A black and white logo

Description automatically generated

Nour yasser 212001212

Shahd reda 211010168

x/o game

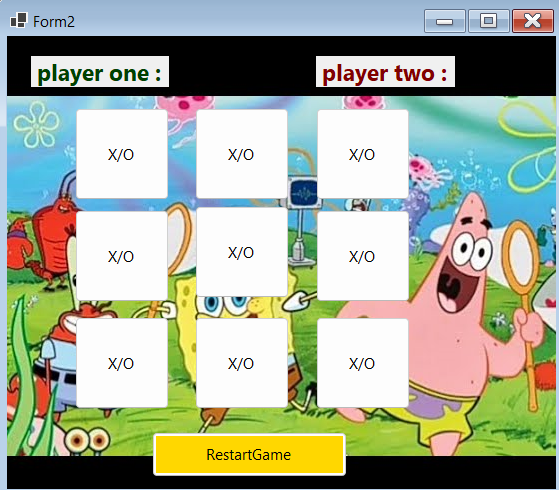
c sharp (c#)

Overview:

The project is a Windows Forms application implemented in C# that allows users to play x/o game against a computer opponent or with another player. The game features two modes: a single-player mode against a computer opponent (CPU) and a two-player mode for local multiplayer.

Game Mode :

Two player mode (form2)



Single player mode (form1)

A screenshot of a game

Description automatically generated

Score tracking :

If the player x or o win the score increase and display message box which one win the game

Restart game :

Can player restart game again

Write/ Read from file :

Code write and read from file is called output.txt

First check if the file exit or not then the game start consider we are in mode cpu message box will appear there is file to written score and create is done when game finish if the player is win , the file will update the player is win and increase count and increase high score and the same thing on cpu and anther mode local player

If there problem in file will appear message box there is an error at file to exit

A screenshot of a computer error

Description automatically generated

A screenshot of a game

Description automatically generated

A screenshot of a computer error

Description automatically generated

A screenshot of a computer

Description automatically generated

The code :

Form1 ( cpu )

namespace WinFormsApp1

{

public partial class Form1 : Form

{

public enum player

{

x, o

}

player currentplayer;

Random Random = new Random();

int cpuwincount = 0;

int playerwincount = 0;

List<Button> buttons;

public Form1()

{

InitializeComponent();

Restartgame();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

private void cpumove(object sender, EventArgs e)

{

if (buttons.Count > 0)

{

int index = Random.Next(buttons.Count);

Button selectedButton = buttons[index];

selectedButton.Enabled = false;

currentplayer = player.o;

selectedButton.Text = currentplayer.ToString();

selectedButton.BackColor = Color.DarkSalmon;

buttons.RemoveAt(index);

cheackgame();

cputimer.Stop();

}

}

private void playerclickbutton(object sender, EventArgs e)

{

var button = (Button)sender;

currentplayer = player.x;

button.Text = currentplayer.ToString();

button.Enabled = false;

button.BackColor = Color.Cyan;

buttons.Remove(button);

cheackgame();

cputimer.Start();

}

private void Restgame(object sender, EventArgs e)

{

Restartgame();

}

private void cheackgame()

{

// Winning conditions for Player X

if ((button1.Text == "x" && button2.Text == "x" && button3.Text == "x") ||

(button4.Text == "x" && button5.Text == "x" && button6.Text == "x") ||

(button7.Text == "x" && button8.Text == "x" && button9.Text == "x") ||

(button1.Text == "x" && button6.Text == "x" && button9.Text == "x") ||

(button2.Text == "x" && button5.Text == "x" && button8.Text == "x") ||

(button3.Text == "x" && button4.Text == "x" && button7.Text == "x") ||

(button1.Text == "x" && button5.Text == "x" && button7.Text == "x") ||

(button3.Text == "x" && button5.Text == "x" && button9.Text == "x"))

{

cputimer.Stop();

MessageBox.Show("Player Wins!", "Game Over");

playerwincount++;

label1.Text = "Player Wins: " + playerwincount;

Restartgame();

return;

