

# Handreichung: Lightbot



## Inhalt

1	Beschreibung .....	2
2	Überblick über die Grundfunktionen der Menüs.....	2
3	Informatischer Inhalt.....	4
3.1	Kapitel 1 - Grundlagen – Bewegung der Spielfigur.....	4
3.2	Kapitel 2 - Prozeduren / Funktionen .....	5
3.3	Kapitel 3 - Schleifen .....	7
4	Kommentierte Musterlösungen.....	8
4.1	Kapitel 1 - Grundlagen .....	8
4.2	Kapitel 2 - Prozeduren / Funktionen .....	11
4.3	Kapitel 3 - Schleifen .....	15
5	Übungsaufgaben:.....	18
6	Platz für persönliche Notizen.....	20
	Kopiervorlage Bildkarten.....	21
	Kopiervorlage Programmtabelle.....	22
	Kopiervorlage Programmtabelle.....	22
	Kopiervorlage Isopapier:.....	24

## 1 Beschreibung

Bei dem Programmierspiel *Lightbot code hour* - aus dem Genre *Programming puzzles* - muss eine kleine Roboter-Spielfigur so programmiert werden, dass sie sich durch kleine Hindernissparcour bewegt und auf bestimmten Feldern eine Lampe einschaltet.

Dazu werden einzelne Befehle grafisch zu einem Programm zusammengestellt. Die SuS erleben dabei, dass sich Programme aus einzelnen Befehlen, die nacheinander ausgeführt werden, zusammensetzen. Im Verlauf des Spiels werden die Informatischen Konzepte „**Schleife**“ und „**Funktion**“ eingeführt. Zu Beginn einiger Levels werden dazu neue Befehle und Konzepte mittels Sprechblasen eingeführt.

Das Spiel ist grundsätzlich kindgerecht für die Grundschule aufbereitet, insbesondere bei den höheren Schwierigkeitsstufen brauchen Kinder jedoch erklärende Unterstützung und Anleitung.

## 2 Überblick über die Grundfunktionen der Menüs

Lightbot code hour bietet drei Hauptansichten: Das **Hauptmenü**, die **Levelauswahl** und die „**Spielansicht**“.

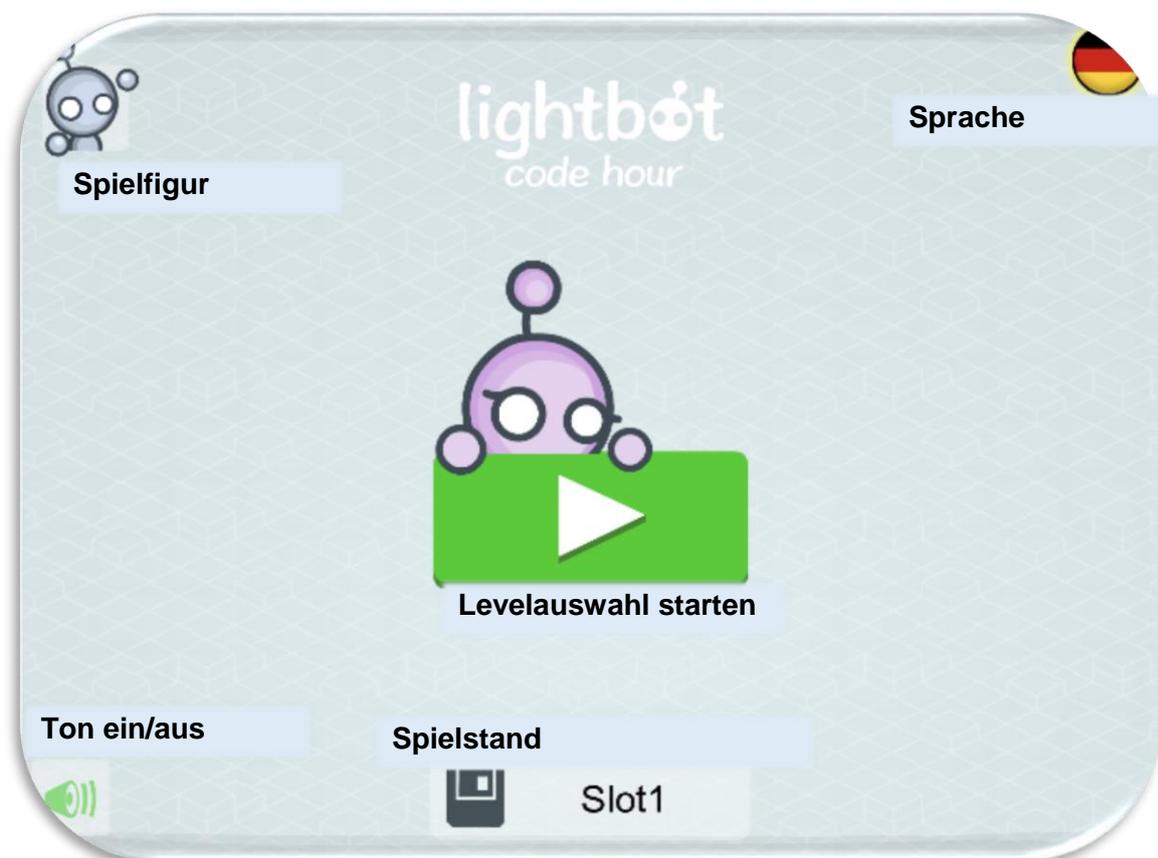


Abbildung 1: Ansicht Hauptmenü mit Funktionsbeschreibung

Im Hauptmenü kann ein Spielstand angelegt oder geladen werden, die Sprache sowie Spielfigur gewechselt werden. Das Anlegen und Laden von Spielständen ist recht selbsterklärend.

Startet man die Levelauswahl, kann man zwischen drei Kapiteln mit jeweils mehreren Level auswählen, die einzelnen Levels müssen allerdings der Reihe nach gelöst werden und werden zunehmend schwieriger und komplexer.



Abbildung 2: Ansicht Levelauswahl

Das Spielfeld besteht zum einem aus dem Programm „**Main**“ (rechter Rand), dem „**Befehlsmenü**“ (unterer Rand) und dem **Hindernissparcour** (Mitte) sowie einigen Steuerschaltflächen. Die Befehle am unteren Fensterrand können durch Tippen oder Ziehen dem Programm hinzugefügt bzw. entfernt werden. Mit „**Start**“ (oben) wird das Programm gestartet und die Spielfigur in Bewegung gesetzt. Auf den blauen Feldern soll die Spielfigur ihre Lampe einschalten. Das Level ist beendet, wenn alle blauen Felder beleuchtet sind.

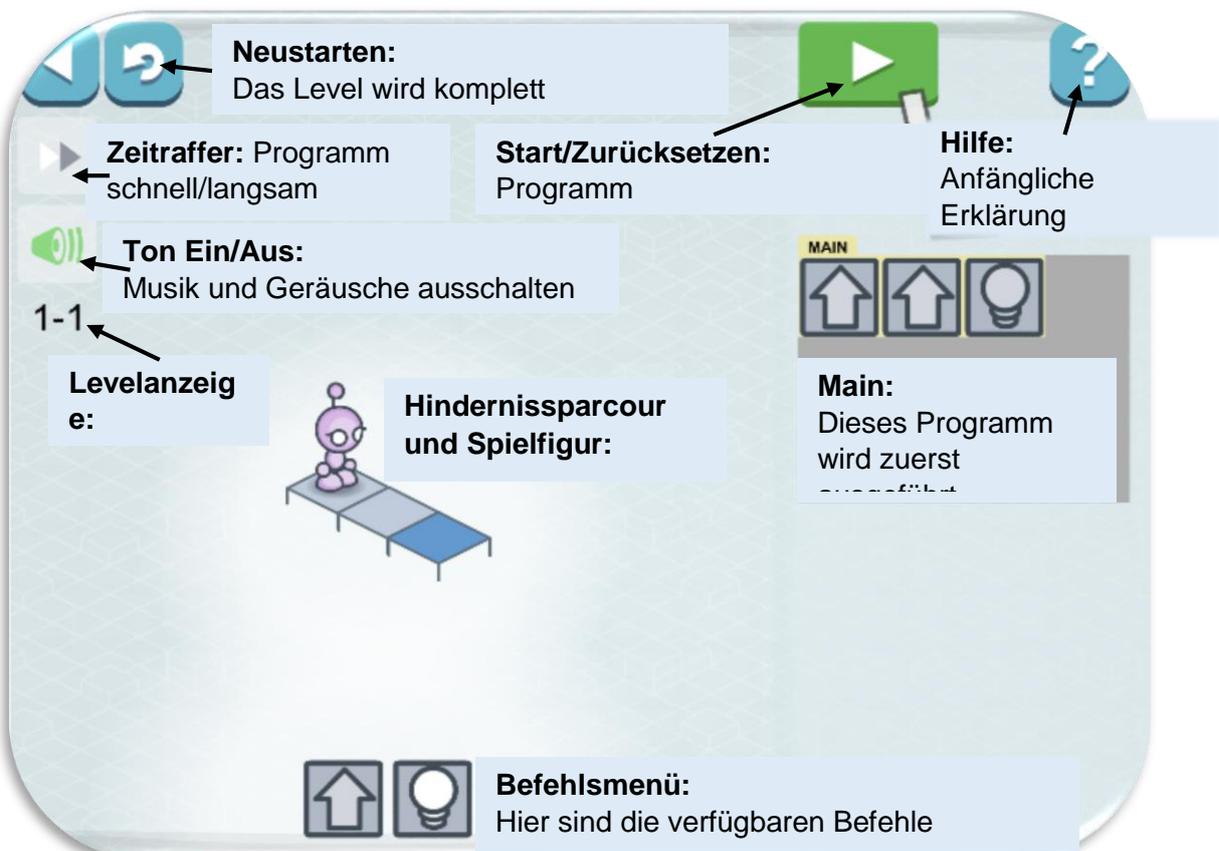
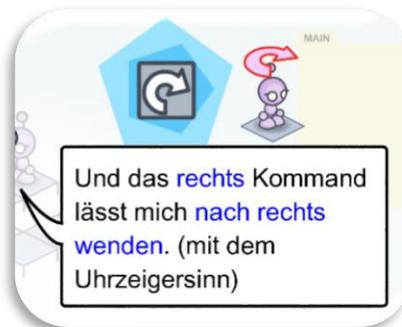
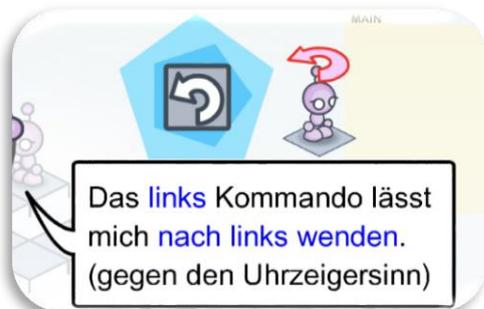


Abbildung 3: Übersicht Spielansicht

### 3 Informatischer Inhalt

#### 3.1 Kapitel 1 - Grundlagen – Bewegung der Spielfigur

Die Grundbefehle der Spielfigur sind **Vorwärts gehen**, **Beleuchten**, **Links Wenden**, **Rechts Wenden** und **Springen**. Diese Befehle sind für die SuS in der Regel selbsterklärend, lediglich die Orientierung beim Wenden müssen einige durch ausprobieren lösen.



