

Черкаський національний університет імені Богдана Хмельницького
Кафедра інформаційних технологій

КУРСОВА РОБОТА

З дисципліни “Програмування та алгоритмічні мови”

На тему: “Гра “Сто сірників””

Студента 1 курсу КН-19 групи
спеціальності 122 Комп’ютерні науки
Олексюка Б.Ю.

Керівник: ст. викладач Царик Т.Ю.

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії _____

(підпис) (прізвище та ініціали)

(підпис) (прізвище та ініціали)

(підпис) (прізвище та ініціали)

м. Черкаси – 2020 рік

Зміст курсової роботи

Вступ.....	3
Розділ 1. Огляд алгоритмів та методів для створення гри.....	5
1.1. Огляд способів створення гри.....	5
1.2. Опис та вибір алгоритму гри.....	5
1.3. Огляд та вибір графічного зображення гри.....	6
1.4. Висновок.....	7
Розділ 2.....	8
2.1. Створення алгоритму роботи програми.....	8
2.2. Створення основного алгоритму.....	9
2.3. Створення допоміжних алгоритмів.....	9
2.4. Висновок.....	10
Розділ 3.....	12
3.1. Реалізація основного алгоритму гри.....	12
3.2. Реалізація допоміжних алгоритмів гри.....	14
3.3. Реалізація графічного зображення гри.....	15
3.4. Висновок.....	19
Висновки.....	20
Список інформаційних джерел.....	21
Додатки.....	22

Вступ

Актуальність теми дослідження. В сучасному світі, який переповнений комп'ютерами, ноутбуками, смартфонами та іншими гаджетами існує безліч різних програмних продуктів, які спрямовані на полегшення життя людини. Це насамперед програми, які призначені для роботи, проведення вільного часу чи відпочинку, але не так багато є ігор, які спрямовані не лише на розваги, а й на розвиток та навчання. Дана тема, на мою думку, є досить актуальною, адже розробка комп'ютерних ігор – це одна із складових частин щоденного користування комп'ютером. Щодня мільйони людей використовують ПК, ноутбуки, смартфони та інші пристрої для вирішення повсякденних проблем, на роботі, для відпочинку чи спілкування. Деякі ж вбачають в комп'ютері лиш засіб для розваг. Обрана мною тема несе в собі не лише розважальний характер для користувача, а й навчальний. Гра “Сто сірників” – це програмний продукт, який націлений як на людей старшого віку, так і на людей молодшого віку. Кожен зможе знайти щось для себе. Хтось може вбачати в цій грі лиш розвагу, а хтось може використовувати її для саморозвитку.

Мета полягає у вивченні необхідних для розробки даного проекту розділів програмування на мові C#, розробці програмного продукту, складання блок-схем до програми.

Завдання роботи:

1. Вивчити мову програмування, володіти нею на достатньому рівні для реалізації курсової роботи.
2. Дослідити схожі програмні продукти задля кращої реалізації.
3. Скласти алгоритм створення програмного продукту.
4. Побудувати блок-схеми до гри.
5. Визначити з метод реалізації даного проекту.
6. Написати програмний код.
7. Візуально оформити гру.
8. Написати записку до курсової роботи.

Правила гри: З купи, яка спочатку містить 100 сірників двоє гравців по черзі беруть кілька сірників: не менше одного, не більше десяти. Програє той, хто взяв останній сірник.

Розділ 1. Огляд алгоритмів та методів для створення гри

1.1. Огляд способів створення гри

При виборі методів слід пам'ятати, що є безліч варіантів як можна створити даний програмний продукт. Можна створити консольну програму гри, можна для створення використати Windows Forms[2], а можна використати інші програми, які створені саме для проектування ігор, такі як от наприклад Unity. Даний проект реалізовано за допомогою Windows Forms, адже цей метод є одним із найдоступніших та цікавих способів створення гри.

1.2. Опис та вибір алгоритму гри

Існує кілька способів, кілька алгоритмів, за допомогою яких можна створити дану гру. Обирати слід серед двох варіантів: з використанням циклів та без використання циклів. Розглянемо перший алгоритм (з циклами)[3]. Даний алгоритм полягає в тому, що спочатку задається початкова кількість сірників (100 сірників), після чого створюється кілька числових перемінних задля подальшої роботи гри. В циклі описується хід гравця та комп'ютера, кількість сірників вводиться вручну, тому відбувається також перевірка на правильність вводу числа, воно має бути строго менше десяти і більше одного, якщо перевірка показала, що все введено правильно, то після цього від початкового значення кількості сірників поступово віднімаються взяті сірники гравцем та комп'ютером, кількість сірників, які бере комп'ютер визначається випадковим чином. Якщо ж перевірка показала, що введено некоректне число, виводиться повідомлення про неправильність вводу та гравцю надається нова спроба витягти сірники. В разі, якщо гравець взяв останній сірник, перемога дістається комп'ютеру, і навпаки, якщо комп'ютер бере останній сірник, то перемога дістається гравцю. Даний алгоритм дуже зручно використовувати при створенні гри у консольному виді.

