

Bertl mit Funktionen

In diesem Abschnitt sollen die Mechanismen für die Spezifikation neuer Befehle zum Roboter Vokabular erklärt werden.

4 Funktionen durch Blöcke

Funktionen sind Programmabschnitte (Gruppe von Anweisungen/Befehlen), die eine bestimmte Aufgabe erfüllen und

getrennt vom Hauptprogramm (hier die grüne Fahne:  oder in C: `main()` – ist auch eine Funktion die beim Programmstart ausgeführt wird) stehen. Die Funktionen werden aufgerufen (ausgeführt), wenn sie in einem Programm für die Durchführung bestimmter Aktionen benötigt werden.

Die Implementation einer zusätzlichen Funktion (Neuer Block in Scratch) benötigt zumindest zwei Schritte:

1. **Definition einer Funktion** (beinhaltet die Befehle):



2. **Aufruf einer Funktion** (Ausführen der Aufgabe):



4.1 Erzeugen einer deutschen Roboter Programmiersprache

Bisher haben wir die in den Roboter (urBertl) eingebauten primitiven Befehle (Move, TurnLeft, ...) verwendet, die in der englischen Sprache enthalten sind. In diesem Abschnitt wollen wir diesen Bertl so erweitern, dass wir die Befehle in unserer natürlicheren deutschen Sprache verwenden können. Dafür brauchen wir obigen Mechanismus der Funktionen.

```
Bewege ();  
LinksDrehen ();  
RechtsDrehen ();  
PagerAufheben ();  
PagerHinlegen ();
```



Erzeugung neuer Befehle (Creating a Custom Block - Function)

Um eine Funktion (einen benutzerdefinierten Block) erstellen zu können, öffnen Sie die "Weitere Blöcke, engl. **More Blocks** " Kategorie und drücken Sie die "Neuer Block, engl. **Make a Block**" Button. Ein neues Block Menü öffnet sich, in dem Sie Ihre individuellen Blocks benennen können (hier **Bewege**). Sie können festlegen, was Ihre benutzerdefinierten Blocks tut, indem ein Skript auf seine definierten Blocks schnappen.

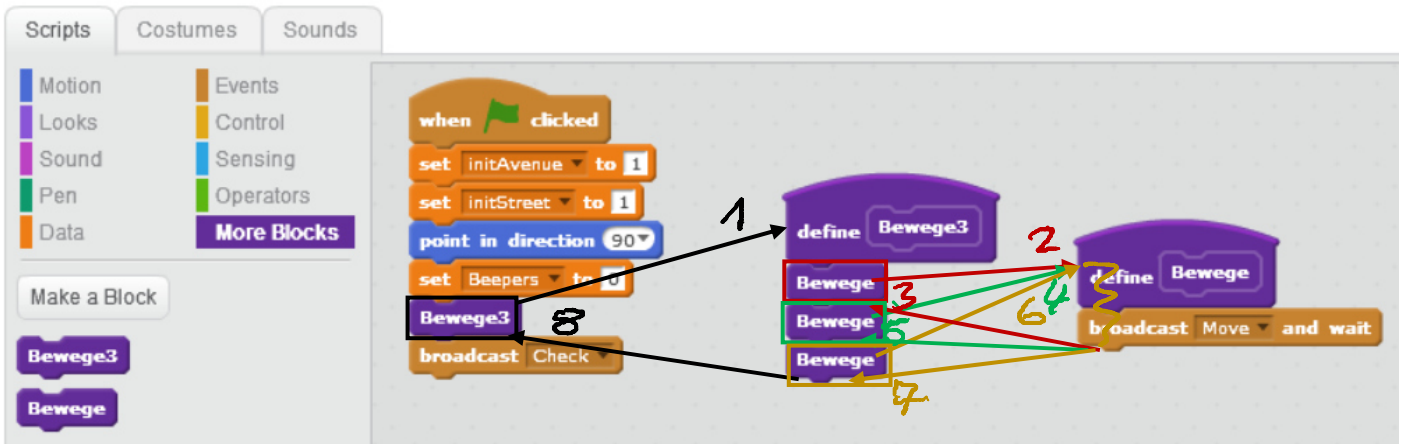


4.2 Funktionen mit Funktionsaufrufen

Natürlich können neue Funktionen auch bestehende Funktionen aufrufen, z.B. Bewege3 soll drei Mal die obige Funktion Bewege aufrufen:

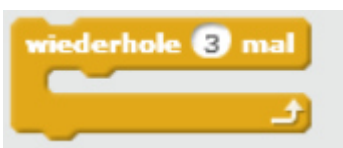
Weitere Blöcke → Neuer Block: **Bewege3**.

