

ВСТУП

В даній дипломній роботі ми зосередимось на розробці програмного додатку «Сапер» з використанням технології WPF, що дозволить нам створити візуально привабливий та функціональний інтерфейс для користувача. Метою цього проекту є реалізація класичної гри «Сапер» з усіма її основними елементами, такими як поле з мінами, логіка гри, підрахунок очків та можливість рекордів.

Дослідження в цій роботі охоплює детальний аналіз гри «Сапер» та визначення його основних правил та вимог до розробки програмного додатку. Також буде проведений аналіз сучасних підходів до розробки ігрових додатків, зокрема використання технології WPF та мови програмування C#. Це дозволить нам визначити найкращі практики та підходи до розробки «Сапера» з використанням цих технологій.

Далі у роботі буде розглянуто процес розробки програмного додатку на базі технології WPF, включаючи проектування архітектури додатку, розробку графічного інтерфейсу користувача, реалізацію логіки гри та тестування. Також буде проведений аналіз продуктивності та оптимізація додатку для забезпечення плавної ігрової динаміки та задоволення користувачів. Застосування технології WPF дозволить нам створити привабливий інтерфейс, з використанням векторної графіки, анімації та інших ефектів, що покращать візуальний досвід користувача.

Окрім того, в рамках дослідження буде проаналізовано можливості Microsoft Visual Studio 2018, середовища розробки програмного забезпечення, яке надає широкий спектр інструментів для зручного розробки та налагодження додатків на базі технології WPF. Використання цього середовища значно спростить процес розробки та дозволить ефективно використовувати різноманітні функціональні можливості технології WPF.

Розробка програмного додатку на базі технології Windows Presentation Foundation (WPF) - це захоплюючий процес, який дозволяє створити ігровий додаток заснований на класичній грі «Сапер». Гра «Сапер» відома своєю захоплюючою геймплейною механікою та вимагає від користувача логічного мислення, швидкості прийняття рішень та уміння виявляти міни в мінне поле.

Розробка програмного додатку на базі технології WPF гра «Сапер» є цікавим і актуальним завданням. Цей проект дозволить нам дослідити можливості технології WPF та мови програмування C# в контексті створення ігрових додатків. Результатом роботи буде функціональний та привабливий додаток, який задовольнить потреби користувачів та надасть їм можливість насолоджуватись класичною грою «Сапер».

Предметом дослідження є процес розробки програмного додатку гри «Сапер» з використанням технології WPF. Дослідження спрямоване на вивчення та впровадження різних аспектів гри, включаючи логіку гри, візуальний дизайн, функціональні можливості та взаємодію з користувачем. Окрім того, у рамках дослідження будуть розглянуті питання оптимізації та покращення продуктивності додатку.

Цей дослідницький проект має на меті розробити функціональний та привабливий додаток гри «Сапер», використовуючи переваги технології WPF.

Результатом цього дослідження буде функціональний та ігровий додаток гри «Сапер», який буде задовольняти потреби користувачів та забезпечувати їм захоплюючий геймплей та графічну привабливість. Дослідження дозволить також встановити ефективні підходи до розробки ігрових додатків з використанням технології WPF та сприяти подальшій розвитку цього напрямку програмування.

РОЗДІЛ 1

ТЕОРЕТИЧНИЙ ОГЛЯД СТВОРЕННЯ КОМП'ЮТЕРНОЇ ГРИ «САПЕР»

1.1. Аналітичний огляд комп'ютерних ігор

Аналітичний огляд комп'ютерних ігор є процесом оцінки, аналізу та опису характеристик ігор з метою надання корисної інформації гравцям. Огляди ігор можуть містити докладні описи геймплею, графіки, звуку, історії, механік гри та інших аспектів, які впливають на враження від гри.

Аналітичний огляд зазвичай починається з визначення жанру гри, наприклад, пригодницька гра, шутер від першої особи, стратегічна гра або рольова гра. Потім огляд може описувати головні особливості геймплею, такі як система керування, механіки, рівні складності, можливості розвитку персонажів і т.д.

Крім того, аналітичні огляди можуть включати оцінку технічних аспектів гри, таких як якість графіки, звуку, фізичний движок, оптимізація продукту для різних платформ та системних вимог. Також можуть бути зазначені плюси і мінуси гри, оцінка інноваційності, оригінальності та загальної якості гри.

У аналітичному огляді можуть бути надані рекомендації щодо цільової аудиторії гри, наприклад, чи вона підходить для любителів певного жанру, новачків у геймінгу або досвідчених гравців. Також можуть бути вказані схожі ігри, які можуть зацікавити гравців з подібними смаками.

Аналітичні огляди комп'ютерних ігор часто пишуться професійними гейм-журналістами, блогерами та іншими експертами у галузі геймінгу. Їхні огляди можуть бути корисно і як для виробників ігор, які можуть використати ці огляди для поліпшення своїх продуктів, так і для гравців, які шукають об'єктивну інформацію перед придбанням або грати в певну гру.

Аналітичні огляди комп'ютерних ігор можуть мати різну структуру, але зазвичай вони включають наступні елементи:

1. Вступ: Вводиться гра і розповідається про загальні враження або контекст, в якому вона була створена.
2. Геймплей: Описується, як гра працює, які механіки використовуються, які особливості є у геймплеї та які взаємодії можна очікувати.
3. Графіка і звук: Оцінюється якість графіки, дизайну рівнів, анімації та звукового супроводу. Можуть наводитись приклади конкретних моментів гри.
4. Історія та персонажі: Розповідається про сюжет гри, характеристики головних персонажів і їхні мотивації. Оцінюється наскільки захоплюючими або цікавими вони є.
5. Мультиплеєр (за наявності): Якщо гра має мультиплеєрний режим, то розглядається, як він працює, наскільки забезпечена гра на різних платформах, які можливості взаємодії з іншими гравцями.
6. Висновок: Підсумовуються основні враження від гри, можливо, надаються рекомендації для певної аудиторії.

Останнім часом спостерігається значний ріст популярності ігрового програмного забезпечення, тоді як продажі інших типів програмного забезпечення не виявляють такого ж високого темпу зростання. Зокрема, ігри для платформи Windows становлять найбільшу частку продажів в цій галузі. Варто відзначити, що ігрове програмне забезпечення займає третє місце за обсягом доходів у сфері програмного забезпечення, випереджаючи навіть такі продукти, як текстові процесори та електронні таблиці.

Точну кількість комп'ютерних ігор важко визначити, оскільки їх асортимент постійно змінюється і розширюється. Але можна приблизно оцінити, що на ринку завжди присутні тисячі різних найменувань ігор,

пропонуючи широкий спектр жанрів, геймплею та цікавих концепцій для гравців різних вікових груп.

Цей ріст популярності ігрового програмного забезпечення свідчить про значну важливість розвитку індустрії грального софту. Розробка якісних та захоплюючих ігор для платформи Windows є перспективним напрямком для розробників програмного забезпечення, оскільки вони мають можливість залучити широку аудиторію гравців та отримати значні прибутки від продажу своїх продуктів.

Одним з видів класифікації комп'ютерних ігор є їх поділ на пригодницькі ігри. Пригодницькі ігри пропонують захоплюючу історію, в якій гравець може контролювати головного персонажа або навіть декілька персонажів. Вони створюють мультимедійну розповідь, де деталі оповіді можуть бути візуально привабливими та деталізованими. Пригодницькі ігри можуть охоплювати широкий спектр жанрів, від трагічних історій до науково-фантастичних пригод, фантазійних світів і навіть розповідей, що містять елементи обману і підступу. Основою таких ігор є розв'язання різних головоломок і подолання перешкод, які зустрічаються на шляху гравця, по мірі розвитку сюжету.

Ось кілька категорій комп'ютерних ігор:

1. Фантастичні рольові ігри: це комп'ютеризована версія розповідей Dungeons and Dragons, де гравець керує групою персонажів у світі, що може бути середньовічним або сучасним, відбуватися у космосі або майбутньому.
2. Імітатори повітряних і космічних битв: ігри, які точно імітують польоти та битви на реальних літаках або в космосі.
3. Імітатори наземних і морських битв: симулятори військового устаткування, де гравець може керувати підводними лодками, кораблями або танками і випробувати реальні сценарії битв.

4. Ігри з літаками, поїздами і автомобілями: ці ігри дозволяють гравцям керувати транспортними засобами, від екстремальних перегонів до реалістичного польоту літаком.
5. Спортивні імітатори: ігри, де можна керувати спортивними командами або професіоналами, виконувати стратегії та змагатися у віртуальних спортивних подіях.
6. Аркадні ігри/ігри дій: швидкі ігри, де рефлекс та швидкість рухів гравця важливі для досягнення успіху.
7. Настільні ігри, солітер і салонові ігри: комп'ютерні версії традиційних настільних ігор, які можна грати самотійно або з комп'ютерними противниками.

1.2. Опис предметної області

Предметна область даної дипломної роботи – гра «Сапер». Розробка програмного додатку на базі технології WPF (Windows Presentation Foundation) гра «Сапер» відноситься до галузі комп'ютерних ігор та програмування. «Сапер» - це класична гра, яка широко відома серед користувачів операційних систем Windows.

При розробці програмного додатку «Сапер» на базі технології WPF використовується спеціалізована платформа для створення графічних інтерфейсів Windows. WPF надає розробникам можливість створювати багатофункціональні та привабливі візуальні елементи і дозволяє ефективно взаємодіяти з користувачем.

Гра «Сапер» має на меті забезпечити гравцю веселі та захоплюючі години гри, де гравець має виявити приховані міни на ігровому полі. Ігрове поле складається з сітки, розмір якої може бути налаштований, але зазвичай вона має прямокутну форму. На початку гри, деякі клітини на полі містять міни, а інші - пусті. Гравець повинен розгадати місця розташування мін та

позначити їх прапорцями. Мета гри полягає в тому, щоб виявити всі безпечні клітини без попадання на міну. Якщо гравець виявляє міну, гра закінчується.

У програмному додатку «Сапер» на базі технології WPF реалізується відображення графічного інтерфейсу гри, включаючи ігрове поле, кнопки для взаємодії з грою (розташування прапорців, відкриття клітин), показник часу та кількості виявлених мін, а також повідомлення про результати гри.

