

Міністерство освіти і науки України
Криворізький національний університет
Кафедра моделювання та програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавра

з напрямку підготовки 121 «Інженерія програмного забезпечення»

На тему: Розробка стратегічної розвиваючої комп'ютерної гри

*Засвідчую, що в цій
кваліфікаційній роботі немає
запозичень із праць інших
авторів без відповідних посилань.
Студент гр. ПЗ-20-2
_____ Шумило О. Д.*

Керівник кваліфікаційної
роботи

/ Стрюк А. М. /

Завідувач кафедри

/ Стрюк А. М. /

Кривий Ріг
2024

Криворізький національний університет
Факультет: Інформаційних технологій
Кафедра: Моделювання та програмного забезпечення
Освітньо-кваліфікаційний рівень: бакалавр
Спеціальність: 121 "Інженерія програмного забезпечення"

ЗАТВЕРДЖУЮ
Зав. кафедри
Стрюк А. М.
«__» _____ 2024__ р.

ЗАВДАННЯ
на кваліфікаційну роботу

студенту групи ППЗ-20-2 Шумило Олександр Дмитровичу

1. Тема: Розробка стратегічної розвиваючої комп'ютерної гри затверджено наказом по КНУ №__ від «__» лютого 2024 р.
2. Термін подання студентом закінченого проекту «10» червня 2024 р.
3. Вихідні дані по роботі: Пояснювальна записка: 70 сторінок, 26 рисунків, 1 додаток, 30 використаних у роботі джерел
4. Зміст пояснювальної записки (перелік питань, що їх треба розробити): .
5. Перелік графічного демонстраційного матеріалу: Приклади існуючих програм. Функціональна схема. Блок-схема основного алгоритму. Структура бази даних. Приклади роботи розробленої програми.

Календарний план:

№	Найменування етапів кваліфікаційної роботи	Термін виконання етапів роботи
1	<i>Формулювання мети та задач роботи</i>	13.02.2024-20.02.2024
2	<i>Аналіз інформаційних джерел</i>	21.02.2024-09.03.2024
3	<i>Визначення вимог до програмного забезпечення</i>	10.03.2024-18.03.2024
4	<i>Розробка функціональної схеми</i>	19.03.2024-23.03.2024
5	<i>Розробка алгоритмів</i>	24.03.2024-31.03.2024
6	<i>Розробка бази даних</i>	01.04.2024-14.04.2024
7	<i>Розробка програмного забезпечення</i>	15.04.2024-13.05.2024
8	<i>Тестування програмного забезпечення</i>	14.05.2024-20.05.2024
9	<i>Оформлення пояснювальної записки</i>	21.05.2024-12.06.2024
10	<i>Розробка демонстраційних матеріалів</i>	13.06.2024-17.06.2024

Дата видачі завдання: «__» _____ 2024 р.
Студент _____ Шумило О. Д.
Керівник роботи _____ Стрюк А. М.

РЕФЕРАТ

КОМП'ЮТЕРНА ГРА, ГРА ЗІ СЛОВАМИ, C++, C++ BUILDER, ПРОГРАМУВАННЯ, ІНТЕРФЕЙС, АЛГОРИТМ.

Пояснювальна записка: 70 с., 26 рис., 30 джерел.

В представленій кваліфікаційній роботі було спроектовано та реалізовано програмне забезпечення стратегічної розвиваючої гри за мотивами гри в слова.

Метою роботи є створення розвиваючої комп'ютерної гри.

Було зроблено історичний огляд розробки ігор зі словами. Зроблено огляд комп'ютерних ігор зі словами. Враховуючи досвід інших розробок ігор зі словами, ми сформулювали вимоги до гри «Вгадай слово».

Спроектовано основні алгоритми гри «Вгадай слово», серед яких можна виділити алгоритм загальної організації гри та алгоритм перевірки слова та вгаданих гравцем букв.

Для розробки програмного забезпечення гри «Вгадай слово» нами було обрано одну з найпопулярніших мов – C++. А середовищем розробки обрано C++ Builder.

За допомогою інструментальних засобів було спроектовано інтерфейс гри «Вгадай слово», реалізовані алгоритми та протестовано програмне забезпечення.

ABSTRACT

COMPUTER GAME, WORDS, C ++, C ++ BUILDER, PROGRAMMING, INTERFACE, ALGORITHM.

Explanatory note: 70 pp., 26 Fig., 30 sources.

In the presented qualification work, the software of the game "Guess the Word" was designated and implemented.

The purpose of the work is to create a computer game "Guess the Word".

A historical review of the development of games with words was made. Computer games have been reviewed with words. Given the experience of other game developments with words, we formulated the requirements for the game "Guess the Word".

The basic algorithms of the game "Guess the Word" are designated, among which you can distinguish the algorithm of the general organization of the game and the algorithm for verifying the word and the player visited by the player.

To develop the software "Guess the Word", we have chosen one of the most popular languages – C ++. And C ++ Builder is selected.

With the help of tools, the interface of the game "Guess the Word", implemented algorithms and tested software was used.

ЗМІСТ

ВСТУП	7
1 ДОСЛІДЖЕННЯ ДОСВІДУ РОЗРОБКИ ІГОР ЗІ СЛОВАМИ	9
1.1 Історичний огляд розробки ігор зі словами	9
1.2 Огляд комп'ютерних ігор зі словами.....	14
1.3 Визначення вимог до гри «Вгадай слово».....	24
2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ГРИ «ВГАДАЙ СЛОВО»	25
2.1 Функціональна схема програмного забезпечення гри «Вгадай слово»	25
2.2 Розробка основних алгоритмів гри «Вгадай слово»	26
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ГРИ «ВГАДАЙ СЛОВО»	29
3.1 Вибір мови програмування та засобів розробки гри «Вгадай слово»	29
3.2 Проєктування інтерфейсу гри «Вгадай слово».....	30
3.3 Приклади роботи програми	32
ВИСНОВКИ	40
Перелік посилань	42
Додаток А Текст програми.....	44

ВСТУП

Дослідники вважають, що прагнення грати – природне і властиве не лише людям, а навіть деяким тваринам. Але саме розвиток людської цивілізації породив велику кількість різноманітних ігор, наповнив їх різноманітним змістом.

Багато ігор передбачають високу фізичну активність, але є й такі, що орієнтовані на інтелектуальну активність гравців. До таких ігор, зокрема відносяться так звані логічні та стратегічні ігри, у тому числі ігри зі словами.

За час розвитку мовлення та писемності з'явилося багато ігор зі словами. Всім відомі, наприклад, кросворди, ребуси та багато інших ігор. Відомо, що вони розвивають не лише словниковий запас, але й розвивають логіку, здатність до аналізу, прогнозування, асоціативного мислення тощо.

З появою комп'ютера з'явилась і нова категорія ігор – комп'ютерні ігри. Більшість комп'ютерних ігор мають виключно розважальний характер, акцент робиться на візуальних ефектах. Проте, існує і значний пласт логічних та розвиваючих комп'ютерних ігор. До таких ігор відносяться і ігри зі словами.

Справжнім феноменом стала мобільна гра «Wordly». Ця гра з'явилась у 2021 році і за кілька місяців стала шалено популярною. Про неї писали статті і робили телевізійні передачі. Ідея гри дуже проста. Гравцю потрібно відгадати загадане слово за шість спроб. Після кожної спроби гравець бачить, які букви він вгадав. Якщо вгадав букву та її позицію, вона підсвічується зеленим. Якщо лише букву, то така буква підсвічується жовтим.

Зважаючи на популярність та оригінальність гри Wordly в своїй кваліфікаційній роботі ми вирішили розробити україномовну версію цієї гри під назвою «Вгадай слово».

Метою роботи є створення комп'ютерної гри «Вгадай слово».

Задачами, що потрібно виконати є наступні:

- проаналізувати особливості ігор зі словами;
- визначити вимоги до програмного забезпечення гри «Вгадай слово»;

- спроектувати функціональну схему програмного забезпечення гри;
- розробити основні алгоритми програми;
- спроектувати інтерфейс комп'ютерної гри «Вгадай слово»;
- розробити та протестувати програмне забезпечення гри «Вгадай слово».

1 ДОСЛІДЖЕННЯ ДОСВІДУ РОЗРОБКИ ІГОР ЗІ СЛОВАМИ

1.1 Історичний огляд розробки ігор зі словами

За енциклопедичними визначеннями «гра» – це діяльність, що моделює інший вид діяльності з метою розваги або навчання. Гра зазвичай не ставить перед собою якоїсь корисної мети, хоча сама гра може бути корисною [1]. Зазвичай словом «гра» позначають різні не схожі між собою явища. Відзначається, що ця діяльність обумовлена сукупністю правил та прийомів.

Дослідники вважають, що прагнення грати – природне і властиве не лише людям, а навіть деяким тваринам. Але саме розвиток людської цивілізації породив велику кількість різноманітних ігор, наповнив їх різноманітним змістом. Ігри супроводжують нас все свідоме життя від самого народження. В дитинстві ми навчаємось через гру, в дорослому віці – наповнюємо грою своє дозвілля. Граючись, ми без сумніву отримуємо задоволення, але в той же час, велика кількість ігор приносить нам користь, розвиваючи певні здібності, формуючи ті чи інші навички тощо.

Історія зберегла згадки про ігри з прадавніх часів. З розвитком цивілізації ігри ускладнювались, урізноманітнювались і з часом стали офоплювати і відображати практично всі сфери життя людини та суспільства. Ще у XV ст. Йохан Хьойзінга стверджував, що поняття «людина, що грає» можна прирівняти до виразу «людина творить». Аристотель вбачав у грі джерело душевної рівноваги, гармонії душі й тіла. Платон підкреслював, що завдяки грі людина пізнає властивості своєї природи. Дж. Локк в педагогічних трактатах говорив, що ігри так само необхідні, як праця та їжа. І сьогодні багато дослідників підтверджують вплив ігор, в які ми граємо на формування особистості.

Таким чином, «гра» – це дуже багатомірне поняття, що включає в себе діяльність, пізнання, тренінг, творчість, спілкування, розвагу та творчість. Багато ігор передбачають високу фізичну активність, але є й такі, що

орієнтовані на інтелектуальну активність гравців. До таких ігор, зокрема відносяться так звані логічні ігри, у тому числі ігри зі словами.

За час розвитку мовлення та писемності з'явилося багато ігор зі словами. Всім відомі, наприклад, кросворди, ребуси та багато інших ігор (рис. 1.1). Відомо, що вони розвивають не лише словниковий запас, але й розвивають логіку, здатність до аналізу, прогнозування, асоціативного мислення тощо.

