**Detailed code analysis**

**Module: Main**

Public rinc As Integer, cinc As Integer

Dim r() As Integer, c() As Integer

Sub StartGame()

' Initializes the game board and starts the game loop

' Creates the initial snake and sets up game elements

Range("B2:S26").Interior.Color = vbBlack

ReDim r(2)

ReDim c(2)

r(0) = 20: r(1) = 21: r(2) = 22

c(0) = 10: c(1) = 10: c(2) = 10

rinc = 0: cinc = 0

bindKeys

ShowSnake

AddApple

apples = 0

End Sub

Sub ShowSnake()

' Displays the snake on the game board

For i = UBound(r) To 1 Step -1

Cells(r(i), c(i)).Interior.Color = vbGreen

Next i

Cells(r(0), c(0)).Interior.Color = vbRed

End Sub

Sub MoveSnake()

' Moves the snake based on user input and handles game logic

If rinc <> 0 Or cinc <> 0 Then

tail = UBound(r)

Cells(r(tail), c(tail)).Interior.Color = vbBlack

' Repositions body of the snake

For i = tail To 1 Step -1

r(i) = r(i - 1)

c(i) = c(i - 1)

Next i

' Moves the head of the snake

r(0) = r(0) + rinc

c(0) = c(0) + cinc

' Checks the movement of the snake

If Cells(r(0), c(0)).Interior.Color = vbYellow Then

' Got the apple

apples = apples + 1

ReDim Preserve r(UBound(r) + 1)

ReDim Preserve c(UBound(c) + 1)

r(UBound(r)) = r(UBound(r) - 1)

c(UBound(c)) = c(UBound(c) - 1)

AddApple

ElseIf Cells(r(0), c(0)).Interior.Color <> vbBlack Then

' Game over if the snake collides with itself or game boundaries

StopTimer

MsgBox "Game Over"

Exit Sub

End If

ShowSnake

End If

End Sub

Sub AddApple()

' Adds a new apple to a random location on the game board

Randomize

arow = Int(Rnd \* 24) + 2

acol = Int(Rnd \* 14) + 2

Cells(arow, acol).Interior.Color = vbYellow

End Sub

**Module: Keys**

Sub bindKeys()

' Binds arrow keys to corresponding movement functions

Application.OnKey "{LEFT}", "moveLeft"

Application.OnKey "{RIGHT}", "moveRight"

Application.OnKey "{UP}", "moveUp"

Application.OnKey "{DOWN}", "moveDown"

End Sub

Sub moveLeft()

' Moves the snake left

If cinc <> 1 Then

cinc = -1

rinc = 0

MoveSnake

End If

End Sub

Sub moveRight()

' Moves the snake right

If cinc <> -1 Then

cinc = 1

rinc = 0

MoveSnake

End If

End Sub

Sub moveUp()

' Moves the snake up

If rinc <> 1 Then

cinc = 0

rinc = -1

MoveSnake

End If

End Sub

Sub moveDown()

' Moves the snake down

If rinc <> -1 Then

cinc = 0

rinc = 1

MoveSnake

End If

End Sub

Sub freeKeys()

' Unbinds arrow keys

Application.OnKey "{LEFT}"

Application.OnKey "{RIGHT}"

Application.OnKey "{UP}"

Application.OnKey "{DOWN}"

End Sub

**Module: Timer**

Option Explicit

#If Win64 Then

' Declare 64-bit compatible functions\*(source below)

Public Declare PtrSafe Function SetTimer Lib "User32" ( \_

ByVal hwnd As LongLong, \_

ByVal nIDEvent As LongLong, \_

ByVal uElapse As LongLong, \_

ByVal lpTimerFunc As LongLong) As LongLong

Public Declare PtrSafe Function KillTimer Lib "User32" ( \_

ByVal hwnd As LongLong, \_

ByVal nIDEvent As LongLong) As LongLong

Public TimerID As LongLong

#Else

' Declare 32-bit compatible functions

Public Declare PtrSafe Function SetTimer Lib "User32" ( \_

ByVal hwnd As Long, \_

ByVal nIDEvent As Long, \_

ByVal uElapse As Long, \_

ByVal lpTimerFunc As Long) As Long

Public Declare PtrSafe Function KillTimer Lib "User32" ( \_

ByVal hwnd As Long, \_

ByVal nIDEvent As Long) As Long

Public TimerID As Long

#End If

Sub StartTimer()

' Starts the game timer

If TimerID <> 0 Then

KillTimer 0, TimerID

TimerID = 0

End If

TimerID = SetTimer(0, 0, 200, AddressOf TimerEvent)

End Sub

Sub TimerEvent()

' Timer event handler to move the snake periodically

On Error Resume Next

Call MoveSnake

End Sub

Sub StopTimer()

' Stops the game timer and unbinds arrow keys

KillTimer 0, TimerID

TimerID = 0

freeKeys

End Sub

**Sources:**

\*: https://ramblings.mcpher.com/classes/promises-in-vba/settimer-and-vba/