

Guess the number

Software Documentation

Author: matiwa

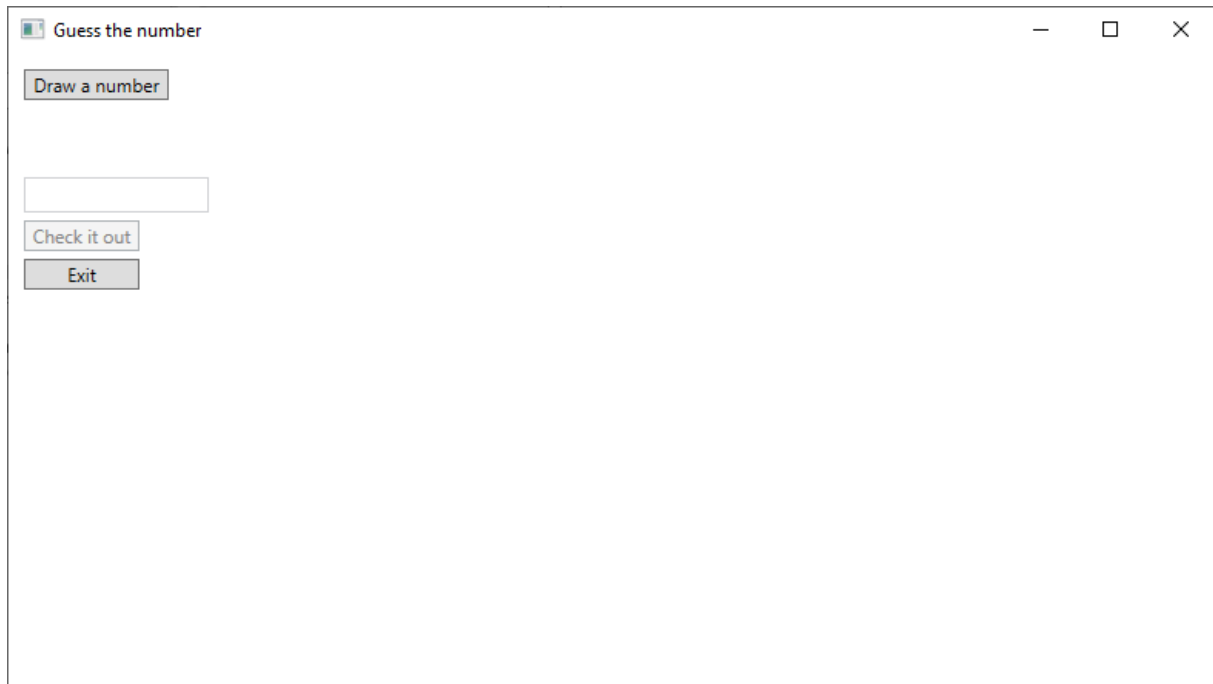
Table of contents

Table of contents.....	2
Introduction.....	3
Describing of the application's operation.....	3
What is needed for use?.....	11
Algorithm used.....	11
Interface description.....	11
Source code description.....	12
List of drawings.....	15
List of listings.....	15

Introduction

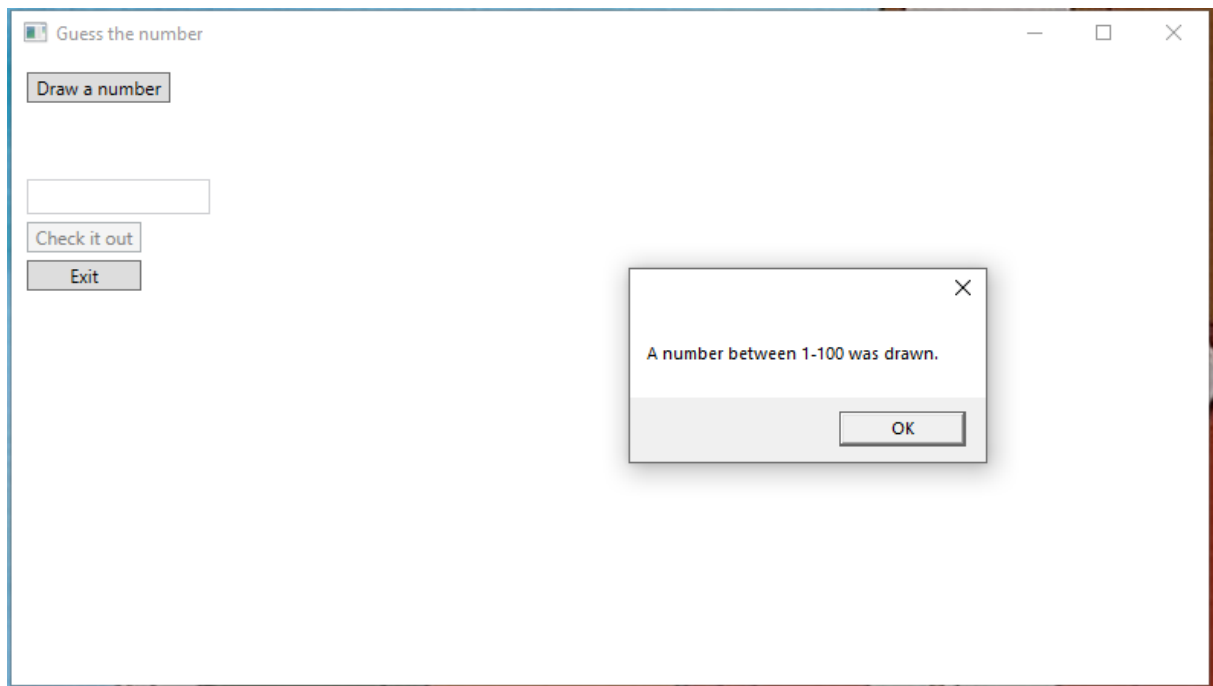
This software documentation includes: description of the application's operation, what is needed for use, algorithms used, interface description and source code description. This application is used to guess the drawn number.

