OOP - Final Excercise

Your goal in this exercise is to write a program that will:

- Accept a JSON file with instructions to draw a drawing made of shapes.
- Draw the requested drawing.
- Present the drawing on the screen and save it as a JPG image.

Shapes

The program should support the following shapes:

Basic Shapes

- Point
- Line
- Triangle
- Rectangle
- Circle

Each basic shape should support line color and fill color.

Composite Shapes

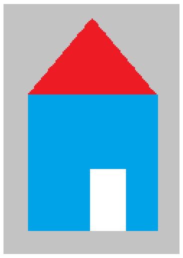
A composite shape is a shape defined in the JSON file as a collection of existing shapes.

The program should support two ways to define composite shapes:

1) Definition as a part of the instructions in the JSON file
2) Definition in another JSON file of instructions.

Example:

You can define the composite shape *"House"* with two rectangles and a triangle:



Afterwards, you can use it to define the composite shape *"Street"*:

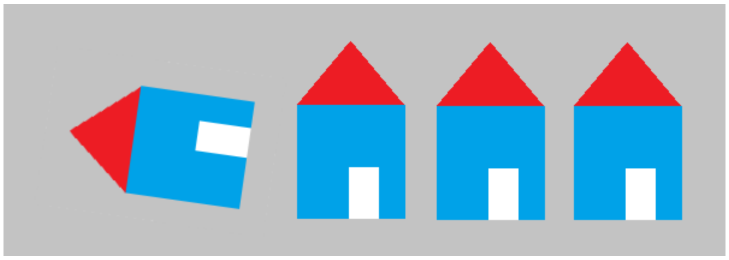
Operations On Shapes

Each shape needs to support the following operations:

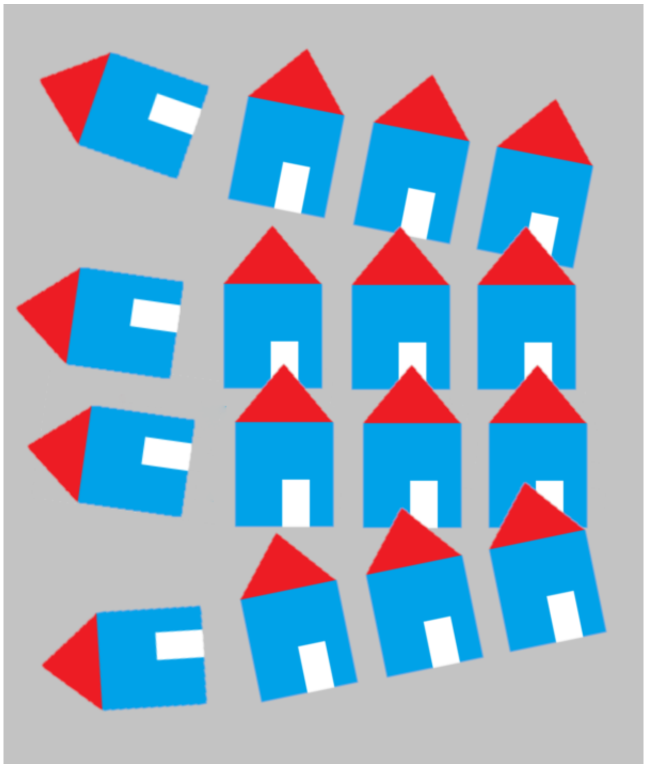
- Rotation
- Translation
- Scale (Resize)

<u>Example</u>

"A street where one house got knocked down because of a hurricane" Is a composite shape that was defined by several *"House"* shapes with different angles and locations.



We can go on with that example and get *"A city where all the left houses got knocked down because of a hurricane"*



Project Steps

There are three steps to this project:

1) JSON file format definition
2) Class and Interface design
3) Implementation

After every step you need to get approval from your instructor to continue to the next step.

General Notes

- Don't forget OOP principles - SOLID, Inheritance, abstract classes (use ABC)
- The definition of the JSON format is up to you. You can (and should) discuss it with your instructor. It is recommended not to change the definition once you've settled on one.
- The definition of the scale and rotation axis is up to you. It is recommended to define them in the most intuitive way possible (remember you are writing this for a client!)
- You should not implement functions you can find in libraries:
    - JSON parsing - use python's `json`
    - Shape rendering - use python's `cv2`
- Use well known design patterns!
- Write your code in a modular fashion - seperate to files, classes and methods.