPanel», які успадковуються від «JPanel»;

- «JLabel». Цей клас використовується для відображення тексту або зображень. У вашій програмі використовуються «letterLabels», «resultLabel», «hintLabel», «definitionLabel», «attemptsLabel» та «timerLabel» як мітки, які відображають різні тексти;

- «JButton». Цей клас представляє кнопку, на яку можна натискати. Ви використовуєте його для створення кнопок «newGameButton», «hintButton» та «letterButtons», які користувач може натискати для взаємодії з грою;

- «BoxLayout». Цей клас використовується для управління розміщенням компонентів в контейнері. Ви використовуєте його для налаштування вертикального розташування компонентів у «resultPanel»;

- «GridLayout». Цей клас використовується для розташування компонентів у вигляді таблиці з заданим числом рядків і стовпців. У вашій програмі використовується для організації «letterLabels» у рядки та стовпці у «wordPanel» та «letterButtons» у «buttonPanel»;

– «SwingUtilities». Цей клас надає різні корисні методи для роботи з бібліотекою Swing. У вашому коді використовується його метод «invokeLater» для запуску гри у потоці з подіями Swing.

3.2.4.2 Поля класу

Поля класу – це змінні, які визначені в межах класу і зберігають стан або дані, які відносяться до об'єктів цього класу. Приклади полів:

- «wordPanel», «buttonPanel», «resultPanel» – панелі для відображення слів, кнопок та результатів гри;
- «letterLabels», «letterButtons» – масиви міток та кнопок, які представляють літери;
- «newGameButton», «hintButton» – кнопки для початку нової гри та використання підказки;
- «resultLabel», «hintLabel», «definitionLabel», «attemptsLabel», «timerLabel» – мітки для відображення результатів гри, підказок, визначення та часу;
- «easyTerms», «mediumTerms», «hardTerms» – списки слів для різних рівнів складності;
- «terms» – список слів для поточного рівня складності;
- «selectedTerm», «wordToGuess» – вибране слово та слово, яке потрібно відгадати;
- «wrongGuesses» – кількість неправильних спроб відгадати слово;
- «hintIndex» – індекс підказки, яку гравець використав;
- «hintUsed» – прапорець, що вказує, чи була використана підказка;
- «maxAttempts» – максимальна кількість спроб відгадати слово;
- «remainingAttempts» – кількість залишених спроб;
- «timer» – об'єкт «Timer» для відліку часу гри;
- «secondsPassed» – кількість секунд, які пройшли під час гри;

– «gameDuration» – тривалість гри в секундах.

3.2.4.3 Методи класу

Методи класу використовуються для виконання певних дій або операцій з об'єктами цього класу. Вони визначають поведінку об'єктів та дозволяють взаємодіяти з ними. Приклад методів класу в програмі:

- конструктор «GuessTheWord» – налаштовує вікно гри та ініціалізує поля;
- «chooseDifficulty()» – виводить діалогове вікно для вибору рівня складності;
- «setDifficulty(int difficulty)» – встановлює поточний рівень складності на основі вибору гравця;
- «newGame()» – починає нову гру, ініціалізує слово для відгаду та відповідні мітки;
- «startTimer()» – запускає таймер гри;
- «stopTimer()» – зупиняє таймер гри;
- «checkLetter(String letter)» – перевіряє, чи міститься відповідна літера у слові для відгаду, оновлює мітки та перевіряє умови перемоги або поразки;
- «checkWin()» – перевіряє, чи всі літери відгадані;
- «disableLetterButtons()» – вимикає кнопки літер, коли гра закінчена;
- «actionPerformed(ActionEvent e)» – реагує на дії користувача, такі як натискання кнопок нової гри, підказки або літер;
- «main(String[] args)» – метод «main», який запускає гру.

3.3 Інструкція користувача

При запуску гри перед користувачем відкриються 2 вікна: гра та вибір складності (рис. 3.2).

