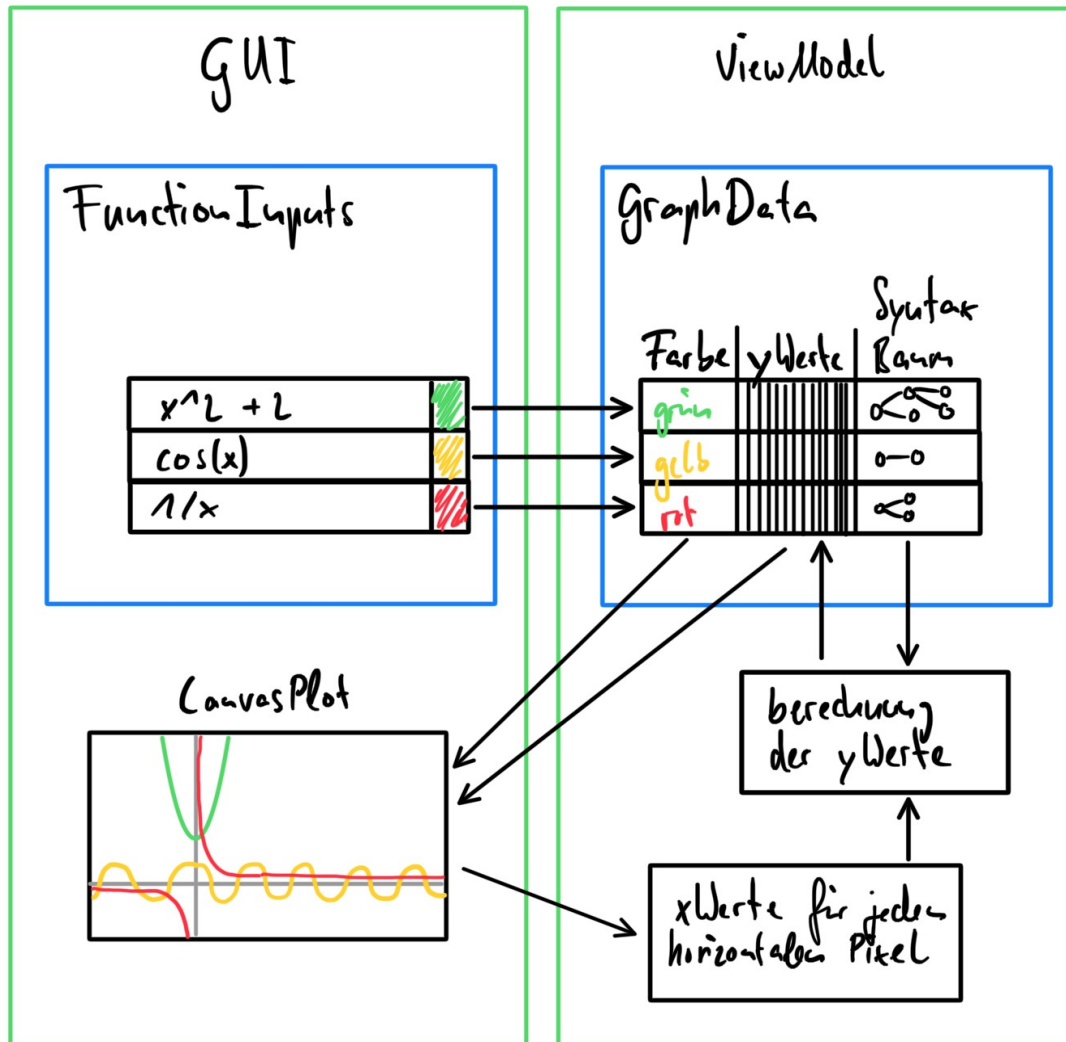


## ViewModel

Das ViewModel dient als Speicher für die Daten der Graphen. Die Interaktion zwischen GUI und ViewModel wird im folgendem Diagramm gezeigt:



Jeder einzelne `FunctionInput` registriert sich bei seiner Instanziierung beim `ViewModel`. Dieses legt für jeden Input ein `GraphData` Objekt in einer `HashMap` an. Daraufhin können bei Änderung der Farbe oder des Eingabestrings die hinterlegten Daten, auf die die `CanvasPlot` Klasse zum Zeichnen der Graphen zugreift aktualisiert werden. Die `CanvasPlot` Klasse aktualisiert die gespeicherten xWerte jedes mal, wenn diese sich ändern, was die Neuberechnung aller y Werte jeder einzelnen Funktion zur Folge hat. Wenn sich der Eingabestring eines `FunctionInputs` ändert werden nur die y Werte des `GraphData` Objekts neu berechnet welches mit dem jeweiligem `FunctionInputs` korrespondiert.

### Berechnung der y Werte

Bei der Berechnung der y Werte wird die Methode `double[] evalAll(double[] values)`, welche alle x Werte auf einmal evaluiert anstatt der Methode `double eval(double value)`, welche einen einzelnen x Wert evaluiert genutzt, da damit ein Schwachpunkt der iterativen Berechnung der y Werte teilweise behoben wird. Beim zeichnen des Funktionsgraphs wird zwischen jedem Punkt welcher sich aus dem x-Wert jedes Pixels der x-Achse und des errechneten y-Werts besteht eine Linie gezeichnet. Dadurch erhält man einen durchgehenden

Graphen anstatt einzelner Punkte ohne Verbindung zueinander. Bei Funktionen wie zum Beispiel  $1/x$  oder  $\tan(x)$ , welche im sichtbaren Bereich einen oder sogar mehrere Grenzwerte haben, werden Paare von Punkten, welche einen Grenzwert zwischen sich haben somit auch verbunden, obwohl diese im Graph nicht verbunden sein sollten. Um dies zu umgehen, wird bei allen **SyntaxNodes**, bei deren Evaluierung ein Grenzwert auftauchen kann (Division, Tangens, Logarithmus, Wurzel), in der Methode `evalAll` nach Paaren von Punkten „gesucht“ bei denen ein Grenzwert überschritten wurde um diese auf diesen Grenzwert zu korrigieren. Beim Zeichnen des Graphen dürfen somit Punkte mit Unendlichen y Werten (`Double.POSITIVE_INFINITY` oder `Double.NEGATIVE_INFINITY`) nicht gezeichnet werden, damit diese im Graphen nicht verbunden werden.

