МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСТИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій
Кафедра програмного забезпечення

ЗВІТ

До лабораторної роботи № 4
На тему: «Компоненти Borland C++ Builder 6 для представлення даних»
З дисципліни: «Об'єктно-орієнтоване програмування»

Лектор:
доц. кафедри ПЗ
Коротєєва Т. О.

Виконав:
ст. гр. ПЗ-16
Чаус О. М.

Прийняв:
асист. кафедри ПЗ
Дивак І. В.

« ____ » _____ 2022 р.

∑= _____ .....:

Львів-2022

Тема роботи: Компоненти **Borland C++ Builder 6** для представлення даних.

Мета роботи: Створити віконний проект та продемонструвати використання компонент призначених для відображення та опрацювання даних.

Теоретичні відомості

Компонента **dataGridView**. Компонента **dataGridView** є таблицею, що містить рядки. Дані таблиці можуть бути лише для читання або такими, що можуть бути редаговані. Таблиця може мати смуги прокрутки, причому задане число перших рядків і стовпців може бути фіксованим і не прокручуватися. Таким чином, можна задати заголовки стовпців і рядків, що постійно присутні у вікні компоненти. Компонента призначена в першу чергу для представлення текстової інформації, однак є можливість відображати і графічну.

Властивості **Rows** і **Cells** визначають відповідно кількість стовпців і рядків. Колір фону фіксованих комірок задається властивістю **BackColor**, а колір тексту в **Value** – **ForeColor.**

Властивість **ReadOnly** визначає чи можна змінювати значення комірок**.**

Компонента **Timer** дозволяє вимірювати час. Властивість інтрвал визначає інтервал виміру в мілісекундах. Властивість **Start** і **Stop** запускають вимір і зупиняють відповідно.

Завдання для лабораторної роботи

1. Ознайомитись із компонентою **dataGridView**.
2. Реалізувати гру.