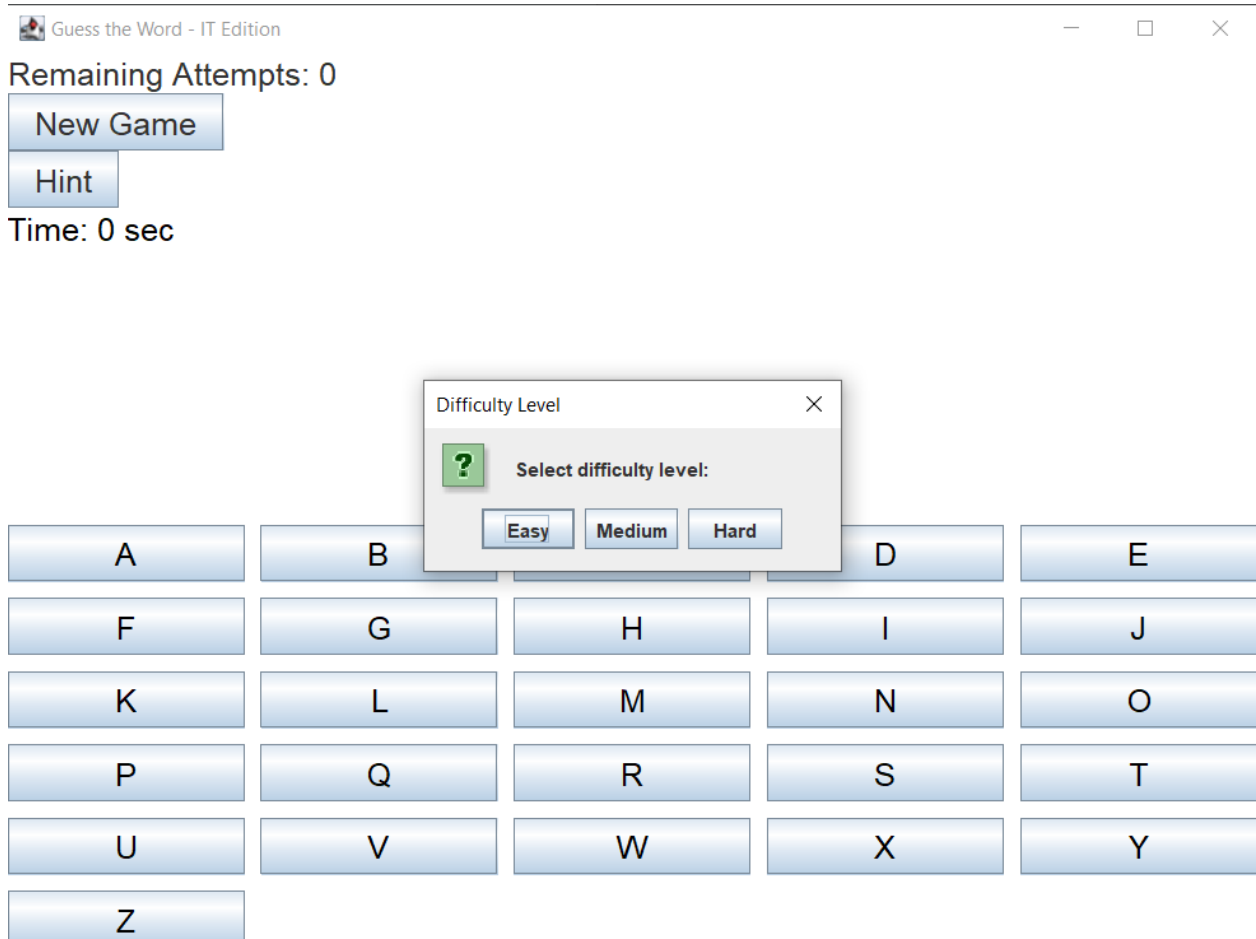


Рисунок 3.2 – Два вікна при відкритті гри

Тут можна побачити, що у грі є 3 рівня складності: легкий, середній та важкий. Обираючи різні рівні складності гра буде пропонувати відгадати слова в залежності від рівня складності, який буде обрано.

Для того, щоб продовжити користування програмою потрібно обрати рівень складності. Після чого вікно гри стане доступним (рис. 3.3). Приклад роботи програми буде продемонстровано на рівні складності «Medium».

