Схема кода компонента GameGrid

Public Class GameGrid

Inherits DataGridView

Const WM\_LBUTTONDOWN As Integer = &H201

Const WM\_LBUTTONDBLCLK As Integer = &H203

Const WM\_KEYDOWN As Integer = &H100

Const VK\_LEFT As Integer = &H25

Const VK\_RIGHT As Integer = &H27

Const VK\_DOWN As Integer = &H28

Const VK\_UP As Integer = &H26

'custom events

Public Event IncrementScore(newPoints As Integer)

Public Event ShapeChanged(shapePoints() As Point, shapeColor As String)

Private rowCounter As Integer = 0

Private Const Caption As String = "Ошибка"

Shared ReadOnly stringManager As Resources.ResourceManager

Public Sub New()

Me.DoubleBuffered = True

End Sub

Protected Overrides Sub OnRowPrePaint(ByVal e As System.Windows.Forms.DataGridViewRowPrePaintEventArgs)

Try

e.PaintParts = e.PaintParts And Not DataGridViewPaintParts.Focus

MyBase.OnRowPrePaint(e)

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Protected Overrides Sub WndProc(ByRef m As System.Windows.Forms.Message)

Try

If m.Msg = WM\_LBUTTONDBLCLK OrElse m.Msg = WM\_LBUTTONDOWN Then

Return

ElseIf m.Msg = WM\_KEYDOWN Then

If m.WParam.ToInt32 = VK\_LEFT Then

MoveLeft()

ElseIf m.WParam.ToInt32 = VK\_RIGHT Then

MoveRight()

ElseIf m.WParam.ToInt32 = VK\_DOWN Then

MoveDown()

ElseIf m.WParam.ToInt32 = VK\_UP Then

RotateShape()

End If

Return

End If

MyBase.WndProc(m)

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Private WithEvents Tmr As New Timer With {.Interval = 500}

Private WithEvents Flashtmr As New Timer With {.Interval = 125}

Private flashCounter As Integer = 1

Private flashRow As Integer

Private missATick As Boolean = False

Private gameGrid()() As String

Private currentShape As Shape

Private ReadOnly listShapes As New List(Of Shape)

Private ReadOnly r As New Random

Private moveCounter As Integer = 0

Public Sub NewGame()

Try

Tmr.Stop()

Tmr.Interval = 500

listShapes.Clear()

moveCounter = 0

ReDim gameGrid(29)

For x As Integer = 1 To 30

Dim row(19) As String

gameGrid(x - 1) = DirectCast(row.Clone, String())

Next

NewShape()

currentShape = listShapes(0)

RaiseEvent ShapeChanged(currentShape.CurrentPoints, currentShape.ShapeColor)

rowCounter = 0

Tmr.Start()

Flashtmr.Start()

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Public Sub StopGame()

Try

Tmr.Stop()

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Private Sub NewShape()

Try

Dim sc() As String = {"R", "G", "B", "Y"}

Dim ns As New Shape(r.Next(0, 7), sc(r.Next(0, 4)))

listShapes.Add(ns)

'currentShape = ns

AddHandler ns.TouchDown, AddressOf CurrentShape\_TouchDown

HasChanged(gameGrid, False, -1)

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Public Sub MoveLeft()

Try

If CType(Form1.lblScore.Text, Integer) >= CType(Form1.RichTextBox4.Text, Integer) Then Exit Sub

If Form1.Button2.BackColor = Color.Red Then Exit Sub

If currentShape Is Nothing Then Return

gameGrid = currentShape.MoveLeft(gameGrid)

HasChanged(gameGrid, False, -1)

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Public Sub MoveRight()

Try

If CType(Form1.lblScore.Text, Integer) >= CType(Form1.RichTextBox4.Text, Integer) Then Exit Sub

If Form1.Button2.BackColor = Color.Red Then Exit Sub

If currentShape Is Nothing Then Return

gameGrid = currentShape.MoveRight(gameGrid)

HasChanged(gameGrid, False, -1)

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Public Sub MoveDown()

Try

If CType(Form1.lblScore.Text, Integer) >= CType(Form1.RichTextBox4.Text, Integer) Then Exit Sub

If Form1.Button2.BackColor = Color.Red Then Exit Sub

Do

For x As Integer = 0 To listShapes.Count - 1

If x > listShapes.Count - 1 Then Continue Do

gameGrid = listShapes(x).MoveDown(gameGrid)

