

**Dietrich Boles**

# **Hamstroid**

**-**

**Hamster-Simulator für Android-Smartphones**

**Version 1.2**

## **Benutzungshandbuch**

**02.01.2015**

## Inhalt

1	Einleitung.....	5
2	Hamstroid-Versionen.....	7
2.1	Version 1.0 .....	7
2.2	Version 1.1 .....	7
2.3	Version 1.2 .....	7
3	Installation und Starten der Hamstroid-App.....	8
4	Das Hamster-Modell.....	9
5	Hamstroid-Programme .....	11
5.1	Hamster-Befehle.....	11
5.2	Anweisungen und Programme .....	11
5.3	Prozeduren.....	12
5.3.1	Prozedurdefinition .....	12
5.3.2	Prozeduraufruf .....	12
5.3.3	Start-Prozedur.....	12
5.3.4	Verlasse-Anweisung .....	12
5.4	Hamster-Testbefehle .....	12
5.5	Boolesche Ausdrücke.....	13
5.6	Auswahanweisungen.....	14
5.6.1	Bedingte Anweisung .....	14
5.6.2	Alternativanweisung.....	14
5.7	Wiederholungsanweisungen .....	15
5.7.1	Solange-Anweisung .....	15
5.7.2	Wiederhole-Anweisung .....	15
5.7.3	Wiederhole-Bis-Anweisung.....	15
5.7.4	Wiederhole-Anzahl-Anweisung.....	15
5.8	Boolesche Funktionen.....	16
5.8.1	Liefere-Anweisung .....	16
5.8.2	Funktionsdefinition .....	16
5.8.3	Funktionsaufruf .....	16
6	Bedienung der Hamstroid-App .....	18
6.1	Genereller Aufbau .....	18
6.1.1	Ebenen .....	18
6.1.2	Menüleiste.....	19

6.2	Einstellungen-Bildschirm .....	19
6.2.1	Bildschirmausrichtung.....	19
6.2.2	Territoriumsgröße .....	19
6.2.3	Sound .....	19
6.3	Hilfe-Bildschirm.....	20
6.4	Info-Bildschirm.....	20
6.5	Start-Bildschirm .....	20
6.6	Programme-Bildschirm .....	21
6.7	Programm-Bildschirm .....	22
6.7.1	Prozeduren-Bereich .....	23
6.7.2	Funktionen-Bereich.....	23
6.7.3	Territorien-Bereich .....	24
6.8	Prozedur-Bildschirm .....	25
6.8.1	Anweisungsblöcke .....	25
6.8.2	Anweisungsblöcke mit Bedingungen bzw. Zahlenwerten .....	26
6.8.3	Anweisungen mit mehreren Blöcken.....	26
6.8.4	Anweisungen hinzufügen .....	27
6.8.5	Weitere Menü-Einträge .....	27
6.9	Funktion-Bildschirm .....	28
6.10	Bedingung-Bildschirm .....	29
6.10.1	Ausdrucksblöcke.....	29
6.10.2	Komplexe Ausdrücke.....	30
6.10.3	„+“-Symbol in der Menüleiste.....	30
6.11	Anweisungen-Kategorien-Bildschirm.....	31
6.12	Ausdrücke-Kategorien-Bildschirm .....	32
6.13	Territorium-Bildschirm .....	33
6.13.1	Gestaltung eines Territoriums.....	33
6.13.2	Ausführung eines Programms .....	34
6.14	Aufgabenstellung-Bildschirm .....	35
6.15	Aufgabenkategorien-Bildschirm .....	36
6.16	Aufgaben-Bildschirm .....	37
6.17	Aufgabe-Bildschirm .....	38
7	Ausblick .....	39



# 1 Einleitung

Mit dem **Hamster-Modell** wird Programmieranfängern ein einfaches, aber mächtiges Modell zur Verfügung gestellt, mit dessen Hilfe Grundkonzepte der Programmierung auf spielerische Art und Weise erlernt werden können. Programmierer entwickeln so genannte **Hamster-Programme**, mit denen sie virtuelle Hamster durch eine virtuelle Landschaft steuern und bestimmte Aufgaben lösen lassen.

Zum Hamster-Modell existieren drei im Springer-Vieweg-Verlag erschienene Bücher. In dem ersten Buch „Programmieren spielend gelernt mit dem Java-Hamster-Modell“ werden allgemeine Grundlagen der Programmierung erläutert sowie Konzepte der imperativen Programmierung (Anweisungen, Schleifen, Prozeduren, Typen, Variablen, Parameter, Rekursion, ...) eingeführt. Darauf aufbauend behandelt das zweite Buch „Objektorientierte Programmierung spielend gelernt mit dem Java-Hamster-Modell“ alle wichtigen Konzepte der objektorientierten Programmierung (Objekte, Klassen, Vererbung, Polymorphie, Interfaces, Exceptions, Zugriffsrechte, Pakete, ...). Im dritten Band „Parallele Programmierung spielend gelernt mit dem Java-Hamster-Modell“ geht es um die parallele Programmierung mit Java-Threads (Threads, Kommunikation, Synchronisation, Deadlocks, ...). Ergänzende Materialien werden im Internet auf der **Java-Hamster-Modell-Homepage** unter <http://www.java-hamster-modell.de/> bereitgestellt.

Auf der Java-Hamster-Modell-Homepage steht auch der **Hamster-Simulator** kostenlos zur Verfügung; ein Programm, mit dem Hamster-Programme erstellt und ausgeführt werden können. Neben den drei Büchern kommt dem Hamster-Simulator dabei eine ganz wichtige Bedeutung zu, denn Programmieren lernt man nicht durch Lesen. Man muss üben, üben, üben. Und genau dazu dient der Simulator. Als Programmiersprachen unterstützt der Hamster-Simulator dabei in erster Linie Java, aber auch andere Sprachen wie Python, Ruby, Scheme und Prolog.

Der Hamster-Simulator ist auf allen Computern lauffähig, für die eine Java Platform Standard Edition – kurz JavaSE – verfügbar ist. Für Smartphones ist das leider nicht der Fall. Diese Lücke soll nun die sogenannte **Hamstroid-App** – kurz **Hamstroid** - schließen. Hierbei handelt es sich um eine Android-App, die es ermöglicht, mit Android-Smartphones Hamster-Programme zu entwickeln und zu testen. Unterstützt werden Smartphones (und auch Tablets) mit Android ab Version 4.0. Die Programmierung erfolgt dabei nicht textuell, sondern mit Hilfe einer visuellen Programmiersprache, die an die visuelle Sprache der Scratch-Umgebung (siehe <http://scratch.mit.edu/>) oder auch Google-Blockly (<https://developers.google.com/blockly/>) angelehnt ist. Als Vorbild für den generellen Aufbau einer solchen App diente die App Pocket Code (<https://play.google.com/store/apps/details?id=org.catrobat.catroid>).

Hamstroid soll dabei den Hamster-Simulator nicht ersetzen sondern ergänzen. Er ist für Situationen gedacht, wo man gerade ein wenig Zeit und Lust hat, sich mit dem Programmieren lernen zu beschäftigen, aber ein Computer oder Laptop nicht zur Verfügung steht, bspw. im Bus oder Zug oder zwischendurch in einer Pause. Wenn sie also demnächst in solchen Situationen zum Smartphone greifen und spielen oder „rumdaddeln“ wollen, halten Sie vielleicht kurz einen Moment inne: Eigentlich könnte ich ja auch was sinnvollereres tun und ein bisschen „hamstern“.

Sie halten hier in Händen bzw. sehen am Bildschirm das Benutzungshandbuch der Hamstroid-App. Im Prinzip ist diese sehr einfach zu bedienen. Wenn Sie bereits etwas Erfahrung mit Smartphones haben, sollte Ihnen seine Handhabung keine Probleme bereiten. Trotzdem gibt es sicher Situationen, in denen Sie nicht weiterkommen oder in denen Ihnen nicht bewusst ist, welche Möglichkeiten Sie überhaupt haben. Für genau diese Fälle ist dieses Benutzungshandbuch konzipiert. Es dient allerdings nicht als Ersatz für die oben genannten Hamster-Bücher. In den Hamster-Büchern (insbesondere dem ersten Buch) werden die Grundkonzepte der Programmierung beschrieben und sie dienen weiterhin als theoretische Grundlage für das dann praktische Programmieren lernen mit dem Hamster-Simulator und Hamstroid.

## **2 Hamstroid-Versionen**

### **2.1 Version 1.0**

Version 1.0 vom 27.11.2014 ist die erste im Google-Play-Store bereitgestellte Version von Hamstroid.

### **2.2 Version 1.1**

In Version 1.1 vom 03.12.2014 wurde ein Fehler behoben, der zum Speichern eines fehlerhaften Zustands führen konnte.

### **2.3 Version 1.2**

Gegenüber Version 1.1 enthält Version 1.2 der Hamstroid-App folgende Veränderungen:

- Eine Wiederhole-Bis-Anweisung wurde eingeführt (siehe Abschnitt 5.7.3)
- Eine Wiederhole-Anzahl-Anweisung wurde eingeführt (siehe Abschnitt 5.7.4)
- Sound kann an- oder abgestellt werden (siehe Abschnitt 6.2.3)
- Das Look-and-Feel beim Erstellen von Prozeduren, Funktionen und Bedingungen wurde dem Look-and-Feel von Scratch bzw. Google-Blockly angeglichen.
- Im Kontext-Menü von Anweisungsblöcken für Steueranweisungen gibt es einen weiteren Eintrag „Löschen mit Inhalt“ (siehe Abschnitt 6.8.1)
- Bei der Bearbeitung von Prozeduren und Funktionen gibt es drei neue Menü-Einträge: „Anweisungen löschen“, „in Prozedur auslagern“ und „in Funktion auslagern“. Hierüber können Sie mehrere Anweisungen selektieren und dann gleichzeitig löschen oder in eine neue Prozedur bzw. Funktion auslagern (siehe Abschnitt 6.8.5).
- Es wurden ein paar kleinere Fehler behoben.

### **3 Installation und Starten der Hamstroid-App**

Die Hamstroid-App kann aus Google-Play-Store geladen und wie in Android üblich installiert und gestartet werden. Unterstützt werden Smartphones und Tablets mit Android ab Version 4.0. Hamstroid verlangt keine besonderen Rechte und auch keine besonderen Hardware-Eigenschaften. Die Hamstroid-App ist völlig kostenlos und nervt auch nicht mit Werbung.

Beim Beenden der Hamstroid-App wird der aktuelle Zustand, d.h. insbesondere alle Programme, automatisch gespeichert. Programme gehen also niemals verloren, sondern sind bei einem erneuten Starten der App wieder verfügbar.

