

## Programmieren I, WS 2011/12

### Übung: Textadventure

Um gut programmieren zu lernen, gehört nicht nur, Aufgaben exakt nach Vorgaben und anhand von Hilfestellungen zu bearbeiten. Meistens kennt man nur das Ergebnis – man weiß also, wo man hinkommen möchte, aber wie man das genau anstellt, das muss man sich selbst überlegen.

Genau in diese Richtung wollen wir innerhalb der kommenden Wochen einmal einen ersten Gehversuch machen. Aber keine Angst! So schwer ist das gar nicht. Und meistens macht das knobeln über dem eigenen Code sogar Spaß! Wenn ihr doch einmal völlig hilflos in der Luft hängen bleiben solltet, dann sind Google und das Galileo OpenBook eure besten Freunde. Und wenn alle Stricke reißen, meldet ihr euch und wir suchen das Problem gemeinsam. :-)

#### Aufgabe 1

Die allerersten Gehversuche im Bereich Spieleentwicklung produzierten keine ultra-realistischen 3D-Spiele, wie man sie heute sehen kann: Früher hat man sogenannte „Textadventures“ programmiert. Dabei hat man sich per Texteingabe der Himmelsrichtungen („n“, „s“, „w“, „o“) von Raum zu Raum bewegt. In jedem Raum hat man einen Beschreibungstext erhalten, wo man sich gerade befindet, und eventuell auch, was man dort tun kann. Die Rätsel konnte man nur durch das Aufsammeln und geschickte Kombinieren von Gegenständen und Mechanismen lösen.

Keine Sorge, das ist gar nicht so schwer! Und für den Anfang wollen wir es nicht übertreiben. Wir wollen uns zunächst ein klares Ziel setzen: Der Spieler soll in einem bestimmten Raum starten. Dann soll er die Möglichkeit haben, sich zwischen drei bis vier Räumen hin- und her bewegen zu können. Was benötigen wir dazu alles?

Auf jeden Fall eine Benutzeroberfläche, welche den Spieler um eine Eingabe bittet. Diese könnte so aussehen:

```
Was möchtest du tun?
```

```
-
```

Damit wir auch tatsächlich unbegrenzt lange in unseren Räumen herumlaufen können, sollte diese Benutzerabfrage also in einer Schleife erfolgen. Am Besten in der Main-Funktion: so können wir alle Funktionalität, die eventuell später noch hinzu kommt, mit in diese Schleife packen.

Als zweites brauchen wir Räume. Es bietet sich an, eine Skizze der Räume auf Papier zu machen. Jeder der Räume soll außerdem mehrere Eigenschaften in sich speichern: Zum

Beispiel die Himmelsrichtungen, durch welche du den Raum verlassen kannst. Welche Datenstruktur würde sich hierfür eignen? Denke daran, dass es die „eine“, richtige Lösung bei dieser Aufgabe nicht mehr gibt! Vielleicht probierst du es mit Dictionaries, oder mit einer Listen/ Tupel-Kombination?

Unser erster Gehversuch in unserem Spiel sollte am Ende so ähnlich aussehen:

```
Du befindest dich hier: Bismarckplatz.  
Was möchtest du tun?  
- O
```

```
Du befindest dich hier: Hauptstrasse.  
Was möchtest du tun?  
- W
```

```
Du befindest dich hier: Bismarckplatz.  
Was möchtest du tun?  
- N
```

```
Du befindest dich hier: Theodor-Heuss-Brücke.  
Was möchtest du tun?  
-
```

## Aufgabe 2

Wenn du die ersten Schritte hinbekommen hast, können wir einen Schritt weiter gehen. Die Räume sind nun „hart gecodet“, das heißt fest im Code verankert. Schöner wäre es, wenn unser kleines Programm da etwas flexibler wäre. Wie wäre es zum Beispiel, wenn wir die Räume in eine Textdatei schreiben könnten, und das Programm erstellt daraus automatisch unser Spiel?

Überlege dir, wie wir das anstellen könnten. Wir brauchen derzeit vor allem zwei Informationen zu jedem Raum: Wie heißt er, und wohin kannst du dich von dort aus bewegen. Etwas unsauber könnten wir also zum Beispiel ein festes Format beschließen: Wir lesen die Datei ein, und jede neue Zeile beinhaltet die Infos für genau einen Raum. Oder du widmest jedem Raum eine feste Anzahl an Zeilen: In der ersten steht der Name, danach folgen die möglichen Himmelsrichtungen.

Die Datei könnte dann beispielsweise so aussehen:

```
Bismarckplatz
n Theodor-Heuss-Brücke
s
w
o Hauptstrasse
```

```
Hauptstrasse
n
s
w Bismarckplatz
o
```

...

Wenn du mit dem Format deiner Datei nun zufrieden bist, dann musst du die Inhalte nur noch mit einfachen Datei- und Stringfunktionen einlesen und dann in das gleiche Format bringen, in dem du sie zuvor hart gecodet hattest.

Jetzt kann sich jeder Spieler seine eigenen Räume erstellen!

## Aufgabe 3

Folgt...