A-level
COMPUTER SCIENCE
(7517/1E)
Paper 1 VB.NET

Skeleton Program
Module Module1
    Sub Main()
        Dim Choice As Integer = 0
        While Choice <> 9
            DisplayMenu()
            Choice = GetMainMenuChoice()
            Select Case Choice
                Case 1
                    Dim MyGame As New Game(False)
                Case 2
                    Dim MyGame As New Game(True)
            End Select
        End While
    End Sub

Public Sub DisplayMenu()
    Console.WriteLine("MAIN MENU")
    Console.WriteLine()  
    Console.WriteLine("1.  Start new game")
    Console.WriteLine("2.  Play training game")
    Console.WriteLine("9.  Quit")
    Console.WriteLine()  
    Console.Write("Please enter your choice: ")
End Sub

Public Function GetMainMenuChoice() As Integer
    Dim Choice As Integer
    Choice = CInt(Console.ReadLine())
    Console.WriteLine()
    Return Choice
End Function

Structure CellReference
    Dim NoOfCellsEast As Integer
    Dim NoOfCellsSouth As Integer
End Structure

Class Game
    Const NS As Integer = 4
    Const WE As Integer = 6
    Private Player As New Character
    Private Cavern As New Grid(NS, WE)
    Private Monster As New Enemy
    Private Flask As New Item
    Private Trap1 As New Trap
    Private Trap2 As New Trap
    Private TrainingGame As Boolean

    Public Sub New(ByVal IsATrainingGame As Boolean)
        TrainingGame = IsATrainingGame
        Randomize()
        SetUpGame()
        Play()
    End Sub

    Public Sub Play()
        Dim Count As Integer
        Dim Eaten As Boolean
        Dim FlaskFound As Boolean
        Dim MoveDirection As Char
Dim ValidMove As Boolean
Dim Position As CellReference
Eaten = False
FlaskFound = False
Cavern.Display(Monster.GetAwake)
Do
    Do
        DisplayMoveOptions()
        MoveDirection = GetMove()
        ValidMove = CheckValidMove(MoveDirection)
    Loop Until ValidMove
    If MoveDirection <> "M" Then
        Cavern.PlaceItem(Player.GetPosition, " ")
        Player.MakeMove(MoveDirection)
        Cavern.PlaceItem(Player.GetPosition, "*" )
        Cavern.Display(Monster.GetAwake)
        FlaskFound = Player.CheckIfSameCell(Flask.GetPosition)
        If FlaskFound Then
            DisplayWonGameMessage()
        End If
        Eaten = Monster.CheckIfSameCell(Player.GetPosition)
        'This selection structure checks to see if the player has
        triggered one of the traps in the cavern
        If Not Monster.GetAwake And Not FlaskFound And Not Eaten And
        (Player.CheckIfSameCell(Trap1.GetPosition) And Not Trap1.GetTriggered Or
        Player.CheckIfSameCell(Trap2.GetPosition) And Not Trap2.GetTriggered) Then
            Monster.ChangeSleepStatus()
            DisplayTrapMessage()
            Cavern.Display(Monster.GetAwake)
        End If
        If Monster.GetAwake And Not Eaten And Not FlaskFound Then
            Count = 0
            Do
                Cavern.PlaceItem(Monster.GetPosition, " ")
                Position = Monster.GetPosition
                Monster.MakeMove(Player.GetPosition)
                Cavern.PlaceItem(Monster.GetPosition, "M")
                If Monster.CheckIfSameCell(Flask.GetPosition) Then
                    Flask.SetPosition(Position)
                    Cavern.PlaceItem(Position, "F")
                End If
                Eaten = Monster.CheckIfSameCell(Player.GetPosition)
                Console.WriteLine()
                Console.WriteLine("Press Enter key to continue")
                Console.ReadLine()
                Count = Count + 1
            Loop Until Count = 2 Or Eaten
        End If
        If Eaten Then
            DisplayLostGameMessage()
        End If
    End If
Loop Until Eaten Or FlaskFound Or MoveDirection = "M"
End Sub

Public Sub DisplayMoveOptions()
    Console.WriteLine()
    Console.WriteLine("Enter N to move NORTH")
    Console.WriteLine("Enter S to move SOUTH")
    Console.WriteLine("Enter E to move EAST")
    Console.WriteLine("Enter W to move WEST")
Public Function GetMove() As Char
    Dim Move As Char
    Move = Console.ReadLine
    Console.WriteLine()
    Return Move
End Function

Public Function CheckValidMove(ByVal Direction As Char) As Boolean
    Dim ValidMove As Boolean
    ValidMove = True
    If Not (Direction = "N" Or Direction = "S" Or Direction = "W" Or Direction = "E" Or Direction = "M") Then
        ValidMove = False
    End If
    Return ValidMove
End Function

