



Ozobots und mehr ...

**Workshop beim Informatiklehrer-Tag Bayern
am 19. September 2024
an der TU München**

Dr.-Ing. Ulrich Kiesmüller
Simon-Marius-Gymnasium Gunzenhausen
ulrich.kiesmueller@smg-gun.de

„Was du mir sagst, das vergesse ich.

Was du mir zeigst, daran erinnere ich mich.

Was du mich tun lässt, das verstehe ich.“

Konfuzius (551 - 479 v. Chr.), chinesischer Philosoph

- Die Schülerinnen und Schüler setzen unter sinnvoller Nutzung algorithmischer Bausteine einfache Algorithmen mithilfe geeigneter Programmierwerkzeuge um.

(ISB: LehrplanPLUS (Gymnasium) – Natur und Technik 7 – Informatik)

- Fachbegriffe: Algorithmus, Anweisung, Sequenz, ein- und zweiseitig bedingte Anweisung, Wiederholung mit fester Anzahl, Wiederholung mit Bedingung

(ISB: LehrplanPLUS (Gymnasium) – Natur und Technik 7 – Informatik)

Robot Karol

- Strukturelemente zusammenklickbar
- Programmcode aus viel Text
- keine selbst definierbaren Variablen
- Struktogramm generierbar

„... wir ham da was mit so nem doofen Roboter gemacht – was, weiß ich nicht ... – ... er konnte jedenfalls nix ...“

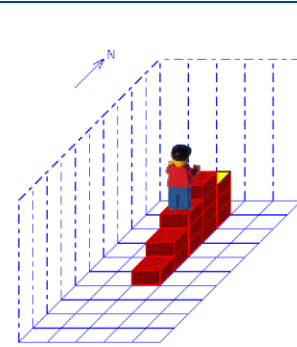
```
Anweisung Umdrehen
  LinksDrehen
  LinksDrehen
  LinksDrehen
*Anweisung

Anweisung ZweiSchritt
  Schritt
  Schritt
*Anweisung

Anweisung EineSchichtLegen
  wiederhole solange NichtIstMarke
    Hinlegen
    Schritt
  *wiederhole
*Anweisung

Anweisung HinabSteigen
  wiederhole solange IstZiegel
    Schritt
  *wiederhole
*Anweisung

Programm
  MarkeSetzen
  wiederhole 7 mal
    ZweiSchritt
    Umdrehen
    EineSchichtLegen
    Umdrehen
    HinabSteigen
  *wiederhole
  Schritt
  Umdrehen
*Programm
```



Hauptprogramm

MarkeSetzen
wiederhole 7 mal
ZweiSchritt()
Umdrehen()
EineSchichtLegen()
Umdrehen()
HinabSteigen()
Schritt
Umdrehen()

Anweisung/Methode: Umdrehen()

LinksDrehen
LinksDrehen

Anweisung/Methode: ZweiSchritt()

Schritt
Schritt

Anweisung/Methode: EineSchichtLegen()

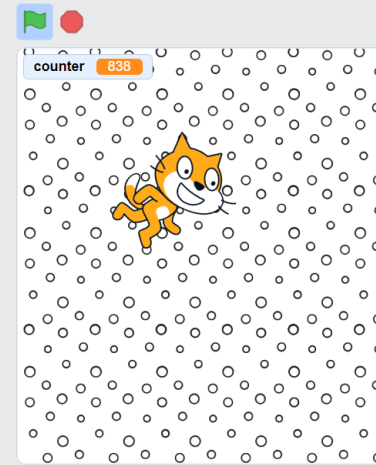
wdh. solange NichtIstMarke
Hinlegen
Schritt

Anweisung/Methode: HinabSteigen()

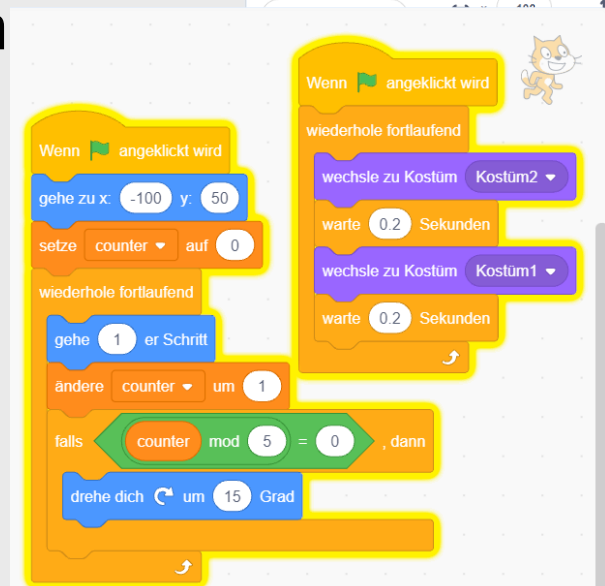
wdh. solange IstZiegel
Schritt

Scratch

- Blockly-Programmierung per drag&drop
- selbst gestaltbare Figuren/Hintergründe
- kreative Möglichkeiten
- parallele Ausführung von Programmteilen
- selbst erstellbare Programmierbausteine



