

## Square Collimator

When the collimator was a circle , we drew an arc of degree 360 .Below is the code snippet with circle as collimator .

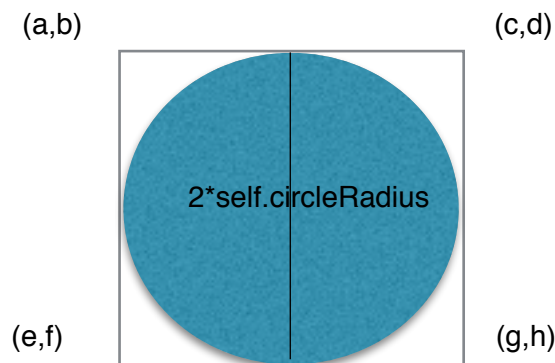
```
[path addArcWithCenter:self.circleCenter
    radius:self.circleRadius
    startAngle:0.0
    endAngle:DEGREES_TO_RADIANS(360)
    clockwise:YES];
```

Now in order to include a larger surface area we have changed the collimator to a square so that even the edges are also taken into account.

The change in the code would be to create a square such that the edge length would be the same as the diameter used to create the circle .

```
[path moveToPoint:CGPointMake(a, b)];
[path addLineToPoint:CGPointMake(c, d)];
[path addLineToPoint:CGPointMake(e, f)];
[path addLineToPoint:CGPointMake(g, h)];
[path addLineToPoint:CGPointMake(a, b)];
```

Where



So far we have created the collimator and we are successful in changing the collimator shape from circle to square .

Our next attempt is to change the **dimensions** of the square as we move in and out the collimator .

Consider the code snippet below ,

```
CGFloat xMove = nowPoint.x - previousPoint.x;
CGFloat yMove = nowPoint.y - previousPoint.y;
CGFloat distance = sqrt ((xMove * xMove) + (yMove * yMove));
```

nowPoint gives us the current point of touch on the screen and the previousPoint gives us the previous touch on the screen. This helps us to obtain the distance of movement from both the x axis and y axis perspective .

**Note:-** Distance between 2 points (x,y) and (p,q) is given by the equation  $\sqrt{(x-p)^2 + (y-q)^2}$  which is exactly used in the above piece of code .

Now we need to determine the new (a,b) , (c,d) , (e,f) and (g,h) .Consider the code snippet below .

```
if (previousPoint.y<nowPoint.y)
{
    if (squarePressed)
    {
        a=a+distance ;
        b=b+distance ;
        c=c+distance ;
        d=d-distance ; //y
        e=e-distance ; //x
        f=f-distance ; //y
        g=g-distance ; //x
        h=h+distance ;
    }
}
```

When the y value of previousPoint is < y value of nowPoint would mean we are reducing the square collimator value (in layman terms we are shrinking the collimator size) .

So now the new corner coordinate values would be ,

```
a=a+distance ;
b=b+distance ;
c=c+distance ;
d=d-distance ; //y
e=e-distance ; //x
f=f-distance ; //y
g=g-distance ; //x
h=h+distance ;
```

Similarly we update the values of coordinates when we increase the size of the square collimator .  
Placing the code here .

```
else
{
    if (squarePressed)
    {
        a=a-distance ;
        b=b-distance ;
        c=c-distance ;
        d=d+distance ; //y
        e=e+distance ; //x
        f=f+distance ; //y
        g=g+distance ; //x
        h=h-distance ;
    }
}
```

\*\*\*\*\*THE END\*\*\*\*\*

