Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра ЕОМ



**Звіт**

З лабораторної роботи №3

З дисципліни: «Автоматизоване проектування комп’ютерних систем»

На тему: «Створення клієнта і сервера»

Виконав:

ст.гр. КІ-403

Соболь К.Д.

Перевірив:

Кіцера А.О.

**Львів 2024**

**Хід роботи**

Написав скетч для Arduino UNO:

#include <ArduinoJson.h>

enum Move { Rock, Paper, Scissors };

// Easy mode

Move getStupidMove(Move playerMove) {

  switch (playerMove) {

    case Rock: return Scissors;

    case Paper: return Rock;

    case Scissors: return Paper;

    default: return getRandomMove();

  }

}

// Normal move

Move getRandomMove() {

  int randMove = random(0, 3);  // Випадковий вибір між 0, 1 і 2

  return static\_cast<Move>(randMove);

}

// Impossible mode

Move getCounterMove(Move playerMove) {

  switch (playerMove) {

    case Rock: return Paper;

    case Paper: return Scissors;

    case Scissors: return Rock;

    default: return getRandomMove();

  }

}

// Convert move to string

String getMoveString(Move move) {

  switch (move) {

    case Rock: return "Rock";

    case Paper: return "Paper";

    case Scissors: return "Scissors";

    default: return "Unknown";

  }

}

// Convert string to Move enum

Move getMoveFromString(String move) {

  move.trim();

  if (move == "Rock") return Rock;

  if (move == "Paper") return Paper;

  if (move == "Scissors") return Scissors;

  return Rock;

}

void setup() {

  Serial.begin(9600);

  randomSeed(analogRead(0));

  delay(2000);

}

void loop() {

  if (Serial.available()) {

    String message = Serial.readStringUntil('\n');

    message.trim();

    int spaceIndex = message.indexOf(' ');

    if (spaceIndex != -1) {

      String playerMove = message.substring(0, spaceIndex);

      String difficulty = message.substring(spaceIndex + 1);

      Move player = getMoveFromString(playerMove);

      Move arduinoMove;

      if (difficulty == "Easy") {

        arduinoMove = getStupidMove(player);

      }

      else if (difficulty == "Normal") {

        arduinoMove = getRandomMove();

      }

      else if (difficulty == "Impossible") {

        arduinoMove = getCounterMove(player);

      } else {

        arduinoMove = getRandomMove();

      }

      String arduinoMoveString = getMoveString(arduinoMove);

      // Create JSON

      StaticJsonDocument<200> doc;

      doc["player\_move"] = playerMove;

      doc["arduino\_move"] = arduinoMoveString;

      doc["difficulty"] = difficulty;

      String output;

      serializeJson(doc, output);

      Serial.println(output);

    }

  }

}

Зробив дизайн вікна за допомогою мови розмітки XAML:  
<Window x:Class="ArduinoClient.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="RockPaperScissors" WindowState="Maximized" Height="1080" Width="1920" Background="#FFF5E6">

<Grid>

<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center" Margin="10">

<TextBlock x:Name="GameName" Text="Rock Paper Scissors" HorizontalAlignment="Center" FontSize="50" Visibility="Visible" Margin="0 0 0 50"/>

<TextBlock x:Name="CreateBy" Text="Created by Sobol K." HorizontalAlignment="Center" FontSize="30" Visibility="Visible" Margin="0 0 0 50"/>

<Button x:Name="EasyButton" Content="Easy mode" Click="Easy\_Mode\_Click" Visibility="Visible" Margin="10" Style="{StaticResource PixelButtonStyle}"/>

<Button x:Name="NormalButton" Content="Normal mode" Click="Normal\_Mode\_Click" Visibility="Visible" Margin="10" Style="{StaticResource PixelButtonStyle}"/>

<Button x:Name="ImpossibleButton" Content="Impossible mode" Click="Impossible\_Mode\_Click" Visibility="Visible" Margin="10" Style="{StaticResource PixelButtonStyle}"/>

<TextBlock x:Name="ChoiseText" Text="Make Your Choise:" HorizontalAlignment="Center" FontSize="30" Visibility="Collapsed"></TextBlock>

<StackPanel x:Name="MakeChoise" Orientation="Horizontal" Visibility="Visible">

<Button x:Name="Rock\_Button" Content="Rock" Click="ChoiseRock\_Click" Visibility="Collapsed" Margin=" 20" Style="{StaticResource PixelButtonStyle}"/>

