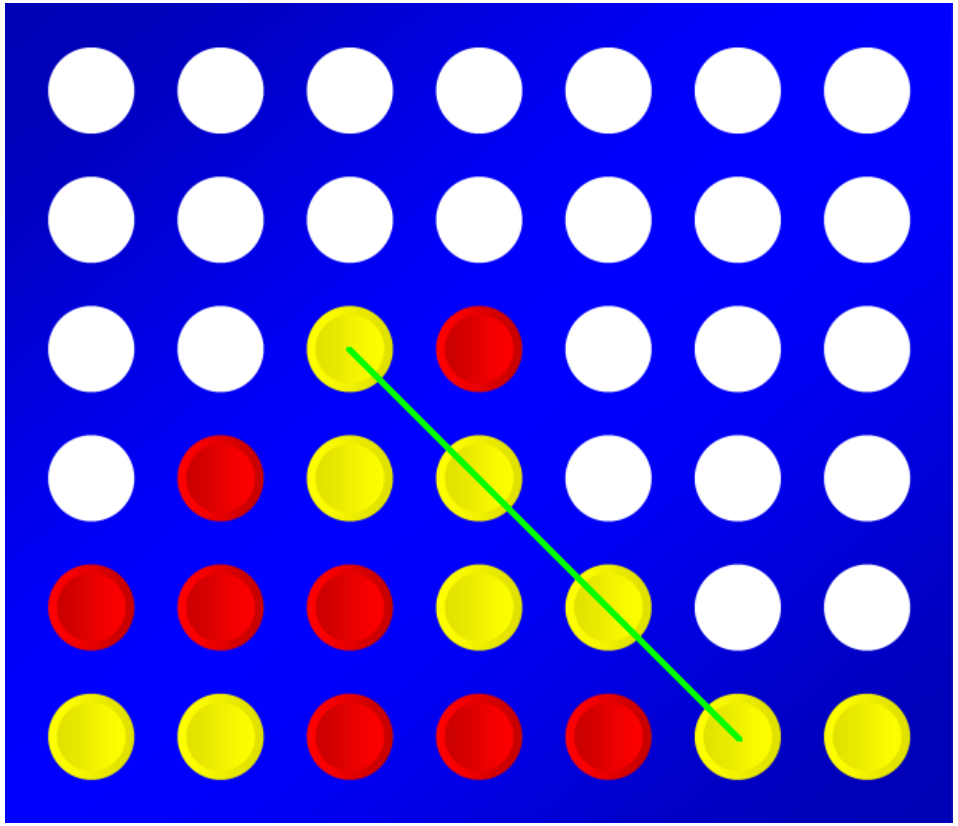
Vier Gewinnt (1)

- Im Folgenden soll als Anwendungsbeispiel das Spiel Vier Gewinnt implementiert werden.
- Die beiden Spieler setzen abwechselnd einen Spielstein (in der Abbildung rot bzw. gelb) auf das unterste freie Feld einer Spalte. (In Hardware-Ausführungen des Spiels wird der Stein von oben in eine Spalte fallen-gelassen.) Wer zuerst eine waagrechte, senkrechte oder diagonale Reihe aus vier eigenen Steinen erreicht, hat gewonnen.
- Das Spielfeld besteht aus 6 Zeilen und 7 Spalten.



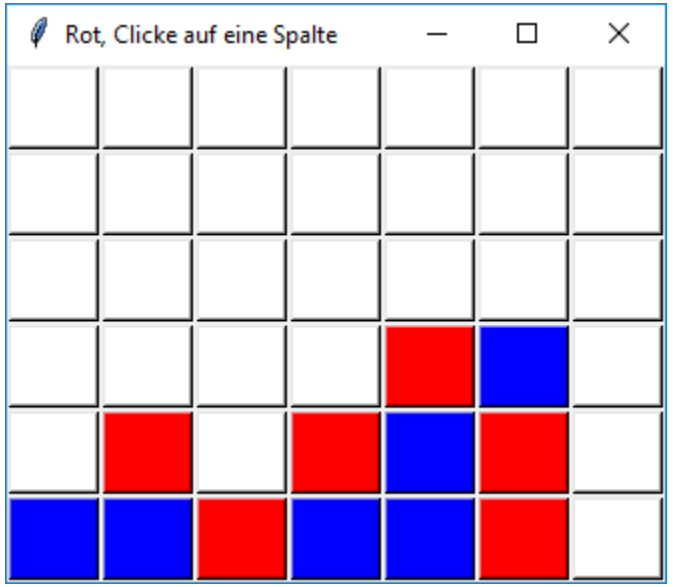
Bildquelle: https://commons.wikimedia.org/wiki/File:Connect4_Wins.PNG
Ghabes at Hungarian Wikipedia [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)]

