

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«Розробка ігрових та мультимедійних систем»

Лабораторна робота № 1

на тему:

Лабораторна робота №1
**«Розробка ігрового проекту з використанням засобів та технологій
програмування»**

Виконав:	Сініло О. А.	Перевірила:	Меркулова К. В.
Група	ІПЗ-44	Дата перевірки	
Форма навчання	денна	Бали	
Спеціальність	121		

Мета роботи: освоїти основні прийоми та етапи розробки ігрових систем для ОС Windows. Лабораторна робота орієнтована на ознайомлення студентів з основними елементами середовищ програмування ігрових систем для ОС Windows та створення елементарної гри.

Задача: Розробити сюжет гри. Узгодити з викладачем. Реалізувати розроблену гру, описати в звіті всі етапи. Наявність звукового супроводу гри є обов'язковою.

Робоче завдання: Виконання роботи передбачає наступну послідовність дій:

1. Узгодження ігри з викладачем.
2. Розробка сюжету гри.
3. Реалізація розробленої гри.
4. Оформлення результатів та звіту

Завдання:

Розробка гри «Динозавр» (також відома як Chrome Dino). Гравець веде піксельного Тиранозавра Рекса по ландшафту, що прокручується збоку, уникаючи перешкод, щоб отримати вищий бал. Завершенням гри є зіткнення моделі динозавра із перешкодою.

Сюжет:

Гра «Динозавр» — це гра, розроблена Google і вбудована у веб-переглядач Google Chrome. Гру створили учасники команди Chrome UX у 2014 році.

Під час гри Самотній Т-Рекс безперервно рухається зліва направо по чорно-білому пустельному ландшафту, при цьому гравець намагається уникати зустрічних перешкод, таких як кактуси, стрибаючи. Натискання пробілу призведе до того, що динозавр «стрибне». По ходу гри швидкість гри поступово зростає, доки користувач не натрапить на перешкоду, що призведе до миттєвого завершення гри.

Опис алгоритму реалізації:

Код розроблений на C# у вигляді Windows форми. Програма складається з форми, на якій знаходяться перешкоди, динозавр та підлога, а також з наступних методів:

- Подія натискання клавіші. При натисканню пробілу обробляється фізика стрибка, а при завершенні гри і натисканню клавіші R рестарт гри.
- Подія відпускання клавіші. При відпусканню клавіші R відбувається рестарт гри. Також якщо динозавр стрибав, то включення логіки падіння.
- Кожні 20 мілісекунд виконання обробки поточного стану гри, тобто обробка максимального стрибка, зміна позиції перешкод, перевірка зштовхування динозавра з перешкодою.
- Метод скидання поточних даних до початкових значень.

Лістинг програми з коментарями:

```
using System;
using System.Media;
using System.Windows.Forms;

namespace T_Rex_Runner_Game
{
    public partial class Form1 : Form
    {
        bool jumping = false; // boolean to check if player is jumping or not
        int jumpSpeed; // integer to set jump speed
        int force = 12; // force of the jump in an integer
        int score = 0; // default score integer set to 0
        int obstacleSpeed = 10; // the default speed for the obstacles
        Random rand = new Random(); // create a new random class
        bool isGameOver = false; // boolean to check is game over
        int topPosition; // integer to set trex initial top position
        SoundPlayer scoreSound; // class for score sounds
        SoundPlayer gameOverSound; // class for game over sound
        public Form1()
        {
            InitializeComponent();
            topPosition = floor.Top - trex.Height; // set trex initial top position
            scoreSound = new SoundPlayer(T_Rex_Runner_Game.Properties.Resources.score);
            gameOverSound = new SoundPlayer(T_Rex_Runner_Game.Properties.Resources.game_over);

            resetGame(); // run the reset game function
        }

        private void gameEvent(object sender, EventArgs e)
        {
            trex.Top += jumpSpeed; // move trex upper when jumping

            scoreText.Text = "Score: " + score;

            // reset jumping
            if (jumping == true && force < 0)
            {
                jumping = false;
            }
        }
    }
}
```

```

// gradually decrease jump force speed
if (jumping == true)
{
    jumpSpeed = -12;
    force -= 1;
}
else
{
    jumpSpeed = 12;
}

// reset jump and position params
if (trex.Top > topPosition - 1 && jumping == false)
{
    force = 12;
    trex.Top = topPosition;
    jumpSpeed = 0;
}

foreach (Control x in this.Controls)
{
    if (x is PictureBox && (string)x.Tag == "obstacle")
    {
        x.Left -= obstacleSpeed;

        // set new position for disappeared obstacle
        if (x.Left < -100)
        {
            x.Left = this.ClientSize.Width + rand.Next(300, 500) + (x.Width * 15);
            score++;
            scoreSound.Play();
        }

        // make game over when intersect with obstacle
        if (trex.Bounds.Intersects(x.Bounds))
        {
            gameTimer.Stop();
            trex.Image = Properties.Resources.dead;
            scoreText.Text += " Press R to restart the game!";
            isGameOver = true;
            gameOverSound.Play();
        }
    }
}

// gradually increase game difficulty with increasing score
if (score > 5)
{
    obstacleSpeed = 12;
} else if (score > 10)
{
    obstacleSpeed = 15;
}

}

private void keyisup(object sender, KeyEventArgs e)
{
    // if the R key is pressed and released then we run the reset function
    if (e.KeyCode == Keys.R && isGameOver == true)
    {
        resetGame();
    }

    //when the keys are released we check if jumping is true
    // if so we need to set it back to false so the player can jump again