Гра «Сапер» - це популярна логічна гра, яка широко відома серед користувачів операційних систем Windows. Головна мета гри полягає в тому, щоб виявити всі безпечні клітини на ігровому полі без попадання на міни.

Правила гри «Сапер» наступні:

1. Гравець має справу з ігровим полем, яке складається з сітки клітин. Розмір сітки може бути налаштований, але зазвичай вона має прямокутну форму.
2. Деякі клітини на полі містять міни, а інші - пусті. Міни розташовуються на полі випадковим чином перед кожною грою.
3. Гравець може взаємодіяти з клітинами, натискаючи на них:
 - Якщо гравець відкриває клітину і виявляє міну, гра закінчується поразкою.
 - Якщо гравець відкриває клітину і вона є пустою, то вона відображає число, яке показує, скільки мін межують з цією клітиною (від 0 до 8). Це число вказує на кількість мін у сусідніх клітинах. За допомогою цих чисел гравець може визначити, де розташовані міни.
 - Якщо гравець виявляє клітину з числом 0, це означає, що жодна з сусідніх клітин не містить міни. У такому випадку гравець автоматично відкриває всі сусідні клітини, розкриваючи велику область пустоти.

4. Гравець також може позначити клітини, які, на його думку, містять міни, розмістивши на них прапорець. Це допомагає гравцеві позначити потенційно небезпечні клітини.
5. Гра продовжується до тих пір, поки гравець не відкриє всі безпечні клітини на ігровому полі. Якщо гравець відкриває всі безпечні клітини, він перемагає.
6. Гра може мати різні рівні складності, що впливають на розмір ігрового поля та кількість мін. Зазвичай є декілька варіантів рівнів складності, таких як початковий, середній і високий.
7. На початку гри гравець може мати деякі підказки, які допомагають йому зрозуміти, де розташовані міни. Наприклад, перша клітина, яку гравець відкриває, завжди є безпечною, що дає стартову інформацію для подальшої гри.
8. У деяких версіях гри "Сапер" може бути збільшена складність, де окрім звичайних мін, є також "кроватьі мін". Кроватьі мін - це групи мін, які розташовуються поруч одна з одною і показують, скільки мін знаходиться в групі. Це додає додатковий елемент стратегії та розгляду гравцем можливих варіантів розташування мін.

Гра «Сапер» є захопливою та вимагає від гравця логічного мислення та уваги. Вона надає можливість розвивати навички аналізу та прийняття рішень.

1.3. Критичний аналіз програм комп'ютерної гри «Сапер»

Комп'ютерна гра «Сапер» є класичною грою, яка має свою популярність серед користувачів операційних систем Windows. Однак, вона також має деякі недоліки і обмеження. Нижче наведений критичний аналіз комп'ютерної гри «Сапер»:

1. *Елемент випадковості:* Одним з недоліків гри «Сапер» є наявність елемента випадковості. Розташування мін на ігровому полі залежить від випадкових факторів. Це може призводити до ситуацій, коли на початку гри гравець відкриває клітину з міною, що призводить до неможливості подальшої гри.
2. *Обмежений геймплей:* Гра «Сапер» має обмежений геймплей і складається з обмеженого набору правил. Після деякого часу гра може стати повторюваною і втратити інтерес для деяких гравців. Відсутність різноманітності у геймплеї може призвести до втрати зацікавленості з боку користувачів.
3. *Відсутність багатощаровості:* Гра «Сапер» не має багатощарового режиму або можливості грати з іншими гравцями в режимі онлайн. Це обмежує можливості взаємодії з іншими користувачами та можливість вирішувати головоломки разом.
4. *Відсутність розвитку сюжету:* У грі «Сапер» відсутній розвиток сюжету або історії. Гравець просто відкриває клітини і намагається уникнути мін. Це може призвести до швидкої втрати зацікавленості користувачів, які шукають глибоких і захоплюючих ігрових досвідів.

Існує кілька реальних програмних рішень для гри Сапер:

- *Microsoft Minesweeper:* Це офіційна версія гри Сапер, розроблена компанією Microsoft для операційних систем Windows. Вона має графічний інтерфейс, різні режими гри та рівні складності. Переваги: Офіційна версія гри Сапер від Microsoft, з високою якістю і надійністю. Має добре розроблений графічний інтерфейс та різні режими гри. Недоліки: Обмежена доступність лише для операційних систем Windows.



Рис. 1.1 Приклад інтерфейсу програми Microsoft Minesweeper

- *Minesweeper X*: Це безкоштовне програмне забезпечення гри Сапер, доступне для Windows, macOS та Linux. Minesweeper X має розширені функції, налаштування полів гри та можливість змагатися з іншими гравцями в режимі онлайн. Переваги: Безкоштовне програмне забезпечення, доступне для Windows, macOS та Linux. Має розширені функції та налаштування полів гри, а також можливість змагатися з іншими гравцями в режимі онлайн. Недоліки: Можуть виникати проблеми з сумісністю на деяких платформах, оскільки це незалежне програмне забезпечення.

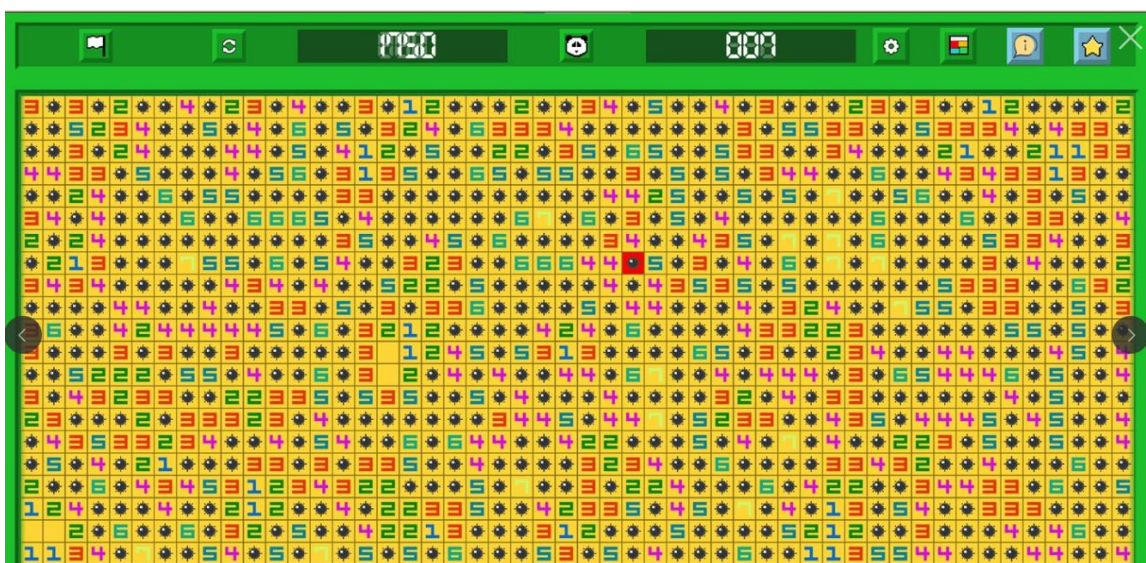


Рис. 1.2 Приклад інтерфейсу гри Minesweeper X

- *Minesweeper Classic*: Це мобільна гра Сапер для платформ Android та iOS. Вона пропонує класичний геймплей Сапера з різними розмірами поля, варіантами складності та підтримкою рекордів гравців. Переваги: Мобільна версія гри Сапер для Android та iOS, доступна безкоштовно. Має класичний геймплей з різними розмірами поля та рівнями складності. Недоліки: Можливість відображення реклами та обмеження функцій у безкоштовній версії.

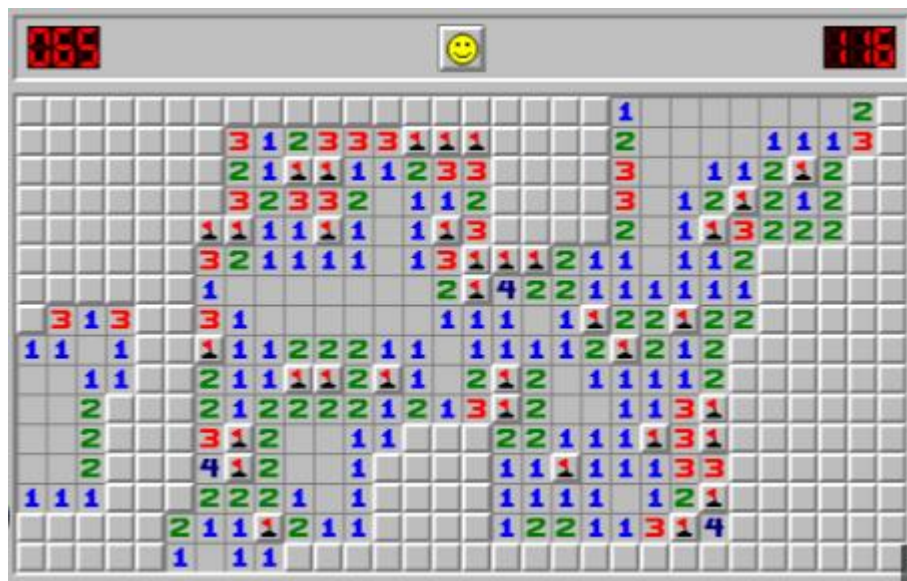


Рис. 1.3 Приклад інтерфейсу гри Minesweeper Classic

- *Minesweeper Genius*: Це інноваційна версія гри Сапер для мобільних пристроїв Android та iOS. Вона поєднує елементи гри Сапер з головоломками та логічними завданнями, створюючи унікальний геймплейний досвід. Переваги: Інноваційна версія гри Сапер, що поєднує головоломки та логічні завдання. Має унікальний геймплей та варіативність викликів. Недоліки: Деякі гравці можуть відчувати, що вона віддаляється від класичної гри Сапер.

Представлені в роботі програмні рішення для гри Сапер добре відомі і використовуються гравцями по всьому світу. Вони демонструють різні платформи та можливості для насолоди грою «Сапер».



