CHALLENGE INNOVATE CONNECT

SOFE 3980:
Software Quality

Assignment 2: Automation Testing

Saleh Nawar | 100536488

Number Wizard Challenge

Introduction:

Number Wizard Challenge (NWC) is a simple 2d game that predicts any number between 1 to 1000 that the user chooses. The game achieves this by guessing a random number between two min and max bounds, then depending on the user input the game sets new bounds appropriate to the new parameters and repeats the process.

Software Development Model:

Due to the simplicity of the game and limited features, the best approach to develop the software would be a waterfall model. In This waterfall model, the development team will apply a straight through run to research, design, develop, and test the software.

Requirements:

NWC requires very limited resources due to its simplicity. Therefore, there is a wide array of options on which programing language or game development engine to use in order to implement this game. These options include developing the game using languages such Python, Java, C# or C++ as well as using game engines such as Unreal Engine or Unity. While Unreal Engine is more powerful and provides more features, Unity is more accessible and well suited to handle small games. Therefore, the best approach to developing NWC is using Unity Engine. Unity uses c# as a scripting language coupled with virtual studio as its Scripting IDE. Furthermore, newer versions of Unity implements Nunit framework for automation testing.

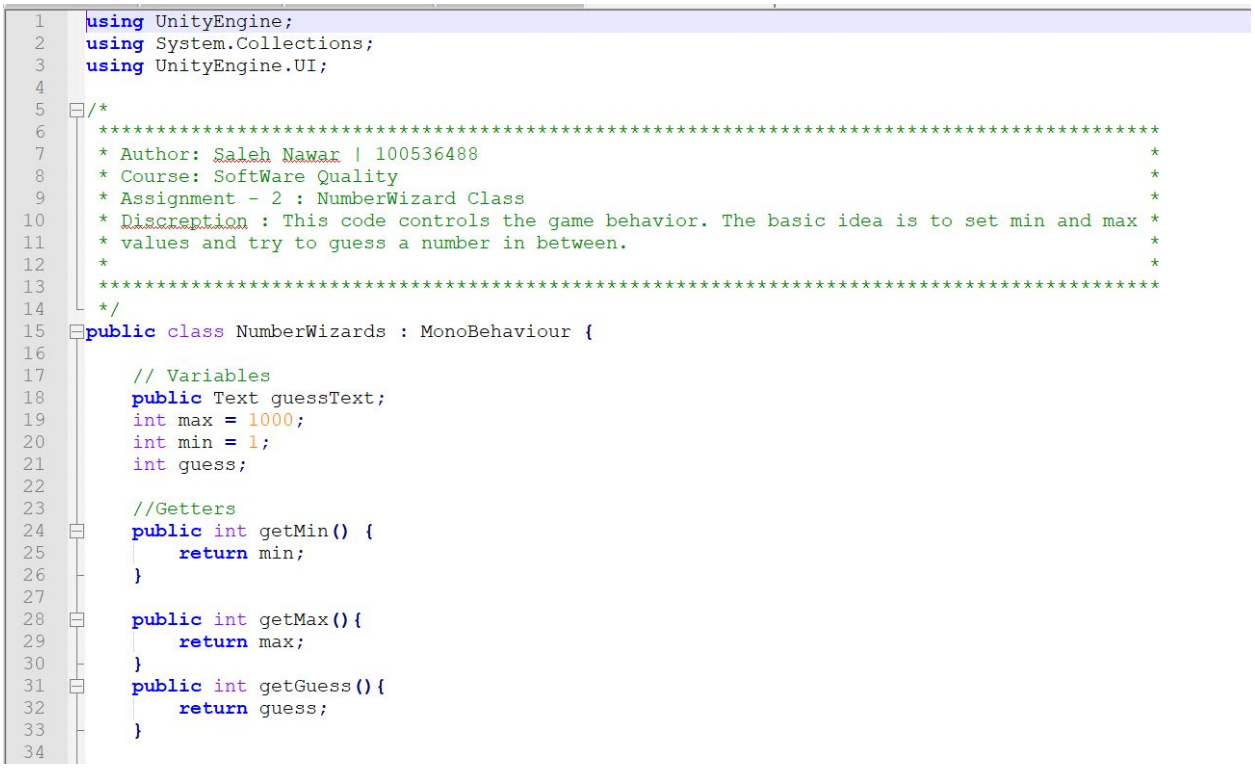
Design:

At its heart, NWC is basically a search algorithm that is searching for a number that the user chooses. This makes it very easy to implement. Initially,, the game will guess a random number between the two bounds and present it to the user, to which the user will input if the chosen number is higher, lower or equal to the guessed number. Afterwards, the game will use three functions, Guesshigher(), GuessLower() or GuessEqual() to react to the user input. The game will also use other functions such as StartGame() and GuessNext() to perform other functions.

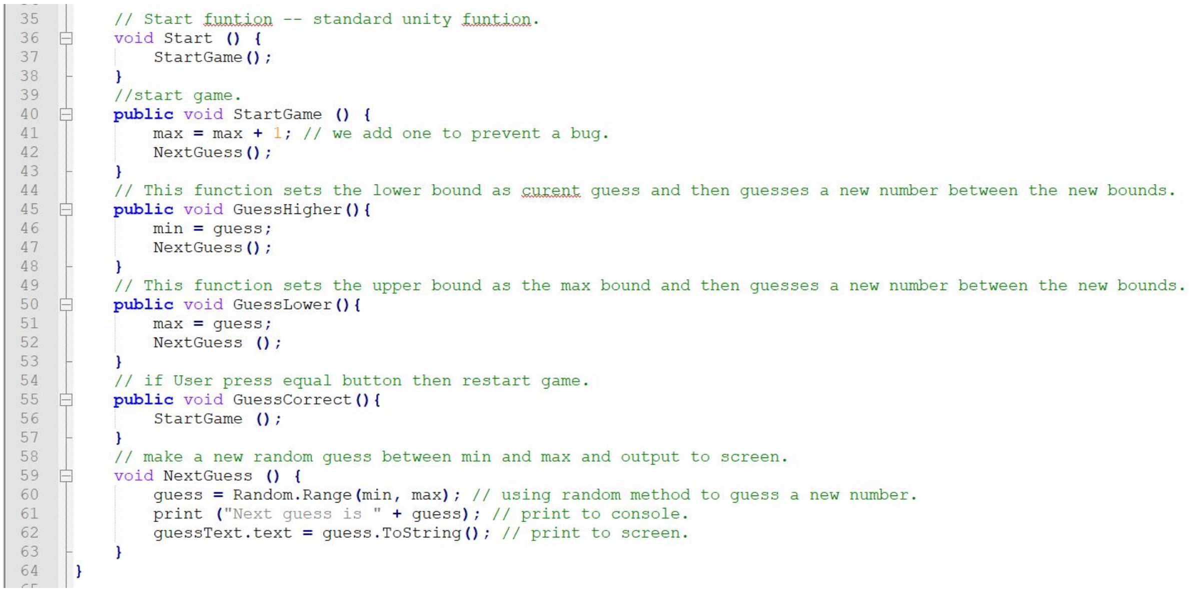
Implementation:

This section will only present the NumberWizard class which handles the main logic of the game.

