

## Einführung in die Programmierung ÜBUNGSBLATT 5 Vier gewinnt

### Aufgabe 5.1: Das Spiel

Testfälle zu Aufgabe 5.2 a).

#### 1. Implementieren Sie das Spiel für zwei Spieler...:

```
Please enter a number of players between 1 and 2 >>> 3  
This is no valid input.  
Please enter a number of players between 1 and 2 >>> TERM environment variable not set.
```

In der Aufgabenstellung wird die Eingabe einer Spieleranzahl zwischen 1 (aufgrund von Teilaufgabe c)) und 2 Spielern verlangt. Damit das gewährleistet werden kann testen wir, ob das Programm erkennt, dass die Eingabe nicht innerhalb des angegebenen Bereiches liegt.

Das Programm erkennt die Fehleingabe, löscht die Konsole und meldet „invalid input“. Es verlangt nach einer neuen Eingabe.

#### 2. Implementieren Sie das Spiel für zwei Spieler...:

```
Please enter a number of players between 1 and 2 >>> 2  
You chose 2  
TERM environment variable not set.  
Please enter your name Player 1 >>> Felix  
TERM environment variable not set.  
Player 1, your name is Felix  
Would you like to keep the name Felix? y/n >>> n  
Please enter your name Player 1 >>> TERM environment variable not set.  
Felix  
TERM environment variable not set.  
Player 1, your name is Felix  
Would you like to keep the name Felix? y/n >>>
```

Nach der korrekten Eingabe einer Spielerzahl haben die Spieler die Möglichkeit ihre Namen einzugeben. Falls der Name versehentlich falsch eingegeben wurde können Spieler diesen noch ändern. In diesem Testfall überprüfen wir ob die Änderung vom Programm richtig wahrgenommen wurde und der geänderte Name unter dem jeweiligen Spieler gespeichert wurde. Wir erkennen, dass es funktioniert daran, dass das Programm uns wissen lässt, welcher Spieler den neuen Namen erhalten hat. Dieser Testfall ist notwendig um zu überprüfen, dass der geänderte Name nicht versehentlich dem anderen Spieler zugeteilt wird.

#### 3. Implementieren Sie das Spiel für zwei Spieler...:

```

Please enter a number of players between 1 and 2 >>> 2
TERM environment variable not set.
You chose 2
Please enter your name Player 1 >>> Felix
Player 1, your name is Felix
TERM environment variable not set.
Would you like to keep the name Felix? y/n >>> Yes
TERM environment variable not set.
Please enter your name Player 2 >>> Dan
Player 2, your name is Dan
TERM environment variable not set.
Would you like to keep the name Dan? y/n >>> what?
TERM environment variable not set.
TERM environment variable not set.
This is no valid input
Would you like to keep the name Dan? y/n >>>

```

Wir überprüfen nun ob das Program statt der Eingabe eines ‚y‘ oder ‚n‘ auch ein ‚yes‘ oder ‚no‘ akzeptiert, sowie ob es akzeptiert wenn ein Buchstabe groß geschrieben wurde. Das machen wir, damit die Eingabe dem User erleichtert wird. Er muss nicht bei jeder Eingabe auf Groß- oder Kleinschreibung achten.

Auch überprüfen wir in dem selben Testfall ob eine falsche Eingabe (in diesem Fall ‚what?‘ von Dan) erkannt wird. Das ist wichtig, da eine klare Ansage vom User gefordert wird um Missverständnisse zu vermeiden.

#### 4. Daraufhin machen Spieler abwechselnd Eingaben, in welche Spalte Sie den Stein ihrer Farbe fallen lassen.....:

```

Now it's Felix's turn.

(To quit enter: 'q'. To restart enter: 'r')

Please enter the column that you want to play >>>
1  2  3  4  5  6  7  8  9  10
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
TERM environment variable not set.
1  2  3  4  5  6  7  8  9  10
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .  .  .
Now it's Dan's turn.

```

Das Programm könnte bei einer falschen Programmierung den Coin in die falsche Spalte werfen, da es die Spalten von 0 beginnend bis 9 (statt wie beim User Interface von 1 bis 10) zählt. Um dem vorzubeugen müssen wir überprüfen ob die Eingabe eines Spielers korrekt erkannt, sein Spielstein in die richtige Spalte ‚geworfen‘ wurde und der nächste Spieler dran kommt.

```

Now it's Dan's turn.

(To quit enter: 'q'. To restart enter: 'r')

Please enter the column that you want to play >>> what??
1 2 3 4 5 6 7 8 9 10
TERM environment variable not set.
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
Invalid input, please enter a column index between 1 and 10.
(To quit enter: 'q'. To restart enter: 'r')

Please enter the column that you want to play >>>

```

Anschliessend überprüfen wir ob das Programm bei Falscheingaben denselben Spieler zu einer erneuten Eingabe auffordert. Würde es Falscheingaben nicht erkennen würde das Programm zu einem Absturz führen, da es würde die Fortführung des weiteren Codes ein integer zwischen 1 und 10 benötigt.

