

(FIGURA OPCIONAL)

**MINESWEEPER**

**Relatório da Prova de Aptidão Tecnológica**

Aluno: André Medeiros Melo

12º J nº36094

Índice

[1. Introdução 4](#_Toc389471816)

[2. Desenvolvimento do Minesweeper 5](#_Toc389471817)

[2.1. Porquê escolher o Minesweeper? 5](#_Toc389471818)

[2.2. Realização do Minesweeper 5](#_Toc389471819)

[2.2.1. MineCell 5](#_Toc389471820)

[2.2.2. Lodin 10](#_Toc389471821)

[2.2.3.Score 10](#_Toc389471822)

[2.2.4. Globais 12](#_Toc389471823)

[2.2.5. Tabuleiro 12](#_Toc389471824)

[3. Extras do jogo 21](#_Toc389471825)

[3.1. Nível de Dificuldade do Jogo (Tabuleiro) 21](#_Toc389471826)

[3.2. Estilo (Tabuleiro) 22](#_Toc389471827)

[4. Conclusão 24](#_Toc389471828)

1. Introdução

O projeto realizado foi a criação de uma versão do conhecido jogo “Minesweeper”, um velho jogo que já vinha nas antigas versões do Windows. A escolha deste projeto foi por dois motivos: recriar um jogo da minha infância e por à prova o meu raciocínio matemático.

Este projeto tem os seguintes objetivos:

- Recriar um jogo antigo;

- Criar os níveis de dificuldade;

- Fazer uma tabela de pontuações;

- Aprender a utilizar o método de gravar um jogo;

- Personalizar o jogo à escolha do jogador.

2. Desenvolvimento do Minesweeper

**2.1. Porquê escolher o Minesweeper?**

Escolhi o Minesweeper por ser um jogo clássico que contém muito raciocínio matemático na sua programação o que faz com que este projeto seja desafiante.

**2.2. Realização do Minesweeper**

Iniciei o projeto pela seleção do programa a utilizar e a minha escolha foi o Visual Basic, porque tenho experiência em trabalhar com este programa. Antes de começar a programar, procurei por imagens de Minas e Bandeiras para utilizar no Minesweeper e encontrei um website onde podia desenhar as imagens através de código. Assim, comecei a programar o meu Minesweeper pela imagem.

No geral, o meu trabalho contem cinco formulários:

- **MineCell:** onde ficou programado o desenho das minas, bandeiras, números e botões do tabuleiro. Também algumas das funções que contêm o próprio jogo Minesweeper. Esse formulário é um UserControl;

- **Tabuleiro:** onde o formulário terá toda a base do jogo;

- **Globais:** contém as variáveis globais para poder ser mais fácil organizar o trabalho.

- **Score:** contém o projeto das pontuações;