Code: This code can be found in NumberWizardChallenge>Assets>Scripts NumberWizard.cs



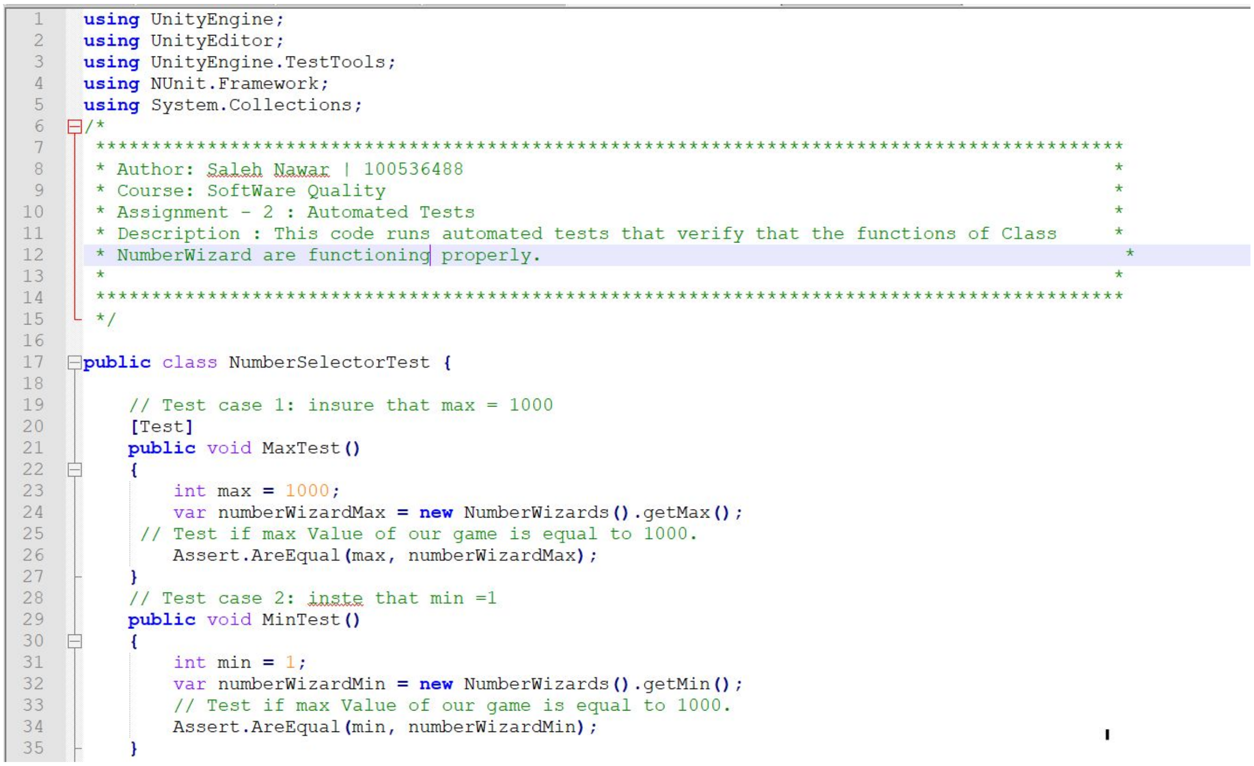
```
using UnityEngine;
using System.Collections;
using UnityEngine.UI;

/*
 ********************************************************************************************
 * Author: Saleh Nawar | 100536488                                                          *
 * Course: SoftWare Quality                                                                 *
 * Assignment - 2 : NumberWizard Class                                                      *
 * Discreption : This code controls the game behavior. The basic idea is to set min and max *
 * values and try to guess a number in between.                                             *
 *                                                                                          *
 ********************************************************************************************
 */
public class NumberWizards : MonoBehaviour {

    // Variables
    public Text guessText;
    int max = 1000;
    int min = 1;
    int guess;

    //Getters
    public int getMin() {
        return min;
    }

    public int getMax(){
        return max;
    }
    public int getGuess(){
        return guess;
    }

```



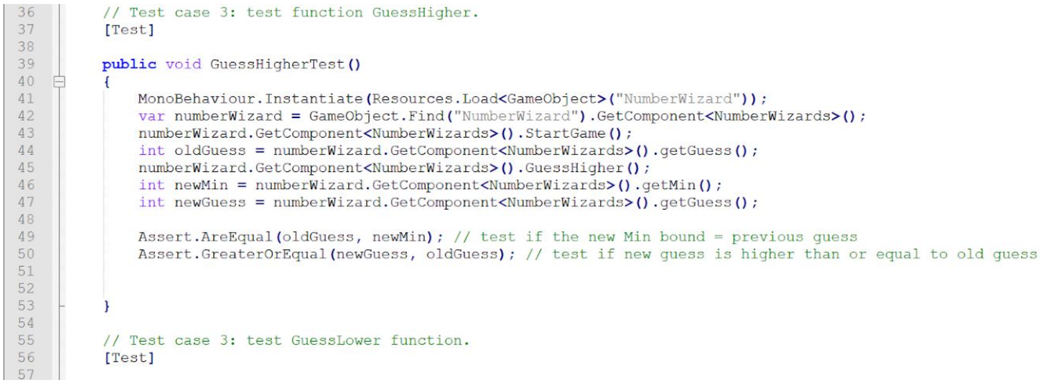
```
    // Start funtion -- standard unity funtion.
    void Start () {
        StartGame();
    }
    //start game.
    public void StartGame () {
        max = max + 1; // we add one to prevent a bug.
        NextGuess();
    }
    // This function sets the lower bound as curent guess and then guesses a new number between the new bounds.
    public void GuessHigher(){
        min = guess;
        NextGuess();
    }
    // This function sets the upper bound as the max bound and then guesses a new number between the new bounds.
    public void GuessLower(){
        max = guess;
        NextGuess ();
    }
    // if User press equal button then restart game.
    public void GuessCorrect(){
        StartGame ();
    }
    // make a new random guess between min and max and output to screen.
    void NextGuess () {
        guess = Random.Range(min, max); // using random method to guess a new number.
        print ("Next guess is " + guess); // print to console.
        guessText.text = guess.ToString(); // print to screen.
    }
}
```

Testing:

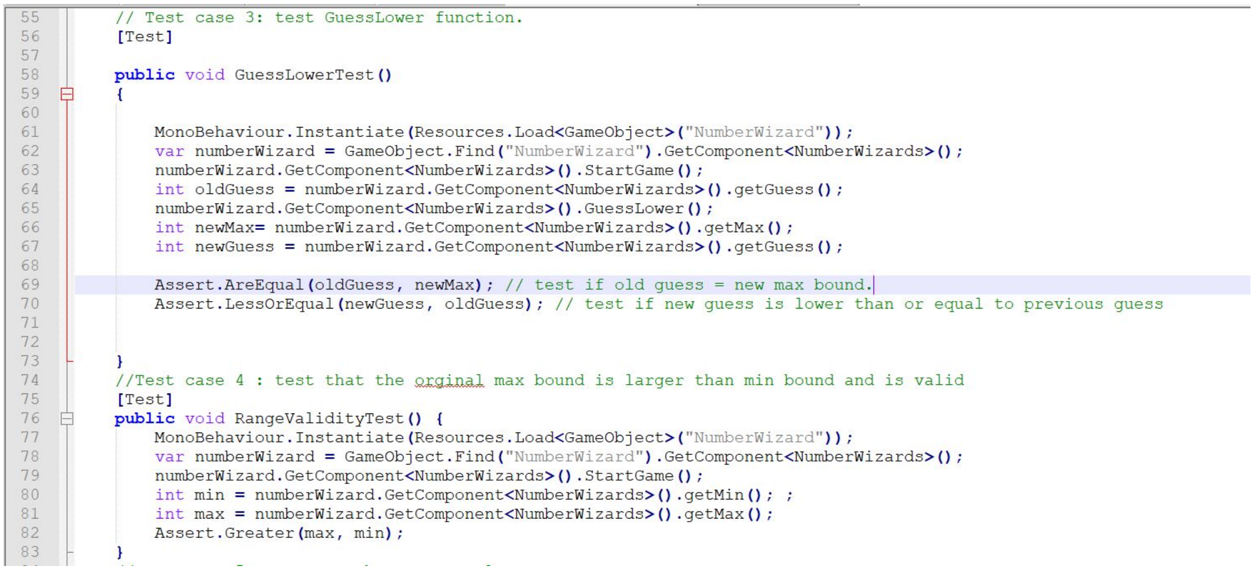
As mentioned previously unity uses Nunit framework for automation testing. The following class tests the functionality of various functions of NumberWizard.cs as well as and automation test of the game that assigns a random integer between 1 : 1000 and test if the game can guess that number. The following code can be found in NumberWizardChallenge>Assets>Editor NumberSelectorTest.cs