}

// Winning conditions for CPU (Player O)

if ((button1.Text == "o" && button2.Text == "o" && button3.Text == "o") ||

(button4.Text == "o" && button5.Text == "o" && button6.Text == "o") ||

(button7.Text == "o" && button8.Text == "o" && button9.Text == "o") ||

(button1.Text == "o" && button6.Text == "o" && button9.Text == "o") ||

(button2.Text == "o" && button5.Text == "o" && button8.Text == "o") ||

(button3.Text == "o" && button4.Text == "o" && button7.Text == "o") ||

(button1.Text == "o" && button5.Text == "o" && button7.Text == "o") ||

(button3.Text == "o" && button5.Text == "o" && button9.Text == "o"))

{

cputimer.Stop();

MessageBox.Show("CPU Wins!", "Game Over");

cpuwincount++;

label2.Text = "CPU Wins: " + cpuwincount;

Restartgame();

return;

}

if (buttons.All(b => !b.Enabled))

{

cputimer.Stop();

MessageBox.Show("It's a Draw!", "Game Over");

Restartgame();

}

}

private void Restartgame()

{

buttons = new List<Button> { button1, button2, button3, button4, button5, button6, button7, button8, button9 };

foreach (Button x in buttons)

{

x.Enabled = true;

x.Text = "X/O";

x.BackColor = DefaultBackColor;

}

currentplayer = player.x;

}

private void NAV3(object sender, EventArgs e)

{

this.Hide();

Form3 f3 = new Form3();

f3.Show();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void UpdateHighScore()

{

string path = "output.txt";

if (File.Exists(path))

{

try

{

string[] lines = File.ReadAllLines(path);

foreach (string line in lines)

{

if (line.StartsWith("Player Wins:"))

{

string winsString = line.Substring("Player Wins:".Length);

if (int.TryParse(winsString, out int playerWins))

{

label3.Text = "High Score: " + playerWins;

break;

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Error reading file: " + ex.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void label3\_Click(object sender, EventArgs e)

{

UpdateHighScore();

}

}

}

}

}

Form2 (two player )

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WinFormsApp1

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

resetgame();

}

public enum player

{

X, O

}

player CURRENTPLAYER;

int playerone = 0;

int playertwo = 0;

List<Button> buttons;

private void CheckGame()

{

if (check\_game(player.X))

{

MessageBox.Show("Player\_X\_win", "Dark\_Warrior");

playerone++;

label1.Text = "Playerone Wins: " + playerone;

resetgame();

}

else if (check\_game(player.O))

{

MessageBox.Show("Player\_O\_win", "Dark\_Warrior");

playertwo++;

label2.Text = "Playertwo Wins: " + playertwo;

resetgame();

}

else if (buttons.Count == 0)

{

MessageBox.Show("It's a draw!", "Game Over");

resetgame();

}

}

private bool check\_game(player CURRENTPLAYER)

{

if (button1.Text == "X" && button2.Text == "X" && button3.Text == "X" ||

button4.Text == "X" && button5.Text == "X" && button6.Text == "X" ||

button7.Text == "X" && button8.Text == "X" && button9.Text == "X" ||

button1.Text == "X" && button6.Text == "X" && button9.Text == "X" ||

button2.Text == "X" && button5.Text == "X" && button8.Text == "X" ||

button3.Text == "X" && button4.Text == "X" && button7.Text == "X" ||

button1.Text == "X" && button5.Text == "X" && button7.Text == "X" ||

button3.Text == "X" && button5.Text == "X" && button9.Text == "X")

{

return CURRENTPLAYER == player.X;

}

else if (button1.Text == "O" && button2.Text == "O" && button3.Text ==

"O" ||

button4.Text == "O" && button5.Text == "O" && button6.Text == "O" ||

button7.Text == "O" && button8.Text == "O" && button9.Text == "O" ||

button1.Text == "O" && button6.Text == "O" && button9.Text == "O" ||

button2.Text == "O" && button5.Text == "O" && button8.Text == "O" ||

button3.Text == "O" && button4.Text == "O" && button7.Text == "O" ||

button1.Text == "O" && button5.Text == "O" && button7.Text == "O" ||

button3.Text == "O" && button5.Text == "O" && button9.Text == "O")

