

## Programmieren 1 – WS 2015/16

Prof. Dr. Michael Rohs, Henning Pohl, M.Sc., Oliver Beren Kaul, M.Sc.

# Übungsblatt Challenge

Dieses Übungsblatt darf einzeln oder in Zweiergruppen bearbeitet werden. Dieses Übungsblatt zählt nicht als reguläres Übungsblatt. Die Bearbeitung ist freiwillig.

Abgabe bis Freitag den 8.1. um 23:59 Uhr über <https://assignments.hci.uni-hannover.de>. Erstellen Sie alle zur Abgabe gehörigen Dateien im Verzeichnis `assignment_challenge` (s.u.). Entfernen Sie alle vom Compiler generierten Dateien (z.B. `exe`-Dateien) aus dem Verzeichnis und komprimieren Sie es zu `assignment_challenge.zip`. Laden Sie diese Datei dann bitte hoch.

### Allgemeine Hinweise zur Kompilierung

Wenn noch nicht geschehen, laden Sie die Programming I C Library herunter: <http://hci.uni-hannover.de/files/prog1lib-0.5.zip>. Laden Sie `assignment_challenge.zip` mit den Template-Dateien für dieses Übungsblatt aus Stud.IP herunter und speichern Sie `assignment_challenge.zip` im gleichen Verzeichnis wie `prog1lib-0.5.zip`. Folgende Kommandos sind auf der Konsole erforderlich, um die Bibliothek zu kompilieren und das Template dieser Aufgabe zu nutzen:

<code>unzip prog1lib-0.5.zip</code>	← Bibliothek entpacken
<code>unzip assignment_challenge.zip</code>	← Template-Dateien dieser Übung entpacken
<code>cd prog1lib/lib</code>	← change directory to prog1lib/lib
<code>make</code>	← create libprog1.a
<code>cd ../../assignment_challenge</code>	← change directory to assignment_challenge
<code>make adventure</code>	← create executable

Hinweise:

- Wenn Sie beim Kompilieren die Fehlermeldung bekommen, dass `base.h` nicht gefunden wurde, dann weicht Ihre Verzeichnisstruktur von der geforderten ab. Überprüfen Sie erneut die eingangs angegebenen Schritte.
- Die Dokumentation der `prog1lib` finden Sie unter <http://hci.uni-hannover.de/files/prog1lib/index.html>

## Aufgabe 1: Text-Adventure

In dieser Aufgabe soll ein Text-Adventure ausgestaltet werden. Eine rudimentäre Implementierung, die im Rahmen der Aufgabe weiter ausgebaut werden soll, findet sich in der Template-Datei `adventure.c`. Das Spielbrett besteht aus 10 x 10 Feldern. Ziel des Spiels ist es, den Weihnachtsbaum einzusammeln, der sich auf einem der Felder befindet. Die Spielfigur steht zu Beginn links oben auf dem Spielbrett (Position 0, 0). Die Spielfigur kann mit folgenden Kommandos bewegt werden:

- q: quit the game
- l: go left
- r: go right
- u: go up
- d: go down

Auf jedem Feld können sich mehrere Objekte befinden, z.B. ein Schlüssel und ein Apfel. Felder können auch mit einem Mauerstück oder einer Tür belegt sein. Manche Objekte lassen sich von der Spielfigur aufsammeln (`get`). Gesammelte Objekte können auf Feldern abgelegt werden (`put`). Die zugehörigen Kommandos lauten:

- g <object>: get the object, example: g key (Anfangsbuchstabe reicht: g k)
- p <object>: put the object, example: p apple

Manche Hindernisse lassen sich nur überwinden, wenn der Spieler bestimmte Objekte mit sich führt. Im Template befindet sich z.B. auf Feld (0, 1) eine Tür, die nur geöffnet werden kann, wenn die Spielfigur den Schlüssel mit sich führt. Wenn bestimmte von der Spielfigur mitgeführte Objekte abgelegt werden, kann dies den Spielzustand beeinflussen. Im Template befindet sich z.B. auf Feld (0, 2) ein hungriges Monster. Dies kann besänftigt und zum Schlafen gebracht werden, indem der Spieler den Apfel auf dem Feld ablegt. Danach kann das Feld gefahrlos verlassen werden. Wenn das Monster nicht gefüttert wird, frisst es den Spieler beim Verlassen des Feldes.

Die Spielfigur kann außerdem Äußerungen von sich geben. Das `say`-Kommando lautet:

- s <statement>: says something, example: s hello, how are you?

Hinweise zu wesentlichen Daten und Funktionen im Template:

- `List stuff` ist eine Liste von Strings. Sie repräsentiert die Objekte, die von der Spielfigur mitgeführt werden.
- `List cannot_get` ist eine Liste von Strings. Sie repräsentiert Objekte, die nicht eingesammelt werden können, wie z.B. Mauern.
- `Position position` repräsentiert die aktuelle Position des Spielers auf dem Spielbrett.
- `Position previous_position` repräsentiert die vorherige Position des Spielers auf dem Spielbrett.
- Die Funktion `initialize_game` setzt das Spiel in den Grundzustand.



- Die Funktion `print_situation` gibt den aktuellen Spielzustand aus.
- Die Funktion `evaluate_situation` evaluiert den aktuellen Spielzustand und ändert ihn ggf. entsprechend der Spiellogik.
- Die Funktion `play_adventure` verarbeitet Eingaben von der Kommandozeile. Die Funktion ruft `evaluate_situation` und `print_situation` auf.

Die Aufgabe besteht darin, das Template zu einem möglichst interessanten Text-Adventure auszubauen. Die im Template verwendeten Spielobjekte dürfen auch komplett ersetzt werden. Dazu ist es sinnvoll, die Funktionen `initialize_game`, `print_situation` und `evaluate_situation` sowie ggf. die Spielfeldgröße anzupassen.

Das beste Text-Adventure wird mit einem 3D-gedruckten Pokal ausgezeichnet. Die Zweit- und Drittplatzierten bekommen eine Medaille. Die Gewinner werden von den Teilnehmern der Veranstaltung per Abstimmung über Stud.IP ermittelt. Dazu werden die abgegebenen Text-Adventures auf Stud.IP anonymisiert zur Verfügung gestellt.

Wir sind gespannt auf Ihre Abgaben!