**Hangman Design**

**Introduction:**

The problem:

People love to play games and in particular the use of video games. Bringing an all-time favourite to the video game world may attract a wider range of gamers and show non-gamers that not every game is violent such as “Grand Theft Auto” or ”Call of Duty” which a lot of people seem to think are violent and are respective of all video games.

How I intend to solve this problem:

I will make a game in Visual Basic that will allow the user to play hangman on a computer without access to the game on a website or having to pay to play. I will make this program easy and enjoyable to play for newcomers. Playing hangman using technology will be easy to use and won’t require any other resources such as paper or pen.

Success Criteria:

In my program I will need to:

* An easy and self-explanatory program that anyone can use.
* Let the player choose the category for the word they are guessing.
* Show the length of the letters so they can guess words based on length.
* Allow the player to guess a letter and see used letters that may have been either rejected or found.
* The word should change if a letter from it is guessed and show the word.
* Show a ‘hanging man’ who grows as wrong letters are guessed.
* Show the player if they have won or lost and allow the user to play the game again.
* If the same letter is entered twice then the program will notify the user and won’t count the letter.

Testing:

My program will be tested by making sure the functions work by starting out with a basic word that is set and making sure the program will work then with words and lives. Once the testing is done in the first stage I will allow the user to change the word category with multiple words which I will need to test through multiple words. Once the program is fully functioning then I will start making it more aesthetically pleasing to the user and laid out better for the user.

**Design:**

Inputs and Outputs:

The inputs that I will be using in my program are:

* Letter guess for the word
* Full word guess
* Category for the word

The outputs that I will be using in my program are:

* Letters used – correct or incorrect
* Letters guessed in the word in the blank letter spaces
* Number of letters in a word
* Category selected by the user
* If the word has been guessed or not

User Interface:

\*See the design sheet!\*

Visual Basic: I will use Visual Basic as you can code efficiently and while using Visual Studio it will allow me to design a user interface that is easy to create and navigate.

Menu strip: Allows user easy access to reset the word, close the program, change categories and find help to use the program.

Reset button: If the user wants to finds a new word as some words people may not know how to spell.

Exit button: Once the user has finished using the program I will have an exit button to allow the user a quick and alternative way to exit the program.

Flow Chart:

\*See the design sheet!\*

Validation:

With my validation I need to take account of a few things:

* A single letter can only be entered
* Anything numerical or a symbol will be rejected
* Same letter entered in twice
* Presence check

Variables:

Integers: ‘lives’ + ‘letter\_count’ + ‘word\_number’ + ‘letters\_used\_count’

Strings: ‘word’ + ‘array\_length’ + ‘rand\_num’ + ‘letters\_used’

Code:

Public Class Hangman

Dim word, lives, letters\_left, letter, underscores As String

'Declares the variables

Dim food() As String = {"Pizza", "Burger", "Lasagne", "Curry", "Quiche"}

Dim sport() As String = {"Football", "Basketball", "Cricket", "Badminton", "Volleyball"}

Dim countries() As String = {"Portugal", "Paraguay", "Jamaica", "Malaysia", "Ethiopia"}

'Sets the categories in an array

Dim Dice As New Random

Dim DiceResult As Integer = Dice.Next(1, 5)

'Declares the random for the word selection

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

Call reset()

'Calls the reset sub

End Sub

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

Call lives\_check()

'Calls the lives sub

letter = txtLetter.Text

letter = letter.Trim

word = UCase(word)

txtLetter.MaxLength = 1

'Sets up features for the letter guess

If mnuCategoriesFood.Enabled = True Or mnuCategoriesSport.Enabled = True Or mnuCategoriesCountries.Enabled = True Then

MsgBox("You need to select a category from the menu or the combo box!", 48, "Category needs selecting!")

txtLetter.Clear()

Exit Sub

End If

'Checks if a category has not been selected

If letter.Trim = "" Then

MsgBox("You need to enter in a letter!", vbExclamation, "No letter!")

