11 SCHERE-STEIN-PAPIER

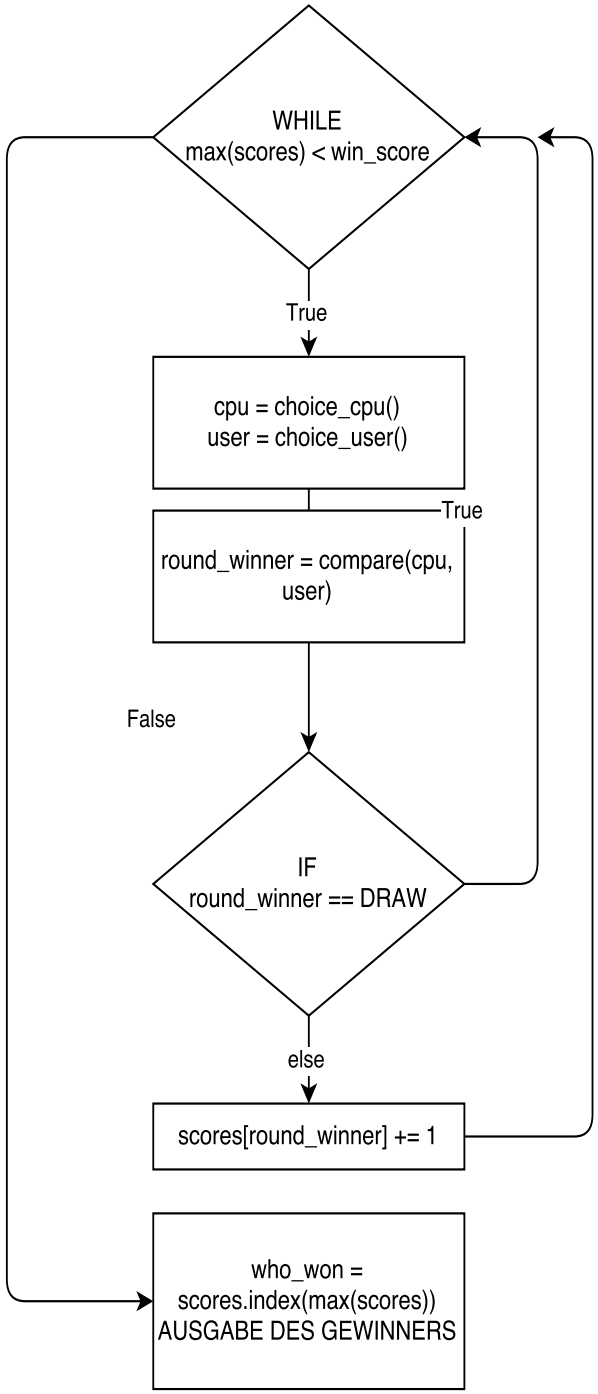
Yanick DICKBAUER, Patrick MOSER, Manuel PERNER

Aufgabenstellung

- Simulieren Sie das Spiel „Schere-Stein-Papier“, wobei der Computer der Gegner ist. Der Benutzer gibt eine der drei Möglichkeiten ein, welche das Programm mit simulierten Werten vergleicht. Die Anzahl der Runden ist frei wählbar und beim Programmende wird der Sieger ausgegeben. Stellen Sie im Rahmen der Präsentation den Ablauf des Programmes anhand von selbstgewählten Zufallszahlen vor.

Flow Chart

MAIN FUNCTIONALITY



Selbstgewählte Zufallszahlen

- win_score = 3 (DEFAULT)
- choices and winner:

cpu =	0	1	1	2	1
user =	1	1	2	1	2
round_winner=	1	2	1	0	1

- Ergebnis:

cpu	user	draw
1	3	1

FUNCTION choice_cpu()

- benutzt random_number_from_interval

```
47 ▼ def choice_cpu():
48       """This function creates and returns the CPU's choice."""
49       cpu = int(random_number_from_interval(0,len(OPTIONS)))
50       if DEBUG:
51           print('Hint: CPU chose {}'.format(OPTIONS[cpu]))
52       return cpu
```

- gibt entweder 0, 1, oder 2 zurück
- Wert gibt den Index des entsprechenden Items der Liste OPTIONS

FUNCTION choice_user()

- fragt den User nach seiner Wahl

```
54 ▼ def choice_user():
55       """This function asks the user for his/her choice and returns it."""
56       valid = False
57 ▼     while not valid:
58           option_string = '/'.join(OPTIONS)
59           user_input = input('Choose an item out of {}: '.format(option_string)).lower()
60           if user_input in OPTIONS:
61               valid = True
62           else:
63               print('Incorrect input, please choose again.')
64       return OPTIONS.index(user_input)
```

- Eingabeüberprüfung über while
  - option_string wird benötigt, um User-Eingabe (String) überprüfen zu können
  - wenn die Eingabe des Users in option_string enthalten ist, wird die Eingabe akzeptiert und zurückgegeben
  - wenn die Eingabe nicht enthalten ist, wird der User erneut nach einer korrekten Eingabe gebeten

FUNCTION compare(cpu, user)

- vergleicht die Werte auf cpu und user, indem die entsprechenden Werte als Indizes der Entscheidungsmatrix betrachtet werden (matrix)

```
67  def compare(cpu, user):
68      """
69          This function compares the random CPU-choice with the user's input
70          and returns who won (CPU, USER, DRAW)
71      """
72      # rows: cpu   cols: user
73          #ROCK   PAPER  SCISSORS
74      matrix = [
75          [DRAW,  USER,  CPU],  #ROCK
76          [CPU,   DRAW, USER],  #PAPER
77          [USER,  CPU,  DRAW]   #SCISSORS
78      ]
79      return matrix[cpu][user]
```

- der an der stelle (cpu, user) gespeicherte Wert entscheidet über das Rundenergebnis

FUNCTION main()

```
24 ▼ def main():
25       win_score = user_input([['Please enter the score to win the game', int, 3]], DEBUG)
         [0]
26
27       scores = [0, 0]
28 ▼     while max(scores) < win_score:
29           cpu = choice_cpu()
30           user = choice_user()
31           round_winner = compare(cpu, user)
32           if round_winner == DRAW:
33               pass
34 ▼         else:
35               scores[round_winner] += 1
36
37           print(ROUND_MSG[round_winner])
38
39           print('Score: cpu {} vs. {} you\n'.format(scores[CPU], scores[USER]))
40
41       who_won = scores.index(max(scores))
42       if who_won == CPU:
43           print('CPU won the game')
44 ▼     else:
45           print('USER won the game')
```