Рис. 1.4 Приклад інтерфейсу програми Minesweeper Genius

- *Minesweeper Online*: Це веб-додаток гри Сапер, який дозволяє грати в Сапер в онлайн-режимі. Він надає можливість змагатися з іншими гравцями по всьому світу, переглядати статистику та встановлювати рекорди. Переваги: Веб-додаток, що дозволяє грати в Сапер в онлайн-режимі. Можливість змагатися з гравцями з усього світу та переглядати статистику. Недоліки: Залежить від наявності Інтернет-з'єднання.

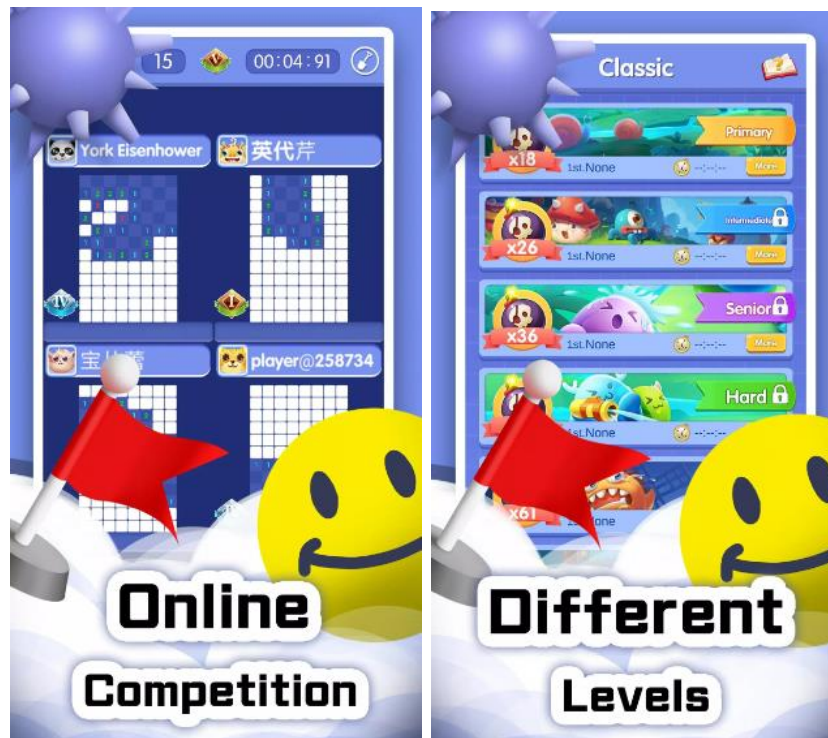


Рис. 1.5 Приклад інтерфейсу програм Minesweeper Online

- Створюємо нову групу з перетинаючихся клітинок (23, ?).
- Розраховуємо кількість мін у новій групі, що дорівнює кількості мін у групі з більшою кількістю мін (234, 2) мінус кількість залишених клітинок в іншій групі після відокремлення перетинаючихся клітинок (1, ?). Тобто $2-1 = 1$. Отримуємо (23, 1).
- Якщо розрахована кількість мін у новій групі (23, 1) не дорівнює кількості мін у групі з меншою кількістю мін (123, 1), то припиняємо перетворення.
- Віднімаємо новоутворену групу від обох перетинаючихся груп.
 $(123, 1) - (23, 1) = (1, 0)$, $(234, 2) - (23, 1) = (4, 1)$.
- Додаємо новоутворену групу до списку груп. Отже, (234, 2) і (123, 1) $\rightarrow$ (1, 0) і (23, 1) і (4, 1).

5. Повторюємо кроки з 1 до тих пір, поки не відбудеться жодних змін:

У підсумку у нас буде три типи груп:

- Кількість мін дорівнює нулю.
- Кількість мін дорівнює кількості клітинок у групі.
- Кількість мін більша за кількість клітинок у групі.

Отже, клітинки з першої групи можна відкрити безпечно, а клітинки з другої групи можна позначити як міні. В цьому і полягає суть основного алгоритму.

Часто зустрічаються ситуації, коли немає достовірного рішення. Тоді на допомогу приходить наступний алгоритм, який використовує теорію ймовірностей. Алгоритм працює у два етапи:

1. Визначення ймовірності в окремих клітинках з урахуванням впливу кількох відкритих клітинок.
2. Коригування ймовірностей з урахуванням взаємного впливу груп з спільними клітинками одна на одну.

Розглянемо, як цей метод працює на прикладі ситуації, коли відкриті всього дві сусідні клітинки зі значеннями 4 і 2. Ймовірності наявності мін у клітинках 4 і 2 окремо дорівнюють $4/7 = 0,57$ та $2/7 = 0,28$ відповідно.

	0,57	0,57	0,57	
	0,57	4	0,57	
	0,28		0,28	
	0,57	2	0,57	
	0,28		0,28	
	0,28	0,28	0,28	

Рис. 2.2 Ситуації, коли відкриті всього дві сусідні клітинки

Для обчислення ймовірності знаходження міни в клітинці поруч з декількома відкритими клітинками використовується формула для обчислення ймовірності принаймні одного події: Ймовірність настання події А, яка полягає в появі принаймні однієї з подій $A_1, A_2, \dots, A_n$, незалежних в сукупності, дорівнює різниці між одиницею і добутком ймовірностей протилежних подій. $A = 1 - (1 - A_1) * (1 - A_2) * \dots * (1 - A_n)$.

У випадку суміжних клітинок після застосування цієї формули результат дорівнює $1 - (1 - 0,57) * (1 - 0,28) = 0,69$.

	0,57	0,57	0,57	
	0,69	4	0,69	
	0,69	2	0,69	
	0,28	0,28	0,28	

Рис. 2.3 Розрахунок суміжних клітинок

Проте слід пам'ятати, що сума ймовірностей у кожній групі клітинок повинна дорівнювати кількості мін у групі. Тому всі значення в кожній групі

потрібно помножити так, щоб в результаті їх сума дорівнювала числу мін. Це число рівне кількості мін у групі, поділене на поточну суму ймовірностей клітинок групи:

$$4 / (0,57 + 0,57 + 0,57 + 0,69 + 0,69 + 0,69 + 0,69) = 0,895 \quad 0,57 * 0,895 = 0,485, \quad 0,69 * 0,895 = 0,618$$

Тепер ті клітинки, які мали ймовірність 0,57, мають 0,485, а ті, які мали 0,69, тепер мають 0,618. Аналогічний розрахунок для другої групи виконуємо враховуючи коригування від попередньої групи.

$$2 / (0,28 + 0,28 + 0,28 + 0,618 + 0,618 + 0,618 + 0,618) = 0,604 \quad 0,28 * 0,604 = 0,169, \quad 0,618 * 0,604 = 0,373.$$

Бачимо, що ймовірність у загальних клітинках знову змінилась, тому подібне вирівнювання для кожної групи потрібно повторити кілька разів, доки система не досягне деяких стійких значень, які і будуть справжніми ймовірностями знаходження мін у клітинках.

$$4 / (0,485 + 0,485 + 0,485 + 0,373 + 0,373 + 0,373 + 0,373) = 1,357 \quad 0,485 * 1,357 = 0,658, \quad 0,373 * 1,357 = 0,506 \quad 2 / (0,169 + 0,169 + 0,169 + 0,506 + 0,506 + 0,506 + 0,506) = 0,79 \quad 0,169 * 0,79 = 0,134, \quad 0,506 * 0,79 = 0,4$$

	0,79	0,79	0,79	
	0,41	4	0,41	
	0,41	2	0,41	
	0,12	0,12	0,12	

Рис. 2.4 Ймовірність у загальних клітинках

Далі необхідно вибрати одну з клітинок з мінімальною ймовірністю та зробити хід в цю клітинку.

На заключному етапі гри велику роль відіграє кількість не позначених мін. Знаючи цю кількість, можна шляхом перебору підставляти їх у невідомі

комірки та позначати відповідні варіанти. У процесі перебору підходящих варіантів для кожної комірки підраховуємо кількість позначок. Розділивши отримані значення на загальну кількість позначок, отримуємо ймовірність знаходження мін для кожної комірки. Наприклад, у цій ситуації, яка здається має лише одне достовірне рішення, останній метод (LastTurns) знайшов 3 комірки з ймовірністю 0% знаходження міни.

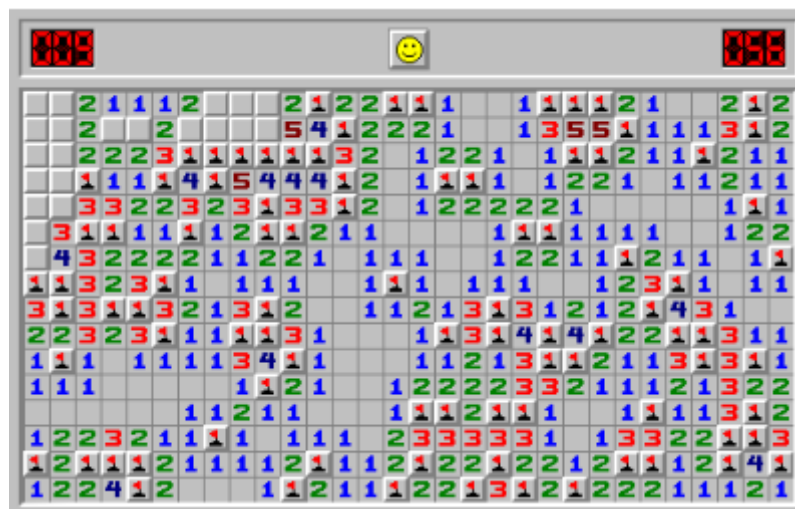


Рис. 2.5 Заключний етап гри «Сапер»

LastTurns(9,21) перевірів 144 відповідні комбінації з 293930 можливих і знайшов 3 комірки з мінімальною ймовірністю 0%.

З точки зору зрозумілості ідеї, цей метод є найпростішим. На практиці, за наявності достатньої кількості вибірок, розрахункові та фактичні значення ймовірностей знаходження міни в комірни майже збігаються.

2.2. Розробка функціональної схеми

Автоматизація інформаційних потоків роботи гри «Сапер» може бути здійснена за допомогою використання програмного коду із наступними функціями:

1. Генерація поля гри: Автоматичне створення поля гри з випадковим розташуванням мін та обчисленням числа сусідніх мін для кожної комірки.

