**Лабораторна робота № 5**

Тема: Інтерфейс користувача та елементи керування

Мета:Навчитися створювати динамічні додатки, що працюють за принципами керування подіями

**Теоретичні відомості**

**Створення динамічних додатків на мові C#**

Використовуючи елемент управління pictureBox, створимо просту гру "Перегони жуків". Суть гри полягає в наступному: перемагає той, чий жук прибіжить до фінішу першим. Для гри потрібні два рисунки жучків, один буде червоним, інший – синім. Рисунки можна намалювати, наприклад, в редакторі Paint, або завантажити до себе в папку із проектом представлені тут рисунки:

Рисунок 5.1 – Зображення жуків для гри «Перегони жуків»

Створивши новий проект, варто розташувати на формі наступні елементи управління (рис. 2): 1) textBox1, textBox2; 2)label1, label2; 3)button1, button2; 4) panel1; 5) pictureBox1, pictureBox2; 6)menuStrip1; 7) timer1.

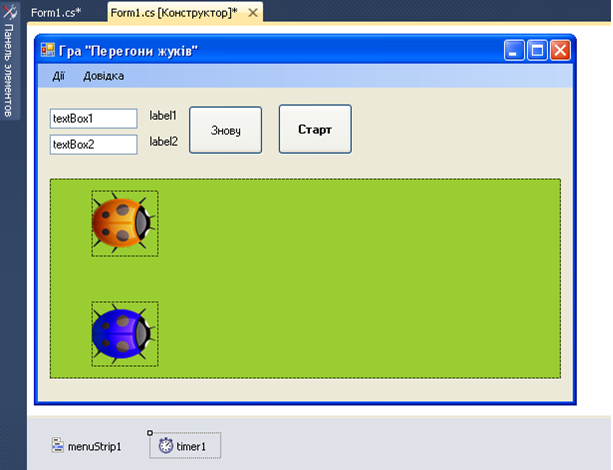


Рисунок 5.2 – Створення інтерфейсу користувача для гри «Перегони жуків»

В об'єкти pictureBox1 і pictureBox2 необхідно завантажити рисунки жуків, для цього спочатку натисніть на кнопку «…» біля властивості Image цих елементів, а потім на кнопку Import на панелі ресурсів.

Коли рисунки будуть завантажені, треба привласнити панелі panel1 у властивості BackColor такий же колір, як і тло рисунків жуків.

Для визначення координати фінішу треба провести наступні обчислення:

Finish = panel1.Size.Width - pictureBox1.Width;

panel1.Size.Width - ширина панелі panel1,

pictureBox1.Width - ширину рисунку жука.

Отримана цифра буде означати фінішну координату.

Подвійним клацанням на виділеній формі варто створити подію й прописати в ній початкові налаштування нашої програми:

private void Form1\_Load(object sender, EventArgs e)

  {

    textBox1.Text = "Гравець 1";

    textBox2.Text = "Гравець 2";

    label1.Text = "0";

    label2.Text = "0";

    pictureBox1.Left = 1;

    pictureBox2.Left = 1;

Finish = panel1.Size.Width - pictureBox1.Width;

  }

Кнопці button1 у властивості Text варто привласнити значення "СТАРТ", а button2 значення "Знову".

Запустимо додаток на виконання; форма повинна прийняти вигляд, показаний на рисунку 3:

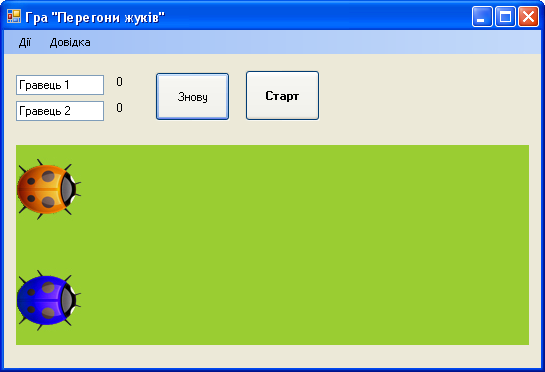


Рисунок 5.3 – Інтерфейс користувача гри «Перегони жуків»

Далі, створимо в коді програми декілька глобальних змінних:

public partial class Form1 : Form

  {

   int flag1;

   int x1, x2;

int Finish;

   public Form1()

  {

...

