

## Taller 2

Fecha de entrega: lunes 31 de agosto de 2020, 8 pm

### 1. Cree una clase llamada **imageShape**:

#### a. Constructor:

- i. recibe dos parámetros de imagen: width y height

#### b. Métodos:

- i. **generateShape**: genera y almacena (como *self.shape*) una imagen de tamaño (width, height) correspondiente a un triángulo (0), cuadrado (1), rectángulo (2), o círculo (3). El tipo de figura se determina generando un número aleatorio uniformemente distribuido que toma los valores {0, 1, 2, 3}.

Parámetros de las figuras:

**triangle**: equilátero, lado =  $\min(\text{width}, \text{height})/2$

**square**: lado =  $\min(\text{width}, \text{height})/2$ , rotado 45 grados

**rectangle**: lado\_horizontal = width/2, lado\_vertical = height/2

**circle**: radio =  $\min(\text{width}, \text{height})/4$

La figura debe estar centrada en la imagen de tamaño (width, height). La figura generada se dibuja en color Cyan sobre un fondo negro.

Cada llamado a **generateShape** se genera una nueva imagen.

- ii. **showShape**: visualiza la imagen disponible en *self.shape* durante 5 segundos. Si no hay imagen disponible, visualiza una imagen en negro.
- iii. **getShape**: retorna la imagen generada y un string con el nombre de figura generada: 'triangle', 'square', 'rectangle' o 'circle'
- iv. **whatShape**: recibe una imagen de entrada (de fondo negro y objeto claro) que contiene una figura y la clasifica como 'triangle', 'square', 'rectangle' o 'circle'. Este método retorna un string con el nombre de tipo de figura resultante.

**Diseñe su método utilizando umbralización y análisis de contornos para clasificar la figura de entrada en una de las 4 clases.**

### 2. Cree un script (main.py) de Python que:

- a. Utilice la clase **imageShape**
- b. Pida al usuario las dimensiones de la imagen
- c. Genere una figura y la visualice por 5 segundos
- d. Clasifique la figura en 1 de las 4 clases y muestre el resultado en pantalla
- e. Indique si la clasificación realizada es correcta o incorrecta

### 3. El código debe estar debidamente indentado y comentado.

### 4. Suba a GitHub su repositorio del taller 2 antes de la fecha y hora de entrega. Agregue como colaborador a [n.ribero@javeriana.edu.co](mailto:n.ribero@javeriana.edu.co) a su repositorio para que tenga acceso a su solución.