Lab 2 : 2D Graphics

This laboratory session serves as an introduction to 2D graphics by utilizing OpenCV (Open Source Computer Vision Library) as a tool for the creation and manipulation of 2D graphics. While OpenCV is extensively employed in real-time computer vision applications, its functionalities encompass a broader scope. This session will explore how OpenCV's image processing capabilities can be employed to programmatically generate and modify 2D graphical elements.

Throughout this educational experience, we will explore the fundamental algorithms employed for 2D graphics within the OpenCV and Python programming environment. The focus will be on practical implementation, equipping you with the necessary skills to draw various geometric shapes directly onto images. These shapes will include lines, rectangles, circles, and even curves.

Installing OpenCV with Python (using Miniconda3)

Before we begin, we need to ensure everyone has OpenCV installed on their systems. We'll use Miniconda3, a lightweight package manager for Python, to install OpenCV in a clean and isolated environment.

Prerequisites:

- Download and install Miniconda3 from https://docs.anaconda.com/free/miniconda/miniconda-install (ensure you choose the appropriate installer for your operating system).

Installation Steps:

1. **Open a terminal or command prompt window.**
2. **(Optional) Create a new environment:** It's recommended to create a separate environment for this lab to avoid conflicts with other projects. Use the following command, replacing your_env_name with a desired name:

```
conda create -n your_env_name python=YOUR_PYTHON_VERSION
```

Replace YOUR_PYTHON_VERSION with the desired Python version (e.g., 3.8, 3.9).

3. **Activate the environment (if created):**

```
conda activate your_env_name
```

4. **Install OpenCV:** Use the following command to install OpenCV from the conda-forge channel:

```
conda install -c conda-forge opencv
pip install opencv-contrib-python
```

(The second command will install the full package if the first command cannot successfully install the opencv python package.)

Verification:

Once the installation is complete, open a Python interpreter (type python or python3 in your terminal) and try importing OpenCV:

```
import cv2
print(cv2.__version__)
```

Running this code should print the installed OpenCV version, indicating successful installation.

Assignment – Working with 2D graphics

Here is the outline of this lab assignment. There are four tasks to be completed.

1. Implement Bresenham's line algorithm. Please use the pseudocode in Slide 20 or the reference link.
2. Implement a function drawing rectangles using cv2.line() method. Please use the pseudocode in Slide 22.
3. Implement a function to draw ovals using cv2.line() method. Please use the pseudocode in Slide 24.
4. Implement a function to draw curves using cubic and quadratic Bezier curves. The equation is in Slide 28.

Each task has its own base code and it is waiting for you to fill in the incomplete codes.

1. Please implement task 1 in 'lab2_1_bresenham_line.py'.
2. Please implement task 2 in 'lab2_2_rectangle.py'.
3. Please implement task 3 in 'lab2_3_oval.py'.
4. Please implement task 4 in 'lab2_4_1_quadratic_bezier_curve.py' and 'lab2_4_2_cubic_bezier_curve.py'.