President University

Report of Programming Assignment 3

Computer Graphics and Animations

Ida Bagus Bhaskara (001201500076)

Rahmad Martin (001201500033)

Vera Debora Vitamas (001201500076)

CIT 2 2015

**Table of Contents**

Title 1

Table of Contents 2

[I. Introduction 3](#_Toc495226771)

[II. Basic Theory 3](#_Toc495226772)

[III. Implementation 3](#_Toc495226773)

[IV. Design 3](#_Toc495226774)

[V. Evaluation 3](#_Toc495226775)

[VI. Work Log 3](#_Toc495226776)

[VII. Conclusion and Remarks 3](#_Toc495226777)

# **Introduction**

To display objects in a computer screen, certain pixels on the screen must be filled to represent the object on the screen. Certain algorithms need to be executed to avoid the user to fill the pixels themselves. Taking circles and ellipses as the main object focus of this program, the midpoint algorithm will be the algorithm to determine which pixels are to be set. Other algorithms exist to determine the pixels, but the midpoint algorithm is proven to be able to correctly determine the pixels. This will be covered further in the Basic Theory section.

This program was created using Visual Basic programming language. In this report, we cover: basic theory, how to use the application, evaluation of the main features, explanation of data structure, work log, conclusion and remarks.

# **Basic Theory**

In theory, there are several approaches to the algorithms used to determine the set pixels of a circle. One of them is by using the mathematical formula of finding a circle’s area, which the algorithm is as follows:

x traversal [0..R]

y = round(sqrt(R\*R – x\*x))

Set(xc+x,yc+y)

Considering that it’s how we do it mathematically, it should work. But in reality, this will not generate an incomplete circle with missing pixels and moreover, straight lines which means that this approach does not generate a circle.

Using the midpoint algorithm, however, the correct circle set pixels can be determined. This is also true for the ellipse set pixels. The algorithm is as follows:

x 🡨 0; y 🡨 r; d 🡨 1 - r

Iterate

Set(xc + x, yc + y)

Stop x ≥ y

If d < 0 then {choose R}

d 🡨 d + 2 \* x + 3

Else {d ≥ 0, choose DR}

d 🡨 d + 2 \* (x – y) + 5

y 🡨 y – 1

x 🡨 x + 1

The algorithm starts by setting the pixel as shown by Set(xc + x, yc + y). It begins with positioning the set pixel at the start position. Then, it continues until x is equals to or bigger than y. When that happens, it should choose between R(Right) or DR(Down Right) depending on the calculated d. But beforehand, the d is recalculated at the variable assignment as d 🡨 1 – r. It then checks whether d < 0 or d ≥ 0. If it’s d < 0, then it will choose R, recalculates the d by doing d + 2 \* x + 3, and increments x by 1, leaving the y untouched. If it’s d ≥ 0, however, then it will choose DR, recalculates d by doing d + 2 \* (x – y) + 5, and increments x by 1, also decrements y by 1.

For an ellipse we have to divide an ellipse into 4 quadrants and The curve is divided into two regions. In region I, the slope on the curve is greater than –1 while in region II less than –1. The algorithm to draw ellipse is as follows :

Procedure DrawEllipse(xc : integer,

yc : integer, a : integer, b : integer)

//as the input

For region I :

x 🡨 0; y 🡨 b

d 🡨 4b2 – 4a2b + a2

Iterate

Set(xc+x,yc+y)

Stop 2b2(x + 1) ≥ a2(2y – 1)

If d > 0 then {choose DR}

y 🡨 y – 1

d 🡨 d + b2(8x+12)+a2(8-8y)

Else {choose R}

d 🡨 d + b2(8x+12)

x 🡨 x + 1

For region II :

d 🡨 b2(2x+1)2 + 4a2(y-1)2 – 4a2b2

While y>0 do

If d < 0 then {choose DR}

x 🡨 x + 1

d 🡨 d + b2(8x+8)+a2(12-8y)

Else {choose D}

d 🡨 d + a2(12-8y)

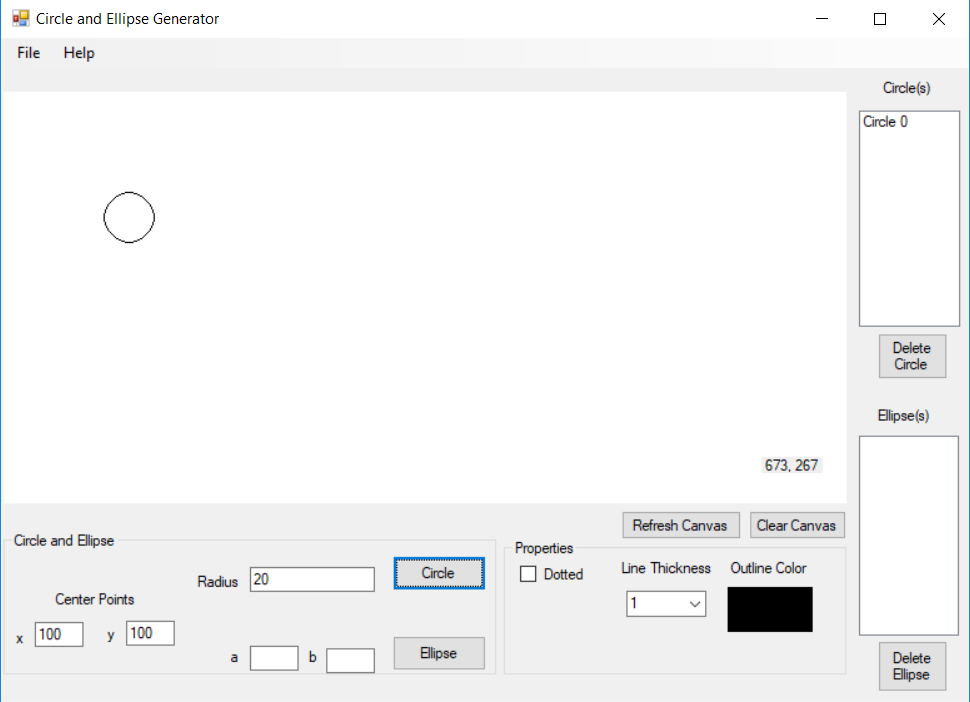
y 🡨 y – 1

Set(xc+x,yc+y)

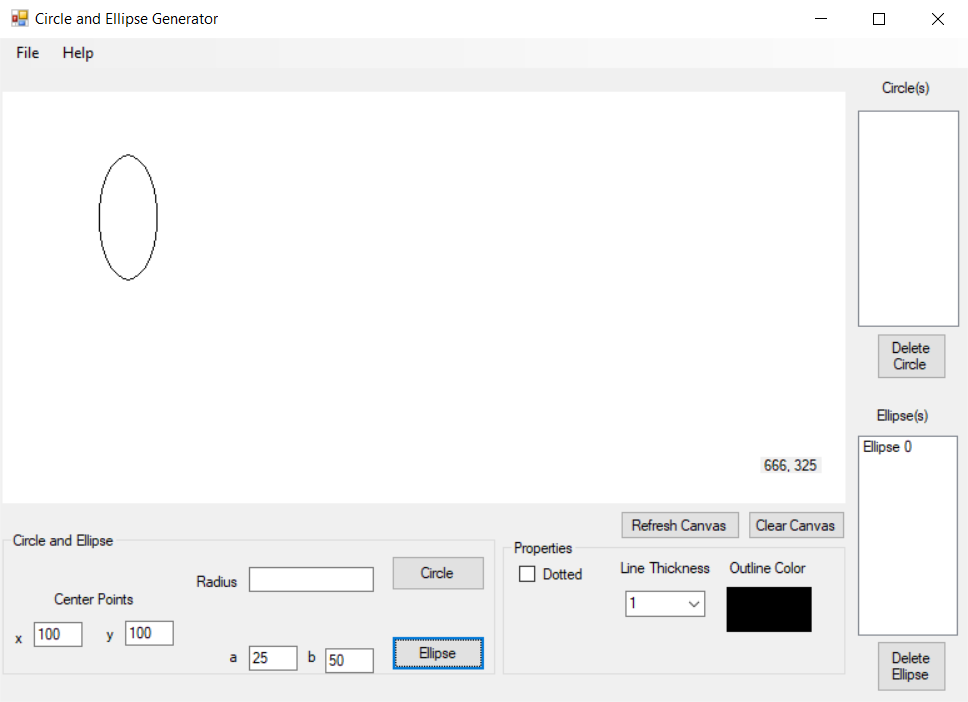
# **Implementation**

This is the implementation about how does the program works:

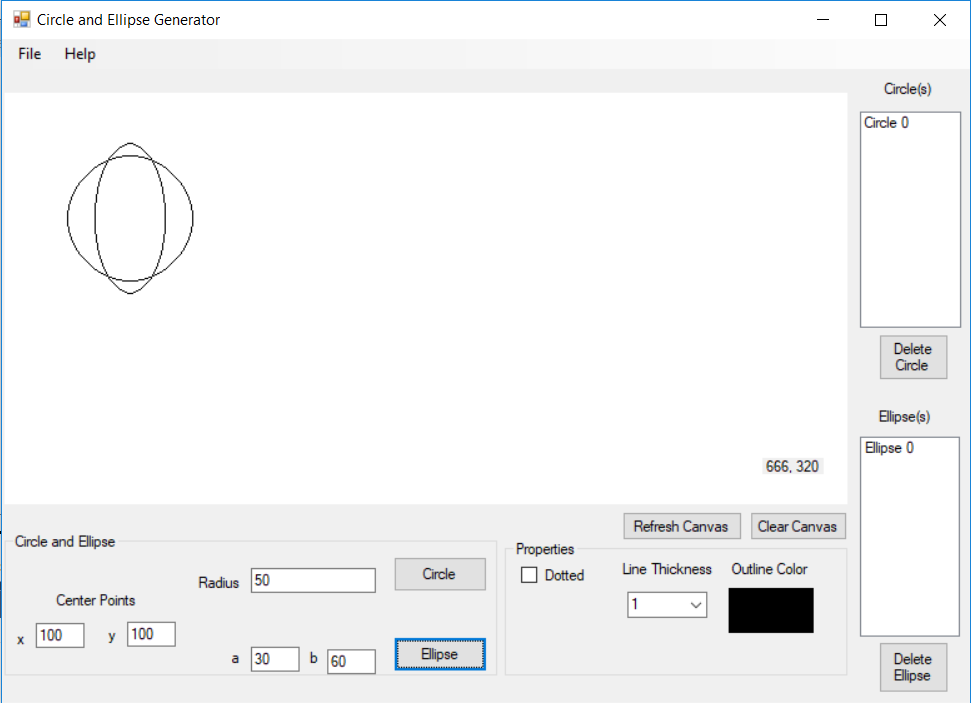
If we put some number inside x,y and the Radius and we click circle. The Output is:



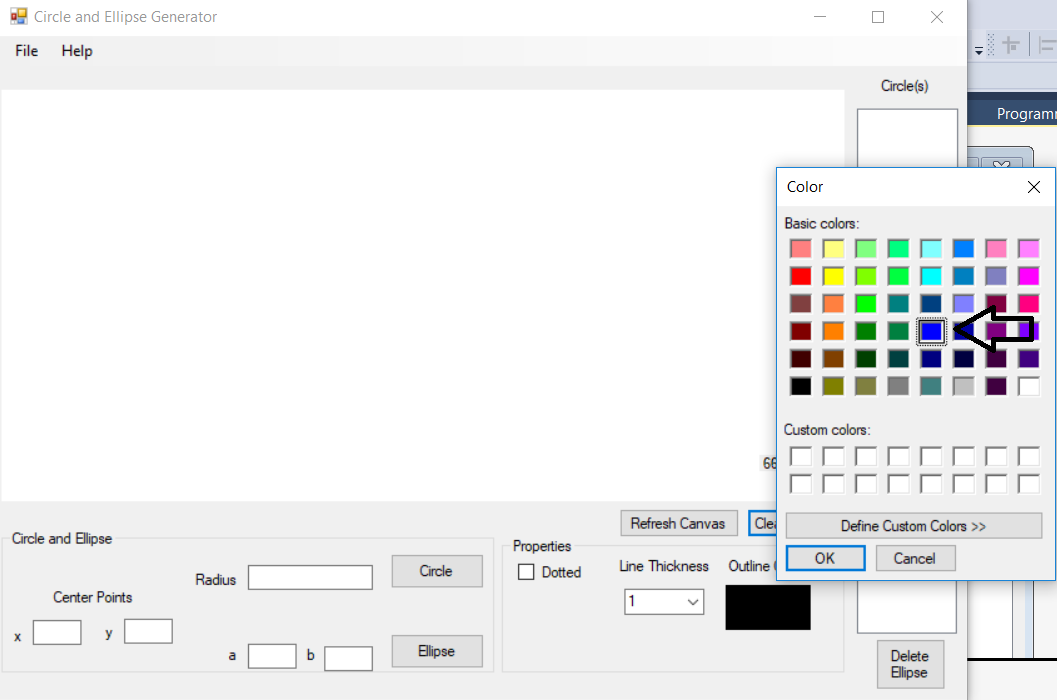
If we put some number inside x,y and a,b after that we click ellipse. The output is:



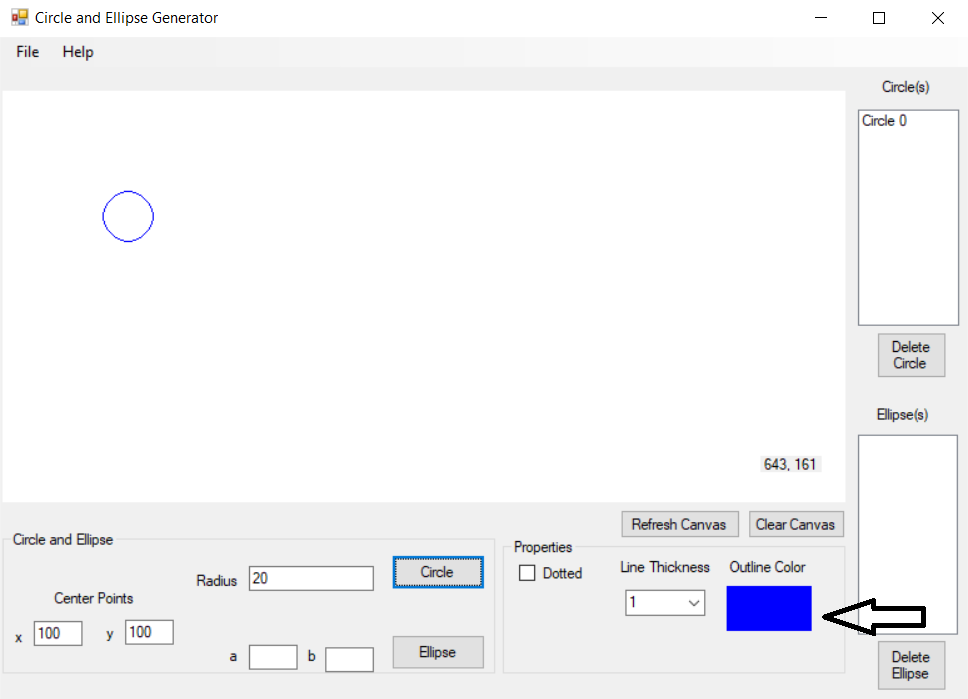
If we want to make a circle and an ellipse in one picture, we make a circle after that an ellipse. The output is:



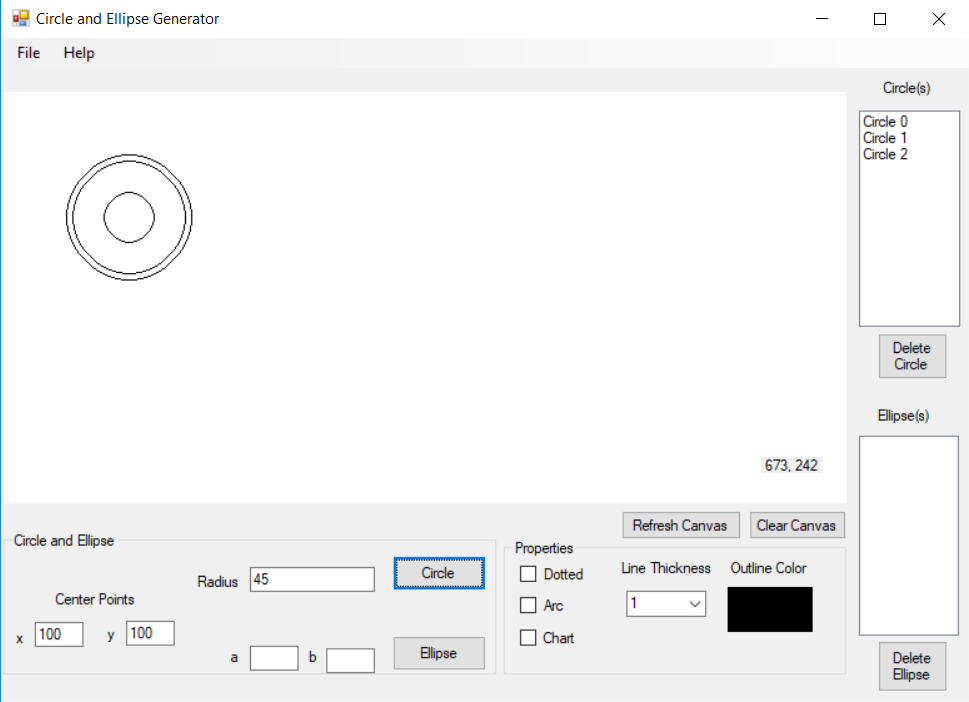
If we want to make a color for a circle or an ellipse you can choose the color first



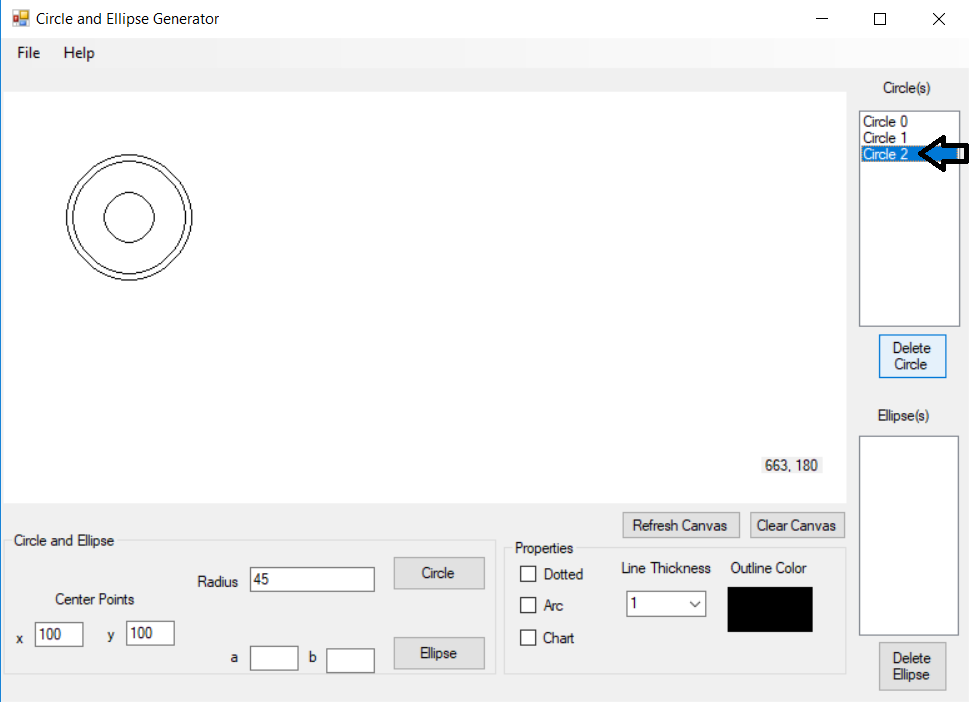
After that we choose the color now we make an ellipse or a circle, for an example we make a line of circle become blue. This is the output:



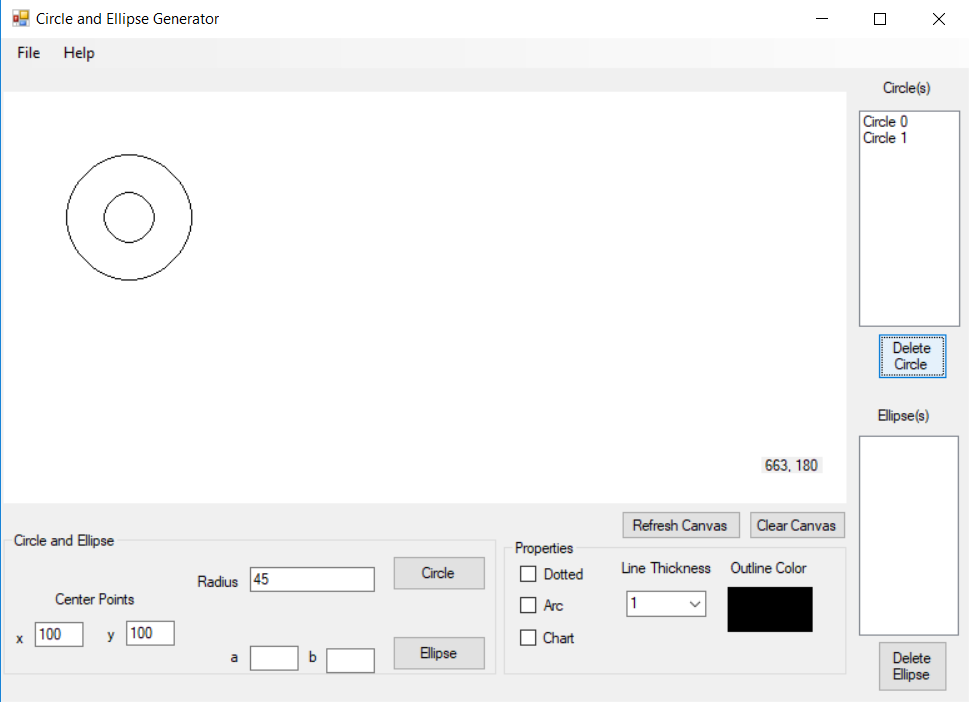
If we want to delete a circle(s) in one picture, although the circle(s) are stacked we still can delete it, but first after we draw a circle(s). This the output:



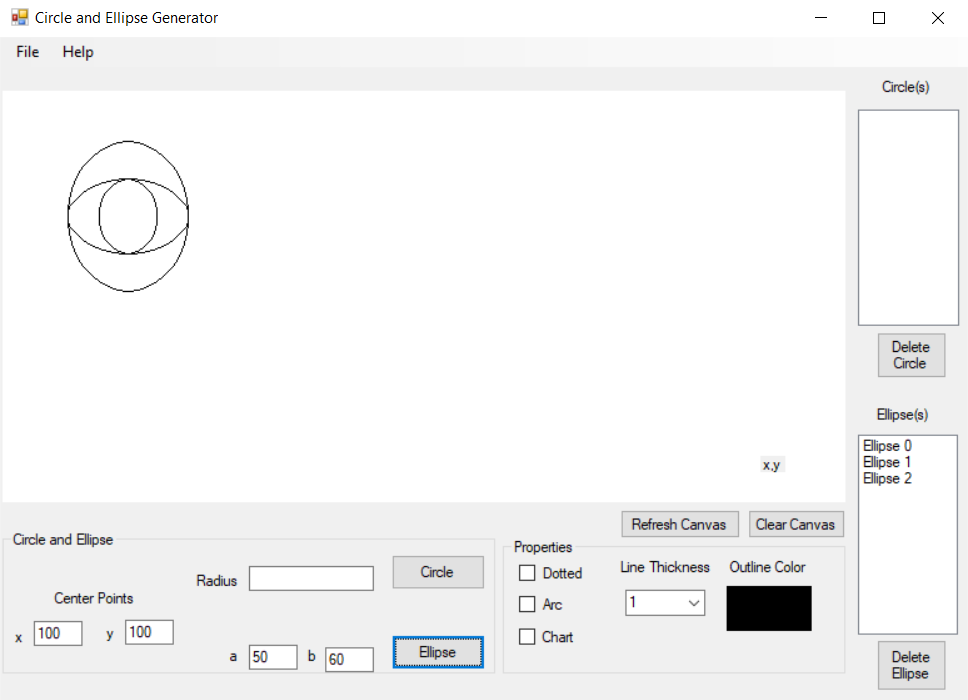
After we draw the circle(s) we have to choose the latest circle that we have been put. This is the output:



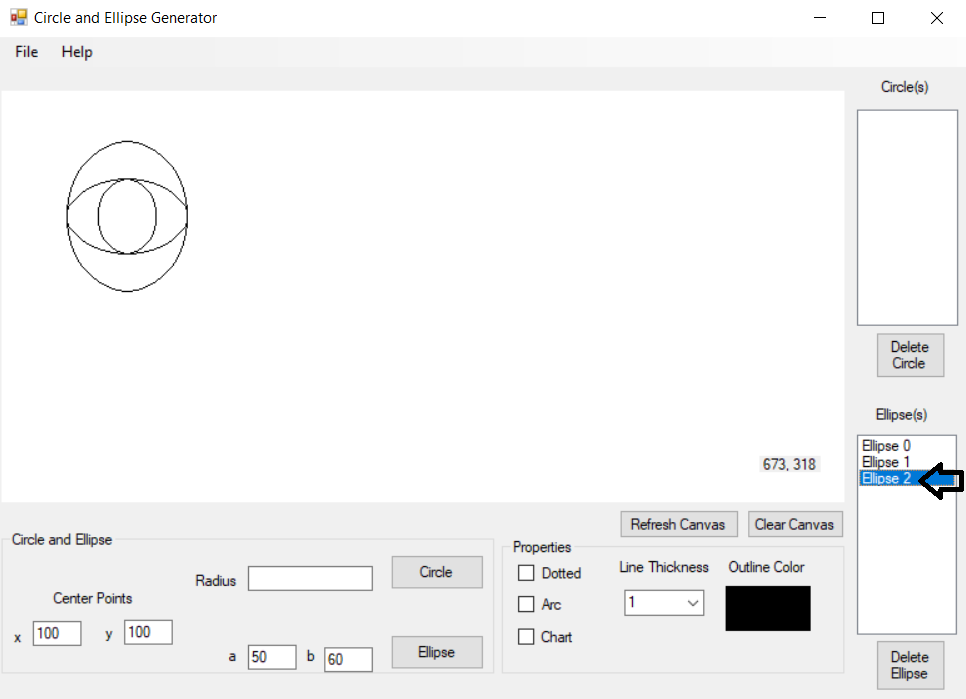
And it’s already deleted after we click the delete circle button. This the output:



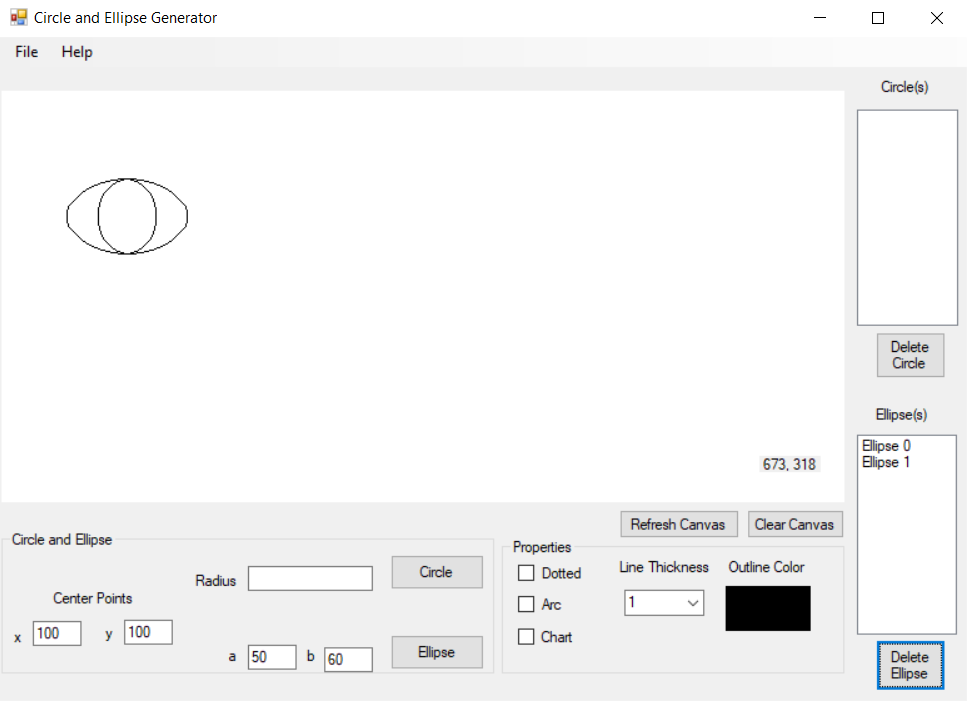
This is if we want to delete an ellipse like the same what circle did. First we have to draw an ellipse(s). This is the output:



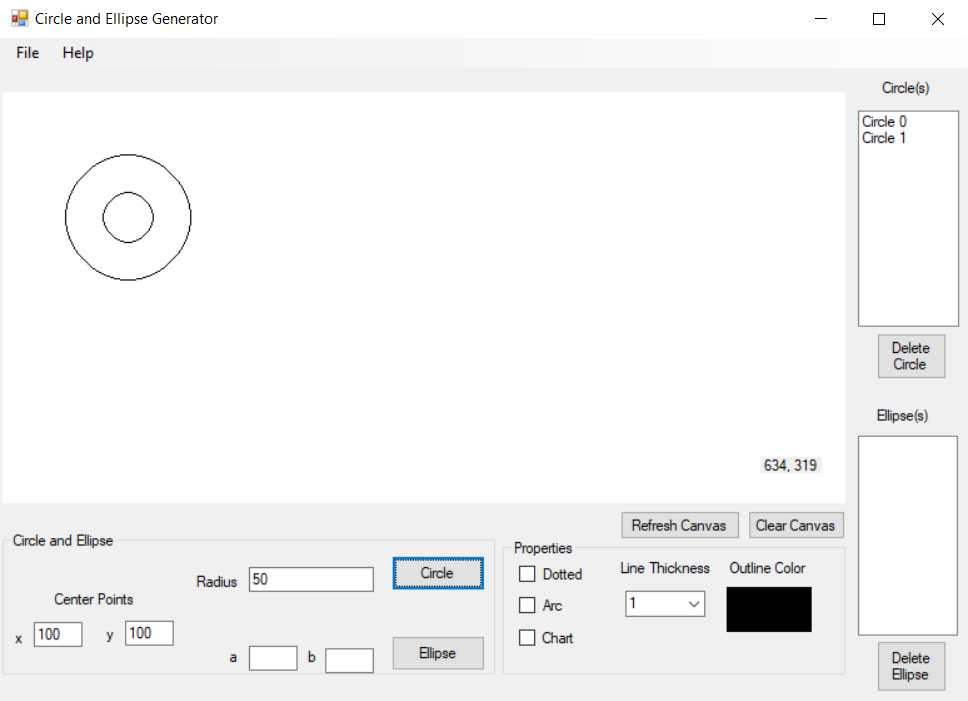
After we draw the ellipse(s) we have to choose the latest circle that we have been put. This is the output:



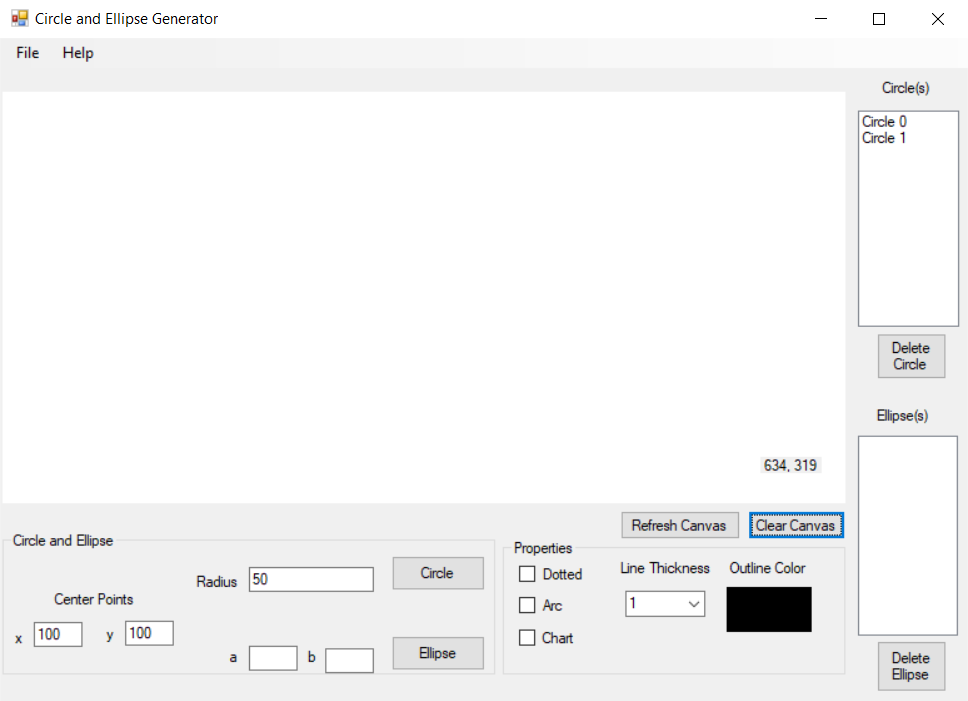
And it’s already deleted after we click the delete ellipse button. This the output:



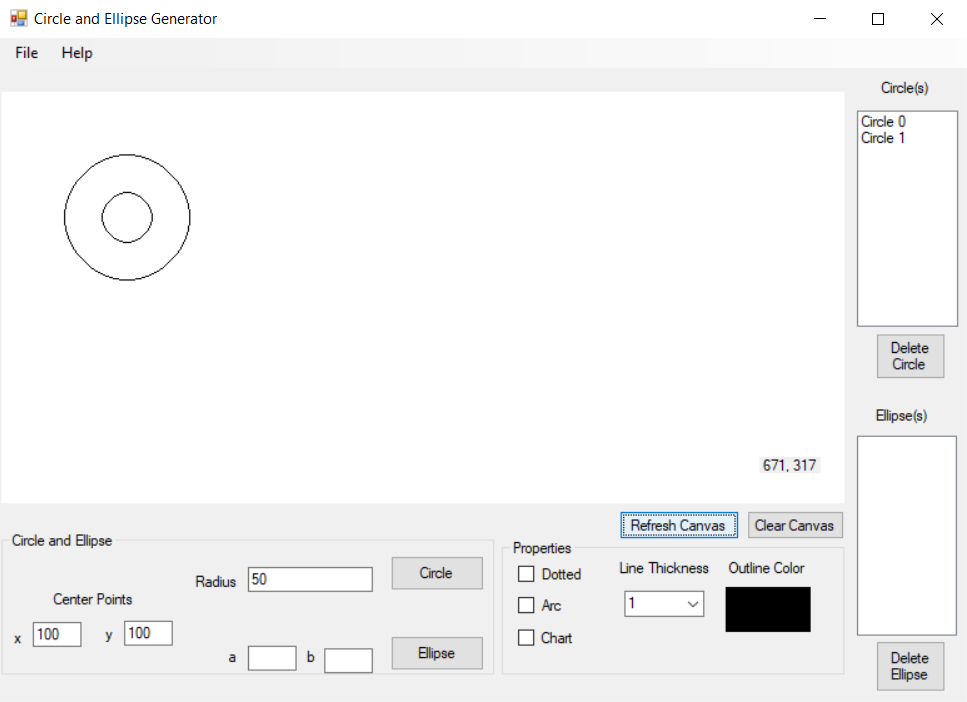
This program also has a refresh canvas. Before we use the refresh canvas we have to draw something on Picture Box. For example we draw a circle(s). This the output:



And this what happen if we click the button Clear Canvas. This is the output:

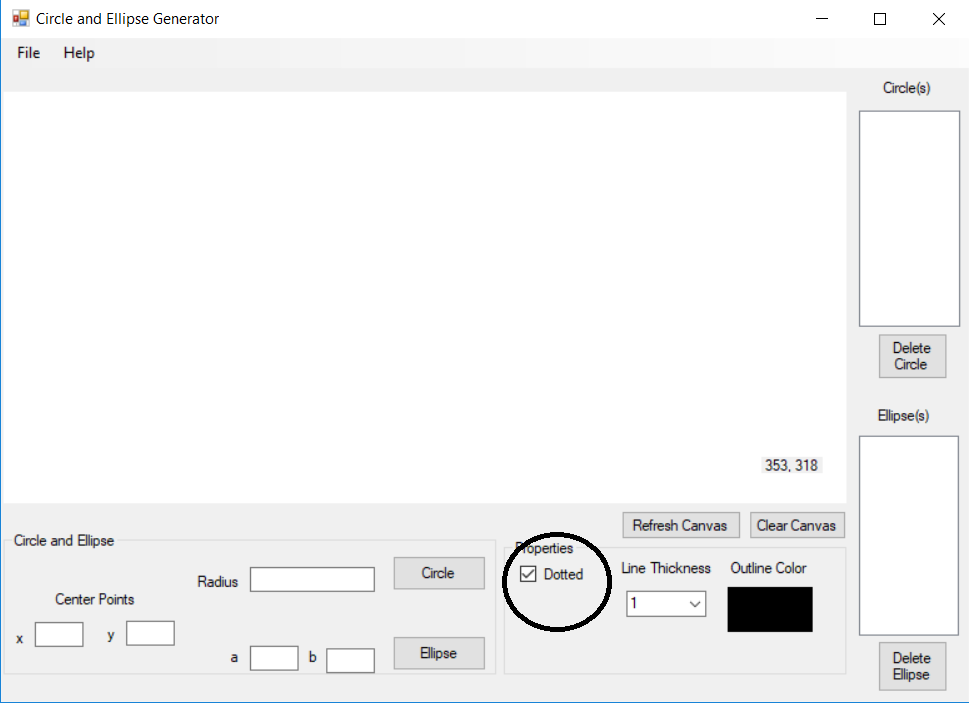


There is also a refresh button after we clear the canvas we also can refresh it, and we can again all the circle(s) that have been deleted. This is the output:

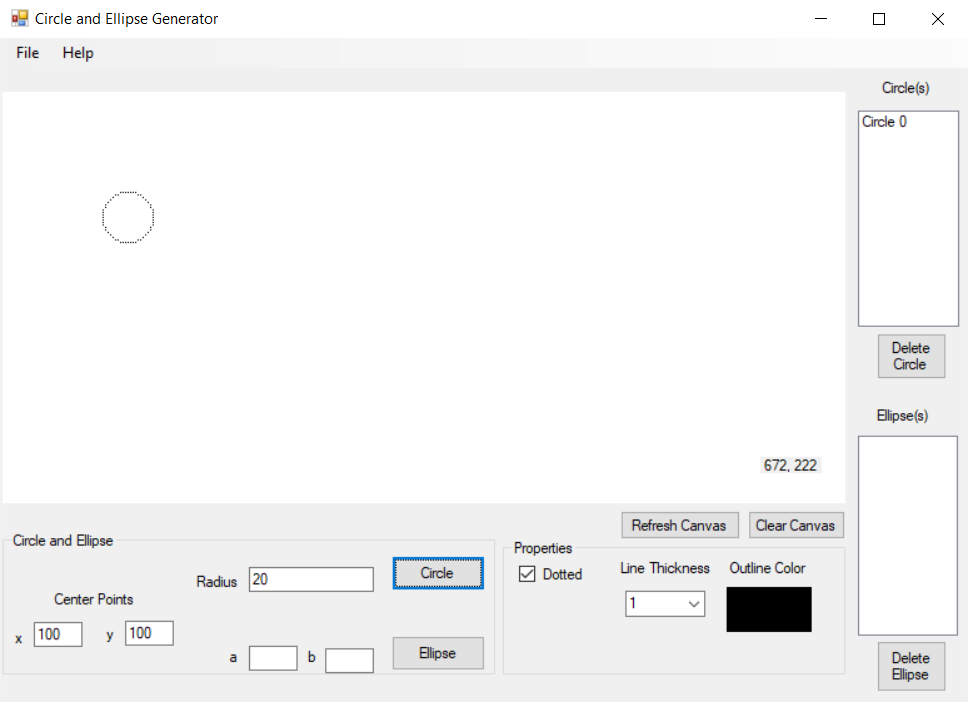


This program also can make a dotted. For example if we want to make a dotted circle.