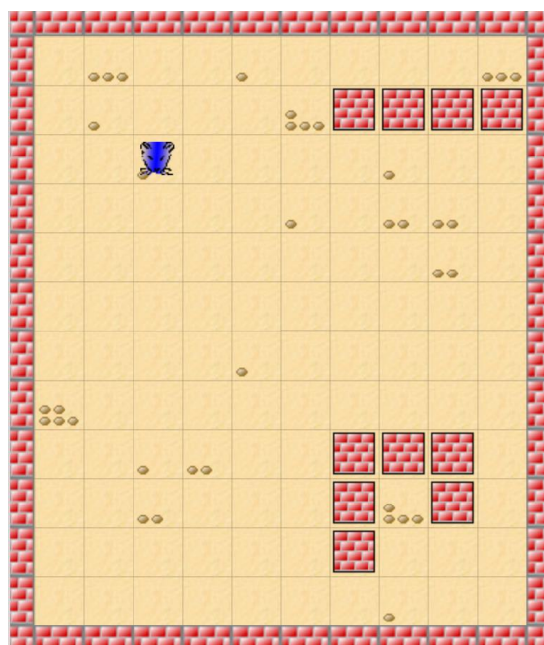


## 4 Das Hamster-Modell

Das Hamster-Modell ist ein didaktisches Modell, bei dem nicht primär das Erlernen einer Programmiersprache, sondern das Erlernen der Programmierung im Vordergrund steht, d.h. das Erlernen grundlegender Programmierkonzepte und das Erlernen des Problemlösungs- bzw. des Programmentwicklungsprozesses. Der Lernprozess wird im Hamster-Modell durch spielerische Elemente unterstützt.

Die Grundidee des Hamster-Modells ist ausgesprochen einfach: Sie als Programmierer müssen einen (virtuellen) Hamster durch eine (virtuelle) Landschaft steuern und ihn gegebene Aufgaben lösen lassen.

Die Welt, in der der Hamster lebt, wird durch eine gekachelte Ebene repräsentiert. Die folgende Abbildung zeigt eine typische Hamster-Landschaft, auch Hamster-Territorium genannt. Das Territorium in der Abbildung besteht dabei aus 12 Reihen und 10 Spalten, d.h. insgesamt 120 Kacheln.



Auf einzelnen Kacheln können ein oder mehrere Körner liegen. Auf den Kacheln können weiterhin auch Mauern stehen, das bedeutet, dass diese Kacheln blockiert sind. Der Hamster kann sie nicht betreten. Es ist nicht möglich, dass sich auf einer Kachel sowohl eine Mauer als auch Körner befinden. Ein Hamster-Territorium ist immer an allen Seiten von Mauern abgeschlossen.

In Hamstroid existiert in jedem Territorium immer genau ein Hamster. Der Hamster steht dabei auf einer der Kacheln des Territoriums. Diese Kachel darf nicht durch eine Mauer blockiert sein, sie kann jedoch Körner enthalten. Der Hamster kann in vier unterschiedlichen Blickrichtungen (Nord, Süd, West, Ost) auf den Kacheln stehen. Je nach Blickrichtung wird der Hamster durch eine entsprechende Ausrichtung des Hamster-Symbols repräsentiert. Körner können sich nicht nur auf einzelnen Kacheln, sondern auch im Maul des Hamsters befinden.

Mit Hilfe bestimmter Befehle kann ein Programmierer den Hamster durch ein gegebenes Territorium steuern. Der Hamster kann dabei von Kachel zu Kachel hüpfen, er kann sich drehen, Körner fressen und Körner wieder ablegen. Sie können sich den Hamster quasi als einen virtuellen Prozessor vorstellen, der im Gegensatz zu realen Prozessoren (zunächst) keine arithmetischen und logischen Operationen ausführen kann, sondern in der Lage ist, mit einem kleinen Grundvorrat an Befehlen ein Territorium zu „erforschen“.

Ihnen als Hamster-Programmierer werden nun bestimmte Aufgaben gestellt, die Sie durch die Steuerung des Hamsters durch ein Territorium zu lösen haben. Diese Aufgaben werden Hamster-Aufgaben und die entstehenden Lösungsprogramme Hamster-Programme genannt. Zusätzlich zu einer Aufgabe werden dabei entweder bestimmte Hamster-Landschaften fest vorgegeben, in denen der Hamster die Aufgabe zu lösen hat, oder die Territorien werden nur noch durch vorgegebene Eigenschaften charakterisiert, d.h. der Hamster wird in der Lage sein, eine bestimmte Aufgabe in unterschiedlichen (aber gleichartigen) Territorien zu lösen.

## 5 Hamstroid-Programme

In diesem Kapitel werden die grundlegenden Programmierkonzepte der Programmierung mit Hamstroid kurz vorgestellt. Ausführlichere Informationen hierzu finden Sie in dem ersten Buch zum Hamster-Modell „Programmieren spielend gelernt mit dem Java-Hamster-Modell“.

### 5.1 Hamster-Befehle

Die Aufgabe eines Hamster-Programmierers besteht darin, den Hamster durch eine Landschaft zu steuern, um dadurch gegebene Hamster-Aufgaben zu lösen. Zur Steuerung des Hamsters müssen ihm Anweisungen in Form von Befehlen gegeben werden. Der Hamster besitzt dabei die Fähigkeit, vier verschiedene Befehle zu verstehen und auszuführen:

- **vor**: Der Hamster hüpft eine Kachel in seiner aktuellen Blickrichtung nach vorn.
- **linksUm**: Der Hamster dreht sich auf der Kachel, auf der er gerade steht, um 90 Grad nach links.
- **nimm**: Der Hamster frisst von der Kachel, auf der er sich gerade befindet, genau ein Korn, d.h. anschließend hat der Hamster ein Korn mehr im Maul und auf der Kachel liegt ein Korn weniger als vorher.
- **gib**: Der Hamster legt auf der Kachel, auf der er sich gerade befindet, genau ein Korn aus seinem Maul ab, d.h. er hat anschließend ein Korn weniger im Maul und auf der Kachel liegt ein Korn mehr als vorher.

### 5.2 Anweisungen und Programme

In imperativen Programmiersprachen werden Verarbeitungsvorschriften durch sogenannte **Anweisungen** ausgedrückt. Anweisungen, die nicht weiter zerlegt werden können, werden elementare Anweisungen genannt. In der Hamster-Sprache sind die vier Hamster-Befehle elementare Anweisungen.

Eine Folge von Anweisungen wird als **Anweisungssequenz** bezeichnet. Die einzelnen Anweisungen einer Anweisungssequenz werden in der angegebenen Reihenfolge hintereinander ausgeführt. Eine Anweisungssequenz gilt selber wieder als (komplexe) Anweisung.

Hamstroid-Programme werden in der sogenannten **Start-Prozedur** formuliert. Beim Start eines Hamstroid-Programms wird die Anweisung bzw. Anweisungssequenz innerhalb der Start-Prozedur ausgeführt. Nach Ausführung der letzten Anweisung im Rumpf der Start-Prozedur ist ein Hamster-Programm beendet.

## 5.3 Prozeduren

Stellen Sie sich **Prozeduren** als eine Möglichkeit vor, auf der Basis vorhandener Befehle neue Befehle zu definieren und aufzurufen.

### 5.3.1 Prozedurdefinition

Durch eine **Prozedurdefinition** wird ein neuer Befehl vereinbart. In der Definition muss zum einen angegeben werden, wie der Befehl heißt (Prozedurname) und zum anderen muss festgelegt werden, was der Hamster tun soll, wenn er den neuen Befehl erhält (Prozedurrumpf). Der Prozedurrumpf besteht dabei aus einer Anweisung oder Anweisungssequenz.

### 5.3.2 Prozeduraufruf

Nach der Definition einer Prozedur kann diese wie die Grundbefehle des Hamsters verwendet werden. Führt man eine Prozedur aus (man nennt dies auch **Prozeduraufruf**), so werden die Anweisungen des entsprechenden Prozedurrumpfes ausgeführt.

Klassisches Beispiel für eine Prozedur ist die Definition einer Prozedur mit dem Namen *rechtsUm*. Der Rumpf der Prozedur enthält dabei eine Anweisungssequenz mit drei Anweisungen *linksUm*. Ruft man die Prozedur *rechtsUm* auf, dreht sich der Hamster dreimal um 90 Grad nach links, was einer Drehung um 90 Grad nach rechts entspricht.

### 5.3.3 Start-Prozedur

Eine besondere Prozedur, die jedes Hamster-Programm besitzt und die auch nicht gelöscht oder umbenannt werden kann, ist die Prozedur mit dem Namen **Start-Prozedur**. Beim Start eines Hamster-Programms werden immer die Anweisungen im Rumpf dieser Prozedur ausgeführt.

### 5.3.4 Verlasse-Anweisung

Eine Prozedur ist beendet, wenn die letzte Anweisung ihres Rumpfes ausgeführt worden ist. Zum (vorzeitigen) Verlassen einer Prozedur gibt es eine weitere Anweisung: die **Verlasse-Anweisung**. Ihre Ausführung bewirkt das unmittelbare Verlassen der Prozedur, d.h. Anweisungen, die sich hinter der verlasse-Anweisung im Prozedurrumpf befinden, werden nicht mehr ausgeführt. Die Ausführung einer Verlasse-Anweisung in der Start-Prozedur führt zum unmittelbaren Programmende.

## 5.4 Hamster-Testbefehle

Wie Sie vielleicht schon festgestellt haben, können bei den Befehlen *vor*, *nimm* und *gib* Probleme auftreten:

- Der Hamster bekommt den Befehl *vor* und die Kachel in Blickrichtung vor ihm ist durch eine Mauer blockiert.

- Der Hamster bekommt den Befehl *nimm* und auf der Kachel, auf der er sich gerade befindet, liegt kein einziges Korn.
- Der Hamster bekommt den Befehl *gib* und er hat kein einziges Korn im Maul.

Bringen Sie den Hamster in diese für ihn unlösbaren Situationen, dann ist der Hamster derart von Ihnen enttäuscht, dass er im Folgenden nicht mehr bereit ist, weitere Befehle auszuführen. Derartige Fehler werden **Laufzeitfehler** genannt. Sie treten während der Ausführung eines Programms auf. Programme, die zu Laufzeitfehlern führen können, sind nicht korrekt!

Zur Vermeidung der Fehler und zur Überprüfung bestimmter Bedingungen kennt der Hamster drei **Testbefehle**. Testbefehle liefern boolesche Werte, also wahr oder falsch:

- **vornFrei**: Liefert den Wert *wahr*, falls sich auf der Kachel in Blickrichtung vor dem Hamster keine Mauer befindet. Ist die Kachel durch eine Mauer blockiert, dann wird der Wert *falsch* geliefert.
- **maulLeer**: Liefert den Wert *falsch*, falls der Hamster ein oder mehrere Körner im Maul hat. Befinden sich keine Körner im Maul des Hamsters, dann wird der Wert *wahr* geliefert.
- **kornDa**: Liefert den Wert *wahr*, falls auf der Kachel, auf der der Hamster gerade steht, ein oder mehrere Körner liegen. Befindet sich kein Korn auf der Kachel, dann wird der Wert *falsch* geliefert.

## 5.5 Boolesche Ausdrücke

**Boolesche Ausdrücke** sind Ausdrücke, die einen Wahrheitswert (wahr oder falsch) liefern. Sie können zur Überprüfung einer Bedingung verwendet werden. Die drei Hamster-Testbefehle stellen die elementaren booleschen Ausdrücke der Hamstroid-Sprache dar. Weiterhin gibt es die sogenannten Literale **wahr** und **falsch**, die den entsprechenden Wahrheitswert liefern.