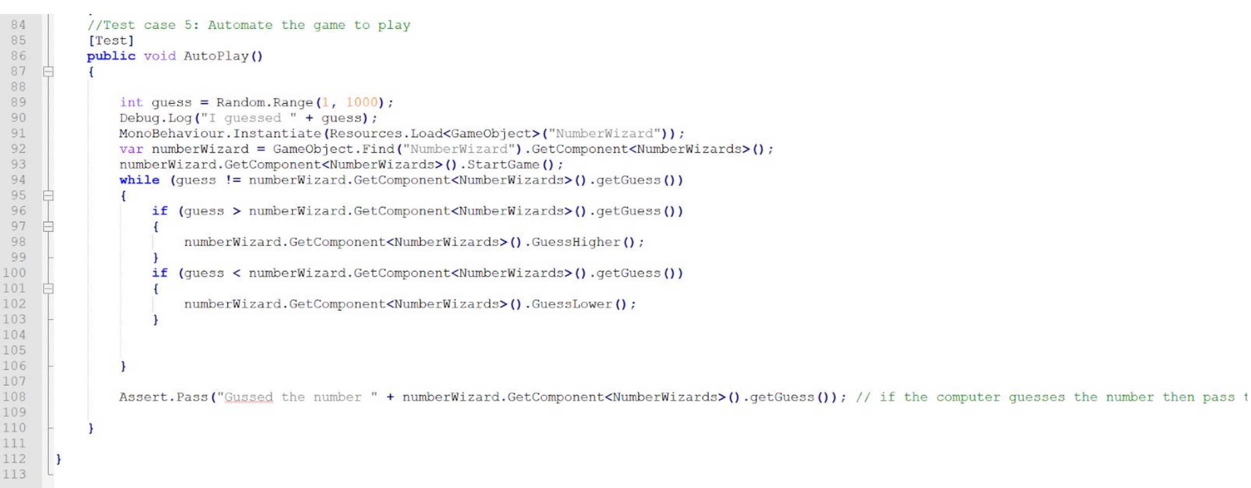
```
using UnityEngine;
using UnityEditor;
using UnityEngine.TestTools;
using NUnit.Framework;
using System.Collections;
/*
 *********************************************************************************************
 * Author: Saleh Nawar | 100536488                                                           *
 * Course: SoftWare Quality                                                                  *
 * Assignment - 2 : Automated Tests                                                          *
 * Description : This code runs automated tests that verify that the functions of Class     *
 * NumberWizard are functioning properly.                                                     *
 *                                                                                           *
 *********************************************************************************************
 */

public class NumberSelectorTest {

    // Test case 1: insure that max = 1000
    [Test]
    public void MaxTest()
    {
        int max = 1000;
        var numberWizardMax = new NumberWizards().getMax();
     // Test if max Value of our game is equal to 1000.
        Assert.AreEqual(max, numberWizardMax);
    }
    // Test case 2: inste that min =1
    public void MinTest()
    {
        int min = 1;
        var numberWizardMin = new NumberWizards().getMin();
        // Test if max Value of our game is equal to 1000.
        Assert.AreEqual(min, numberWizardMin);
    }
```



```
    // Test case 3: test function GuessHigher.
    [Test]

    public void GuessHigherTest()
    {
        MonoBehaviour.Instantiate(Resources.Load<GameObject>("NumberWizard"));
        var numberWizard = GameObject.Find("NumberWizard").GetComponent<NumberWizards>();
        numberWizard.GetComponent<NumberWizards>().StartGame();
        int oldGuess = numberWizard.GetComponent<NumberWizards>().getGuess();
        numberWizard.GetComponent<NumberWizards>().GuessHigher();
        int newMin = numberWizard.GetComponent<NumberWizards>().getMin();
        int newGuess = numberWizard.GetComponent<NumberWizards>().getGuess();

        Assert.AreEqual(oldGuess, newMin); // test if the new Min bound = previous guess
        Assert.GreaterOrEqual(newGuess, oldGuess); // test if new guess is higher than or equal to old guess


    }

    // Test case 3: test GuessLower function.
    [Test]
```

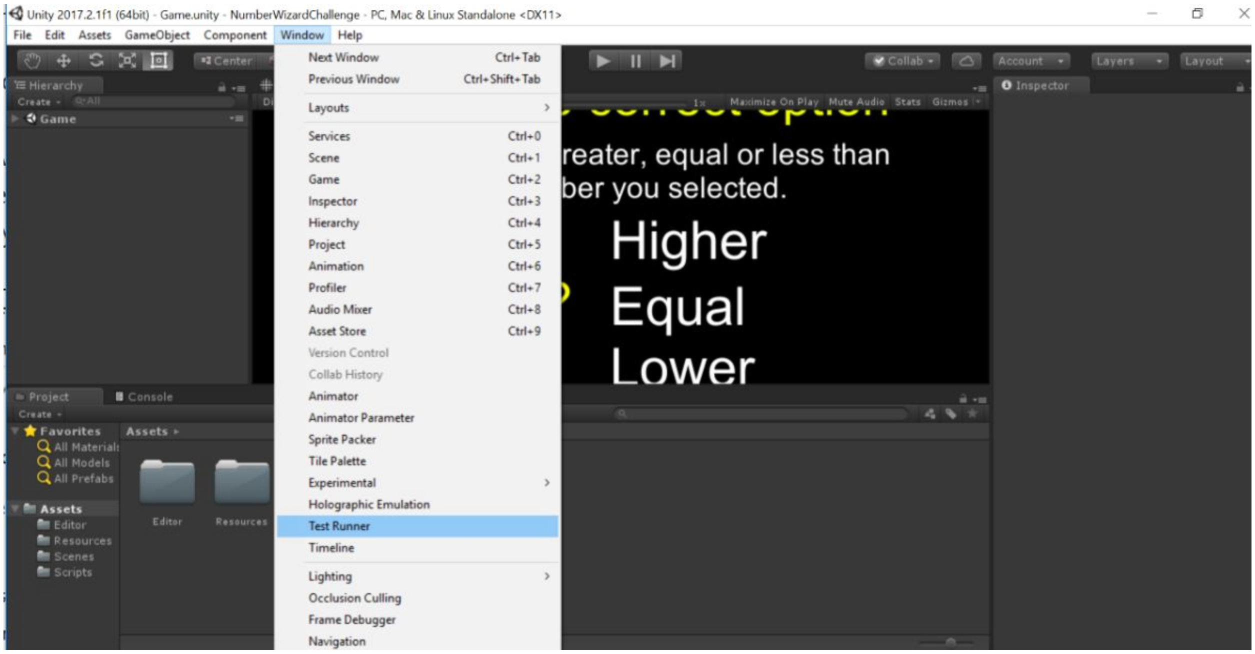


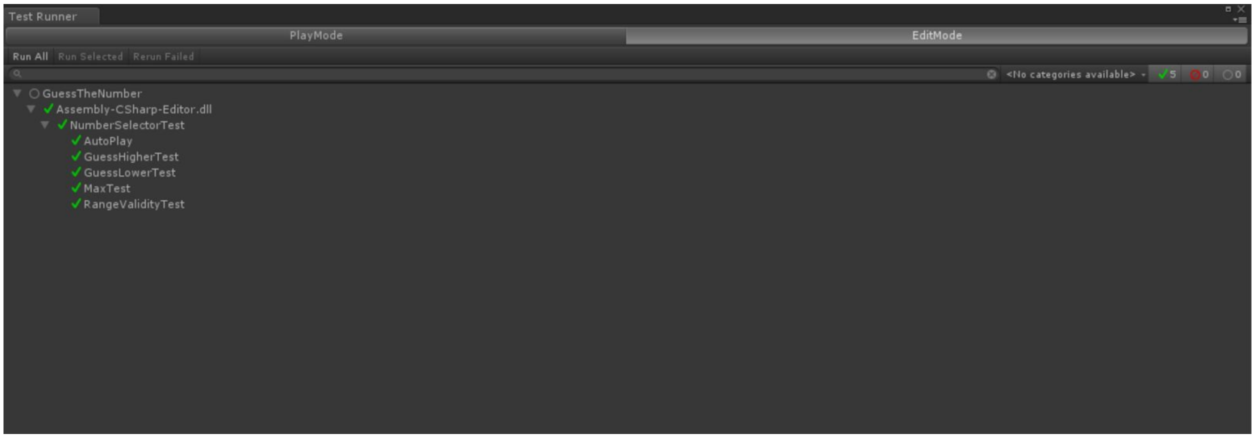
```
    // Test case 3: test GuessLower function.
    [Test]

    public void GuessLowerTest()
    {

        MonoBehaviour.Instantiate(Resources.Load<GameObject>("NumberWizard"));
        var numberWizard = GameObject.Find("NumberWizard").GetComponent<NumberWizards>();
        numberWizard.GetComponent<NumberWizards>().StartGame();
        int oldGuess = numberWizard.GetComponent<NumberWizards>().getGuess();
        numberWizard.GetComponent<NumberWizards>().GuessLower();
        int newMax= numberWizard.GetComponent<NumberWizards>().getMax();
        int newGuess = numberWizard.GetComponent<NumberWizards>().getGuess();

        Assert.AreEqual(oldGuess, newMax); // test if old guess = new max bound.
        Assert.LessOrEqual(newGuess, oldGuess); // test if new guess is lower than or equal to previous guess


    }
    //Test case 4 : test that the orginal max bound is larger than min bound and is valid
    [Test]
    public void RangeValidityTest() {
        MonoBehaviour.Instantiate(Resources.Load<GameObject>("NumberWizard"));
        var numberWizard = GameObject.Find("NumberWizard").GetComponent<NumberWizards>();
        numberWizard.GetComponent<NumberWizards>().StartGame();
        int min = numberWizard.GetComponent<NumberWizards>().getMin(); ;
        int max = numberWizard.GetComponent<NumberWizards>().getMax();
        Assert.Greater(max, min);
    }
```