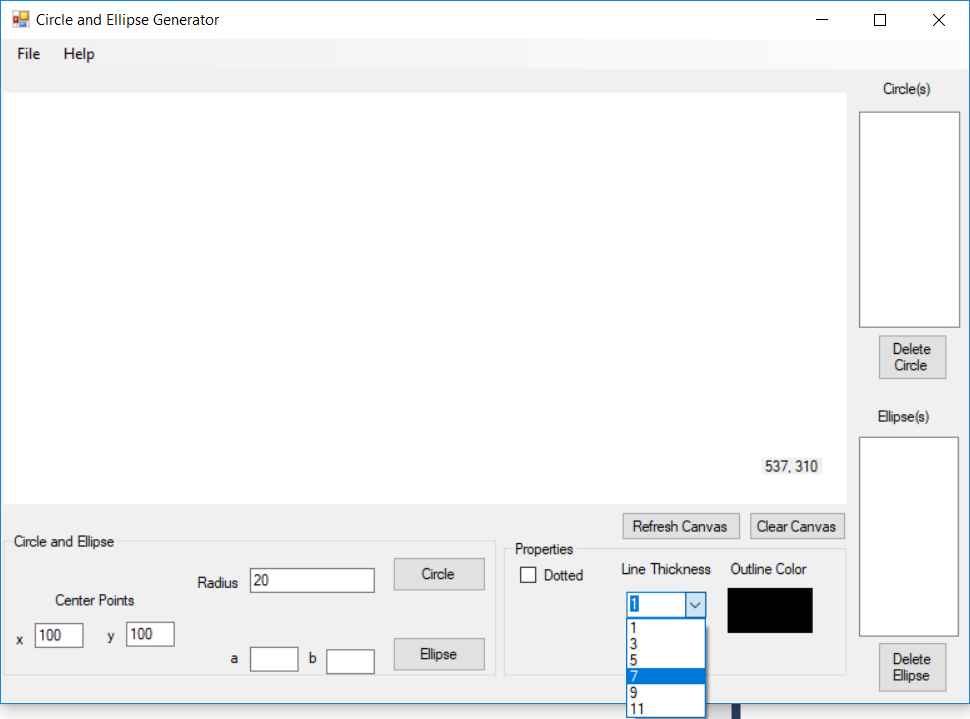
First we have to check the check box on dotted. This is for an example:



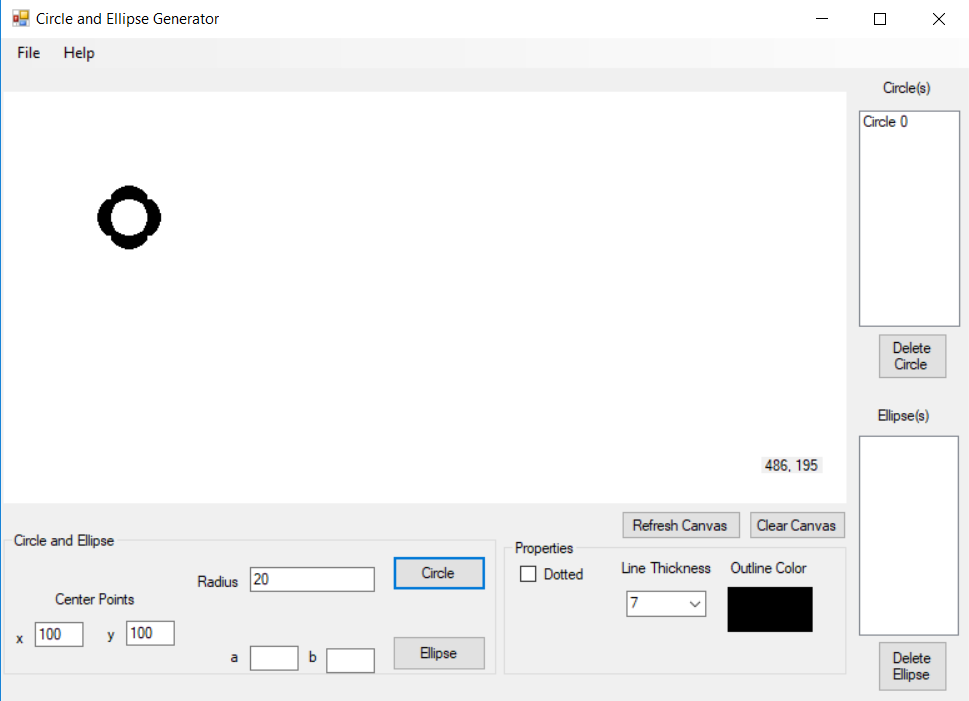
Now we see the line of the circle is already dotted. This is the output:



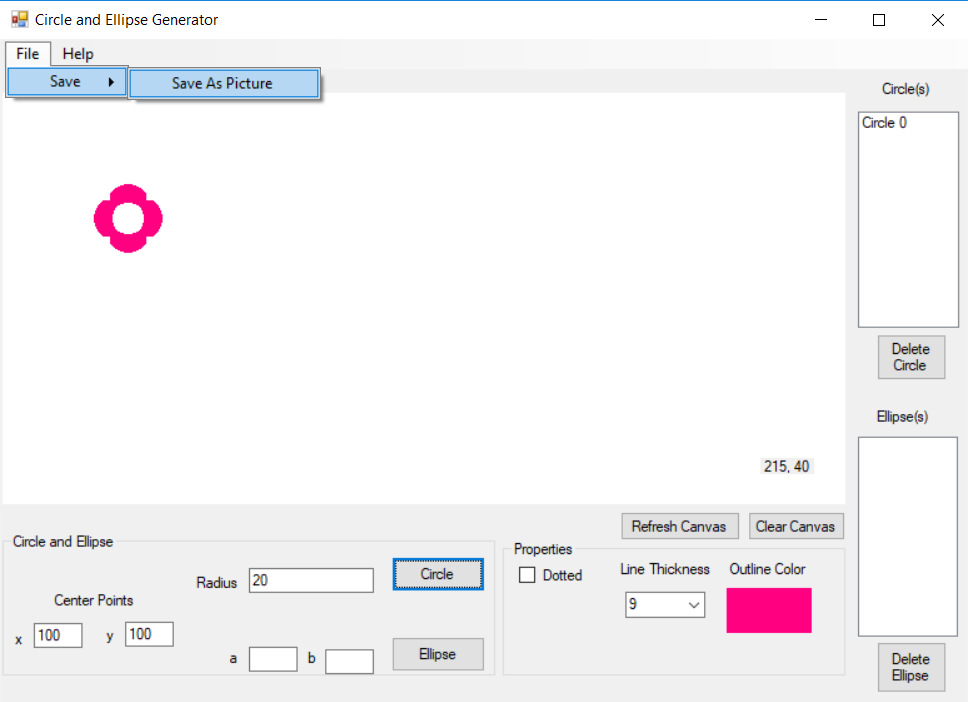
We also can make the line becom thick. We choose making a circle for example, firs we set x,y and also the radius, after that we set the line thinkness. Like this:



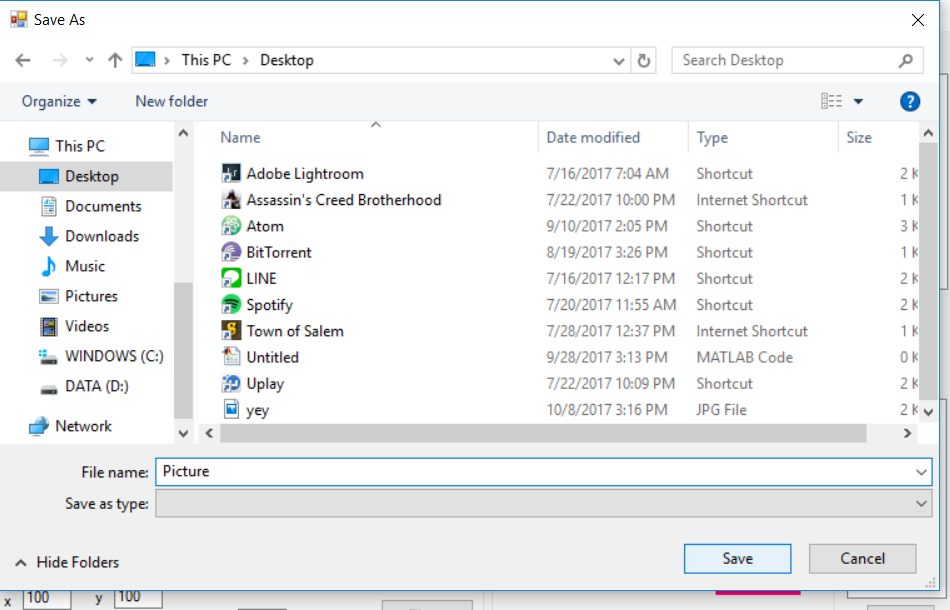
After we set it all. The output gonna be:



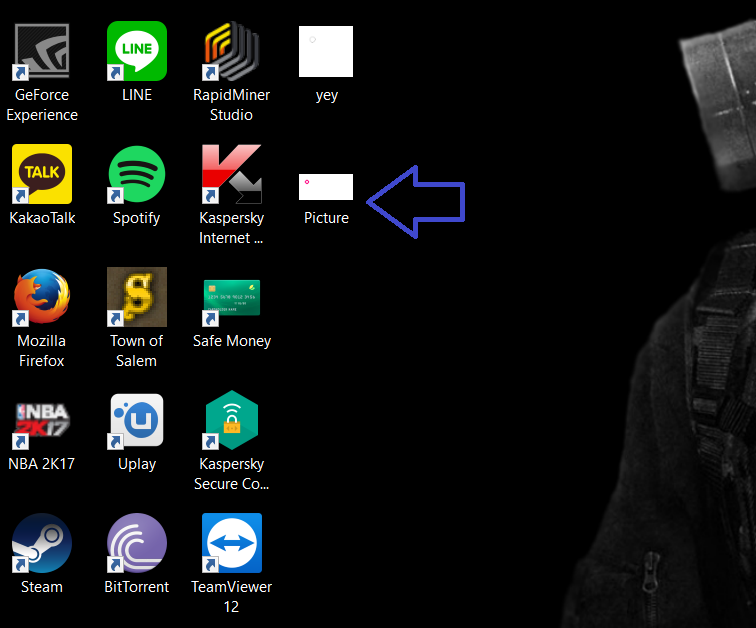
We also can save the picture, what we have been make on the Picture Box. Like this:



After we click Save As Picture, it’s will shows where do we want to put the picture. Like this:



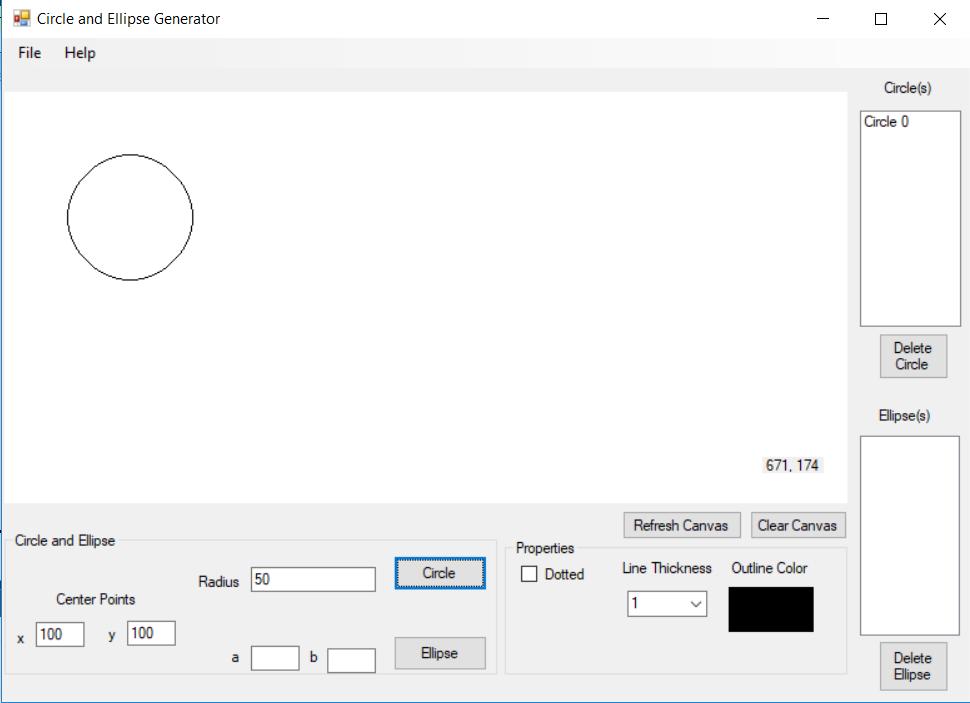
We put it on the desktop, and it will shows like this:



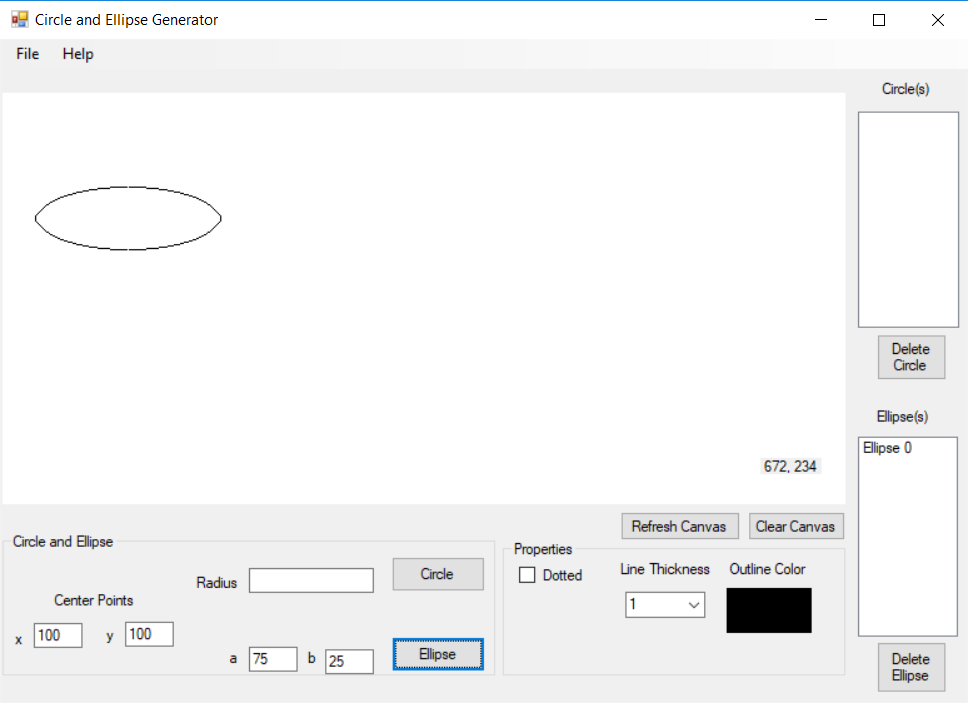
# **Design**

# **Evaluation**

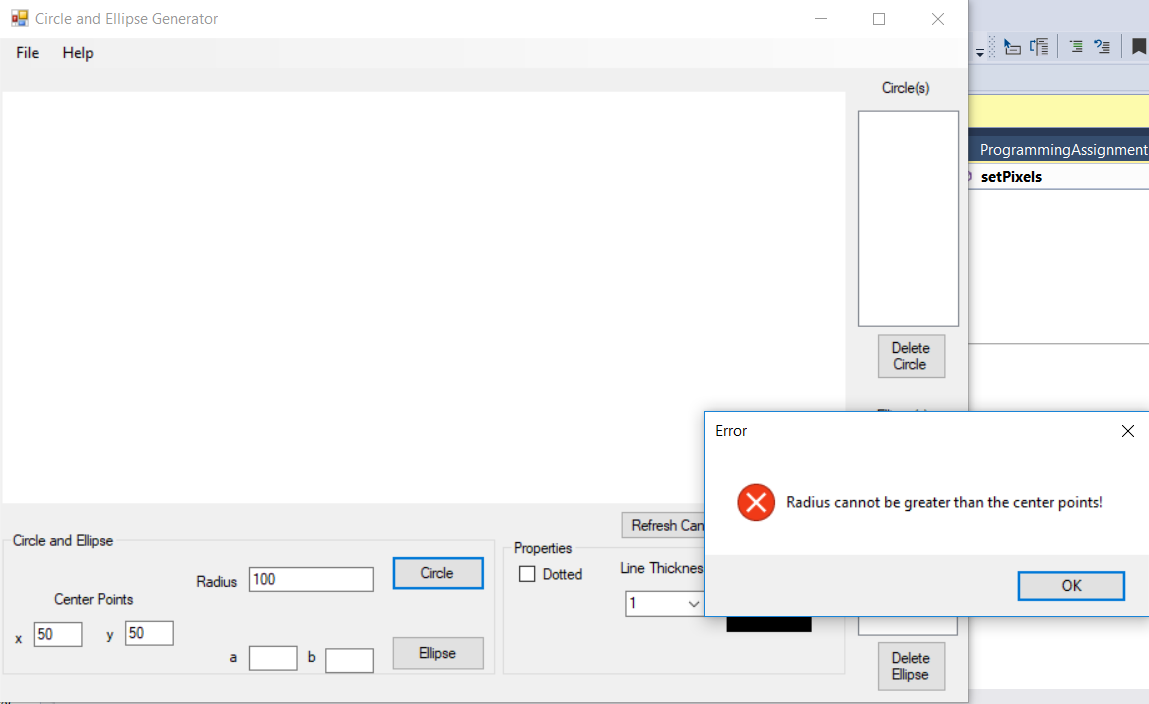
* 1. Circle(100, 100, 50)