Код програми

```
Random rnd;
              int cellNum = 50;
              bool onBlack = false;
              bool redGenerated = false;
              bool startClicked = false;
              bool cellsGenerated = false;
              int score = 0;
              int bestScore = 0;

private: System::Void Game_Load(System::Object^ sender, System::EventArgs^ e) {
       for (int i = 0; i < 10; i++)
       {
              dataGridView1->Columns->Add("","");
              dataGridView1->Rows->Add();
       }
       dataGridView1->ClearSelection();
       for (int i = 0; i < 10; i++)
       {
              for (int j = 0; j < 10; j++)
              {
                     dataGridView1->Rows[i]->Cells[j]->Style->BackColor = Color::White;
              }
       }
}
private: System::Void buttonStart_Click(System::Object^ sender, System::EventArgs^ e) {
       int Column, Row;
       if (!startClicked)
       {
              onBlack = false;
              for (int i = 0; i < 10; i++)
              {
                     for (int j = 0; j < 10; j++)
                            dataGridView1->Rows[i]->Cells[j]->Style->BackColor =
Color::White;
              }
              Column = rnd.Next(0, 10);
              Row = rnd.Next(0, 10);
```

```
			dataGridView1->Rows[Column]->Cells[Row]->Style->BackColor = Color::Cyan;
			startClicked = true;
			timer1->Start();
		}
	}
	private: System::Void GameOver()
	{
		timer1->Stop();
		for (int i = 0; i < 10; i++)
		{
			for (int j = 0; j < 10; j++)
				dataGridView1->Rows[i]->Cells[j]->Style->BackColor = Color::Black;
		}
		cellNum = 50;
		startClicked = false;
		MessageBox::Show("You lost!\nFinal score: " + score);
		if (score > bestScore)
			bestScore = score;
		label1->Text = "Best score: " + bestScore;
		label2->Text = "Score: 0";
		score = 0;
	}
	private: System::Void timer1_Tick(System::Object^ sender, System::EventArgs^ e) {
		if (redGenerated)
		{
			for (int i = 0; i < 10; i++)
			{
				for (int j = 0; j < 10; j++)
				{
					if (dataGridView1->Rows[i]->Cells[j]->Style->BackColor !=
Color::Cyan)
					{
						dataGridView1->Rows[i]->Cells[j]->Style->BackColor =
Color::White;
					}
				}
			}
			onBlack = false;
			redGenerated = false;
		}
		if (!cellsGenerated)
		{
			for (int i = 0; i < cellNum; i++)
			{
				int Column = rnd.Next(0, 10);
				int Row = rnd.Next(0, 10);
				if (dataGridView1->Rows[Row]->Cells[Column]->Style->BackColor ==
Color::Cyan)
				{
					i--;
				}
				else dataGridView1->Rows[Row]->Cells[Column]->Style->BackColor =
Color::Black;
			}
			if (cellNum != 1)
			{
				cellNum /= 2;
			}
			if(cellNum == 1)
				cellNum = 1 + rnd.Next(0, 4);
			cellsGenerated = true;
		} else
		{
			for (int i = 0; i < 10; i++)
			{
				for (int j = 0; j < 10; j++)
				{
```

```
                        if (dataGridView1->Rows[i]->Cells[j]->Style->BackColor ==
Color::White)
                        {
                                dataGridView1->Rows[i]->Cells[j]->Style->BackColor =
Color::Red;
                        }
                }
        }
        cellsGenerated = false;
        redGenerated = true;
        if (!onBlack)
        {
                GameOver();
        }
        else
        {
                score++;
                label2->Text = "Score: " + score;
        }
    }
}
private: System::Void buttonStart_KeyDown(System::Object^ sender,
System::Windows::Forms::KeyEventArgs^ e)
{
    int index;
    bool stop = false;
    switch (e->KeyCode)
    {
    case Keys::S:
        for (int i = 0; i < 10; i++)
        {
                for (int j = 0; j < 10; j++)
                {
                        if (dataGridView1->Rows[i]->Cells[j]->Style->BackColor ==
Color::Cyan)
                        {
                                if (i == 9)
                                        index = 0;
                                else index = i + 1;
                                if (dataGridView1->Rows[index]->Cells[j]->Style->BackColor
== Color::Red)
                                {
                                        GameOver();
                                        stop = true;
                                        break;
                                }
                                if (onBlack)
                                {
                                        dataGridView1->Rows[i]->Cells[j]->Style->BackColor
= Color::Black;
                                }
                                else dataGridView1->Rows[i]->Cells[j]->Style->BackColor =
Color::White;
                                if (dataGridView1->Rows[index]->Cells[j]->Style->BackColor
== Color::Black)
                                {
                                        onBlack = true;
                                }
                                else onBlack = false;
                                dataGridView1->Rows[index]->Cells[j]->Style->BackColor =
Color::Cyan;
                                stop = true;
                                break;
                        }
                }
                if (stop)
                {
                        stop = false;
```

```
                        break;
                    }