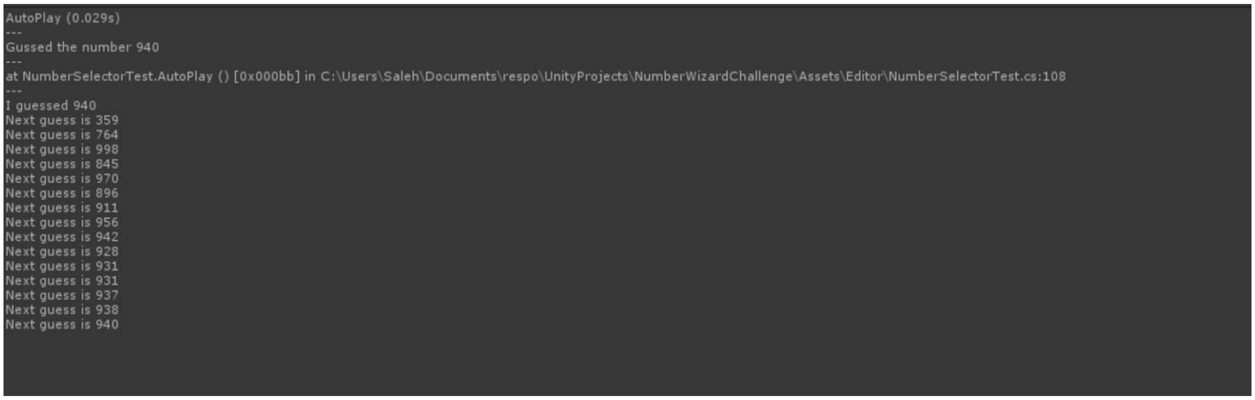
```
    //Test case 5: Automate the game to play
    [Test]
    public void AutoPlay()
    {

        int guess = Random.Range(1, 1000);
        Debug.Log("I guessed " + guess);
        MonoBehaviour.Instantiate(Resources.Load<GameObject>("NumberWizard"));
        var numberWizard = GameObject.Find("NumberWizard").GetComponent<NumberWizards>();
        numberWizard.GetComponent<NumberWizards>().StartGame();
        while (guess != numberWizard.GetComponent<NumberWizards>().getGuess())
        {
            if (guess > numberWizard.GetComponent<NumberWizards>().getGuess())
            {
                numberWizard.GetComponent<NumberWizards>().GuessHigher();
            }
            if (guess < numberWizard.GetComponent<NumberWizards>().getGuess())
            {
                numberWizard.GetComponent<NumberWizards>().GuessLower();
            }


        }

        Assert.Pass("Gussed the number " + numberWizard.GetComponent<NumberWizards>().getGuess()); // if the computer guesses the number then pass t

    }

}
```

Automated test requires using Unity Editior which can be installed for free. I’m using Unity 2017 to run this test. To run the test Open the NumberWizardChallenge project using Unity Editor 2017 then click on Window>TestRunner -> run all
Download Link of Unity: https://store.unity.com/download?ref=personal





```
○ GuessTheNumber
  ✓ Assembly-CSharp-Editor.dll
    ✓ NumberSelectorTest
      ✓ AutoPlay
      ✓ GuessHigherTest
      ✓ GuessLowerTest
      ✓ MaxTest
      ✓ RangeValidityTest
```



```
AutoPlay (0.029s)
---
Gussed the number 940
---
at NumberSelectorTest.AutoPlay () [0x000bb] in C:\Users\Saleh\Documents\respo\UnityProjects\NumberWizardChallenge\Assets\Editor\NumberSelectorTest.cs:108
---
I guessed 940
Next guess is 359
Next guess is 764
Next guess is 998
Next guess is 845
Next guess is 970
Next guess is 896
Next guess is 911
Next guess is 956
Next guess is 942
Next guess is 928
Next guess is 931
Next guess is 931
Next guess is 937
Next guess is 938
Next guess is 940
```

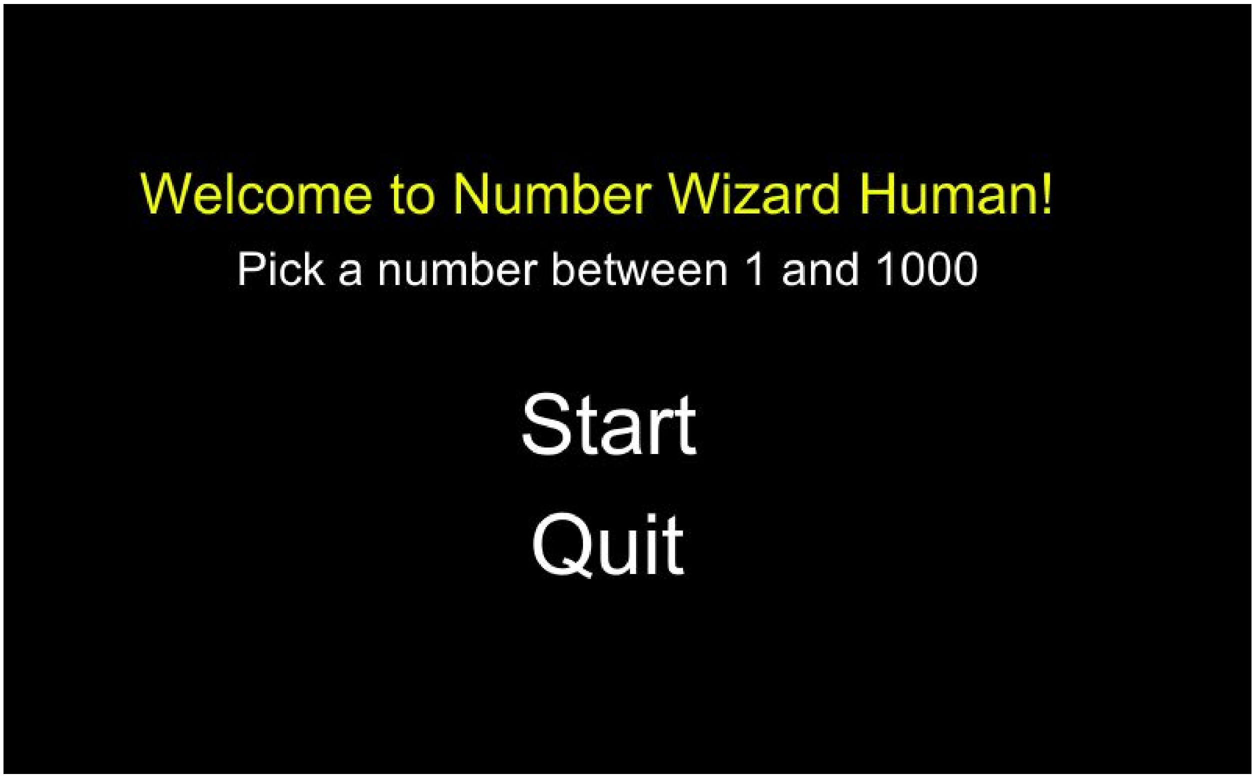
Found Bugs:

1. Found a bug where the NWC will guess a number out of bounds (Fixed).
2. Found a bug where NWC GuessHigher() had inncorect implementation causing it to choose guess as max value instead of min. (Fixed)
3. Found a bug where NWC will repeat the same value for the guessed number due to the random method of nominating a new guess. As the range gets smaller, the probability of the bug occuring increases.(Not Fixed - Requires re-implementing GuessNext())

Executable:

Executable can be found in NumberWizardChallenge> NWC.exe

The Following run code is a sample run where I guessed the number 467



Welcome to Number Wizard Human!