## Testfälle zu Aufgabe 5.2 b).

### 1. Nach jedem Zug wird überprüft ob ein Spieler gewonnen hat....:

```

1 2 3 4 5 6 7 8 9 10
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
Now it's Felix's turn.

(To quit enter: 'q'. To restart enter: 'r')

Please enter the column that you want to play >>>

```

Damit die Regeln von Vier-Gewinnt eingehalten werden, schauen wir bei diesem Testfall erst einmal ob das Programm erkennt, dass noch niemand gewonnen hat und es dennoch weiter läuft.

### 2. Nach jedem Zug wird überprüft ob ein Spieler gewonnen hat....:

```

. . . . .
TERM environment variable not set.
1 2 3 4 5 6 7 8 9 10
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
Now it's Dan's turn.

(To quit enter: 'q'. To restart enter: 'r')

Please enter the column that you want to play >>>

```

Bei diesem Testfall sollte das Programm eigentlich erkennen, dass Spieler Felix

gewonnen hat. Vermutlich liegt ein Fehler in der Gewinnererkennung des Codes vor, daher läuft das Spiel, trotz erfolgreichem Legen der Figur weiter. Wir werden natürlich versuchen den Fehler zu beheben. Dieser Testfall ist notwendig, da das Spiel nach erfolgreichem Legen auch beendet, und der Gewinner genannt werden sollte.

### 3. Gibt es einen Sieger soll diese Spielrunde beendet werden:

```
TERM environment variable not set.
 1  2  3  4  5  6  7  8  9 10
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
Dan has won the game!
Press enter if you want to restart the game and enter 'q' if you want to quit the game!
TERM environment variable not set.
```

Wir stellen fest, ob das Programm auch bei weiteren Figuren keinen Gewinner ausruft. In diesem Fall funktioniert es jedoch. So wissen wir, dass der Fehler eventuell nur an der Codierung der einen Figur liegt und können diesen einfacher beheben. Auch sehen wir dadurch, dass die Spielrunde beim Erkennen eines Siegers beendet wird und nicht noch wie bei obigem Testfall weiterläuft.

### 4. Gibt es einen Sieger soll diese Spielrunde beendet werden:

```
TERM environment variable not set.
 1  2  3  4  5  6  7  8  9 10
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
x  x  x  x  x  x  x  x  x  x
Felix has won the game!
Press enter if you want to restart the game and enter 'q' if you want to quit the game!
```

Um weitere Fehler zu vermeiden gehen wir sicher, dass das Programm auch bei der ‚verwandten‘ Figur (spiegelverkehrt zu Figur aus Testfall 2) funktioniert und erkennt, dass in diesem Fall ein Spieler gewonnen hat.

### Testfälle zu Aufgabe 5.2 c).

Wir überprüfen nun in folgenden Testfällen, ob der User in der Lage ist einen Spieler auszuwählen und gegen die KI zu spielen.

#### 1. ...Einzelspielermodus...:

```
Please enter a number of players between 1 and 2 >>> 1
TERM environment variable not set.
You chose 1
Please enter your name Player 1 >>> Ian
TERM environment variable not set.
Player 1, your name is Ian
Would you like to keep the name Ian? y/n >>> y
TERM environment variable not set.
TERM environment variable not set.
Ian, your coin is X
```

Damit der Einzelspielermodus funktioniert, muss das Programm akzeptieren, dass nur 1 Spieler eingegeben wird. Das ist nach diesem Testfall nach zu urteilen möglich.

## 2. KI plant mindestens einen Zug voraus und versucht zu gewinnen...:

[illegible]

Bei diesem Testfall haben wir print Funktionen im Code gelassen um zu überprüfen ob die KI die Spielzüge und die Wahrscheinlichkeiten bei jedem Zug zu gewinnen richtig berechnet, und anhand dieser eine Spalte aussucht.

Der Ausgabe nach zu urteilen müsste sie sich für die Spalte mit den am wenigsten notwendigen Zügen (in diesem Fall die Zahl 3) und der höchsten Wahrscheinlichkeit (in diesem Fall 8) entscheiden.

### 3. randomisierte Auswahl...:

[illegible]

Wir überprüfen nun, ob die KI in der Lage ist bei zwei ‚gleichwertigen‘ Spielfeldern sich für eine ‚random‘ zu entscheiden, ohne eine Fehlermeldung auszuwerfen. Bei diesem Testfall ist mir aufgefallen, dass die KI aktuell die Spielzüge mit einem Punkt zu wenig berechnet. Zum Erzielen eines Gewinns benötigt Sie nach Ablegen eines Coins bei column 5 noch mindestens zwei, statt einem Zug. (Die KI berechnet auch die notwendigen Spielzüge des Gegenspielers. Sie werden jedoch in der list of score für column 5 nicht angezeigt, da der Gegenspieler in dieser Spalte 5 Spielzüge bräuchte und sie im Vergleich zu den nur 2 notwendigen Zügen der KI nicht relevant sind. )

#### 4. KI Spielzug:

The Computer has chosen column 6 !

Es soll die Information ausgegeben werden wo die KI ihren Coin abgelegt hat. Dies ist zwar nicht notwendig für die Funktionalität des Spiels, nimmt dem User jedoch die Arbeit ab nach dem abgeworfenen Coin der KI im Spielfeld zu suchen. Bei diesem Testfall überprüfen wir ob trotz while Schleife im Code und Löschen der Konsole die Information angezeigt wird.