Exit Sub

End If

'Checks if anything has been entered

If letter Like "\*[a-z]\*" Then

letter = UCase(letter)

End If

'Sets a lower case letter to upper case

If letter Like "\*[A-Z]\*" Then

Dim count As Integer

For count = 0 To pnlAlphabet.Controls.Count - 1

If pnlAlphabet.Controls.Item(count).Text = letter Then

If pnlAlphabet.Controls.Item(count).ForeColor <> SystemColors.ControlText Then

MsgBox("You have already guessed that letter!", vbExclamation, "Double Guess!")

Exit Sub

End If

End If

Next

'Checks if the letter has been used

If word.Contains(letter) Then

For count = 0 To pnlAlphabet.Controls.Count - 1

If pnlAlphabet.Controls.Item(count).Text = letter Then

pnlAlphabet.Controls.Item(count).ForeColor = Color.Green

End If

Next

'Sets a correct letter to green

Dim Current\_Word As String = lblUnderscores.Text

Dim word\_characters() = Current\_Word.ToCharArray()

'Declares varaibles for the underscores

For count = 0 To Len(word) - 1

If word.Chars(count) = letter Then

word\_characters(2 \* count) = letter

End If

Next

'Goes through the length of the word to check underscores

Dim new\_word As String = word\_characters

lblUnderscores.Text = new\_word

'Declares more varaibles for underscores

Else

For count = 0 To pnlAlphabet.Controls.Count - 1

If pnlAlphabet.Controls.Item(count).Text = letter Then

pnlAlphabet.Controls.Item(count).ForeColor = Color.Red

End If

Next

'Sets a wrong letter to red

Call Wrong\_Guess()

'Calls the wrong guess sub

End If

ElseIf letter Like " \*[0-9]\*" Then

MsgBox("You entered a number! You cannot enter numbers!", vbExclamation, "Number entered!")

txtLetter.Clear()

Exit Sub

'Checks if a number is entered

Else

MsgBox("You have entered in a symbol! You cannot enter symbols!", vbExclamation, "Symbol entered!")

txtLetter.Clear()

Exit Sub

'Checks if a symbol is entered through process of elimination

End If

txtLetter.Clear()

'Clears the text box

Call full\_word()

Call lives\_check()

'Calls two subs for validation

End Sub

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

Dim Input\_Word As String = InputBox("What is the word?")

'Sets a input box for the word guess

If LCase(Input\_Word) = LCase(word) Then

Call You\_Win()

'Calls the win case

Else

Call Wrong\_Guess()

'Calls for wrong case

End If

End Sub

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

Call reset()

'Calls reset case

End Sub

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

If mnuCategoriesFood.Checked = True Then

Call disable()

Call makeUnderscores(word)

'Calls the disable and underscores sub

comboCategories.Text = "Food"

word = food(DiceResult.ToString)

lblUnderscores.Visible = True

'Category and word selection

End If

End Sub

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

If mnuCategoriesSport.Checked = True Then

Call disable()

Call makeUnderscores(word)

'Calls the disable and underscores sub

comboCategories.Text = "Sport"

word = sport(DiceResult.ToString)

lblUnderscores.Visible = True

'Category and word selection

End If

End Sub

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

If mnuCategoriesCountries.Checked = True Then

Call disable()

Call makeUnderscores(word)

'Calls the disable and underscores sub

comboCategories.Text = "Countries"

word = countries(DiceResult.ToString)

lblUnderscores.Visible = True

'Category and word selection

End If

End Sub

Private Sub cbmoCategories\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles comboCategories.SelectedIndexChanged

Call Clear\_Checked()

'Calls the clear check to reset checked

If comboCategories.SelectedIndex = 0 Then

mnuCategoriesFood.Checked = True

Call mnuCategoriesFood\_Click(sender, e)