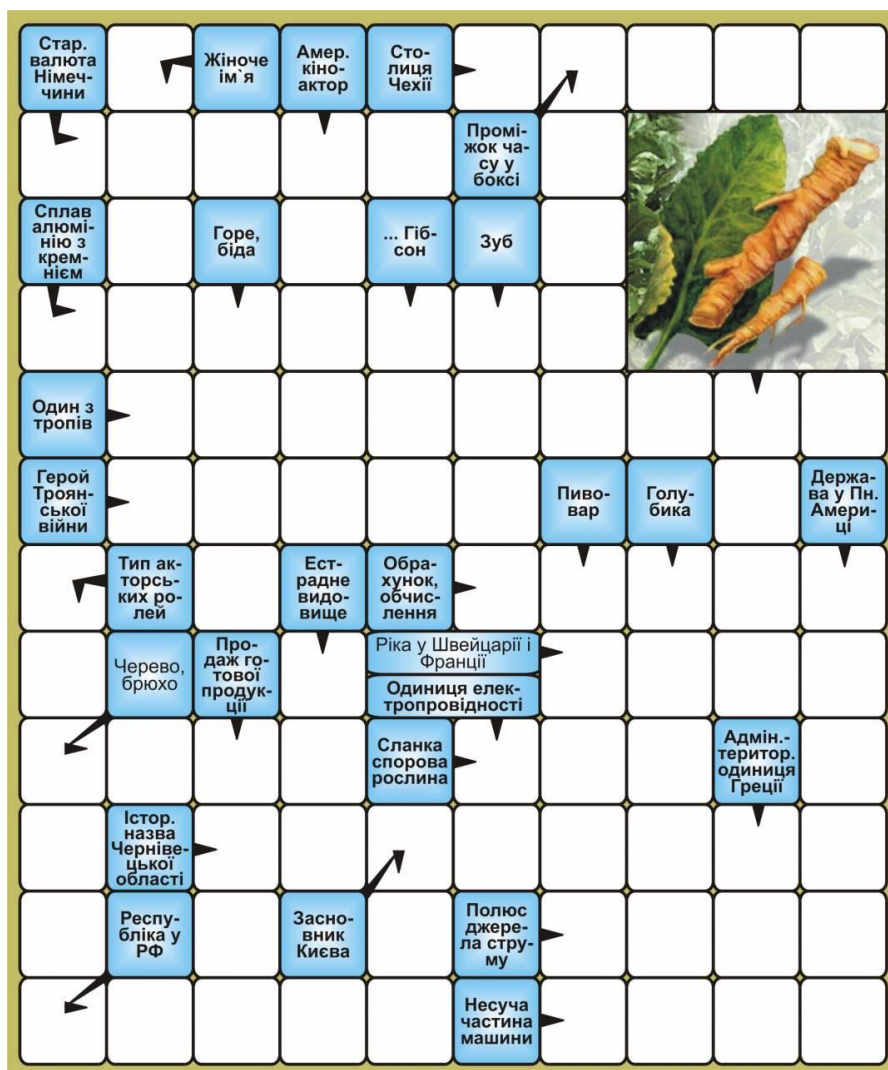


Рисунок 1.1 – Приклад сканворду

Широко відомою є, наприклад, гра «Слова зі слів». В цій грі учасники обирають довільне слово з великої кількості літер, а потім намагаються утворити з його букв інші слова. Умови перемоги в грі можуть варіюватись. Можна грати на час: хто більше складе слів за певний час. Можна пропонувати слова по черзі і програє той, хто не зможе запропонувати нового слова.

В великих компаніях популярна гра «Остання буква». В цій грі кожен учасник має сказати слово, що починається на останню букву попереднього слова. Слова не мають повторюватись. Можна звизити кількість дозволених слів якоюсь тематикою, наприклад, міста або професії.

Іншим прикладом ігри в компанії може слугувати гра «Контакт». В цій грі обирається ведучий, який загадує слово та називає його першу букву. Інші учасники намагаються вгадати його. Вони задають завуальоване питання, про слово, що починається з цієї букви. Якщо ведучий здогадався, яке слово мав на увазі гравець, то називає його і передає спробу наступному гравцю, не відкриваючи наступної букви. Якщо ведучий не здогадався, але здогадався інший гравець, то цей гравець каже «Контакт!» і на рахунок три він, та учасник, що загадував слово, називають це слово. Якщо слова гравців співпали, але це слово не є розгадкою, ведучий називає наступну букву. Гра відбувається, поки слово не буде вгадано.

Для деяких ігор можуть знадобитись найпростіші приладдя, наприклад, аркуш паперу та ручка або олівець. До таких ігор зі словами відносяться ігри «Балда» (рис. 1.2), «Слова на одну букву».

Для більш складних ігор може знадобитись спеціально підготовлене ігрове поле. Прикладом такої гри може стати на популярна настільна гра «Scrabble» (рис. 1.3). Відома також її українська адаптація, що отримала назву «Ерудит».

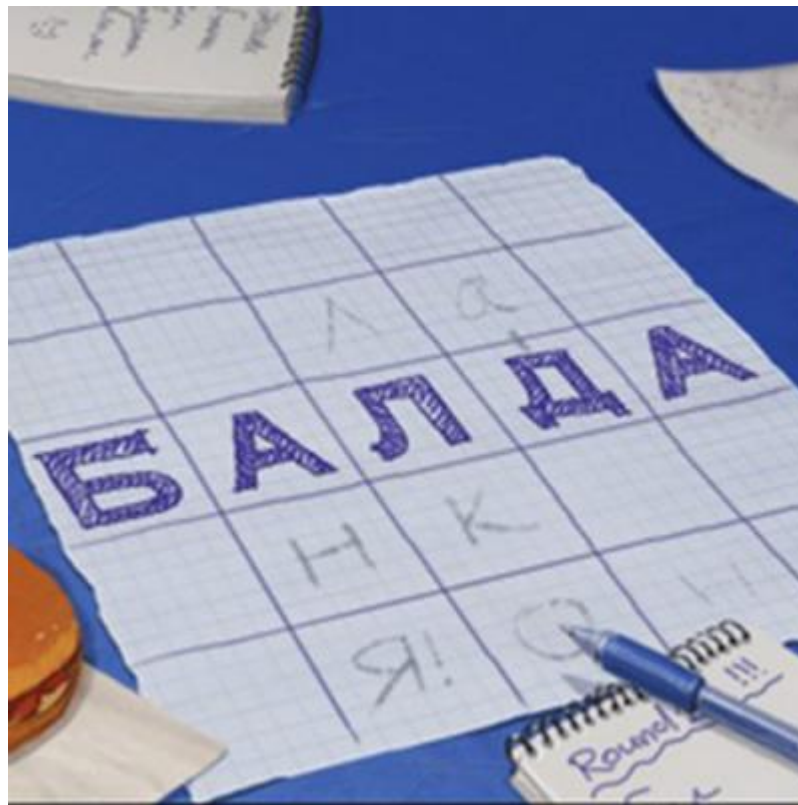


Рисунок 1.2 – Приклад гри «Балда»

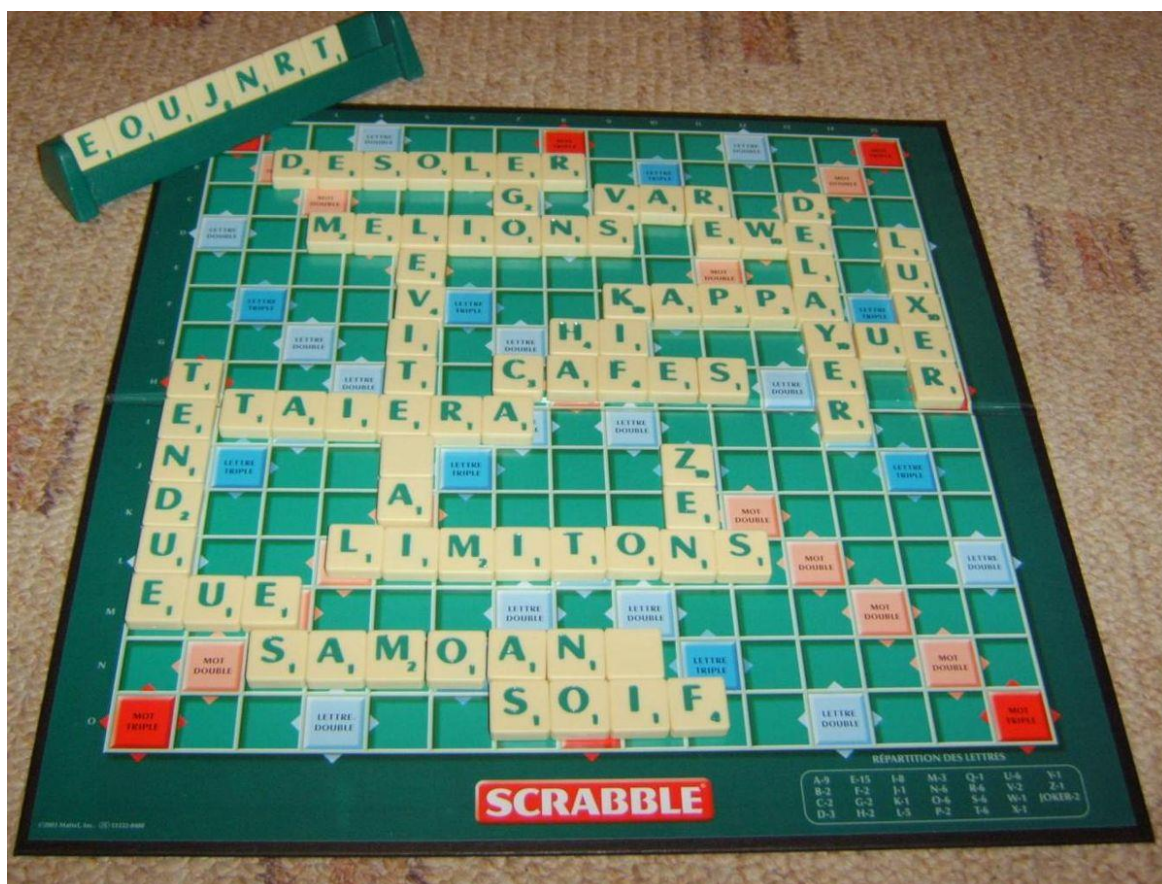


Рисунок 1.3 – Приклад ігрового поля «Scrabble»

Існує думка, що Scrabble добре розвиває лише словниковий запас. Проте це не так. Крім розвитку словникового запасу, гра навчає стратегії. Гравцю також доводиться продумувати свої наступні кроки в умовах невизначеності. Іноді цю гру порівнюють зі грою в шахи, але замість фігур – букви. Також відзначається, що Scrabble розвиває і просторове мислення. Гравцю доводиться багатосторонньо оцінювати ситуацію на полі, подумки викладати слова в різних комбінаціях, обирати той варіант, що може принести найбільше балів тощо. Відповідно також тренуються певною мірою і математичні уміння. Учасникам гри доводиться кожен раз підраховувати отримані бали. [2]

З появою комп'ютера з'явилась і нова категорія ігор – комп'ютерні ігри. Більшість комп'ютерних ігор мають виключно розважальний характер, акцент робиться на візуальних ефектах. Проте, існує і значний пласт логічних та розвиваючих комп'ютерних ігор. До таких ігор відносяться і ігри зі словами. Тож проаналізуємо, які на сьогодні ігри зі словами існують на ринку та мають популярність.

1.2 Огляд комп'ютерних ігор зі словами

Комп'ютерні ігри зі словами за сюжетом гри можна розділити на дві основні категорії:

- ігри на основі традиційних ігор зі словами;
- унікальні комп'ютерні ігри зі словами.

Ігри на основі традиційних, це, фактично, ті самі ігри, в які можна грати з папером та олівцем або взагалі без них, просто грати в них тепер можна на комп'ютері або мобільному пристрої. А унікальні комп'ютерні ігри були створенні саме для гри на комп'ютері, і без спеціального пристрою в них грати або неможливо, або занадто складно.

Комп'ютерна гра «Балда» є однією з найросповсюдженіших ігор зі словами, що має на нашу думку найбільше реалізацій. Є численні варіанти гри для комп'ютера, браузера, мобільних пристроїв. До того ж, це одна з найстаріших ігор зі словами, що були реалізовані на комп'ютері. Так, зокрема, відома реалізація цієї гри ще 1992 року [3] під операційну систему MS-DOS (рис. 1.4). В цьому варіанті гри потрібно було грати з комп'ютером.



Рисунок 1.4 – Комп'ютерна гра «Балда», 1992 р.

Пізніше з'явилась версія цієї ж гри під Windows (рис. 1.5).



Рисунок 1.5 – Приклад гри «Балда» під Windows

З появою Word Wide Web стали популярними онлайн ігри. У тому числі з'явилися реалізації браузерної онлайн гри «Балда» (рис. 1.6). Перевагою такого варіанту гри є те, що через Інтернет можна грати з іншою людиною, де б вона не знаходилась. Можна влаштовувати онлайн турніри з великою кількістю учасників тощо.



Рисунок 1.6 – Пример онлайн игры «Балда»

А з появою смартфонів почали з'являтися мобільні ігри. В тому числі з'явилися і мобільні реалізації «Балди» (рис. 1.7) [4]. Мобільні версії гри надають можливість грати не лише з комп'ютером або іншим гравцем онлайн, а й робити це в будь-якому місці, на відпочинку, в транспорті тощо.



Рисунок 1.7 – Приклад мобільної гри «Балда»

В англomовних країнах більш популярною грою зі словами була і залишається гра Scrabble. Ця гра також має багато комп'ютерних реалізацій під різними платформами. Зокрема відомими та популярними були реалізації під персональні комп'ютери на базі системи Windows (рис. 1.8).



