LAPLAND UNIVERSITY OF APPLIED SCIENCES

Assignments

DotNET Application Development

A1301794 – Mert E. Musul

TABLE OF CONTENTS

ASSIGNMENT 1

TASK

Write a program that plays “guess the number” as follows: Your program chooses the number to be guessed by selecting an int at random in the range 1–1000. The program then displays the following text in a label:

I have a number between 1 and 1000--can you guess my number?

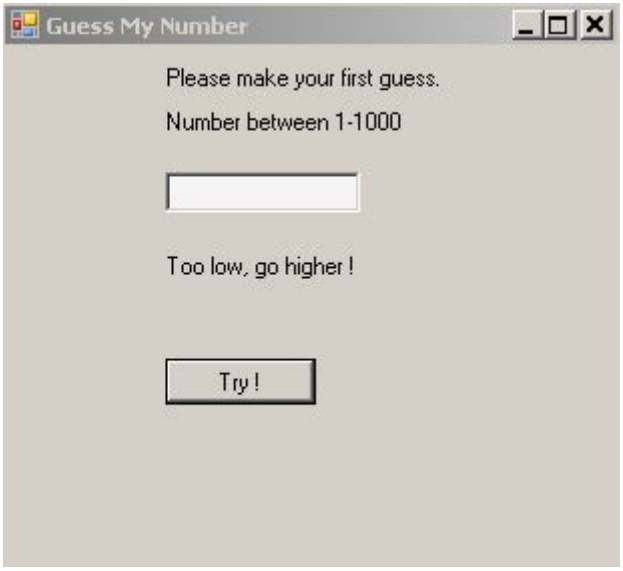
Please enter your first guess.

A TextBox should be used to input the guess. As each guess is input, the background color should change to red or blue. Red indicates that the user is getting “warmer,” blue that the user is getting “colder.” A Label should display either “Too High” or “Too Low,” to help the user zero in on the correct answer. When the user guesses the correct answer, display “Correct!” in a message box, change the Form’s background color to green and disable the TextBox. Recall that a TextBox (like other controls) can be disabled by setting the control’s Enabled property to false. Provide a Button that allows the user to play the game again. When the Button is clicked, generate a new random number, change the background to the default color and enable the TextBox.

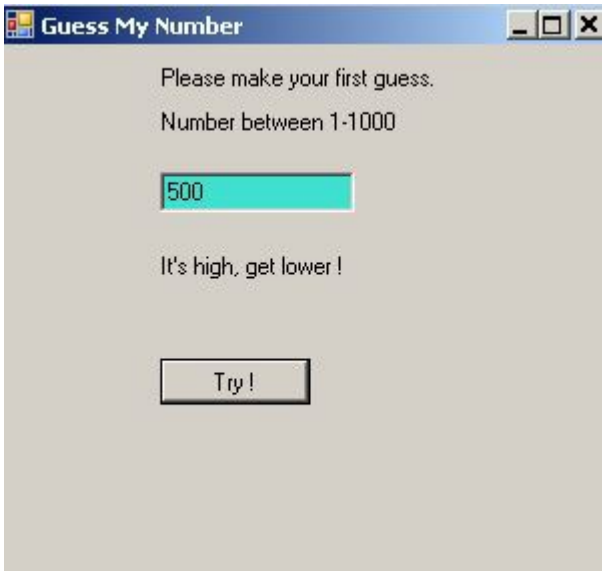
Instructions:

- Declare a property for each member variable
- Create at least two classes and several methods inside them. Do not put all the code inside the main method.