ElseIf comboCategories.SelectedIndex = 1 Then

mnuCategoriesSport.Checked = True

Call mnuCategoriesSport\_Click(sender, e)

ElseIf comboCategories.SelectedIndex = 2 Then

mnuCategoriesCountries.Checked = True

Call mnuCategoriesCountries\_Click(sender, e)

Else

comboCategories.Text = "Unknown"

End If

'Re-checks menu strip tabs

End Sub

Private Sub makeUnderscores(word As String)

Dim blank\_string As String = ""

Dim count As Integer

'Declares variables for underscores

underscores = Len(word)

letters\_left = Len(word)

lblLettersCount.Text = letters\_left

'Underscore validation

For count = 0 To Len(word) - 1

blank\_string += "\_" & " "

Next

'Checkes the word for guessed letter

lblUnderscores.Text = blank\_string

'Sets the underscores on the label

End Sub

Sub Wrong\_Guess()

lives = lives - 1

MsgBox("You have guessed wrong! You will lose a life!", vbExclamation, "Wrong Guess!")

'Removes one life

Call lives\_check()

'Calls lives check

End Sub

Sub You\_Win()

MsgBox("You win!", vbInformation, "Congratulations!")

lblUnderscores.Text = word

txtLetter.Clear()

'Allows user to win

End Sub

Sub Clear\_Checked()

mnuCategoriesFood.Checked = False

mnuCategoriesSport.Checked = False

mnuCategoriesCountries.Checked = False

'Unchecks all menu strip tabs

End Sub

Sub lives\_check()

If lives = 6 Then

picHangman.Image = My.Resources.six\_lives

ElseIf lives = 5 Then

picHangman.Image = My.Resources.five\_lives

ElseIf lives = 4 Then

picHangman.Image = My.Resources.four\_lives

ElseIf lives = 3 Then

picHangman.Image = My.Resources.three\_lives

ElseIf lives = 2 Then

picHangman.Image = My.Resources.two\_lives

ElseIf lives = 1 Then

picHangman.Image = My.Resources.one\_life

ElseIf lives = 0 Then

picHangman.Image = My.Resources.zero\_lives

MsgBox("Hangman!", vbCritical, "Hangman!")

End If

lblLivesCount.Text = lives

'Checks the lives count

End Sub

Sub full\_word()

If lblUnderscores.Text = word Then

MsgBox("You win!", vbOKOnly, "Congratulations!")

lblUnderscores.Text = word

txtLetter.Clear()

End If

'Checks if full word is guessed

End Sub

Sub reset()

lives = 6

letters\_left = 0

lblLettersCount.Text = letters\_left

comboCategories.Text = "Unknown"

txtLetter.Clear()

mnuCategoriesFood.Enabled = True

mnuCategoriesSport.Enabled = True

mnuCategoriesCountries.Enabled = True

mnuCategoriesFood.Checked = False

mnuCategoriesSport.Checked = False

mnuCategoriesCountries.Checked = False

lblUnderscores.Visible = False

picHangman.Image = My.Resources.six\_lives

Dim count As Integer

For count = 0 To pnlAlphabet.Controls.Count - 1

If pnlAlphabet.Controls.Item(count).Text = letter Then

pnlAlphabet.Controls.Item(count).ForeColor = SystemColors.ControlText

End If

Next

'Resets all features of form

End Sub

Sub disable()

mnuCategoriesFood.Enabled = False

mnuCategoriesSport.Enabled = False

mnuCategoriesCountries.Enabled = False

'Disables menu strip tabs

End Sub

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

MsgBox("This Is a game of Hangman made in Visual Basic by Tom H.", vbInformation, "Details: ")

'Shows the user programmers and programs admin details

End Sub

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

Me.Close()

'Close the form

End Sub

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

MsgBox("This is my hangman game which you can play right now! Choose a category from the menu and the guess a letter!", vbInformation, "Help Centre")

'Shows the user a help centre

End Sub

End Class



