

Anexo 1. Programa de estabelecimento de situação experimental

' C:\Documents and Settings\Sylvio B.
Cruz\Desktop\Experimentos\Experimento 12B\Pro
grama\Exp.ebs

' Generated on: 2/22/2010 6:35:02

' This file generated with E-Studio
interface.

' E-Prime Copyright © 2002 Psychology
Software Tools.

' ALL RIGHTS RESERVED

' Legal use of this experiment script
requires a full E-Prime or Runtime License.

' Author:

' Klaus Tiedemann (Home)

Option CStrings On
Dim ebContext as Context

'-----
' Class Declarations
'-----

'-----
' InitTextDisplayDefaults
'-----

Sub InitTextDisplayDefaults(theTextDisplay As
TextDisplay)

 If theTextDisplay Is Nothing Then Exit
Sub

 theTextDisplay.X = "center"
 theTextDisplay.Y = "center"
 theTextDisplay.Width = "100%"
 theTextDisplay.Height = "100%"

 theTextDisplay.ForeColor =
CColor("black")
 theTextDisplay.BackColor =
CColor("white")
 theTextDisplay.BackStyle = "opaque"
 theTextDisplay.BorderColor =
CColor("black")
 theTextDisplay.BorderWidth =
CLng("0")
 theTextDisplay.XAlign = "center"
 theTextDisplay.YAlign = "center"
 theTextDisplay.AlignHorizontal =
"center"
 theTextDisplay.AlignVertical = "center"
 theTextDisplay.WordWrap = True
 theTextDisplay.ClearAfter =
CLogical("No")
 theTextDisplay.FontName = "Courier
New"
 theTextDisplay.FontSize = "18"
 theTextDisplay.FontBold =
CLogical("Yes")
 theTextDisplay.FontItalic =
CLogical("No")
 theTextDisplay.FontUnderline =
CLogical("No")
 theTextDisplay.FontStrikeout =
CLogical("No")

End Sub

'-----
' InitImageDisplayDefaults
'-----

Sub InitImageDisplayDefaults(theImageDisplay
As ImageDisplay)

 If theImageDisplay Is Nothing Then Exit
Sub

 theImageDisplay.X = "center"
 theImageDisplay.Y = "center"
 theImageDisplay.Width = "100%"
 theImageDisplay.Height = "100%"
 theImageDisplay.BackColor =
CColor("black")
 theImageDisplay.BackStyle = "opaque"

```

        theImageDisplay.BorderColor =
CColor("black")
        theImageDisplay.BorderWidth =
CLng("0")
        theImageDisplay.XAlign = "center"
        theImageDisplay.YAlign = "center"
        theImageDisplay.AlignHorizontal =
"center"
        theImageDisplay.AlignVertical =
"center"
        theImageDisplay.ClearAfter =
CLogical("No")
        theImageDisplay.UseSourceColorKey =
CLogical("No")
        theImageDisplay.SourceColorKey =
CColor("black")
        theImageDisplay.MirrorLeftRight =
CLogical("No")
        theImageDisplay.MirrorUpDown =
CLogical("No")
        theImageDisplay.Stretch =
CLogical("No")
End Sub

'-----
'InitSlideStateDefaults
'-----
Sub InitSlideStateDefaults(theSlideState As
SlideState)

    If theSlideState Is Nothing Then Exit Sub

        theSlideState.X = "center"
        theSlideState.Y = "center"
        theSlideState.Width = "100%"
        theSlideState.Height = "100%"
        theSlideState.BackColor =
CColor("white")
        theSlideState.BackStyle = "opaque"
        theSlideState.BorderColor =
CColor("black")
        theSlideState.BorderWidth = CLng("0")
        theSlideState.XAlign = "center"
        theSlideState.YAlign = "center"
        theSlideState.ClearAfter =
CLogical("No")
        theSlideState.Enabled = CLogical("Yes")

```

```

End Sub

'-----
'InitSlideVisualStimDefaults
'-----
Sub
InitSlideVisualStimDefaults(theSlideVisualStim
As SlideVisualStim)

    If theSlideVisualStim Is Nothing Then
Exit Sub

        theSlideVisualStim.X = "center"
        theSlideVisualStim.Y = "center"
        theSlideVisualStim.Width = "25%"
        theSlideVisualStim.Height = "25%"
        theSlideVisualStim.ForeColor =
CColor("black")
        theSlideVisualStim.BackColor =
CColor("white")
        theSlideVisualStim.BackStyle =
"opaque"
        theSlideVisualStim.BorderColor =
CColor("black")
        theSlideVisualStim.BorderWidth =
CLng("0")
        theSlideVisualStim.XAlign = "center"
        theSlideVisualStim.YAlign = "center"
        theSlideVisualStim.AlignHorizontal =
"center"
        theSlideVisualStim.AlignVertical =
"center"
End Sub

'-----
'InitSlideTextDefaults
'-----
Sub InitSlideTextDefaults(theSlideText As
SlideText)

    If theSlideText Is Nothing Then Exit Sub

        InitSlideVisualStimDefaults theSlideText

        theSlideText.WordWrap = True
        theSlideText.FontName = "Courier New"
        theSlideText.FontSize = "18"
        theSlideText.FontBold =
CLogical("Yes")
        theSlideText.FontItalic = CLogical("No")

```

```

        theSlideText.FontUnderline =
CLogical("No")
        theSlideText.FontStrikeout =
CLogical("No")

End Sub

'-----
'InitSlideImageDefaults
'-----
Sub InitSlideImageDefaults(theSlideImage As
SlideImage)

    If theSlideImage Is Nothing Then Exit
Sub

    InitSlideVisualStimDefaults
theSlideImage

        theSlideImage.Stretch = CLogical("No")
        theSlideImage.MirrorLeftRight =
CLogical("No")
        theSlideImage.MirrorUpDown =
CLogical("No")
        theSlideImage.UseSourceColorKey =
CLogical("No")
        theSlideImage.SourceColorKey =
CColor("black")

End Sub

'-----
'InitSlideSoundBufferDefaults
'-----
Sub
InitSlideSoundBufferDefaults(theSoundBuffer As
SoundBuffer)

    If theSoundBuffer Is Nothing Then Exit
Sub

        theSoundBuffer.Loop = CLogical("No")
        theSoundBuffer.StopAfter =
CLogical("Yes")
        theSoundBuffer.StartOffset = "0"
        theSoundBuffer.StopOffset = "0"

End Sub

```

```

'-----
' Instance Declarations
'-----
Dim Display As DisplayDevice
Dim Sound As SoundDevice
Dim Keyboard As KeyboardDevice
Dim Mouse As MouseDevice
Dim Coleta As Procedure

Dim PtoTeste As TextDisplay
Dim PtoTesteEchoClients As
EchoClientCollection

Dim PtoTiro As TextDisplay
Dim PtoTiroEchoClients As EchoClientCollection

Dim Procedimento As Procedure

Dim Lista As List

Dim PtoResultado As TextDisplay
Dim PtoResultadoEchoClients As
EchoClientCollection

Dim Tiro As ImageDisplay

Dim Teste As ImageDisplay

Dim Resultado As Slide
Dim Resultado_State As SlideState
Dim Resultado_SlideSoundOut As
SlideSoundOut
Dim Resultado_SoundBuffer As SoundBuffer
Dim Resultado_SoundBufferInfo As
SoundBufferInfo

Dim DurResultado1 As Wait
Dim DurResultado1EchoClients As
EchoClientCollection

Dim PreTiro As TextDisplay
Dim PreTiroEchoClients As EchoClientCollection

Dim PreTeste As TextDisplay

```

```

Dim PreTesteEchoClients As
EchoClientCollection
'-----
'-----

Dim PreResultado As TextDisplay
Dim PreResultadoEchoClients As
EchoClientCollection
'-----
'-----

Dim DurTiro As Wait
Dim DurTiroEchoClients As
EchoClientCollection
' User Script
'-----
'-----

Dim DurTeste As Wait
Dim DurTesteEchoClients As
EchoClientCollection
Const Cursor$ = "Cursor.txt"
Const int2% = 2, dblM# = 0.5
Const dblPid2# = dblM# * Pi
Const booF = False
Const booT = True
Const cHeight as Integer = 7
Const cWidth as Integer = 600
Const cNLin% = 8, cNCol% = 2, cLQua% = 5,
cAQua% = 4, cDHQua% = 6, cDVQua% = 5,
cNPix% = 7, cMinMin! = 0.0715
'Const cFaixa% = 60

Dim Instrução As TextDisplay
Dim InstruçãoEchoClients As
EchoClientCollection
Const cCurr% = 320 'Total de pontos = tentativas
COM CONTINGÊNCIA + pontos iniciais
Dim theMouseResponseData As
MouseResponseData
Dim dblAlfa#, dblSin#, dblSinf#, dblCos#,
dblCosf#, dblX#, dblY#, dblR#, dblRf#, dblDR#,
dblML#, dblML2#, dblMLp#, dblMLp2#,
dblDX#, dblDY#, nCurrentP#, nCurrentF#,
nCurrentR#
Dim lngSQua&, lngALin&, lngLQP&,
lngDHQP&, lngDVQP&, lngAQP&, nCurrent&,
lngTara&
Dim intXLoc%, intYLoc%, intLin%, intULin%,
intCol%, intUCol%, intBar%, intXCir%,
intYCir%, intLCir%, intRCir%, intDXRes%
Dim lngLCan&, lngACan&
Dim sngACurr!, sngBCurr!, sngCCurr!
Dim xLoc&, yLoc&
Dim cnvs As Canvas
Dim cnvs1 As Canvas
Dim pntTiro As point, pntCerto As Point,
pntErrado As Point
Dim pntTiroC As point, pntCertoC As Point,
pntErradoC As Point
Dim pntTiroD As point, pntCertoD As Point,
pntErradoD As Point
Dim intResultado%, intCerto%, intXPress%,
intYPress%
Dim booCon As boolean
Dim booCerto As Boolean
Dim booExi As Boolean
Dim booVai As Boolean
Dim booTara As Boolean
Dim intTemResultado%
Dim lngOnSet1&, lngTVis&

'-----
'-----
' Package Declare Script

```

```

Sub subTCE(pntTirof As Point, pntCertof As Point, pntErradof As Point)
    dblSin# = dblSin# * dblRf#
    dblCos# = dblCos# * dblRf#
    pntTirof.x = xLoc + lngLCan & \ 2 ' int(dblX#) +
    dblCos# ' = int(dblX#)
    pntTirof.y = yLoc - lngLCan & ' int(dblY#) -
    dblSin# ' = int(dblY#)
    dblSin# = intCerto% * dblSin#
    dblCos# = intCerto% * dblCos#
    pntCertof.x = int(dblX# - dblSin#)
    pntCertof.y = int(dblY# - dblCos#)
    pntErradof.x = int(dblX# + dblSin#)
    pntErradof.y = int(dblY# + dblCos#)
End Sub

Function fboFC(pntTCEf As Point, dblMLf2#) As Boolean
    dblDX# = intXPress% - pntTCEf.x
    dblDY# = intYPress% - pntTCEf.y
    fboFC = CBool(dblMLf2# < dblDX# * dblDX# +
    dblDY# * dblDY#)
End Function

Function fboForaC(pntTCE As Point, pntTCed As Point, pntTCEc As Point) As Boolean
    intXPress% = theMouseResponseData.CursorX
    intYPress% = theMouseResponseData.CursorY
    fboForaC = fboFC(pntTCE, dblML2#) ' Or
    Not(fboFC(pntTCed, dblMLp2#) And
    fboFC(pntTCEc, dblMLp2#))
    ' fboForaC = fboFC(pntTCed, dblMLp2#)
End Function

Sub subErro
    sngACurr! = (cCurr% - 0.98 *
    nCurrentF#)/(sngCCurr! - nCurrent&)
    sngBCurr! = cCurr% - cCurr% * sngACurr!
End Sub

Sub subME
    Mouse.ShowCursor booF
    ' subErro
    ' nCurrentR# = sngACurr! * nCurrent& +
    sngBCurr!
    ' Debug.Print nCurrentR# & " = " & sngACurr! &
    " * " & nCurrent& & " + " & sngBCurr!
End Sub

Sub subCan(LQP As Long, AQP as Long)
    ' Debug.Print intXLoc% & "=" & intCol% & "="
    & lngDHQP& & "=" & intYLoc% & "=" &
    intLin% & "=" & lngDVQP& & "=" & intBar% &
    "=" & LQP & "=" & AQP

```

```

    cnvs.Rectangle intXLoc% + intCol% *
    lngDHQP&, intYLoc% + intLin% * lngDVQP&
    + intBar% * cNPix%, LQP, AQP
End Sub

Sub subSoQua
    intLin% = 0
    intCol% = 0
    intBar% = 0
    lngALin& = lngSQua& * cNCol%
    lngLQP& = cLQua% * cNPix%
    lngDHQP& = cDHQua% * cNPix%
    lngDVQP& = cDVQua% * cNPix%
    lngAQP& = cAQua% * cNPix%
    nCurrentP# = nCurrentF#
    Do While nCurrentP# >= lngALin&
        intULin% = intLin%
        intUCol% = cNCol - 1
        For intCol% = 0 To intUCol%
            subCan lngLQP&, lngAQP&
        Next
        intCol% = 0
        intLin% = intLin% + 1
        ' Debug.Print "L" & intLin% & "$" & nCurrentP#
        nCurrentP# = nCurrentP# - lngALin&
    Loop
    Do While nCurrentP# >= lngSQua&
        intULin% = intLin%
        intUCol% = intCol%
        ' Debug.Print intULin% & intUCol% & "$" &
        nCurrentP#
        subCan lngLQP&, lngAQP&
        intCol% = intCol% + 1
        nCurrentP# = nCurrentP# - lngSQua&
    Loop
End Sub

Sub subQua
    subSoQua
    Do While nCurrentP# >= cLQua%
        subCan lngLQP&, cNPix%
        intBar% = intBar% + 1
        nCurrentP# = nCurrentP# - cLQua%
    Loop
    If nCurrentP# > cMinMin! Then
        subCan Int(nCurrentP# * cNPix% + dblM#),
        cNPix%
    End If
End Sub

Sub subPisca
    intBar% = 0
    intLin% = intULin%
    intCol% = intUCol%
    ' Debug.Print intULin% & intUCol%
    If booTara Then

```

```

cnvs.PenColor = CColor("gray")
cnvs.FillColor = CColor("gray")
subSoQua
End If
End Sub

'-----
'-----
' Package Global Script
'-----

'-----
'-----
' Package Routines
'-----

'-----
'-----
' Implementation
'-----

Sub Coleta_Run(c as Context)

    '-----
    ' InLine - Preliminar BEGIN
    '-----

    Dim intMC%
    dblR# = dblM# * 133 '85.5
    dblDR# = dblM# * 66.5 '57 '28.5
    dblML# = dblM# * 19 '47.5 '85.5 '57
    dblML2# = dblML# * dblML#
    dblMLp# = dblM# * 19 '19
    dblMLp2# = dblMLp# * dblMLp#
    If FileExists(Cursor$) Then
        Open Cursor$ For Input As #1
        Input #1, intMC%
        Close
        Mouse.Cursor = intMC%
    End If
    lngSqua& = cLQua% * cAQua%
    lngLCan& = cNCol% * cDHQua% * cNPix% - 1
    lngACan& = cNLin% * cDVQua% * cNPix% - 1
    intDXRes% = Display.XRes
    xLoc& = Int((intDXRes% - lngLCan&) * 0.9)
    yLoc& = Int((Display.YRes - lngACan&) * 0.9)
    intXLoc% = Int(xLoc& - dblM# * cNPix% *
(cLQua% - cDHQua%)) + 1
    intYLoc% = Int(yLoc& - dblM# * cNPix% *
(cAQua% - cDVQua%)) + 1

```

```

lngLCan& = lngLCan& + 2
lngACan& = lngACan& + 2
intXCir% = Int(intDXRes% * 0.2)
intYCir% = (2 * yLoc& + lngACan&) \ 2
intLCir% = 60
intRCir% = 50
sngCCurr! = cCurr% + cMinMin!
nCurrent& = 40 'Pontos iniciais = Total de pontos
- tentativas COM CONTINGÊNCIA
nCurrentF# = nCurrent&
nCurrentR# = nCurrent&
lngTara& = (nCurrent& \ lngSqua& + 1) *
lngSqua&
sngACurr! = 1
sngBCurr! = 0
DurResultado4.Duration = 0
DurResultado3.Duration = 0
DurResultado2.Duration = 0
DurResultado1.Duration = 0
'-----
' InLine - Preliminar END
'-----

Instrução.InputMasks.Reset

InstruçãoEchoClients.RemoveAll
Instrução.InputMasks.Add
Mouse.CreateInputMask("1", "",
CLng(Instrução.Duration), CLng("1")),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "")

Instrução.Run
Lista.Run c

Agradecimento.Run

c.Log
End Sub

Sub Procedimento_Run(c as Context)

    '-----
    ' InLine - Leitura BEGIN
    '-----

    Dim rc As Rect
    dblAlfa# = c.GetAttrib("Alfa")
    dblX# = dblM# + dblM# * Display.XRes '128 +
c.GetAttrib("Pos")
    dblY# = dblM# + dblM# * Display.YRes '128 +
50
    intCerto% = c.GetAttrib("Certo")
    dblSinf# = sin(dblAlfa#)

```

```

dblCosf# = cos(dblAlfa#)
dblRf# = dblR#
subTCE pntTiro, pntCerto, pntErrado
dblRf# = dblR# - dblDR#
subTCE pntTiroD, pntCertoD, pntErradoD
dblRf# = dblR# + dblDR#
subTCE pntTiroC, pntCertoC, pntErradoC
booCon = c.GetAttrib("Contingente")
booExi = c.GetAttrib("Exibe")
booVai = c.GetAttrib("Continua")
PreTeste.Tag = 0
lngOnSeT1& = 0
lngTVis& = 0
Goto Ordem
.....
' InLine - Leitura END
.....

.....
' Label - CaosTiro BEGIN
.....

CaosTiro:
  If Err.Number = ebInputAccepted Then
    Err.Clear
    Resume CaosTiroResume
  ElseIf Err.Number <> 0 Then
    'NOTE: If you receive a runtime
error here, it
    ' is because a runtime error other
than ebInputAccepted
    ' was thrown (ebInputAccepted
for catching input masks that jump).
    'You are encouraged to either
handle the error so that
    ' it is not thrown in the future or
will have to set up
    ' your own error handler, which
will also need to take
    ' into account for any input
masks that jump.
    ,
    'Raise the error so the default
error handler will show the message
    Err.Raise Err.Number
  End If

```

```

CaosTiroResume:
.....
' Label - CaosTiro END
.....

```

```

PtoTiro.InputMasks.Reset

PtoTiroEchoClients.RemoveAll
PtoTiro.InputMasks.Add
Mouse.CreateInputMask("{-ANY}", "",
CLng(PtoTiro.Duration), CLng("1"),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "")

PtoTiro.Run

.....
' Label - Ordem BEGIN
.....

Ordem:
  If Err.Number = ebInputAccepted Then
    Err.Clear
    Resume OrdemResume
  ElseIf Err.Number <> 0 Then
    'NOTE: If you receive a runtime
error here, it
    ' is because a runtime error other
than ebInputAccepted
    ' was thrown (ebInputAccepted
for catching input masks that jump).
    'You are encouraged to either
handle the error so that
    ' it is not thrown in the future or
will have to set up
    ' your own error handler, which
will also need to take
    ' into account for any input
masks that jump.
    ,
    'Raise the error so the default
error handler will show the message
    Err.Raise Err.Number
  End If

```

```

OrdemResume:
.....
' Label - Ordem END
.....

```

```

On Error GoTo CaosTiro

PreTiro.InputMasks.Reset

PreTiroEchoClients.RemoveAll
PreTiro.InputMasks.Add
Mouse.CreateInputMask("1", "",
CLng(PreTiro.Duration), CLng("1"),

```

```
ebEndResponseActionJump, CLogical("Yes"), "",
"", "")
```

```
PreTiro.Run
```

```
.....
' Label - Ordem1 BEGIN
.....
```

```
Ordem1:
```

```
If Err.Number = ebInputAccepted Then
    Err.Clear
    Resume Ordem1Resume
ElseIf Err.Number <> 0 Then
    'NOTE: If you receive a runtime
```

error here, it

' is because a runtime error other than ebInputAccepted

' was thrown (ebInputAccepted for catching input masks that jump).