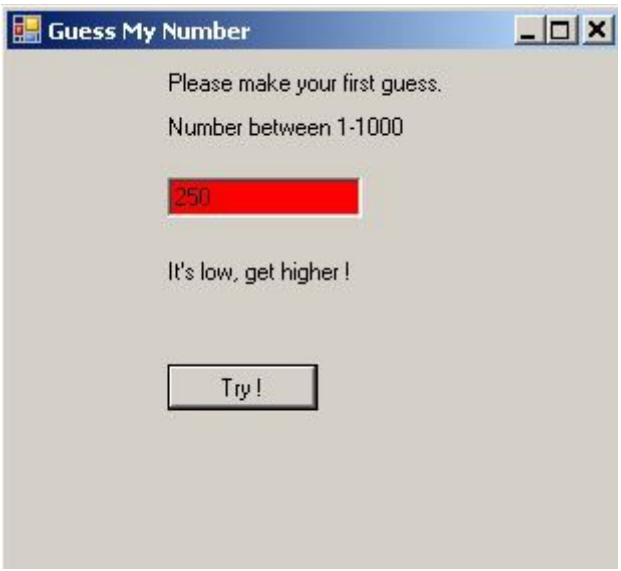
SOLUTION



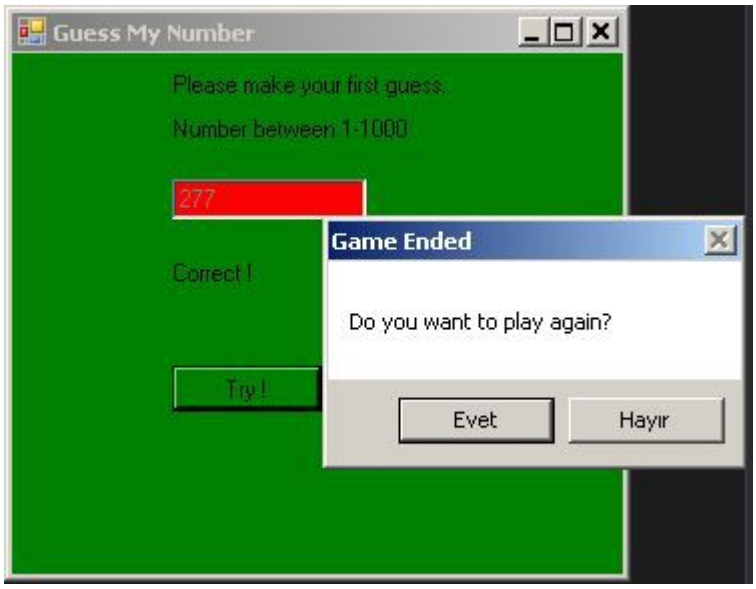
First, welcoming screen.



When player is away from the number “colder”, textbox color is blue.



And when player is getting closer, its warmer, So, textbox is red.



When game ends, a dialog box appear and asks the player, if he wants to play again or not.

If answer is “Yes”, game starts again, “No” means exit.

```
ï»¿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;




namespace GuessMyNumber
{


    public partial class Form1 : Form
    {
       int thenumber;   // the holden number by pc
       int guess;       // the guessed number by players
       int choosenbutton;  // dialog box's answer


        private void GenerateRandoms()        // function to randomize numbers
{
    // create random number variable
    Random random = new Random();
    thenumber = random.Next(0, 1000);        // randoming the number and assigning

        }

        public Form1()
        {
            InitializeComponent();
            GenerateRandoms();

        }

        private void button1_Click_1(object sender, EventArgs e)
        {

           Int32.TryParse(textBox1.Text, out guess); // converting string to int
                                                     // to compare based on integers
           if (guess > thenumber)
               label2.Text = "It's high, get lower !";

           if (guess < thenumber)                          // comparing guess and
               label2.Text = "It's low, get higher !";     // holden number by pc

           if (guess == thenumber)
           {
               label2.Text = "Correct !";             // when guess is right
               this.BackColor = System.Drawing.Color.Green; //form's background green
               textBox1.Enabled = false;               // textbox is disabled

               DialogResult choosenbutton = MessageBox.Show("Do you want to play
again?","Game Ended", MessageBoxButtons.YesNo);  // asking question to play again
```

```
                if (choosenbutton == DialogResult.Yes)
                    Application.Restart();              // if answer is yes, restart
                else
                    Application.Exit();                 // if answer is no, exit
            }

            if (guess > thenumber + 100 || guess < thenumber - 100 )
            {
                textBox1.BackColor = Color.Turquoise; // getting away from the number
                          }
                else
            {
                textBox1.BackColor =  Color.Red;  // getting closer to the number

}
            }



    }
}
```

