

Черкаський національний університет імені Богдана Хмельницького

(повне найменування вищого навчального закладу)

Кафедра інформаційних технологій

(повна назва кафедри)

КУРСОВА РОБОТА

з дисципліни

“Програмування та алгоритмічні мови”

НА ТЕМУ «гра Вгадай слово»

Студента 1 курсу, групи КН-19

напряму підготовки «Інформаційні технології»

спеціальності «Комп’ютерні науки»

Зав’ялова Д.С.

(прізвище та ініціали)

Керівник кандидат технічних наук,

старший викладач кафедри ІТ Царик Т.Ю.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала: _____

Кількість балів: _____ Оцінка: ECTS _____

Члени комісії

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

м. Черкаси – 2020 рік

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ ім. Б. ХМЕЛЬНИЦЬКОГО (НАЗВА КАФЕДРИ, НА ЯКІЙ
ВИКОНУЄТЬСЯ КУРСОВА РОБОТА)**

ЗАТВЕРДЖУЮ

Завідувач кафедри _____,

(підпис)

„_____” _____ 2020 р.

**ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу**

студента Зав'ялова Дмитра Сергійовича групи КН-19 першого курсу

ТЕМА гра Вгадай слово

Вихідні дані до курсової роботи: _____

Зміст Пояснювальної записки до курсової роботи:

Індивідуальне завдання

Вступ

1 Огляд ...

2 Розробка схеми алгоритму ...

3 Розробка програми ...

Висновки

Список літератури

Додатки (за необхідності)

Перелік наочного матеріалу:

Програмний продукт ..., пояснювальна записка, презентація результатів роботи.

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу: до ...	Примітка
1.	Отримання завдання на курсову роботу.	04.02.2015	
2.	Огляд джерел технічної інформації за темою роботи.	18.02.2015	
3.	Виконання аналізу методів реалізації завдання	04.03.2015	
3.	Розробка алгоритму реалізації завдання	18.03.2015	
4.	Програмування алгоритму, створення програмного продукту	08.04.2015	
5.	Тестування розробленого програмного продукту	15.04.2015	
6.	Написання пояснювальної записки	01.05.2015	
7.	Подача роботи керівнику для написання відгуку.	07.05.2015	
8.	Корегування роботи за результатами розгляду керівника. Остаточне оформлення пояснювальної роботи.	14.05.2015	
9.	Написання доповіді, створення слайдів.	15.05.2015	
10.	Захист курсової роботи	16.05.2015	

Студент _____ / Зав'ялов Д.С. /

(підпис)

(Прізвище, ініціали)

Науковий керівник _____ / Царик Т.І. /

(підпис)

(Прізвище, ініціали)

Завдання видане «03» Березня 2020 року

Вступ

Комп'ютерні ігри є невід'ємною частиною поп-культури, а їх створення потребує чималої кількості ресурсів. Історія їх зародження починається з середини 20-го сторіччя. Спершу, це були програми, як “Шахи” чи “Хрестики-нулики”, створені армією США, щоб навчити комп'ютер передбачати дії противника. Проте першою повноцінною відеогрою(та за сумісництвом першу багатокористувацьку) все ж прийнято вважати гру “Tennis For Two”, розроблену фізиком Віллі Хігінботемом, де на електронний дисплей виводилося зображення тенісного корту, сітки, м'яча та його траєкторії . Це дало поштовх в подальшому розвитку ігор. І на сьогоднішній день, індустрія відеоігор є однією з найприбутковіших.

Перед початком створення будь-якої гри, потрібно чітко спланувати безліч деталей. Спершу треба визначитись із жанром і сеттінгом(бо це напряму впливає на аудиторію вашої гри). Далі потрібно вибрати ігровий рушій. Потім придумати сюжет, і створити концепт-арти в залежності від обраного сеттінгу та сюжету. Все це потребує багато часу та енергії(особливо, коли ви створюєте гру самотужки, або в команді з декількох чоловік), і тому цьому треба приділити особливу увагу.

Темою даної роботи є створення гри “Вгадай слово”, в якій гравець муситиме написати слово з наданих йому для цього букв.

Розділ 1. Концепт гри.

Метою цієї роботи є створення гри “Вгадай слово”, в якій гравцеві потрібно буде скласти слова з букв, які він бачитиме на екрані. По ходу гри, гравець буде отримувати бали за кожне вгадане слово(без підказок).