<Button x:Name="Paper\_Button" Content="Paper" Click="ChoisePaper\_Click" Visibility="Collapsed" Margin=" 20" Style="{StaticResource PixelButtonStyle}"/>

<Button x:Name="Scissors\_Button" Content="Scissors" Click="ChoiseScissors\_Click" Visibility="Collapsed" Margin=" 20" Style="{StaticResource PixelButtonStyle}"/>

<StackPanel x:Name="GameOver\_Menu" Orientation="Vertical" Visibility="Collapsed" Margin="10">

<TextBlock Text="Result Of Game:" HorizontalAlignment="Center" FontSize="30"></TextBlock>

<TextBlock x:Name="TextAferGame" Text=" " HorizontalAlignment="Center" FontSize="25" Margin=" 0 10 0 0" ></TextBlock>

<TextBlock Text="Your answer was:" HorizontalAlignment="Center" FontSize="30" Margin=" 0 30 0 0"></TextBlock>

<TextBlock x:Name="Choise" Text=" " HorizontalAlignment="Center" FontSize="25" Margin=" 0 10 0 0"></TextBlock>

<TextBlock Text="AI answer was:" HorizontalAlignment="Center" FontSize="30" Margin=" 0 30 0 0"></TextBlock>

<TextBlock x:Name="Responce" Text=" " HorizontalAlignment="Center" FontSize="25" Margin=" 0 10 0 0"></TextBlock>

<Button x:Name="OneMoreGame\_Button" Content="One More Time" Click="OneMoreGame\_Click" Margin=" 20" Style="{StaticResource PixelButtonStyle}"/>

<Button x:Name="ChangeDif\_Button" Content="Change Difficulty" Click="ChangeDif\_Click" Margin=" 20" Style="{StaticResource PixelButtonStyle}"/>

</StackPanel>

</StackPanel>

</StackPanel>

</Grid>

</Window>

Написав логіку для вікна:  
using System;

using System.IO;

using System.IO.Ports;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Media;

using Newtonsoft.Json.Linq;

namespace ArduinoClient

{

public partial class MainWindow : Window

{

private SerialPort serialPort;

public string difficulty = "";

public MainWindow()

{

InitializeComponent();

InitializeSerialPort();

}

private void InitializeSerialPort()

{

serialPort = new SerialPort("COM5", 9600)

{

NewLine = "\r\n"

};

serialPort.Open();

}

private void Easy\_Mode\_Click(object sender, RoutedEventArgs e)

{

MakeDifButtonsCollapsed();

difficulty = "Easy";

}

private void Normal\_Mode\_Click(object sender, RoutedEventArgs e)

{

MakeDifButtonsCollapsed();

difficulty = "Normal";

}

private void Impossible\_Mode\_Click(object sender, RoutedEventArgs e)

{

MakeDifButtonsCollapsed();

difficulty = "Impossible";

}

private async void ChoiseRock\_Click(object sender, RoutedEventArgs e)

{

await SendMessageAsync("Rock", difficulty);

}

private async void ChoisePaper\_Click(object sender, RoutedEventArgs e)

{

await SendMessageAsync("Paper", difficulty);

}

private async void ChoiseScissors\_Click(object sender, RoutedEventArgs e)

{

await SendMessageAsync("Scissors", difficulty);

}

private async Task SendMessageAsync(string message, string dif)

{

try

{

if (serialPort != null && serialPort.IsOpen)

{

serialPort.WriteLine(message + " " + dif);

}

string response = await Task.Run(() => serialPort.ReadLine());

JObject jsonResponse = JObject.Parse(response);

string playerMove = jsonResponse["player\_move"].ToString();

string arduinoMove = jsonResponse["arduino\_move"].ToString();

string difficulty = jsonResponse["difficulty"].ToString();

ChoiseText.Visibility = Visibility.Collapsed;

Rock\_Button.Visibility = Visibility.Collapsed;

Paper\_Button.Visibility = Visibility.Collapsed;

Scissors\_Button.Visibility = Visibility.Collapsed;

GameOver\_Menu.Visibility = Visibility.Visible;

if (arduinoMove == "Rock")

{

if (message == "Paper")

{

TextAferGame.Text = "You won!";

TextAferGame.Foreground = Brushes.Green;

}

else if (message == "Scissors")

