
Der Vorkurs ist ein Angebot der (teilweise ehemaligen) Programmierung 2 Tutoren. Der Kurs ist keine offizielle Lehrveranstaltung. Es gibt keine CP, die Teilnahme ist optional, und eine HISPOS-Anmeldung ist weder möglich noch erforderlich. Bei Fragen zum Vorkurs und Programmierung 2 könnt ihr euch gerne an die Dozenten und Tutoren wenden. Wir wünschen euch ein erfolgreiches Semester und freuen uns, euch in Programmierung 2 wieder zu sehen.

Aufgabe 1.0: Laufen

Bringen Sie die Eule zum Ziel! Die erlaubten Befehle sind `move`, `turn left` und `turn right`.

Aufgabe 1.1: Immer geradeaus

Dieses Labyrinth ist sehr einfach, es geht nur geradeaus. Lösen Sie das Labyrinth mit nur einem `move` Befehl! Dazu kann der `jump` Befehl hilfreich sein.

Aufgabe 1.2: Wände erkunden

Auch in diesem Labyrinth geht es zunächst nur geradeaus. Sobald die Eule auf eine Wand gestoßen ist fehlt nur noch ein Schritt nach rechts. Erweitern Sie die Lösung der vorherigen Aufgabe! Hier können die Befehle `explore` und `branch` hilfreich sein.

Aufgabe 1.3: Unterprogramme

In diesem Labyrinth kann das Ziel erreicht werden, indem man geradeaus bis zur nächsten Wand geht, sich dann nach rechts dreht und diese beiden Schritte so lange wiederholt bis das Ziel erreicht wurde. Schreiben Sie zunächst ein Unterprogramm `geradeausBisWand`, indem die Eule geradeaus bis zur nächsten Wand geht. Lösen Sie dann das Labyrinth mit Hilfe von diesem Unterprogramm.

Aufgabe 1.4: Im Kreis

Der Strategie aus der vorherigen Aufgabe führt bei diesem Labyrinth dazu, dass die Eule im Kreis läuft. Ändern Sie die Lösung aus der vorherigen Aufgabe so, dass das aktuelle Feld vor jeder Rechtsdrehung markiert wird. Sobald ein markiertes Feld zum zweiten Mal erreicht wird, soll die Rechtsdrehung durch eine Linksdrehung ersetzt werden.

Aufgabe 1.5: Mit anderen reden und arbeiten

Die Aufgabe ist es nun mit dieser gegebenen Hilfsfunktion ihren Weg aus dem Labyrinth zu finden! Wann immer Sie nichtmehr gerade aus laufen können, rufen Sie das Unterprogramm `guess` auf. Nach der Ausführung gibt `guess` 0 oder 1 als 0. Argument zurück. Wenn Sie die Zahl 0 erhalten müssen Sie links abbiegen. Sollte das Ergebnis 1 sein, müssen Sie nach rechts um zum Ziel zu gelangen. Die Funktionsweise von `guess` ist dabei nicht relevant.

Aufgabe 1.6: Wahlen nach Zahlen

Erstellen und updaten Sie einen Counter, der die Anzahl an Linksdrehung gegen die Anzahl an Rechtsdrehungen aufzählt. Wann immer Sie links **und** rechts abbiegen können, wählen Sie die Richtung die (laut Counter) weniger oft gewählt wurde. Ansonsten gehen Sie gerade aus, oder biegen Sie in die einzig freie Richtung ab. Bei Gleichstand gehen Sie immer zuerst nach rechts. (Die Karte für diese Aufgabe enthält keine Sackgassen.)

Aufgabe 1.7: Bonus: Brute Force

Probleme lassen sich *theoretisch* dadurch lösen, alle Möglichkeiten auszuprobieren. Dieses Verfahren heiSSt Brute Force.

Ziel dieser Aufgabe ist es, das Labyrinth mit einer speziellen Strategie zu lösen, welche in vielen Labyrinthen erfolgreich ist: Prüfen Sie vor jedem Schritt die Markierungen auf dem aktuellen Feld. Sind alle Richtungen markiert, bleiben Sie stehen. Andernfalls drehen Sie die Eule in eine noch nicht markierte Richtung, markieren diese und gehen einen Schritt vorwärts.