

Steuerung mit Buttons (Umsetzung)

Umsetzung der
Steuerung der Möbelobjekte
mit Hilfe von Buttons

Steuerung mit Buttons

Basis: Kursergebnis von Termin 7:

Raumplaner-vollstaendig-Gui-Kurs.zip

- Erster Schritt: „Aufräumen“
 - Datei *Demobeispiel_mit_Menue.py* umbenennen in *gui.py*
 - Klasse umbenennen in `Gui`
 - App-Bereich in `RaumplanerApp` verschieben und editieren

Steuerung mit Buttons

Basis: Kursergebnis von Termin 7:
Raumplaner-vollstaendig-Gui-Kurs.zip

- Erster Schritt: „Aufräumen“
 - Datei *Demobeispiel_mit_Menue.py* umbenennen in *gui.py*



Steuerung mit Buttons

Basis: Kursergebnis von Termin 7:

Raumplaner-vollstaendig-Gui-Kurs.zip

- Erster Schritt: „Aufräumen“
 - Klasse umbenennen in Gui

```
### -----  
class Gui(wx.Frame):  
    def __init__(  
        self, parent, ID, title, pos=wx.DefaultPosition,  
        size=wx.DefaultSize, style=wx.DEFAULT_FRAME_STYLE  
    ):  
  
        wx.Frame.__init__(self, parent, ID, title, pos, size, style)  
        panel = wx.Panel(self, -1)
```

Steuerung mit Buttons

Basis: Kursergebnis von Termin 7:
Raumplaner-vollstaendig-Gui-Kurs.zip

- Erster Schritt: „Aufräumen“
 - App-Bereich in RaumplanerApp verschieben und editieren

```
from gui import Gui

### -----

class RaumplanerApp(wx.App):
    """Testanwendung fuer Raumplaner-Grafik
    ShellFrame: Interaktion mit laufender Anwendung
    FillingFrame: Anzeige der Umgebung"""
    def OnInit(self):
        self.gui = Gui(None, 1000, "Demobeispiel")
        self.SetTopWindow(self.gui)
        self.gui.Show(True)

        self.fenster = GrafikFenster(self.gui, "Raumplaner-Grafik")
        self.fenster.Show(True)
```

Steuerung mit Buttons

Projekt 1:

Uebung-Drehung-mit-Buttons

- Zweiter Schritt: Button zur Steuerung
 - einbauen
 - Ereignisbehandlung verknüpfen
 - Methoden dafür anlegen

Steuerung mit Buttons

Projekt 1:

Uebung-Drehung-mit-Buttons

- Zweiter Schritt: Button zur Steuerung
 - einbauen
 - Ereignisbehandlung verknüpfen

```
dreheLinksButton = wx.Button(panel, 1004, "drehe links")
dreheLinksButton.SetPosition((15, 115))
self.Bind(wx.EVT_BUTTON, self.OnDreheLinks, dreheLinksButton)
```

```
dreheRechtsButton = wx.Button(panel, 1005, "drehe rechts")
dreheRechtsButton.SetPosition((185, 115))
self.Bind(wx.EVT_BUTTON, self.OnDreheRechts, dreheRechtsButton)
```

Steuerung mit Buttons

Projekt 1:

Uebung-Drehung-mit-Buttons

- Zweiter Schritt: Button zur Steuerung

```
def OnDreheLinks(self, event):  
    print('drehe links')
```

```
def OnDreheRechts(self, event):  
    print('drehe rechts')
```

- Methoden dafür anlegen

Steuerung mit Buttons

Projekt 1:

Uebung-Drehung-mit-Buttons

- Dritter Schritt: Anforderungen klären
 - in Schritten von jeweils 30° rechts herum (*im Uhrzeigersinn*) oder links herum
 - Wer wird gesteuert: Auswahl klären
 - Voraussetzungen dafür klären

Steuerung mit Buttons

Projekt 1:

Uebung-Drehung-mit-Buttons

- Dritter Schritt: Anforderungen klären
 - Voraussetzungen dafür klären

```
self.Bind(wx.EVT_CLOSE, self.OnCloseWindow)
```

```
self.liste = []  
self.auswahl = -1
```

im Konstruktor

```
def OnStuhl(self, event):  
    print('Stuhl erzeugen')  
    self.liste.append(Stuhl(sichtbar=True))  
  
def OnTisch(self, event):  
    print('Tisch erzeugen')  
    self.liste.append(Tisch(sichtbar=True))  
  
def OnSessel(self, event):  
    print('Sessel erzeugen')  
    self.liste.append(Sessel(sichtbar=True))
```

Steuerung mit Buttons

Projekt 1:

Uebung-Drehung-mit-Buttons

- Dritter Schritt: Anforderungen klären
 - Wer wird gesteuert: Auswahl klären

```
def OnWaehle(self, event):  
    print('waehle')  
    self.auswahl += 1  
    if self.auswahl == len(self.liste):  
        self.auswahl = -1
```

Steuerung mit Buttons

Projekt 1:

Uebung-Drehung-mit-Buttons

- Vierter Schritt: Anforderungen umsetzen
 - Methoden auch inhaltlich umsetzen

```
def OnDreheLinks(self, event):  
    print('drehe links')  
    if self.auswahl != -1:  
        self.liste[self.auswahl].Drehe(-30)  
  
def OnDreheRechts(self, event):  
    print('drehe rechts')  
    if self.auswahl != -1:  
        self.liste[self.auswahl].Drehe(30)
```