Обраний мною алгоритм є трішки інакшим від описаного вище. Оскільки продукт створюється, використовуючи Windows Forms, можна використати

інший спосіб реалізації програмного продукту. Суть полягає в створенні форми, в якій присутні 10 кнопок, пронумеровані від 1 до 10. Початкова кількість сірників може бути задана за допомогою елемента Label, а ходи описані використовуючи елемент Listbox. Для кнопок, які пронумеровані, створюється один і той самий клік, тобто функція в них буде одна й та сама, проте число витягнутих сірників (число на кнопці) буде іншим. Першим ходитиме гравець, йому лиш слід обрати кнопку та натиснути на неї. Хід комп'ютера, кількість сірників, які він витягне, буде визначатись теж випадковим способом. По черзі гравець та комп'ютер витягуватимуть сірники, в кінці, коли залишиться менше десяти сірників і хід буде належати комп'ютеру, він забиратиме всі сірники, тим самим віддаючи перемогу гравцю.

1.3. Огляд та вибір графічного зображення гри

Існує дуже багато способів графічного зображення. Можна використати консоль, та деякі елементи, які можна зобразити в ній для графічного відображення гри. Можна застосувати інші програми, але також можна скористатися засобами Windows Forms. Існує кілька способів: з використанням лише однієї форми та з використанням декількох форм. В першому випадку створюється одна єдина форма, на якій вміщуються кнопки, Listbox та Label. Меню до гри можна створити за допомогою елемента MenuStrip, додавши його в нашу форму та створивши потрібні нам кнопки меню. По закінченню гри в Listbox виводитиметься текст зі вказанням переможця. Другий спосіб полягає у створенні кількох форм, а саме: форма головного меню, форма власне гри та форма, в якій буде відображена інформація про гру. В головному меню буде три кнопки: "Start game", "Info", "Exit". При натисненні кнопки "Start game" буде відкриватись форма власне гри, при натисненні "Info" відкриватиметься форма з інформацією про продукт, а при натисненні "Exit" гра буде закриватись. Форма власне гри матиме вигляд як і в попередньому варіанті оформлення, проте може бути додано кілька нових кнопок для рестарту гри та виходу із неї. Можна також змінити колір кнопок та всіх елементів форм задля привабливого вигляду

продукту. Також можна додати задній фон форми, обравши фото, створене раніше, або ж за допомогою графіки мови програмування C# створити власний фон у формі.

1.4. Висновок

Отже, в даному розділі було описано різні варіанти, методи та алгоритми створення програмного продукту. Найбільш вдалим, на мою думку, буде варіант з використанням кількох форм.

Розділ 2. Проектування гри

2.1. Створення алгоритму роботи програми

При створенні алгоритму роботи програми слід пам'ятати, що гра має бути насамперед зручною у використанні, тобто мати зручний інтерфейс та бути візуально привабливою, щоб заохотити користувача грати в цю гру. Дана програма матиме кілька форм (кілька вікон). При запуску гри відкривається головне меню, на якому присутні три кнопки: "Start Game", "Info", "Exit". При натисненні кнопки "Exit" відбувається вихід з гри, при натисненні "Info" відкривається нове вікно, в якому вказані правила гри та деяка інформація, якщо ж буде натиснута кнопка "Start Game" відкриється вікно власне гри, де гравець зможе грати в гру, при цьому вікно головного меню зникне з поля зору гравця, щоб запобігти проблем з перемиканнями вікон. У вікні власне гри реалізовано десять кнопок, пронумерованих від одного до десяти, кнопка виходу зі гри "Exit Game", кнопка рестарту гри "Restart", Listbox, в якому буде відбуватися інформування гравця про його хід, чи хід комп'ютера та Label, за допомогою якого буде висвітлено кількість сірників у купі. На Рис.2.1. зображена блок-схема взаємодії з головним меню та з вікном власне гри.