Zur Verknüpfung von booleschen Ausdrücken und damit zur Bildung komplexer boolescher Ausdrücke existieren drei Operatoren:

- **NICHT**: Der NICHT-Operator (Negation) hat einen booleschen Ausdruck als Operanden. Seine Ausführung bewirkt die Änderung des Wahrheitswertes des Operanden, d.h. liefert der Operand *wahr*, so liefert die Negation *falsch*; liefert der Operand *falsch*, so liefert die Negation *wahr*.
- **UND**: Der UND-Operator (Konjunktion) hat zwei boolesche Ausdrücke als Operanden. Seine Ausführung bewirkt folgende Verknüpfung der beiden Operanden: Er liefert genau dann den Wahrheitswert *wahr*, wenn beide Operanden den Wahrheitswert *wahr* liefern. In allen anderen Fällen wird *falsch* geliefert. Liefert bereits der erste Operand den Wert *falsch*, liefert die Konjunktion direkt *falsch* und der zweite Operand wird gar nicht mehr ausgewertet. Das ist unter Umständen bei der Verwendung von Funktionen (siehe Abschnitt 5.8 (Boolesche Funktionen)) von Bedeutung.

- **ODER:** Der ODER-Operator (Disjunktion) hat zwei boolesche Ausdrücke als Operanden. Seine Ausführung bewirkt folgende Verknüpfung der beiden Operanden: Er liefert genau dann den Wahrheitswert *falsch*, wenn beide Operanden den Wahrheitswert *falsch* liefern. In allen anderen Fällen wird *wahr* geliefert. Liefert bereits der erste Operand den Wert *wahr*, liefert die Disjunktion direkt *wahr* und der zweite Operand wird gar nicht mehr ausgewertet.

## 5.6 Auswahanweisungen

**Auswahanweisungen** sind komplexe Anweisungen, die es erlauben auszudrücken, dass eine bestimmte Anweisung bzw. Anweisungssequenz nur unter einer bestimmten Bedingung ausgeführt werden soll. Hamstroid unterstützt zwei Auswahanweisungen, die bedingte Anweisung und die Alternativanweisung.

### 5.6.1 Bedingte Anweisung

Die **bedingte Anweisung** („falls <Bedingung> tue <Wahr-Anweisung> ende falls“) besteht aus einer Bedingung (ein boolescher Ausdruck) und einer sogenannten Wahr-Anweisung. Die Wahr-Anweisung kann dabei eine elementare Anweisung, eine komplexe Anweisung oder eine Anweisungssequenz sein. Wird eine bedingte Anweisung ausgeführt, wird zunächst die Bedingung überprüft, d.h. der boolesche Ausdruck ausgewertet. Nur in dem Fall, dass er *wahr* liefert, wird anschließend die Wahr-Anweisung ausgeführt.

Mit Hilfe der bedingten Anweisung lässt sich also folgendermaßen der Fehler vermeiden, dass der Hamster vor eine Mauer läuft: „falls vornFrei tue vor ende falls“.

### 5.6.2 Alternativanweisung

Die **Alternativanweisung** („falls <Bedingung> tue <Wahr-Anweisung> sonst <Falsch-Anweisung> ende falls“) besteht aus einer Bedingung (ein boolescher Ausdruck), einer sogenannten Wahr-Anweisung und einer sogenannten Falsch-Anweisung. Die Wahr-Anweisung und die Falsch-Anweisung können dabei eine elementare Anweisung, eine komplexe Anweisung oder eine Anweisungssequenz sein. Wird eine Alternativanweisung ausgeführt, wird zunächst die Bedingung überprüft, d.h. der boolesche Ausdruck ausgewertet. Liefert er *wahr*, wird anschließend die Wahr-Anweisung, nicht aber die Falsch-Anweisung ausgeführt. Liefert er *falsch*, wird anschließend die Falsch-Anweisung, nicht aber die Wahr-Anweisung ausgeführt.

Bei der Ausführung der Alternativanweisung „falls vornFrei tue vor sonst linksUm ende falls“, springt der Hamster also auf die nächste Kachel in Blickrichtung vor ihm, wenn dort keine Mauer platziert ist. Befindet sich dort eine Mauer, dreht er sich um 90 Grad nach links.

## 5.7 Wiederholungsanweisungen

**Wiederholungsanweisungen** sind komplexe Anweisungen, die es erlauben auszudrücken, dass eine bestimmte Anweisung bzw. Anweisungssequenz in Abhängigkeit einer bestimmten Bedingung wiederholt ausgeführt werden soll. Hamstroid unterstützt zwei Wiederholungsanweisungen, die Solange-Anweisung und die Wiederhole-Anweisung.

### 5.7.1 Solange-Anweisung

Die **Solange-Anweisung** („solange <Bedingung> wiederhole <Schleifenanweisung> ende solange“) besteht aus einer Bedingung (ein boolescher Ausdruck) und einer sogenannten Schleifenanweisung. Die Schleifenanweisung kann dabei eine elementare Anweisung, eine komplexe Anweisung oder eine Anweisungssequenz sein. Wird eine Solange-Anweisung ausgeführt, wird zunächst die Bedingung überprüft, d.h. der boolesche Ausdruck ausgewertet. Falls er *falsch* liefert, ist die Solange-Anweisung unmittelbar beendet. Wenn er *wahr* liefert, wird die Schleifenanweisung einmal ausgeführt. Anschließend wird die Bedingung erneut ausgewertet. Falls sie immer noch erfüllt ist, wird die Schleifenanweisung ein weiteres Mal ausgeführt. Dieser Prozess („Überprüfung der Schleifenbedingung und falls diese erfüllt ist, Ausführung der Schleifenanweisung“) wiederholt sich solange, bis irgendwann einmal die Bedingung nicht mehr erfüllt ist.

Mit Hilfe der Solange-Anweisung lässt sich also der Hamster, unabhängig davon, wo er sich gerade befindet, bis in seiner Blickrichtung unmittelbar vor die nächste Mauer steuern: „solange vornFrei wiederhole vor ende solange“.

### 5.7.2 Wiederhole-Anweisung

Die **Wiederhole-Anweisung** („wiederhole <Schleifenanweisung> solange <Bedingung>“) besteht wie die Solange-Anweisung aus einer Bedingung (ein boolescher Ausdruck) und einer Schleifenanweisung. Die Schleifenanweisung kann dabei eine elementare Anweisung, eine komplexe Anweisung oder eine Anweisungssequenz sein. Der Unterschied der Wiederhole-Anweisung zur Solange-Anweisung besteht einzig und allein darin, dass auf jeden Fall einmal die Schleifenanweisung ausgeführt wird. Erst danach kommt es zur erstmaligen Überprüfung der Bedingung und dann gegebenenfalls zur wiederholten Ausführung der Schleifenanweisung, solange die Bedingung erfüllt ist.

### 5.7.3 Wiederhole-Bis-Anweisung

Die **Wiederhole-Bis-Anweisung** („wiederhole <Schleifenanweisung> bis <Bedingung>“) ist analog zur Wiederhole-Anweisung. Der Unterschied liegt einzig und allein darin, dass die Schleifenanweisung nicht solange **wie** sondern solange **bis** die Bedingung erfüllt ist, ausgeführt wird.

### 5.7.4 Wiederhole-Anzahl-Anweisung

Die **Wiederhole-Anzahl-Anweisung** („wiederhole <anzahl>-mal <Schleifenanweisung>“) besteht aus einem Zahlenwert und einer Schleifenanweisung. Die Schleifenanweisung

kann dabei eine elementare Anweisung, eine komplexe Anweisung oder eine Anweisungssequenz sein. Der Zahlenwert definiert die Anzahl an Wiederholungen, d.h. er gibt an, wie oft die Schleifenanweisung ausgeführt werden soll.

## 5.8 Boolesche Funktionen

So wie Prozeduren dazu dienen, neue Befehle einzuführen, dienen **boolesche Funktionen** dazu, neue Testbefehle zu definieren.

### 5.8.1 Liefere-Anweisung

Boolesche Funktionen liefern Wahrheitswerte. Zum Verlassen einer Funktion und zum Liefern eines Wahrheitswertes dient die **Liefere-Anweisung**. Sie besitzt eine Bedingung, d.h. booleschen Ausdruck. Wird in einem Funktionsrumpf eine Liefere-Anweisung ausgeführt, so wird die Funktion unmittelbar verlassen und der Wahrheitswert geliefert, den die Auswertung der Bedingung ergibt.

### 5.8.2 Funktionsdefinition

Durch eine **Funktionsdefinition** wird ein neuer Testbefehl vereinbart. Wie bei einer Prozedurdefinition muss auch bei der Definition einer Funktion zum einen angegeben werden, wie der Testbefehl heißt (Funktionsname) und zum anderen muss festgelegt werden, was der Hamster tun soll, wenn er den neuen Testbefehl erhält (Funktionsrumpf). Der Funktionsrumpf besteht dabei aus einer Anweisung oder Anweisungssequenz. Diese sollte in jedem möglichen Durchlauf eine Liefere-Anweisung besitzen.

### 5.8.3 Funktionsaufruf

Nach der Definition einer Funktion kann diese wie die Testbefehle des Hamsters zur Überprüfung von Bedingungen verwendet werden. Führt man eine Funktion aus (man nennt dies auch **Funktionsaufruf**), so werden die Anweisungen des entsprechenden Prozedurrumpfes ausgeführt. Wird dabei eine Liefere-Anweisung ausgeführt, wird der Funktionsrumpf sofort verlassen und der gelieferte Wert als Wahrheitswert des Funktionsaufrufes genutzt.

Klassisches Beispiel für eine Funktion ist die Definition einer Funktion mit dem Namen *linksFrei*, die überprüft, ob sich links vom Hamster eine Mauer befindet. Die Funktion bedient sich dabei einer definierten Prozedur *rechtsUm*:

```
linksFrei:
    linksUm
    falls vornFrei tue
        rechtsUm
        liefere wahr
    sonst
        rechtsUm
        liefere falsch
    ende falls
```



Ruft man die Prozedur *linksFrei* auf, dreht sich der Hamster zunächst nach links. Dann überprüft er ob sich dort eine Mauer befindet und bevor er den entsprechenden Wert liefert, begibt er sich zurück in seine Ausgangsposition.

In folgendem Beispielprogramm sucht der Hamster also links von sich eine Nische und begibt sich dort hinein, sobald er eine findet:

Start-Prozedur:

```
    solange vornFrei UND NOT linksFrei wiederhole
        vor
    ende solange
    falls linksFrei tue
        linksUm
        vor
    ende falls
```

Wird bei der Ausführung einer Funktion keine Liefere-Anweisung ausgeführt, so wird die Funktion mit Ausführung der letzten Anweisung beendet und liefert *wahr*.

## 6 Bedienung der Hamstroid-App

In diesem Kapitel wird die Bedienung der Hamstroid-App ausführlich beschrieben. Es wird dabei vorausgesetzt, dass Sie Grundkenntnisse zur Bedienung eines Android-Smartphones besitzen.

### 6.1 Genereller Aufbau

Die Hamstroid-App besteht aus einer Menge an einzelnen Bildschirmseiten (man spricht in Android auch von Activities). Jede Bildschirmseite besitzt ganz oben eine Menüleiste (Toolbar). Darunter befindet sich der Inhaltsbereich.

