ANIMATE++: an vector animation library in C++

Wode "Nimo" Ni
Columbia University
wn2155@columbia.edu

Xuanyuan Zhang
Columbia University
xz2580@columbia.edu

1. Motivation

Vector graphics with animation has been popular ever since the era of Adobe Flash in the last 2 decades. Recently, an alternative to create vector-based animation, Structured Vector Graphics (SVG) and HTML5 Canvas has became more popular among front-end developers and artists. The problem with these data formats, however, are often difficult for human to parse and interact with. Therefore, many libraries in JavaScript[1, 2] and direct manipulation tools[3] emerged. Sadly, due to JavaScripts weaker support for object-oriented programming (OOP), the library are sometimes less intuitive to interact with and lack language affordances such as type-checking. The direct manipulation tools, on the other hand, have more user-friendly interfaces but lack high-level abstractions and ways to encode complex behaviors, which can be tedious to work with.

2. Objective

The goal of this project is to programmatically create vector animation in an intuitive, object-oriented manner. Taking advantages of C++s object oriented programming features as well as static typing, we believe our library will bring up an intuitive environment where users compose new vector graphics primitives, import existing ones, and eventually generate beautiful animations in a light-weight manner.

3. Related Work

- **Snap SVG**: a JavaScript Library with support for creating SVG based animations
- **Paper.js**: The Swiss Army Knife of Vector Graphics Scripting, another JS library for vector graphics animation. It shows some interesting uses of the JS language: it implemented (a very limited version of) operator overloading in the library, which turns out to be extremely complicated to implement given the JS language. We believe Animate++ can replicate and enhance its effort in simplifying the library interface using C++, which has strong language features.
- **SVGator**: a direct manipulation tool to animate SVG files. We plan to learn from its user interface and provided functionalities to build our abstractions and library functions to model them. We believe that a programmatic tool will be more powerful in that users can compose customized functions to automate the design process and encode complex behaviors (e.g. complex paths that objects travel) programmatically.

4. Proposed Solution

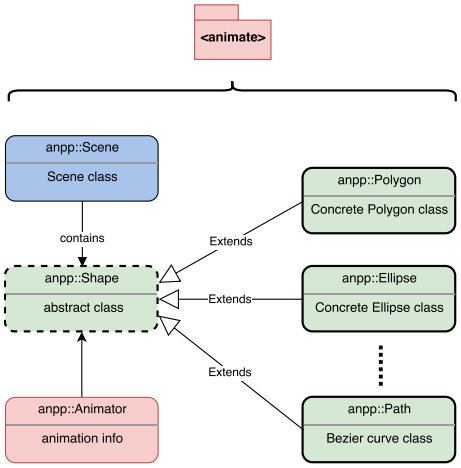

Figure 1. Structure of ANIMATE++ class hierarchy.

We propose to build a C++ library, ANIMATE++, with class structures that exploits inheritance, polymorphism, and many other modern C++ features and primitives for vector graphics animation (shown in Figure 1). Specifically, we plan to build classes of graphical primitives with animation description objects attached, abstractions for other concepts such as scenes, and high-level functions to generate common animation elements.

5. Team Member Resposibilities

- **Wode "Nimo" Ni**: system architecture, library interface design, graphics related implementation, implementation of demonstrations
- **Xuanyuan Zhang**: system implementation, test cases, implementation of demonstrations

References

[1] Paper.js offical site. http://paperjs.org/. Accessed: 2018-03-22.

[2] Snap svg offical site. http://snapsvg.io/. Accessed: 2018-03-22.

[3] Svgator offical site. https://www.svgator.com/. Accessed: 2018-03-22.