

# Buzzword Bingo auf der Kommandozeile

Von

**Niraj Rao**

**Yusuf Fuat Sarac**

**Duc Anh Hoang**

# Inhaltsverzeichnis

- Vorführung des Spieles

1 Module

2 Grundfunktionen

3 Hauptteil des Codes (Mainmethode)

- 3.1 Spielverlauf
- 3.2 Einzelspielermodus
- 3.3 Laufschrift

# 1 Module

1

## **Curses**

Bildschirmmal- und Tastaturhandhabungsfunktion für textbasierte Terminals

2

## **PrettyTable**

PrettyTable veranschaulicht Strings in einer tabellarischen Form mit ASCII-Zeichen

3

## **OS**

eine tragbare Möglichkeit zur Verwendung betriebssystem-abhängiger Funktionen

4

## **sys**

Über den Python Interpreter stellt uns das sys-Modul Informationen in Konstanten, Funktionen und Methoden zu Verfügung.

5

## **tkinter**

Tkinter ist ein Modul für eine grafische Oberfläche (GUI) Python. Sie ist in Mac OS und Windows im Lieferumfang mit dabei.

## 2 Grundfunktionen

1

### **def Karte\_erstellen()**

- Erstellt die Bingokarte, basierend auf der Höhe und Breite
- „X“ in der Mitte bei 5x5 und 7x7

2

### **def Karte\_Ausgeben()**

Gibt die Karten in dem Prettytable Format aus

3

### **def gezogenes\_wort()**

- Überprüft das fallende Wort mit den Wörtern im Spielfeld
- Markiert entsprechendes Wort mit „X“

4

### **def prüfen()**

Prüft den horizontalen, vertikalen und diagonalen Gewinn

5

### **def restart() & beenden()**

- Startet das Spiel neu bei Fehlinputs des Spielers
- Beendet das Spiel, wenn „exit“ eingetippt wird

```

1  #erforderliche Module
2  import random
3  import sys
4  import os
5  from colorama.ansi import Back
6  from prettytable import PrettyTable
7  import tkinter as tk
8  from colorama import init, Fore, Style
9  init(autoreset=True)
10

```

```

8  def Karte_Erstellen(höhe,breite,liste):
9      Karte = {
10         }
11         for x in range(0,höhe):
12             Karte[str(x+1)]=[random.sample(liste,k=breite)]
13         for Zahl in Karte:
14             if Zahl=='3':
15                 if höhe==5 and breite==5:
16                     Karte[Zahl][0][2] = "X"
17                 elif Zahl=='4':
18                     if höhe==7 and breite==7:
19                         Karte[Zahl][0][3] = "X"
20             return Karte
21
22
23 def Karte_Ausgeben(Karte):
24     if höhe == 3 & breite == 3:
25         t = PrettyTable(['1', '2', '3'])
26     elif höhe == 5 & breite == 5:
27         t = PrettyTable(['1', '2', '3', '4', '5'])
28     elif höhe == 7 & breite == 7:
29         t = PrettyTable(['1', '2', '3', '4', '5', '6', '7'])
30
31     for Zahl in Karte:
32         t.add_row((Karte[Zahl][0]))
33     print(t)
34
35
36 def gezogenes_wort(Karte, liste):
37     wort_eingabe = input("Bitte geben sie das Wort ein was sie Markiert haben wollen" + "\n")
38     for Zahl in Karte:
39         y = 0
40         if wort_eingabe == "exit":
41             quit()
42         for wort in Karte[Zahl][0]:
43             if wort.casefold() != wort_eingabe.casefold():
44                 print("Geben sie bitte ein Wort aus der Liste ein")
45             elif wort.casefold() == wort_eingabe.casefold():
46                 Karte[Zahl][0][y] = "X"
47
48         y += 1
49
50     return wort_eingabe