{

return CURRENTPLAYER == player.O;

}

return false;

}

private void resetgame()

{

buttons = new List<Button>

{ button1, button2, button3 , button4, button5,

button6, button7, button8, button9 };

foreach (Button b in buttons)

{

b.Enabled = true;

b.Text = "X/O";

b.BackColor = DefaultBackColor;

}

}

private void resetgame(object sender, EventArgs e)

{

resetgame();

}

private void chooseone(object sender, EventArgs e)

{

var button = (Button)sender;

button.Enabled = false;

if (CURRENTPLAYER == player.X)

{

button.Text = player.X.ToString();

button.BackColor = Color.Cyan;

}

else

{

button.Text = player.O.ToString();

button.BackColor = Color.Pink;

}

buttons.Remove(button);

CheckGame();

SwitchPlayer();

}

private void SwitchPlayer()

{

CURRENTPLAYER = (CURRENTPLAYER == player.X) ? player.O : player.X;

}

private void Form2\_Load(object sender, EventArgs e)

{

}

private void nav4(object sender, EventArgs e)

{

this.Hide();

Form3 f3 = new Form3();

f3.Show();

}

private void writeinfile()

{

string path = "output.txt";

int highScore = Math.Max(playerone, playertwo);

using (StreamWriter writer = new StreamWriter(path))

{

writer.WriteLine("Player one Wins: " + playerone);

writer.WriteLine("Player two Wins: " + playertwo);

writer.WriteLine("High Score: " + highScore); // Write the high score

}

MessageBox.Show("Wins and High Score have been written to " + path, "Success",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void UpdateHighScore()

{

string path = "output.txt";

if (File.Exists(path))

{

try

{

string[] lines = File.ReadAllLines(path);

foreach (string line in lines)

{

if (line.StartsWith("High Score:"))

{

string highScoreString = line.Substring("High Score:".Length).Trim();

if (int.TryParse(highScoreString, out int highScore))

{

label3.Text = "High Score: " + highScore;

return; // Exit after finding the high score

}

}

}

}

catch (Exception ex)

{

MessageBox.Show("Error reading file: " + ex.Message, "Error",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void label3\_Click(object sender, EventArgs e)

{

writeinfile();

UpdateHighScore();

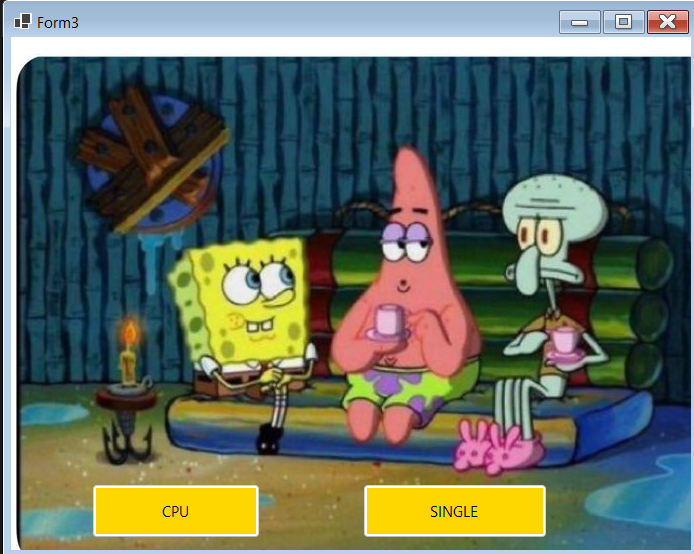
} }

}

Form3(run code)

Choose CPU if you want form1 to Play with CPU

Choose single if you want form2 to player with anther player



using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WinFormsApp1

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

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

{

Button clickedButton = sender as Button;

if (clickedButton == SINGLE)

{

this.Hide();

Form2 f2 = new Form2();

f2.Show();

}

else if (clickedButton == CPU)

{

this.Hide();

Form1 f1 = new Form1();

f1.Show();

}

}

}

}

Main :

namespace WinFormsApp1

{

internal static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form3());

}

}

}

Anther screenshots :



A screenshot of a computer game

Description automatically generated

A screenshot of a game

Description automatically generated