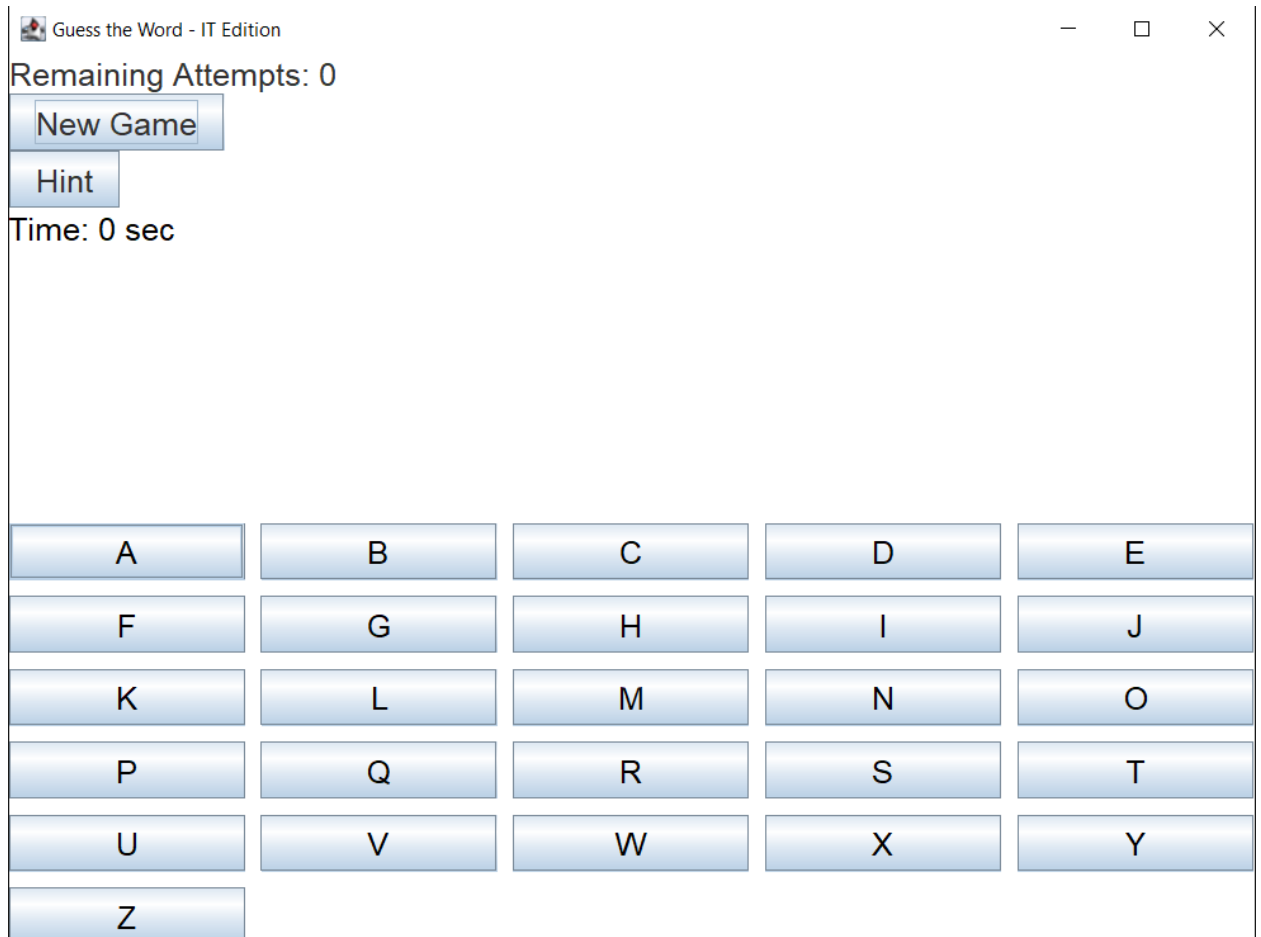


Рисунок 3.3 – Вікно гри

В цьому вікні присутня графічна клавіатура, клавішу «New Game» та «Hint», кількість доступних спроб вгадати слово та пройдений час.

Щоб почати гру потрібно натиснути кнопку «New Game». Це змусить програму обрати випадкове слово із одного з масивів слів в залежності від обраного рівня складності, запустити таймер, дати визначення слова, яке потрібно відгадати, показати кількість букв у слові, яке потрібно відгадати та відобразити кількість доступних спроб (рис. 3.4).

Після кожного разу як користувач виграє чи програє потрібно натискати клавішу «New Game» щоб зіграти у гру знову.

Для того, щоб вгадати слово потрібно натискати на клавіші графічної клавіатури. При натисканні літери якої нема у слові клавіша графічної клавіатури, яка відповідає за цю літеру, блокується, а кількість доступних спроб зменшується на один (рис. 3.5).