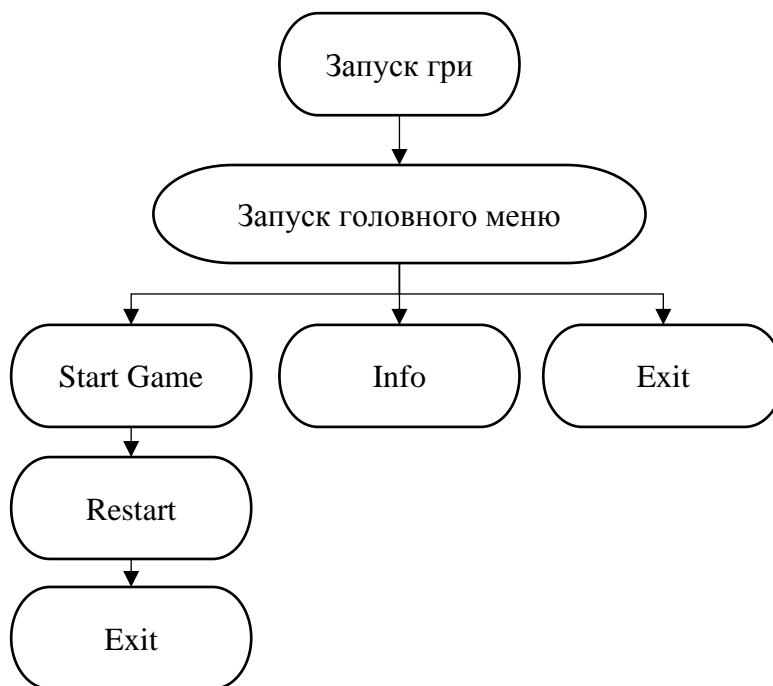


Рис.2.1. Алгоритм взаємодії з ігровими меню

2.2. Створення основного алгоритму

Створення даного алгоритму має на меті реалізувати взаємодію гри з гравцем. Для цього після запуску вікна власне гри буде висвітлено дане вікно задля гри в гру. В цьому вікні буде розміщено кнопки, Listbox та Label. Десять кнопок, пронумерованих від одного до десяти матимуть одну й ту саму функцію (один й той самий клік). Текст на кнопці, тобто нумерація, буде згодом використана для ініціалізації кількості взятих сірників гравцем, тобто, якщо гравець натисне на кнопку з написом вісім, це означатиме, що гравець взяв рівно вісім сірників. Після натискання кнопки від загальної кількості сірників буде відніматись кількість витягнутих гравцем. Початкова кількість сірників буде висвітлена за допомогою елемента Label. З поступовим витягуванням сірників гравцем та комп'ютером значення кількості буде зменшуватись відповідно до кількості взятих сірників. В Listbox будуть відображатись ходи та буде описана кількість взятих сірників та оновлена кількість сірників у купі. По закінченні гри буде висвітлено переможця.

2.3. Створення допоміжних алгоритмів

Допоміжні алгоритми-це ті алгоритми, які допомагають гравцю взаємодіяти з грою, наприклад, руху між формами. В даній роботі буде висвітлено кілька таких алгоритмів. Це насамперед переміщення з головного меню в меню інформації про гру "Info". Даний алгоритм є дуже простим, його роботу словесно можна описати так: при натисненні кнопки "Info" в головному меню, відкривається вікно інформації, в якій гравець може ознайомитися з правилами гри Рис.2.2. Після завершення ознайомлення з правилами гравець повертається до головного меню, де далі може обрати план своїх дій: почати гру, або ж вийти з неї. Інший схожий алгоритм виконує функції відкриття вікна власне гри та згорнення головного меню.

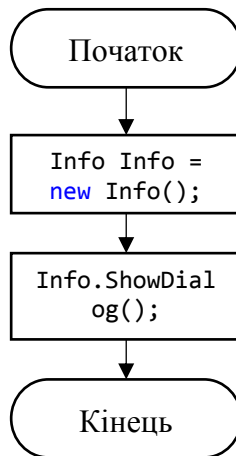


Рис.2.2. Алгоритм відкриття вікна інформації

Також у виконанні даної роботи буде використано алгоритм закриття всієї гри в цілому. Цей алгоритм можна описати так: при натисненні на кнопку "Exit" або "Exit game" гра буде закриватися Рис.2.3.

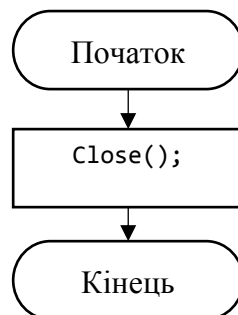


Рис.2.3. Алгоритм закриття гри

В даній частині розділу описано лише деякі допоміжні алгоритми. Більш детально з ними можна буде ознайомитися у розділі 3.

2.4. Висновок

Отже, в цьому розділі було описано алгоритми, які будуть використані у ході реалізації програмного продукту, наведені блок-схеми до деяких частин коду, та освітлено весь концепт майбутньої гри. Задачі до створення гри:

1. Простота у використанні
2. Створення вікна головного меню.
3. Створення вікна інформації.

4. Створення вікна власне гри.
5. Написання основного коду.
6. Написання допоміжних кодів.
7. Цікаве візуальне оформлення гри.
8. Створення лічильника сірників.
9. Реалізація виводу результатів гри за допомогою Listbox.

Розділ 3. Реалізація гри

3.1. Реалізація основного алгоритму гри

Основний алгоритм був реалізований наступним чином: при потраплянні у вікно гри перед гравцем стоїть ціль обрати кількість сірників, яку він хоче витягнути. Початкова кількість рівна ста, від цього значення віднімається кількість взятих сірників гравцем, при цьому оновлюється значення кількості сірників у купі. Після цього хід переходить комп'ютеру. Коли сірників на купі не залишиться виводиться результат про переможця. По закінченню сірників на купі баланс сірників рівняється нулю. На рисунку 3.1. зображено фрагмент коду, який відповідає за зчитування даних з кнопок та хід гравця.

```
Button but = (Button)sender;
int count = Convert.ToInt32(but.Text);
int balance = Convert.ToInt32(label1.Text);
if (balance <= 0)
{
    return;
}
balance -= count;
string match = "match";
if (count > 1)
{
    match = "matches";
}
listBox1.Items.Add("Player tooks " + count.ToString() + match + " there are only " + balance + "matches");
if (balance <= 0)
{
    label1.Text = "0";
    listBox1.Items.Add("Computer win ");
    return;
}
```