Vier Gewinnt (2)

- Das Python-Skript four.py soll sowohl eine Konsolenausgabe als auch eine einfache grafische Benutzungsoberfläche haben.

```
_ _ _ _ _ _ _
_ _ _ _ _ _ _
_ _ _ _ _ _ _
_ _ _ _ A B _
_ _ _ _ _ A _
_ A _ A B A _
B B A B B A _

Setze A:
```



Vier Gewinnt (3)

- Welche Daten werden für das Spiel benötigt?

Name	Datentyp	Initialwert

- Legen Sie die Daten als globale Variablen an.

Vier Gewinnt (4)

- Wir brauchen die folgenden Funktionen:

def ausgabe():

Parameter: keiner

Rückgabewert: keiner

Die Funktion gibt das Spielfeld auf die Konsole aus.

def setze(spalte):

Parameter: int spalte : Spaltenindex

Rückgabewert: keiner

Die Funktion prüft, ob spalte ein zulässiger Index zwischen 0 und 6 ist und ob die Spalte noch nicht vollständig gefüllt ist. Falls das so ist, wird in das unterste nicht belegte Feld der Spalte der String 'A, oder 'B , (je nachdem welcher Spieler dran ist) eingetragen. Danach ist der nächste Spieler dran.

Vier Gewinnt (5)

def finde_reihe():

Parameter: keiner

Rückgabewert: None oder eine Liste mit 4 Tupeln, die aus je zwei ganzen Zahlen bestehen

Die Funktion prüft, ob es eine waagrechte, senkrechte oder diagonale Reihe aus vier Steinen eines Spielers gibt. Falls ja, wird eine Liste mit Tupel der Indizes zurückgegeben.
Beispiel: In der folgenden Abbildung gibt es eine diagonale Reihe von Steinen des Spielers A. Der Rückgabewert von finde_reihe ist in diesem Fall: [(5, 2), (4 ,3), (3 ,4), (2, 5)]

```
_ _ _ _ _ _ _
_ _ _ _ _ _ _
_ _ _ _ _ A _
_ _ _ _ A B _
_ A _ A B A _
B B A B B A _
```

Vier Gewinnt (6)

- (Unvollständiges) Programm four.py:

```
# Hier die globalen Variablen einfuegen

def ausgabe():
    # Hier den Funktionskoerper einfuegen

def setze(spalte):
    # Hier den Funktionskoerper einfuegen

def finde_reihe():
    # Hier den Funktionskoerper einfuegen

while gewonnen == None:
    s = int(input("Setze " + dran + ": "))
    setze(s)
    gewonnen = finde_reihe()
    ausgabe()
```

Vier Gewinnt (7)

- Wenn Sie wollen, probieren Sie die folgende Ergänzung von four.py:

```
# Hier die globalen Variablen einfuegen

def ausgabe():
    # Hier den Funktionskoerper einfuegen

def setze(spalte):
    # Hier den Funktionskoerper einfuegen

def finde_reihe():
    # Hier den Funktionskoerper einfuegen

def random_spalte():
    return random.randint(0, 6)
```

```
while gewonnen == None:
    if dran == 'A':
        s = int(input("Setze " + dran + ": "))
        setze(s)
    elif dran == 'B':
        input("Druecke Return, um " + dran + " zufaellig zu setzen")
        setze(random_spalte())

    gewonnen = finde_reihe()
    ausgabe()
```