Module Module1

Sub Main()

Dim loopCount As Integer = 0 'added to allow them to play games as many as they want

Dim NewBingoGame As New BingoGame

'added = True

Do

If NewBingoGame.Menu() = True Then

Console.WriteLine("Do you want to play again? - Y or N")

If UCase(Console.ReadLine) = "Y" Then

Else

loopCount = 1 'added to check if end the system

End If

Else loopCount = 1

End If

Loop Until loopCount = 1 'added

End Sub

End Module

Public Class BingoGame

Private winCounter As Integer = 0

Private lossCounter As Integer = 0

Private numbers As New NumberMachine

Private round As Integer = 0 'added so we can have correct grammer for the first ball - first ball is ... instead of next ball is for round 1

Private callernum As Integer 'added variable (same name as caller (current variable name and the name of the function it is in) which contains value of next number being called

Private stringsetGame As String = "" 'added - string that stores all the numbers previously called in the normal bingo game

Private stringsetBonus As String = "" 'added - string that stores all the numbers previously called in the bingo bonus game

Public Sub New()

Console.WriteLine("$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$")

Console.WriteLine("\* B-I-N-G-O S-I-M \*")

Console.WriteLine("$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$")

End Sub

Public Sub PlayGame()

Dim playerCard As New BingoCard

Dim won As Boolean

Dim responser As String 'added - converts the number that is currently being called into string to be added to the list of numbers stored

stringsetGame = ""

playerCard.Displaycard()

Console.WriteLine("Eyes Down... ")

Do

responser = CStr(Caller()) 'added responser = cstr()

Console.WriteLine("The numbers rolled so far are: ") 'added

stringsetGame = stringsetGame + (responser) + " " 'added

Console.WriteLine(stringsetGame) 'added

playerCard.checkNum(stringsetGame)

Console.WriteLine("Did you win?")

Console.WriteLine("Enter 1 for yes and 0 for no?")

Do

Try 'added try catch to validate the data entry

won = Console.ReadLine()

Catch ex As Exception

Console.WriteLine("That was not a valid option. Please enter 1 or 0")

won = Console.ReadLine()

End Try

Loop Until won = "0" Or won = "1"

Console.Clear()

playerCard.Displaycard()

Loop Until won

Console.WriteLine("You have matched ") '& playerCard.GameOver(numbers.getNumbers, numbers.getBack))

If playerCard.GameOver(numbers.getBack) = 15 Then

Console.WriteLine("Yes you have won!")

Else

Console.WriteLine("Sorry you stopped too early")

Console.WriteLine("You only matched " & playerCard.GameOver(numbers.getBack))

Console.WriteLine("GAME OVER")

End If

End Sub

Public Sub PlayBonusGame()

Dim playerCard As New BingoBonusCard

Dim won As Boolean

Dim responser As String 'added - converts the number that is currently being called into string to be added to the list of numbers stored

playerCard.Displaycard()

stringsetBonus = ""

Console.WriteLine("Eyes Down... ")

Do

Console.WriteLine("\*\*BINGO BONUS\*\*")

responser = CStr(Caller()) 'added responser = cstr()

Console.WriteLine("The numbers rolled so far are: ") 'added so we know what numbers have been called so far

stringsetBonus = stringsetBonus + (responser) + " " 'added - this stores the number been called with the rest of the numbers that have already been called

Console.WriteLine(stringsetBonus) 'added - this outputs the numbers called so far

playerCard.checkNum(stringsetBonus)

Console.WriteLine("Did you win?")

Console.WriteLine("Enter 1 for yes and 0 for no?")

Do

Try 'added try catch to validate the data entry

won = Console.ReadLine()

Catch ex As Exception

Console.WriteLine("That was not a valid option. Please enter 1 or 0")

won = Console.ReadLine()

End Try

Loop Until won = "0" Or won = "1"

'won = console.readline() 'removed this

Console.Clear()

playerCard.Displaycard()

Loop Until won

Select Case playerCard.GameOver(numbers.getBack)

Case 1

Console.WriteLine("Well Done Full House")

Case 2

Console.WriteLine("Well Done Horizontal Lines")

Case 3

Console.WriteLine("Well Done Corners")

Case 0

Console.WriteLine("Sorry you stopped too early, GAME OVER")

End Select

End Sub

Private Function Caller() As Integer

'caller = numbers.nextBall 'removed line

callernum = numbers.nextBall 'added - stores next number to be called

If round = 0 Then 'added

Console.WriteLine("The first ball is ... ") 'added for grammer purposes

round = round + 1 'added

ElseIf callernum = -1 Then 'changed from caller to callernum

Console.WriteLine("Sorry there are no more available numbers left in the game") 'added so users know to not bother rolling and more since there are no available numbers left

Else

Console.WriteLine("and the next ball is.....")

round = round + 1

End If

If callernum = 11 Then 'changed from caller to callernum

Console.WriteLine("legs 11")

ElseIf callernum = 22 Then 'added 'changed from caller to callernum

Console.WriteLine("two little ducks") 'added - special names thing

Else

Console.WriteLine(callernum) 'changed from caller to callernum

End If

Return callernum 'added - so it returns smth

End Function

Public Function Menu() As Boolean

Dim choice As Integer

numbers = New NumberMachine

Do

Try 'added try catch to validate data entry

MenuOptions()

choice = Console.ReadLine()

Catch ex As Exception

End Try

Select Case choice

Case 0

Console.WriteLine("Goodbye")

Return False 'added to fix error when enter 0 in menu

Case 1

Console.Clear() 'added - clear console for user

PlayGame()

Case 2

Console.Clear() 'added - clear console for user

PlayBonusGame()

Case 3

numbers.PracticeGame()

Console.Clear()

Console.WriteLine("Practice Game Initiated")

PlayGame() 'added - play game (normal bingo)

Case 4

instructions()

Case Else

Console.WriteLine("Not an Option")

End Select

Loop Until choice = 1 Or choice = 2 Or choice = 0 Or choice = 3 'added choice = 3 (for practice game), 4 (to display instructions), 5 (tally up games won and store onto file)

Return True

End Function

Public Sub instructions()

Console.WriteLine("The rules")

Console.WriteLine("The game is typically won by winning a full house")

Console.WriteLine("If you are playing BINGO BONUS you can also win by having all 4 corners or a horizonal line of 5 numbers on the ticket")

End Sub

Public Sub MenuOptions()

Console.WriteLine("---------------------------------------")

Console.WriteLine("Choose 1 play Bingo")

Console.WriteLine("Choose 2 play Bingo Bonus")

Console.WriteLine("Choose 3 play practice game")

Console.WriteLine("Choose 4 for the instructions")

Console.WriteLine("Choose 5 to see the number of times you have won")

Console.WriteLine("Choose 0 to exit")

End Sub

End Class

Public Class BingoCard

Protected numbers(2, 8) As Integer

Protected callednum(2, 8) As Boolean

Protected alreadyCalled As String = ""

Public Sub New()

numbers = AssignNumbers()

End Sub

Private Function AssignNumbers() As Integer(,)

Randomize() '

Dim row1(4) As Integer

Dim row2(4) As Integer

Dim row3(4) As Integer

Dim cardnumbers(2, 8) As Integer

row1 = AssignRowPlaces()

row2 = AssignRowPlaces()

row3 = AssignRowPlaces()

For x = 0 To 4

cardnumbers(0, row1(x)) = repo.NewRandom(1, 8) + (10 \* row1(x))

Next

For x = 0 To 4

Dim base As Integer

base = (cardnumbers(0, row2(x)) Mod 10) + 1

base = repo.NewRandom(base, 9)

cardnumbers(1, row2(x)) = base + (10 \* row2(x))

Next

For x = 0 To 4

Dim base As Integer

If cardnumbers(1, row3(x)) = 0 Then

base = (cardnumbers(0, row3(x)) Mod 10) + 1

Else

base = (cardnumbers(1, row3(x)) Mod 10) + 1

End If

base = repo.NewRandom(base, 10)

cardnumbers(2, row3(x)) = base + (10 \* row3(x))

Next

Return cardnumbers

End Function

Public Sub Displaycard()

For x = 0 To 2

Console.Write("|")

For y = 0 To 8

' Console.Write(numbers(x, y) & ",")

If Len(CStr(numbers(x, y))) = 2 Then 'this if statement is for spacing and alignment of the board

If callednum(x, y) = True Then 'added this nested IF statement in order to display X when the number is picked

Console.Write(" X | ")

Else

Console.Write(numbers(x, y) & " | ") 'changed the , to a | so it is easier to see the columns separately

End If

ElseIf Len(CStr((numbers(x, y)))) = 1 Then

If callednum(x, y) = True Then

Console.Write(" X | ")

Else

Console.Write(" " & numbers(x, y) & " | ")

End If

End If

Next y

Console.WriteLine()

Next x

End Sub

Private Function AssignRowPlaces() As Integer()

Dim numberCount As Integer

Dim row(4) As Integer

Dim match As Boolean

For x = 0 To 4

row(x) = -1

Next

Do While numberCount <= 4

match = False

row(numberCount) = repo.NewRandom(0, 8)

numberCount += 1

For x = 0 To 4

If row(numberCount - 1) = row(x) And numberCount - 1 <> x Then

match = True

End If

Next

If match Then

numberCount -= 1

End If

Loop

row = rearrage(row, 4)

Return row

End Function

Private Function rearrage(ByVal dataSet As Integer(), ByVal size As Integer) As Integer()

Dim i, j As Integer

For i = 0 To size - 1

For j = 0 To size - 1

If (dataSet(j) > dataSet(j + 1)) Then

Dim temp As Integer = dataSet(j)

dataSet(j) = dataSet(j + 1)

dataSet(j + 1) = temp

End If

Next

Next

Return dataSet

End Function

Public Overridable Function GameOver(ByVal tail As Integer) As Integer

Dim matched As Integer

For x = 0 To 2

For y = 0 To 8

If callednum(x, y) = True Then

matched += 1

End If

Next

Next

Return matched

End Function

Sub checkNum(ByVal callerNumbers As String)

Dim callerNums As Integer

If callerNumbers <> " " Then

callerNums = CInt(Mid(callerNumbers, Len(alreadyCalled) + 1, Len(callerNumbers) - 1))

End If

alreadyCalled = callerNumbers

If callerNums <> 0 Then

For x = 0 To 8

For y = 0 To 2

If numbers(y, x) = callerNums Then

callednum(y, x) = True

End If

Next

Next

End If

End Sub

End Class

Public Class BingoBonusCard

Inherits BingoCard

Public Overrides Function GameOver(ByVal tail As Integer) As Integer

Dim matched As Integer

Dim corner As Integer

Dim hLines(2) As Integer

matched = MyBase.GameOver(tail)

corner = Corners()

Console.WriteLine("You have matched " & corner & " corners.

You have matched " & matched & " numbers.")

For x = 0 To 2

hLines(x) = Horizontal(x)

Console.WriteLine("You have matched " & hLines(x) & " numbers in hoizontal line " & x + 1 & ".")

If corner = 4 Then

Return 3

ElseIf hLines(x) = 5 Then

Return 2

ElseIf matched = 15 Then

Return 1

End If

Next

Console.WriteLine("You only matched " & matched)

Return 0

End Function

'Checks the 4 corners

Private Function Corners() As Integer

Dim matched As Integer

Dim left As Boolean = False

Dim count As Integer = 0

Dim right As Boolean = False

For x = 0 To 2

While x <> 1 And left = False

If numbers(x, count) <> 0 Then

left = True

If callednum(x, count) Then

matched += 1

End If

Else count += 1

End If

End While

count = 0

While x <> 1 And right = False

If numbers(x, 8 - count) <> 0 Then

right = True

If callednum(x, 8 - count) Then

matched += 1

End If

Else count += 1

End If

End While

count = 0

left = False

right = False

Next

'OLD CORNERS CODE (DOES NOT WORK)

'For x = 0 To 2

' For y = 0 To 8

' If x <> 1 Then

' If first = False And numbers(x, y) <> 0 Then

' first = True

' If left = False Then

' If callednum(x, y) = True Then

' matched += 1

' left = True

' End If

' End If

' End If

' End If

' Next

' While right <> 8

' If last = False And numbers(x, 8 - right) <> 0 Then

' last = True

' If callednum(x, 8 - right) = True And rightbool = False Then

' matched += 1

' rightbool = True

' End If

' End If

' right += 1

' End While

' rightbool = False

' left = False

' last = False

' first = False

'Next

Return matched

End Function

'Checks horizontal lines

Private Function Horizontal(ByVal x As Integer) As Integer

Dim matched() As Integer = {0, 0, 0}

For y = 0 To 8

If callednum(x, y) = True Then

matched(x) += 1

End If

Next

Return matched(x)

Return False

End Function

End Class

Public Class NumberMachine

Private numberOrder(89) As Integer

Private current As Integer

Private back As Integer = 89 'changed from 74 to 89 so that now it does 1 - 90 instead of 1 - 75

Public Sub New()

Dim temp, num1, num2 As Integer

For x = 1 To 90 'changed from 75 to 90 so that now it does 1 - 90 instead of 1 - 75

numberOrder(x - 1) = x

Next x

For x = 1 To 1000

num1 = repo.NewRandom(0, 89)

num2 = repo.NewRandom(0, 89)

temp = numberOrder(num1)

numberOrder(num1) = numberOrder(num2)

numberOrder(num2) = temp

Next

End Sub

'Some numbers were wrong

Public Sub PracticeGame()

numberOrder(0) = 4

numberOrder(1) = 11

numberOrder(2) = 47

numberOrder(3) = 57

numberOrder(4) = 65

numberOrder(5) = 33

numberOrder(6) = 48

numberOrder(7) = 58

numberOrder(8) = 68

numberOrder(9) = 78

numberOrder(10) = 5

numberOrder(11) = 18

numberOrder(12) = 50

numberOrder(13) = 59

numberOrder(14) = 80

End Sub

Public Function nextBall() As Integer

If current < back Then

current += 1

Return numberOrder(current - 1)

Else

Return -1

End If

End Function

Public Function getNumbers() As Integer()

Return numberOrder

End Function

Public Function getBack() As Integer

Return current

End Function

End Class

Public Class repo

Public Shared Function NewRandom(ByVal lowerbound As Integer, ByVal upperbound As Integer)

Randomize() 'added // uncommented

Return CInt(Math.Floor((upperbound - lowerbound + 1) \* Rnd())) + lowerbound

End Function

End Class