Vector Based Drawing Application

Lab 2

Names:

- Enas Morsy Muhammad
- Sara Muhammad Youssef

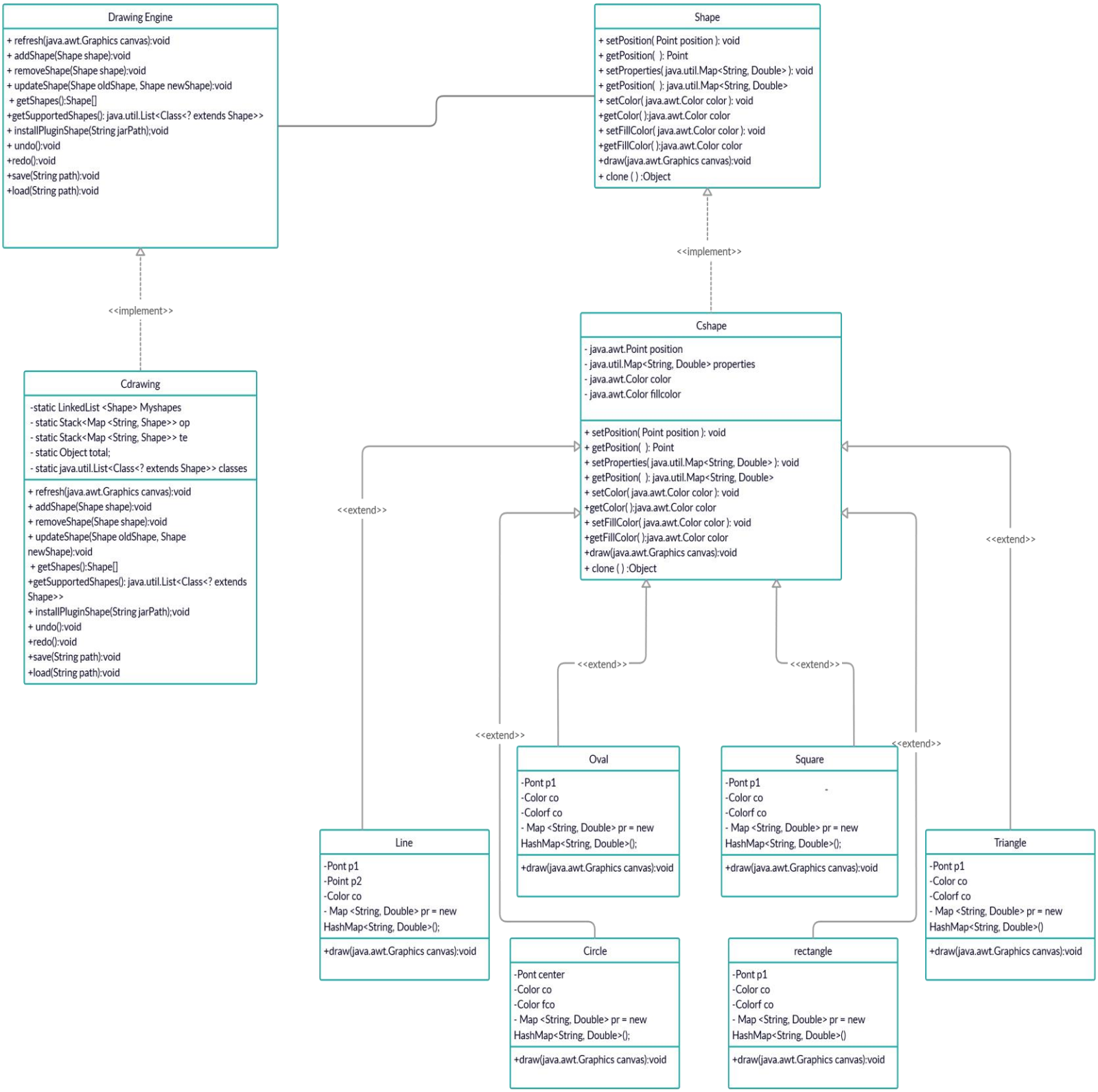
IDs:

- 20
- 31

programming

Dr. Khaled Nagy

UML diagram:


Drawing Engine
+ refresh(java.awt.Graphics canvas):void
+ addShape(Shape shape):void
+ removeShape(Shape shape):void
+ updateShape(Shape oldShape, Shape newShape):void
+ getShapes():Shape[]
+getSupportedShapes(): java.util.List<Class<? extends Shape>>
+ installPluginShape(String jarPath);void
+ undo():void
+redo():void
+save(String path):void
+load(String path):void
Shape
+ setPosition(Point position): void
+ getPosition( ): Point
+ setProperties(java.util.Map<String, Double>): void
+ getPosition( ): java.util.Map<String, Double>
+ setColor(java.awt.Color color): void
+getColor( ):java.awt.Color color
+ setFillColor(java.awt.Color color): void
+getFillColor( ):java.awt.Color color
+draw(java.awt.Graphics canvas):void
+ clone ( ) :Object
<<implement>>
<<implement>>
Cdrawing
-static LinkedList <Shape> Myshapes
- static Stack<Map <String, Shape>> op
- static Stack<Map <String, Shape>> te
- static Object total;
- static java.util.List<Class<? extends Shape>> classes
+ refresh(java.awt.Graphics canvas):void
+ addShape(Shape shape):void
+ removeShape(Shape shape):void
+ updateShape(Shape oldShape, Shape newShape):void
+ getShapes():Shape[]
+getSupportedShapes(): java.util.List<Class<? extends Shape>>
+ installPluginShape(String jarPath);void
+ undo():void
+redo():void
+save(String path):void
+load(String path):void
Cshape
- java.awt.Point position
- java.util.Map<String, Double> properties
- java.awt.Color color
- java.awt.Color fillcolor
+ setPosition(Point position): void
+ getPosition( ): Point
+ setProperties(java.util.Map<String, Double>): void
+ getPosition( ): java.util.Map<String, Double>
+ setColor(java.awt.Color color): void
+getColor( ):java.awt.Color color
+ setFillColor(java.awt.Color color): void
+getFillColor( ):java.awt.Color color
+draw(java.awt.Graphics canvas):void
+ clone ( ) :Object
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<extend>>
Line
-Pont p1
-Point p2
-Color co
- Map <String, Double> pr = new HashMap<String, Double>();
+draw(java.awt.Graphics canvas):void
Oval
-Pont p1
-Color co
-Colorf co
- Map <String, Double> pr = new HashMap<String, Double>();
+draw(java.awt.Graphics canvas):void
Square
-Pont p1
-Color co
-Colorf co
- Map <String, Double> pr = new HashMap<String, Double>();
+draw(java.awt.Graphics canvas):void
Triangle
-Pont p1
-Color co
-Colorf co
- Map <String, Double> pr = new HashMap<String, Double>()
+draw(java.awt.Graphics canvas):void
Circle
-Pont center
-Color co
-Color fco
- Map <String, Double> pr = new HashMap<String, Double>();
+draw(java.awt.Graphics canvas):void
rectangle
-Pont p1
-Color co
-Colorf co
- Map <String, Double> pr = new HashMap<String, Double>()
+draw(java.awt.Graphics canvas):void

user guide:

It is a simple application where the user only deals with buttons to choose an action to be performed.

- **To draw a shape:**
  1. choose border color or fill color of your shape
  2. press the shape button from the right ones
  3. go to drawing area
  4. press, drag and release with your mouse to determine location and size of the shape