Задачі цієї роботи:

- a) Виконати аналіз програм, для створення гри.
- b) Скласти структурну схему гри.
- c) Реалізувати гру у вигляді програмного продукту.
- d) Зробити висновки щодо розробленої програми.

1.1 Вибір програми чи ігрового рушія для створення гри

Як було описано у Вступі, вибір ігрового рушія є одним з основних і важливих моментів під час створення гри. На щастя, їх існує доволі велика кількість, тому вибрати є з чого. Найпопулярнішими рушіями є:

- Unreal Engine
- Unity
- GameMaker
- Godot
- CryEngine

Більшість з них є безкоштовними, або мають Trial-версію. Такі рушії як Unity та Godot Engine є одними з найкращим вибором для новачків, тому що мають доволі зрозумілий інтерфейс, і в інтернеті можна знайти безліч пакетів з матеріалами для створення гри.

Всі ці рушії в більшій чи меншій мірі потребують навичок програмування. Є звісно й такі, які не потребують жодних навичок (наприклад Construct 2), проте вони не надають таких гнучких інструментів які потрібні для розробки більш-менш серйозних проєктів.

Але так як завданням даної роботи є створення текстової гри, в якій не обов'язково використовувати графіку чи тим паче фізику, було вирішено зробити її за допомогою Windows Forms, так як в ньому є всі необхідні інструменти, для того щоб добре реалізувати поставлену задачу.

1.2 Використані джерела та загальна інформація

Під час виконання роботи, були використані такі джерела:

1) Руководство по программированию в Windows Forms

<https://metanit.com/sharp/windowsforms/>. Перевірено: 9.05.2020.

Для виконання цієї роботи, основ, які надає дане джерело було достатньо, тому інших джерел немає.

При першому запуску програми, перше що кидається в очі це початкова Форма. Вона виглядає як стандартне вікно Windows: має рядок заголовка, кнопки «Згорнути», «Розгорнути» та «Закрити». Над цією та іншими формами буде вестися робота під час виконання проєкту. Для того, щоб працювати з формою, потрібно щоб на ній були об'єкти. Вони знаходяться на панелі елементів Toolbox. Щоб відкрити цю панель, потрібно перейти в меню View і там обрати Toolbox(також можна скористатися комбінацією клавіш Ctrl+Alt+X). Ця панель містить в собі велику кількість вкладок та інструментів, але в цій роботі будуть використовуватися елементи з вкладки «Common Controls», а саме:

- Button – створює кнопку
- Label – дозволяє відображати звичайний текст на формі
- TextBox – текстове поле, яке дозволяє вводити та редагувати текст

Також можна змінювати властивості елементів. Для того, щоб викликати цю панель(якщо її нема за замовчування), треба перейти в меню View та обрати Properties Window. При виконанні роботи, основними зміненими властивостями будуть:

- BackColor – задає колір фону

- `BackgroundImage` – дозволяє встановити зображення як фон
- `Font` – задає шрифт
- `FormBorderStyle` – задає стилі границь форми
- `Text` – дозволяє змінювати текст
- `Size` – дозволяє змінювати розмір елементу/форми
- `StartPosition` – задає початкову позицію формі
- `MaximizeBox` – дозволяє приховувати/відображати кнопку «Розгорнути»
- `MinimizeBox` – дозволяє приховувати/відображати кнопку «Згорнути»
- `Location` – змінює розміщення елементу

Щоб помістити будь-який елемент на форму, потрібно клікнути по ньому в `Toolbox` і потім клікнути на формі. Для зміни його розміру найпростішими способами будуть просто розтягнути його, потягнувши за один з його кутів, або скористатися властивістю `Size` в панелі `Properties`. Всі зміни під час роботи програми будуть виконуватися за допомогою подій(`Events`). Вся робота програми базується на подіях. Наприклад, щоб задати подію для натискання кнопки, треба розмістити її на формі, і далі зробити по ній подвійний клік(так само можна робити з будь-якими елементами). Після цього відкриється вікно з кодом, в якому вже буде записана подія кліку по кнопці. Після цього треба буде записати саму подію, тобто те, що буде відбуватися при натисканні на кнопку під час виконання програми. Для цього треба написати код в середині фігурних дужок. В подіях також можна змінювати властивості елементів форми.