' You are encouraged to either handle the error so that

' it is not thrown in the future or will have to set up

' your own error handler, which will also need to take

' into account for any input masks that jump.

' Raise the error so the default error handler will show the message

```
Err.Raise Err.Number
End If
```

```
Ordem1Resume:
```

```
.....
' Label - Ordem1 END
.....
```

```
Tiro.Run
c.SetAttrib "Tiro.OnsetTime",
```

```
Tiro.OnsetTime
```

```
c.SetAttrib "Tiro.Duration",
```

```
Tiro.Duration
```

```
.....
' InLine - Barra BEGIN
.....
```

```
Mouse.ShowCursor booT
```

```
If booCon then
```

```
Resultado.Duration = 0
```

```
' Dim rc As Rect
```

```
Set cnvs = Display.CreateCanvas
```

```
cnvs.PenColor = CColor("gray")
```

```
cnvs.FillColor = CColor("gray")
```

```
cnvs.Rectangle xLoc&, yLoc&, lngLCan&,
```

```
lngACan&
```

```
nCurrentF# = cCurr%
```

```
cnvs.PenColor = CColor("black")
```

```
cnvs.FillColor = CColor("black")
```

```
subQua
```

```
nCurrentF# = nCurrentR#
```

```
' Debug.Print sngACurr! & " * " & nCurrent& & "
```

```
+ " & sngBCurr! & "=" & nCurrentF#
```

```
cnvs.PenColor = CColor("white")
```

```
cnvs.FillColor = CColor("white")
```

```
subQua
```

```
subPisca
```

```
'Ready the offscreen canvas for copy
```

```
rc.Left = xLoc&
```

```
rc.Top = yLoc&
```

```
rc.Right = xLoc& + lngLCan&
```

```
rc.Bottom = yLoc& + lngACan&
```

```
'Copy the offscreen canvas to the display
```

```
Display.Canvas.Copy cnvs, rc, rc
```

```
'Release references
```

```
Set cnvs = Nothing
```

'This example draws two arcs in different colors to form a circle.

```
Set cnvs = Display.Canvas
```

```
cnvs.PenColor = CColor("white")
```

```
cnvs.FillColor = CColor("white")
```

```
cnvs.Circle xLoc + lngLCan& \ 2, yLoc -
```

```
lngLCan&, 10
```

```
' cnvs.Rectangle intXCir% - intLCir%, intYCir% -
```

```
intLCir%, 2 * intLCir%, 2 * intLCir%
```

```
'Set the width and color of the pen
```

```
' cnvs.PenWidth = 3
```

```
' cnvs.PenColor = CColor ("black")
```

```
' cnvs.Arc intXCir%, intYCir%, IntRCir%, 0, 360
```

```
' cnvs.FillColor = CColor("white")
```

```
' Set cnvs = Nothing
```

```
Else
```

```
.....
' InLine - Barra END
.....
```

```
DurResultado4.InputMasks.Reset
```

```
DurResultado4EchoClients.RemoveAll
```

```
DurResultado4.InputMasks.Add
```

```
Mouse.CreateInputMask("1", "",
```

```
CLng(DurResultado4.Duration), CLng("1"),
```

```
ebEndResponseActionTerminate,
```

```
CLogical("Yes"), "", "", "")
```



```

DurResultado4.Run
.....
' InLine - DepResultado4 BEGIN
.....
If DurResultado4.RT > 0 Then
DurResultado4.Duration =
DurResultado4.Duration - DurResultado4.RT
subME
Goto CaosTiro
Else
DurResultado4.Duration = 0
Set cnvs = Display.Canvas
cnvs.PenColor = CColor("white")
cnvs.FillColor = CColor("white")
subSoQua
'Set the width and color of the pen
' cnvs.PenWidth = 5
' cnvs.PenColor = CColor("white")
' cnvs.Arc intXCir%, intYCir%, intRCir%, 0, 90
Set cnvs = Nothing
End If
.....
' InLine - DepResultado4 END
.....

DurResultado3.InputMasks.Reset

DurResultado3EchoClients.RemoveAll
DurResultado3.InputMasks.Add
Mouse.CreateInputMask("1", "",
CLng(DurResultado3.Duration), CLng("1"),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "")

DurResultado3.Run
.....
' InLine - DepResultado3 BEGIN
.....
If DurResultado3.RT > 0 Then
DurResultado3.Duration =
DurResultado3.Duration - DurResultado3.RT
subME
Goto CaosTiro
Else
DurResultado3.Duration = 0
Set cnvs = Display.Canvas
subPisca
'Set the width and color of the pen
' cnvs.PenWidth = 5
' cnvs.PenColor = CColor("white")
' cnvs.Arc intXCir%, intYCir%, intRCir%, 270,
90
Set cnvs = Nothing
End If
.....
' InLine - DepResultado3 END
.....

DurResultado2.InputMasks.Reset

DurResultado2EchoClients.RemoveAll
DurResultado2.InputMasks.Add
Mouse.CreateInputMask("1", "",
CLng(DurResultado2.Duration), CLng("1"),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "")

DurResultado2.Run
.....
' InLine - DepResultado2 BEGIN
.....
If DurResultado2.RT > 0 Then
DurResultado2.Duration =
DurResultado2.Duration - DurResultado2.RT
subME
Goto CaosTiro
Else
DurResultado2.Duration = 0
Set cnvs = Display.Canvas
cnvs.PenColor = CColor("white")
cnvs.FillColor = CColor("white")
subSoQua
'Set the width and color of the pen
' cnvs.PenWidth = 5
' cnvs.PenColor = CColor("white")
' cnvs.Arc intXCir%, intYCir%, intRCir%, 180,
90
Set cnvs = Nothing
End If
.....
' InLine - DepResultado2 END
.....

DurResultado1.InputMasks.Reset

DurResultado1EchoClients.RemoveAll
DurResultado1.InputMasks.Add
Mouse.CreateInputMask("1", "",
CLng(DurResultado1.Duration), CLng("1"),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "")

```

```

DurResultado1.Run
.....
' InLine - DepResultado1 BEGIN
.....
If DurResultado1.RT > 0 Then
DurResultado1.Duration =
DurResultado1.Duration - DurResultado1.RT
subME
Goto CaosTiro
'ElseIf booVai Then
' Goto Continua
End If
.....
' InLine - DepResultado1 END
.....

' Label - Ordem2 BEGIN
.....

Ordem2:
If Err.Number = ebInputAccepted Then
Err.Clear
Resume Ordem2Resume
ElseIf Err.Number <> 0 Then
'NOTE: If you receive a runtime
error here, it
' is because a runtime error other
than ebInputAccepted
' was thrown (ebInputAccepted
for catching input masks that jump).
'You are encouraged to either
handle the error so that
' it is not thrown in the future or
will have to set up
' your own error handler, which
will also need to take
' into account for any input
masks that jump.
'
'Raise the error so the default
error handler will show the message
Err.Raise Err.Number
End If

Ordem2Resume:
.....
' Label - Ordem2 END
.....

' InLine - MouTiro BEGIN
.....
Mouse.ShowCursor booT

Resultado.Duration = 0
' Dim rc As Rect
' Set cnvs = Display.CreateCanvas
' cnvs.PenColor = CColor("gray")
' cnvs.FillColor = CColor("gray")
' cnvs.Rectangle xLoc&, yLoc&, lngLCan&,
lngACan&
' nCurrentF# = cCurr%
' cnvs.PenColor = CColor("black")
' cnvs.FillColor = CColor("black")
' subQua
' Debug.Print nCurrentR#
' nCurrentF# = nCurrentR#
' cnvs.PenColor = CColor("white")
' cnvs.FillColor = CColor("white")
' subQua
' subPisca
'Ready the offscreen canvas for copy
' rc.Left = xLoc&
' rc.Top = yLoc&
' rc.Right = xLoc& + lngLCan&
' rc.Bottom = yLoc& + lngACan&
'Copy the offscreen canvas to the display
' Display.Canvas.Copy cnvs, rc, rc
'Release references
' Set cnvs = Nothing
.....
' InLine - MouTiro END
.....

DurTiro.InputMasks.Reset

DurTiroEchoClients.RemoveAll
DurTiro.InputMasks.Add
Mouse.CreateInputMask("1", "",
CLng(DurTiro.Duration), CLng("1"),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "")

DurTiro.Run
c.SetAttrib "DurTiro.RT", DurTiro.RT
.....
' InLine - DepTiro BEGIN
.....
Mouse.ShowCursor booF

```

```

Set theMouseResponseData =
CMouseResponseData(DurTiro.InputMasks.Resp
onses(1))
If fboForaC(pntTiro, pntTiroD, pntTiroC) Then
Goto CaosTiro

```

```

' InLine - DepTiro END

```

```

' Label - CaosTeste BEGIN

```

CaosTeste:

```

If Err.Number = ebInputAccepted Then
Err.Clear
Resume CaosTesteResume
ElseIf Err.Number <> 0 Then
'NOTE: If you receive a runtime
error here, it
' is because a runtime error other
than ebInputAccepted
' was thrown (ebInputAccepted
for catching input masks that jump).
' You are encouraged to either
handle the error so that
' it is not thrown in the future or
will have to set up
' your own error handler, which
will also need to take
' into account for any input
masks that jump.
'
' Raise the error so the default
error handler will show the message
Err.Raise Err.Number
End If

```

CaosTesteResume:

```

' Label - CaosTeste END

```

```

PtoTeste.InputMasks.Reset

```

```

PtoTesteEchoClients.RemoveAll
PtoTeste.InputMasks.Add
Mouse.CreateInputMask("{-ANY}", "",
CLng(PtoTeste.Duration), CLng("1"),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "")

```

```

PtoTeste.Run
c.SetAttrib "PtoTeste.FinishTime",
PtoTeste.FinishTime
c.SetAttrib "PtoTeste.RTTime",
PtoTeste.RTTime
c.SetAttrib "PtoTeste.RT", PtoTeste.RT

```

```

On Error GoTo CaosTeste

```

```

PreTeste.InputMasks.Reset

```

```

PreTesteEchoClients.RemoveAll
PreTeste.InputMasks.Add
Mouse.CreateInputMask("1", "",
CLng(PreTeste.Duration), CLng("1"),
ebEndResponseActionJump, CLogical("Yes"), "",
"", "")

```

```

PreTeste.Run
c.SetAttrib "PreTeste.OnsetTime",
PreTeste.OnsetTime
c.SetAttrib "PreTeste.Duration",
PreTeste.Duration
c.SetAttrib "PreTeste.StartTime",
PreTeste.StartTime
c.SetAttrib "PreTeste.FinishTime",
PreTeste.FinishTime
c.SetAttrib "PreTeste.Tag", PreTeste.Tag

```

```

Teste.Filename = c.GetAttrib("FTeste")
Teste.Load
Teste.Run
c.SetAttrib "Teste.OnsetTime",
Teste.OnsetTime
c.SetAttrib "Teste.Duration",
Teste.Duration
c.SetAttrib "Teste.StartTime",
Teste.StartTime
c.SetAttrib "Teste.FinishTime",
Teste.FinishTime
c.SetAttrib "Teste.Tag", Teste.Tag

```

```

' InLine - MouTeste BEGIN

```

```

Mouse.ShowCursor booT
If booExi Then
' Set cnvs1 = Display.Canvas
' cnvs1.Oval 50, 150, 38, 38
End If

```

```

' InLine - MouTeste END
.....

DurTeste.InputMasks.Reset

DurTesteEchoClients.RemoveAll
DurTeste.InputMasks.Add
Mouse.CreateInputMask("1", "",
CLng(DurTeste.Duration), CLng("1"),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "")

DurTeste.Run
c.SetAttrib "DurTeste.StartTime",
DurTeste.StartTime
c.SetAttrib "DurTeste.RTTime",
DurTeste.RTTime
c.SetAttrib "DurTeste.ACC",
DurTeste.ACC
c.SetAttrib "DurTeste.RT", DurTeste.RT
c.SetAttrib "DurTeste.Tag",
DurTeste.Tag

.....

' InLine - DepTeste BEGIN
.....

Mouse.ShowCursor booF
lngOnSeT1& = lngOnSeT1& + (1 -
Sgn(PreTeste.Tag)) * Teste.OnsetTime
lngTVis& = lngTVis& + DurTeste.RTTime -
Teste.OnsetTime
Set theMouseResponseData =
CMouseResponseData(DurTeste.InputMasks.Res
ponses(1))
booCerto = fboForaC(pntErrado, pntErradoD,
pntErradoC)
If fboForaC(pntCerto, pntCertoD, pntCertoC) And
booCerto then
PreTeste.Tag = PreTeste.Tag + 1
Goto CaosTeste
Else
Teste.Tag = Teste.OnsetTime - lngOnSeT1&
DurTeste.Tag = lngTVis
'intResultado% = DurResultado.OnsetTime -
DurResultado.OffsetTime
If booCon Then
nCurrent& = nCurrent& + 1
If booCerto Then
DurTeste.ACC = 1
Resultado.ActiveState = "Certo"
intTemResultado% = 0
Else
DurTeste.ACC = 0

```

```

Resultado.ActiveState = "Errado"
intTemResultado% = 0
subErro
'sngACurr! = (cCurr% - 0.97 *
nCurrentF#)/(sngCCurr! - nCurrent&)
'sngBCurr! = cCurr% - cCurr% * sngACurr!
End If
Else
If booCerto Then
DurTeste.ACC = 1
Else
DurTeste.ACC = 0
End If
Resultado.ActiveState = "Indefinido"
intTemResultado% = 0
End If
nCurrentR# = sngACurr! * nCurrent& +
sngBCurr!
booTara = (nCurrentR# >= lngTara&)
If booTara Then
Resultado.Duration = 1300
lngTara& = lngTara& + lngSQua&
intTemResultado% = 350
End If
DurResultado4.Duration = intTemResultado%
DurResultado3.Duration = intTemResultado%
DurResultado2.Duration = intTemResultado%
DurResultado1.Duration = intTemResultado%
End If
Goto Aqui1
.....

' InLine - DepTeste END
.....

.....

' Label - CaosTampa BEGIN
.....

CaosTampa:
If Err.Number = ebInputAccepted Then
Err.Clear
Resume CaosTampaResume
ElseIf Err.Number <> 0 Then
'NOTE: If you receive a runtime
error here, it
' is because a runtime error other
than ebInputAccepted
' was thrown (ebInputAccepted
for catching input masks that jump).
'You are encouraged to either
handle the error so that
' it is not thrown in the future or
will have to set up
' your own error handler, which
will also need to take

```

```

' into account for any input
masks that jump.
'
'Raise the error so the default
error handler will show the message
Err.Raise Err.Number
End If

CaosTampaResume:
'
' Label - CaosTampa END
'

PtoTampa.InputMasks.Reset

PtoTampaEchoClients.RemoveAll
PtoTampa.InputMasks.Add
Mouse.CreateInputMask("{-ANY}", "",
CLng(PtoTampa.Duration), CLng("1"),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "")

PtoTampa.Run

On Error GoTo CaosTampa

PreTampa.InputMasks.Reset

PreTampaEchoClients.RemoveAll
PreTampa.InputMasks.Add
Mouse.CreateInputMask("1", "",
CLng(PreTampa.Duration), CLng("1"),
ebEndResponseActionJump, CLogical("Yes"), "",
"", "")

PreTampa.Run

Tampa.Filename =
c.GetAttrib("FTampa")
Tampa.Load
Tampa.Run
'
' InLine - MouTampa BEGIN
'
Mouse.ShowCursor booT
'Set cnvs1 = Display.Canvas
'cnvs1.Oval 50, 150, 38, 38

'cnvs1.Rectangle 0, 0, intDXRes%, cFaixa

```

```

' InLine - MouTampa END
'

DurTampa.InputMasks.Reset

DurTampaEchoClients.RemoveAll
DurTampa.InputMasks.Add
Mouse.CreateInputMask("1", "",
CLng(DurTampa.Duration), CLng("1"),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "")

DurTampa.Run

' InLine - DepTampa BEGIN
'
Mouse.ShowCursor booF
'Set theMouseResponseData =
CMouseResponseData(DurTampa.InputMasks.Re
sponses(1))
'If theMouseResponseData.CursorY > 0 Then
Goto CaosTampa
If DurTampa.RT > 0 Then Goto CaosTampa Else
Goto Aqui
' InLine - DepTampa END
'

' Label - CaosResultado BEGIN
'

CaosResultado:
If Err.Number = ebInputAccepted Then
Err.Clear
Resume CaosResultadoResume
ElseIf Err.Number <> 0 Then
'NOTE: If you receive a runtime
error here, it
' is because a runtime error other
than ebInputAccepted
' was thrown (ebInputAccepted
for catching input masks that jump).
'You are encouraged to either
handle the error so that
' it is not thrown in the future or
will have to set up
' your own error handler, which
will also need to take
' into account for any input
masks that jump.

```

```

'
'Raise the error so the default
error handler will show the message
Err.Raise Err.Number
End If

CaosResultadoResume:

'
' Label - CaosResultado END
'

PtoResultado.InputMasks.Reset

PtoResultadoEchoClients.RemoveAll
PtoResultado.InputMasks.Add
Mouse.CreateInputMask("{-ANY}", "",
CLng(PtoResultado.Duration), CLng("1"),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "")

PtoResultado.Run

'
' Label - Aqui BEGIN
'

Aqui:
If Err.Number = ebInputAccepted Then
Err.Clear
Resume AquiResume
ElseIf Err.Number <> 0 Then
'NOTE: If you receive a runtime
error here, it
' is because a runtime error other
than ebInputAccepted
' was thrown (ebInputAccepted
for catching input masks that jump).
'You are encouraged to either
handle the error so that
' it is not thrown in the future or
will have to set up
' your own error handler, which
will also need to take
' into account for any input
masks that jump.
'
'Raise the error so the default
error handler will show the message
Err.Raise Err.Number
End If

AquiResume:

```

```

'
' Label - Aqui END
'

On Error GoTo CaosResultado

PreResultado.InputMasks.Reset

PreResultadoEchoClients.RemoveAll
PreResultado.InputMasks.Add
Mouse.CreateInputMask("1", "",
CLng(PreResultado.Duration), CLng("1"),
ebEndResponseActionJump, CLogical("Yes"), "",
"", "")

PreResultado.Run

' InLine - AntResultado BEGIN
'
'intResultado% = intResultado% +
DurResultado.OffsetTime -
DurResultado.OnsetTime
'DurResultado.Duration = intTemResultado% -
intResultado%
' InLine - AntResultado END

'
' Label - Aqui1 BEGIN
'

Aqui1:
If Err.Number = ebInputAccepted Then
Err.Clear
Resume Aqui1Resume
ElseIf Err.Number <> 0 Then
'NOTE: If you receive a runtime
error here, it
' is because a runtime error other
than ebInputAccepted
' was thrown (ebInputAccepted
for catching input masks that jump).
'You are encouraged to either
handle the error so that
' it is not thrown in the future or
will have to set up
' your own error handler, which
will also need to take
' into account for any input
masks that jump.

```

```

'
'Raise the error so the default
error handler will show the message
Err.Raise Err.Number
End If

Aqui1 Resume:

.....
'      Label - Aqui1 END
.....

Select Case Resultado.ActiveState
    Case "Certo"

        Set
Resultado_SlideSoundOut =
CSlideSoundOut(Resultado.States.Item("Certo").
Objects(1))
        Set
Resultado_SoundBuffer =
Resultado_SlideSoundOut.Buffers(1)
        Set
Resultado_SoundBuffer = Nothing
        Set
Resultado_SlideSoundOut = Nothing
        Case "Errado"

            Case "Indefinido"

End Select

Resultado.Run

.....
' InLine - Para BEGIN
.....

If booVai Then Goto Continua
.....
' InLine - Para END
.....

.....
'      Label - Parada BEGIN
.....

Parada:
If Err.Number = ebInputAccepted Then
    Err.Clear
    Resume ParadaResume
ElseIf Err.Number <> 0 Then

```

```

'NOTE: If you receive a runtime
error here, it
' is because a runtime error other
than ebInputAccepted
' was thrown (ebInputAccepted
for catching input masks that jump).
'You are encouraged to either
handle the error so that
' it is not thrown in the future or
will have to set up
' your own error handler, which
will also need to take
' into account for any input
masks that jump.
'
'Raise the error so the default
error handler will show the message
Err.Raise Err.Number
End If

ParadaResume:

.....
'      Label - Parada END
.....

Aguarde.InputMasks.Reset

AguardeEchoClients.RemoveAll
Aguarde.InputMasks.Add
Keyboard.CreateInputMask("{ANY}", "",
CLng(Aguarde.Duration), CLng("1"),
ebEndResponseActionTerminate,
CLogical("Yes"), "", "", "ResponseMode:All
ProcessBackspace:Yes")

Aguarde.Run

.....
'      Label - Continua BEGIN
.....

Continua:
If Err.Number = ebInputAccepted Then
    Err.Clear
    Resume ContinuaResume
ElseIf Err.Number <> 0 Then
'NOTE: If you receive a runtime
error here, it
' is because a runtime error other
than ebInputAccepted
' was thrown (ebInputAccepted
for catching input masks that jump).

