Coding Challenge Specification

Table of Contents

[1. Program - Rasterizing circles: 1](#_Toc492933551)

[1) Design & Algorithm: 2](#_Toc492933552)

[ List of Classes used: 2](#_Toc492933553)

[2) Improvements considered but not implemented: 3](#_Toc492933554)

[3) How to Test: 4](#_Toc492933555)

[2. Program - Generates best-fit Circle: 4](#_Toc492933556)

[1) Design & Algorithm: 5](#_Toc492933557)

[ List of Classes used: 5](#_Toc492933558)

[2) Improvements considered but not implemented: 5](#_Toc492933559)

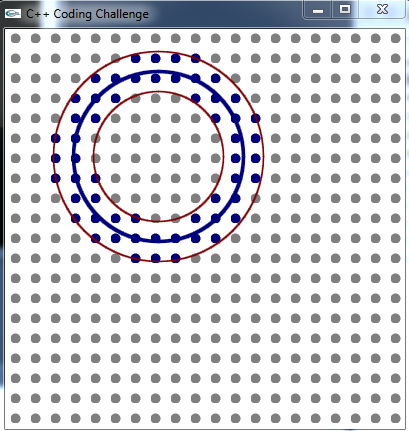
[3) How to Test: 5](#_Toc492933560)

[3. Program - Generates best-fit Eclipse: 6](#_Toc492933561)

[1) How to Test: 6](#_Toc492933562)

# Program - Rasterizing circles:

Rasterization is the task of taking an image described in a vector graphics format (shapes) and converting it into a raster image (pixels or dots) for output on a video display or printer, or for storage in a bitmap file format. It refers to both rasterization of models and 2D rendering primitives such as polygons, line segments, etc.



## Design & Algorithm:

To solve this problem, we must consider the following mathematical equations:

**R = sqrt((x1-x2) ^2 + (y1-y2) ^2)**

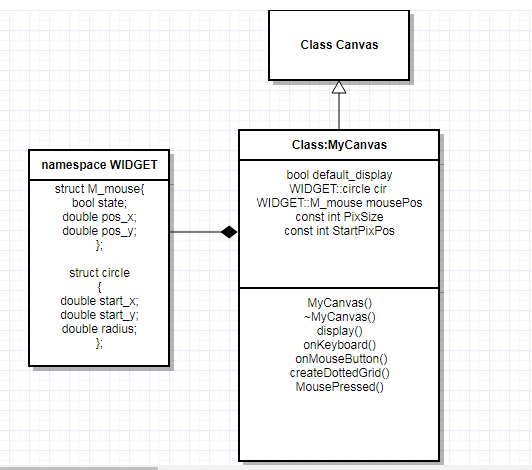
Where R = Radius of Circle,

(x1, y1) – Origin of circle,

(x2, y2) – Any point at circumference of circle.

### List of Classes used:

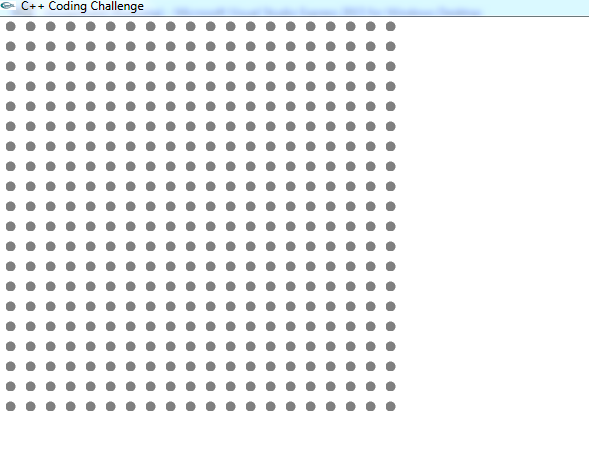
There are two classes used, in this class Canvas & Class MyCanvas.



## Improvements considered but not implemented:

While doing this task, I faced some issue which needs to be address latter:

* When we maximize the window, dot spacing & size should be increased. This part is pending for next release. In current solution, grid present at the top corner.



* When the mouse released outside the WIDGET, what action we need to take, is missing in problem.

Action Like: We have to show the maximum circle across that origin.

In my solution for these cases I did not create the circle, logging the reason as ***“Radius is higher”***.