Наприклад, щоб змінити текст в `Label`, треба написати ім'я цього об'єкту(наприклад `label1`), і через крапку звернутися до його властивостей і присвоїти потрібне значення: `label1.Text = "Новий текст"`.

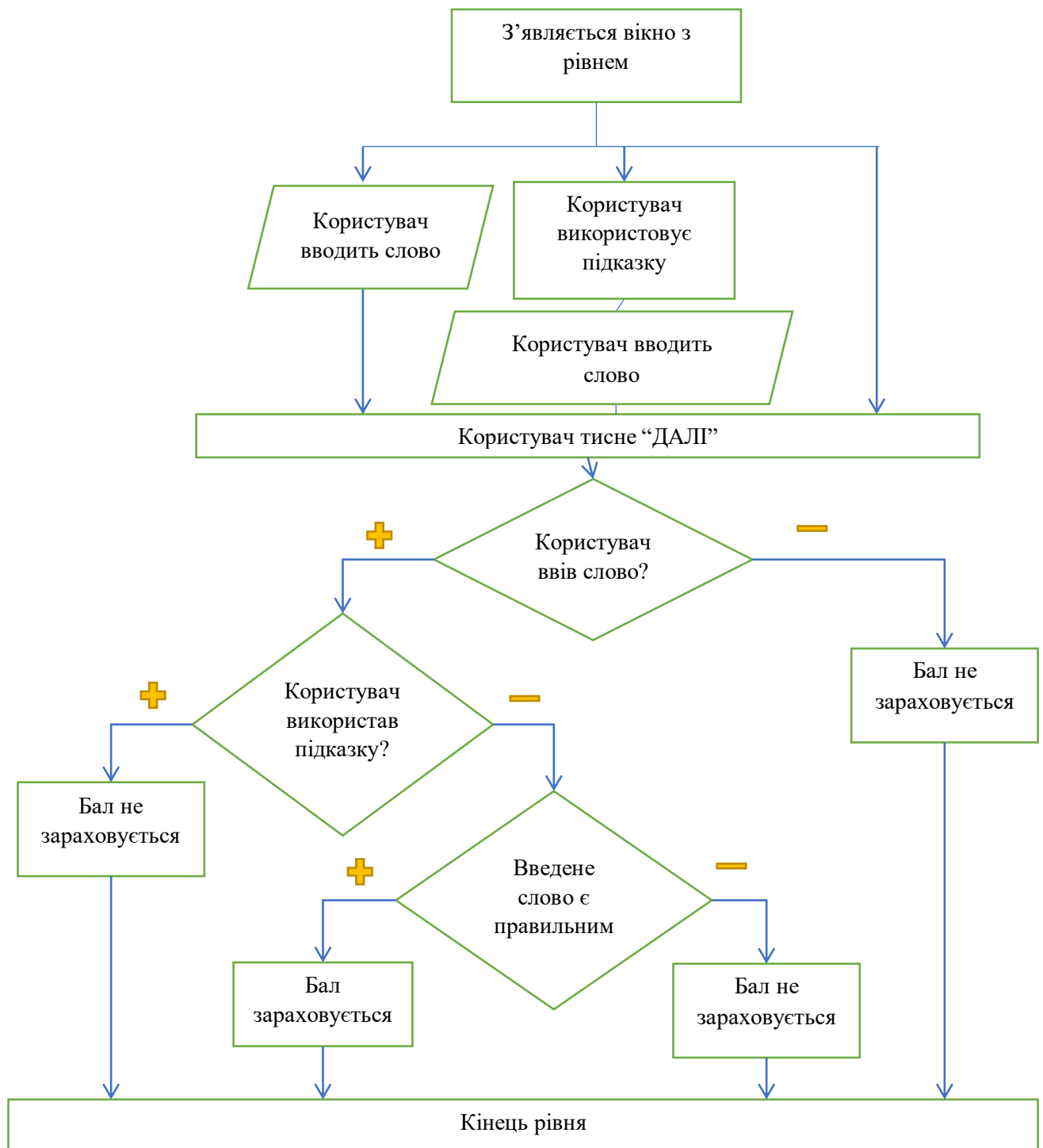
Розділ 2. Проектування продукту

Спочатку потрібно вирішити як саме буде працювати гра. Після запуску з'являється початкове вікно гри, в якому буде знаходитися кнопка початку гри, та деяка інформація. Вікна з'являтимуться в центрі екрану, їх не можна буде розтягувати, згортати та розгортати. Після того, як користувач розпочне гру, стартове вікно(вікно привітання) зникне, і з'явиться вікно з першим рівнем на ньому, як і на всіх наступних вікнах, будуть присутні наступні елементи:

- Назва вікна
- Кнопка «ЗАКРИТИ»
- Фонове зображення
- Напис «Вгадайте слово:»
- Букви, з яких потрібно скласти слово
- Текстове поле, в якому потрібно буде написати слово
- Кнопка переходу на наступний рівень
- Кнопка, яка показує слово

По ходу гри користувач муситиме складати слово з букв, і записувати його в текстове поле. Коли слово буде записане, і користувач натисне кнопку “ДАЛІ”, вікно, яке було запущене зникне(в незалежності від того чи правильне слово записав користувач), і на його місці з'явиться нове, вже з іншим словом. За кожне правильно вгадане слово, гравець отримує 1 бал. Також, якщо користувач не знає що за слово потрібно скласти, він може скористатися підказкою і натиснути на кнопку “ПОКАЗАТИ СЛОВО”. Правильне слово з'явиться нижче, але користувач бал за цей рівень не отримає. Після проходження гри, в кінці з'явиться повідомлення про закінчення гри, яке покаже скільки балів отримав гравець. Максимум можна отримати 5 балів.

Рівень буде проходити наступним чином:

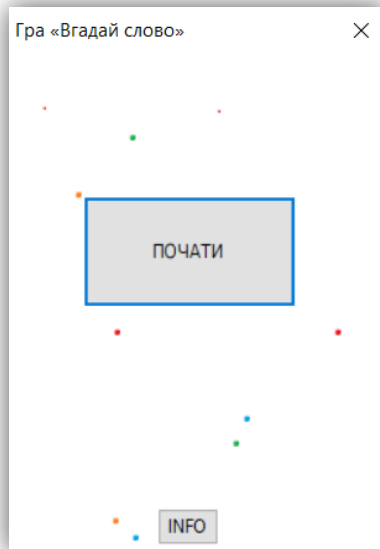


Такий алгоритм буде використовуватися у всіх рівнях, крім початкового вікна і останнього рівня.

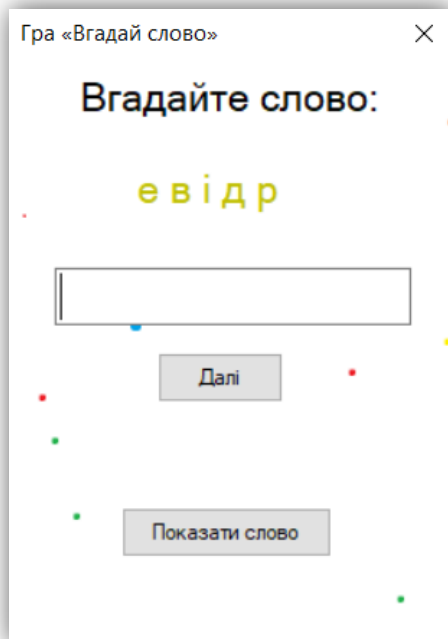
Розділ 3. Розробка гри.

3.1 Опис інтерфейсу

Інтерфейс програми буде представляти з себе вікно, розмірів 350;450 пікселів. Початкове вікно міститиме в собі кнопку початку гри та кнопку з інформацією.



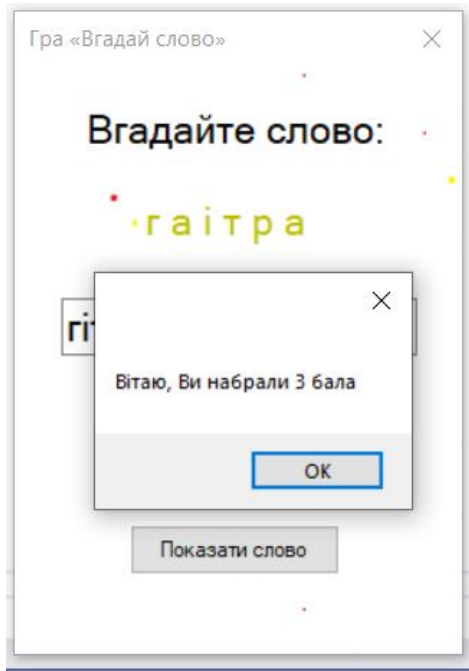
Після натискання на кнопку початку гри, з'явиться вікно з першим рівнем:



В текстове поле користувач вводитиме слово, і тиснутиме кнопку “Далі”.