Describing of the application's operation

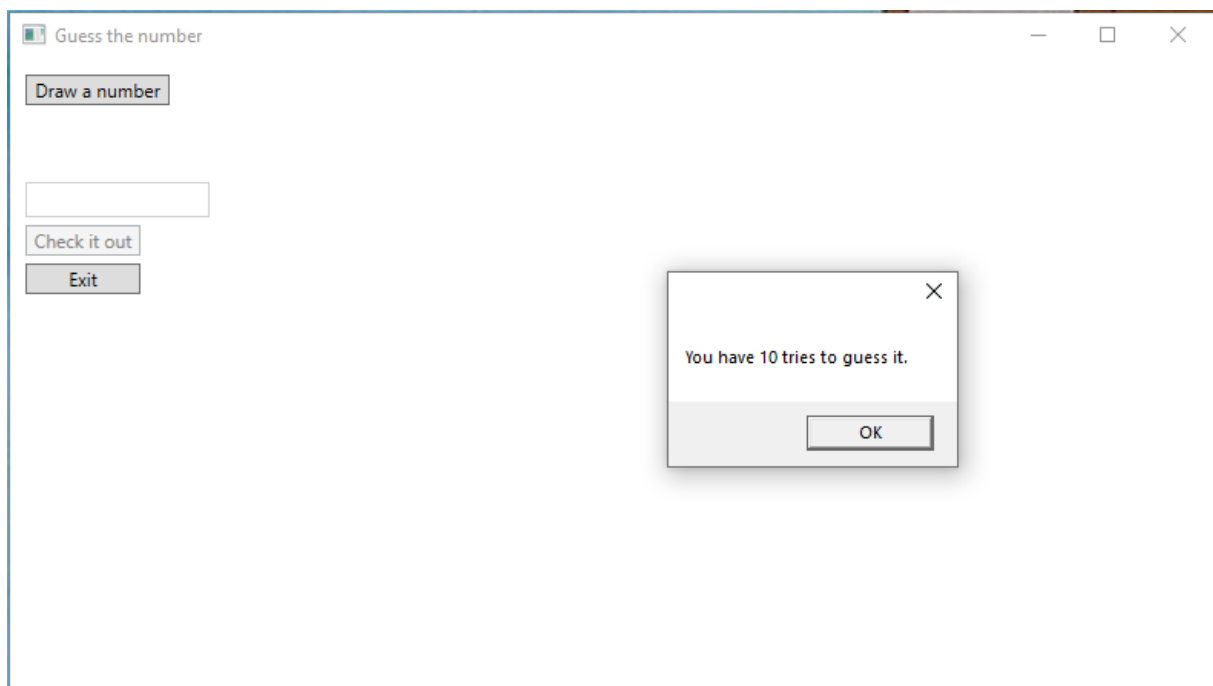


Drawing 1: The beginning of the application's operation [own study]

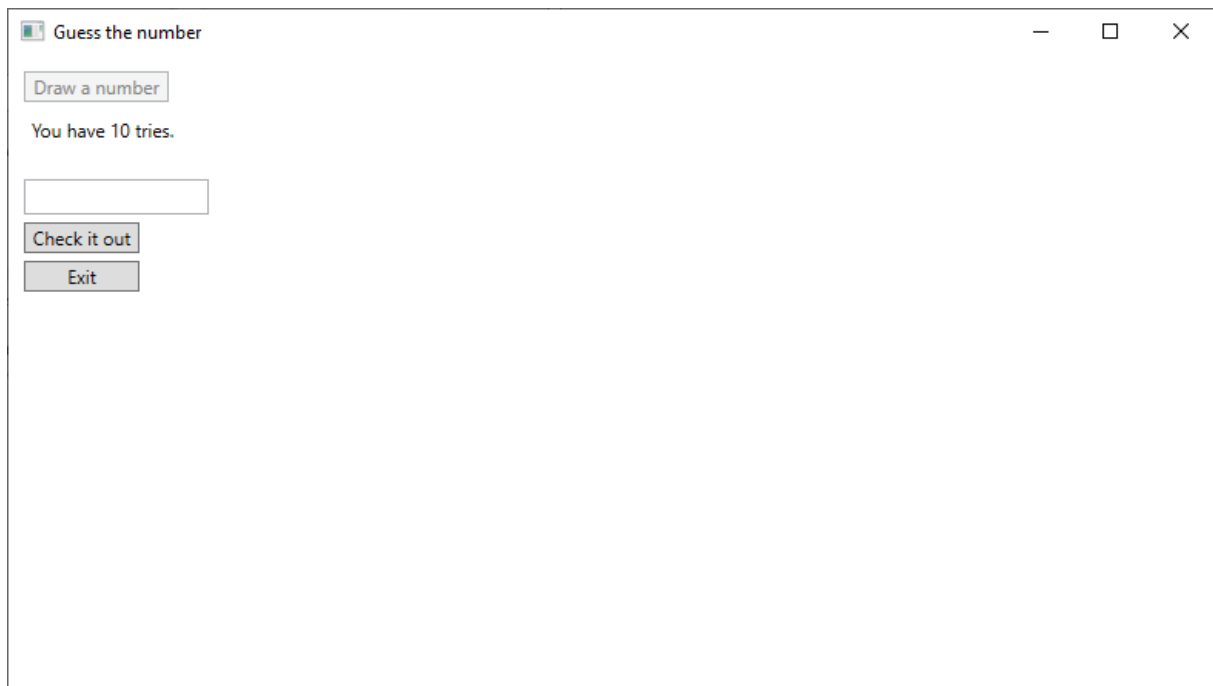
The user needs the "Draw a number" button in order for the program to draw a number. After this action, an appropriate message will appear that a number between 1 and 100 has been drawn. After the user confirms that the user has read the message, another message appears stating that there are 10 attempts to guess the number. The complete test sequence is shown below. There are messages with hints about the drawn number if the number entered is too small or too large. There are also messages that tell you the number of tries you have left. There is a message about a guessed number and a win. There is also a loss message with information about the number that the user was looking for. The characteristics of the application and the accompanying messages characterize a typical game of chance. After guessing (winning) or losing, the user can play again. The loss message was presented without any mistakes.