Pick a number between 1 and 1000

Select the correct option

Is this number greater, equal or less than the number you selected.

Higher

529 Equal

Lower

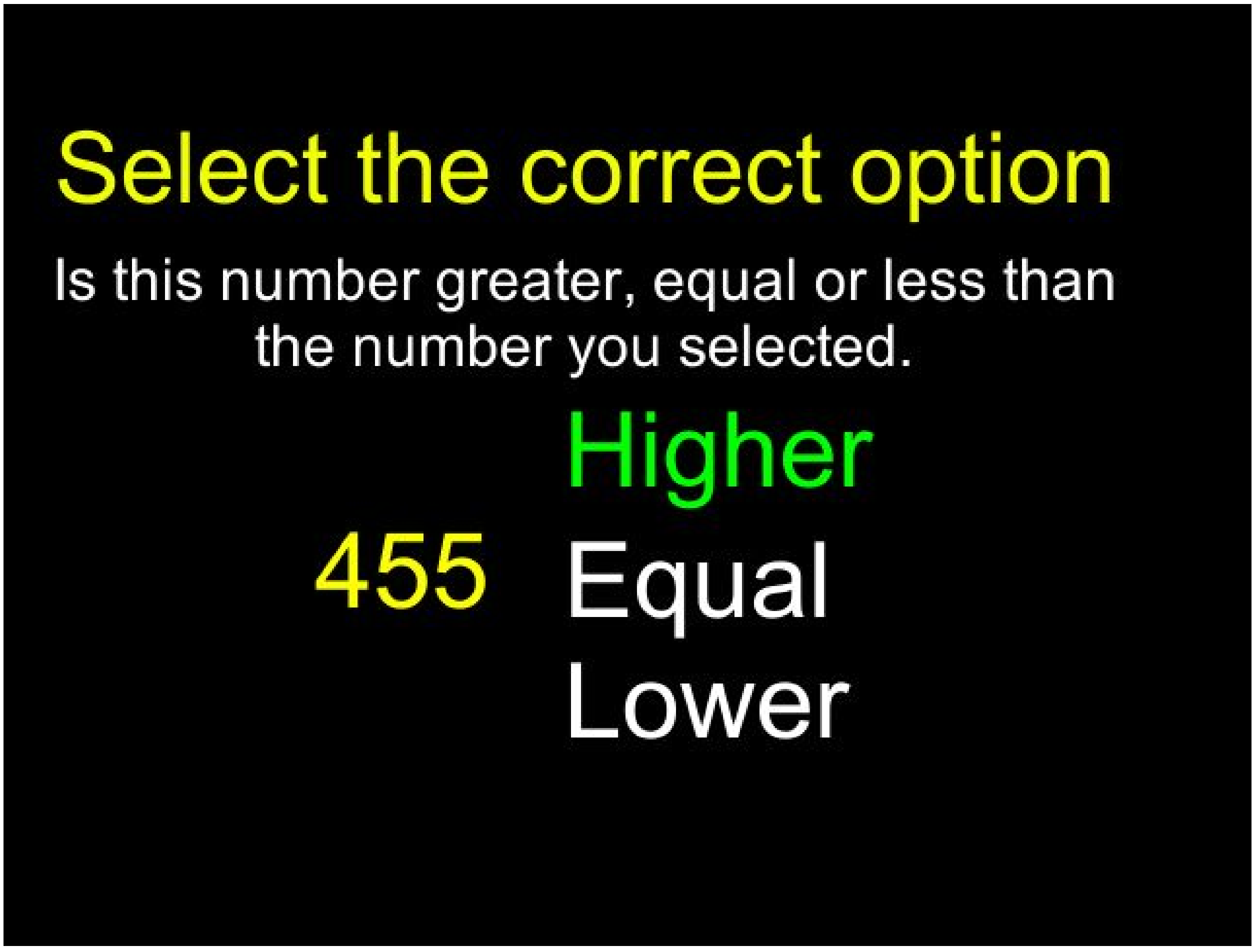
Select the correct option

Is this number greater, equal or less than the number you selected.

[✓ Higher]

105 Equal

Lower



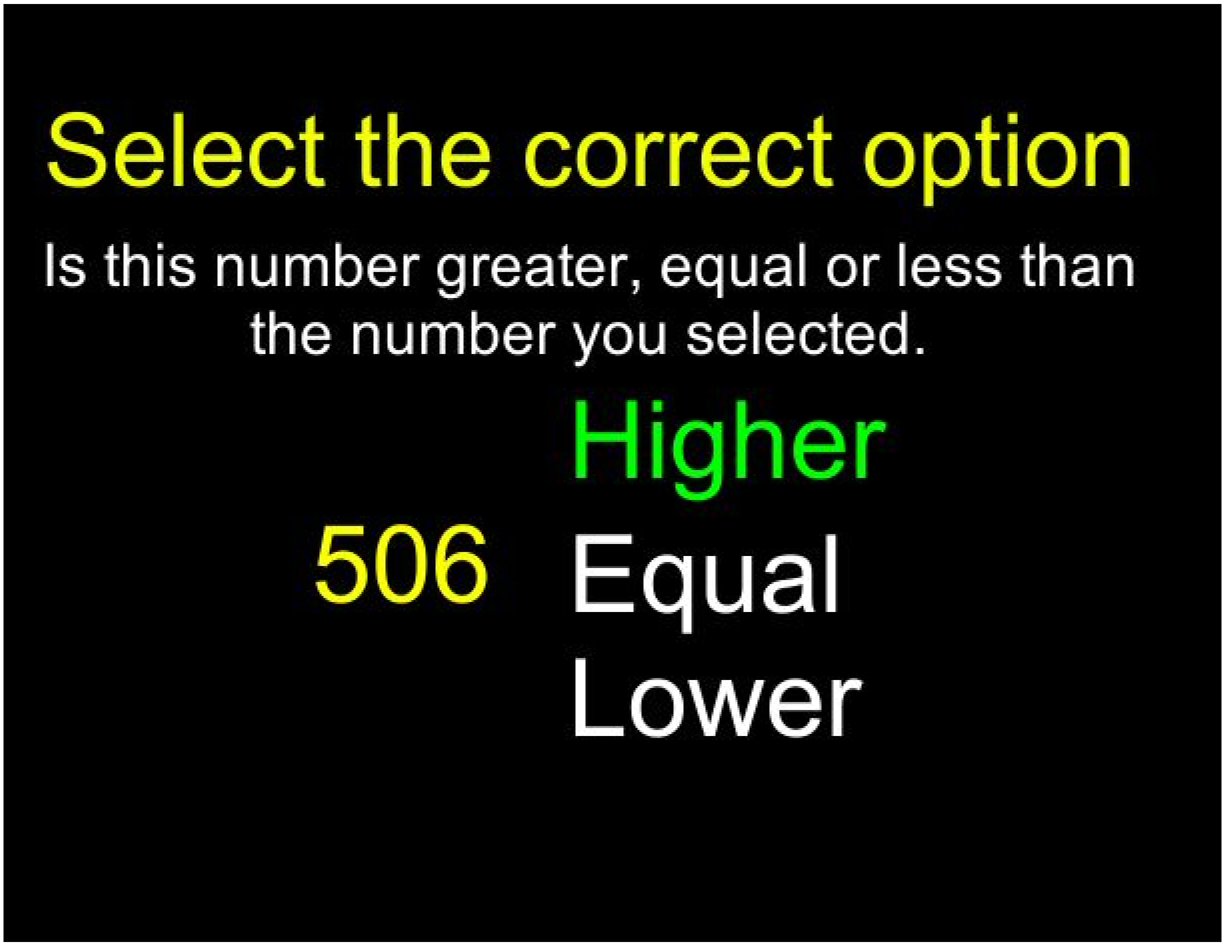
Select the correct option

Is this number greater, equal or less than the number you selected.

