

Forum:[Hack Forums](#)

Thread: [Lockscreen with Keypad | Source](#)

CottleSilver - August 11, 2014, 6:41 am

This is my first post and release here and also the first project of this kind. This is just a lockscreen with keypad, the idea I took from rainmeter, there you can't do a keyboard hook or close a process, and this blocks all keys except ctrl+alt+delete and winkey+I but it dose close task manager. This is how it looks like:

Spoiler [\(Click to View\)](#)

Keep in mind that all pictures can be changed from the external folder with images.

Credits: Me and DarthDestroyer

I am far from an expert with vb.net, so here is the source with binary: [Download Link](#) (File updated v1.1)

Update:

v1.0

-initial release

v1.1

-optimized timer for closing processes (credits: DarthDestroyer)

-added keyboard numbers exception (credits: DarthDestroyer)

-other updates

-possible last version, check page 2-3 for any update

v1.2

version not working properly because of an error in button gui

-gui buttons added (credits: DarthDestroyer)

-other small changes (font, time format, etc...)

Special thx to DarthDestroyer

Jokeh - August 11, 2014, 6:42 am

Would look better centered bro, other than that, good job

CottleSilver - August 11, 2014, 6:45 am

It is centered, fullscreen (I recorded like that) and It adds a form, fullscreen too, on any other display with just the time, oh and the encryption for the code is made by me, I know is not much of an encryption but I try

Aur3xis - August 11, 2014, 6:45 am

Nice share bro! Will use it on my notebook! :)

Only3 - August 11, 2014, 6:54 am

the idea is nice but why images u can do it with gdi

DarthDestroyer - August 11, 2014, 8:08 am

This is a funny Way to optain, what you want... ;)

I have two fast suggestions for you...

1. You forgot to set the FrmScreen DoubleBuffered Option ;)
2. You could optimize the CloseTimer a Lot! It's not necessary that you loop (for each) multiple times through the ProcessList...

And... then... A Hint: Why not allow the Numbers from the Keyboard? ;)

I like it! Good Job!

CottleSilver - August 11, 2014, 8:34 am

(Aug 11, 2014 06:54 AM)Only3 Wrote: [\[CITE\]](#) the idea is nice but why images u can do it with gdi I know that its posible but I don't know exacly how so I did the way I did it, but if you want to do it I will be happy to add it and give you the credit

(Aug 11, 2014 08:08 AM)DarthDestroyer Wrote: [\[CITE\]](#) This is a funny Way to optain, what you want... ;)

I have two fast suggestions for you...

1. You forgot to set the FrmScreen DoubleBuffered Option ;)
2. You could optimize the CloseTimer a Lot! It's not necessary that you loop (for each) multiple times through the ProcessList...

And... then... A Hint: Why not allow the Numbers from the Keyboard? ;)

I like it! Good Job!

Yea I know its not the best way to do it but if you got a better way and want to add it I will be happy to use it, give you the credit of course
Oh and it was easier to block everything

DarthDestroyer - August 12, 2014, 1:45 am

OK... here we go...

TimerClose Example:

```
Code Private Sub TimerClose_Tick(sender As Object, e As EventArgs) Handles TimerClose.Tick
    Try
        For Each prog As Process In Process.GetProcesses
            Select Case prog.ProcessName
                Case "Taskmgr"
                    prog.Kill()
                Case "taskmgr"
                    prog.Kill()
                Case "cmd"
                    prog.Kill()
                Case "bsplayer"
                    prog.Kill()
                Case "vlc"
                    prog.Kill()
                Case "chrome"
                    prog.Kill()
                Case "firefox"
```

```

        prog.Kill()
        Case "iexplore"
        prog.Kill()
        Case "opera"
        prog.Kill()
    End Select
Next
Catch ex As Exception
End Try
End Sub
KeyboardHook_KeyUp Example:

```

```

Code Private Sub AddKeyToLabelCode(ByVal input As String)
    If Not LabelCode.Text.Length > 9 Then
        LabelCode.Text &= input
    End If
End Sub

```

```

Private Sub KeyboardHook_KeyUp(sender As Object, e As KeyEventArgs) Handles
KeyboardHook.KeyUp
    Select Case e.KeyData
    Case Keys.D1
        AddKeyToLabelCode("1")
    Case Keys.NumPad1
        AddKeyToLabelCode("1")
    Case Keys.D2
        AddKeyToLabelCode("2")
    Case Keys.NumPad2
        AddKeyToLabelCode("2")
    Case Keys.D3
        AddKeyToLabelCode("3")
    Case Keys.NumPad3
        AddKeyToLabelCode("3")
    Case Keys.D4
        AddKeyToLabelCode("4")
    Case Keys.NumPad4
        AddKeyToLabelCode("4")
    Case Keys.D5
        AddKeyToLabelCode("5")
    Case Keys.NumPad5
        AddKeyToLabelCode("5")
    Case Keys.D6
        AddKeyToLabelCode("6")
    Case Keys.NumPad6
        AddKeyToLabelCode("6")
    Case Keys.D7
        AddKeyToLabelCode("7")
    Case Keys.NumPad7
        AddKeyToLabelCode("7")
    Case Keys.D8
        AddKeyToLabelCode("8")
    Case Keys.NumPad8
        AddKeyToLabelCode("8")
    
```

```

Case Keys.D9
    AddKeyToLabelCode("9")
Case Keys.NumPad9
    AddKeyToLabelCode("9")
Case Keys.Enter
    PerformCheck()
Case Else
    e.Handled = True
End Select
End Sub
AND... because i had some time... a little Gift:

```

Your ImageThingie as a GDI Button:

```

Code Public Class PiCButton
    Inherits Control

    Enum MouseState As Byte
        None = 0
        Over = 1
        Down = 2
        Block = 3
    End Enum

    Private _BorderColour As Color = Color.FromArgb(255, 255, 255)
    Private _FontColour As Color = Color.FromArgb(255, 255, 255)
    Private _FontHoverColour As Color = Color.FromArgb(255, 255, 255)
    Private _FontClickColour As Color = Color.FromArgb(255, 255, 255)
    Private _MainColour As Color = Color.Transparent
    Private _HoverColour As Color = Color.FromArgb(85, 255, 255, 255)
    Private _PressedColour As Color = Color.FromArgb(85, 0, 0, 0)
    Private _OffSetLeft As Integer = 0
    Private _OffSetTop As Integer = 0
    Private State As New MouseState

    Protected Overrides Sub OnMouseDown(e As MouseEventArgs)
        MyBase.OnMouseDown(e)
        State = MouseState.Down : Invalidate()
    End Sub
    Protected Overrides Sub OnMouseUp(e As MouseEventArgs)
        MyBase.OnMouseUp(e)
        State = MouseState.Over : Invalidate()
    End Sub
    Protected Overrides Sub OnMouseEnter(e As EventArgs)
        MyBase.OnMouseEnter(e)
        State = MouseState.Over : Invalidate()
    End Sub
    Protected Overrides Sub OnMouseLeave(e As EventArgs)
        MyBase.OnMouseLeave(e)
        State = MouseState.None : Invalidate()
    End Sub

    <Category("Colours")>

```

```
Public Property BorderColour As Color
    Get
        Return _BorderColour
    End Get
    Set(value As Color)
        _BorderColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property FontColour As Color
    Get
        Return _FontColour
    End Get
    Set(value As Color)
        _FontColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property FontHoverColour As Color
    Get
        Return _FontHoverColour
    End Get
    Set(value As Color)
        _FontHoverColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property FontClickColour As Color
    Get
        Return _FontClickColour
    End Get
    Set(value As Color)
        _FontClickColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property BaseColour As Color
    Get
        Return _MainColour
    End Get
    Set(value As Color)
        _MainColour = value
        Invalidate()
    End Set
End Property
```

```

<Category("Colours")>
Public Property HoverColour As Color
    Get
        Return _HoverColour
    End Get
    Set(value As Color)
        _HoverColour = value
        Invalidate()
    End Set
End Property

```

```

<Category("Colours")>
Public Property PressedColour As Color
    Get
        Return _PressedColour
    End Get
    Set(value As Color)
        _PressedColour = value
        Invalidate()
    End Set
End Property

```

```

<Category("OffSet")>
Public Property OffSetLeft As Integer
    Get
        Return _OffSetLeft
    End Get
    Set(value As Integer)
        _OffSetLeft = value
        Invalidate()
    End Set
End Property

```

```

<Category("OffSet")>
Public Property OffSetTop As Integer
    Get
        Return _OffSetTop
    End Get
    Set(value As Integer)
        _OffSetTop = value
        Invalidate()
    End Set
End Property

```

```

Sub New()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or _
        ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer Or _
        ControlStyles.SupportsTransparentBackColor, True)
    DoubleBuffered = True
    Font = New Font("Arial", 30)
    Size = New Size(50, 50)
    MaximumSize = New Size(50, 50)

```

```
MinimumSize = New Size(50, 50)
BackColor = _MainColour
End Sub
```

```
Protected Overrides Sub OnPaint(e As PaintEventArgs)
    Dim G = e.Graphics
    Dim FontBrush As Brush = Brushes.White
    With G
        .TextRenderingHint = TextRenderingHint.ClearTypeGridFit
        .SmoothingMode = SmoothingMode.HighQuality
        .PixelOffsetMode = PixelOffsetMode.HighQuality
        .InterpolationMode = CType(7, InterpolationMode)
        Select Case State
            Case MouseState.None
                FontBrush = Brushes.White
                .DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))
            Case MouseState.Over
                FontBrush = Brushes.White
                .FillEllipse(New SolidBrush(_HoverColour), New Rectangle(2, 2, Width - 4, Height - 4))
                .DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))
            Case MouseState.Down
                FontBrush = Brushes.White
                .FillEllipse(New SolidBrush(_PressedColour), New Rectangle(2, 2, Width - 4, Height - 4))
                .DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))
        End Select
        .DrawString(Text, Font, FontBrush, New Point(CInt((Width - 4) / 2) + _OffsetLeft, CInt((Height + 4) / 2) + _OffsetTop), New StringFormat With {.Alignment = StringAlignment.Center, .LineAlignment = StringAlignment.Center})
    End With
End Sub
```

End Class

It should explain itself...

For your Reset and Confirm Button, user "Webdings"-Font ;)

I had remade your entire Code... i would post it, if you want ?

Cornflower Blue ???-?— - August 12, 2014, 1:55 am

(Aug 12, 2014 01:45 AM)DarthDestroyer Wrote: [\[CITE\]](#) SNIPPED :D

Awesome HQ post. You seem to be a good coder, you should release some of your stuff. I would love to see it.

Enjoy the green apple.

SaketSaket - August 12, 2014, 2:52 am

Nice work ;) Keep up the good work ;)

DarthDestroyer - August 12, 2014, 3:11 am

Thank you, Mate!

I'm happy, that you like it... ?

I can't Release something atm. I code for a Company only, atm.
These Solutions require an Active Directory or a SQL Server...

The only thing, that i could Release is an SEPA-XML Reader without Dependencies in German Language... I dont know, if it would be usefull for you ?

BTT: I have uploaded the changed Files!

<http://www.mediafire.com/download/voueg35gpq755gb/LockscreenNumpad.rar>

In this Archive are only the Files for the Main-From! Nothing else!
I hadn't spend a lot of time in the functionality, but it "should" work ?

