

Der Bewegungssensor - Neustart und Geschwindigkeit

So, das was ihr bis jetzt geschafft habt, ist ja schon mal ein erster Erfolg, aber im Moment fehlt dem kleinen Zauberer noch die gewisse Ruhe...

Es wäre ja irgendwie gut, wenn der Zauberer auch stehen bleiben könnte. Daher soll im Code als nächstes ein Bereich geschaffen werden, in dem die Geschwindigkeit auf 0 gesetzt wird.

Als weitere Optimierung soll das Spiel neu gestartet werden können, um den Punktestand zurück zu setzen.

Auch hierfür gibt es einen Befehl, der über die Konsole gesendet werden kann und zwar '#n'.

Dieser Code sollte gesendet werden, wenn der Taster gedrückt ist.

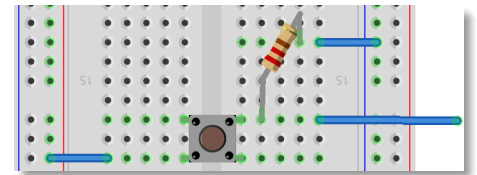


Abb. 1 : Tasterschaltung



1. Erweitert euren Code so, dass es einen Bereich gibt, in dem der Zauberer stehen bleibt, also die Geschwindigkeit auf 0 gesetzt wird – sonst soll die Geschwindigkeit auf einem festen Wert $\neq 0$ sein
2. Fügt zudem in euren Code eine Abfrage ein, die den Taster überprüft und, falls der Taster gedrückt ist, den Befehl zum Neustart sendet. In dem Falle sollte dann auch kein links-rechts-Befehl gesendet werden.
3. Testet erneut das Spiel

Sehr gut, damit seid ihr schon einen großen Schritt voran gekommen!

Nun ist es euch vielleicht schon passiert, dass der Zauberer etwas zu langsam oder zu schnell war und ihr habt vielleicht sogar intuitiv versucht, den Gamecontroller weiter oder weniger weit in die Richtung zu drehen in die der Zauberer gehen sollte. Dieses Verhalten sollt ihr als nächstes implementieren. Ihr habt hierzu ja bereits die minimalen und maximalen Werte entlang der x-Achse bestimmt und auch dort einen Ruhebereich festgelegt, in dem der Zauberer sich nicht bewegt. Da die Geschwindigkeit jedoch auf Werte zwischen 0 und 20 begrenzt ist, müssen nun die Bereiche für links und rechts auf diesen Geschwindigkeitsbereich 'gemapt' werden.

Der Arduino bietet hierfür bereits eine integrierte Funktion, mit der ihr dies bewerkstelligen könnt:

```
map( Eingangswert: value, Eingangsgrenzen: fromLow, fromHigh, Ausgangsgrenzen: toLow, toHigh );
```

Der Bewegungssensor - Neustart und Geschwindigkeit

Die folgende Beschreibung der Funktion aus der Arduino-Dokumentation erklärt dies noch detaillierter:

Description

Re-maps a number from one range to another. That is, a **value** of **fromLow** would get mapped to **toLow**, a value of **fromHigh** to **toHigh**, values in-between to values in-between, etc.

Does not constrain values to within the range, because out-of-range values are sometimes intended and useful. The `constrain()` function may be used either before or after this function, if limits to the ranges are desired.

Note that the "lower bounds" of either range may be larger or smaller than the "upper bounds" so the `map()` function may be used to reverse a range of numbers, for example

```
y = map(x, 1, 50, 50, 1);
```

The function also handles negative numbers well, so that this example

```
y = map(x, 1, 50, 50, -100);
```

is also valid and works well.

The `map()` function uses integer math so will not generate fractions, when the math might indicate that it should do so. Fractional remainders are truncated, and are not rounded or averaged.

Abb. 2 : Dokumentation zur Map-Funktion



Beachtet: die `map`-Funktion liefert einen Integer zurück, der erst noch in einen String umgewandelt werden muss

Hier empfiehlt es sich die Befehlsübersicht zu Rate zu ziehen



1. Erweitert euren Code so, dass die Geschwindigkeit mit Hilfe der `map`-Funktion für den rechten und den linken Bereich auf die entsprechende Geschwindigkeit abgebildet wird
2. Testet ... ach ihr wisst schon

HERZLICHEN GLÜCKWUNSCH!!!

Euer Projekt war ein voller Erfolg. Jetzt könnt ihr eigene Ideen umsetzen. Schaut dazu einmal in die Readme-Datei im WizzyWizards-Ordner. Wenn ihr Anregungen braucht, fragt einen Betreuer.

Quellenverzeichnis:

Abb. 1 - Quelle: Erstellt mit Fritzing-Software (<http://fritzing.org>)

Abb. 2 - Quelle: Arduino-Dokumentation (<http://www.arduino.cc>)

Alle weiteren Grafiken/Icons - Quelle: InfoSphere