Він може також натиснути на кнопку “Показати слово”, щоб подивитися відповідь.

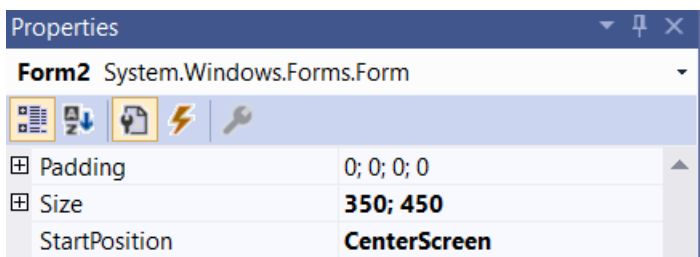
Після проходження гри, йому буде показане повідомлення:



Повідомлення в кінці можна викликати, використавши `MessageBox.Show()`.

3.2 Створення гри

Спочатку, змінимо розміри вікна. Для цього у властивостях вікна потрібно знайти параметр `Size` і змінити його.



В нашому випадку треба поставити значення 350;450. В цих же властивостях, одразу можна змінити наступні параметри:

- `StartPosition` – `CenterScreen`
- `MaximizeBox` – `False`
- `MinimizeBox` – `False`

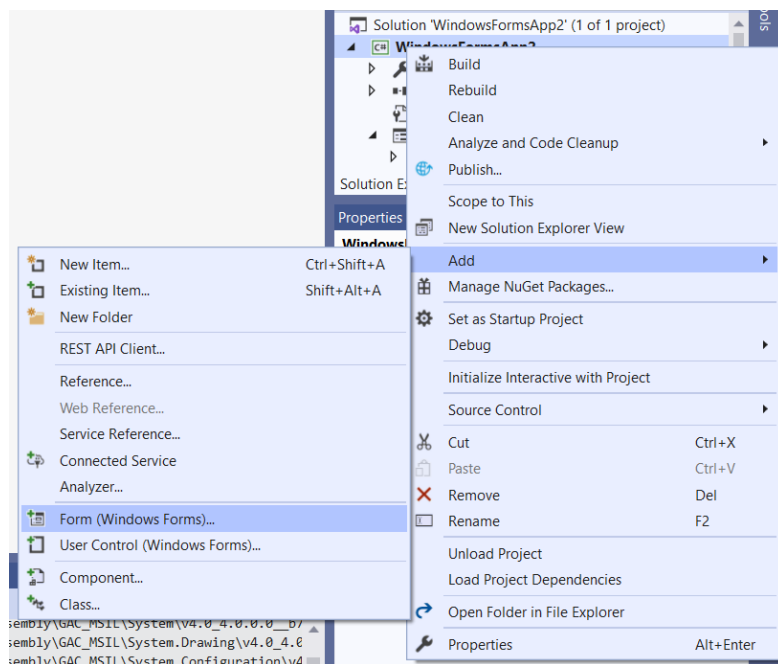
- Text - Гра «Вгадай слово»
- FormBorderStyle – FixedDialog
- BackColor – White
- BackGroundImage – обрати фонове зображення

Тепер вікно є налаштованим. Далі треба додати на нього необхідні елементи. Так як це меню гри, то в ньому буде міститися кнопка початку гри, і кнопка, при натиску на яку буде показана інформація про нарахування балів. Для цього потрібно викликати ToolBox, і в ньому знайти елемент Button. Розмістити на формі дві кнопки, і змінити їх текст на “ПОЧАТИ” та “INFO” відповідно. Тепер додамо подію натискання на кнопку “INFO”:

```
1 reference
private void button2_Click(object sender, EventArgs e)
{
    MessageBox.Show("За кожне вгадане слово Ви будете отримувати один бал. " +
        "За використання підказок бали не нараховуються.");
}
```

Під час натиску на кнопку, буде відображатися MessageBox з інформацією.

Тепер треба створити іншу форму, яку буде викликати початкова форма.



