# Task 1

An event driven program or an EDP is a type of program that the flow of the program is determined by events that a user would do. An event, for example, would be a person click on a button or selecting something from a menu, this event would trigger another piece of code in the program to run, this is why it is called an event driven program.

There are four key parts of an event driven program; the form, which is where all the controls (buttons, icons and objects) are stored, and each control can have an event; the event loop, which is a piece of code that runs in the background, that constantly checks for events and runs in a loop until an event has happened; the trigger function, which is used by event loops to identify which code to run when a particular event happens and the event handlers which is the part of the code that runs once the particular event has occurred.

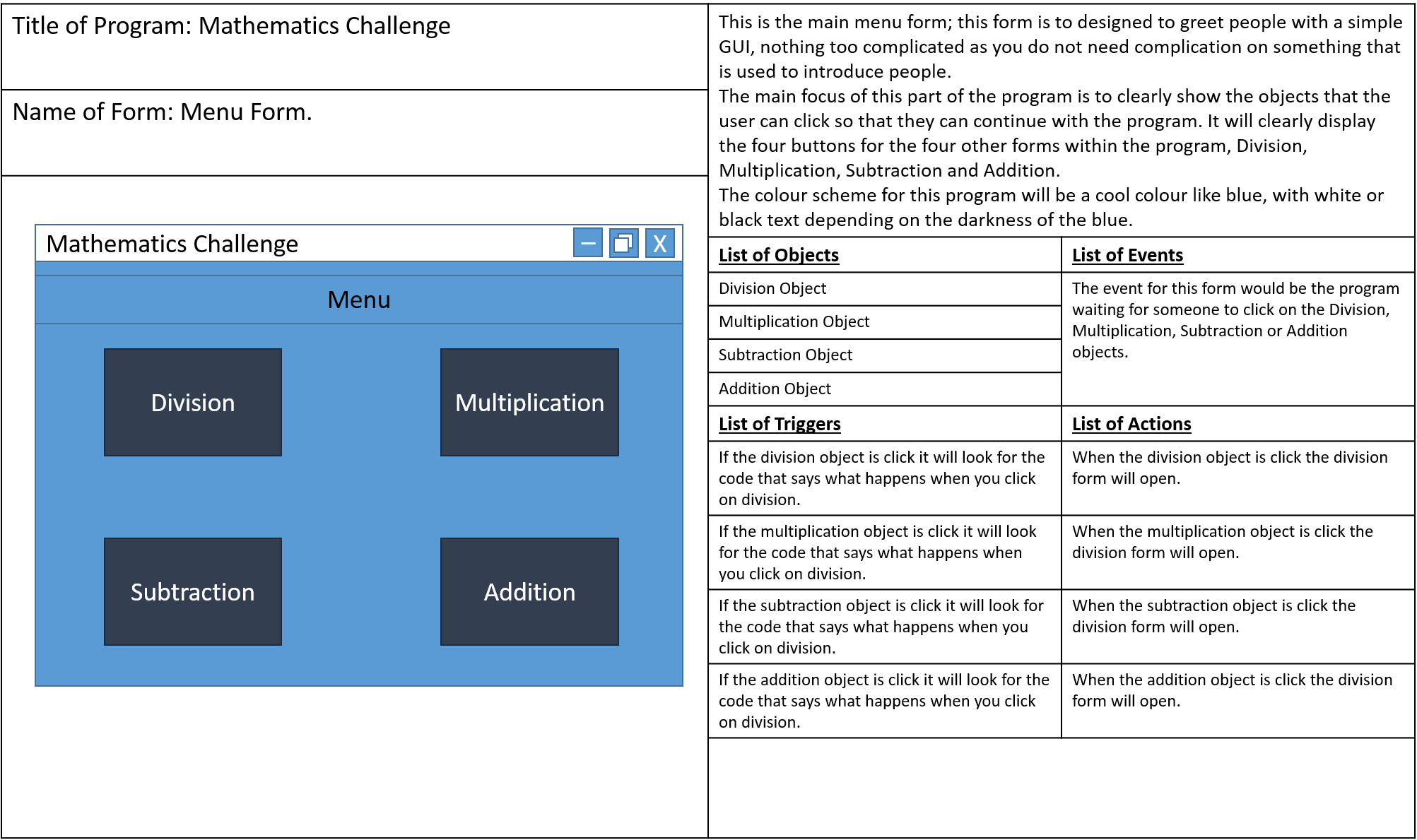
An GUI operating system, like Windows, runs in exactly the same way; even the non-graphical parts of the operating system is a part of the EDF. In terms of Windows the form would be the GUI interface itself; when you first load the operating system the program loads the Windows Explorer (explorer.exe) which is the form in windows where all the buttons, icons and objects (controls) are stored.