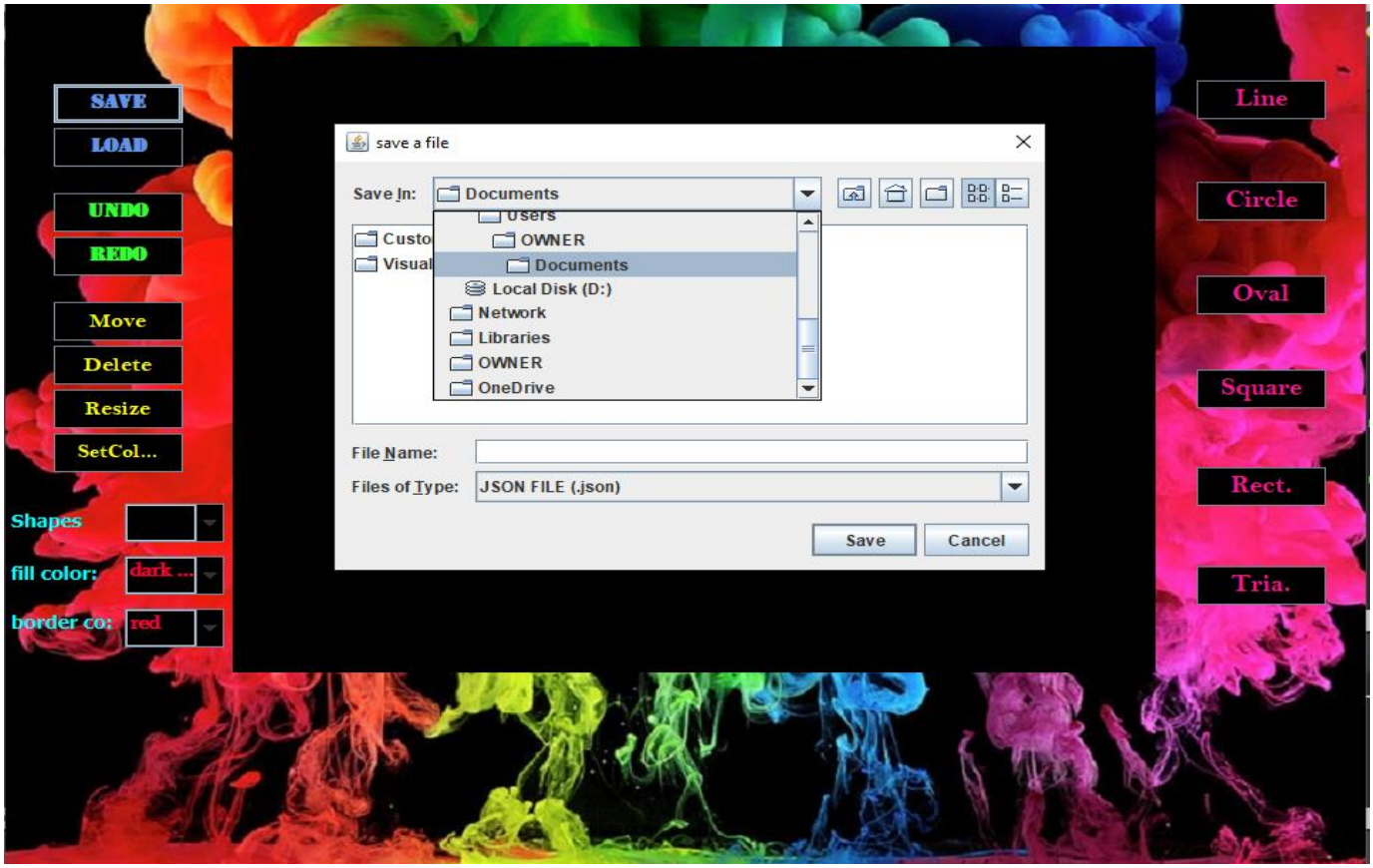
- **to modify an existing shape:**
    1. select the shape name from the Combo-Box **“shapes”**
    2. to **reset its color**; choose the required color from “border co” or “fill color” in case of full colored shape then press **“SetColor”**
    3. to **move it** press the button**” Move”,** go to drawing area and drag it to the new position
    4. to **resize it** press the button **”Resize”,** go to drawing area, press and drag until you reach the required size
    5. to **delete it** press the button**” Delete”**

- **You can undo any action and redo it again up to 20 actions by pressing undo or redo button.**

