**Vector Based Drawing Application**

***Design :***

First of all we create the interfaces "Shape" and "DrawingEngine" given in the project's description then, we create "TheShape" class that implements the interface "Shape" then, creating class "Engine" that implements "DrawingEngine" interface   
   
we create our shapes that need to be drawn in our application :

-Class Triangle

-Class Circle

-Class Rectangle

-Class Square

-Class Ellipse

-Class Line

-Class Plugin

And all this classes inherit from TheShape class.

***Engine:***

There will be a class called Engine that inherit from DrawingEngine interface.

***Canvas:***

Class Canvas extends JPanel and over ride paintComponant and it contains two methods one called draw and the ather called edit that the GUI call and they access the methods in Engine.

***GUI***

GUI calls canvas and the engine classes

Canvas: contains a method called paint component that draw all the shapes in the array list in the Engine   
and contains another method draw that create an object by reflection and add it to the array list of the shapes in the Engine, it also contains a method "edit" that get the shape that is selected by the user from the combobox in the GUI then, change the properties of the shape according to the user choices.

***Data Structure used:***

We used Liskedlists, arraylists , stacks, Hash Map.

***User Guide:***

->First: we have a button for each shape to draw it, when you click on any button of a shape, a dialog appears to set the position and the dimensions of this shape, you click agree to draw the shape on the screen.

->Second: we have buttons that control the operations on the shapes (move, delete, edit, colorize, copy, undo, redo):

#before using the previous buttons you have to choose a shape from the combo box from the drawn shapes.

-move button: you click on this button and set the new position of the shape you have drawn.

-delete button: that button deletes the shape chosen from the combo box.

-edit button: it takes the new position and the new dimensions of the shape chosen.

-colorize: that button give your shape a color, it has a dialog contains fill color and border color, on pressing a button from these two buttons the color dialog appears and you choose a color.

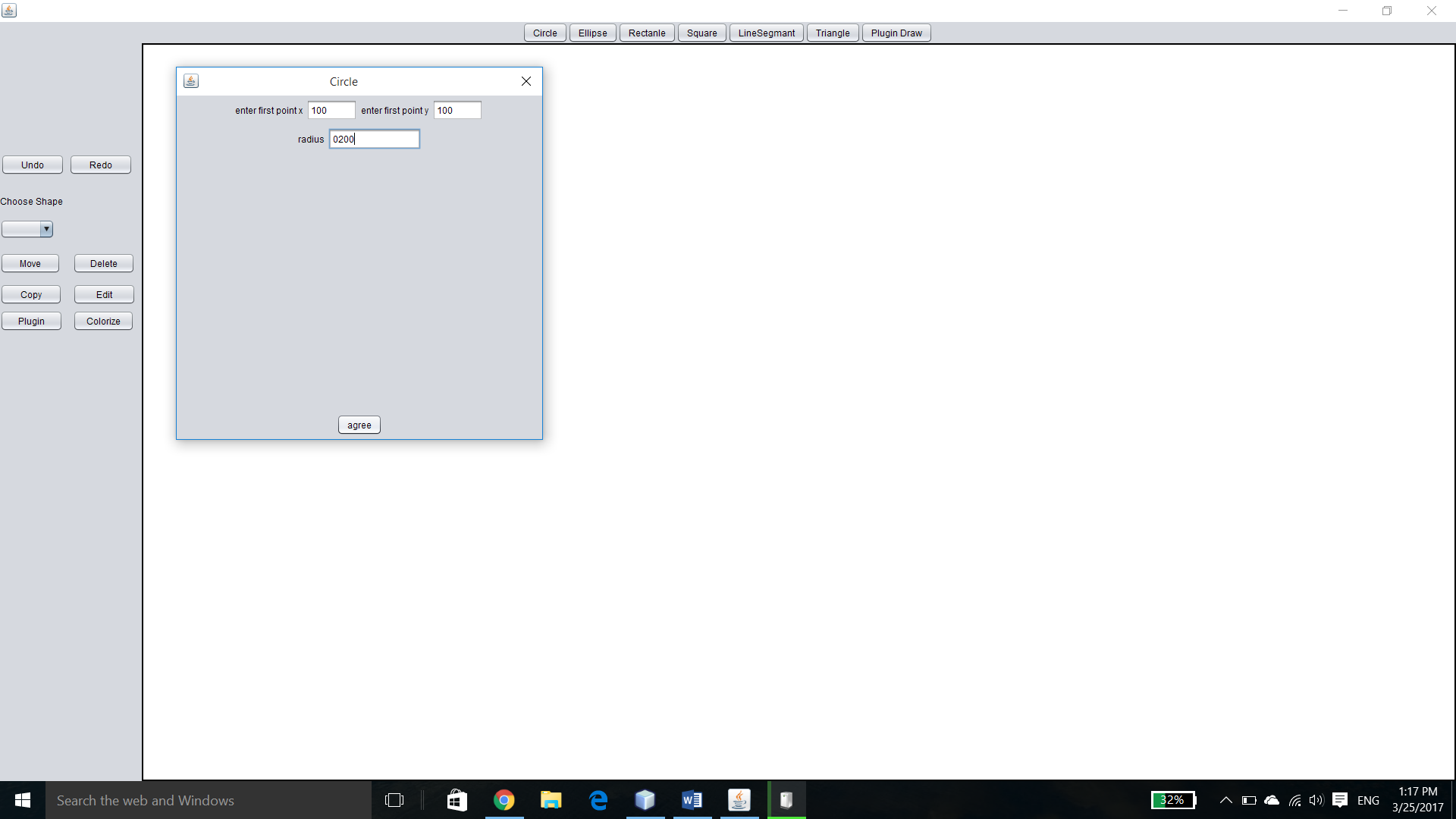
-copy: it copies the shape by taking the new position of the new shape and the old one still on the screen.

-undo: it gets the last action you have done (delete a shape or add the shape).

