Informatik Klasse 13, Foliensatz 7
Maus-Ereignisse

Prof. G. Kemnitz

Institut für Informatik, Technische Universität Clausthal
29. Oktober 2009

Mouse-Ereignisse

Syntax: Zeichenkette der Form "<*Ereignisname*>"

- Button-i, Double-Button-i, Triple-Button-i: ein- zwei- und dreifacher Maus-Klick
- Bi-Motion: Maus-Bewegung bei gedrückter Taste
- ButtonRelease-i: Loslassen der Maus-Taste
- Enter: Maus-Zeiger tritt in das Fenster ein
- Leave: Maus-Zeiger verlässt dast Fenster

($i \in \{1, 2, 3\}$ – 1: linke, 2: mittlere, 3: rechte Maus-Taste).

Attribute des Ereignisobjekts:

- x, y: Mouse-Zeiger-Koordinaten in Pixeln relativ zur oberen linken Fensterecke
- widget: Zeiger auf das Widget-Objekt.

Experiment mit Maus-Ereignissen

```
from Tkinter import *
class MyLabel(Label):
 name=""
class MyFrame(Frame):
 name="Frame"
```

Hier die Definition der Klasse »MyApp« von der nächsten Folie einfügen.

```
root = Tk();
app = MyApp(root);
root.mainloop()
```

- Was bewirken die Klassendefinitionen?
- In »MyApp« soll »Leave« nacheinander durch alle Maus-Ereignistypen ersetzt werden.

```
class MyApp:
 l=[0, 0, 0]

 def __init__(self, master):
  frame=MyFrame(master)
  frame.pack()
  frame.bind("<Leave>", self.Action)
  for idx in range(3):
   self.l[idx] = MyLabel(frame, width=12, height=2,
     relief=RIDGE, bg="#ffaaaa")
   self.l[idx].name = "Label "+str(idx);
   self.l[idx].grid(row=idx, column=idx)
   self.l[idx].bind("<Leave>", self.Action)

 def Action(self, EvObj):
  self.l[0].config(text=EvObj.widget.name)
  self.l[1].config(text="x="+str(EvObj.x))
  self.l[2].config(text="y="+str(EvObj.y))
```

- Wenn zwei Widgets übereinander angeordnet sind, wird das Maus-Ereignis an beide oder nur an das obere weitergeleitet?
- Gilt eine Bewegung in ein Vordergrundfenster als Verlassen des Hintergrundfensters?

Klausuraufgaben

- Aufgabe 1: Klasse mit Attributen und Methoden aus einer Textaufgabenstellung konstruieren; Testbeispiel für die Klasse entwickeln
- Aufgabe 2: Eine Oberfläche nach Bildschirmfoto aus Labeln, Button und Eingabe-Widgets zusammensetzen; Attribute (Farben, Schrift, Umrandung, Cursor) und Anordnung; keine Methoden
- Aufgabe 3: Klasse mit Labeln und an Maus-Ereignisse gebundene Methoden plus Instanz-Erzeugung aus einer Textaufgabe und Bildschirmfoto entwickeln; Instanz-Aufruf

Alle Lösungen in vorgegenen Verzeichnissen unter vorgegebenen Namen zu speichern. Erlaubte Hilfsmittel: Folien, Dokumente auf der Webseite, eigenen Lösungen und Notizen auch schriftlich

Aufgabe 7.1: Untersuchung von Mehrfach-Klicks

- Werden bei einem Triple-Klick-Ereignis keine, einzelne oder mehrere Einzel- und Doppel-Klick-Ereignisse ausgelöst?

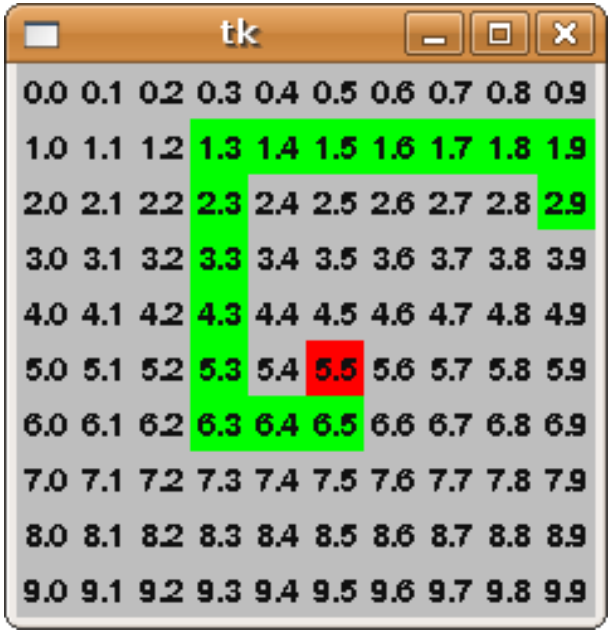
Überlegen Sie sich eine geeignete Test-Applikation, mit der Sie diese Frage beantworten können.