Figur



Mit zusätzlichen Geräten

- Einstieg über Robotik ("alle zeichnen einen Roboter")
- Begriffsfestlegung:

*Ein **Roboter** ist eine*

- *technische Apparatur*
- *die dazu dient, Menschen bei häufig wiederkehrenden (mechanischen) Arbeiten zu unterstützen.*
- *Hierbei haben sie über **Sensoren** (z. B. für Wärme, (Ultra-)Schall, UV-Licht) Kontakt mit ihrer Umwelt.*
- ***Aktoren** (z. B. Motoren, Displays) dienen zur Bewältigung der Aufgaben.*
- *Sie werden gesteuert von Algorithmen.*

- Einstieg über Robotik
- Übergang zur Algorithmik
- Begriffsfestlegung:

*Ein **Algorithmus** ist eine*

- *endliche Folge von*
- *eindeutigen Anweisungen, die bei*
- *einheitlichen Voraussetzungen zu einem*
- *eindeutigen Ergebnis führen.*

Anweisung

einen Schritt geradeaus vor

Sequenz

einen Schritt geradeaus vor

einen Schritt geradeaus vor

drehe um 180° nach rechts

einen Schritt geradeaus vor

Bedingte Anweisung

Wand?	
ja	nein
drehe 90° rechts	Schritt nach vorne

Wiederholung (mit Zähler)

Zähle von	1	bis	3
Schritt nach vorne			

Wiederholung (mit Bedingung)

Solange nichtWand erfüllt, tue
Schritt nach vorne

Calliope Mini

- Blockly-Programmierung per drag&drop mittels Onlineanwendung MakeCode
- Vorschaumöglichkeit



Microsoft | CALLIOPE mini

Blöcke JavaScript

Suche...

- Grundlagen
- Eingabe
- Musik
- LED
- Schleifen
- Logik
- Variablen
- Mathematik
- Motoren

dauerhaft

zeige Symbol

pausiere (ms) 100

zeige Symbol

Calliope Mini

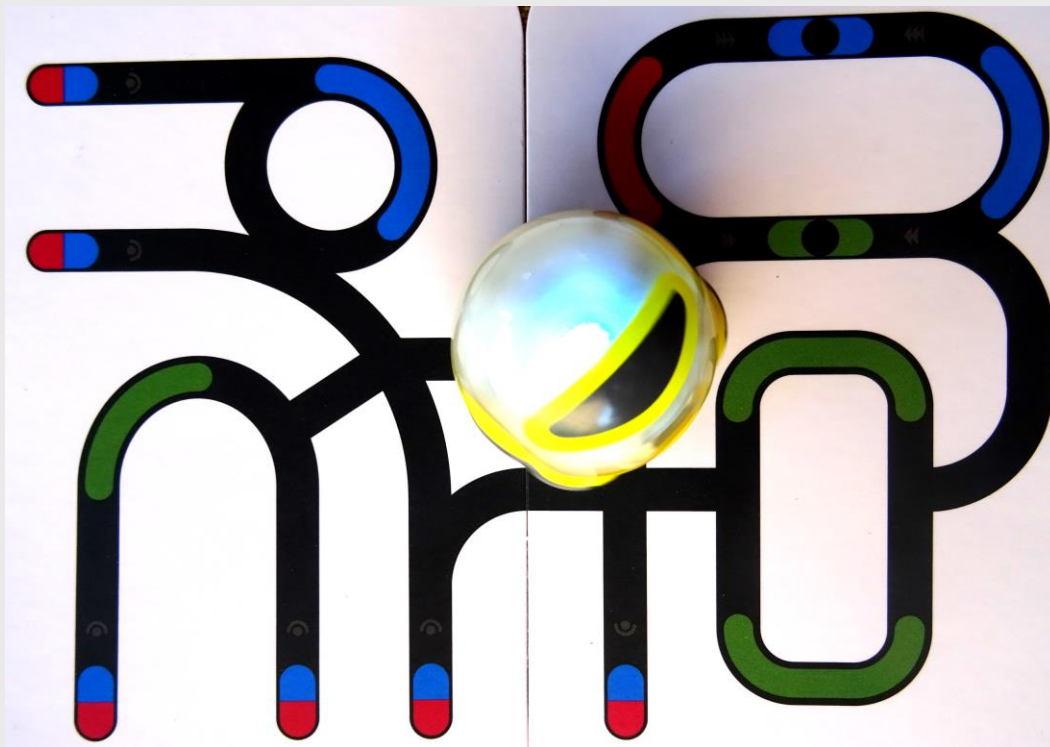
- Blockly-Programmierung per drag&drop mittels Onlineanwendung MakeCode
- Übersetzungsmöglichkeit in Python- oder Java-Script
- Vorschaumöglichkeit (auch Tasten, Pins, Schüttelsensor)
- Programmübertragung mittels USB-Verbindung
- Eingabemöglichkeiten: Taste, Kontakt, Sensor
- Ausgabemöglichkeiten: Zahl, Text, LED, LED-Feld, Sound
- Erweiterungsmöglichkeiten