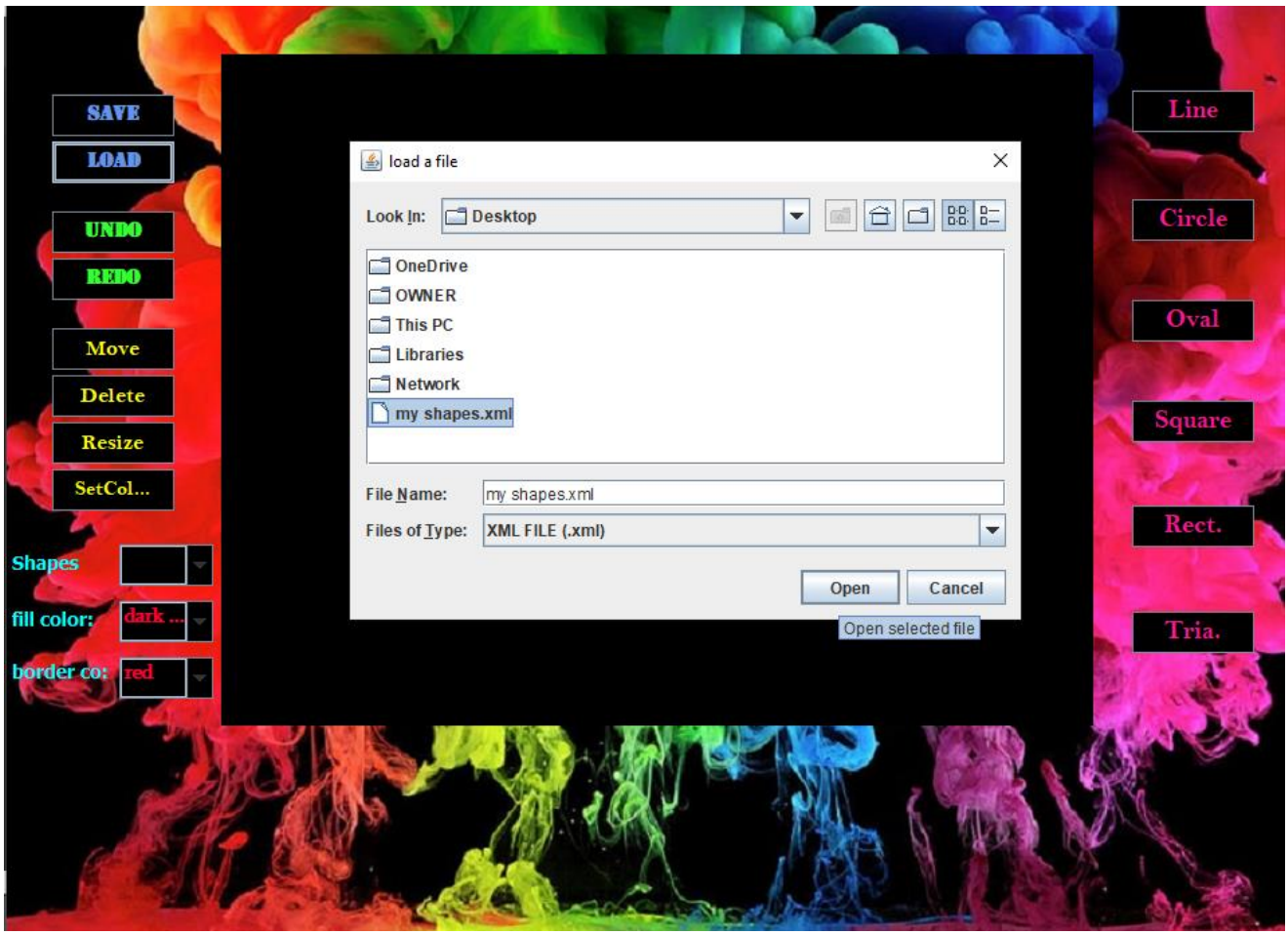
- **You can add any external jarfile that represents a class of unsupported shape and implement it into your environment to be able to use it through pressing “Add” button but some features may not be available for it.**

- **SAVE & LOAD:**

1. To **save your current shapes** press the button **“SAVE”,** choose whether to save them as **XML** file or **JASON** file, enter your file name, choose directory for your file to be saved in and press save.