Higher

455 Equal

Lower



Select the correct option

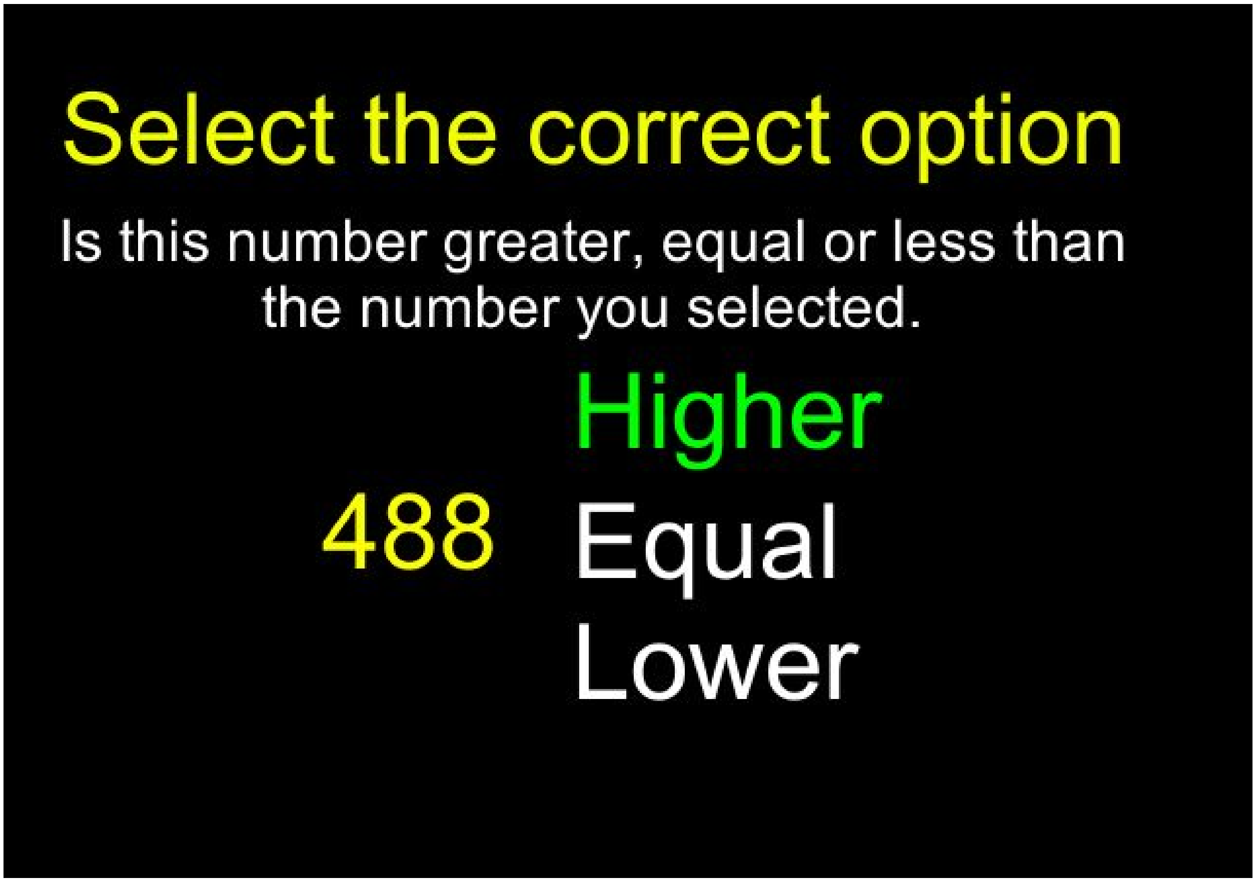
Is this number greater, equal or less than the number you selected.

506

Higher

Equal

Lower

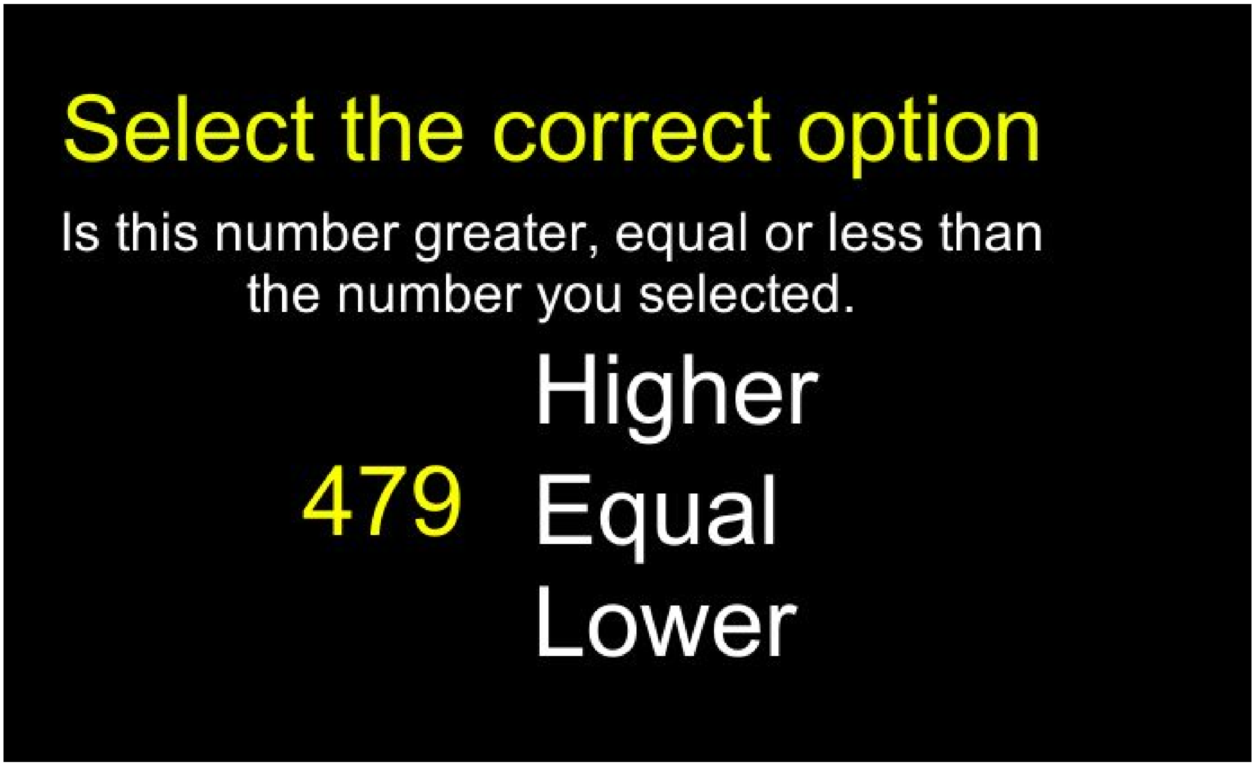


Select the correct option

Is this number greater, equal or less than the number you selected.