#### 6.1.1 Ebenen

Die Bildschirmseiten sind miteinander verknüpft und folgendermaßen in Ebenen angeordnet:

Start-Bildschirm

    Programme-Bildschirm

        Programm-Bildschirm

            Prozedur-Bildschirm

                Anweisungen-Kategorien-Bildschirm

                Bedingung-Bildschirm

                    Ausdrücke-Kategorien-Bildschirm

        Funktion-Bildschirm

            Anweisungen-Kategorien-Bildschirm

            Bedingung-Bildschirm

                Ausdrücke-Kategorien-Bildschirm

    Territorium-Bildschirm

    Aufgabenstellung-Bildschirm

Aufgabenkategorien-Bildschirm

    Aufgaben-Bildschirm

        Aufgabe-Bildschirm

            Aufgabenstellung-Bildschirm

### 6.1.2 Menüleiste

Über die Menüleiste einer Bildschirmseite kann man wichtige allgemeine und auch Bildschirmseiten-bezogene Funktionalitäten abrufen. Ganz links befindet sich (außer auf dem Start-Bildschirm) ein „<“-Zeichen sowie das Hamstroid-Icon. Beim Klick auf dieses Zeichen erfolgt in der Regel ein Wechsel zum Bildschirm auf der nächst-oberen Ebene.

Ganz rechts in der Menüleiste befindet sich ein Zeichen mit drei übereinander stehenden Punkten. Hier hinter verbirgt sich ein Menü. Beim Klick auf das Zeichen wird das Menü geöffnet. Es enthält neben Seiten-spezifischen Funktionalitäten immer folgende Einträge:

- Einstellungen: Beim Klick auf diesen Eintrag öffnet sich der Einstellungen-Bildschirm.
- Hilfe: Beim Klick auf diesen Eintrag öffnet sich der Hilfe-Bildschirm.
- Über Hamstroid: Beim Klick auf diesen Eintrag öffnet sich der Info-Bildschirm.

Je nach Bildschirm erscheinen weitere Einträge in der Menüleiste. Deren Bedeutung wird jeweils bei der Beschreibung der einzelnen Bildschirme erläutert.

## 6.2 Einstellungen-Bildschirm

Der Einstellungen-Bildschirm enthält drei Einträge namens *Bildschirmausrichtung*, *Territoriumsgröße* und *Sound*.

### 6.2.1 Bildschirmausrichtung

Nach dem Klick auf den Eintrag *Bildschirmausrichtung* können Sie in einer Dialogbox die Bildschirmausrichtung festlegen:

- Immer Landscape: Hamstroid wird immer in der waagerechten Ausrichtung angezeigt.
- Immer Portrait: Hamstroid wird immer in der senkrechten Ausrichtung angezeigt.
- Automatischer Wechsel: Je nachdem wie Sie ihr Smartphone halten, wird Hamstroid in der waagerechten oder senkrechten Ausrichtung angezeigt.

### 6.2.2 Territoriumsgröße

Nach dem Klick auf den Eintrag *Territoriumsgröße* können Sie in einer Dialogbox die Anzahl an Reihen und Spalten festlegen, die neu erzeugte Territorien besitzen sollen. Standardmäßig sind hier 12 Reihen und 10 Spalten voreingestellt.

### 6.2.3 Sound

Sie können durch Anklicken des Schalter-Buttons einstellen, ob Sound an- oder ausgestellt werden soll. Standardmäßig ist der Sound angestellt.

## 6.3 Hilfe-Bildschirm

Auf dem Hilfe-Bildschirm wird dieses Benutzungshandbuch angezeigt.

## 6.4 Info-Bildschirm

Auf dem Info-Bildschirm werden allgemeine Informationen zur Hamstroid-App angezeigt.

## 6.5 Start-Bildschirm

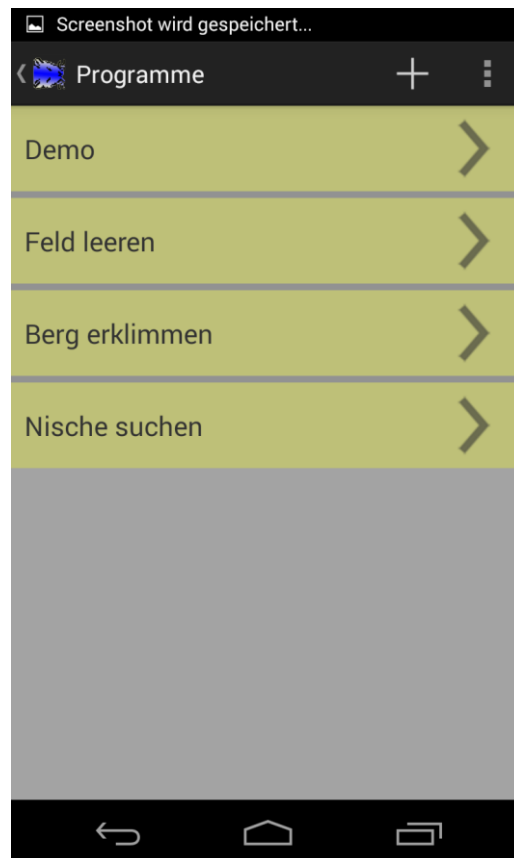


Nach dem Starten der Hamstroid-App öffnet sich der Start-Bildschirm. Er enthält folgende Einträge:

- **Fortsetzen** (<Programmname>): Beim Klick auf diesen Eintrag wird zum Programm-Bildschirm des Programms gewechselt, dass Sie zuletzt bearbeitet haben. Beim erstmaligen Start von Hamstroid ist das ein Demonstrationsprogramm namens *Demo*.
- **Neu**: Beim Klick auf diesen Eintrag können Sie ein neues Programm erzeugen. Es erscheint eine Dialogbox, in der Sie den Namen des neuen Programms eingeben müssen. Programmnamen müssen eindeutig sein, d.h. der eingegebene Name darf nicht bereits an ein anderes Programm vergeben sein. Anschließend erfolgt direkt ein Wechsel zum Programm-Bildschirm des neuen Programms.

- **Programme:** Beim Klick auf diesen Eintrag gelangen Sie zum Programme-Bildschirm.
- **Aufgaben:** Ein Klick auf diesen Eintrag führt Sie zum Aufgabenkategorien-Bildschirm.
- **Hilfe:** Beim Klick auf diesen Eintrag gelangen Sie zum Hilfe-Bildschirm.

## 6.6 Programme-Bildschirm



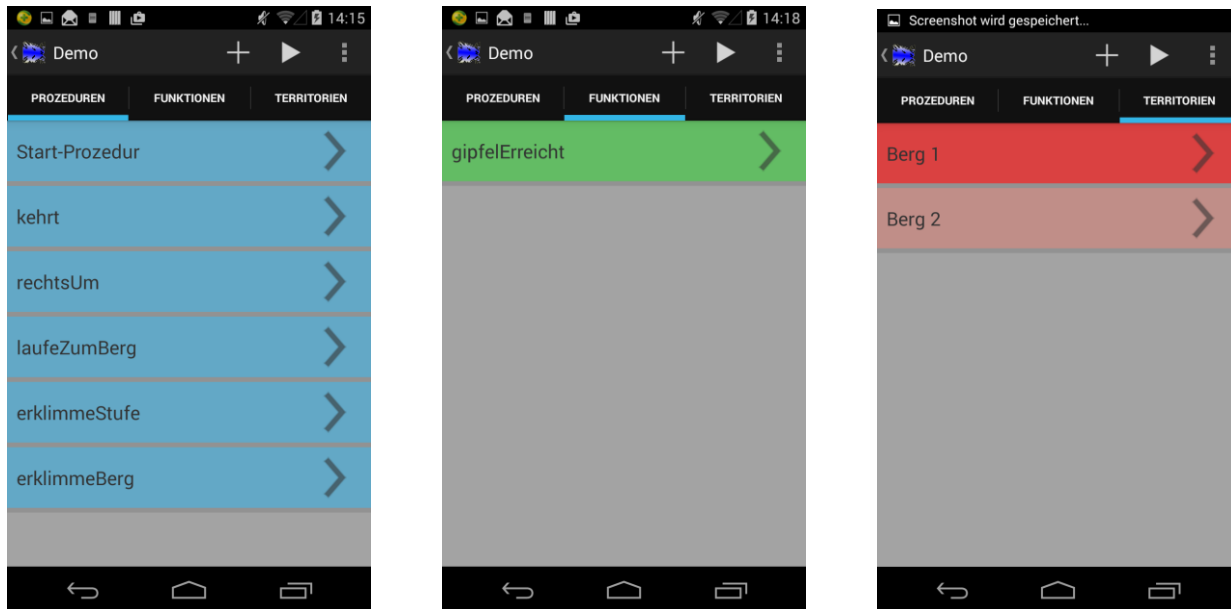
Prinzipiell können Sie beliebig viele Programme in Hamstroid erstellen und verwalten. Der Programme-Bildschirm enthält im Inhaltsbereich für alle Programme je einen Eintrag, der durch den Namen des jeweiligen Programms gekennzeichnet ist. Durch Klick auf einen Eintrag wechseln Sie zum Programm-Bildschirm des entsprechenden Programms.

Wenn Sie länger auf einen Programm-Eintrag klicken, erscheint eine Dialogbox mit dem Namen des angeklickten Programms. Die Dialogbox enthält folgende Einträge:

- **Bearbeiten:** Durch Klick auf diesen Eintrag wechseln Sie zum Programm-Bildschirm für dieses Programm.
- **Umbenennen:** Bei Klick auf diesen Eintrag können Sie den Namen des Programms ändern. Programmnamen müssen eindeutig sein.
- **Löschen:** Bei Klick auf diesen Eintrag erscheint ein Löschen-Dialog, über den Sie das Programm unwiderruflich löschen können.

- **Kopieren:** Bei Klick auf diesen Eintrag können Sie das Programm kopieren. Sie werden aufgefordert, einen Namen für das kopierte Programm einzugeben. Programmnamen müssen eindeutig sein.

## 6.7 Programm-Bildschirm



In der Menüleiste des Programme-Bildschirms erscheint neben dem Hamstroid-Icon der Name des aktuell bearbeiteten Programms.

Unter der Menüleiste erscheinen drei Tabs, die drei verschiedene Bereiche eines Programms repräsentieren

- **Prozeduren:** Beim Klick auf diesen Tab erscheint die Liste der aktuell definierten Prozeduren des Programms.
- **Funktionen:** Beim Klick auf diesen Tab erscheint die Liste der aktuell definierten Funktionen des Programms.
- **Territorien:** Beim Klick auf diesen Tab erscheint die Liste der aktuell definierten Territorien für das Programm.

Ein Tab- bzw. Bereichswechsel ist alternativ auch mit der Wischgeste möglich. Welcher Bereich aktuell selektiert ist, erkennt man an dem blauen Balken unterhalb des entsprechenden Tabs.

Befindet sich das Smartphone im Landscape-Modus werden die Tabs übrigens durch ein Listen-Menü ersetzt. Durch Klick auf einen Listeneintrag wechselt man hier in den entsprechenden Bereich.

Weiterhin befindet sich in der Menüleiste ein „+“-Symbol und ein Pfeil. Je nachdem welcher Tab aktuell selektiert ist, kann man durch Klick auf das „+“-Symbol eine neue Prozedur oder Funktion definieren oder ein neues Territorium erzeugen. Es erscheint eine Dialogbox, in der man den Namen für die neue Prozedur oder Funktion oder das neue

Territorium angeben muss. Namen für Prozeduren und Funktionen sowie für Territorien müssen eindeutig sein. Anschließend wird direkt zum entsprechenden Prozedur-Bildschirm, Funktion-Bildschirm bzw. Territorium-Bildschirm gewechselt.

Das Pfeil-Symbol repräsentiert den Programmstart. Bei Klick auf das Pfeil-Symbol wird das aktuelle Programm im aktuell selektierten Territorium gestartet.