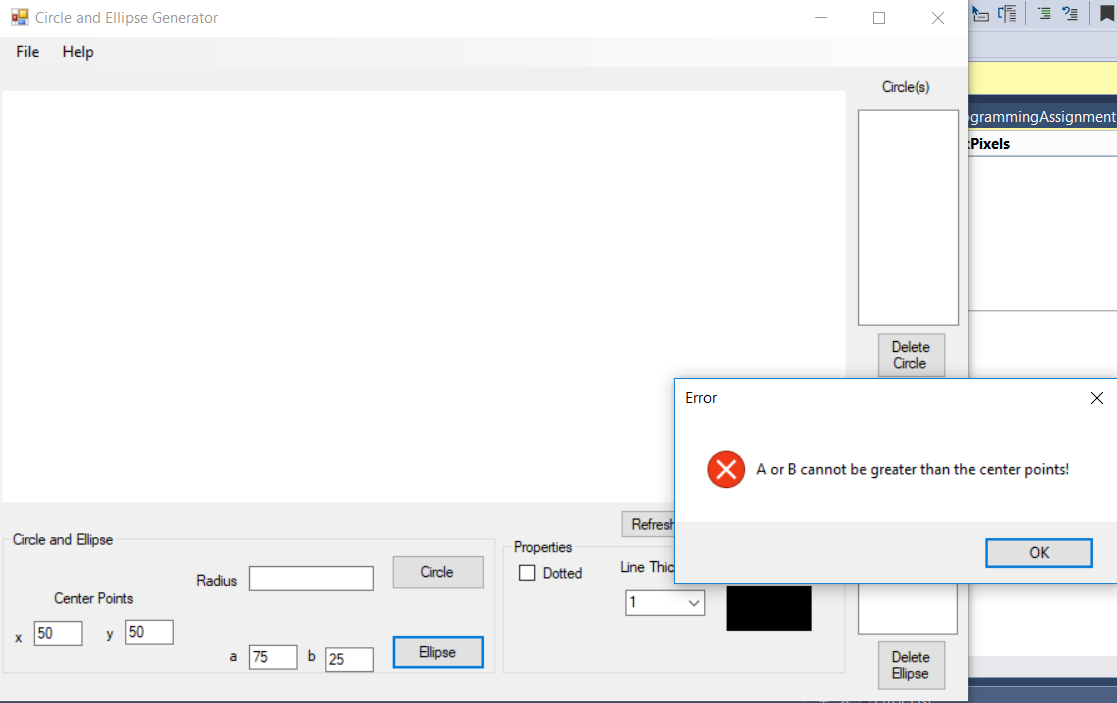
* 1. Ellipse(100, 100, 75, 25)



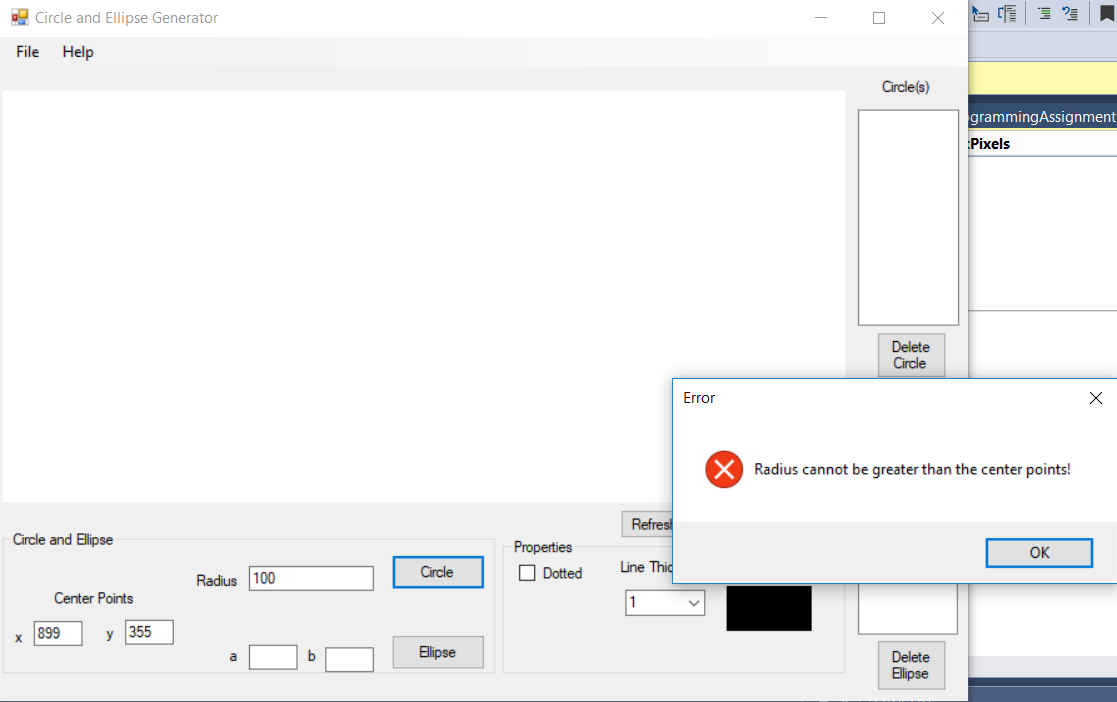
* 1. Circle(50, 50, 100)



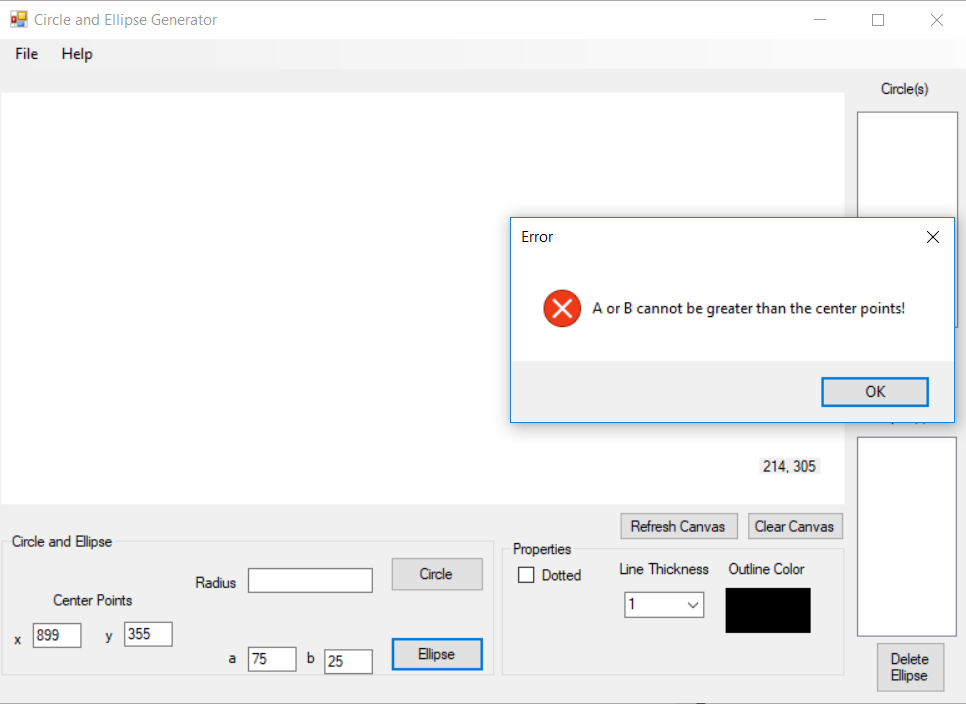
* 1. Ellipse(50, 50, 75, 25)