```

```

        'You are encouraged to either
handle the error so that
        ' it is not thrown in the future or
will have to set up
        ' your own error handler, which
will also need to take
        ' into account for any input
masks that jump.
        '
        'Raise the error so the default
error handler will show the message
        Err.Raise Err.Number
    End If

```

ContinuaResume:

```

.....
'      Label - Continua END
.....

.....
'      Label - Natal BEGIN
.....

```

Natal:

```

    If Err.Number = ebInputAccepted Then
        Err.Clear
        Resume NatalResume
    ElseIf Err.Number <> 0 Then
        'NOTE: If you receive a runtime
error here, it
        ' is because a runtime error other
than ebInputAccepted
        ' was thrown (ebInputAccepted
for catching input masks that jump).
        'You are encouraged to either
handle the error so that
        ' it is not thrown in the future or
will have to set up
        ' your own error handler, which
will also need to take
        ' into account for any input
masks that jump.
        '
        'Raise the error so the default
error handler will show the message
        Err.Raise Err.Number
    End If

```

NatalResume:

```

.....
'      Label - Natal END
.....

```

```

        c.SetAttrib "Tiro.OnsetTime",
Tiro.OnsetTime
        c.SetAttrib "Tiro.Duration",
Tiro.Duration

        c.SetAttrib "DurTiro.RT", DurTiro.RT

        c.SetAttrib "PtoTeste.FinishTime",
PtoTeste.FinishTime
        c.SetAttrib "PtoTeste.RTTime",
PtoTeste.RTTime
        c.SetAttrib "PtoTeste.RT", PtoTeste.RT

        c.SetAttrib "PreTeste.OnsetTime",
PreTeste.OnsetTime
        c.SetAttrib "PreTeste.Duration",
PreTeste.Duration
        c.SetAttrib "PreTeste.StartTime",
PreTeste.StartTime
        c.SetAttrib "PreTeste.FinishTime",
PreTeste.FinishTime
        c.SetAttrib "PreTeste.Tag", PreTeste.Tag

        c.SetAttrib "Teste.OnsetTime",
Teste.OnsetTime
        c.SetAttrib "Teste.Duration",
Teste.Duration
        c.SetAttrib "Teste.StartTime",
Teste.StartTime
        c.SetAttrib "Teste.FinishTime",
Teste.FinishTime
        c.SetAttrib "Teste.Tag", Teste.Tag

        c.SetAttrib "DurTeste.StartTime",
DurTeste.StartTime
        c.SetAttrib "DurTeste.RTTime",
DurTeste.RTTime
        c.SetAttrib "DurTeste.ACC",
DurTeste.ACC
        c.SetAttrib "DurTeste.RT", DurTeste.RT
        c.SetAttrib "DurTeste.Tag",
DurTeste.Tag

        c.Log
    End Sub

```

```

-----
-----
' InitDevices
,

```



```

'-----
-----
Sub InitDevices(c As Context)

    SetOSThreadPriority 3

    Set Display = New DisplayDevice
    Display.Name = "Display"

    Dim DisplayDisplayDeviceInfo As
DisplayDeviceInfo
    DisplayDisplayDeviceInfo.XRes = 640
    DisplayDisplayDeviceInfo.YRes = 480
    DisplayDisplayDeviceInfo.ColorDepth =
16
    DisplayDisplayDeviceInfo.RefreshRate =
0
    DisplayDisplayDeviceInfo.NumPages =
0

    Display.Open DisplayDisplayDeviceInfo
c.SetAttrib "Display.RefreshRate",
Format$(Display.CalculatedRefreshRate, "0.000")

    Set Sound = New SoundDevice
    Sound.Name = "Sound"

    Dim SoundSoundOutDeviceInfo As
SoundDeviceInfo
    SoundSoundOutDeviceInfo.Channels = 2
    SoundSoundOutDeviceInfo.SamplesPerS
econd = 22050
    SoundSoundOutDeviceInfo.BitsPerSamp
le = 16
    Sound.Open SoundSoundOutDeviceInfo

    Set Keyboard = New KeyboardDevice
    Keyboard.Name = "Keyboard"

    Dim KeyboardKeyboardDeviceInfo as
KeyboardDeviceInfo
    KeyboardKeyboardDeviceInfo.Collectio
nMode = ebPressesOnly
    KeyboardKeyboardDeviceInfo.CapsLock
= ebCapsLockOff
    KeyboardKeyboardDeviceInfo.NumLock
= ebNumLockOn
    Keyboard.Open
KeyboardKeyboardDeviceInfo

    Set Mouse = New MouseDevice
    Mouse.Name = "Mouse"

```

```

    Dim MouseMouseDeviceInfo as
MouseDeviceInfo
    MouseMouseDeviceInfo.OpenMode =
ebMouseOpenModeDirect
    MouseMouseDeviceInfo.CollectionMode
= ebPressesAndReleases
    MouseMouseDeviceInfo.ShowCursor =
True
    Mouse.Open MouseMouseDeviceInfo

    SetOSThreadPriority 0

End Sub

'-----
-----
' InitObjects
'-----
-----
Sub InitObjects(c As Context)

    Set Coleta = New Procedure
    Coleta.Name = "Coleta"
    Coleta.Tag = ""
    Coleta.Subroutine = "Coleta_Run"

    Set PtoTeste = New TextDisplay
    PtoTeste.Name = "PtoTeste"
    PtoTeste.Tag = ""

    Set PtoTesteEchoClients = New
EchoClientCollection

    InitTextDisplayDefaults PtoTeste

    PtoTeste.Text = ""
    PtoTeste.BackColor = CColor("black")
    PtoTeste.ForeColor = CColor("white")
    PtoTeste.Duration = CLng("-1")
    PtoTeste.TimingMode =
ebTimingModeEvent
    PtoTeste.PreRelease = Val("0")

    PtoTeste.OnsetSync = 1
    PtoTeste.OffsetSync = 0

    Set PtoTiro = New TextDisplay
    PtoTiro.Name = "PtoTiro"
    PtoTiro.Tag = ""

    Set PtoTiroEchoClients = New
EchoClientCollection

```

```

InitTextDisplayDefaults PtoTiro

PtoTiro.Text = ""
PtoTiro.BackColor = CColor("black")
PtoTiro.ForeColor = CColor("white")
PtoTiro.Duration = CLng("-1")
PtoTiro.TimingMode =
ebTimingModeEvent
PtoTiro.PreRelease = Val("0")

PtoTiro.OnsetSync = 1
PtoTiro.OffsetSync = 0

Set Procedimento = New Procedure
Procedimento.Name = "Procedimento"
Procedimento.Tag = ""
Procedimento.Subroutine =
"Procedimento_Run"

Set Lista = New List
Lista.Name = "Lista"
Lista.Tag = ""

'Initialization for Lista

Set Lista.Order = New SequentialOrder
Set Lista.Deletion = NoDeletion
Lista.ResetEveryRun = False

' Create the column headings
Lista.AddAttrib "FTiro"
Lista.AddAttrib "FTeste"
Lista.AddAttrib "FTampa"
Lista.AddAttrib "Certo"
Lista.AddAttrib "Contingente"
Lista.AddAttrib "Alfa"
Lista.AddAttrib "Pos"
Lista.AddAttrib "Exibe"
Lista.AddAttrib "Continua"
Lista.FileName = ""
Lista.LoadMethod =
ebLoadMethodEmbedded

Lista.Load

Set Lista.TerminateCondition =
Cycles(1)
Set Lista.ResetCondition = Samples(295)
Lista.Reset

Set PtoResultado = New TextDisplay
PtoResultado.Name = "PtoResultado"
PtoResultado.Tag = ""

Set PtoResultadoEchoClients = New
EchoClientCollection

InitTextDisplayDefaults PtoResultado

PtoResultado.Text = ""
PtoResultado.BackColor =
CColor("black")
PtoResultado.ForeColor =
CColor("white")
PtoResultado.Duration = CLng("-1")
PtoResultado.TimingMode =
ebTimingModeEvent
PtoResultado.PreRelease = Val("0")

PtoResultado.OnsetSync = 1
PtoResultado.OffsetSync = 0

Set Tiro = New ImageDisplay
Tiro.Name = "Tiro"
Tiro.Tag = ""
Tiro.FileName = "Vazio.bmp"
Tiro.Load

InitImageDisplayDefaults Tiro

Tiro.Duration = CLng("0")
Tiro.TimingMode =
ebTimingModeEvent
Tiro.PreRelease = Val("0")

Tiro.OnsetSync = 1
Tiro.OffsetSync = 0

Set Teste = New ImageDisplay
Teste.Name = "Teste"
Teste.Tag = ""

InitImageDisplayDefaults Teste

Teste.Duration = CLng("0")
Teste.TimingMode =
ebTimingModeEvent
Teste.PreRelease = Val("0")

Teste.OnsetSync = 1
Teste.OffsetSync = 0

Set Resultado = New Slide
Resultado.Name = "Resultado"
Resultado.Tag = ""
Resultado.Duration = CLng("0")
Resultado.TimingMode =
ebTimingModeEvent
Resultado.PreRelease = Val("0")

```

```

Resultado.OnsetSync = 1
Resultado.OffsetSync = 0

Set Resultado_State = New SlideState
Resultado_State.Name = "Certo"

InitSlideStateDefaults Resultado_State

Resultado_State.BackColor =
CColor("black")

Resultado.States.Add Resultado_State,
"Certo"

Set Resultado_SlideSoundOut = New
SlideSoundOut
Resultado_SlideSoundOut.Name =
"SoundOut1"
Resultado_SoundBufferInfo.MaxLength
= 5000
Resultado_SoundBufferInfo.VolumeCont
rol = CLogical("no")
Resultado_SoundBufferInfo.PanControl
= CLogical("no")
Set Resultado_SoundBuffer =
Sound.CreateBuffer(Resultado_SoundBufferInfo)
Resultado_SlideSoundOut.Buffers.Add
Resultado_SoundBuffer
Resultado.States.Item("Certo").Objects.A
dd Resultado_SlideSoundOut, "SoundOut1"

Resultado_SoundBuffer.FileName =
"tada.wav"
Resultado_SoundBuffer.Load

InitSlideSoundBufferDefaults
Resultado_SoundBuffer

Resultado_SoundBuffer.EndSoundActio
n = 0

Set Resultado_SoundBuffer = Nothing

Set Resultado_State = New SlideState
Resultado_State.Name = "Errado"

InitSlideStateDefaults Resultado_State

Resultado_State.BackColor =
CColor("black")

Resultado.States.Add Resultado_State,
"Errado"

Set Resultado_State = New SlideState

Resultado_State.Name = "Indefinido"

InitSlideStateDefaults Resultado_State

Resultado_State.BackColor =
CColor("black")

Resultado.States.Add Resultado_State,
"Indefinido"

Resultado.ActiveState = "Certo"

Set DurResultado1 = New Wait
DurResultado1.Name = "DurResultado1"
DurResultado1.Tag = ""

Set DurResultado1EchoClients = New
EchoClientCollection
DurResultado1.Duration = CLng("0")
DurResultado1.TimingMode =
ebTimingModeEvent
DurResultado1.PreRelease = Val("0")

DurResultado1.OnsetSync = 1
DurResultado1.OffsetSync = 0

Set PreTiro = New TextDisplay
PreTiro.Name = "PreTiro"
PreTiro.Tag = ""

Set PreTiroEchoClients = New
EchoClientCollection

InitTextDisplayDefaults PreTiro

PreTiro.Text = ""
PreTiro.BackColor = CColor("black")
PreTiro.Duration = CLng("1000")
PreTiro.TimingMode =
ebTimingModeCumulative
PreTiro.PreRelease = Val("800")

PreTiro.OnsetSync = 1
PreTiro.OffsetSync = 0

Set PreTeste = New TextDisplay
PreTeste.Name = "PreTeste"
PreTeste.Tag = ""

Set PreTesteEchoClients = New
EchoClientCollection

InitTextDisplayDefaults PreTeste

PreTeste.Text = ""
PreTeste.BackColor = CColor("black")

```



```

Aguarde.PreRelease = Val("0")

Aguarde.OnsetSync = 1
Aguarde.OffsetSync = 0

Set DurResultado4 = New Wait
DurResultado4.Name = "DurResultado4"
DurResultado4.Tag = ""

Set DurResultado4EchoClients = New
EchoClientCollection
DurResultado4.Duration = CLng("0")
DurResultado4.TimingMode =
ebTimingModeEvent
DurResultado4.PreRelease = Val("0")

DurResultado4.OnsetSync = 1
DurResultado4.OffsetSync = 0

Set DurResultado3 = New Wait
DurResultado3.Name = "DurResultado3"
DurResultado3.Tag = ""

Set DurResultado3EchoClients = New
EchoClientCollection
DurResultado3.Duration = CLng("0")
DurResultado3.TimingMode =
ebTimingModeEvent
DurResultado3.PreRelease = Val("0")

DurResultado3.OnsetSync = 1
DurResultado3.OffsetSync = 0

Set DurResultado2 = New Wait
DurResultado2.Name = "DurResultado2"
DurResultado2.Tag = ""

Set DurResultado2EchoClients = New
EchoClientCollection
DurResultado2.Duration = CLng("0")
DurResultado2.TimingMode =
ebTimingModeEvent
DurResultado2.PreRelease = Val("0")

DurResultado2.OnsetSync = 1
DurResultado2.OffsetSync = 0

Set PtoTampa = New TextDisplay
PtoTampa.Name = "PtoTampa"
PtoTampa.Tag = ""

Set PtoTampaEchoClients = New
EchoClientCollection

InitTextDisplayDefaults PtoTampa

PtoTampa.Text = ""
PtoTampa.BackColor = CColor("black")
PtoTampa.Duration = CLng("-1")
PtoTampa.TimingMode =
ebTimingModeEvent
PtoTampa.PreRelease = Val("0")

PtoTampa.OnsetSync = 1
PtoTampa.OffsetSync = 0

Set PreTampa = New TextDisplay
PreTampa.Name = "PreTampa"
PreTampa.Tag = ""

Set PreTampaEchoClients = New
EchoClientCollection

InitTextDisplayDefaults PreTampa

PreTampa.Text = ""
PreTampa.BackColor = CColor("black")
PreTampa.Duration = CLng("1000")
PreTampa.TimingMode =
ebTimingModeCumulative
PreTampa.PreRelease = Val("100")

PreTampa.OnsetSync = 1
PreTampa.OffsetSync = 0

Set Tampa = New ImageDisplay
Tampa.Name = "Tampa"
Tampa.Tag = ""

InitImageDisplayDefaults Tampa

Tampa.Duration = CLng("0")
Tampa.TimingMode =
ebTimingModeEvent
Tampa.PreRelease = Val("0")

Tampa.OnsetSync = 1
Tampa.OffsetSync = 0

Set DurTampa = New Wait
DurTampa.Name = "DurTampa"
DurTampa.Tag = ""

Set DurTampaEchoClients = New
EchoClientCollection
DurTampa.Duration = CLng("500")
DurTampa.TimingMode =
ebTimingModeEvent
DurTampa.PreRelease = Val("0")

DurTampa.OnsetSync = 1
DurTampa.OffsetSync = 0

```

End Sub	Set Coleta = Nothing
	Set PtoTeste = Nothing
'----- ----- ' InitPackages '----- -----	Set PtoTesteEchoClients = Nothing
Sub InitPackages(c As Context)	Set PtoTiro = Nothing
End Sub	Set PtoTiroEchoClients = Nothing
	Set Procedimento = Nothing
	Set Lista = Nothing
	Set PtoResultado = Nothing
'----- ----- '----- ----- ' UnInitDevices '----- -----	Set PtoResultadoEchoClients = Nothing
Sub UnInitDevices()	Set Tiro = Nothing
Display.Close	Set Teste = Nothing
Set Display = Nothing	Set Resultado = Nothing
Sound.Close	Set DurResultado1 = Nothing
	Set DurResultado1 EchoClients = Nothing
Keyboard.Close	Set PreTiro = Nothing
Set Keyboard = Nothing	Set PreTiroEchoClients = Nothing
	Set PreTeste = Nothing
Mouse.Close	Set PreTesteEchoClients = Nothing
Set Mouse = Nothing	Set PreResultado = Nothing
End Sub	Set PreResultadoEchoClients = Nothing
	Set DurTiro = Nothing
'----- ----- '----- ----- ' UnInitPackages '----- -----	Set DurTiroEchoClients = Nothing
Sub UnInitPackages()	Set DurTeste = Nothing
End Sub	Set DurTesteEchoClients = Nothing
	Set Instrução = Nothing
'----- ----- ' UnInitObjects '----- -----	Set InstruçãoEchoClients = Nothing
Sub UnInitObjects()	Set Agradecimento = Nothing

```

Set Aguarde = Nothing
Set AguardeEchoClients = Nothing
Set DurResultado4 = Nothing
Set DurResultado4EchoClients =
Nothing
Set DurResultado3 = Nothing
Set DurResultado3EchoClients =
Nothing
Set DurResultado2 = Nothing
Set DurResultado2EchoClients =
Nothing
Set PtoTampa = Nothing
Set PtoTampaEchoClients = Nothing
Set PreTampa = Nothing
Set PreTampaEchoClients = Nothing
Set Tampa = Nothing
Set DurTampa = Nothing
Set DurTampaEchoClients = Nothing

End Sub

'-----
' Main
'-----
Sub Main()

    CreateDefaultPort

    ' Create and initialize the default context,
data file,
    ' and provide global access to the context.
    Dim c As Context
    Set c = New Context
    Set c.DataFile = New DataFile
    c.PushNewFrame
    Set ebContext = c

    ' Set the log level names
    c.SetLogLevelName 1, "Session"
    c.SetLogLevelName 2, "Block"
    c.SetLogLevelName 3, "Trial"
    c.SetLogLevelName 4, "SubTrial"
    c.SetLogLevelName 5, "LogLevel5"
    c.SetLogLevelName 6, "LogLevel6"
    c.SetLogLevelName 7, "LogLevel7"
    c.SetLogLevelName 8, "LogLevel8"
    c.SetLogLevelName 9, "LogLevel9"
    c.SetLogLevelName 10, "LogLevel10"

    ' Set standard logging items
    ebContext.SetAttrib "Experiment", "Exp"
    ebContext.SetAttrib "SessionDate",
Date$
    ebContext.SetAttrib "SessionTime",
Time$
    ebContext.SetAttrib "RandomSeed",
PRNG.GetSeed()

    If Basic.OS = ebWin32 Then
        WinActivate "E-Run
Experiment Window"
    End If

    ' Get the StartupInfo
    ' Set default for GroupNumber
    c.SetAttrib "Group", "1"

    ' Set the defaults for all of the StartupInfo
    c.SetAttrib "Subject", "1"
    c.SetAttrib "Session", "1"

    Dim vAnswer As Variant
    StartupInfo_Begin:

    StartupInfoPrompt_Subject:
        vAnswer = AskBox("Please enter the
Subject Number (1-32767, 0=No Data Logging):",
c.GetAttrib("Subject"))
        If Not IsEmpty(vAnswer) then
            If Not IsNumeric(vAnswer) then
                MsgBox "Please enter
an integer value"
                GoTo
StartupInfoPrompt_Subject
            ElseIf CLng(vAnswer) < 0 Then
                MsgBox "The value for
Subject must not be less than 0"
                GoTo
StartupInfoPrompt_Subject
            ElseIf CLng(vAnswer) > 32767
Then