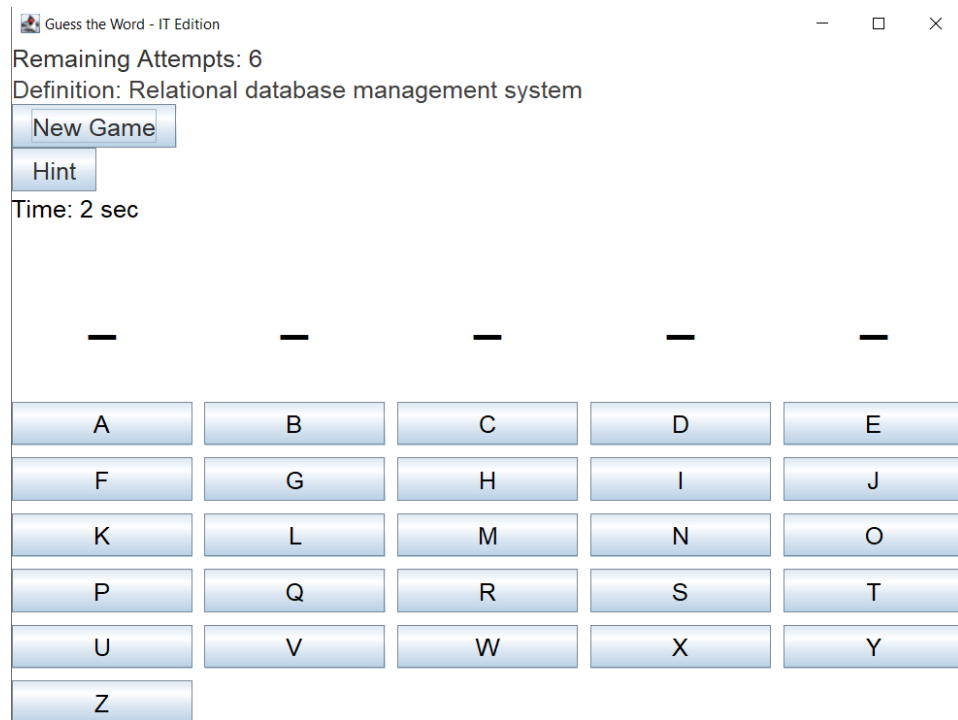


Рисунок 3.4 – Гра після натискання кнопки «New Game»

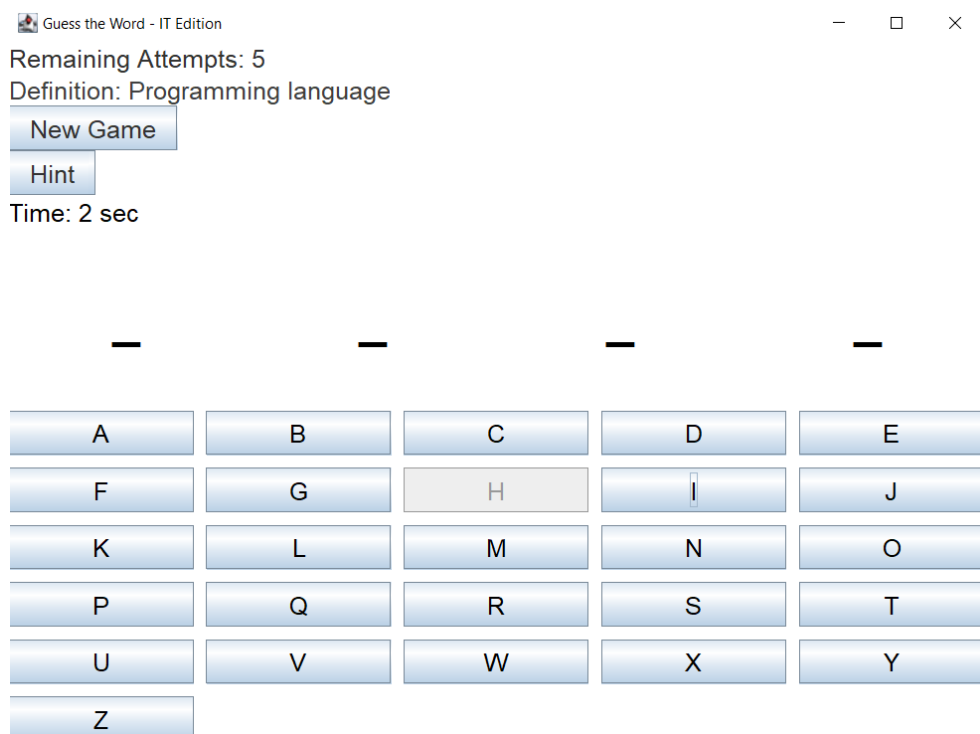


Рисунок 3.5 – Приклад заблокованої клавіші зменшення доступних спроб через натискання літери, якої нема у слові

При використанні усіх доступних спроб гра завершується. За правилами гри, користувач програє і для того, щоб допомогти йому вивчити термін, який користувач не відгадав, програма пише правильну відповідь (рис. 3.6).

