

Lösungsidee

Um die Aufgabe zu lösen, müssen zuerst alle drei Formen des Turniers (Liga, KO, KO5) durchlaufen werden. Dabei müssen Punkte an die jeweiligen Gewinner vergeben werden, somit kann man später bestimmen welcher Spieler mit welcher Punktzahl wie oft in einer Turnierform gewonnen hat.

Für das Ligaturnier müssen zuerst jedem Spieler seine Gegner zugeordnet werden. Da jeder Spieler aber nur ein mal mit dem jeweiligen anderem Spieler spielen muss, sollten Spieler mit denen bereits gespielt wurden, aus der Gegnerliste der einzelnen Spieler entfernt werden. Dann muss jeder Spieler gegen seine entsprechenden Spieler spielen und die Punktzahl bei dem Gewinner erhöht werden.

Wenn alle Spieler gegeneinander gespielt haben, wird der Hauptgewinner ermittelt. Da aber alle Turnierformen mehrmals ausgeführt werden, wird hier noch einmal eine Punkteliste erstellt und jedem Hauptgewinner des jeweiligen Turniers einen Punkt hinzugefügt. Dieser Schritt geschieht für alle Formen der Turniere.

Für das KO-Turnier müssen Spielerpaare eingeteilt werden, welche dann gegeneinander antreten. Das muss so lange geschehen, bis es nur noch einen Spieler gibt, welcher dann gewonnen hat. Auch hier wird dem Hauptgewinner bei der Hauptpunktzahl ein Punkt vergeben.

Das KO5 Turnier muss genau so ablaufen wie das KO-Turnier, nur dass die jeweiligen Spielerpaare fünf mal hintereinander antreten. Hierbei wird ebenfalls bei jedem Gewinn ein Punkt an den Gewinner vergeben. Nach den fünf Spielen wird der Gewinner des Spiels bestimmt und wie bei dem normalem KO-Turnier dieser Prozess so lange wiederholt, bis nur ein Spieler verbleibt. Auch diesem wird ein Punkt hinzugefügt.

Umsetzung

Die Lösungsidee wird in Python umgesetzt. Hierbei wird die random- und sys- Bibliothek genutzt. Die random-Bibliothek wird genutzt um beim antreten von zwei Spielern mit den jeweiligen Spielerstärken den Gewinner zu ermitteln. Die sys-Bibliothek wird genutzt um das Programm mit einem Argument, der Spielerstärken-Datei, zu starten.

Sobald das Programm mit den richtigen Argumenten gestartet wird, wird die Datei mit Spielerstärken zeilenweise in eine Liste eingelesen. Daraus wird die Anzahl der Spieler entnommen, der Zeilenumbruch entfernt und dann aus der Liste gelöscht. Übrig bleiben nur noch die Spielerstärken. Diese werden dann mittels einer for-Schleife aus der Liste entnommen und der formatierungsbedingte Zeilenumbruch wird entfernt. Die einzelnen Spielerstärken werden in einer Liste gespeichert. Dadurch kann jetzt jedem Spieler ein Index zugeordnet werden.

Um den Gewinner aus zwei Spielern und deren entsprechenden Spielerstärken zu bestimmen, gibt es eine eigenständige Funktion, welche mit einem Zufallsgenerator und den Spielerstärken den Gewinner ermittelt und diesen dann zurückgibt. Diese Funktion ist ein Hauptbestandteil des gesamten Skripts:

```

def fight(dPlayer1,dPlayer2):
    dTotal = dPlayer1 + dPlayer2
    dRandom = randint(1,dTotal)
    if(dRandom<=dPlayer1):
        return(1)
    else:
        return(2)

```

Für das Ligaturnier gibt es einen eigenständigen Matchmaker, welcher mit zwei ineinander stehenden for-Schleifen jedem Spieler Gegner zuordnet, ohne dass Spieler mehrmals gegeneinander antreten. Das ist möglich, da nach laden des Spielers (welchem die Gegner zugeteilt werden), aber vor jeder Zuteilung der erste Spieler aus der Spielerliste gelöscht wird. Dadurch verringert sich vor jeder Zuteilung die Liste der Gegner, weshalb es dann nicht mehr möglich ist, dass zwei Spieler mehrmals gegeneinander antreten. Die erste for-Schleife erledigt genau das, während die zweite Schleife innerhalb der ersten, die Gegner zuordnet. Da aber jedes mal ein Spieler entfernt wird, muss eine Verschiebung der Index der Gegner stattfinden. Diese Verschiebung erhöht sich nach jedem verarbeiteten Spieler und somit wird jedem Spieler einmalig Gegner zugeordnet:

```

def LigaMatchMaker(dPlayers):
    dMatches = []
    offset = 1
    for i in range(len(dPlayers)):
        dPlayers.pop(0)
        dOpponents = []
        for iO in range(len(dPlayers)):
            iO += offset
            dOpponents.append(iO)
        dMatches.append(list(dOpponents))
        offset += 1
    return(dMatches)

```

Die zweite Funktion des Liga-Turniers lässt jeden Spieler gegen seine Gegner antreten und erhöht bei dem Gewinner die Punktzahl. Nachdem alle Spieler gegeneinander angetreten sind, wird der Gesamtgewinner des Turniers bestimmt. Hierbei wird der Spieler mit der höchsten Punktzahl entnommen.

Das KO-System besteht ebenfalls aus zwei Funktionen. In der KoMiniMatch-Funktion, treten alle Spieler „einer Schicht“ gegeneinander an. Hier werden Spielerpaare mit einer while-Schleife zugeteilt. Dabei wird jeweils der Index (Nummer des Spielers), zusammen mit dem darauffolgendem Spieler, als ein Paar gespeichert und das einzelne Paar in einer großen Paarliste

gespeichert. Sobald die Zuteilung erfolgt ist, treten mit einer for-Schleife alle Spielerpaare gegeneinander an. In der Schleife wird mit einer if-Bedingung der Gewinner bestimmt und dieser dann an eine Liste mit den anderen Gewinnern angehängt. Somit wird immer nur „eine Schicht“ des Turniers verarbeitet:

```
def KoMiniMatch(iPlayers,dPlayers):  
    counter = 0  
    dpairs = []  
    dWinner = []  
    while(counter != len(iPlayers)):  
        dpair = [iPlayers[counter],iPlayers[counter+1]]  
        dpairs.append(list(dpair))  
        counter += 2  
    for imatch in range(len(dpairs)):  
        dmatch = dpairs[imatch]  
        dPlayer1 = dmatch[0]  
        dPlayer2 = dmatch[1]  
        if(fight(int(dPlayers[dPlayer1]), int(dPlayers[dPlayer2]))==1):  
            dWinner.append(dPlayer1)  
        else:  
            dWinner.append(dPlayer2)  
    return(dWinner)
```

Die zweite Funktion des KO-Systems, lässt das KO-Turnier überhaupt erst funktionieren. Zuerst wird mit einer for-Schleife eine Spielerliste mit deren Index erstellt. Da nämlich in der KoMiniMatch-Funktion Spieler ausscheiden, kann man diese nicht mehr an ihrem Index der Liste bestimmen, sondern muss diese separat speichern. In einer while-Schleife, treten die Spieler mit der MiniMatch-Funktion gegeneinander an. Deren Sieger treten dann erneut gegeneinander an, so lange bis nur noch ein Spieler verbleibt, das ist der Gewinner. Dieser wird dann zurückgegeben.

Das KO5-System funktioniert fast genau so wie das normale KO-System, bis auf dass in der Ko5MiniMatch-Funktion alle Spielerpaare mit einer for-Schleife fünf mal gegeneinander antreten und bei jedem Gewinn die Punktzahl des Gewinners um 1 erhöht wird. Nach den 5 Spielen wird mit einer if-Bedingung der Spieler mit der höheren Punktzahl ermittelt und ab dann verläuft der Rest weiterhin so wie im normalem KO-System.

Die drei Hauptfunktionen der einzelnen Turniere werden im Hauptbestandteil, einer for-Schleife aufgerufen. Die for-Schleife läuft so oft, wie der Nutzer in einer Inputabfrage angegeben hat. Hier wird dem Hauptgewinner eines Turniers ein Punkt hinzugefügt:

```

for runs in range(int(input("Anzahl der Durchläufe für jede Turnierform: "))):
    ligawinner = LigaMatch(players)
    ligascores[ligawinner] += 1
    kowinner = KoMatch(players)
    koscores[int(kowinner[0])] += 1
    ko5winner = Ko5Match(players)
    ko5scores[int(ko5winner[0])] += 1

```

Sobald alle Turniere oft genug durchliefen, werden alle drei Gesamtgewinner der Turnierformen bestimmt und mit ihren Spielerstärken an den Nutzer ausgegeben.

Beispiele

Wir rufen nun das Programm mit der Beispieldatei „spielstaerken1.txt“ auf:

```
python Aufgabe3.py spielstaerken1.txt
```

Daraufhin erhalten wir eine Inputabfrage:

Anzahl der Durchläufe für jede Turnierform:

Damit das Programm richtig funktioniert, muss nun Minimum ein Wert von 1 eingegeben werden.

Für das Beispiel geben wir 20 ein. Daraufhin wird nun 20 mal jede Turnierform mit den Spielerstärken der Beispieldatei durchlaufen und wir erhalten folgendes Ergebnis:

Bei dem Ligaturnier gewann Spieler 8 mit Spielerstärke 100

Bei dem KO-Turnier gewann Spieler 8 mit Spielerstärke 100

Bei dem KO5-Turnier gewann Spieler 8 mit Spielerstärke 100

In diesem Beispiel hat Spieler 8 in allen Turnierformen gewonnen. Das liegt daran, dass er mit Spielstärke 100 und somit 40 Spielerstärken Abstand zu dem Zweitstärksten Spieler einen großen Vorteil hat.

Wenn wir das Programm erneut durchführen aber diesmal mit der Beispieldatei „spielstaerken4.txt“ und erneut mit 20 Durchläufen, erhalten wir folgendes:

Bei dem Ligaturnier gewann Spieler 1 mit Spielerstärke 100

Bei dem KO-Turnier gewann Spieler 5 mit Spielerstärke 95

Bei dem KO5-Turnier gewann Spieler 9 mit Spielerstärke 95

In diesem Beispiel ist der Abstand zwischen den Spielerstärken nur bei 5. Es gibt 16 Spieler, alle haben eine Spielerstärke von 95 außer Spieler 1 mit 100. Hier lässt sich also erkennen dass nicht nur der stärkste Spieler gewinnt.

Empfehlung an Tobi

Bei einer höheren Anzahl an Wiederholungen der Turniere, erhöht sich die Wahrscheinlichkeit dass der Stärkste Spieler gewinnt enorm. Trotzdem ist die Wahrscheinlichkeit dass der Stärkste Spieler gewinnt auch bei allen andern Formen hoch. Vor allem aber bei wenigen Durchläufen gewinnt der Stärkste Spieler vor allem bei den Liga-Turnieren häufig. Das liegt daran dass er hier gegen alle Spieler antritt und die Chance auf Siege höher ist.