Рис3.1. Алгоритм ходу гравця

З рисунку видно, що задається дві перемінних типу `int`, яким присвоюється значення числа на кнопці (`count`) та значення початкової кількості сірників (`balance`). Далі, якщо баланс менший або рівний нулю, відбувається зупинка програми. Після того, як було зчитано значення взятих сірників (`count`) гравцем, від початкової кількості віднімається значення витягнутих сірників та присвоюється нове значення з уже оновленими даними. Після цього створюється ще одна перемінна типу `string`. Вона створена для зручності виводу результатів, адже коли гравець бере 1 сірник на екран виведеться саме “match”, а не “matches”. Після цієї перевірки відбувається вивід результату на екран. Далі виконується ще одна перевірка, вона зроблена для останнього ходу, коли в купі

буде дуже мало сірників, її суть така: якщо баланс буде менший або дорівнюватиме нулю, ми присвоюємо балансу значення нуль, та виводимо переможця на екран, після чого робота програми зупиняється і програма очікує на подальші дії: на рестарт гри або вихід з неї. Хід комп'ютера зображений на Рис3.2.

```
if (balance > 20)
{
    Random rand = new Random();
    count = rand.Next(10) + 1;
}
else
    if (balance < 10)
    {
        count = balance;
    }
match = " match";
if (count > 1 && balance > 1 || balance <= 0)
{
    match = " matches";
}
balance -= count;
label1.Text = balance.ToString();
listBox1.Items.Add("Computer tooks " + count.ToString() + match + " there are only " + balance + " matches");
if (balance <= 0)
{
    label1.Text = "0";
    listBox1.Items.Add("Player win ");
    return;
}
```

Рис.3.2. Алгоритм ходу комп'ютера

З фрагменту коду видно, що кількість взятих сірників комп'ютером визначається випадковим чином, за це відповідає умовний оператор if, який перевіряє, чи кількість сірників на купі більша двадцяти, якщо так, обирається випадкове число з діапазону від одного до десяти. Якщо ж кількість в купі менша 10, то комп'ютер автоматично забирає всі сірники. Точно так само, як і у випадком з ходом гравця, від загальної кількості сірників на купі віднімається кількість витягнутих та присвоюється нове значення кількості сірників у купі. Після ходу обох гравців оновлюється значення сірників у купі та виводиться на екран оновлене число. Далі ж відбувається вивід на екран результатів ходу комп'ютера, після чого виконується перевірка, чи кількість сірників більша рівна нулю задля розуміння завершення гри. Якщо ж ця кількість менша або рівна нулю, баланс прирівнюється до нуля, виводиться на екран результат гри, після чого наша гра очікує на подальші дії: на рестарт гри або вихід з неї. Як і у випадку

з ходом гравця, для правильності виводу результатів використана перемінна “match”.

3.2. Реалізація допоміжних алгоритмів гри

Одним із таких алгоритмів є алгоритм переходу з вікна головного меню у вікно власне гри. Він дуже простий, його можна описати так: при натисненні кнопки “Start Game” автоматично відкривається вікно гри, а головне меню згортається. Даний алгоритм зображено на Рис.3.3.

```
private void StartButton_Click(object sender, EventArgs e)
{
    Game game = new Game();
    game.Show();
    Hide();
}
```

Рис.3.3. Алгоритм відкриття вікна власне гри

Інший схожий алгоритм використаний для відображення вікна інформації, в якому вказано правила гри. Даний алгоритм не має на меті згорнути головне меню. Це зроблено для того, щоб гравець при необхідності міг одночасно дивитися у правила та грати в гру. Цей алгоритм висвітлено на Рис3.4.

```
private void InfoButton_Click(object sender, EventArgs e)
{
    Info Info = new Info();
    Info.ShowDialog();
}
```

Рис.3.4. Алгоритм відображення вікна інформації

Наступний ще простіший алгоритм, який відповідає за закриття гри показано на Рис3.5. Його суть полягає в тому, що при натисненні кнопки ”Exit” гра припинятиме свою роботу, тобто закриватиметься.

```
private void ExitButton_Click(object sender, EventArgs e)
{
    Close();
}
```

Рис.3.5. Алгоритм закриття гри

Алгоритм рестарту гри описується теж просто. При натисненні кнопки “Restart” значення балансу присвоюється число сто, тобто оновлюється лічильник та очищується Listbox, тим самим гра починається знову. Даний алгоритм зображено на Рис.3.6.

```
private void NewGame_Click(object sender, EventArgs e)
{
    label1.Text = "100";
    listBox1.Items.Clear();
}
```

Рис.3.6. Алгоритм рестарту гри

3.3. Реалізація графічного зображення гри

Як уже було зазначено вище гра матиме кілька вікон (Windows Forms). При запуску гри відбувається відкриття першої форма, де вміщено три кнопки: “Start Game”, “Info” та “Exit”. Перед створенням кнопок у форму було розміщено елемент TableLayoutPanel, якому було задано такі параметри: кількість рядків дорівнює три, а кількість стовпців дорівнює один, а також було змінено його колір заднього фону на червоний. В комірки даного елементу було вміщено три кнопки. Кольори кнопок “Start Game”, “Info” та “Exit” було змінено на зелений, жовтий та червоний кольори відповідно. Розмір тексту кнопок був збільшений, а шрифт змінено на Open Sans. Також для головного меню було створено задній фон, за допомогою графічного редактора Adobe Photoshop. На фоні зображено сірники та змінено колір фону на фіолетовий. Загальний вид головного меню зображено на Рис.3.7.