Рисунок 1.8 – Версія Scrabble для IBM PC

Аналогічно гра Scrabble має численні браузерні варіанти (рис. 1.9)



Рисунок 1.9 – Онлайн версія гри Scrabble

І, звичайно ж, існує велика кількість мобільних версій гри Scrabble, зокрема така, як Word Master (рис. 1.10).

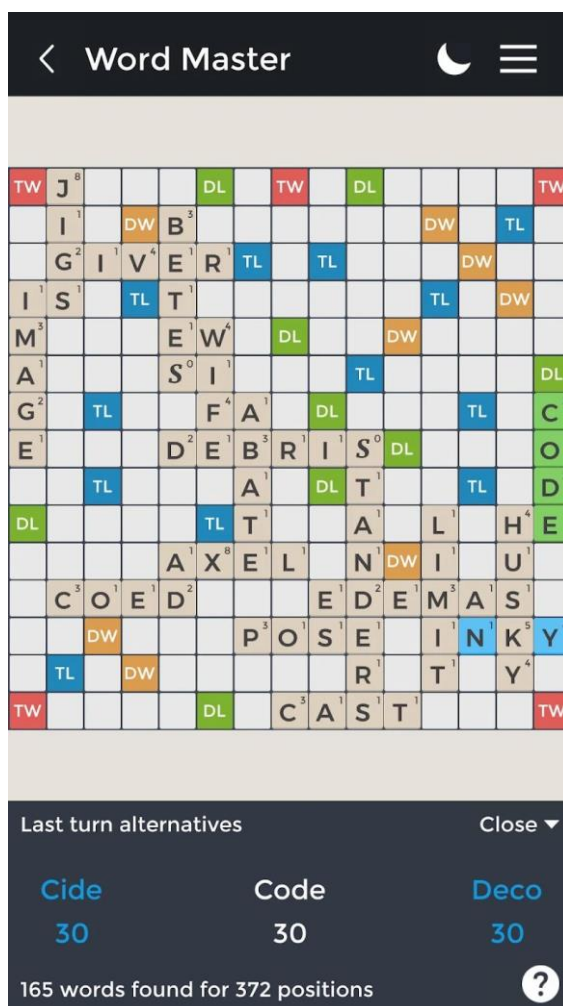


Рисунок 1.10 – Приклад мобільної реалізації гри Scrabble

Так само звичайні кросворди отримали комп'ютерні, браузерні та мобільні реалізації (рис. 1.11).



Рисунок 1.11 – Приклад мобільної реалізації кросворду

Цікавою реалізацією ігор зі словами є застосунок «Слова зі слова» (рис. 1.12).

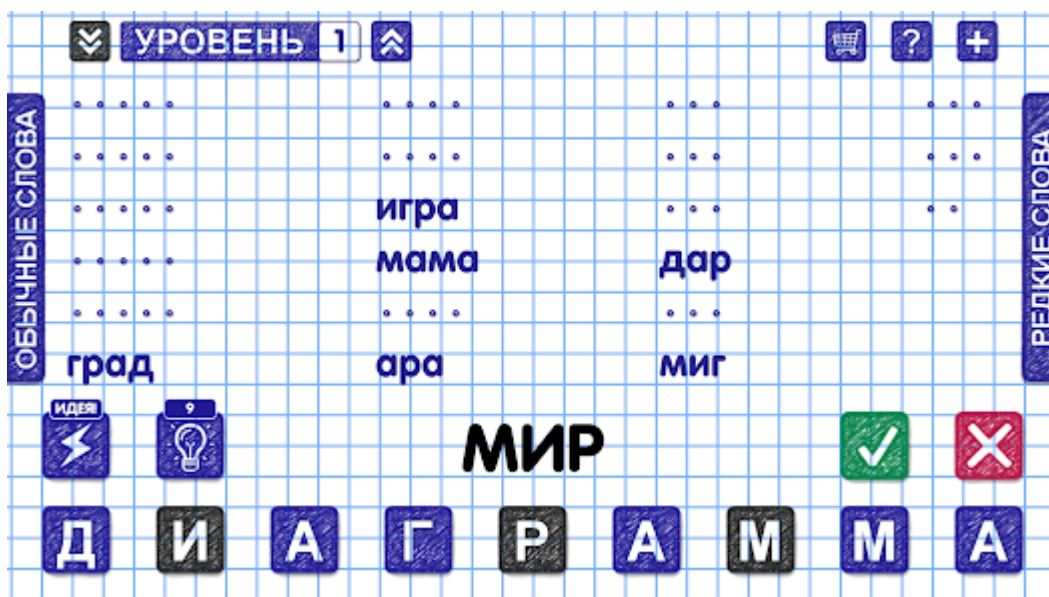


Рисунок 1.12 – Гра «Слова зі слова»

Цікавим є те, що цей застосунок пропонує цілий набір з мініігор, серед яких є такі як «Філворди», «Башта слів», «Королівський квадрат» (рис. 1.13).

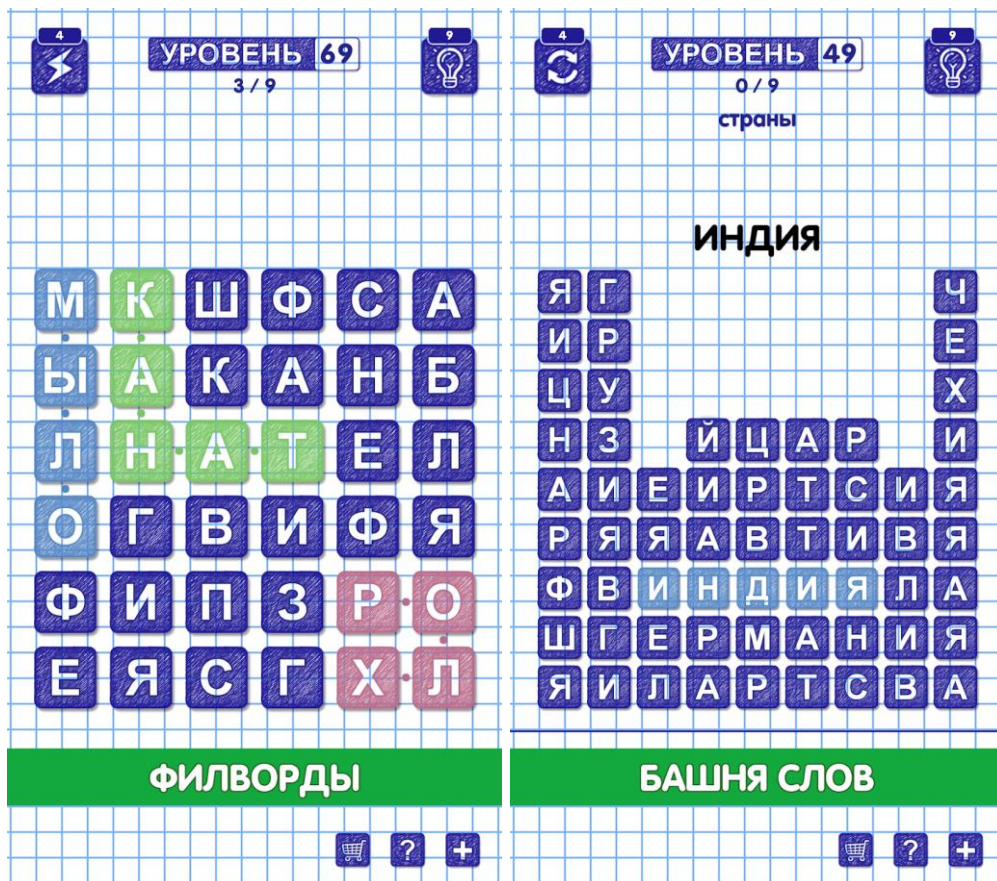


Рисунок 1.12 – Міні-ігри «Філворды», «Башта слів», «Королівський квадрат»

Але найбільш цікавими є ігри, що є суто комп'ютерними та не мають традиційних аналогів. Таких ігор також доволі багато. Яскравим прикладом є гра WOW (рис. 1.12). Однією з переваг гри є наявність її української локалізації.



Рисунок 1.12 – Мобільна гра зі словами «WOW»

А минулого року справжнім феноменом стала мобільна гра «Wordly» (рис. 1.13). Ця гра з'явилась у 2021 році і за кілька місяців стала шалено популярною. Про неї писали статті і робили телевізійні передачі.

Ідея гри дуже проста. Гравцю потрібно відгадати загадане слово за шість спроб. Після кожної спроби гравець бачить, які букви він вгадав. Якщо вгадав букву та її позицію, вона підсвічується зеленим. Якщо лише букву, то така буква підсвічується жовтим.

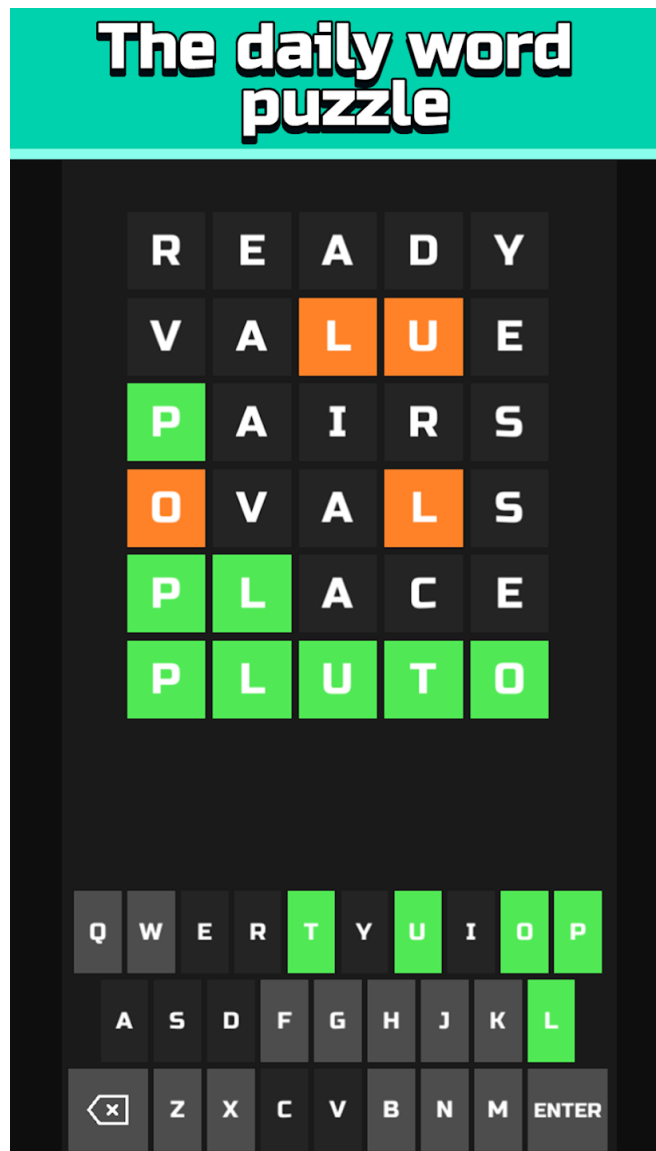


Рисунок 1.13 – Мобільна гра Wordly

Зважаючи на популярність та оригінальність гри Wordly в своїй кваліфікаційній роботі ми вирішили розробити україномовну версію цієї гри під назвою «Вгадай слово».

1.3 Визначення вимог до гри «Вгадай слово»

Вивчивши досвід розробок комп'ютерних ігор зі словами ми можемо сформулювати вимоги до гри «Вгадай слово».

Гра має випадковим чином обирати зі словника слово з 5 букв. Користувач не бачить цього слова і має вгадати його за шість спроб.

Під час кожної спроби користувач має вводити слово. Програма порівнює його з загаданим. Якщо користувач вгадав слово, гра закінчується. Якщо слова не існує, користувач має повторити спробу. Якщо слово не вгадано, програма має визначити, які букви вгадані. Якщо вгадана лише буква, вона має бути підсвічена жовтим кольором. Якщо вгадано букву і місце в слові, така буква підсвічена зеленим. Букви, які відсутні в загаданому слові, мають підсвічуватись сірим кольором.

