# Vorbereitung 3 – Ereignisbehandlung

## Nachschlagen bei…

Unterkapitel von <https://www.inf-schule.de/software/gui/entwicklung_tkinter/ereignisverarbeitung> und <https://www.python-kurs.eu/tkinter_canvas.php>.

## Erklärung

Die Methode mainloop() ist eine ständige Schleife, die nach Ereignissen im Fenster Ausschau hält. Ereignisse sind zum Beispiel Mausklicks oder Tastendrücke. Diese Ereignisse können aufgefangen und mit bestimmten Funktionen zur Behandlung des Ereignisses verknüpft werden.

## Codebeispiel

from tkinter import \*

# Ereignisbehandlung

def linksVerschieben(event):

 (x0, y0, x1, y1) = tuple(canvas.coords(id\_kreis))

 if x0 >= 10:

 canvas.coords(id\_kreis, (x0-10, y0, x1-10, y1))

def rechtsVerschieben(event):

 (x0, y0, x1, y1) = tuple(canvas.coords(id\_kreis))

 if x1 <= 580:

 canvas.coords(id\_kreis, (x0+10, y0, x1+10, y1))

def obenVerschieben(event):

 (x0, y0, x1, y1) = tuple(canvas.coords(id\_kreis))

 if y0 >= 10:

 canvas.coords(id\_kreis, (x0, y0-10, x1, y1-10))

def untenVerschieben(event):

 (x0, y0, x1, y1) = tuple(canvas.coords(id\_kreis))

 if y1 <= 380:

 canvas.coords(id\_kreis, (x0, y0+10, x1, y1+10))

def objektLoeschen(event):

 canvas.delete(id\_kreis)

def vergroessern(event):

 (x0, y0, x1, y1) = tuple(canvas.coords(id\_kreis))

 canvas.coords(id\_kreis, (x0, y0, x1+10, y1+10))

def verkleinern(event):

 (x0, y0, x1, y1) = tuple(canvas.coords(id\_kreis))

 if x1-x0 > 10 and y1-y0 > 10:

 canvas.coords(id\_kreis, (x0, y0, x1-10, y1-10))

def plazieren(event):

 (x, y) = (event.x, event.y)

 (x0, y0, x1, y1) = tuple(canvas.coords(id\_kreis))

 canvas.coords(id\_kreis, (x, y, x+(x1-x0), y+(y1-y0)))

def entfernen(event):

 (x, y) = (event.x, event.y)

 (x0, y0, x1, y1) = tuple(canvas.coords(id\_kreis))

 if x0 <= x and x <= x1 and y0 <= y and y <= y1:

 canvas.delete(id\_kreis)

# Erzeugung der Komponenten

tkFenster = Tk()

tkFenster.title('Demo - Ereignisse')

tkFenster.geometry('400x300')

# Leinwand zum Zeichnen

canvas = Canvas(master=tkFenster, background='white')

canvas.place(x=10, y=10, width=380, height=280)

# Grafikobjekte

id\_kreis = canvas.create\_oval(100, 100, 120, 120, fill='red')

# Ereignisse

tkFenster.bind('<KeyPress-g>', vergroessern)

tkFenster.bind('<KeyPress-k>', verkleinern)

tkFenster.bind('<KeyPress-Up>', obenVerschieben)

tkFenster.bind('<KeyPress-Down>', untenVerschieben)

tkFenster.bind('<KeyPress-Left>', linksVerschieben)

tkFenster.bind('<KeyPress-Right>', rechtsVerschieben)

canvas.bind('<Button-1>', plazieren)

canvas.bind('<Button-3>', entfernen)

# Aktivierung der Ereignisschleife

tkFenster.mainloop()

Weitere Beispiele für Ereignisse:

| **Event-Sequenz** | **Ereignis** |
| --- | --- |
| '<Button-1>' | die linke Maustaste wird angeklickt |
| '<Button-3>' | die rechte Maustaste wird angeklickt |
| '<Double-Button-1>' | die linke Maustaste wird doppelt angeklickt |
| '<ButtonRelease-1>' | die linke Maustaste loslassen |
| '<Enter>' | der Mauszeiger wird auf eine GUI-Komponente bewegt |
| '<Leave>' | der Mauszeiger verlässt eine GUI-Komponente |
| '<Motion>' | der Mauszeiger wird innerhalb einer GUI-Komponente bewegt |
| '<KeyPress-A>' | die A-Taste wird gedrückt |
| '<KeyPress-Left>' | die Pfeil-nach-links-Taste wird gedrückt |
| '<KeyRelease-Bt>' | die B-Taste wird losgelassen |

Folgender Code ist brauchbar für das Rechteck aus der letzten Übung:

#defines all keypresses

def keypress(event):

 if event.keysym == "Left":

 rechteck.xvel=-10

 elif event.keysym == "Right":

 rechteck.xvel=10

def keyrelease(event):

 if event.keysym == "Left":

 rechteck.xvel=0

 elif event.keysym == "Right":

 rechteck.xvel=0

fenster.bind("<Key>",keypress)

fenster.bind("<KeyRelease>",keyrelease)

## Es werden nur die Ereignisse „Taste drücken“ und „Taste loslassen“ an das Fenster gebunden. In der Funktion keypress() wird dann mit event.keysym abgefragt, um welche Taste es sich handelt und entsprechend reagiert.

## Aufgaben

1. Kopier das Beispielprogramm in eine Datei ereignisbehandlung.pyw und löse alle Ereignisse, die vorkommen aus.
2. Speichere die Datei rechteck.pyw unter dem neuen Namen rechteck\_keyevent.pyw. Kopiere den Code mit keypress(), keyrelease() und fenster.bind() in das neue Programm. Ändere den vorhandenen Code so ab, dass sich das Rechteck nicht mehr von selbst bewegt, sondern nur noch, wenn man die linke oder rechte Pfeiltaste drückt. Die Kontrolle, ob das Rechteck am Rand ankommt und umkehren muss entfällt damit auch.