In der Funktionsdefinition kommt der Funktionsaufruf von **Bewege**:



Einwurf: Schleifen – Loops (kommt später genauer)

Anstelle der drei Aufrufe von **Bewege**, kann man auch eine Schleife (loop) verwenden. In Scratch wird eine Schleife mit einer fixen Anzahl von Durchläufen in **Steuerung** engl. **Control** durch den Block:



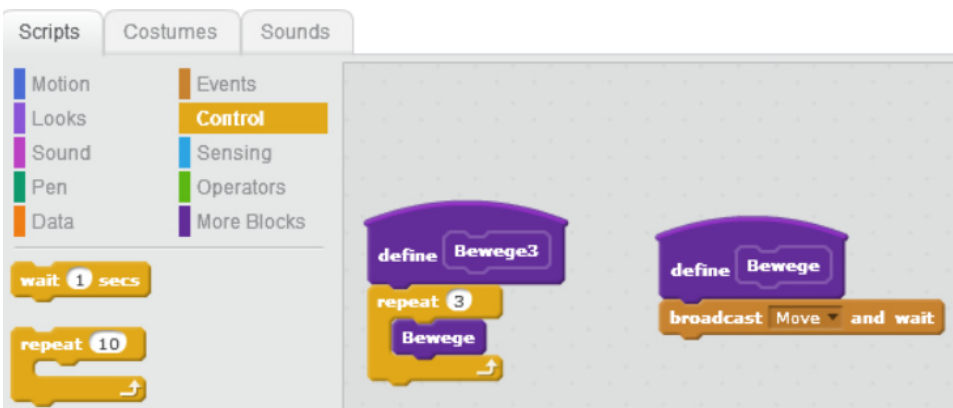
oder engl.



verwendet, wobei die Anzahl der Durchläufe (hier: 3)

im weißen Kreis eingetragen wird.

Damit ergibt sich für die obere Funktion: **Bewege3** folgender Code mit gleicher Aufgabe die Funktion Move über Bewege drei Mal aufzurufen:



Will man nicht nur immer drei Moves machen, sondern unterschiedlich oft, so muss der Funktion mittels eines Parameters mitgeteilt werden wie oft der Bertl „Moven“ soll:

4.3 Erzeugung neuer Befehle (Function) mit Argumenten (Parametern)

Um eine Funktion (einen benutzerdefinierten Block) erstellen zu können, öffnen Sie die "Weitere Blöcke, engl. **More Blocks**" Kategorie und drücken Sie die "Neuer Block, engl. **Make a Block**" Button. Ein neues Block Menü öffnet sich, in dem Sie Ihre individuellen Blocks benennen können (hier **BewegeXmal**).

Durch Drücken der "**Einstellungen** oder engl. **Options**" können Sie Zahlen (Number), Text (String), bzw. boolesche Felder hinzuzufügen. Sie können auch mehrere Einträge in den Block, hinzufügen oder ohne Bildschirmaktualisierung laufen lassen. Nach der Benennung und dem hinzufügen von Argumenten, wird ein definieren Block erstellt. Die Einstellungen können später durch einen Rechtsklick auf den definierten Block geändert werden; wählen Sie "**Bearbeiten** engl. **Edit**". Sie können festlegen, was Ihre benutzerdefinierten Blocks tun sollen, indem Sie ein Skript auf seinen definierten Block schnappen.



Beim Aufruf der Funktion **BewegeXmal(3)** wird in das Argument (Parameter) **steps** die Zahl 3 geschrieben und der Inhalt (Bewege) der Schleife **repeat steps**-Mal (hier: 3 Mal) ausgeführt, sodass die Funktion Bewege 3 Mal aufgerufen wird.

4.4 Aufgabe:

Programmieren Sie die Stages aus dem letzten Mal mit Funktionen und möglichst wenig Befehlen in main(). 