```

```

63 def Prüfen(Karte,höhe,breite):
64
65     sieg = False
66     #horizontaler Gewinn
67
68     for Zahl in Karte:
69         counter = 0
70         for x in range(0,int(höhe)):
71             if Karte[Zahl][0][x]=='\033[40m'+'\033[31m' + "      X      " + '\033[39m' + '\033[49m':
72                 counter+= 1
73
74             if counter is int(höhe):
75                 sieg = True
76                 return sieg
77
78     #vertikaler Gewinn
79     for x in range(0,int(höhe)):
80         counter=0
81
82         for Zahl in Karte:
83             if Karte[Zahl][0][x]=='\033[40m'+'\033[31m' + "      X      " + '\033[39m' + '\033[49m':
84                 counter+=1
85         if counter==int(höhe):
86             sieg=True
87             return sieg
88
89     #diagonaler Gewinn links oben nach rechts unten
90     for x in range (0,int(höhe)):
91         y=0
92         counter=0
93         for Zahl in Karte:
94             if Karte[Zahl][0][y]=='\033[40m'+'\033[31m' + "      X      " + '\033[39m' + '\033[49m':
95                 counter+=1
96                 y+=1
97         if counter==int(höhe):
98             sieg=True
99             return sieg
100
101     #diagonaler Gewinn von rechts oben nach links unten
102     for x in range (int(höhe)):
103         y=int(höhe)-1
104         counter=0
105         for Zahl in Karte:
106             if Karte[Zahl][0][y]=='\033[40m'+'\033[31m' + "      X      " + '\033[39m' + '\033[49m':
107                 counter+=1
108                 y-=1
109         if counter==int(höhe):
110             sieg=True
111             return sieg
112

```

```

101
102 def restart():
103     print("Spiel startet neu aufgrund von eines FehlInputs" + "\n" + "-----" + "\n")
104     os.execv(sys.executable, ['/bin/python3'] + sys.argv)
105 def restarthöhebreite():
106     print("Spiel startet neu aufgrund von eines FehlInputs")
107     print("Bitte geben sie nur 3 x 3, 5 x 5, 7 x 7 ein" + "\n" + "-----" + "\n")
108     os.execv(sys.executable, ['/bin/python3'] + sys.argv)
109 def beenden():
110     user_input = input()
111     if user_input == "exit":
112         quit()
113

```

### 3 Hauptteil des Codes (Mainmethode)

- Willkommensbildschirm
- Textdatei einlesen & Wörter trennen
- Bestimmung Größe Spielfeld & Spielmodi durch Userinput
- Fehlerprävention durch try – except Block

[illegible]

## 3.1 Spielverlauf

- Ausführung bei Einzelspielermodus
- Ausführung bei Mehrspielermodus – Angabe von Personenanzahl durch Userinput
- Exportfunktion bei Mehrspielermodus  
Karten werden im Prettytable-Format exportiert

```
161
162 spielfeld=[]
163
164
165 if modi == 1:
166     einzelspieler = Karte_Erstellen(höhe,breite,liste)
167     spielfeld.append(einzelspieler)
168     anzahl_spieler = 1
169     for x in range(0,anzahl_spieler):
170         Karte = Karte_Ausgeben(spielfeld[x])
171 elif modi == 2:
172     anzahl_spieler=input('Bitte geben Sie an, wie viele Personen am Bingospiel teilnehmen möchten!'+ "\n")
173     for x in range(0,int(anzahl_spieler)):
174         spielfeld.append(Karte_Erstellen(höhe,breite,liste))
175
176     if höhe == 3 & breite == 3:
177         t = PrettyTable(['1', '2', '3'])
178     elif höhe == 5 & breite == 5:
179         t = PrettyTable(['1', '2', '3', '4', '5'])
180     elif höhe == 7 & breite == 7:
181         t = PrettyTable(['1', '2', '3', '4', '5', '6', '7'])
182
183
184
185
186     for x in range(0,int(anzahl_spieler)):
187         f = open("Karte"+str(x+1), 'w')#karten erzeugt
188         Karte=spielfeld[x]
189
190         for Zahl in Karte:
191             t.add_row(Karte[Zahl][0])
192         f.write(str(t))
193
194
195         for Zahl in Karte:
196             for Zahl in range(0,1):
197                 t.del_row(Zahl)
198
199
200
201     f.close
202
```



