

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;

namespace Morse\_Code\_Translator

{

public partial class morseTranslationForm : Form

{

public morseTranslationForm()

{

InitializeComponent();

}

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

{

this.Close();

}

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

{

rawTextBox.Text = "";

rawTextBox.Focus();

translatedTextBox.Text = "";

translatedLabel.Visible = false;

translatedTextBox.Visible = false;

}

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

{

//take the user's input and determine whether there are any invalid characters

//if there are any then we will instruct the user to delete the character.

//use the IsletterOrDigit for each character as well as the isPunctuation but

//set up the valid chars as ?., and the IsWhiteSpace BUT after trimming

try

{

translatedLabel.Visible = true; //show the label describing the translation

translatedTextBox.Visible = true; //show the label that will hold the translation

string userString = rawTextBox.Text; //set the user's input to a variable

string translation = ""; //new string to set the translation

char[] letters = {' ','a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0', '.', ',', '?','!' };

string[] morseLetters = {"|",".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..", ".---", "-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", "...", "-", "..-", "...-", ".--", "-..-", "-.--", "--..", ".----", "..---", "...--", "....-", ".....", "-....", "--...", "---..", "----.", "-----", ".-.-.-", "--..--", "..--..","-.-.--" };

// remove whitespace at the beginning and end of the input in case the

//user is a qa tester in disguise

userString = userString.Trim();

userString = userString.ToLower();

//iterate through the string to verify that the character falls into

//one of the three categories

foreach (char character in userString)

{

if (letters.Contains(character))

{

for (int index = 0; index < letters.Length; index++)

{

if (character == letters[index])

{

translation += (morseLetters[index] + " ");

}

}

}

else

{

MessageBox.Show("the charcter " + character + " is not allowed so I'm choosing to ignore it");

}

}

translatedTextBox.Text = translation.Trim();

} catch (Exception ex)

{

//just in case

MessageBox.Show(ex.Message);

}

}

}

}