Das Menü der Menüleiste enthält weiterhin einen Eintrag „Aufgabenstellung“. Durch Klick hierauf gelangen Sie zum Aufgabenstellung-Bildschirm, über den Sie eine Aufgabenstellung zum aktuellen Programm anschauen, eingeben oder ändern können.

### 6.7.1 Prozeduren-Bereich

Der Prozeduren-Bereich enthält für alle aktuell im Programm definierten Prozeduren je einen Eintrag, der durch den Namen der jeweiligen Prozedur gekennzeichnet ist. Durch Klick auf einen Eintrag wechseln Sie zum Prozedur-Bildschirm der entsprechenden Prozedur.

Wenn Sie länger auf einen Prozedur-Eintrag klicken, erscheint eine Dialogbox mit dem Namen der angeklickten Prozedur. Die Dialogbox enthält folgende Einträge:

- **Bearbeiten:** Durch Klick auf diesen Eintrag wechseln Sie zum Prozedur-Bildschirm für diese Prozedur.
- **Umbenennen:** Bei Klick auf diesen Eintrag können Sie den Namen der Prozedur ändern. Dabei gilt: Innerhalb eines Programms sind Namen für Prozeduren und Funktionen eindeutig, dürfen also nur einmal verwendet werden. Die Start-Prozedur kann nicht umbenannt werden.
- **Löschen:** Bei Klick auf diesen Eintrag erscheint ein Löschen-Dialog, über den Sie die Prozedur unwiderruflich löschen können. Ein Löschen ist dabei nur dann möglich, wenn die Prozedur nicht verwendet, d.h. nirgendwo aufgerufen wird. Die Start-Prozedur kann nicht gelöscht werden.
- **Kopieren:** Bei Klick auf diesen Eintrag können Sie die Prozedur kopieren. Sie werden aufgefordert, einen Namen für die kopierte Prozedur einzugeben.

### 6.7.2 Funktionen-Bereich

Der Funktionen-Bereich enthält für alle aktuell im Programm definierten Funktionen je einen Eintrag, der durch den Namen der jeweiligen Funktion gekennzeichnet ist. Durch Klick auf einen Eintrag wechseln Sie zum Funktion-Bildschirm der entsprechenden Funktion.

Wenn Sie länger auf einen Funktion-Eintrag klicken, erscheint eine Dialogbox mit dem Namen der angeklickten Funktion. Die Dialogbox enthält folgende Einträge:

- **Bearbeiten:** Durch Klick auf diesen Eintrag wechseln Sie zum Funktion-Bildschirm für diese Funktion.

- **Umbenennen:** Bei Klick auf diesen Eintrag können Sie den Namen der Funktion ändern. Dabei gilt: Innerhalb eines Programms sind Namen für Prozeduren und Funktionen eindeutig, dürfen also nur einmal verwendet werden.
- **Löschen:** Bei Klick auf diesen Eintrag erscheint ein Löschen-Dialog, über den Sie die Funktion unwiderruflich löschen können. Ein Löschen ist dabei nur dann möglich, wenn die Funktion nicht verwendet, d.h. nirgendwo aufgerufen wird.
- **Kopieren:** Bei Klick auf diesen Eintrag können Sie die Funktion kopieren. Sie werden aufgefordert, einen Namen für die kopierte Funktion einzugeben.

### 6.7.3 Territorien-Bereich

Der Territorien-Bereich enthält für alle aktuell im Programm erzeugten Territorien je einen Eintrag, der durch den Namen des jeweiligen Territoriums gekennzeichnet ist. Durch Klick auf einen Eintrag wechseln Sie zum Territorium-Bildschirm des entsprechenden Territoriums.

Wenn Sie länger auf einen Territorium-Eintrag klicken, erscheint eine Dialogbox mit dem Namen des angeklickten Territoriums. Die Dialogbox enthält folgende Einträge:

- **Bearbeiten:** Durch Klick auf diesen Eintrag wechseln Sie zum Territorium-Bildschirm für dieses Territorium.
- **Umbenennen:** Bei Klick auf diesen Eintrag können Sie den Namen des Territoriums ändern. Dabei gilt: Innerhalb eines Programms sind Namen für Territorien eindeutig, dürfen also nur einmal verwendet werden.
- **Löschen:** Bei Klick auf diesen Eintrag erscheint ein Löschen-Dialog, über den Sie das Territorium unwiderruflich löschen können. Dabei gilt: Es muss pro Programm immer ein Territorium existieren, d.h. existiert nur ein Territorium, kann dieses nicht gelöscht werden.
- **Kopieren:** Bei Klick auf diesen Eintrag können Sie das Territorium kopieren. Sie werden aufgefordert, einen Namen für das kopierte Territorium einzugeben.

Einer der Einträge der Territorien-Liste ist dunkelrot markiert. Hierbei handelt es sich um das **aktuelle Territorium**. Es ist das einzige bzw. das zuletzt bearbeitete Territorium. Beim Start eines Programms wird dieses immer im aktuellen Territorium ausgeführt.



## 6.8 Prozedur-Bildschirm



In der Menüleiste des Prozedur-Bildschirms erscheint neben dem Hamstroid-Icon der Name des aktuell bearbeiteten Programms.

Weiterhin befindet sich in der Menüleiste ein „+“-Symbol und ein Pfeil. Das „+“-Symbol wird weiter unten in Abschnitt 6.8.4 erläutert. Das Pfeil-Symbol repräsentiert den Programmstart. Bei Klick auf das Pfeil-Symbol wird das aktuelle Programm im aktuell selektierten Territorium gestartet.

Unterhalb der Menüleiste befindet sich eine weitere Leiste, in der mittig der Name der sich aktuell in Bearbeitung befindenden Prozedur angezeigt wird. Enthält ein Programm mehrere Prozeduren werden rechts und links die Namen der weiteren Prozeduren angeführt. Mit der Wischgeste kann direkt zwischen den einzelnen Prozeduren gewechselt werden.

Im Inhaltsbereich werden die Anweisungen der bearbeiteten Prozedur in Form von Blöcken angezeigt. Sollte eine Prozedur ausgeführt werden, werden diese Anweisungen in der Reihenfolge von oben nach unten ausgeführt.

### 6.8.1 Anweisungsblöcke

Ein Anweisungsblock enthält im linken Bereich den Namen der Anweisung und im rechten Bereich ein Verschiebe-Symbol, das durch vier übereinander liegende Streifen gekennzeichnet ist. Drückt man auf das Verschiebe-Symbol, kann man den

Anweisungsblock innerhalb der aktuellen Prozedur an eine andere Stelle verschieben. Dazu muss der Finger auf dem Bildschirm verbleiben. Stellen, an die der Block verschoben werden kann, werden beim Verschiebevorgang rot markiert. Hebt man in diesem Fall den Finger, wird der Block an der alten Stelle entfernt und an der neuen Stelle eingefügt. Lässt man den Finger an einer nicht erlaubten Stelle los, verbleibt der Anweisungsblock an der alten Stelle.

Wenn Sie länger auf einen Anweisungsblock klicken, erscheint eine Dialogbox mit dem Namen der angeklickten Anweisung. Die Dialogbox enthält folgende Einträge:

- **Löschen:** Bei Klick auf diesen Eintrag wird die Anweisung unmittelbar gelöscht.
- **Löschen mit Inhalt:** Dieser Eintrag erscheint nur bei Steuerungsanweisungen. Bei Klick auf diesen Eintrag werden die Anweisung selber sowie alle inneren Anweisungen gelöscht.
- **Prozedur bearbeiten:** Dieser Eintrag erscheint nur, wenn es sich bei der Anweisung um einen Prozeduraufruf handelt. Durch Klick auf den Eintrag kann man direkt in den Prozedurbereich der entsprechenden Prozedur wechseln und diese bearbeiten.
- **Bedingung bearbeiten:** Dieser Eintrag erscheint nur, wenn es sich bei der Anweisung um eine Anweisung mit einer Bedingung handelt. Klickt man auf den Eintrag erfolgt ein Wechsel zum Bedingung-Bildschirm, wo man die Bedingung bearbeiten kann.

### 6.8.2 Anweisungsblöcke mit Bedingungen bzw. Zahlenwerten

Bei Anweisungen, die Bedingungen bzw. boolesche Ausdrücke enthalten, werden diese im entsprechenden Anweisungsblock hinter dem Namen der Anweisung in einem grünen Kästchen angezeigt. Durch Klick auf das grüne Kästchen kann man zum Bedingung-Bildschirm wechseln und die Bedingung bearbeiten.

Bei der Wiederhole-Anzahl-Anweisung wird die Anzahl der Wiederholungen in einem roten Kästchen hinter dem Namen der Anweisung angezeigt. Beim Klicken auf das Kästchen erscheint eine Dialogbox, über die man den Zahlenwert ändern kann.

### 6.8.3 Anweisungen mit mehreren Blöcken

Anweisungen, die eine Auswahl- oder Wiederholungsanweisung repräsentieren, bestehen aus mehreren Blöcken, insbesondere einem Anfangs- und einem Endblock. Diese sind prinzipiell unabhängig voneinander verschiebbar. Allerdings ist eine Verschiebung nur in gewissen Bereichen möglich, so dass keine Überlappungen zwischen mehreren dieser komplexen Anweisungen entstehen. Beim Löschen eines Blockes werden automatisch auch die anderen Blöcke der Anweisung gelöscht.

Welche Anweisungen zur Wahr-, Falsch oder Schleifenanweisungssequenz gehören, wird durch Einrückung der entsprechenden Blöcke nach rechts zwischen den Blöcken einer komplexen Anweisung angezeigt.

#### 6.8.4 Anweisungen hinzufügen

Durch Drücken des „+“-Symbols in der Menüleiste kann man neue Anweisungen zu einer Prozedur hinzufügen. Hierbei wird in den Anweisungen-Kategorien-Bildschirm gewechselt. Klickt man dort eine Anweisung an, erfolgt automatisch wieder ein Wechsel in den Prozedur-Bildschirm. Der Anweisungsblock der neuen Anweisung erscheint im oberen Bereich des Bildschirms und liegt oberhalb der anderen Anweisungsblöcke. Diesen Block kann man nun an die gewünschte Position verschieben. Ein Klick außerhalb des Anweisungsblockes der neuen Anweisung oder eine Verschiebung des Blockes an eine ungültige Stelle führen zum Löschen der Anweisung.