## 3.2 Einzelspielermodus

- Spielstart im Einzelspielermodus
- Dokumentation des Spielverlaufs  
zählt eingetippte Wörter mit
- Gewinnprüfung durch Prüfen() - Methode

```
232 print('''
233 Das Spiel beginnt jetzt!
234 ''')#prints ändern
235
236 #Einzelspielermodus Gewinnbekanntgabe mit Detail(wie viele Wörter gezogen wurden)
237 sieg = False
238
239 for x in range(0,int(anzahl_spieler)):
240
241     sieg=Prüfen(spielfeld[x],int(höhe),int(breite))
242     words_till_win = 0
243     wort_eingabe = input("Bitte geben sie das Wort ein was sie Markiert haben wollen" + "\n")
244     while not sieg and wort_eingabe != "exit":
245         wort_eingabe = gezogenes_wort(spielfeld[x], liste, wort_eingabe)
246         words_till_win += 1
247         #Solange das Spiel noch nicht gewonnen ist, wird jedes eingegebene Wort mitgezählt
248
249         print(f"\nGeschriebenes Wort: {wort_eingabe}.")
250         print(f"Anzahl der eingegeben Wörtern: {words_till_win}.\n")
251         Karte_Ausgeben(spielfeld[x])
252
253     #Gewinn wird überprüft, wenn True --> Laufschrift
254     sieg=Prüfen(spielfeld[x],int(höhe),int(breite))
255
```

```

#Laufschrift bei Sieg
if sieg==True:

    print("\033[1;35;40m\n"+"\033[35m'+f'\nSie haben Gewonnen!\nSoviele Wörter wurden gezogen: {words_till_win}."+
    '\033[39m' + "\033[1;35;40m\n"]
    APP_TITEL = "LAUFSCHRIFT - GEWINN"
    xposition = 300
    yposition = 300
    weite_fenster = 300
    höhe_fenster = 300
    SCROLLEN_TEXT = "GLÜCKWUNSCH! Sie haben GEWONNEN!"

class Application(tk.Frame):

    def __init__(self, master):
        self.master = master
        tk.Frame.__init__(self, master)

        self.canvas = tk.Canvas(self, bg='cyan', highlightthickness=0)
        self.canvas.pack(expand=True)

        xpos = weite_fenster
        ypos = 100
        self.canvas.create_text(xpos, ypos, anchor='w', text=SCROLLEN_TEXT,
                                font=('Helvetica 30 bold'), tags='text')

        text_anfang = self.canvas.bbox('text')[0]
        text_ende = self.canvas.bbox('text')[2]
        self.text_länge = text_ende - text_anfang

        self.scroll_text()

    def scroll_text(self):
        self.canvas.move('text', -2, 0)
        text_ende = self.canvas.bbox('text')[2]
        if text_ende < 0:
            self.canvas.move('text', weite_fenster + self.text_länge, 0)

        self.canvas.after(20, self.scroll_text)

def anwenden():
    app_win = tk.Tk()
    app_win.title(APP_TITEL)
    app_win.geometry("+{}+{}".format(xposition, yposition))
    app_win.geometry("{}x{}".format(weite_fenster, höhe_fenster))

    app = Application(app_win).pack(fill='both', expand=True, padx=0, pady=0)

    app_win.mainloop()
    quit()

if __name__ == '__main__':
    anwenden()

```

## 3.3 Laufschrift

- Laufschrift bei Gewinn Einzelspielermodus
- Aufpopendes Fenster durch Tkinter
- Funktionen bestimmen Design des Fensters  
Größe, Schriftart, Schriftgröße,  
Schriftfarbe, Vorder- und  
Hintergrundfarbe, Scrollrichtung,  
Scrollgeschwindigkeit

**Danke für Ihre Aufmerksamkeit!**