Перейменуємо нову форму в Form11. Далі створимо подію на кнопку “ПОЧАТИ”:

```

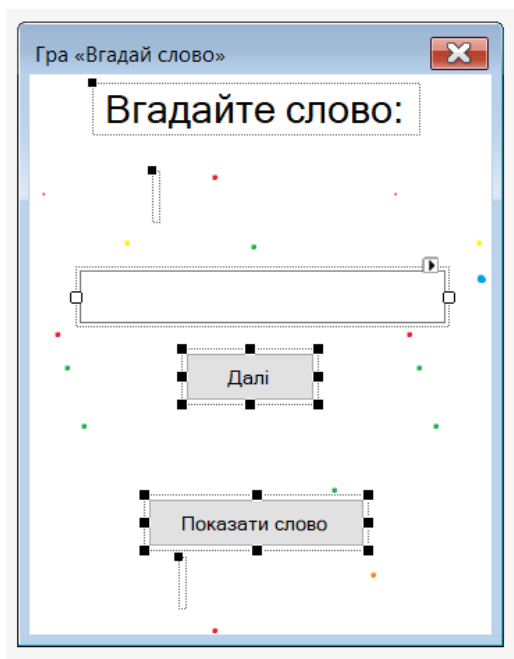
1 reference
private void button1_Click(object sender, EventArgs e)
{
    Form11 frm = new Form11();
    frm.Show();
    this.Hide();
}

```

Тут ми створюємо об'єкт форми першого рівня, запускаємо її, і ховаємо початкову форму.

Далі для першої форми(і для всіх наступних) задаємо ті самі параметри як і для початкової.

Тепер треба додати необхідні елементи на форму першого рівня. Спочатку додамо три елемента Label. У властивостях одного з них, в тексті треба написати: “Вгадайте слово:”. Другий Label треба розмістити трохи нижче першого, і у властивостях в полі Text стерти значення. Останній Label розмістимо в самому низу вікна, і так само видалимо значення. Тепер додамо дві кнопки. Перша буде знаходитися в центрі вікна, інша трохи нижче, над останнім Label. І наостанок залишилося додати TextBox. Розмістимо його над першою кнопкою. Ось що ми повинні отримати:



Тепер перейдемо до коду. Тиснемо F7. В середині класу записуємо масив із слів:

```
static string[] wordsArray = new string[5] { "гітара", "пальто", "двері", "вікно", "лампа" }; // масив слів
string[] newArr = new string[4];
```

І після цього створюємо новий масив, з кількістю елементів на 1 менше ніж в попередньому.

Тепер додамо ще декілька змінних:

```
int click = 0;
int score = 0;
int indx;
```

Змінна click буде перевіряти чи була натиснута кнопка, яка показує підказку.

Змінна score зберігає в собі кількість балів, які отримує гравець протягом гри.

Змінна indx – це номер елемента(слова) в масиві, яке потрібно вгадати.

Далі двічі клікаємо по формі, щоб записати подію, яка буде відбуватися при запуску цієї форми. Спочатку запишемо сюди наступне:

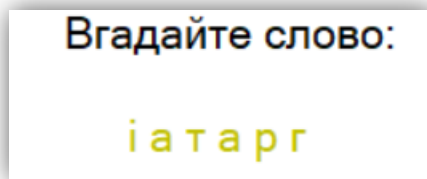
```
private void Form1_Load(object sender, EventArgs e)
{
    List<string> wordsLst = new List<string>(wordsArray); // Створюється список, в який входять е-ти масиву
    var rndm = new Random();

    indx = rndm.Next(0, wordsLst.Count);

    List<char> ltrsLst = new List<char>(wordsArray[indx]);
    var rnd = new Random();
    while (ltrsLst.Count > 0)
    {
        int index = rnd.Next(0, ltrsLst.Count);
        label2.Text += Convert.ToString(ltrsLst[index]) + " ";
        ltrsLst.RemoveAt(index);
    }
    wordsLst.RemoveAt(indx);
}
```

Спочатку ми з масиву wordsArray створюємо список wordsLst. Далі змінній indx присвоюємо порядковий номер довільного елемента(слова) зі списку wordsLst. Далі з літер, з яких складається це слово створюємо список ltrsLst. Потім беремо довільний елемент з ltrsLst, записуємо його в label2, і видаляємо його зі списку щоб він не повторювався. Це повторюється поки ми не переберемо всі букви в слові. Далі видаляємо це слово зі списку wordsLst.

