

Übung 5: Externe Daten und Mausinteraktion

In dieser Übung lernen Sie den Umgang mit externen Datenquellen und die Verarbeitung von Nutzeraktionen, wie das Bewegen der Maus.

Aufgabe 1: Externe Daten

Processing bietet die folgenden Befehle für das Einlesen und Verarbeiten einfacher (Text-)Dateien:

Funktion	Beschreibung
<code>loadStrings(dateiname)</code>	Lädt die Zeilen der Datei und gibt sie als String-Array zurück.
<code>splitTokens(zeile, trennzeichen);</code>	Trennt den gegebenen String <code>zeile</code> an allen Vorkommen von <code>trennzeichen</code> und gibt die Einzelteile als String-Array zurück.

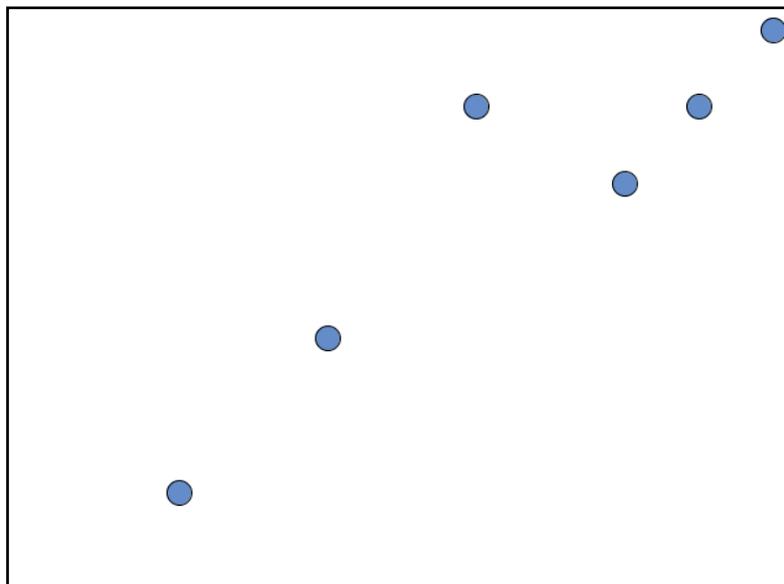
```
10;10
20;30
30;60
40;50
45;60
50;70
...
```

Mit Hilfe dieser Befehle soll nun eine externe Datei `punkte.txt` eingelesen werden. Diese Datei enthält eine Reihe von Punkten in einer Tabelle mit zwei Spalten (x-Wert, y-Wert). Der Inhalt der Datei kann z.B. so aussehen wie links angegeben.

→ Lesen Sie die Punkte aus der Datei zur Laufzeit ein und zeichnen Sie einen entsprechenden Scatterplot. Die eingelesenen x/y-Werte können z.B. in zwei Integer-Arrays gespeichert werden. Alternativ kann auch die Klasse `PVector` verwendet werden.

→ Erweitern Sie das Programm nun so, dass es den maximalen x-Wert und den maximalen y-Wert der Punkte bestimmt. Nutzen Sie diese Werte, um den Scatterplot auf die Fläche des Sketches zu strecken bzw. einzupassen.

Das Ergebnis könnte z. B. so aussehen:

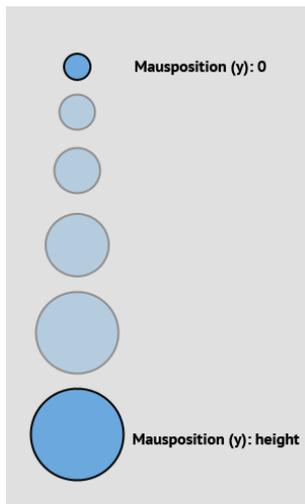


Aufgabe 2: Interaktion

Um die Mauskoordinaten auszulesen, sowie auf Klicks und Bewegungen der Maus reagieren zu können, stehen in Processing folgende Befehle bereit:

Funktion	Beschreibung
<code>mouseX, mouseY</code>	Zugriff auf die aktuellen Koordinaten der Maus
<code>pmouseX, pmouseY</code>	Zugriff die Koordinaten der Maus im letzten Frame
<code>mouseClicked()</code>	Wird aufgerufen sobald eine Maustaste gedrückt und losgelassen wird
<code>mousePressed()</code>	Wird aufgerufen, sobald eine Maustaste gedrückt wird
<code>mouseReleased()</code>	Wird aufgerufen, sobald eine Maustaste losgelassen wird
<code>mouseMoved()</code>	Wird aufgerufen, wenn die Maus bewegt und keine Maustaste gedrückt wird
<code>mouseDragged()</code>	Wird aufgerufen, wenn die Maus bewegt wird und eine Maustaste gedrückt wird

Mit Hilfe dieser Befehle soll der Sketch aus der vorherigen Aufgabe nun um Interaktionsmöglichkeiten erweitert werden.



→ Erweitern Sie den Sketch aus Aufgabe 1 so, dass ein linker Mausklick die Form der im Scatterplot dargestellten Datenpunkte verändert (z.B. Umschalten zwischen einer Darstellung als Kreis und einer Darstellung als Rechteck).

Des Weiteren soll die Größe der Datenpunkte mit der Maus verändert werden können. Während die rechte Maustaste gedrückt wird, soll sich beim Bewegen der Maus die Größe der Datenpunkte entsprechend der y-Koordinate der Maus anpassen. Befindet sich die Maus am oberen Rand der Zeichenfläche, so soll die Form eine Minimalgröße besitzen, am unteren Bildschirmrand entsprechend eine Maximalgröße. Dazwischen soll ein weicher Übergang zwischen den Größen stattfinden (Tipp: Die Funktion `map()` könnte hier sehr nützlich sein).

Das Ergebnis könnte z. B. so aussehen:

