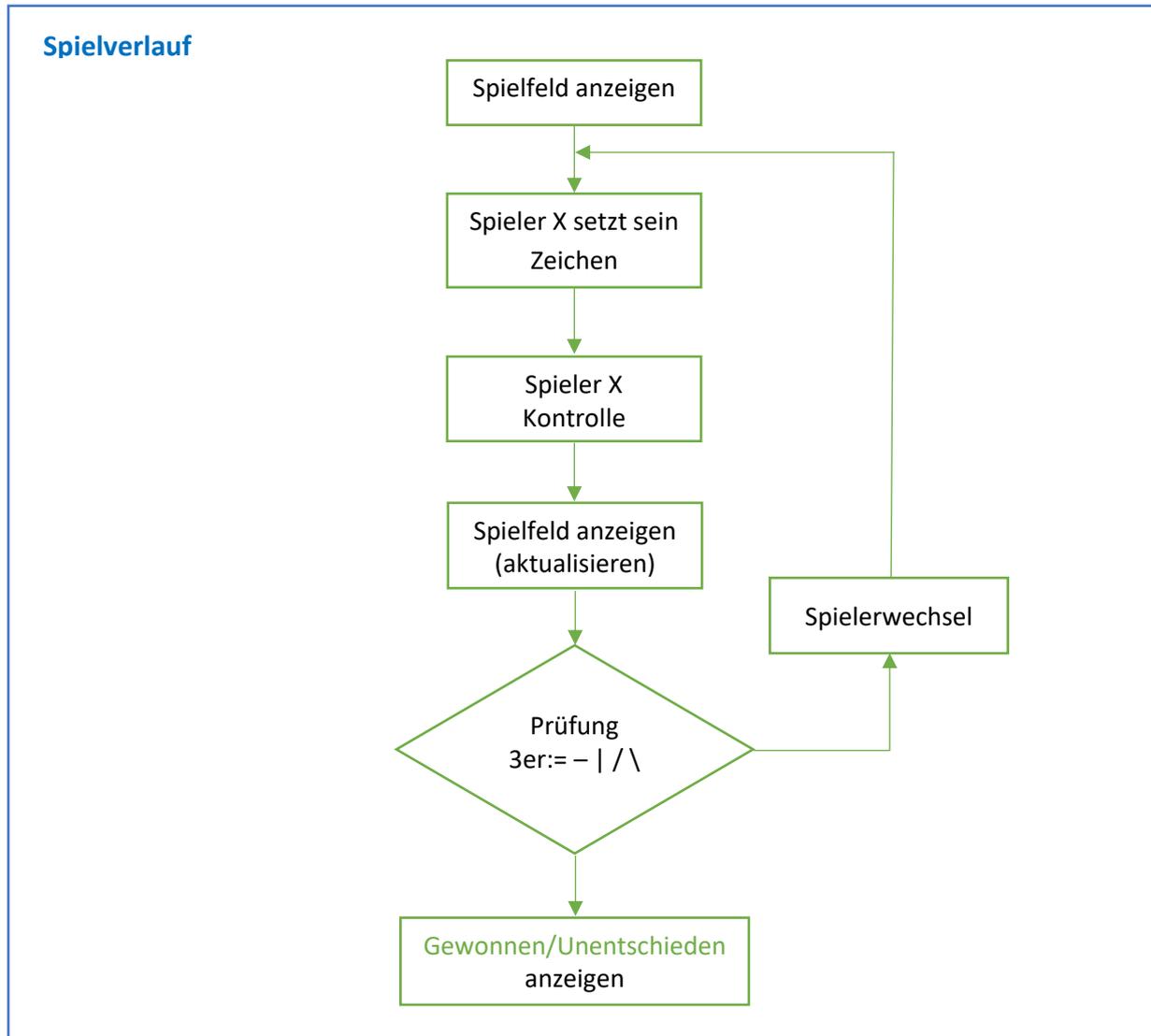
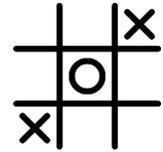


### Aufgabe 7b (schwer) - (Teil 4 von TIC TAC TOE)

**Situation:** Laut einer aktuellen VuMA-Umfrage spielen mehr als 40 Prozent der Deutschen Computer- und Videospiele. In der Altersgruppe der 14- bis 29-Jährigen beträgt der Anteil der Videospieler sogar über 70 Prozent. Die meisten Gamer interessieren sich für Action-Spiele/ Ego-Shooter, Abenteuer-Spiele und Geschicklichkeitsspiele.



Setzen wir das bekannte Spiel **TIC TAC TOE** als Python-Programm fort. Vom **Spielverlauf** haben wir bereits einige **Funktionen** (grün markiert) erstellt. Lösen wir die nächsten zwei Probleme:



Folgende **Funktionen** leiten wir vom Spielverlauf ab:

Spielfeld anzeigen

Spieler Eingabe

Kontrolle der Spielereingabe

Spielerwechsel

Prüfung 3er

Prüfung unentschieden

`spielfeld_ausgeben()`

`spieler_eingabe()`

`spieler_eingabekontrolle()`

`spieler_wechsel()`

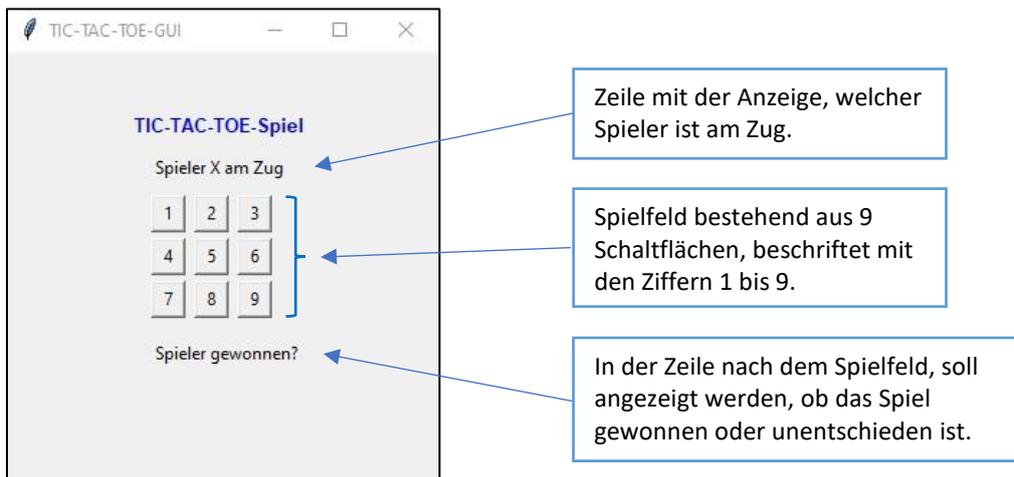
`spiel_gewonnen()`

`spiel_unentschieden()`

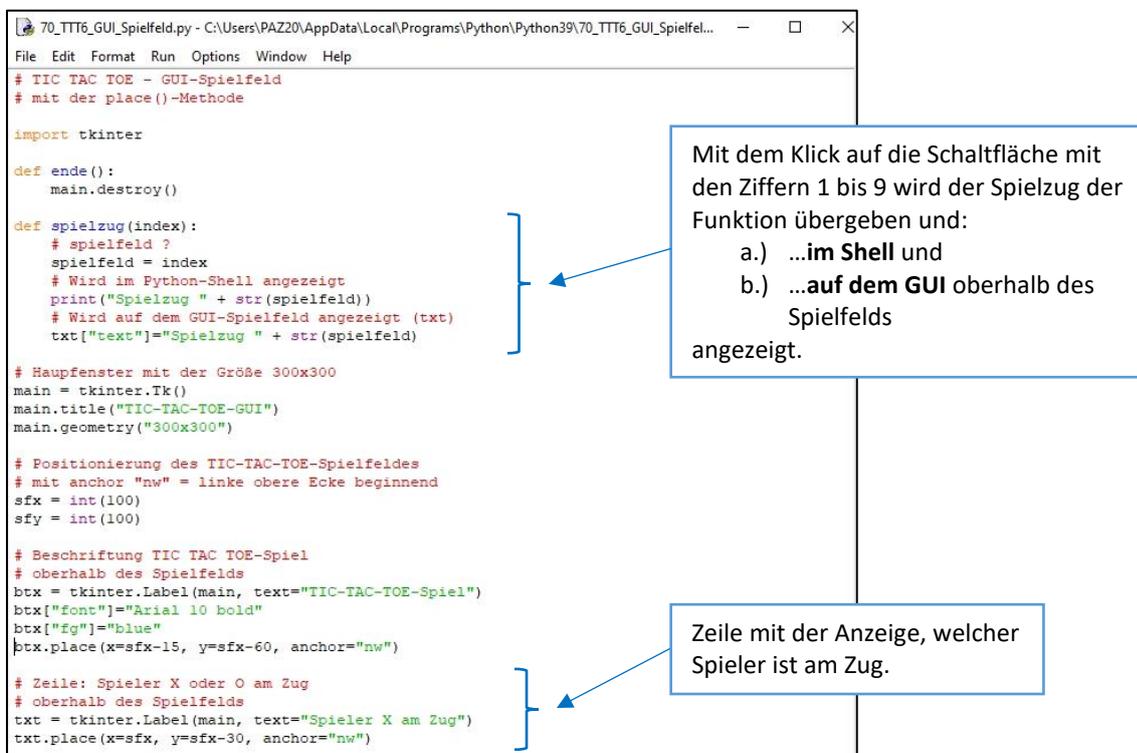
Das Spiel zu TIC TAC TOE haben wir fertig programmiert und es läuft im Python Shell. Wir haben auch die letzte Funktion mit dem Unentschieden gelöst. Nun wollen wir eine Benutzeroberfläche (GUI-Oberfläche) erstellen.

