






Einheit 7

Listen in Ozoblockly

Infos für LehrerInnen

In dieser Einheit beschäftigen sich die Lernenden mit dem Datentyp „Liste“ in Ozoblockly. Dabei lösen sie eine fortgeschrittene Programmieraufgabe unter Verwendung von Schleifen, Variablen und bedingten Anweisungen.

INFO BOX bit evo

-  **Ozoblockly**
-  **7.Schulstufe (12-14 J.)**
-  **100 min.**

Lernziele

- ✓ Die Lernenden verstehen die Verwendung von Listen (als Datentyp). (Sie können Elemente in die Liste einfügen und Werte an einer bestimmten Stelle der Liste auslesen.)
- ✓ Die Lernenden verstehen den Einsatz von Variablen.
- ✓ Die Lernenden können den Timer einsetzen.
- ✓ Die Lernenden kennen verschiedene Abbruchbedingungen von Schleifen.
- ✓ (Optional) Die Lernenden verstehen die Verwendung von Unterprogrammen.

Benötigte Materialien

- ✓ Ozobots
- ✓ Aufgabenstellung („Ozoblockly - Listen“)
- ✓ Computer (mit Internetverbindung)



Stundenablauf

- ✓ Teilen Sie den SchülerInnen das Aufgabenblatt („Listen in Ozoblockly“) aus.
- ✓ Besprechen Sie gemeinsam die Aufgabenstellung.
 - Die SchülerInnen beschäftigen sich in dieser Einheit mit dem Datentyp „Listen“ in *Ozoblockly*.
 - Die Lösung erfordert die Verwendung von Schleifen, Variablen und des Timers.
 - Zur Durchführung der Aufgabe muss in Ozoblockly in Level 5 (Master) gewechselt werden.
- ✓ Sollten die Lernenden noch keine Erfahrungen mit *Ozoblockly* haben bzw. wurden die vorangegangenen Einheiten nicht durchgeführt, dann sollte diese Übung nicht vorgezogen werden.
- ✓ Besprechen Sie, je nach Vorwissen der SchülerInnen, die Rolle von Listen (oder allgemein Datenstrukturen) in der Programmierung und erklären Sie die entsprechenden Befehle zur Erstellung und Manipulation von Listen in *Ozoblockly*.
- ✓ Die SchülerInnen sollen abschließend die erstellten Programme auf der Strecke, die auf dem Aufgabenblatt abgebildet ist, testen.
- ✓ Fordern Sie die SchülerInnen auf, die Korrektheit der Programme im Praxistest zu überprüfen.

TIPPS

- ✓ Vergewissern Sie sich im Vorhinein, ob die Lernenden die vorausgesetzten Konzepte auch tatsächlich verstanden haben.
- ✓ Überprüfen Sie im Vorhinein, ob der Ozobot die Farben des Ausdrucks auch korrekt erkennt.
- ✓ Überprüfen Sie die Korrektheit der erstellten Programme bzw. lassen Sie die Lernenden die Programme gegenseitig vorführen und auf Korrektheit prüfen.
- ✓ Bei Problemen mit *Ozoblockly* können Sie folgendes Video als Hilfestellung heranziehen: <https://youtu.be/RucE5oZTW2c>



Einheit 7

Listen in Ozoblockly

In dieser Übung geht es um Listen, genauer gesagt um den Datentyp „Liste“ in der Programmierung. Daten können in Form einer Liste im Speicher abgelegt werden. Das ist in vielen Fällen, insbesondere bei der hier vorgegebenen Strecke und Aufgabe, sehr praktisch.

Zum Lösen der nachfolgenden Aufgabe sollst du Listen in *Ozoblockly* verwenden.

- ✓ Erstelle dazu ein Programm in Ozoblockly.
- ✓ Solange der Ozobot fährt, soll bei jeder Kreuzung, zu der der Ozobot kommt, die Farbe der Kreuzung in einer Liste gespeichert werden.
- ✓ Wenn der Ozobot zum Stehen kommt, dann soll der Ozobot die Farben der Kreuzungen, die er passiert hat, in der korrekten Reihenfolge ausgeben.

Deine Aufgabe:

Schreibe nun drei verschiedene Programme (a-c) mit den folgenden Stopp-Kriterien:

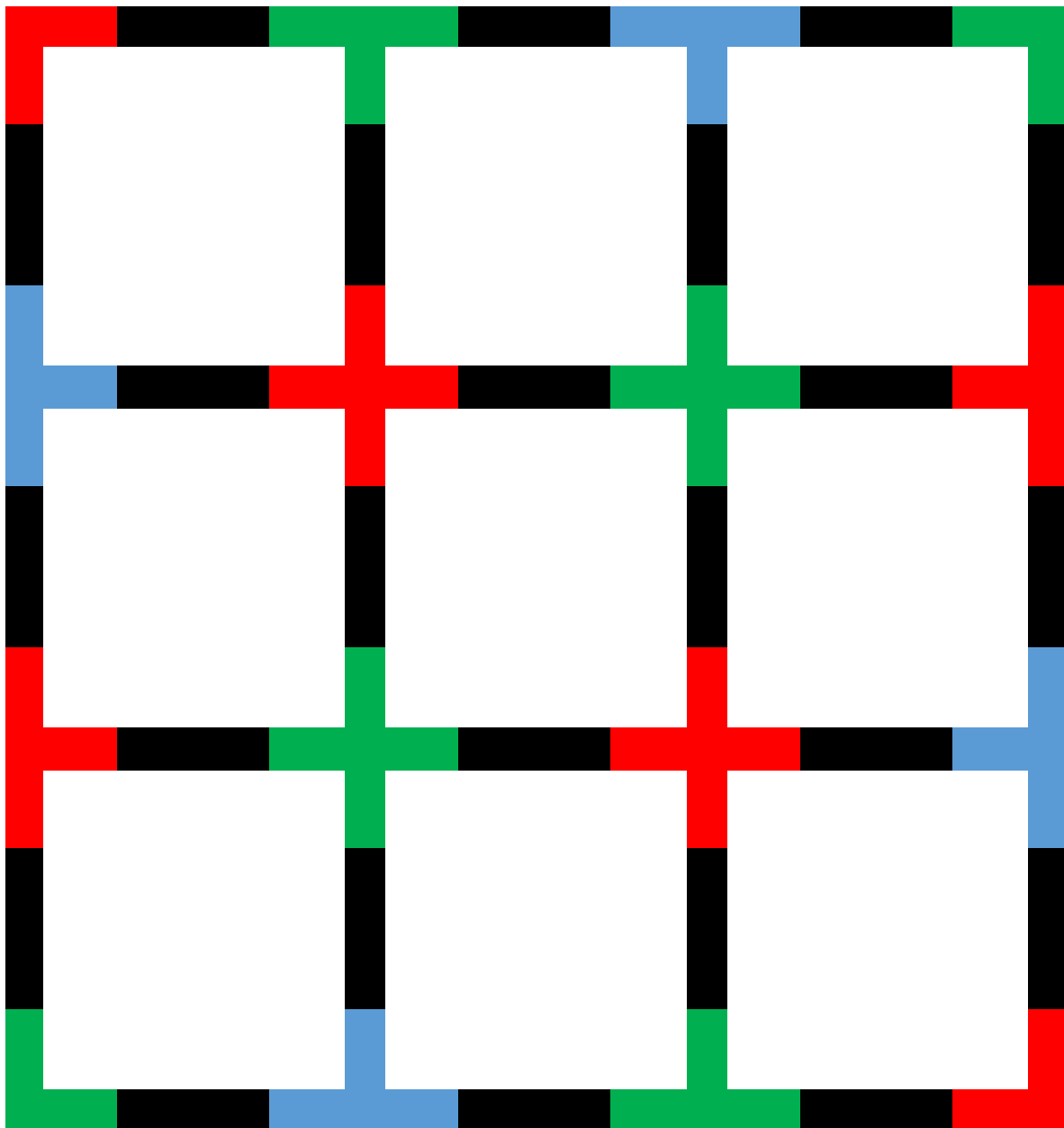
- a) Der Ozobot soll nach einer Minute stoppen.
- b) Der Ozobot soll nach 15 passierten Kreuzungen stoppen.
- c) Der Ozobot soll dann stoppen, wenn er 3 rote Kreuzungen passiert hat.

Beachte die folgenden Punkte:

- ✓ *Ozoblockly* muss mindestens in Level 5 (Master) verwendet werden.
- ✓ Das erste Element der Liste hat den Index 0.
- ✓ Für Aufgabe c) musst du auch über Variablen in der Programmierung Bescheid wissen.
- ✓ Die Ausgabe der Farbreihenfolge kann mit der LED des Ozobot gemacht werden. Hast du einen Ozobot EVO, so kannst du auch die Soundausgabe zur Aufgabe der Farbreihenfolge verwenden.



Einheit 7
Teste deinen Code hier!





Einheit 7 Listen in Ozoblockly - Lösungen

bit evo

```

to sayColors
  repeat while length of List ≠ 0
  do
    say color get element 0 from List
    remove element at index 0 from List
  
```

bit evo

```

to showColors
  turn top light off
  repeat while length of List ≠ 0
  do
    set top light color get element 0 from List
    wait 1 0 second(s)
    remove element at index 0 from List
    turn top light off
    wait 0 5 second(s)
  
```

Für a)

```

repeat while get timer value (s) < 10
do
  follow line to next intersection or line end
  append value get surface color to the end of List
sayColors
  
```

Für b)

```

repeat while length of List < 15
do
  follow line to next intersection or line end
  append value get surface color to the end of List
sayColors
  
```

Für c)

```

set red to 0
repeat while red < 3
do
  follow line to next intersection or line end
  append value get surface color to the end of List
  if get surface color = surface color red
  do
    change red by 1
sayColors
  
```