* API::gultMotionFunc is not available, because of that we are not able to show the circle while dragging the mouse. We depends on mouse released (GLUT\_UP) to calculate radius.
* No API for changing the pen (thin/thick). To make thick line I make three circles nearby.

## How to Test:

This solution is developed on Visual Studio 2015 & compatible to Visual Studio 2013.

To test this solution on windows, I run the few scenarios:

* Run the solution, press the mouse RIGHT Key, after release the mouse nothing happens on the WIDGET.
* Press Mouse Left Key, Release at same place. No Circle forms on window as radius is zero.
* Press Mouse Left Key, drag to some place within widget. Three circles are formed. Two with thin RED line & one with thick BLUE line.
* Press Mouse Left Key, drag to outside the widget. No circles are formed.

# Program - Generates best-fit Circle:

In this question we have to use the mouse click event for toggling the Points on Widget.

And by providing the **KeyPress** event we need to perform certain things.

* When “G” is pressed best-fit circle creates.
* When “C” is pressed, it clears the widget.

## Design & Algorithm:

To solve this problem, we must consider the following mathematical equations:

**R = sqrt((x1-x2) ^2 + (y1-y2) ^2)**

Where R = Radius of Circle,

(x1, y1) – Origin of circle,

(x2, y2) – Any point at circumference of circle.

& well know modified Kaden’s algorithm to get the maximum area with largest BLUE dots.

Note: Kaden’s algorithm is for the max area in 1D.

### List of Classes used:

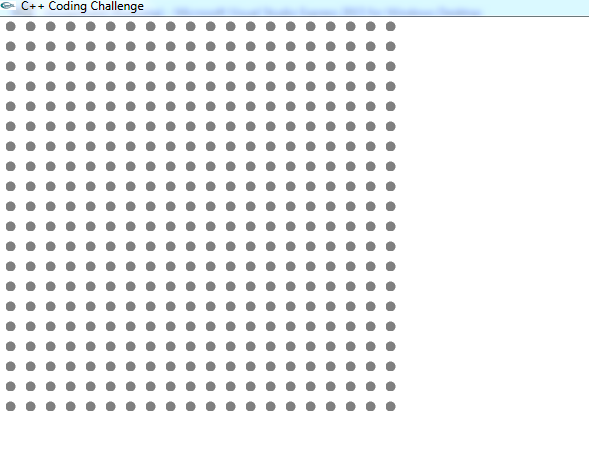
There are two classes used, in this class Canvas & Class MyCanvas.

Same as above.

## Improvements considered but not implemented:

While doing this task, I faced some issue which needs to be address latter:

* When we maximize the window, dot spacing & size should be increased. This part is pending for next release. In current solution, grid present at the top corner.



* If the selected dots are grouped in two parts located at the different position, in this best fit circle is present only in larger group. There is an ambiguity in requirement.
* Color of the best fit circle is not defined in requirement, I assume it RED.
* What happens when ‘g’ or ‘c’? I do the same thing what ‘G’ or ‘E’ doing.

## How to Test:

This solution is developed on Visual Studio 2015 & compatible to Visual Studio 2013.

To test this solution on windows, I run the few scenarios:

* Run the solution, press the mouse RIGHT Key, after release the mouse nothing happens on the WIDGET.
* Press Mouse Left Key at a Dot or on white space, color of dot changed to “BLUE”.

If we click the same Dot, it become grey again.

* Keep selecting the Dot by **Mouse Left key** press event & than provide the keyboard event “G”, a circle creates in RED.
* Any time, press the “C”, all dots on widget become **GRAY,** if a circle presents it is also removed.

# Program - Generates best-fit Eclipse:

Everything is same as the chapter 2 (Generates best-fit Circle), only we need to modify the base class Canvas. Added one API to create the Eclipse.

var x = h + r\_width\*Math.cos(theta) ;

var y = k - 0.5 \* r\_height\*Math.sin(theta) ;

*where (x,y) is coordinates on circumference*

*(h,k) : coordinates at origin*

*Math = Constant= sqrt(2)*

*theta : angle*

*r\_width = rectangular width*

*r\_height = rectangular height*

For Reference: <http://www.mathopenref.com/coordcirclealgorithm.html>

## How to Test:

To test the eclipse, instead of ‘G’ or ’g’, press ‘E’ or ‘e’ . Other conditions are same.