

WERKSTÜCK A
ALTERNATIVE 3

SCHIFFE VERSENKEN

VON: ZAKARIA MOWAFEK, MERVE ILPESIN & SAMMY ACOSTA

ÜBERSICHT

- Warum genau Schiffe Versenken?
- Inspiration
- Menü
- Aufbau des Menüs
- Player vs. Computer
- Player vs. Player
- Herausforderungen
- Fazit



WARUM GENAU SCHIFFE VERSENKEN?

❖ ALLE GRUPPENMITGLIEDER SIND GAMING AFFIN

❖ PROJEKT HAT UNS ZUGESAGT

❖ IDEEN SIND SCHNELL ENTSTANDEN



Inspiration



Spiel Logo



Inspiration zum Logo



Python-cfonts

MENÜ

- ❖ FÜR DIE EINE LEICHTERE BENUTZERFREUNDLICHKEIT
- ❖ UM ZWISCHEN VERSCHIEDENEN OPTIONEN ZU WECHSELN
- ❖ EIN MENÜ GEHÖRT ZU EINEM SPIEL DAZU



AUFBAU DES MENÜS

❖ WHILE – SCHLEIFEN, IF – ANWEISUNGEN, TRY – EXCEPT BLÖCKE, INPUT – ANWEISUNGEN

❖ MENÜ SOLLTE FARBIG SEIN

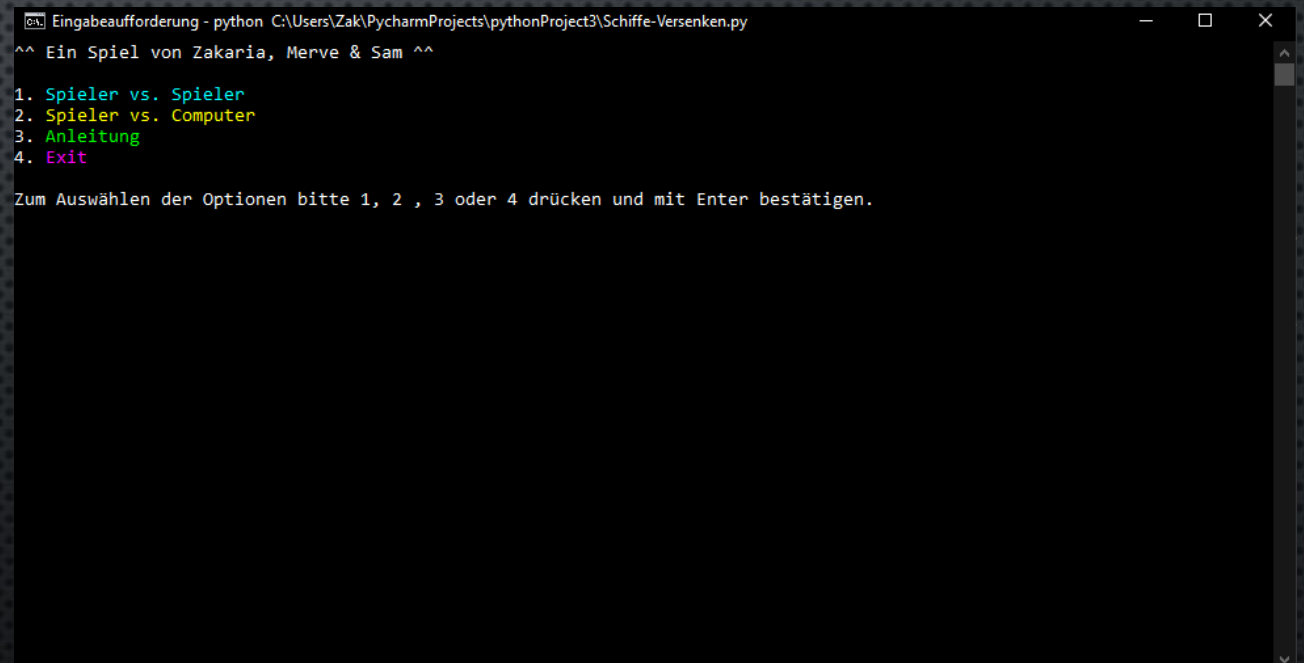
- CFONTS

- COLORAMA

❖ UNENDLICHE SCHLEIFEN MIT EXIT

- OK = 1

WHILE OK == 1:



```
Eingabeaufforderung - python C:\Users\Zak\PycharmProjects\pythonProject3\Schiffe-Versenken.py
^^ Ein Spiel von Zakaria, Merve & Sam ^^
1. Spieler vs. Spieler
2. Spieler vs. Computer
3. Anleitung
4. Exit

Zum Auswählen der Optionen bitte 1, 2 , 3 oder 4 drücken und mit Enter bestätigen.
```


