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;

using System.IO;

using cLibrary;

namespace PVM7\_6615

{

public partial class Form1 : Form

{

Random r = new Random();

Timer tgame = new Timer();

int score = 0, time = 180;

int benar = 0, salah = 0;

List<word> words;

List<int> highscore = new List<int>();

public Form1()

{

InitializeComponent();

this.Text = "Word Finder";

this.MaximizeBox = false;

words = new List<word>();

tgame.Interval = 1000;

tgame.Tick += Tgame\_Tick;

setResource();

worddDGV.Columns.Add("", "Kata");

}

private void setResource()

{

updateScore();

updateTimer();

}

private void updateScore()

{

scoreLbl.Text = score.ToString();

}

private void updateTimer()

{

timerLbl.Text = time.ToString();

}

private void createBoard()

{

int rlocation = r.Next(1, 3);

if (gameDGV.Columns.Count < 10)

{

for (int i = 0; i < 10; i++)

{

gameDGV.Columns.Add("", "Col" + (i + 1));

gameDGV.Rows.Add();

}

int rand = r.Next(1, 3);

if (rand == 1) putVerti();

else putHori();

fillCells();

gameDGV.CellDoubleClick += GameDGV\_CellDoubleClick;

}

else MessageBox.Show("Kolom sudah 10!");

}

char[] fill = new char[] {'a','b','c','d','e'};

private void fillCells()

{

for (int i = 0; i < 10; i++)

{

for (int j = 0; j < 10; j++)

{

int rhuruf = r.Next(5);

if (gameDGV.Rows[i].Cells[j].Value == null) gameDGV.Rows[i].Cells[j].Value = fill[rhuruf];

}

}

}

private void putVerti()

{

for (int i = 0; i < words.Count; i++)

{

for (int j = 0; j < words[i].Splitted.Count; j++)

{

gameDGV.Rows[j].Cells[i].Value = words[i].Splitted[j];

}

}

}

private void putHori()

{

for (int i = 0; i < words.Count; i++)

{

for (int j = 0; j < words[i].Splitted.Count; j++)

{

gameDGV.Rows[i].Cells[j].Value = words[i].Splitted[j];

}

}

}

int[] temp;

private void Tgame\_Tick(object sender, EventArgs e)

{

time--;

updateTimer();

if (time == 0)

{

tgame.Stop();

MessageBox.Show("Jumlah Kata Benar: " + benar + "\n" +

"Jumlah Kata Salah: " + salah + "\n" +

"Score: " + score);

highscore.Add(score);

StreamReader sr = new StreamReader("highscore.txt");

while (!sr.EndOfStream)

{

string s = sr.ReadLine();

int val = int.Parse(s);

highscore.Add(val);

}

sr.Close();

temp = highscore.ToArray();

Array.Sort(temp);

Array.Reverse(temp);

StreamWriter sw = new StreamWriter("highscore.txt");

if (highscore.Count > 0)

{

for (int i = 0; i < temp.Length; i++)

{

sw.WriteLine(temp[i].ToString());

}

}

sw.Close();

}

}

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

{

string kata = wordText.Text;

if (words.Count < 9)

{

word wtemp = new word(kata);

words.Add(wtemp);

worddDGV.Rows.Add(wtemp);

}

else MessageBox.Show("Kata sudah 10!");

}

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

{

if (words.Count < 9) MessageBox.Show("Kata kurang dari 10!");

else

{

for (int i = 0; i < words.Count; i++) words[i].split();

MessageBox.Show("Untuk bermain pilihlah huruf secara urut, sesuai dari soal");

tgame.Start();

createBoard();

}

}

List<char> check = new List<char>();

private void GameDGV\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

char huruf = (char)gameDGV.Rows[e.RowIndex].Cells[e.ColumnIndex].Value;

for (int i = 0; i < words.Count; i++)

{

for (int j = 0; j < words[i].Splitted.Count; j++)

{

if (huruf == words[i].Splitted[j])

{

check.Add(huruf);

gameDGV.Rows[e.RowIndex].Cells[e.ColumnIndex].Style.BackColor = Color.Green;

benar++;

break;

}

else

{

check.Remove(huruf);

gameDGV.Rows[e.RowIndex].Cells[e.ColumnIndex].Style.BackColor = Color.Red;

salah++;

}

}

break;

}

char[] temp = check.ToArray();

string s = new string(temp);

for (int i = 0; i < words.Count; i++)

{

if ((s == words[i].ToString() && s.Length == words[i].ToString().Length) &&

worddDGV.Rows[i].Cells[0].Style.BackColor != Color.Green)

{

check.Clear();

worddDGV.Rows[i].Cells[0].Style.BackColor = Color.Green;

score += 100 \* s.Length;

break;

}

}

updateScore();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Drawing;

namespace cLibrary

{

public class word

{

private string kata;

private List<char> splitted= new List<char>();

public word(string kata)

{

this.kata = kata;

}

public void split()

{

for (int i = 0; i < this.kata.Length; i++)

{

this.splitted.Add(this.kata[i]);

}

}

public override string ToString()

{

return this.kata;

}

public List<char> Splitted

{

set { this.splitted = value; }

get { return this.splitted; }

}

}

}