

Forum:[Hack Forums](#)

Thread: [\[WINGLESS\] Perfect Rounded Rectangle using WINAPI+MSAA !!](#)

tusharthegreat - February 22, 2016, 6:34 pm

Perfect ???

As some of you know, the usual RoundRect function which uses arcs and ellipses to draw roundrectangles is not very efficient. Artifacts can be seen everywhere, and AA sucks with it.

The top left rounded corner is fine, but the rest of them far from fine. So much that the bottom right seems completely distinguishable from the top left. Now this is not symmetrical and human eye praises symmetry so, they are bad.

Most of the time you will find yourself compromising to the situation but no more !

MSAA+WinAPI to the rescue

What is this MSAA ?

MSAA is a method of anti aliasing used to smooth out jagged lines. In this method, the target image is rendered at a higher resolution than normal, and the shrunk to the original size. Downscaling largely removes aliasing, and the render resolution is, smoother is the image.

WinAPI ?

I saw a function in WinAPI that produces roundrects. Named CreateRoundRectRgn, it takes the bounds of the rectangle and radius of the rounds as parameters, and gives to us a perfectly rounded-rectangle region. You can now use this region as you wish.

MSAA + WinAPI ?

Apparently the region that you create with CreateRoundRectRgn is a very jagged one, with harsh lines everywhere. It may be perfectly round, but actually is far from perfect. So I decided to get the best of both worlds, combine CreateRoundRectRgn with 8x MSAA and Voila ! there it was, the PERFECT roundrect.

This one does not have any [normally visible] jagged lines, all the rounds are symmetrical, and with the function I have written, you can draw a roundrect in just one statement, complete with border and fill color. (Set the fill/border color to color.transparent if you do not want them to be showed.)

Code Class RoundedRectangle

```
<DllImport("gdi32.dll")> _
Shared Function CreateRoundRectRgn(ByVal x1 As Integer, ByVal y1 As Integer, ByVal x2 As Integer, ByVal y2 As Integer, ByVal cx As Integer, ByVal cy As Integer) As IntPtr
End Function

Shared Function DrawRoundRectangle(rct As Rectangle, r As Integer, MSAA As Integer, FillColor As Color, BorderColor As Color)
    Dim b As New Bitmap(rct.Width * MSAA + MSAA, rct.Height * MSAA + MSAA)
    Dim g As Graphics = Graphics.FromImage(b)
    Dim RegH As IntPtr = CreateRoundRectRgn(rct.Left, rct.Top, (rct.Width) * MSAA, (rct.Height) * MSAA, r * MSAA, r * MSAA)
    Dim RegH1 As IntPtr = CreateRoundRectRgn(rct.Left + MSAA / 1, rct.Top + MSAA / 1,
```

```
(rct.Width - 1) * MSAA, (rct.Height - 1) * MSAA, (r - 2) * MSAA, (r - 2) * MSAA)
```

```
Dim Reg As Region = Region.FromHrgn(RegH)
```

```
Dim Reg1 As Region = Region.FromHrgn(RegH1)
```

```
g.InterpolationMode = InterpolationMode.HighQualityBicubic
```

```
g.FillRegion(New SolidBrush(BorderColor), Reg)
```

```
g.SetClip(Reg1, CombineMode.Replace)
```

```
g.Clear(FillColor)
```

```
Return b
```

```
End Function
```

```
End Class
```

The function DrawRoundRectangle returns a bitmap which you can draw as in the following example:

```
Code Protected Overrides Sub OnPaint(e As PaintEventArgs)
```

```
Dim g As Graphics = e.Graphics
```

```
g.Clear(Parent.BackColor)
```

```
g.DrawImage(RoundedRectangle.DrawRoundRectangle(New Rectangle(5, 5, Width - 10, Height - 10), 10, 8, Color.FromArgb(28, 28, 28), Color.FromArgb(51, 51, 51)), 5, 5, Width - 10, Height - 10)
```

```
End Sub
```

Here the base rectangle is rect(5,5,width-10,height-10),

radius is 10 and

MSAA multiplier is 8(x)

Border color is color.fromargb(51,51,51)

Fill Color is color.fromargb(28,28,28)

image:--- i65.tiny pic. com/ 2wp01uc.png

Also, do not forget to use the

```
G.InterpolationMode = 7
```

(I have not used it in the code) Without it, your borders may be different in width.