Змінна flag1 - призначена для контролю ігрової ситуації, якщо flag1 дорівнює 0, то жуки стоять на місці, а якщо flag1 не дорівнює 0, то жуки починають рухатися. Змінні x1 і x2 призначені для зберігання горизонтальних координат положення жуків на біговій доріжці.

Створимо подію для кнопки "Знову", ця подія очищає всі змінні й встановлює жуків у стартову позицію.

private void button2\_Click(object sender, EventArgs e)

  {

   label1.Text = "0";

   label2.Text = "0";

   x1 = 1; x2 = 1;

   pictureBox1.Left = x1;

   pictureBox2.Left = x2;

   flag1 = 0;

  }

Для кнопки "СТАРТ" теж створимо подію, вона буде дуже простою, потрібно привласнити змінній flag1 будь-яке значення не рівне 0.

private void button1\_Click(object sender, EventArgs e)

  {

    flag1 = 1;

  }

Тепер залишилося створити подію, що буде переміщати жуків. Для цього знадобиться компонент таймер timer1. Встановимо компонент на формі й зробимо деякі настроювання.



Рисунок 5.4 – Компонент таймер timer1

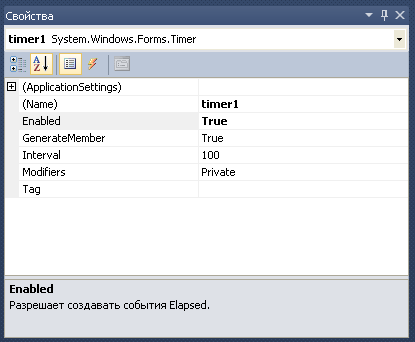


Рисунок 5.5 – Налаштування властивостей елементу керування «Таймер»

Властивість GenerateMember і Enabled повиннібути True.Властивість Interval дорівнює 100 - це один тік таймера в секунду.Якщо жуки будуть рухатися повільно, то це значення можна зменшити, наприклад до 50.

**Для того щоб визначити скільком секундам відповідає значення задане у властивості Interval ­– розділіть це число на 1000. Напр., якщо Interval=5000, то дії вказані в обробнику таймера будуть відбуватися з інтервалом в 5 секунд.**

Подвійним клацанням по іконці timer1 створимо подію руху жучків:

private void timer1\_Tick(object sender, EventArgs e)

  {

   if (flag1 != 0) // Якщо дано старт :

   {

   Random a = new Random(); // Включаємо генератор випадкових чисел

// У змінну count записуємо випадкове число в діапазоні від 0 до 8

   int count = a.Next(8);

// Нарощуємо значення координати x1 на випадкове число count

   x1 += count;

// Виводимо значення пройденого шляху для першого гравця

   label1.Text = Convert.ToString(x1);

// Зміщаємо першого жучка на випадкову величину

   pictureBox1.Left = x1;

// Створюємо випадкове число для другого гравця й повторюємо все те ж саме

   count = a.Next(8);

   x2 += count;

   label2.Text = Convert.ToString(x2);

   pictureBox2.Left = x2;

  }

// Перевіряємо, якийізгравцівдійшовдофінішуйзупиняємопроцес

  if ((x1 >= Finish) || (x2 >= Finish))

  {

   flag1 = 0;

  }

}

Запустимо програму на виконання й, натиснувши кнопку "СТАРТ" зіграємо кілька партій, щоб переконатися, що гравці приходять до фінішу зі змінним успіхом.