MENÜ 1/3

- OBERFLÄCHE WIRD ABGEBILDET
- ES WIRD NACH EINEM INPUT GEFRAGT
- KLAISSCHE IF-ANWEISUNG

```
26 def mainMenu() :
27     ok = 1
28     while ok == 1:
29         try:
30             pvp = (f"{Fore.CYAN}Spieler vs. Spieler")
31             pve = (f"{Fore.YELLOW}Spieler vs. Computer")
32             info = (f"{Fore.GREEN}Anleitung")
33             ext = (f"{Fore.MAGENTA}Exit")
34
35             print("^^ Ein Spiel von Zakaria, Merve & Sam ^^")
36             print("\n1. " + pvp)
37             print("2. " + pve)
38             print("3. " + info)
39             print("4. " + ext + "\n")
40
41             eingabe = input("Zum Auswählen der Optionen bitte 1, 2 , 3 oder 4 drücken und mit Enter bestätigen.")
42             int(eingabe)
43             liste = [1,2,3, 4]
44             if "1" in eingabe:
45                 print("Eingabe erfolgreich - es wurde " + pvp + " ausgewählt.")
46                 ok == 2
47                 break
48             elif "2" in eingabe:
49                 print("Eingabe erfolgreich - es wurde " + pve + " ausgewählt.")
50                 ok == 2
51                 break
52             elif "3" in eingabe:
53                 print("\nEingabe erfolgreich - es wurde " + info + " ausgewählt.")
54                 ok == 2
55                 break
56
57             elif "4" in eingabe:
58                 print("\nEingabe erfolgreich - es wurde " + ext + " ausgewählt.")
59                 ok == 2
60                 break
61
62             else:
63                 print("\nFehler - Bitte 1, 2, 3 oder 4 auswählen.")
64
65             except:
66                 print("\nFehler - Bitte 1, 2, 3 oder 4 auswählen.")
67
68         return eingabe
69
```

MENÜ 2/3

- WEITERE INPUT ANWEISUNGEN

- SICHERHEITSFRAGEN

- TRY-CATCH VARIANTE

```
36 def finale(check):
37     try:
38         if "1" == check:
39             #pvp
40             bestaetigung = input("Ausführen? [j/n]:")
41             if bestaetigung == "j":
42                 #hier kommt pvp rein
43                 print("Spiel würde jetzt ausgeführt werden.")
44                 bs = "n"
45
46             elif bestaetigung == "n":
47                 bs = "n"
48
49             elif bestaetigung != "j" or bestaetigung != "n":
50                 print("Bitte nur j für ja und n für nein bestätigen.\n")
51                 bs = "n"
52
53         elif "2" == check:
54             #pve
55             bestaetigung = input("Ausführen? [j/n]:")
56             if bestaetigung == "j":
57                 # hier kommt pve rein
58                 print("Spiel würde jetzt ausgeführt werden.")
59                 bs = "n"
60
61             elif bestaetigung == "n":
62                 bs = "n"
63
64             elif bestaetigung != "j" or bestaetigung != "n":
65                 print("Bitte nur j für ja und n für nein bestätigen.\n")
66                 bs = "n"
67
68         elif "3" == check:
69             #ext
70             bestaetigung = input("Möchten Sie das Spiel wirklich beenden? [j/n]:")
71             if bestaetigung == "j":
72                 print("Vielen Dank fürs Spielen.")
73                 bs = "j"
74
75             elif bestaetigung == "n":
76                 bs = "n"
77
78             elif bestaetigung != "j" or bestaetigung != "n":
79                 print("Bitte nur j für ja und n für nein bestätigen.\n")
80                 bs = "n"
81         else:
82             print("Bitte nur j für ja und n für nein bestätigen.\n")
83     except:
84         print("")
85         print("Fehler - Eingabe überprüfen.")
86
87     return bs
```