2. Відображення поля гри: Автоматичне відображення поля гри користувачу у текстовому або графічному вигляді.
3. Взаємодія з користувачем: Автоматизована обробка введення користувача, яка включає обробку кліків на комірки та позначення комірок як мін.
4. Перевірка стану гри: Автоматична перевірка стану гри після кожної дії користувача, щоб визначити, чи гра завершена перемогою або програшем.
5. Інформаційні повідомлення: Автоматичне відображення повідомлень користувачу про результати їх дій, такі як програш, перемога або кількість сусідніх мін у відкритій комірці.
6. Збереження та завантаження стану гри: Можливість автоматичного збереження поточного стану гри та завантаження гри з попередньо збереженого стану.
7. Статистика гри: Збір та відображення статистичних даних про гру, таких як кількість перемог, програшів, середній час гри тощо.

Ці функції можуть бути реалізовані за допомогою програмного коду, включаючи структури даних для зберігання інформації про поле гри та комірки, функції для обробки введення користувача, функції перевірки стану гри, а також логіку для відображення повідомлень та статистики.

Функціональна діаграма представлена на рисунку 2.6 відображає взаємодію між функціями гри "Сапер". Головний пакет "Гра" містить функції для розпочатку гри, відкриття комірки, позначення комірки як міни та перевірки стану гри. Пакет "Поле" включає функції для отримання комірки та розгортання сусідніх комірок. Пакет "Комірка" містить функції для отримання стану комірки, отримання числа сусідніх мін

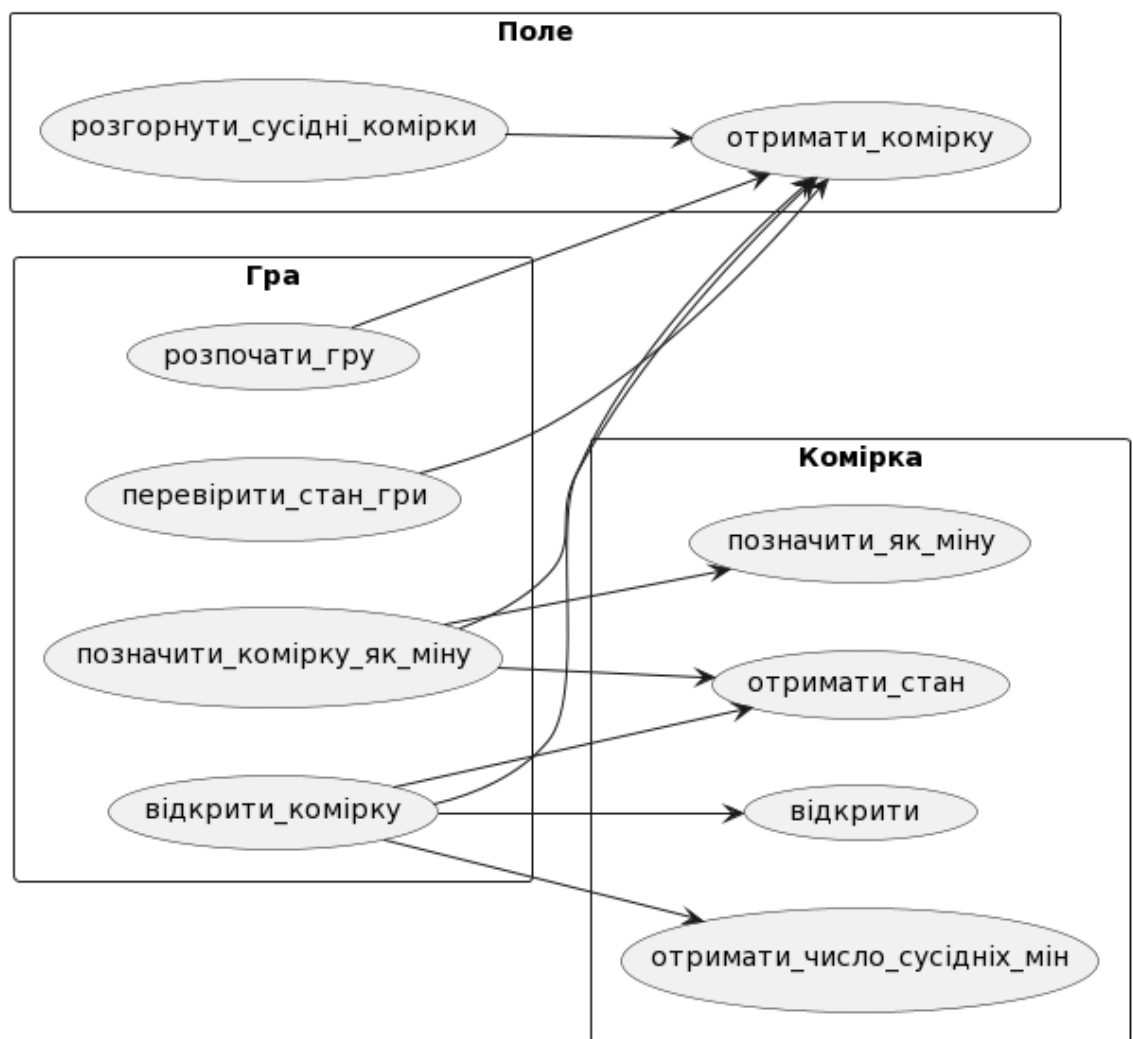


Рис. 2.6 Функціональна діаграма гри «Сапер»

Контекстна діаграма демонструє взаємодію користувача з грою "Сапер". Користувач взаємодіє з грою шляхом відображення поля гри, взаємодії з комірками (відкриття комірки або позначення її як міни) та перевірки стану гри (перевірка на перемогу або програш). Гра реагує на дії користувача та виконує відповідні дії відповідно до правил гри "Сапер".

Ця діаграма допоможе вам уявити основні компоненти та взаємодію в грі "Сапер" на високому рівні.

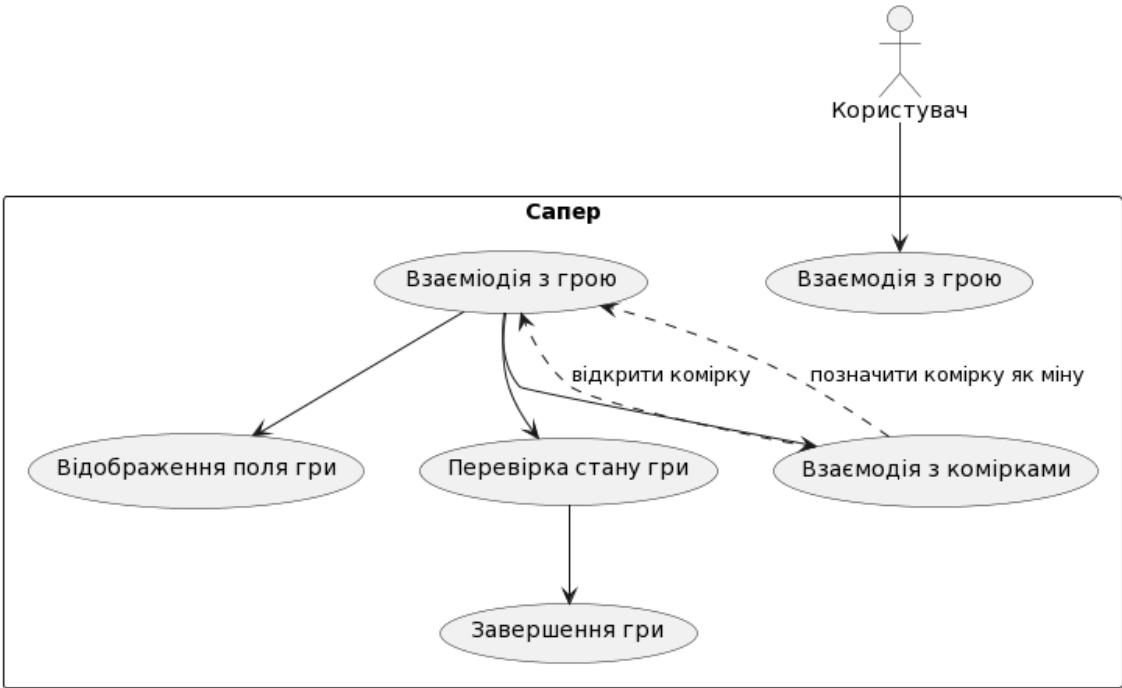


Рис. 2.7 Контекстна діаграма основних компонентів гри «Сапер»

У контекстній діаграмі, що представлена на рисунку 2.7, зображений зовнішній агент, яким є користувач гри "Сапер", та основні компоненти гри: "Гра", "Поле" і "Комірка". Взаємодія між користувачем та грою відбувається через функції гри, яка керує полем та комірками. Поле взаємодіє з комірками, отримуючи запити щодо стану та дій над ними.

Ця контекстна діаграма допомагає зрозуміти зовнішній контекст гри "Сапер" та взаємозв'язки між її основними компонентами та користувачем.

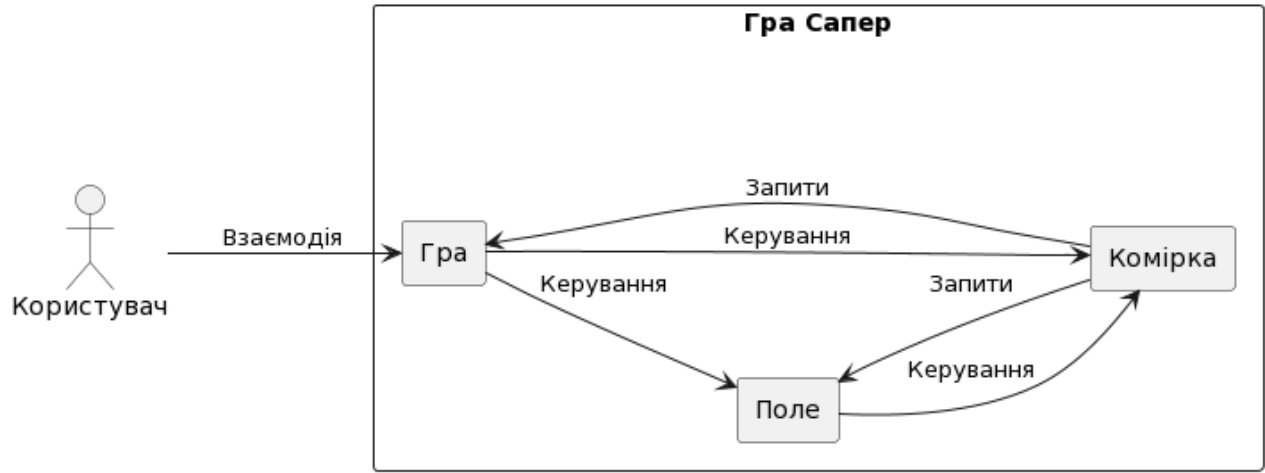


Рис. 2.2 Контекстна діаграма зовнішнього контексту гри

У UML діаграмі класів, що представлено на рисунку 2.8, є основні класи, що використовуються в грі "Сапер". Клас Гра має залежність до класу Поле, яке містить масив об'єктів Комірка. Клас Комірка має атрибути для збереження стану комірки та числа сусідніх мін. Також використовується перерахування СтанКомірки, яке визначає можливі стани комірки.