## 3.2 Kapitel 2 - Prozeduren / Funktionen

Schwieriger zu verstehen ist die Verwendung von Prozeduren (PROC1) und gelingt den SuS nicht selbstständig. Informatisch gesehen ist dies eine eigenständige **Funktion** oder **Prozedur**. Dies kann am ehesten als eine Zusammenfassung von Befehlen verstanden werden, die in dieser Reihenfolge mehrfach vorkommen. Beim Programmieren dient dies der Übersichtlichkeit des Programmcodes und der Auslagerung von Programmteilen.

Die SuS verstehen dies häufig fälschlich zuerst als Verlängerung von *Main*, nicht als eigenständige Befehlsabfolge.

### Beispiel:

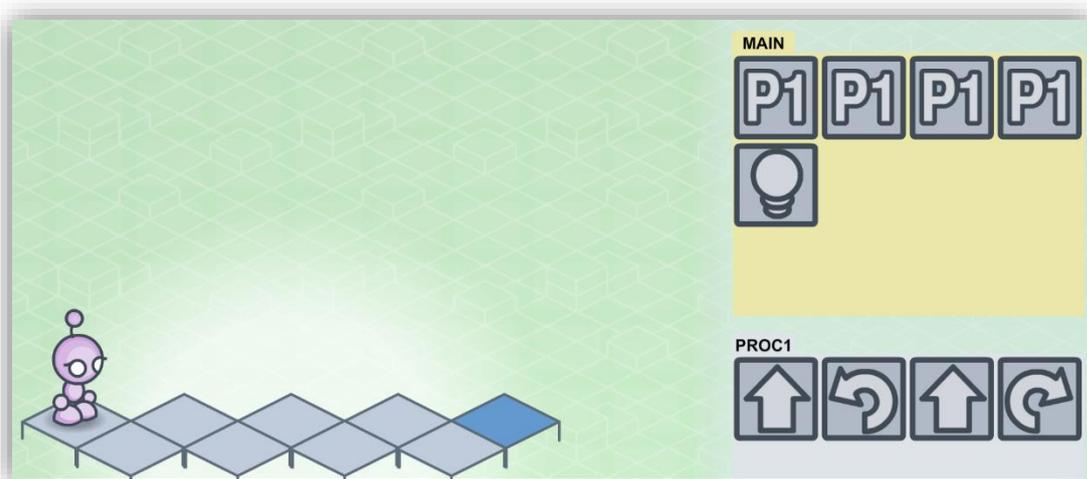
**Statt:** „Zieh deine Schuhe an, Binde den Schuh links, Binde den Schuh rechts, öffne die Türe, geht raus, mach sie wieder zu, Setz deinen Fuß 1250 mal nach vorne ...“

**sagt man** (zumindest theoretisch): „Geh zur Schule!“

**Statt jeden Abend zu sagen:** „Zahnbürste raus, Zahnpasta auf, Zahnpasta drauf ...“.

**Sagt man jeden Abend:** „Zähneputzen!“

Trotz der Zusammenfassung in einen Befehl, werden diese detaillierten Bewegungsabläufe ausgeführt. Im Spiel ist dies ein neuer Programmteil zusätzlich zum Bereich *Main* und kann durch den neuen Befehl **P1** aufgerufen werden:



### Erklärung:

**Statt:** [Vorwärts+links wenden+Vorwärts+rechts wenden+Vorwärts+links wenden+Vorwärts+rechts wenden+Vorwärts+links wenden+Vorwärts+rechts wenden+Lampe ein]

wird die Befehlskette **zusammengefasst zu:**

[4x(Vorwärts+links wenden+Vorwärts+rechts wenden)+Lampe ein]

Die Aufgabe für die SuS besteht vor allem darin, sich wiederholende Programmteile zu erkennen und diese in Proc1 auszulagern.

Dies kann durch legen des Programmcodes mit Bildkärtchen und Finden von Wiederholungen geschehen. Dazu sind Bildkärtchen zum Ausschneiden als Kopiervorlage hinterlegt.

Die Erklärung von Proc1 durch die App überspringen die SuS in der Regel und verstehen diese auch nicht. An dieser Stelle muss die Lehrkraft diese Aufgabe übernehmen.

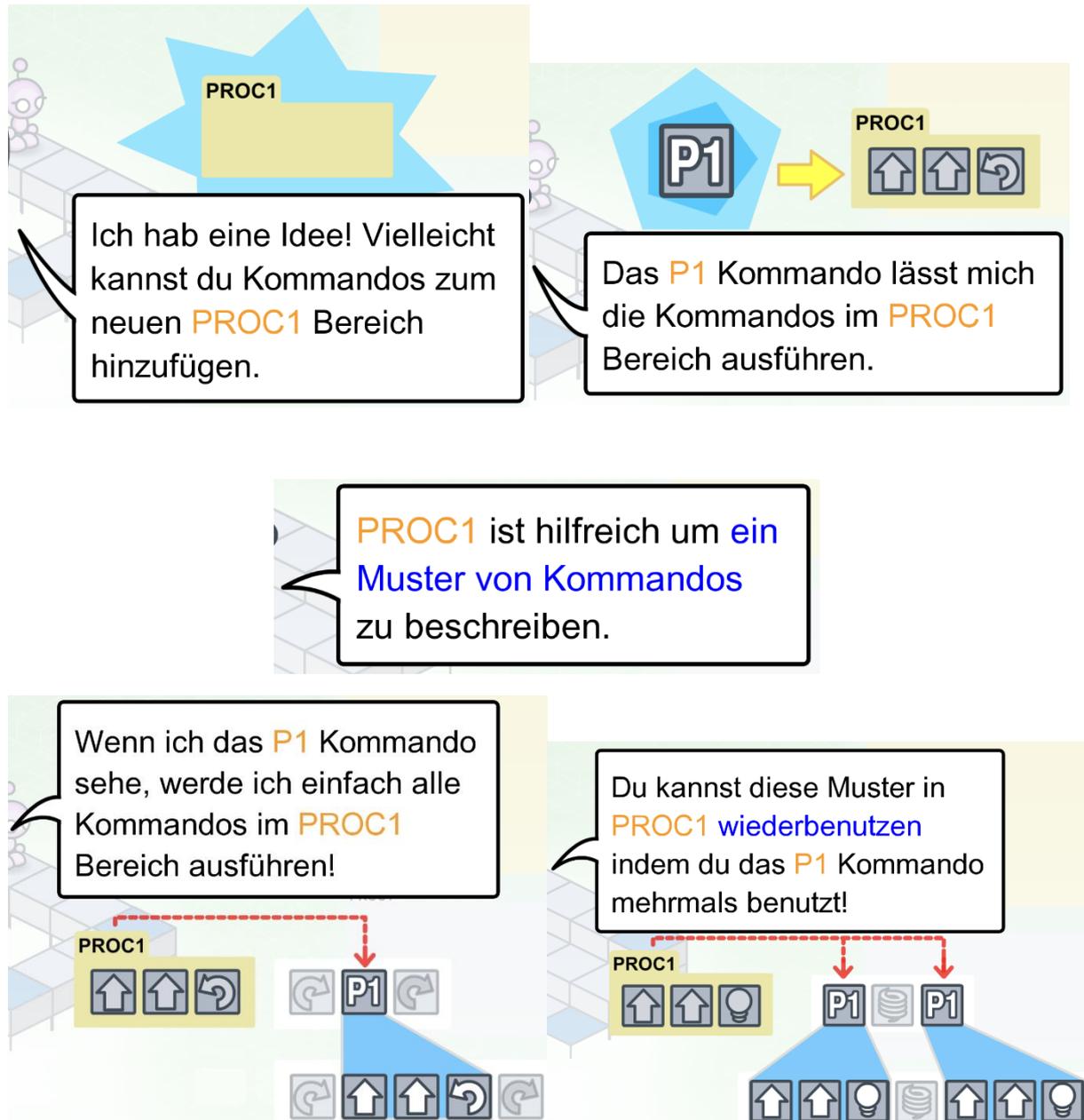


Abbildung 4: Erklärung der Verwendung von Proc1 durch Lightbot

### 3.3 Kapitel 3 - Schleifen

Schleifen sind ein Mittel in der Informatik, mit denen einzelne Programmteile mehrfach oder unendlich oft ausgeführt werden können.

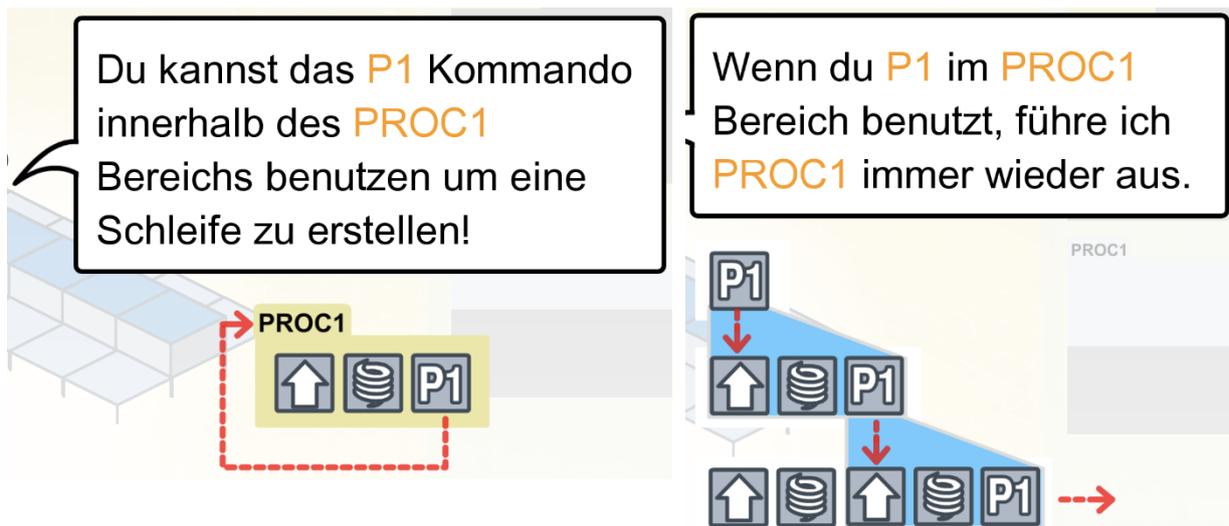
#### z.B.:

- 1000 mal eine Zahl addieren und diese Ausgeben.
- Immer wieder eine Temperatur messen.
- Temperatur messen, heizen bis eine Temperatur erreicht ist.
- jede 10 Minuten den Provider fragen, ob eine E-Mail eingegangen ist.