Параметри головного меню були виставлені таким чином. Мінімальний розмір вікна має такі параметри: 600 пікселів по горизонталі та 600 пікселів по вертикалі. Додано задній фон форми, та змінено назву на “100 Matches”. Також при створенні було враховано той факт, що користувач захоче змінювати розмір вікна, тому вікно зроблено так, щоб при збільшенні та зменшенні елементи вікна не втрачали свої розміри та розташування, а залишалися на центрі, змінюючи лише свій розмір.при цьому задній фон розширюється і не втрачає привабливого

виду після збільшення вікна, а елемент `TableLayoutPanel` збільшується або зменшується у відповідності до розтягнення вікна.

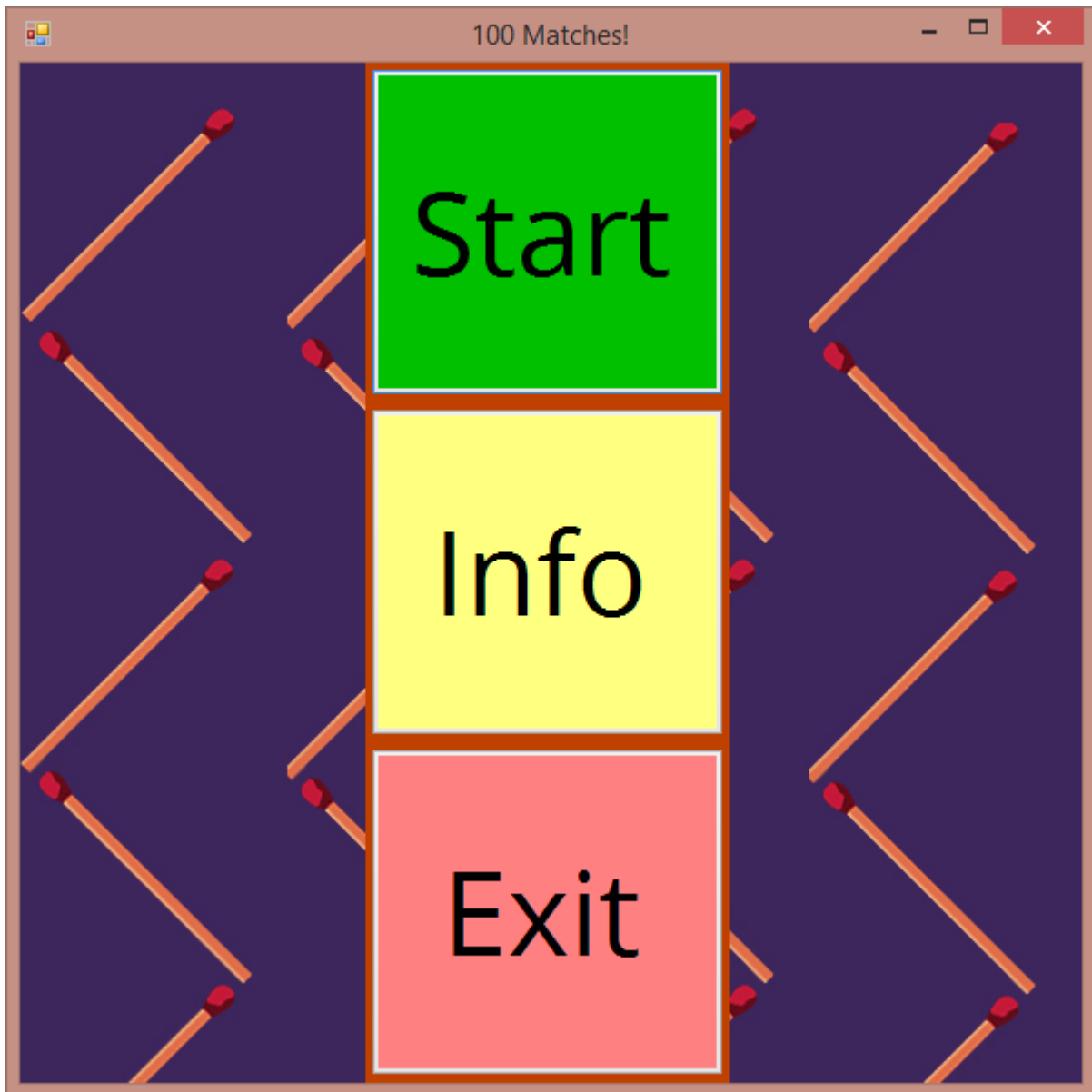


Рис.3.7. Зображення головного меню

При створенні вікна інформації було використано всі ті самі параметри, що і для головного меню, проте колір заднього фону було змінено з фіолетового на сірий, та додано елемент `Label` для відображення тексту (правил гри), який в свою чергу поміщений в елемент `TableLayoutPanel`. Розмір шрифту був збільшений, а сам шрифт замінено на `Open Sans`. Також були виставлені мінімальний розмір вікна (600 пікселів по вертикалі та 600 пікселів по горизонталі) і змінено назву вікна на "Info". Вікно зроблено таким чином, щоб

під час збільшення елементи залишались відцентрованими та не з'їжджали зі своїх місць. Вікно інформації зображено на Рис.3.8.