#### 6.8.5 Weitere Menü-Einträge

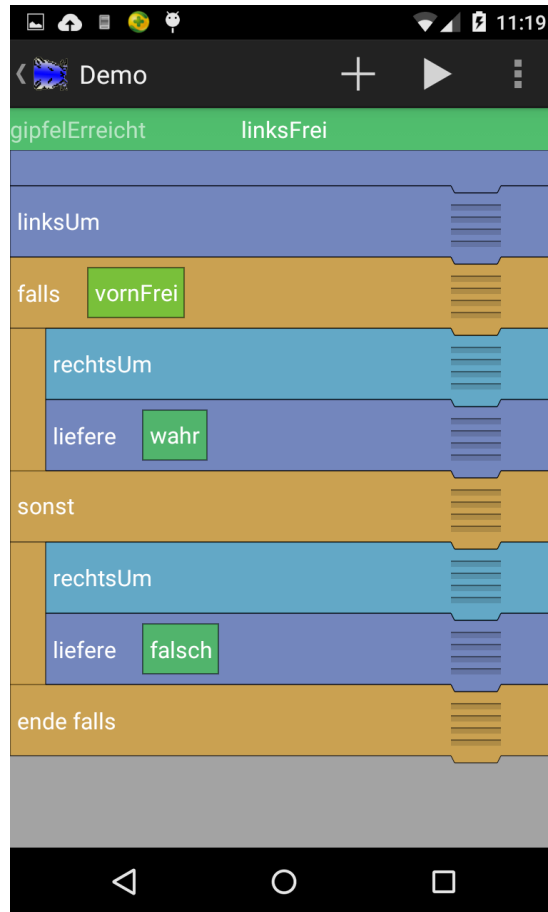
Über das Menü der Menüleiste lassen sich weitere Funktionalitäten abrufen: „Anweisungen löschen“, „in Prozedur auslagern“ und „in Funktion auslagern“. Wenn Sie einen dieser Menü-Einträge anklicken, gelangen Sie in den Selektionsmodus für Anweisungen innerhalb der Prozedur. Durch Anklicken einer Anweisung wird diese selektiert (grüner Punkt rechts) oder beim erneuten Anklicken wieder deselektiert. Verlassen können Sie den Selektionsmodus durch Anklicken des Häkchen-Symbols links in der Menüleiste.

**Anweisungen löschen:** Beim Anklicken des Buttons „Löschen“ in der Menüleiste werden alle aktuell selektierten Anweisungen gelöscht.

**In Prozedur auslagern:** Beim Anklicken des Buttons „in Prozedur auslagern“ können Sie die aktuell selektierten Anweisungen in eine neue Prozedur auslagern. Dazu erscheint eine Dialogbox, in der Sie den Namen der neuen Prozedur eingeben müssen. Nach Klicken auf „Ok“ wird die neue Prozedur mit den entsprechenden Anweisungen erzeugt. Diese werden in der aktuellen Prozedur gelöscht und durch einen Prozeduraufruf der neuen Prozedur ersetzt.

**In Funktion auslagern:** Beim Anklicken des Buttons „in Funktion auslagern“ können Sie die aktuell selektierten Anweisungen in eine neue Funktion auslagern. Dazu erscheint eine Dialogbox, in der Sie den Namen der neuen Funktion eingeben müssen. Nach Klicken auf „Ok“ wird die neue Funktion mit den entsprechenden Anweisungen erzeugt. Diese werden in der aktuellen Prozedur gelöscht.

## 6.9 Funktion-Bildschirm



Der Funktion-Bildschirm ist völlig analog zum Prozedur-Bildschirm aufgebaut. Die Bearbeitung von Funktionen im Funktion-Bildschirm erfolgt analog zur Bearbeitung von Prozeduren im Prozedur-Bildschirm.

## 6.10 Bedingung-Bildschirm



Klickt man im Prozedur- oder Funktion-Bildschirm auf eine Bedingung eines Anweisungsblockes, wird zum Bedingung-Bildschirm gewechselt.

### 6.10.1 Ausdrucksblöcke

Im Inhaltsbereich des Bedingung-Bildschirms wird die zu bearbeitende Bedingung in Form von einem oder mehreren Blöcken angezeigt. Ersteres ist bei einfachen, letzteres bei komplexen Ausdrücken, die mit Hilfe der booleschen Operatoren gebildet werden, der Fall. Bei der Auswertung einer Bedingung werden diese Blöcke in der Reihenfolge von oben nach unten ausgewertet.

Ein Ausdrucksblock enthält im linken Bereich den Namen des booleschen Ausdrucks und im rechten Bereich ein Verschiebe-Symbol, das durch vier übereinanderliegende Streifen gekennzeichnet ist. Drückt man auf das Verschiebe-Symbol, kann man den Ausdrucksblock an eine andere Stelle verschieben. Dazu muss der Finger auf dem Bildschirm verbleiben. Stellen, an die der Block verschoben werden kann, werden beim Verschiebevorgang rot markiert. Hebt man in diesem Fall den Finger, wird der verschobene Block an der alten Stelle gelöscht und durch einen Wahr-Block ersetzt und der verschobene Block ersetzt anschließend den bisherigen Block an der neuen Stelle. Dieser wird gelöscht.

Wenn Sie länger auf einen Ausdrucksblock klicken, erscheint eine Dialogbox mit dem Namen des angeklickten Ausdrucks. Die Dialogbox enthält folgende Einträge:

- **Löschen:** Bei Klick auf diesen Eintrag wird der Ausdruck unmittelbar gelöscht. Ein gelöschter Ausdruck wird immer durch einen Wahr-Ausdruck ersetzt.
- **Bedingung ändern:** Klickt man auf den Eintrag erfolgt ein Wechsel zum Ausdrücke-Kategorien-Bildschirm. Klickt man dort einen Ausdruck an, erfolgt automatisch wieder ein Wechsel in den Bedingung-Bildschirm. Der Ausdruck, für den die Dialogbox geöffnet wurde, ist dabei durch den Ausdruck, der im Ausdrücke-Kategorien-Bildschirm ausgewählt wurde, ersetzt worden.
- **Funktion bearbeiten:** Dieser Eintrag erscheint nur, wenn es sich bei dem Ausdruck um einen Funktionsaufruf handelt. Durch Klick auf den Eintrag kann man direkt in den Funktionsbereich der entsprechenden Funktion wechseln und diese bearbeiten.

## 6.10.2 Komplexe Ausdrücke

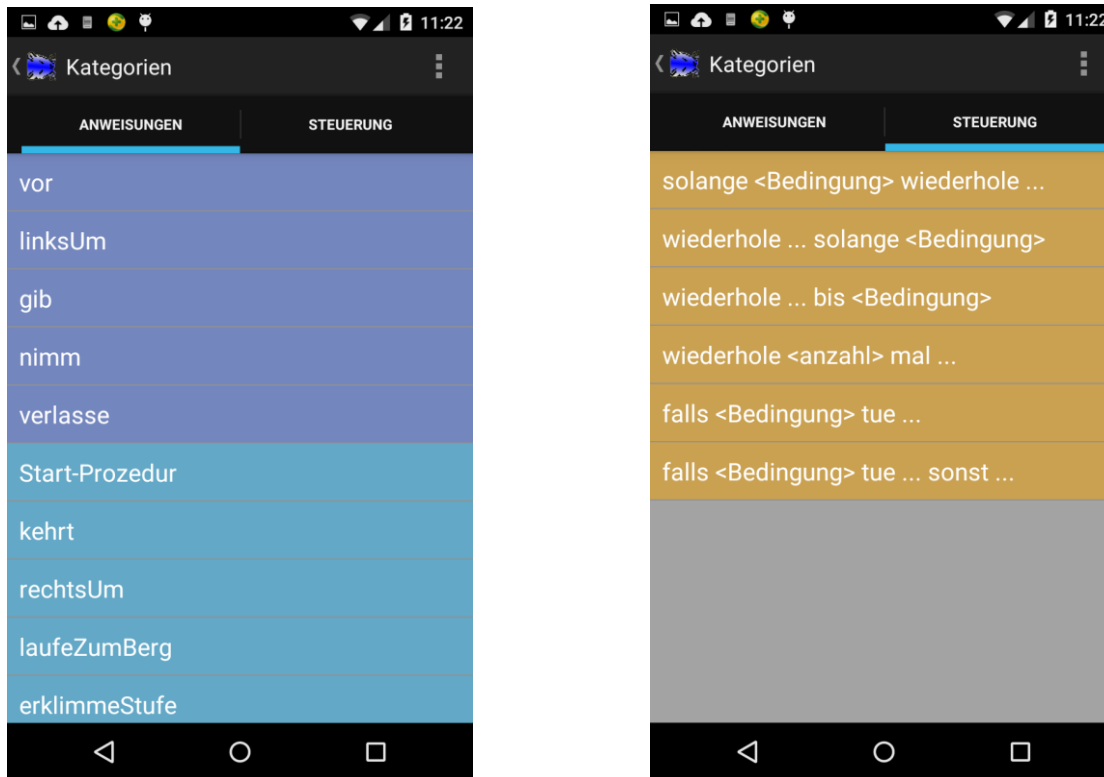
Komplexe Ausdrücke werden im Fall der Negation von zwei und im Fall der Konjunktion und Disjunktion durch drei Blöcke repräsentiert. Die mit „nicht“, „und“ bzw. „oder“ benannten Blöcke repräsentieren dabei den entsprechenden Operator, die anderen Blöcke die Operanden. Beim Löschen eines der Operator-Blöcke und beim Verschieben eines anderen Blockes auf einen Operator-Block werden auch die Operanden-Blöcke mit gelöscht. Wird ein Operator-Block verschoben, werden entsprechend auch die Operanden-Blöcke mit verschoben.

Komplexe Ausdrücke lassen sich schachteln. An der Einrückung der Blöcke nach rechts lässt sich die Priorität der Auswertung der Operanden erkennen, vergleichbar mit der Klammersetzung.

## 6.10.3 „+“-Symbol in der Menüleiste

Das „+“-Symbol in der Menüleiste bietet eine Alternative zum Ändern von Bedingungen. Klickt man auf das „+“-Symbol erfolgt ein Wechsel zum Ausdrücke-Kategorien-Bildschirm. Klickt man dort einen Ausdruck an, erfolgt automatisch wieder ein Wechsel in den Bedingung-Bildschirm. Der Ausdrucksblock des ausgewählten Ausdrucks erscheint im oberen Bereich des Bildschirms und liegt oberhalb der anderen Ausdrucksblöcke. Diesen Block kann man nun an die gewünschte Position verschieben, wo er den bisherigen Ausdruck ersetzt. Ein Klick außerhalb des Ausdrucksblockes des neuen Ausdrucks oder eine Verschiebung des Blockes an eine ungültige Stelle führt zum Löschen der Ausdrucksblockes.

## 6.11 Anweisungen-Kategorien-Bildschirm



In den Anweisungen-Kategorien-Bildschirm gelangt man, wenn man im Prozedur-Bildschirm oder Funktion-Bildschirm auf das „+“-Symbol in der Menüleiste klickt. Im Anweisungen-Kategorien-Bildschirm wählt man dann die Anweisung aus, die neu in die Prozedur bzw. Funktion eingefügt werden soll. Hierzu klickt man einfach auf die entsprechende Anweisung in der entsprechenden Liste des Inhaltsbereiches. Anschließend wird automatisch wieder zum Prozedur- bzw. Funktion-Bildschirm gewechselt.