{

TextAferGame.Text = "You lose!";

TextAferGame.Foreground = Brushes.Red;

}

else

{

TextAferGame.Text = "Draw!";

TextAferGame.Foreground = Brushes.Black;

}

Responce.Text = arduinoMove;

Choise.Text = message;

}

else if (arduinoMove == "Paper")

{

if (message == "Paper")

{

TextAferGame.Text = "Draw!";

TextAferGame.Foreground = Brushes.Black;

}

else if (message == "Scissors")

{

TextAferGame.Text = "You won!";

TextAferGame.Foreground = Brushes.Green;

}

else

{

TextAferGame.Text = "You lose!";

TextAferGame.Foreground = Brushes.Red;

}

Responce.Text = arduinoMove;

Choise.Text = message;

}

else if (arduinoMove == "Scissors")

{

if (message == "Paper")

{

TextAferGame.Text = "You lose!";

TextAferGame.Foreground = Brushes.Red;

}

else if (message == "Scissors")

{

TextAferGame.Text = "Draw!";

TextAferGame.Foreground = Brushes.Black;

}

else

{

TextAferGame.Text = "You won!";

TextAferGame.Foreground = Brushes.Green;

}

Responce.Text = arduinoMove;

Choise.Text = message;

}

}

catch (TimeoutException)

{

MessageBox.Show("No response from Arduino.", "Timeout", MessageBoxButton.OK, MessageBoxImage.Warning);

}

catch (IOException ex)

{

MessageBox.Show($"Communication error: {ex.Message}", "Error", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void OneMoreGame\_Click(object sender, RoutedEventArgs e)

{

GameOver\_Menu.Visibility = Visibility.Collapsed;

MakeDifButtonsCollapsed();

}

private void ChangeDif\_Click(object sender, RoutedEventArgs e)

{

GameOver\_Menu.Visibility = Visibility.Collapsed;

MakeDifButtonsVisible();

}

private void MakeDifButtonsCollapsed()

{

EasyButton.Visibility = Visibility.Collapsed;

NormalButton.Visibility = Visibility.Collapsed;

ImpossibleButton.Visibility = Visibility.Collapsed;

GameName.Visibility = Visibility.Collapsed;

CreateBy.Visibility = Visibility.Collapsed;

ChoiseText.Visibility = Visibility.Visible;

Rock\_Button.Visibility = Visibility.Visible;

Paper\_Button.Visibility = Visibility.Visible;

Scissors\_Button.Visibility = Visibility.Visible;

}

private void MakeDifButtonsVisible()

{

GameName.Visibility = Visibility.Visible;

CreateBy.Visibility = Visibility.Visible;

EasyButton.Visibility = Visibility.Visible;

NormalButton.Visibility = Visibility.Visible;

ImpossibleButton.Visibility = Visibility.Visible;

ChoiseText.Visibility = Visibility.Collapsed;

Rock\_Button.Visibility = Visibility.Collapsed;

Paper\_Button.Visibility = Visibility.Collapsed;

Scissors\_Button.Visibility = Visibility.Collapsed;

}

private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)

{

if (serialPort != null && serialPort.IsOpen)

{

serialPort.Close();