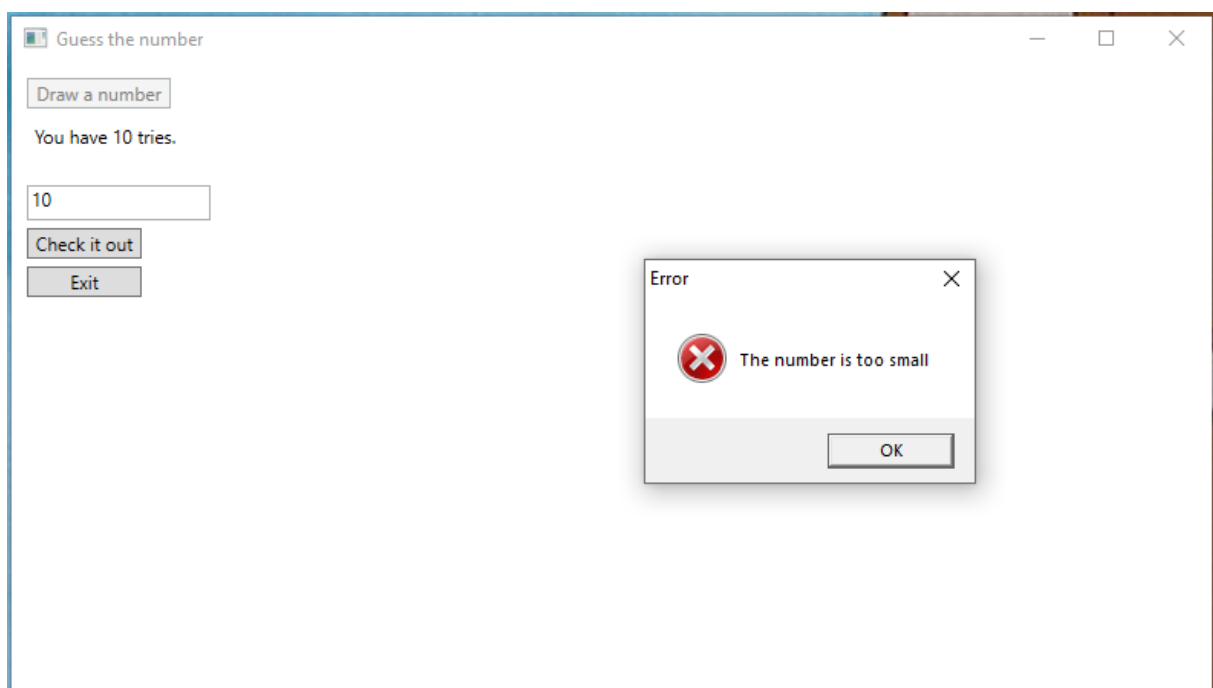
Drawing 2: First step [own study]



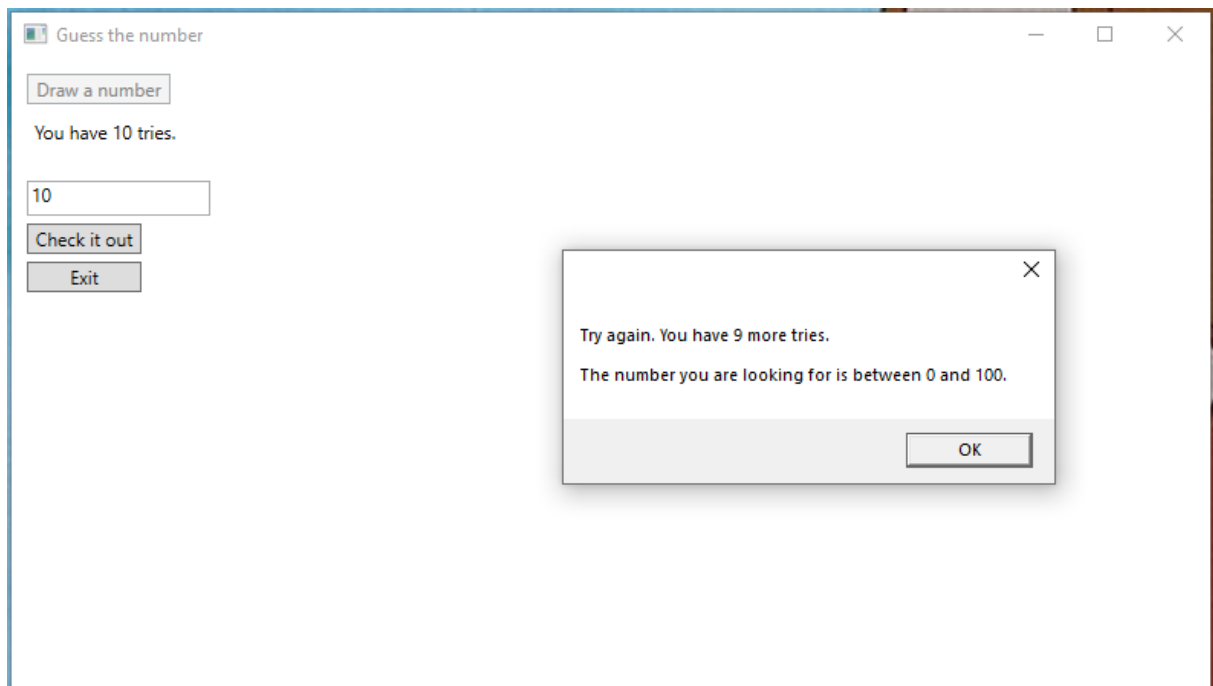
Drawing 3: Second step [own study]



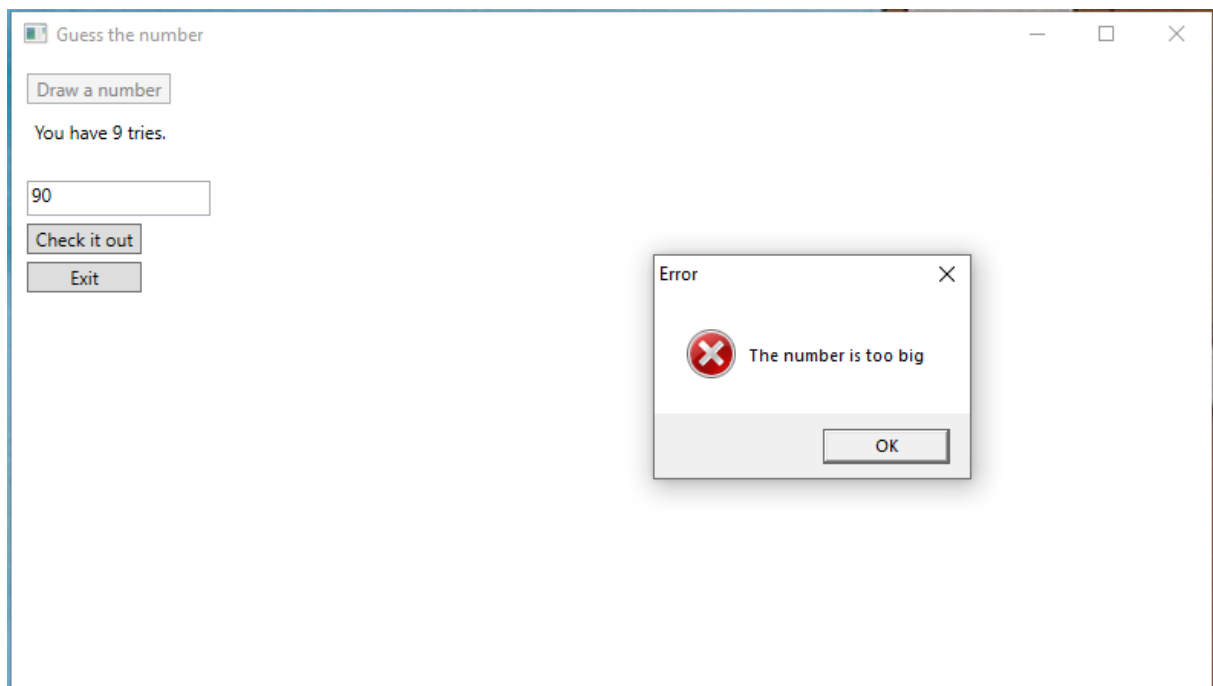
Drawing 4: The application is ready for the first attempt [own study]



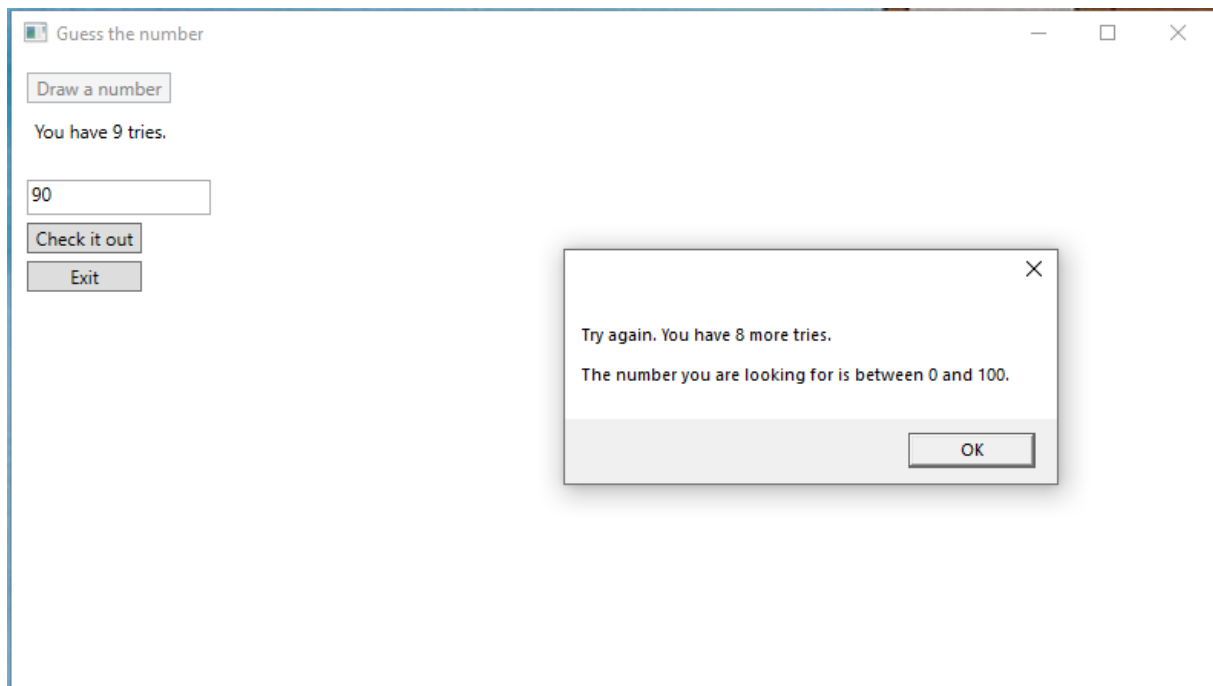
Drawing 5: Prompt [own study]



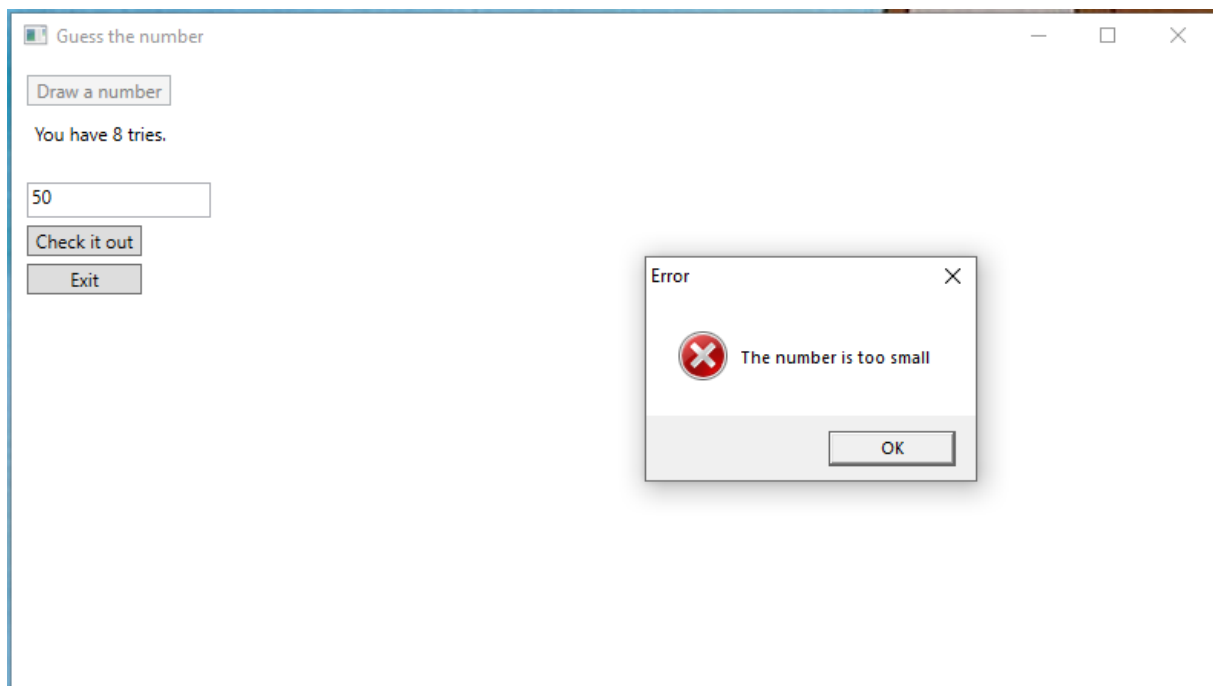
Drawing 6: Information about the number of attempts [own study]



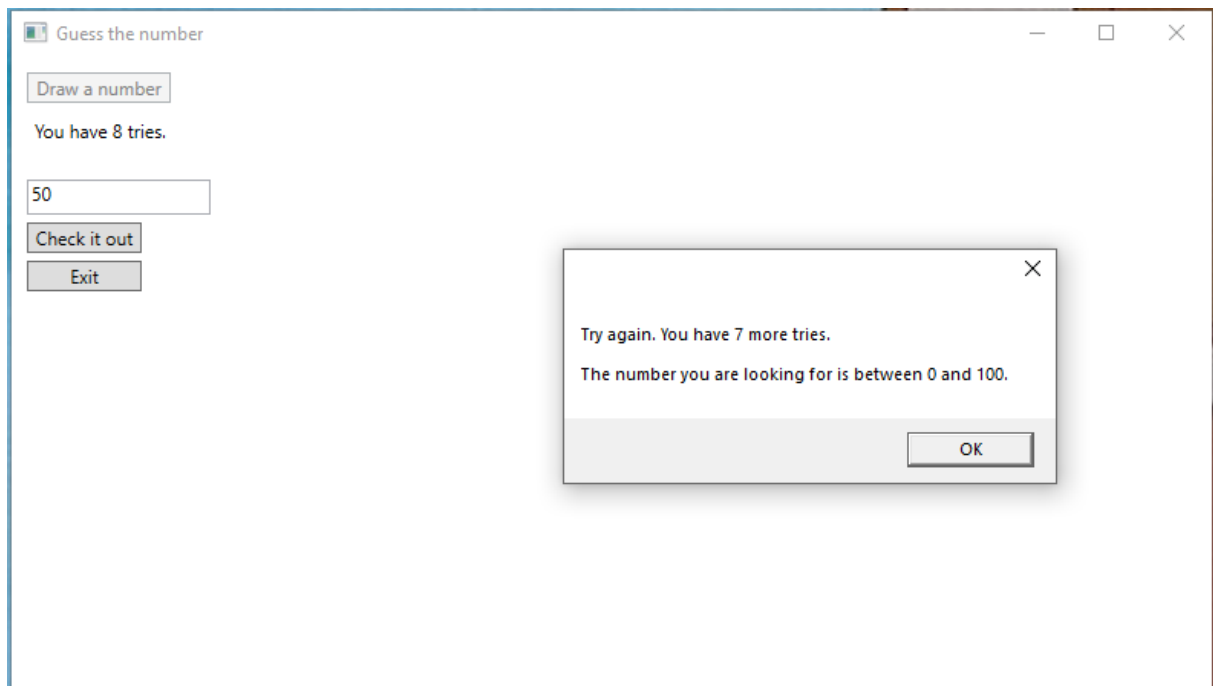
Drawing 7: Prompt [own study]



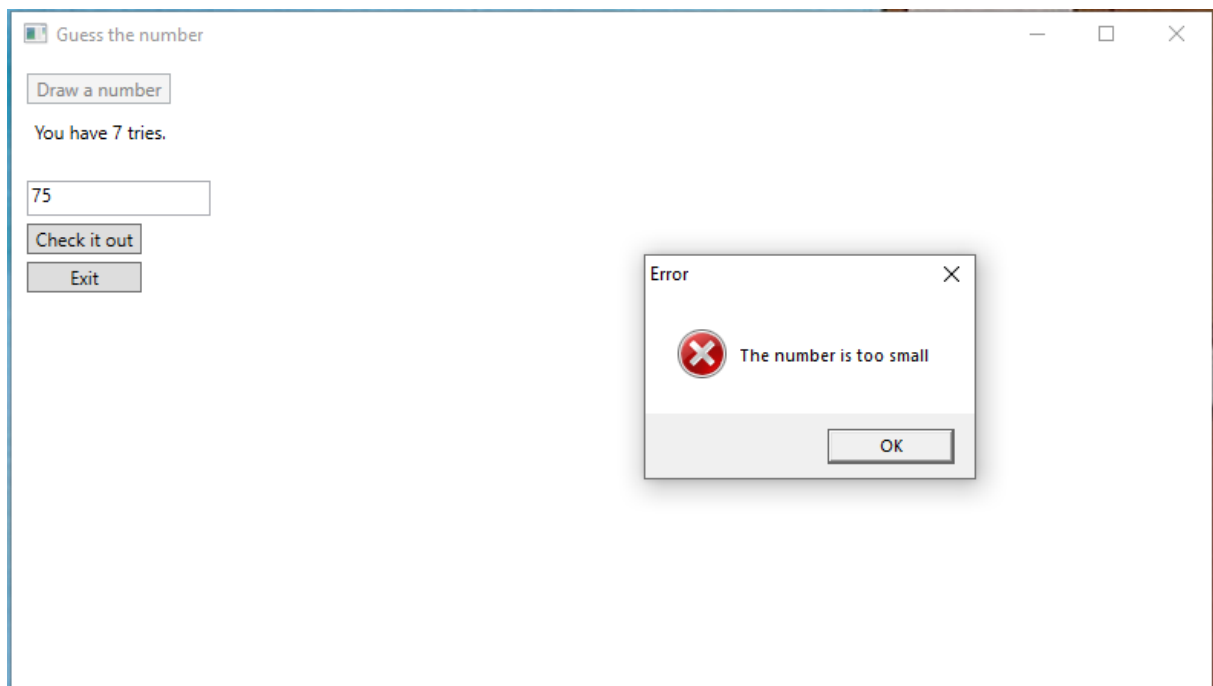
Drawing 8: Information about the number of attempts [own study]