#### Auch SuS arbeiten in Schleifen:

- Ein Halbjahr lang jede Woche den gleichen Stundenplan abarbeiten, außer es sind Ferien
- 3 Minuten lang die Zähne Putzen  
Einatmen, Ausatmen....

In Lightbot werden Schleifen dadurch umgesetzt, dass sich das Programm selbst wieder aufruft. Die SuS finden es zu Beginn auch witzig, die Spielfigur nur immer weiter springen oder einen Tanz ausführen zu lassen.



Für die SuS kann dieses Vorgehen, dass sich das Programm selbst wieder aufruft, an dem nebenstehendem Kinderlied verdeutlicht werden:

*Ein Mops kam in die Küche  
und stahl dem Koch ein Ei.  
Da nahm der Koch den Löffel  
und schlug den Mops entzwei.*

*Da kamen viele Möpse  
und gruben ihm sein Grab  
und setzten ihm nen Grabstein,  
auf dem geschrieben stand:*

*Ein Mops kam in die Küche  
und stahl dem Koch ein Ei.  
Da nahm der Koch den Löffel  
und schlug den Hund entzwei...*

Abbildung 5: Beispiel für eine rekursive Schleife

## 4 Kommentierte Musterlösungen

Im Folgenden sind Lösungsvorschläge für die einzelnen Level dargestellt. Bei einigen Level gibt es mehrere Lösungsmöglichkeiten.

