

OZOBOT IN DER GRUNDSCHULE

WAS BRINGT DAS AM BK?

CODING offline | online

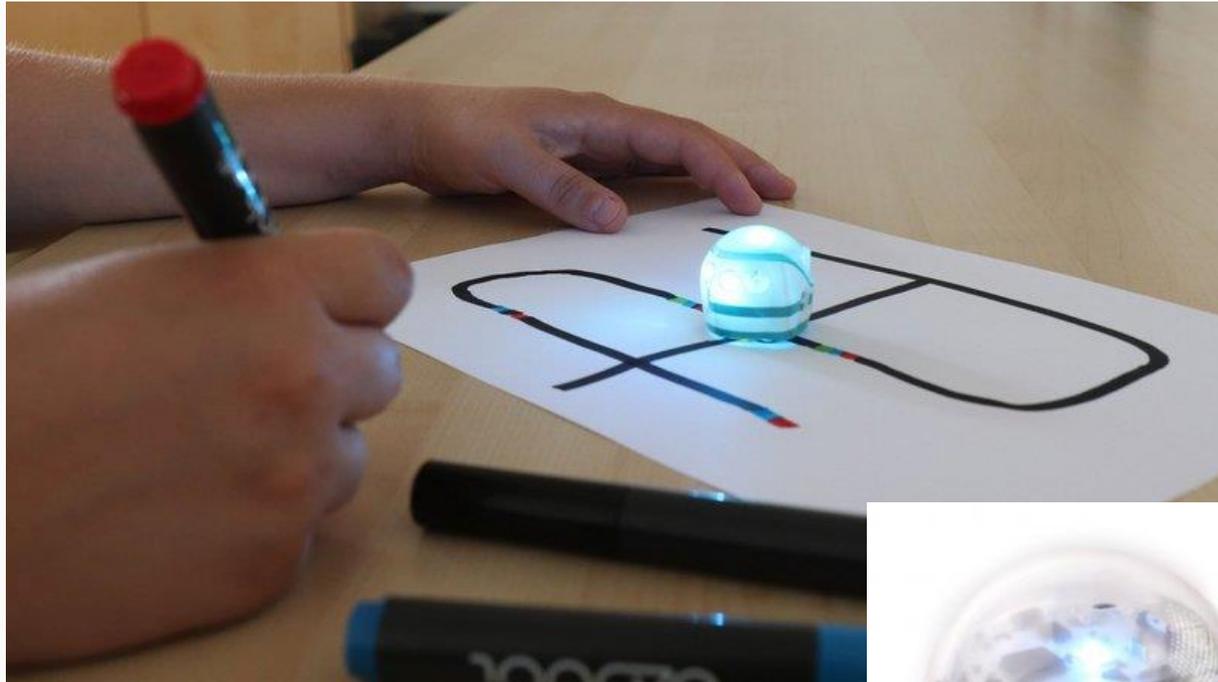
Sandra.Mueller@brk.nrw.schule
Tom.Schardt@brk.nrw.schule
05/2024

Ozobot Modul 1: Offline-Coding

- 1 Vorstellung des Ozobot
- 2 Kurze Einführung
- 3 Ausprobieren mit Demo-Material
- 4 Mögliche Aufgabenstellungen
- 5 Konkrete Einbindung in den Unterricht?



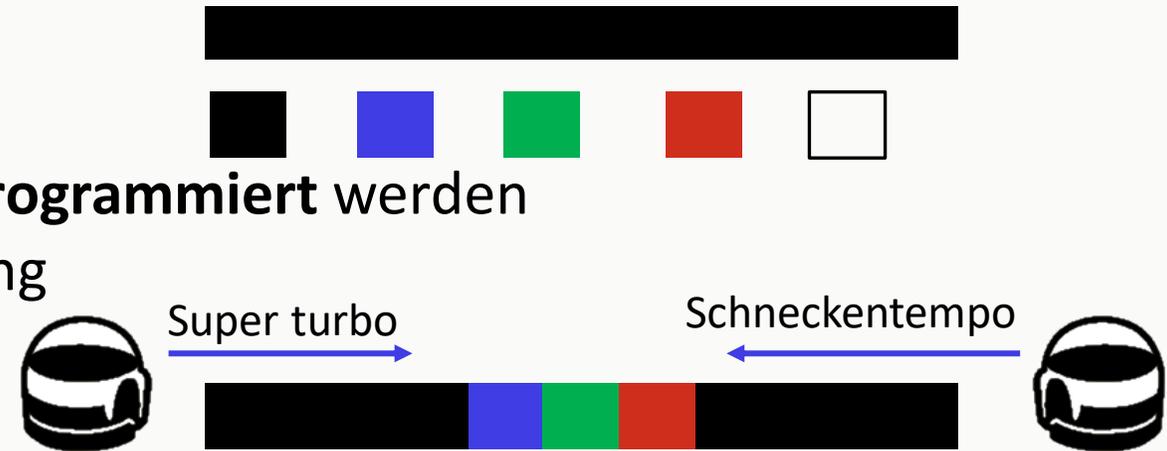
Vorstellung des Ozobot



- Spiel-Roboter für den Grundschulbereich
- Sensoren (Untergrundfarbe, Linien)
- Aktoren (Lampen, Bewegungen)
- **Online-Modus** (nächstes Modul)
Echtes Programmieren über Web-Seite
- **Offline-Modus**
Linien verfolgen und auf Farbcodes reagieren
 - keine Vorkenntnisse erforderlich
 - keine spez. Technik erforderlich
 - Ein Stapel DinA4 Papier reicht aus

Kurze Einführung

- Der Ozobot folgt schwarzen Linien
- Er erkennt **fünf Farben**
dadurch kann er mit Hilfe von **Farbcodes programmiert** werden
- Er liest die Farbcodes immer in Fahrtrichtung
- Start und Ende lassen sich markieren, d.h. der Ozobot erkennt anhand eines Codes, wann er am Ziel ist



- Der Ozobot muss ggf .bei Inbetriebnahme und bei Veränderungen der Lichtverhältnisse kalibriert werden
- Farbcodes werden am besten bei Tageslicht gelesen
- Beim Zeichnen der Codes müssen bestimmte Vorgaben eingehalten werden

Ausprobieren mit dem Demomaterial

- 3 Fertige einfache Aufgabenblätter mit Parkour
- 1 Karte mit Farbcodes
- Papier und Stifte zum Selberspielen ;)

Ozobot Projektideen phsz 1

Ozobot kennen lernen

Worum geht es?

Der Ozobot ist ein kleiner Roboter, der selbstständig dunklen Linien folgt. An Kreuzungen wählt er zufällig eine Abzweigung aus. An seiner Unterseite hat der Ozobot mehrere Sensoren, die wie kleine Kameras die Farbe des Untergrunds erkennen.



Was brauchst du?

- 1 Ozobot
- 1 weisses Papier (möglichst gross)
- 1 breiter schwarzer Filzstift (3-5mm)

Was lernst du?

- Du kennst das grundlegende Verhalten des Ozobots.
- Du kannst den Ozobot mit seiner Umgebung vertraut machen.
- Du kannst kreative Ideen bei der Gestaltung einer Fahrbahn für den Ozobot entwickeln und umsetzen.

Wie funktioniert es?

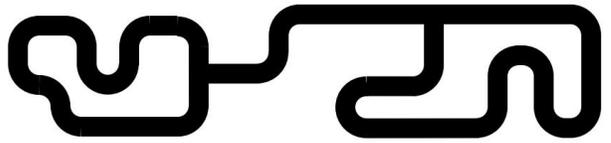
1. Ozobot in Betrieb nehmen
Sollte der Akku deines Ozobots leer sein, muss er zunächst über ein USB-Kabel aufgeladen werden. Der Ozobot hat nur einen einzigen Knopf zum An- und Abschalten an der Seite. Die eingebauten Sensoren des Ozobots sollten bei der ersten Verwendung immer mit der Umgebung vertraut gemacht werden (Kalibrierung). Dazu stellst du den Ozobot ausgeschaltet auf den schwarzen Kreis auf dieser Seite. Drücke dann die Taste am Ozobot für mindestens 5 Sekunden. Der Ozobot wird weiss blinken und danach selbstständig vom schwarzen Kreis herunterfahren. Wenn alles geklappt hat, wird er kurz grün blinken. Blinkt er rot, musst du die Schritte wiederholen. Wenn du den Ozobot auf einem Bildschirm (Tablet, Notebook) verwendest, stelle ihn auf eine freie weisse Fläche statt auf den schwarzen Kreis.

2. Folge der schwarzen Linie
Der Ozobot ist so gebaut, dass er immer versucht einer dunklen Linie nachzufahren. An Kreuzungen entscheidet er zufällig, wohin er fährt. (Weiter hinten in dieser Broschüre erfährst du, wie du dem Ozobot die Richtung vorgeben kannst.) Stelle deinen Ozobot auf die Fahrbahn am Ende dieser Seite und beobachte ihn, wie er der schwarzen Linie folgt.

Zeichne eine eigene Fahrbahn auf weisses Papier. Die Linien sollten in etwa so dick wie hier auf der Seite sein (ca. 5mm). Wenn die Linien zu dünn sind, wird dein Ozobot anhalten. Füge Kreuzungen ein und beobachte, wie sich der Ozobot verhält. Ihr könnt auch eine gemeinsame Fahrbahn zeichnen und eure Blätter zusammenlegen. Es können auch mehrere Ozobots gleichzeitig darauf fahren.



Ozobot hier kalibrieren



Mögliche Aufgabenstellungen für Farbprogrammierung

- Malen Sie zwei ähnliche Strecken auf und überlegen Sie, wie Sie mithilfe des Ozobots messen können, welche Strecke die längere ist
- Entwerfen Sie einen Endlosparkour mit mindestens einer Kreuzung, einer Sackgasse und einem Geschwindigkeitswechsel
Testen Sie anschließend, ob Ihr Parkour auch mit zwei Ozobots funktioniert
- „Würfeln“ mit dem Ozobot
- Irrgarten Challenge
Gestalten Sie einen beliebigen Irrgarten mit einem definierten Start und Ziel.
Überlegen Sie ohne Testen, wie Sie die Farbcodes einsetzen müssen, um vom Start zum Ziel zu gelangen und malen Sie die erforderlichen Farbcodes vorab auf Code-Vorlagen.
Testen Sie anschließend Ihre Lösung.

Tipp: Einsatz des IPad

- Kleine Ozobot-Welten lassen sich auch mit PowerPoint auf dem IPad gestalten und nutzen
- Keine Stifte und Papier nötig
- Somit ist auch ein Abspeichern der Welten möglich



Konkrete Einbindung in den Unterricht???

- Ist das witzig oder blöd?
- Gibt es neue Ideen?

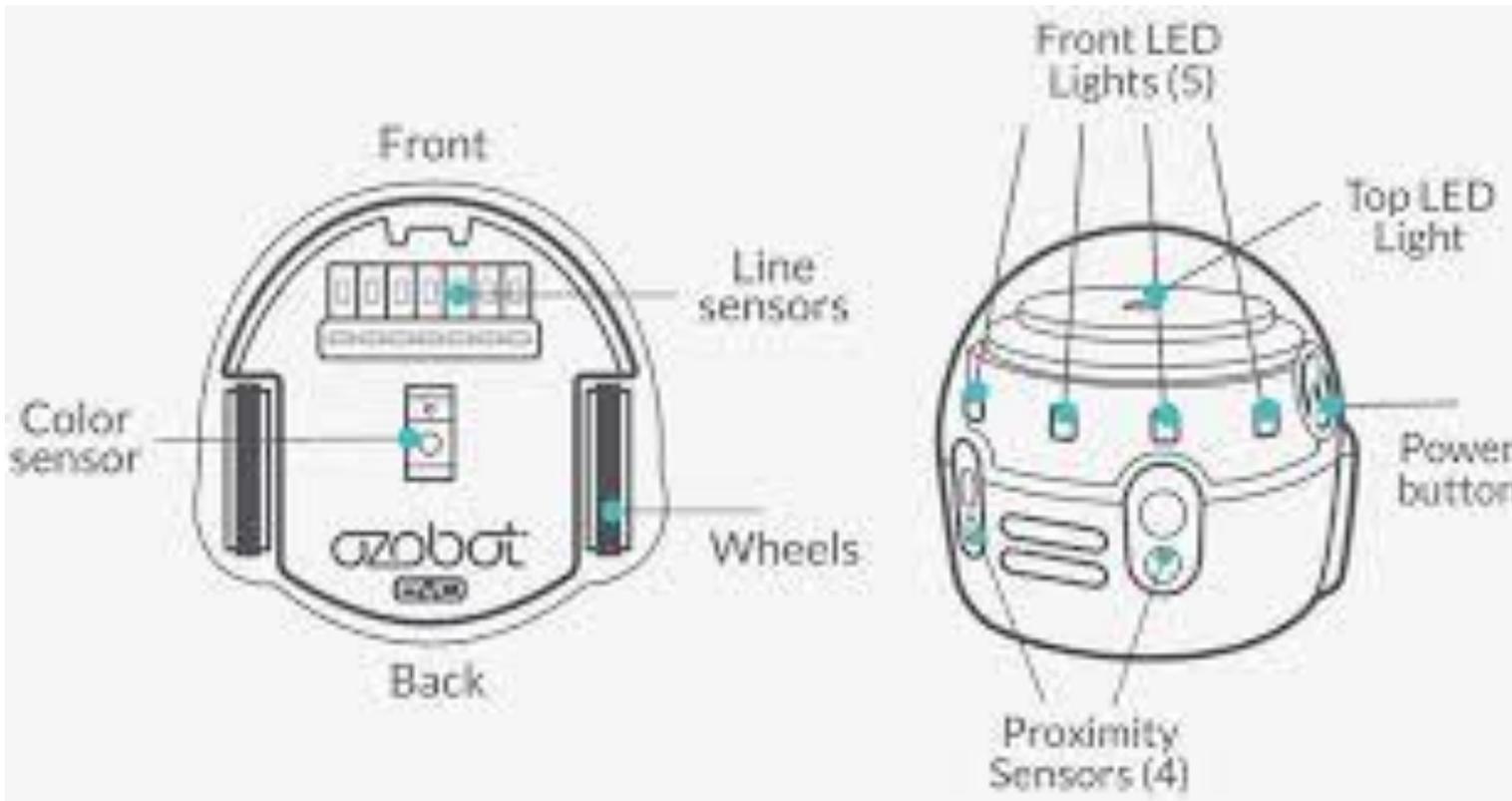


Ozobot Modul 2: Online-Coding

- 1 Vorstellung des Ozobot
- 2 Kurze Einführung
- 3 Ausprobieren mit Demo-Material
- 4 Mögliche Aufgabenstellungen
- 5 Konkrete Einbindung in den Unterricht?



Fähigkeiten des Ozobot Evo



Aktoren

- TopLED (Zustand/Farbe)
- FrontLED (Blinken)
- Töne
- Bewegung (Fahren)

Sensoren

- Erkennen von vier Farben
weiß, schwarz, rot, grün, blau
- Verfolgen von Linien
- Abstandssensor

Quelle: [OzoBlockly.com/editor](https://www.ozoblockly.com/editor)

„Freies Programmieren“ mit OzoBlockly.com/editor

The screenshot shows the OzoBlockly.com editor interface. On the left, there is a sidebar with the OzoBlockly logo, an 'Evo' button, and five difficulty levels (1-5). Below the levels are categories: Movement, Light Effects, Timing, Loops, and Sounds. The main workspace displays a program titled 'Lauf bis schwarz' with the following code blocks:

```
set top light color [white]
repeat forever
do
  move forward distance 1 step speed medium
  if (get surface color = surface color)
  do
    break out of loop
set top light color [red]
```

On the right side of the workspace, there are buttons for 'Programs' and 'Menu'.

- Befehle als Blöcke
- 5 Schwierigkeitslevel
- Befehle nehmen mit jedem Level zu
- Übertragung per Lichtsequenz
- Teilen der Programme per temp. Link oder lokal

Quelle: OzoBlockly.com/editor

Ozobot Blockly: Ipad & Web Aufgaben zum Üben

The screenshot shows the Ozobot Blockly interface. On the left, there is a sidebar with a close button (X), an 'Evo' button, and a 'Levels' section with a '2' button. Below this are several icons: a right arrow, a lightbulb, a clock, a refresh icon, and a speaker. The main workspace is titled 'Lauf bis schwarz' and contains a Blockly script with 'set to', 'repeat forever', 'do', and 'set to' blocks. A menu is open over the workspace, listing: Programs, Help, Glossary, Examples, Challenges, Code Preview, and Settings. The 'Challenges' section is active, showing three challenge cards:

- Rectangle Walk**
Level: 2 or higher
Program Ozobot to walk in a rectangle. The challenge is to make Ozobot move and its LED light up at certain points during the walk.
Show Instructions
- Slot Car Race Track**
Level: 3 or higher
Author: Prof. Richard Born
Slot car racing has been of interest to many for years and years. In this challenge you will be creating an ozoBlockly program that makes Ozobot Bit behave like a slot car. Create your program so that Ozobot will run five laps using the provided track.
Show Instructions Download Map
- Color Visit Counts**

At the bottom of the challenge section, there is a link: 'Ready to code a real robot? [Shop Ozobots](#)'

- Evo App
- Bereich Meta
- Virtuellen Parkour bauen

Aufgabe 1: 1 Schritt vor, 2 zurück (Level 1)

← ↻ 🏠 <https://ozoblockly.com/editor?lang=en&robot=evo&mode=1> A

Movement

Bit+ Evo
MetaBot

Levels
1

→

💡

🕒

🔊

The screenshot shows the Ozoblockly editor interface. On the left, there is a sidebar with a teal robot icon, a 'Levels' section with '1' selected, and several icons: a right arrow, a lightbulb, a clock, and a speaker. The main workspace is titled 'Movement' and contains a grid of yellow blocks. The first row has two blocks: one with two right-pointing chevrons and one with four right-pointing chevrons. The second row has four blocks, each with a right-pointing arrow and a number above it: 1, 2, 5, and 10. The third row has four blocks, each with a left-pointing arrow and a number above it: 1, 2, 5, and 10. The fourth row has three blocks, each with a circular arrow indicating a 90-degree turn. On the right side of the workspace, two blocks are stacked vertically. The top block has a right-pointing arrow with the number '1' above it. The bottom block has a left-pointing arrow with the number '2' above it.

Aufgabe 1: 1 Schritt vor, 2 zurück (Level 2)

← ↻ 🏠 <https://ozoblockly.com/editor?lang=en&robot=evo&mode=2> 🔊

Movement

- move forward distance 1 step speed medium
- ↻ rotate slight left
- ⚡ zigzag medium
- 🛹 skate medium forward
- 🌀 spin left
- 🕒 small circle medium forward left 1 second
- 🕒 big circle medium forward left 1 second

Bit+ Evo MetaBot Levels 2

→

💡

🕒

🔄

🔊

Lev

→ move forward distance 1 step speed medium

→ move backward distance 1 step speed medium

Aufgabe: TopLicht als Blaulicht (Level 1 - klappt nicht richtig)

← ↻ 🏠 <https://ozoblockly.com/editor?lang=en&robot=evo&mode=1> 🔊

Movement

Bit+ Evo
MetaBot

Levels
1

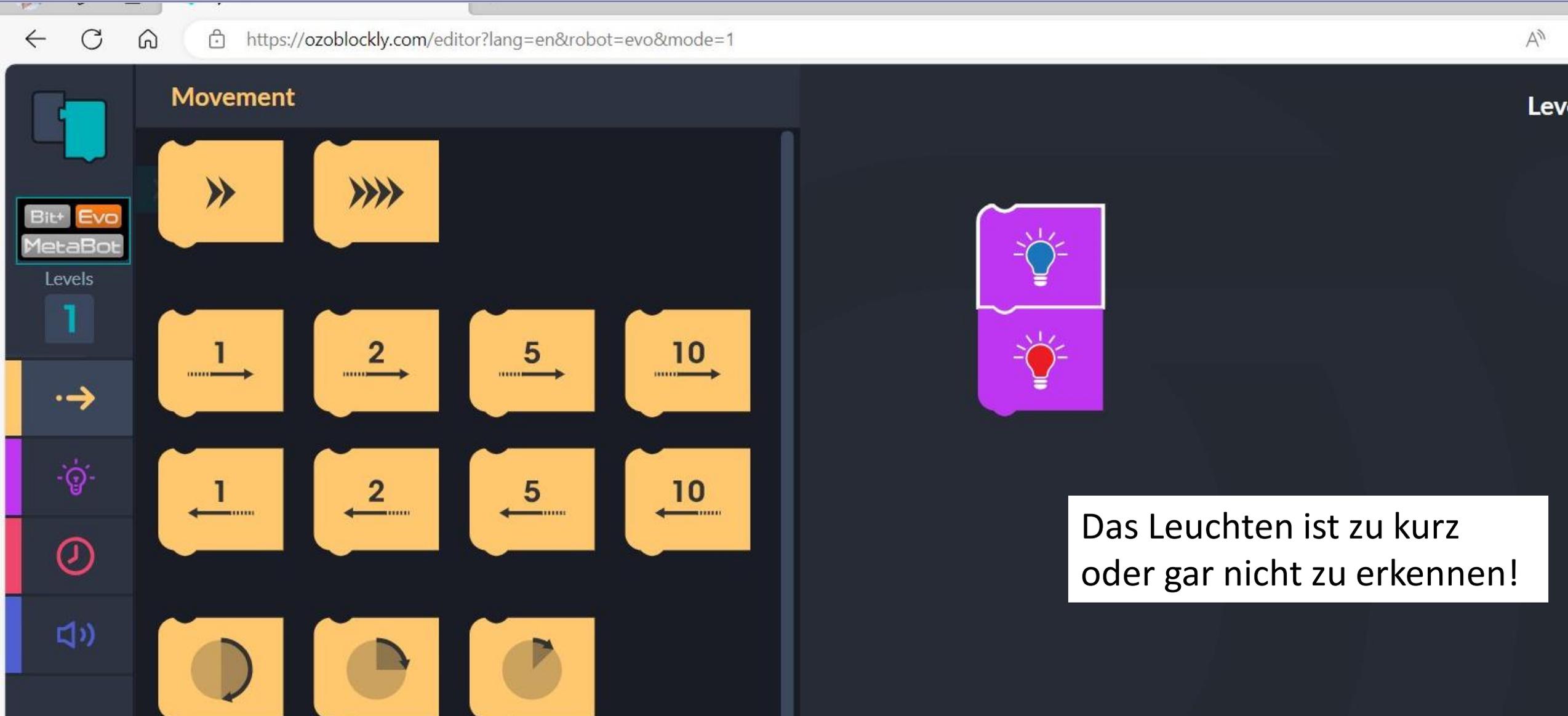
→

💡

🕒

🔊

Lev



The screenshot shows the Ozoblockly editor interface. On the left, there is a sidebar with a 'Levels' section set to '1'. Below it are icons for movement (arrow), light (lightbulb), time (clock), and sound (speaker). The main workspace is titled 'Movement' and contains a grid of yellow blocks: two blocks with double and triple right-pointing arrows, two rows of blocks with right-pointing arrows labeled '1', '2', '5', and '10', and two rows of blocks with left-pointing arrows labeled '1', '2', '5', and '10'. On the right side of the workspace, there are two purple blocks stacked vertically, each containing a lightbulb icon. The top lightbulb is blue, and the bottom one is red.

Das Leuchten ist zu kurz
oder gar nicht zu erkennen!

Aufgabe: TopLicht als Blaulicht (Level 1 - besser)



Light Effects

Bit+ Evo
MetaBot

Levels
1

→

Light Effects palette:

- Row 1: Red, Green, Blue lightbulbs
- Row 2: Yellow, Cyan, Magenta lightbulbs
- Row 3: Drum, Traffic light, Christmas tree
- Row 4: Rainbow, Fireworks, Moon

Script stack:

- Blue lightbulb
- Timer 1
- Red lightbulb
- Timer 1

Es blinkt nur ein Mal...

Aufgabe: TopLicht als Blaulicht (Level 2 - richtig mit Schleife)

← ↻ 🏠 <https://ozoblockly.com/editor?lang=en&robot=evo&mode=2> 🔊

OZO Blockly

Bit+ Evo
MetaBot

Levels

1 2 3 4 5

→ **Movement**

💡 **Light Effects**

🕒 **Timing**

🔄 **Loops**

🔊 **Sounds**

Loops

- repeat forever
do
- repeat 2 times
do

turn top light off

repeat forever

do

- 💡 set top light color [red]
- 🕒 wait 1 second(s)
- 💡 set top light color [blue]
- 🕒 wait 1 second(s)

turn top light off

Aufgabe: TopLicht als Blaulicht (Level 5 - toll, mit Abschalten)

The screenshot shows the Ozoblockly editor interface. The browser address bar displays <https://ozoblockly.com/editor?lang=en&robot=evo&mode=5>. On the left sidebar, the robot is identified as 'Evo' and the current level is '5'. The main workspace contains the following script:

- when green flag clicked** (implied start block)
- if object in front** (green flag block)
- capture button press events** (green flag block) set to **true**
- set button press count** (green flag block) set to **0**
- repeat forever** (blue loop block) containing:
 - do** (purple block) containing:
 - set top light color** (purple block) set to **red**
 - wait** (red block) set to **1** **second(s)**
 - turn top light off** (purple block)
 - set top light color** (purple block) set to **blue**
 - wait** (red block) set to **1** **second(s)**
 - turn top light off** (purple block)
- if get button press count > 0** (green flag block)
- do terminate program and switch to idle** (red block)

Python Programmierung mit beta.python.ozobot.com

The screenshot shows a web browser window with the URL `beta.python.ozobot.com`. The page title is "tree-maze-letter". The interface includes a toolbar with buttons for "Run Python code (without a robot)", "tree_DFS", "Open", and "Download". A dropdown menu for "Quick-start with a template:" is set to "Tree traversal (DFS)".

```
1 import ozobot
2 """
3 Ozobot demonstrates a Depth First traversal while navigating a binary tree maze.
4
5 Prerequisite: print or draw the map at https://ozoblockly.com/resource/maps/tree-maze-letter.pdf
6
7 Note: There is a bug in latest Evo FW with intersection detection on thick lines. As a workaround, make sure the line width is
8 exactly 4mm. This can be achieved by printing the track in 80% scale.
9
10 Start Ozobot Evo at the root of the tree and it will methodically explore all of the tree branches
11 and return to the starting point. Then the program will report the number of leaves
12 Ozobot visited while traversing the maze.
13 """
14
15 robot = ozobot.get_robot()
16
17 def searchSubtree(intersection):
18     """Depth First traversal in a binary tree"""
19     if (intersection == ozobot.Directions.BACK.value):
20         # Robot keeps track of visited leaves (line ends)
21         robot.navigation.navigate(ozobot.Directions.BACK)
22         return 1
23     else:
24         # currently, the algorithm only supports intersections of type "T", as in the example maze
```

Python Programmierung mit beta.python.ozobot.com

Quick-start with a template:

Tree traversal (DFS) ▾

- please choose any -

Set Lights example

Square walk example

Music example

Concurrency example

Tree traversal (DFS)

Python Programmierung mit beta.python.ozobot.com

Run Python code (without a robot)

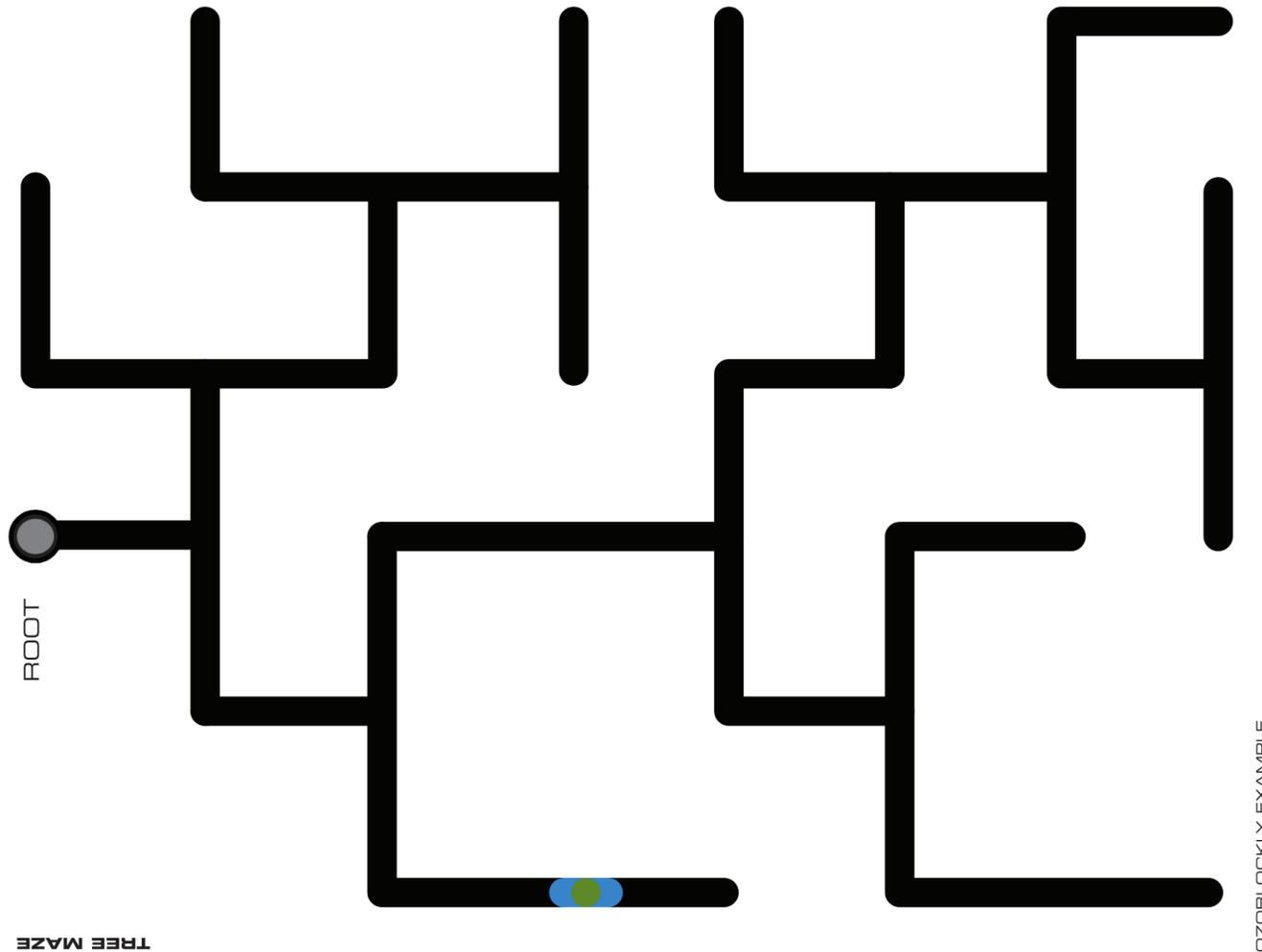
setLED

Open

Download

```
1  import ozobot
2
3  r = ozobot.get_robot()
4
5  # This program runs indefinitely. To stop it, press the "Restart Python" button in top-left corner.
6  while True:
7      r.light_effects.set_light_color( ozobot.SurfaceColor.RED, ozobot.Lights.ALL_FRONT)
8      ozobot.time.sleep(1)
9      r.light_effects.set_light_color_rgb( 1.0, 1.0, 1.0, ozobot.Lights.ALL_FRONT )
10     ozobot.time.sleep(1)
11     r.light_effects.set_light_color_rgb( 0, 0, 0, ozobot.Lights.FRONT_1 | ozobot.Lights.FRONT_5)
12     ozobot.time.sleep(0.3)
13     r.light_effects.set_light_color_rgb( 0, 0, 0, ozobot.Lights.FRONT_2 | ozobot.Lights.FRONT_4)
14     ozobot.time.sleep(0.3)
15     r.light_effects.set_light_color_rgb( 0, 0, 0, ozobot.Lights.FRONT_3)
16     ozobot.time.sleep(1)
```

Python Programmierung: Tiefen-Suche



- Darstellung auf dem Ipad
- Bildschirmhelligkeit maximal

Mögliche Aufgaben mit OzoBlockly (Level 1, 2)

Browser Edge: URL <https://ozoblockly.com/editor>

1. Lass das Toplicht abwechselnd rot und blau blinken
Tipp: Nutze den Timer, um das Licht lange genug leuchten zu lassen
2. Fahre einen Schritt vor, und zwei zurück
Tipp: Nutze den Code aus Aufgabe 1, um in Programm 2 anzuzeigen, wenn das Programm zuende ist
3. Fahre in einem großen Rechteck
Frage: Was musst Du tun, um das Rechteck zu vergrößern?

Mögliche Aufgaben mit OzoBlockly (Level 3-5)

4. Gefängnis: Male einen großen schwarzen Kreis und lasse Ozobot solange in eine Richtung fahren, bis die Grenze erreicht ist.
Wie dick muss man die Grenze malen, damit Ozobot sie erkennt.
Tipp: Mache Dir vorher Notizen, welche Befehle Du für diese Aufgabe nutzen möchtest und bringe Sie in die richtige Reihenfolge.
5. Erweitere Aufgabe 4 so, dass das Toplight des Ozobots immer die Farbe des Untergrundes anzeigt.
6. Mars Expedition: Male einen großen schwarzen Kreis und einige rote und blaue Felder am inneren Rand des Kreises. Der Ozobot soll nun erkunden, welche Farbe der Untergrund am Rande der Fläche hat. Dafür soll er einfach in eine Richtung losfahren, bis zum Rand, erkennen welche Untergrundfarbe dort herrscht, in dieser Farbe die Lampe anschalten und zurückkommen.
Erweiterung: Wie weit weg ist der Rand eigentlich?

Mögliche Aufgaben mit OzoBlockly (Level 3-5)

4. Gefängnis: Male einen großen schwarzen Kreis und lasse Ozobot solange in eine Richtung fahren, bis die Grenze erreicht ist.
Wie dick muss man die Grenze malen, damit Ozobot sie erkennt.
Tipp: Mache Dir vorher Notizen, welche Befehle Du für diese Aufgabe nutzen möchtest und bringe Sie in die richtige Reihenfolge.
5. Erweitere Aufgabe 4 so, dass das Toplight des Ozobots immer die Farbe des Untergrundes anzeigt.
6. Mars Expedition: Male einen großen schwarzen Kreis und einige rote und blaue Felder am inneren Rand des Kreises. Der Ozobot soll nun erkunden, welche Farbe der Untergrund am Rande der Fläche hat. Dafür soll er einfach in eine Richtung losfahren, bis zum Rand, erkennen welche Untergrundfarbe dort herrscht, in dieser Farbe die Lampe anschalten und zurückkommen.
Erweiterung: Wie weit weg ist der Rand eigentlich?