488

Higher
Equal
Lower

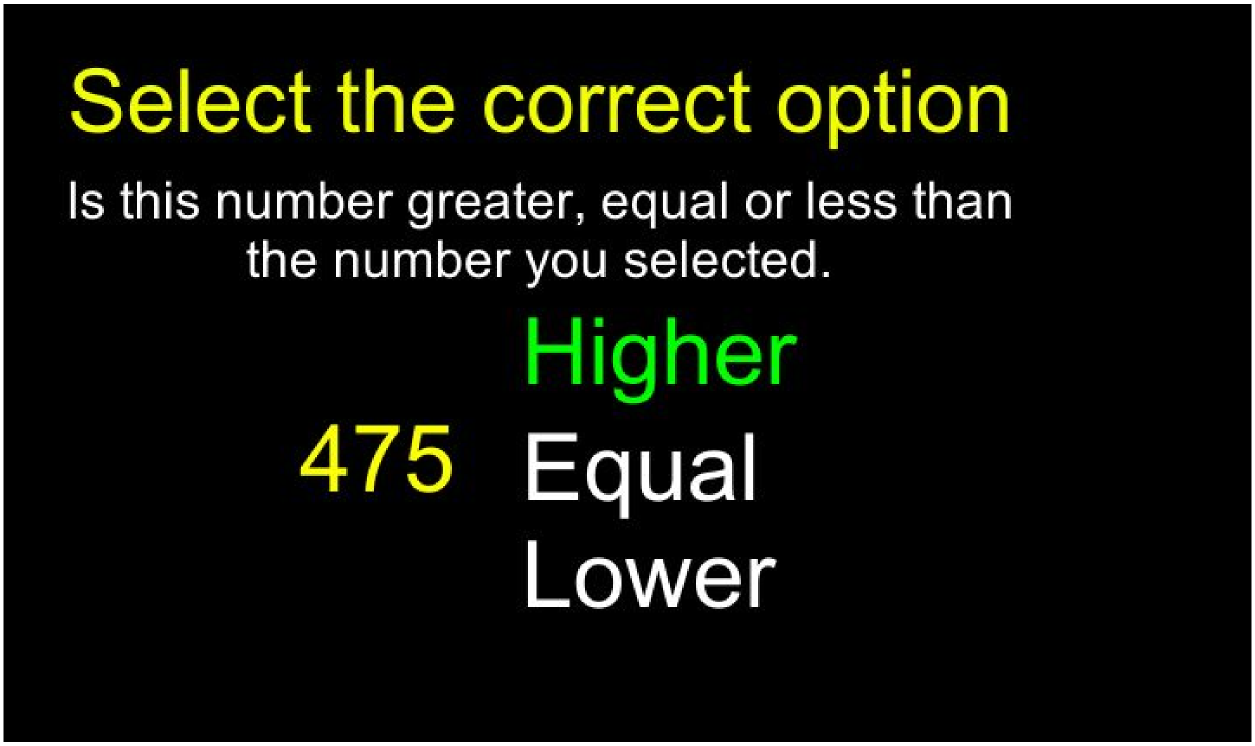


Select the correct option

Is this number greater, equal or less than the number you selected.

479

Higher
Equal
Lower



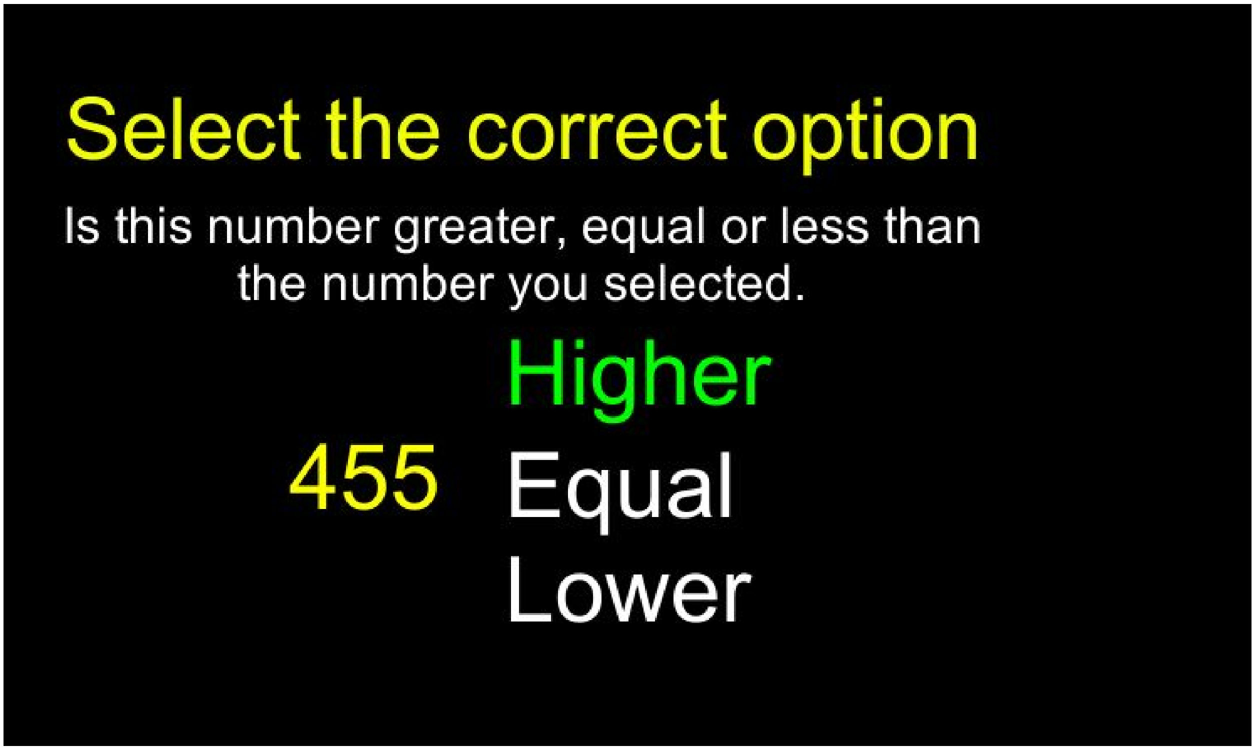
Select the correct option

Is this number greater, equal or less than the number you selected.

Higher

475 Equal

Lower



Select the correct option

Is this number greater, equal or less than the number you selected.

Higher

455 Equal

Lower

Select the correct option

Is this number greater, equal or less than
the number you selected.

461

Higher

Equal

Lower

Select the correct option

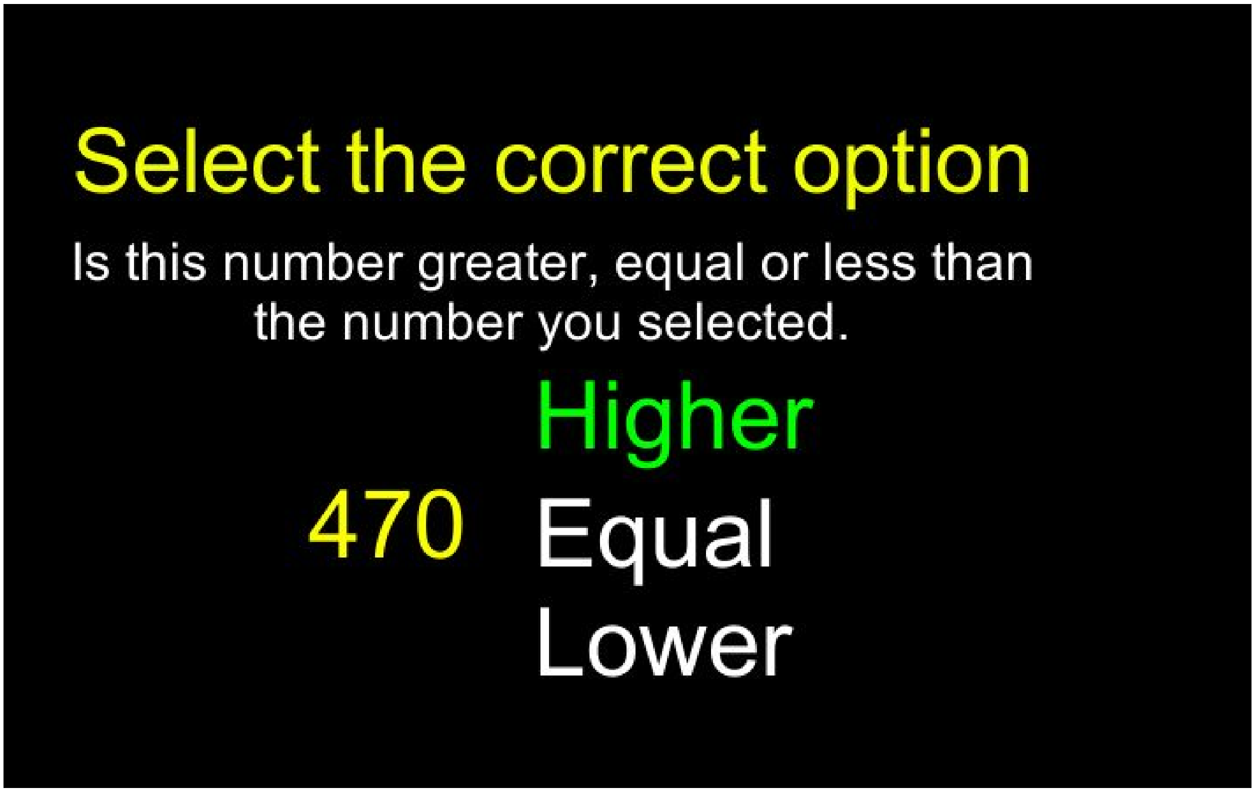
Is this number greater, equal or less than
the number you selected.

