**FLOW OF CURVETOPIA PROJECT**

### 1. ****Input Data Preparation****

* **Input Format:** Start with a CSV file containing poly-lines that represent the shapes you want to process.
* **Loading Data:** Read and parse the CSV file into a structured format, where each shape is represented by a set of 2D points.

### 2. ****Curve Regularization****

* **Identify Regular Shapes:** Analyze the poly lines to identify simple shapes like lines, circles, ellipses, and polygons.
* **Regularize Curves:** Fit the identified shapes to their ideal geometric forms (e.g., straight lines, perfect circles).

### 3. ****Symmetry Detection****

* **Analyze Symmetry:** For each regularized shape, check for reflection symmetry (mirror symmetry).
* **Classify Symmetry:** Categorize shapes based on whether they have reflection symmetry or not.

### 4. ****Curve Completion****

* **Detect Incomplete Curves:** Identify any curves that are incomplete or have gaps.
* **Complete Curves:** Use geometric algorithms to fill in missing parts of the curves, ensuring they are closed and smooth.

### 5. ****Visualization****

* **Plotting Results:** Visualize the regularized, completed, and symmetry-analyzed curves using matplotlib.
* **Inspect Output:** Display the processed curves to verify and refine the transformations.

### 6. ****Output****

* **Save Results:** Optionally, save the processed curves to a new CSV or SVG file.
* **Review Output:** Compare the final curves with the original input to assess the effectiveness of the regularization and completion processes.

### 7. ****Iteration****

* **Refinement:** Based on visual inspection, adjust parameters or rerun the process to achieve better results.

This flow ensures that raw, hand-drawn shapes are transformed into clean, regularized, and visually pleasing curves, ready for further use or analysis.

4o