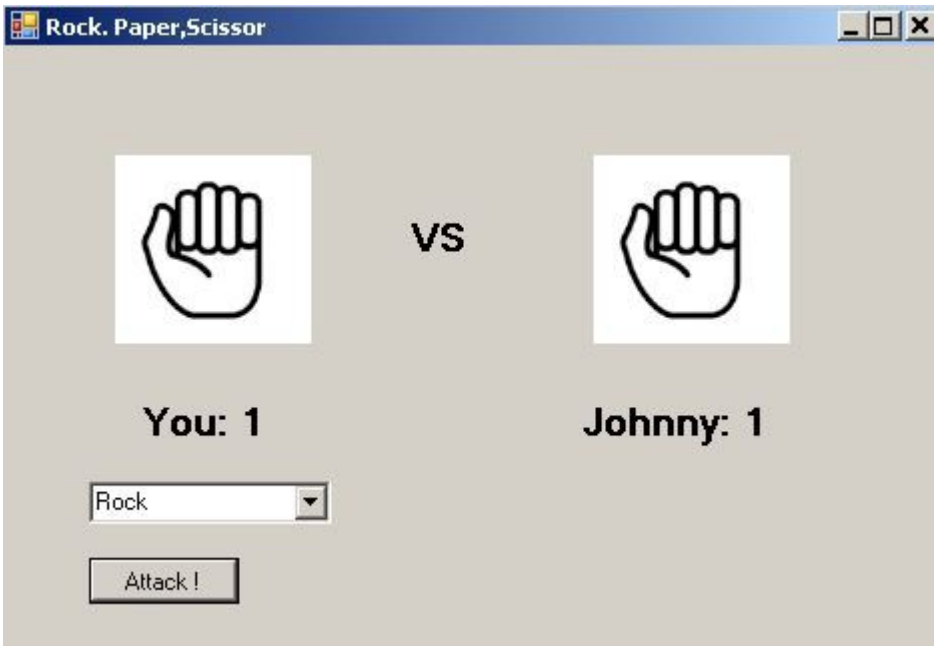
ASSIGNMENT 2

TASK

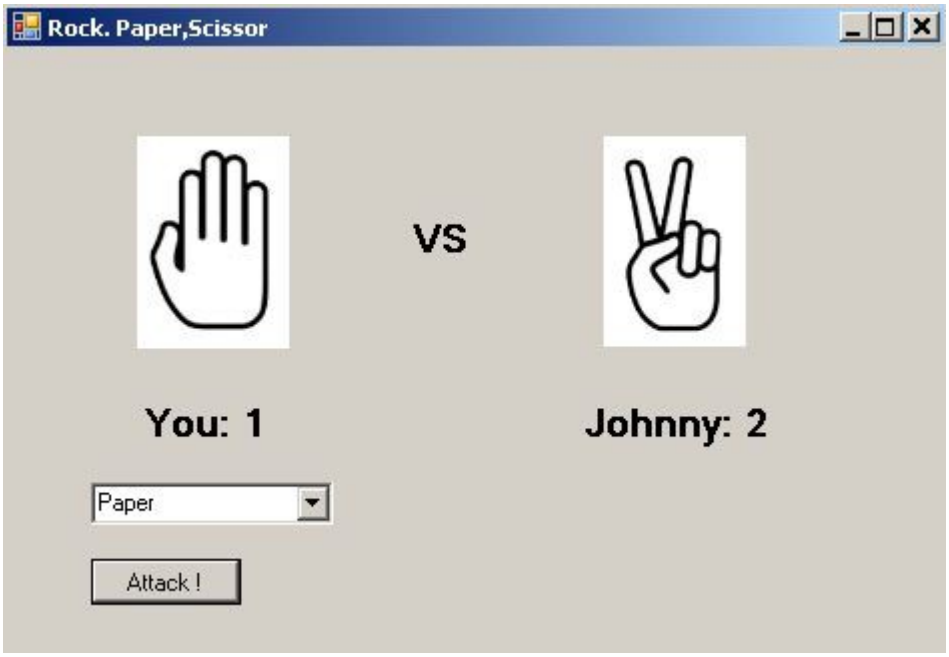
There is only one assignment to be done. Since I have been interested to develop games. I have challenge myself to do a “Rock, Paper, Scissor” game. I'd like to present it also.