Якщо гравець зробив шість спроб і не вгадав слово, гра закінчується.

Гра має вести статистику.

- скільки було загадано слів;
- скільки слів вгадано;
- скільки слів вгадано з 1, 2, 3, 4, 5 та 6-ї спроби.

2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ГРИ «ВГАДАЙ СЛОВО»

2.1 Функціональна схема програмного забезпечення гри «Вгадай слово»

Інтерфейс комп'ютерної гри зазвичай простий, розроблений так, щоб користувачу було зручно і він якомога більше уваги міг приділити самому процесу гри. Тому і функціональну схему ми максимально спрощуємо. Фактично гра складається з двох функціональних блоків (рис. 2.1):

- стартова сторінка;
- ігрове поле.

На стартовій сторінці користувач може переглянути статистику своїх результатів, а також кнопку для переходу до ігрового поля.

Ігрове поле власне призначено для реалізації гри. Серед структур даних, що використовуються під час гри, це список слів для гри та файл з результатами, в якому зберігається статистика гравця.

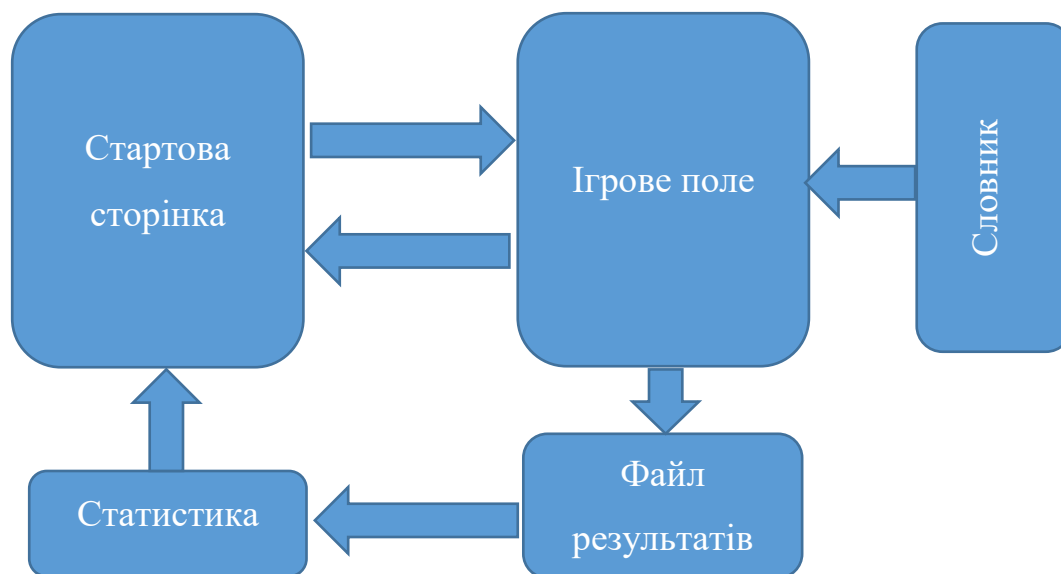


Рисунок 2.1 – Функціональна схема програмного забезпечення для реалізації навчальних квестів

2.2 Розробка основних алгоритмів гри «Вгадай слово»

Основний алгоритм гри «Вгадай слово» представлено на рисунку 2.2.

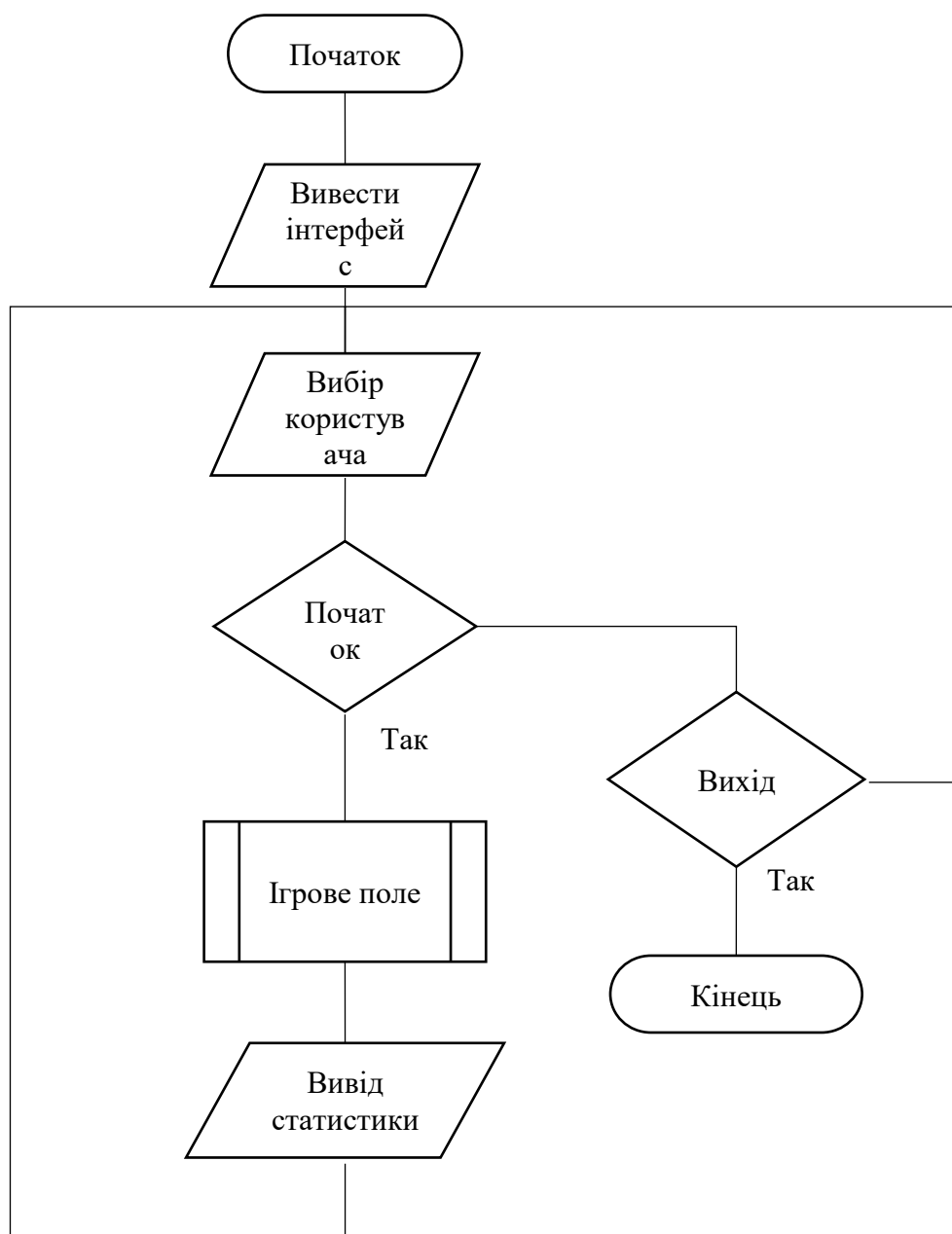


Рисунок 2.2 – Блок-схема основного алгоритму гри «Вгадай слово»

Робота програми починається з виведення інтерфейсу та пропозиції користувачу перейти до гри. Якщо користувач обирає «Почати гру», запускається модуль гри. Після сеансу гри, оновлюється статистика і користувач знову повертається до вибору. Якщо користувач обирає завершити програму, робота алгоритму закінчується.

Алгоритм модуля гри детально представлено на рисунку 2.3.

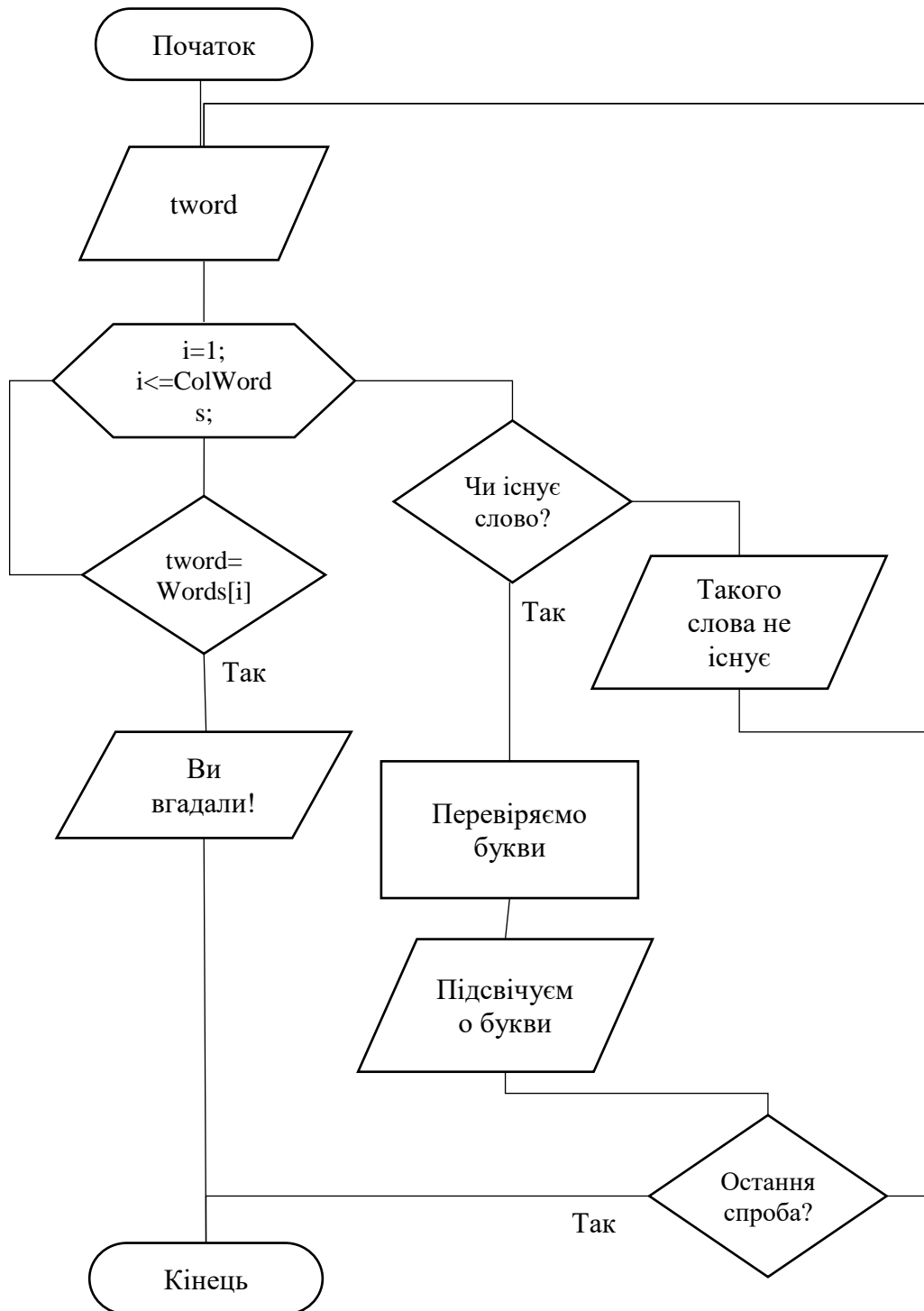


Рисунок 2.3 – Блок-схема алгоритму ігрового модуля

Гра починається з того, що користувач робить спробу і вводить слово.

Алгоритм перевіряє, чи співпадає це слово з загаданим. Якщо так, то виводиться повідомлення, що «Ви вгадали слово!».

Якщо слово не вгадано, то алгоритм перевіряє, чи існує це слово взагалі і порівнює введене слово з кожним словом у словнику. Якщо співпадінь немає, то виводиться повідомлення, що такого слова не існує і користувачу пропонується ввести інше.

Якщо слово є в словнику, то наступним етапом є визначення літер, які користувач вгадав. Якщо літера вгадана і по місцю, то вона підсвічується зеленим, якщо вгадана лише буква, то вона підсвічується жовтим.

Букви, яких в загаданому слові немає, підсвічуються сірим.

