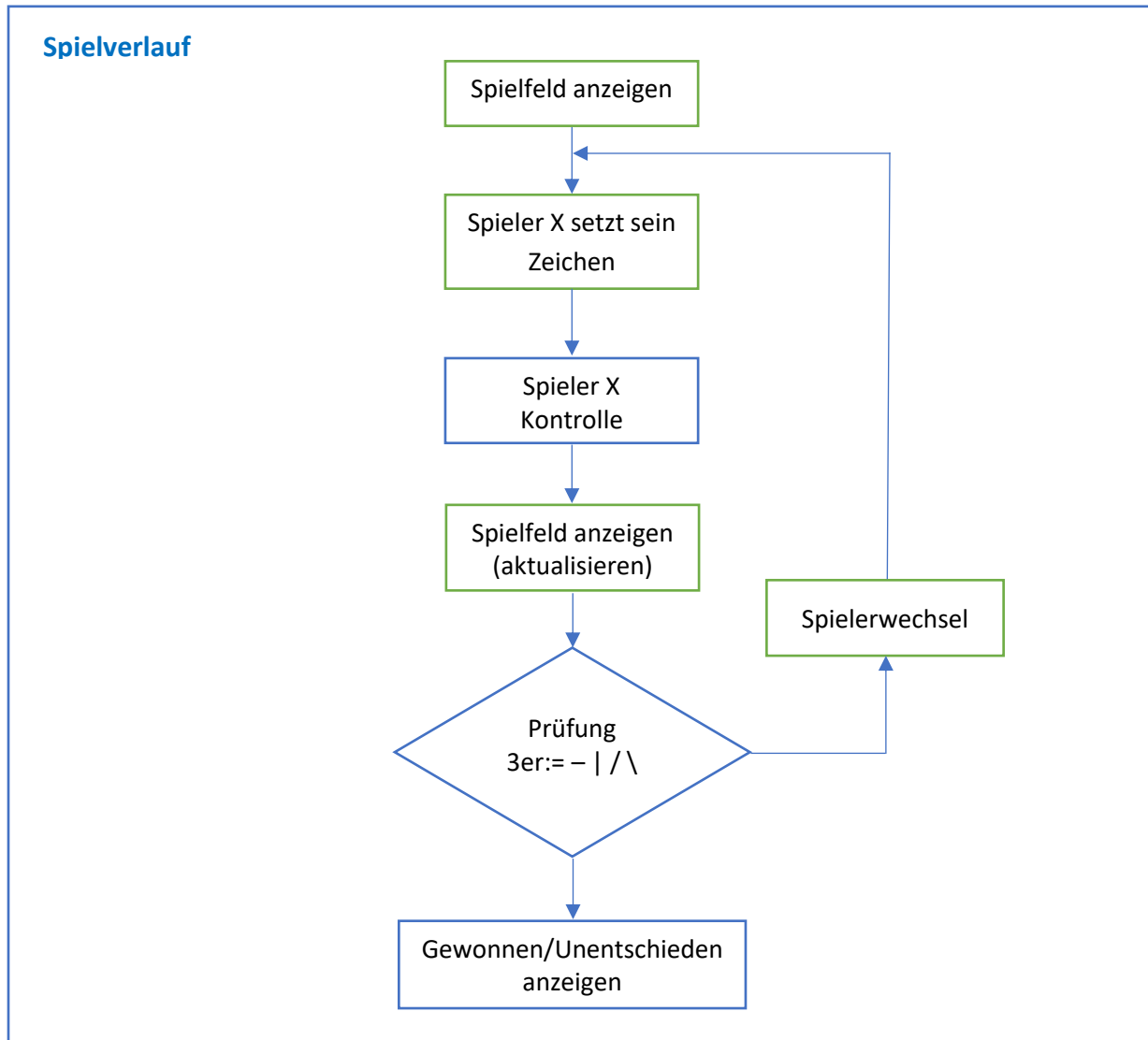
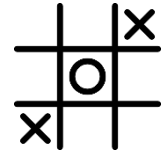