}

}

}

}

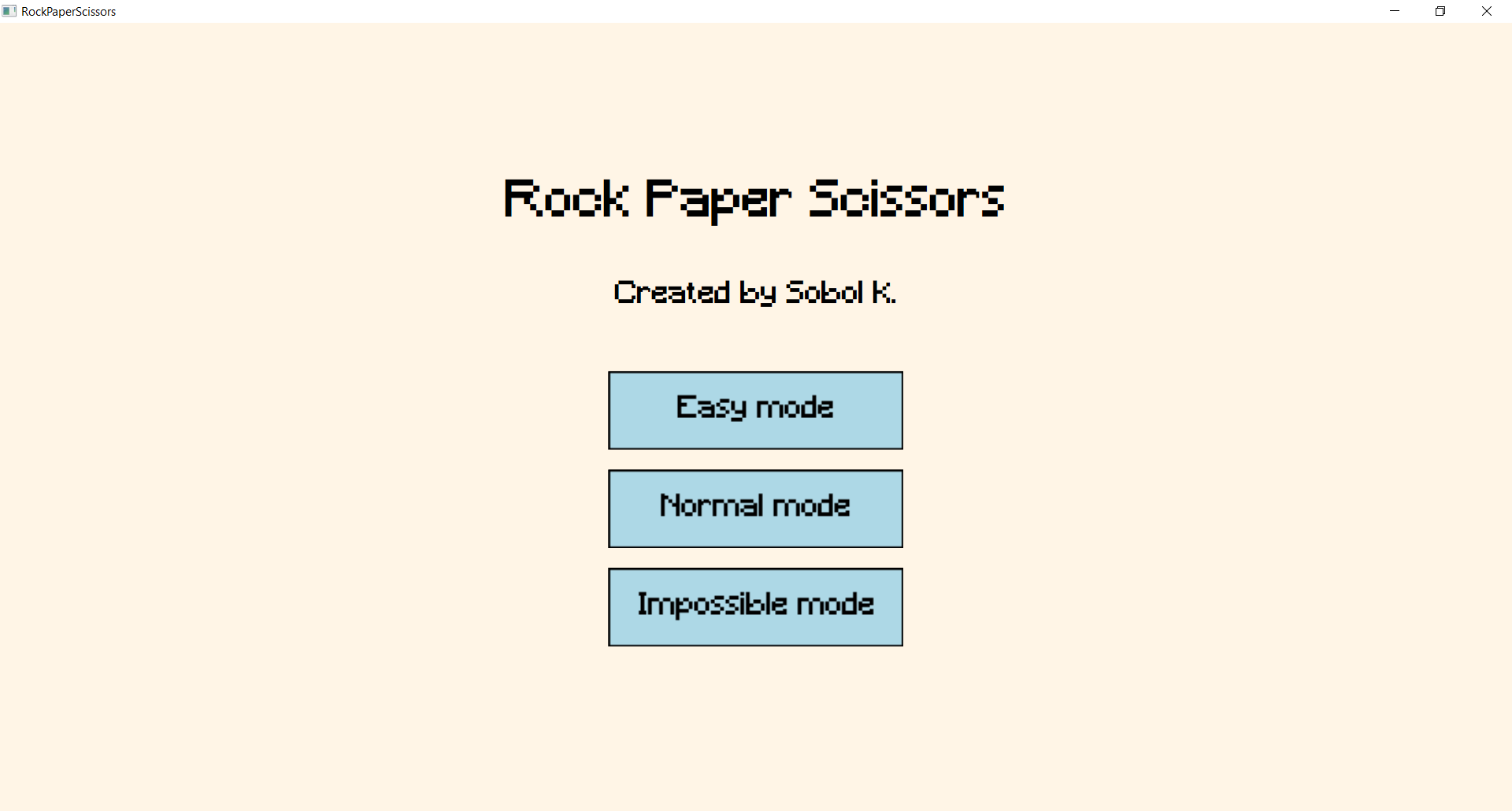


Рис. 1 Меню гри

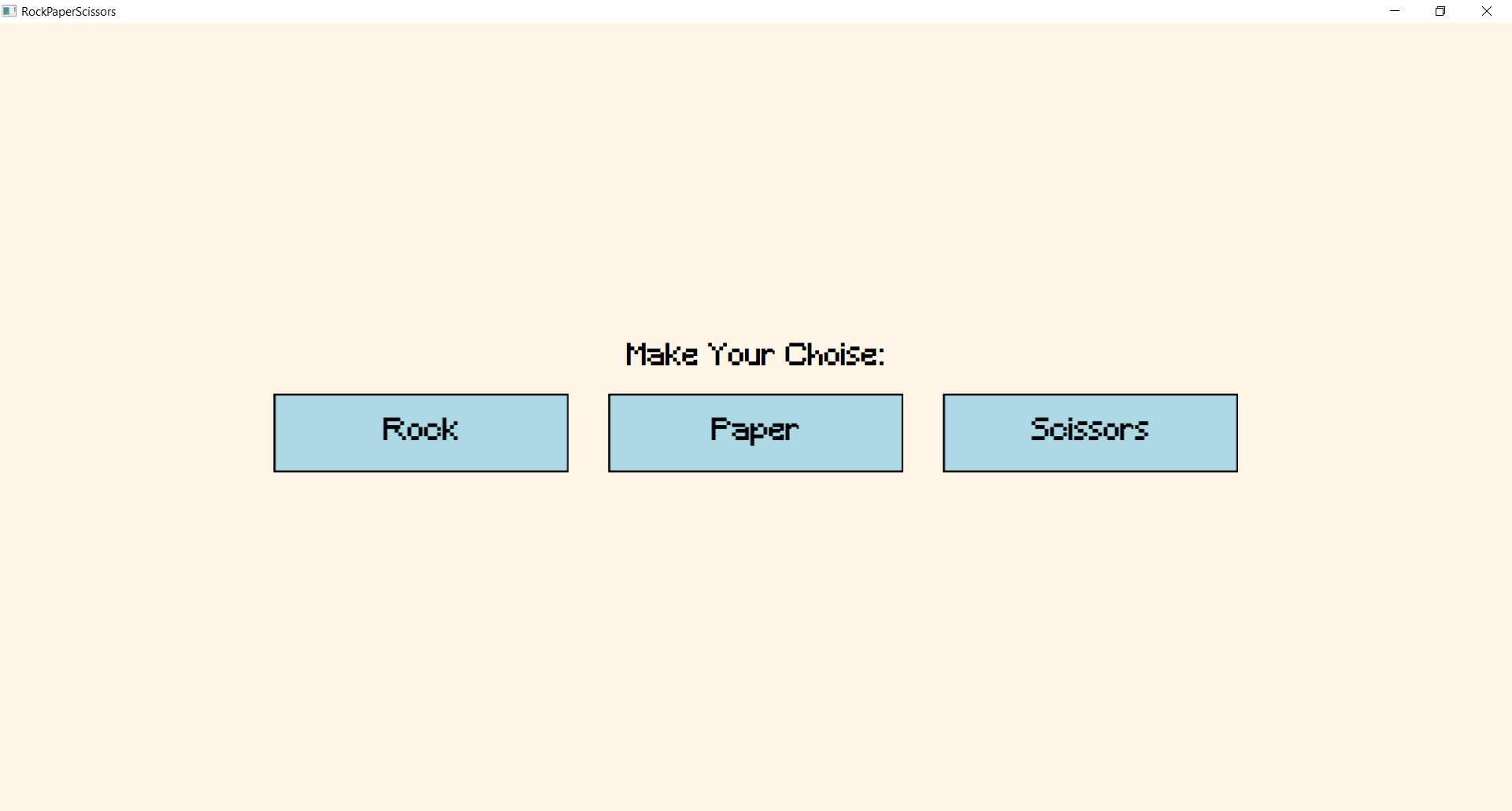


Рис. Вибір ходу гравця

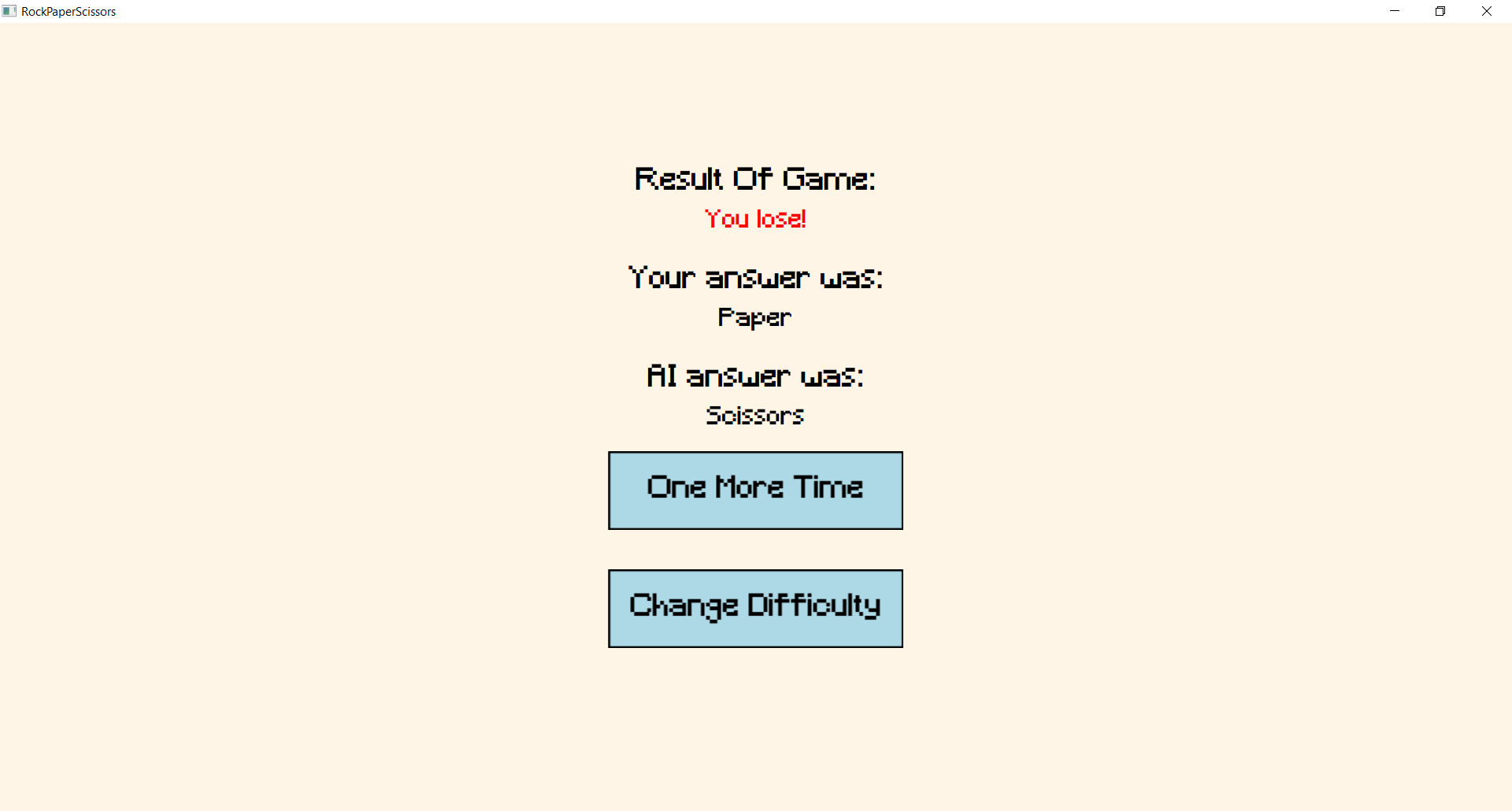


Рис.3 Вікно результату гри

Написав yml файл:  
  
name: Build, Test, and Deploy

on:

  push:

    branches:

      - develop

      - Task3

  pull\_request:

    branches:

      - develop

      - Task3

jobs:

  build:

    runs-on: windows-latest

    steps:

    # 1. Checkout репозиторій

    - name: Checkout repository

      uses: actions/checkout@v2

    # 2. Встановлення .NET SDK

    - name: Set up .NET SDK