Ця UML діаграма дозволяє візуалізувати взаємозв'язки між класами та структуру гри "Сапер". Вона може служити основою для подальшої реалізації гри, а також для розуміння взаємодії між компонентами гри.

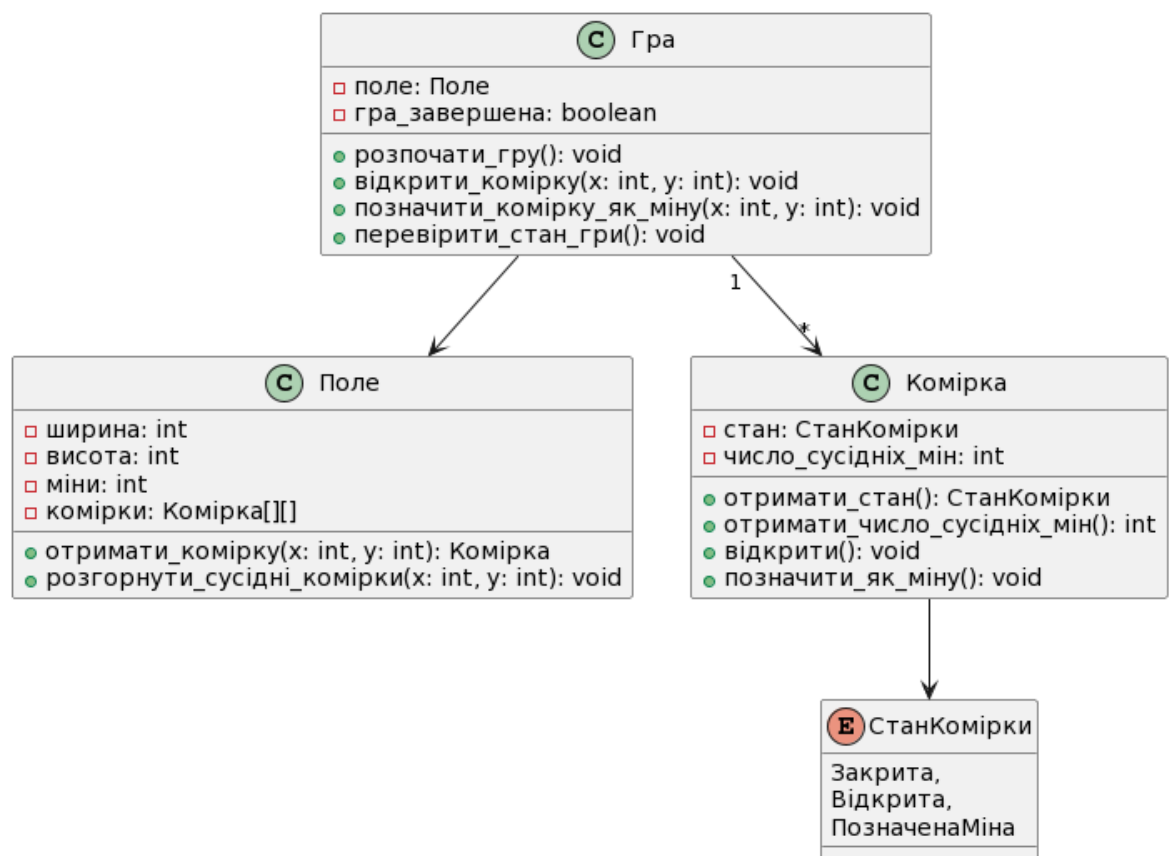


Рис. 2.8 UML діаграма взаємозв'язків між класами та структурою гри

2.3. Обґрунтування програмних засобів розробки

Для створення системи гри «Сапер» була обрана мова програмування C# та середовище розробки Microsoft Visual Studio 2019 з використанням технології Windows Presentation Foundation (WPF).

Обґрунтування вибору мови програмування C# та середовища розробки Microsoft Visual Studio 2019 з використанням технології Windows Presentation Foundation (WPF) для розробки гри "Сапер" базується на кількох ключових факторах:

Популярність та підтримка: Мова програмування C# є однією з найпопулярніших мов у сфері розробки програмного забезпечення, особливо для платформи .NET. Її широке використання гарантує наявність багато ресурсів, документації, форумів та спільноти розробників, що полегшує процес розробки, пошук рішень та розв'язання можливих проблем.

Середовище розробки Microsoft Visual Studio 2019: Visual Studio є потужним та комплексним середовищем розробки з багатьма інструментами та функціями, які сприяють зручності та продуктивності розробки. Вона надає широкі можливості для розробки, налагодження, тестування та розгортання програмного забезпечення. Крім того, Microsoft постійно оновлює та підтримує Visual Studio, забезпечуючи нові можливості та виправлення помилок.

Windows Presentation Foundation (WPF): WPF є потужною технологією для створення графічних інтерфейсів користувача (GUI) в Windows-подібних додатках. Вона надає багато функцій і можливостей для створення привабливих, динамічних інтерфейсів з використанням векторної графіки, анімації, стилізації та інших ефектів. WPF сприяє розробці сучасних та естетичних інтерфейсів, що можуть забезпечити кращий користувацький досвід.

Сумісність та розширення: Використання C# та технології WPF у поєднанні з Microsoft Visual Studio 2019 дозволяють створити гнучку та розширювану архітектуру для гри "Сапер". C# забезпечує потужну об'єктно-орієнтовану парадигму програмування, що сприяє організації коду, полегшує управління станом гри та взаємодію з користувачем.

Технологія WPF надає широкі можливості для створення графічного інтерфейсу гри, дозволяючи використовувати різноманітні елементи управління, налаштовувати їх зовнішній вигляд та анімацію, а також реагувати на події користувача. WPF також забезпечує розділення логіки та представлення, що полегшує розробку та підтримку коду.

Застосування Microsoft Visual Studio 2019 дозволяє ефективно використовувати можливості мови C# та технології WPF. Інтегроване середовище розробки надає розширення, інструменти для налагодження коду, підтримку контролю версій та інші корисні функції, що сприяють зручності та продуктивності розробки.

Отже, обрання мови програмування C# та середовища розробки Microsoft Visual Studio 2019 з використанням технології WPF для розробки гри "Сапер" є обґрунтованим рішенням, яке дозволить створити функціональний, зручний у використанні та привабливий графічний інтерфейс гри.

Windows Presentation Foundation (WPF) є графічною підсистемою в складі .NET Framework, призначеною для розробки візуально привабливих клієнтських додатків для Windows. WPF базується на мові XAML і надає розширені можливості взаємодії з користувачем. Вона встановлена в різних версіях операційних систем Windows, починаючи з Windows Vista.

Основні особливості WPF полягають у використанні векторної системи візуалізації, яка незалежна від дозволу пристрою виведення, і побудована з урахуванням сучасного графічного обладнання. WPF надає засоби для створення візуального інтерфейсу, такі як мова XAML, елементи управління, прив'язка даних, макети, дво- та тривимірна графіка, анімація, стилі, шаблони, документи, текст, мультимедіа і оформлення.

З використанням C# та WPF в Microsoft Visual Studio 2017 можна реалізувати функціональність гри «Сапер» і створити привабливий інтерфейс взаємодії з користувачем.

WPF (Windows Presentation Foundation) використовує графічну технологію DirectX, на відміну від Windows Forms, який використовує GDI/GDI+. Завдяки використанню апаратного прискорення через DirectX, WPF досягає вищої продуктивності порівняно з GDI+. Крім того, існує урізана версія CLR, відома як WPF/E або Silverlight.

Для розробки WPF додатків можна використовувати будь-яку мову, яка є сумісною з платформою .NET. Серед таких мов: C#, F#, VB.NET, C++, Ruby, Python, Delphi (з використанням Prism), Lua та багато інших. Для повноцінної роботи з WPF можна використовувати як Visual Studio, так і Expression Blend. Visual Studio більше спрямована на програмування, тоді як Expression Blend надає інструменти для дизайну і дозволяє здійснювати багато операцій без потреби в ручному редагуванні XAML. Приклади таких операцій включають анімацію, стилізацію, управління станами та створення елементів управління.

WPF надає розширений і гнучкий набір графічних можливостей. Основною одиницею виміру є апаратно-незалежний піксель, що дозволяє створювати графіку, яка не залежить від дозволу екрану або пристроїв. Також WPF має вбудовану підтримку графіки та анімації, спрощуючи їх програмування та автоматичне керування анімацією. Графічна система WPF також використовує апаратне прискорення, що дозволяє зменшити навантаження на ЦП, використовуючи можливості графічного обладнання.

WPF (Windows Presentation Foundation) - це фреймворк розробки, який надає різноманітні можливості для створення багатофункціональних десктопних додатків. Основні характеристики WPF включають:

1. Сумісність з мовами: WPF можна використовувати з різними мовами, такими як C#, F#, VB.NET, C++, Ruby, Python, Delphi (Prism), Lua та іншими, якщо вони сумісні з .NET.
2. Засоби розробки: WPF можна розробляти за допомогою Visual Studio, яка орієнтована на програмування, або Expression Blend,

яка спрямована на дизайн і дозволяє візуально редагувати XAML без ручного введення. Expression Blend підтримує функції, такі як анімація, стилізація, створення елементів управління тощо.