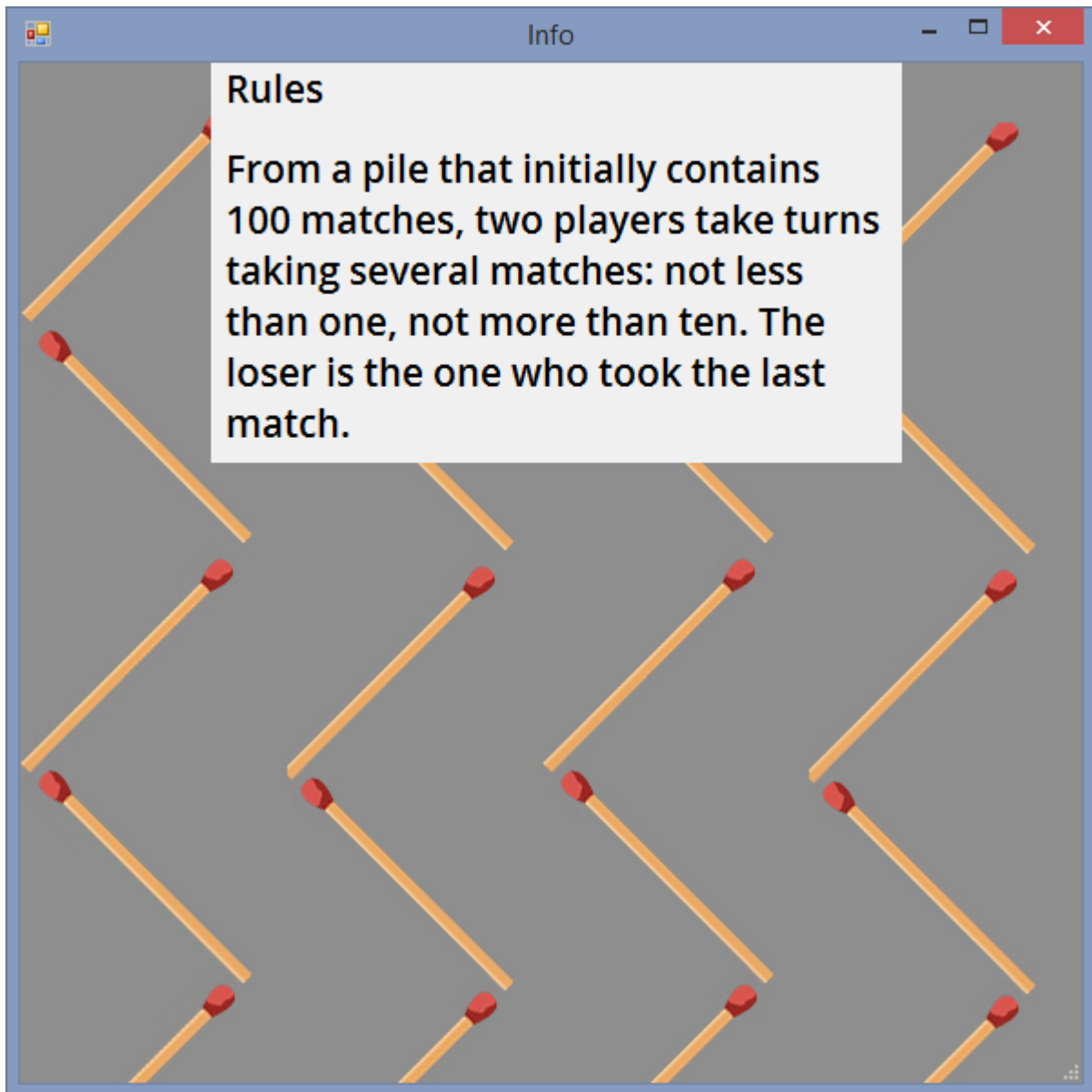


Рис.3.8. Зображення вікна інформації

При створенні вікна власне гри також було використано елемент `TableLayoutPanel`, проте уже двічі. Перший раз для розміщення пронумерованих кнопок, вдруге для розміщення кнопок виходу з гри та рестарту. Також додано елемент `Label` та `Listbox`. Задній фон було додано такий самий як і в попередніх двох випадках, проте колір було змінено на менш яскравий фіолетовий. Всі елементи закріплені так, щоб при збільшенні жоден з них не змінював свого положення. Колір пронумерованих кнопок був змінений на помаранчевий, кольори `TableLayoutPanel1` та `TableLayoutPanel 2` на жовтий, кнопки виходу та

рестарту мають червоний та зелений кольори відповідно. На Рис.3.9. зображено вікно власне гри.