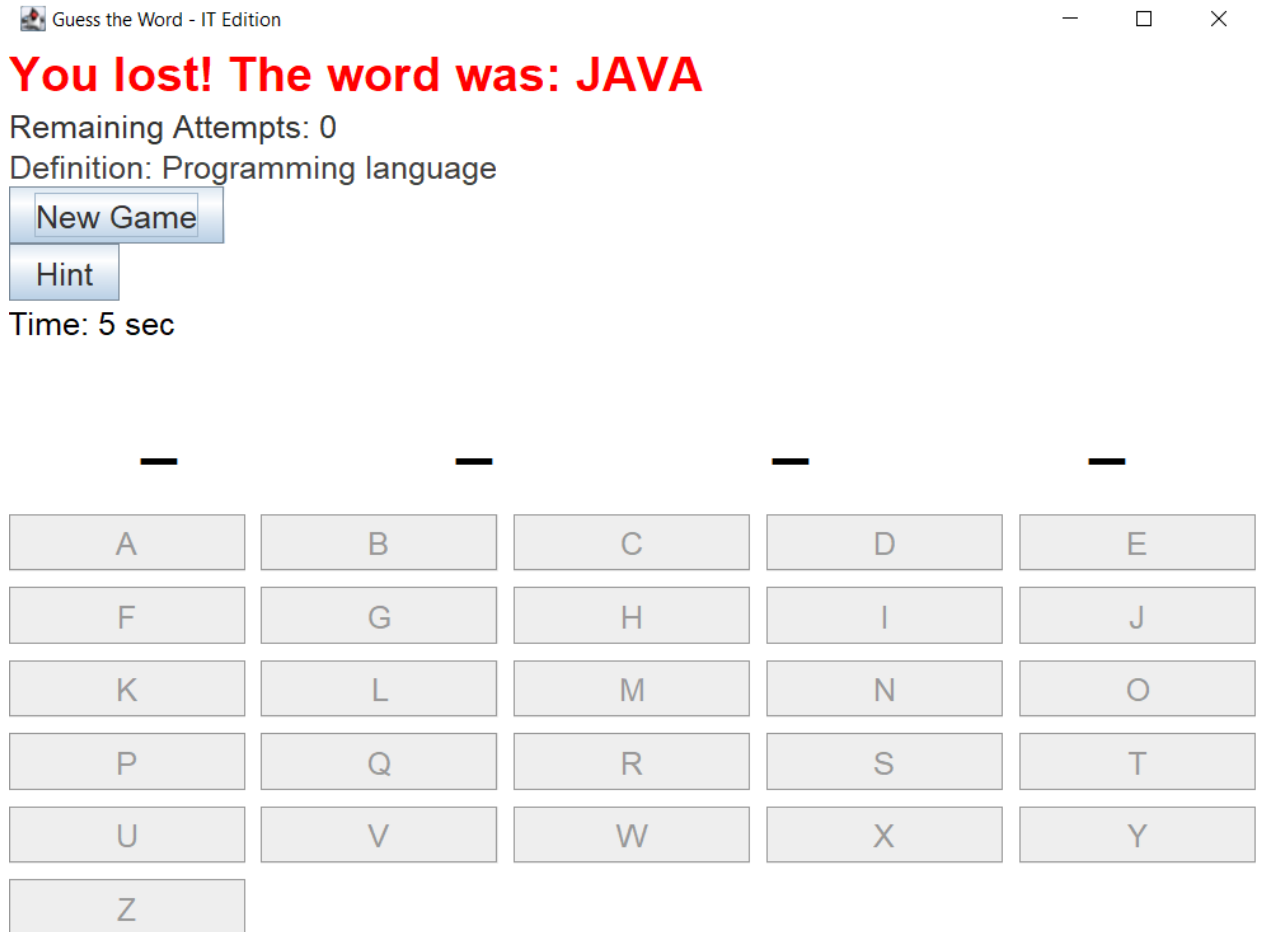


Рисунок 3.6 – Приклад програшу у грі шляхом використання всіх доступних спроб

Якщо користувач вгадує літеру у слові, то клавіша із цією літерою блокується, а вгадана літера з'являється на місці де вона повинна стояти. За вдалу спробу відгадати літеру в гравця не відіймається кількість доступних спроб. Якщо вгадана літера зустрічається у слові декілька разів, то вона з'являється на всіх місцях де вона повинна стояти. Не потрібно натискати правильну літеру декілька разів і в такому випадку вона блокується з першого натискання на неї (рис. 3.7).

