

VCL Final Project: Tiny SVG Renderer

郑懿轩

2026 年 1 月 12 日

1 选题介绍

可缩放矢量图形 (Scalable Vector Graphics, SVG) 是一种基于 XML 标记语言, 用于描述二维矢量图形的图片格式。不同于 PNG 等位图格式, SVG 直接描述组成图形的点、线、面, 然后通过渲染器渲染成屏幕可以显示的位图。

本项目基于课程 [Lab0](#) 的代码框架开发, 实现了一个简易的 SVG 渲染器。主要包含以下功能:

- 支持解析常见的 SVG 几何形状, 如 Rect, Circle, Ellipse, Line, Polygon, Path 等
- 支持解析常见的 SVG 变换 (transform), 如 Translate, Rotate, Scale, Skew, Matrix
- 支持解析常见的样式 (Style) 属性, 如 Fill, Stroke, Opacity 等
- 提供了 SSAA 反走样选项, 支持 1-16 倍采样倍数
- 支持导出图片

如果您已经配置好了 Xmake, 请在项目根目录下使用以下命令运行该渲染器:

```
xmake
xmake run svg
```

您可以查看根目录下的 README.md 获取更多详细信息。

2 实现思路

2.1 XML 解析模块

利用 [tinyxml2](#) 提供的解析方法进行解析。具体来说, 一个 SVG 文件就是一棵 DOM 树, 我们按照广度优先顺序遍历文件中的每个元素进行解析, 以下是主要元素及其属性的解析思路。

2.1.1 图元解析

为了简便起见, 我们可以将所有图元统一看作 Path (例如, 矩形可以看作从起始点向右走 width, 再向下走 height, 再向左走 width, 再回到起点)。有关 Path 的详细介绍可以在 [SVG tutorial](#) 中找到。简单地说, 一个 Path 是由多个命令绘制出来的一条路径, 为了渲染的方便, 我们在解析的时候将路径离散化为一条一条的线段。对于直线命令这是容易的, 只需要保存端点即可。但是 Path 需要我们支持更加灵活的曲线命令。曲线命令包括二次、三次贝塞尔曲线命令和弧命令。对于贝塞尔曲线命令, 我们可以沿用 Lab1 的 de Casteljau 算法的思想, 将曲线不断细分为多个更小的贝塞尔曲线, 直到它们足够平滑。具体实现参考了 [这篇博客](#)。弧命令要求我们画出一段旋转后的椭圆曲线, 在 [SVG 1.1 标准的附录 F.6 Elliptical arc implementation notes](#) 中有详细介绍实现细节。

2.1.2 变换解析

类比前向运动学的思路，我们可以在遍历过程中维护一个矩阵栈，用于保存当前变换矩阵，并使用齐次坐标简化数学运算。

2.1.3 样式解析

SVG 标准中提供了许多样式属性，由于时间有限，我只实现了基础的填充，描边，透明度属性。这部分的解析技术难点主要集中在字符串操作上。

2.2 渲染模块

该模块主要分为填充部分的渲染和描边部分的渲染。

2.2.1 填充

在解析模块中，我们已经将路径转化为一条条的线段，因此可以利用扫描线填充算法进行填充。值得注意的点是 SVG 有两种填充规则，分别是 nonzero 和 evenodd，对于 nonzero 我们对每条线段还要维护它的方向（+1 代表向上，-1 代表向下），经过每条边方向计数就加上它的方向。只有在方向计数不为 0 的区域才进行填充；对于 evenodd 就是简单的在奇数边和偶数边之间填充。

2.2.2 描边

SVG 支持多种描边样式，描边可以理解为在线段周围特定区域进行填充，具体区域的选择由描边样式决定。

2.2.3 反走样

注意到直接渲染出的图像锯齿现象较为严重，因此我复用了 Lab1 中的 SSAA 反走样技术的代码进行反走样。

2.3 UI 模块

2.3.1 文件操作

利用 [portable-file-dialogs](#) 库提供文件对话框，方便用户加载 SVG 文件和保存图片。

2.3.2 渲染完成提示

在每次 recompute 之后，设置一个消息显示时间。在 OnSetupPropsUI 函数中根据当前消息显示时间决定是否显示消息，若是则显示消息并相应减少时间。

3 实现效果

3.1 UI 界面说明

1. 放大镜工具，开启后可通过鼠标悬停放大区域像素（7.）
2. 加载您的 SVG 文件
3. 控制 SSAA 的采样频率，通过鼠标拖动更改

4. 应用 SSAA 反走样技术，注意拖动后需要应用更改才会生效
5. 将当前渲染好的图片导出为 PNG 格式图片
6. 画布，您的 SVG 文件将在这里渲染
7. 放大镜效果