SOLUTION

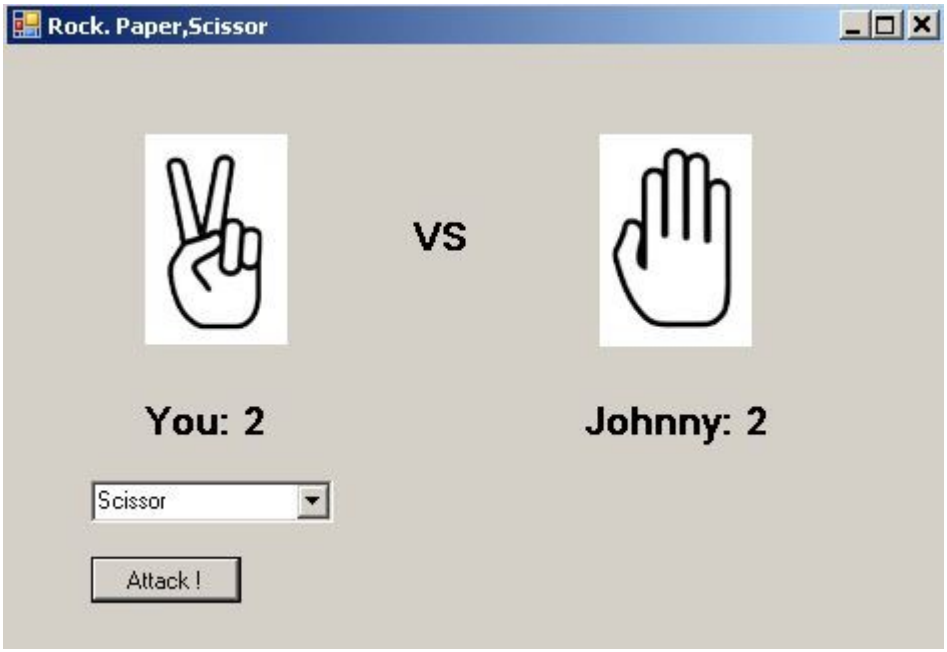
Firts player make the attacks.