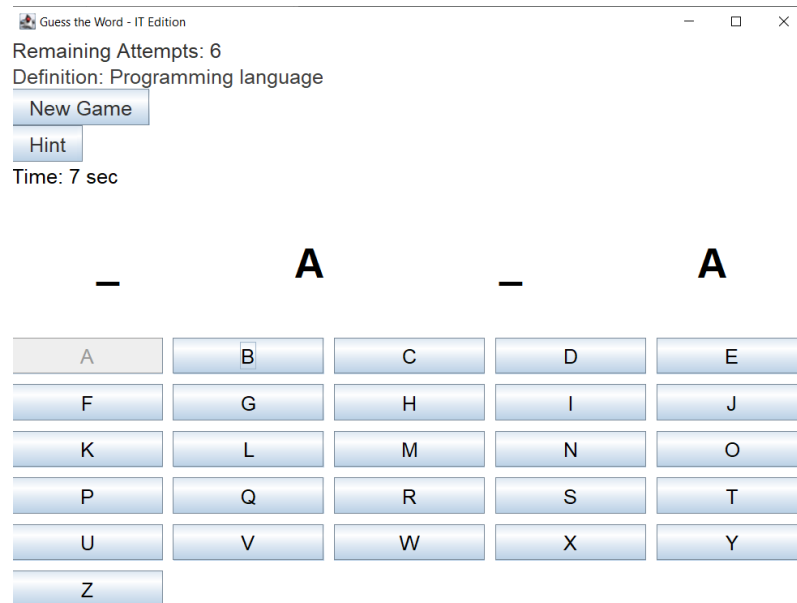


Рисунок 3.7 – Приклад натискання правильної клавіші, яка зустрічається у слові декілька разів

Після вдалої спроби відгадати слово гра привітає вас із перемогою (рис. 3.8). Після усіх варіантів закінчення гри таймер зупиняється, завдяки чому користувач може оцінити на скільки швидше він відгадав потрібне слово та скільки помилок він допустив поки його відгадував.

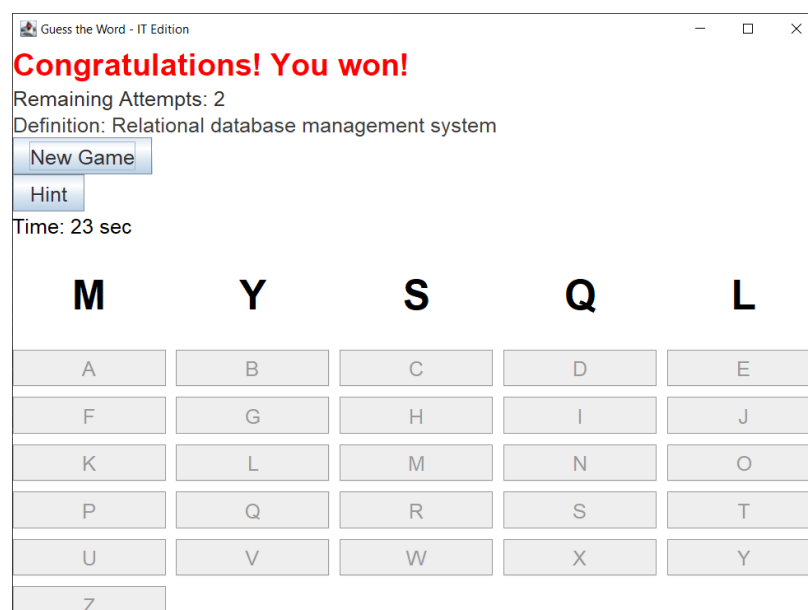


Рисунок 3.8 – Приклад роботи програми при повному відгадуванні слова

В цій грі таймер не тільки допомагає дізнатися наскільки поліпшився результат, а ще й є невід'ємною частиною ігрового процесу. Якщо користувач не встигне відгадати слово за 60 секунд, то гра автоматично завершується програшом користувача (рис. 3.9). Щоб почати гру з початку потрібно знову натиснути клавішу «New Game».