3. Графіка: WPF надає масштабовану та гнучку систему графіки. Вона використовує апаратно-незалежні пікселі як основну одиницю виміру, що дозволяє роботі з графікою бути незалежною від роздільної здатності та пристроїв. WPF спрощує програмування графіки шляхом автоматичного керування анімацією та використання апаратного прискорення для зменшення використання ЦП.
4. Веб-подібна модель компоновання: WPF пропонує гнучку систему компоновання, яка динамічно розміщує елементи управління на основі їх вмісту, а не фіксованих координат. Це дозволяє створювати інтерфейси, які адаптуються до відображення динамічного контенту або підтримки різних мов.

При виборі мови програмування важливими критеріями є її зручність, потужність та надійність програм, написаних на цій мові. C# є системною мовою, яка дозволяє створювати надійні та стійкі до помилок компоненти, які можуть бути повторно використані в різних ситуаціях. Вона також сприяє кращій структуризації проекту та повному використанню можливостей платформи .NET.

Середовище розробки Visual Studio 2019 надає зручні можливості для редагування та налагодження програм. Воно використовує технологію IntelliSense, яка підказує, як завершити код під час його набору, містить заготовки часто використовуваних фрагментів коду та дозволяє проводити рефакторинг (переструктуризацію коду, інтелектуальні виправлення і т.д.). Під час набору тексту програми швидко аналізується код і попереджується про можливі помилки.

Visual Studio 2019 також надає можливість налаштування комбінацій клавіш для команд, додавання та видалення кнопок на панелях інструментів. Це дозволяє швидко виконувати різні дії, такі як коментування/розкоментування фрагментів коду, пошук, перестановка рядків програми та інше. Робота з вікнами також організована зручно, дозволяючи їх розміщення у різних частинах екрану, прикріплення до боків головного вікна, закріплення або автоматичне приховування.

У Visual Studio 2019 також зручно відлагоджувати програми. Процес відлагодження відбувається по кроках інтелектуально, зупиняючись на кожному операторі, якщо на одному рядку

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1. Розробка алгоритмів

Розробка блок-схеми програми гри «Сапер» представлена на рисунку

3.1.

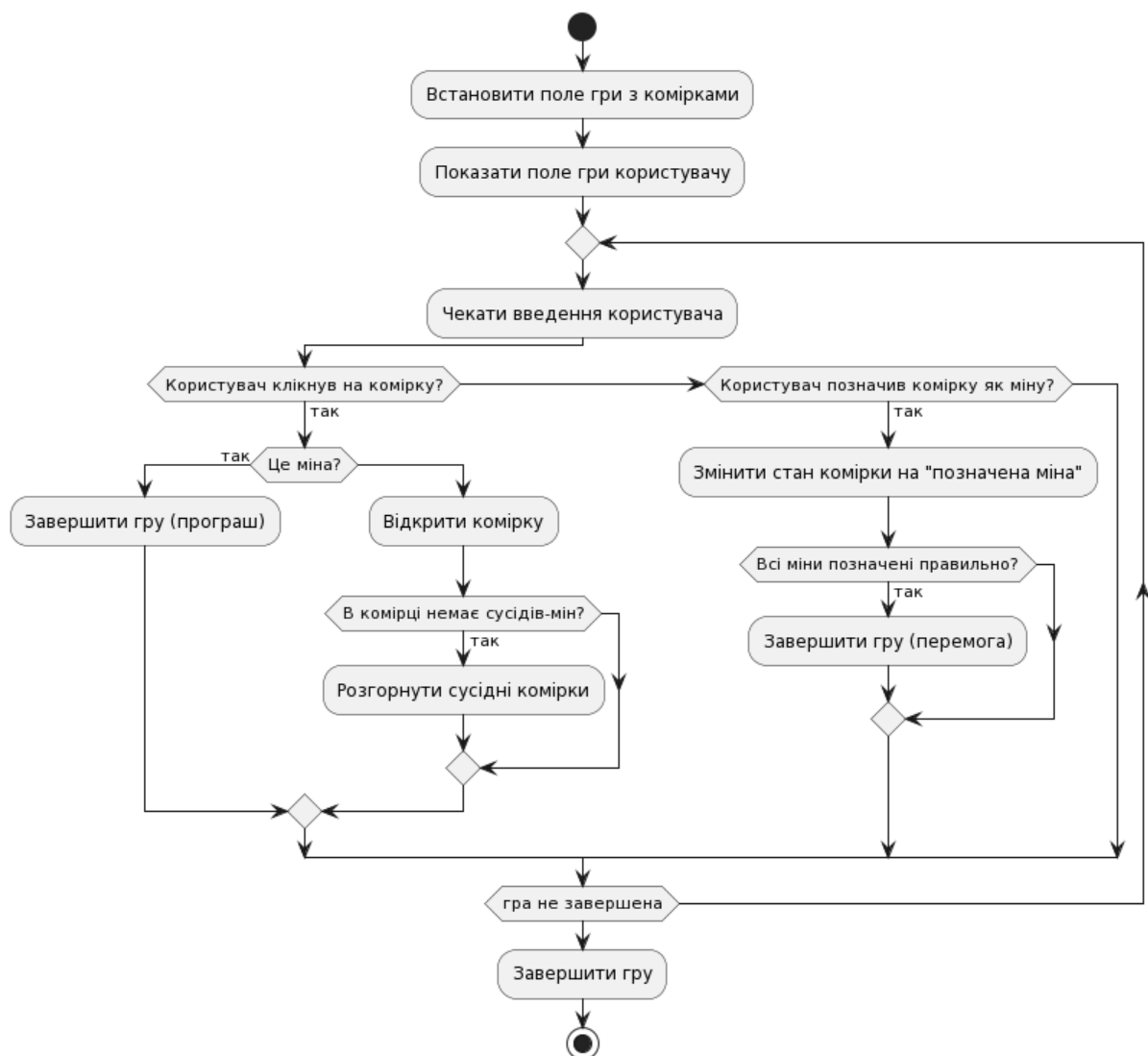


Рис. 3.1 Блок-схема алгоритму

Рисунок 3.1 відображає функціональну схему гри «Сапер». Він включає початок гри, взаємодію з користувачем, перевірку стану комірок та завершення гри у разі перемоги або програшу.

3.2. Вимоги до технічних засобів

Технічні вимоги до комп'ютерної гри «Сапер» можуть варіюватися в залежності від конкретної реалізації гри та її комплексності. Однак, основні загальні вимоги до системи можуть включати:

Операційна система: Гра може вимагати певної операційної системи, такої як Windows, macOS, Linux і т. д. Важливо перевірити, яку платформу підтримує гра перед її встановленням.

Процесор: Мінімальні вимоги до процесора можуть бути досить низькими, наприклад, Intel Core 2 Duo або еквівалентний.

Оперативна пам'ять: Мінімальний обсяг оперативної пам'яті може становити 2 ГБ, але для кращої продуктивності може знадобитися більше.

- **Відеокарта:** Якщо гра має графічні ефекти або вимагає 3D-прискорення, може бути необхідна сумісна відеокарта.
- **Місце на диску:** Гра може потребувати певного обсягу місця на жорсткому диску для встановлення.
- **Звукова карта:** Якщо гра має звукові ефекти або музику, може знадобитися сумісна звукова карта.
- **Драйвери:** Важливо мати оновлені драйвери для графічної та звукової карти, щоб гарантувати сумісність та належну роботу гри.

Автоматизована система вимагає встановлення додаткового програмного пакету Visual Studio 2019. Для успішної установки Visual Studio 2019 потрібні такі системні вимоги:

- Вимоги до обладнання:

- Процесор з частотою 1,6 ГГц або більше.
- Оперативна пам'ять (ОЗУ) об'ємом 1 ГБ (1,5 ГБ для роботи на віртуальній машині).
- 10 ГБ вільного місця на жорсткому диску.
- Жорсткий диск з частотою обертання 5 400 об/хв.
- Відеокарта з підтримкою DirectX 9 і дозволом екрану 1024x768 або вище.

Вимоги до операційної системи: У Windows 8.1 і Windows Server 2012 R2 потрібне оновлення 2919355 для успішної установки Visual Studio 2015.

Додаткові вимоги: Для розробки додатків для Магазину Windows і універсальних програм для Windows потрібна Windows 8.1 або більш пізня версія.

Для розробки додатків Windows Phone 8.0 потрібна Windows 8.1 (x64) з оновленням або більш пізня версія.

Для емуляторів Windows потрібна Windows 8.1 Професійна (x64) або вище, а також процесор, який підтримує клієнт Hyper-V і перетворення адрес другого рівня (SLAT).

Visual Studio - це повноцінне середовище розробки, яке надає засоби для створення веб-додатків ASP.NET, XML (веб-служби), настільних і мобільних додатків. У ньому використовуються мови програмування, такі як Visual Basic, Visual C# і Visual C++. Інтегроване середовище розробки (IDE) дозволяє зручно використовувати ці мови та спрощує процес створення рішень, базованих на декількох мовах.

Для встановлення Visual Studio 2019 та його різних версій (Enterprise, Professional, Community) потрібні певні мінімальні системні вимоги. Вимоги можуть варіюватися в залежності від версії Visual Studio та додаткових компонентів, які ви обираєте для установки. Основні системні вимоги для Visual Studio 2019 включають наступне:

Visual Studio 2019 є комплексним середовищем розробки, яке надає широкі можливості для створення різноманітних додатків. Це включає веб-додатки на платформі ASP.NET, розробку XML-веб-служб, настільні програми і мобільні додатки. Усі ці можливості доступні для розробки з використанням мов програмування, таких як Visual Basic, Visual C# і Visual C++.

Окрім цього, Visual Studio 2019 забезпечує інтегроване середовище розробки (IDE), що спрощує процес створення проєктів, що складаються з декількох мов програмування. Інтерфейс IDE забезпечує зручний редактор коду з підсвічуванням синтаксису, автодоповненням, відладчиком та іншими корисними інструментами, які полегшують розробку програмного забезпечення.