Hiiro - February 22, 2016, 7:22 pm

Re-uploaded so I can display it here for you.

tusharthegreat - February 22, 2016, 10:04 pm

Thx.

Also, if you want to get the region for clipping purposes, you can edit my function to return the region directly.

Earn - February 22, 2016, 10:20 pm

That is far from perfect bro.

My Computer - February 22, 2016, 10:26 pm

(Feb 22, 2016 11:20 PM)Earn Wrote: [\[CITE\]](#) That is far from perfect bro.

I believe his rectangle is the dark gray one because in his source code he has dark RGB colors. He probably just drew the red one as an example to show how it is compared to his.

Isley - February 22, 2016, 10:32 pm

(Feb 22, 2016 11:26 PM)EggsAndToast Wrote: [\[CITE\]](#) I believe his rectangle is the dark gray one because in his source code he has dark RGB colors. He probably just drew the red one as an example to show how it is compared to his.

This, especially since the dark grey one has, yknow, perfect rounded edges.

Earn - February 22, 2016, 10:37 pm

(Feb 22, 2016 11:26 PM)EggsAndToast Wrote: [\[CITE\]](#) I believe his rectangle is the dark gray one because in his source code he has dark RGB colors. He probably just drew the red one as an example to show how it is compared to his.

Ah, okay. My bad.

Nice work OP.

SaketSaket - February 22, 2016, 11:49 pm

Thanks for sharing... Will look into your code... ;)

6e756c6c - February 23, 2016, 12:39 am

First off, great contribution - nice to see something new! In regards to the issue you are referring to though, the solution in GDI is to simply set the Graphics.PixelOffsetMode to HighQuality. Though it is a bit.. quirky. In my experience (if I recall correctly), you should set the PixelOffsetMode when anti-aliasing isn't being used for nice sharp, symmetrical corners, otherwise disable it. That or it was something more bizarre like only using it to 'fill' a rectangle but not for 'drawing' a rectangle. It's one of those two, I'm 75% sure! :D

tusharthegreat - February 23, 2016, 3:25 am

Thx man.

You contributed to me more than I did! HAHA!

Oh, and do not set the MSAA multiplier to greater than 64x, it will cause an error.

Anyway, 8x is more than enough for anything and for big shapes I do not recommend more than 4x. If I remember correctly, just drawing a text, a roundrect, an ellipse and a line of sizes like 200x100 px took 163 ms with 16x MSAA. That's a LOT of time.

playonlinux03 - February 23, 2016, 3:57 am

good job man. =) it's perfect tuto

tusharthegreat - February 23, 2016, 4:02 am

I just realized, if you pass the borderwidth argument as a negative value of the decimal part not equal to 0, it gives you a ready made drop shadow !

Eg- pass borderwidth as 0.5 and the border color as transparent.

This works really well with container controls.

?MrRoot - February 23, 2016, 9:46 am

Why are you dividing by 1?

Never knew how MSAA worked so thanks a lot for the concept.

tusharthegreat - February 24, 2016, 4:22 am

(Feb 23, 2016 10:46 AM)SecuroServ Wrote: [\[CITE\]](#) Why are you dividing by 1?

It's nothing, initially I was trying to check what happens if you divided it by 2 or 4 but then I realised it was of no use. Remove it if you want to.

GammaRay - March 7, 2016, 4:01 am

The fix for the problem is really simple with GDI+. This is so silly. You first fill the rounded rectangle and then you draw it. The drawing does the anti-aliasing for you.

tusharthegreat - March 19, 2016, 9:31 am

(Mar 7, 2016 05:01 AM)GammaRay Wrote: [\[CITE\]](#) The fix for the problem is really simple with GDI+. This is so silly. You first fill the rounded rectangle and then you draw it. The drawing does the anti-aliasing for you.

That's what I did. The function fills the rounded rectangle on a bitmap of suitable size and then that image is drawn on the required surface.

GammaRay - March 20, 2016, 5:42 pm

(Mar 19, 2016 09:31 AM)tusharthegreat Wrote: [\[CITE\]](#) That's what I did. The function fills the rounded rectangle on a bitmap of suitable size and then that image is drawn on the required surface. It's incredibly messy and inefficient.

