

Programmieren 1 (PRG1)

Übung 7

7.2 Analyse

Programmierkonventionen:

Konventionen weitestgehend eingehalten. Einige Variablen wurden schlecht benannt (Zeile 24, 39, 73, 121: `for i in...`; Zeile 35: `for p in...`).

Zeile 37: keine Leerstellen zwischen „players“, + und 1

Zeilen 62/68/125: Geschlossene Klammer nicht unterhalb von mehrzeiligem String

Kommentare:

Docstrings sind passend gewählt und kurzgehalten. Allerdings fehlt am Anfang des Moduls eine Zusammenfassung aller enthaltenen Funktionen und deren Aufgabe.

Leider ungenügende Kommentierung. Außerhalb von Docstrings wird dem Leser des Codes keine Hilfestellung gegeben zu verstehen, wie die unterschiedlichen Funktionen arbeiten, warum sie existieren oder warum die unterschiedlichen Operationen in ihnen durchgeführt werden.

Hier muss dringend nachgebessert werden!

Bsp.:

Zeilen 16 und 17

```
assert faces >= 1
```

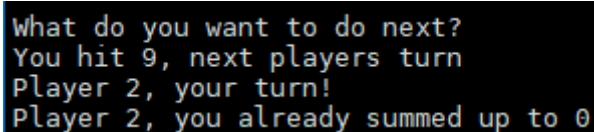
```
assert number >= 0
```

Der Leser erhält keine Information wofür diese Variablen verwendet werden oder warum sie `>= 1/0` sein müssen

User-Interface:

Das UI ist grundsätzlich solide designt, allerdings wäre es gut ein paar zusätzliche Leerspalten zwischen Texten einzufügen, um die Lesbarkeit zu verbessern.

Bsp:



```
What do you want to do next?
You hit 9, next players turn
Player 2, your turn!
Player 2, you already summed up to 0
```

Leerzeile zwischen „You hit 9...“ und „Player 2,...“ einfügen.

```
This is a fun game for 5 players
Your options:
roll the dice: <Enter>new game: new
Next player: n
Rules: r
Quit: exit
Player 1, your turn!
Player 1, you already summed up to 0
```

Leerstelle zwischen „Quit:...“ und „Player 1,...“ einfügen.

Fehlender „\n“ Befehl in String am Ende von „<Enter>“

Verbesserung der Lesbarkeit:

roll\_cheating\_dice kompliziert programmiert. Zeilen 35-37 evtl. redundant, da nur überprüft wird ob modifizierte Chancen korrekt sind. Allerdings hat User keinen Zugriff darauf, nur ein Programmierer der den Code ändern will. Zeilen 38 bis 42 schwer zu verstehen. Umsetzung scheint zu funktionieren, allerdings wäre eine Kommentierung an dieser Stelle sehr sinnvoll gewesen.

Das Einfügen von ein Leerzeilen zwischen abgeschlossenen Zeilen könnte die Lesbarkeit des Codes verbessern. (Bsp: Zwischen Zeilen 78 und 79, 96 und 97, 100 und 101, etc.)

Funktionalitäten:

Ausführbarkeit:

Programm lässt sich problemlos in der Konsole ausführen.

Würfeln:

Funktioniert ohne Probleme.

Nächster Spieler:

Funktioniert ohne Probleme.

Spiel neu starten:

Funktioniert ohne Probleme. Funktion aber aufgrund eines Fehlenden „\n“ in der Spielbeschreibung sehr leicht zu übersehen.

Spiel beenden:

Funktioniert ohne Probleme, allerdings erhält der User kein wirkliches Feedback, dass das Spiel beendet wurde.

Zusätzliche Funktionen:

Ausgabe der Spielregeln zu jedem Zeitpunkt. Komfortable Funktion, die dem User die Interaktion mit dem Programm erleichtert

Programm öffnet jedes Mal nach der Eingabe der Spielerzahl ein neues cmd Fenster, um ein „sauberes“ Fenster für das Spiel zu haben. Dies ist eine nützliche Implementierung, kann aber bei Testläufen behindern oder wenn der User die Aktionen in seinem Vorherigen Fenster behalten möchte. Evtl. wäre es sinnvoller ein neues Fenster zu öffnen, dass sich automatisch schließt, wenn das Spiel vorbei ist.

### 7.3

a)

Liste an Vorgenommenen Veränderungen:

Zeile (neu)	Änderung
8 – 12	Eigene Author Variablen eingefügt, für den Fall Richtlinien für .py abgaben in dieser Aufgabe zutreffen
15	Zahl der Würfel von 1 auf 3 hochgesetzt
47 – 69	Einleitender Text und Regelwerk aktualisiert
86	Entfernt (#)
90 – 98	Entfernt (#)
99 – 194	Neuen Code eingefügt
164 – 183	Code aus Original recycled (Original Zeilen 83 – 94)
219 – 230	Code aus Original recycled (Original Zeilen 119 – 126)
196 – 217	Entfernt (#)
227	Zusätzliche Leerzeilen eingefügt um UI zu verbessern
237 – 247	Ausgabertext verbessert, an Bedingung Spielerzahl $\geq 2$ angepasst und Gatekeeper eingefügt, der Eingabe bei Spielerzahl $< 2$ erneut aufruft

b)

Das Programm war grundsätzlich gut Strukturiert und bot eine solide Basis für Anpassung/ Optimierung. Ein Beispiel hierfür ist die einfache Änderung der Anzahl der geworfenen Würfel mittels einer Variablen. Anfang und Ende des Programms waren so allgemein geschrieben, dass sie (mit Ausnahme des UI einfach so übernommen werden konnten. Besonders der Umgang mit Userbefehlen außerhalb des Spiels („new“, „exit“ und „r“ (rules)) und variablen prints (über {} und .format() ) war elegant umgesetzt und hat bei der Modifikation des Programmes sehr gut geholfen.

Alles in allem war der Code nicht übermäßig flexibel, aber sauber programmiert, größtenteils gut verständlich und im Rahmen seiner Möglichkeiten offen für Anpassungen.

(Anmerkung: Allerdings müssen das nächste Mal unbedingt Kommentare eingefügt werden)