CottleSilver - August 12, 2014, 4:10 am

(Aug 12, 2014 01:45 AM)DarthDestroyer Wrote: [\[CITE\]](#) OK... here we go...

TimerClose Example:

```
Code Private Sub TimerClose_Tick(sender As Object, e As EventArgs) Handles TimerClose.Tick
    Try
        For Each prog As Process In Process.GetProcesses
            Select Case prog.ProcessName
                Case "Taskmgr"
                    prog.Kill()
                Case "taskmgr"
                    prog.Kill()
                Case "cmd"
                    prog.Kill()
                Case "bsplayer"
                    prog.Kill()
                Case "vlc"
                    prog.Kill()
                Case "chrome"
                    prog.Kill()
                Case "firefox"
                    prog.Kill()
                Case "iexplore"
                    prog.Kill()
                Case "opera"
                    prog.Kill()
            End Select
        Next
        Catch ex As Exception
        End Try
    End Sub
```

KeyboardHook_KeyUp Example:

```
Code Private Sub AddKeyToLabelCode(ByVal input As String)
    If Not LabelCode.Text.Length > 9 Then
        LabelCode.Text &= input
    End If
End Sub
```

Private Sub KeyboardHook_KeyUp(sender As Object, e As KeyEventArgs) Handles

```

KeyboardHook.KeyUp
    Select Case e.KeyData
    Case Keys.D1
    AddKeyToLabelCode("1")
    Case Keys.NumPad1
    AddKeyToLabelCode("1")
    Case Keys.D2
    AddKeyToLabelCode("2")
    Case Keys.NumPad2
    AddKeyToLabelCode("2")
    Case Keys.D3
    AddKeyToLabelCode("3")
    Case Keys.NumPad3
    AddKeyToLabelCode("3")
    Case Keys.D4
    AddKeyToLabelCode("4")
    Case Keys.NumPad4
    AddKeyToLabelCode("4")
    Case Keys.D5
    AddKeyToLabelCode("5")
    Case Keys.NumPad5
    AddKeyToLabelCode("5")
    Case Keys.D6
    AddKeyToLabelCode("6")
    Case Keys.NumPad6
    AddKeyToLabelCode("6")
    Case Keys.D7
    AddKeyToLabelCode("7")
    Case Keys.NumPad7
    AddKeyToLabelCode("7")
    Case Keys.D8
    AddKeyToLabelCode("8")
    Case Keys.NumPad8
    AddKeyToLabelCode("8")
    Case Keys.D9
    AddKeyToLabelCode("9")
    Case Keys.NumPad9
    AddKeyToLabelCode("9")
    Case Keys.Enter
    PerformCheck()
    Case Else
    e.Handled = True
    End Select
End Sub

```

AND... because i had some time... a little Gift:

Your ImageThingie as a GDI Button:

```

Code Public Class PiCButton
    Inherits Control

    Enum MouseState As Byte
    None = 0

```

```
Over = 1
Down = 2
Block = 3
End Enum
```

```
Private _BorderColour As Color = Color.FromArgb(255, 255, 255)
Private _FontColour As Color = Color.FromArgb(255, 255, 255)
Private _FontHoverColour As Color = Color.FromArgb(255, 255, 255)
Private _FontClickColour As Color = Color.FromArgb(255, 255, 255)
Private _MainColour As Color = Color.Transparent
Private _HoverColour As Color = Color.FromArgb(85, 255, 255, 255)
Private _PressedColour As Color = Color.FromArgb(85, 0, 0, 0)
Private _OffsetLeft As Integer = 0
Private _OffsetTop As Integer = 0
Private State As New MouseState
```

```
Protected Overrides Sub OnMouseDown(e As MouseEventArgs)
    MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
End Sub
```

```
Protected Overrides Sub OnMouseUp(e As MouseEventArgs)
    MyBase.OnMouseUp(e)
    State = MouseState.Over : Invalidate()
End Sub
```

```
Protected Overrides Sub OnMouseEnter(e As EventArgs)
    MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
End Sub
```

```
Protected Overrides Sub OnMouseLeave(e As EventArgs)
    MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
End Sub
```

```
<Category("Colours")>
Public Property BorderColour As Color
    Get
        Return _BorderColour
    End Get
    Set(value As Color)
        _BorderColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property FontColour As Color
    Get
        Return _FontColour
    End Get
    Set(value As Color)
        _FontColour = value
        Invalidate()
    End Set
```

End Property

```
<Category("Colours")>
Public Property FontHoverColour As Color
    Get
        Return _FontHoverColour
    End Get
    Set(value As Color)
        _FontHoverColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property FontClickColour As Color
    Get
        Return _FontClickColour
    End Get
    Set(value As Color)
        _FontClickColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property BaseColour As Color
    Get
        Return _MainColour
    End Get
    Set(value As Color)
        _MainColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property HoverColour As Color
    Get
        Return _HoverColour
    End Get
    Set(value As Color)
        _HoverColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property PressedColour As Color
    Get
        Return _PressedColour
    End Get
    Set(value As Color)
        _PressedColour = value
    End Set
End Property
```

```
Invalidate()  
End Set  
End Property
```

```
<Category("OffSet")>  
Public Property OffSetLeft As Integer  
Get  
Return _OffsetLeft  
End Get  
Set(value As Integer)  
_OffsetLeft = value  
Invalidate()  
End Set  
End Property
```

```
<Category("OffSet")>  
Public Property OffSetTop As Integer  
Get  
Return _OffsetTop  
End Get  
Set(value As Integer)  
_OffsetTop = value  
Invalidate()  
End Set  
End Property
```

```
Sub New()  
SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or _  
ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer Or _  
ControlStyles.SupportsTransparentBackColor, True)  
DoubleBuffered = True  
Font = New Font("Arial", 30)  
Size = New Size(50, 50)  
MaximumSize = New Size(50, 50)  
MinimumSize = New Size(50, 50)  
BackColor = _MainColour  
End Sub
```

```
Protected Overrides Sub OnPaint(e As PaintEventArgs)  
Dim G = e.Graphics  
Dim FontBrush As Brush = Brushes.White  
With G  
.TextRenderingHint = TextRenderingHint.ClearTypeGridFit  
.SmoothingMode = SmoothingMode.HighQuality  
.PixelOffsetMode = PixelOffsetMode.HighQuality  
.InterpolationMode = CType(7, InterpolationMode)  
Select Case State  
Case MouseState.None  
FontBrush = Brushes.White  
.DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))  
Case MouseState.Over  
FontBrush = Brushes.White  
.FillEllipse(New SolidBrush(_HoverColour), New Rectangle(2, 2, Width - 4, Height - 4))
```

```

.DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))
Case MouseState.Down
FontBrush = Brushes.White
.FillEllipse(New SolidBrush(_PressedColour), New Rectangle(2, 2, Width - 4, Height - 4))
.DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))
End Select
.DrawString(Text, Font, FontBrush, New Point(CInt((Width - 4) / 2) + _OffsetLeft, CInt((Height +
4) / 2) + _OffsetTop), New StringFormat With {.Alignment = StringAlignment.Center, .LineAlignment
= StringAlignment.Center})
End With
End Sub

```

End Class

It should explain itself...

For your Reset and Confirm Button, user "Webdings"-Font ;)

I had remade your entire Code... i would post it, if you want ?
Yea sure post it, I like to see what else you done with the code

New version added with small changes

DarthDestroyer - August 12, 2014, 5:15 am

(Aug 12, 2014 04:10 AM)CottleSilver Wrote: [\[CITE\]](#) Yea sure post it, I like to see what else you done with the code

New version added with small changes

<http://www.mediafire.com/download/voueg35gpq755gb/LockscreenNumpad.rar> ?

I had started with your idea...

You could use ONE of these Btn-Images and draw the Text on it with the Btn_Paint Event...
Then, the User has to change only one Image...

The Idea with these GDI Buttons is, that they have Alpha-Enabled Colors. That means, that they are transparent. The User can choose any Background, but dont need to change the whole Buttons. I had done that too, with the "one-image"-idea...

<http://www.mediafire.com/view/d9ug2err5obuyxq/Btn.png#>

Another Idea is, you could use a Config-File (for User-Edits) where you can select, which form the button should have... Its Easy to implement more forms for the buttons! Rectangle, Ellipse, Polygon... Whatever you want! It would be easier for Users to change a line of text instead of drawing hours. ?

Or... you could implement BOTH Versions ;)

OffCode - August 12, 2014, 5:56 am

Nice work man! Keep it up! And thanks for source! :)

CottleSilver - August 12, 2014, 6:00 am

The last source has an error at the button gui, it will be updated in the next half hour

App downgraded to v1.1, posible fix for v1.2 will come soon

Only3 - August 12, 2014, 6:38 am

(Aug 12, 2014 01:45 AM)DarthDestroyer Wrote: [\[CITE\]](#) OK... here we go...

TimerClose Example:

Code Private Sub TimerClose_Tick(sender As Object, e As EventArgs) Handles TimerClose.Tick

```
Try
For Each prog As Process In Process.GetProcesses
Select Case prog.ProcessName
Case "Taskmgr"
prog.Kill()
Case "taskmgr"
prog.Kill()
Case "cmd"
prog.Kill()
Case "bsplayer"
prog.Kill()
Case "vlc"
prog.Kill()
Case "chrome"
prog.Kill()
Case "firefox"
prog.Kill()
Case "iexplore"
prog.Kill()
Case "opera"
prog.Kill()
End Select
Next
Catch ex As Exception
End Try
End Sub
```

KeyboardHook_KeyUp Example:

Code Private Sub AddKeyToLabelCode(ByVal input As String)

```
If Not LabelCode.Text.Length > 9 Then
LabelCode.Text &= input
End If
End Sub
```

Private Sub KeyboardHook_KeyUp(sender As Object, e As KeyEventArgs) Handles

```
KeyboardHook.KeyUp
Select Case e.KeyData
Case Keys.D1
AddKeyToLabelCode("1")
Case Keys.NumPad1
AddKeyToLabelCode("1")
Case Keys.D2
AddKeyToLabelCode("2")
Case Keys.NumPad2
AddKeyToLabelCode("2")
Case Keys.D3
AddKeyToLabelCode("3")
```

```

Case Keys.NumPad3
AddKeyToLabelCode("3")
Case Keys.D4
AddKeyToLabelCode("4")
Case Keys.NumPad4
AddKeyToLabelCode("4")
Case Keys.D5
AddKeyToLabelCode("5")
Case Keys.NumPad5
AddKeyToLabelCode("5")
Case Keys.D6
AddKeyToLabelCode("6")
Case Keys.NumPad6
AddKeyToLabelCode("6")
Case Keys.D7
AddKeyToLabelCode("7")
Case Keys.NumPad7
AddKeyToLabelCode("7")
Case Keys.D8
AddKeyToLabelCode("8")
Case Keys.NumPad8
AddKeyToLabelCode("8")
Case Keys.D9
AddKeyToLabelCode("9")
Case Keys.NumPad9
AddKeyToLabelCode("9")
Case Keys.Enter
PerformCheck()
Case Else
e.Handled = True
End Select
End Sub

```

AND... because i had some time... a little Gift:

Your ImageThingie as a GDI Button:

```

Code Public Class PiCButton
Inherits Control

```

```

Enum MouseState As Byte
None = 0
Over = 1
Down = 2
Block = 3
End Enum

```

```

Private _BorderColour As Color = Color.FromArgb(255, 255, 255)
Private _FontColour As Color = Color.FromArgb(255, 255, 255)
Private _FontHoverColour As Color = Color.FromArgb(255, 255, 255)
Private _FontClickColour As Color = Color.FromArgb(255, 255, 255)
Private _MainColour As Color = Color.Transparent
Private _HoverColour As Color = Color.FromArgb(85, 255, 255, 255)
Private _PressedColour As Color = Color.FromArgb(85, 0, 0, 0)

```

```
Private _OffsetLeft As Integer = 0
Private _OffsetTop As Integer = 0
Private State As New MouseState
```

```
Protected Overrides Sub OnMouseDown(e As MouseEventArgs)
    MyBase.OnMouseDown(e)
    State = MouseState.Down : Invalidate()
End Sub
```

```
Protected Overrides Sub OnMouseUp(e As MouseEventArgs)
    MyBase.OnMouseUp(e)
    State = MouseState.Over : Invalidate()
End Sub
```

```
Protected Overrides Sub OnMouseEnter(e As EventArgs)
    MyBase.OnMouseEnter(e)
    State = MouseState.Over : Invalidate()
End Sub
```

```
Protected Overrides Sub OnMouseLeave(e As EventArgs)
    MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
End Sub
```

```
<Category("Colours")>
Public Property BorderColour As Color
    Get
        Return _BorderColour
    End Get
    Set(value As Color)
        _BorderColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property FontColour As Color
    Get
        Return _FontColour
    End Get
    Set(value As Color)
        _FontColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property FontHoverColour As Color
    Get
        Return _FontHoverColour
    End Get
    Set(value As Color)
        _FontHoverColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property FontClickColour As Color
    Get
        Return _FontClickColour
    End Get
    Set(value As Color)
        _FontClickColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property BaseColour As Color
    Get
        Return _MainColour
    End Get
    Set(value As Color)
        _MainColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property HoverColour As Color
    Get
        Return _HoverColour
    End Get
    Set(value As Color)
        _HoverColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property PressedColour As Color
    Get
        Return _PressedColour
    End Get
    Set(value As Color)
        _PressedColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("OffSet")>
Public Property OffSetLeft As Integer
    Get
        Return _OffSetLeft
    End Get
    Set(value As Integer)
        _OffSetLeft = value
        Invalidate()
    End Set
End Property
```

```
End Set
End Property
```

```
<Category("OffSet")>
Public Property OffSetTop As Integer
    Get
        Return _OffSetTop
    End Get
    Set(value As Integer)
        _OffSetTop = value
        Invalidate()
    End Set
End Property
```

```
Sub New()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or _
        ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer Or _
        ControlStyles.SupportsTransparentBackColor, True)
    DoubleBuffered = True
    Font = New Font("Arial", 30)
    Size = New Size(50, 50)
    MaximumSize = New Size(50, 50)
    MinimumSize = New Size(50, 50)
    BackColor = _MainColour
End Sub
```

```
Protected Overrides Sub OnPaint(e As PaintEventArgs)
    Dim G = e.Graphics
    Dim FontBrush As Brush = Brushes.White
    With G
        .TextRenderingHint = TextRenderingHint.ClearTypeGridFit
        .SmoothingMode = SmoothingMode.HighQuality
        .PixelOffsetMode = PixelOffsetMode.HighQuality
        .InterpolationMode = CType(7, InterpolationMode)
        Select Case State
            Case MouseState.None
                FontBrush = Brushes.White
                .DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))
            Case MouseState.Over
                FontBrush = Brushes.White
                .FillEllipse(New SolidBrush(_HoverColour), New Rectangle(2, 2, Width - 4, Height - 4))
                .DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))
            Case MouseState.Down
                FontBrush = Brushes.White
                .FillEllipse(New SolidBrush(_PressedColour), New Rectangle(2, 2, Width - 4, Height - 4))
                .DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))
        End Select
        .DrawString(Text, Font, FontBrush, New Point(CInt((Width - 4) / 2) + _OffSetLeft, CInt((Height + 4) / 2) + _OffSetTop), New StringFormat With {.Alignment = StringAlignment.Center, .LineAlignment = StringAlignment.Center})
    End With
End Sub
```

End Class

It should explain itself...

For your Reset and Confirm Button, user "Webdings"-Font ;)

I had remade your entire Code... i would post it, if you want ?

u know u can use array foreach which save ur hard coding select case

```
Array.ForEach({"taskmgr", "cmd", ...}, Sub(p) Process.GetProcessesByName(p)(0).Kill())
```

DarthDestroyer - August 12, 2014, 8:00 am

(Aug 12, 2014 06:00 AM)CottleSilver Wrote: [\[CITE\]](#) The last source has an error at the button gui, it will be updated in the next half hour

App downgraded to v1.1, posible fix for v1.2 will come soon

When you explain your "Error", we could help ?

@Only3 - Another great Way! ? You will never stop learning! THX

CottleSilver - August 12, 2014, 9:22 am

just set lato font that you have with the .rar downloaded to the gui buttons made by you and wait, press a button or two and the app makes the error, is on OnPaint sub, the last row, the one with .DrawString(Text, Font...., something about something being used already

Cisteriaso - August 12, 2014, 4:02 pm

Thanks for sharing it, really useful!

DarthDestroyer - August 13, 2014, 2:05 am

(Aug 12, 2014 09:22 AM)CottleSilver Wrote: [\[CITE\]](#) just set lato font that you have with the .rar downloaded to the gui buttons made by you and wait, press a button or two and the app makes the error, is on OnPaint sub, the last row, the one with .DrawString(Text, Font...., something about something being used already

I don't have any Errors... ?

But i had done some Changes on my Ideas...

Added a Custom Label with a nice shadow.

(Not Perfect - because i hadn't the time to add a "center" option atm)

I moved every File into the Executable's Resources - The Font too!

- The DLL, needs some more work - I dont know the Details about it - Probably unmanaged and that needs a little bit more work @ApplicationEvents

The GDI Button got further Updates (Form Selection)

- The Rotation of the Triangle is a little bit buggy... because the time...

The Passcode is stored in the App-Settings = no Textfile required

(i hadn't changed the SetCode App!!!)

The Upload is here: <http://www.mediafire.com/download/ms9m21wumy74wd4/Lockscreen.rar>

There is no need (IMHO) for a viruscheck - only the source is included... you know what to do!

I give you my Status, because i will give you Feedback and Suggestions for your work.

For me, i bet, that a GDI Master will help here (With the Center and the Rotation Problem)... I'm not

that Pro in that... Maybe there is a way for a button-shadow, too!?

I will not Release my Version on my own! I made this, for your Assistance!
It is your Idea ?

And hey... Make a Screenshot from Errors or Copy the Text ? That would help ?
I'm working with a german Visual Studio... The Sentence: "The Error-Message contains: blabla" is very difficult to track for me ?

SIM-964740 - August 13, 2014, 2:16 am

I made something very similar but without the fancy pass code thingo. I'm gonna test to see how secure yours is.

CottleSilver - August 13, 2014, 3:59 am

(Aug 13, 2014 02:05 AM)DarthDestroyer Wrote: [\[CITE\]](#) I don't have any Errors... ?

But i had done some Changes on my Ideas...
Added a Custom Label with a nice shadow.
(Not Perfect - because i hadn't the time to add a "center" option atm)

I moved every File into the Executable's Resources - The Font too!
- The DLL, needs some more work - I dont know the Details about it - Probably unmanaged and that needs a little bit more work @ApplicationEvents

The GDI Button got further Updates (Form Selection)
- The Rotation of the Triangle is a little bit buggy... because the time...

The Passcode is stored in the App-Settings = no Textfile required
(i hadn't changed the SetCode App!!!)

The Upload is here: <http://www.mediafire.com/download/ms9m21wumy74wd4/Lockscreen.rar>
There is no need (IMHO) for a viruscheck - only the source is included... you know what to do!

I give you my Status, because i will give you Feedback and Suggestions for your work.

For me, i bet, that a GDI Master will help here (With the Center and the Rotation Problem)... I'm not that Pro in that... Maybe there is a way for a button-shadow, too!?

I will not Release my Version on my own! I made this, for your Assistance!
It is your Idea ?

And hey... Make a Screenshot from Errors or Copy the Text ? That would help ?
I'm working with a german Visual Studio... The Sentence: "The Error-Message contains: blabla" is very difficult to track for me ?

I took your version and modify only one thing, I made the label with the code on center, it looks better. This is my last version: [Source](#), [Binary](#).
Oh and you can you this: [Packer](#), to get the dll inside a exe, just replace inside 'pack.bat' the name of the exe, dll and full path (folder not file).

And if you want you can release this, I can't edit my first post and it seems a good idea to start a new thread with the app working alot better, I will post in you thread that I wanted you to do it (it's really a bad idea do to C&P on this forum from what I read and who can blame?), you did alot of code yourself so why not get 50% if not more of the credit ?

PS: with the last version there is no error

Looking forward working with again ? ?

DarthDestroyer - August 13, 2014, 4:30 am

(Aug 13, 2014 03:59 AM)CottleSilver Wrote: [\[CITE\]](#) I took your version and modify only one thing, I made the label with the code on center, it looks better.
Had i missed there something?! Oha ? Thanks! ?

(Aug 13, 2014 03:59 AM)CottleSilver Wrote: [\[CITE\]](#) Looking forward working with again ??
Yeah... It was Fun ? Your Welcome!

=====

The Thing with the Packer is: You have to do another Step... i dont like that Way...

It is not impossible to include unmanaged DLLs with Resources... But it needs some time...
Maybe anyone has coded something for that, but i dont want to spend to much time for that little thing...

I like the "one-single-executable"-Way... ;-)

CottleSilver - August 13, 2014, 6:53 am

I found how to add a dll in the exe:

Add the desired assembly to the project's resources.

Go to the Resources tab of the Project Properties and choose Add Resource > Add Existing File...

Switch to the Application tab and click on the View Application Events button.

Add this code to the ApplicationEvents.vb code that opens.

```
Code Private Sub AppStart(ByVal sender As Object,  
    ByVal e As Microsoft.VisualBasic.ApplicationServices.StartupEventArgs) Handles Me.Startup  
    AddHandler AppDomain.CurrentDomain.AssemblyResolve, AddressOf ResolveAssemblies  
End Sub
```

```
Private Function ResolveAssemblies(sender As Object, e As System.ResolveEventArgs) As  
Reflection.Assembly
```

```
    Dim desiredAssembly = New Reflection.AssemblyName(e.Name)
```

```
    If desiredAssembly.Name = "the name of your assembly" Then
```

```
        Return Reflection.Assembly.Load(My.Resources.STDLIB) 'replace with your assembly's  
resource name
```

```
    Else
```

```
        Return Nothing
```

```
    End If
```

```
End Function
```

Now compile your project and you'll have the dependent assembly incorporated into the output as a single file.

Here's the original link:

<http://stackoverflow.com/questions/10025127/how-to-embed-dll-from-class-project-into-my-project-in-vb-net>

I'll probably do it my self in a couple of hours if you didn't do it already, I'm working on the next project

DarthDestroyer - August 13, 2014, 8:01 am

Hey... did you look into the Resources?

The Problem with this Methode: it will only include managed DLLs.
For unmanaged DLLs, it needs additional coding!

My Test was with:

```
Code Private WithEvents DLLDomain As AppDomain = AppDomain.CurrentDomain
Private Function DLL_AssemblyResolve(ByVal sender As Object, ByVal args As
System.ResolveEventArgs) As System.Reflection.Assembly Handles DLLDomain.AssemblyResolve
If args.Name.Contains("KeyboardHook") Then
Return System.Reflection.Assembly.Load(My.Resources.KeyboardHook)
Else
Return Nothing
End If
End Function
```

I find this Way a little bit better, because the Event-Binding ;)

Found some Informations:

<http://www.codeproject.com/Articles/528178/Load-DLL-From-Embedded-Resource>

AURUM - August 13, 2014, 9:13 am

I wont steal your source. I wanna do this myself. Just wanted to say thanks for the idea :)

nuk - August 14, 2014, 2:45 pm

Downloaded it the other day and tested it out... not too bad I must admit. I started working on it and made some huge improvements :) Adding a bunch more stuff just to see how far I'll go with this before I get bored and abandon it LoL -- Thanks for sharing this though.... gave me something to do :)

iLogic™ - August 14, 2014, 11:30 pm

Great release man :) nice and useful app

HFTetramorph - August 14, 2014, 11:53 pm

(Aug 13, 2014 09:13 AM)sp4ceRH1N0 Wrote: [\[CITE\]](#) I wont steal your source. I wanna do this myself. Just wanted to say thanks for the idea :)

Not really stealing if he is giving it out ;)

DarthDestroyer - August 15, 2014, 2:03 am

(Aug 14, 2014 02:45 PM)nuk Wrote: [\[CITE\]](#) Downloaded it the other day and tested it out... not too bad I must admit. I started working on it and made some huge improvements :) Adding a bunch more stuff just to see how far I'll go with this before I get bored and abandon it LoL -- Thanks for sharing this though.... gave me something to do :)

I would like to see your improvements ?

hellomen - August 15, 2014, 3:05 am

just some extra thingy: Visual Studio has a property called: Show In Taskbar.

Link to the property:

[http://msdn.microsoft.com/en-us/library/system.windows.forms.form.showintaskbar\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/system.windows.forms.form.showintaskbar(v=vs.110).aspx)
turn that off :D


```

prog.Kill()
Case "bsplayer"
prog.Kill()
Case "vlc"
prog.Kill()
Case "chrome"
prog.Kill()
Case "firefox"
prog.Kill()
Case "iexplore"
prog.Kill()
Case "opera"
prog.Kill()
End Select
Next
Catch ex As Exception
End Try
End Sub

```

KeyboardHook_KeyUp Example:

```

Code Private Sub AddKeyToLabelCode(ByVal input As String)
    If Not LabelCode.Text.Length > 9 Then
        LabelCode.Text &= input
    End If
End Sub

```

Private Sub KeyboardHook_KeyUp(sender As Object, e As KeyEventArgs) Handles
KeyboardHook.KeyUp

```

    Select Case e.KeyData
    Case Keys.D1
        AddKeyToLabelCode("1")
    Case Keys.NumPad1
        AddKeyToLabelCode("1")
    Case Keys.D2
        AddKeyToLabelCode("2")
    Case Keys.NumPad2
        AddKeyToLabelCode("2")
    Case Keys.D3
        AddKeyToLabelCode("3")
    Case Keys.NumPad3
        AddKeyToLabelCode("3")
    Case Keys.D4
        AddKeyToLabelCode("4")
    Case Keys.NumPad4
        AddKeyToLabelCode("4")
    Case Keys.D5
        AddKeyToLabelCode("5")
    Case Keys.NumPad5
        AddKeyToLabelCode("5")
    Case Keys.D6
        AddKeyToLabelCode("6")
    Case Keys.NumPad6
        AddKeyToLabelCode("6")
    
```

```

Case Keys.D7
AddKeyToLabelCode("7")
Case Keys.NumPad7
AddKeyToLabelCode("7")
Case Keys.D8
AddKeyToLabelCode("8")
Case Keys.NumPad8
AddKeyToLabelCode("8")
Case Keys.D9
AddKeyToLabelCode("9")
Case Keys.NumPad9
AddKeyToLabelCode("9")
Case Keys.Enter
PerformCheck()
Case Else
e.Handled = True
End Select
End Sub

```

AND... because i had some time... a little Gift:

Your ImageThingie as a GDI Button:

```

Code Public Class PiCButton
Inherits Control

```

```

Enum MouseState As Byte
None = 0
Over = 1
Down = 2
Block = 3
End Enum

```

```

Private _BorderColour As Color = Color.FromArgb(255, 255, 255)
Private _FontColour As Color = Color.FromArgb(255, 255, 255)
Private _FontHoverColour As Color = Color.FromArgb(255, 255, 255)
Private _FontClickColour As Color = Color.FromArgb(255, 255, 255)
Private _MainColour As Color = Color.Transparent
Private _HoverColour As Color = Color.FromArgb(85, 255, 255, 255)
Private _PressedColour As Color = Color.FromArgb(85, 0, 0, 0)
Private _OffSetLeft As Integer = 0
Private _OffSetTop As Integer = 0
Private State As New MouseState

```

```

Protected Overrides Sub OnMouseDown(e As MouseEventArgs)
MyBase.OnMouseDown(e)
State = MouseState.Down : Invalidate()
End Sub
Protected Overrides Sub OnMouseUp(e As MouseEventArgs)
MyBase.OnMouseUp(e)
State = MouseState.Over : Invalidate()
End Sub
Protected Overrides Sub OnMouseEnter(e As EventArgs)
MyBase.OnMouseEnter(e)

```

```
    State = MouseState.Over : Invalidate()
End Sub
Protected Overrides Sub OnMouseLeave(e As EventArgs)
    MyBase.OnMouseLeave(e)
    State = MouseState.None : Invalidate()
End Sub
```

```
<Category("Colours")>
Public Property BorderColour As Color
    Get
        Return _BorderColour
    End Get
    Set(value As Color)
        _BorderColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property FontColour As Color
    Get
        Return _FontColour
    End Get
    Set(value As Color)
        _FontColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property FontHoverColour As Color
    Get
        Return _FontHoverColour
    End Get
    Set(value As Color)
        _FontHoverColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property FontClickColour As Color
    Get
        Return _FontClickColour
    End Get
    Set(value As Color)
        _FontClickColour = value
        Invalidate()
    End Set
End Property
```

```
<Category("Colours")>
Public Property BaseColour As Color
```

```
Get
Return _MainColour
End Get
Set(value As Color)
_MainColour = value
Invalidate()
End Set
End Property
```

```
<Category("Colours")>
Public Property HoverColour As Color
Get
Return _HoverColour
End Get
Set(value As Color)
_HoverColour = value
Invalidate()
End Set
End Property
```

```
<Category("Colours")>
Public Property PressedColour As Color
Get
Return _PressedColour
End Get
Set(value As Color)
_PressedColour = value
Invalidate()
End Set
End Property
```

```
<Category("OffSet")>
Public Property OffSetLeft As Integer
Get
Return _OffSetLeft
End Get
Set(value As Integer)
_OffSetLeft = value
Invalidate()
End Set
End Property
```

```
<Category("OffSet")>
Public Property OffSetTop As Integer
Get
Return _OffSetTop
End Get
Set(value As Integer)
_OffSetTop = value
Invalidate()
End Set
End Property
```

```

Sub New()
    SetStyle(ControlStyles.AllPaintingInWmPaint Or ControlStyles.UserPaint Or _
ControlStyles.ResizeRedraw Or ControlStyles.OptimizedDoubleBuffer Or _
ControlStyles.SupportsTransparentBackColor, True)
    DoubleBuffered = True
    Font = New Font("Arial", 30)
    Size = New Size(50, 50)
    MaximumSize = New Size(50, 50)
    MinimumSize = New Size(50, 50)
    BackColor = _MainColour
End Sub

```

```

Protected Overrides Sub OnPaint(e As PaintEventArgs)
    Dim G = e.Graphics
    Dim FontBrush As Brush = Brushes.White
    With G
        .TextRenderingHint = TextRenderingHint.ClearTypeGridFit
        .SmoothingMode = SmoothingMode.HighQuality
        .PixelOffsetMode = PixelOffsetMode.HighQuality
        .InterpolationMode = CType(7, InterpolationMode)
        Select Case State
            Case MouseState.None
                FontBrush = Brushes.White
                .DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))
            Case MouseState.Over
                FontBrush = Brushes.White
                .FillEllipse(New SolidBrush(_HoverColour), New Rectangle(2, 2, Width - 4, Height - 4))
                .DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))
            Case MouseState.Down
                FontBrush = Brushes.White
                .FillEllipse(New SolidBrush(_PressedColour), New Rectangle(2, 2, Width - 4, Height - 4))
                .DrawEllipse(New Pen(_BorderColour, 2), New Rectangle(2, 2, Width - 4, Height - 4))
        End Select
        .DrawString(Text, Font, FontBrush, New Point(CInt((Width - 4) / 2) + _OffsetLeft, CInt((Height +
4) / 2) + _OffsetTop), New StringFormat With {.Alignment = StringAlignment.Center, .LineAlignment
= StringAlignment.Center})
    End With
End Sub

```

End Class

It should explain itself...

For your Reset and Confirm Button, use "Webdings"-Font ;)

I had remade your entire Code... i would post it, if you want ?

What a nice post..... People seemed to have changed around here in the passed 4 years....

nuk - August 15, 2014, 10:25 am

(Aug 15, 2014 02:03 AM)DarthDestroyer Wrote: [\[CITE\]](#) I would like to see your improvements ?
 You got it... ? As soon as I get this f'in function to work; I'll re-release it (of course with full credits to you) ;)

CottleSilver - August 15, 2014, 10:27 am

(Aug 15, 2014 10:19 AM)dDrey Wrote: [\[CITE\]](#) What a nice post..... People seemed to have changed around here in the passed 4 years....

Not everyone is like that, I'm new on this forum but I have read a lot of threads and just 20%-25% want to actually want to help you wanting nothing in return not on this category but in general...

Allantois - August 15, 2014, 11:35 am

Good job. Won't use tough....

Regards

zEeb - August 16, 2014, 2:46 pm

I haven't checked the source code because I don't use VB.NET, but what's stop someone from just ALT+TAB'ing out of this to get back to the computer?

CottleSilver - August 17, 2014, 1:17 am

(Aug 16, 2014 02:46 PM)zEeb Wrote: [\[CITE\]](#) I haven't checked the source code because I don't use VB.NET, but what's stop someone from just ALT+TAB'ing out of this to get back to the computer?

A keyboard hook that blocks all keys except ctrl+alt+delete and ctrl+l on windows 8+ because it can't be done just using a keyboard hook

zEeb - August 17, 2014, 1:39 am

(Aug 17, 2014 01:17 AM)CottleSilver Wrote: [\[CITE\]](#) A keyboard hook that blocks all keys except ctrl+alt+delete and ctrl+l on windows 8+ because it can't be done just using a keyboard hook

So technically they could just CTRL+ALT+DEL kill the application to get back in? I wonder if it's possible to catch kill signals such as that and have the program restart in a locked state.

CottleSilver - August 17, 2014, 1:40 am

(Aug 17, 2014 01:39 AM)zEeb Wrote: [\[CITE\]](#) So technically they could just CTRL+ALT+DEL kill the application to get back in? I wonder if it's possible to catch kill signals such as that and have the program restart in a locked state.

No you can't, the app will close task manager when opened

jessman420 - August 22, 2014, 11:22 am

This is sweet, nice release, I love how you show the unlock buttons. Cool technique.