Oberfläche mit GridBagSizer

Die Gestaltung einer
Oberfläche mit GridBagSizer
zum Raumplaner

Oberfläche mit GridBagSizer

- Ziel ist, zum Raumplaner eine Oberfläche zu gestalten, die (*weitgehend*) ohne Menue auskommt.
- Da nun sehr viele Elemente auf dem Panel des Grafikfensters dargestellt werden, wird eine Darstellung mit einem *Sizer* hergestellt, der eine automatisiertes einheitliches Layout gewährleistet.
- Wir verwenden einen GridBagSizer.

Oberfläche mit GridBagSizer

- Basis sind
  - das Raumplanerprojekt, in das die Gui vollständig integriert ist.
  - die Datei *Demobeispiel_GridBag-Sizer.py*

Oberfläche mit GridBagSizer

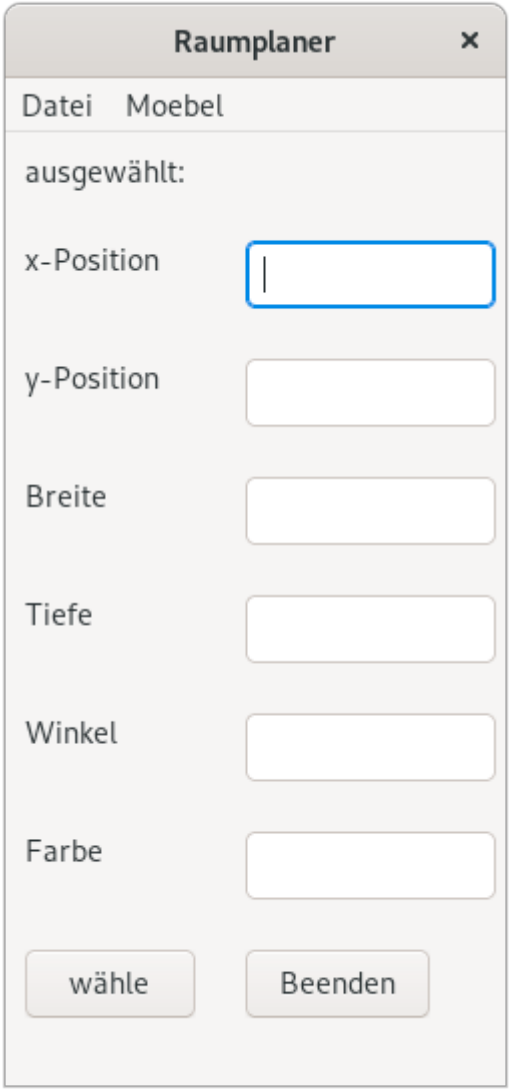
- Auf der Oberfläche sollen zeilenweise jeweils mit Beschriftung (Label -> *StaticText*) und Eingabefeld (*TextCtrl*) dargestellt werden:
  - x-Position
  - y-Position
  - Breite
  - Tiefe
  - Winkel (Orientierung)
  - Farbe

Oberfläche mit GridBagSizer

- Außerdem sollen Buttons zum Auswählen der Moebel-Objekte und zum Beenden vorhanden sein.
- Hilfreich ist auch eine Meldung, welches Objekt gerade ausgewählt ist. Das kann ein StaticText sein, da nichts einzugeben ist, sondern nur ein Wert dargestellt werden soll.

Oberfläche mit GridBagSizer

- Um alles unterzubringen muss die Fenstergröße geändert werden.



Oberfläche mit GridBagSizer

- Arbeitsschritte in der Klasse Gui
  - Erzeugen des GridBagSizer-Objekts

```
gbs = self.gbs = wx.GridBagSizer(vgap=5, hgap=5)
```

  - Erzeugen der Elemente, beispielsweise:

```
labelX = wx.StaticText(panel, -1, "x-Position")
self.textCtrlX = wx.TextCtrl(panel, -1,"",
                             size=(125, -1),
                             style=wx.TE_PROCESS_ENTER)
```

Oberfläche mit GridBagSizer

- Aufbau in gbs definieren

gbs.Add(label1, (0,0), flag=wx.ALL, border=10)
gbs.Add(self.labelName, (0,1), flag=wx.ALL, border=10)
gbs.Add(labelX, (1,0), flag=wx.ALL, border=10)
gbs.Add(self.textCtrlX, (1,1), flag=wx.ALL, border=10)
gbs.Add(labelY, (2,0), flag=wx.ALL, border=10)
gbs.Add(self.textCtrlY, (2,1), flag=wx.ALL, border=10)
...
gbs.Add(labelW, (5,0), flag=wx.ALL, border=10)
gbs.Add(self.textCtrlW, (5,1), flag=wx.ALL, border=10)
gbs.Add(labelF, (6,0), flag=wx.ALL, border=10)
gbs.Add(self.textCtrlF, (6,1), flag=wx.ALL, border=10)
gbs.Add(waehleButton, (7,0), flag=wx.ALL, border=10)
gbs.Add(beendenButton, (7,1), flag=wx.ALL, border=10)

Oberfläche mit GridBagSizer

- gbs dem Panel-Objekt zuweisen und Layout umsetzen lassen.

panel.SetSizerAndFit(gbs)

Oberfläche mit GridBagSizer

- Ereignisbehandlung für das Ereignis EVT_TEXT_ENTER der TextCtrl-Objekte binden:

```
self.Bind(wx.EVT_TEXT_ENTER, self.OnTextEnterX, self.textCtrlX)
self.Bind(wx.EVT_TEXT_ENTER, self.OnTextEnterY, self.textCtrlY)
self.Bind(wx.EVT_TEXT_ENTER, self.OnTextEnterB, self.textCtrlB)
self.Bind(wx.EVT_TEXT_ENTER, self.OnTextEnterT, self.textCtrlT)
self.Bind(wx.EVT_TEXT_ENTER, self.OnTextEnterW, self.textCtrlW)
self.Bind(wx.EVT_TEXT_ENTER, self.OnTextEnterF, self.textCtrlF)
```

Oberfläche mit GridBagSizer

- Ereignisbehandlung realisieren (beispielsweise):

```
def OnTextEnterX(self, event):
    ausgewaehlt=self.__modell.GibAusgewaehltes()
    if ausgewaehlt==None: return
    alterWert=ausgewaehlt.GibX()
    inhalt=self.textCtrlX.GetValue()
    try:
        neuerWert=int(inhalt)
        self.__modell.BewegeHorizontal(neuerWert-alterWert)
    except ValueError as e:
        self.textCtrlX.SetValue('Eingabefehler')
        return
```

Oberfläche mit GridBagSizer

- Waehle neu:

```
def OnWaehle(self, event):
    ausgewaehlt=self.__modell.Waehle()
    if ausgewaehlt==-1:
        self.labelName.SetLabel('')
        self.textCtrlX.SetValue('')
        ...
        return
    if type(ausgewaehlt)==int: ausgewaehlt+=1
    self.labelName.SetLabel(str(ausgewaehlt))
    aktuell=self.__modell.GibAusgewaehltes()
    self.textCtrlX.SetValue(str(aktuell.GibX()))
    ...
    self.textCtrlF.SetValue(aktuell.GibFarbe())
```