Після підсвічування букв перевіряється, чи не остання це була спроба.

Якщо це не остання спроба, то користувачу пропонується ввести інше слово.

Якщо спроба остання, то виводиться повідомлення, що ви не вгадали слово, і алгоритм завершується.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ГРИ «ВГАДАЙ СЛОВО»

3.1 Вибір мови програмування та засобів розробки гри «Вгадай слово»

Для розробки програмного забезпечення гри «Вгадай слово» нами було обрано одну з найпопулярніших та найуніверсальніших мов – C++. Сьогодні існує багато компіляторів та інтегрованих середовищ розробки, які використовують мову C++ для розробки програм під різні платформи. Найпопулярнішими є ті, що надають можливість створювати кросплатформні застосунки. Одним з таких середовищ є C++ Builder, що входить до IDE RAD Studio.

C++ Builder – це середовище швидкої розробки додатків (RAD) для побудови програм C++, які можуть бути орієнтовані на Windows (як IA-32, так і X64), iOS, MacOS та Android. Спочатку він був розроблений Borland і зараз належить Embarcadero Technologies.

C++ Builder поєднує в собі бібліотеку візуальних компонентів та IDE з декількома C++ компіляторами. Завдяки цьому в RAD Studio дійсно можна швидко розробляти програми, будувати інтерфейси користувача, писати код, що може без змін переноситись на різні платформи. Це відкриває значні можливості щодо подальшого розвитку програм і робить саме це середовище привабливим для розробки.

3.2 Проектування інтерфейсу гри «Вгадай слово»

Розробка інтерфейсу стартової форми гри «Вгадай слово» зображено на рисунку 3.1. На стартовій формі розташовані компоненти для відображення фонового рисунку, назви гри, привітання користувача. Також на стартовій формі відображається статистика ігор:

- скільки було загадано слів;
- скільки слів вгадано;
- скільки слів вгадано з 1, 2, 3, 4, 5 та 6-ї спроби.

Також на стартовій формі розташовано кнопку «Почати гру!»

Після натискання на цю кнопку користувач має перейти до ігрового поля і може робити спроби вгадати слово.

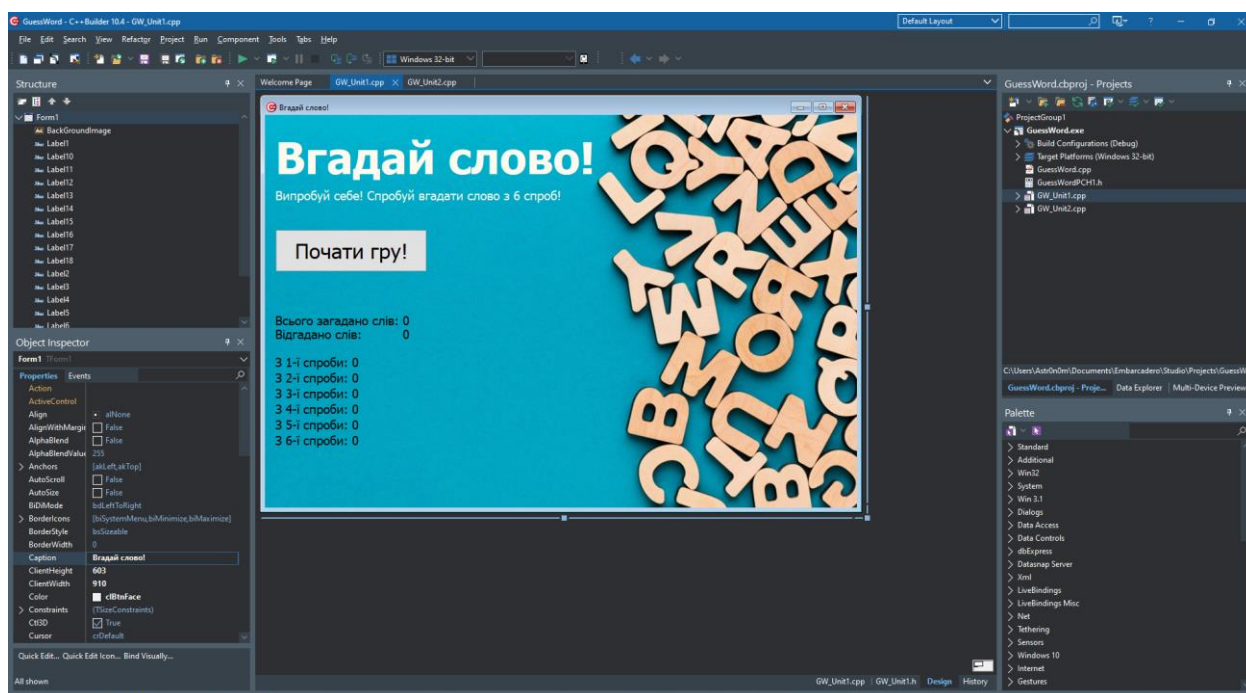


Рисунок 3.1 – Проектування інтерфейсу стартової форми гри

Макет інтерфейсу ігрового поля показано на рисунку 3.2.

На ігровому полі присутні компоненти для послідовного відображення шести спроб, що надаються користувачу для вгадування слова. Кожен цей компонент може змінювати свій колір, що необхідно нам для підсвічування літер.

Під полем для спроб розміщуються кнопки з буквами. Це необхідно, щоб користувач, коли робить спробу, також бачив, які букви присутні в слові, а які відсутні.

Також на формі присутній окремий компонент для зберігання та редагування словника. Цей компонент не буде бачити користувач під час роботи програми.

Також присутня кнопка «Зробити спробу», натиснувши на яку користувач може перевірити, чи вгадав він слово.

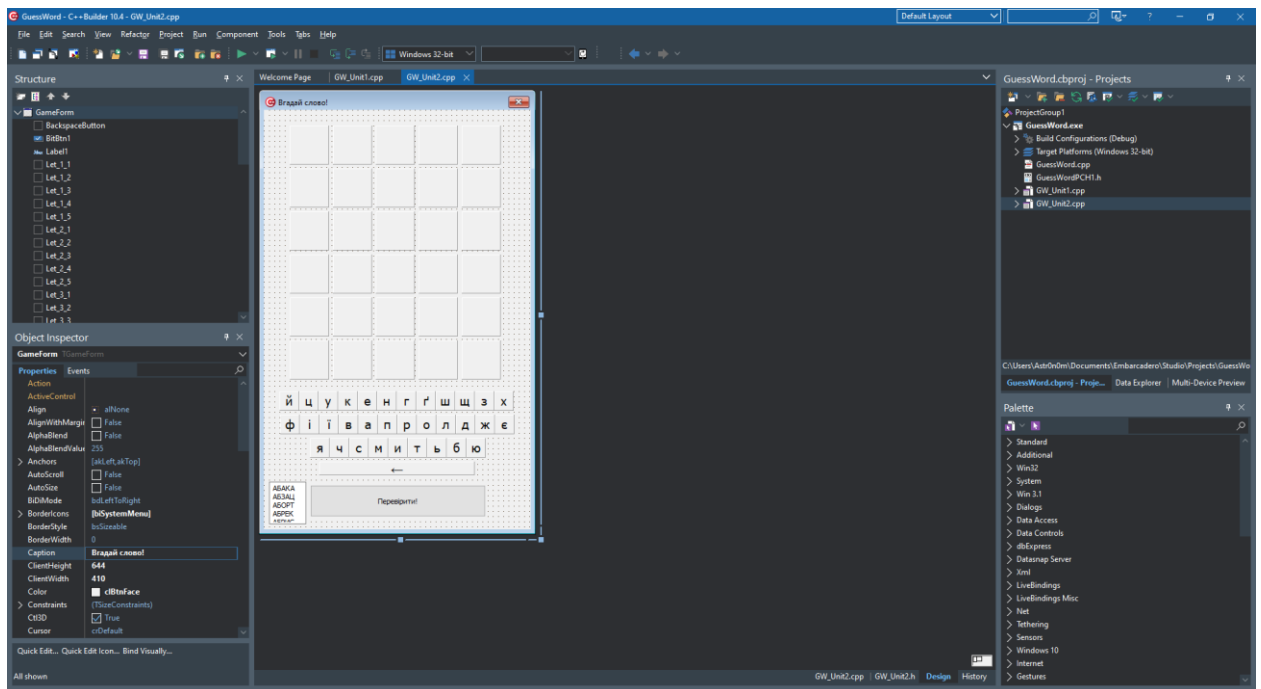


Рисунок 3.2 – Проектування інтерфейсу ігрового поля

3.3 Приклади роботи програми

Робота програми починається з відображення вітальної форми (рис. 3.3). На цій форми розміщується привітання користувача, назва гри та заохочення до гри.

Внизу вікна відображається статистика гравця – кількість спроб, кількість вгаданих слів, скільки слів з якої спроби було вгадано.

При натисканні на кнопку «Почати гру!» починається сама гра, відкривається вікно з основним полем гри (рис. 3.4).

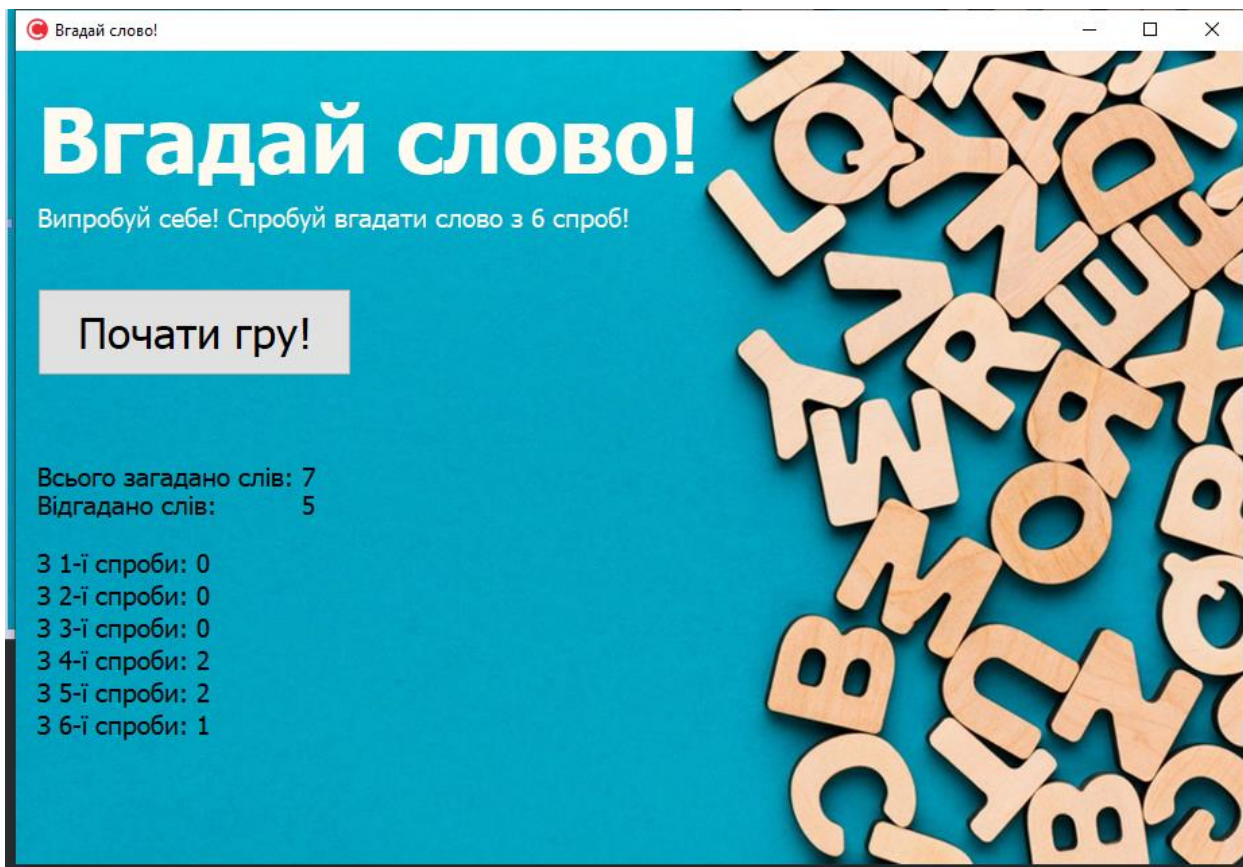


Рисунок 3.3 – Вітальна форма гри

Ігрове поле на початку гри чисте, жодна буква не підсвічена. Користувач може вводити букви, щоб сформувати перше слово (рис. 3.5).