MENÜ 3/3

- UNENDLICHE SCHLEIFE
- BEI EXIT WIRD DIE SCHLEIFE DURCHBROCHEN

```
1167     check = mainMenu()
1168     bs = finale(check)
1169     def aufseher(bs):
1170         af = "n"
1171         while af == "n":
1172             check = mainMenu()
1173             bs = finale(check)
1174             if bs == "j":
1175                 af = "y"
1176                 break
1177             elif bs == "n":
1178                 print("Wiederholung der Schleife.")
1179
1180     if bs != "j":
1181         aufseher(bs)
1182     else:
1183         print("Vielen Dank fürs Spielen.")
1184
1185
```

PLAYER VS. COMPUTER

AUFBAU:

SCHLEIFEN, IF - ANWEISUNGEN, TRY - EXCEPT BLÖCKE, METHODEN, LISTS, INPUT - ANWEISUNGEN, RANDOM

ZUSÄTZLICH:

- TIME-SLEEP(3)
- CLEAR=LAMBDA:OS.SYSTEM('CLS')
- CLEAR
- CFONTS FÜR DEN ENDScreen

PLAYER VS. COMPUTER

MIT COMPUTER IST EIN ALGORITHMUS GEMEINT, DER TAKTISCH AUF SCHIFFE SCHIEßT

1. SPIELER STELLT SCHIFFE AUF DAS FELD
2. SCHIFFE WERDEN NOCH EINMAL ANGEZEIGT
3. SPIELER SCHIEßT AUF SEINE GEWÜNSCHTEN KOORDINATEN MIT HILFE VON INPUT – ANWEISUNGEN
4. DER ALGORITHMUS SCHIEßT AUF DAS FELD DES SPIELERS
5. BEIDE SCHIEßEN ABWECHSELND AUF EINANDER BIS EINER KEINE SCHIFFE MEHR ÜBRIG HAT
6. DER JENIGE, DER ALLE SCHIFFE DES GEGNERS VERSENKT GEWINNT

PLAYER VS. COMPUTER: BESONDERHEITEN

```
770 def pruefung_boat_comp(b, start, richt, bereits_ausgewaehlt):
771     boat = []
772     if richt == 1:
773         for i in range(b):
774             boat.append(start - i * 10)
775     elif richt == 2:
776         for i in range(b):
777             boat.append(start + i)
778     elif richt == 3:
779         for i in range(b):
780             boat.append(start + i * 10)
781     elif richt == 4:
782         for i in range(b):
783             boat.append(start - i)
784     boat = pruefung_auf_fehler(boat, bereits_ausgewaehlt)
785     return boat
786
787 def boate_bauen(bereits_ausgewaehlt, anzahl_schiffe):
788     schiffe = []
789     # anzahl_schiffe = [5,4,3,3,2,2]
790     for b in anzahl_schiffe:
791         boat = [-1]
792         while boat[0] == -1:
793             auswahl_schiff_start = random2.randrange(99)
794             auswahl_schiff_richt = random2.randrange(1, 4)
795             boat = pruefung_boat_comp(b, auswahl_schiff_start, auswahl_schiff_richt,
796                                     bereits_ausgewaehlt)
797             schiffe.append(boat)
798             bereits_ausgewaehlt = bereits_ausgewaehlt + boat
799             # print(schiffe)
800
801     return schiffe, bereits_ausgewaehlt
```

```
910 def taktik_strategie(schuss, taktik, zuvorGeschosseneSchuesse, hit):
911     temp = []
912     if len(taktik) < 1:
913         temp = [schuss - 1, schuss + 1, schuss - 10, schuss + 10]
914     else:
915         if schuss - 1 in hit:
916             temp = [schuss + 1]
917             for num in [2, 3, 4, 5, 6, 7, 8]:
918                 if schuss - num not in hit:
919                     temp.append(schuss - num)
920                     break
921         elif schuss + 1 in hit:
922             temp = [schuss - 1]
923             for num in [2, 3, 4, 5, 6, 7, 8]:
924                 if schuss + num not in hit:
925                     temp.append(schuss + num)
926                     break
927         if schuss - 10 in hit:
928             temp = [schuss + 10]
929             for num in [20, 30, 40, 50, 60, 70, 80]:
930                 if schuss - num not in hit:
931                     temp.append(schuss - num)
932                     break
933         elif schuss + 10 in hit:
934             temp = [schuss - 10]
935             for num in [20, 30, 40, 50, 60, 70, 80]:
936                 if schuss + num not in hit:
937                     temp.append(schuss + num)
938                     break
939     # erweiterte Strategie
940     erws = []
941     for i in range(len(temp)):
942         if temp[i] not in zuvorGeschosseneSchuesse and temp[i] < 100 and temp[i] > -1:
943             erws.append(temp[i])
944     random2.shuffle(erws)
945
946     return erws
```