```

```

                MsgBox "The value for
Subject must be not be greater than 32767"
                GoTo
StartupInfoPrompt_Subject
            End If
        Else
            GoTo ExperimentAbort
        End if

        c.SetAttrib "Subject", CStr(vAnswer)

StartupInfoPrompt_Session:
    vAnswer = AskBox("Please enter the
Session Number (1-32767):",
c.GetAttrib("Session"))
    If Not IsEmpty(vAnswer) then
        If Not IsNumeric(vAnswer) then
            MsgBox "Please enter
an integer value"
            GoTo
StartupInfoPrompt_Session
        ElseIf CLng(vAnswer) < 1 Then
            MsgBox "The value for
Session must not be less than 1"
            GoTo
StartupInfoPrompt_Session
        ElseIf CLng(vAnswer) > 32767
Then
            MsgBox "The value for
Session must be not be greater than 32767"
            GoTo
StartupInfoPrompt_Session
        End If
    Else
        GoTo ExperimentAbort
    End if

    c.SetAttrib "Session", CStr(vAnswer)

    ' Display the summary
    Dim strSummary As String
    strSummary = "Subject: " &
c.GetAttrib("Subject") & "\n"
    strSummary = strSummary & "Session:
" & c.GetAttrib("Session") & "\n"
    strSummary = strSummary &
"\nContinue with the above startup info?"

    Dim nSummaryAnswer As Integer
    nSummaryAnswer =
MsgBox(strSummary, ebYesNoCancel +
ebQuestion, "Summary of Startup Info")
    If nSummaryAnswer = ebNo Then
        GoTo StartupInfo_Begin
    ElseIf nSummaryAnswer = ebCancel
Then

```

```

                GoTo ExperimentAbort
            End If

            ' Set the default Data Filename
            c.DataFile.Filename =
CStr(c.GetAttrib("Experiment")) & "-" &
CStr(c.GetAttrib("Subject")) & "-" &
CStr(c.GetAttrib("Session")) & ".txt"

            ' If we are logging data, then prompt to
overwrite the data file if it exists
            If CLng(c.GetAttrib("Subject")) <> 0
Then
                If
FileExists(c.DataFile.Filename) Then
                    If ebYes <>
MsgBox("WARNING: The data file and/or
recovery file already exists:\nFILE: " &
c.DataFile.Filename & "\n\nDo you want to
overwrite?", ebYesNo + ebQuestion) Then
                        GoTo
ExperimentAbort
                    End If
                End If
            End If

            ' Initialize all system devices, packages,
and objects
            InitDevices c
            InitPackages c
            InitObjects c

            ' If we are logging data, then open the
datafile
            If CLng(c.GetAttrib("Subject")) <> 0
Then
                c.DataFile.Open
                c.LogHeader
            End If

            ' Start the running of the Experiment
Coleta.Run c

            ' Clean up the context and close the
datafile
            If CLng(c.GetAttrib("Subject")) <> 0
Then
                c.DataFile.Close
                ' Attempt to convert the recovery
file into a data file
                Dim nConvert As Long
                nConvert =
c.DataFile.Convert(ebProgressSimple)
                If nConvert = 0 Then

```


' Settings in E-Studio are set to not remove E-Recovery file		TRUE	0.7064981	-20	TRUE
Else	1	True			
' The datafile failed to convert!		Procedimento		A002P.bmp	
		B002P.bmp	B002P.bmp	-1	
		TRUE	0.7064981	-20	TRUE
MsgBox "ERROR: The datafile did not convert!\nFILE: " & c.DataFile.FileName & "\n\nIt is recommended that you recover your data with the E-Recovery utility"	1	True			
		Procedimento		A002P.bmp	
		B002P.bmp	B002P.bmp	-1	
		TRUE	0.7064981	-20	TRUE
MsgBox c.DataFile.GetLastErrorMessages() End If	1	True			
		Procedimento		A001P.bmp	
		B001P.bmp	B001P.bmp	1	
		TRUE	0.7064981	-20	TRUE
End If ExperimentFinish:	1	True			
		Procedimento		A001P.bmp	
		B001P.bmp	B001P.bmp	1	
		TRUE	0.7064981	-20	TRUE
UnInitObjects		True			
		Procedimento		A002P.bmp	
		B002P.bmp	B002P.bmp	-1	
		TRUE	0.7064981	-20	TRUE
UnInitPackages UnInitDevices	1	True			
		Procedimento		A002P.bmp	
ExperimentAbort:		B002P.bmp	B002P.bmp	-1	
		TRUE	0.7064981	-20	TRUE
' Clean up the context c.PopFrame Set c = Nothing Set ebContext = Nothing	1	True			
		Procedimento		A001P.bmp	
		B001P.bmp	B001P.bmp	1	
		TRUE	0.7064981	-20	TRUE
DestroyDefaultPort	1	True			
		Procedimento		A001P.bmp	
		B001P.bmp	B001P.bmp	1	
		TRUE	0.7064981	-20	TRUE
End Sub	1	True			
		Procedimento		A001P.bmp	
		B001P.bmp	B001P.bmp	1	
		TRUE	0.7064981	-20	TRUE
[DataSection_Lista(1)]		True			
Weight Nested Procedure FTiro	1	Procedimento		A002P.bmp	
FTeste FTampa Certo Contingente		B002P.bmp	B002P.bmp	-1	
Alfa Pos Exibe Continua		TRUE	0.7064981	-20	TRUE
1		Procedimento		A001P.bmp	
		B001P.bmp	B001P.bmp	1	
		TRUE	0.7064981	-20	TRUE
		True			
		Procedimento		A001P.bmp	
		B001P.bmp	B001P.bmp	1	
		TRUE	0.7064981	-20	TRUE
		True			
		Procedimento		A001P.bmp	
		B001P.bmp	B001P.bmp	1	
		TRUE	0.7064981	-20	TRUE
		True			
		Procedimento		A002P.bmp	
		B002P.bmp	B002P.bmp	-1	
		TRUE	-0.1814761	-20	TRUE
		True			
		Procedimento		A003.bmp	
		B003.bmp	B003.bmp	1	
		TRUE	2.4582167	-20	TRUE
		True			
		Procedimento		A004.bmp	
		B004.bmp	B004.bmp	-1	

	TRUE	-0.0521086	-20	TRUE		TRUE	0.8681677	-20	TRUE
	True					True			
1		Procedimento	A005.bmp		1		Procedimento	A019.bmp	
	B005.bmp	B005.bmp	1			B019.bmp	B019.bmp	1	
	TRUE	-2.1914057	-20	TRUE		TRUE	-1.0174773	-20	TRUE
	True					True			
1		Procedimento	A006.bmp		1		Procedimento	A020.bmp	
	B006.bmp	B006.bmp	1			B020.bmp	B020.bmp	-1	
	TRUE	1.0013471	-20	TRUE		TRUE	2.3728106	-20	TRUE
	True					True			
1		Procedimento	A007.bmp		1		Procedimento	A021.bmp	
	B007.bmp	B007.bmp	-1			B021.bmp	B021.bmp	1	
	TRUE	-0.3469991	-20	TRUE		TRUE	2.6704525	-20	TRUE
	True					True			
1		Procedimento	A008.bmp		1		Procedimento	A022.bmp	
	B008.bmp	B008.bmp	-1			B022.bmp	B022.bmp	-1	
	TRUE	-1.2585296	-20	TRUE		TRUE	2.7760790	-20	TRUE
	True					True			
1		Procedimento	A009.bmp		1		Procedimento	A023.bmp	
	B009.bmp	B009.bmp	-1			B023.bmp	B023.bmp	1	
	TRUE	-0.6376980	-20	TRUE		TRUE	-0.6117926	-20	TRUE
	True					True			
1		Procedimento	A010.bmp		1		Procedimento	A024.bmp	
	B010.bmp	B010.bmp	1			B024.bmp	B024.bmp	-1	
	TRUE	-1.8159985	-20	TRUE		TRUE	1.0840803	-20	TRUE
	True					True			
1		Procedimento	A011.bmp		1		Procedimento	A025.bmp	
	B011.bmp	B011.bmp	1			B025.bmp	B025.bmp	-1	
	TRUE	0.3049970	-20	TRUE		TRUE	1.8263444	-20	TRUE
	True					True			
1		Procedimento	A012.bmp		1		Procedimento	A026.bmp	
	B012.bmp	B012.bmp	1			B026.bmp	B026.bmp	-1	
	TRUE	-0.2595993	-20	TRUE		TRUE	-2.5298020	-20	TRUE
	True					True			
1		Procedimento	A013.bmp		1		Procedimento	A027.bmp	
	B013.bmp	B013.bmp	1			B027.bmp	B027.bmp	-1	
	TRUE	-1.9132719	-20	TRUE		TRUE	2.3875900	-20	TRUE
	True					True			
1		Procedimento	A014.bmp		1		Procedimento	A028.bmp	
	B014.bmp	B014.bmp	-1			B028.bmp	B028.bmp	-1	
	TRUE	-2.8068368	-20	TRUE		TRUE	2.8312311	-20	TRUE
	True					True			
1		Procedimento	A015.bmp		1		Procedimento	A029.bmp	
	B015.bmp	B015.bmp	1			B029.bmp	B029.bmp	1	
	TRUE	1.9396432	-20	TRUE		TRUE	-0.8923430	-20	TRUE
	True					True			
1		Procedimento	A016.bmp		1		Procedimento	A030.bmp	
	B016.bmp	B016.bmp	-1			B030.bmp	B030.bmp	1	
	TRUE	-1.0525628	-20	TRUE		TRUE	-0.5579437	-20	TRUE
	True					True			
1		Procedimento	A017.bmp		1		Procedimento	A031.bmp	
	B017.bmp	B017.bmp	1			B031.bmp	B031.bmp	-1	
	TRUE	-0.0546438	-20	TRUE		TRUE	-0.8626865	-20	TRUE
	True					True			
1		Procedimento	A018.bmp		1		Procedimento	A032.bmp	
	B018.bmp	B018.bmp	-1			B032.bmp	B032.bmp	1	

	TRUE	-0.2829322	-20	TRUE		TRUE	-1.3113207	-20	TRUE
	True					True			
1		Procedimento	A033.bmp		1		Procedimento	A047.bmp	
	B033.bmp	B033.bmp	1			B047.bmp	B047.bmp	1	
	TRUE	-1.0377645	-20	TRUE		TRUE	-1.2262578	-20	TRUE
	True					True			
1		Procedimento	A034.bmp		1		Procedimento	A048.bmp	
	B034.bmp	B034.bmp	1			B048.bmp	B048.bmp	-1	
	TRUE	1.0856593	-20	TRUE		TRUE	1.2567470	-20	TRUE
	True					True			
1		Procedimento	A035.bmp		1		Procedimento	A049.bmp	
	B035.bmp	B035.bmp	1			B049.bmp	B049.bmp	1	
	TRUE	-2.0533405	-20	TRUE		TRUE	0.6906600	-20	TRUE
	True					True			
1		Procedimento	A036.bmp		1		Procedimento	A050.bmp	
	B036.bmp	B036.bmp	-1			B050.bmp	B050.bmp	-1	
	TRUE	1.5006327	-20	TRUE		TRUE	0.9505399	-20	TRUE
	True					True			
1		Procedimento	A037.bmp		1		Procedimento	A051.bmp	
	B037.bmp	B037.bmp	1			B051.bmp	B051.bmp	-1	
	TRUE	-1.7244310	-20	TRUE		TRUE	0.9319233	-20	TRUE
	True					True			
1		Procedimento	A038.bmp		1		Procedimento	A052.bmp	
	B038.bmp	B038.bmp	-1			B052.bmp	B052.bmp	1	
	TRUE	-1.1244002	-20	TRUE		TRUE	-0.2328878	-20	TRUE
	True					True			
1		Procedimento	A039.bmp		1		Procedimento	A053.bmp	
	B039.bmp	B039.bmp	-1			B053.bmp	B053.bmp	-1	
	TRUE	1.2822712	-20	TRUE		TRUE	-1.9312225	-20	TRUE
	True					True			
1		Procedimento	A040.bmp		1		Procedimento	A054.bmp	
	B040.bmp	B040.bmp	1			B054.bmp	B054.bmp	-1	
	TRUE	-0.1921720	-20	TRUE		TRUE	1.2160625	-20	TRUE
	True					True			
1		Procedimento	A041.bmp		1		Procedimento	A055.bmp	
	B041.bmp	B041.bmp	-1			B055.bmp	B055.bmp	1	
	TRUE	0.1757368	-20	TRUE		TRUE	-2.8731516	-20	TRUE
	True					True			
1		Procedimento	A042.bmp		1		Procedimento	A056.bmp	
	B042.bmp	B042.bmp	-1			B056.bmp	B056.bmp	1	
	TRUE	-1.8503708	-20	TRUE		TRUE	2.8650559	-20	TRUE
	True					True			
1		Procedimento	A043.bmp		1		Procedimento	A057.bmp	
	B043.bmp	B043.bmp	1			B057.bmp	B057.bmp	-1	
	TRUE	-1.0933923	-20	TRUE		TRUE	1.0226444	-20	TRUE
	True					True			
1		Procedimento	A044.bmp		1		Procedimento	A058.bmp	
	B044.bmp	B044.bmp	1			B058.bmp	B058.bmp	1	
	TRUE	0.1692197	-20	TRUE		TRUE	-0.7061500	-20	TRUE
	True					False			
1		Procedimento	A045.bmp		1		Procedimento	A059.bmp	
	B045.bmp	B045.bmp	-1			B059.bmp	B059.bmp	-1	
	TRUE	-2.9061660	-20	TRUE		TRUE	-3.0185548	-20	TRUE
	True					True			
1		Procedimento	A046.bmp		1		Procedimento	A060.bmp	
	B046.bmp	B046.bmp	1			B060.bmp	B060.bmp	-1	

	TRUE	-2.6194666	-20	TRUE		TRUE	2.8222735	150	TRUE
	True					True			
1		Procedimento	A061.bmp		1		Procedimento	A075.bmp	
	B061.bmp	B061.bmp	1			C075.bmp	C075.bmp	-1	
	TRUE	1.5814740	-20	TRUE		TRUE	1.3963007	150	TRUE
	True					True			
1		Procedimento	A062.bmp		1		Procedimento	A076.bmp	
	B062.bmp	B062.bmp	-1			C076.bmp	C076.bmp	1	
	TRUE	0.9496165	-20	TRUE		TRUE	-2.4282560	150	TRUE
	True					True			
1		Procedimento	A063.bmp		1		Procedimento	A077.bmp	
	B063.bmp	B063.bmp	1			C077.bmp	C077.bmp	1	
	TRUE	-0.6927701	-20	TRUE		TRUE	0.8490685	150	TRUE
	True					True			
1		Procedimento	A064.bmp		1		Procedimento	A078.bmp	
	B064.bmp	B064.bmp	-1			C078.bmp	C078.bmp	-1	
	TRUE	0.1309338	-20	TRUE		TRUE	-1.2574159	150	TRUE
	True					True			
1		Procedimento	A065.bmp		1		Procedimento	A079.bmp	
	B065.bmp	B065.bmp	-1			C079.bmp	C079.bmp	1	
	TRUE	0.1235645	-20	TRUE		TRUE	1.7693021	150	TRUE
	True					True			
1		Procedimento	A066.bmp		1		Procedimento	A080.bmp	
	B066.bmp	B066.bmp	1			C080.bmp	C080.bmp	-1	
	TRUE	2.5565180	-20	TRUE		TRUE	-2.9238924	150	TRUE
	True					True			
1		Procedimento	A067.bmp		1		Procedimento	A081.bmp	
	C067.bmp	C067.bmp	-1			C081.bmp	C081.bmp	-1	
	TRUE	-1.1003286	150	TRUE		TRUE	1.4993480	150	TRUE
	True					True			
1		Procedimento	A068.bmp		1		Procedimento	A082.bmp	
	C068.bmp	C068.bmp	-1			C082.bmp	C082.bmp	1	
	TRUE	2.2557455	150	TRUE		TRUE	1.0378905	150	TRUE
	True					True			
1		Procedimento	A069.bmp		1		Procedimento	A083.bmp	
	C069.bmp	C069.bmp	1			C083.bmp	C083.bmp	-1	
	TRUE	-1.2946679	150	TRUE		TRUE	-1.3610298	150	TRUE
	True					True			
1		Procedimento	A070.bmp		1		Procedimento	A084.bmp	
	C070.bmp	C070.bmp	-1			C084.bmp	C084.bmp	-1	
	TRUE	2.2976932	150	TRUE		TRUE	1.9977343	150	TRUE
	True					True			
1		Procedimento	A071.bmp		1		Procedimento	A085.bmp	
	C071.bmp	C071.bmp	1			C085.bmp	C085.bmp	-1	
	TRUE	-2.7213543	150	TRUE		TRUE	-0.5937537	150	TRUE
	True					True			
1		Procedimento	A072.bmp		1		Procedimento	A086.bmp	
	C072.bmp	C072.bmp	1			C086.bmp	C086.bmp	1	
	TRUE	-2.9348902	150	TRUE		TRUE	-2.8424318	150	TRUE
	True					True			
1		Procedimento	A073.bmp		1		Procedimento	A087.bmp	
	C073.bmp	C073.bmp	1			C087.bmp	C087.bmp	-1	
	TRUE	2.9766377	150	TRUE		TRUE	2.7672128	150	TRUE
	True					True			
1		Procedimento	A074.bmp		1		Procedimento	A088.bmp	
	C074.bmp	C074.bmp	-1			C088.bmp	C088.bmp	-1	

	TRUE	0.4926588	150	TRUE		TRUE	-1.2131293	150	TRUE
	True					True			
1		Procedimento	A089.bmp		1		Procedimento	A103.bmp	
	C089.bmp	C089.bmp	-1			C103.bmp	C103.bmp	1	
	TRUE	-1.7659654	150	TRUE		TRUE	-0.1542142	150	TRUE
	True					True			
1		Procedimento	A090.bmp		1		Procedimento	A104.bmp	
	C090.bmp	C090.bmp	-1			C104.bmp	C104.bmp	1	
	TRUE	1.1487539	150	TRUE		TRUE	-0.6267599	150	TRUE
	True					True			
1		Procedimento	A091.bmp		1		Procedimento	A105.bmp	
	C091.bmp	C091.bmp	-1			C105.bmp	C105.bmp	1	
	TRUE	-1.7966678	150	TRUE		TRUE	-0.5911661	150	TRUE
	True					True			
1		Procedimento	A092.bmp		1		Procedimento	A106.bmp	
	C092.bmp	C092.bmp	-1			C106.bmp	C106.bmp	-1	
	TRUE	1.6773271	150	TRUE		TRUE	3.1233675	150	TRUE
	True					True			
1		Procedimento	A093.bmp		1		Procedimento	A107.bmp	
	C093.bmp	C093.bmp	1			C107.bmp	C107.bmp	1	
	TRUE	1.4089266	150	TRUE		TRUE	-0.7358107	150	TRUE
	True					True			
1		Procedimento	A094.bmp		1		Procedimento	A108.bmp	
	C094.bmp	C094.bmp	1			C108.bmp	C108.bmp	-1	
	TRUE	0.1536394	150	TRUE		TRUE	-0.1996959	150	TRUE
	True					True			
1		Procedimento	A095.bmp		1		Procedimento	A109.bmp	
	C095.bmp	C095.bmp	-1			C109.bmp	C109.bmp	1	
	TRUE	-1.0107923	150	TRUE		TRUE	-2.5376445	150	TRUE
	True					True			
1		Procedimento	A096.bmp		1		Procedimento	A110.bmp	
	C096.bmp	C096.bmp	-1			C110.bmp	C110.bmp	1	
	TRUE	2.9363508	150	TRUE		TRUE	2.2424023	150	TRUE
	True					True			
1		Procedimento	A097.bmp		1		Procedimento	A111.bmp	
	C097.bmp	C097.bmp	1			C111.bmp	C111.bmp	-1	
	TRUE	0.4685708	150	TRUE		TRUE	-0.2636117	150	TRUE
	True					True			
1		Procedimento	A098.bmp		1		Procedimento	A112.bmp	
	C098.bmp	C098.bmp	1			C112.bmp	C112.bmp	-1	
	TRUE	2.0797741	150	TRUE		TRUE	-0.6649810	150	TRUE
	True					True			
1		Procedimento	A099.bmp		1		Procedimento	A113.bmp	
	C099.bmp	C099.bmp	1			C113.bmp	C113.bmp	1	
	TRUE	-2.4129567	150	TRUE		TRUE	-0.7358107	150	TRUE
	True					True			
1		Procedimento	A100.bmp		1		Procedimento	A114.bmp	
	C100.bmp	C100.bmp	1			C114.bmp	C114.bmp	-1	
	TRUE	-0.0629643	150	TRUE		TRUE	-0.7182596	150	TRUE
	True					True			
1		Procedimento	A101.bmp		1		Procedimento	A115.bmp	
	C101.bmp	C101.bmp	1			C115.bmp	C115.bmp	1	
	TRUE	-0.5621274	150	TRUE		TRUE	-0.8534708	150	TRUE
	True					True			
1		Procedimento	A102.bmp		1		Procedimento	A116.bmp	
	C102.bmp	C102.bmp	-1			C116.bmp	C116.bmp	1	