### 4.1 Kapitel 1 - Grundlagen

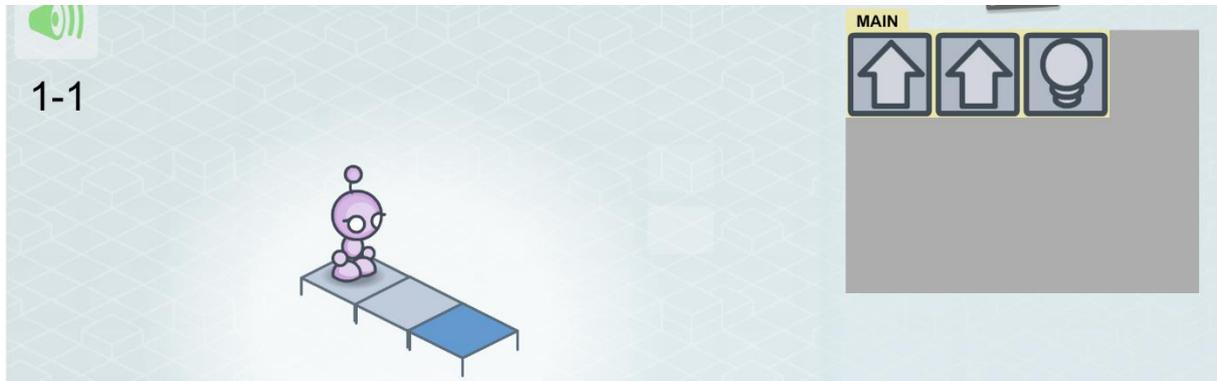


Abbildung 6: Lösung 1-1: Einführung der ersten Befehle und des Spielprinzips

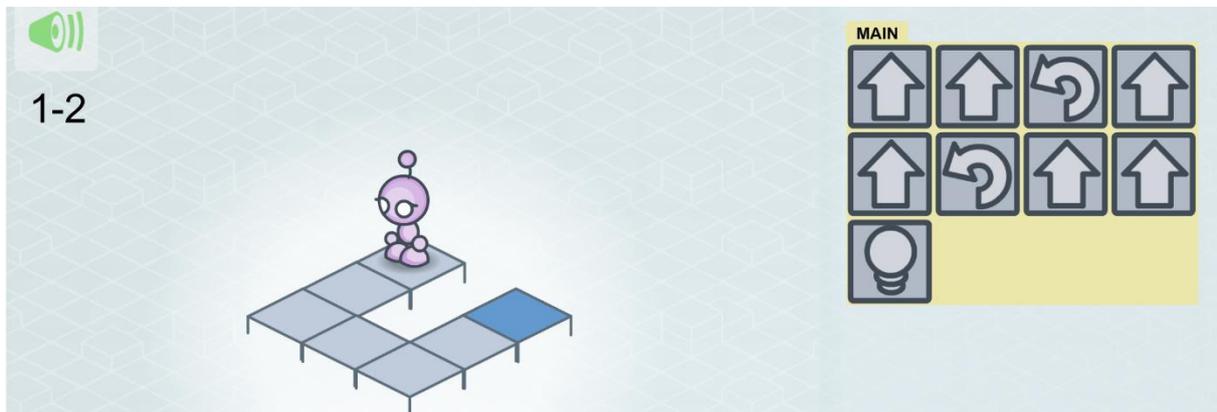


Abbildung 7: Lösung 1-2: Einführung der Wendung

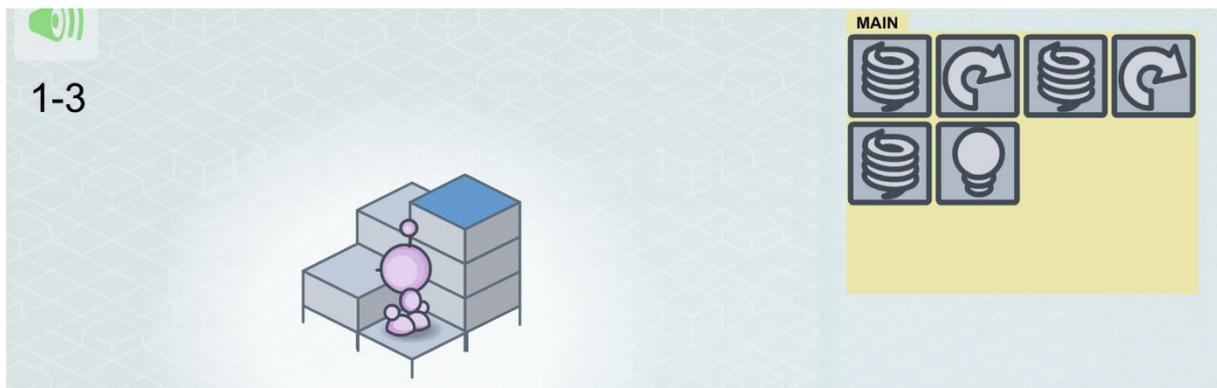


Abbildung 8: Lösung 1-3: Einführung des Springens

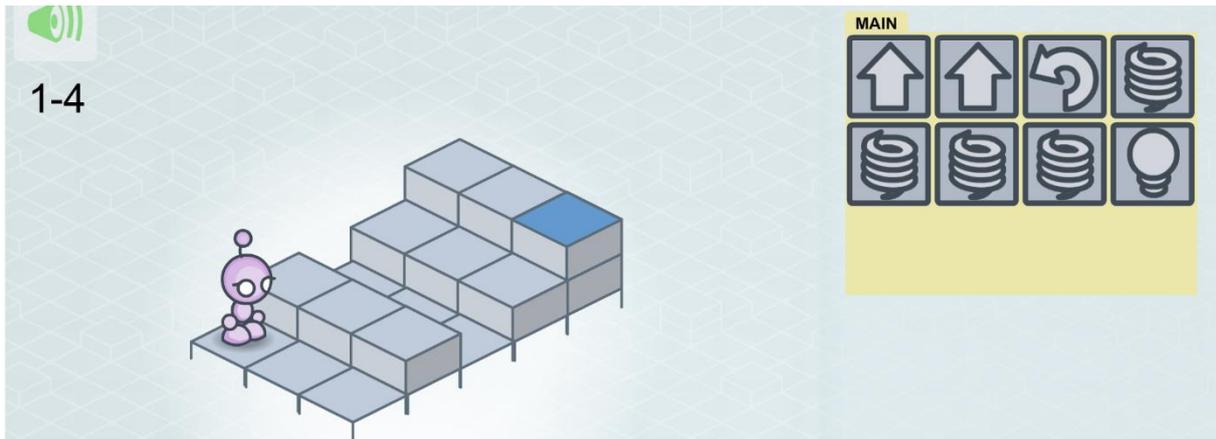


Abbildung 9: Lösung 1-4: Anwenden des Springens

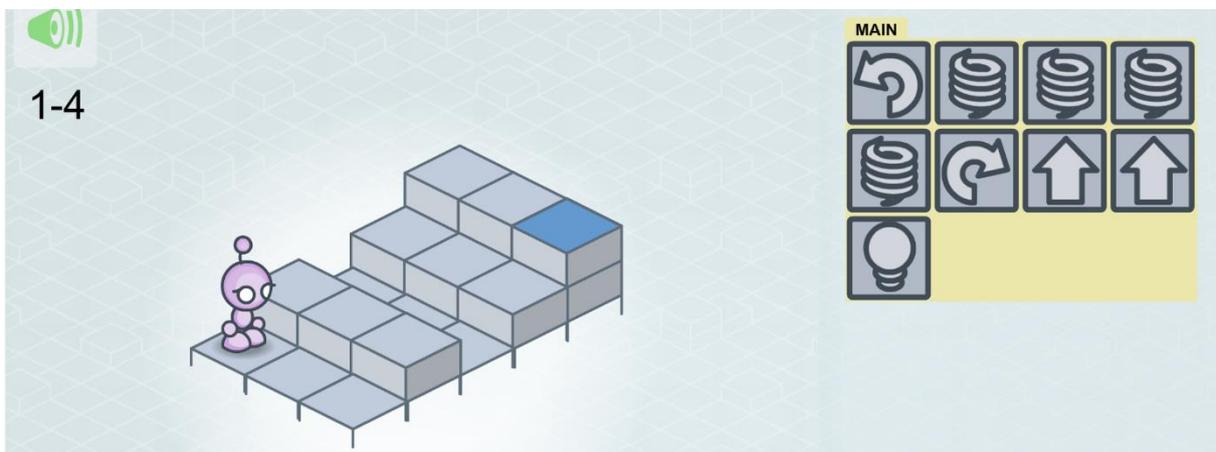


Abbildung 10: 1-4: Anwenden des Springens, alternative Lösung

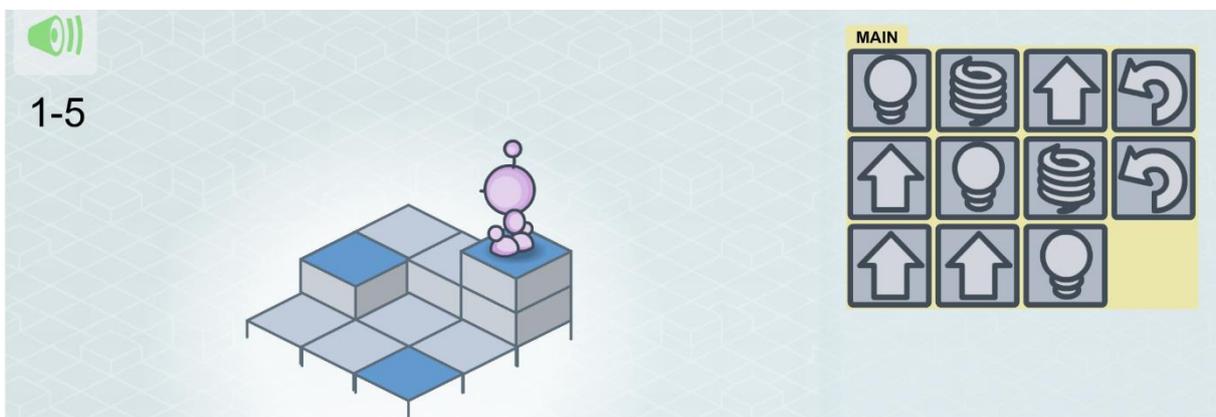


Abbildung 11: Lösung 1-5: Mehrere blaue Felder

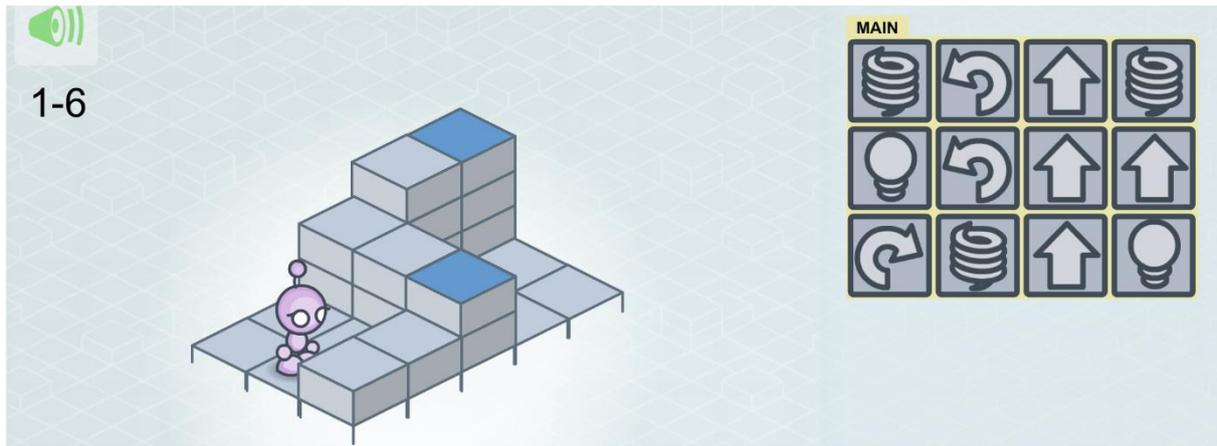


Abbildung 12: Lösung 1-6

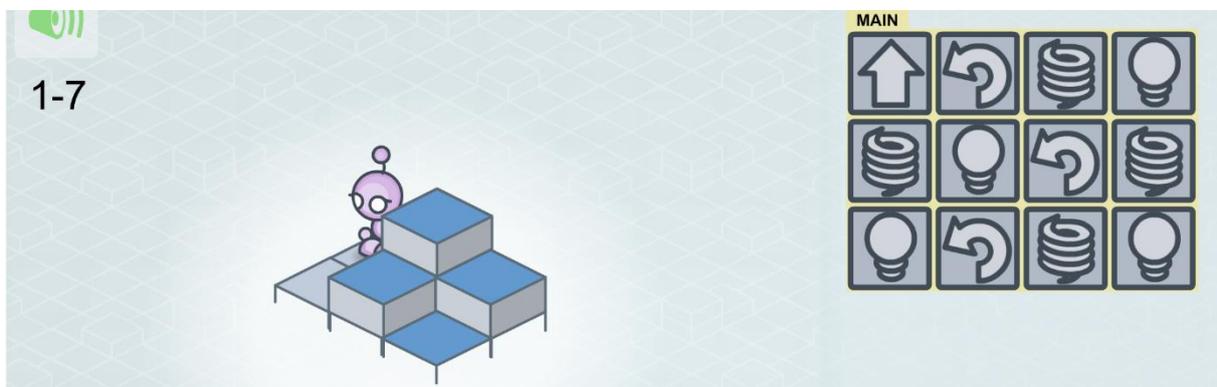


Abbildung 13: Lösung 1-7

Bei diesem Level wollen die SuS häufig oben über die Treppe und dann heruntergehen. Dieses Vorgehen ist prinzipiell richtig, dazu fehlt jedoch ein Befehlsfeld und führt häufig zu Verwirrung. Der Korrekte weg führt unten entlang.

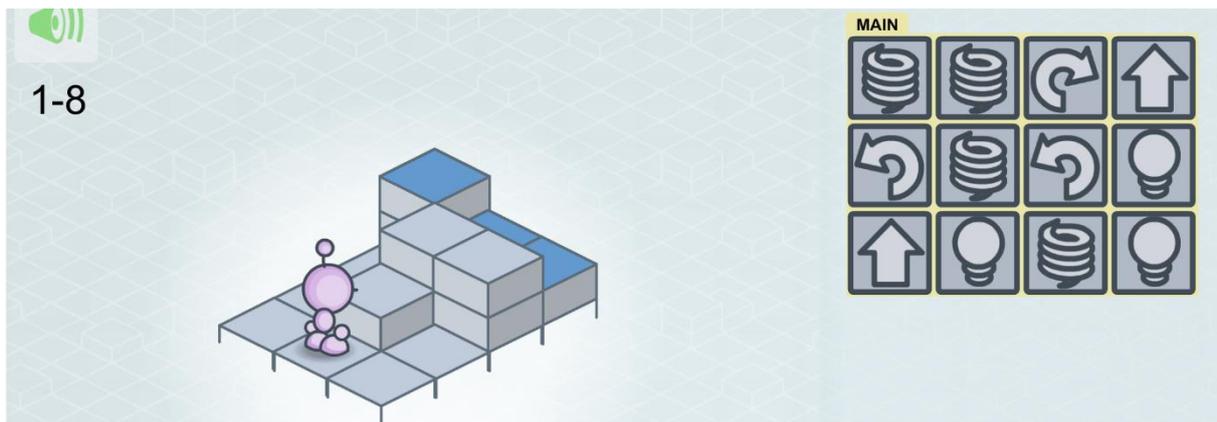


Abbildung 14: Lösung 1-8

## 4.2 Kapitel 2 - Prozeduren / Funktionen

Zum Herausarbeiten der Funktionen kann die Leitfrage dienen: „Was macht die Figur immer gleich?“ (z.B. drei Schritte nach vorne gehen). Können ähnliche Ausgangspunkte gefunden werden?

Eine weitere Herangehensweise kann es sein, anhand eines Lösungsversuchs die Bewegung der Spielfigur vorherzusagen und gemeinsam zu diskutieren. Insbesondere bei zwei Prozeduren ist es bereits herausfordernd, nachzuvollziehen und zu beschreiben wie das Programm im Detail abläuft.