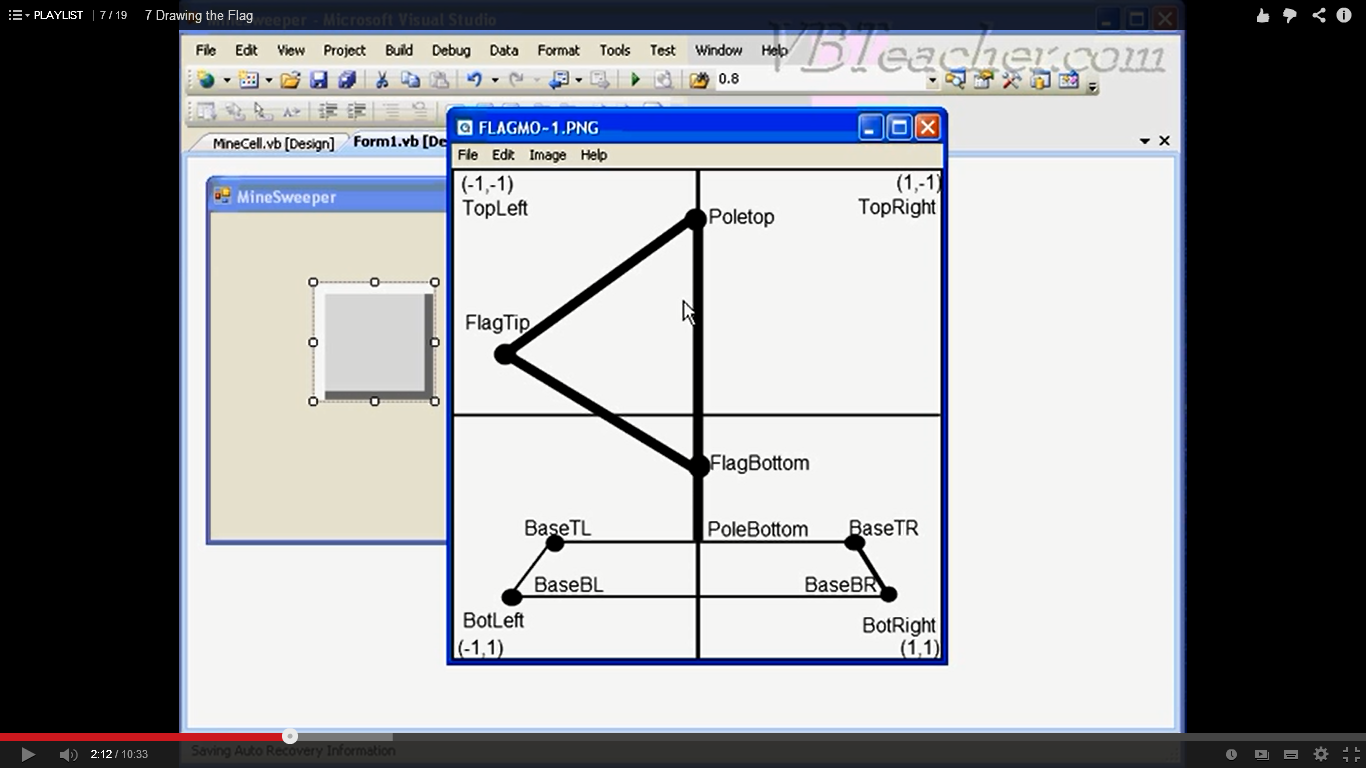
- **Lodin:** algo que foi necessário conter para ligar aos “saves” do jogo.

**2.2.1. MineCell**

Neste formulário foram criadas as imagens necessárias ao jogo Minesweeper (Mina, Bandeira, Número e o Botão) em linha de código.

Cada imagem foi criada com base numa imagem que continha os eixos X e Y do UserControl para se ter noção dos pontos que se tinha de utilizar para desenhar as imagens. A Figura 1 demonstra como foi desenhada a bandeira a partir de vários pontos do referencial para se poder criar a base, o poste e a bandeira.

Figura 1. Desenho da bandeira



**Programação utilizada para desenhar a bandeira**

Case VisaoMineCell.Bandeira

With e.Graphics

.ResetTransform()

.TranslateTransform(Me.Width / 2, Me.Height / 2)

.ScaleTransform(Me.Width / 2, Me.Height / 2)

.Clear(Color.LightGray)

'Contronos

Dim TopLeft As New Point(-1, -1)

Dim TopRight As New Point(1, -1)

Dim BotLeft As New Point(-1, 1)

Dim BotRight As New Point(1, 1)

Dim mPen As New Pen(Color.Gray, 0.3)

.DrawLine(mPen, BotRight, BotLeft)

.DrawLine(mPen, BotRight, TopRight)

mPen = New Pen(Color.White, 0.3)

.DrawLine(mPen, TopLeft, BotLeft)

.DrawLine(mPen, TopLeft, TopRight)

'Pontos da Bandeira

Dim PoleTop As New PointF(0, -0.7)

Dim PoleBottom As New PointF(0, 0.5)

Dim FlagTip As New Point(-0.7, -0.3)

Dim FlagBottom As New PointF(0, 0.1)

Dim BaseBT As New PointF(-0.5, 0.5)

Dim BaseBL As New PointF(-0.7, 0.7)

Dim BaseTR As New PointF(0.5, 0.5)

Dim BaseBR As New PointF(0.7, 0.7)

'pole

mPen = New Pen(Color.Brown, 0.1)