e.) Benutzeroberfläche zu TIC TAC TOE

Jetzt geht es darum unserem Spiel TIC TAC TOE eine Benutzeroberfläche zu geben. Dazu werden wir folgende Vorlage des GUI-Spiels verwenden.



Zuerst konzentrieren wir uns auf die Umsetzung unseres Spiels, dann können wir weiter am Design des TIC-TAC-TOE-GUIs arbeiten. Für das Layout der GUI-Anwendung wurde nicht das pack() sondern das **place()** Geometriemanager verwendet.



...weiter auf nächster Seite

## Programmierübungen zum Kurs: Python – Einstieg in die Programmierung

```
# Spielfeld aus Buttons mit fester Größe
b1 = tkinter.Button(main, text="1", command=lambda:spielzug(1), height = 1, width = 2)
b1.place(x=sfx, y=sfy, anchor="nw")
b2 = tkinter.Button(main, text="2", command=lambda:spielzug(2), height = 1, width = 2)
b2.place(x=sfx+30, y=sfy, anchor="nw")
b3 = tkinter.Button(main, text="3", command=lambda:spielzug(3), height = 1, width = 2)
b3.place(x=sfx+60, y=sfy, anchor="nw")

b4 = tkinter.Button(main, text="4", command=lambda:spielzug(4), height = 1, width = 2)
b4.place(x=sfx, y=sfy+30, anchor="nw")
b5 = tkinter.Button(main, text="5", command=lambda:spielzug(5), height = 1, width = 2)
b5.place(x=sfx+30, y=sfy+30, anchor="nw")
b6 = tkinter.Button(main, text="6", command=lambda:spielzug(6), height = 1, width = 2)
b6.place(x=sfx+60, y=sfy+30, anchor="nw")

b7 = tkinter.Button(main, text="7", command=lambda:spielzug(7), height = 1, width = 2)
b7.place(x=sfx, y=sfy+60, anchor="nw")
b8 = tkinter.Button(main, text="8", command=lambda:spielzug(8), height = 1, width = 2)
b8.place(x=sfx+30, y=sfy+60, anchor="nw")
b9 = tkinter.Button(main, text="9", command=lambda:spielzug(9), height = 1, width = 2)
b9.place(x=sfx+60, y=sfy+60, anchor="nw")

# Zeile: Spiel entschieden
# unterhalb des Spielfelds
ext = tkinter.Label(main, text="Spieler gewonnen?")
ext.place(x=sfx, y=sfy+100, anchor="nw")

main.mainloop()
```

Die Schaltfläche  
mit den Ziffern  
1 bis 9.  
Spielzug wird  
der Funktion

Zeile mit der Anzeige, ob Spiel  
gewonnen oder unentschieden.

### Hinweis:

Der Python-Programmcode stehen für den Spielablauf (aus Afg 6) und für das GUI **online** bereit.