Рисунок 3.4 – Ігрове поле на початку гри

Після того, як користувач ввів по буквах слово, гравець може натиснути кнопку «Перевірити», щоб взнати вгадав він слово або ні, і які букви він вгадав, які ні.



Рисунок 3.5 – Приклад введення слова

Результат перевірки введеного слова показано на рисунку 3.6. Ті букви, яких немає в загаданому слові, підсвічені сірим кольором. Якщо буква вгадана, але не на своєму місці, то вона підсвічується жовтим кольором.



Рисунок 3.6 – Результат перевірки слова

Якщо вгадана буква і її місце в слові, то така буква підсвічується зеленим кольором (рис. 3.7).

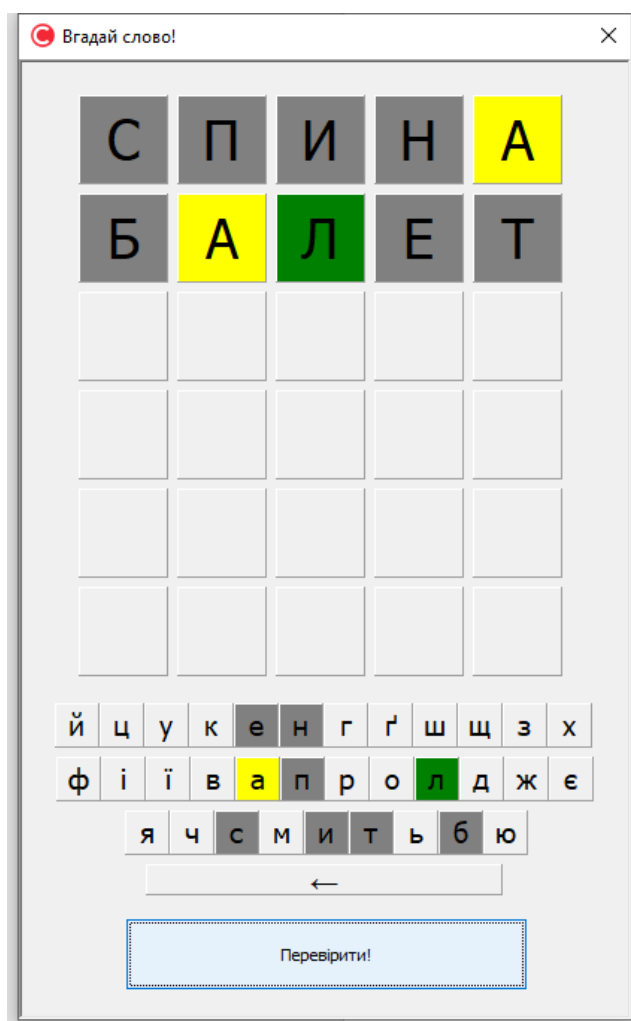


Рисунок 3.7 – Приклад виділення вгаданих букв

Якщо користувач вводить слово, якого не існує, то відображається повідомлення «Такого слова не існує! Повторіть спробу.» (рис. 3.8).

В разі введення неіснуючого слова, користувач не втрачає спробу і може повторити спробу, ввівши інше слово.

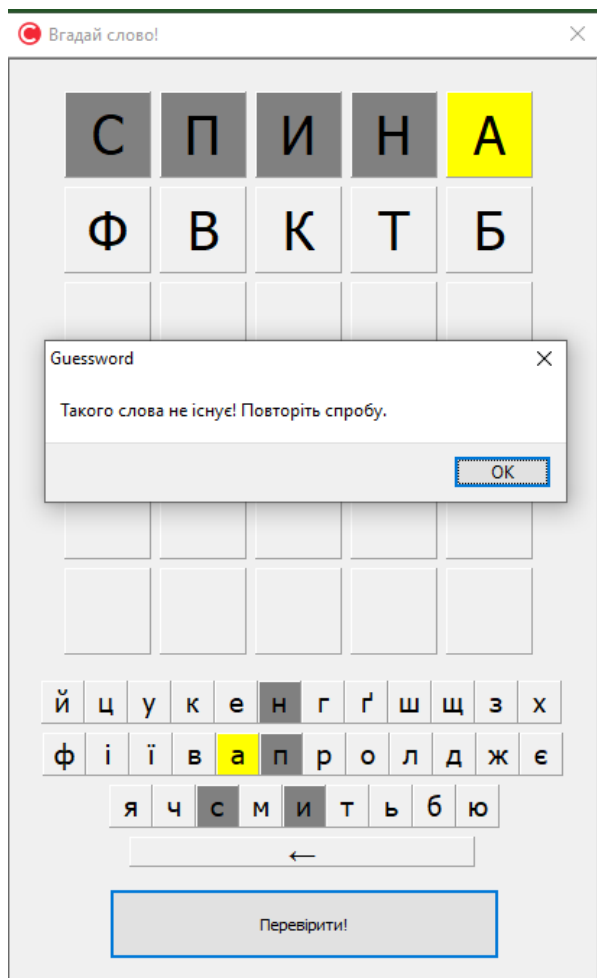


Рисунок 3.8 – Повідомлення при введенні слова, якого не існує

Якщо гравець вгадав слово, відображається повідомлення «Вітаємо! Ви вгадали слово!» (рис. 3.9).

Після успішно вгаданого слова, користувач повертається до вітальної форми, може перевірити зміни статистики та за бажання зіграти ще одну гру.

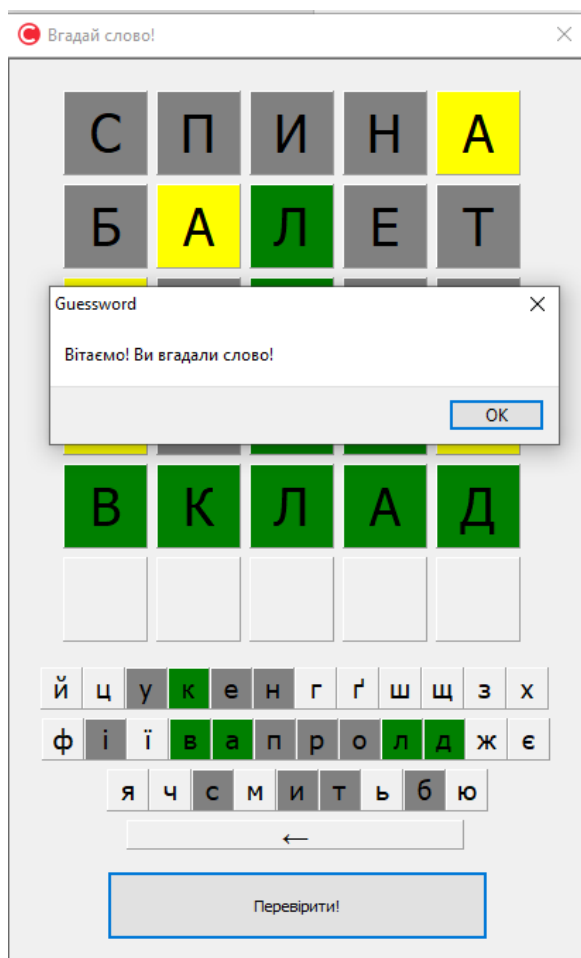


Рисунок 3.9 – Повідомлення про те, що користувач вгадав слово

Якщо користувач використав всі шість спроб, але не зміг вгадати слово, то після шостої спроби відображається повідомлення «Прикро, але ви не вгадали слово» (рис. 3.10).

Після цього користувач повертається до вітальної форми і отримує можливість зіграти ще раз, але вже з іншим словом.

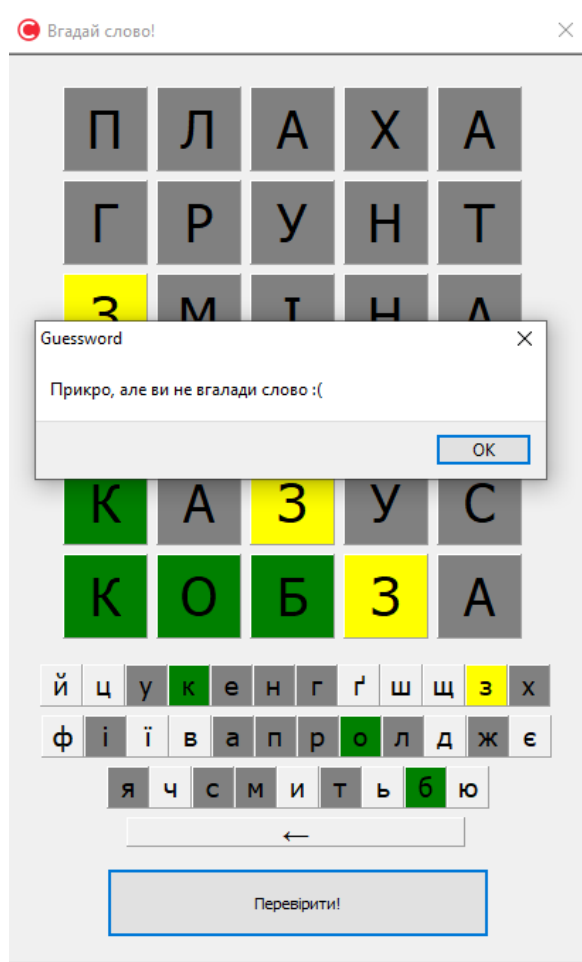


Рисунок 3.10 – Приклад повідомлення, коли слово не вгадано

ВИСНОВКИ

В представленій кваліфікаційній роботі було спроектовано та реалізовано програмне забезпечення гри «Вгадай слово».

Нами було зроблено історичний огляд розробки ігор зі словами. Визначено, що вони розвивають не лише словниковий запас, але й розвивають логіку, здатність до аналізу, прогнозування, асоціативного мислення тощо.

Зроблено огляд комп'ютерних ігор зі словами. Визначено, що комп'ютерні ігри зі словами за сюжетом гри можна розділити на дві основні категорії:

- ігри на основі традиційних ігор зі словами;
- унікальні комп'ютерні ігри зі словами.

Ігри на основі традиційних, це, фактично, ті самі ігри, в які можна грати з папером та олівцем або взагалі без них, просто грати в них тепер можна на комп'ютері або мобільному пристрої. А унікальні комп'ютерні ігри були створенні саме для гри на комп'ютері, і без спеціального пристрою в них грати або неможливо, або занадто складно.

Цікавою розробкою останніх років стала мобільна гра «Wordly». Ідея гри дуже проста. Гравцю потрібно відгадати загадане слово за шість спроб. Після кожної спроби гравець бачить, які букви він вгадав. Якщо вгадав букву та її позицію, вона підсвічується зеленим. Якщо лише букву, то така буква підсвічується жовтим.

Зважаючи на популярність та оригінальність гри Wordly в своїй кваліфікаційній роботі ми вирішили розробити україномовну версію цієї гри під назвою «Вгадай слово».

Враховуючи досвід інших розробок ігор зі словами, ми сформулювали вимоги до гри «Вгадай слово».

Інтерфейс комп'ютерної гри зазвичай простий, розроблений так, щоб користувачу було зручно і він якомога більше уваги міг приділити самому

процесу гри. Тому і функціональну схему ми максимально спрощуємо. Фактично гра складається з двох функціональних блоків (рис. 2.1):

- стартова сторінка;
- ігрове поле.

На стартовій сторінці користувач може переглянути статистику своїх результатів, а також кнопку для переходу до ігрового поля.

Ігрове поле власне призначено для реалізації гри. Серед структур даних, що використовуються під час гри, це список слів для гри та файл з результатами, в якому зберігається статистика гравця.

Спроектовано основні алгоритми гри «Вгадай слово», серед яких можна виділити алгоритм загальної організації гри та алгоритм перевірки слова та вгаданих гравцем букв.