                }
                break;
        case Keys::D:
                for (int i = 0; i < 10; i++)
                {
                    for (int j = 0; j < 10; j++)
                    {
                        if (dataGridView1->Rows[i]->Cells[j]->Style->BackColor ==
Color::Cyan)
                        {
                            if (j == 9)
                                index = 0;
                            else
                                index = j + 1;
                            if (dataGridView1->Rows[i]->Cells[index]->Style->BackColor
== Color::Red)
                            {
                                GameOver();
                                stop = true;
                                break;
                            }
                            if (onBlack)
                            {
                                dataGridView1->Rows[i]->Cells[j]->Style->BackColor
= Color::Black;
                            }
                            else dataGridView1->Rows[i]->Cells[j]->Style->BackColor =
Color::White;
                            if (dataGridView1->Rows[i]->Cells[index]->Style->BackColor
== Color::Black)
                            {
                                onBlack = true;
                            }
                            else onBlack = false;
                            dataGridView1->Rows[i]->Cells[index]->Style->BackColor =
Color::Cyan;
                            stop = true;
                            break;
                        }
                    }
                    if (stop)
                    {
                        stop = false;
                        break;
                    }
                }
                break;
        case Keys::W:
                for (int i = 0; i < 10; i++)
                {
                    for (int j = 0; j < 10; j++)
                    {
                        if (dataGridView1->Rows[i]->Cells[j]->Style->BackColor ==
Color::Cyan)
                        {
                            if (i == 0)
                                index = 9;
                            else index = i - 1;
                            if (dataGridView1->Rows[index]->Cells[j]->Style->BackColor
== Color::Red)
                            {
                                GameOver();
                                stop = true;
                                break;
                            }
                            if (onBlack)
```

```
                                {
                                        dataGridView1->Rows[i]->Cells[j]->Style->BackColor
= Color::Black;
                                }
                                else dataGridView1->Rows[i]->Cells[j]->Style->BackColor =
Color::White;
                                if (dataGridView1->Rows[index]->Cells[j]->Style->BackColor
== Color::Black)
                                {
                                        onBlack = true;
                                }
                                else onBlack = false;
                                dataGridView1->Rows[index]->Cells[j]->Style->BackColor =
Color::Cyan;
                                stop = true;
                                break;
                        }
                }
                if (stop)
                {
                        stop = false;
                        break;
                }
        }
        break;
    case Keys::A:
        for (int i = 0; i < 10; i++)
        {
                for (int j = 0; j < 10; j++)
                {
                        if (dataGridView1->Rows[i]->Cells[j]->Style->BackColor ==
Color::Cyan)
                        {
                                if (j == 0)
                                        index = 9;
                                else index = j - 1;
                                if (dataGridView1->Rows[i]->Cells[index]->Style->BackColor
== Color::Red)
                                {
                                        GameOver();
                                        stop = true;
                                        break;
                                }
                                if (onBlack)
                                {
                                        dataGridView1->Rows[i]->Cells[j]->Style->BackColor
= Color::Black;
                                }
                                else dataGridView1->Rows[i]->Cells[j]->Style->BackColor =
Color::White;
                                if (dataGridView1->Rows[i]->Cells[index]->Style->BackColor
== Color::Black)
                                {
                                        onBlack = true;
                                }
                                else onBlack = false;
                                dataGridView1->Rows[i]->Cells[index]->Style->BackColor =
Color::Cyan;
                                stop = true;
                                break;
                        }
                }
                if (stop)
                {
                        stop = false;
                        break;
                }
        }
```

```
            break;
        }
}
};
}
```