	TRUE	-1.8084770	150	TRUE		TRUE	0.0080949	150	TRUE
	True					True			
1		Procedimento	A117.bmp		1		Procedimento	A131.bmp	
	C117.bmp	C117.bmp	-1			C131.bmp	C131.bmp	1	
	TRUE	-1.8966877	150	TRUE		TRUE	2.0961919	150	TRUE
	True					True			
1		Procedimento	A118.bmp		1		Procedimento	A132.bmp	
	C118.bmp	C118.bmp	1			C132.bmp	C132.bmp	-1	
	TRUE	0.4408625	150	TRUE		TRUE	1.2838508	150	TRUE
	True					True			
1		Procedimento	A119.bmp		1		Procedimento	A133.bmp	
	C119.bmp	C119.bmp	-1			C133.bmp	C133.bmp	1	
	TRUE	-1.1598976	150	TRUE		TRUE	-0.5427919	150	TRUE
	True					True			
1		Procedimento	A120.bmp		1		Procedimento	A134.bmp	
	C120.bmp	C120.bmp	1			C134.bmp	C134.bmp	-1	
	TRUE	-1.7444250	150	TRUE		TRUE	-2.5037672	150	TRUE
	True					True			
1		Procedimento	A121.bmp		1		Procedimento	A135.bmp	
	C121.bmp	C121.bmp	1			C135.bmp	C135.bmp	1	
	TRUE	0.0127077	150	TRUE		TRUE	-3.0142713	150	TRUE
	True					True			
1		Procedimento	A122.bmp		1		Procedimento	A136.bmp	
	C122.bmp	C122.bmp	1			C136.bmp	C136.bmp	1	
	TRUE	0.9891633	150	TRUE		TRUE	-2.6686729	150	TRUE
	True					True			
1		Procedimento	A123.bmp		1		Procedimento	A137.bmp	
	C123.bmp	C123.bmp	1			C137.bmp	C137.bmp	-1	
	TRUE	0.2848274	150	TRUE		TRUE	-2.7074241	150	TRUE
	True					True			
1		Procedimento	A124.bmp		1		Procedimento	A138.bmp	
	C124.bmp	C124.bmp	1			C138.bmp	C138.bmp	-1	
	TRUE	-0.3267116	150	TRUE		TRUE	-1.7487009	150	TRUE
	True					True			
1		Procedimento	A125.bmp		1		Procedimento	A139.bmp	
	C125.bmp	C125.bmp	1			C139.bmp	C139.bmp	1	
	TRUE	2.9540177	150	TRUE		TRUE	-1.8511752	150	TRUE
	True					True			
1		Procedimento	A126.bmp		1		Procedimento	A140.bmp	
	C126.bmp	C126.bmp	-1			C140.bmp	C140.bmp	1	
	TRUE	0.4348145	150	TRUE		TRUE	-3.0837844	150	TRUE
	True					True			
1		Procedimento	A127.bmp		1		Procedimento	A141.bmp	
	C127.bmp	C127.bmp	-1			C141.bmp	C141.bmp	-1	
	TRUE	2.0735283	150	TRUE		TRUE	0.4254235	150	TRUE
	True					True			
1		Procedimento	A128.bmp		1		Procedimento	A142.bmp	
	C128.bmp	C128.bmp	-1			C142.bmp	C142.bmp	1	
	TRUE	1.1607091	150	TRUE		TRUE	-1.8491394	150	TRUE
	True					True			
1		Procedimento	A129.bmp		1		Procedimento	A143.bmp	
	C129.bmp	C129.bmp	-1			C143.bmp	C143.bmp	-1	
	TRUE	2.8402134	150	TRUE		TRUE	2.2307781	150	TRUE
	True					True			
1		Procedimento	A130.bmp		1		Procedimento	A144.bmp	
	C130.bmp	C130.bmp	-1			C144.bmp	C144.bmp	-1	

	TRUE	1.0941713	150	TRUE		TRUE	0.3322607	150	TRUE
	True					True			
1		Procedimento	A145.bmp		1		Procedimento	A159.bmp	
	C145.bmp	C145.bmp	1			C159.bmp	C159.bmp	-1	
	TRUE	1.9234764	150	TRUE		TRUE	0.9998550	150	TRUE
	True					True			
1		Procedimento	A146.bmp		1		Procedimento	A160.bmp	
	C146.bmp	C146.bmp	-1			C160.bmp	C160.bmp	1	
	TRUE	2.1563530	150	TRUE		TRUE	-1.2207154	150	TRUE
	True					True			
1		Procedimento	A147.bmp		1		Procedimento	A161.bmp	
	C147.bmp	C147.bmp	1			C161.bmp	C161.bmp	-1	
	TRUE	2.9369745	150	TRUE		TRUE	-0.9254364	150	TRUE
	True					True			
1		Procedimento	A148.bmp		1		Procedimento	A162.bmp	
	C148.bmp	C148.bmp	1			C162.bmp	C162.bmp	1	
	TRUE	-1.9695370	150	TRUE		TRUE	-2.8818495	150	TRUE
	True					True			
1		Procedimento	A149.bmp		1		Procedimento	A163.bmp	
	C149.bmp	C149.bmp	1			C163.bmp	C163.bmp	1	
	TRUE	0.9696498	150	TRUE		TRUE	0.4663582	150	TRUE
	True					True			
1		Procedimento	A150.bmp		1		Procedimento	A164.bmp	
	C150.bmp	C150.bmp	1			C164.bmp	C164.bmp	-1	
	TRUE	2.2181600	150	TRUE		TRUE	-2.6806745	150	TRUE
	True					True			
1		Procedimento	A151.bmp		1		Procedimento	A165.bmp	
	C151.bmp	C151.bmp	-1			C165.bmp	C165.bmp	-1	
	TRUE	1.0695618	150	TRUE		TRUE	-2.2794350	150	TRUE
	True					True			
1		Procedimento	A152.bmp		1		Procedimento	A166.bmp	
	C152.bmp	C152.bmp	-1			C166.bmp	C166.bmp	-1	
	TRUE	-0.7489806	150	TRUE		TRUE	2.5999351	150	TRUE
	True					True			
1		Procedimento	A153.bmp		1		Procedimento	A167.bmp	
	C153.bmp	C153.bmp	-1			C167.bmp	C167.bmp	1	
	TRUE	1.4466116	150	TRUE		TRUE	1.3882973	150	TRUE
	True					True			
1		Procedimento	A154.bmp		1		Procedimento	A168.bmp	
	C154.bmp	C154.bmp	-1			C168.bmp	C168.bmp	1	
	TRUE	1.9854405	150	TRUE		TRUE	-2.9657260	150	TRUE
	True					True			
1		Procedimento	A155.bmp		1		Procedimento	A169.bmp	
	C155.bmp	C155.bmp	1			C169.bmp	C169.bmp	1	
	TRUE	-2.7960500	150	TRUE		TRUE	-1.2898675	150	TRUE
	True					True			
1		Procedimento	A156.bmp		1		Procedimento	A170.bmp	
	C156.bmp	C156.bmp	-1			C170.bmp	C170.bmp	1	
	TRUE	1.6455815	150	TRUE		TRUE	-1.7467072	150	TRUE
	True					False			
1		Procedimento	A157.bmp		1		Procedimento	A171.bmp	
	C157.bmp	C157.bmp	-1			B171.bmp	B171.bmp	1	
	TRUE	-1.0506651	150	TRUE		TRUE	1.1180632	-20	TRUE
	True					True			
1		Procedimento	A158.bmp		1		Procedimento	A172.bmp	
	C158.bmp	C158.bmp	-1			B172.bmp	B172.bmp	-1	

	TRUE	-1.4385909	-20	TRUE		TRUE	-2.8850236	-20	TRUE
	True					True			
1		Procedimento	A173.bmp		1		Procedimento	A187.bmp	
	B173.bmp	B173.bmp	1			B187.bmp	B187.bmp	-1	
	TRUE	-0.7242598	-20	TRUE		TRUE	-2.7663950	-20	TRUE
	True					True			
1		Procedimento	A174.bmp		1		Procedimento	A188.bmp	
	B174.bmp	B174.bmp	-1			B188.bmp	B188.bmp	-1	
	TRUE	1.4971983	-20	TRUE		TRUE	1.7897057	-20	TRUE
	True					True			
1		Procedimento	A175.bmp		1		Procedimento	A189.bmp	
	B175.bmp	B175.bmp	-1			B189.bmp	B189.bmp	1	
	TRUE	-0.9827765	-20	TRUE		TRUE	0.2503835	-20	TRUE
	True					True			
1		Procedimento	A176.bmp		1		Procedimento	A190.bmp	
	B176.bmp	B176.bmp	1			B190.bmp	B190.bmp	1	
	TRUE	1.5457500	-20	TRUE		TRUE	0.7699146	-20	TRUE
	True					False			
1		Procedimento	A177.bmp		1		Procedimento	A191.bmp	
	B177.bmp	B177.bmp	1			B191.bmp	B191.bmp	1	
	TRUE	1.7527088	-20	TRUE		TRUE	1.3987943	-20	TRUE
	True					True			
1		Procedimento	A178.bmp		1		Procedimento	A192.bmp	
	B178.bmp	B178.bmp	1			B192.bmp	B192.bmp	1	
	TRUE	-0.1351013	-20	TRUE		TRUE	-0.7358107	-20	TRUE
	True					True			
1		Procedimento	A179.bmp		1		Procedimento	A193.bmp	
	B179.bmp	B179.bmp	1			B193.bmp	B193.bmp	1	
	TRUE	1.0682676	-20	TRUE		TRUE	-1.8606546	-20	TRUE
	True					True			
1		Procedimento	A180.bmp		1		Procedimento	A194.bmp	
	B180.bmp	B180.bmp	1			B194.bmp	B194.bmp	-1	
	TRUE	-2.8037882	-20	TRUE		TRUE	0.3413802	-20	TRUE
	False					True			
1		Procedimento	A181.bmp		1		Procedimento	A195.bmp	
	B181.bmp	B181.bmp	-1			B195.bmp	B195.bmp	-1	
	TRUE	-1.7804356	-20	TRUE		TRUE	-1.6062496	-20	TRUE
	True					True			
1		Procedimento	A182.bmp		1		Procedimento	A196.bmp	
	B182.bmp	B182.bmp	1			B196.bmp	B196.bmp	1	
	TRUE	-0.3162854	-20	TRUE		TRUE	-2.8541290	-20	TRUE
	True					True			
1		Procedimento	A183.bmp		1		Procedimento	A197.bmp	
	B183.bmp	B183.bmp	-1			B197.bmp	B197.bmp	1	
	TRUE	1.4311744	-20	TRUE		TRUE	1.5964719	-20	TRUE
	True					True			
1		Procedimento	A184.bmp		1		Procedimento	A198.bmp	
	B184.bmp	B184.bmp	1			B198.bmp	B198.bmp	-1	
	TRUE	-1.7058369	-20	TRUE		TRUE	0.9102107	-20	TRUE
	True					True			
1		Procedimento	A185.bmp		1		Procedimento	A199.bmp	
	B185.bmp	B185.bmp	1			B199.bmp	B199.bmp	-1	
	TRUE	-1.2469053	-20	TRUE		TRUE	1.3057988	-20	TRUE
	True					True			
1		Procedimento	A186.bmp		1		Procedimento	A200.bmp	
	B186.bmp	B186.bmp	-1			B200.bmp	B200.bmp	-1	

	TRUE	0.9377360	-20	TRUE		TRUE	2.9473446	-20	TRUE
	False					True			
1		Procedimento	A201.bmp		1		Procedimento	A215.bmp	
	B201.bmp	B201.bmp	-1			B215.bmp	B215.bmp	-1	
	TRUE	-1.7440036	-20	TRUE		TRUE	2.4906733	-20	TRUE
	True					True			
1		Procedimento	A202.bmp		1		Procedimento	A216.bmp	
	B202.bmp	B202.bmp	-1			B216.bmp	B216.bmp	1	
	TRUE	-2.3247549	-20	TRUE		TRUE	-1.7817871	-20	TRUE
	True					True			
1		Procedimento	A203.bmp		1		Procedimento	A217.bmp	
	B203.bmp	B203.bmp	-1			B217.bmp	B217.bmp	1	
	TRUE	-2.6649255	-20	TRUE		TRUE	-0.9620098	-20	TRUE
	True					True			
1		Procedimento	A204.bmp		1		Procedimento	A218.bmp	
	B204.bmp	B204.bmp	1			B218.bmp	B218.bmp	1	
	TRUE	2.6026033	-20	TRUE		TRUE	2.6587854	-20	TRUE
	True					True			
1		Procedimento	A205.bmp		1		Procedimento	A219.bmp	
	B205.bmp	B205.bmp	-1			B219.bmp	B219.bmp	1	
	TRUE	1.7743025	-20	TRUE		TRUE	3.0053243	-20	TRUE
	True					True			
1		Procedimento	A206.bmp		1		Procedimento	A220.bmp	
	B206.bmp	B206.bmp	1			B220.bmp	B220.bmp	-1	
	TRUE	0.5662828	-20	TRUE		TRUE	-2.4724898	-20	TRUE
	True					False			
1		Procedimento	A207.bmp		1		Procedimento	A221.bmp	
	B207.bmp	B207.bmp	1			B221.bmp	B221.bmp	-1	
	TRUE	2.1651129	-20	TRUE		TRUE	2.8298246	-20	TRUE
	True					True			
1		Procedimento	A208.bmp		1		Procedimento	A222.bmp	
	B208.bmp	B208.bmp	-1			B222.bmp	B222.bmp	-1	
	TRUE	1.4064262	-20	TRUE		TRUE	-1.0172918	-20	TRUE
	True					True			
1		Procedimento	A209.bmp		1		Procedimento	A223.bmp	
	B209.bmp	B209.bmp	-1			B223.bmp	B223.bmp	1	
	TRUE	2.1720514	-20	TRUE		TRUE	2.5061714	-20	TRUE
	True					True			
1		Procedimento	A210.bmp		1		Procedimento	A224.bmp	
	B210.bmp	B210.bmp	-1			B224.bmp	B224.bmp	1	
	TRUE	-0.4654296	-20	TRUE		TRUE	1.6308601	-20	TRUE
	True					True			
1		Procedimento	A211.bmp		1		Procedimento	A225.bmp	
	B211.bmp	B211.bmp	1			B225.bmp	B225.bmp	-1	
	TRUE	-1.1810184	-20	TRUE		TRUE	-1.7869213	-20	TRUE
	True					True			
1		Procedimento	A212.bmp		1		Procedimento	A226.bmp	
	B212.bmp	B212.bmp	-1			B226.bmp	B226.bmp	-1	
	TRUE	-1.1984282	-20	TRUE		TRUE	0.8038668	-20	TRUE
	True					True			
1		Procedimento	A213.bmp		1		Procedimento	A227.bmp	
	B213.bmp	B213.bmp	-1			B227.bmp	B227.bmp	1	
	TRUE	1.9628589	-20	TRUE		TRUE	0.4524826	-20	TRUE
	True					True			
1		Procedimento	A214.bmp		1		Procedimento	A228.bmp	
	B214.bmp	B214.bmp	-1			B228.bmp	B228.bmp	1	

	TRUE	-0.0290974	-20	TRUE		TRUE	-2.4374297	150	TRUE
	True					True			
1		Procedimento	A229.bmp		1		Procedimento	A243.bmp	
	B229.bmp	B229.bmp	-1			C243.bmp	C243.bmp	-1	
	TRUE	-1.6946334	-20	TRUE		TRUE	-1.1491489	150	TRUE
	True					True			
1		Procedimento	A230.bmp		1		Procedimento	A244.bmp	
	B230.bmp	B230.bmp	1			C244.bmp	C244.bmp	1	
	TRUE	2.2549576	-20	TRUE		TRUE	-2.8975615	150	TRUE
	True					True			
1		Procedimento	A231.bmp		1		Procedimento	A245.bmp	
	B231.bmp	B231.bmp	-1			C245.bmp	C245.bmp	-1	
	TRUE	-2.9432567	-20	TRUE		TRUE	0.1916288	150	TRUE
	True					True			
1		Procedimento	A232.bmp		1		Procedimento	A246.bmp	
	B232.bmp	B232.bmp	1			C246.bmp	C246.bmp	1	
	TRUE	-2.2957315	-20	TRUE		TRUE	0.0566668	150	TRUE
	True					True			
1		Procedimento	A233.bmp		1		Procedimento	A247.bmp	
	B233.bmp	B233.bmp	-1			C247.bmp	C247.bmp	-1	
	TRUE	-3.0825544	-20	TRUE		TRUE	-0.3470227	150	TRUE
	True					True			
1		Procedimento	A234.bmp		1		Procedimento	A248.bmp	
	B234.bmp	B234.bmp	1			C248.bmp	C248.bmp	-1	
	TRUE	-0.6576618	-20	TRUE		FALSE	-1.2953604	150	TRUE
	True					True			
1		Procedimento	A235.bmp		1		Procedimento	A249.bmp	
	B235.bmp	B235.bmp	-1			C249.bmp	C249.bmp	-1	
	TRUE	-1.7577582	-20	TRUE		TRUE	1.0375267	150	TRUE
	True					True			
1		Procedimento	A236.bmp		1		Procedimento	A250.bmp	
	B236.bmp	B236.bmp	1			C250.bmp	C250.bmp	1	
	TRUE	0.2163953	-20	TRUE		TRUE	0.5074478	150	TRUE
	True					True			
1		Procedimento	A237.bmp		1		Procedimento	A251.bmp	
	B237.bmp	B237.bmp	1			C251.bmp	C251.bmp	1	
	TRUE	1.0555678	-20	TRUE		TRUE	-0.7453183	150	TRUE
	True					True			
1		Procedimento	A238.bmp		1		Procedimento	A252.bmp	
	B238.bmp	B238.bmp	-1			C252.bmp	C252.bmp	-1	
	TRUE	-2.5517211	-20	TRUE		TRUE	-2.0078745	150	TRUE
	True					True			
1		Procedimento	A239.bmp		1		Procedimento	A253.bmp	
	B239.bmp	B239.bmp	-1			C253.bmp	C253.bmp	-1	
	TRUE	-2.4399900	-20	TRUE		FALSE	2.7091292	150	TRUE
	True					True			
1		Procedimento	A240.bmp		1		Procedimento	A254.bmp	
	B240.bmp	B240.bmp	1			C254.bmp	C254.bmp	1	
	TRUE	0.5642562	-20	TRUE		FALSE	0.5417681	150	TRUE
	False					True			
1		Procedimento	A241.bmp		1		Procedimento	A255.bmp	
	C241.bmp	C241.bmp	1			C255.bmp	C255.bmp	-1	
	TRUE	2.1619420	150	TRUE		FALSE	-1.6088535	150	TRUE
	True					True			
1		Procedimento	A242.bmp		1		Procedimento	A256.bmp	
	C242.bmp	C242.bmp	1			C256.bmp	C256.bmp	1	

