Erste Schritte - Spielobjekte steuern




Im Einstieg hing die Figur noch an losen Variablen. Ab jetzt ist sie ein **Objekt**: der namenlose Held. Er muss in jedem Level alle Coins einsammeln und dabei verschiedene Hindernisse überwinden.

Aufgabe 1 – Die Umgebung kennenlernen

Es ist stets das Ziel des Spiels, mit dem Helden alle Coins im Level einzusammeln. Um den Weg in einem Level zu planen, dürfen Sie dies zunächst mit der Tastatur umsetzen.

a) Testen Sie im **Tastaturmodus** die Tasten ⬆ ⬅ ➡, **Enter**, **C**, **V** und **F** und sammeln Sie alle Coins im Level ein.

(noch nicht bearbeitet)

Tastenbelegung:

Ergänzen Sie, was die Tasten bewirken:

Enter bewirkt: _____

C bewirkt: _____

V bewirkt: _____

F bewirkt: _____

c) Worin unterscheiden sich die Tür und das Tor?

(noch nicht beantwortet)

Aufgabe 2 – Den Helden programmieren

Wir beginnen mit einem einfachen Level, das nun durch Programmieren des Helden gelöst werden soll. Unser Held kennt die Befehle `geh()`, `links()`, `rechts()` und `nimm_coin()`. Damit der Held einen Befehl ausführt, schreiben Sie `held.` davor, z. B. `held.geh()`.

a) Öffnen Sie **Level 1**. Der Coin liegt direkt vor dem Helden. Tippen Sie `held.geh()` und in die nächste Zeile `held.nimm_coin()` von Hand ein und führen Sie das Programm aus.

(noch nicht bearbeitet)

Ab jetzt müssen Sie die Befehle des Helden nicht mehr vollständig selbst tippen: Sie stehen im Level neben der Welt unter „Verfügbare Funktionen“ – ein Klick fügt sie in Ihr Programm ein.

b) Schreiben Sie ein Programm, um **Level 1A** zu lösen.

(noch nicht bearbeitet)

c) Lösen Sie jetzt **Level 2A** als Übung.

(noch nicht bearbeitet)

Aufgabe 3 – Andere Objekte

a) Öffnen Sie **Level 3**. Dort finden Sie neben dem Helden auch noch eine weitere Figur, nämlich den **Knappen**. Er versteht die gleichen Befehle wie der Held, wird jedoch über die Variable `knappe` angesprochen. Tippen Sie die Befehle für den Knappen von Hand ein, damit er seinen Coin holt. Den Helden steuern Sie wie gewohnt über die verfügbaren Funktionen.

(noch nicht bearbeitet)

Die verfügbaren Funktionen wachsen mit

Unter „Verfügbare Funktionen“ steht immer nur, was Sie schon kennen. Sobald Sie etwas Neues gelernt haben, passt sich die Auswahl an: Ab dem nächsten Level können Sie dort auch den `knappe` auswählen – dann hängt ein Klick die Befehle an den Knappen statt an den Helden.

b) Öffnen Sie **Level 3A**. Wieder sind Held und Knappe getrennt. Wählen Sie unter „Verfügbare Funktionen“ aus, wer den Befehl ausführen soll, und sammeln Sie mithilfe des Knappen und des Helden alle Coins ein.

(noch nicht bearbeitet)

c) Öffnen Sie **Level 4A**. Leider scheint das Level unlösbar zu sein, deswegen bedienen wir uns einer kleinen Mogelei. Vielleicht lässt sich das Monster ja über die Variable `monster` ansprechen …

(noch nicht bearbeitet)

Aufgabe 4 – Reflexion

Welche Rückschlüsse können Sie über die Funktionsweise der Methoden `geh()`, `links()` und `rechts()` des Knappen ziehen? Überlegen Sie dazu noch einmal, wie die Bewegung ohne den objektorientierten Ansatz umgesetzt wurde (Heft 0). Welche Voraussetzungen müssen erfüllt sein, damit die Methoden genau das gewünschte Verhalten zeigen und sich z. B. nur der Held bewegt, nicht aber der Knappe?

(noch nicht bearbeitet)

Merke:

Vor dem Punkt steht die _____ des Objekts, hinter dem Punkt die _____. `held.geh()` bewegt also nur den _____, `knappe.geh()` nur den _____.

Aufgabe 5 – Schleifen verwenden

Je größer das Level ist, desto aufwendiger ist auch die Lösung des Problems. Daher möchten wir die Lösung (falls möglich) effizienter gestalten. Dies erreichen wir durch Nutzen von Schleifen, z. B. der `for`-Schleife.

Erinnerung: Die Schleife `for i in range(5):` führt die eingerückten Befehle 5-mal aus.

a) Nutzen Sie eine Schleife, um **Level 5A** effizient zu lösen.

(noch nicht bearbeitet)

b) Nutzen Sie Schleifen, um **Level 6A** effizient zu lösen.

(noch nicht bearbeitet)

Unterprogramme

Sie können jederzeit auch Unterprogramme verwenden, um wiederkehrende Muster nicht mehrfach aufschreiben zu müssen. Zum Beispiel lässt dieses Unterprogramm den Helden `n` Schritte weit gehen:

```
def lauf(n):
    for i in range(n):
        held.geh()

lauf(3)
```

Für Experten: Ein Unterprogramm `lauf(n, figur)` kann jede übergebene Figur (`held`, `knappe` oder `monster`) laufen lassen.

Aufgabe 6 – Weitere Methoden

Held und Knappe besitzen noch die Fähigkeiten `lese_spruch()`, `bediene_tor()` und `sage_spruch()`. Nutzen Sie diese zusätzlich zu Schleifen und Unterprogrammen.

a) Lösen Sie **Level 7A**.

(noch nicht bearbeitet)

b) Lösen Sie **Level 8A**. Vor fast jedem Coin steht ein Tor – ein Fall für eine Schleife.

(noch nicht bearbeitet)

Aufgabe 7 – Fallen und Herzen

In manchen Leveln liegen **Fallen**: Wer hineintritt, verliert ein Leben. **Herzen** heilen ein Leben. Wer kein Leben mehr hat, ist k. o. Wie viele Leben der Held noch hat, steht unter der Spielwelt.

a) Lösen Sie **Level 8B**. Der Held hat nur ein Leben – holen Sie sich vorher das Herz.

(noch nicht bearbeitet)

Außerdem gibt es **Teleporter**: Sie gehören immer paarweise zusammen – wer den einen betritt, steht sofort auf dem anderen. Ein **Pfeilfeld** schiebt den Helden in Pfeilrichtung weiter.

b) Lösen Sie **Level 8C**. Probieren Sie ruhig zuerst aus, was beim Betreten der neuen Felder passiert.

(noch nicht bearbeitet)

Im **Bonuslevel 8X** geht es im Slalom durch zwei Reihen von Fallen. Mit nur einem Leben ist kein Fehltritt erlaubt. Finden Sie das Muster und lösen Sie das Level mit einer Schleife.

Bonuslevel 8X (freiwillig)