.DrawLine(mPen, PoleTop, PoleBottom)

'Bandeira

Dim FPts() As PointF = {PoleTop, FlagTip, FlagBottom}

Dim FBrush As New SolidBrush(Color.Red)

.FillPolygon(FBrush, FPts)

'Base

Dim BPts() As PointF = {BaseBL, BaseBT, BaseTR, BaseBR}

FBrush.Color = Color.Black

.FillPolygon(FBrush, BPts)

End With

**Programação utilizada para desenhar a mina:**

Case VisaoMineCell.Mina

With e.Graphics

.ResetTransform()

.TranslateTransform(Me.Width / 2, Me.Height / 2)

.ScaleTransform(Me.Width / 2, Me.Height / 2)

.Clear(Color.WhiteSmoke)

'Circulo

Dim CRect As New RectangleF(-0.6, -0.6, 1.2, 1.2)

Dim CBrush As New SolidBrush(Color.Black)

.FillEllipse(CBrush, CRect)

'Picos

Dim IRad As Single = 0.5

Dim ERad As Single = 0.8

Dim PPen As New Pen(Color.Black, 0.15)

PPen.EndCap = Drawing2D.LineCap.Round 'Formato das Pontas

For Angulo As Single = 0 To 1.75 \* Math.PI Step 0.25 \* Math.PI

Dim Interior As New PointF(IRad \* Math.Cos(Angulo), IRad \* Math.Sin(Angulo))

Dim Esterior As New PointF(ERad \* Math.Cos(Angulo), ERad \* Math.Sin(Angulo))

.DrawLine(PPen, Interior, Esterior)

Next

'Brilho

Dim HRect As New RectangleF(-0.3, -0.3, 0.2, 0.2)

Dim WBrush As New SolidBrush(Color.White)

.FillRectangle(WBrush, HRect)

'Contronos da caixa da mina

Dim BRect As New Rectangle(-1, -1, 2, 2)

Dim BPen As New Pen(Color.Gray, 0.05)

.DrawRectangle(BPen, BRect)

End With

Para desenhar a mina, tive de usar as funções matemáticas cosseno e seno devido aos ângulos dos picos da mina. Para o círculo da mina não foi necessário porque foi feito numa simples linha de código.

**Programação utilizada para os números:**

Case VisaoMineCell.Numero

Dim NColors() As Color = {Color.Blue, Color.Green, Color.Red, Color.Navy, Color.DarkGreen, Color.DarkRed, Color.Brown, Color.Black}

With e.Graphics

.ResetTransform()

.TranslateTransform(Me.Width / 2, Me.Height / 2)

.ScaleTransform(Me.Width / 2, Me.Height / 2)

.Clear(Color.WhiteSmoke)

'Sair o Numero

If mNumero > 0 And mNumero <= 8 Then

Dim NBrush As New SolidBrush(NColors(mNumero - 1))

Dim myFont As New Font("Times", 1.5, FontStyle.Bold, GraphicsUnit.World)

Dim SS As SizeF = .MeasureString(mNumero.ToString, myFont)

.DrawString(mNumero.ToString, myFont, NBrush, -SS.Width / 2, -SS.Height / 2)

End If

'Controno da caixa

Dim BRect As New Rectangle(-1, -1, 2, 2)

Dim BPen As New Pen(Color.Gray, 0.05)

.DrawRectangle(BPen, BRect)

End With

Os números foram mais simples de programar porque foi só preciso introduzir o estilo de letra para cada número de 1 a 8 - máximo que pode sair- e escolher a cor usando um arrey.

**Programação utilizada para os botões:**

Case VisaoMineCell.Botao

With e.Graphics

.ResetTransform()

.TranslateTransform(Me.Width / 2, Me.Height / 2)

.ScaleTransform(Me.Width / 2, Me.Height / 2)

.Clear(mButtonColor)

'Contronos

Dim TopLeft As New Point(-1, -1)

Dim TopRight As New Point(1, -1)

Dim BotLeft As New Point(-1, 1)

Dim BotRight As New Point(1, 1)

Dim mPen As New Pen(Color.Gray, 0.3)

.DrawLine(mPen, BotRight, BotLeft)

.DrawLine(mPen, BotRight, TopRight)

mPen = New Pen(Color.White, 0.3)

.DrawLine(mPen, TopLeft, BotLeft)

.DrawLine(mPen, TopLeft, TopRight)

End With

End Select

O desenho do botão foi muito simples, alteraram-se as cores nos contornos para dar um estilo de botão.

