

Einführung in die Programmierung

Prof. Dr. Franziska Matthäus
Prof. Dr. Matthias Kaschube
Dr. Karsten Tolle
Lukas Müller



Übungsblatt ÜE-05

Ausgabe: 24.11.2021
Abgabe: 11.12.2021
16:00 Uhr

Ein Spiel

Hinweis:

- *Es sind grundsätzlich Rechenwege anzugeben, es sei denn es findet sich ein expliziter Hinweis, dass dies nicht nötig ist.*
- *Es 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.*
- *Digitale Abgaben, die nicht im Format **.pdf** oder **.txt** für Texte oder **.py** für Code erfolgen, werden nicht bewertet. Bei Abgaben mehrerer Dateien müssen diese als **.zip** zusammengefasst werden.*
- *Achten Sie darauf die Variable **__author__** in allen Quellcode Dateien korrekt zu setzen (am Anfang des Quellcodes):*
`__author__ = "<Matr-Nr>, <Nachname>"` bzw. bei Zweiertams
`__author__ = "<Matr-Nr_1>, <Nachname_1>, <Matr-Nr_2>, <Nachname_2>"`
- *Außerdem muss Ihr Name in jeder abgegebenen **.pdf** und **.txt** Datei zu finden sein.*
- *Abgaben, die per Hand geschrieben und eingescannt werden, sind **nicht** erlaubt (bzw. geben 0 Punkte und werden nicht korrigiert).*
- *Beim Programmieren und Kommentieren halten Sie sich die Regeln im Programmierhandbuch, siehe Moodle-Kurs ([Programmierhandbuch WiSe 21/22 \(Style Guide\)](#)). Im Zweifelsfall gilt PEP 8.*

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: Dokumentation (auch die Benutzerführung sollte erklärt werden (Benutzerhandbuch), was bei dem BMI-Beispiel durch fehlende Komplexität nicht nötig war), Docstrings (<http://www.python.org/dev/peps/pep-0257>), Strukturierung und Einhalten des Style-Guides.

Wir empfehlen dringend in Zweiertams zu arbeiten. Teamarbeit (e.g. Pair-Programming) ist ja auch Teil des Lerninhaltes. Es ist vorteilhaft, dass die Teams aus einem Tutorium stammen. Die Tutoren sind angehalten in den Tutorien die Teambildung zu unterstützen. Die finale Bewertung durch den Tutor erfolgt dabei erst nach einem erfolgreichem Code-Review. Dort wird geprüft, ob von allen Teammitgliedern der abgegebene Inhalt auch verstanden wurde. ... es geht also nicht darum die Arbeit einfach aufzuteilen! **Jeder muss alles verstehen!**

Pro Team nur eine Abgabe in Moodle!

Aufgabe – Wizard mal anders**20 Punkte**

Implementieren Sie ein Spiel, das ähnlich zu dem ist, was Sie im letzten Aufgabenblatt umgesetzt haben. Das Spiel soll wie folgt funktionieren:

- Es wird mit 60 Karten (15 * 4 Farben) gespielt (Kartendeck).
- Das Spiel kann mit 2-5 Spieler*innen gespielt werden.
- Es wird ein Startspieler*in und eine Reihenfolge festgelegt.
- Es werden eine Anzahl an Runden gespielt (abhängig davon wie viele mitspielen). Die Anzahl der Runden ergibt sich aus der Formel:

$$\text{Anzahl Runden} = \left\lfloor \frac{\text{Anzahl Karten insgesamt} - 1}{\text{Anzahl Spielenden}} \right\rfloor$$

- In jeder Runde wird:
 - Jedem Spielenden werden Karten ausgeteilt (die Anzahl der Karten soll der Rundenzahl entsprechen). In der ersten Runde erhält jede*r eine Karte, in der zweiten Runde erhält jede*r zwei Karten und so weiter.
 - Die Trumpffarbe für die Runde wird ermittelt: Hierzu wird die oberste Karte vom verbliebenen Kartendeck umgedreht. Die Farbe dieser Karte ist die Trumpffarbe.
 - Ab jetzt wie beim Aufgabenblatt EPR 04: Jede*r legt eine Karte (beginnend bei dem*der Startspieler*in), der*die Spieler*in mit der größten Karte gewinnt den Stich (was beim letzten Aufgabenblatt auch Runde hieß). Die nächste interne Runde wird dann vom nächsten Spielenden eröffnet. Je nach der Anzahl der ausgeteilten Karten, gibt es also in dieser Runde entsprechend viele Stiche.
 - Am Ende der Runde werden alle Karten zusammengetan und neu gemischt. In der nächsten Runde startet der*die nächste Spieler*in in der Reihenfolge als Startspieler.
- Ermittlung des Gewinners:
 - Hier bitte mindestens zwei Varianten umsetzen und über Einstellungen beim Start des Spieles zur Verfügung stellen. Denkbar wäre der*die Spieler*in: a) mit den meisten Stichen, b) mit den meisten gewonnen Runden, c) mit den meisten Punkten in den Stichen ...
Hinweis: *Dictionaries* könnten für die Speicherung der Wertungen hilfreich sein ☺

Zusatzanforderungen und Hinweise (bei Nichteinhaltung gibt es Punktabzüge):

- Erstellen Sie **zwei verschiedene** Computer-Gegner, welche die Aktionen von Spielern, die als Computer-Gegner definiert werden, durchführen können (z.B. einen sehr einfachen – Random und einen „schlaueren“). **(3 Punkte)**
- Entwickeln Sie eine einfache Benutzungsschnittstelle (User Interface) auf Konsolenebene, die sich durch Tastatureingaben bedienen lässt und den Benutzer sinnvoll durch das Programm leitet und mindestens folgende Möglichkeiten bietet: **(6 Punkte)**
 - Dem Benutzer zu Beginn mitteilen, was gültige Eingaben sind und was diese bewirken.
 - Kennzeichnung, welcher Spieler gerade am Zug ist.
 - Der Benutzer soll die Möglichkeit haben das Spiel zu beenden oder eine neue Runde zu starten.
 - Ihr User Interface soll robust angelegt sein, sodass falsche Eingaben des Benutzers nicht zu einem Absturz führen.
- Für die Umsetzung sollen alle drei Funktionen aus dem Modul *cards* verwendet werden. Dieses Modul enthält die Funktionen des Blatts EPR 04 (Teil-Aufgaben a, b und c). Die Funktionen haben Sie schon im

Blatt 04 programmiert, diese können sie erst einmal verwenden. Das Modul *cards* wird erst nach Abgabefrist vom Blatt 04 im Moodle-Kurs veröffentlicht.

- Strukturieren/ergänzen Sie ihre Implementierung mindestens durch die Erstellung weiterer **eigener drei** Funktionen (mit entsprechenden Testfällen).
- (nochmals ;-)) Neben der Implementierung wird bei der Bearbeitung Wert auf die folgenden Punkte gelegt: Dokumentation (auch die Benutzerführung sollte erklärt werden (Benutzerhandbuch), was bei dem BMI-Beispiel durch fehlende Komplexität nicht nötig war), Docstrings (<http://www.python.org/dev/peps/pep-0257>), Strukturierung und Einhalten des Style-Guides.
- Für fehlende Angaben in der Aufgabestellungen dürfen Annahmen aufgestellt werden, die in der Dokumentation beschreiben werden.