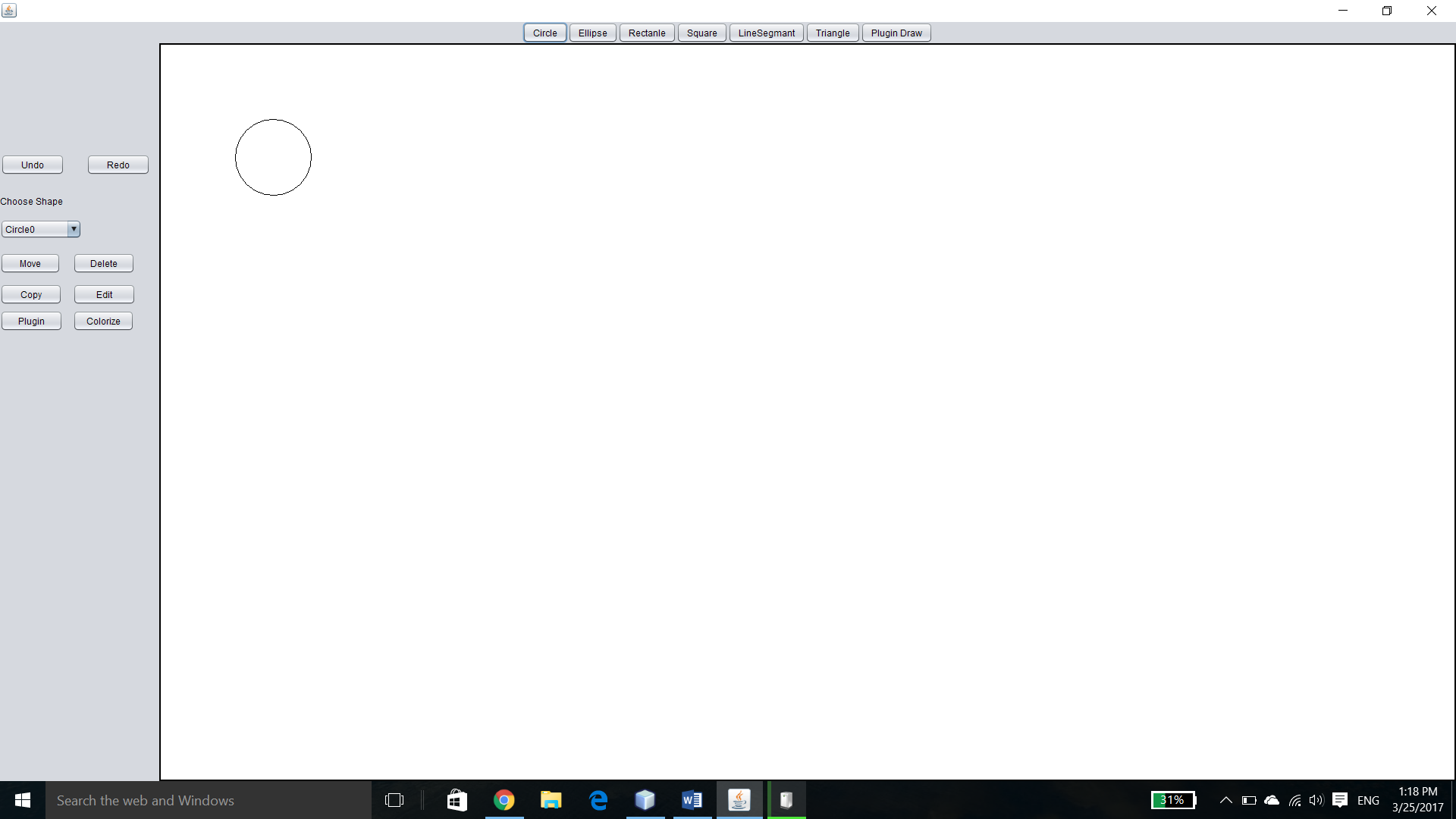
-redo: that gets back the undo action.

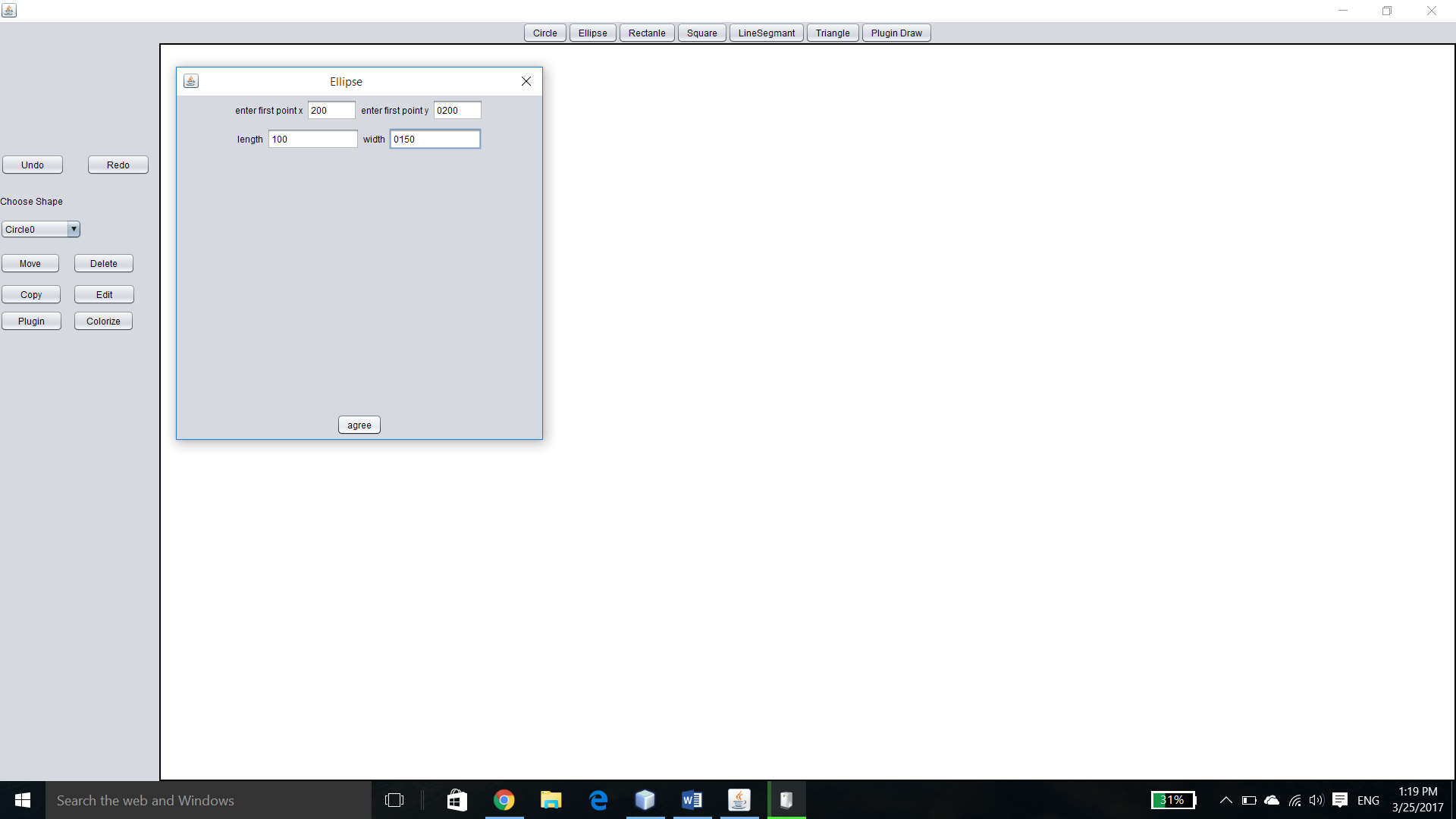
->Third: we have a plugin button that imports a new class and make the last operations and the plugindraw button draw the loaded class

Screenshots from the program:

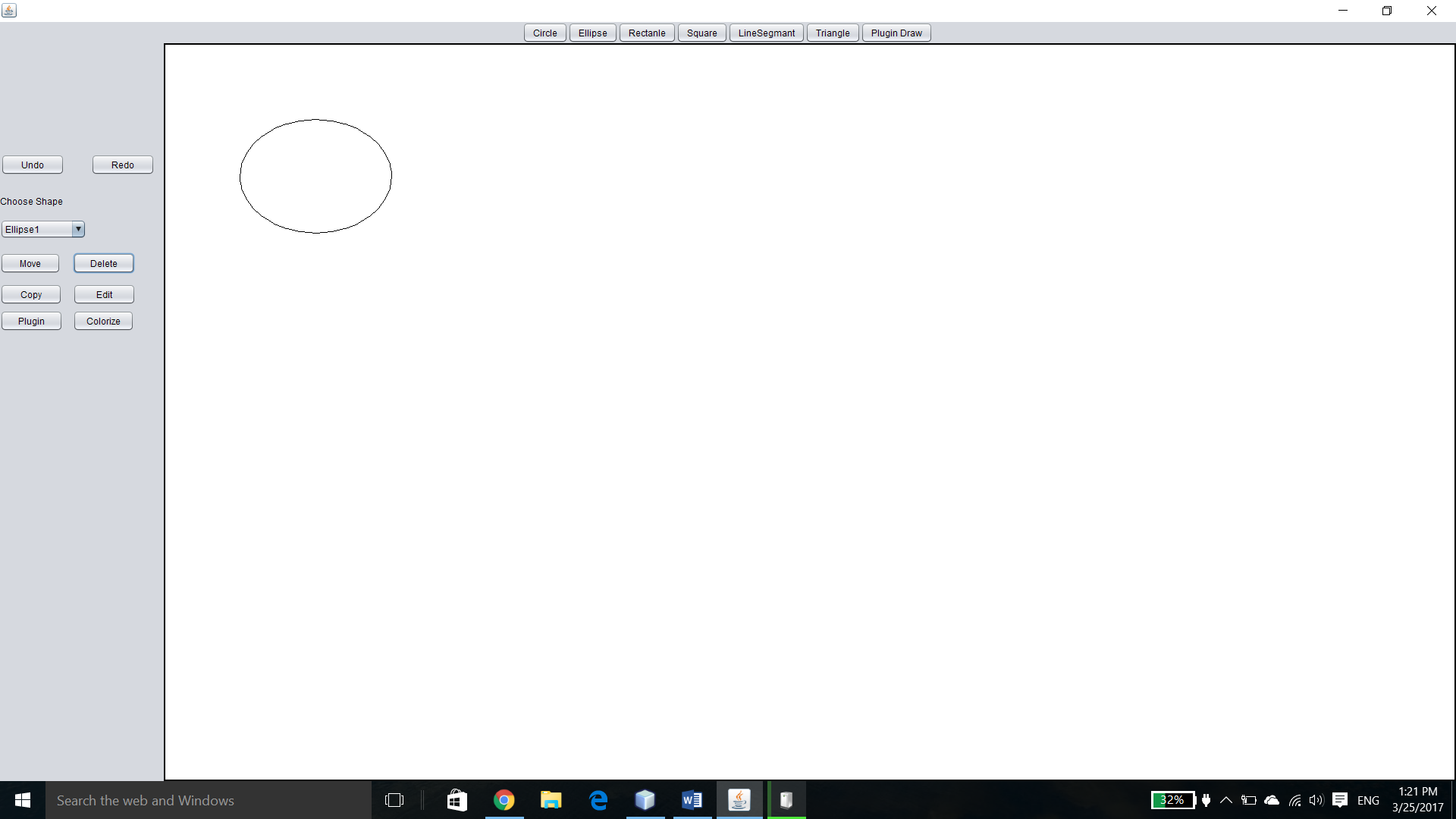
Circle dialog:

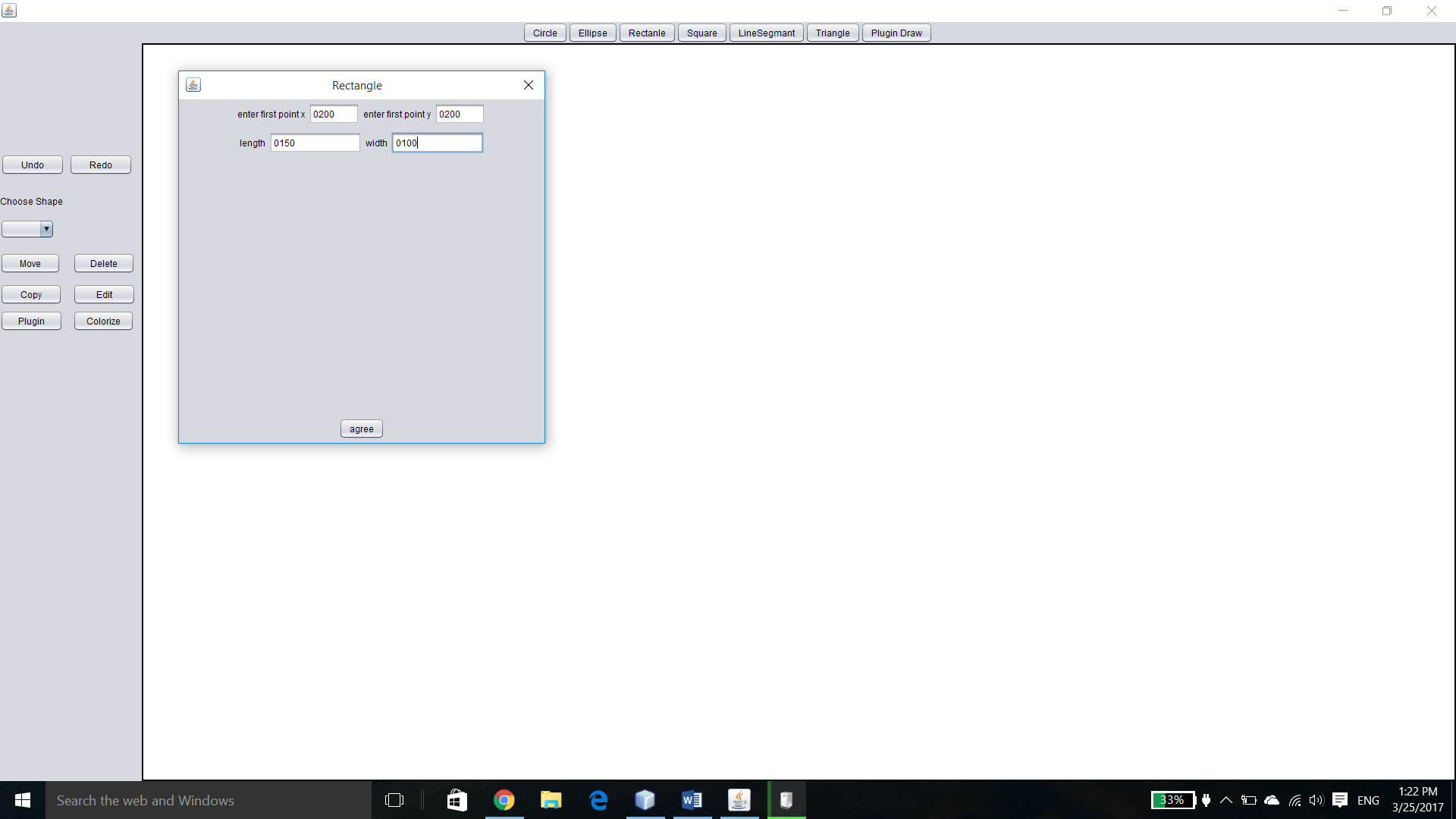
Circle drawing:



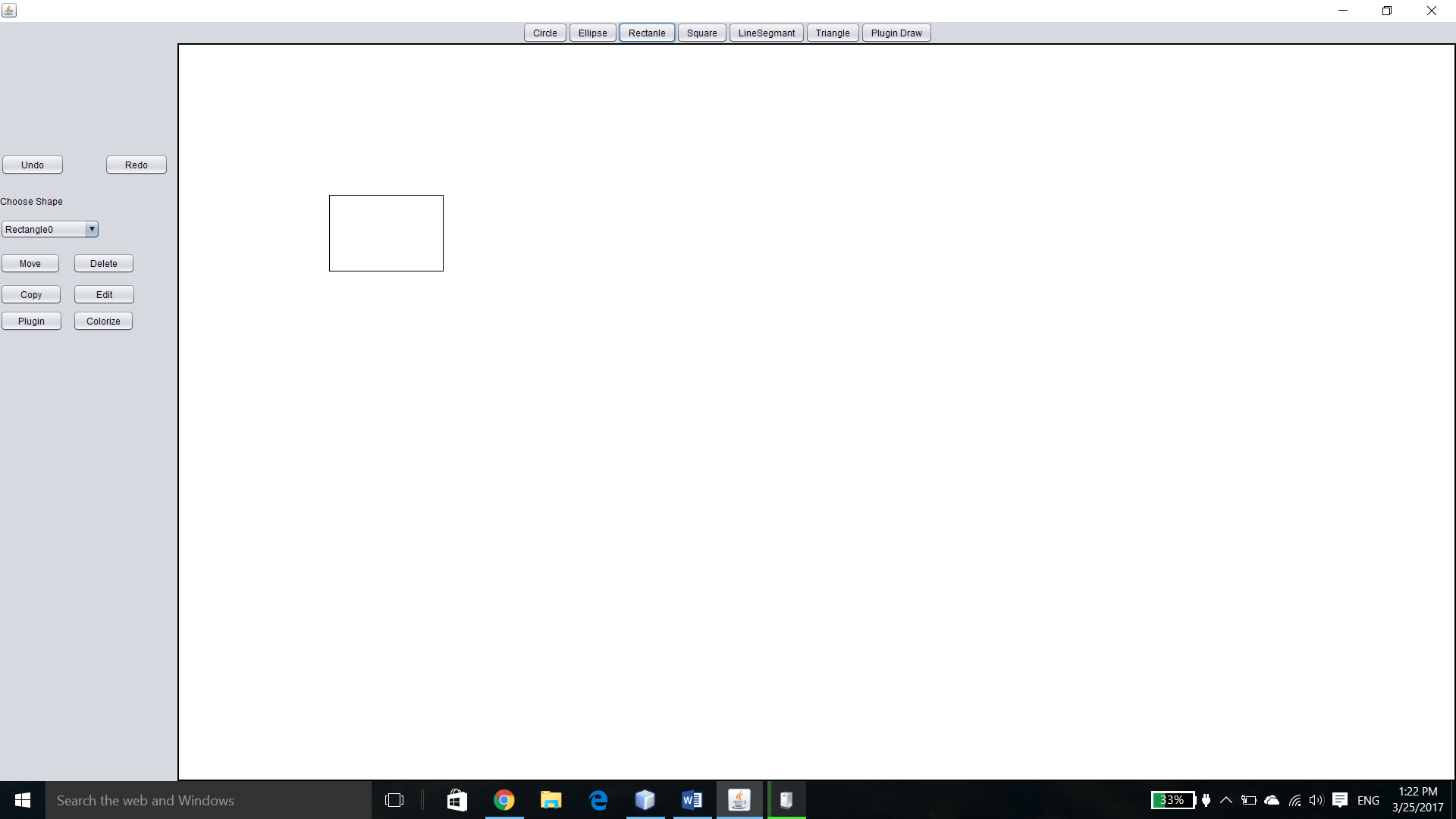
Ellipse dialog:

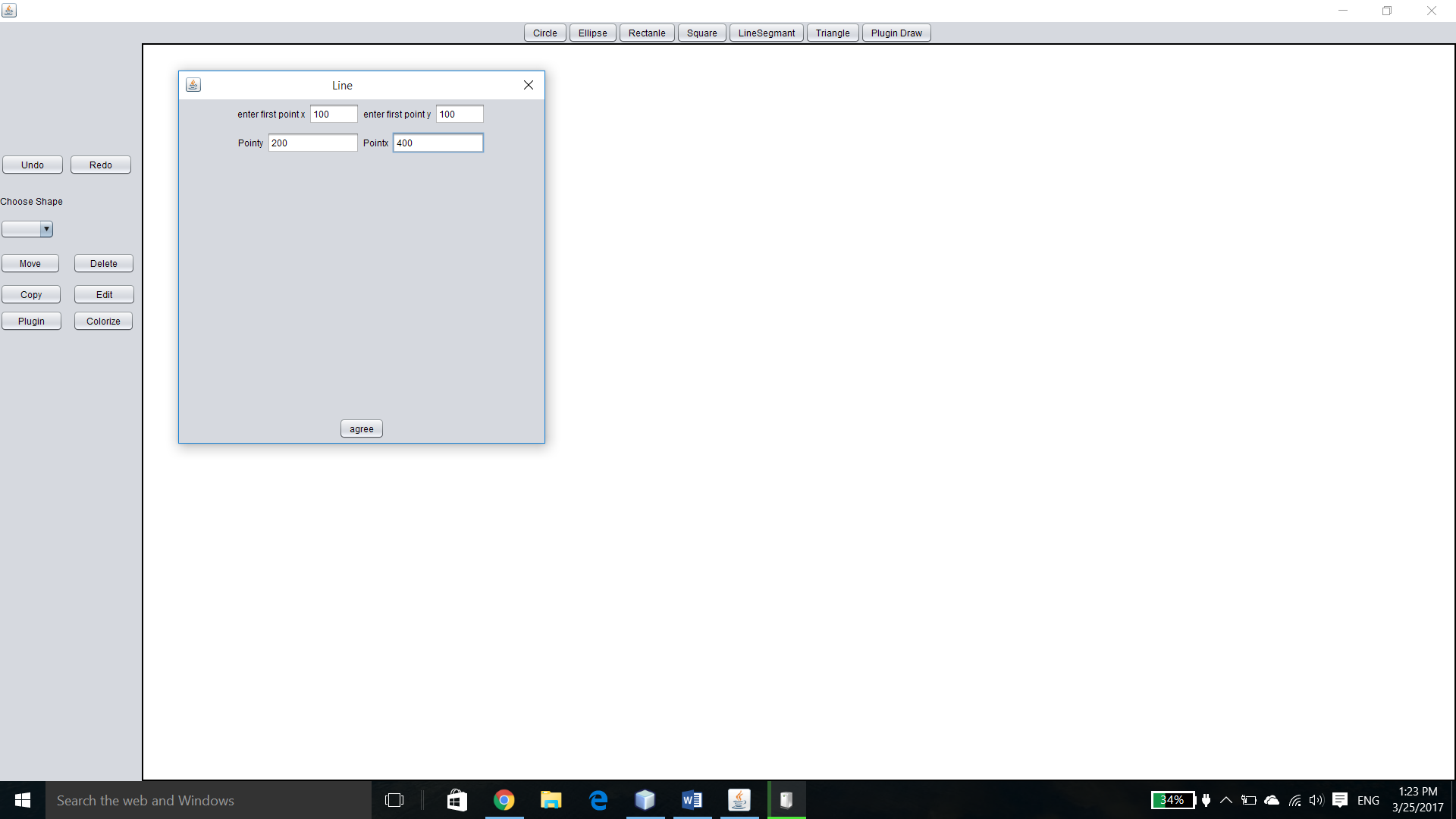
Ellipse drawing:



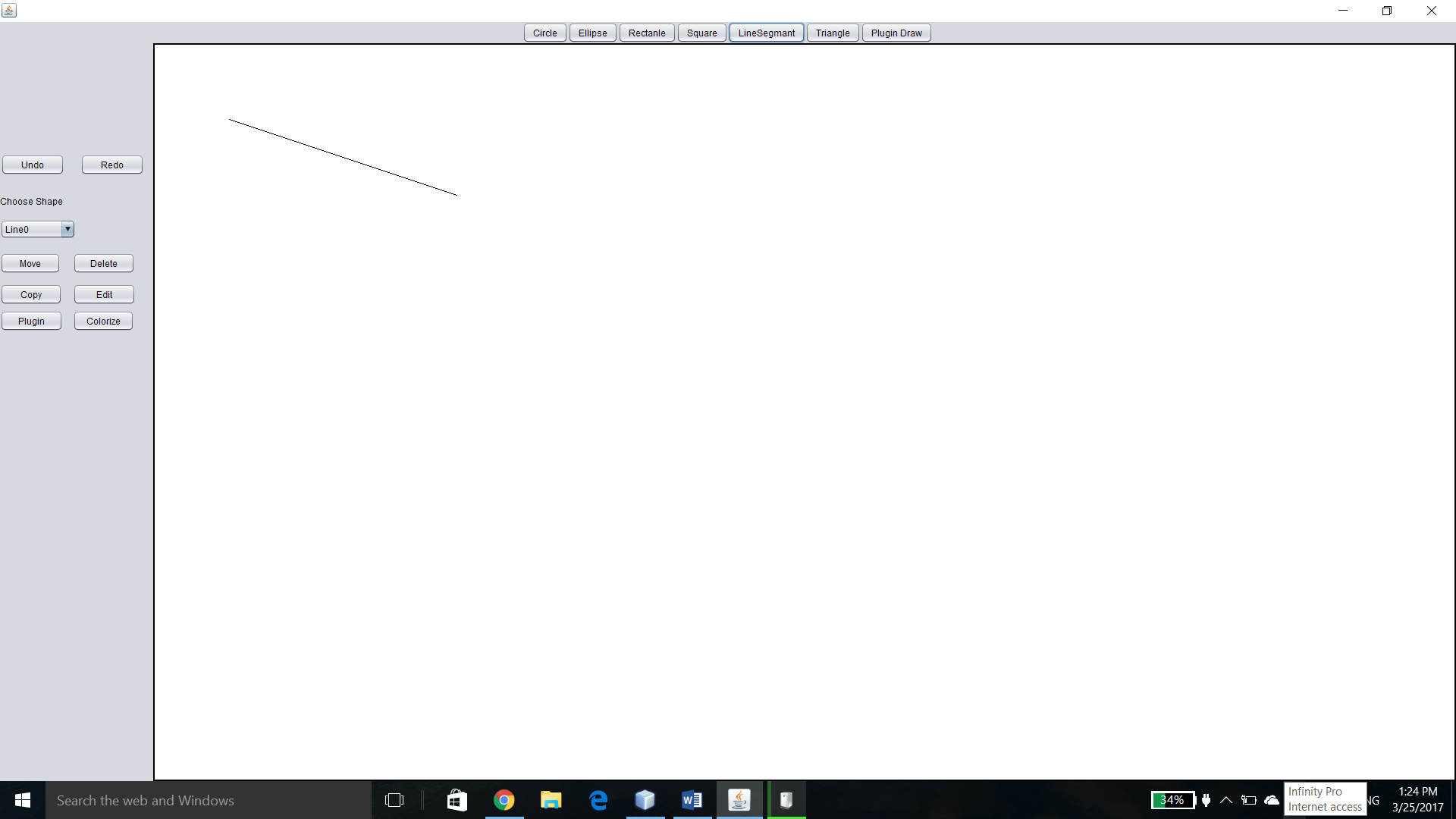
Rectangle dialog:

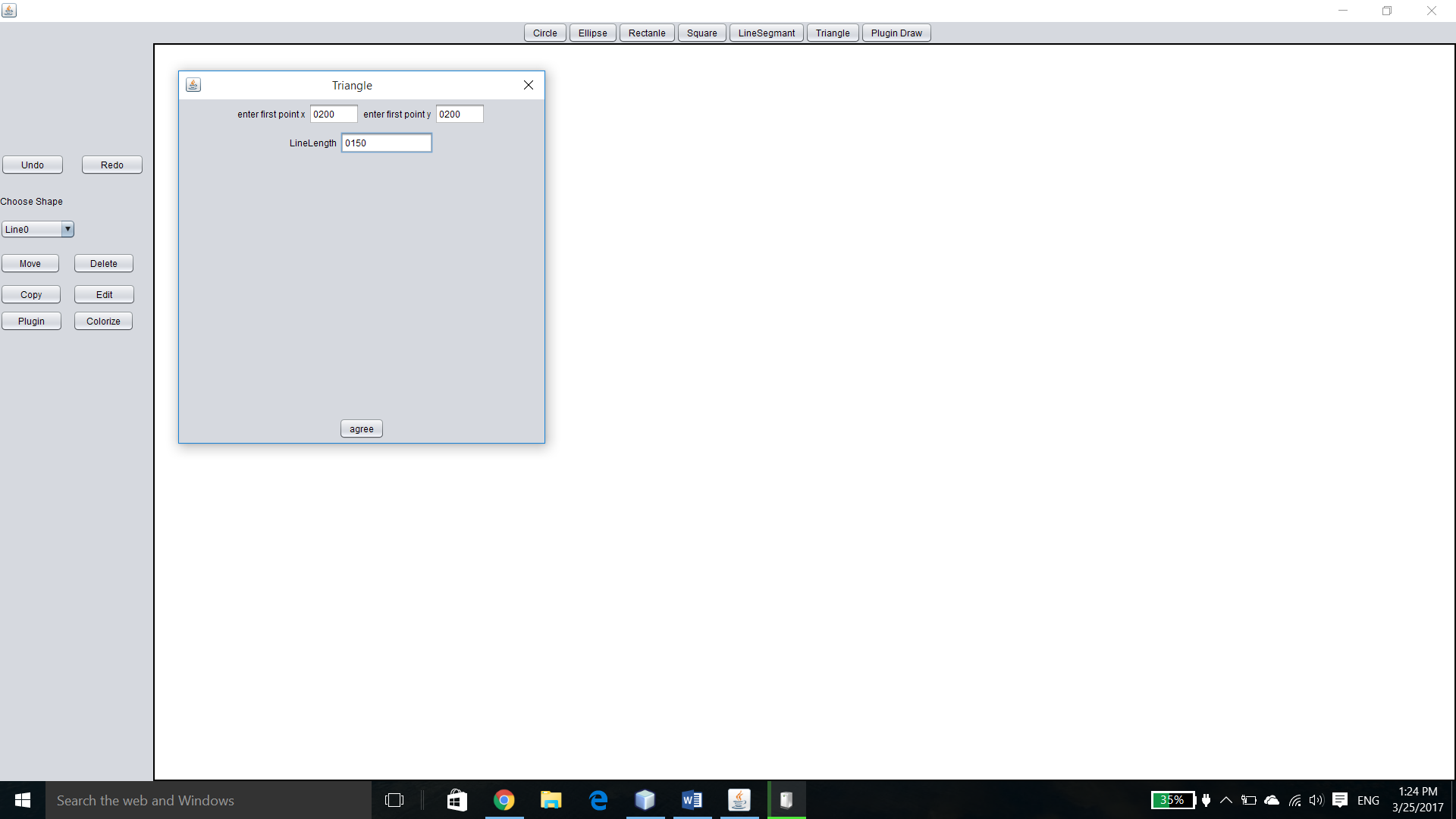
Rectangle drawing:



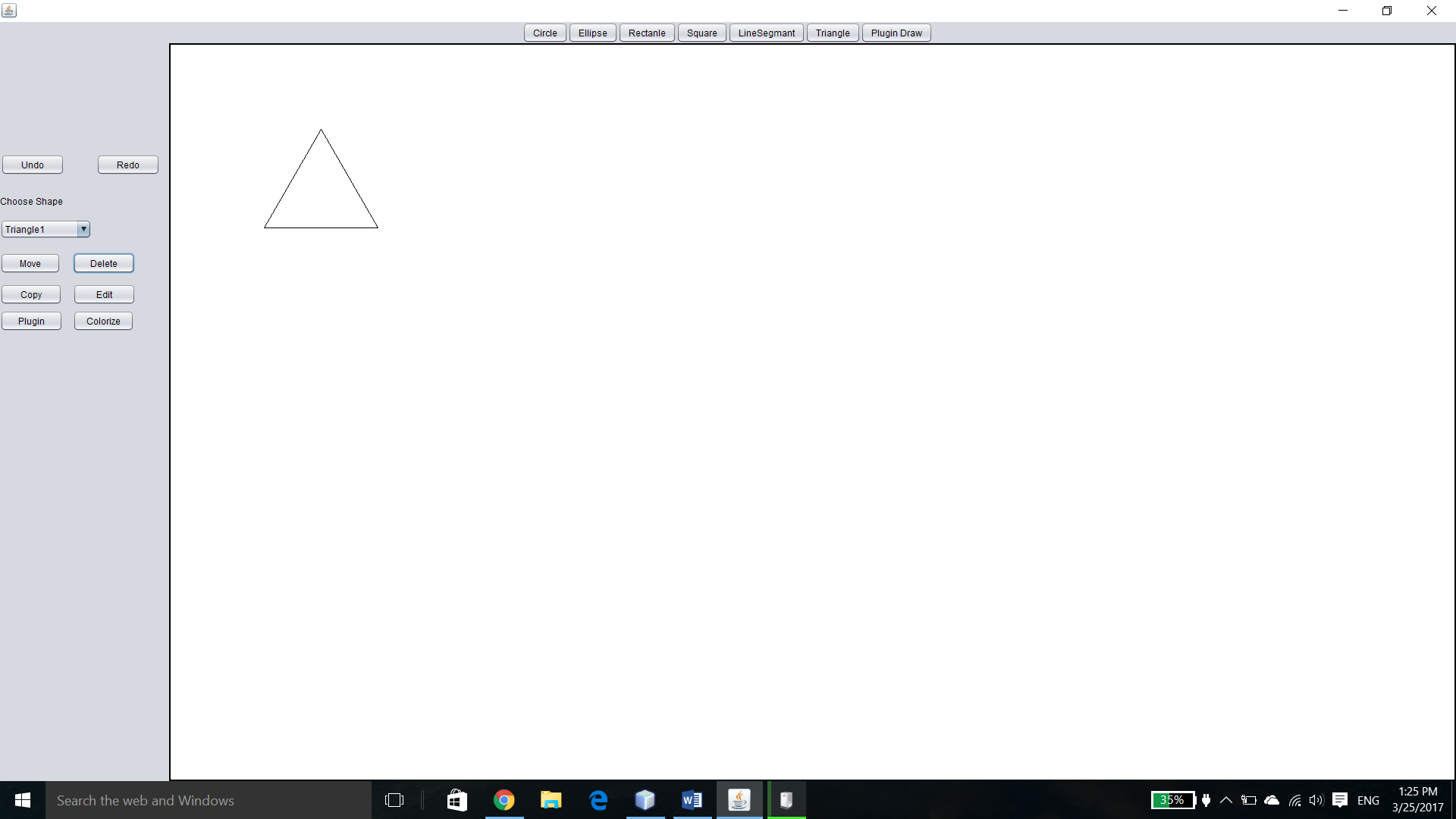
Line dialog:

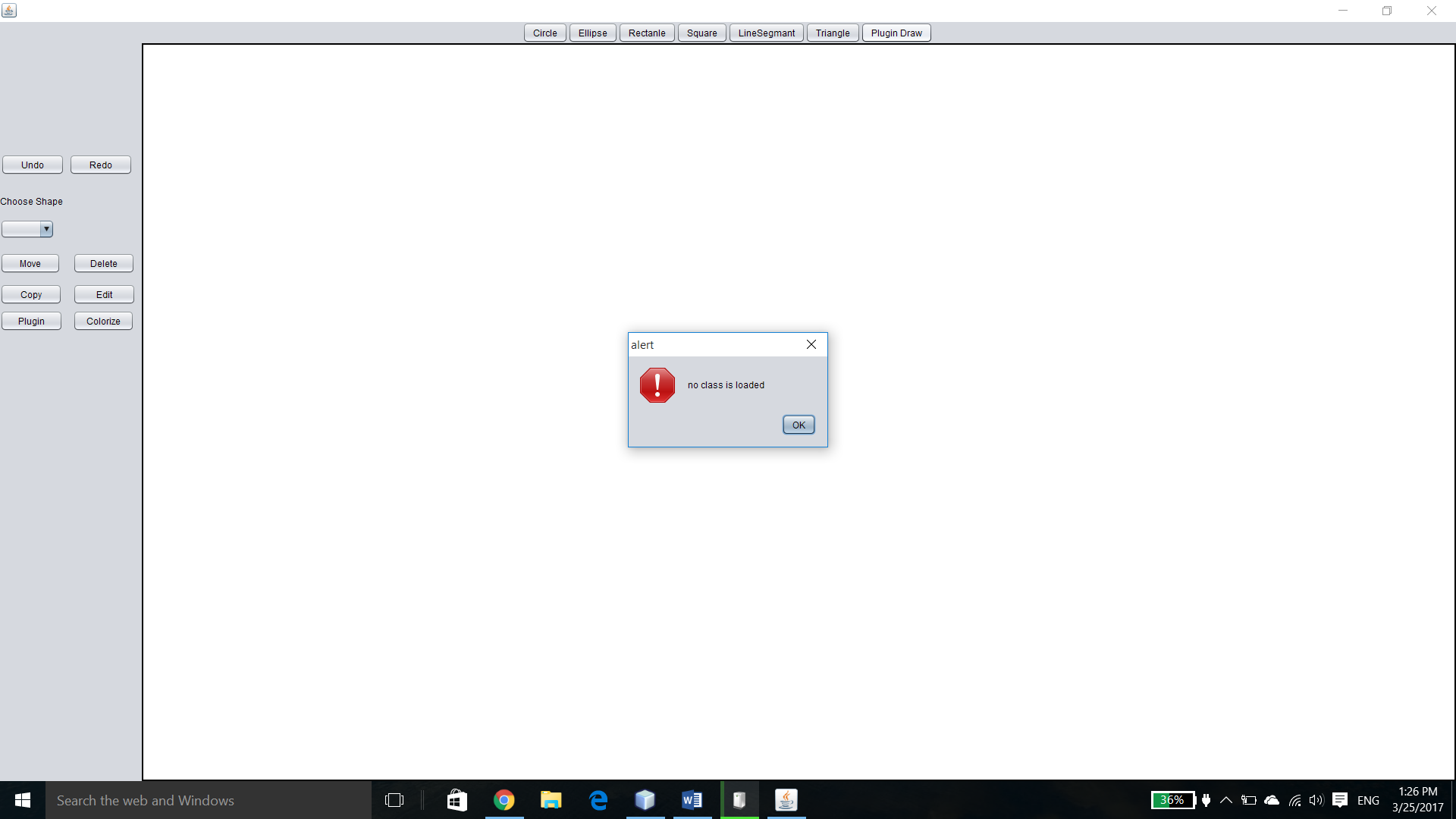
Line drawing:



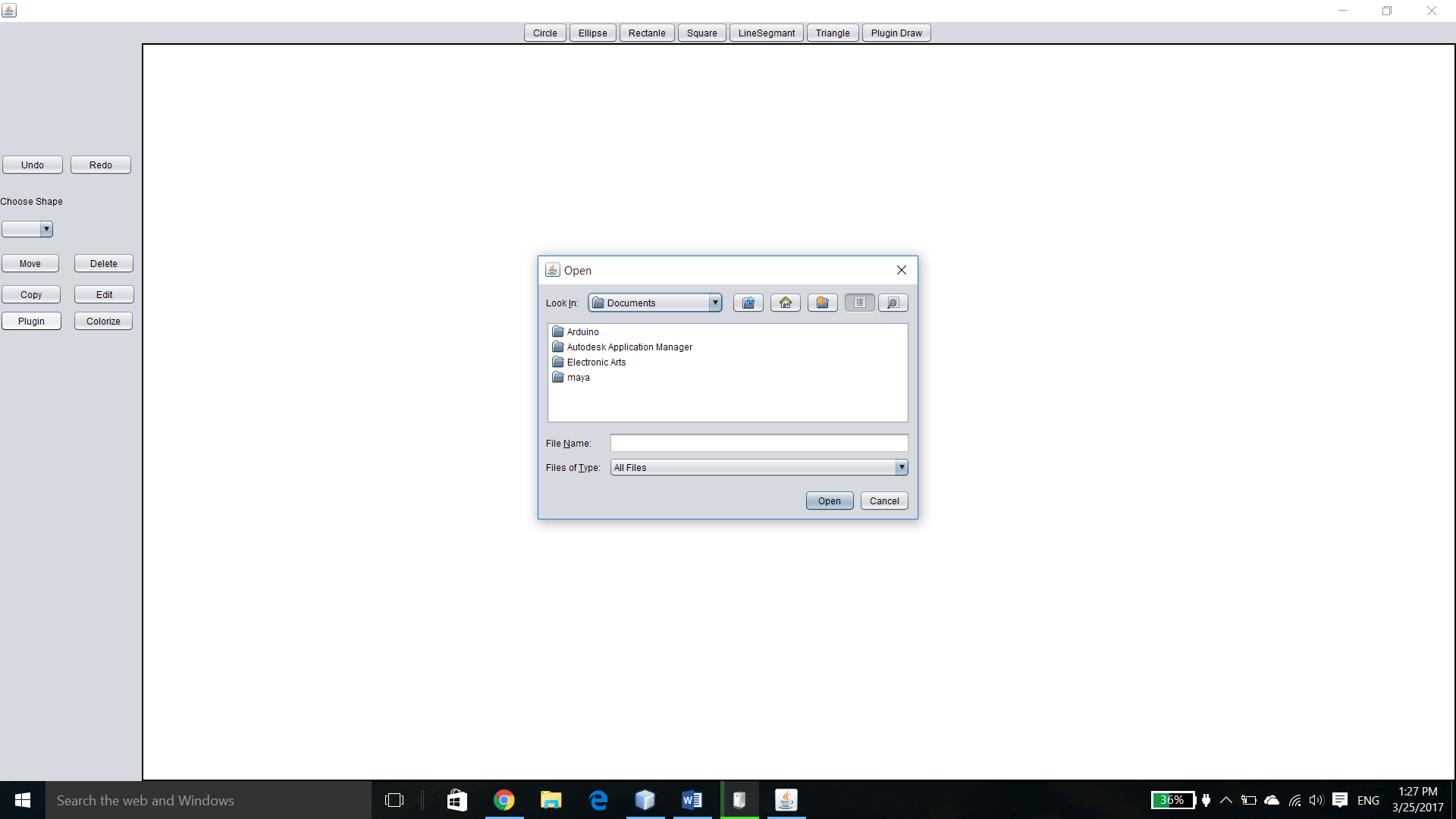
Triangle dialog:

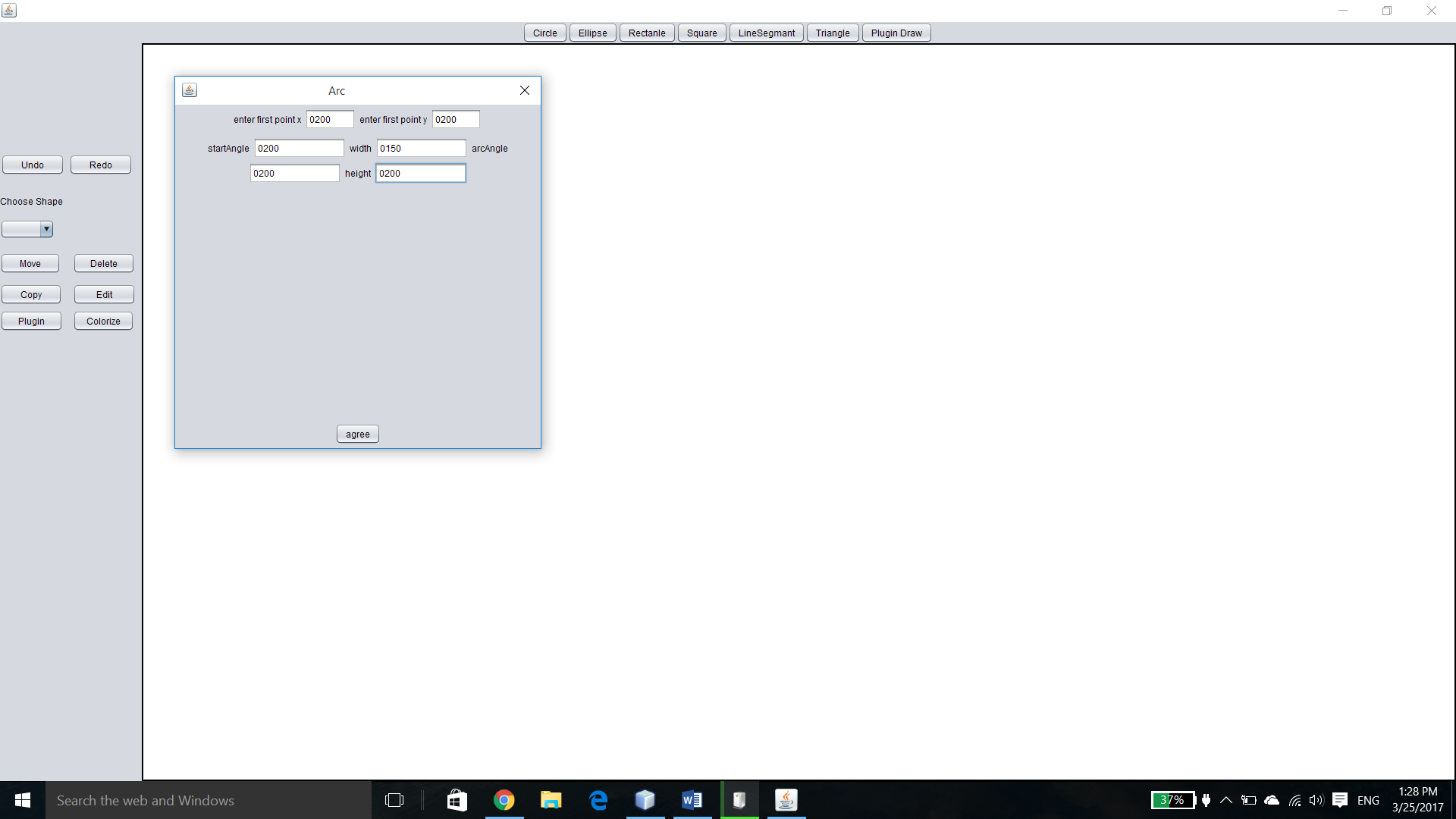
Triangle drawing:



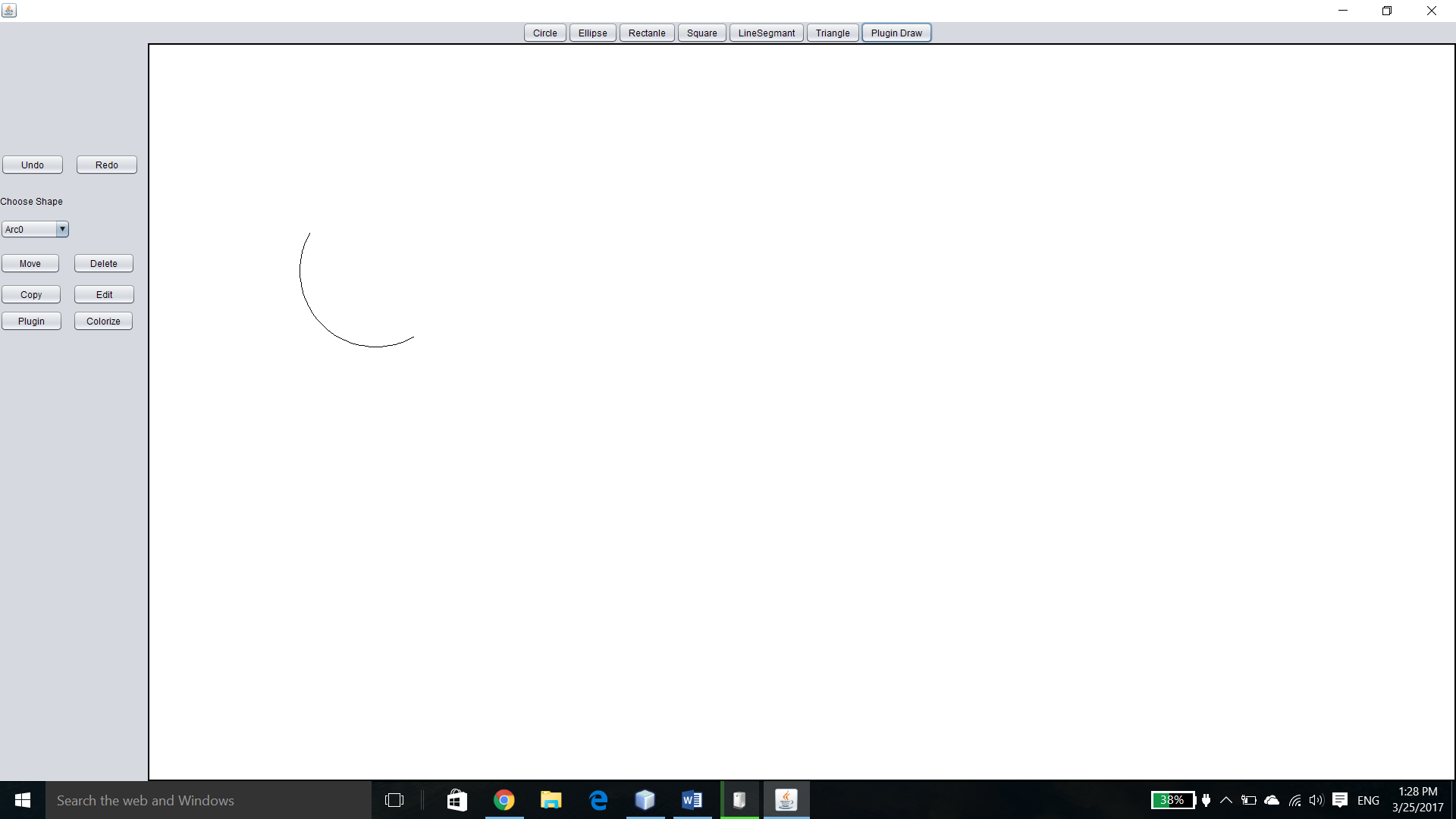
Pressing on the Plugin drawing before importing a class:

importing a new class as a Plugin:



Plugin(Arc) dialog:

Plugin(Arc) drawing:



UML diagram screenshot:

