

# **Robot Karol blockbasiert - eine Umsetzung des bekannten Tools in Snap! Version 1.1**

Eine Integration von Robot Karol von Uli Freiburger in die graphische  
Programmierungsumgebung Snap!

Verfasser: Franziska Rieger  
E-Mail: [snap@prosystem24.de](mailto:snap@prosystem24.de)

# 1 Vorwort

Robot Karol von Uli Freiburger ist ein Tool, in dem eine kleine LEGO-Figur textuell programmiert wird und sich durch eine Welt bewegt. Dieses Tool wird insbesondere in bayerischen Gymnasien in der Jahrgangsstufe 7 zur Vermittlung der Grundkenntnisse über Algorithmik eingesetzt.

Sowohl die textuelle als auch die graphische Programmierung haben Vor- und Nachteile. Um die Vorteile beider Varianten nutzen zu können, empfiehlt sich im Laufe des Schuljahres ein Wechsel der Oberflächen.

Um einen schnellen Wechsel von Robot Karol auf eine graphische Oberfläche zu ermöglichen wurde das Beispiel und die Welt von Karol in das Tool Snap! integriert. Die Anweisungen und die Funktionsweise des Karol in Snap! ist analog zur textuellen Programmierung. Alle Abbildungen basieren auf dem Programm Robot Karol von Uli Freiburger.

Dieses Tool ist im Rahmen einer Seminararbeit entstanden. Es bietet noch Erweiterungsmöglichkeiten und soll als Basis gesehen werden. Hinweise, Tipps und Ideen werden gerne entgegengenommen.

Im Folgenden werden die Blöcke für die Lehrkraft sowie die Blöcke, mit denen die SuS programmieren vorgestellt. Um den Nutzen von Karol in Snap! aufzuzeigen werden einige Beispielaufgaben besprochen. Den Aufbau des Programmes sowie die Hintergrundprogrammierung werden im pdf vorgestellt.

## 2 Blöcke Lehrkraft

Einige der Blöcke sind nur von der Lehrkraft und nicht von den SuS zu verwenden. Diese Blöcke bauen die Welt auf und müssen von den SuS vor der eigentlichen Programmierung ausgeführt werden. Es empfiehlt sich diese Blöcke in beispielsweise dem Block *Start* zu kapseln. Dann müssen die SuS diesen Block als Black-Box ausführen und kommen mit den Lehrer-Blöcken nicht in Berührung.

Zum Festlegen der Größe der Welt gibt es zwei Blöcke (siehe Abbildung 1).

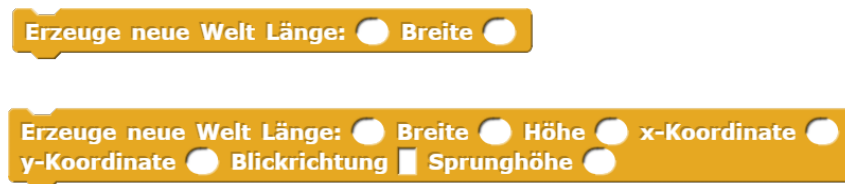


Abbildung 1: Blöcke zum Erzeugen der Welt

Beim minimalen Block wird lediglich die Länge und Breite der Welt festgelegt. Der Roboter wird dann auf die Position (1/1) mit Blickrichtung Süden gesetzt. Die Sprunghöhe beträgt 1. Beim erweiterten Block können diese Parameter gesetzt werden.

Nach dem Festlegen dieser Parameter können Ziegel in der Welt verteilt werden. Hierbei kann die Position, die Farbe und die Anzahl der Ziegel gewählt werden (siehe Abbildung 2).



Abbildung 2: Block zum Verteilen der Ziegel in der Welt

## 3 Blöcke Schüler

Die wichtigsten Anweisungen und Bedingungen von Robot Karol sind auch in Snap! umgesetzt. Die Anweisungen sind unter der Kategorie *Bewegung* zu finden. Die Bedingungen unter *Fühlen*.

Die Anweisungen *Schritt*, *LinksDrehen* und *RechtsDrehen* bewegen den Karol über die Welt. Um einen Ziegel hinzulegen bzw. aufzuheben werden die Anweisungen [ ] *Hinlegen* und *Aufheben* verwendet. Beim Hinlegen kann die Farbe des Ziegels gewählt werden (rot, blau, grün, gelb). Die Anweisungen sind in Abbildung 3



Abbildung 3: Anweisungen zur Programmierung des Karol

Die umgesetzten Bedingungen sind *IstWand*, *NichtIstWand*, *IstZiegel* und *NichtIstZiegel*. Um die Blickrichtung des Karol abzufragen können die Bedingungen *IstNorden*, *IstSüden*, *IstOsten* und *IstWesten* verwendet werden (siehe Abbildung 4).



Abbildung 4: Bedingungen zur Programmierung des Karol

## **4 Beispielaufgaben**

### **4.1 Aufgabe mit Snap! stellen**

Wird Robot Karol mit Snap! im Unterricht eingesetzt, so umfasst diese Phase mehrere Schritte.

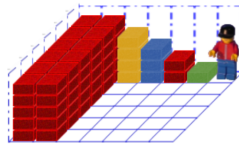
1. Die Lehrkraft formuliert eine Aufgabenstellung und überlegt sich die Startposition von Karol und den Aufbau der Welt.
2. Die Lehrkraft öffnet die Vorlage von Robot Karol in Snap! und formuliert gemäß der Aufgabenstellung den Start-Block. Dieser umfasst zunächst die den Aufbau der Welt und die Startposition von Karol. Anschließend werden ggf. Ziegel in der Welt verteilt.
3. Das Projekt wird entweder heruntergeladen oder online im Account der Lehrkraft als neues Projekt gespeichert.
4. Die Lehrkraft teilt das gespeicherte Projekt und kopiert den Link oder stellt den SuS die heruntergeladene Datei zur Verfügung.
5. Die SuS öffnen den Link des gespeicherten Projektes bzw. öffnen Snap! und laden die Datei der Lehrkraft hoch.
6. Als Abgabe bietet sich ein Bild aller programmierten Blöcke an.

## 4.2 Beispielaufgabe Methode

### 4.2.1 Aufgabenstellung

#### Aufgabe 1:

Programmiere eine Methode Treppe, die eine 1-Ziegel breite Treppe baut (siehe Abbildung).



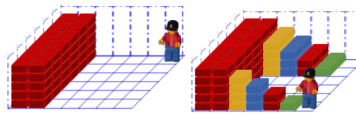
Beispiel für die 1-Ziegel breite Treppe

#### Hinweis:

Eine Methode wird in Snap! durch einen neuen Block repräsentiert. Erstelle einen neuen Block in der Kategorie Bewegung und programmiere in diesem den Bau deine Treppe.

#### Aufgabe 2:

Nutze die Methode aus Aufgabe 2 um ein Treppenhaus zu bauen. Die Treppen in den ersten Stock sind allerdings unterschiedlich breit.



Beispiel für das Treppenhaus

### 4.2.2 Startblock

Bei dieser Aufgabe ist die Startposition des Karols verändert (also nicht bei (1/1)) und es werden Ziegel in der Welt verteilt.



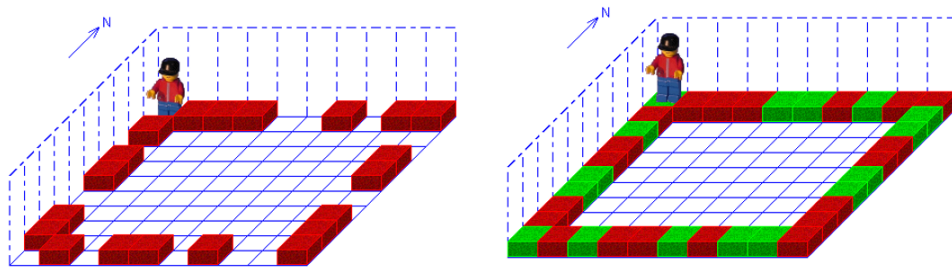
Start-Block mit Ziegel

## 4.3 Beispielaufgabe bedingte Anweisung

### 4.3.1 Aufgabenstellung

#### Aufgabe 1:

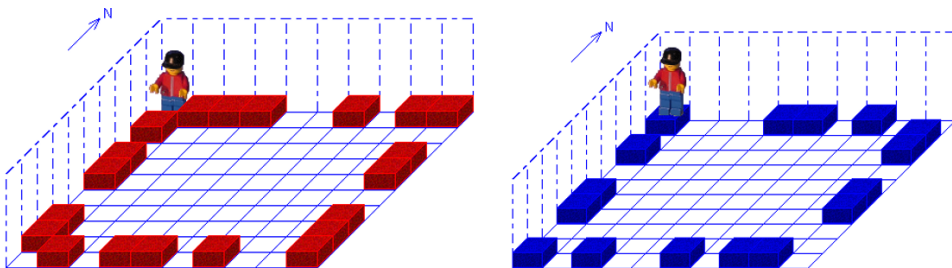
- Schreibe ein Programm, sodass der Roboter einmal den Rand der Welt (Größe: 6x6) abläuft. Nutze falls möglich die Wiederhole-Struktur.
- Ändere dein Programm aus (a) so ab, dass er an jeder Stelle ohne Ziegel einen farbigen Ziegel hinlegt.



Beispiel für das Hinlegen der Ziegel am Rand der Welt

#### Aufgabe 2:

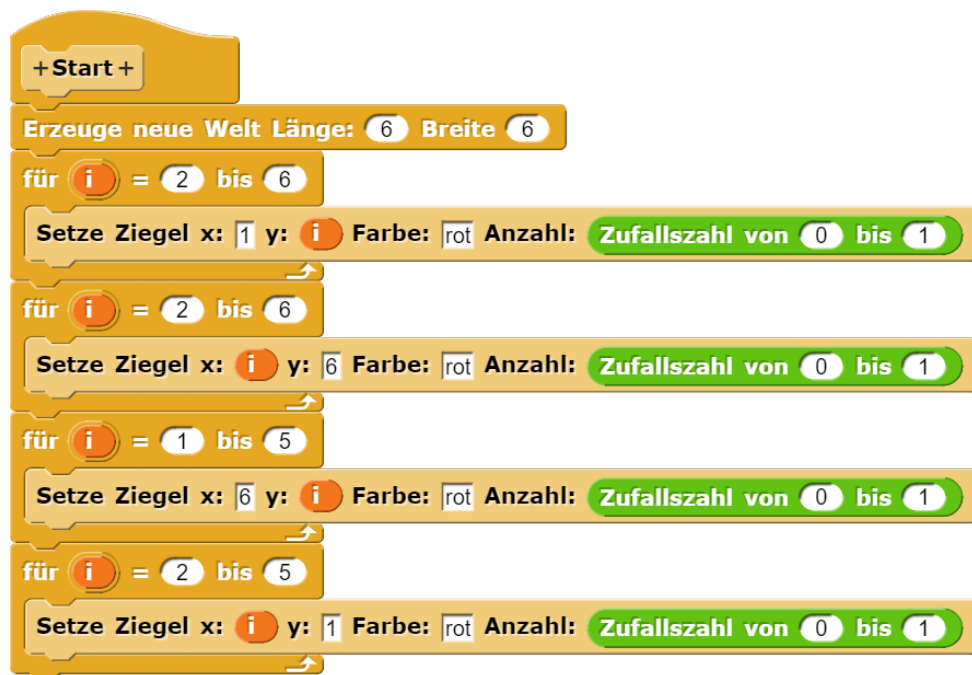
Ändere dein Programm aus Aufgabe 1 (a) so ab, dass der Roboter jeder Stelle, an der ein Stein liegt, diesen wegnimmt. An der Stelle, an der kein Stein liegt, soll er weiterhin einen hinlegen.



Beispiel für das Hinlegen der Ziegel am Rand der Welt

### 4.3.2 Startblock

Bei dieser Aufgabe beginnt Karol in der Ecke (1/1), wodurch der einfache Block zum Erzeugen der Welt verwendet werden kann. Zusätzlich werden hier zufällig Ziegel am Rand der Welt verteilt.