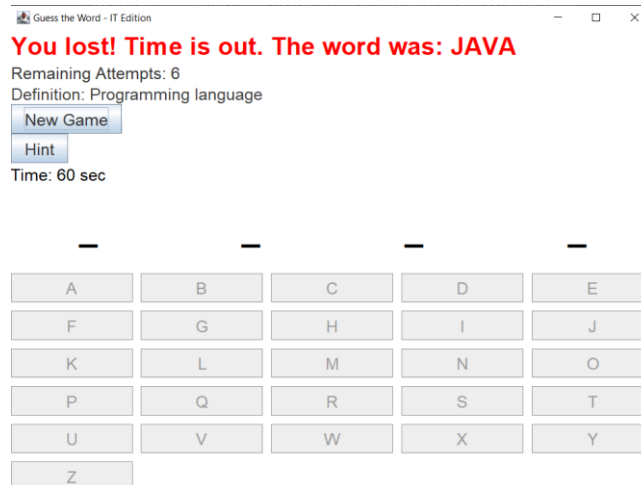


Рисунок 3.9 – Приклад завершення гри через спливання часу

В цій грі існує клавіша «Hint». Якщо на неї натиснути після початку гри, то з'явиться повідомлення в якому буде вказана випадкова літера в слові, яке користувач буде відгадувати на той момент. Сама клавіша «Hint» при цьому блокується і нею неможливо скористатися до початку нової гри (рис. 3.10).

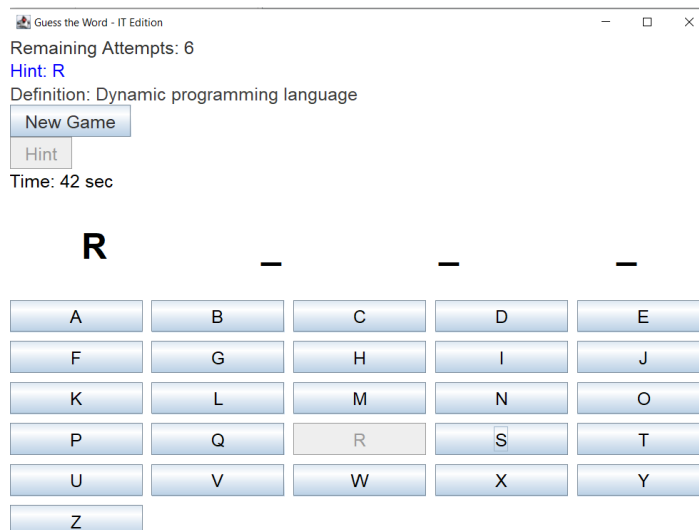


Рисунок 3.10 – Приклад роботи клавіші «Hint»

ВИСНОВКИ

У рамках кваліфікаційної роботи була розроблена і реалізована гра на мові програмування Java для Windows, здатна на навчання користувачів базовим термінам з ІТ теми.

Терміни та їх визначення зберігаються локально, що дозволяє користуватися програмою за відсутності зв'язку з мережею інтернет та досить легко додавати нові терміни та їх визначення до програми.

Практичні рекомендації із проведеної роботи полягають у покращенні функціоналу гри та оптимізуванні коду для більш швидкої роботи програми.

Практична значущість роботи полягає в покращенні якості відеоігор та їх позитивний вплив на життя людини.

Перспективи розробки навчальних ігор можуть бути пов'язані із швидким розвитком технологій та поширенням популярності відеоігор не тільки як засіб розваги та відпочинку, а ще й як цікавий спосіб навчання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Tools for fast metric data search in structural methods for image classification, *IEEE Access*, 10, pp. 124738-124746.
2. Kuhn, J. (2018). Minecraft: education edition. *Calico journal*, 35(2), 214-223.
3. Higdon, J., Miller, S., & Paul, N. (2009, October). Educational Gaming for the Rest of Us: Thinking Worlds and WYSIWYG Game Development. In *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (pp. 359-362). Association for the Advancement of Computing in Education (AACE).
4. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.
5. Shen, Y. (2009, March). Teaching game development using microsoft xna game studio. In *Proceedings of the 2009 Spring Simulation Multiconference* (pp. 1-6).
6. Гороховатський, В. О., & Творошенко, І. С. (2022). Аналіз багатовимірних даних за описом у формі множини компонент.
7. Ahmad, M. A., Tvoroshenko, I., Baker, J. H., & Lyashenko, V. (2019). Modeling the structure of intellectual means of decision-making using a system-oriented nfo approach.
8. Fahrurozi, A., Maesaroh, S., Suwanto, I., & Nursyahidah, F. (2018). Developing learning trajectory based instruction of the congruence for ninth grade using central java historical building. *JRAMathEdu (Journal of Research and Advances in Mathematics Education)*, 3(2), 78-85.