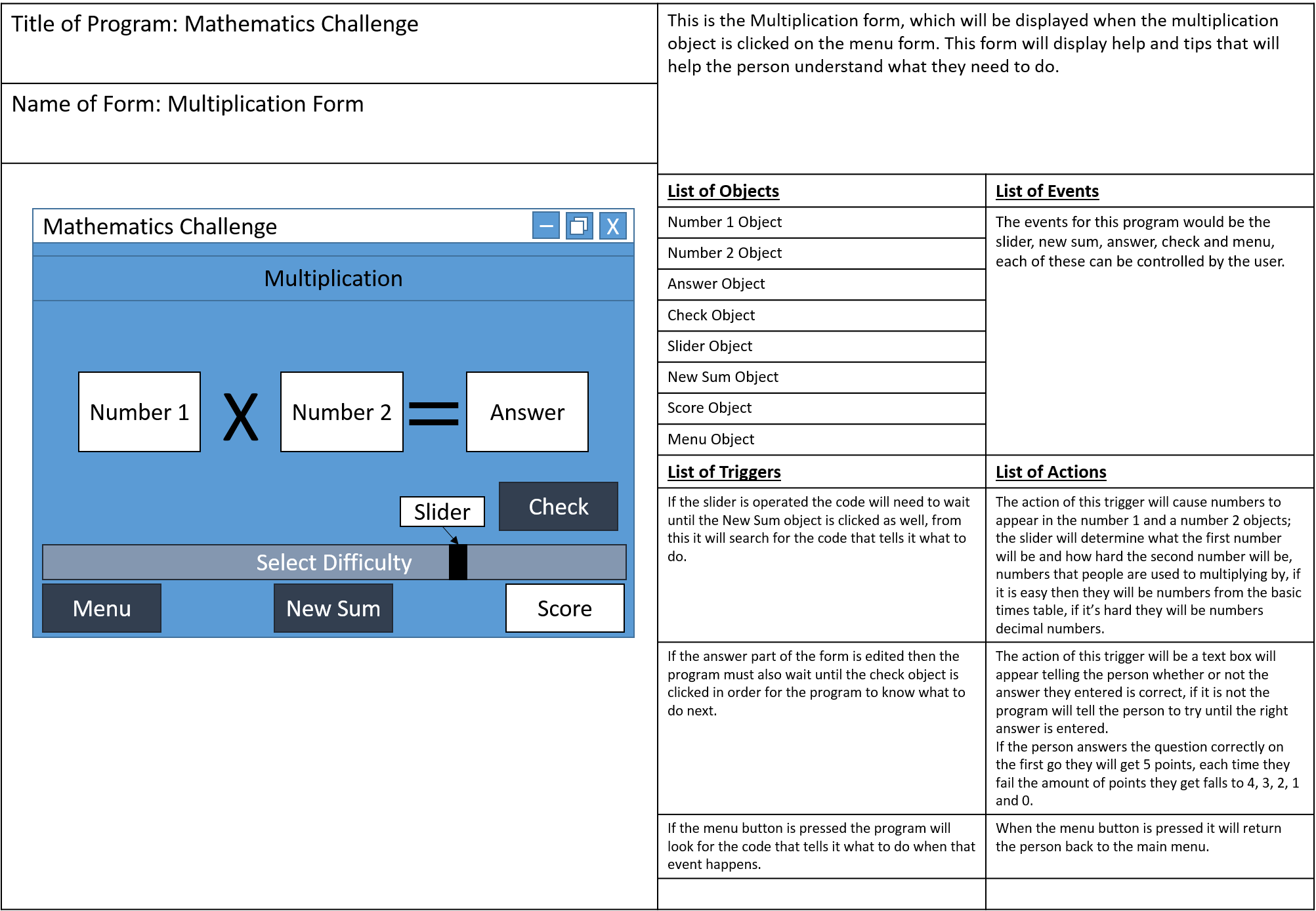


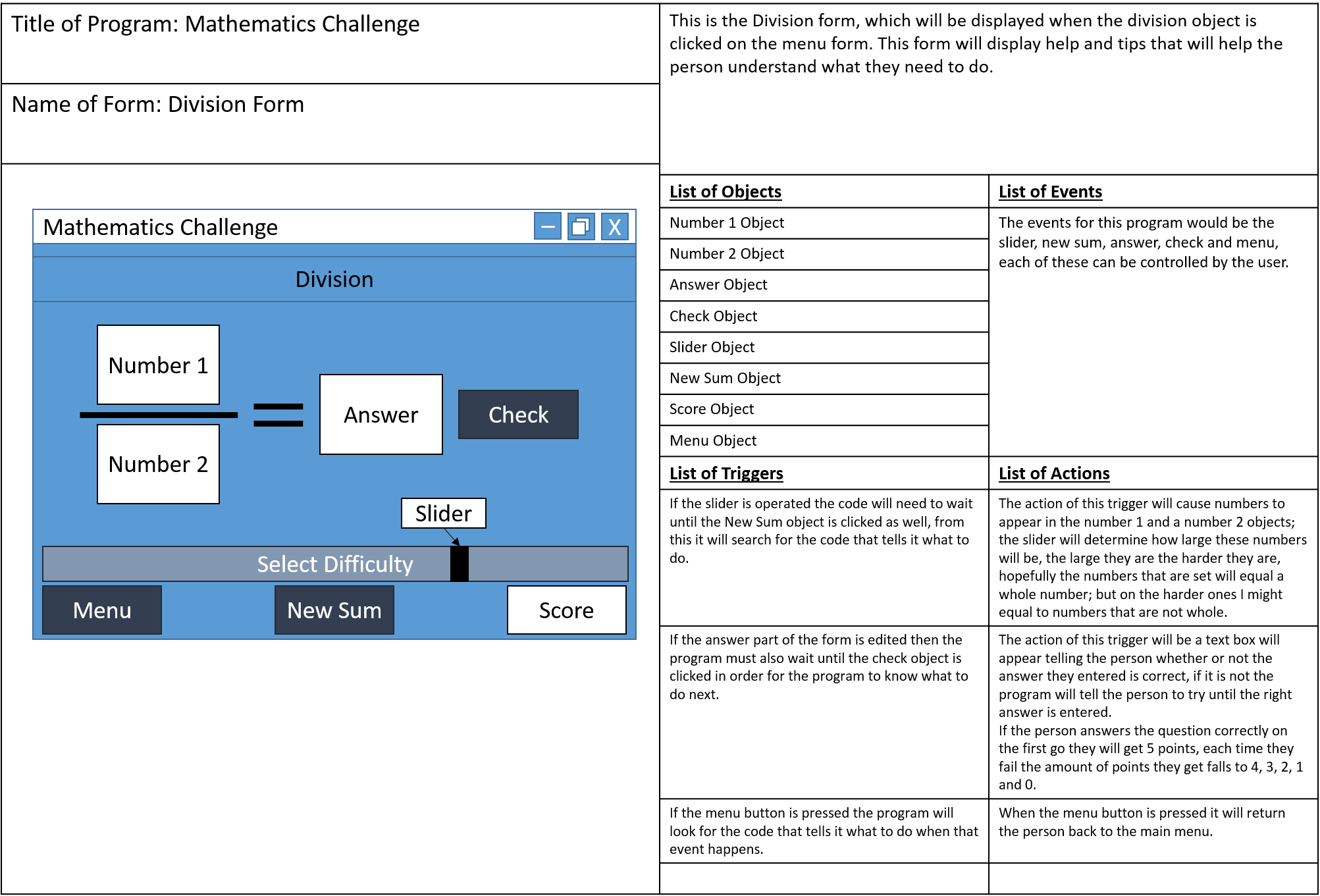
Figure 1: Controls of the Windows GUI

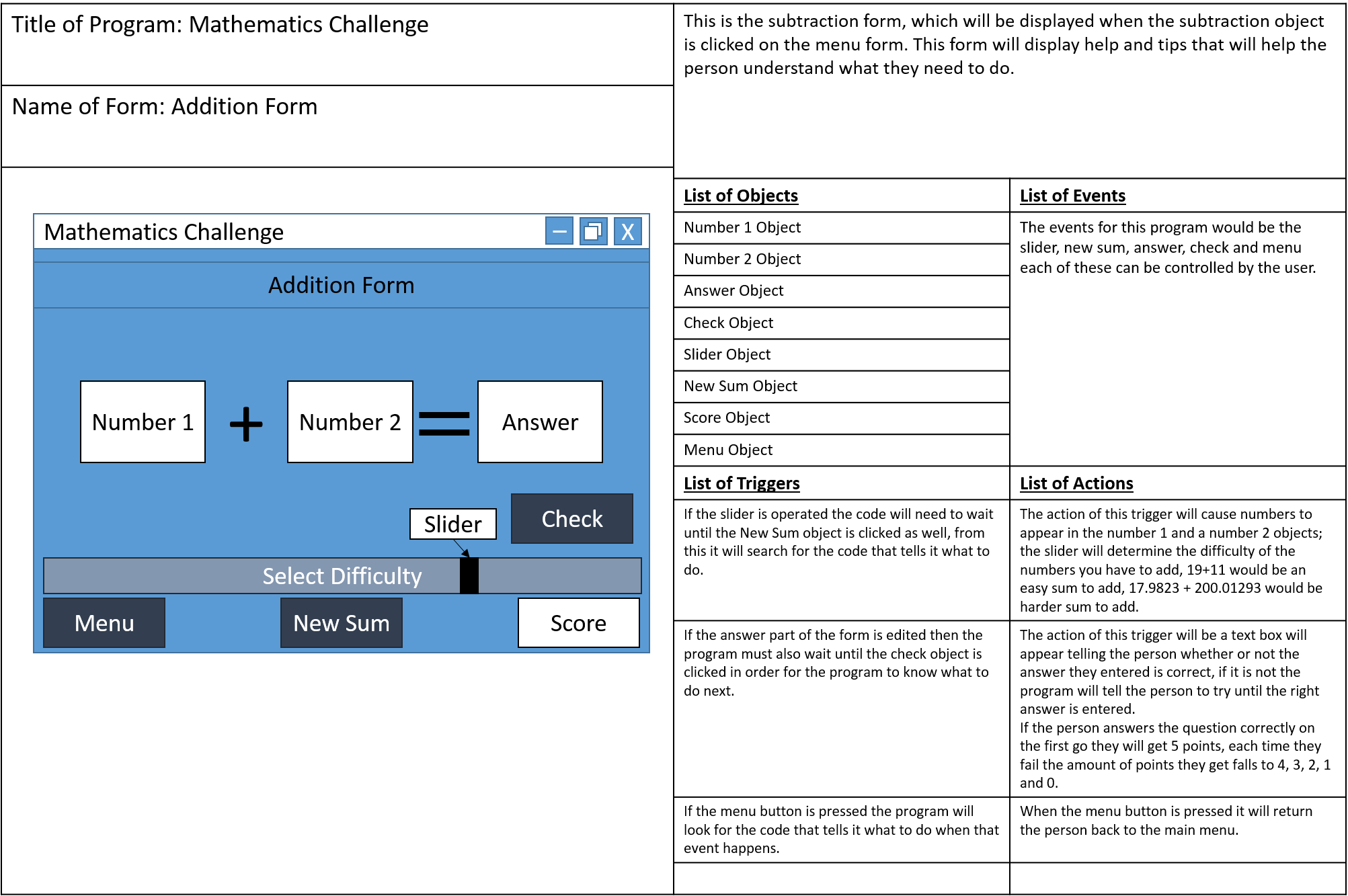
The Windows operating system also has an event loop, Windows will run a loop and wait until the user does something, this will run in the background until the system itself is turned off. Windows will also have a trigger function that will be use to find a piece of code to run when a program, or a button is pressed; for example, when the user clicks on the start menu button, the trigger will then look for the piece of code that needs to run when the start menu button is pressed, this is an event. The final part that Windows will do is run the code that commands the start menu, this is the event handler, and is also the fourth stage of the EDP.

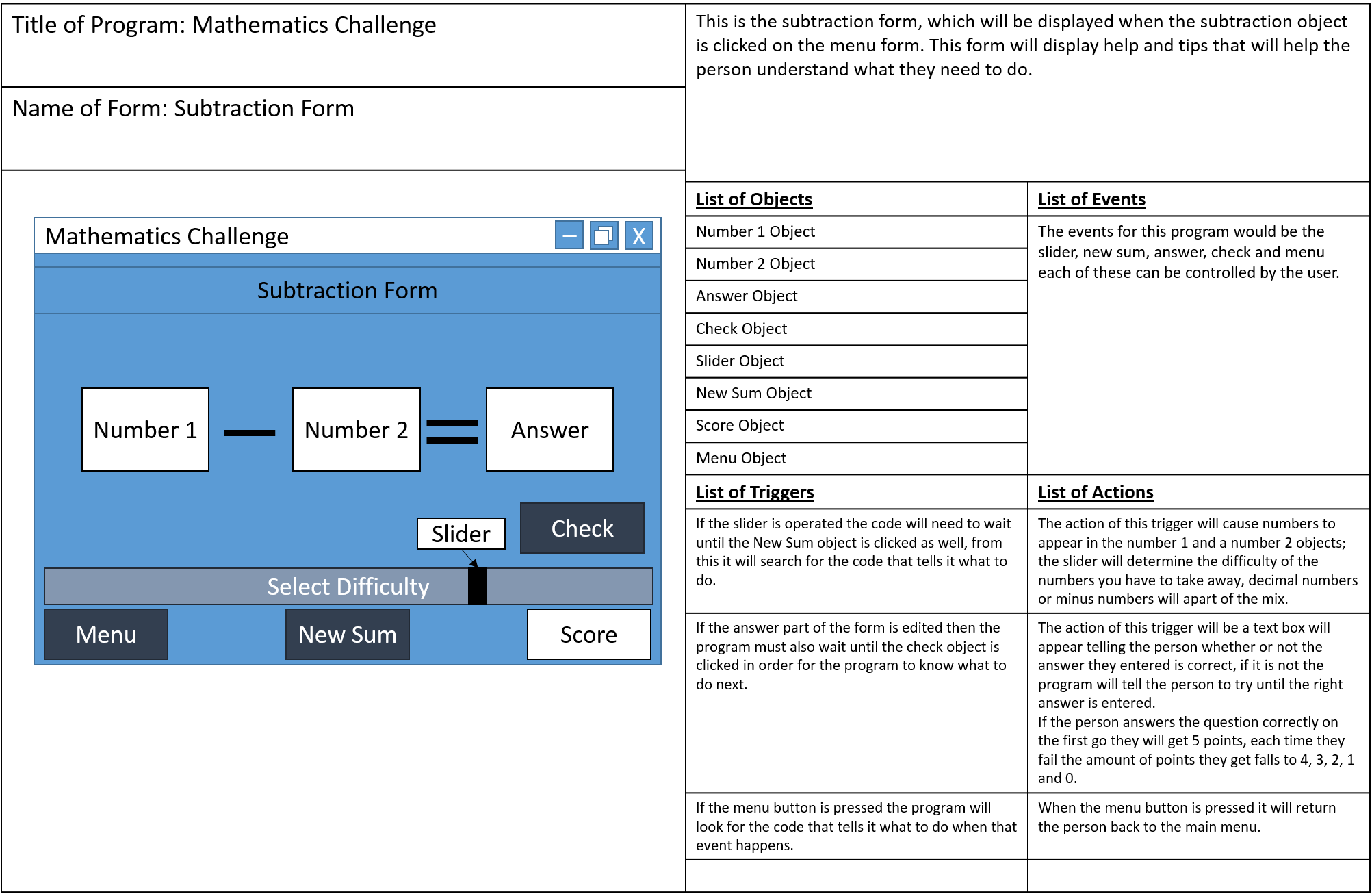
# Task 2 – Design of Program











# Pseudo Code

## Part 1

Public Class frmMainMenu

'Menu Form

'Addition From

'Subtraction From

'Multiplication Form

'Division Form

End Class

## Part 2

'Menu Form

Public Class frmMainMenu

Private Sub frmMainMenu\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

'Addition Form Button

Private Sub btnAddition\_Click(sender As Object, e As EventArgs) Handles btnAddition.Click

Me.Hide() 'Hide Menu Form

Addition.Show() 'Show Addition Form

End Sub

'Subtraction Form Button

Private Sub btnSubtraction\_Click(sender As Object, e As EventArgs) Handles btnSubtraction.Click

Me.Hide() 'Hide Menu Form

Subtraction.Show() 'Show Subtraction Form

End Sub

'Multiplication Form Button

Private Sub btnMulitplication\_Click(sender As Object, e As EventArgs) Handles btnMulitplication.Click

Me.Hide() 'Hide Menu Form

Multiplication.Show() 'Show Multiplication Form

End Sub

'Divison Form Button

Private Sub btnDivision\_Click(sender As Object, e As EventArgs) Handles btnDivision.Click

Me.Hide() 'Hide Menu Form

Division.Show() 'Show Division Form

End Sub

'Exit Form Button

Private Sub btnExit\_Click(sender As Object, e As EventArgs) Handles btnExit.Click

'Ask person if they are sure that they want to quit

Dim result As Integer = MessageBox.Show("Are you sure you want to close?", "Attention", MessageBoxButtons.YesNo, MessageBoxIcon.Question)

'If yes is pressed then close, if no is pressed tehn stay open.

If result = DialogResult.Yes Then

Close() 'Closes this form

Addition.Close() 'Close Addition if hidden

Subtraction.Close() 'Close Subtraction if Hidden

Multiplication.Close() 'Close Multiplication if hidden

Division.Close() 'Close Division if Open

ElseIf result = DialogResult.No Then

End If

End Sub

End Class

# Test Plan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Test No* | *Run No* | *Test Description* | *Expected Outcome* | *Actual Outcome* | *Action Required* |
| 1 | 1 | Menu Form – To test if the addition button in the form works as it should, the button should open another form within the program and close the one that it was on. | The Addition Button, when clicked should cause the menu form to hide and should open the Addition Form. |  |  |
| 2 | 1 | Menu Form – To test if the subtraction button in the form works as it should, the button should open another form within the program and close the one that it was on. | The subtraction button, when clicked, should cause the menu form to hide and should open the subtraction form. |  |  |
| 3 | 1 | Menu Form – To test if the multiplication button in the form works as it should, the button should open another form within the program and close the one that it was on. | The multiplication button, when clicked, should cause the menu form to hide and should open the multiplication form. |  |  |
| 4 | 1 | Menu Form – To test if the division button in the form works as it should, the button should open another form within the program and close the one that it was on. | The division button, when clicked should cause the menu form to hide and open the division form. |  |  |
| 5 | 1 | Menu Form – To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits yes then the program should close. |  |  |
| 6 | 1 | Menu Form – To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits no the program should remain open. |  |  |
| 7 | 1 | Addition form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far left as it can go | Small, simple numbers should appear in txtNum1 and txtNum2 that makes an easy sum to complete. |  |  |
| 8 | 1 | Addition form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is in the middle | More complex numbers should appear, numbers with at least two decimal places should appear |  |  |
| 9 | 1 | Addition form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far right as it can go | Complex numbers with at least three decimal places should appear, making it the hardest to calculate the sum. |  |  |
| 10 | 1 | Addition form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is correct then a dialogue box will appear saying that the answer is correct, the score should increase by one. |  |  |
| 11 | 1 | Addition form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is incorrect then a dialogue box will appear say that the answer in incorrect and tells you try again, the score will increase by 0. |  |  |
| 12 | 1 | Addition form – Error Handling – What happens if a complex character was entered into the txtNum1 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 13 | 1 | Addition form – Error Handling – What happens if a complex character was entered into the txtNum2 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 14 | 1 | Addition form – Error Handling – What happens if a complex character was entered into the txtAnswer and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 15 | 1 | Addition form – Error Handling – What happens if a decimal number was entered into the box during an easy sum generation where a decimal number cannot be an answer. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 16 | 1 | Addition form – Error Handling – What happens if a complex character was entered into the txtScore, being a text box people will be able to put a value into it. | If the answer is incorrect or correct the score box should resume from what the score value was, entering a character shouldn’t cause a problem. |  |  |
| 17 | 1 | Addition form - To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits yes then the program should close. |  |  |
| 18 | 1 | Addition form – To test to see if btnMenu closes the current form and takes the user back to main menu. | The form that user is on should hide from and frmMainMenu should open. |  |  |
| 19 | 1 | Subtraction form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far left as it can go | Small, simple numbers should appear in txtNum1 and txtNum2 that makes an easy sum to complete. |  |  |
| 20 | 1 | Subtraction form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is in the middle | More complex numbers should appear, numbers with at least two decimal places should appear |  |  |
| 21 | 1 | Subtraction form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far right as it can go | Complex numbers with at least three decimal places should appear, making it the hardest to calculate the sum. |  |  |
| 22 | 1 | Subtraction form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is correct then a dialogue box will appear saying that the answer is correct, the score should increase by one. |  |  |
| 23 | 1 | Subtraction form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is incorrect then a dialogue box will appear say that the answer in incorrect and tells you try again, the score will increase by 0. |  |  |
| 24 | 1 | Subtraction form – Error Handling – What happens if a complex character was entered into the txtNum1 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 25 | 1 | Subtraction form – Error Handling – What happens if a complex character was entered into the txtNum2 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 26 | 1 | Subtraction form – Error Handling – What happens if a complex character was entered into the txtAnswer and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 27 | 1 | Subtraction form – Error Handling – What happens if a decimal number was entered into the box during an easy sum generation where a decimal number cannot be an answer. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 28 | 1 | Subtraction form – Error Handling – What happens if a complex character was entered into the txtScore, being a text box people will be able to put a value into it. | If the answer is incorrect or correct the score box should resume from what the score value was, entering a character shouldn’t cause a problem. |  |  |
| 29 | 1 | Subtraction form - To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits yes then the program should close. |  |  |
| 30 | 1 | Subtraction form – To test to see if btnMenu closes the current form and takes the user back to main menu. | The form that user is on should hide from and frmMainMenu should open. |  |  |
| 31 | 1 | Multiplication form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far left as it can go | Small, simple numbers should appear in txtNum1 and txtNum2 that makes an easy sum to complete. |  |  |
| 32 | 1 | Multiplication form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is in the middle | More complex numbers should appear, numbers with at least two decimal places should appear |  |  |
| 33 | 1 | Multiplication form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far right as it can go | Complex numbers with at least three decimal places should appear, making it the hardest to calculate the sum. |  |  |
| 34 | 1 | Multiplication form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is correct then a dialogue box will appear saying that the answer is correct, the score should increase by one. |  |  |
| 35 | 1 | Multiplication form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is incorrect then a dialogue box will appear say that the answer in incorrect and tells you try again, the score will increase by 0. |  |  |
| 36 | 1 | Multiplication form – Error Handling – What happens if a complex character was entered into the txtNum1 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 37 | 1 | Multiplication form – Error Handling – What happens if a complex character was entered into the txtNum2 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 38 | 1 | Multiplication form – Error Handling – What happens if a complex character was entered into the txtAnswer and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 39 | 1 | Multiplication form – Error Handling – What happens if a decimal number was entered into the box during an easy sum generation where a decimal number cannot be an answer. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 40 | 1 | Multiplication form – Error Handling – What happens if a complex character was entered into the txtScore, being a text box people will be able to put a value into it. | If the answer is incorrect or correct the score box should resume from what the score value was, entering a character shouldn’t cause a problem. |  |  |
| 41 | 1 | Multiplication form - To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits yes then the program should close. |  |  |
| 42 | 1 | Multiplication form – To test to see if btnMenu closes the current form and takes the user back to main menu. | The form that user is on should hide from and frmMainMenu should open. |  |  |
| 43 | 1 | Division form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far left as it can go | Small, simple numbers should appear in txtNum1 and txtNum2 that makes an easy sum to complete. |  |  |
| 44 | 1 | Division form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is in the middle | More complex numbers should appear, numbers with at least two decimal places should appear |  |  |
| 45 | 1 | Division form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far right as it can go | Complex numbers with at least three decimal places should appear, making it the hardest to calculate the sum. |  |  |
| 46 | 1 | Division form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is correct then a dialogue box will appear saying that the answer is correct, the score should increase by one. |  |  |
| 47 | 1 | Division form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is incorrect then a dialogue box will appear say that the answer in incorrect and tells you try again, the score will increase by 0. |  |  |
| 48 | 1 | Division form – Error Handling – What happens if a complex character was entered into the txtNum1 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 49 | 1 | Division form – Error Handling – What happens if a complex character was entered into the txtNum2 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 50 | 1 | Division form – Error Handling – What happens if a complex character was entered into the txtAnswer and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 51 | 1 | Division form – Error Handling – What happens if a decimal number was entered into the box during an easy sum generation where a decimal number cannot be an answer. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 52 | 1 | Division form – Error Handling – What happens if a complex character was entered into the txtScore, being a text box people will be able to put a value into it. | If the answer is incorrect or correct the score box should resume from what the score value was, entering a character shouldn’t cause a problem. |  |  |
| 53 | 1 | Division form – Error Handling – What would happen if someone manually typed a zero in the denominator box (txtNum2) and tried to calculate it (Can’t divide by zero) | Because the value for num2 is stored as a integer that has to be created by the computer a person putting a zero in txtNum2 shouldn’t cause the program to crash. |  |  |
| 54 | 1 | Division form - To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits yes then the program should close. |  |  |
| 55 | 1 | Division form – To test to see if btnMenu closes the current form and takes the user back to main menu. | The form that user is on should hide from and frmMainMenu should open. |  |  |

# Test Log

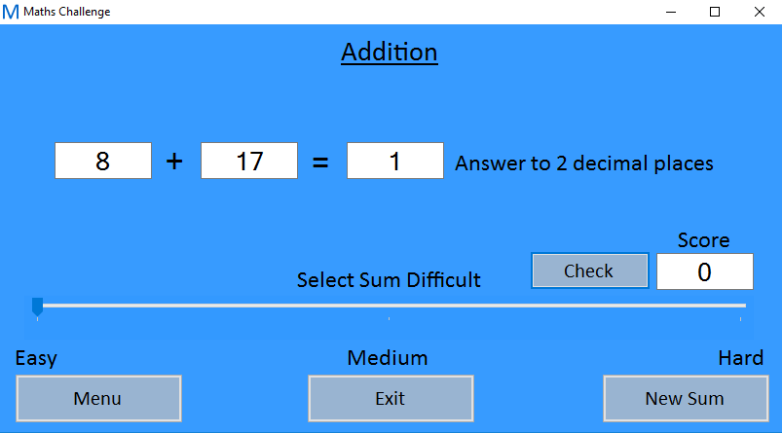
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Test No* | *Run No* | *Test Description* | *Expected Outcome* | *Actual Outcome* | *Action Required* |
| 1 | 1 | Menu Form – To test if the addition button in the form works as it should, the button should open another form within the program and close the one that it was on. | The Addition Button, when clicked should cause the menu form to hide and should open the Addition Form. | As Expected |  |
| 2 | 1 | Menu Form – To test if the subtraction button in the form works as it should, the button should open another form within the program and close the one that it was on. | The subtraction button, when clicked, should cause the menu form to hide and should open the subtraction form. | As Expected |  |
| 3 | 1 | Menu Form – To test if the multiplication button in the form works as it should, the button should open another form within the program and close the one that it was on. | The multiplication button, when clicked, should cause the menu form to hide and should open the multiplication form. | As Expected |  |
| 4 | 1 | Menu Form – To test if the division button in the form works as it should, the button should open another form within the program and close the one that it was on. | The division button, when clicked should cause the menu form to hide and open the division form. | As Expected |  |
| 5 | 1 | Menu Form – To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits yes then the program should close. | As Expected |  |
| 6 | 1 | Menu Form – To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits no the program should remain open. | As Expected |  |
| 7 | 1 | Addition form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far left as it can go | Small, simple numbers should appear in txtNum1 and txtNum2 that makes an easy sum to complete. | As Expected |  |
| 8 | 1 | Addition form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is in the middle | More complex numbers should appear, numbers with at least two decimal places should appear | As Expected |  |
| 9 | 1 | Addition form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far right as it can go | Complex numbers with at least three decimal places should appear, making it the hardest to calculate the sum. | As Expected |  |
| 10 | 1 | Addition form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is correct then a dialogue box will appear saying that the answer is correct, the score should increase by one. | Correct Answer was incorrect according to the code. | Check the code for erros, in Private Sub easySum() the code was set to num1 + 2 instead of num1+ num2. |
| 10 | 2 | Addition form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is correct then a dialogue box will appear saying that the answer is correct, the score should increase by one. | As Expected |  |
| 11 | 1 | Addition form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is incorrect then a dialogue box will appear say that the answer in incorrect and tells you try again, the score will increase by 0. | As Expected |  |
| 12 | 1 | Addition form – Error Handling – What happens if a complex character was entered into the txtNum1 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 13 | 1 | Addition form – Error Handling – What happens if a complex character was entered into the txtNum2 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 14 | 1 | Addition form – Error Handling – What happens if a complex character was entered into the txtAnswer and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 15 | 1 | Addition form – Error Handling – What happens if a decimal number was entered into the box during an easy sum generation where a decimal number cannot be an answer. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 16 | 1 | Addition form – Error Handling – What happens if a complex character was entered into the txtScore, being a text box people will be able to put a value into it. | If the answer is incorrect or correct the score box should resume from what the score value was, entering a character shouldn’t cause a problem. |  |  |
| 17 | 1 | Addition form - To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits yes then the program should close. | As Expected |  |
| 18 | 1 | Addition form – To test to see if btnMenu closes the current form and takes the user back to main menu. | The form that user is on should hide from and frmMainMenu should open. | As Expected |  |
| 19 | 1 | Subtraction form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far left as it can go | Small, simple numbers should appear in txtNum1 and txtNum2 that makes an easy sum to complete. | As Expected |  |
| 20 | 1 | Subtraction form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is in the middle | More complex numbers should appear, numbers with at least two decimal places should appear | As Expected |  |
| 21 | 1 | Subtraction form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far right as it can go | Complex numbers with at least three decimal places should appear, making it the hardest to calculate the sum. | As Expected |  |
| 22 | 1 | Subtraction form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is correct then a dialogue box will appear saying that the answer is correct, the score should increase by one. | As Expected |  |
| 23 | 1 | Subtraction form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is incorrect then a dialogue box will appear say that the answer in incorrect and tells you try again, the score will increase by 0. | As Expected |  |
| 24 | 1 | Subtraction form – Error Handling – What happens if a complex character was entered into the txtNum1 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 25 | 1 | Subtraction form – Error Handling – What happens if a complex character was entered into the txtNum2 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 26 | 1 | Subtraction form – Error Handling – What happens if a complex character was entered into the txtAnswer and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 27 | 1 | Subtraction form – Error Handling – What happens if a decimal number was entered into the box during an easy sum generation where a decimal number cannot be an answer. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 28 | 1 | Subtraction form – Error Handling – What happens if a complex character was entered into the txtScore, being a text box people will be able to put a value into it. | If the answer is incorrect or correct the score box should resume from what the score value was, entering a character shouldn’t cause a problem. |  |  |
| 29 | 1 | Subtraction form - To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits yes then the program should close. | As Expected |  |
| 30 | 1 | Subtraction form – To test to see if btnMenu closes the current form and takes the user back to main menu. | The form that user is on should hide from and frmMainMenu should open. | As Expected |  |
| 31 | 1 | Multiplication form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far left as it can go | Small, simple numbers should appear in txtNum1 and txtNum2 that makes an easy sum to complete. | Nothing Happens | Check code to see where the error is. Error was located as the button names did not match the code, due from copying the code from one form to another |
| 2 | As Expected |  |
| 32 | 1 | Multiplication form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is in the middle | More complex numbers should appear, numbers with at least two decimal places should appear | Nothing Happens | Check code to see where the error is. Error was located as the button names did not match the code, due from copying the code from one form to another |
| 2 | As Expected |  |
| 33 | 1 | Multiplication form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far right as it can go | Complex numbers with at least three decimal places should appear, making it the hardest to calculate the sum. | Nothing happens | Check code to see where the error is. Error was located as the button names did not match the code, due from copying the code from one form to another |
| 2 | Multiplication form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far right as it can go | Complex numbers with at least three decimal places should appear, making it the hardest to calculate the sum. | As Expected |  |
| 33 | 2 | Multiplication form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far right as it can go | Complex numbers with at least three decimal places should appear, making it the hardest to calculate the sum. | As Expected |  |
| 34 | 1 | Multiplication form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is correct then a dialogue box will appear saying that the answer is correct, the score should increase by one. | Nothing Happens | Check code to see where the error is. Error was located as the button names did not match the code, due from copying the code from one form to another |
| 2 | As Expected |  |
| 35 | 1 | Multiplication form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is incorrect then a dialogue box will appear say that the answer in incorrect and tells you try again, the score will increase by 0. | Nothing Happens | Check code to see where the error is. Error was located as the button names did not match the code, due from copying the code from one form to another |
| 2 |  | As Expected |  |
| 36 | 1 | Multiplication form – Error Handling – What happens if a complex character was entered into the txtNum1 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 37 | 1 | Multiplication form – Error Handling – What happens if a complex character was entered into the txtNum2 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 38 | 1 | Multiplication form – Error Handling – What happens if a complex character was entered into the txtAnswer and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 39 | 1 | Multiplication form – Error Handling – What happens if a decimal number was entered into the box during an easy sum generation where a decimal number cannot be an answer. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. |  |  |
| 40 | 1 | Multiplication form – Error Handling – What happens if a complex character was entered into the txtScore, being a text box people will be able to put a value into it. | If the answer is incorrect or correct the score box should resume from what the score value was, entering a character shouldn’t cause a problem. |  |  |
| 41 | 1 | Multiplication form - To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits yes then the program should close. | As Expected |  |
| 42 | 1 | Multiplication form – To test to see if btnMenu closes the current form and takes the user back to main menu. | The form that user is on should hide from and frmMainMenu should open. | As Expected |  |
| 43 | 1 | Division form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far left as it can go | Small, simple numbers should appear in txtNum1 and txtNum2 that makes an easy sum to complete. | As Expected |  |
| 44 | 1 | Division form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is in the middle | More complex numbers should appear, numbers with at least two decimal places should appear | As Expected |  |
| 45 | 1 | Division form – btnNewSum should generate a number in txtNum1 and txtNum2 based on the location of the track bar. This is to test when the track bar is as far right as it can go | Complex numbers with at least three decimal places should appear, making it the hardest to calculate the sum. | As Expected |  |
| 46 | 1 | Division form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is correct then a dialogue box will appear saying that the answer is correct, the score should increase by one. | Even though the answer is correct, the messagebox says that it is incorrect. | Rewrite code for btnCheck |
|  | 2 | Even though the answer is correct the messagebox still says that it is incorrect | Rewrite code for whole form, to locate bugs. |
| 47 | 1 | Division form – btnCheck should check the answer in txtAnswer with the real answer and message box will appear, and the txtScore should respond in kind. | If the answer is incorrect then a dialogue box will appear say that the answer in incorrect and tells you try again, the score will increase by 0. | As Expected |  |
| 48 | 1 | Division form – Error Handling – What happens if a complex character was entered into the txtNum1 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. | As Expected |  |
| 49 | 1 | Division form – Error Handling – What happens if a complex character was entered into the txtNum2 and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. | As Expected |  |
| 50 | 1 | Division form – Error Handling – What happens if a complex character was entered into the txtAnswer and click check. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. | As Expected |  |
| 51 | 1 | Division form – Error Handling – What happens if a decimal number was entered into the box during an easy sum generation where a decimal number cannot be an answer. | Because the answer would be incorrect, the dialogue box that says that the answer you gave is incorrect should appear. | As Expected |  |
| 52 | 1 | Division form – Error Handling – What happens if a complex character was entered into the txtScore, being a text box people will be able to put a value into it. | If the answer is incorrect or correct the score box should resume from what the score value was, entering a character shouldn’t cause a problem. | As Expected |  |
| 53 | 1 | Division form – Error Handling – What would happen if someone manually typed a zero in the denominator box (txtNum2) and tried to calculate it (Can’t divide by zero) | Because the value for num2 is stored as a integer that has to be created by the computer a person putting a zero in txtNum2 shouldn’t cause the program to crash. | As Expected |  |
| 54 | 1 | Division form - To test if the exit button in the form works as it should, it should open a prompt, and if the correct result is entered the program should close. | The exit button, when clicked, should open a dialogue box that asks the person if they are sure that they want to leave the program, if the person hits yes then the program should close. | As Expected |  |
| 55 | 1 | Division form – To test to see if btnMenu closes the current form and takes the user back to main menu. | The form that user is on should hide from and frmMainMenu should open. | As Expected |  |

# Evaluation

## Menu Form

The good thing about the menu form is that it clearly shows the separate part of the programs so that the person can make their choice, it clearly shows whether they want to choose addition, subtraction, multiplication and division. Another good thing about this form is the exit button, if you hit the X at the top of the form it does not close all the other forms that are open in the program, even if they are hidden, the exit button will close the program safely. The problem with this form is that it is very bland, there’s not much going on in terms of graphics, another bad feature of this form is that the buttons are faced far apart, but this is only because of the size of the other forms, keeping the form size consistent amongst the other forms would not have been possible if the size of this form was smaller, also making the buttons doesn’t work neither, too big just makes the program look unprofessional.

## Addition, Subtraction, Multiplication and Division forms

One feature of this form that makes it work so well is the scroll bar. Instead of a scroll bar being used; however, a track-bar was used instead; this was because a scroll bar was harder to program properly than what the track bar was, instead of setting the values of the scrollbar, the track-bar has pre-set values, and they are a lot easier to change. Another good aspect of this form is the ability to choose the difficulty of the sum, the easy sums are simple, small whole numbers, the medium sums are simple, small decimal numbers, but the hard sums are decimal and large numbers, and the answer box is set to two decimal places; this is a good feature because in forms like division, sum numbers, when they are divided by each other, end up with recurring fractions. One final feature that is really good in the forms is the fact that when the forms first open and the boxes are blank you cannot tally the score up, even if nothing times nothing is nothing, because of the way it has been programmed it will prevent this

The features that are not so good would be features like the ability to check the score over and over again if you have the right answer, the problem with this is that it will keep on tallying the score, because logically the answer that you have entered is correct. Another feature that is not good is the colour of the track bar, unfortunately in visual studio the colours are different to what they appear on the screen, probably due to some compatibility issues, and this causes the colour of the track bar to blend into the background, and the trackers on the track bar are barely visible.

### Improvements

One thing that could be improved is the colour scheme, changing the colour, or adding a background to the program could make it look more appealing, and would prevent the track bar from blending into the background. The reason why this didn’t happen was because it would have taken a lot long to find the right colour scheme that would allow the two to match.

The next improvement I could make would be to make the whole program appear on one form, the problem with making a program on one form would be the amount of debugging you would have to do if something was to go wrong, however if I had more time to do it this would have been a possible option, and I would have been able to learn how to hide certain things once a certain button is pressed instead of hiding the whole form.

The final thing that I would want to improve on would be the check score. Every time you hit check the program will run a sub routine that works out if the answer that was entered is correct, and would display a dialogue box accordingly; however, once you have the correct answer you could keep on clicking check and this would cause the score to tally up indefinitely. If I had time to fix this I would prevent people from being able to do, maybe adding a new sub routine to run when the check button is clicked to prevent people from abusing the bug.