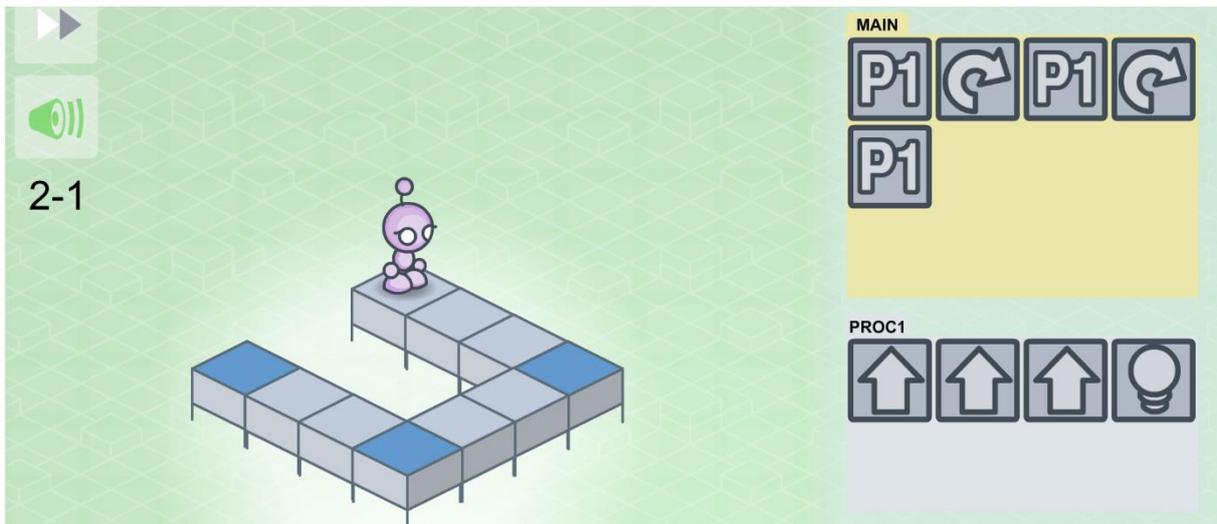


Abbildung 15: Lösung 2-1: Einführung Proc1, noch nicht die ideale Lösung

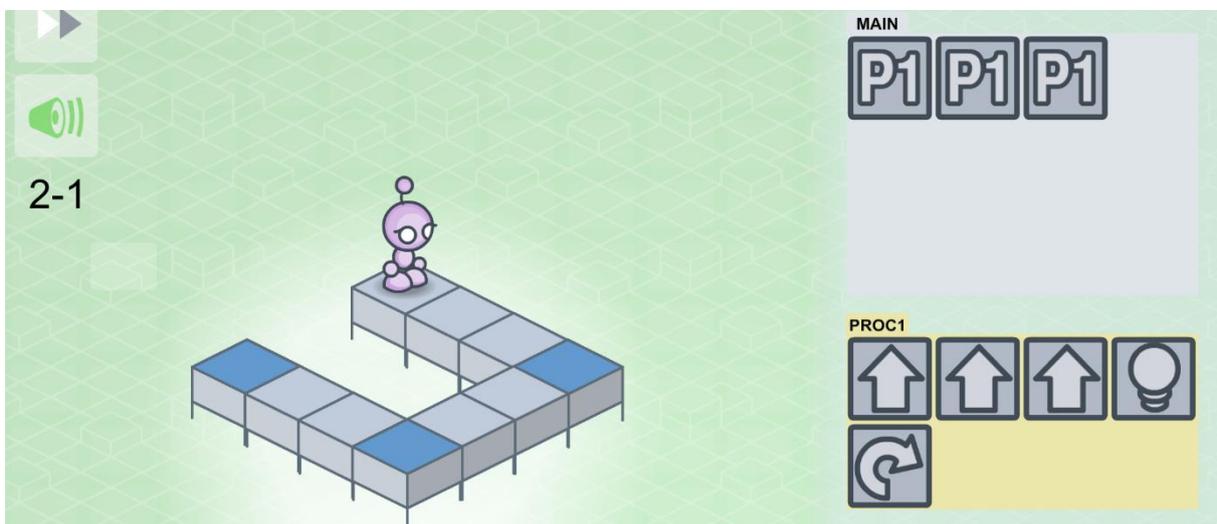


Abbildung 16: Lösung 2-1: Alternative Lösung, die „Drehung“ und „Lampe einschalten“ wurden in Proc1 integriert.

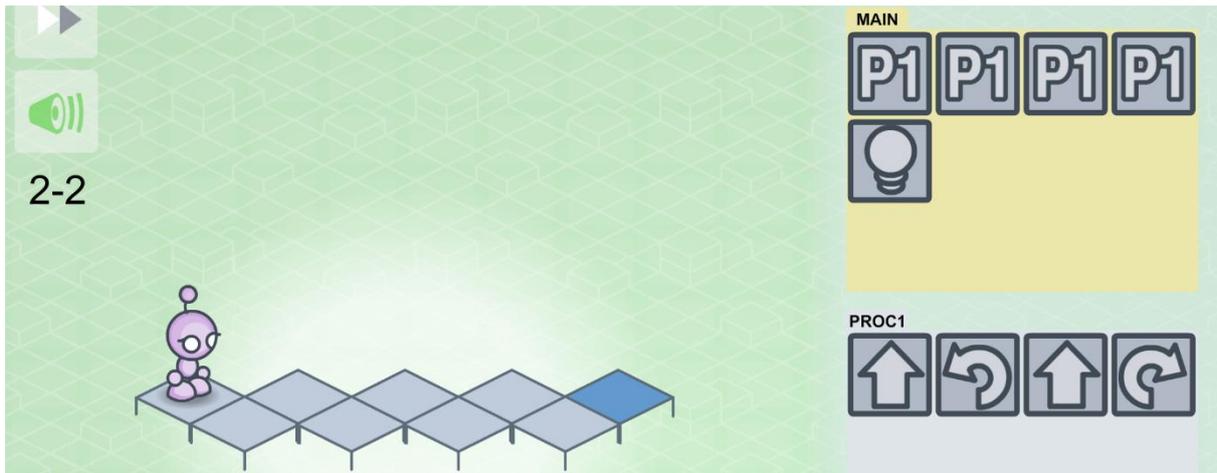


Abbildung 17: Lösung 2-2

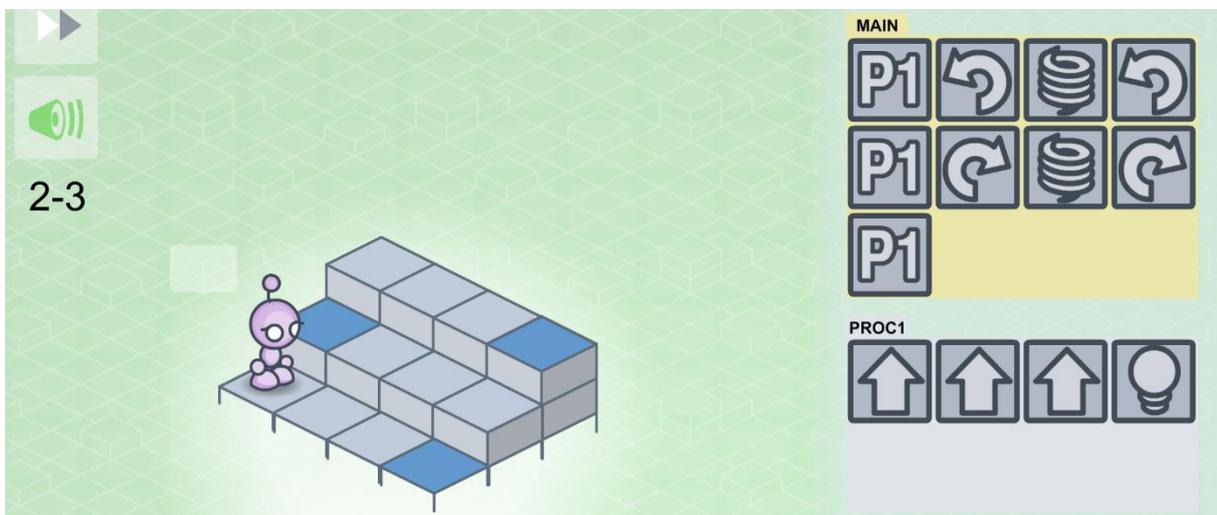


Abbildung 18: Lösung 2-3

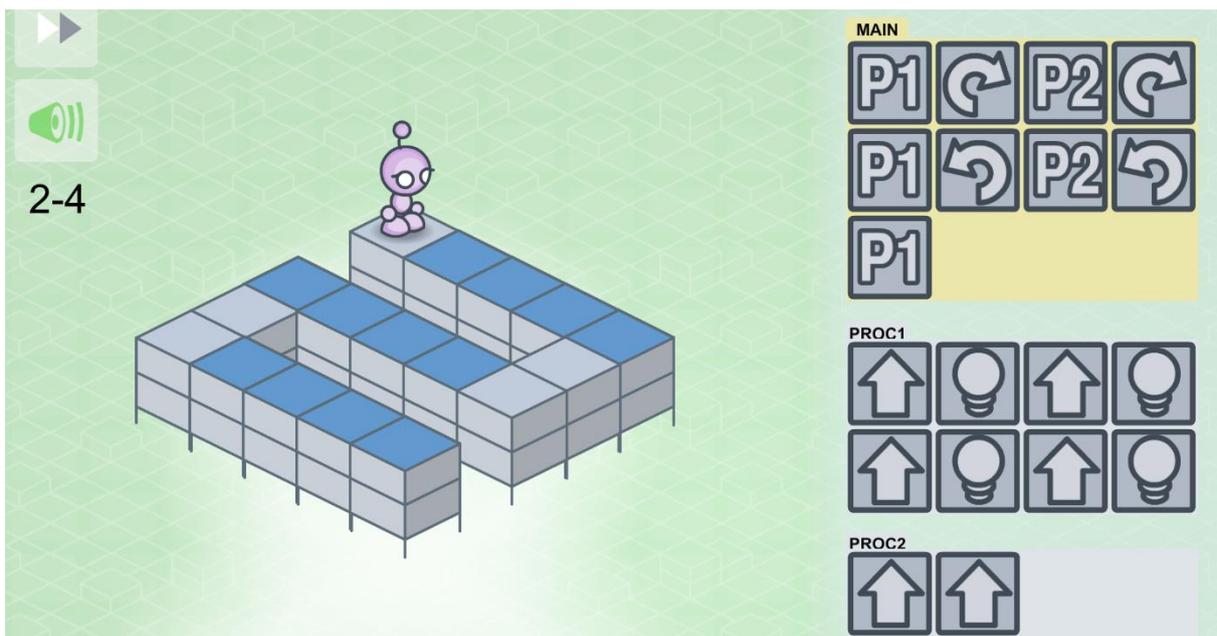


Abbildung 19: Lösung 2-4: Einführung Proc2

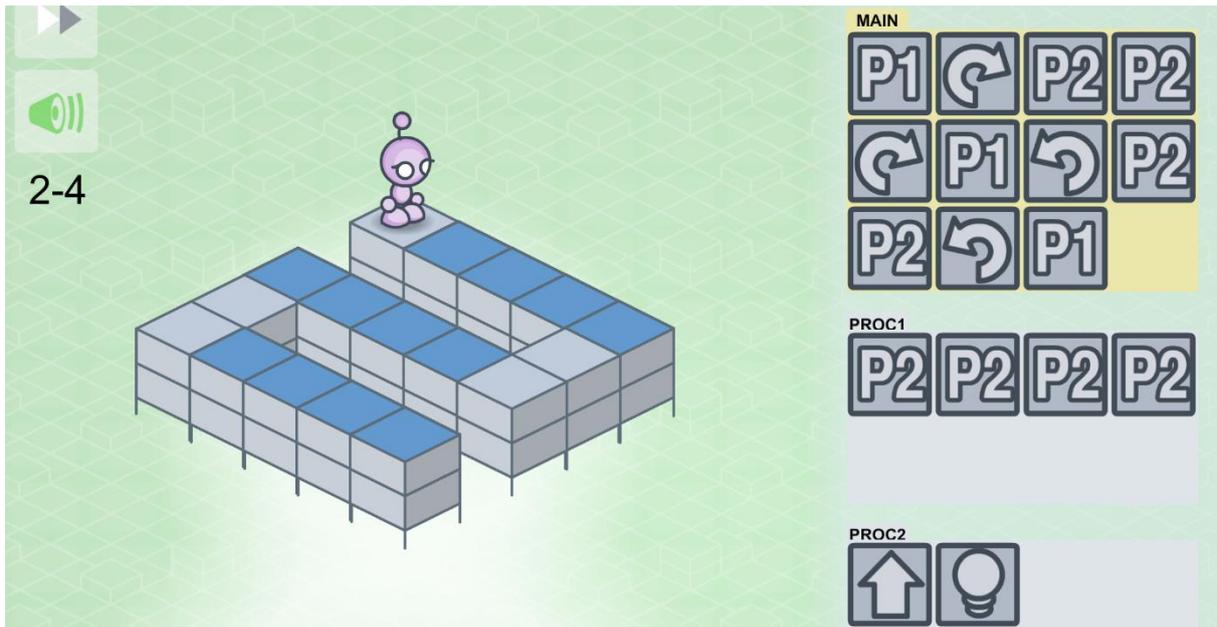


Abbildung 20: Lösung 2-4: alternative Lösung. Die Spielfigur führt zu viele Befehle aus, kommt dennoch zum Ziel.