Erzeugen der Welt (einfacher Block) mit verteilen von Ziegeln



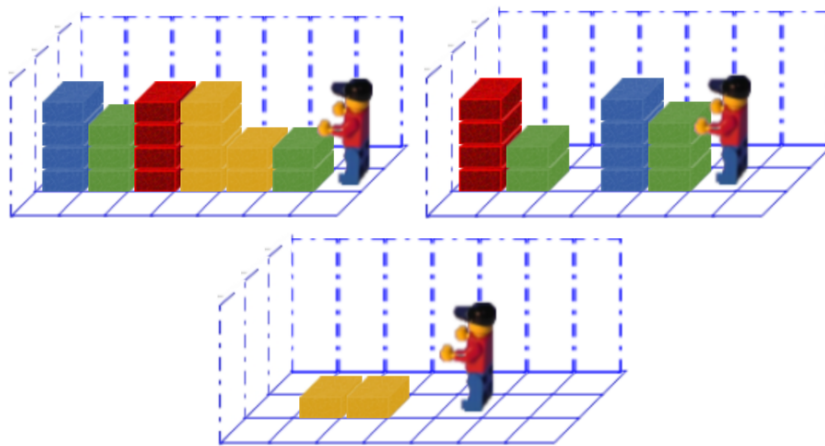
## 4.4 Bedingte Wiederholung

### 4.4.1 Aufgabenstellung

#### Aufgabe:

Karol soll die Welt aufräumen. Alle Ziegel liegen in einer Linie vor ihm. Allerdings ist unbekannt, wie viele Ziegel auf den Feldern liegen und wie weit es bis zur Wand ist.

- a) Lasse Karol bis zur Wand laufen.
- b) Auf dem Weg soll er alle Ziegel aufheben.



Beispiel für die Ziegel, die vor Karol liegen

### 4.4.2 Startblock

Bei dieser Aufgabe ist sowohl die Startposition als auch die Anzahl der Ziegel variabel. Zusätzlich wurde bei diesem Start-Block auch deren Farbe variiert.



Start Block der Aufgabe

## 5 Vorteil von Snap!-Karol

Warum sollte die Integration von Robot Karol in Snap! im Unterricht verwendet werden? Die graphische und textuelle Programmierung haben jeweils Vor- und Nachteile. Durch einen Wechsel zwischen diesen beiden Arten der Programmierung können alle Vorteile genutzt werden.

Durch die Integration von Robot Karol in Snap! müssen sich die SuS nicht in ein neues Beispiel und deren Anweisungen und Bedingungen einarbeiten. Sie können sich ganz auf die Struktur der Programmierkonzepte und Algorithmen konzentrieren.

Für heterogene Klassen kann Snap! für erweiterte Aufgabenstellungen genutzt werden. Snap! unterstützt die Konzepte der Variable und Liste und ermöglicht starken SuS anspruchsvolle Aufgabenstellungen zu bearbeiten.

Der wohl größte Vorteil von Snap! gegenüber Robot Karol ist die Zufallsfunktion. Diese kann beim Erstellen der Welt zu einer variablen Position des Karols oder eine wechselnde Position und Anzahl der Ziegel verwendet werden. Insbesondere bei den Inhalten bedingte Anweisung und bedingte Wiederholung verhindert dies ein Abzählen durch die SuS und Lösen der Aufgaben durch bloßes Aneinanderreihen von Anweisungen.

## 6 Darauf ist zu achten

Als Fehlerquellen haben sich bisher folgende Punkte ergeben:

1. Die SuS löschen oder verändern den Start-Block.
2. Die Hilfsmethoden zur Umsetzung von Robot Karol in Snap! werden angezeigt. Außerdem sind alle Blöcke, die Snap! bereit stellt ebenfalls sichtbar. Es könnte dazu kommen, dass einige SuS mit diesen statt mit den speziell für Karol benötigte Blöcke programmieren.
3. Das Konzept des Quaders und des Rucksacks sind in Snap! noch nicht umgesetzt.
4. Die Ansicht der Bühne kann nicht verschoben werden. Allerdings kann die Größe der Bühne verändert werden.
5. Die Höhe der Welt wird, unabhängig vom Wert, immer mit dem gleichen Symbol dargestellt. Das bedeutet, die Welt mit Höhe 1 sieht genauso aus wie die Welt mit Höhe 5.

## **7 Erweiterungsmöglichkeiten**

Das Programm bietet noch viele Möglichkeiten, es zu erweitern und zu verbessern.

Beispielsweise können weitere farbige Ziegeln mit eingebaut werden. Außerdem könnte die Figur des Roboters durch eine andere Figur ersetzt werden. Dies birgt die Möglichkeit, die Aufgabenstellung auch für Schülerinnen motivierender zu gestalten.

Des weiteren gibt es noch Konzepte von Robot Karol, die in Snap! nicht umgesetzt wurden. Hierzu zählen der Quader und der Rucksack.