Hinweis: Sie benötigen für alle drei Ereignistypen einen Ereignishandler, der die Ereignisse so mitschreibt, dass Sie sie anhand der Maus-Koordinaten einander zuordnen können.

Aufgabe 7.2: Maus-Spur 1

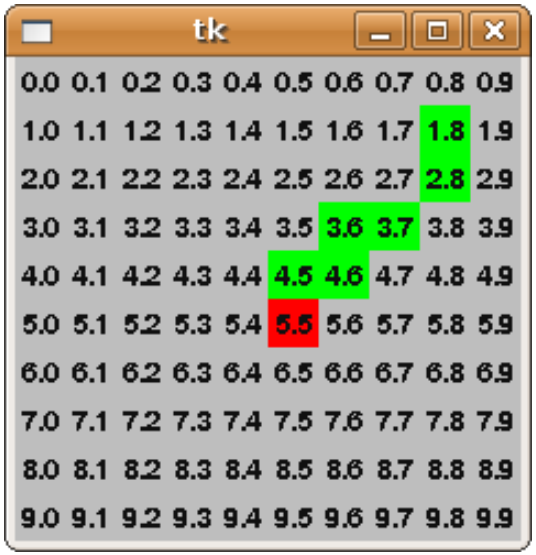
Es ist eine Anwendung mit einer 10×10 Label-Matrix als Oberfläche zu entwickeln.

- In jedem Label sollen Zeilen und Spaltennummer stehen.
- Das Label, über dem sich der Maus-Zeiger gerade befindet, ist rot und verlassene Label sind grün darzustellen.



Aufgabe 7.3: Maus-Spur 2

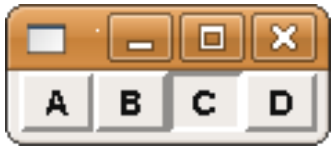
Wie Aufgabe zuvor, nur sollen immer nur die letzten sechs verlassen Label grün dargestellt und die übrigen auf grau zurückgesetzt werden.



Hinweis: Es ist eine Liste mit den verlassenen Labeln mitzuführen, aus der immer das erste Label entnommen und grau gefärbt und und das neue grün gefärbte Label hinten angehängt wird.

Aufgabe 7.4: Tasten-Auswahlfeld

Schreiben Sie eine Appliktion mit einem Tastenfeld aus vier Labeln mit den Beschriftungen »A« bis »D«, bei dem zum Programmstart die erste Taste mit »relief=SUNKEN« und die übrigen mit »relief=RAISED« dargestellt werden. Bei einem Einfach-Klick auf eine Taste soll diese mit »relief=SUNKEN« und die anderen mit »relief=RAISED« dargestellt werden.



Hinweis:
Anordnung von links nach rechts mit »pack(side=LEFT)«.

Aufgabe 7.5: Feinmotorik-Test 1

Ändern Sie die Applikation mit dem Tastenfeld aus Aufgabe 6.3 so ab, dass im Ausgabenfeld unten bei Tastenbetätigung der Tastenwert und der relative Abstand des Maus-Zeigers zum Mittelpunkt mal 100 als »Treffer« angezeigt wird.

Hinweise:

- Button-Größe: width=2 und height=1, Cursor »circle«
- Pixelgröße experimentell mit »Leave« bestimmen
- der Mittelpunkt des Labels hat »Treffer 100« und die Eckpunkte haben »Treffer 0«

Aufgabe 7.6: Feinmotorik-Test 2

Erweitern Sie das Programm um eine Start-Taste, eine Stop-Taste und eine Ausgabe, in der nach Betätigung der Stopp-Taste die mittlere Anzahl der Klicks pro Sekunde und die mittlere Trefferanzahl angezeigt wird.

Hinweis: Bestimmung der aktuellen Zeit in Sekunden als Gleitkommazahl:

```
import time
t = time.time()
```