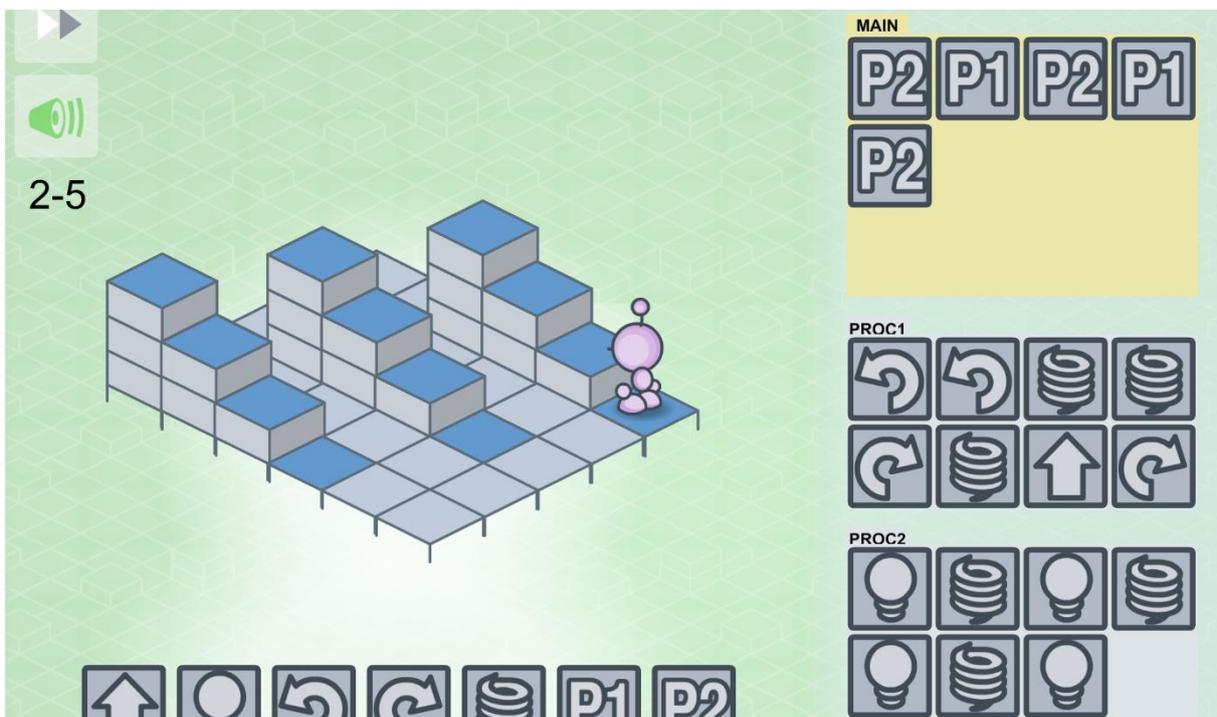


Abbildung 21: Lösung 2-5

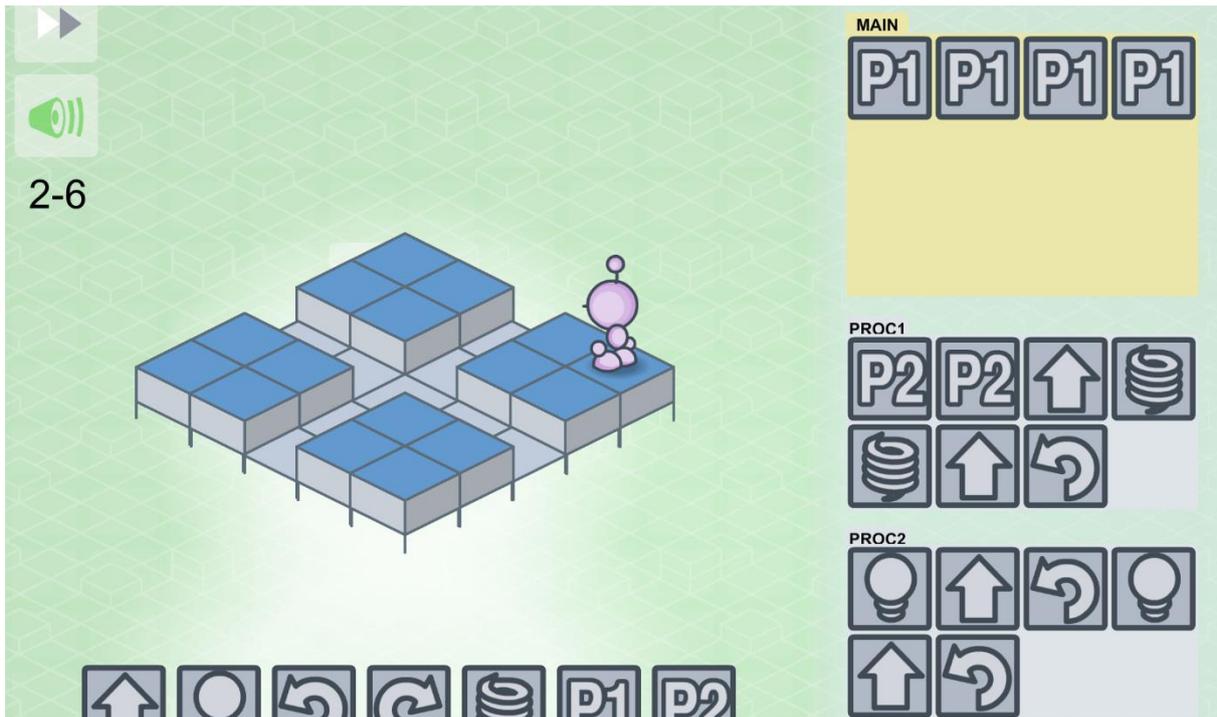


Abbildung 22: Lösung 2-6

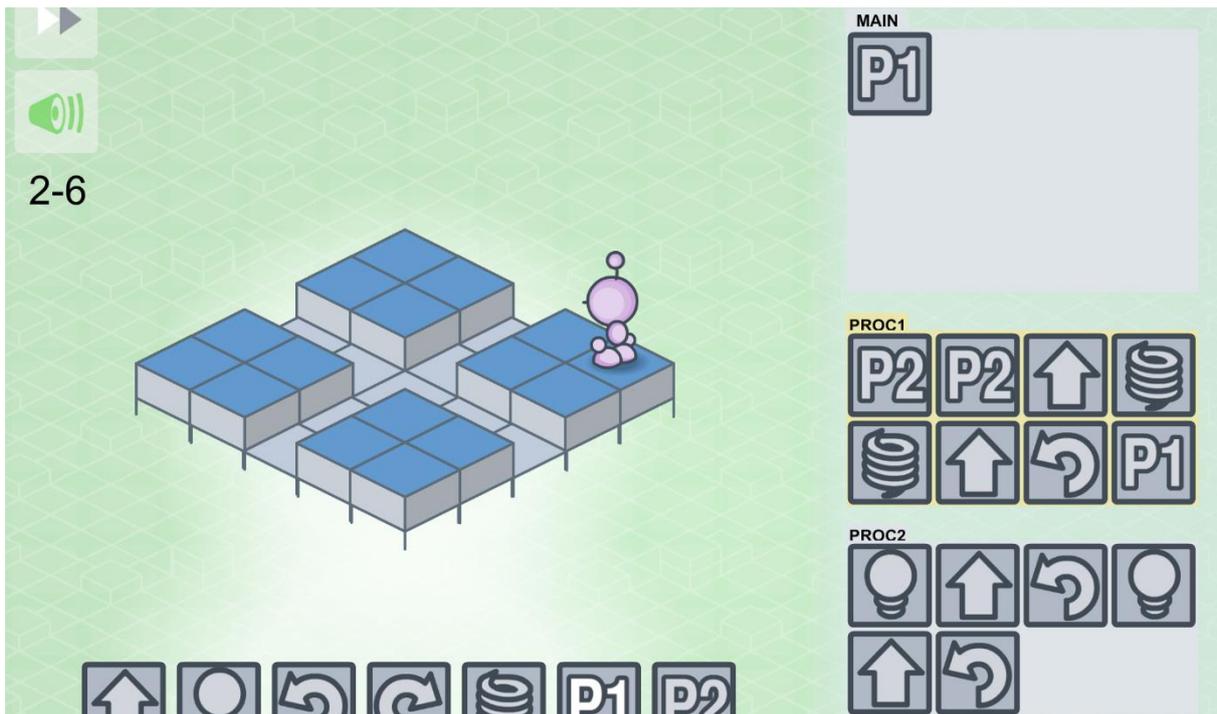
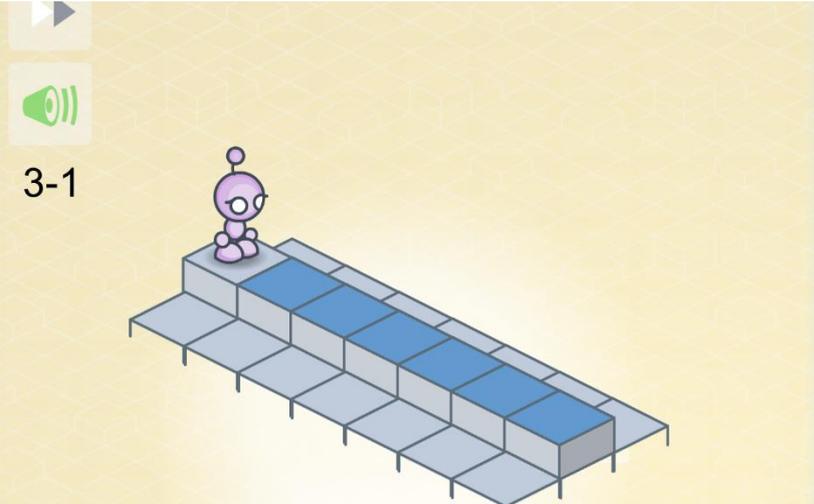