	FALSE	-0.4595570	150	TRUE		TRUE	1.2298951	150	TRUE
	True					True			
1		Procedimento	A257.bmp		1		Procedimento	A269.bmp	
	C257.bmp	C257.bmp	-1			C269.bmp	C269.bmp	-1	
	TRUE	1.7494465	150	TRUE		FALSE	-2.6504282	150	TRUE
	True					True			
1		Procedimento	A258.bmp		1		Procedimento	A270.bmp	
	C258.bmp	C258.bmp	1			C270.bmp	C270.bmp	-1	
	TRUE	0.1252245	150	TRUE		TRUE	2.8769079	150	TRUE
	True					True			
1		Procedimento	A259.bmp		1		Procedimento	A271.bmp	
	C259.bmp	C259.bmp	1			C271.bmp	C271.bmp	1	
	FALSE	2.9886741	150	TRUE		FALSE	0.3725245	150	TRUE
	True					True			
1		Procedimento	A260.bmp		1		Procedimento	A272.bmp	
	C260.bmp	C260.bmp	-1			C272.bmp	C272.bmp	-1	
	TRUE	1.8810875	150	TRUE		FALSE	3.0769215	150	TRUE
	True					True			
1		Procedimento	A261.bmp		1		Procedimento	A273.bmp	
	C261.bmp	C261.bmp	1			C273.bmp	C273.bmp	1	
	TRUE	1.9875847	150	TRUE		FALSE	3.0399173	150	TRUE
	True					True			
1		Procedimento	A262.bmp		1		Procedimento	A274.bmp	
	C262.bmp	C262.bmp	1			C274.bmp	C274.bmp	-1	
	TRUE	-0.8056154	150	TRUE		TRUE	2.2666353	150	TRUE
	True					True			
1		Procedimento	A263.bmp		1		Procedimento	A275.bmp	
	C263.bmp	C263.bmp	1			C275.bmp	C275.bmp	-1	
	TRUE	-1.3412158	150	TRUE		TRUE	0.1369672	150	TRUE
	True					True			
1		Procedimento	A264.bmp		1		Procedimento	A276.bmp	
	C264.bmp	C264.bmp	1			C276.bmp	C276.bmp	-1	
	TRUE	0.1740936	150	TRUE		FALSE	0.1918154	150	TRUE
	True					True			
1		Procedimento	A265.bmp		1		Procedimento	A277.bmp	
	C265.bmp	C265.bmp	-1			C277.bmp	C277.bmp	1	
	FALSE	-0.0437216	150	TRUE		FALSE	-1.6417734	150	TRUE
	True					True			
1		Procedimento	A266.bmp		1		Procedimento	A278.bmp	
	C266.bmp	C266.bmp	1			C278.bmp	C278.bmp	1	
	FALSE	0.6274037	150	TRUE		TRUE	-0.7358107	150	TRUE
	True					True			
1		Procedimento	A267.bmp		1		Procedimento	A279.bmp	
	C267.bmp	C267.bmp	-1			C279.bmp	C279.bmp	-1	
	TRUE	-1.9456219	150	TRUE		TRUE	-2.5764074	150	TRUE
	True					True			
1		Procedimento	A268.bmp		1		Procedimento	A280.bmp	
	C268.bmp	C268.bmp	-1			C280.bmp	C280.bmp	1	
						FALSE	2.3993134	150	TRUE

Anexo 2. Programa gerador de estímulos.

```
Const conZ = 0
Const conU = 1
Const conB = 255
Const conI = 32767
Const conM = 1023
Const conIM = 0.000001
Const conO = 6
Const conW = 5
Const conCj = 3
Const conF = 9
Const conX = 10
Const conS = 100
Const conSz = "0"
Const conData = "Data"
Const conSCj = "Cj"
Const conBar = "\"
Const conEst = "Estímulos"
Const conTel = "Telas"
Const conPar = "Partes"
Const conFac = "Faces"
Const conTex = "txt"
Const conBMP = "bmp"
Const conPto = "."
Const conTimer = "Tempo restante... "
Const conSA = "A"
Const conSB = "B"
Const conSC = "C"
Dim bytUtil As Byte
'Data
Dim sngD1 As Single
Dim sngD2 As Single
Dim sngD3 As Single
Dim sngD4 As Single
Dim sngD5 As Single
Dim sngD6 As Single
Dim intD1 As Integer
Dim intD2 As Integer
Dim bytD1 As Byte
Dim bytD2 As Byte
Dim bytD3 As Byte
Dim bytD4 As Byte
'Configuração
Dim conRdGrau As Single
Dim conGrauRd As Single
Dim bytOOp As Byte
Dim booNTU As Boolean
Dim booCjPar As Boolean
Dim bytCjZ As Byte
Dim sngR As Single
```

```
Dim sngPro As Single
Dim intX As Integer
Dim intY As Integer
Dim strPath As String
Dim strPathA As String
Dim strPathB As String
Dim strFile As String
Dim intO As Integer
Dim booOpts As Boolean
' Objeto
Dim bytO As Byte
Dim sngTam(conZ To conO) As Byte
Dim bytCP(conZ To conO) As Byte
Dim bytCjt(conZ To conO) As Byte
Dim sngAng(conZ To conO) As Single
Dim sngDis(conZ To conO) As Single
Dim sngArAng(conZ To conO) As Single
Dim sngArCom(conZ To conO) As Single
Dim sngArLar(conZ To conO) As Single
Dim sngArCom1 As Single
Dim sngArLar1 As Single
Dim sngArC1 As Single
Dim sngArX1 As Single
Dim sngArY1 As Single
Dim sngArX2 As Single
Dim sngArY2 As Single
Dim sngArR As Single
Dim sngPARX1 As Single
Dim sngPARX2 As Single
Dim sngPARY1 As Single
Dim sngPARY2 As Single
Dim sngAux As Single
Dim sngAux1 As Single
Dim sngAuxT(conU To 3) As Single
Dim bytRGB As Byte
Dim sngObj(conZ To conO, conZ To conM, conZ To conM) As Single
'Tela
Dim bytMod As Byte
Dim intT As Integer
Dim intNT As Integer
Dim bytCj(conZ To conM) As Byte
Dim sngAngulo(conZ To conM) As Single
```

```
Dim sngGiro(conZ To conO, conZ To conM) As Single
Dim sngCro(conZ To conO, conZ To conM) As Single
Dim sngCroA As Single
Dim bytEsq(conZ To conM) As Byte
Dim bytDir(conZ To conM) As Byte
Dim bytAce(conZ To conM) As Byte
Dim bytEsq1 As Byte
Dim bytDir1 As Byte
Dim bytNivel(conZ To conO) As Byte
Dim bytAntiNivel(conZ To conO) As Byte
Dim bytAN1 As Byte
' Parte
Dim bytP As Byte
Dim bytPS As Byte
Dim bytNP(conZ To conCj) As Byte
Dim bytNP1 As Byte
Dim intBola(conZ To conCj, -conU To conB) As Integer
Dim sngRALfa(conZ To conCj, -conU To conB) As Single
Dim sngPD(conZ To conCj, -conU To conB) As Single
Dim sngPALfa(conZ To conCj, -conU To conB) As Single
' Face
Dim bytNF(conZ To conCj) As Byte
Dim bytNF1 As Byte
Dim bytD(conZ To conCj, conZ To conB) As Byte
Dim bytA(conZ To conCj, conZ To conB) As Byte
Dim bytOp(conZ To conCj, conZ To conB) As Byte
Dim bytB(conZ To conCj, conZ To conB) As Byte
' Cálculo
Dim sngTimer As Single
Dim sngTimer1 As Single
Dim sngRALfa1 As Single
Dim sngLogoE As Single
Dim sngLogoOu As Single
Dim strCalc As String
Dim strCalc1 As String
Dim bytFacOb(conZ To conO) As Byte
```

```

Dim sngLogo(conZ To 11) As Single
Dim bytF As Byte
Dim bytConj As Byte
Dim dblPdtp As Double
Dim dblRALfatp As Double
Dim bytCP1 As Byte
Dim dblCPoR As Double
Dim dblAngulo As Double
Dim dblAngulo1 As Double
Dim dblRALfa As Double
Dim dblPALfa As Double
Dim dblAlfa As Double
Dim dblCPoRCA As Double
Dim dblCPoRSA As Double
Dim dblRaio(-conU To conB) As Double
Dim dblRaio1 As Double
Dim dblK As Double
Dim dblX(-conU To conB) As Double
Dim dblY(-conU To conB) As Double
Dim intMX As Integer
Dim intMY As Integer
Dim sngDF(conZ To conB) As Single
Dim sngDFA As Single
Dim sngDFB As Single
Dim dblDX As Double
Dim dblDY As Double
Dim dblDX1 As Double
Dim dblDY1 As Double
Dim sngTama As Single
Dim intP As Integer
' Suave
Dim bytSV(conZ To conM, conZ To conM) As Byte
Dim intDX As Integer
Dim intDY As Integer
Dim dblRX(conZ To conU, conU To conB, conZ To conM) As Double
Dim dblRY(conZ To conU, conU To conB, conZ To conM) As Double
Dim sngCooXM As Single
Dim sngCooYM As Single
Dim dblCooXM As Double
Dim dblCooYM As Double
Dim dblCooM As Double
Dim dblCooSin As Double
Dim dblCooCos As Double
Dim dblCooArc As Double
Dim intCooX As Integer
Dim intCooY As Integer

```

```

Dim intQsi As Integer
Dim intQed As Integer
Dim intQis As Integer
Dim intQde As Integer
Dim sngDF1 As Single
Dim sngX1 As Single
Dim sngY1 As Single
' Bitmap
Dim intVis As Integer
Dim intVis1 As Integer
Dim intVis2 As Integer
Dim intPXMin(conZ To conO) As Integer
Dim intPXMax(conZ To conO) As Integer
Dim intPYMin(conZ To conO, conZ To conM) As Integer
Dim intPYMax(conZ To conO, conZ To conM) As Integer
Dim intPXMin1 As Integer
Dim intPXMax1 As Integer
Dim intPYMin1 As Integer
Dim intPYMax1 As Integer
Dim intPYLMin1 As Integer
Dim intPYLMax1 As Integer
Dim intPX As Integer
Dim intPY As Integer
Dim intPYy As Integer
Dim bytAux1 As Byte
Dim intAux1 As Integer
Dim bytAux(conU To 3) As Byte
Dim intAux(conU To 3) As Integer
Dim intPCont As Integer
Dim booPCont As Boolean
Dim intRCont As Integer
Dim intInt(conZ To conI) As Integer
Dim bytCor(conU To 3, conZ To conI) As Byte
Dim strNome As String
Dim bytState As Byte
' Cor
Dim sngCRGB(conU To 3, conZ To conCj, conZ To conB) As Single
'Dim sngCorR(conZ To conCj, conZ To conB) As Single
'Dim sngCorG(conZ To conCj, conZ To conB) As Single

```

```

'Dim sngCorB(conZ To conCj, conZ To conB) As Single
Dim sngTom(conU To 3, conZ To conO) As Single
Dim bytRGB1 As Byte

Private Function funVinc() As Boolean

    funVinc = optTela.Value Or Not frmConfig.Enabled

End Function

Private Sub subFOpt(ByVal Ob As Byte, ByVal Hab As Boolean)

    optFace0(Ob).Enabled = Hab
    optFace1(Ob).Enabled = Hab
    optFace2(Ob).Enabled = Hab
    optFace3(Ob).Enabled = Hab
    optFace4(Ob).Enabled = Hab
    optFace5(Ob).Enabled = Hab
    optFace6(Ob).Enabled = Hab
    optFace7(Ob).Enabled = Hab
    optFace8(Ob).Enabled = Hab
    optFace9(Ob).Enabled = Hab

End Sub

Private Sub subFace(ByVal Obj As Byte, ByVal Habil As Boolean)

    frmFace(Obj).Enabled = Habil
    subFOpt Obj, Habil

End Sub

Private Sub subConjunto(ByVal Hab As Boolean)

    frmConjunto.Enabled = Hab
    opt1.Enabled = Hab
    opt2.Enabled = Hab
    opt3.Enabled = Hab

End Sub

Private Sub subCj(ByVal Ob As Byte, ByVal Hab As Boolean)

```

```

frmCj(Ob).Enabled = Hab
optCj0(Ob).Enabled = Hab
optCj1(Ob).Enabled = Hab

End Sub

Private Sub subCP(ByVal Ob
As Byte, ByVal Hab As
Boolean)

frmCP(Ob).Enabled = Hab
optCP0(Ob).Enabled = Hab
optCP1(Ob).Enabled = Hab

End Sub

Private Sub subTela(ByVal
Hab As Boolean)

frmTela.Enabled = Hab
optA.Enabled = Hab
optB.Enabled = Hab
optC.Enabled = Hab

End Sub

Private Sub subVis(ByVal Ob
As Byte, ByVal Hab As
Boolean)

frmVis(Ob).Enabled = Hab
optVis0(Ob).Enabled = Hab
optVis1(Ob).Enabled = Hab

End Sub

Private Sub subCorInt()

For bytRGB1 = 1 To 3
  bytAux1 =
  bytAux(bytRGB1)
  bytCor(bytRGB1, intPCont)
= bytAux1
  intAux(bytRGB1) = bytAux1
Next
intInt(intPCont) = intPY

End Sub

Private Sub subCalTela()

intPCont = conZ
For intPX = conZ To intX
  For bytRGB1 = conU To 3
    intAux(bytRGB1) = -conU
    Next
    For intPY = conZ To intY
      For bytRGB1 = conU To 3
        sngAuxT(bytRGB1) = conZ
        Next
        sngAux1 = conZ
        intVis1 = intVis
        For bytO = conZ To conO
          bytAN1 =
          bytAntiNivel(bytO)
          bytMod = intVis1 Mod 2
          sngAux = bytMod *
          sngObj(bytAN1, intPX,
          intPY)
          sngCroA = sngCro(bytAN1,
          intT)
          For bytRGB1 = conU To 3
            sngAux1 =
            sngAuxT(bytRGB1)
            sngAux1 = sngAux1 +
            (sngCroA *
            sngTom(bytRGB1, bytAN1) +
            (conU - sngCroA) / 2) *
            sngAux - sngAux1 * sngAux
            sngAuxT(bytRGB1) =
            sngAux1
            Next
            intVis1 = (intVis1 -
            bytMod) \ 2
            Next
            booPCont = True
            For bytRGB1 = conU To 3
              bytAux1 = Int(conB *
              sngAuxT(bytRGB1))
              bytAux(bytRGB1) =
              bytAux1
              booPCont = booPCont And
              (bytAux1 =
              intAux(bytRGB1))
              Next
              If Not booPCont Then
                intPCont = intPCont +
                conU
                subCorInt
                End If
                Next
                intPCont = intPCont +
                Sgn(intY - intInt(intPCont))
                subCorInt
                Next
                intPX = conZ
                intPY = conZ
                Picture1.Width = intX +
                conU
                Picture1.Height = intY +
                conU
                intPCont = intPCont - conU
                For intRCont = conU To
                intPCont
                  intPYy = intInt(intRCont +
                  conU)
                  bytAux1 = Sgn(intPYy)
                  intAux1 = conU - bytAux1
                  intPYy = bytAux1 * intPYy
                  + intAux1 * (intPY + conU)
                  Picture1.Line (intPX, intPY)-
                  (intPX, intPYy),
                  RGB(bytCor(conU,
                  intRCont), bytCor(2,
                  intRCont), bytCor(3,
                  intRCont))
                  intPX = intPX + intAux1
                  intPY = bytAux1 * intPYy
                Next
                Picture1.Visible = True
              End Sub

Private Function funPriva(X
As Double, Y As Double) As
Byte

  funPriva = Fix(0.5 +
  Sgn(dblRai01 - X - Y))

End Function

Private Sub subParX()

  sngParX1 = intPX -
  sngArX1
  sngParX2 = intPX -
  sngArX2
  sngParX1 = sngParX1 *
  sngParX1
  sngParX2 = sngParX2 *
  sngParX2

End Sub

Private Function funDEI() As
Boolean

  sngParY1 = intPY -
  sngArY1
  sngParY2 = intPY -
  sngArY2
  funDEI = CBool(sngArR >
  Sqr(sngParX1 + sngParY1 *
  sngParY1) + Sqr(sngParX2
  + sngParY2 * sngParY2))

```

End Function

Private Function

funSEx(ByVal Coord As Double) As Double

funSEx = Coord + conIM * (conU - Abs(Sgn(Coord - Int(Coord))))

End Function

Private Sub subMudaMin(Min As Integer, Coor As Integer)

Min = Min + (Sgn(Fix(0.5 - Sgn(Min)))) * (Coor + conU)

End Sub

Private Function

funNumZU(MUM As Integer, Corte As Single) As Byte

funNumZU = Sgn(conU + Sgn(MUM - Corte))

End Function

Private Sub subCooX(C4oX As Integer)

intCooX = intPX + C4oX
sngCooXM = intCooX - 0.5

End Sub

Private Sub subCooY(C4oY As Integer)

intCooY = intPY + C4oY
sngCooYM = intCooY - 0.5

End Sub

Private Sub subCoo(CooX As Integer, CooY As Integer)

subCooX CooX
subCooY CooY

End Sub

Private Function funFD(CooX As Integer, CooY As Integer) As Byte

subCoo CooX, CooY

funFD =

funNumZU(Sgn((sngCooXM - dblRX(conZ, bytP, intCooY)) * (dblRX(conU, bytP, intCooY) - sngCooXM)) + Sgn((sngCooYM - dblRY(conZ, bytP, intCooX)) * (dblRY(conU, bytP, intCooX) - sngCooYM)), 1.5)

End Function

Private Sub subRaioXY()

dblRaio1 = dblRaio(bytP)
sngX1 = dblX(bytP)
sngY1 = dblY(bytP)

End Sub

Private Function funSet(Dist As Double)

Var = intPX + intPY + bytP
dblCooSin = 0.5 * Dist / dblRaio1
dblCooCos = Sqr(conU - dblCooSin * dblCooSin)
dblCooArc = Atn(dblCooSin / dblCooCos)
funSet = (dblCooArc - dblCooSin * dblCooCos) * dblRaio1 * dblRaio1

End Function

Private Sub subDF1(sngAd As Single)

sngDF1 = sngDF1 + sngAd

End Sub

Private Sub subTS(TSX As Integer, TSY As Integer, MaMe As Integer)

subCoo TSX, TSY
dblCooXM = sngCooXM - dblRX(funNumZU(intCooX, sngX1), bytP, intCooY)
dblCooYM = sngCooYM - dblRY(funNumZU(intCooY, sngY1), bytP, intCooX)

subDF1 MaMe * 0.5 *

Abs(dblCooXM * dblCooYM) + funSet(Sqr(dblCooXM * dblCooXM + dblCooYM * dblCooYM))

End Sub

Private Function funCas(CP1 As Single, QP1 As Integer, CP2 As Single, QP2 As Single, Dist1 As Double) As Double

funCas = funNumZU(Sgn((CP1 - QP1 + 0.5) * (QP1 + 0.5 - CP1)) + Sgn(dblRaio1 - Abs(CP2 - QP2)), 1.5) * funSet(Dist1)

End Function

Private Sub subCasV(CVY As Integer, SoSuV As Integer)

subCooY CVY
subDF1 SoSuV * funCas(sngX1, intPX, sngY1, sngCooYM, dblRX(conU, bytP, intCooY) - dblRX(conZ, bytP, intCooY))

End Sub

Private Sub subCasH(CHX As Integer, SoSuH As Integer)

subCooX CHX
subDF1 SoSuH * funCas(sngY1, intPY, sngX1, sngCooXM, dblRY(conU, bytP, intCooX) - dblRY(conZ, bytP, intCooX))

End Sub

Private Sub subOV()

intVis = conZ
For intO = conO To conZ Step -conU
intVis = intVis + intVis
If optVis1(bytAntiNivel(intO)).


```

Value Then intVis = intVis +
conU
Next

```

```
End Sub
```

```
Private Sub subPre()
```

```

intT = intT + conU
opt1.Value =
CBool(bytCj(intT) = conU)
opt2.Value =
CBool(bytCj(intT) = 2)
opt3.Value =
CBool(bytCj(intT) = 3)
txtAngulo = conRdGrau *
sngAngulo(intT)
For bytO = conZ To conO
txtGiro(bytO) = conRdGrau
* sngGiro(bytO, intT)
txtCro(bytO) =
sngCro(bytO, intT)
Next
bytEsq1 = bytEsq(intT)
bytDir1 = bytDir(intT)
optFace1(conU).Value =
CBool(bytEsq1 = conU)
optFace2(conU).Value =
CBool(bytEsq1 = 2)
optFace3(conU).Value =
CBool(bytEsq1 = 3)
optFace4(conU).Value =
CBool(bytEsq1 = 4)
optFace5(conU).Value =
CBool(bytEsq1 = 5)
optFace6(conU).Value =
CBool(bytEsq1 = 6)
optFace1(3).Value =
CBool(bytDir1 = conU)
optFace2(3).Value =
CBool(bytDir1 = 2)
optFace3(3).Value =
CBool(bytDir1 = 3)
optFace4(3).Value =
CBool(bytDir1 = 4)
optFace5(3).Value =
CBool(bytDir1 = 5)
optFace6(3).Value =
CBool(bytDir1 = 6)
optFace7(conO).Value = True
optFace8(conZ).Value =
CBool(bytAce(intT) = conU)
optFace9(conZ).Value = Not
optFace8(conZ).Value

