

## Planter Bot

### Task 1A – Detect objects and enlist their properties

#### Goal:

To learn Python and use it to:

- Identify shape of objects.
- Identify colour of objects.
- Identify position of objects in the image

Please find the *Task-1A\_main.py* file in the nested folder named “2. Task\_Description\Test Images”. Modify the sections of *Task-1A\_main.py* marked for the same, to accomplish the following:

#### Given:

A set of five images named as **test1.png**, **test2.png**, **test3.png**, **test4.png** and **test5.png** in a folder named as “Test Images”. A sample set is shown in Figure 1. This folder is a sub-folder inside the “Task\_Description” folder.

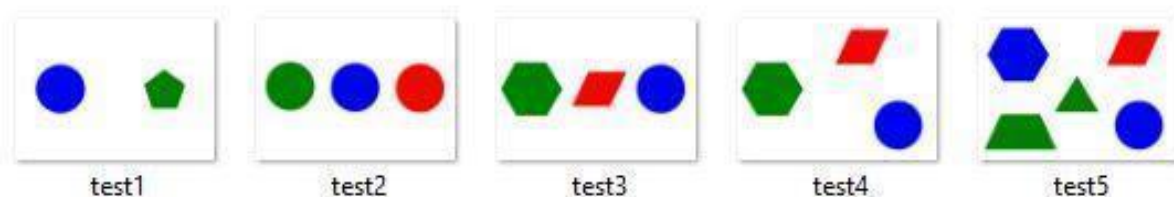


Figure 1: Sample Image Set containing five test images

Every image has a certain number of **objects**. Each Object is defined by three features, viz. **Color, Shape and Position in the image**. Color can be one of three primary colors: **Red, Green and Blue**. Shape can be one of any of the six shapes, viz. **Triangle, Rhombus, Trapezium, Pentagon, Hexagon and Circle**. Objects in a given image will vary in color, shape and position in the image and can appear multiple times in an image.

## Problem Description:

For each image in the “**Test Images**” folder, add to the Python script file your methods or functions in order to find the color, shape and X and Y coordinates of centroid (or center) of each object in that image.

Your code should return a List of Lists; where each individual List contains - image name followed by List of **Color-Shape-centroidX-centroidY** in a comma separated values List.

Each List must have as many elements of **Color-Shape-centroidX-centroidY** as number of objects in that image.

You have to find out the **color, Shape and X and Y coordinates of centroid** of each object in the image and concatenate their values into one string in the format given below then store it in the List.

Considering the image “**Sample1.png**”, shown in Figure 2, the output would be as follows:

```
[“sample1.png”, [“Green-Rhombus-191-361”], [“Blue-Circle-524-361”], [“Red-Triangle-831-392”]]
```

