

## Hilfetext zur Einfachen Grafik unter Scheme

Die einfache Grafik unter Scheme entstammt dem Paket graphics, das mit **(require graphics/graphics)** geladen wird.<sup>1</sup> Dann muss das Paket initialisiert werden mit **(open-graphics)**

### Grafikfenster erzeugen

Es muss zunächst ein Grafikfenster erzeugt werden. Dazu dient die Funktion **open-viewport**, die nach der Hilfe aufgerufen werden muss mit **(open-viewport name horiz vert)**. Ein viewport ist also einfach ein Fenster der grafischen Oberfläche mit einer standardmäßig weißen Zeichenfläche. Die Funktion gibt ein Fensterobjekt<sup>2</sup> zurück, das einer Variablen zugewiesen werden muss, um weiter darauf zugreifen zu können. In unserem Projekt können wir also beispielsweise **(define fenster (open-viewport "tsp" 500 500))** schreiben und bekommen dann ein 500 x 500 Pixel großes Fenster mit dem Namen **fenster** und der Titelzeile (Beschriftung) **tsp**.

Das Fenster wird nach Ausführen dieser Zeile sofort angezeigt und ist mit den einfachen folgenden Befehlen über seinen Namen **fenster** ansprechbar.

### Zeichnen

Es gibt mehrere vordefinierte Funktionen für Zeichenobjekte. Wir benötigen eigentlich nur Linien, die man mit **((draw-line viewport) p1 p2 [color])** zeichnen kann. Die Schreibweise zeigt, dass zunächst das Fensterobjekt eine Funktion zum Zeichnen einer Linie zurückgeben muss, der wir dann die benötigten Parameter übergeben. Das sind neben der Farbe, die man hier definieren kann, aber nicht muss, die Anfangsposition und die Endposition der Strecke.

### Positionen

Eine Position muss über eine **structure** erzeugt werden. Das geschieht mit **make**, der Aufruf heißt also **(make-posn x y)**.

### Eine Linie zeichnen

Wir können damit nun eine Linie zeichnen, indem wir der vom Fenster übergebenen Funktion die beiden Positionen und ggf. die Farbe übergeben.

**((draw-line fenster) (make-posn 10 20) (make-posn 30 100))**

zeichnet also eine schwarze Linie vom Punkt (10 / 20) zum Punkt (30 / 100). Die Achsorientierung ist dabei so, wie wir sie auch von Java kennen mit dem Ursprung in der linken oberen Ecke des Fensters, der x-Achse nach rechts und der y-Achse nach unten.

### Und die wird man die Grafik wieder los?

Zum Beenden dienen die beiden Aufrufe **(close-viewport fenster)** und **(close-graphics)**, die man geschickterweise in eine Hüllfunktion packt, die auf Mausclicks aus dem Fenster reagiert.

---

1 Hilfe-link dazu: [Racket/doc/graphics/index.html?q=graphics](http://Racket/doc/graphics/index.html?q=graphics), am besten Hilfecenter aufrufen und das Stichwort **graphics** eingeben.

2 Das hört sich nicht nur OO modelliert an.

```
(define
  (close)
  (close-viewport fenster) (close-graphics))
```

```
(define
  (ende)
  (viewport-flush-input fenster)
  ; leert den Puffer
  (get-mouse-click fenster)
  ; wartet auf Maus-klickt
  (close))
```