Team Project: SVG Reader

Overviews

The objective of this project is to design and implement an SVG file rendering application using **C++** programming language and **Object-Oriented Programming** (OOP) principles. The application will parse an SVG file, render the vector graphics, and handle user interactions if applicable.

Functional Requirements

SVG Reading and Parsing

- The program must have the capability to read SVG files and parse their content.
- SVG file parsing should handle various SVG elements, attributes, and their hierarchical structure.
- Implement classes representing SVG elements (e.g., `Circle`, `Path`, `Text`) using object-oriented definition.

Rendering

- Implement a rendering engine that utilizes the parsed SVG data to draw vector graphics on the screen.
- Use object-oriented design patterns to represent different SVG elements as objects and render them accordingly.
- Support for rendering basic shapes, text, border, and fill is mandatory.

Interactivity (optional)

- Zoom in/out
- Rotate

Object-Oriented Design

- Implement a well-organized class hierarchy representing different SVG elements.
- Utilize inheritance, encapsulation, and polymorphism to create a robust and flexible design.
- Consider implementing design patterns for object creation and traversing SVG elements.

Documentation

- Provide detailed documentation describing the class hierarchy, methods, and their functionality.
- Include UML diagrams representing the class relationships and interactions within the rendering module.
- Comment the source code thoroughly to enhance readability and understanding for future developers.