Ozobot

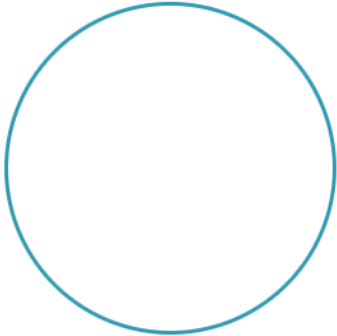
- Programmierung über Farbcodes
- Blockly-Programmierung per drag&drop mittels Onlineanwendung [OzoBlockly](#)
- mit [Simulator](#) lösbare [Online-Aufgaben](#) (nur Sequenz und Wiederholung mit Zähler)
- Programmübertragung per Flashing-Code
- Programmübertragung per App (Android, Apple)
- mehrere Sensoren (Farbe, Hindernis)
- mehrere Aktoren (LEDs, Motoren, Sound)



- Programmierung über Farbcodes zum Kennenlernen
- Untersuchung der Fahrt auf vorgefertigten Karten
- Austesten von Farbcodes mit selbstgemalten Pfaden



- Programmierung mit OzoBlockly
- Je nach eingesetztem Bot / eingestellter Niveaustufe unterschiedlicher Befehlsvorrat
- Einstieg mit einfachen Sequenzen
- Programmübertragung per flashing



Calibration spot




Your bot will blink GREEN if calibration was successful. If your bot blinks RED, try again.



- schrittweise Wahl höherer Niveaustufen ermöglicht Umsetzung weiterer algorithmischer Grundstrukturen
- Wiederholungen mit Zähler zur Programmierung komplexerer Bewegungsmuster
- vorgefertigte Blöcke zur Linienverfolgung
- bedingte Anweisungen (Überprüfung von Sensorwerten)
- Bewältigung einfacher Aufgaben (Hindernisparcours)
- selbst erzeugte Variablen einsetzbar
- kreative Planung und Implementierung eines einfachen „Adventure Games“

Anleitungen für das erste selbst programmierte Adventure-Game

19.12.16

Bedienungsanleitung: Schatzsuche

mit deinen Fingern.
vorne.

Ozogame

Der Ozobot hat die Mission zwei Diamanten über die Grenze zu schmuggeln und in sein Lager zu bringen. Dafür muss er gut aufpassen, dass er nicht erwischt wird, da ihm sonst die Diamanten abgedreht werden und er von vorne anfangen muss.

Zuerst muss man den Ozobot auf die beiden Diamanten (rot und blau) steuern. Wenn er sie eingesammelt hat, muss er versuchen mit ihnen über die Grenze zu kommen (grün). Schafft er es, muss man ihn nur noch zum Versteck navigieren und hat gewonnen (schwarz). Wenn nicht, muss man nochmal von vorne anfangen. (dann wurde er erwischt! ☹)

Um den Ozobot zu steuern muss man das tun:

- nach vorne fahren: Hand hinter den Ozobot halten
- nach rechts fahren: Hand links vor den Ozobot halten
- nach links fahren: Hand rechts vor den Ozobot halten

Mayaexpedition mit Ozobot ev0

Abenteuerer!
mir, Ozobot ev0, auf die Suche
schwarzen Mayaschatz! Es ist ganz einfach!
ger hinten
er rechts vorne
er links vorne
den zwei Schlüsseln, aber vorsicht!
sehr langsam, also versuche nicht in den
fallen. Was passiert? Finde es heraus!

Arduino Uno

- Blockly-Programmierung per drag&drop mittels *mBlock*
- Programmübertragung per USB-Verbindung
- viele Sensoren verfügbar (Helligkeit, IR, UV, Sound, Berührung, ...)
- viele Aktoren verfügbar (LEDs, LCD-Anzeige, ...)
- Fächerübergreif zu Physik möglich

mBlock - Based On Scratch From the MIT Media Lab(v3.4.11) - Nicht verbunden - Nicht gespeichert

Datei Bearbeiten Verbinden Boards Extensions Sprache Hilfe

Skripte

Bewegung Aussehen Klang Malstift Daten&Blöcke Ereignisse Steuerung Fühlen Operatoren Roboter

Arduino

Arduino Programm

setze digitalen Pin 9 als Ausgang

lese digitalen Pin 9

lese analogen Pin (A) 0

lese Puls-Pin 13 Timeout 20000

setze digitalen Pin 9 auf HIGH

setze PWM-Pin 5 Ausgang auf 0

spiele an Pin 9 den Ton C4 für halbe

setze Servo-Pin 9 Winkel auf 90 Grad

LED-Strip Port 6 Led 1 rot 0 gruen

LED-Strip alle LEDs Port 6 rot 0 grue

sende Text an seriellen Port Hallo

serial available bytes

Arduino Programm

LCD an

wiederhole fortlaufend

setze farbe auf 255

wiederhole bis farbe = 0

LCD set background color (red, green, blue) farbe 255 - farbe

ändere farbe um -1

LCD Pos Zeile: 0 Spalte: 0

LCD zeige text verbinde r mit verbinde konvertiere farbe *

LCD Pos Zeile: 0 Spalte: 8

LCD zeige text verbinde b mit 0

LCD Pos Zeile: 1 Spalte: 0

LCD zeige text verbinde g mit verbinde konvertiere 255 - f

warte 0.2 Sek.

wiederhole bis farbe = 255

LCD set background color (red, green, blue) 0 255 - farbe f

ändere farbe um 1

LCD Pos Zeile: 0 Spalte: 0

Zurück Upload zum Arduino Mit der Arduino IDE

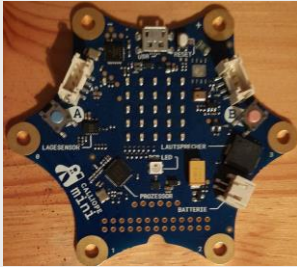
```

18 void loop() {
19     farbe = 255;
20     while (!((farbe) == (0)))
21     {
22         _loop();
23         lcd.setRGB(farbe, (255) - (farbe), 0);
24         farbe += -1;
25         lcd.setCursor(0,0);
26         lcd.print(String("r:") + String(((farbe)
27         lcd.setCursor(8,0);
28         lcd.print(String("b:") + String("0"));
29         lcd.setCursor(0,1);
30         lcd.print(String("g:") + String(((255) -
31         _delay(0.2);
32     }
    
```

send encode mode ☒ Binär-Modus ☐ Zeichen-Modus

recv encode mode ☒ Binär-Modus ☐ Zeichen-Modus

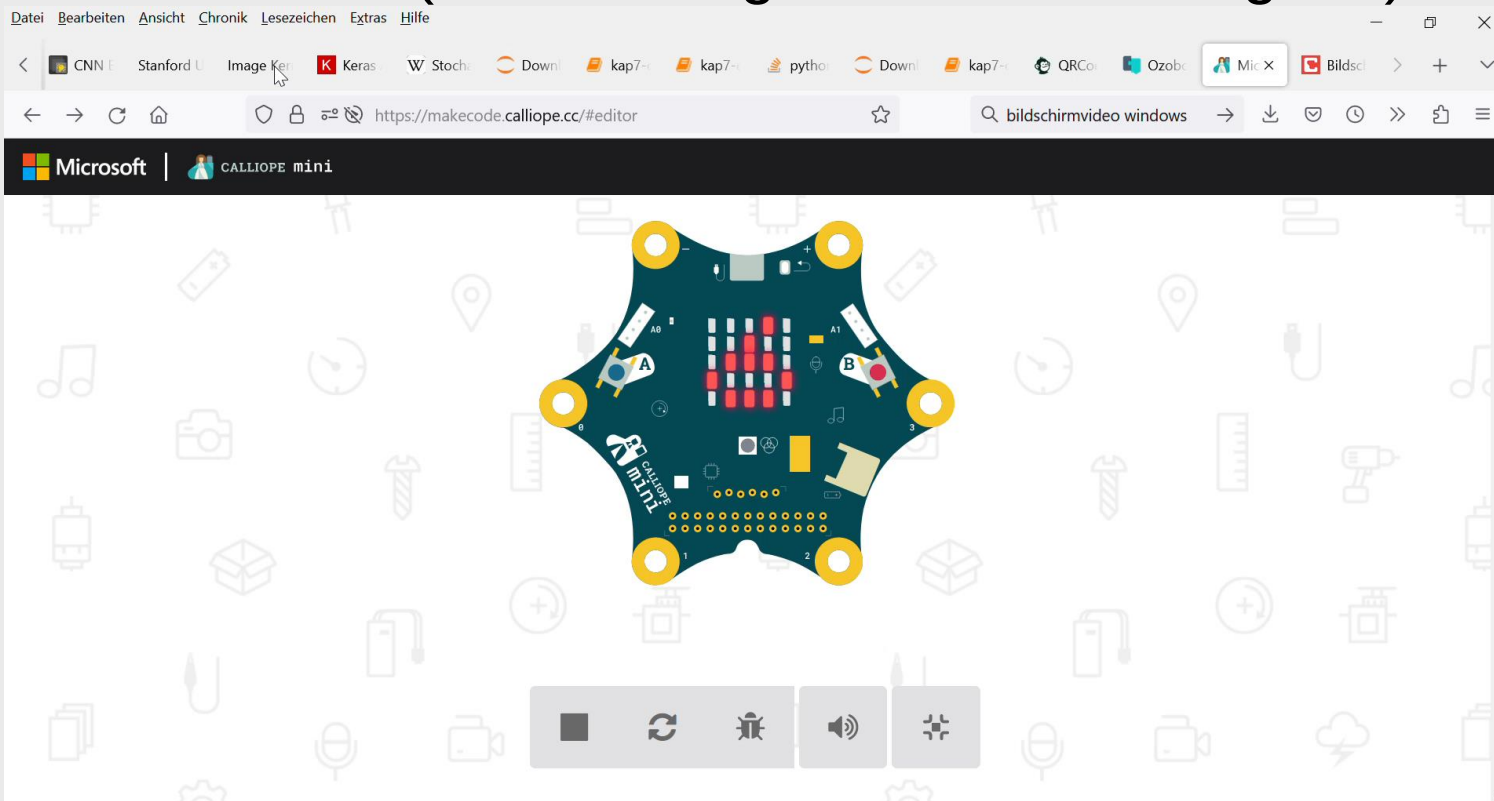
- Materialien zum Einsatz von OzoBots im Unterricht der Universität Münster unter <https://www.uni-muenster.de/Lernroboter/video/#ozobot>
- zur Binnendifferenzierung geeignet: Aufgaben von <https://blockly.games>
- für Prüfungen geeignet: Trainings-Aufgaben des Jugendwettbewerbs Informatik (die blockly-Aufgaben)



Egal wie – Hauptsache Spaß! 😊

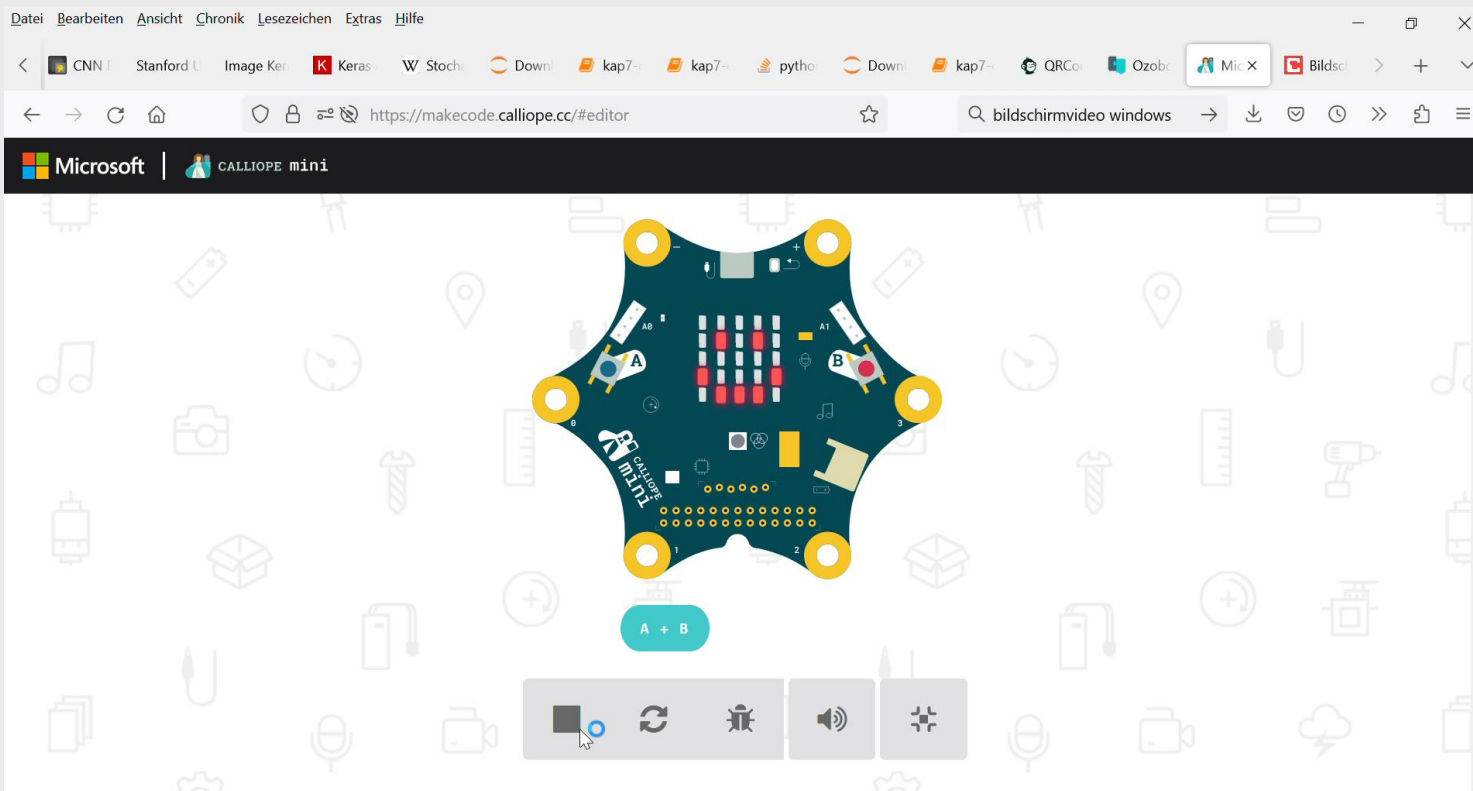
Challenge 1

- Erstelle eine Countdownanzeige bei der am Ende eine Rakete startet (LED-Anzeige und Soundausgabe)



Challenge 2

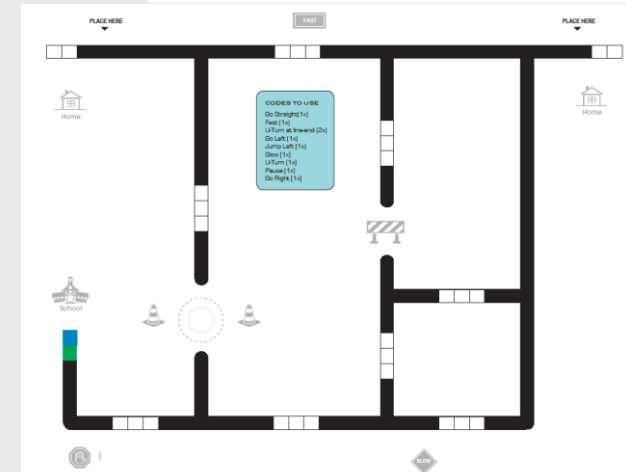
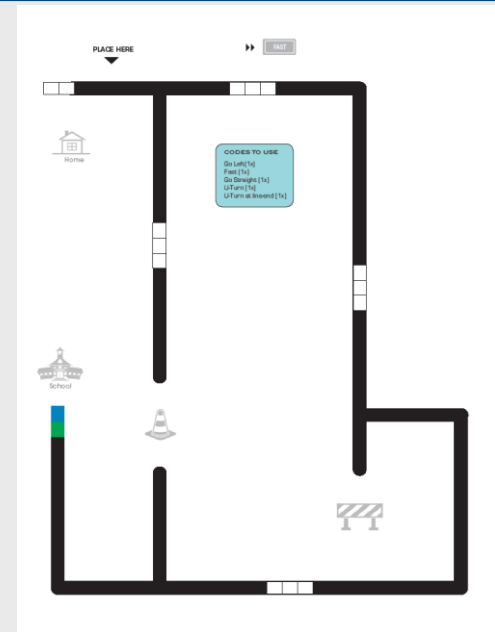
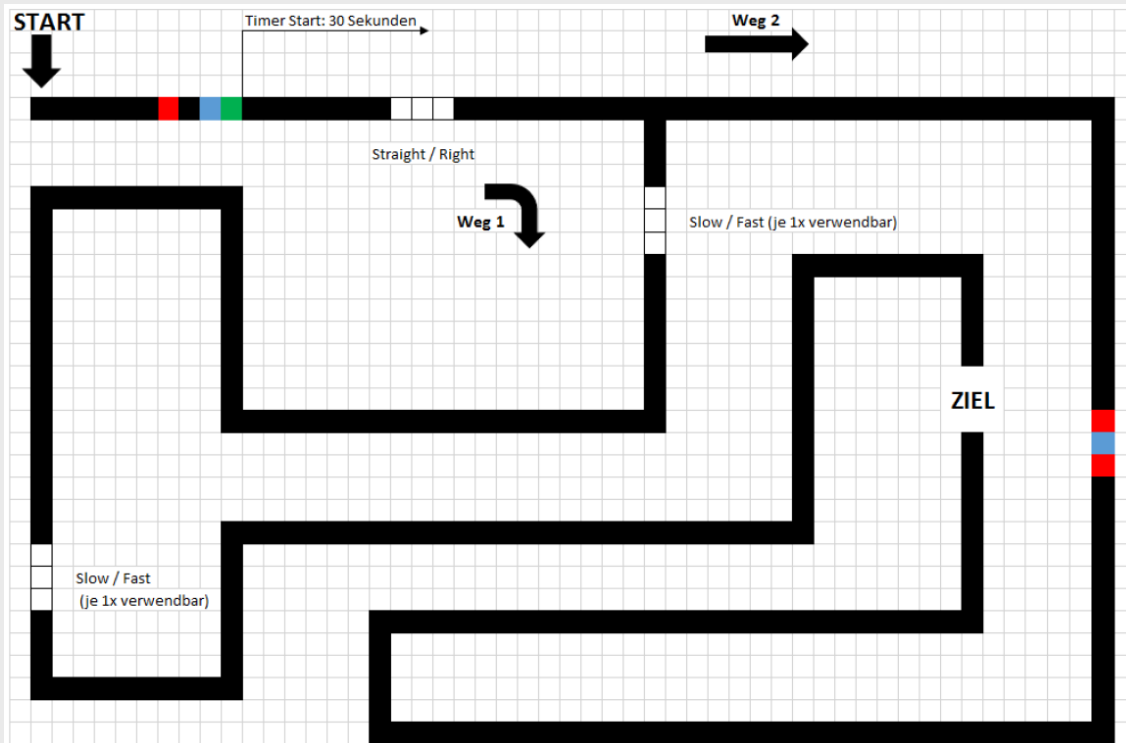
- Erstelle einen Rechentrainer für die Grundschule mit LED-Anzeige und Soundausgabe



Dr.-Ing. Ulrich Kiesmüller
ulrich.kiesmueller@smg-gun.de
Workshop Ozobots und mehr ...

Task 2

- Bewältige die gestellten Aufgaben
- Ergänze geeignete Farbcodes



Tasks

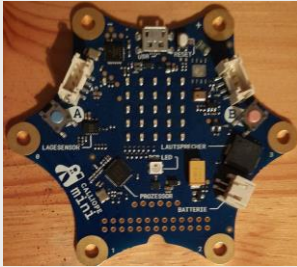
- Erstelle ein eigenes Programm freier Wahl (z. B. "*Model auf Laufsteg*"), übertrage es auf einen OzoBot und lass es ablaufen
- Erstelle ein Programm "*Schneeräumer*", mit dessen Hilfe der OzoBot bei einer Geradeausbewegung alle Hindernisse aus dem Weg räumt.
- Entwickle Dein eigenes "*Adventure-Game*"

It's your turn ... ☺

Arduino uno

Tasks

- Erstelle ein Programm "*Farbwechsler*", das die LCD-Hintergrundfarbe permanent wechseln lässt in der Reihenfolge *blau-gelb-rot-grün-weiß*.
- Erstelle ein Programm, das die LCD-Anzeige nacheinander in allen Farben des RGB-Farbsystems aufleuchten lässt – außerdem sollen die Intensitätswerte der Farbkanäle jeweils mit einer Textausgabe auf dem Display erscheinen.



Viel Spaß beim Arbeiten! 😊



OzoBlockly – Niveaustufen

The screenshot displays the OzoBlockly programming environment with several categories of blocks:

- Mover:** Includes 'Bit', 'Evo', and 'MetaBot' blocks, as well as movement blocks like '1' with a right arrow, '1' with a left arrow, and a '2' block with a right arrow.
- Sounds:** Features blocks with various robot face expressions (happy, sad, surprised, etc.) and sound icons.
- Other Functional Blocks:** Includes blocks for light bulbs (green, blue, cyan, pink), a traffic light, a Christmas tree, fireworks, and a moon.

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 ▾

Light Effects

 turn top light off

 set top light color
 red

 police car lights

 rainbow

 traffic lights

 disco

 christmas tree

 firework

Timing

 wait
 1 ▾ second(s)

Sounds

 play
 happy ▾

 say color
 red ▾

 say direction
 left ▾

 say number
 1 ▾

 play note
 C ▾

Loops

repeat forever

do

repeat
 2 ▾ times

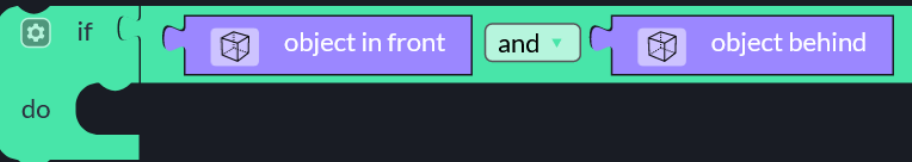
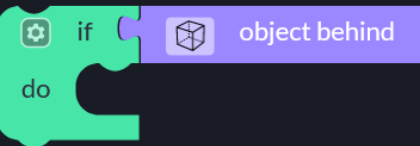
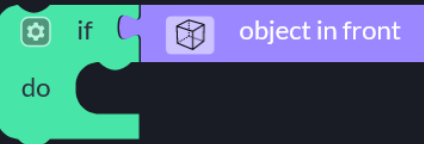
do



OzoBlockly – Niveaustufen



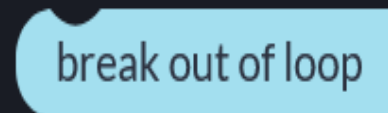
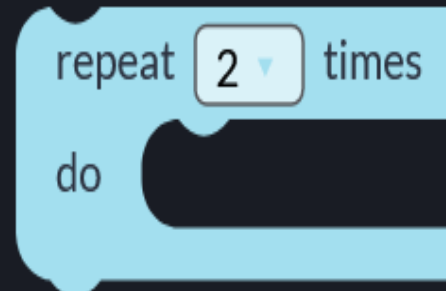
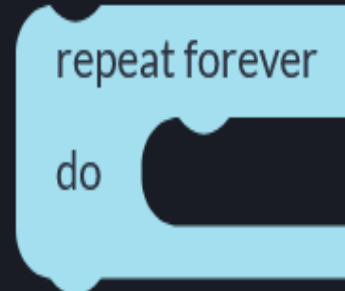
Sensors



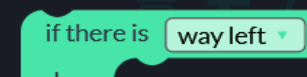
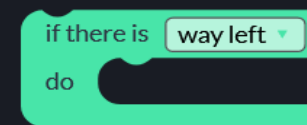
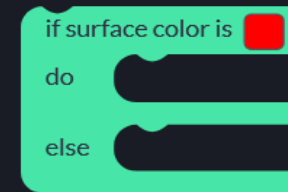
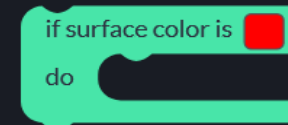
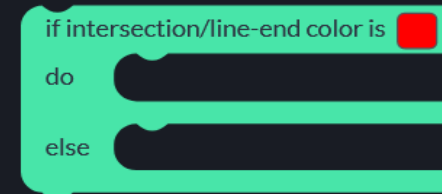
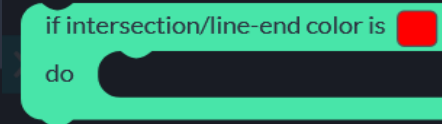
Line Navigation



Loops



Logic





OzoBlockly – Niveaustufen



Line Navigation

follow line to next intersection or line end

pick direction: left

there is way left

set line-following speed: 30 mm/s

get line-following speed

get intersection/line-end color

intersection/line-end color

Timing

wait 1 - 0 second(s)

wait 100 × 10 ms

Loops

repeat 10 times

repeat while true

count with i from 1 to 10 by

break out of loop

Terminate

terminate program and turn Ozobot off

terminate program and continue line following

terminate program and switch to idle

Logic

if do

=

and

not

true

test if true if false

Variables

Create variable...

set counter to

counter

Math

0

+

absolute

is even

change item by 1

remainder of ÷

constrain low 1 high 100

random integer from 1 to 100

Sensors

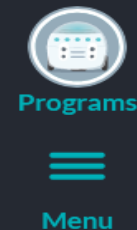
if read proximity sensor left front > 10

do

Catwalk *

```

set counter to 0
say direction forward
repeat while counter ≤ 5
do
  move distance: 20 mm speed: 30 mm/s
  set counter to counter + 1
  say number counter
rotate angle: 30 deg speed: 30 mm/s
say direction left
set light color
  lights (light top front)
  color surface color
say color surface color
rotate angle: -60 deg speed: 30 mm/s
say direction right
set light color
  lights (light top front)
  color surface color
say color surface color
rotate angle: 30 deg speed: 30 mm/s
  
```



Timing

wait 1 . 0 second(s)

wait 100 × 10 ms

set timer value 0

get timer value (s)

check timer tick event

Button

capture button press events true

get button press count

set button press count 0

wait for button press

Math

0

+

absolute

is even

change item by 1

remainder of ÷

constrain low 1 high 100

random integer from 1 to 100

set random number generator seed 42

bitwise AND

bitwise shift LEFT 1

List

append value to the end of List

in List set element to value

get element from List

get last element and remove it from List

remove element at index from List

length of List

Arrays

declare array arrayName of length 10