```

```

        if (jumping)
        {
            jumping = false;
        }
    }

    private void keyisdown(object sender, KeyEventArgs e)
    {
        //if the player pressed the space key and jumping boolean is false
        // then we set jumping to true
        if (e.KeyCode == Keys.Space && !jumping)
        {
            jumping = true;
        }
    }

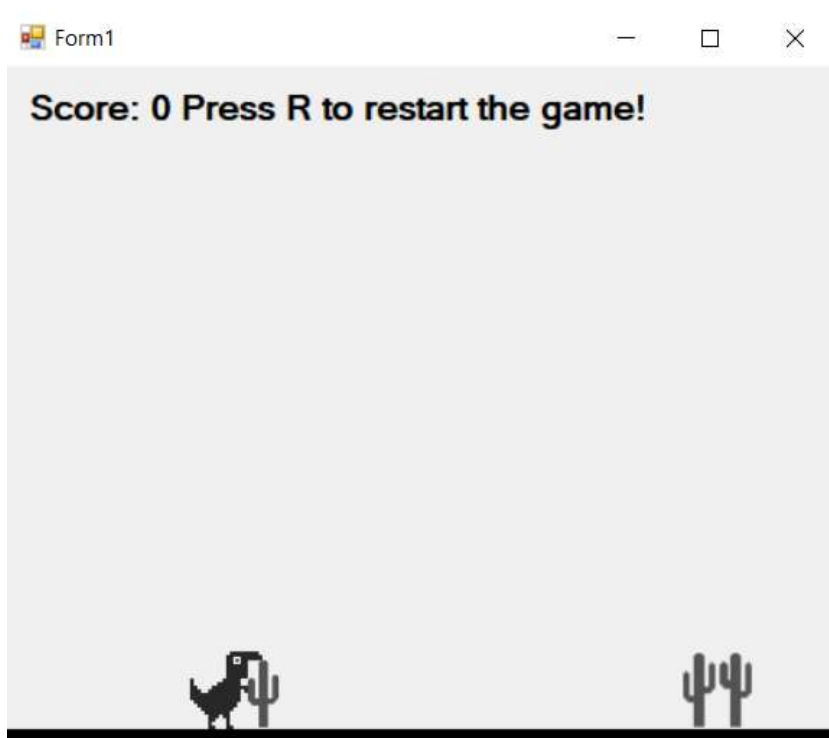
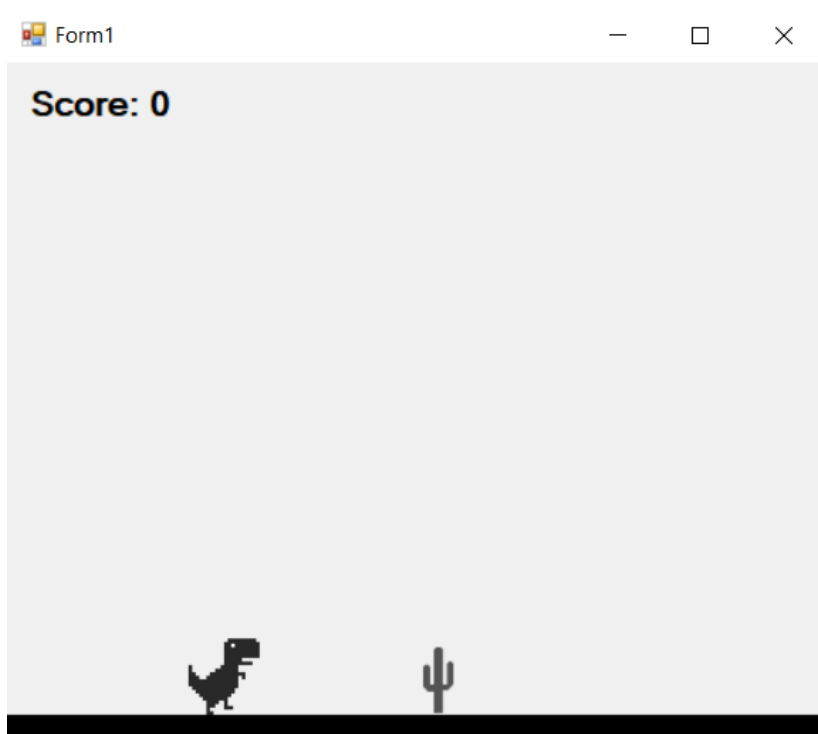
    public void resetGame()
    {
        // This is the reset function
        force = 12; // set the force to 12
        jumpSpeed = 0; // set the jump speed to 0
        jumping = false; // change jumping to false
        score = 0; // set score to 0
        obstacleSpeed = 10; // set obstacle speed back to 10
        scoreText.Text = "Score: " + score; // change the score text to just show the
score
        trex.Image = Properties.Resources.running; // change the t rex image to running
        isGameOver = false;
        trex.Top = topPosition;

        foreach (Control x in this.Controls)
        {
            // is X is a picture box and it has a tag of obstacle
            if (x is PictureBox && x.Tag == "obstacle")
            {
                // generate a random number in the position integer between 600 and 1000
                // change the obstacles left position to a random location beginning of the
game
                x.Left = this.ClientSize.Width + rand.Next(500, 800) + (x.Width * 10);
            }
        }

        gameTimer.Start(); // start the timer
    }
}

```

Скріншоти інтерфейсу:



Висновок:

Отже, в даній лабораторній роботі було виконано поставлене завдання. Окрім цього було прийнято рішення щодо технологій розробки та алгоритму створення модуля гри. При ході виконання роботи було вивчено основні прийоми та етапи розробки ігрових систем. Було покращено навички використання

середовища розробки ігрових систем – VS, збережено та завантажено проект розробки на гіт.