Calculating point on a circle's circumference from angle in C#?

Asked 15 years, 9 months ago Modified 7 years, 4 months ago Viewed 46k times

42

I imagine that this is a simple question, but I'm getting some strange results with my current code and I don't have the math background to fully understand why. My goal is simple, as stated in the title: I just want to find the point at some distance and angle from a center point.

My current code:

```
Point centerPoint = new Point ( 0, 0 );
Point result      = new Point ( 0, 0 );
double angle      = 0.5; //between 0 and 2 * PI, angle is in radians
int distance      = 1000;

result.Y = centerPoint.Y + (int)Math.Round( distance * Math.Sin( angle ) );
result.X = centerPoint.X + (int)Math.Round( distance * Math.Cos( angle ) );
```

In general, this seems to work fairly reasonably, but I get problems at various spots, most notably when the angle corresponds to points in the negative x and y axis. Clearly I'm doing something wrong -- thoughts on what that is?

UPDATE: This was my mistake, this code works fine -- the few outliers that were not working were actually due to a bug in how the angle for 1.5PI was being calculated. I thought I had checked that well enough, but evidently had not. Thanks to everyone for their time, hopefully the working code above will prove helpful to someone else.

c# math geometry angle

Share
Improve this question
Follow

edited Feb 6, 2012 at 14:21

Jonas
128k ● 100 ● 326 ● 405

asked Mar 23, 2009 at 16:57

Chris McElligott Park
3,127 ● 3 ● 28 ● 25

What do you mean "i get problems at various spots"? – Restore The Data Dumps Again Mar 23, 2009 at 16:59

is centerPoint deliberately unused? Shouldn't centerPoint.X and .Y be added to result? – spender Mar 23, 2009 at 17:05

can you provide an example of a problem? – Andrew Grant Mar 23, 2009 at 17:05

I don't expect you'll get more than shots in the dark until you describe what exactly is wrong with your results. – P Daddy Mar 23, 2009 at 17:17

Please do not refer to your distance as X and angle as Y. That is counter intuitive. Distance is your radius and angle is your... angle. – Joe Phillips Mar 23, 2009 at 17:20

6 Answers

Sorted by: Highest score (default)

35

You forgot to add the center point:

```
result.Y = (int)Math.Round( centerPoint.Y + distance * Math.Sin( angle ) );
result.X = (int)Math.Round( centerPoint.X + distance * Math.Cos( angle ) );
```

The rest should be ok... (what strange results were you getting? Can you give an exact input?)

Share Improve this answer Follow

answered Mar 23, 2009 at 17:19

MartinStettner
29.1k ● 15 ● 82 ● 106

Ah, this is a good point about the centerpoint -- it is unused in this case. The specific point I'm working with at present is with a centerpoint of 0,0, so it's irrelevant in that case, but in other cases it would certainly make a difference. I've updated the example to properly include it. – Chris McElligott Park Mar 23, 2009 at 17:54

6

Firstly, since you're in radians it's probably beneficial to define your angle as such:

```
double angle = (Math.PI / 3); // 60 degrees...
```

The functions themselves are working fine. The rounding will only affect your answer if your distance is sufficiently small enough. Other than that, the answers should come out just fine.

If it's the rounding you're worried about, remember that by default, .NET does banker's rounding, and you may want:

```
result.X = (int)Math.Round(centerPoint.X + distance * Math.Cos(angle),
MidpointRounding.AwayFromZero);
result.Y = (int)Math.Round(centerPoint.Y + distance * Math.Sin(angle),
MidpointRounding.AwayFromZero);
```

instead.

Additionally, in the question you want *distance X and angle Y*... I assume you're not relating that to the point `(X,Y)`, because that's completely different.

The distance formula is:

```
double distance = Math.Sqrt((centerPoint.X + result.X)^2 + (centerPoint.Y +
result.Y)^2);
```

Share Improve this answer Follow

edited Mar 23, 2009 at 18:11

answered Mar 23, 2009 at 17:17

John Rasch
63.4k ● 19 ● 109 ● 141

> John, Thanks for the fish. I was blissfully unaware that C# was rounding double calculations at all, let alone awfully (IMHO). – corlettk Sep 25, 2012 at 7:30

3

A Swift3 version

```
func pointOnCircle(radius: Double, angleInDegrees: Double, origin: CGPoint) ->
CGPoint {
    let x = abs(Double(origin.x) + radius * cos(angleInDegrees * (.pi / 180)))
    let y = abs(Double(origin.y) - radius * sin(angleInDegrees * (.pi / 180)))

    return CGPoint(x: x, y: y)
}
```

Share Improve this answer Follow

answered Aug 8, 2017 at 19:43

Joe Susnick
6,772 ● 7 ● 48 ● 55

1

Without more information on the exact errors it's hard to tell what's wrong. The equations look right and should work. Are you sure the angles you are passing in are correct for angles > 90 degrees? The only other thing I could think of would be that you're multiplying distance (an int) by the result of Math.sin (double) but that shouldn't really be an issue.

Share Improve this answer Follow

answered Mar 23, 2009 at 18:12

JonBWalsh
321 ● 2 ● 10

1

```
-(NSMutableArray *)GetPointsForCircle
    {
       NSMutableArray *Points = [[NSMutableArray alloc] init];
       CGPoint CenterPoint = CGPointMake(160, 230);
```

```
        CGPoint Point;
        for (float Angel = 0; Angel <= 360; Angel+= 60)
          {
            Point.x = CenterPoint.x + 100 * cos(Angel);
            Point.y = CenterPoint.y + 100 * sin(Angel);
            [Points addObject:[NSValue valueWithCGPoint:Point]];
          }
        return Points;
    }

    - (CGPoint)pointOnCircle:(int)thisPoint withTotalPointCount:
(int)totalPoints
     {
        CGPoint centerPoint = CGPointMake(self.view.frame.size.width / 2,
self.view.frame.size.height / 2);
        float radius = 100.0;
        float angle = ( 2 * M_PI / (float)totalPoints ) * (float)thisPoint;
        CGPoint newPoint;
        newPoint.x = (centerPoint.x) + (radius * cosf(angle));
        newPoint.y = (centerPoint.y) + (radius * sinf(angle));
        return newPoint;
    }
```

Share

Improve this answer

Follow

edited Sep 20, 2012 at 6:04

answered Sep 20, 2012 at 5:58

Tahir Waseem

149 ● 2 ● 7

Return 6 Points for a circle of radius 100. The parametric equation for a circle is x = cx + r * cos(a) y = cy + r * sin(a) Where r is the radius, cx,cy the origin, and a the angle from 0..2PI radians or 0..360 degrees. – Tahir Waseem Sep 20, 2012 at 6:00

-1

I don't know c#, anyway if you are trying to draw the points somewhere you have to consider the fact that the Y axis crease from the top to the bottom of the screen, so your **sin** element should have be **-sin(...)** and not *+sin(...)*

so

```
result.Y = centerPoint.Y + (int)Math.Round( distance * Math.Sin( angle ) );
```

should become:

```
result.Y = centerPoint.Y - (int)Math.Round( distance * Math.Sin( angle ) );
```

If you are not trying to draw them I could not imagine what the problem is, can you give some example?

Share Improve this answer Follow

answered Mar 23, 2009 at 18:18

Andrea Ambu
39.4k ● 14 ● 56 ● 77

result.Y should use cos not sin check this link webdeveloper.com/forum/showthread.php?t=190439 – RVN Feb 16, 2011 at 13:10

@RVN your link is dead – AntonioCS Feb 19, 2019 at 18:40