**2.2.2. Lodin**

Public Class Lodin

Private Sub Lodin\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Me.CenterToScreen()

If Not System.IO.Directory.Exists(Globais.pasta) Then

System.IO.Directory.CreateDirectory(Globais.pasta)

End If

If Not System.IO.File.Exists(Globais.pasta + "\" + Globais.Gravado) Then

System.IO.File.Create(Globais.pasta + "\" + Globais.Gravado)

End If

Me.Close()

End Sub

End Class

Esta parte do código serve para quando se abrir o Minesweeper confirmar se já existe a pasta e o ficheiro “.txt” onde ficam gravados os scores do jogo, caso não exista a pasta cria uma nova e o ficheiro “.txt”. Este formulário é o primeiro a ser aberto.

**2.2.3.Score**

Public Class Score

Private Sub Score\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

ValoresScore()

End Sub

Private Sub ValoresScore()

Facil1.Text = "1º - " & Globais.Fprimeiro & " segundos"

Facil2.Text = "2º - " & Globais.Fsecundo & " segundos"

Facil3.Text = "3º - " & Globais.Fterceiro & " segundos"

Medio1.Text = "1º - " & Globais.Mprimeiro & " segundos"

Medio2.Text = "2º - " & Globais.Msecundo & " segundos"

Medio3.Text = "3º - " & Globais.Mterceiro & " segundos"

Dificil1.Text = "1º - " & Globais.Dprimeiro & " segundos"

Dificil2.Text = "2º - " & Globais.Dsecundo & " segundos"

Dificil3.Text = "3º - " & Globais.Dterceiro & " segundos"

End Sub

End Class

Ao clicar em “Pontuações” no menu do Minesweeper, abre-se uma janela que demostra os melhores resultados conseguidos até à data.

Private Sub Eliminar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Eliminar.Click

Globais.Fprimeiro = 0

Globais.Fsecundo = 0

Globais.Fterceiro = 0

Globais.Mprimeiro = 0

Globais.Msecundo = 0

Globais.Mterceiro = 0

Globais.Dprimeiro = 0

Globais.Dsecundo = 0

Globais.Dterceiro = 0

ValoresScore()

End Sub.

Ao clicar em “Eliminar” as pontuações são reiniciadas e faz um *refresh* à janela, por outras palavras mete tudo a zero.

**2.2.4. Globais**

Public Class Globais

Public Shared teste As Integer

Public Shared Tamanho As Integer = 1

Public Shared pasta As String = "C:\Users\André Melo\Desktop\Minesweeper\Save"

Public Shared Gravado As String = "Gravacao.txt"

Public Shared ECor As Color = Color.LightGray

Public Shared Fprimeiro As Integer

Public Shared Fsecundo As Integer

Public Shared Fterceiro As Integer

Public Shared Mprimeiro As Integer

Public Shared Msecundo As Integer

Public Shared Mterceiro As Integer

Public Shared Dprimeiro As Integer

Public Shared Dsecundo As Integer

Public Shared Dterceiro As Integer

End Class

Como foi referido no início, este Code File só contém variáveis globais para facilitar o trabalho de programar. Estas variáveis vão ser usadas/alteradas em vários formulários (Tabuleiro, Score, Lodin e MineCell).

**2.2.5. Tabuleiro**

Esta parte por ser a base do programa que liga todos os formulários é muito complexa e, por isso, vai ser explicada em várias partes. Começa-se por explicar onde começa o programa a percorrer as linhas de código.

Public Class Tabuleiro

Dim CellSize As Integer = 25

Dim BoardRows As Integer = 9

Dim BoardCols As Integer = 9

Dim MinasJogo As Integer = 10

Dim Secundos As Integer = 0

Dim MinasMarcadas As Integer = 0

Dim MineField(8, 8) As MineCell

…

Logo no início introduza uns dados por tabuleiro do jogo sendo assim o seu default.