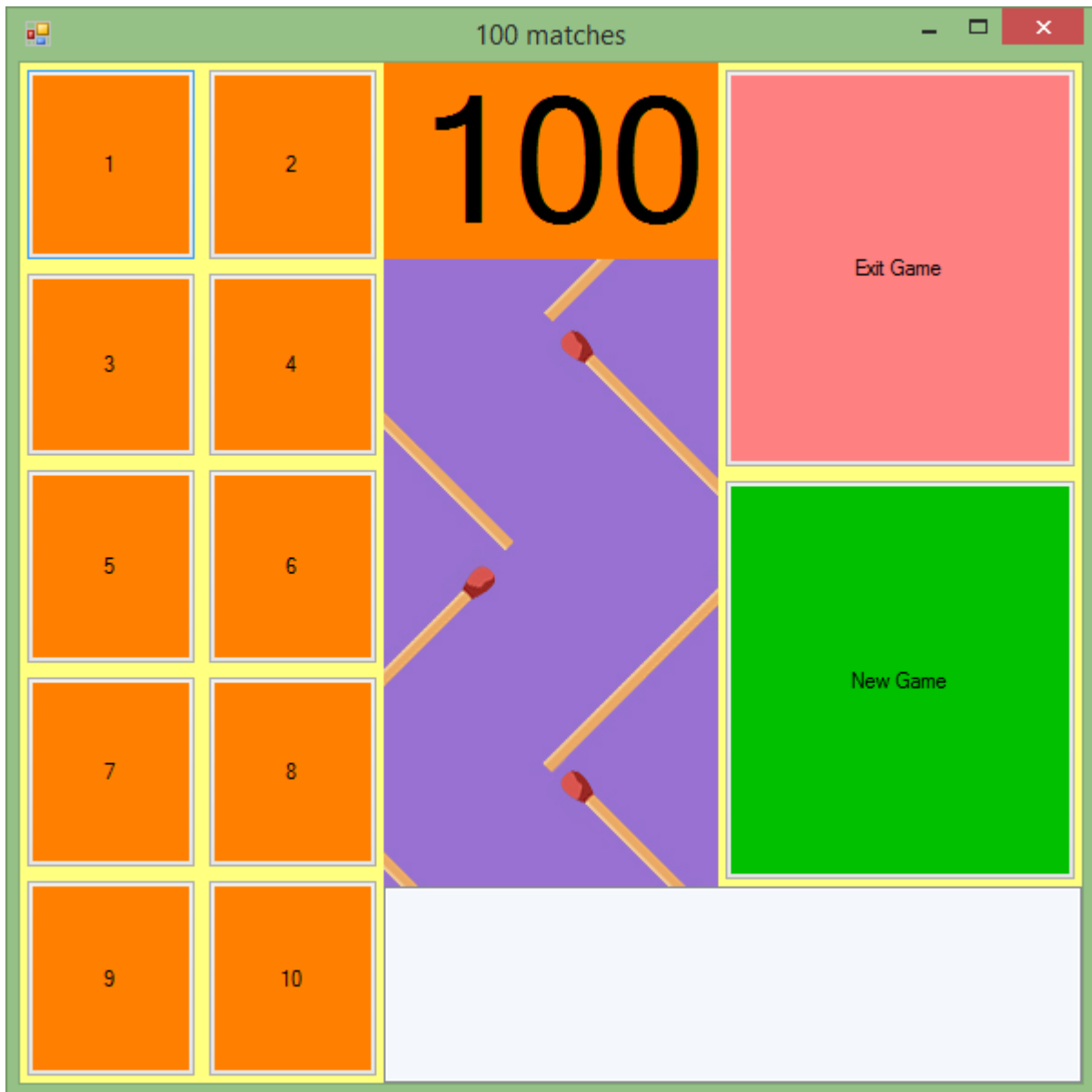


Рис.3.9. Зображення вікна власне гри

Також слід зауважити, що при натисненні кнопки “Exit game” у цьому вікні автоматично закривається повністю гра з усіма вікнами. Колір лічильника було змінено на помаранчевий, а текст збільшено, щоб краще бачити кількість сірників у купі.

Результат роботи програми проілюстровано на Рис.3.10. На рисунку видно, що після певної кількості ходів виграв гравець, оскільки комп’ютер забрав останні сірники з купи. Видно також, що кількість сірників на купі зменшилась до нуля, тобто гравець та комп’ютер витягнули всі сірники.