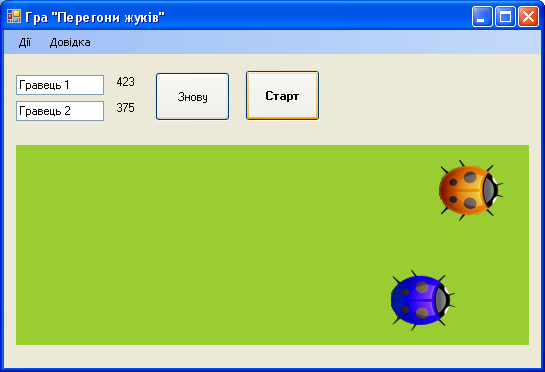


Рисунок 5.6 – Результати роботи програми «Перегони жуків»

**Завдання до лабораторної роботи:**

1. Реалізуйте описану в теоретичній частині гру «Перегони жуків». Додайте в програму поле, в якому буде виводитися ім'я переможця. Реалізуйте можливість управління програмою через меню користувача та створіть вікно довідки про програму.

2. Виконайте одне з наступних завдань за своїм варіантом (використовуючи елемент управління Timer):

1) Зобразити червоне коло, що випадковим чином пересувається по білому квадрату у вікні програми при натисненні кнопки «Старт» та зупиняється при натисненні кнопки «Стоп».

2) Виводити на екран питання та варіанти відповідей на нього, давати певний інтервал часу на відповідь. Якщо користувач відповів невірно, або не відповів по завершенню заданого інтервалу на поточне питання, віднімати 3 бали, якщо відповів вірно додавати 6 балів; після чого виводити наступне питання. Виводити проміжну та остаточну кількість балів на екран. Зробити 5 питань по 3 варіанти відповіді по мові програмування С#.

3) Виводити на екран N точок, що повинні рухатися по вікну випадковим чином, періодично (через інтервали часу тривалістю М) визначати скільки точок потрапило у ділянку вікна відокремлену квадратом nxn. Використати 2 різні елементи Timer.

4) Виводити у вікні програми через кожні 3с 6 різних та 2 однакові випадкові зображення, якщо користувач встигатиме послідовно натиснути на 2 однакові зображення, то нараховувати йому 2 бали, якщо лише на одне з них то 1 бал. По завершенні 1хв зупинити гру та вивести загальну кількість балів.

5) Зобразити на екрані таймер у форматі *години:хвилини:секунди*, що починає відлік при натисненні кнопки «Старт», зупиняється при натисненні кнопки «Стоп» та обнуляється при натисненні кнопки «Обнулити».

6) Вивести у вікні зображення світлофора (три кола), світлофор кожні N секунд повинен змінювати колір (червоний, жовтий, зелений). Релізувати функцію встановлення користувачем значення N та вибору відтінків кольорів сфітлофора.

7) Виводити на екран графік функції y = k\*sin(x) з деякою затримкою (користувач повинен мати можливість змінювати значення затримки), щоб було видно як він послідовно промальовується. При зміні користувачем коефіцієнту k, перемальовувати графік.

8) Через випадкові інтервали часу в проміжку від 3 до 8 секунд виводити на екран різні зображення тварин. Внизу вікна повинні міститися кнопки з назвами всіх використаних в програмі зображень. Якщо користувач встигатиме натиснути на відповідну поточному зображенню кнопку – нараховувати йому 10 балів.

9) Виводити у вікні фігуру, що рухається по ньому випадковим чином, та кожні 6 секунд змінює форму та колір.

10) Користувач вводить у текстове поле речення, програма змінює кожні 3с порядок слів у реченні випадковим чином (також є кнопка «старт/стоп»).

***Контрольні запитання:***

1) Які елементи керування необхідно було використати в даній лабораторній роботі? Що вони дозволяють реалізувати та які властивості мають?

2) Для чого використовується елемент керування Timer? Навіщо виконувати періодично певні дії?

3) Опишіть алгоритм Вашої програми.