

Vier gewinnt

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-Guides^{3,4}. 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)
- 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 nicht bewertet, dabei gilt als Maßstab NUR die Ausführbarkeit in der Konsole! Bitte achten Sie auf die Einhaltung der **Programmierrichtlinie**. Sonst gibt es leider Punktabzüge.

Σ ___ / 20

Aufgabe 5.1: Das Spiel

Ziel dieses Übungsblattes ist es, eine Variation von *Vier gewinnt* zu implementieren. Bei *Vier gewinnt* wird auf einem aufrecht stehenden Spielbrett gespielt, in welches die Spieler abwechselnd Spielsteine fallen lassen.

Bei dieser Variante gewinnt man allerdings *nicht*, indem man vier Steine in einer Reihe erhält, sondern indem die gelegten Steine in ihrer Form einer der folgenden Figuren entsprechen:

1.	X X X X	2.	X X X X	3.	X X X X	4.	X X X X
----	------------	----	------------	----	---------------	----	---------------

Überlegen und notieren Sie zu allen bearbeiteten Teilaufgaben jeweils mindestens 4 geeignete Testfälle. Beachten Sie die Punkteverteilung, welche unter den Hinweisen auf der ersten Seite angegeben ist!

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/>

⁴https://moodle.studiumdigitale.uni-frankfurt.de/moodle3/pluginfile.php/56989/mod_resource/content/7/Programmierhandbuch-2018_V18-19-1.pdf

Aufgabe 5.2: Implementierung

Punkte: ____ / 12

Bedingung: Schreiben Sie ein Benutzungshandbuch und erläutern Sie, wie ihre Implementierung zu bedienen ist.

- (a) (6 Punkte) Implementieren Sie das Spiel für zwei Spieler. Es soll zu Beginn des Spiels automatisch ein Spielfeld von der Größe 9×10 angelegt werden. Daraufhin machen die Spieler abwechselnd Eingaben, in welche Spalte Sie den Stein ihrer Farbe fallen lassen.

Hinweis: Bevor Sie mit der Implementierung beginnen, überlegen Sie, wie Sie das Spielfeld verwalten wollen. Manchmal kann es sinnvoll sein, zwischen der Ausgabe des Spielfeldes auf dem Monitor und der internen Repräsentation zu unterscheiden.

- (b) (3 Punkte) Erweitern Sie Ihr Programm dahingehend, dass nach jedem Zug überprüft wird, ob ein Spieler gewonnen hat. Gibt es einen Sieger soll diese Spielrunde beendet werden.

- (c) (3 Punkte) Erweitern Sie Ihr Programm um die Möglichkeit eines Einzelspielermodus, bei dem der Computer die Rolle des Gegners übernimmt.

Punktevergabe auf die Lösungsvorschläge:

- Randomisierte Auswahl: 1 Punkt.
- Eine rein Greedy-basierte Auswahl: 2 Punkte.
- Die KI plant mindestens 1 Zug voraus und versucht zu gewinnen oder die Gegenspieler am Gewinn zu hindern. 3 Punkte

Aufgabe 5.3: User Interface

Punkte: ____ / 5

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

1. Dem Benutzer mitteilen, was gültige Eingaben sind.
2. Kennzeichnung, welcher Spieler gerade am Zug ist.
3. Die nächsten Züge abzufragen, bis keine Züge mehr möglich sind.
4. Dem Benutzer des Spieles die Möglichkeit zu bieten, das Spiel jederzeit zu beenden oder eine neue Runde zu starten.
5. Nach jedem Zug das aktuelle Spielfeld ausgeben.

Aufgabe 5.4: Fehlerbehandlung

Punkte: ____ / 3

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