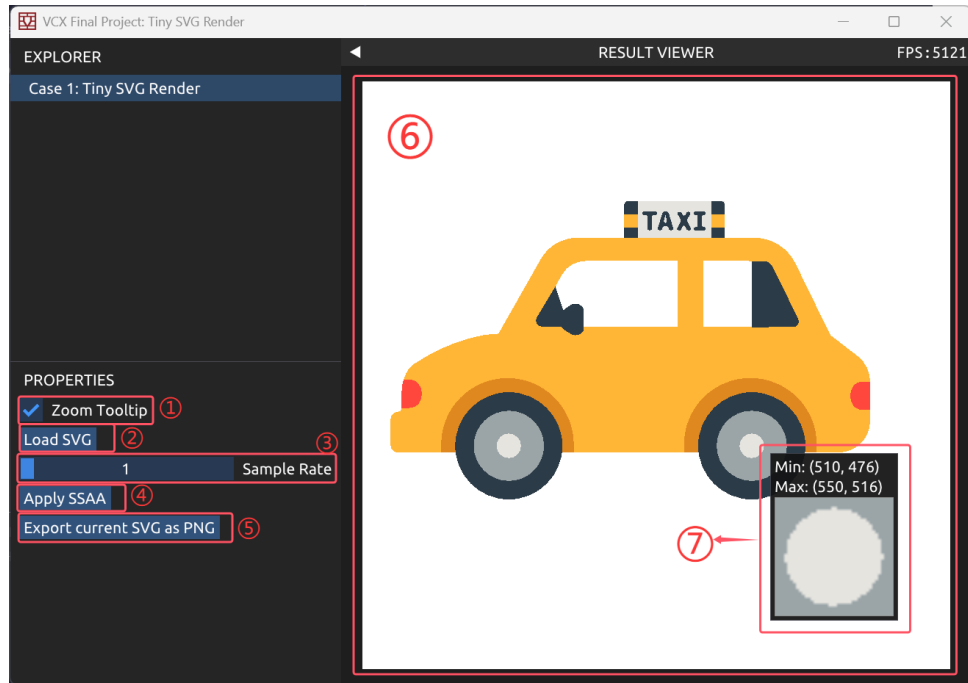


图 3.1: UI 界面

3.2 部分 SVG 图片渲染效果

在 `assets/images` 文件夹中提供了多张 SVG 图片，您也可以选择大部分不含高级样式的 SVG 图片尝试。以下展示 `assets/images/taxi.svg` 在无反走样和 16 倍 SSAA 反走样下的渲染结果：

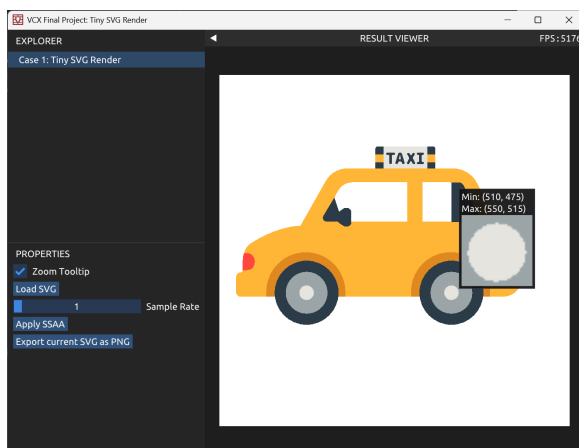


图 3.2: SSAA 1x

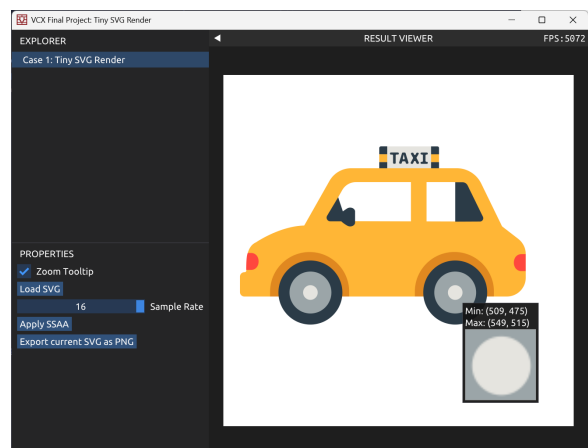


图 3.3: SSAA 16x