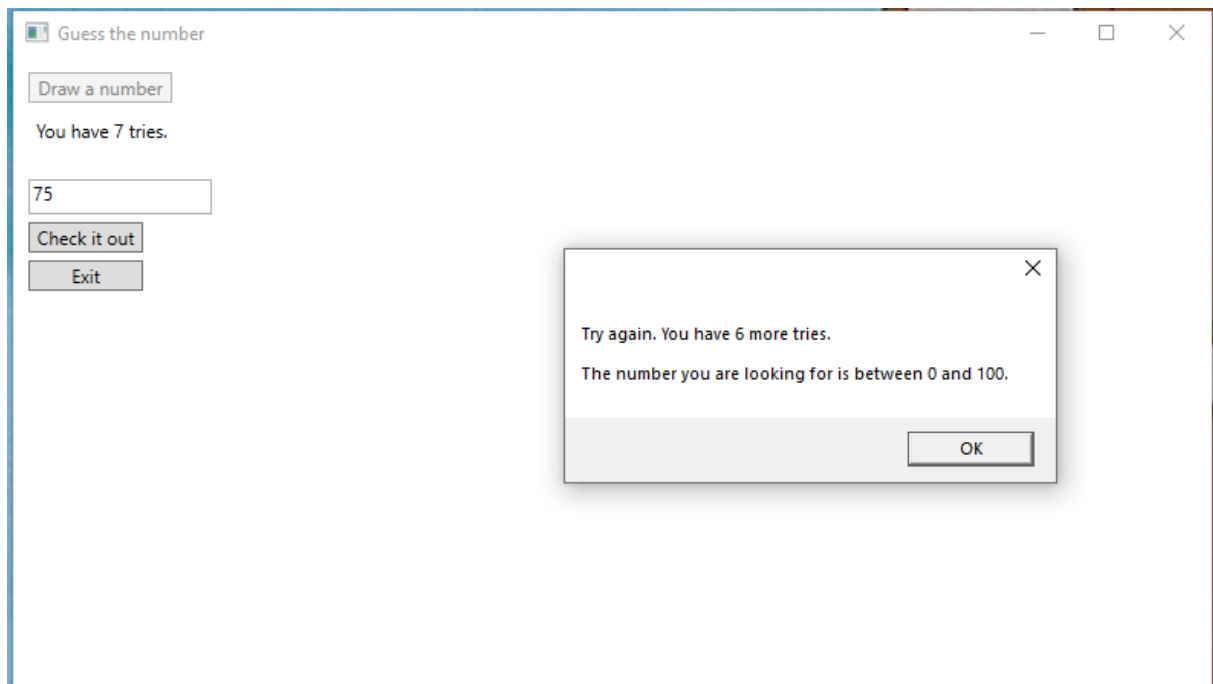
Drawing 9: Prompt [own study]



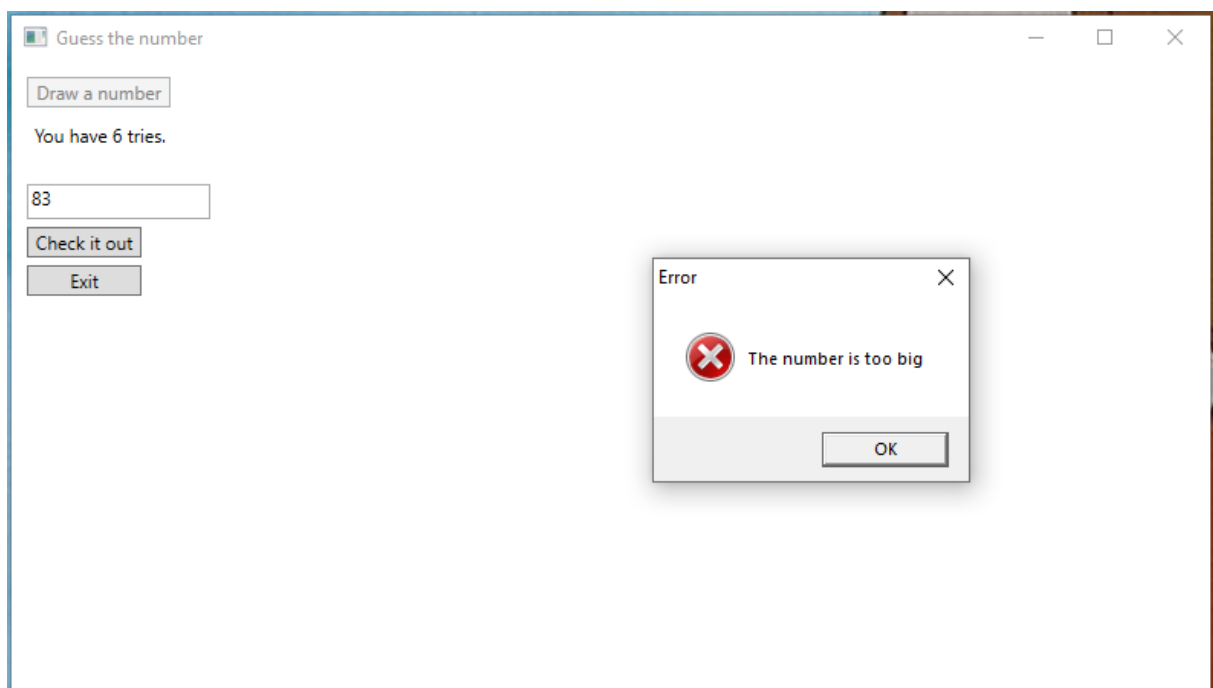
Drawing 10: Information about the number of attempts [own study]



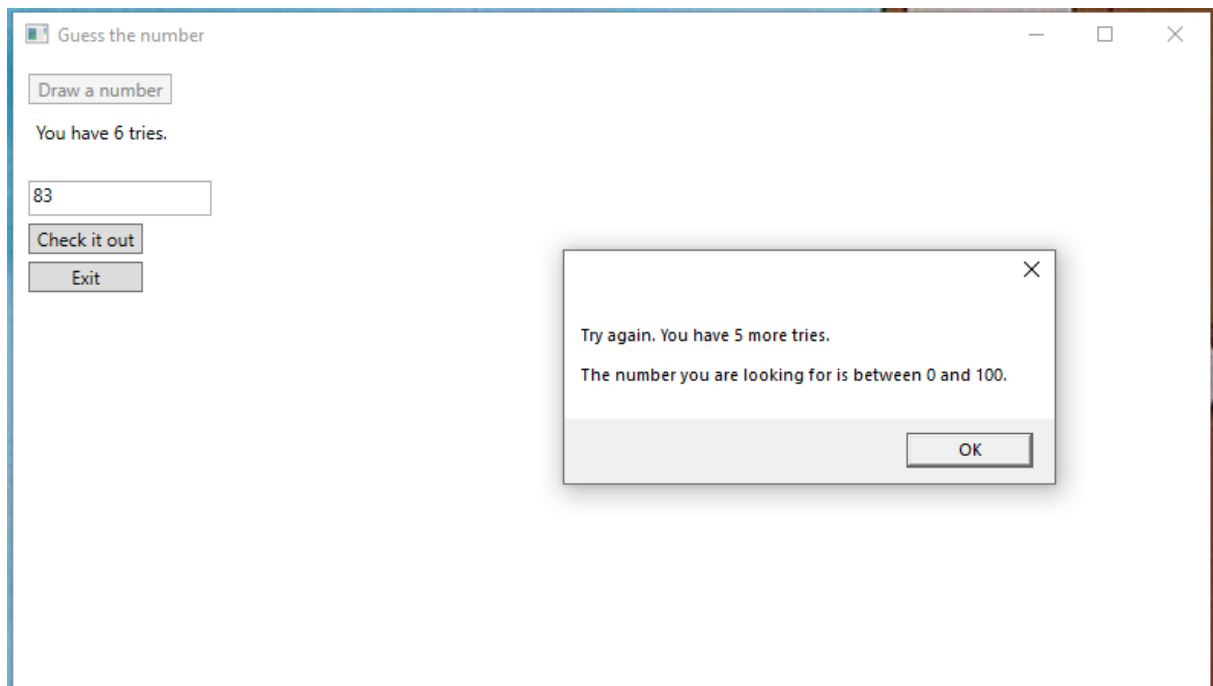
Drawing 11: Prompt [own study]