HasChanged(gameGrid, False, -1)

Next

Exit Do

Loop

moveCounter += 1

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Public Sub RotateShape()

Try

If CType(Form1.lblScore.Text, Integer) >= CType(Form1.RichTextBox4.Text, Integer) Then Exit Sub

If Form1.Button2.BackColor = Color.Red Then Exit Sub

If currentShape Is Nothing Then Return

gameGrid = currentShape.RotateShape(gameGrid)

HasChanged(gameGrid, False, -1)

RaiseEvent ShapeChanged(currentShape.CurrentPoints, currentShape.ShapeColor)

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Private Sub Tmr\_Tick(sender As Object, e As EventArgs) Handles Tmr.Tick

Try

If missATick Then Return

If moveCounter >= 27 Then

moveCounter = 0

NewShape()

If listShapes.Count = 1 Then

currentShape = listShapes(0)

RaiseEvent ShapeChanged(currentShape.CurrentPoints, currentShape.ShapeColor)

End If

End If

MoveDown()

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Private Sub CurrentShape\_TouchDown(sender As Shape)

Try

If sender.CurrentPoints.Any(Function(p) p.Y < 0) Then Tmr.Stop()

RemoveHandler currentShape.TouchDown, AddressOf CurrentShape\_TouchDown

listShapes.Remove(sender)

If listShapes.Count < 1 Then

currentShape = Nothing

moveCounter = 27

Else

currentShape = listShapes(0)

RaiseEvent ShapeChanged(currentShape.CurrentPoints, currentShape.ShapeColor)

End If

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Private Sub Flashtmr\_Tick(sender As Object, e As EventArgs) Handles Flashtmr.Tick

Try

Select Case flashCounter

Case 1

flashRow = FindFullRow()

If flashRow > -1 Then

flashCounter = 2

HasChanged(gameGrid, True, flashRow)

End If

Case 2

flashCounter = 3

HasChanged(gameGrid, False, -1)

Case 3

flashCounter = 4

HasChanged(gameGrid, True, flashRow)

Case 4

Dim newGrid As New List(Of String())(gameGrid)

For Each p As Point In listShapes.Last.CurrentPoints

If p.Y > -1 Then

newGrid(p.Y)(p.X) = ""

End If

Next

Dim newRow(19) As String

newGrid.RemoveAt(flashRow)

newGrid.Insert(0, newRow)

missATick = True

gameGrid = newGrid.ToArray

flashCounter = 1

MoveDown()

HasChanged(gameGrid, False, -1)

missATick = False

rowCounter += 1

If rowCounter Mod 10 = 0 Then

Tmr.Interval -= 40

RaiseEvent IncrementScore(CInt(((1000 - Tmr.Interval) \* 0.35)))

ElseIf rowCounter Mod 5 = 0 Then

Tmr.Interval -= 20

RaiseEvent IncrementScore(CInt(((1000 - Tmr.Interval) \* 0.25)))

Else

RaiseEvent IncrementScore(CInt(((1000 - Tmr.Interval) \* 0.05)))

End If

End Select

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

Private Function FindFullRow() As Integer

For x As Integer = 29 To 0 Step -1

If gameGrid(x).All(Function(s) Not String.IsNullOrEmpty(s)) Then Return x

Next

Return -1

End Function

Private Sub HasChanged(grid As String()(), flash As Boolean, flashRow As Integer)

Try

Dim colors As New Dictionary(Of String, Color) From {{"R", Color.Red}, {"G", Color.Green}, {"B", Color.Blue}, {"Y", Color.Yellow}}

Dim flashColors As New Dictionary(Of String, Color) From {{"R", Color.FromArgb(255, 165, 165)}, {"G", Color.FromArgb(165, 255, 165)}, {"B", Color.FromArgb(165, 165, 255)}, {"Y", Color.FromArgb(255, 255, 230)}}

For y As Integer = 0 To 29

For x As Integer = 0 To 19

If String.IsNullOrEmpty(grid(y)(x)) Then

Me.Rows(y).Cells(x).Style.BackColor = Color.Black

Else

If Not flash OrElse (flash And Not flashRow = y) Then

Me.Rows(y).Cells(x).Style.BackColor = colors(grid(y)(x))

Else

Me.Rows(y).Cells(x).Style.BackColor = flashColors(grid(y)(x))

End If

End If

Next

Next

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

End Sub

End Class