9. Shafronenko, A., Pliss, I., & Ye, B. (2013). The evolving adaptive neural network for data processing with missing observations. *Радіоелектроніка, інформатика, управління*, (2 (29)), 119-127.
10. Gorokhovatsky, V. (2014). Structural analysis and intellectual data processing in computer vision. *SMIT: Kharkiv, Ukraine*.
11. M., Otón, S., & Hilera, J. R. (2011). Games-Based Learning Framework. *Computer Technology and Application*, 906.
12. Gorokhovatskiy, A. V., Gorokhovatskiy, V. A., Vlasenko, A. N., & Vlasenko, N. V. (2014). Quality Criteria for Multidimensional Object Recognition Based Upon Distance Matrices. *Telecommunications and Radio Engineering*, 73(18).
13. Гороховатський, В. О., Гадецька, С. В., & Пономаренко, Р. П. (2019). Статистичні розподіли та ланцюжкове подання даних при визначенні релевантності структурних описів візуальних об'єктів.
14. Bodyanskiy, Y., Shafronenko, A., & Pliss, I. (2022). Clusterization of vector and matrix data arrays using the combined evolutionary method of fish schools. *Системні дослідження та інформаційні технології: міжнародний науково-технічний журнал, № 4*.
15. Творошенко, І. С. (2018). Особливості застосування сучасних принципів штучного інтелекту до розробки ефективних механізмів моделювання складних систем. *Science and Technology of the Present Time: Priority Development Directions of Ukraine and Poland*, 118-121.
16. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.
17. Shanahan, J. (2009, March). Students create game-based online learning environment that teaches Java programing. In *Proceedings of the 47th Annual Southeast Regional Conference* (pp. 1-4).

18. Kickmeier-Rust, M., & Albert, D. (2012). Developing Educational Games. *An Alien's Guide to Multi-Adaptive Educational Computer Games*, 21.
19. Гороховатський, В. О., & Творошенко, І. С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник.
20. Xu, C. W., & Xu, C. W. (2018). Java Programming and Game Development: A Project HelloWorld. *Learning Java with Games*, 3-15.
21. Tvoroshenko, I. S., & Gorokhovatskyi, V. O. (2019). Modification of the branch and bound method to determine the extremes of membership functions in fuzzy intelligent systems. *Telecommunications and Radio Engineering*, 78(20).
22. Alam, A. (2022). Design and Development of Digital Game-Based Learning Software for Incorporation into School Syllabus and Curriculum Transaction. *Mohanty, A., Alam, A., Sarkar, R., & Chaudhury, S.(2021). Design and Development of Digital Game-Based Learning Software for Incorporation into School Syllabus and Curriculum Transaction. Design Engineering*, 4864-4900.
23. Гороховатський, В. О., Гадецька, С. В., & Стяглик, Н. І. (2019). Вивчення статистичних властивостей моделі блочного подання для множини дескрипторів ключових точок зображень властивостей моделі блочного подання для множини дескрипторів ключових точок зображень. *Gutiérrez, J. J.*
24. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), С. 5-12.
25. Ross, J. M. (2002). Guiding students through programming puzzles: value and examples of java game assignments. *ACM SIGCSE Bulletin*, 34(4), 94-98.
26. Zhang, X., & Liu, X. (2017). JAVA Jigsaw Puzzle. *International Journal of Advanced Pervasive and Ubiquitous Computing (IJAPUC)*, 9(3), 1-14.
27. Qiang, T., Xiaofeng, Z., Likai, L., Sheng, Y., & Han, W. (2018, October). The Design of Game Platform Based on Java. In *IOP Conference Series: Materials Science and Engineering* (Vol. 435, No. 1, p. 012014). IOP Publishing.

28. Sierra, K., & Bates, B. (2005). *Head First Java: A Brain-Friendly Guide*. "O'Reilly Media, Inc."
29. Ingles, B. (2006, March). The future of Java™ game development. In *Proceedings of the 44th annual southeast regional conference* (pp. 698-701).
30. Fan, J., Ries, E., & Tenitchi, C. (1996). *Black art of Java game programming* (Vol. 112). Waite Group Press.