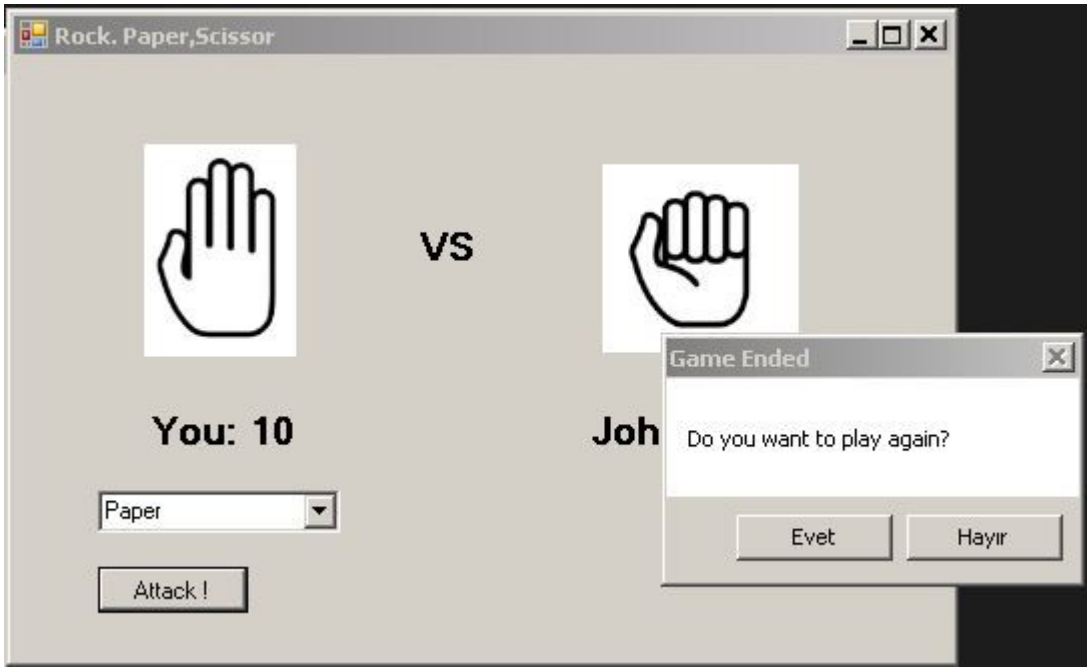
And AI answers



Different situations.



By the way, Johnny is our AI based player.



When game ends, application asks to player, if he wants to play again.

Catching the error, if player forgets to choose an attack style but tries to click on Attack button.

```
ï»¿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;

namespace RockPaperScissor
{
    public partial class Form1 : Form
    {

        int npc;           // NPC's decisions assigned to integer
        int player;        // Player's decisions assigned to integer
        int sc_player=0;   // Player's score
        int sc_npc=0;      // NPC's score
        int choosen;       // Dialog box
        private void play_npc()
        {

            Random random = new Random(); // random number variable
            npc= random.Next(0, 3);

        }


        public Form1()
        {
            InitializeComponent();
        }


        private void button1_Click(object sender, EventArgs e)
        {

            if (comboBox1.Text != "Rock" && comboBox1.Text != "Paper" &&
comboBox1.Text != "Scissor")        // Catching the error
                MessageBox.Show("Choose an attack style !", "Error");
            else
            {

                play_npc();    // NPC decides (randomly)


                if (npc == 0)                   // Showing figures according to
                {                               // NPC's decision
                    Paper2.Visible = true;
                    Scissor2.Visible = false;    // 0 = Paper
                    Rock2.Visible = false;       // 1 = Scissor
                }                                // 2 = Rock
                if (npc == 1)
                {
                    Paper2.Visible = false;
                    Scissor2.Visible = true;
                    Rock2.Visible = false;
                }
```

```
                if (npc == 2)
                {
                    Paper2.Visible = false;
                    Scissor2.Visible = false;
                    Rock2.Visible = true;
                }

                if (player == npc)      // comparing values of decisions
                {
                    sc_player++;       // if they are equal, its draw. Both player
                    sc_npc++;          // get scores.
                }
                if (player == 0 && npc == 1)  // Scissor cuts the paper
                    sc_npc++;
                if (player == 0 && npc == 2)  // Paper wraps the rock
                    sc_player++;
                if (player == 1 && npc == 0)  // Scissor cuts the paper
                    sc_player++;
                if (player == 1 && npc == 2)  // Rock breaks the scissor
                    sc_npc++;
                if (player == 2 && npc == 0)  // Paper wraps the rock
                    sc_npc++;
                if (player == 2 && npc == 1)  // Rock breaks the scissor
                    sc_player++;

                label1.Text = "You: " + sc_player.ToString();
                label2.Text = "Johnny: " + sc_npc.ToString();

                if (sc_player == 10 || sc_npc == 10)   // When one of the each side
                {                                      // comes to ten, he wins !
                    DialogResult choosenbutton = MessageBox.Show("Do you want to play
again?", "Game Ended", MessageBoxButtons.YesNo);
                    if (choosenbutton == DialogResult.Yes)
                        Application.Restart();
                    else
                        Application.Exit();
                }
            }
        }

        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
                                           // Situations when player choose a style
            if (comboBox1.Text == "Paper")
            {
              Paper1.Visible = true;          // Player chooses Paper
              Scissor1.Visible = false;
              Rock1.Visible = false;
              player = 0;                     // And value is assigned to 0
            }
           if (comboBox1.Text == "Scissor")
           {
               Paper1.Visible = false;
               Scissor1.Visible = true;
               Rock1.Visible = false;
               player = 1;
           }
            if (comboBox1.Text == "Rock")
            {
                Paper1.Visible = false;
```

```
                Scissor1.Visible = false;
                Rock1.Visible = true;
                player = 2;
            }


        }
    }
}
```