Aufgabe 5 - (Teil 2 von TIC TAC TOE)

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 Videospiele 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()`

In der heutigen Aufgabe geht es, um die Prüfung der Spielereingabe. Das Spielfeld zu TIC TAC TOE haben wir mit dem Datentyp „Liste“ erstellt.

b.) Spieler Eingabekontrolle

Der aktive Spieler gibt seinen Spielzug ein. Nach der Eingabe (1 bis 9) müssen wir prüfen, ob ein Zeichen nicht schon bereits an der Stelle in der Liste mit „X“ oder „O“ belegt ist.
Bsp.

1 2 3	<u>Spieler „O“ ist am Zug:</u>	
4 5 6	Wählt der Spieler „O“	Feld 8 → Meldung,
7 X 9		ansonsten darf er den Zug setzen

Dies erfolgt mit einer if-else Bedingung. Falls an der Stelle im Spielfeld ein „X“ oder „O“ vorhanden ist, soll die Ausgabe „Das Feld ist bereits belegt – wählen Sie ein anderes Feld!“, andernfalls wird der Spielzug akzeptiert.

```
print("TIC TAC TOE")
spiel_aktiv = True
spieler_aktuell = "X"
spielfeld = ["X",
             "1", "2", "3",
             "4", "5", "6",
             "7", "8", "9"]

def spielfeld Ausgabe():
    print(spielfeld[1]+" | "+spielfeld[2]+" | "+spielfeld[3])
    print(spielfeld[4]+" | "+spielfeld[5]+" | "+spielfeld[6])
    print(spielfeld[7]+" | "+spielfeld[8]+" | "+spielfeld[9])

def spieler_eingabe():
    global spiel_aktiv
    spielzug=input("Bitte Feld eingeben: ")
    spielzug=int(spielzug)
    #Prüfung, ob Spielfeld frei
    if spielzug >= 1 and spielzug <= 9:
        return spielzug
    else:
        print("Eingabe muss zwischen 1 und 9 liegen")

def spieler_wechsel():
    global spieler_aktuell
    if spieler_aktuell == "X":
        spieler_aktuell = "O"
    else:
        spieler_aktuell = "X"

#Hauptprogramm
spielfeld_Ausgabe()
while spiel_aktiv:
    print("Spieler", spieler_aktuell, "ist am Zug")
    spielzug = spieler_eingabe()
    if spielzug:
        spielfeld[spielzug] = spieler_aktuell
        spielfeld_Ausgabe()
        # Eingabe prüfen
        # Hat Spieler_aktuell gewonnen
        # Spiel unentschieden
        #Spieler Wechsel
        spieler_wechsel()
```

b.) Eingabekontrolle
If spielfeld[spielzug] gleich „X“ oder „O“, dann „Feld belegt“ sonst return spielzug

c.) Gewonnen?
siehe nächste Seite!

c.) Spieler hat gewonnen

Der Spieler hat gewonnen, wenn alle 3 Felder gleich sind – horizontal, vertikal oder diagonal

Horizontale-Möglichkeiten	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
---------------------------	--	--	--

Vertikale-Möglichkeiten	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
-------------------------	--	--	--

Diagonale-Möglichkeiten	1 2 3 4 5 6 7 8 9	1 2 3 4 5 6 7 8 9
-------------------------	--	--

Ob der Spieler gewonnen hat, schreiben wir eine Funktion **spiel_gewonnen()**. Die Funktion soll alle 8 Möglichkeiten (drei Horizontale-, drei Vertikale- und zwei Diagonale-Möglichkeiten, wie oben veranschaulicht) testen.

Die Funktion **spiel_gewonnen()** soll aus dem Hauptprogramm (siehe vorherige Seite) aufgerufen werden. Überlegen Sie sich einen Rückgabewert für die Funktion. Der Gewinner soll angezeigt werden, bevor das TIC TAC TOE-Spiel beendet wird.