**INDEX**

|  |  |
| --- | --- |
| **SR NO** | **CONTENT** |
| 1. | ABSTRACT |
| 2. | INTRODUCTION |
| 3. | HARDWARE AND SOFTWARE REQ. |
| 4. | ALGORITHM |
| 5. | FLOWCHART |
| 6. | SOURCE CODE |
| 7. | OUTPUT |
| 8. | APPLICATION |
| 9. | CONCLUSION |
| 10. | REFERENCE |

**INTRODUCTION**

This Visual Basic .NET program is a traditional rendition of the timeless Tic-Tac-Toe game, designed for two players. The interface is straightforward, presenting a 3x3 grid where players can place their marks, X or O, by clicking on the desired square. Players take turns, with the game tracking the sequence to determine the current player. After each move, the program checks for a winning combination, analyzing rows, columns, and diagonals. If a player successfully aligns three marks, they win the game, prompting a celebratory display. In the event that all squares are filled without a winner, the game concludes in a draw. A "New Game" button resets the board for endless rounds of entertainment.

**HARDWARE AND SOFTWARE REQ.**

**Software Requirements:**

- Operating System: Windows 7 or later

- .NET Framework: .NET Framework 4.5 or later

- Integrated Development Environment (IDE): Visual Studio 2010 or later (to open and modify the project)

**Hardware Requirements:**

- Processor: 1 GHz or faster processor

- RAM: 1 GB (32-bit) or 2 GB (64-bit) RAM

- Storage: 50 MB of available hard disk space for installation

**ABSTRACT**

This Visual Basic .NET program is a classic implementation of the Tic-Tac-Toe game using Windows Forms. The game provides a grid of buttons representing the game board. Players alternate turns, clicking on empty buttons to place their mark (X or O) on the board.

After each move, the game checks for a winning condition by examining the rows, columns, and diagonals of the board. If a player successfully places their mark in a row, column, or diagonal, the game declares them the winner and highlights the winning combination of buttons. If no player achieves a winning combination and all buttons are filled, the game declares a draw.

The interface includes a label to display the current player's turn and a button to start a new game. Clicking the new game button resets the board, allowing players to play again. Overall, this program provides a fun and interactive way to play the classic Tic-Tac-Toe game on the Windows platform.

**APPLICATIONS**

**1.Initialize the Game:**

- Create a 3x3 grid to represent the game board.

- Initialize all grid cells to empty.

**2. Game Loop:**

- Repeat until the game is over:

- Player X makes a move:

- Wait for player X to click on an empty cell.

- Mark the cell as X.

- Check for a win or a draw.

- Player O makes a move:

- Wait for player O to click on an empty cell.

- Mark the cell as O.

- Check for a win or a draw.

**3. Check for Win or Draw:**

- Check rows, columns, and diagonals for three consecutive X's or O's.

- If a win is found, end the game and declare the winner.

- If all cells are filled and no win is found, end the game as a draw.

**4. End of Game:**

- Display the result (win, draw) to the players.

- Offer the option to start a new game.

**SOURCE CODE**

Public Class VB\_TIC\_TAC\_TOE

Private Sub VB\_TIC\_TAC\_TOE\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

' add event to all buttons inside the panel2

For Each c As Control In Panel2.Controls

If c.GetType() = GetType(Button) Then

AddHandler c.Click, AddressOf btn\_Click

End If

Next

End Sub

Dim XorO As Integer = 0

' create button event

Private Sub btn\_Click(sender As Object, e As EventArgs)

Dim btn As Button = sender

' we will clear buttons text later

If btn.Text.Equals("") Then

If XorO Mod 2 = 0 Then

btn.Text = "X"

btn.ForeColor = Color.Red

Label1.Text = "[O] Turn"

getTheWinner()

Else

btn.Text = "O"

btn.ForeColor = Color.Blue

Label1.Text = "[X] Turn"

getTheWinner()

End If

XorO += 1

End If

End Sub

Dim win As Boolean = False

' create a function to get the winner

Private Sub getTheWinner()

If Not Button1.Text.Equals("") AndAlso Button1.Text.Equals(Button2.Text) AndAlso Button1.Text.Equals(Button3.Text) Then

win = True

winEffect(Button1, Button2, Button3)

End If

If Not Button4.Text.Equals("") AndAlso Button4.Text.Equals(Button5.Text) AndAlso Button4.Text.Equals(Button6.Text) Then

win = True

winEffect(Button4, Button5, Button6)

End If

If Not Button7.Text.Equals("") AndAlso Button7.Text.Equals(Button8.Text) AndAlso Button7.Text.Equals(Button9.Text) Then

win = True

winEffect(Button7, Button8, Button9)

End If

If Not Button1.Text.Equals("") AndAlso Button1.Text.Equals(Button4.Text) AndAlso Button1.Text.Equals(Button7.Text) Then

win = True

winEffect(Button1, Button4, Button7)

End If

If Not Button2.Text.Equals("") AndAlso Button2.Text.Equals(Button5.Text) AndAlso Button2.Text.Equals(Button8.Text) Then

win = True

winEffect(Button2, Button5, Button8)

End If

If Not Button3.Text.Equals("") AndAlso Button3.Text.Equals(Button6.Text) AndAlso Button3.Text.Equals(Button9.Text) Then

win = True

winEffect(Button3, Button6, Button9)

End If

If Not Button1.Text.Equals("") AndAlso Button1.Text.Equals(Button5.Text) AndAlso Button1.Text.Equals(Button9.Text) Then

win = True

winEffect(Button1, Button5, Button9)

End If

If Not Button3.Text.Equals("") AndAlso Button3.Text.Equals(Button5.Text) AndAlso Button3.Text.Equals(Button7.Text) Then

win = True

winEffect(Button3, Button5, Button7)

End If

' if no one win later

' 9 buttons with X or O mean 9 char = no button is empty

If allbuttonsTextLength() = 9 AndAlso win = False Then

Label1.Text = "NO Winner"

End If

End Sub

Function allbuttonsTextLength() As Integer

Dim btnsTxtLength As Integer = 0

For Each c As Control In Panel2.Controls

If c.GetType() = GetType(Button) Then

btnsTxtLength += c.Text.Length

End If

Next

Return btnsTxtLength

End Function

Private Sub winEffect(ByVal b1 As Button, ByVal b2 As Button, ByVal b3 As Button)

b1.BackColor = Color.Red

b2.BackColor = Color.Red

b3.BackColor = Color.Red

b1.ForeColor = Color.White

b2.ForeColor = Color.White

b3.ForeColor = Color.White

Label1.Text = b1.Text + " Win"

End Sub

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

XorO = 0

win = False

Label1.Text = "Play"

For Each c As Control In Panel2.Controls

If c.GetType() = GetType(Button) Then

c.BackColor = Color.White

c.Text = ""

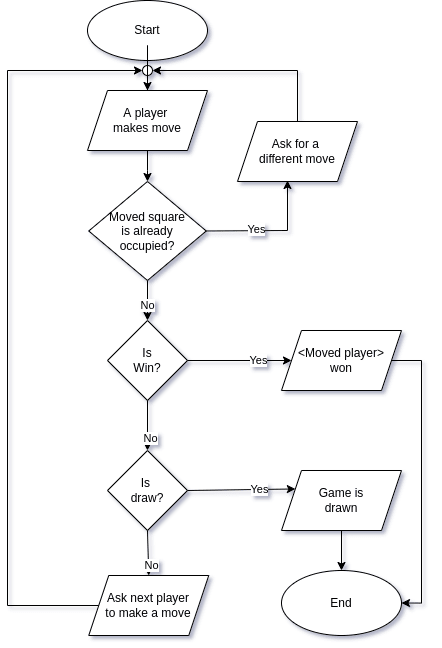
End If

Next

End Sub

End Class

**FLOWCHART**



**APPLICATIONS**

1. **Entertainment:** Provide a fun and engaging game for users to play in their free time, either against each other or against a computer opponent.

2. **Learning Tool:** Use the game to teach basic programming concepts such as loops, conditionals, and event handling in a practical and interactive way.

3. **Problem Solving:** Develop problem-solving skills by strategizing to win the game or create an AI opponent with varying levels of difficulty.

4. **User Interface Design:** Showcase good user interface design practices by creating an intuitive and visually appealing game interface.

5. **Project Template:** Use the game as a template for other Windows Forms projects, demonstrating how to structure and organize code for similar applications.

6. **Multiplayer Gaming:** Extend the game to support multiplayer functionality over a network, allowing users to play against each other remotely.

**CONCLUSION**

In conclusion, this Visual Basic .NET implementation of Tic-Tac-Toe provides a simple yet effective example of game development using Windows Forms. The game offers an engaging experience for two players, challenging them to strategically place their marks to win the game. With its clear and intuitive interface, the game not only entertains but also serves as a learning tool for programming concepts and user interface design. Additionally, the game can be further enhanced with features like AI opponents, multiplayer functionality, and mobile adaptation, showcasing its adaptability and potential for expansion. Overall, this Tic-Tac-Toe game demonstrates the versatility and educational value of Windows Forms programming in Visual Basic .NET.

**REFERENCE**

1. <https://learn.microsoft.com/en-us/dotnet/visual-basic/>
2. <https://www.tutorialspoint.com/vb.net/index.htm>
3. <https://www.javatpoint.com/vb-net>

**OUTPUT**