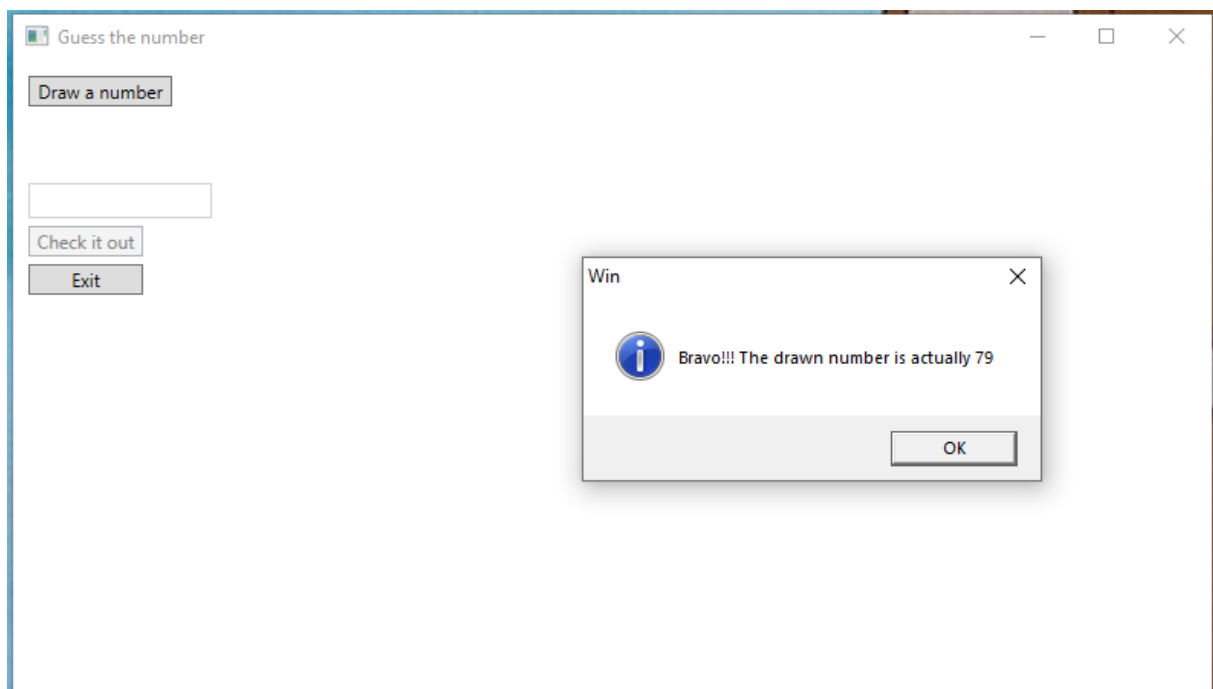
Drawing 12: Information about the number of attempts [own study]



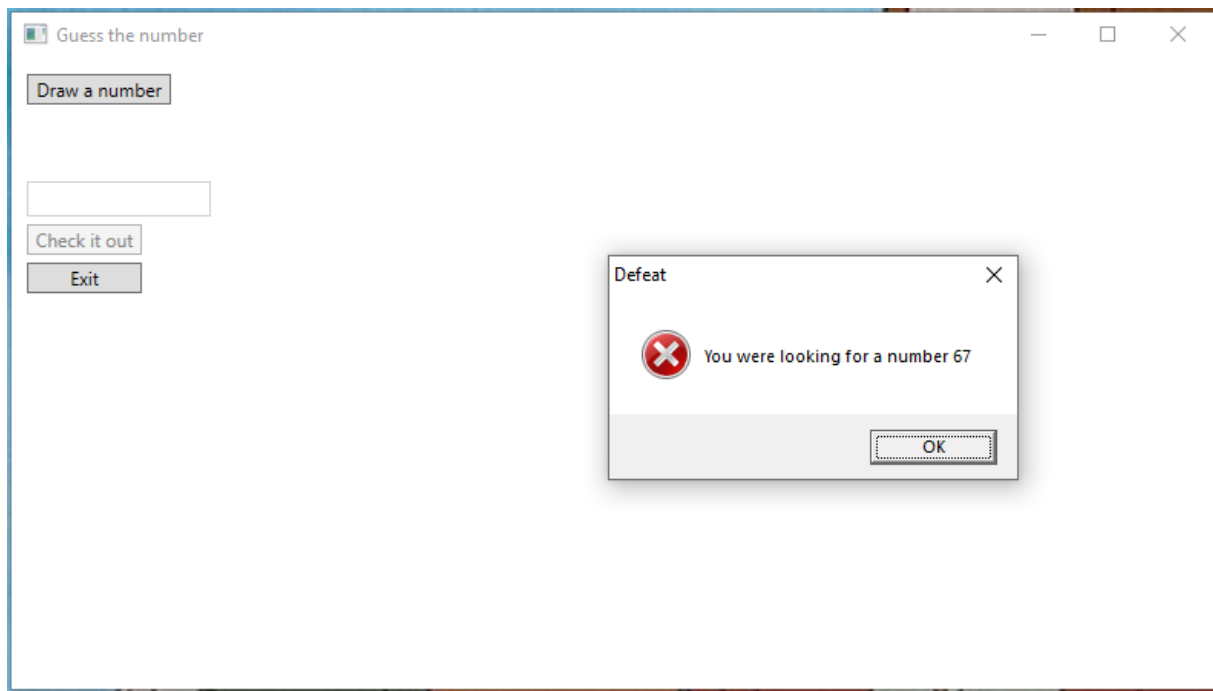
Drawing 13: Prompt [own study]



Drawing 14: Information about the number of attempts [own study]



Drawing 15: Information about the win [own study]



Drawing 16: Information about the loss [own study]