```

```
End Sub
```

```
Private Function funCalc() As
Boolean
```

```

' homogeniza
dados de cálculo e
strCalc = sngR & intX & intY
& sngPro & dblAngulo1 &
bytCJz
For bytO = conZ To conO
strCalc = strCalc &
bytAntiNivel(bytO) &
sngTam(bytO) &
sngAng(bytO) &
sngGiro(bytO, intT) &
sngCro(bytO, intT) &
sngDis(bytO) & bytCP(bytO)
& bytCjt(bytO) &
bytFacOb(bytO)
Next
funCalc = CBool(strCalc <>
strCalc1) 'verifica mudanças
que justifiquem cálculo
If funCalc Then
intMX = intX \ 2
intMY = intY \ 2
Picture1.Visible = False
lblTimer.Visible = True
sngTimer = Timer
For bytO = conZ To conO
intPXMax1 =
intPXMax(bytO)
intPXMin1 =
intPXMin(bytO)
For intPX = intPXMin1 To
intPXMax1
intPYMin1 =
intPYMin(bytO, intPX)
intPYMax1 =
intPYMax(bytO, intPX)
For intPY = intPYMin1 To
intPYMax1
sngObj(bytO, intPX,
intPY) = conZ
Next
Next
sngTama = sngPro *
sngTam(bytO)
bytCP1 = bytCP(bytO)
' bytCP1 = 0
bytConj = bytCjt(bytO) *
bytCJz
bytNP1 = bytNP(bytConj)
bytNF1 = bytNF(bytConj)
dblCPoR = bytCP1 *
sngDis(bytO) * sngPro

```

```

' dblCPoR = 0
dblAlfa = sngAng(bytO) +
dblAngulo1
' dblAngulo = (conU -
bytCP1) * dblAlfa
dblAngulo = dblAlfa +
sngGiro(bytO, intT)
dblCPoRCA = intMX +
dblCPoR * Cos(dblAlfa)
dblCPoRSA = intMY -
dblCPoR * Sin(dblAlfa)
sngArCom1 =
sngArCom(bytO)
sngArLar1 =
sngArLar(bytO)
sngArC1 = Sqr(2 *
(sngArCom1 * sngArCom1 -
sngArLar1 * sngArLar1))
sngRAIfa(bytConj, conZ) =
conZ
sngPD(bytConj, conZ) =
sngArC1
sngPAIfa(bytConj, conZ) =
sngArAng(bytO)
sngRAIfa(bytConj, -conU) =
conZ
sngPD(bytConj, -conU) = -
sngArC1
sngPAIfa(bytConj, -conU) =
sngArAng(bytO)
For intP = -conU To bytNP1
sngRAIfa1 =
sngRAIfa(bytConj, intP)
dblRAIfatp = sngTama *
sngRAIfa1
dblPdtp = sngTama *
sngPD(bytConj, intP)
dblRAIfa = dblAngulo +
sngRAIfa1
dblPAIfa = dblAngulo +
sngPAIfa(bytConj, intP)
dblRaio1 = dblRAIfatp +
Abs(intBola(bytConj, intP)) *
(sngR - dblRAIfatp)
dblRaio(intP) = dblRaio1
dblK = intBola(bytConj,
intP) * dblRaio1
dblX(intP) =
funSEx(dblCPoRCA +
dblPdtp * Cos(dblPAIfa) -
dblK * Sin(dblRAIfa))
dblY(intP) =
funSEx(dblCPoRSA - dblPdtp
* Sin(dblPAIfa) - dblK *
Cos(dblRAIfa))
Next

```

```

sngArX1 = dblX(-conU)
sngArY1 = dblY(-conU)
sngArX2 = dblX(conZ)
sngArY2 = dblY(conZ)
sngArR = Sqr(2) * 2 *
sngArCom1 * sngTama
intPXMax1 = -4
intPYLMax1 = intPXMax1
intPXMin1 = -conU
intPYLMin1 = conM
For intPX = conZ To intX
intPYMax1 = -4
intPYMin1 = -conU
sngPArX1 = intPX -
sngArX1
sngPArX2 = intPX -
sngArX2
sngPArX1 = sngPArX1 *
sngPArX1
sngPArX2 = sngPArX2 *
sngPArX2
For intPY = conZ To intY
sngPArY1 = intPY -
sngArY1
sngPArY2 = intPY -
sngArY2
If sngArR > Sqr(sngPArX1
+ sngPArY1 * sngPArY1) +
Sqr(sngPArX2 + sngPArY2 *
sngPArY2) Then
intPXMax1 = intPX
intPYMax1 = intPY
subMudaMin intPXMin1,
intPX
subMudaMin intPYMin1,
intPY
End If
Next
If intPYLMax1 <
intPYMax1 Then
intPYLMax1 = intPYMax1
If (intPYLMin1 >
intPYMin1) And (intPYMin1
> -conU) Then intPYLMin1 =
intPYMin1
intPYMax(bytO, intPX) =
intPYMax1 + conU
intPYMin(bytO, intPX) =
intPYMin1 - conU
Next
intPXMax1 = intPXMax1 +
2
intPXMin1 = intPXMin1 -
conU
intPXMax(bytO) =
intPXMax1

```

```

intPXMin(bytO) =
intPXMin1
intPYLMax1 = intPYLMax1
+ 2
intPYLMin1 = intPYLMin1
- conU
For bytP = conU To bytNP1
subRaioXY
dblRaio1 = dblRaio1 *
dblRaio1
For intPX = intPXMin1 To
intPXMax1
dblDX = sngX1 - intPX +
0.5
dblDX = dblRaio1 - dblDX
* dblDX
If dblDX >= conZ Then
dblDX = dblDX + (conU -
Sgn(dblDX)) * conIM
dblDX = Sqr(dblDX)
dblDY = sngY1 - dblDX
dblRY(conZ, bytP, intPX)
= dblDY
dblDY = sngY1 + dblDX
dblRY(conU, bytP, intPX)
= dblDY
' If Abs(dblRY(conZ, bytP,
intPX) - dblRY(conU, bytP,
intPX)) < conU Then Stop
Else
dblRY(conZ, bytP, intPX)
= conZ
dblRY(conU, bytP, intPX)
= conZ
End If
Next
For intPY = intPYLMin1
To intPYLMax1
dblDY = sngY1 - intPY +
0.5
dblDY = dblRaio1 - dblDY
* dblDY
If dblDY >= conZ Then
dblDY = dblDY + (conU -
Sgn(dblDY)) * conIM
dblDY = Sqr(dblDY)
dblDX = sngX1 - dblDY
dblRX(conZ, bytP, intPY)
= dblDX
dblDX = sngX1 + dblDY
dblRX(conU, bytP, intPY)
= dblDX
Else
dblRX(conZ, bytP, intPY)
= conZ

```

```

dblRX(conU, bytP, intPY)
= conZ
End If
Next
Next
intPXMax1 = intPXMax1 -
conU
For intPX = intPXMin1 To
intPXMax1
intPYMin1 =
intPYMin(bytO, intPX)
intPYMax1 =
intPYMax(bytO, intPX)
For intPY = intPYMin1 To
intPYMax1
For bytP = conU To
bytNP1
subRaioXY
Var = dblRaio1
intQed =
funNumZU(intPX, sngX1)
intQsi =
funNumZU(intPY, sngY1)
intQde = conU - intQed
intQis = conU - intQsi
sngDF1 = conZ
Select Case 8 *
funFD(intQed, intQsi) + 4 *
funFD(intQed, intQis) + 2 *
funFD(intQde, intQsi) +
funFD(intQde, intQis)
Case conZ
subCasV intQis, conU
subCasH intQde, conU
Case conU
subTS intQde, intQis,
conU
Case 3
subCoo intQde, intQis
dblCooXM =
Abs(dblRX(funNumZU(intCo
oX, sngX1), bytP, intCooY) -
sngCooXM)
subCooY intQsi
dblCooYM =
Abs(dblRX(funNumZU(intCo
oX, sngX1), bytP, intCooY) -
sngCooXM)
dblCooM =
Abs(dblCooXM - dblCooYM)
subDF1 0.5 *
Abs(dblCooXM +
dblCooYM) +
funSet(Sqr(dblCooM *
dblCooM + conU))
subCasH intQed, -conU

```

```

Case 5
  subCoo intQde, intQis
  dblCooXM =
Abs(dblRY(funNumZU(intCo
oY, sngY1), bytP, intCooX) -
sngCooYM)
  subCooX intQed
  dblCooYM =
Abs(dblRY(funNumZU(intCo
oY, sngY1), bytP, intCooX) -
sngCooYM)
  dblCooM =
Abs(dblCooXM - dblCooYM)
  subDF1 0.5 *
Abs(dblCooXM +
dblCooYM) +
funSet(Sqr(dblCooM *
dblCooM + conU))
'
  subCasV intQsi, -conU
Case 7
  sngDF1 = conU
  subTS intQed, intQsi, -
conU
Case Else
  sngDF1 = conU
End Select
If sngDF1 > conU + conU
/ conB Then MsgBox "Tela "
& intT & "(" & intPX & ", " &
intPY & ")=" & sngDF1,
vbOKOnly, "Casca"
  sngDF(bytP + conS) =
sngDF1 + (conU - sngDF1) *
Int(sngDF1)
  Next
  For bytF = conU To
bytNF1
  sngDFA =
sngDF(bytA(bytConj, bytF))
  sngDFB =
sngDF(bytB(bytConj, bytF))
  sngLogoE = sngDFA *
sngDFB
  sngLogo(conZ) =
sngLogoE
  sngLogoOu = sngDFA +
sngDFB - sngLogoE
  sngLogo(conU) =
sngLogoOu
  sngLogo(2) = sngDFA -
sngLogoE
  sngLogo(10) = sngDFA
  sngDF(bytD(bytConj,
bytF)) =
sngLogo(bytOp(bytConj,
bytF))
  Next
  sngObj(bytO, intPX,
intPY) =
sngDF(bytFacOb(bytO))
  Next
  Next
  sngTimer1 = Timer
  lblTimer.Caption =
conTimer & (sngTimer1 -
sngTimer) * (conO - bytO) \
(conO + conU) & " s."
  DoEvents
  Next
  lblTimer.Visible = False
  strCalc1 = strCalc
End If
End Function

Private Sub subOptVis(ByVal
TelaA As Boolean)

For bytO = conU To conW
  optVis0(bytO).Value =
TelaA
  optVis1(bytO).Value = Not
TelaA
Next
  optVis0(conZ).Value = TelaA
Or CBool(bytAce(intT) =
conZ)
  optVis1(conZ).Value = Not
optVis0(conZ).Value
  optVis0(conW).Value = Not
TelaA
  optVis1(conW).Value =
TelaA
  optVis0(conO).Value = True
  subVis conZ, False
End Sub

Private Sub subEAP(Hab As
Boolean)

  cmmExibir.Enabled = Hab
  cmmArq.Enabled = Hab
  cmmParar.Enabled = Hab
End Sub

Private Sub subFile(Nm As
String)

  strFile = strPathA & conBar
& Nm & conPto
End Sub

End Sub

Private Sub subOpen(Nome
As String)

  subFile Nome
  strFile = strFile & conTex
  Open strFile For Input As
#conU
End Sub

Private Sub subArq()

' If optA.Value Then strNome
= conSA Else strNome =
conSB
  If intT < 100 Then
    strNome = strNome & conSz
    If intT < 10 Then strNome =
strNome & conSz
  End If
  strNome = strNome & intT
  strPathA = strPathB &
conBar & conBMP & conBar
& conSCj & bytCJz
  subFile strNome
  strFile = strFile & conBMP
  SavePicture Picture1.Image,
strFile
End Sub

Private Sub subCap(ABC As
String)

  strNome = ABC
  Form1.Caption = conEst &
conBar & intT & ABC &
intNT
  DoEvents
End Sub

Private Sub subExibir()

  subEAP False
  If booNTU Then
' verifica mudanças que
justifiquem cálculo de tela
  subOV
  If funCalc Or CBool(intVis
<> intVis2) Then subCalTela
  Else MsgBox "Sem
modificações", vbInformation
End Sub

```

```

intVis2 = intVis
Else
If optA.Value Then
optB.Value = True
subOV
' subCap "B"
Elseif optB.Value Then
optC.Value = True
subOV
' subCap "C"
Else
subPre
optA.Value = True
subOV
funCalc
' subCap "A"
End If
subCalTela
End If
subEAP True

End Sub

Private Sub cmmArq_Click()

subArq

End Sub

Private Sub subExArq()

subExibir
subArq

End Sub

Private Sub
cmmArqT_Click()

If MsgBox("Tem certeza
(espera aproximada de " &
intNT & "min)?", vbYesNo) =
vbYes Then
byTState =
Form1.WindowState
Form1.WindowState =
vbMinimized
subArq
Do
subExArq
Loop Until intT = intNT
subExArq
subExArq
Form1.Caption = conEst
Form1.WindowState =
byTState

End If

End Sub

Private Sub
cmmExibir_Click()

subExibir
If ((intT = intNT) And
optC.Value) Or (intNT =
conU) Then intT = conZ

End Sub

Private Sub subCorRGB(CorR
As Single, CorG As Single,
CorB As Single)

sngCRGB(conU, bytCJz,
intD1) = CorR
sngCRGB(2, bytCJz, intD1)
= CorG
sngCRGB(3, bytCJz, intD1)
= CorB

End Sub

Private Sub
cmmPadrao_Click()

intVis2 = -conU
optTela.Value = False
optB.Value = False
optC.Value = False
opt1.Value = False
opt2.Value = False
opt3.Value = False
conRdGrau = 45 / Atn(conU)
conGrauRd = conU /
conRdGrau
strPathB = App.Path
strPathA = strPathB &
conBar & conData
subOpen conData
Input #conU, sngD1, sngD2,
intD1, intD2
sngR = sngD1
sngPro = sngD2
intX = intD1
intY = intD2
For bytO = conZ To conO
Input #conU, bytD1, sngD1,
bytD2, bytD3, sngD2, sngD3,
sngD4, sngD5, sngD6
bytNivel(bytO) = bytD1
bytAntiNivel(bytD1) = bytO

sngTam(bytO) = sngD1
bytCP(bytO) = bytD2
bytCjt(bytO) = bytD3
sngAng(bytO) = sngD2
sngDis(bytO) = sngD3
sngArAng(bytO) = sngD4
sngArCom(bytO) = sngD5
sngArLar(bytO) = sngD6
Next
Close #conU
intNT = conZ
subOpen conTel
Do While Not EOF(conU)
intNT = intNT + conU
Input #conU, bytD1, sngD1,
bytD2, bytD3, bytD4, sngD2,
sngD3, sngD4, sngD5
bytCj(intNT) = bytD1
sngAngulo(intNT) = sngD1
bytEsq(intNT) = bytD2
bytDir(intNT) = bytD3
bytAce(intNT) = bytD4
sngGiro(conZ, intNT) =
conZ
sngGiro(conU, intNT) =
sngD2
sngGiro(2, intNT) = sngD2
sngGiro(3, intNT) = sngD3
sngGiro(4, intNT) = sngD3
sngGiro(5, intNT) = conZ
sngGiro(conO, intNT) =
conZ
sngCro(conZ, intNT) = conU
sngCro(conU, intNT) = conU
sngCro(2, intNT) = sngD4
sngCro(3, intNT) = conU
sngCro(4, intNT) = sngD5
sngCro(5, intNT) = conU
sngCro(conO, intNT) = conU
Loop
Close #conU
strPath = strPathA
For bytCJz = conZ To conCj
For intD1 = conZ To conF
subCorRGB conU, conU,
conU
Next
strPathA = strPath & conBar
& conSCj & bytCJz
bytNF1 = conZ
subOpen conFac
Do While Not EOF(conU)
bytNF1 = bytNF1 + conU
Input #conU, bytD1, bytD2,
bytD3, bytD4

```

```

    bytD(bytCJz, bytNF1) =
bytD1
    bytA(bytCJz, bytNF1) =
bytD2
    bytOp(bytCJz, bytNF1) =
bytD3
    bytB(bytCJz, bytNF1) =
bytD4
    Loop
    Close #conU
    bytNF(bytCJz) = bytNF1
    bytNP1 = conZ
    subOpen conPar
    Do While Not EOF(conU)
    Input #conU, intD1, sngD1,
sngD2, sngD3
    If intD1 < conX Then
    bytNP1 = bytNP1 + conU
    intBola(bytCJz, bytNP1) =
intD1
    sngRAlfa(bytCJz, bytNP1)
= sngD1
    sngPD(bytCJz, bytNP1) =
sngD2
    sngPAlfa(bytCJz, bytNP1)
= sngD3
    Else
    intD1 = intD1 - conX
    subCorRGB sngD1, sngD2,
sngD3
    End If
    Loop
    Close #conU
    bytNP(bytCJz) = bytNP1
Next
bytCJz = conZ
' intT = conZ
txtRa.Text = sngR
txtX.Text = intX
txtY.Text = intY
txtPro.Text = sngPro
txtRa.Enabled = False
txtX.Enabled = False
txtY.Enabled = False
txtPro.Enabled = False
subConjunto False
txtAngulo.Enabled = False
subTela False
For bytO = conZ To conO
    booCjPar =
CBool(bytCP(bytO) = conZ)
' booCjPar = True
    optCP0(bytO).Value =
booCjPar
    optCP1(bytO).Value = Not
booCjPar

    txtNivel(bytO) =
bytNivel(bytO)
    txtTam(bytO) =
sngTam(bytO)
    txtAng(bytO) = conRdGrau *
sngAng(bytO)
    txtDis(bytO) = sngDis(bytO)
    booCjPar =
CBool(bytCjt(bytO) = conZ)
    optCj0(bytO).Value =
booCjPar
    optCj1(bytO).Value = Not
booCjPar
    subCP bytO, False
    subVis bytO, False
    txtNivel(bytO).Enabled =
False
    txtR(bytO).Enabled = False
    txtG(bytO).Enabled = False
    txtB(bytO).Enabled = False
    txtTam(bytO).Enabled =
False
    txtAng(bytO).Enabled =
False
    txtDis(bytO).Enabled = False
    subCj bytO, False
    subFace bytO, False
    optFace0(bytO).Enabled =
False
    Next
    optFace0(conW).Value =
True
    booNTU = CBool(intNT =
conU)
    frmConfig.Enabled =
booNTU
    optTela.Enabled = booNTU
    optFace.Enabled = booNTU
    optLivre.Enabled = booNTU
    cmmPadrao.Enabled =
booNTU
    optTela.Value = booNTU
    If booNTU Then subPre Else
    cmmArqT.Enabled = True
    optB.Value = booNTU
    optC.Value = Not booNTU
    intT = conZ
    subExibir

End Sub

Private Sub cmmParar_Click()

    cmmParar.Enabled = False
    cmmParar.Caption =
"Aguarde!"

intT = intNT
cmmParar.Caption = "Parar"

End Sub

Private Sub cmmSair_Click()

    Unload Form1

End Sub

Private Sub subOpt(Conjunto
As Byte)

    bytCJz = Conjunto
    For bytO = conZ To conO
    subTom CInt(bytO),
bytCjt(bytO) * bytCJz,
bytFacOb(bytO)
    Next

End Sub

Private Sub opt1_Click()

    subOpt conU

End Sub

Private Sub opt2_Click()

    subOpt 2

End Sub

Private Sub opt3_Click()

    subOpt 3

End Sub

Private Sub optA_Click()

    subOptVis True
    subCap conSA

End Sub

Private Sub optB_Click()

    subOptVis False
    subVis conZ, booNTU
    subCap conSB

End Sub

