TU Hamburg-Harburg – Institut für Zuverlässiges Rechnen
Prof. Dr. S.M. Rump und Mitarbeiter, Wintersemester 2016/2017

Prozedurale Programmierung, Übungsblatt 05

letzter Abgabetermin 01. Dezember 2016

SVG Funktionsplotter

Aus den Erfahrungen der vergangenen Übungsblätter soll nun das erste kleine Programmierprojekt, ein Funktionsplotter für SVG-Grafiken, erstellt werden. Da SVG-Grafiken nur diskrete Elemente zeichnen können, reicht die Funktion `svg_line` aus der `libSVG.h` vollkommen aus, um eine Annäherung durch Geraden an die zu zeichnende Funktion (s. Abbildung 1 für x^2) zu bekommen. Je feiner das Gitter gewählt wird, umso weniger fällt dem Betrachter auf, dass es sich nicht wirklich um eine kontinuierliche Funktion handelt.

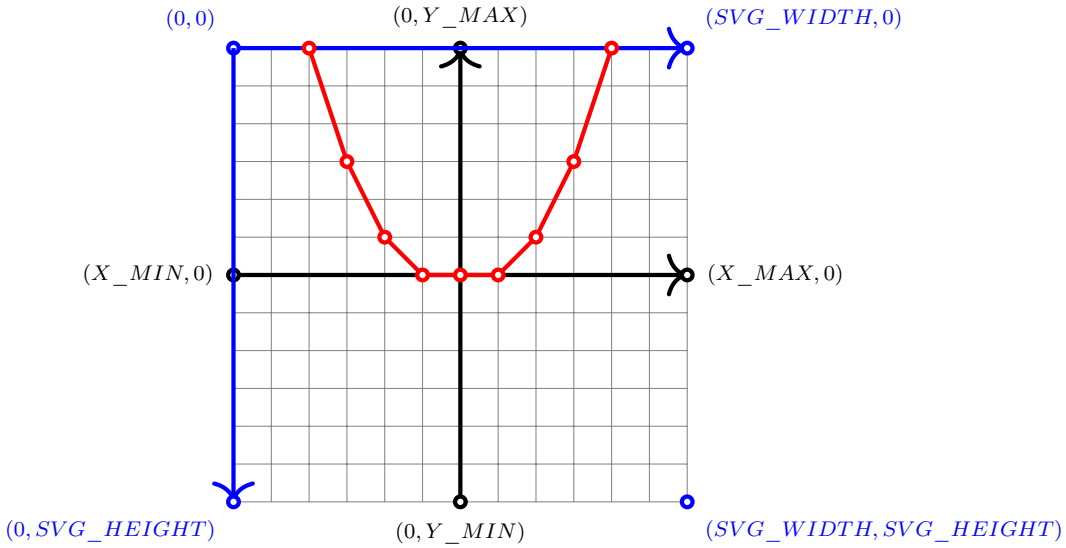

Abbildung 1: Mathematische Koordinaten (schwarz), SVG-Koordinaten (blau) und Funktionsplot aus stückweisen Geraden (rot).

Das Projekt ist sehr komplex. Nutzen Sie das Koordinatensystem (Aufgabe 2 vom Übungsblatt 04) und die `libSVG.h` als Ausgangspunkt für die weiteren Teilaufgaben.

1. Makro Konstanten

   Definieren Sie Makro Konstanten für die SVG- und mathematischen Koordinaten aus Abbildung 1: `SVG_HEIGHT`, `SVG_WIDTH`, `X_MIN`, `X_MAX`, `Y_MIN` und `Y_MAX`. Nutzen Sie diese Konstanten so oft wie möglich in den folgenden Aufgaben!

   (1 Punkt)

2. Übersetzung SVG- zu mathematischen Koordinaten

   In Abbildung 1 wird deutlich, dass sich das SVG- und das mathematische Koordinatensystem unterscheiden. Für eine gegebene SVG X-Koordinate x_{SVG} kann Formel (1) genutzt werden, um die mathematische X-Koordinate x_{mat} zu erhalten. Analog für y_{SVG} die Formel (2).

$$x_{mat} := X_{MIN} + \frac{x_{SVG}(X_{MAX} - X_{MIN})}{SVG_{WIDTH}} \tag{1}$$

$$y_{mat} := Y_{MIN} + \frac{y_{SVG}(Y_{MAX} - Y_{MIN})}{SVG_{HEIGHT}} \tag{2}$$

Schreiben Sie zwei Funktionen `x_svg2mat` und `y_svg2mat`, welche diese Formeln implementieren. Überlegen Sie sich, welche Datentypen für die Ein- und Ausgabe sinnvoll sind und wann in der Berechnung eine explizite Typumwandlung "verlustfrei" ist.

Hinweis: Denken Sie daran, dass bei Formel (2) noch nicht berücksichtigt wurde, dass die mathematische und SVG-Y-Achse entgegengesetzt sind!

(2 Punkte)

3. Übersetzung mathematische zu SVG-Koordinaten

   Stellen Sie analoge Überlegungen für die entgegengesetzte Richtung an und implementieren Sie zwei Funktionen `x_mat2svg` und `y_mat2svg`, welche mathematische in SVG-Koordinaten umrechnen.

   (2 Punkte)

4. Koordinatensystem

   Überarbeiten Sie das Koordinatensystem aus Aufgabe 2 vom Übungsblatt 04, indem Sie die Makrokonstanten, sowie die Umrechnungsfunktionen gebrauchen. Ziel dieser Teilaufgabe ist es, dass ihr Tutor die Konstanten aus dem ersten Aufgabenteil beliebig verändern kann und ein Koordinatensystem gemäß der Vorgaben in der SVG-Grafik gespeichert wird.

   (3 Punkte)

5. Funktionsplot

   Ermöglichen Sie es, dass ihr Tutor eine beliebige Funktion, z.B. aus der `<math.h>` mithilfe Ihres Programms grafisch darstellen kann.

   (2 Punkte)