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

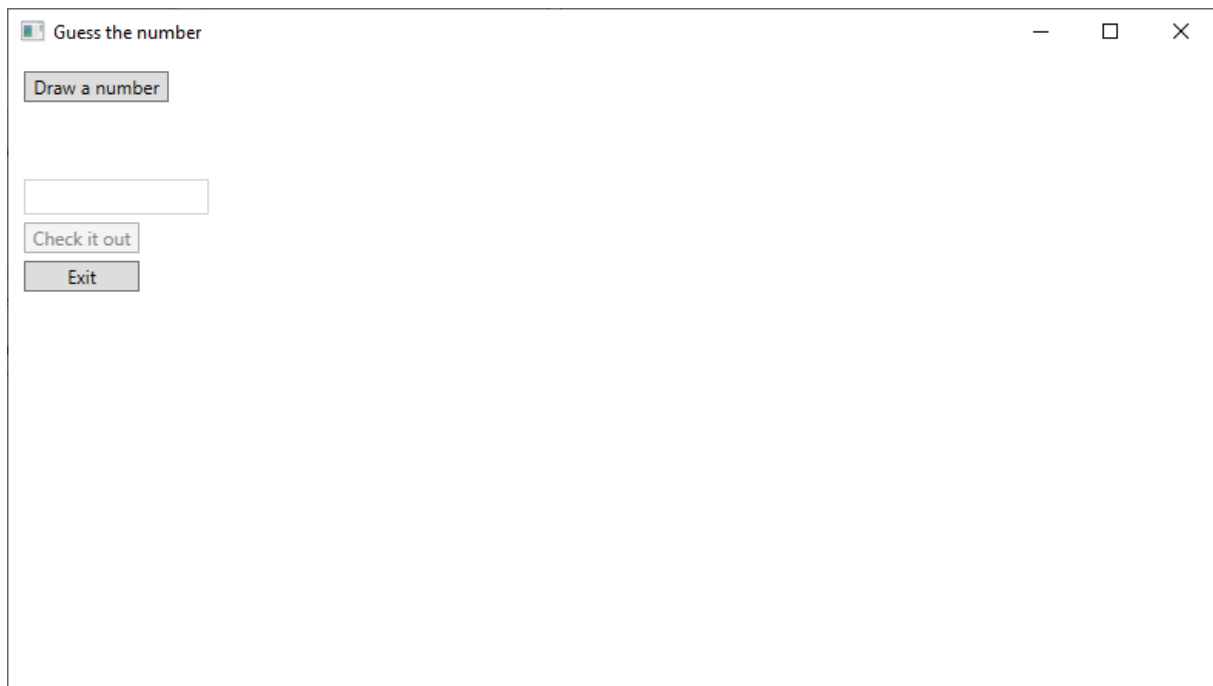
What is needed for use?

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

Algorithm used

The basic form of the algorithm can be deduced from the previous section. To sum up, the application is a typical game of guessing the drawn number. The user has 10 tries to guess it. He has hints on what to follow when guessing when he makes a mistake. He is kept informed about the number of attempts he has left. If he wins, he has information about it. If he loses, he knows it too. After one of these two messages, the game starts all over again.

Interface description



Drawing 17: Graphical interface [own study]

The interface is typical for a Windows Presentation Foundation (WPF) development platform. There are essential components: buttons, text boxes, and label.

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.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Guess_the_number
{
```

```

public partial class MainWindow : Window
{
    private byte not;//number of tries
    private byte nd;//number drawn

    public MainWindow()
    {
        InitializeComponent();
        tv.IsEnabled = false;
        btncio.IsEnabled = false;
        tbean.IsEnabled = false;
    }

    private void btndan_Click(object sender, RoutedEventArgs e)
    {
        MessageBox.Show("A number between 1-100 was drawn.");
        not = 10; //resetting the number of trying
        MessageBox.Show("You have "+not+" tries to guess it.");
        //draw a number
        Random r = new Random();
        nd = Convert.ToByte(r.Next(1, 101));
        //the ability to guess the number
        tv.IsEnabled = true;
        btncio.IsEnabled = true;
        tbean.IsEnabled = true;
        tv.Content = "You have " + not + " tries.";
        //random number locked
        btndan.IsEnabled = false;
    }

    private void btncio_Click(object sender, RoutedEventArgs e)
    {
        try
        {
            byte ne = Convert.ToByte(tbean.Text);//number
entered
            //wrong hit
            if (ne != nd && not > 0)
            {
                if (ne < nd) MessageBox.Show("The number is too
small", "Error", MessageBoxButton.OK,
                MessageBoxImage.Error);
                else if (ne > nd) MessageBox.Show("The number is
too big", "Error", MessageBoxButton.OK,
                MessageBoxImage.Error);
                not--;
                MessageBox.Show("Try again. You have " + not + "
more tries.\r\n\r\n")

```

```

        + "The number you are looking for is between
0 and 100.");
        tv.Content = "You have " + not + " tries.";
    }
    else if (ne == nd)
    {
        //number draw unlocked
        btndan.IsEnabled = true;
        //number quessing locked
        btncio.IsEnabled = false;
        tv.IsEnabled = false; tv.Content = "";
        tbean.IsEnabled = false; tbean.Text = "";
        //displaying the number
        MessageBox.Show("Bravo!!! The drawn number is
actually " + ne, "Win", MessageBoxButton.OK,
            MessageBoxImage.Asterisk);
    }
    if (not < 1)
    {
        //number draw unlocked
        btndan.IsEnabled = true;
        //number quessing locked
        btncio.IsEnabled = false;
        tv.IsEnabled = false; tv.Content = "";
        tbean.IsEnabled = false; tbean.Text = "";
        //displaying the number
        MessageBox.Show("You were looking for a number "
+ nd, "Defeat", MessageBoxButton.OK,
            MessageBoxImage.Stop);
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void btne_Click(object sender, RoutedEventArgs e)
{
    Environment.Exit(0); //closing the application
}
}
}

```

Listing 1: Source code [own study]

List of drawings

Drawing 1: The beginning of the application's operation [own study].....	3
Drawing 2: First step [own study].....	4
Drawing 3: Second step [own study].....	4
Drawing 4: The application is ready for the first attempt [own study].....	5
Drawing 5: Prompt [own study].....	5
Drawing 6: The application is ready for the first attempt [own study].....	6
Drawing 7: Prompt [own study].....	6
Drawing 8: The application is ready for the first attempt [own study].....	7
Drawing 9: Prompt [own study].....	7
Drawing 10: The application is ready for the first attempt [own study].....	8
Drawing 11: Prompt [own study].....	8
Drawing 12: The application is ready for the first attempt [own study].....	9
Drawing 13: Prompt [own study].....	9
Drawing 14: The application is ready for the first attempt [own study].....	10
Drawing 15: Information about the win [own study].....	10
Drawing 16: Information about the loss [own study].....	11
Drawing 17: Graphical interface [own study].....	12

List of listings

Listing 1: Source code [own study].....	12
---	----