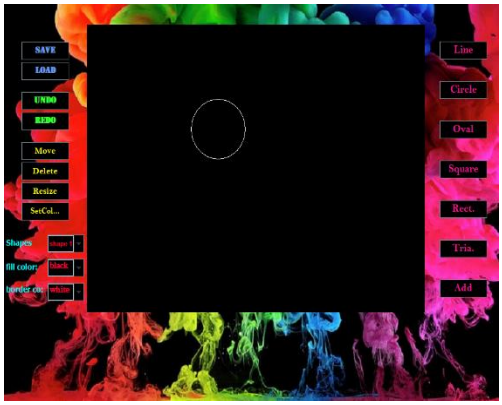


2. To **load saved shapes,** press the button **“LOAD”,** choose your file and press open then your shapes will be automatically saved in the current shapes array.

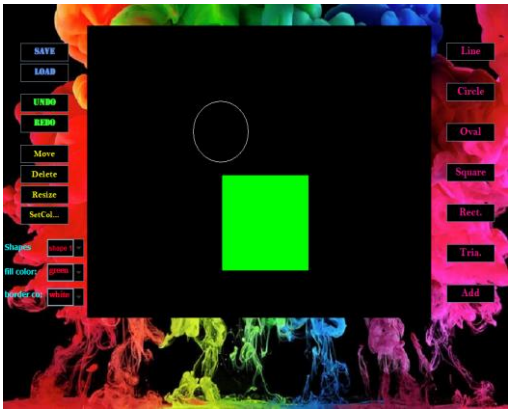


Sample runs:

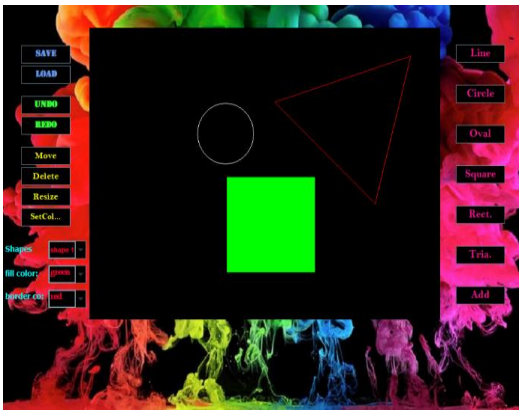
Add a circle:



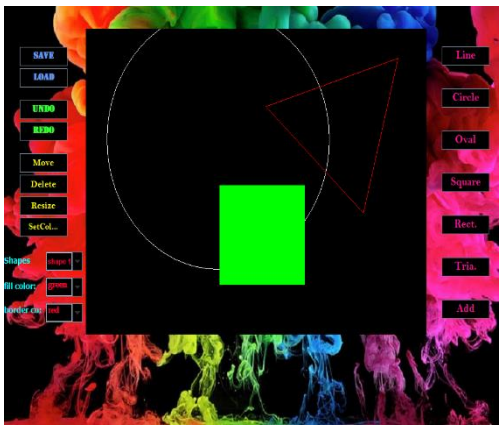
Add full colored square:



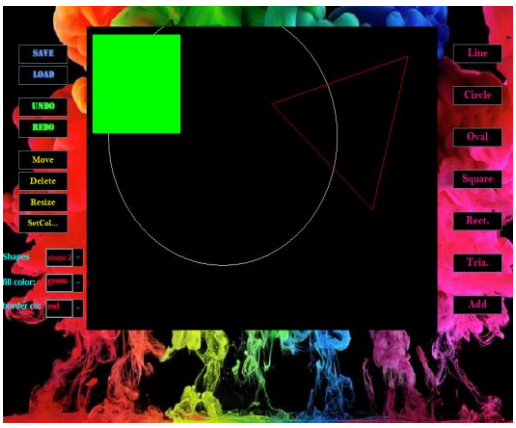
Add border colored triangle:



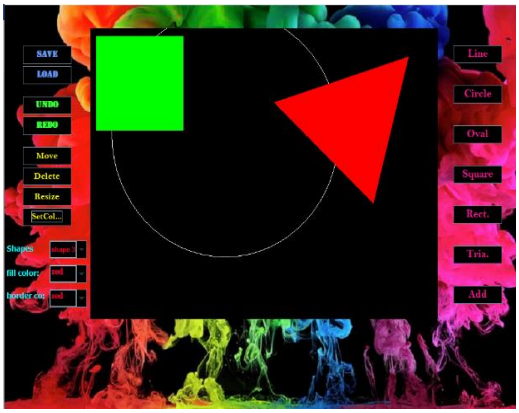
Resize the circle:



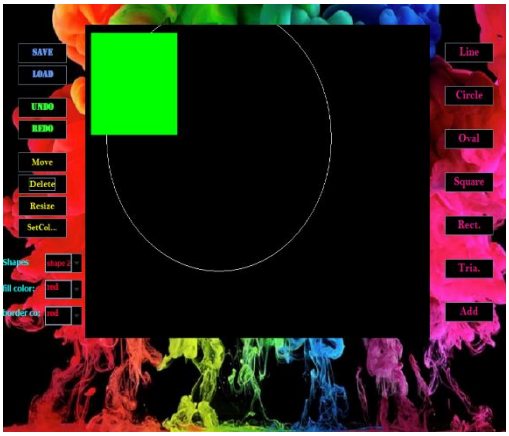
Move the square:



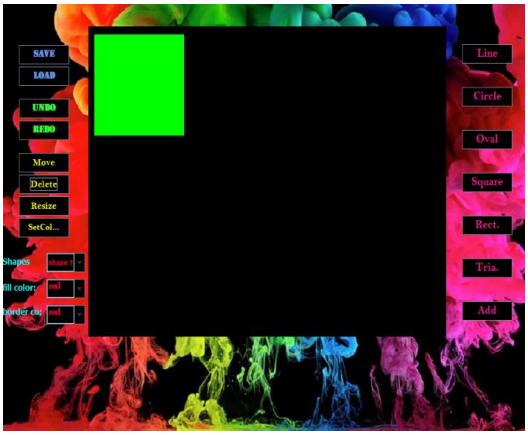
Make the triangle full colored



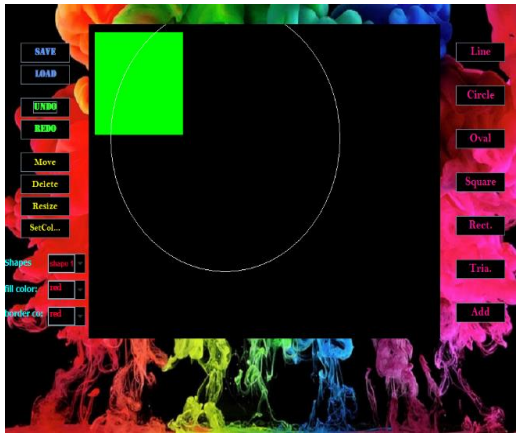
Delete the triangle:

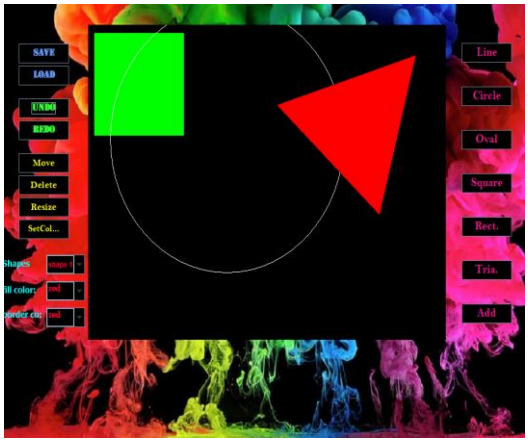


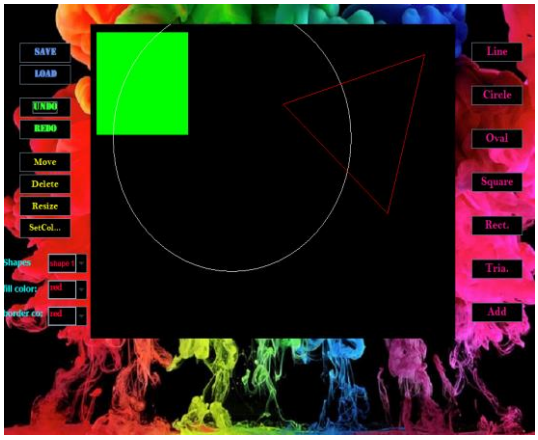
Delete the circle:

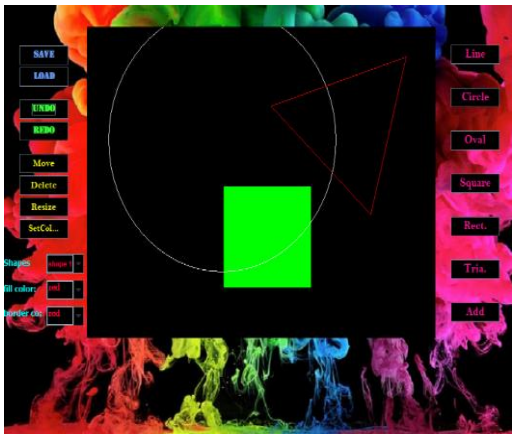


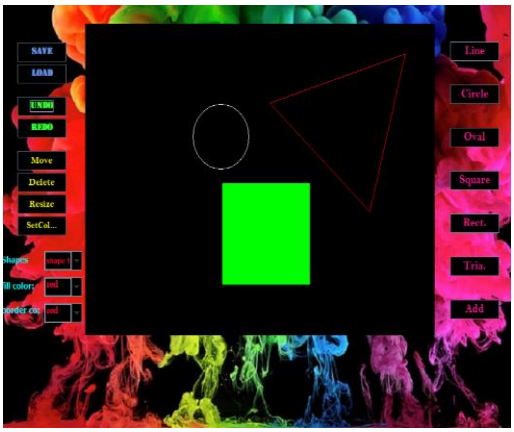
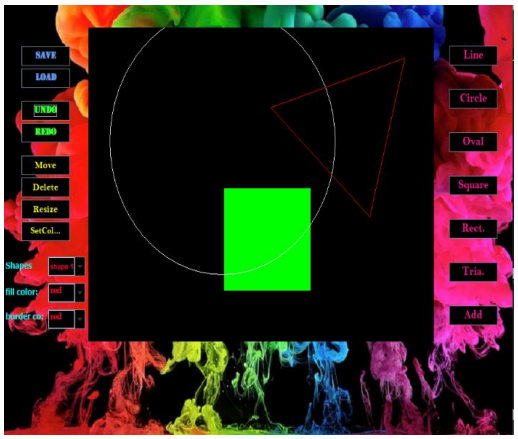
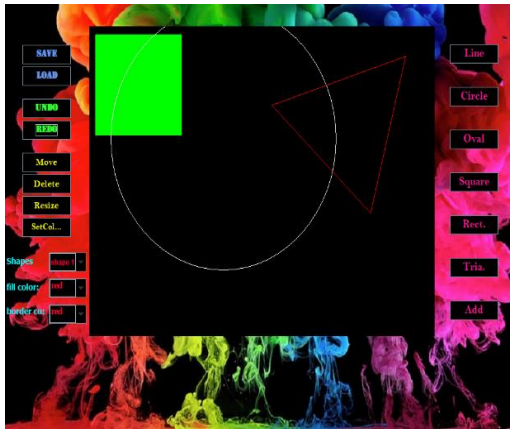
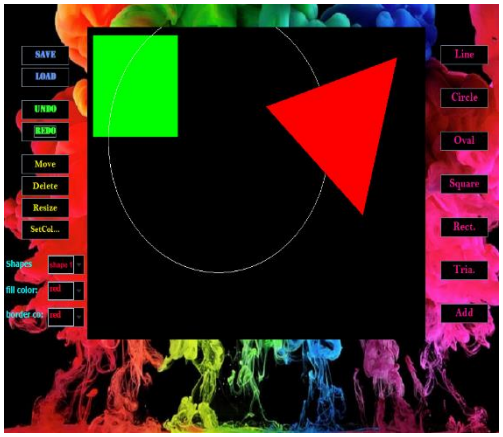
Undo 5 times:











Redo 3 times: