Atbash

Software Documentation

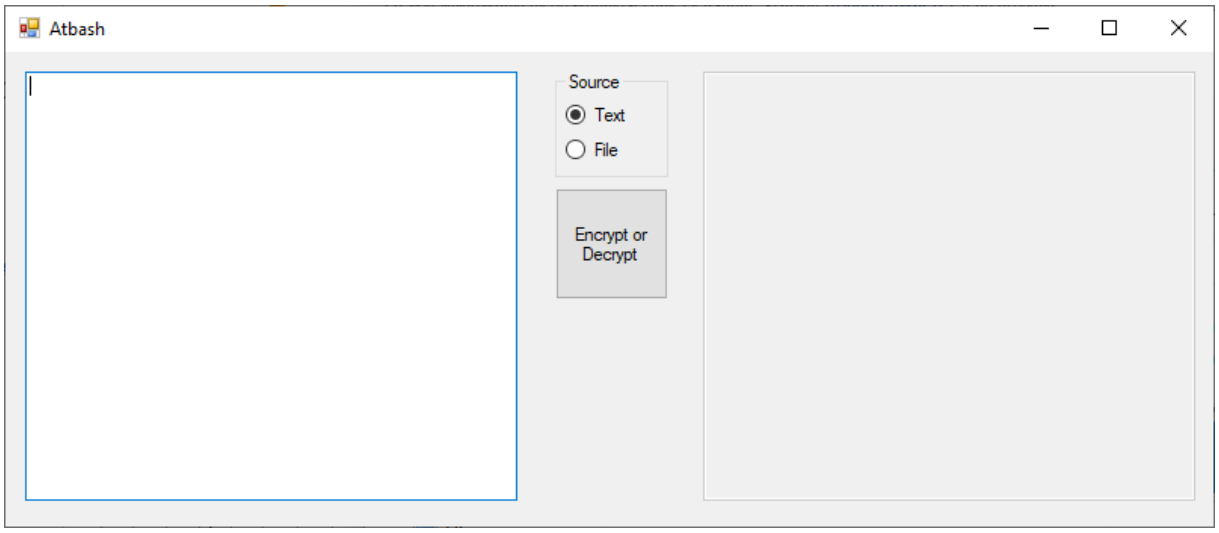
Author: matiwa

Table of contents

Introduction

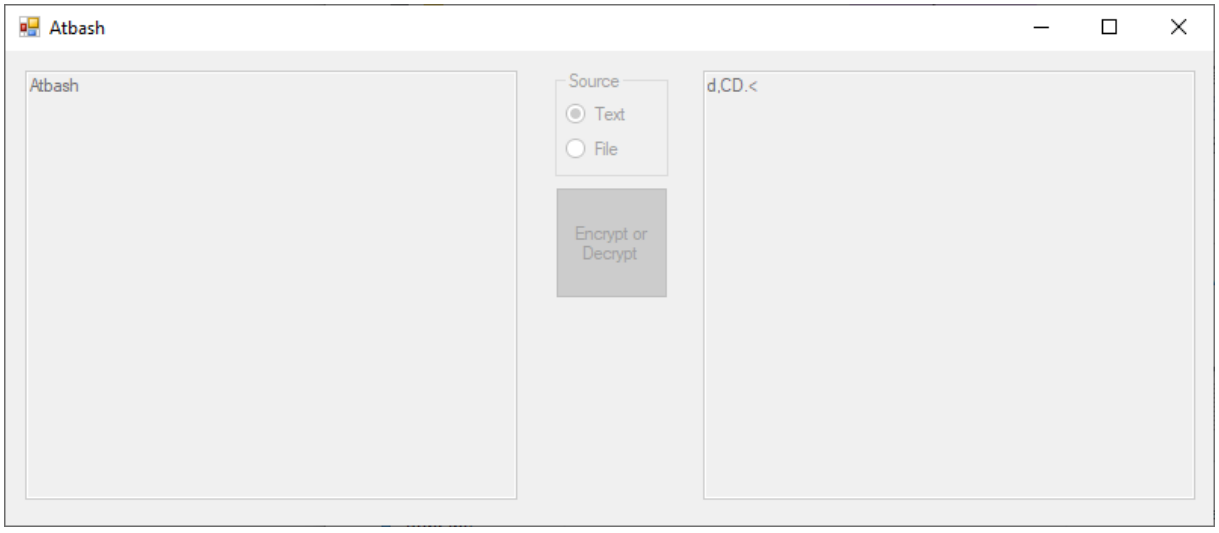
This software documentation includes: description of the application's operation, what is needed for use, algorithms used, interface description and source code description. This application is used to encoding and decoding atbash algorithm.

Describing of the application’s operation

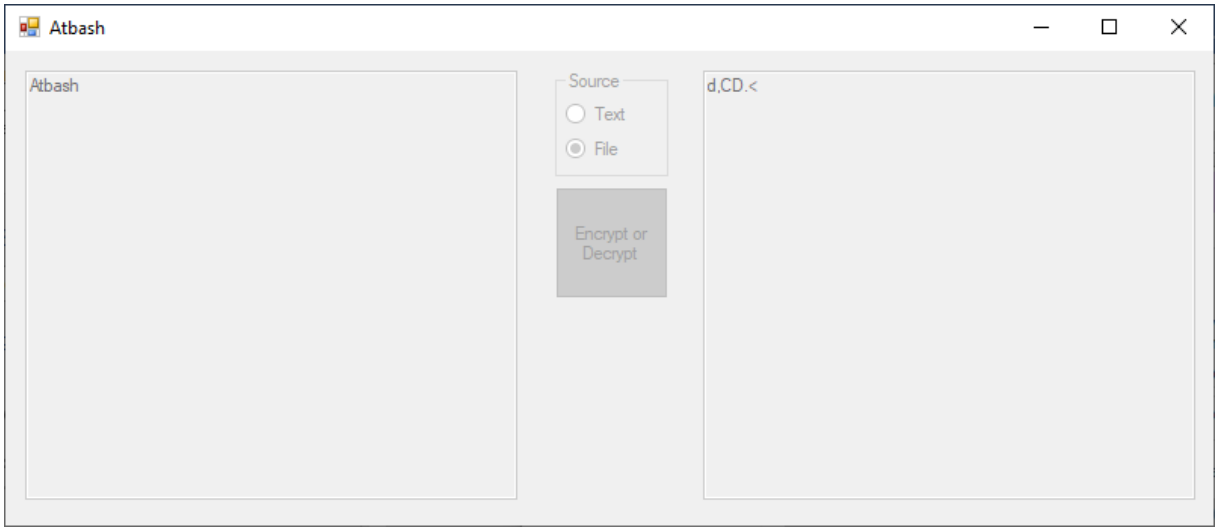


Drawing 1: The beginning of the application's operation [own study]

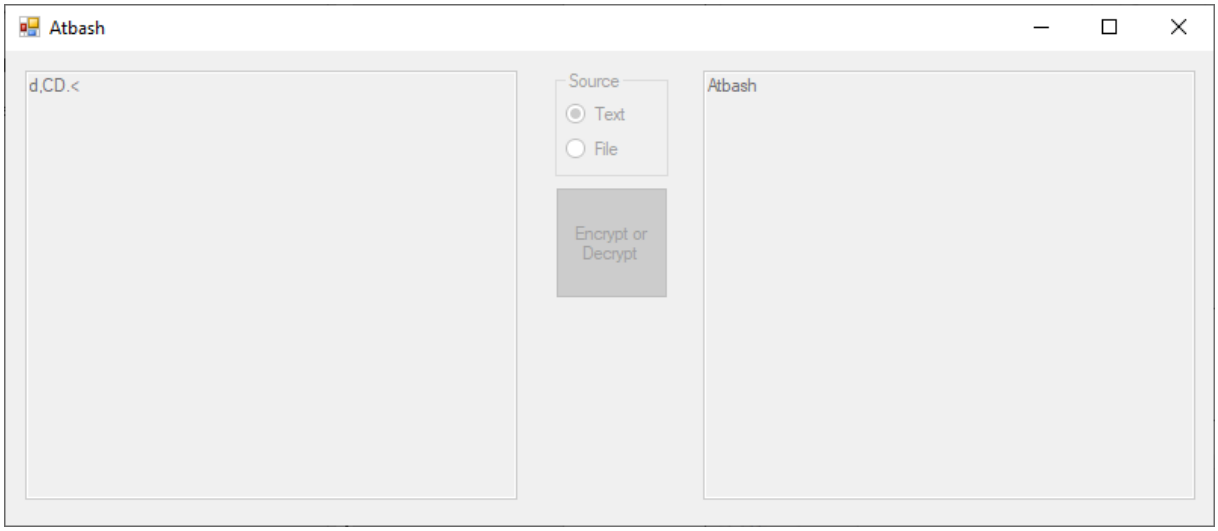
The user has to select the source of the text. If he selects a file, the text box is inactive. If it selects text, the text box is active. When selecting a text source, it is saved to the file as a backup. Selecting a file means that the text will be loaded and printed in the text field. The app has no choice between encryption and decryption as both operations look the same.



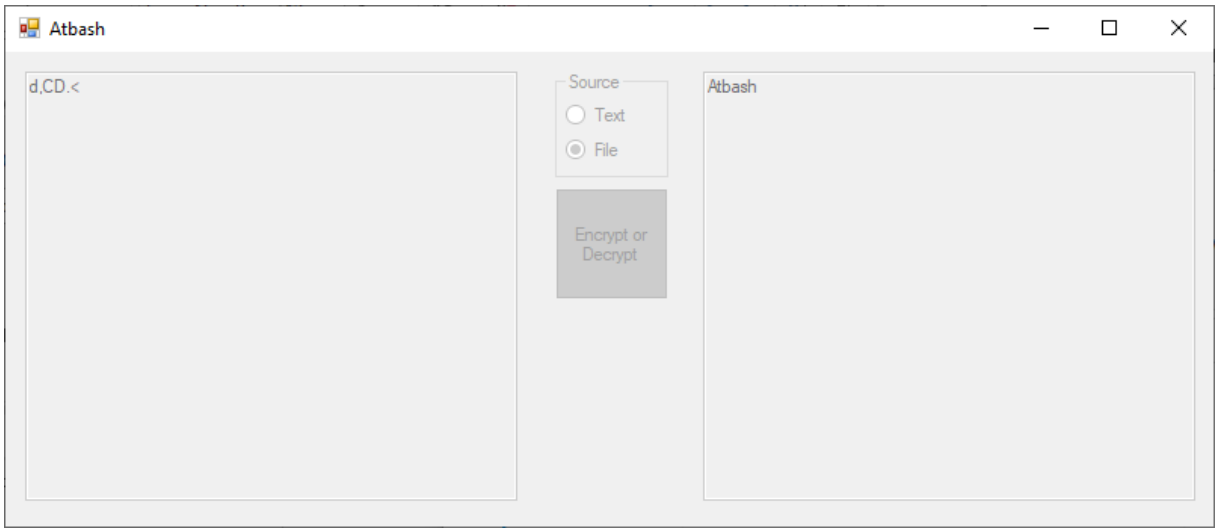
Drawing 2: Text coding example [own study]



Drawing 3: An encoding example from a file [own study]



Drawing 4: Text decoding example [own study]



Drawing 5: An encoding example from a file [own study]

This is the correct use of the program. There may be coding errors that have not been detected by the developer.

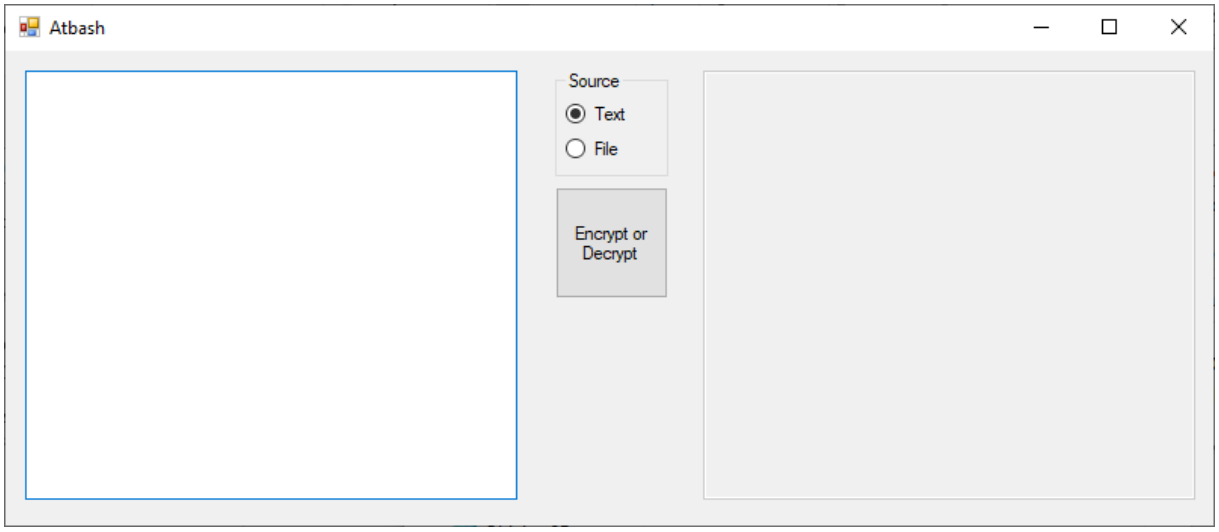
What is needed for use?