```

```

Private Sub optC_Click()
    subOptVis False
    optVis1(conO).Value = True
    subCap conSC
End Sub

Private Sub subTom(I As Integer, Conju As Byte, Face As Byte)
    ' subTom = (sngCorR(Conju, Face) + sngCorG(Conju, Face) + sngCorB(Conju, Face)) / 3
    For bytRGB1 = conU To 3
        sngTom(bytRGB1, I) = sngCRGB(bytRGB1, Conju, Face)
    Next
    txtR(I).Text = sngTom(conU, I)
    txtG(I).Text = sngTom(2, I)
    txtB(I).Text = sngTom(3, I)
End Sub

Private Sub subOptCj(Ind As Integer, BC As Byte)
    bytCjt(Ind) = BC
    subTom Ind, BC * bytCJz, bytFacOb(Ind)
End Sub

Private Sub optCj0_Click(Index As Integer)
    subOptCj Index, conZ
End Sub

Private Sub optCj1_Click(Index As Integer)
    subOptCj Index, conU
End Sub

Private Sub optCP0_Click(Index As Integer)
    bytCP(Index) = conZ
End Sub

Private Sub optCP1_Click(Index As Integer)
    bytCP(Index) = conU
End Sub

Private Sub subOptF(Inde As Integer, FO As Byte)
    bytFacOb(Inde) = FO
    subTom Inde, bytCjt(Inde) * bytCJz, FO
End Sub

Private Sub optFace0_Click(Index As Integer)
    subOptF Index, conZ
End Sub

Private Sub optFace1_Click(Index As Integer)
    subOptF Index, conU
    If funVinc Then
        If Index = conU Then
            optFace1(2).Value = True
            If Index = 3 Then
                optFace1(4).Value = True
            End If
        End Sub
    End Sub

Private Sub optFace2_Click(Index As Integer)
    subOptF Index, 2
    If funVinc Then
        If Index = conU Then
            optFace2(2).Value = True
            If Index = 3 Then
                optFace2(4).Value = True
            End If
        End Sub
    End Sub

Private Sub optFace3_Click(Index As Integer)
    subOptF Index, 3
    If funVinc Then
        If Index = conU Then
            optFace3(2).Value = True
            If Index = 3 Then
                optFace3(4).Value = True
            End If
        End Sub
    End Sub

Private Sub optFace4_Click(Index As Integer)
    subOptF Index, 4
    If funVinc Then
        If Index = conU Then
            optFace4(2).Value = True
            If Index = 3 Then
                optFace4(4).Value = True
            End If
        End Sub
    End Sub

Private Sub optFace5_Click(Index As Integer)
    subOptF Index, 5
    If funVinc Then
        If Index = conU Then
            optFace5(2).Value = True
            If Index = 3 Then
                optFace5(4).Value = True
            End If
        End Sub
    End Sub

Private Sub optFace6_Click(Index As Integer)
    subOptF Index, 6
    If funVinc Then
        If Index = conU Then
            optFace6(2).Value = True
            If Index = 3 Then
                optFace6(4).Value = True
            End If
        End Sub
    End Sub

Private Sub optFace2(4).Value = True
End If
End Sub

```

```

    If Index = 3 Then
    optFace6(4).Value = True
    End If

End Sub

Private Sub
optFace7_Click(Index As
Integer)

    subOptF Index, 7

End Sub

Private Sub
optFace8_Click(Index As
Integer)

    subOptF Index, 8

End Sub

Private Sub
optFace9_Click(Index As
Integer)

    subOptF Index, 9

End Sub

Private Sub optTela_Click()

    subConjunto True
    txtAngulo.Enabled = True
    subTela True
    For bytO = conZ To conO
    subFace bytO, True
    Next
    subFace 2, False
    subFace 4, False
    subFace 5, False
    subFOpt conZ, False
    For intO = 3 To -conU Step -
    2
    booOpts = CBool(intO <
conZ)
    bytO = (intO + Abs(intO)) \
    2
    optFace0(bytO).Enabled =
booOpts
    optFace7(bytO).Enabled =
booOpts
    optFace8(bytO).Enabled =
booOpts
    optFace9(bytO).Enabled =
booOpts

```

```

    Next
    subVis conZ, True
    optFace0(conZ).Enabled =
False

End Sub

Private Sub
txtAng_LostFocus(Index As
Integer)

    sngAng(Index) = conGrauRd
* txtAng(Index).Text

End Sub

Private Sub
txtCro_LostFocus(Index As
Integer)

    sngCro(Index, intT) =
txtCro(Index)

End Sub

Private Sub
txtGiro_LostFocus(Index As
Integer)

    sngGiro(Index, intT) =
conGrauRd * txtGiro(Index)

End Sub

Private Sub
txtAngulo_Change()

    If IsNumeric(txtAngulo.Text)
Then dblAngulo1 =
conGrauRd * txtAngulo.Text

End Sub

Private Sub
txtAngulo_LostFocus()

    If Not
IsNumeric(txtAngulo.Text)
Then txtAngulo.Text =
conRdGrau * dblAngulo1

End Sub

Private Sub
txtDis_LostFocus(Index As
Integer)

```

```

    sngDis(Index) =
txtDis(Index).Text

End Sub

Private Sub
txtNivel_LostFocus(Index As
Integer)

    bytNivel(Index) =
txtNivel(Index)

End Sub

Private Sub
txtPro_LostFocus()

    sngPro = txtPro.Text

End Sub

Private Sub
txtRa_LostFocus()

    sngR = txtRa.Text

End Sub

Private Sub
txtTam_LostFocus(Index As
Integer)

    sngTam(Index) =
txtTam(Index).Text

End Sub

Private Sub
txtR_LostFocus(Index As
Integer)

    sngTom(conU, Index) =
txtR(Index).Text

End Sub

Private Sub
txtG_LostFocus(Index As
Integer)

    sngTom(2, Index) =
txtG(Index).Text

End Sub

```

```
Private Sub  
txtB_LostFocus(Index As  
Integer)
```

```
    sngTom(3, Index) =  
    txtB(Index).Text
```

```
End Sub
```

```
Private Sub txtX_LostFocus()
```

```
    intX = txtX.Text
```

```
End Sub
```

```
Private Sub txtY_LostFocus()
```

```
    intY = txtY.Text
```

```
End Sub
```


Anexo 2. Data.

			3.141592654	0.3			3.141592654	0.2
			0.3				0.2	
		3	-1.2585296	3		3	-1.0174773	4
			4 0				3 0	
			3.141592654				3.141592654	
100000,0.5,255,255			3.141592654	0.2			3.141592654	0.3
5,19,0,0,0,0,0,0.8,0.8			0.2				0.3	
4,19,1,1,-			-0.6376980	3		3	2.3728106	3
1.57079632679489661923132		3	4 0				4 0	
16916398,133,0,4,4			3.141592654				3.141592654	
0,19,1,0,-			3.141592654	0.1			3.141592654	0.4
1.57079632679489661923132			0.1				0.4	
16916398,133,0,4,4			-1.8159985	4		3	2.6704525	4
3,19,1,1,1.5707963267948966		3	3 0				3 0	
192313216916398,133,0,4,4			3.141592654				3.141592654	
1,19,1,0,1.5707963267948966			3.141592654	0			3.141592654	0.5
192313216916398,133,0,4,4			0				0.5	
2,19,0,0,0,0,0,7.5,7.5			0.3049970	4		3	2.7760790	3
6,19,0,0,0,0,0,7.5,7.5		3	3 0				4 0	

Anexo 2. TelasG

			3.141592654				3.141592654	
			3.141592654	0			3.141592654	0.6
			0				0.6	
		3	-0.2595993	4		3	-0.6117926	4
			3 0				3 0	
3	-0.4752882	4	3.141592654				3.141592654	
	3 0		3.141592654	0			3.141592654	0.7
	3.141592654		0				0.7	
	3.141592654	0.9						
	0.9		-1.9132719	4		3	1.0840803	3
			3 0				4 0	
3	-0.1814761	3	3.141592654				3.141592654	
	4 0		3.141592654	0			3.141592654	0.8
	3.141592654		0				0.8	
	3.141592654	0.8						
	0.8		-2.8068368	3		3	1.8263444	3
			4 0				4 0	
3	2.4582167	4	3.141592654				3.141592654	
	3 0		3.141592654	0			3.141592654	0.9
	3.141592654		0				0.9	
	3.141592654	0.7						
	0.7		1.9396432	4		3	-2.5298020	3
			3 0				4 0	
3	-0.0521086	3	3.141592654				3.141592654	
	4 0		3.141592654	0			3.141592654	1
	3.141592654		0				1	
	3.141592654	0.6						
	0.6		-1.0525628	3		3	2.3875900	3
			4 0				4 0	
3	-2.1914057	4	3.141592654				3.141592654	
	3 0		3.141592654	0			3.141592654	1
	3.141592654		0				1	
	3.141592654	0.5						
	0.5		-0.0546438	4		3	2.8312311	3
			3 0				4 0	
3	1.0013471	4	3.141592654				3.141592654	
	3 0		3.141592654	0.1			3.141592654	1
	3.141592654		0.1				1	
	3.141592654	0.4						
	0.4		0.8681677	3		3	-0.8923430	4
			4 0				3 0	
3	-0.3469991	3	3.141592654				3.141592654	
	4 0							
	3.141592654							

	3.141592654	1		3.141592654	0		3.141592654	1
	1			1			0.8	
3	-0.5579437	4	3	0.1757368	4	3	-0.2328878	6
	3	0		6	0		4	0
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			0.2			1	
3	-0.8626865	3	3	-1.8503708	1	3	-1.9312225	4
	4	0		3	0		6	0
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	0		3.141592654	1
	1			1			1	
3	-0.2829322	4	3	-1.0933923	3	3	1.2160625	1
	3	0		1	0		3	0
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	0.6
	1			0			1	
3	-1.0377645	6	3	0.1692197	4	3	-2.8731516	6
	4	0		3	0		4	0
	3.141592654			3.141592654			3.141592654	
	3.141592654	0		3.141592654	1		3.141592654	1
	1			1			1	
3	1.0856593	4	3	-2.9061660	1	3	2.8650559	6
	3	0		3	0		4	0
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	0		3.141592654	1
	1			1			1	
3	-2.0533405	3	3	-1.3113207	3	3	1.0226444	1
	1	0		1	0		3	0
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	0.8
	0			0.2			1	
3	1.5006327	4	3	-1.2262578	6	3	-0.7061500	3
	6	0		4	0		1	0
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	0.4		3.141592654	1
	0			1			1	
3	-1.7244310	4	3	1.2567470	4	3	-3.0185548	3
	3	0		6	0		4	0
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			0.6			1	
3	-1.1244002	3	3	0.6906600	3	3	-2.6194666	3
	4	0		1	0		4	0
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			0.4			1	
3	1.2822712	4	3	0.9505399	3	3	1.5814740	4
	6	0		4	0		3	0
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	0			1			1	
3	-0.1921720	6	3	0.9319233	4	3	0.9496165	1
	4	0		6	0		3	0
	3.141592654			3.141592654			3.141592654	

	3.141592654	1		3.141592654	1		3.141592654	1
	1			0.142857143			1	
3	-0.6927701	4	3	2.8222735	1	3	-0.5937537	1
	3 0			2 0			2 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			0.285714286			1	
3	0.1309338	1	3	1.3963007	2	3	-2.8424318	2
	3 0			3 0			1 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654			3.141592654	1
	1			0.428571429	1		1	
3	0.1235645	3	3	-2.4282560	2	3	2.7672128	2
	4 0			1 0			3 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654			3.141592654	1
	1			0.571428571	1		1	
3	2.5565180	3	3	0.8490685	3	3	0.4926588	3
	1 0			2 0			4 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			0.714285714			1	
3	-1.1003286	1	3	-1.2574159	2	3	-1.7659654	2
	2 0			3 0			3 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654			3.141592654	1
	0			0.857142857	1		1	
3	2.2557455	2	3	1.7693021	3	3	1.1487539	1
	3 0			2 0			2 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	0		3.141592654	1		3.141592654	1
	1			1			1	
3	-1.2946679	3	3	-2.9238924	2	3	-1.7966678	2
	2 0			3 0			3 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	0			1			1	
3	2.2976932	1	3	1.4993480	3	3	1.6773271	2
	2 0			4 0			3 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	0			1			1	
3	-2.7213543	3	3	1.0378905	2	3	1.4089266	3
	2 0			1 0			2 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	0			1			1	
3	-2.9348902	3	3	-1.3610298	2	3	0.1536394	2
	2 0			3 0			1 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	0			1			1	
3	2.9766377	3	3	1.9977343	1	3	-1.0107923	1
	2 0			2 0			2 0	
	3.141592654			3.141592654			3.141592654	

	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			0	
3	2.9363508	1	3	-0.7358107	4	3	0.4408625	6
	2 0			3 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			0.142857143	
3	0.4685708	2	3	-0.1996959	1	3	-1.1598976	4
	1 0			2 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			0.285714286	
3	2.0797741	2	3	-2.5376445	3	3	-1.7444250	6
	1 0			2 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			0.428571429	
3	-2.4129567	4	3	2.2424023	3	3	0.0127077	5
	3 0			2 0			4 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			0.571428571	1
3	-0.0629643	3	3	-0.2636117	5	3	0.9891633	5
	2 0			6 0			4 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	0		3.141592654	
	1			1			0.714285714	1
3	-0.5621274	2	3	-0.6649810	5	3	0.2848274	6
	1 0			6 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	0		3.141592654	1
	1			1			0.857142857	
3	-1.2131293	1	3	-0.7358107	6	3	-0.3267116	6
	2 0			5 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			0			1	
3	-0.1542142	2	3	-0.7182596	2	3	2.9540177	6
	1 0			3 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-0.6267599	2	3	-0.8534708	5	3	0.4348145	4
	1 0			4 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	0		3.141592654	1
	1			1			1	
3	-0.5911661	2	3	-1.8084770	6	3	2.0735283	4
	1 0			5 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			0			1	
3	3.1233675	2	3	-1.8966877	4	3	1.1607091	5
	3 0			5 0			6 0	
	3.141592654			3.141592654			3.141592654	

	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	2.8402134	3	3	-3.0837844	5	3	1.0695618	5
	4 0			4 0			6 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	0.0080949	4	3	0.4254235	5	3	-0.7489806	1
	5 0			6 0			2 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	2.0961919	5	3	-1.8491394	5	3	1.4466116	4
	4 0			4 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	1.2838508	5	3	2.2307781	5	3	1.9854405	3
	6 0			6 0			4 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-0.5427919	2	3	1.0941713	4	3	-2.7960500	6
	1 0			5 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.5037672	4	3	1.9234764	5	3	1.6455815	5
	5 0			4 0			6 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-3.0142713	5	3	2.1563530	4	3	-1.0506651	4
	4 0			5 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.6686729	3	3	2.9369745	3	3	0.3322607	2
	2 0			2 0			3 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.7074241	5	3	-1.9695370	6	3	0.9998550	1
	6 0			5 0			2 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-1.7487009	5	3	0.9696498	2	3	-1.2207154	5
	6 0			1 0			4 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-1.8511752	5	3	2.2181600	6	3	-0.9254364	4
	4 0			5 0			5 0	
	3.141592654			3.141592654			3.141592654	

	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.8818495	5	3	-0.7242598	6	3	-1.7058369	6
	4			3			1	
	0			0			0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	0.4663582	6	3	1.4971983	4	3	-1.2469053	4
	5			6			1	
	0			0			0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.6806745	4	3	-0.9827765	1	3	-2.8850236	1
	5			6			6	
	0			0			0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.2794350	5	3	1.5457500	3	3	-2.7663950	1
	6			1			6	
	0			0			0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	2.5999351	5	3	1.7527088	6	3	1.7897057	3
	6			3			4	
	0			0			0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	1.3882973	4	3	-0.1351013	6	3	0.2503835	6
	3			4			4	
	0			0			0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.9657260	4	3	1.0682676	6	3	0.7699146	3
	3			4			1	
	0			0			0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-1.2898675	5	3	-2.8037882	3	3	1.3987943	4
	4			1			3	
	0			0			0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-1.7467072	6	3	-1.7804356	1	3	-0.7358107	6
	5			4			3	
	0			0			0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	1.1180632	6	3	-0.3162854	6	3	-1.8606546	3
	1			1			1	
	0			0			0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-1.4385909	3	3	1.4311744	4	3	0.3413802	4
	6			6			6	
	0			0			0	
	3.141592654			3.141592654			3.141592654	

	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-1.6062496	3	3	0.5662828	6	3	-0.9620098	6
	6 0			4 0			4 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.8541290	4	3	2.1651129	4	3	2.6587854	6
	1 0			3 0			3 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	1.5964719	6	3	1.4064262	1	3	3.0053243	4
	1 0			6 0			1 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	0.9102107	1	3	2.1720514	4	3	-2.4724898	3
	4 0			6 0			6 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	1.3057988	1	3	-0.4654296	1	3	2.8298246	3
	6 0			6 0			4 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	0.9377360	1	3	-1.1810184	4	3	-1.0172918	1
	4 0			1 0			4 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-1.7440036	1	3	-1.1984282	1	3	2.5061714	6
	3 0			4 0			1 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.3247549	3	3	1.9628589	1	3	1.6308601	6
	4 0			3 0			1 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.6649255	3	3	2.9473446	3	3	-1.7869213	1
	4 0			6 0			3 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	2.6026033	4	3	2.4906733	1	3	0.8038668	1
	3 0			3 0			3 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	1.7743025	1	3	-1.7817871	4	3	0.4524826	4
	6 0			1 0			1 0	
	3.141592654			3.141592654			3.141592654	

	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-0.0290974	6	3	-2.4399900	3	3	0.5074478	6
	1 0			6 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-1.6946334	3	3	0.5642562	4	3	-0.7453183	2
	6 0			3 0			1 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	2.2549576	4	3	2.1619420	4	3	-2.0078745	4
	3 0			3 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.9432567	3	3	-2.4374297	3	3	2.7091292	2
	4 0			2 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.2957315	4	3	-1.1491489	2	3	0.5417681	4
	1 0			3 0			2 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-3.0825544	4	3	-2.8975615	5	3	-1.6088535	3
	6 0			4 0			5 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-0.6576618	3	3	0.1916288	3	3	-0.4595570	4
	1 0			4 0			2 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-1.7577582	1	3	0.0566668	3	3	1.7494465	1
	4 0			2 0			2 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	0.2163953	6	3	-0.3470227	5	3	0.1252245	5
	3 0			6 0			4 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	1.0555678	6	3	-1.2953604	2	3	2.9886741	5
	3 0			4 0			3 0	
	3.141592654			3.141592654			3.141592654	
	3.141592654	1		3.141592654	1		3.141592654	1
	1			1			1	
3	-2.5517211	3	3	1.0375267	1	3	1.8810875	5
	6 0			2 0			6 0	
	3.141592654			3.141592654			3.141592654	

	3.141592654	1	3	-2.6504282	2		3.141592654	
	1			4	0		3.141592654	1
3	1.9875847	5		3.141592654			1	
	4	0		3.141592654	1	3	-0.7358107	6
	3.141592654			1			5	0
	3.141592654	1	3	2.8769079	1		3.141592654	
	1			2	0		3.141592654	1
3	-0.8056154	4		3.141592654			1	
	3	0		3.141592654	1	3	-2.5764074	5
	3.141592654			1			6	0
	3.141592654	1	3	0.3725245	5		3.141592654	
	1			2	0		3.141592654	1
3	-1.3412158	2		3.141592654			1	
	1	0		3.141592654	1	3	2.3993134	5
	3.141592654			1			3	0
	3.141592654	1	3	3.0769215	2		3.141592654	
	1			5	0		3.141592654	1
3	0.1740936	3		3.141592654			1	
	2	0		3.141592654	1			
	3.141592654			1				
	3.141592654	1	3	3.0399173	5			
	1			3	0			
3	-0.0437216	2		3.141592654				
	5	0		3.141592654	1			
	3.141592654			1				
	3.141592654	1	3	2.2666353	4			
	1			5	0			
3	0.6274037	5		3.141592654				
	2	0		3.141592654	1			
	3.141592654			1				
	3.141592654	1	3	0.1369672	3			
	1			4	0	3	0.7064981	4
3	-1.9456219	3		3.141592654			3	0
	4	0		3.141592654	1		3.141592654	
	3.141592654			1			3.141592654	1
	3.141592654	1	3	0.1918154	3		1	
	1			5	0	3	0.7064981	3
3	1.2298951	2		3.141592654			4	0
	3	0		3.141592654	1		3.141592654	
	3.141592654			1			3.141592654	1
	3.141592654	1	3	-1.6417734	4		1	
	1			2	0			

Anexo 2. TelasP

	0.7064981	4
	3	0
	3.141592654	
	3.141592654	1
	1	
	0.7064981	3
	4	0
	3.141592654	
	3.141592654	1
	1	