Abbildung 23: Lösung 2-6 mittels Schleife

### 4.3 Kapitel 3 - Schleifen

3-1



MAIN

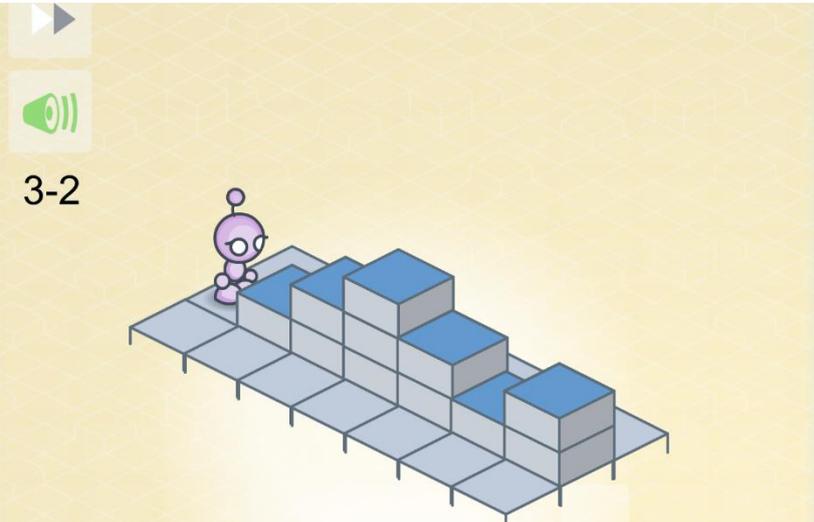
P1

PROC1

↑ ⚡ P1

Abbildung 24: Lösung 3-1

3-2



MAIN

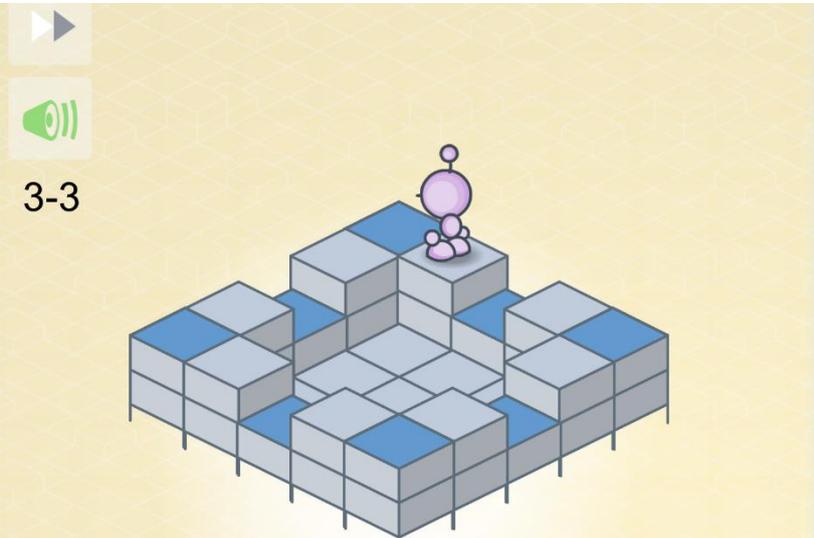
P1

PROC1

⌛ ⚡ P1

Abbildung 25: Lösung 3-2

3-3



MAIN

P1

PROC1

↑ ⚡ ↺ ↑

⌛ ⚡ ⌛ P1

Abbildung 26: Lösung 3-3

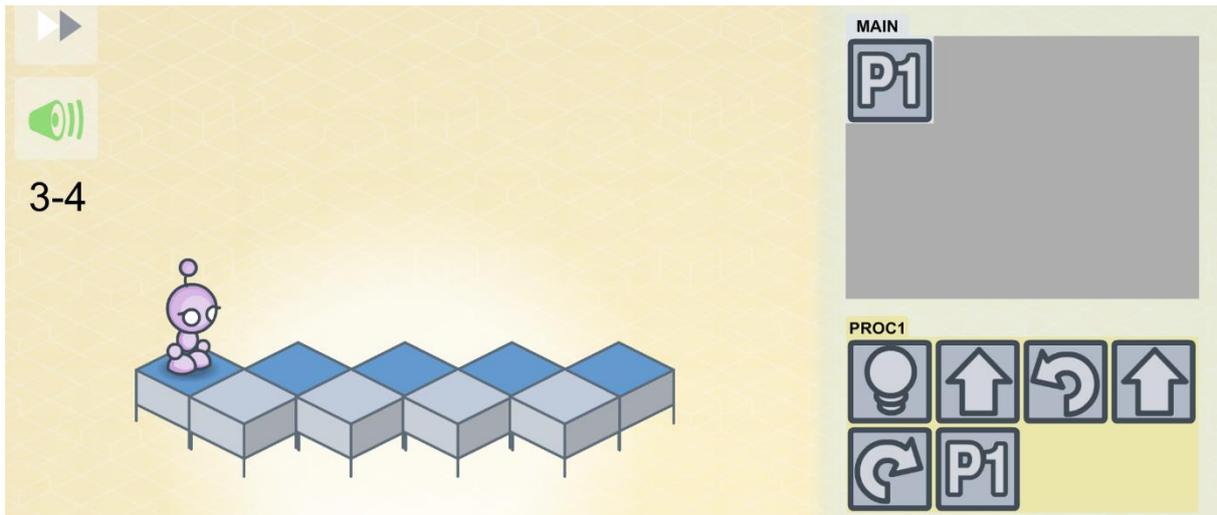


Abbildung 27: Lösung 3-4

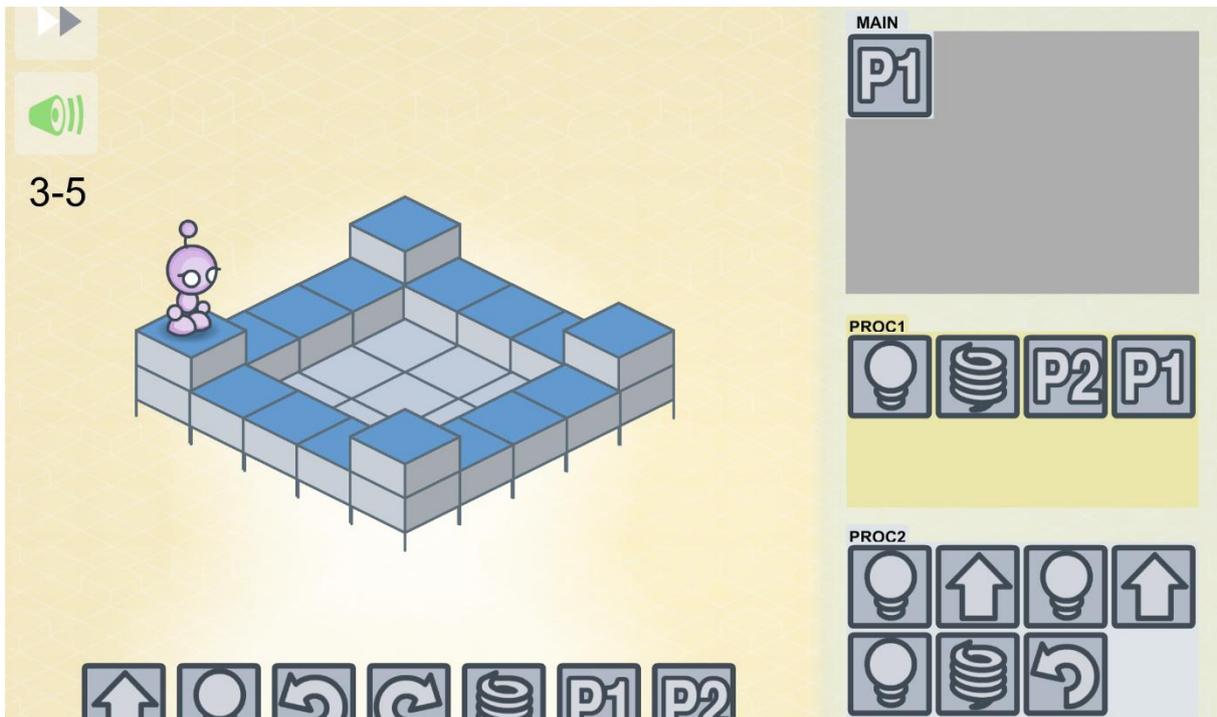


Abbildung 28: Lösung 3-5

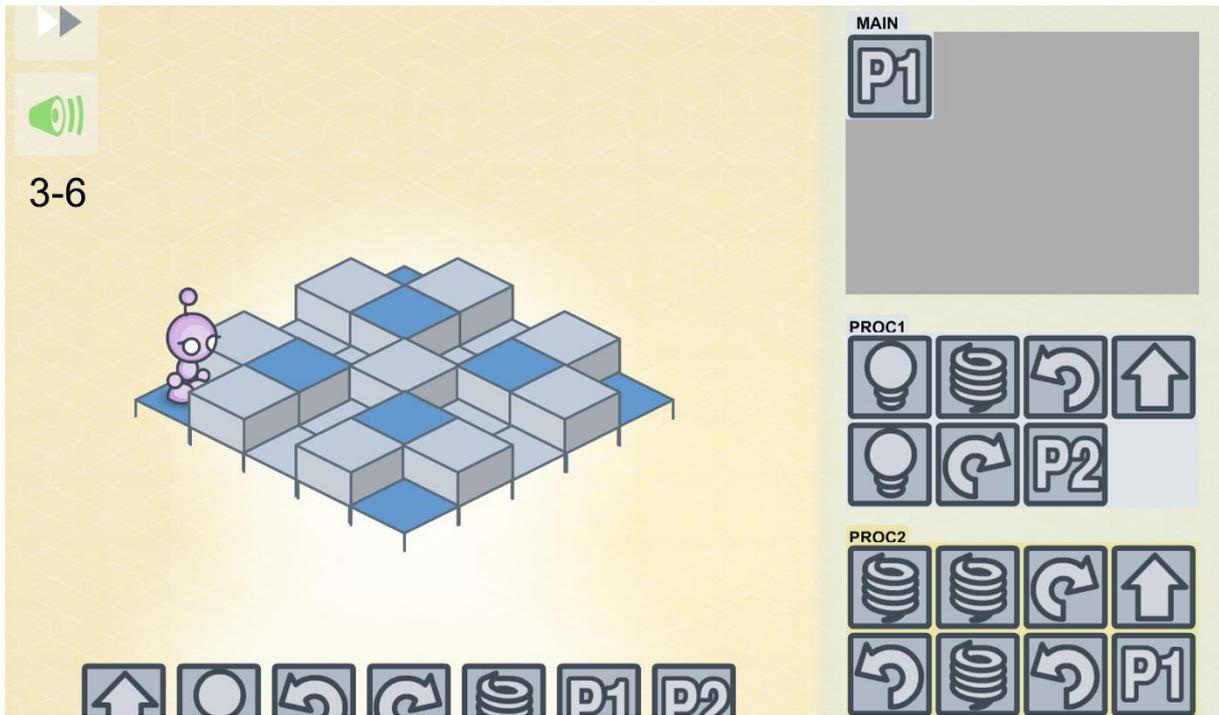


Abbildung 29: Lösung 3-6

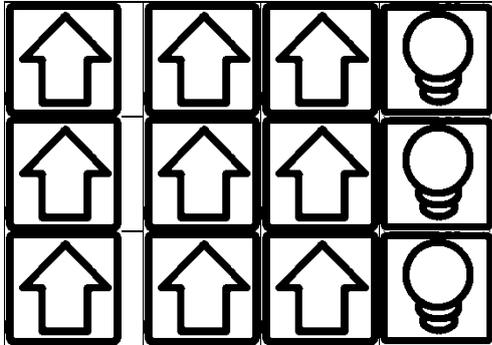
## 5 Übungsaufgaben:

1. Spielen Sie die Level durch.
2. Lassen Sie sich von Ihrem Nachbarn den Programmcode aus 2-2 erklären.
3. Finden Sie eigene Beispiele für Schleifen und Prozeduren in Ihrem Alltag.
4. Suchen Sie sich ein Level aus, schreiben ein Programm und lassen Ihren Nachbarn vorhersagen, wo die Figur stehen wird.