Public Sub SetUpGame()
    Dim Position As CellReference
    Cavern.Reset()
    If Not TrainingGame Then
        Position.NoOfCellsEast = 0
        Position.NoOfCellsSouth = 0
        Player.SetPosition(Position)
        Cavern.PlaceItem(Position, "*")
        Trap1.SetPosition(SetPositionOfItem("T"))
        Trap2.SetPosition(SetPositionOfItem("T"))
    End If
End Sub


```vbnet
Monster.SetPosition(SetPositionOfItem("M"))
Flask.SetPosition(SetPositionOfItem("F"))
Else
    Position.NoOfCellsEast = 4
    Position.NoOfCellsSouth = 2
    Player.SetPosition(Position)
    Cavern.PlaceItem(Position, "*")
    Position.NoOfCellsEast = 6
    Position.NoOfCellsSouth = 2
    Trap1.SetPosition(Position)
    Cavern.PlaceItem(Position, "T")
    Position.NoOfCellsEast = 4
    Position.NoOfCellsSouth = 3
    Trap2.SetPosition(Position)
    Cavern.PlaceItem(Position, "T")
    Position.NoOfCellsEast = 4
    Position.NoOfCellsSouth = 0
    Monster.SetPosition(Position)
    Cavern.PlaceItem(Position, "M")
    Position.NoOfCellsEast = 3
    Position.NoOfCellsSouth = 1
    Flask.SetPosition(Position)
    Cavern.PlaceItem(Position, "F")
End If
End Sub

Public Function GetNewRandomPosition() As CellReference
    Dim Position As CellReference
    Do
        Position.NoOfCellsSouth = Int(Rnd() * (NS + 1))
        Position.NoOfCellsEast = Int(Rnd() * (WE + 1))
    Loop Until Position.NoOfCellsSouth > 0 Or Position.NoOfCellsEast > 0
    Return Position
End Function
End Class

Class Grid
    Private NS As Integer
    Private WE As Integer
    Private CavernState(,,) As Char
    Public Sub New(ByVal S As Integer, ByVal E As Integer)
        NS = S
        WE = E
        ReDim CavernState(NS, WE)
    End Sub
    Public Sub Reset()
        Dim Count1 As Integer
        Dim Count2 As Integer
        For Count1 = 0 To NS
            For Count2 = 0 To WE
                CavernState(Count1, Count2) = " ">
            Next
        Next
    End Sub
    Public Sub Display(ByVal MonsterAwake As Boolean)
        Dim Count1 As Integer
        Dim Count2 As Integer
        For Count1 = 0 To NS
            For Count2 = 0 To WE
                CavernState(Count1, Count2) = " "
            Next
        Next
        Console.WriteLine(" --------- ")
    End Sub
End Class
```
For Count2 = 0 To WE
    If CavernState(Count1, Count2) = " " Or CavernState(Count1, Count2) = "*
    Then
        Console.Write("| " & CavernState(Count1, Count2))
    Else
        Console.Write("| ")
    End If
Next
Console.WriteLine("| ")
Next
Console.WriteLine(" ------------- 
End Sub

Public Sub PlaceItem(ByVal Position As CellReference, ByVal Item As Char)
    CavernState(Position.NoOfCellsSouth, Position.NoOfCellsEast) = Item
End Sub

Public Function IsCellEmpty(ByVal Position As CellReference) As Boolean
    If CavernState(Position.NoOfCellsSouth, Position.NoOfCellsEast) = " " Then
        Return True
    Else
        Return False
    End If
End Function
End Class

Class Enemy
    Inherits Item

    Private Awake As Boolean

    Public Overridable Sub MakeMove(ByVal PlayerPosition As CellReference)
        If NoOfCellsSouth < PlayerPosition.NoOfCellsSouth Then
            NoOfCellsSouth = NoOfCellsSouth + 1
        ElseIf NoOfCellsSouth > PlayerPosition.NoOfCellsSouth Then
            NoOfCellsSouth = NoOfCellsSouth - 1
        ElseIf NoOfCellsEast < PlayerPosition.NoOfCellsEast Then
            NoOfCellsEast = NoOfCellsEast + 1
        Else
            NoOfCellsEast = NoOfCellsEast - 1
        End If
    End Sub

    Public Function GetAwake() As Boolean
        Return Awake
    End Function

    Public Overridable Sub ChangeSleepStatus()
        If Awake Then
            Awake = False
        Else
            Awake = True
        End If
    End Sub

    Public Sub New()
        Awake = False
    End Sub
End Class
Class Character
Inherits Item

Public Sub MakeMove(ByVal Direction As Char)
Select Case Direction
Case "N"
    NoOfCellsSouth = NoOfCellsSouth - 1
Case "S"
    NoOfCellsSouth = NoOfCellsSouth + 1
Case "W"
    NoOfCellsEast = NoOfCellsEast - 1
Case "E"
    NoOfCellsEast = NoOfCellsEast + 1
End Select
End Sub
End Class

Class Trap
Inherits Item
Private Triggered As Boolean

Public Function GetTriggered() As Boolean
    Return Triggered
End Function

Public Sub New()
    Triggered = False
End Sub

Public Sub ToggleTrap()
    Triggered = Not Triggered
End Sub
End Class

Class Item
Protected NoOfCellsEast As Integer
Protected NoOfCellsSouth As Integer

Public Function GetPosition() As CellReference
    Dim Position As CellReference
    Position.NoOfCellsEast = NoOfCellsEast
    Position.NoOfCellsSouth = NoOfCellsSouth
    Return Position
End Function

Public Sub SetPosition(ByVal Position As CellReference)
    NoOfCellsEast = Position.NoOfCellsEast
    NoOfCellsSouth = Position.NoOfCellsSouth
End Sub

Public Function CheckIfSameCell(ByVal Position As CellReference) As Boolean
    If NoOfCellsEast = Position.NoOfCellsEast And NoOfCellsSouth = Position.NoOfCellsSouth Then
        Return True
    Else
        Return False
    End If
End Function
End Class
End Module