      uses: actions/setup-dotnet@v2

      with:

        dotnet-version: '8.0'

    # 3. Завантаження Arduino IDE

    - name: Download Arduino IDE

      run: |

        curl -fsSL https://downloads.arduino.cc/arduino-ide/arduino-ide\_latest\_Windows\_64bit.zip -o arduino-ide.zip

        tar -xf arduino-ide.zip

        mv arduino-ide\*/ C:/ArduinoIDE

    # 4. Тестування C# проекту

    - name: Restore dependencies

      run: dotnet restore ArduinoClient/ArduinoClient.sln

    - name: Build project

      run: dotnet build ArduinoClient/ArduinoClient.sln --configuration Release

    - name: Run tests

      run: dotnet test ArduinoClient/ArduinoClient.sln --configuration Release --logger "trx"

    # 5. Публікація артефактів

    - name: Publish build

      run: dotnet publish ArduinoClient/ArduinoClient.sln --configuration Release --output ./publish

    - name: Upload artifacts

      uses: actions/upload-artifact@v3

      with:

        name: binaries

        path: ./publish

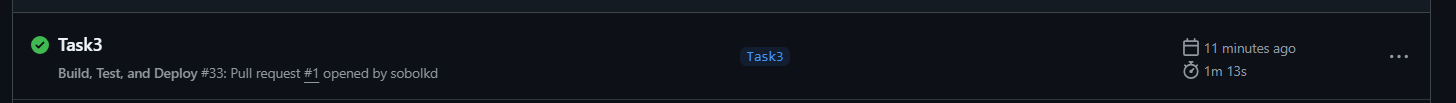
    - name: Upload test reports

      uses: actions/upload-artifact@v3

      with:

        name: test-reports

        path: ./TestResults/\*.trx



Висновок: Я створив клієнт і сервер використовуючи ARDUINO\_IDE та C#.