Для розробки програмного забезпечення гри «Вгадай слово» нами було обрано одну з найпопулярніших та найуніверсальніших мов – C++. А середовищем розробки обрано C++ Builder.

За допомогою інструментальних засобів було спроектовано інтерфейс гри «Вгадай слово», реалізовані алгоритми та протестовано програмне забезпечення.

ПЕРЕЛІК ПОСИЛАНЬ

1. Гра – Вікіпедія [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/%D0%93%D1%80%D0%B0>
2. Ігри в слова: легендарний Scrabble чи інноваційна Pictionary air [Електронний ресурс]. – Режим доступу: <https://bi.ua/ukr/posts/view/igry-v-slova-legendarnyy-scrabble-ili-innovacionnaya-pictionary-air/>
3. Балда [Електронний ресурс]. – Режим доступу: <https://www.old-games.ru/game/5179.html>
4. Балда. Игра в слова [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.sayrex.balda&hl=uk&gl=US>
5. Угадай Слово [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.redboxsoft.guesstheword>
6. Собиратель слов: находи новые слова [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.krutigrama.games.android.words.collector>
7. Слова із Слова 2 [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.justanothertry.slovavk>
8. Word Challenge - Fun Word Game [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.robus.wordgame>
9. Эрудит: скраббл на русском [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.oxothuk.erudit>
10. Крпнто Кроссворды [Електронний ресурс]. – Режим доступу: https://play.google.com/store/apps/details?id=com.fgcoss.crypto_crosswords_ru_kripto_krossvordu
11. Кросвордіум: кросворди українською [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.kulbabix.crosswordium>

12. Архангельский А. Я. Компоненты C++Builder. Справочное и методическое пособие / А. Я. Архангельский. – М.: Бином-Пресс, 2008. – 960 с.
13. Архангельский А. Я. Язык C++ в C++Builder / А. Я. Архангельский. – М.: Бином-Пресс, 2008. – 944 с.
14. C++ Builder [Электронный ресурс]. – Режим доступа: https://uk.wikipedia.org/wiki/C%2B%2B_Builder
15. Швидка розробка програмного забезпечення [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/%D0%A8%D0%B2%D0%B8%D0%B4%D0%BA%D0%B0_%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F

Додаток А
Текст програми

```
//-----  
  
#ifndef GW_Unit1H  
#define GW_Unit1H  
//-----  
#include <System.Classes.hpp>  
#include <Vcl.Controls.hpp>  
#include <Vcl.StdCtrls.hpp>  
#include <Vcl.Forms.hpp>  
#include <Vcl.ExtCtrls.hpp>  
#include <Vcl.Imaging.jpeg.hpp>  
#include <Vcl.Buttons.hpp>  
//-----  
class TForm1 : public TForm  
{  
  __published:// IDE-managed Components  
  TImage *BackGroundImage;  
  TLabel *Label1;  
  TLabel *Label2;  
  TSpeedButton *SpeedButton1;  
  TLabel *Label3;  
  TLabel *Label4;  
  TLabel *Label5;  
  TLabel *Label6;  
  TLabel *Label7;  
  TLabel *Label8;  
  TLabel *Label9;  
  TLabel *Label10;  
  TLabel *Label11;  
  TLabel *Label12;  
  TLabel *Label13;  
  TLabel *Label14;  
  TLabel *Label15;  
  TLabel *Label16;  
  TLabel *Label17;  
  TLabel *Label18;  
  void __fastcall SpeedButton1Click(TObject *Sender);  
  void __fastcall FormCreate(TObject *Sender);  
  void __fastcall FormShow(TObject *Sender);  
private:      // User declarations
```

```

public:          // User declarations
__fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif

//-----

#include <vcl.h>
#pragma hdrstop

#include "GW_Unit1.h"
#include "GW_Unit2.h"

//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm1::SpeedButton1Click(TObject *Sender)
{
    GameForm->Show();
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    // GameForm->Hide();
}
//-----
void __fastcall TForm1::FormShow(TObject *Sender)
{
    GameForm->Hide();
}
//-----

```

```

//-----
#ifndef GW_Unit2H
#define GW_Unit2H
//-----
#include <System.Classes.hpp>
#include <Vcl.Controls.hpp>
#include <Vcl.StdCtrls.hpp>
#include <Vcl.Forms.hpp>
#include <Vcl.Buttons.hpp>
#include <Vcl.ExtCtrls.hpp>
//-----
class TGameForm : public TForm
{
__published:// IDE-managed Components
TPanel *Let_1_1;
TPanel *Let_1_2;
TPanel *Let_1_3;
TPanel *Let_1_4;
TPanel *Let_1_5;
TPanel *Let_2_1;
TPanel *Let_2_2;
TPanel *Let_2_3;
TPanel *Let_2_4;
TPanel *Let_2_5;
TPanel *Let_3_1;
TPanel *Let_3_2;
TPanel *Let_3_3;
TPanel *Let_3_4;
TPanel *Let_3_5;
TPanel *Let_4_1;
TPanel *Let_4_2;
TPanel *Let_4_3;
TPanel *Let_4_4;
TPanel *Let_4_5;
TPanel *Let_5_1;
TPanel *Let_5_2;
TPanel *Let_5_3;
TPanel *Let_5_4;
TPanel *Let_5_5;
TPanel *Let_6_1;
TPanel *Let_6_2;
TPanel *Let_6_3;
TPanel *Let_6_4;
TPanel *Let_6_5;

```

```

TPanel *Let1_Button;
TPanel *Let2_Button;
TPanel *Let3_Button;
TPanel *Let4_Button;
TPanel *Let5_Button;
TPanel *Let6_Button;
TPanel *Let12_Button;
TPanel *Let11_Button;
TPanel *Let10_Button;
TPanel *Let9_Button;
TPanel *Let8_Button;
TPanel *Let7_Button;
TPanel *Let13_Button;
TPanel *Let14_Button;
TPanel *Let15_Button;
TPanel *Let16_Button;
TPanel *Let17_Button;
TPanel *Let18_Button;
TPanel *Let24_Button;
TPanel *Let23_Button;
TPanel *Let22_Button;
TPanel *Let21_Button;
TPanel *Let20_Button;
TPanel *Let19_Button;
TPanel *Let25_Button;
TPanel *Let26_Button;
TPanel *Let27_Button;
TPanel *Let28_Button;
TPanel *Let29_Button;
TPanel *Let33_Button;
TPanel *Let32_Button;
TPanel *Let31_Button;
TPanel *Let30_Button;
TPanel *BackspaceButton;
TBitBtn *BitBtn1;
TMemo *ListWords;
TLabel *Label1;
void __fastcall Let1_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let1_ButtonClick(TObject *Sender);
void __fastcall Let1_ButtonMouseActivate(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y, int HitTest, TMouseActivate &MouseActivate);
void __fastcall Let1_ButtonMouseLeave(TObject *Sender);

```

```

void __fastcall Let1_ButtonMouseUp(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall CheckButtonClick(TObject *Sender);
void __fastcall FormCreate(TObject *Sender);
void __fastcall BitBtn1Click(TObject *Sender);
void __fastcall Let3_ButtonClick(TObject *Sender);
void __fastcall Let2_ButtonClick(TObject *Sender);
void __fastcall Let4_ButtonClick(TObject *Sender);
void __fastcall Let5_ButtonClick(TObject *Sender);
void __fastcall Let6_ButtonClick(TObject *Sender);
void __fastcall Let7_ButtonClick(TObject *Sender);
void __fastcall Let8_ButtonClick(TObject *Sender);
void __fastcall Let9_ButtonClick(TObject *Sender);
void __fastcall Let10_ButtonClick(TObject *Sender);
void __fastcall Let11_ButtonClick(TObject *Sender);
void __fastcall Let12_ButtonClick(TObject *Sender);
void __fastcall Let13_ButtonClick(TObject *Sender);
void __fastcall Let14_ButtonClick(TObject *Sender);
void __fastcall Let15_ButtonClick(TObject *Sender);
void __fastcall Let16_ButtonClick(TObject *Sender);
void __fastcall Let17_ButtonClick(TObject *Sender);
void __fastcall Let18_ButtonClick(TObject *Sender);
void __fastcall Let19_ButtonClick(TObject *Sender);
void __fastcall Let20_ButtonClick(TObject *Sender);
void __fastcall Let21_ButtonClick(TObject *Sender);
void __fastcall Let22_ButtonClick(TObject *Sender);
void __fastcall Let23_ButtonClick(TObject *Sender);
void __fastcall Let24_ButtonClick(TObject *Sender);
void __fastcall Let25_ButtonClick(TObject *Sender);
void __fastcall Let26_ButtonClick(TObject *Sender);
void __fastcall Let27_ButtonClick(TObject *Sender);
void __fastcall Let28_ButtonClick(TObject *Sender);
void __fastcall Let29_ButtonClick(TObject *Sender);
void __fastcall Let30_ButtonClick(TObject *Sender);
void __fastcall Let31_ButtonClick(TObject *Sender);
void __fastcall Let32_ButtonClick(TObject *Sender);
void __fastcall Let33_ButtonClick(TObject *Sender);
void __fastcall BackspaceButtonClick(TObject *Sender);
void __fastcall Let2_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let3_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);

```

```

void __fastcall Let4_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let5_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let6_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let7_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let8_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let9_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let10_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let11_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let12_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let13_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let14_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let15_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let16_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let17_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let18_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);

```



```

void __fastcall Let19_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let20_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let21_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let22_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let23_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let24_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let25_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let26_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let27_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let28_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let29_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let30_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let31_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let32_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);
void __fastcall Let33_ButtonMouseDown(TObject *Sender, TMouseButton
Button, TShiftState Shift,
    int X, int Y);

```

```

void __fastcall BackspaceButtonMouseDown(TObject *Sender,
TMouseButton Button, TShiftState Shift,
int X, int Y);
void __fastcall FormShow(TObject *Sender);
void __fastcall FormClose(TObject *Sender, TCloseAction &Action);
void __fastcall FormDbClick(TObject *Sender);
private: // User declarations
public: // User declarations
__fastcall TGameForm(TComponent* Owner);
};
//-----
extern PACKAGE TGameForm *GameForm;
//-----
#endif

//-----

#include <vcl.h>
#pragma hdrstop

#include "GW_Unit2.h"
#include "GW_Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TGameForm *GameForm;

TPanel *Words[7][6];
int i,j;
String SecretWord;
String
LettersList="ЙЦУКЕНГТШЩЗХФЇВАПРОЛДЖЄЯЧСМИТЬБЮ";
TPanel *LetPanel[33];

//-----
__fastcall TGameForm::TGameForm(TComponent* Owner)
: TForm(Owner)
{
}
//-----

```

```

void __fastcall TGameForm::Let1_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
        TShiftState Shift, int X, int Y)
{
Let1_Button->Color=clGray;
}
//-----
void __fastcall TGameForm::Let1_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="Й";
j++;
}
Let1_Button->Color=clBtnFace;
//Let2_Button->Caption="й";
}
//-----
void __fastcall TGameForm::Let1_ButtonMouseActivate(TObject *Sender,
TMouseButton Button,
        TShiftState Shift, int X, int Y, int HitTest, TMouseActivate
&MouseActivate)

{
Let1_Button->Color=clWhite;
}
//-----
void __fastcall TGameForm::Let1_ButtonMouseLeave(TObject *Sender)
{
Let1_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let1_ButtonMouseUp(TObject *Sender,
TMouseButton Button,
        TShiftState Shift, int X, int Y)
{
//Let1_Button->Color=clWhite;
}
//-----
void __fastcall TGameForm::CheckButtonClick(TObject *Sender)
{
//
if(i<=6)
        {i++; j=1;}
}