466

Higher

Equal

Lower



Select the correct option

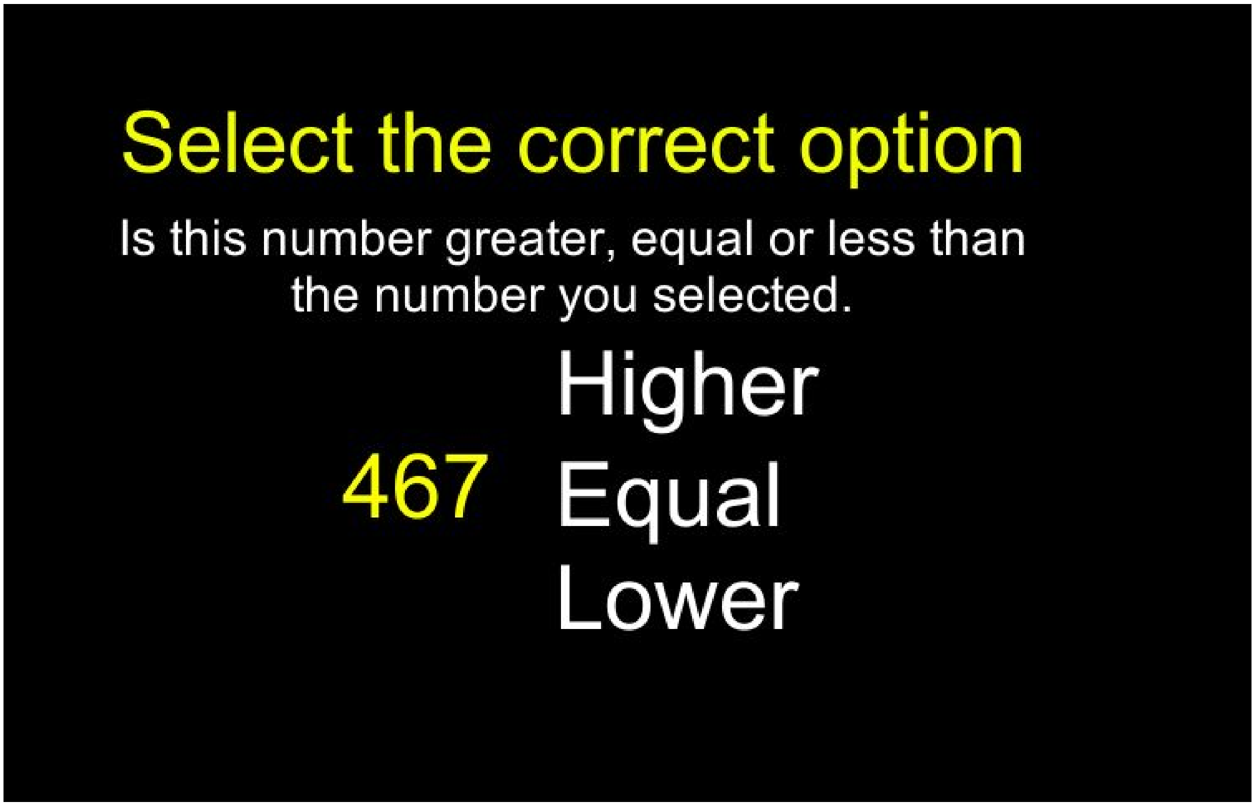
Is this number greater, equal or less than the number you selected.

470

Higher

Equal

Lower

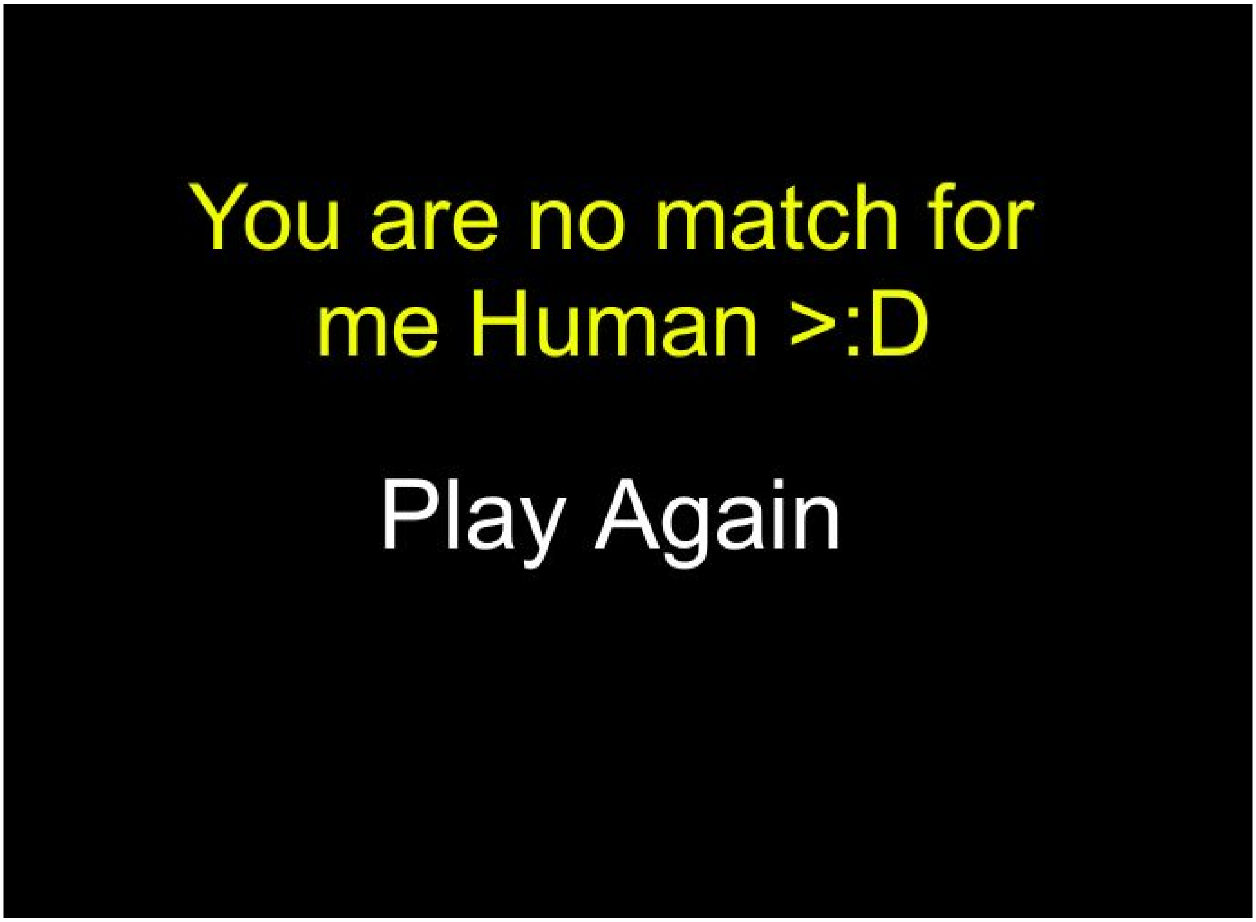


Select the correct option

Is this number greater, equal or less than
the number you selected.

467

Higher
Equal
Lower



You are no match for me Human >:D