Як результат ми отримуємо довільне слово зі списку, в якому букви розміщені в довільному порядку:



Після цього пишемо наступне:

```
int k = 0;
for (int i = 0; i < newArr.Length; i++)
{
    if (wordsArray[k] != wordsArray[indx])
    {
        newArr[i] = wordsArray[k];
        k++;
    }
    else
    {
        newArr[i] = wordsArray[k + 1];
        k += 2;
    }
}
```

Тут ми в новий масив записуємо всі елементи старого, крім того, який ми використали в цій формі.

Тепер запишемо подію для кнопки “Показати слово”:

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    label4.Text = wordsArray[indx];
    click = 1;
}
```

В label4 виводиться наше слово, а змінна click отримує значення 1.

Тепер залишилося записати подію для кнопки “Далі”:

```

1 reference
private void button2_Click(object sender, EventArgs e)
{
    if (textBox1.Text == wordsArray[indx])
    {
        if (click != 1)
        {
            score++;
        }
    }
    Form2 frm = new Form2(score);
    frm.newArr = newArr;
    frm.Show();
    this.Hide();
}

```

Спочатку перевіряється чи записане слово є правильним. Якщо так, то перевіряється чи значення `click = 1`. Якщо гравець не натискав на кнопку “Показати слово”, і значення `click` не дорівнює 1, то йому зараховується 1 бал.

Після цього створюється об'єкт наступної форми, в яку передається новий масив `newArr`, і значення `score`.

Створюємо так само наступну форму, створюємо ті ж самі змінні, тільки значенню `score` не присвоюємо нічого, залишаємо просто `int score`. Тепер:

```

1 reference
public Form2(int score1)
{
    score = score1;
    InitializeComponent();
}

```

Новому значенню `score` ми присвоюємо старе. Далі на треба прийняти значення масиву із попередньої форми, і встановити їх:

```

9 references
public string[] newArr { get; set; }
public string[] newArr2 = new string[3];

```

Після цього створюємо ще один масив, на один елемент менше, як і в попередній формі. Весь інший код залишається незмінним, тільки тепер ми в переписуємо елементи з масиву `newArr` в `newArr2`. Точно так само треба робити і з іншими формами, крім останньої. В останній ми не оголошуємо змінну `indx`, так як і ми знаємо, що порядковий номер елемента 0. Тому в подію запуску форми ми запишемо тільки:

```

1 reference
private void Form5_Load(object sender, EventArgs e)
{
    List<char> ltrslst = new List<char>(newArr4[0]);
    var rnd = new Random();
    while (ltrslst.Count > 0)
    {
        int index = rnd.Next(0, ltrslst.Count);
        label2.Text += Convert.ToString(ltrslst[index]) + " ";
        ltrslst.RemoveAt(index);
    }
}

```

Тепер для закінчення тесту:

```

private void button2_Click_1(object sender, EventArgs e)
{
    if (textBox1.Text == newArr4[0])
    {
        if (click != 1)
        {
            score++;
        }
    }

    string one = " бал";
    string two = " бала";
    string five = " балів";
    if (score == 1)
    {
        MessageBox.Show("Вітаю, Ви набрали" + score + one);
        Application.Exit();
    }
    else if (score >= 2 && score <= 4)
    {
        MessageBox.Show("Вітаю, Ви набрали " + score + two);
        Application.Exit();
    }
    else
    {
        MessageBox.Show("Вітаю, Ви набрали " + score + five);
        Application.Exit();
    }
}

```

Спочатку йде, як і раніше, перевірка на використання підказки. Далі створюються кінцівки для повідомлення про бали. Потім перевіряється скільки балів набрав гравець, і в залежності від кількості балів, буде виводитись повідомлення з різними кінцівками. Після цього програма закривається.

Висновки

Ця робота була виконана за допомогою інструментів, які пропонують середовище Windows Forms. Всі основні елементи гри розташовуються на Формах, з якими взаємодіє користувач. Протягом гри він набирає бали, про кількість яких повідомляється в кінці гри. Загалом, гра є готовою, але потребує кращого візуального оформлення. Також можна було б додати більшу кількість слів, і зробити різні рівні складності, які будуть відрізнятися кількістю букв у слові.

Використані джерела

1) Руководство по программированию в Windows Forms

<https://metanit.com/sharp/windowsforms/>. Перевірено: 9.05.2020.

Додатки