The application does not require installation. It only needs the Windows operating system.

Algorithm used

Atbash - a simple monoalphabetic substitution cipher of Hebrew origin, which works by changing the letter lying at a distance of X from the beginning of the alphabet to a letter lying at a distance of X from its end. In order to decrypt the text encrypted with this cipher, it is enough to re-encrypt it.[1]

Interface description



Drawing 6: Graphical interface [own study]

The interface is typical for a windowing application. There are essential components: button, text boxes, and radio buttons. The button starts the encryption and decryption procedure. Due to the same nature between the two operations, there are no radio buttons for these methods. The message text box is active when the text options button is selected and inactive for the file options.

Source code description

The project was made in the C# programming language, in the Visual Studio Community 2019 programming environment. All work was done on the Windows 10 operating system. The application's source code looks like this.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace Atbash
{
    public partial class Form1 : Form
    {
        FileStream file = new FileStream("message.txt",
FileMode.OpenOrCreate, FileAccess.ReadWrite);
        char[] chars = new char[]{'
','!','"','#','$','%','&','\'','(',')','*','+',',','-','.','/','0',

'1','2','3','4','5','6','7','8','9',':',';','<','=','>','?','@','A',
'Ą','B','C','Ć','D','E',

'Ę','F','G','H','I','J','K','L','Ł','M','N','Ń','O','Ó','P','Q','R',
'S','Ś','T','U','V','W',

'X','Y','Z','Ź','Ż','[','/',']','^','_','`','a','ą','b','c','ć','d',
'e','ę','f','g','h','i',

'j','k','l','ł','m','n','ń','o','ó','p','q','r','s','ś','t','u','v',
'w','x','y','z','ź','ż',
            '{','|','}','~'}; //ASCII code combined with Polish
diacritized letters

        string message = "";
        char[] messagechar;
        char[] codetable;
        public Form1()
        {
            InitializeComponent();
            //creating a code table
            int k =-1;
            codetable = new char[chars.Length];
            for (int i = codetable.Length-1; i >= 0; i--)
            {
```

```
                k++;
                codetable[k] = chars[i];
            }
        }

        //select a message input method
        private void Selectsource(object sender, EventArgs e)
        {
            if (rBText.Checked == true) tBText.Enabled = true;
            else if (rBFile.Checked == true) tBText.Enabled = false;
        }

        private void Btngo_Click(object sender, EventArgs e)
        {
            //loading the message from the text field and saving it
to a file
            if (rBText.Checked == true)
            {
                message = tBText.Text;
                StreamWriter sw = new StreamWriter(file);
                sw.WriteLine(message);
                sw.Close();
            }
            //loading a message from a file and saving it to the
text field
            else if (rBFile.Checked == true)
            {
                StreamReader sr = new StreamReader(file);
                message = sr.ReadLine();
                sr.Close();
                tBText.Text = message;
            }

            //message encryption or decryption
            messagechar = message.ToCharArray();
            for(int i = 0; i < messagechar.Length; i++)
            {
                //encrypting or decrypting messages works the same
                for (int j = 0; j < chars.Length;j++)
                    if (messagechar[i] == chars[j]) tBMessage.Text
+= codetable[j];
            }
            //protection against the next operation
            tBText.Enabled = false;
            btnoperation.Enabled = false;
            gBSource.Enabled = false;
        }
    }
}
```

Listing 1: Source code [own study]

List of drawings

List of listings

Bibliography

[1] https://pl.wikipedia.org/wiki/Atbasz