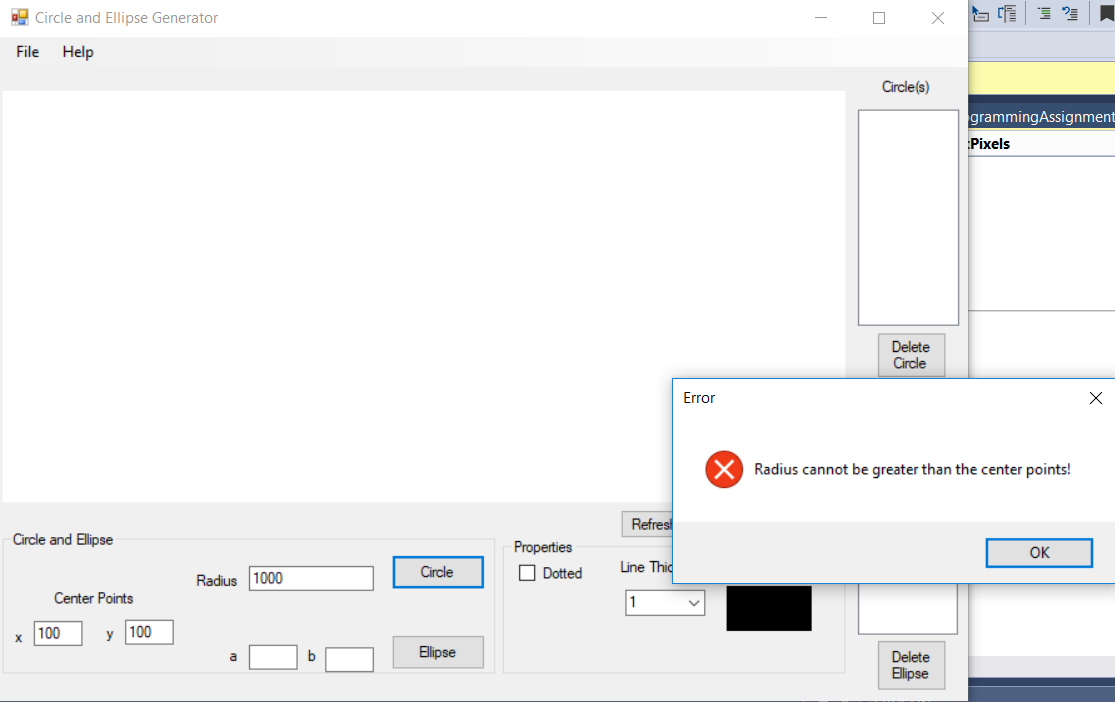


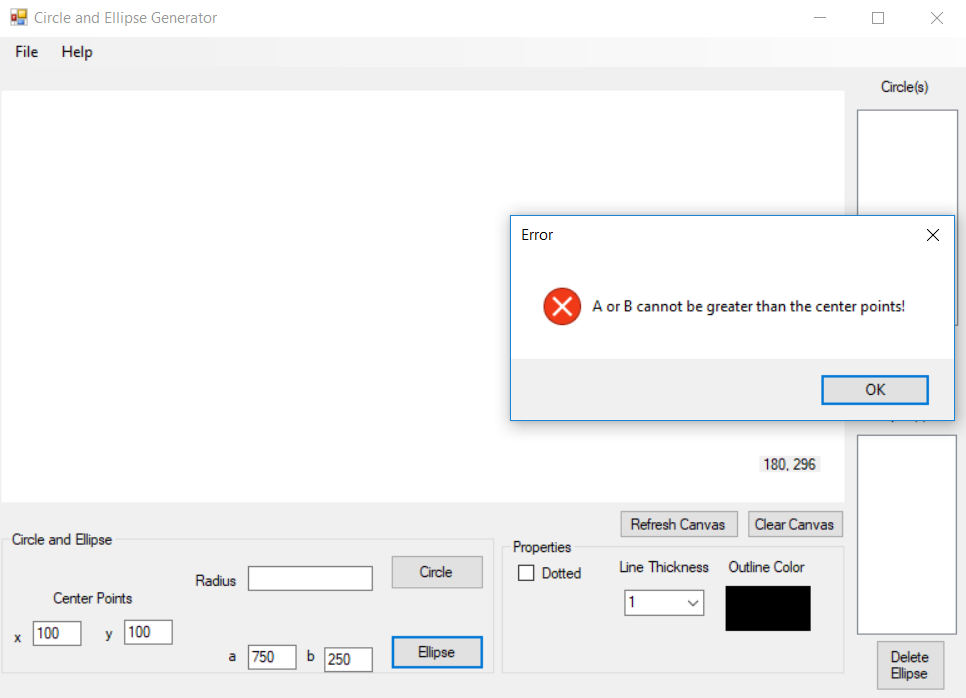
* 1. Circle (*screen width* - 50, *screen height* - 50, 100)



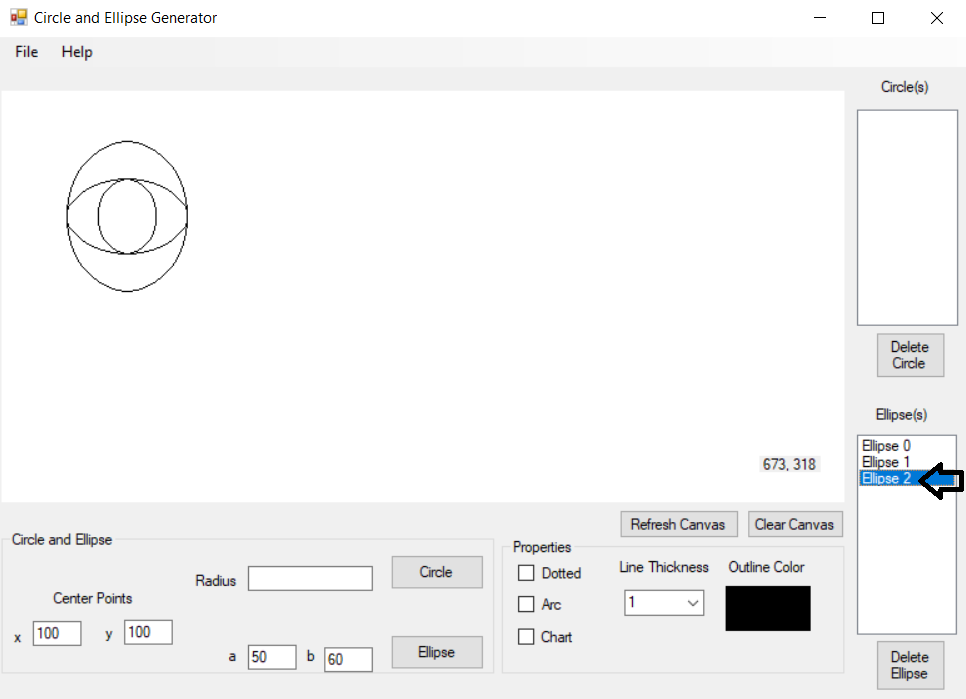
* 1. Ellipse (*screen width* - 50, *screen height* - 50, 75, 25)

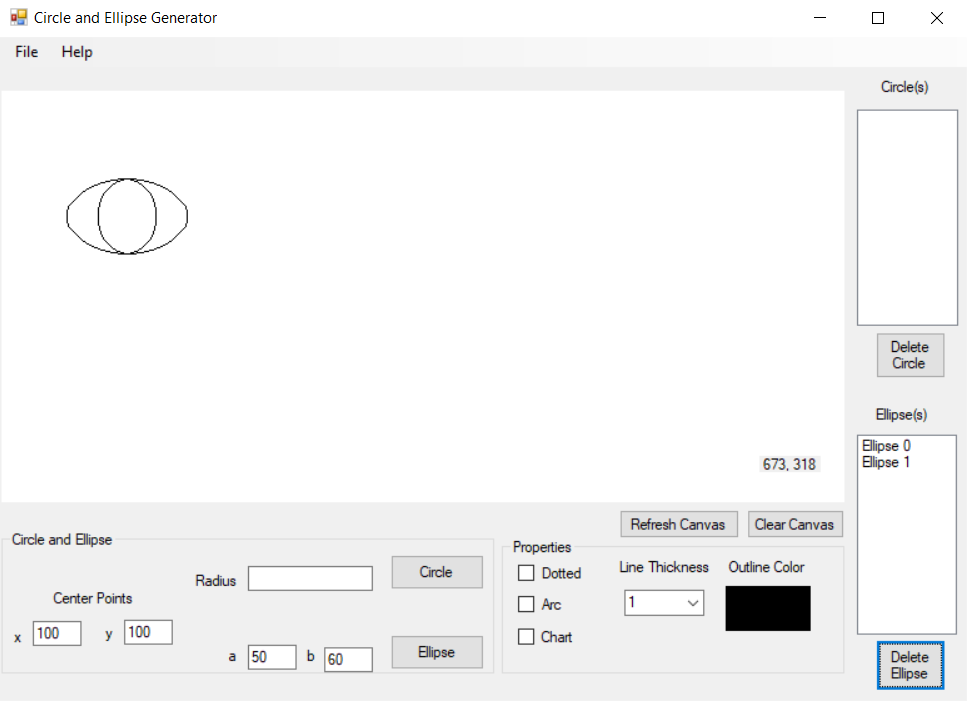


* 1. Circle (100, 100, 1000) (set the radius to a high number)
  2. Ellipse (100, 100, 750, 250)



* 1. Delete a circle/ellipse.





* 1. Also perform a test case for all the bonuses you implemented.

(already explain on Implementation)

* 1. Include screenshots of each test case.

(already explain on Implementation)

* 1. Explain whether each case is successful.

(already explain on Implementation)

# **Work Log**

The work log is extracted directly from Visual Studio’s Git Log History, which is also available publicly at <https://github.com/bakanui/Drawing-Circles-and-Ellipses/commits/master>.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Author | Date | Time | Commit Message |
| 1 | Bhaskara Ida Bagus | 9/25/2017 | 5:33:41 PM | Initialized project |
| 2 | Bhaskara Ida Bagus | 9/25/2017 | 5:37:28 PM | Add PictureBox canvas in Form |
| 3 | Bhaskara Ida Bagus | 9/25/2017 | 9:16:28 PM | Added the ability to draw circle (values are hard-coded) |
| 4 | Rahmad Martin | 9/25/2017 | 10:48:18 PM | #Help, fail ellipse :( |
| 5 | Bhaskara Ida Bagus | 9/26/2017 | 1:38:14 AM | Housekeeping, user input for circle center points and radius (no longer hardcoded) |
| 6 | Rahmad Martin | 9/26/2017 | 12:33:17 PM | #Fix ellipse + input and button |
| 7 | Vera Debora Vitamas | 9/28/2017 | 11:53:32 PM | Creating an array for circle so user can choose to delete any circle. Still buggy, not working as intended |
| 8 | Vera Debora Vitamas | 10/1/2017 | 1:44:15 PM | Trying to put circle values on a List(Of Integer) |
| 9 | Bhaskara Ida Bagus | 10/1/2017 | 7:46:55 PM | Now able to put circles on a list |
| 10 | Vera Debora Vitamas | 10/1/2017 | 8:52:47 PM | Change Circle Color |
| 11 | Bhaskara Ida Bagus | 10/2/2017 | 12:31:37 AM | Circle can now be removed on list |
| 12 | Bhaskara Ida Bagus | 10/5/2017 | 10:30:49 PM | Updated listing algorithm to a much more simpler version |
| 13 | Vera Debora Vitamas | 10/6/2017 | 4:55:02 PM | Trying to make save/load from or to a file |
| 14 | Vera Debora Vitamas | 10/6/2017 | 8:53:01 PM | Making save/load file using an image and it works |
| 15 | Bhaskara Ida Bagus | 10/7/2017 | 3:55:33 AM | Deletion works but only allows bottom-most object to be deleted |
| 16 | Rahmad Martin | 10/7/2017 | 12:58:27 PM | #ellipse can now be deleted #but we can't delete the object when we had both circle and ellipse |
| 17 | Rahmad Martin | 10/7/2017 | 11:42:59 PM | #nomore missing pixel on top+bottom of ellipse #add dotted properties #add cursor position |
| 18 | Vera Debora Vitamas | 10/8/2017 | 11:47:05 AM | Added report document and title page |
| 19 | Bhaskara Ida Bagus | 10/8/2017 | 12:00:05 PM | Added table of contents and introduction in report |
| 20 | Vera Debora Vitamas | 10/8/2017 | 3:20:39 PM | trying to make an interface more friendly |
| 21 | Bhaskara Ida Bagus | 10/8/2017 | 4:26:50 PM | Updated report’s Introduction, Basic Theory and Worklog |
| 22 |  |  |  |  |
| 23 |  |  |  |  |
| 24 |  |  |  |  |
| 25 |  |  |  |  |
| 26 |  |  |  |  |
| 27 |  |  |  |  |

# **Conclusion and Remarks**