Ao iniciar o private sub (BuildBoard()) que está no arranque do jogo ativam-se diversas privade sub e funções que vou explicando à medida que vão sendo chamadas.

Private Sub BuildBoard()

PanelMine.Enabled = True

Secundos = 0

Timer1.Enabled = False

MinasMarcadas = 0

DoLabels()

Me.Hide()

carregar()

'loop entre as colunas e linhas

ReDim MineField(BoardRows - 1, BoardCols - 1)

PanelMine.Controls.Clear()

For Row As Integer = 0 To BoardRows - 1

For col As Integer = 0 To BoardCols - 1

Dim C As New MineCell

PanelMine.Controls.Add(C)

C.Left = CellSize \* col

C.Top = CellSize \* Row

C.Width = CellSize

C.Height = CellSize

C.TivouMina = False

C.Numero = 0

C.X = col

C.Y = Row

MineField(Row, col) = C

AddHandler C.MouseClick, AddressOf MineClick

Next

Next

Esta parte criou o tabuleiro do Minesweeper, onde estão todos botões para serem selecionados, o contador de tempo, a contagem das minas, o menu com várias funções (como é o caso do das Pontuações que já foi referido)

'Random Mine

Dim RX As New Random

For i As Integer = 1 To MinasJogo

Dim X As Integer = 0

Dim Y As Integer = 0

Do

X = RX.Next(0, BoardCols)

Y = RX.Next(0, BoardRows)

Loop Until Not MineField(Y, X).TivouMina

MineField(Y, X).TivouMina = True

'CHEATS

'MineField(Y, X).Ver = MineCell.VisaoMineCell.Mina

'MineField(Y, X).ButtonColor = Color.Red

Next

Aqui criaram-se as minas e colocou-se um random pelo tabuleiro.

'Contar as Minas

For Row As Integer = 0 To BoardRows - 1

For Col As Integer = 0 To BoardCols - 1

If Not MineField(Row, Col).TivouMina Then

For R As Integer = Row - 1 To Row + 1

For C As Integer = Col - 1 To Col + 1

If R >= 0 And R < BoardRows And C >= 0 And C < BoardCols And Not (Row = R And Col = C) Then

If MineField(R, C).TivouMina Then

MineField(Row, Col).Numero += 1

End If

End If

Next

Next

End If

'CHEATS

'If MineField(Row, Col).Numero > 0 Then MineField(Row, Col).Ver = MineCell.VisaoMineCell.Numero

'If MineField(Row, Col).Numero > 0 Then MineField(Row, Col).ButtonColor = Color.Green

Next

Next

Esta parte faz a contagem do número de minas em redor de cada célula. Caso não tivesse nenhuma mina ao redor da célula devolve um espaço em branco.

**Nota:** As duas linhas de código abaixo de onde está escrito CHEATS que estão comentados (na colocação das minas e na contagem de quantas minas estavam ao redor de uma célula) foram utilizados para verificar se o programa estava a funcionar. Sendo a primeira linha de código para a mostrar logo a mina e o número, a seguinte irá mudar a cor para saber onde se situam as células das minas (a vermelho) e a dos números (verde).

'Dimencionar o Painel

Me.Width = BoardCols \* CellSize

Do Until Me.PanelMine.Width = BoardCols \* CellSize

Me.Width += 1

Loop

Me.Height = BoardRows \* CellSize

Do Until Me.PanelMine.Height = BoardRows \* CellSize

Me.Height += 1

Loop

Me.Show()

End Sub

Esta parte dimensionou o tabuleiro de acordo com o nível de dificuldade (por default é o tamanho e dificuldade fácil).

Private Sub DoLabels()

LabelRelogio.Text = Secundos.ToString

LabelMinas.Text = (MinasJogo - MinasMarcadas).ToString

End Sub

Aqui apenas atualiza os valores das duas labels (do Relógio e do Número de minas em jogo). MinasJogo é o número de minas que o jogo contém e MinasMarcadas são as bandeiras colocadas no jogo, para facilitar o jogo ao revelar quantas minas poderá ainda ter o tabuleiro.