Наявність додаткового програмного пакету Visual Studio 2019 дозволяє зробити розробку комп'ютерних ігор, таких як Сапер, більш ефективною і зручною завдяки широкому спектру функцій та інструментів, що надаються. Це дозволяє розробникам створювати складні ігрові сценарії, графіку, звукові ефекти та багато іншого, забезпечуючи високу якість та продуктивність гри.

Середовище розробки Visual Studio являють собою повний набір засобів розробки для створення веб-додатків ASP.NET, XML (веб-служби), настільних додатків і мобільних додатків. Всі мови (Visual Basic, Visual C# і Visual C++) використовують єдине інтегроване середовище розробки (IDE), дозволяє спільно використовувати засоби й спрощує створення рішень на базі декількох мов. Ключовою особливістю платформи .NET Framework є бібліотека базових класів. Бібліотека класів .NET Framework складається з просторів імен. Кожне простір імен містить типи, які можна використовувати в програмах: класи, структури, перерахування, делегати та інтерфейси.

3.3. Інструкція користувача системи

Після запуску програми відкривається інтерфейс головного вікна комп'ютерної гри «Сапер», вигляд якого наведено на рис. 3.2.

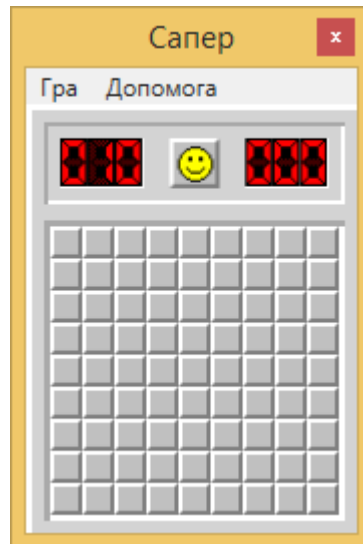


Рис. 3.2 Головна форма гри «Сапер»

В комп'ютерній грі реалізована можливість вибору рівня складності. Оберіть рівень складності гри. В програмі доступні такі рівні, як початківець, шанувальник та професіонал. Кожен рівень має відмінність в розмірі поля та кількості мін. Якщо гравець передумав грати можна натиснути на кнопку «Вихід».

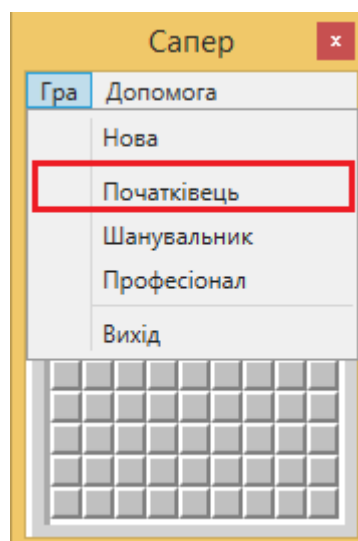


Рис. 3.3 Вибір рівня початківець

Після вибору рівня складності з'явиться ігрове поле, яке складається з квадратних клітинок. Кожна клітинка може мати міну або бути порожньою. Метою гри «Сапер» є відкрити всі порожні клітинки на полі, уникаючи вибуху мін. Ваше завдання - знайти всі міни, помічаючи їх за допомогою прапорців.



Рис. 3.4 Процес гри «Сапер»

Клікніть на клітинку, щоб її відкрити. Якщо вона має міну, гра закінчується. Якщо клітинка порожня, з'явиться число, яке показує, скільки мін знаходиться в сусідніх клітинках, що наведено на рис. 3.4.

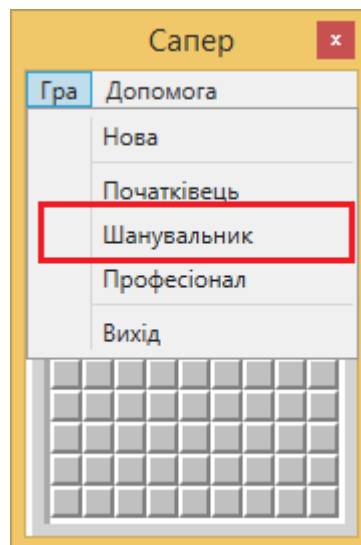


Рис. 3.5 Вибір рівня шанувальник

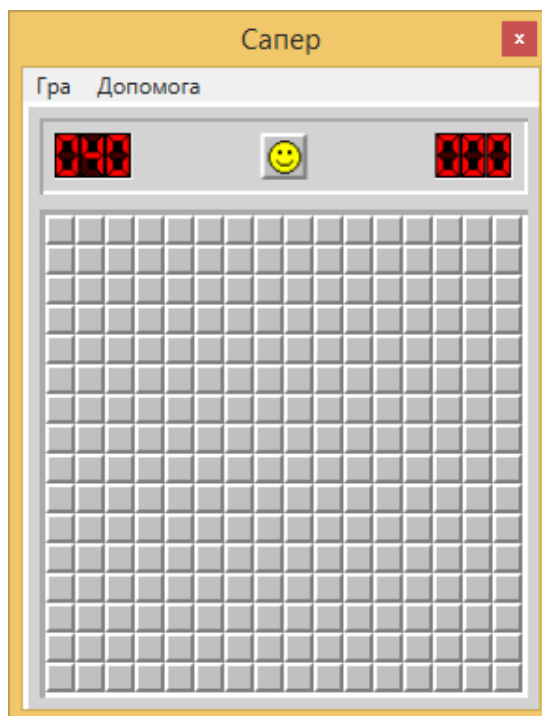


Рис. 3.6 Ігрове поле рівня шанувальник

Якщо ви вважаєте, що певна клітинка має міну, помістіть на неї прапорець, натиснувши правою кнопкою миші або використовуючи відповідну функцію на сенсорному екрані, вигляд якого наведено на рис. 3.7.



Рис. 3.7 Гра закінчилась перемогою

Використовуйте логічне мислення та стратегію для виявлення мін та безпечного відкриття клітинок. Користуйтеся числами на відкритих клітинках, щоб визначити, де можуть бути міни. На рисунку 3.8 представлений варіант програшу гри «Сапер».



Рис. 3.8 Гра закінчилась поразкою

Рівень складності професіонал в грі «Сапер» є найвищим рівнем і призначений для досвідчених гравців, які шукають виклику та складних головоломок. Розмір поля в рівень складності професіонал має великі розміри поля. Це поле розміром 30x16 клітинок.

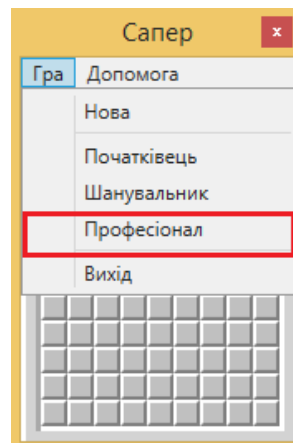


Рис. 3.9 Вибір рівня професіонал

Рівень професіонал характеризується високою кількістю мін на полі. Зазвичай це число становить близько 99 мін, що наведено на рис. 3.10.

Складність розташування мін - вони розташовані в складних та випадкових позиціях. Їх розміщення не має логічного порядку, що робить гру більш випробувальною. На цьому рівні складності важко передбачити місця розташування мін, тому користувачу доведеться багато робити висновків і приймати рішення на основі логічного мислення та інтуїції.

Завдяки великій кількості мін і складному розташуванню, рівень професіонал вимагає від користувача більшої уваги та уважності. Навіть невелика помилка може призвести до поразки.

