

Ein Würfelspiel

Hinweis: Dieses Aufgabenblatt ist mit einer Bearbeitungszeit von zwei Wochen dafür ausgelegt in einem Zweierteam gelöst zu werden. Neben der Implementierung wird bei der Bearbeitung Wert auf die folgenden Punkte gelegt: Dokumentation², Strukturierung und Einhalten des Style-Guides³. Die genaue Aufschlüsselung der Bepunktung ergibt sich wie folgt:

- 50% entfallen auf die reine Funktionalität der Implementierung,
- 25% für die Einhaltung des Style-Guides (insbesondere also auch für Dokumentation) und
- 25% entfallen schließlich auf die angegebenen Testfälle.

Achten Sie darauf die Variable `__author__` in **allen** Quellcode Dateien **korrekt** zu setzen. Aufgaben, die nicht dieser Vorgabe entsprechen, werden **nicht** bewertet! Quellcode muss als `.py` Datei und alles Weitere als `.pdf` Datei abgegeben werden. Bei Abgaben mehrerer Dateien müssen diese als `.zip` zusammengefasst werden. Abgaben, die nicht diesen Regeln entsprechen, werden ebenfalls **nicht** bewertet! Außerdem muss ihr Name in jeder abgegebenen `.pdf` Datei zu finden sein. Nicht lauffähige Programme werden ebenfalls nicht bewertet, dabei gilt als Maßstab NUR die Ausführbarkeit aus der Konsole!

Σ __ / 20

Aufgabe 3.1: Allgemeiner Kommentar

Wir wollen uns mit dieser Aufgabe intensiv mit Funktionen und Prozeduren beschäftigen. In Ihrer Python Installation sind sehr viele vorgefertigte Pakete in der sogenannten „Standard Library“ enthalten, die man unbedingt benutzen sollte, wenn sie zur Problemlösung beitragen können. Viele Funktionen sind in C programmiert, erscheinen jetzt aber einem „Python Kleidchen“, sind aber trotzdem entsprechend effizient. Informieren Sie sich gründlich

1. übersichtsartig darüber, was es so alles in der „Python Standard Library“ gibt, siehe <https://docs.python.org/3/library/index.html>
2. speziell über den Modul `random`, siehe <https://docs.python.org/3/library/random.html>

Aufgabe 3.2: Das Spiel

„16-ist-tot“ ist ein Knobelspiel, das häufig in Kneipen gespielt wird. Oft ist der Einsatz das Bezahlen der nächsten Runde. Gewürfelt wird reihum mit einem Würfel. Jeder Spieler darf solange würfeln, wie er will, die Würfe werden summiert und notiert. Danach wird der Knobelbecher weiter gereicht. Ziel ist es, durch mehrmaliges Würfeln

¹Es dürfen keine Lösungen aus dem Skript, dem Internet oder anderen Quellen abgeschrieben werden. Diese Quellen dürfen nur mit Quellenangaben verwendet werden und es muss ein hinreichend großer Eigenanteil in den Lösungen deutlich zu erkennen sein.

²<http://www.python.org/dev/peps/pep-0257>

³<https://www.python.org/dev/peps/pep-0008/>

und aufsummieren der Würfelergebnisse so nahe an 16 zu kommen wie möglich. Wer 16 oder mehr erreicht, hat sofort verloren – das Spiel ist beendet. Es gelten folgende Sonderregeln:

- Wer als Summenpunkte 9 erreicht, darf nicht mehr würfeln. Der nächste Spieler kommt dran.
- Wer als Summenpunkte 10 erreicht, muss noch einmal werfen. Erfolgt im programmierten Spiel automatisch nach drei Sekunden Wartezeit.

Hat keiner der Spieler direkt verloren (also in Summe 16 oder mehr Punkte erreicht), so verliert(en) diejenigen, die am wenigsten Punkte erreicht haben.

Aufgabe 3.3: Implementierung

Punkte: ____ / 13

Implementieren Sie dieses Spiel!

- (a) (4 Punkte) Implementieren Sie eine Funktion `roll_dice(number=1, faces=6, seed=None)`.

Diese Funktion gibt einen String zurück, in dem die gewürfelten Zahlen im Intervall `[1..faces]` durch Komma getrennt stehen. Es wird mit `number` Würfeln (`number` im Intervall `[1,10]`) gewürfelt, je mit `faces` Flächen. `faces` ist aus dem Intervall `[2,100]` zu wählen. Bei jedem Würfel soll die Verteilung der Zahlen möglichst gleichverteilt sein und im Intervall `[1,faces]` liegen. `seed` ist eine Float-Zahl, die dem Quasi-Zufallszahlen Generator von Python als Parameter übergeben wird. Dies wird genutzt, um Würfelergebnisse wiederholbar (d.h. reproduzierbar) zu machen.

- (b) (4 Punkte) Implementieren Sie eine Funktion `roll_cheating_dice()`. Modifizieren Sie dabei den Würfelwurf eines elektronischen Würfel mit 6 Flächen (nur diesen!) so, dass die Zahl 3 relativ mit der doppelten Wahrscheinlichkeit erscheint, wie die anderen fünf Zahlen.

- (c) (5 Punkte) Implementieren Sie eine Funktion `sixteen_is_dead(players)`, welche den Ablauf des gesamten Spiels implementiert. `players` bezeichnet dabei die Anzahl an Spielern

Aufgabe 3.4: User Interface

Punkte: ____ / 4

Entwickeln Sie eine einfache Benutzungsschnittstelle auf Konsolenebene, die sich durch Tastatureingaben bedienen lässt. Diese soll folgende Möglichkeiten bieten:

1. Dem Benutzer zu Beginn mitteilen, was gültige Eingaben sind und was diese bewirken. Dabei soll Eingabe (Return) das erneute Würfeln bewirken. Das Weiterreichen des Knobelbechers zeigt man durch die Eingabe von `n` (für next) an.
2. Kennzeichnung, welcher Spieler gerade am Zug ist,
3. dem Benutzer des Spieles die Möglichkeit zu bieten, das Spiel jederzeit zu beenden oder eine neue Runde zu starten,

Aufgabe 3.5: Fehlerbehandlung

Punkte: ____ / 3

Ihr *User Interface* soll robust angelegt sein, sodass falsche Eingaben des Benutzers **nicht** zu einem Absturz führen.