```

```

//-----
void __fastcall TGameForm::FormCreate(TObject *Sender)
{
//
Words[1][1]=Let_1_1;
Words[1][2]=Let_1_2;
Words[1][3]=Let_1_3;
Words[1][4]=Let_1_4;
Words[1][5]=Let_1_5;

Words[2][1]=Let_2_1;
Words[2][2]=Let_2_2;
Words[2][3]=Let_2_3;
Words[2][4]=Let_2_4;
Words[2][5]=Let_2_5;

Words[3][1]=Let_3_1;
Words[3][2]=Let_3_2;
Words[3][3]=Let_3_3;
Words[3][4]=Let_3_4;
Words[3][5]=Let_3_5;

Words[4][1]=Let_4_1;
Words[4][2]=Let_4_2;
Words[4][3]=Let_4_3;
Words[4][4]=Let_4_4;
Words[4][5]=Let_4_5;

Words[5][1]=Let_5_1;
Words[5][2]=Let_5_2;
Words[5][3]=Let_5_3;
Words[5][4]=Let_5_4;
Words[5][5]=Let_5_5;

Words[6][1]=Let_6_1;
Words[6][2]=Let_6_2;
Words[6][3]=Let_6_3;
Words[6][4]=Let_6_4;
Words[6][5]=Let_6_5;

LetPanel[0]=Let1_Button;
LetPanel[1]=Let2_Button;
LetPanel[2]=Let3_Button;
LetPanel[3]=Let4_Button;
LetPanel[4]=Let5_Button;

```

```

LetPanel[5]=Let6_Button;
LetPanel[6]=Let7_Button;
LetPanel[7]=Let8_Button;
LetPanel[8]=Let9_Button;
LetPanel[9]=Let10_Button;
LetPanel[10]=Let11_Button;
LetPanel[11]=Let12_Button;
LetPanel[12]=Let13_Button;
LetPanel[13]=Let14_Button;
LetPanel[14]=Let15_Button;
LetPanel[15]=Let16_Button;
LetPanel[16]=Let17_Button;
LetPanel[17]=Let18_Button;
LetPanel[18]=Let19_Button;
LetPanel[19]=Let20_Button;
LetPanel[20]=Let21_Button;
LetPanel[21]=Let22_Button;
LetPanel[22]=Let23_Button;
LetPanel[23]=Let24_Button;
LetPanel[24]=Let25_Button;
LetPanel[25]=Let26_Button;
LetPanel[26]=Let27_Button;
LetPanel[27]=Let28_Button;
LetPanel[28]=Let29_Button;
LetPanel[29]=Let30_Button;
LetPanel[30]=Let31_Button;
LetPanel[31]=Let32_Button;
LetPanel[32]=Let33_Button;

```

```

i=1;
j=1;

```

```

}

```

```

//-----

```

```

void __fastcall TGameForm::BitBtn1Click(TObject *Sender)

```

```

{ int indx, indx2,indx3;

```

```

  String tword;

```

```

  bool WordCheck;

```

```

if (j>5) // ЯКЩО ПОВНІСТЮ ВВЕДЕНО СЛОВО

```

```

{ // Збираємо слово по буквам

```

```

    tword=Words[i][1]->Caption+Words[i][2]->Caption+Words[i][3]-

```

```

>Caption+

```

```

    Words[i][4]->Caption+Words[i][5]->Caption;

```

```

// Перевіряємо, чи вгадали ми слово
if (tword==SecretWord) {
    for(indx=1;indx<=5;indx++) {
        Words[i][indx]->Color=clGreen;
        for(indx2=0;indx2<=32;indx2++){
            if(Words[i][indx]-
>Caption==LettersList[indx2+1])LetPanel[indx2]->Color=clGreen;
        }
    }

    ShowMessage("Вітаємо! Ви вгадали слово!");
    GameForm->Hide();
    Form1->Show();
    return;
}

// Перевіряємо, чи взагалі слово існує
WordCheck=true;
for(indx=0;indx<ListWords->Lines->Count;indx++)
{
    if(tword==ListWords->Lines-
>Strings[indx])WordCheck=false;
}
if(WordCheck){
    ShowMessage("Такого слова не існує! Повторіть спробу.");
    return;
}

// Перевіряємо, які букви вгадані, а які ні
for(indx=1;indx<=5;indx++){

    if(tword[indx]==SecretWord[indx]){ // Якщо вгадали букву і
місце
        Words[i][indx]->Color=clGreen;
        for(indx2=0;indx2<=32;indx2++){
            if(Words[i][indx]-
>Caption==LettersList[indx2+1])LetPanel[indx2]->Color=clGreen;
        }
    }
}

```

```

        for(indx2=1;indx2<=5;indx2++){ // Якщо вгадали лише букву
            if((tword[indx]==SecretWord[indx2])&&(indx!=indx2)){
                WordCheck=false;
                for(indx3=1;indx3<=5;indx3++){

                    if((tword[indx]==SecretWord[indx3])&&(Words[i][indx3]-
>Color==clBtnFace))WordCheck=true;}
                    if(WordCheck){
                        Words[i][indx]->Color=clYellow;
                        for(indx3=0;indx3<=32;indx3++){
                            if(Words[i][indx]-
>Caption==LettersList[indx3+1])LetPanel[indx3]->Color=clYellow;
                                }
                            }
                        }
                    }

                }
                // Всі інші букви робимо сірими
                for(indx=1;indx<=5;indx++){
                    if(Words[i][indx]->Color==clBtnFace){
                        Words[i][indx]->Color=clGray;
                        for(indx2=0;indx2<=32;indx2++){
                            if(Words[i][indx]-
>Caption==LettersList[indx2+1])LetPanel[indx2]->Color=clGray;
                                }
                            }
                        }
                    }

                if(i<6)
                    {i++; j=1;}
                else {
                    ShowMessage("Прикро, але ви не вгадали слово :(");
                    GameForm->Hide();
                    Form1->Show();
                }
            }
        }
//-----
void __fastcall TGameForm::Let3_ButtonClick(TObject *Sender)
{
    if (j<=5) {
        //Words[1][2]->Color=clGreen;
        Words[i][j]->Caption="У";
        j++;
    }
}

```

```

    }
    Let3_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let2_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="I";
    j++;
    }
    Let2_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let4_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="K";
    j++;
    }
    Let4_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let5_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="E";
    j++;
    }
    Let5_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let6_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="H";
    j++;
    }
    Let6_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let7_ButtonClick(TObject *Sender)

```



```

{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="I";
j++;
}
Let7_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let8_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="I";
j++;
}
Let8_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let9_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="III";
j++;
}
Let9_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let10_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="III";
j++;
}
Let10_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let11_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="3";
j++;
}
}

```

```

    }
    Let11_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let12_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="X";
    j++;
    }
    Let12_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let13_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="Φ";
    j++;
    }
    Let13_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let14_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="I";
    j++;
    }
    Let14_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let15_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="İ";
    j++;
    }
    Let15_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let16_ButtonClick(TObject *Sender)

```

```

{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="B";
j++;
}
Let16_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let17_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="A";
j++;
}
Let17_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let18_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="I";
j++;
}
Let18_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let19_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="P";
j++;
}
Let19_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let20_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="O";
j++;
}
}

```

```

    }
    Let20_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let21_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="Л";
    j++;
    }
    Let21_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let22_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="Д";
    j++;
    }
    Let22_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let23_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="Ж";
    j++;
    }
    Let23_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let24_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="Є";
    j++;
    }
    Let24_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let25_ButtonClick(TObject *Sender)

```

```

{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="Я";
j++;
}
Let25_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let26_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="Ч";
j++;
}
Let26_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let27_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="C";
j++;
}
Let27_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let28_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="M";
j++;
}
Let28_Button->Color=clBtnFace;
}
//-----
void __fastcall TGameForm::Let29_ButtonClick(TObject *Sender)
{
if (j<=5) {
//Words[1][2]->Color=clGreen;
Words[i][j]->Caption="И";
j++;
}
}

```

```

    }
    Let29_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let30_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="T";
    j++;
    }
    Let30_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let31_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="B";
    j++;
    }
    Let31_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let32_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="Б";
    j++;
    }
    Let32_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let33_ButtonClick(TObject *Sender)
    {
    if (j<=5) {
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="Ю";
    j++;
    }
    Let33_Button->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::BackspaceButtonClick(TObject *Sender)

```

```

    {
    if (j>1) {
    j--;
    //Words[1][2]->Color=clGreen;
    Words[i][j]->Caption="";

    }
    BackspaceButton->Color=clBtnFace;
    }
    //-----
    void __fastcall TGameForm::Let2_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
    {
    Let2_Button->Color=clGray;
    }
    //-----
    void __fastcall TGameForm::Let3_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
    {
    Let3_Button->Color=clGray;
    }
    //-----
    void __fastcall TGameForm::Let4_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
    {
    Let4_Button->Color=clGray;
    }
    //-----
    void __fastcall TGameForm::Let5_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
    {
    Let5_Button->Color=clGray;
    }
    //-----
    void __fastcall TGameForm::Let6_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
    {
    Let6_Button->Color=clGray;
    }
    //-----

```

```

void __fastcall TGameForm::Let7_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
    Let7_Button->Color=clGray;
}
//-----

```

```

void __fastcall TGameForm::Let8_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
    Let8_Button->Color=clGray;
}
//-----

```

```

void __fastcall TGameForm::Let9_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
    Let9_Button->Color=clGray;
}
//-----

```

```

void __fastcall TGameForm::Let10_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
    Let10_Button->Color=clGray;
}
//-----

```

```

void __fastcall TGameForm::Let11_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
    Let11_Button->Color=clGray;
}
//-----

```

```

void __fastcall TGameForm::Let12_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{

```



```

Let12_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let13_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let13_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let14_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let14_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let15_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let15_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let16_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let16_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let17_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let17_Button->Color=clGray;
}
//-----

```

```

void __fastcall TGameForm::Let18_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let18_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let19_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let19_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let20_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let20_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let21_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let21_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let22_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let22_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let23_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let23_Button->Color=clGray;
}

```

```

}
//-----

void __fastcall TGameForm::Let24_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let24_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let25_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let25_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let26_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let26_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let27_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let27_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let28_ButtonMouseDown(TObject *Sender,
TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
Let28_Button->Color=clGray;
}
//-----

void __fastcall TGameForm::Let29_ButtonMouseDown(TObject *Sender,
TMouseButton Button,

```

```

        TShiftState Shift, int X, int Y)
    {
        Let29_Button->Color=clGray;
    }
    //-----

    void __fastcall TGameForm::Let30_ButtonMouseDown(TObject *Sender,
    TMouseButton Button,
        TShiftState Shift, int X, int Y)
    {
        Let30_Button->Color=clGray;
    }
    //-----

    void __fastcall TGameForm::Let31_ButtonMouseDown(TObject *Sender,
    TMouseButton Button,
        TShiftState Shift, int X, int Y)
    {
        Let31_Button->Color=clGray;
    }
    //-----

    void __fastcall TGameForm::Let32_ButtonMouseDown(TObject *Sender,
    TMouseButton Button,
        TShiftState Shift, int X, int Y)
    {
        Let32_Button->Color=clGray;
    }
    //-----

    void __fastcall TGameForm::Let33_ButtonMouseDown(TObject *Sender,
    TMouseButton Button,
        TShiftState Shift, int X, int Y)
    {
        Let33_Button->Color=clGray;
    }
    //-----

    void __fastcall TGameForm::BackspaceButtonMouseDown(TObject
    *Sender, TMouseButton Button,
        TShiftState Shift, int X, int Y)
    {
        BackspaceButton->Color=clGray;
    }
    //-----

```

```

void __fastcall TGameForm::FormShow(TObject *Sender)
{
    int ci,cj;

    Form1->Hide();
    // Загадуємо слово
    randomize();
    SecretWord=ListWords->Lines->Strings[random(ListWords->Lines-
>Count)];
    // Тимчасова підказка для тестування
    Label1->Caption=SecretWord;

    // Очищуємо форму
    for(ci=1;ci<=6;ci++)
        for(cj=1;cj<=5;cj++){
            Words[ci][cj]->Color=clBtnFace;
            Words[ci][cj]->Caption="";
        }
    for(ci=0;ci<=32;ci++)LetPanel[ci]->Color=clBtnFace;

    i=1;
    j=1;

}
//-----

void __fastcall TGameForm::FormClose(TObject *Sender, TCloseAction
&Action)
{
    Form1->Show();
}
//-----

void __fastcall TGameForm::FormDbClick(TObject *Sender)
{
    if(Label1->Visible)Label1->Hide(); else Label1->Show();
}
//-----

```