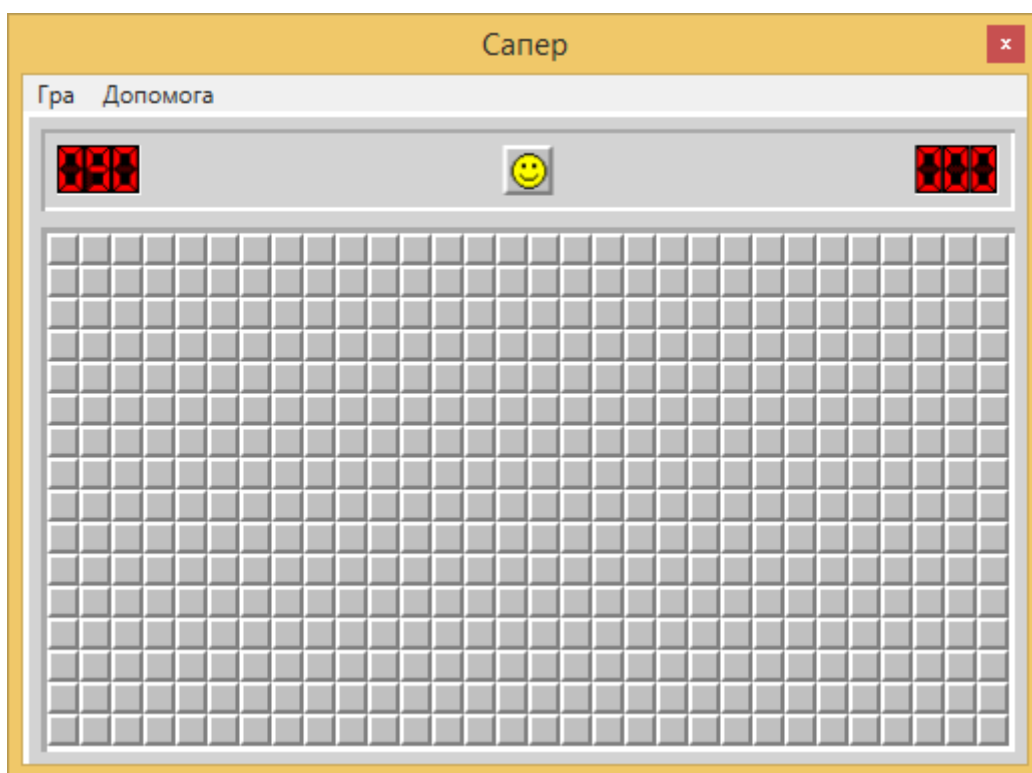


Рис. 3.10 Ігрове поле рівня професіонал

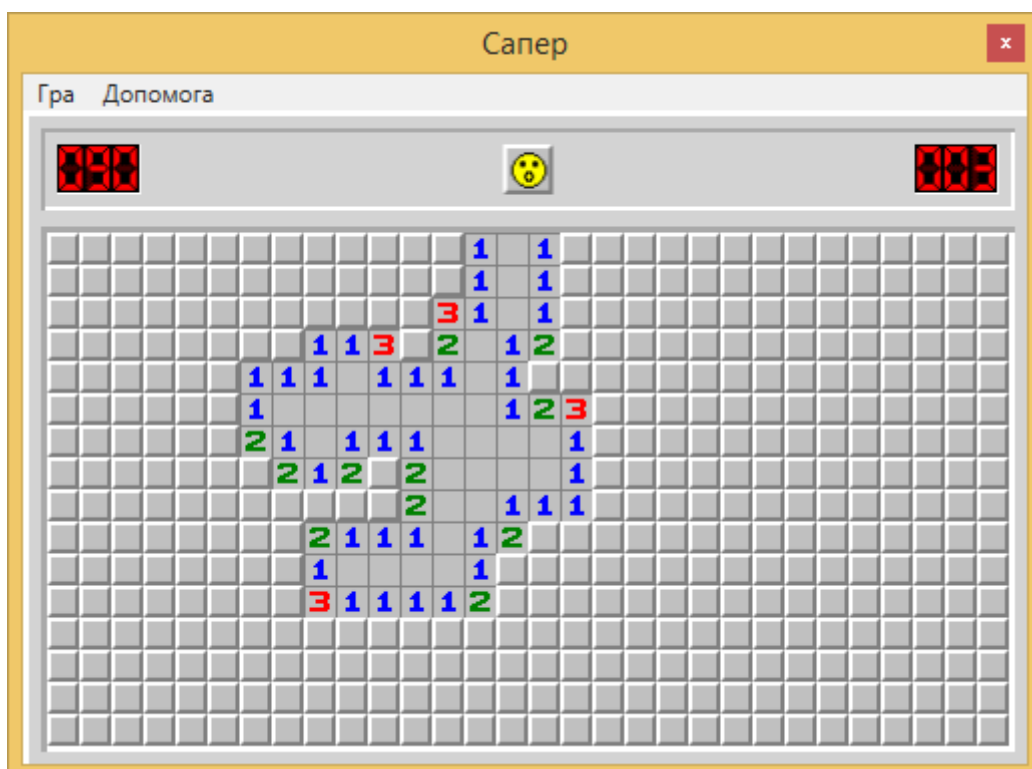


Рис. 3.11 Гра закінчилась перемогою

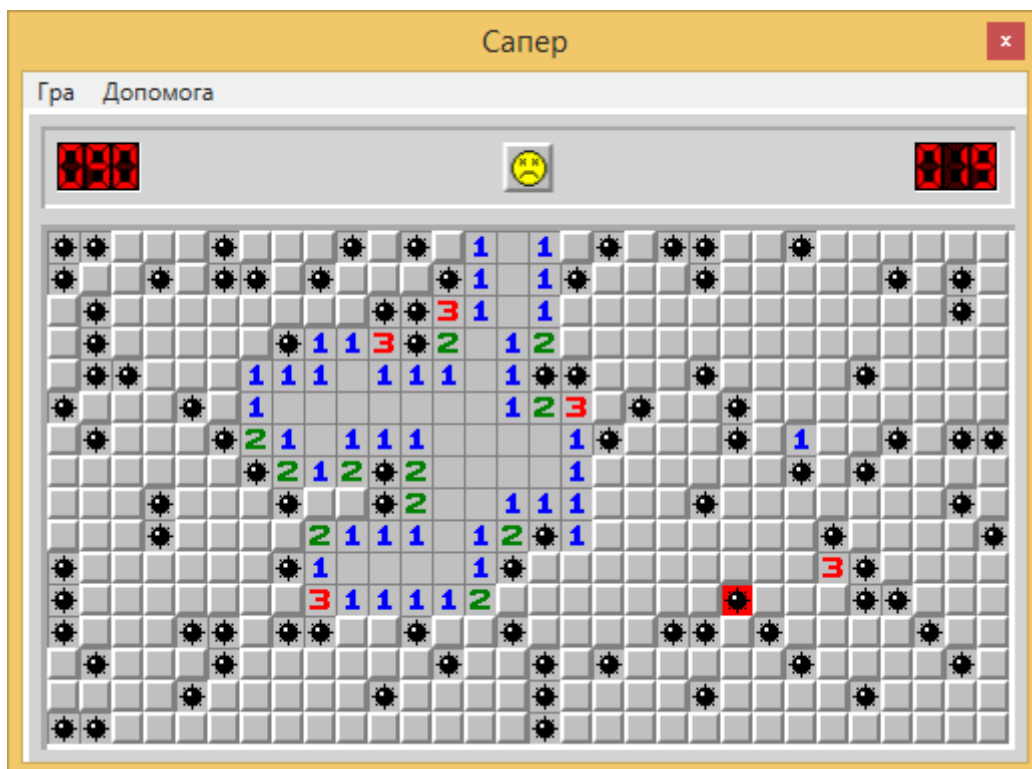


Рис. 3.12 Гра закінчилась поразкою

Рівень складності професіонал в грі «Сапер» є справжнім випробуванням для навичок гравця в іграх логічного мислення та стратегії. Він вимагає ретельного аналізу та розгадування головоломок, роблячи гру захоплюючою. На рисунках 3.13 та 3.14 представлена інформація про розробника програмного забезпечення гри «Сапер».

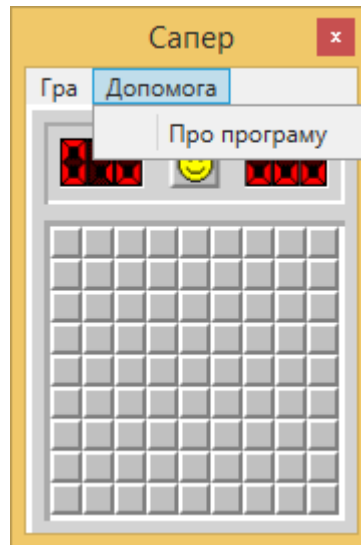


Рис. 3.13 Вибір меню «Допомога»

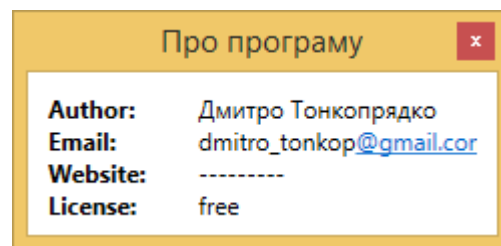


Рис. 3.14 Інформаційне вікно

В даному розділі дипломної роботи була спроектована блок-схема алгоритму роботи програмного забезпечення. Обґрунтовано технічні вимоги системи та інструкція користувача системи. Зроблено обґрунтування про доцільність автоматизувати систему саме засобами розробки платформи .NET в середовищі Microsoft Visual Studio мовою C#.

ВИСНОВКИ

Дипломна робота присвячена розробці програмного забезпечення, що реалізовує логічну гру «Сапер» в режимі користувач – комп'ютер. Додаток дозволить організовувати людині інтелектуальний і водночас емоційний відпочинок від монотонної роботи.

В аналітичній частині були розглянуті існуючі програмні додатки, які більшість своєї роботи реалізують за допомогою інформаційних технологій.

В даній дипломній роботі було проведено детальний аналіз та дослідження комп'ютерної гри «Сапер». Були визначені технічні вимоги до гри, такі як процесор, оперативна пам'ять, простір на жорсткому диску та відеокарта. Також було виявлено, що для успішної установки гри необхідний додатковий програмний пакет Visual Studio 2019.

У процесі дослідження було встановлено, що Visual Studio 2019 надає розробникам широкі можливості для створення різноманітних додатків, включаючи комп'ютерні ігри. Інтегроване середовище розробки (IDE) спрощує процес розробки і забезпечує зручні інструменти для написання коду, відлагодження та розгортання гри.

Застосування Visual Studio 2019 у розробці гри «Сапер» дозволяє покращити продуктивність та якість гри, шляхом розширення можливостей розробників для створення складних ігрових сценаріїв, графіки та звукових ефектів.

Отже, на основі проведеного дослідження можна зробити висновок, що використання Visual Studio 2019 у розробці комп'ютерної гри «Сапер» є ефективним і корисним, що сприяє поліпшенню процесу розробки та якості готового продукту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. М.С. Львов, О.В. Співаковський. Вступ до об'єктно-орієнтованого програмування. 209 с. Mathematics 1989/90. 348 с.
2. Войтенко В.В. Морозов А.В. Теорія та практика (мова C++). - Житомир, 2002.
3. Винник В. Ю. Основи програмування мовою Cі++ - Житомир, 2008.
4. Павловская Т.А. C/C++. Программирование на языке высокого уровня. — СПб.: Питер, 2003. — 461с.
5. Прайс Д., Гандэрлой М. Visual C#.NET. Полное руководство. — Киев: «Век», 2004. — 960 с.
6. Робинсон С., Корнес О., Глинн Дж. и др. C# для профессионалов. В двух томах. — М.: Лори, 2003. — 512 с.
7. Робисон У. C# без лишних слов. — М.: ДМК Пресс, 2002. — 352 с.
8. Секунов Н. Самоучитель C#. Серия «Самоучитель». — СПб.: БХВ-Петербург, 2001. — 576 с.
9. Смайли Д. Учимся программировать на C# вместе с Джоном Смайли. — Киев: «ДиаСофт», 2003. — 528 с.
10. Тай Т., Лэм Х. К. Платформа .NET. Основы. — СПб.: Символ-Плюс, 2003. — 336 с.
11. Троелсен Э. C# и платформа .NET. Библиотека программиста. — СПб.: Питер, 2002. — 796 с.
12. Фролов А. В., Фролов Г. В. Язык C#. Самоучитель. — М.: Диалог-МИФИ, 2003. — 560 с.
13. Шилдт Г. C#: учебный курс. — СПб.: Питер, 2002. — 512 с.: ил.
14. Шилдт Г. Полный справочник по C#. — М.: Издательский дом «Вильямс», 2004. — 752 с.
15. Федоров А. А. Microsoft Visual Studio 2008. Краткий обзор ключевых новинок. 2008.