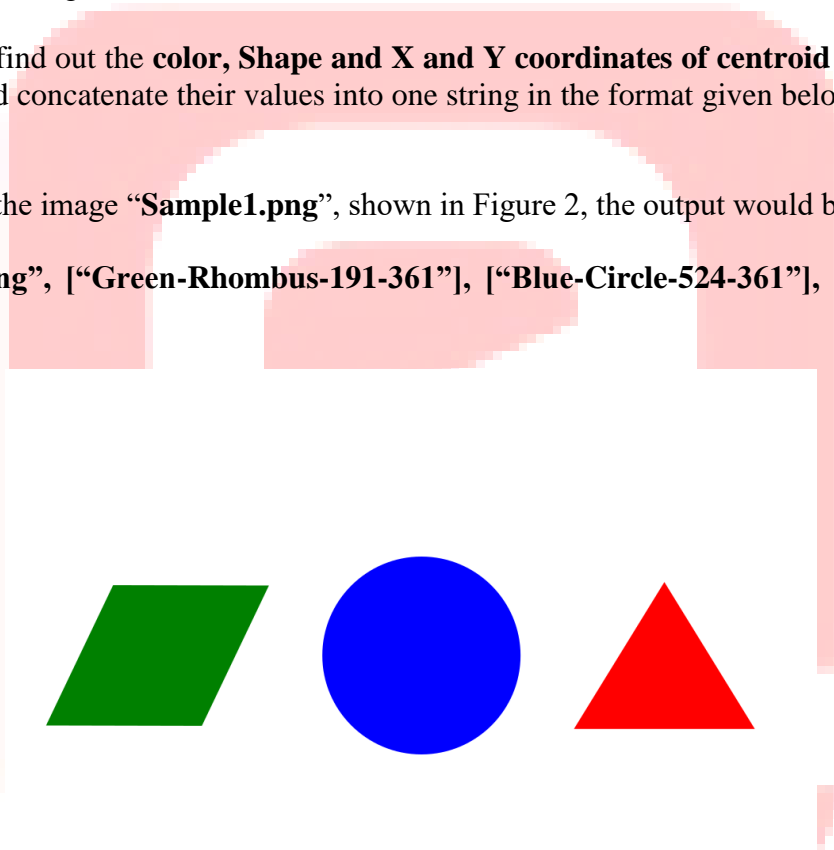


Figure 2: Sample1.png

### Required Output:

1. The Python script file named Task-1A\_main.py with the function “**main**” populated with your logic of solving the task.
2. A csv file with the Lists per image for each of the images in the given set of Test Images on each row i.e. five rows of data for a total of five test images in the “Test Images” folder. Each image will contain a maximum of 5 shapes. For our example, Sample1.png, the csv file would look like:

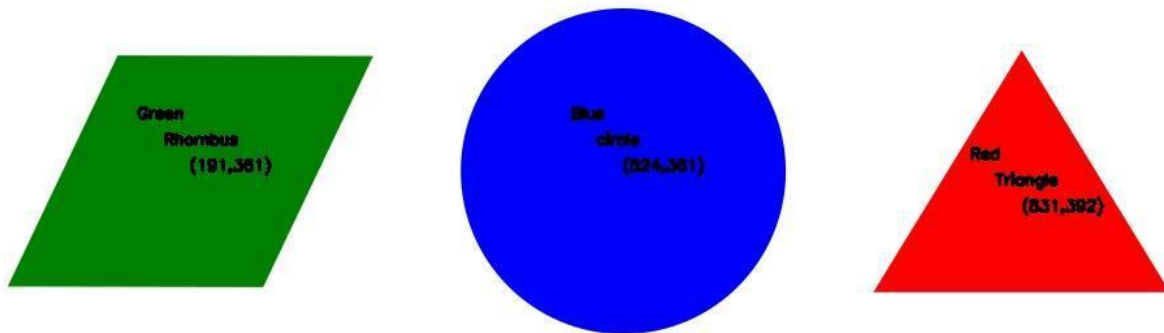
Sample1.png green-rhombus-191-361 red-triangle-831-392 blue-circle-524-361

results\_teamid - Notepad

File Edit Format View Help

Sample1.png,green-rhombus-191-361,red-triangle-831-392,blue-circle-524-361

3. An output image as shown in Figure 5:



*Figure 3: Example Output Image with all properties enlisted, overlaid on the object for Sample1.png image*

Name the output images with respect to its corresponding input image as - “**test1output.png**” for an input image named “**test1.png**” and so on.

**To do:**

1. Open **Task-1A\_main.py** located in the folder named **“Test Images”**. It has two functions:
  - **main():** It returns the required Lists for the **writcsv()** function. It should also write the output image to file. This is the method where you write your logic to solve the task.
  - **writcsv():** This function expects four parameters as arguments-**Color, Shape, CentroidX and CentroidY** and writes the elements of List of Lists one at a time into a **“result1A\_<TeamID>.csv”** file.  
**Do not edit this function. However, you can edit the global variable - “filename” with your TeamId in the Task-1A\_main.py Python code file.** Please use your eYRC team id while actually naming the file.

**IMPORTANT:** Do not change names of any of these functions.

**Rules:**

1. You need to write a **generic program**. Your code should be capable enough to detect any number of objects in an image and enlist their mentioned properties vis. **Color, Shape and Position**. In addition - *your code will be tested on several undisclosed images when you submit your code.*
2. Use basic knowledge of geometry to differentiate between the shapes of objects.
3. **Objects are not rotated or differently oriented with respect to each other.**
4. In case object is not present in the *image*, **return nothing or an empty List.**
5. In case objects are found in the *image*, **return the List of [Color-Shape-centroidX-centroidY] in addition with image name;** where there is one List appended per object in the image and one List of Lists per image.

Happy Learning

All The Best!!!