What someone should do is this:

Use:

PHP Code

<?php

Public Function RoundRect(ByVal rectangle As Rectangle, ByVal curve As Integer) As GraphicsPath

Dim p As GraphicsPath = New GraphicsPath()

Dim arcRectangleWidth As Integer = curve * 2

p.AddArc(New Rectangle(rectangle.X, rectangle.Y, arcRectangleWidth, arcRectangleWidth), -180, 90)

p.AddArc(New Rectangle(rectangle.Width - arcRectangleWidth + rectangle.X, rectangle.Y, arcRectangleWidth, arcRectangleWidth), -90, 90)

p.AddArc(New Rectangle(rectangle.Width - arcRectangleWidth + rectangle.X, rectangle.Height - arcRectangleWidth + rectangle.Y, arcRectangleWidth, arcRectangleWidth), 0, 90)

p.AddArc(New Rectangle(rectangle.X, rectangle.Height - arcRectangleWidth + rectangle.Y, arcRectangleWidth, arcRectangleWidth), 90, 90)

p.AddLine(New Point(rectangle.X, rectangle.Height - arcRectangleWidth + rectangle.Y), New Point(rectangle.X, curve + rectangle.Y))

Return p

End Function

And in your paint event, simply write:

PHP Code

<?php

Dim RoundedRectangle as New Rectangle(20, 20, 20, 20)

Dim RoundedGraphicsPath As GraphicsPath = RoundRect(RoundedRectangle, yourRadius)

e.Graphics.FillPath(yourBrush, RoundedGraphicsPath)

e.Graphics.DrawPath(yourPen, RoundedGraphicsPath)

That simple. No silly API's or Bitmaps and risks of not disposing. Just simple GDI+.

tusharthegreat - March 20, 2016, 6:48 pm

(Feb 22, 2016 07:34 PM)tusharthegreat Wrote: [\[CITE\]](#) Perfect ???

The top left rounded corner is fine, but the rest of them far from fine. So much that the bottom right seems completely distinguishable from the top left. Now this is not symmetrical and human eye praises symmetry so, they are bad.

Most of the time you will find yourself compromising to the situation but no more !

Man, seems like you misunderstood the purpose if this thread. The aim here is not to create super fast lines and arcs, but a well polished, regular rounded rectangle. Sure, it is a but slow because of the MSAA but speed was never the target here. At the same time, I won't suggest anyone to fill their forms with these roundrects everywhere as they do take some time (around 100 ms when msaa multiplier is set to 8x) but then, you don't need 8x. For most of the small tasks, this should be sufficient.

All you did was recreate the usual roundrect function. Believe me, I have experimented enough with them. Tried the usual add arc, Ellipse, transform, image copy etc. The same result every time - unsymmetrical chopped off edges.

Hawk HF - March 21, 2016, 4:19 am

(Mar 20, 2016 06:48 PM)tusharthegreat Wrote: [\[CITE\]](#) Man, seems like you misunderstood the purpose if this thread. The aim here is not to create super fast lines and arcs, but a well polished, regular rounded rectangle. Sure, it is a but slow because of the MSAA but speed was never the target here. At the same time, I won't suggest anyone to fill their forms with these roundrects everywhere as they do take some time (around 100 ms when msaa multiplier is set to 8x) but then, you don't need 8x. For most of the small tasks, this should be sufficient.

All you did was recreate the usual roundrect function. Believe me, I have experimented enough with them. Tried the usual add arc, Ellipse, transform, image copy etc. The same result every time - unsymmetrical chopped off edges.

100ms is a noticable delay. As a user, experiencing constant 100ms UI freezes would be pretty annoying. As others said, this is a good contribution, but it's not the best solution for this.

The chopped off edges come from a bit of math. When you fill a pixel, let's say (0, 0) for example, you're actually filling a rectangle (0, 0, 1, 1). So your width and height will be 1 pixel larger than you're expecting. In most themes, you'll notice many occurrences where rectangles are being painted using a width of this.Width - 1 or a height of this.Height - 1. This is why.

?????? - March 21, 2016, 7:22 am

Nice tut mate :)

Keep up the good work