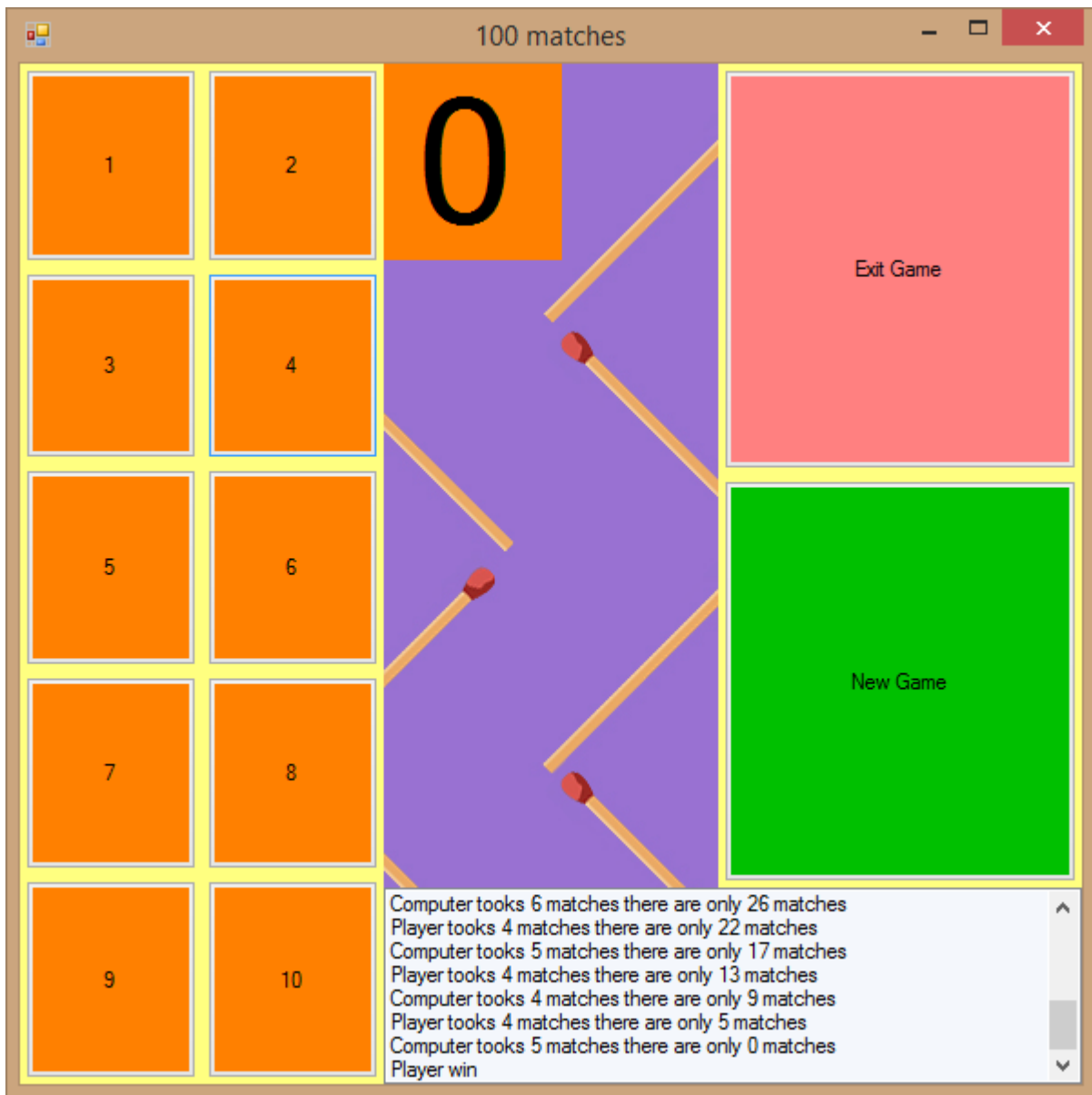


Рис.3.10. Результат роботи гри

3.4. Висновок

Отже, в даному розділі було показано програмний код та інтерфейс програми. Описано детально створення графічного зображення гри. Було проведено перевірку роботи гри та висвітлено результат.

Висновки

Отже, перед розробкою даного проекту мною було опрацьовано багато матеріалу, який було використано у ході розробки даної програми, створено блок-схеми, в тому числі і основного алгоритму (Додаток “А”), проаналізовано та обрано алгоритм створення гри. За допомогою Windows Forms було створено дану гру, яка складається з трьох вікон, кожне з яких має свій функціонал. Даний програмний продукт можуть використовувати люди будь-якого віку та сфери діяльності, які хочуть знайти в іграх не лише розвагу, а ще й спосіб розвиватися.

В майбутньому планується покращити даний програмний продукт. Насамперед оптимізувати його, покращити основний алгоритм роботи, змінити інтерфейс на більш цікавий та розробити план розвитку даної гри.

Список використаних джерел

1. Методичні вказівки до виконання та оформлення курсової роботи з дисциплін “Основи програмування”, “Програмування та алгоритмічні мови”, “Алгоритмізація та програмування”. Черкаси: Вид. від ЧНУ імені Богдана Хмельницького, 2016.-32с.
2. Create a Windows Forms app in Visual Studio with C# [Електронний ресурс] Режим доступу: <https://docs.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2019> Перевірено 24.05.2020
3. Игра 100 спичек [Електронний ресурс] Режим доступу: <https://programmersforum.ru/showthread.php?t=299931> Перевірено 24.05.2020

Додатки

Додаток А