PLAYER VS. PLAYER

AUFBAU:

SCHLEIFEN, IF - ANWEISUNGEN, TRY - EXCEPT BLÖCKE, METHODEN, LISTS, INPUT - ANWEISUNGEN

ZUSÄTZLICH:

- TIME-SLEEP(3)
 - CLEAR=LAMBDA:OS.SYSTEM('CLS') HAUPTSÄCHLICH, DAMIT MAN NICHT SO EINFACH SCHUMMELN KANN
- CLEAR
- CFONTS FÜR DEN ENDSCREEN

DER PLAYER VS. PLAYER TEIL BASIERT AUF DEM PLAYER VS. COMPUTER TEIL UND WURDE ANGEPASST

PLAYER VS PLAYER

```
109 def schiff_Spieler1_hinzufuegen(long, bereits_ausgewaehlt_Spieler1):
110
111     ok = True
112     while ok:
113         try:
114             print("      Schiffe Versenken      ")
115             print("      Spieler 1      ")
116             print("      Bitte plaziere deine Schiffe")
117             print("      A B C D E F G H I J")
118
119             place1 = 0
120             for x in range(10):
121                 row1 = ""
122                 for y in range(10):
123                     cz = " _ "
124                     row1 = row1 + cz
125                     place1 = place1 + 1
126
127                 print(x, " ", row1)
128             schiff_Spieler1 = []
129             # Fordert den Spieler auf eine Koordinate einzugeben.
130             print("Standpunkt von Schiff mit Länge ", long)
131             for i in range(long):
132                 boat_num = input("Bitte Koordinate eingeben, wo sich dein Schiff befinden soll:")
133                 if len(boat_num) != 2:
134                     print("\nFehler - Koordinate befindet sich nicht auf dem Feld.")
135                     print("...versuche stattdessen Bsp: A1, C8, G3\n")
136                 else:
137                     eins = boat_num[0:1]
138                     # print(eins)
139                     zwei = boat_num[-1:]
140                     # print(zwei)
```

```
141
142     if eins == "a" or eins == "A":
143         abc = 0
144     elif eins == "b" or eins == "B":
145         abc = 1
146     elif eins == "c" or eins == "C":
147         abc = 2
148     elif eins == "d" or eins == "D":
149         abc = 3
150     elif eins == "e" or eins == "E":
151         abc = 4
152     elif eins == "f" or eins == "F":
153         abc = 5
154     elif eins == "g" or eins == "G":
155         abc = 6
156     elif eins == "h" or eins == "H":
157         abc = 7
158     elif eins == "i" or eins == "I":
159         abc = 8
160     elif eins == "j" or eins == "J":
161         abc = 9
162     else:
163         print("Falsche Koordinate! Bitte Koordinate innerhalb des Feldes wählen")
164
165     xyz = int(zwei)
166     xyz = (xyz * 10)
167     boat_num = abc + xyz
168
169     schiff_Spieler1.append(int(boat_num))
170     # Prüft auf das Schiff
171     schiff_Spieler1 = pruefung_auf_fehler_Spieler1(schiff_Spieler1,
172                                                    bereits_ausgewaehlt_Spieler1)
173
174     if schiff_Spieler1[0] != -1:
175         bereits_ausgewaehlt_Spieler1 = bereits_ausgewaehlt_Spieler1 + schiff_Spieler1
176         print("...Schiff wurde plaziert")
177         print()
178         print()
179         break
180     else:
181         print("Fehler - Platziertes Schiff überschneidet sich möglicherweise")
182     except:
183         print("      Bitte erneut eingeben")
184         print()
185     return schiff_Spieler1, bereits_ausgewaehlt_Spieler1
```


HERAUSFORDERUNGEN

SICHERHEITSFRAGEN, UM VOR FEHLEINGABEN ZU SCHÜTZEN

TRY – EXCEPT BLÖCKE, WELCHE PROBLEME/FEHLER BEI DER EINGABE HERAUSFILTERT

FAZIT

OBWOHL UNS EINIGES BEKANNT WAR, HABEN WIR VIELE NEUE KENNTNISSE GESAMMELT

IM ANSCHLUSS ZEIGEN WIR IHNEN UNSER FERTIGES SPIEL

VIELEN DANK FÜR IHRE AUFMERKSAMKEIT