Private Sub carregar()

Dim filereader As System.IO.StreamReader

filereader = My.Computer.FileSystem.OpenTextFileReader(Globais.pasta + "\" + Globais.Gravado)

Dim lugar As Integer

lugar = CInt(filereader.ReadLine())

Globais.Fprimeiro = lugar

lugar = CInt(filereader.ReadLine())

Globais.Fsecundo = lugar

lugar = CInt(filereader.ReadLine())

Globais.Fterceiro = lugar

lugar = CInt(filereader.ReadLine())

Globais.Mprimeiro = lugar

lugar = CInt(filereader.ReadLine())

Globais.Msecundo = lugar

lugar = CInt(filereader.ReadLine())

Globais.Mterceiro = lugar

lugar = CInt(filereader.ReadLine())

Globais.Dprimeiro = lugar

lugar = CInt(filereader.ReadLine())

Globais.Dsecundo = lugar

lugar = CInt(filereader.ReadLine())

Globais.Dterceiro = lugar

filereader.Close()

End Sub

Esta parte deu mais trabalho a pôr a funcionar, o que faz é ir buscar os dados do save e introduzi-los nas variáveis globais, atualizando, assim, o formulário Score quando for aberto.

No meio do private sub BuildBroader() contém um AddHandler que ativa estas partes do código (AddHandler C.MouseClick, AddressOf MineClick) onde cada vez que clicas no botão direito ou esquerdo do rato numa das células irá fazer uma função e ativa o contador de segundos.

ElseIf e.Button = Windows.Forms.MouseButtons.Right Then

Select Case M.Ver

Case MineCell.VisaoMineCell.Botao

M.Ver = MineCell.VisaoMineCell.Bandeira

MinasMarcadas += 1

DoLabels()

M.Refresh()

Case Else

M.Ver = MineCell.VisaoMineCell.Botao

MinasMarcadas -= 1

DoLabels()

M.Refresh()

End Select

End If

End Sub

Ao clicar com o botão direito do rato irá ser colocada uma bandeira e diminui o número da label onde indica o número de minas que estão em jogo. Ao clicar outra vez no mesmo sítio com o botão direito do rato irá remover a bandeira e sobe o valor da label que indica o número de minas.

If e.Button = Windows.Forms.MouseButtons.Left Then

If M.TivouMina Then

For r As Integer = 0 To BoardRows - 1

For c As Integer = 0 To BoardCols - 1

Dim CC As MineCell = MineField(c, r)

If CC.TivouMina Then

CC.Ver = MineCell.VisaoMineCell.Mina

CC.Refresh()

Timer1.Enabled = False

PanelMine.Enabled = False

End If

Next

Next

ElseIf M.Numero > 0 Then

M.Ver = MineCell.VisaoMineCell.Numero

M.Refresh()

If Ganhar() Then

Timer1.Enabled = False

PanelMine.Enabled = False

Posicao()

Guardar()

End If

ElseIf M.Numero = 0 Then

EspacoBranco(M)

If Ganhar() Then

Timer1.Enabled = False

PanelMine.Enabled = False

Posicao()

Guardar()

End If

End If

M.Enabled = False

Ao clicar com o botão esquerdo do rato irá percorrer um if, em que confirma se clicaste numa mina, número ou espaço em branco. Se clicar em cima de uma mina o jogo termina, parando o timer (e não irá colocar nas pontuações o tempo que levaste). Se for um número, irá apenas demonstrar o número e o jogo continua. Se clicar num espaço em branco ativa este código que irá abrir vários espaços em branco até “bater” em números permitindo que o jogo continue:

Private Sub EspacoBranco(ByVal M As MineCell)

M.Ver = MineCell.VisaoMineCell.Numero

For R As Integer = M.Y - 1 To M.Y + 1

For C As Integer = M.X - 1 To M.X + 1

If R >= 0 And R < BoardRows And C >= 0 And C < BoardCols Then

Dim MC As MineCell = MineField(R, C)

If MC.Ver = MineCell.VisaoMineCell.Botao Then

If MC.Numero = 0 Then

EspacoBranco(MC)

Else

MC.Ver = MineCell.VisaoMineCell.Numero

End If

End If

MC.Refresh()

MC.Enabled = False

End If

Next

Next

End Sub

Se clicares no último espaço/número e o que sobrar forem apenas as minas, o jogo termina com uma vitória e ativa os private sub’s Posicao() e Guardar().

Private Sub Posicao()

Globais.teste = LabelRelogio.Text

Select Case Globais.Tamanho

Case 1

If Globais.teste < Globais.Fprimeiro Or Globais.Fprimeiro = 0 Then

Globais.Fterceiro = Globais.Fsecundo

Globais.Fsecundo = Globais.Fprimeiro

Globais.Fprimeiro = Globais.teste

ElseIf Globais.teste < Globais.Fsecundo Or Globais.Fsecundo = 0 Then

Globais.Fterceiro = Globais.Fsecundo

Globais.Fsecundo = Globais.teste

ElseIf Globais.teste < Globais.Fterceiro Or Globais.Fterceiro = 0 Then

Globais.Fterceiro = Globais.teste

End If

Case 2

If Globais.teste < Globais.Mprimeiro Or Globais.Mprimeiro = 0 Then

Globais.Mterceiro = Globais.Msecundo

Globais.Msecundo = Globais.Mprimeiro

Globais.Mprimeiro = Globais.teste

ElseIf Globais.teste < Globais.Msecundo Or Globais.Msecundo = 0 Then

Globais.Mterceiro = Globais.Msecundo

Globais.Msecundo = Globais.teste

ElseIf Globais.teste < Globais.Mterceiro Or Globais.Mterceiro = 0 Then

Globais.Mterceiro = Globais.teste

End If

Case Else

If Globais.teste < Globais.Dprimeiro Or Globais.Dprimeiro = 0 Then

Globais.Dterceiro = Globais.Dsecundo

Globais.Dsecundo = Globais.Dprimeiro

Globais.Dprimeiro = Globais.teste

ElseIf Globais.teste < Globais.Dsecundo Or Globais.Dsecundo = 0 Then

Globais.Dterceiro = Globais.Dsecundo

Globais.Dsecundo = Globais.teste

ElseIf Globais.teste < Globais.Dterceiro Or Globais.Dterceiro = 0 Then

Globais.Dterceiro = Globais.teste

End If

End Select

End Sub

O private sub Posicao() irá atualizar as classificações do jogo caso for melhor de uma das posições. Caso for o melhor irá substituir a melhor posição e o valor do 1º irá para 2º, e o que tava no 2º irá para 3º e o que estava no 3º fica eliminado. Antes de confirmar e atualizar as classificações confirma em que grau de dificuldade estava, mais à frente será explicado como se faz essa confirmação.

Private Sub Guardar()

'gravar pontuacoes

Dim file As System.IO.StreamWriter = Nothing

file = My.Computer.FileSystem.OpenTextFileWriter(Globais.pasta + "\" + Globais.Gravado, False)

file.WriteLine(Globais.Fprimeiro)

file.WriteLine(Globais.Fsecundo)

file.WriteLine(Globais.Fterceiro)

file.WriteLine(Globais.Mprimeiro)

file.WriteLine(Globais.Msecundo)

file.WriteLine(Globais.Mterceiro)

file.WriteLine(Globais.Dprimeiro)

file.WriteLine(Globais.Dsecundo)

file.WriteLine(Globais.Dterceiro)

file.Close()

End Sub

O private sub Guardar() atualiza/grava no save do Score os novos valores.

Ficam, assim, descrita a parte mais importante do Minesweeper para se poder jogar, de seguida apresentam-se alguns extras do jogo.

**3. Extras do jogo**

Abaixo explica-se um pormenor que ocorre quando se tenta desligar o Minesweeper e o jogo grava as pontuações para garantir que está bem atualizado.

Private Sub Tabuleiro\_FormClosing(ByVal sender As System.Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles MyBase.FormClosing

Guardar()

End Sub

Private sub Guardar() já está referido em cima como funciona.

**3.1. Nível de Dificuldade do Jogo (Tabuleiro)**

No menu “Jogo” tem as opções fácil, médio e difícil, e cada um deles aumenta o número de minas que contém e o número de colunas e linhas que o jogo irá conter, originando assim mais casas para selecionar.

Private Sub FacilToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FacilToolStripMenuItem.Click

Globais.Tamanho = 1

FacilToolStripMenuItem.Text = "Facil \*"

MedioToolStripMenuItem.Text = "Medio"

DificilToolStripMenuItem.Text = "Dificil"

ComecarJogo(9, 9, 10)

End Sub

Private Sub MedioToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MedioToolStripMenuItem.Click

Globais.Tamanho = 2

FacilToolStripMenuItem.Text = "Facil"

MedioToolStripMenuItem.Text = "Medio \*"

DificilToolStripMenuItem.Text = "Dificil"

ComecarJogo(18, 18, 50)

End Sub

Private Sub DificilToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DificilToolStripMenuItem.Click

Globais.Tamanho = 3

FacilToolStripMenuItem.Text = "Facil"

MedioToolStripMenuItem.Text = "Medio"

DificilToolStripMenuItem.Text = "Dificil \*"

ComecarJogo(22, 22, 75)

End Sub

Ao clicar numa das dificuldades irá colocar um “ \* ” à frente do nome para saber qual o grau de dificuldade se está a jogar. A variável global “Tamanho” está a ser usada para indicar ao programa em que dificuldade está caso o jogador ganhe para atualizar o score no sítio certo.

Private Sub ComecarJogo(ByVal Rows As Integer, ByVal Cols As Integer, ByVal Minas As Integer)

BoardCols = Cols

BoardRows = Rows

MinasJogo = Minas

BuildBoard()

End Sub

Depois de clicar numa das dificuldades, o tabuleiro é redimensionado e recriado.

**3.2. Estilo (Tabuleiro)**

No menu “Estilo”, pode-se trocar a cor de fundo do tabuleiro sendo “Clássico” a cor de default ao iniciar. Esta parte contém uma variável global “ECor” onde no formulário MineCell está a ser chamada no momento em que se altera a cor do botão. Ao trocar a cor (por exemplo azul) irá alterar o valor da variável ECor e irá iniciar o sub private “Buildboard” no formulário Tabuleiro para reiniciar o jogo (caso já tenha começado) e ao criar o jogo outra vez já tem a cor alterada. Também o menu irá trocar um “\*” (como foi feito no nível de dificuldade) pela cor que foi selecionada, por outras palavras, indica a cor que está a ser usada.

Private Sub ClassicoToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ClassicoToolStripMenuItem.Click

Globais.ECor = Color.LightGray

ClassicoToolStripMenuItem.Text = "Classico \*"

VerdeToolStripMenuItem.Text = "Verde"

AzulToolStripMenuItem.Text = "Azul"

VermelhoToolStripMenuItem.Text = "Vermelho"

MagentaToolStripMenuItem.Text = "Magenta"

BuildBoard()

End Sub

Private Sub VerdeToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles VerdeToolStripMenuItem.Click

Globais.ECor = Color.Green

ClassicoToolStripMenuItem.Text = "Classico"

VerdeToolStripMenuItem.Text = "Verde \*"

AzulToolStripMenuItem.Text = "Azul"

VermelhoToolStripMenuItem.Text = "Vermelho"

MagentaToolStripMenuItem.Text = "Magenta"

BuildBoard()

End Sub

…

Não vale a pena mostrar tudo porque o processo é sempre o mesmo só alterando o valor da variável global e onde está escrito o “\*”.

4. Conclusão

Posso concluir que foi divertido fazer um jogo da minha infância, acrescentando um pouco de tecnologia dos dias de hoje sem alterar a dinâmica do jogo. Aprendi que com o Visual Basic é possível criar imagens com base em matemática. O raciocínio matemático foi, sem dúvida, um elemento muito importante deste projeto.

Foram feitos muitos testes ao jogo criado que demonstraram que o mesmo não tem falhas e que proporciona momentos de diversão ao utilizador.