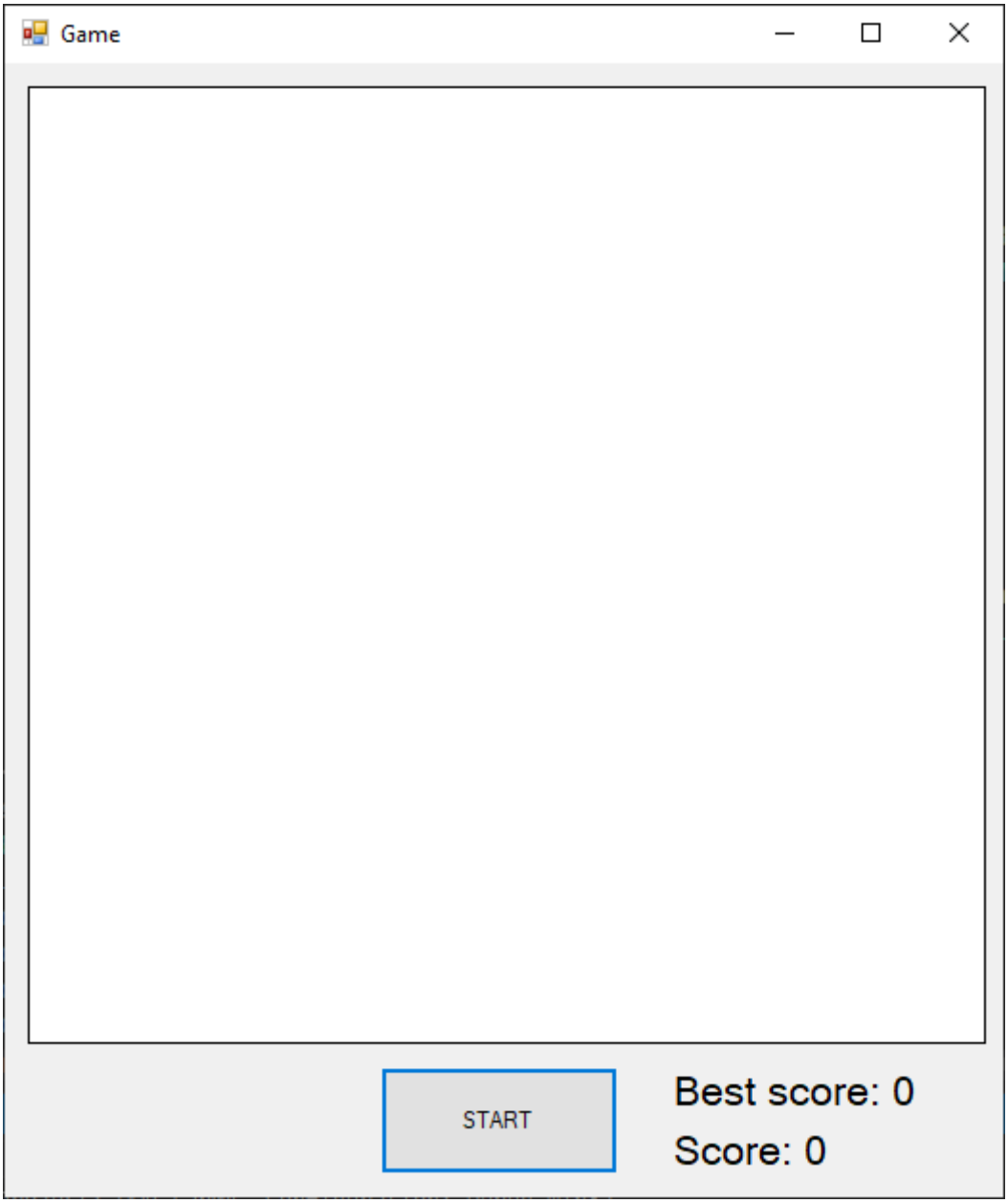

Рис. 1. Стартовий екран

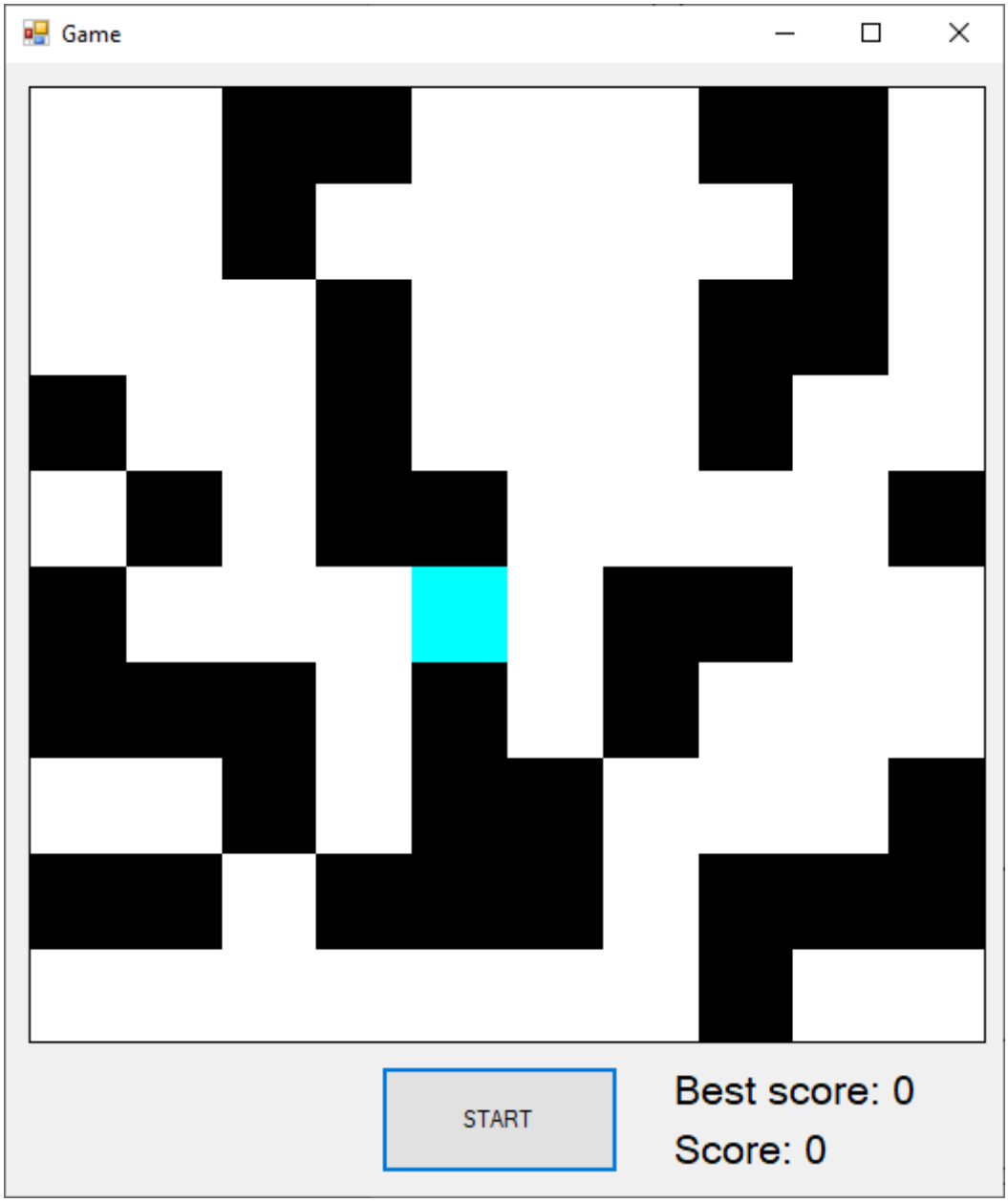

Рис. 2. Геймплей

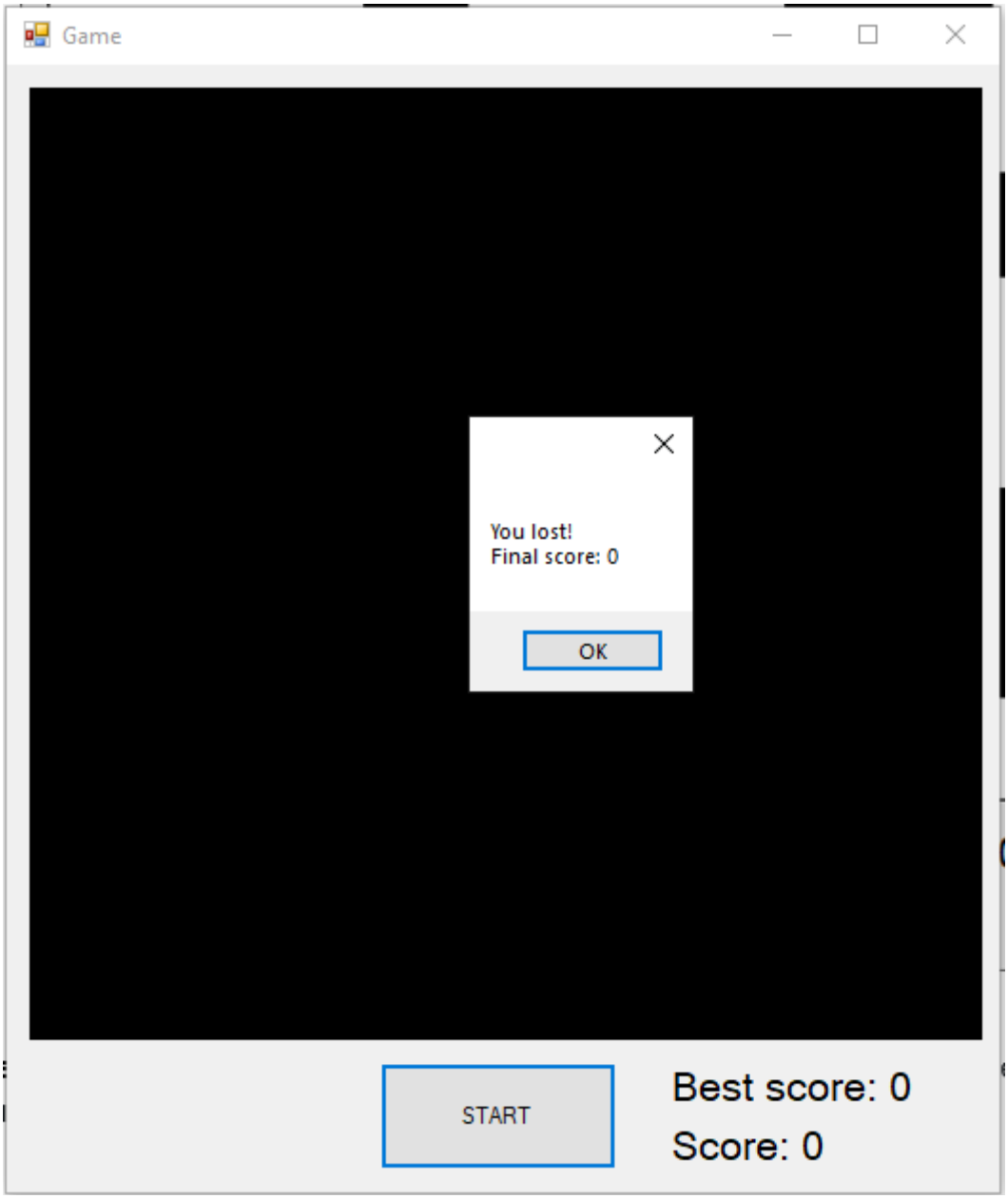

Рис. 3. Екран програшу

Висновок: створив віконний проект та продемонстрував використання компонент призначених для відображення та опрацювання даних.