2. Aufgabenblatt fürs Tutorium

Konzepte der Programmierung

WiSe 2025/26

Kristin Knorr

Aufgabe 1 Abenteuer mit Kontrollfluss-Strukturen

Schreiben Sie ein Programm, in dem der/die Spieler:in in einer Fantasy-Welt eine Drachenhöhle erkundet. Das Programm läuft so ab, dass der spielenden Person ein kleiner Text angezeigt wird, der beschreibt, was gerade passiert. Es werden dann mehrere Handlungsoptionen vorgeschlagen, von denen der/die Spieler:in eine wählen soll. Je nach gewählter Option folgt ein neuer Text, der die Handlung beschreibt und weitere Optionen gibt, bis das Ende der Geschichte erreicht ist.

(a) Geben Sie mit der `print()`-Funktion einen Willkommenstext, der das Spiel erklärt und die Ausgangssituation beschreibt, aus. [1]

(b) Fragen Sie den/die Spieler:in nach dem Namen mittels der `input()`-Funktion und speichern sie diesen ab (und verwenden ihn ab jetzt in den Beschreibungen)

(c) Entwerfen Sie einen Spielverlauf, an dem mindestens drei "richtige" Entscheidungen getroffen werden müssen, damit die Geschichte ein Happy[2] End nimmt. Um die Anzahl der Handlungsstränge zu begrenzen dürfen "falsche" Optionen vorzeitig enden.

Ein Beispiel, wie das aussehen könnte:

```
Du stehst vor der Höhle, was willst du tun? (gib die
    entsprechende Zahl ein)
1 - schreiend in die Höhle rennen
2 - leise in die Höhle schleichen
3 - schreiend in die Höhle schleichen
4 - ein Picknick aufbauen und den Drachen einladen
```

Fordern Sie den/die Spieler:in auf, eine Antwort einzugeben und speichern Sie diese. Prüfen Sie mit `if-else`, welche Option gewählt wurde und lassen die Geschichte je nach gegebener Antwort unterschiedlich weiter verlaufen.

[1] Falls Ihnen nichts einfallen sollte, hier eine Anregung:
Willkommen bei <Ihr Name>s super tollem Kontrollstrukturenabenteuer! Du wirst gleich vor eine Reihe schwieriger Entscheidungen gestellt, bei denen du eine von vier möglichen Vorgehensweisen wählen musst. Damit bestimmst du den Verlauf der Geschichte. Schaffst du es, die Gefahr zu bannen? Und nun beginnt es:
Du erreichst nach einem langen Marsch endlich die Höhle des gefürchtetsten Drachen der gesamten bekannten Welt, dem Draco-Fibonacci. Er mag gerne Ziffern, doch heute wird er zahlen! In deiner Hand das mächtige Schwert Drachus-Köpfus, mit dem du der Bestie den Bauch aufschlitzen wirst, wendest du dich dem Eingang zu, aus dem es riecht, als ob das Untier sich seit der letzten Sonnenfinsternis nicht mehr gewaschen hätte (klar, Drachen mögen kein Wasser).

[2] Was als happy gilt, ist Ihnen überlassen, vielleicht auch happy für den Drachen?

(d) Wer eine zusätzliche Herausforderung will, kann frühere Entscheidungen nachwirken lassen (Wenn die Höhle schreiend betreten wurde, kommen einem z.B. verängstigte Goblins entgegen)

(e) Lassen Sie ihre/n Sitznachbar:in Ihr Spiel spielen (**Ich meine es ernst!**)

Aufgabe 2 Verbesserungen mit Schleifen

(a) Wenn der/die Spieler:in nicht das eingibt, womit das Programm rechnet, kann das ein Problem sein (was passiert dann und warum ist das so?). Schreiben Sie Ihr Abenteuer so um, dass solange nach einer neuen Eingabe gefragt wird, bis eine erwartete Antwort gegeben wurde. (Falls Zahlen eingegeben werden, muss nur abgefangen werden, dass die Zahl zu keiner Antwortmöglichkeit passt.)

(b) Schreiben Sie Ihr Abenteuer so mit einer Schleife um, dass der/die Spieler:in drei Versuche erhält, das "Happy End" zu erreichen. Bei einem vorzeitigen Ende soll wieder von vorne begonnen werden, bei einem geglückten Versuch aber nicht.

Aufgabe 3 Personenbezogene Datenstrukturen

Stellen Sie den Charakter aus Ihrem Abenteuer als eine Liste[3] dar. Darin sind gespeichert: Der Name des Charakters an erster Stelle, sein Level an zweiter Stelle, seine Lieblingszahl an dritter Stelle und der Inhalt seines Beutels (in den beliebig viele Gegenstände passen) an vierter Stelle.

Implementieren Sie Ihre Liste so, dass folgendes Programm funktioniert:

```
my_character = # hier Ihr Code

print("Hallo, ich heiße ", my_character[0])
print("Ich bin schon auf Level ", my_character[1])
print("und meine Lieblingszahl ist ", my_character[2])

print("In meinem Beutel habe ich folgendes:")

for item in my_character[3]:
    print(item)
```

Implementieren Sie eine Abfrage, in der ein/e Nutzer:in einen Gegenstand in den Beutel legen kann. Dann soll der Inhalt des Beutels nochmal ausgegeben werden.

[3] Wie man das eleganter machen kann, lernen Sie gegen Ende des Semesters, aber hier wird nicht gespoilert ;)