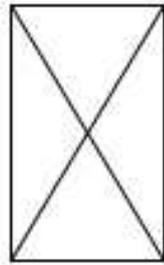
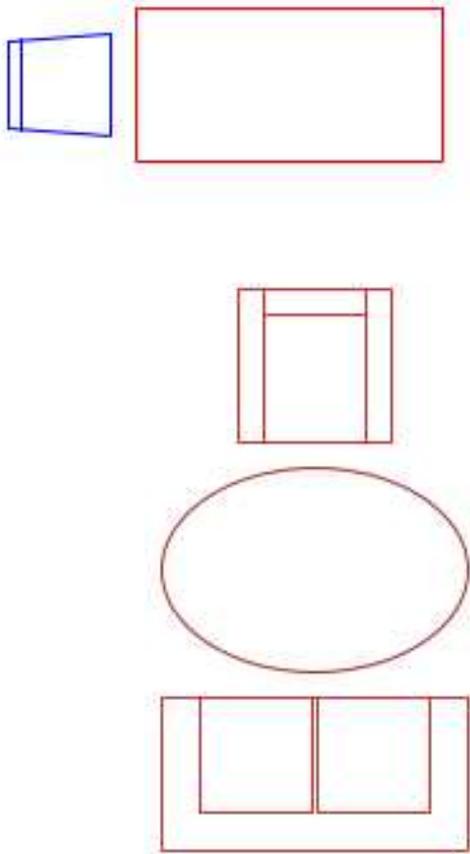


Mausereignisse und Gui

Einbau von Mausereignissen
und
Vervollständigung der
Funktionalität der Gui
in mehreren Schritten

Mausereignisse und Gui



Mausereignisse und Gui

- Im nächsten Schritt werden neben dem Mausklick auch die anderen relevanten Mausereignisse aktiviert.
- Die umfangreichste Erweiterung in RaumplanerModell ergibt sich aus dem Ziehen mit der Maus.
- Dafür muss die Mausposition verwaltet werden. In der Methode Linksklick:

```
def Linksklick(self, x, y):
```

```
...
```

```
self.__clickPosition=(x,y)
```

Mausereignisse und Gui

- Entsprechend für die anderen Methoden:

```
def MausLosgelassen(self,x,y):
```

```
    if self.__ausgewaehlt == -1 : return
```

```
    self.__clickPosition=None
```

```
def GezogenAuf(self,x,y):
```

```
    if self.__ausgewaehlt == -1 : return
```

```
    dX=x-self.__clickPosition[0]
```

```
    dY=y-self.__clickPosition[1]
```

```
    self.__alleMoebel[self.__ausgewaehlt].Bewege(dX,dY)
```

```
    self.__clickPosition=(x,y)
```

Mausereignisse und Gui

- Für den Rechtsklick wird man nicht allein das Auswählen erwarten, sondern dass sich damit ein Dialogfenster zum Konfigurieren des Objekts öffnet.
- Dieses wird in einem Modul `konfigurationsDialog.py` realisiert.
- `RaumplanerModell` ruft auf:

```
dialog=KonfigurationsDialog(Zeichenflaeche.GibZeichenflaeche(), -1,  
                             "Objekt konfigurieren", size=(350, 200),  
                             style=wx.DEFAULT_DIALOG_STYLE)
```

Mausereignisse und Gui

- Im Beispiel erfolgte ein Rechtsklick auf das Symbol des Tischobjekts.
- Alle Attributwerte können bearbeitet werden und die Änderungen werden verarbeitet, wenn auf OK geklickt wird.



The image shows a dialog box titled "Objekt konfigurieren" with a close button (x) in the top right corner. It contains six input fields for configuring an object:

Attribut	Wert
x-Position	60
y-Position	20
Breite	120
Tiefe	60
Winkel	0
Farbe	red

At the bottom of the dialog box, there are three buttons: a help button with a question mark (?), an "Abbrechen" (Cancel) button, and an "OK" button.

Mausereignisse und Gui

- Zusätzlich wird für das Öffnen (und Speichern) je ein Dateidialog eingebaut:

```
dlg = wx.FileDialog(  
    self, message=u"Datei wählen",  
    defaultDir=standardVerzeichnis,  
    defaultFile="daten.dat",  
    style=wx.FD_OPEN)  
if dlg.ShowModal() == wx.ID_OK:  
    dateiname = dlg.GetPath()  
    antwort=self.__modell.Lade(dateiname)  
    ...
```

Mausereignisse und Gui

- Der Rest sind „Feinarbeiten“:
 - Vervollständigung des Moebel-Menues
 - ausgewählt in Textfeld anzeigen (*message an Gui*)
 - dazu muss das RaumplanerModell die Gui kennen
 - vom Ausgewählten wird auch der Klassenname angezeigt
 - Bild speichern mit Dateidialog
 - Bearbeiten-Menue mit Auswahl, Verberge, Zeige, Lösche

Mausereignisse und Gui

- Spätestens jetzt sollte die Frage kommen, ob denn der Begriff RaumplanerModell noch passt.

Mausereignisse und Gui

- Für den Rechtsklick wird man nicht allein das Auswählen erwarten, sondern dass sich damit ein Dialogfenster zum Konfigurieren des Objekts öffnet.
- Dieses wird in einem Modul `konfigurationsDialog.py` realisiert.
- `RaumplanerModell` ruft auf:

```
dialog=KonfigurationsDialog(Zeichenflaeche.GibZeichenflaeche(), -1,  
                             "Objekt konfigurieren", size=(350, 200),  
                             style=wx.DEFAULT_DIALOG_STYLE)
```