5. Optimieren Sie den Programmcode aus der Beispielaufgabe (s. Folgeseite).
6. Zeichnen Sie selbst einen Hindernisparcour (s. Kopiervorlage) und entwickeln Sie das passende Programm dazu.
7. Kneten Sie sich eine Roboterfigur und bauen Ihren Lieblingsparcour aus Holzwürfeln nach. Wie viele Würfel braucht man dazu?

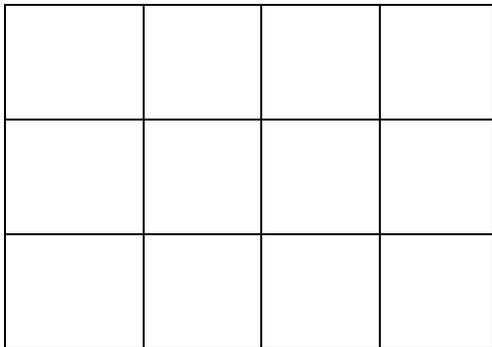
**Aufgabenbeispiel:** Wie könnte eine einfachere Prozedur aussehen?

a)

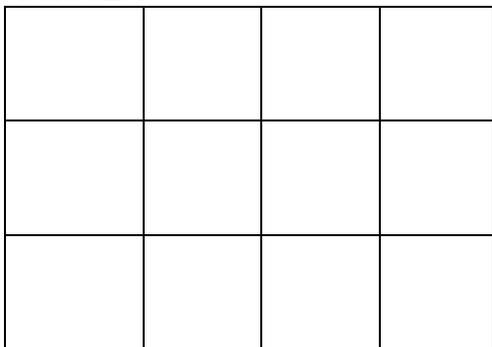
MAIN:



PROC1:

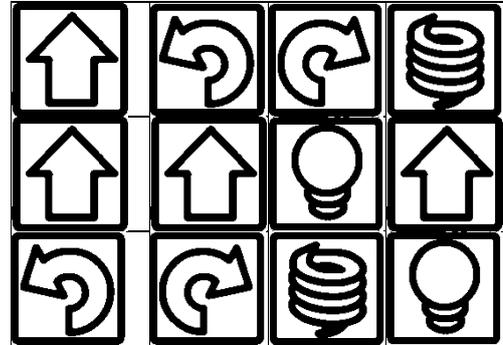


PROC2:

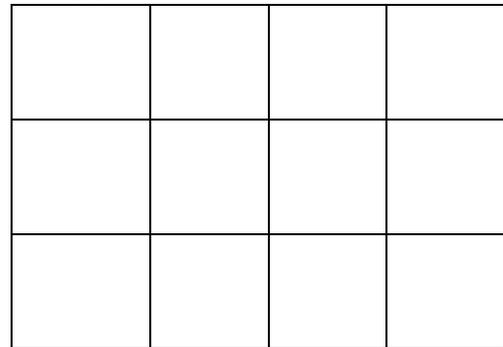


b)

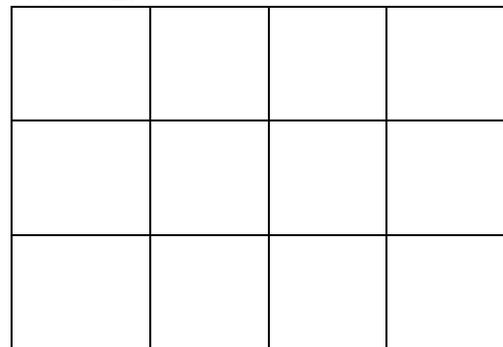
MAIN:



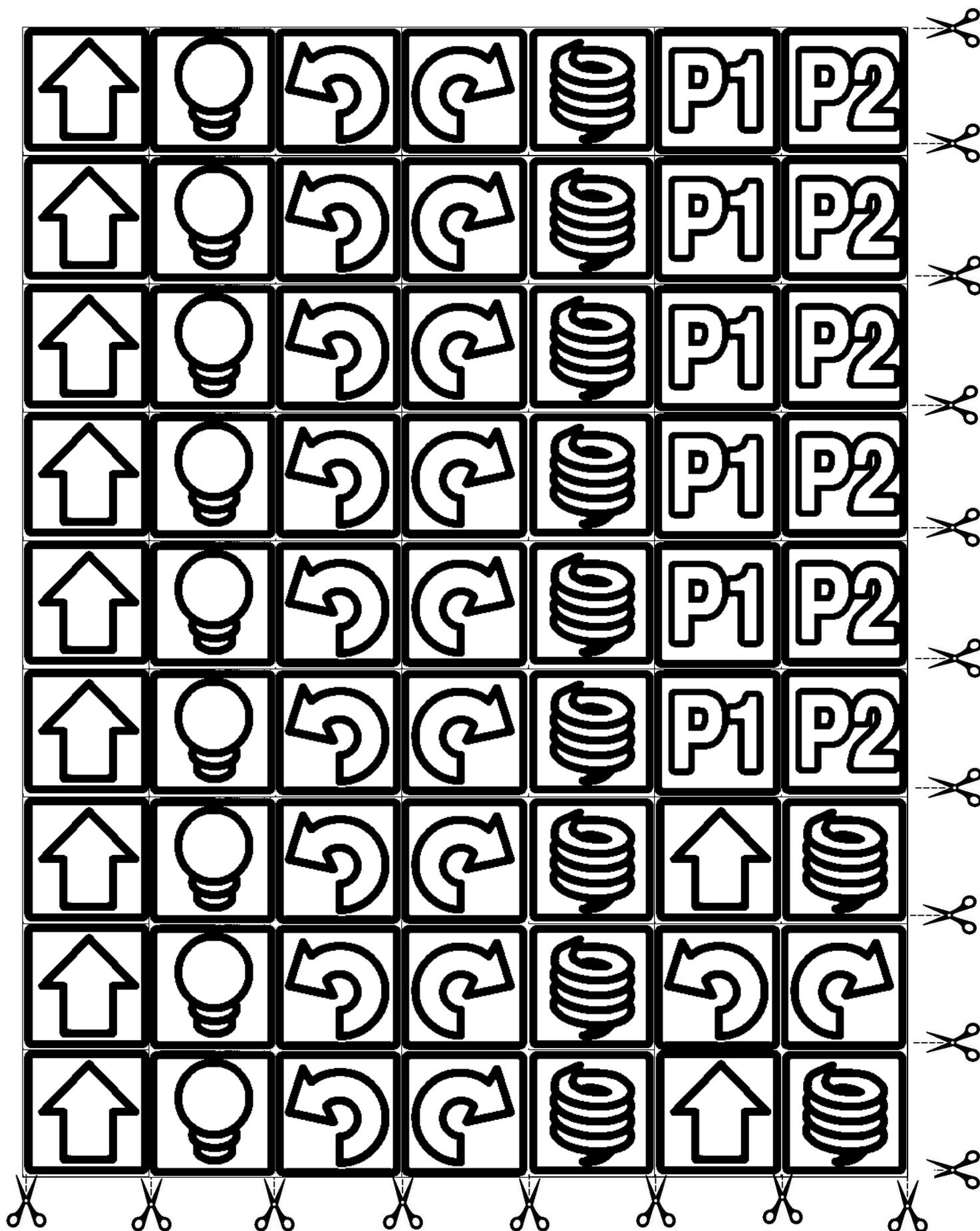
PROC1:



PROC2:







# Kopiervorlage Programmtabelle

für Aufgaben

MAIN:


PROC1:


PROC2:


MAIN:


PROC1:


PROC2:


# Kopiervorlage Programmtabelle

zum legen (Folgesseite)

**MAIN:**


**PROC1:**


**PROC2:**


**Kopiervorlage Isopapier: Level selbst zeichnen**