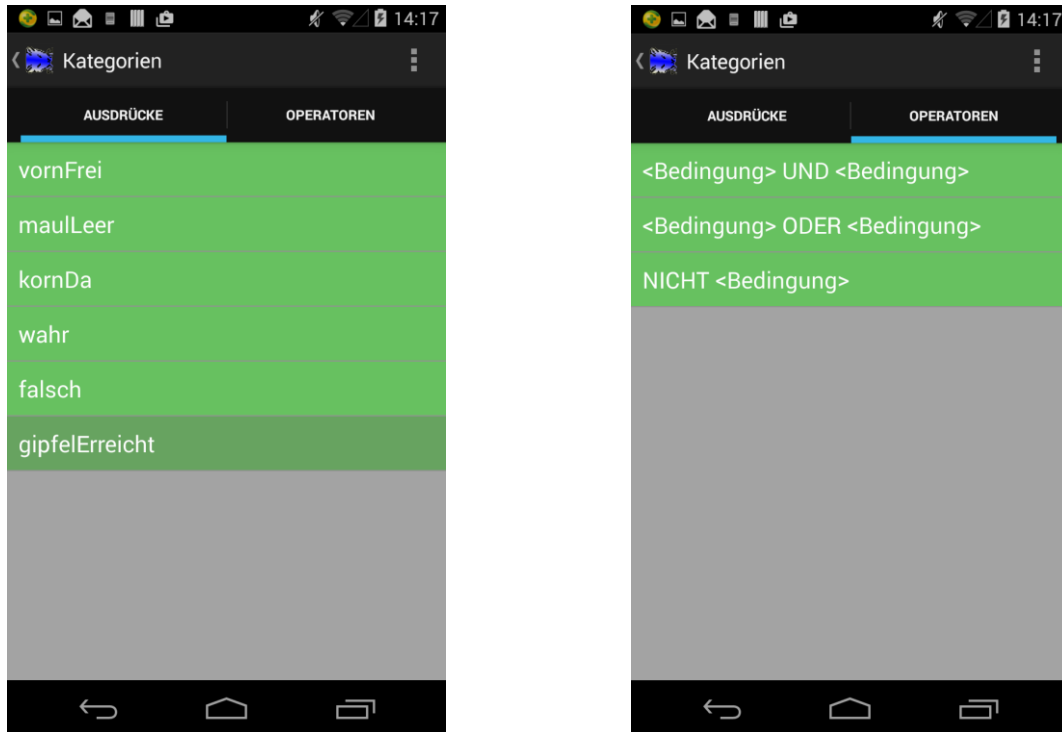
Unter der Menüleiste erscheinen zwei Tabs, die zwei verschiedene Bereiche repräsentieren:

- **Anweisungen:** Beim Klick auf diesen Tab erscheint die Liste der einfachen Anweisungen. Das sind insbesondere die Hamster-Grundbefehle sowie die Prozeduren.
- **Steuerung:** Beim Klick auf diesen Tab erscheint die Liste der komplexen Anweisungen. Das sind die zwei Auswahl- und die vier Wiederholungsanweisungen.

Ein Tab- bzw. Bereichswechsel ist alternativ auch mit der Wischgeste möglich. Welcher Bereich aktuell selektiert ist, erkennt man an dem blauen Balken unterhalb des entsprechenden Tabs.

Befindet sich das Smartphone im Landscape-Modus werden die Tabs übrigens durch ein Listen-Menü ersetzt. Durch Klick auf einen Listeneintrag wechselt man hier in den entsprechenden Bereich.

## 6.12 Ausdrücke-Kategorien-Bildschirm



In den Ausdrücke-Kategorien-Bildschirm gelangt man, wenn man im Bedingung-Bildschirm eine Bedingung ändern möchte, bspw. wenn auf das „+“-Symbol in der Menüleiste klickt. Im Ausdrücke-Kategorien-Bildschirm wählt man dann den neuen Ausdruck aus. Hierzu klickt man einfach auf den entsprechenden Ausdruck in der entsprechenden Liste des Inhaltsbereiches. Anschließend wird automatisch wieder zum Bedingung-Bildschirm gewechselt.

Unter der Menüleiste erscheinen zwei Tabs, die zwei verschiedene Bereiche repräsentieren:

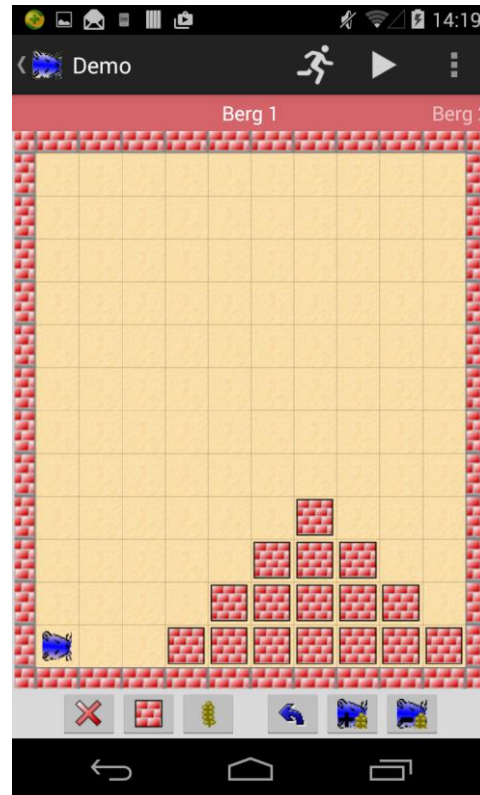
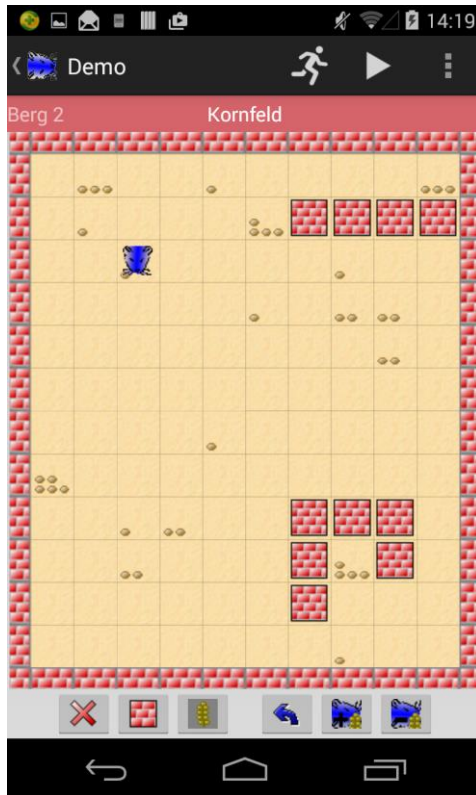
- **Ausdrücke:** Beim Klick auf diesen Tab erscheint die Liste der einfachen Ausdrücke. Das sind die Hamster-Testbefehle sowie die Funktionen.
- **Operatoren:** Beim Klick auf diesen Tab erscheint die Liste der booleschen Operatoren.

Ein Tab- bzw. Bereichswechsel ist alternativ auch mit der Wischgeste möglich. Welcher Bereich aktuell selektiert ist, erkennt man an dem blauen Balken unterhalb des entsprechenden Tabs.

Befindet sich das Smartphone im Landscape-Modus werden die Tabs übrigens durch ein Listen-Menü ersetzt. Durch Klick auf einen Listeneintrag wechselt man hier in den entsprechenden Bereich.



## 6.13 Territorium-Bildschirm



Im Territorium-Bildschirm können ein Territorium gestaltet und ein Programm ausgeführt werden. Unterhalb der Menüleiste befindet sich eine weitere Leiste, in der mittig der Name des sich aktuell in Bearbeitung befindenden Territoriums angezeigt wird. Gibt es zu einem Programm mehrere Territorien, werden rechts und links die Namen der weiteren Territorien angeführt. Mit der Wischgeste kann direkt zwischen den einzelnen Territorien gewechselt werden.

### 6.13.1 Gestaltung eines Territoriums

Standardmäßig haben Territorien in Hamstroid immer eine Größe von 12 Reihen und 10 Spalten, d.h. insgesamt 120 Kacheln. Diese Zahlen können im Einstellungen-Bildschirm verändert werden.

Auf einer der Kacheln befindet sich der Hamster. **Umpositionieren** können Sie den Hamster, in dem Sie ihn Anklicken und bei gedrücktem Finger an die gewünschte Stelle verschieben.

Die **Blickrichtung des Hamsters ändern** Sie, in dem Sie den Button mit dem Pfeil unterhalb des Territoriums anklicken. Bei jedem Klick dreht sich der Hamster um 90 Grad nach links.

Durch Klick auf einen der beiden Buttons rechts vom Umdrehen-Button können Sie die **Anzahl an Körnern im Maul des Hamsters** jeweils um 1 erhöhen oder verringern. Um zu

erfahren, wie viele Körner der Hamster gerade im Maul hat, klicken Sie ihn einfach an. Dann erscheint eine entsprechende Meldung.

Die drei linken Buttons unterhalb des Territoriums dienen zur Gestaltung der Kacheln. Durch Klick auf einen der Buttons aktivieren Sie diesen. Bei erneutem Klick oder bei Klick auf einen der anderen beiden Buttons deaktivieren Sie ihn wieder:

- Wenn der Button mit dem **Körnersymbol** aktiviert ist und sie auf eine Kachel im Territorium klicken, die nicht durch eine Mauer blockiert ist, wird ein zusätzliches Korn auf der Kachel platziert. Theoretisch können Sie beliebig viele Körner auf jeder Kachel platzieren, angezeigt werden durch entsprechende Symbole aber maximal 12.
- Wenn der Button mit dem **Mauersymbol** aktiviert ist und sie auf eine Kachel im Territorium klicken, auf der nicht der Hamster steht, wird auf dieser Kachel eine Mauer platziert. Eventuell vorhandene Körner werden gelöscht.
- Wenn der Button mit dem **Kreuzsymbol** aktiviert ist und sie auf eine Kachel im Territorium klicken, werden eventuell vorhandene Körner oder Mauern von dieser Kachel entfernt.

Im Menü der Menüleiste existiert im Territorium-Bildschirm ein weiterer Eintrag „**Größe ändern**“. Beim Anklicken erscheint eine Dialogbox, über die sie die Anzahl an Reihen und Spalten des Territoriums ändern können. Dabei gilt: Wird das Territorium größer, wird es nach unten bzw. rechts mit leeren Kacheln aufgefüllt. Ist das Territorium kleiner, werden unten bzw. rechts Kacheln entfernt. Unter Umständen wird der Hamster umpositioniert.

### 6.13.2 Ausführung eines Programms

Zur Ausführung eines Hamster-Programms klicken Sie auf das **Start-Symbol** (Pfeil) in der Menüleiste. Es wird durch ein Stopp-Symbol (Kästchen) ersetzt. Das Programm startet unmittelbar mit der Ausführung der Start-Prozedur des Programms. Durch Klicken auf das **Stopp-Symbol** können Sie die Ausführung eines Programms vorzeitig abbrechen. Bei Beendigung eines Programms erscheint automatisch wieder das Start-Symbol in der Menüleiste.

Beim Auftreten eines Laufzeitfehlers, bspw. wenn der Hamster gegen eine Mauer läuft, erscheint eine entsprechende Meldung und das Programm wird unmittelbar beendet. Laufende Programme werden auch beim Bildschirmwechsel bzw. beim Wechseln der Territorien mittels der Wischgeste gestoppt.

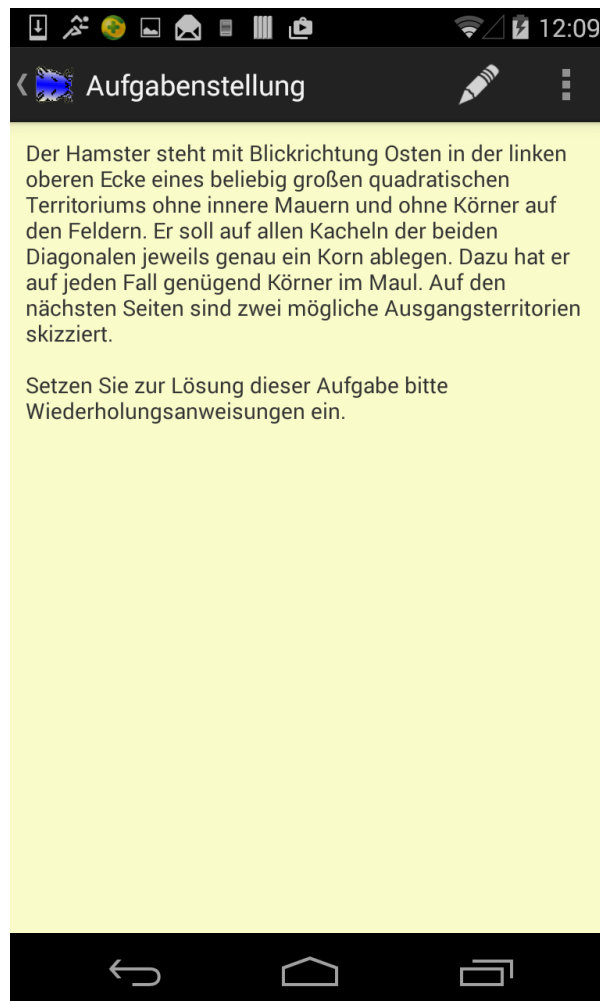
In der Menüleiste des Territorium-Bildschirms existiert weiterhin ein **Läufersymbol**. Hiermit lässt sich die Geschwindigkeit der Ausführung von Programmen regeln. Beim Anklicken des Symbols erscheint ein Schieberegler. Je weiter Sie diesen nach links schieben, umso langsamer wird der Hamster. Je weiter Sie ihn nach rechts schieben, umso schneller flitzt der Hamster durchs Territorium. Ist der Schieberegler sichtbar, können Sie ihn durch Anklicken des Hamstroid-Icons links in der Menüleiste wieder ausblenden.

In der Menüleiste bzw. im Menü der Menüleiste existiert im Territorium-Bildschirm ein weiteres Symbol (kreisförmiger Pfeil) bzw. ein weiterer Eintrag „**Zurücksetzen**“. Beim

Anklicken dieses Symbols bzw. Eintrags wird das Territorium in den Zustand zurückversetzt, der vor dem letzten Start des Hamster-Programms gültig war.

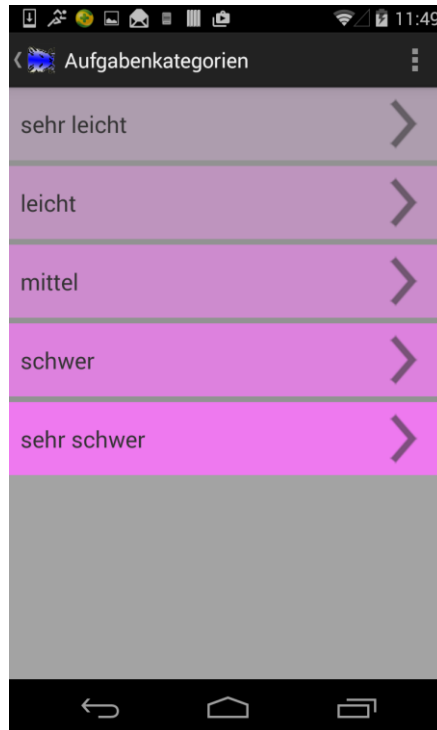
Weiterhin existiert im Menü der Menüleiste ein Eintrag „Startzustand speichern“ bzw. alternativ „Startzustand freigeben“. Klicken Sie auf „**Startzustand speichern**“, wird der aktuelle Zustand des Territoriums gespeichert und das Hamster-Programm immer in dem entsprechenden Zustand gestartet. Klicken Sie auf „**Startzustand freigeben**“ wird der gespeicherte Zustand wieder gelöscht.

## 6.14 Aufgabenstellung-Bildschirm



Im Menü der Menüleiste des Programm-, Prozedur-, Funktion- und Territorium-Bildschirms finden Sie den Eintrag „Aufgabenstellung“. Durch Anklicken dieses Eintrags gelangen Sie zum Aufgabenstellung-Bildschirm. Existiert zu dem aktuellen Programm bereits eine Aufgabenstellung wird diese angezeigt. Durch Klick auf das „Stift“-Symbol bzw. den Eintrag „bearbeiten“ in der Menüleiste öffnet sich ein Eingabefeld, über das Sie eine Aufgabenstellung eintippen oder ändern können.

## 6.15 Aufgabenkategorien-Bildschirm



Eigentlich ist Hamstroid – genauso wie der Hamster-Simulator – dafür gedacht, dass Sie Hamster-Aufgaben lösen, die Sie in den Hamster-Büchern gefunden haben, die Ihnen bspw. Ihr Lehrer gestellt hat oder die sie sich selbst ausgedacht haben. Da Hamstroid jedoch gerade für Situationen konzipiert ist, in denen entsprechende Unterlagen nicht zur Verfügung stehen, sind vorgegebene Aufgaben in Hamstroid integriert. Die Aufgaben sind dabei in 5 Kategorien unterteilt:

- **Sehr leicht:** Diese Kategorie enthält Aufgaben, zu deren Lösung lediglich die 4 Grundbefehle des Hamsters benötigt werden.
- **Leicht:** Aufgaben dieser Kategorie sind dazu gedacht, die Definition und den Einsatz von Prozeduren zu erlernen.
- **Mittel:** Diese Kategorie enthält Basis-Aufgaben zum Erlernen des Einsatzes von Auswahlanweisungen und Wiederholungsanweisungen.
- **Schwer:** Dieser Kategorie zugeordnet sind komplexere Aufgaben zum Erlernen des Einsatzes von Auswahlanweisungen und Wiederholungsanweisungen, Prozeduren und Funktionen. Mit diesen Aufgaben können Sie insbesondere den Programmwurf mit Hilfe des Prinzips der prozeduralen Zerlegung einüben.
- **Sehr schwer:** In dieser Kategorie finden Sie knifflige Aufgaben, bei denen Sie zunächst einen Lösungsansatz finden müssen, um das Problem überhaupt lösen zu können. Für die Lösung einiger der Aufgaben dieser Kategorie ist auch der Einsatz rekursiver Prozeduren und Funktionen notwendig.

Wenn Sie auf einen Eintrag des Aufgabenkategorien-Bildschirms klicken, gelangen Sie zum entsprechenden Aufgaben-Bildschirm.

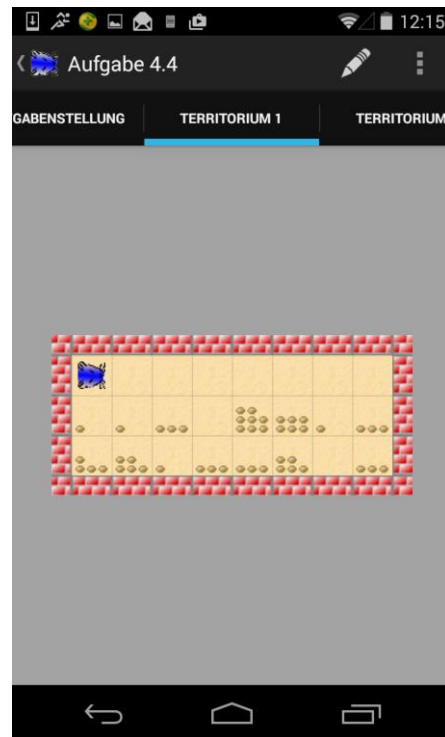
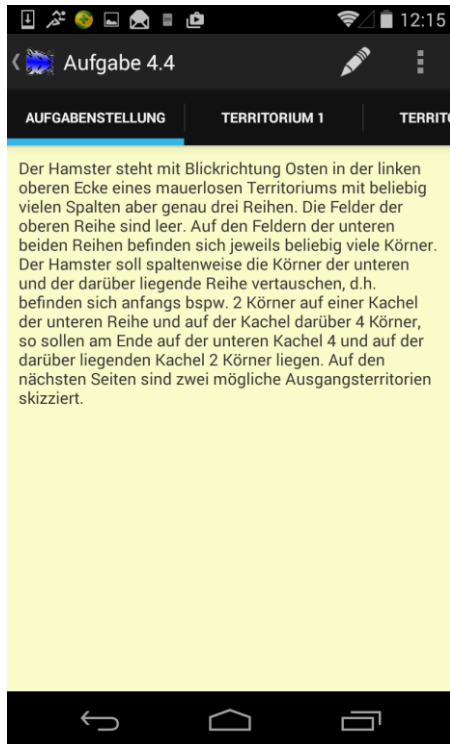
## 6.16 Aufgaben-Bildschirm



In der aktuellen Version von Hamstroid enthält jede Aufgabenkategorie fünf Aufgaben. Durch Anklicken des entsprechenden Eintrags auf dem Aufgaben-Bildschirm gelangen Sie zum Aufgabe-Bildschirm der entsprechenden Aufgabe.

Einträge auf dem Aufgaben-Bildschirm sind grün markiert, wenn Sie zum Lösen der Aufgabe bereits ein Programm angelegt haben. Einträge von Aufgaben ohne bisherige Lösungsversuche sind rot hinterlegt.

## 6.17 Aufgabe-Bildschirm



In der Menüleiste des Aufgabe-Bildschirms erscheint neben dem Hamstroid-Icon der Name der aktuell ausgewählten Aufgabe.

Unter der Menüleiste erscheinen mindestens zwei Tabs, die verschiedene Bereiche einer Aufgabe repräsentieren:

- Aufgabenstellung: Durch Klick auf diesen Tab gelangen Sie zur Aufgabenstellung.
- Territorien: Für jedes Territorium der Aufgabe existiert ein entsprechender Tab. Durch Klick hierauf erscheint das entsprechende Territorium.

Ein Tab- bzw. Bereichswchsel ist alternativ auch mit der Wischgeste möglich. Welcher Bereich aktuell selektiert ist, erkennt man an dem blauen Balken unterhalb des entsprechenden Tabs.

Befindet sich das Smartphone im Landscape-Modus werden die Tabs übrigens durch ein Listen-Menü ersetzt. Durch Klick auf einen Listeneintrag wechselt man hier in den entsprechenden Bereich.

In der Menüleiste des Aufgabe-Bildschirms bzw. im Menü der Menüleiste finden Sie ein „Stift“-Symbol bzw. den Eintrag „bearbeiten“. Wollen Sie eine Aufgabe lösen, klicken Sie bitte hierauf. Wenn Sie dies für die entsprechende Aufgabe zum ersten Mal tun, erscheint eine Dialogbox, in der Sie zunächst den Namen des anschließend neu erzeugten Programms eingeben können. Ansonsten gelangen Sie sofort zum Programm-Bildschirm des entsprechenden Lösungsprogramms. Das Programm enthält übrigens automatisch die Aufgabenstellung als auch die Territorien der Aufgabe.

## 7 Ausblick

Die Hamstroid-App existiert aktuell in der Version 1.2. Es sind weitere Versionen geplant, die insbesondere zusätzliche Funktionalitäten in die App integrieren werden:

- Runterladen vorgefertigter Programme von der Website des Java-Hamster-Modells.
- Austauschen von Programmen zwischen mehreren Android-Smartphones mittels Bluetooth bzw. NFC.
- Generieren von Java-Programmen und verschicken dieser per Email, so dass sie direkt in den Hamster-Simulator geladen werden können.

Wenn Sie weitere Wünsche oder Ideen haben, lassen Sie mich dies gerne wissen. Sie erreichen mich per Email an die Adresse [dietrich@boles.de](mailto:dietrich@boles.de). Aber nun erstmal viel Spaß bei der Hamster-Programmierung mit Hamstroid!