



## Geschwindigkeit

	Schneckentempo
	Langsam
	Normal
	Schnell
	Turbo
	Nitro-Boost

## Verhalten bei Kreuzungen

	Nach Links
	Geradeaus
	Nach Rechts

## Umkehren

	U-Turn
	U-Turn (am Ende)

## Cooler Moves

	Tornado
	Zigzag
	Spin
	Rückwärts fahren

## Line Jumps

	Line Jump Links
	Line Jump Rechts
	Line Jump Geradeaus

## Zähler

	Kreuzungszähler an
	Abbiegezähler an
	Farbwechselzähler an
	Punktezähler auf 5
	Punktezähler +1
	Punktezähler -1

## Ozobot stoppen

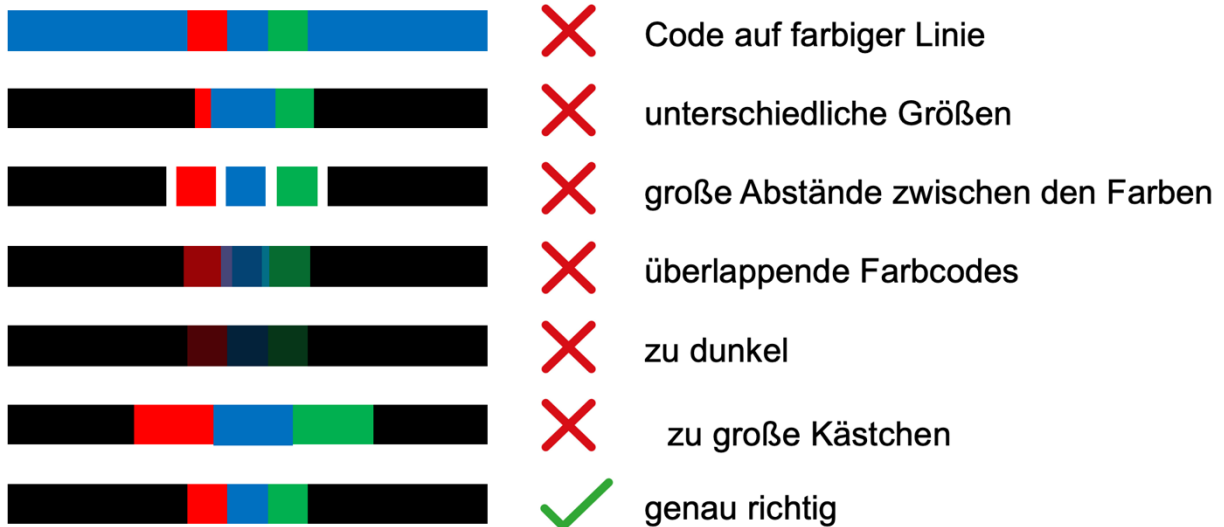
	Gewonnen
	Verloren

## Timer

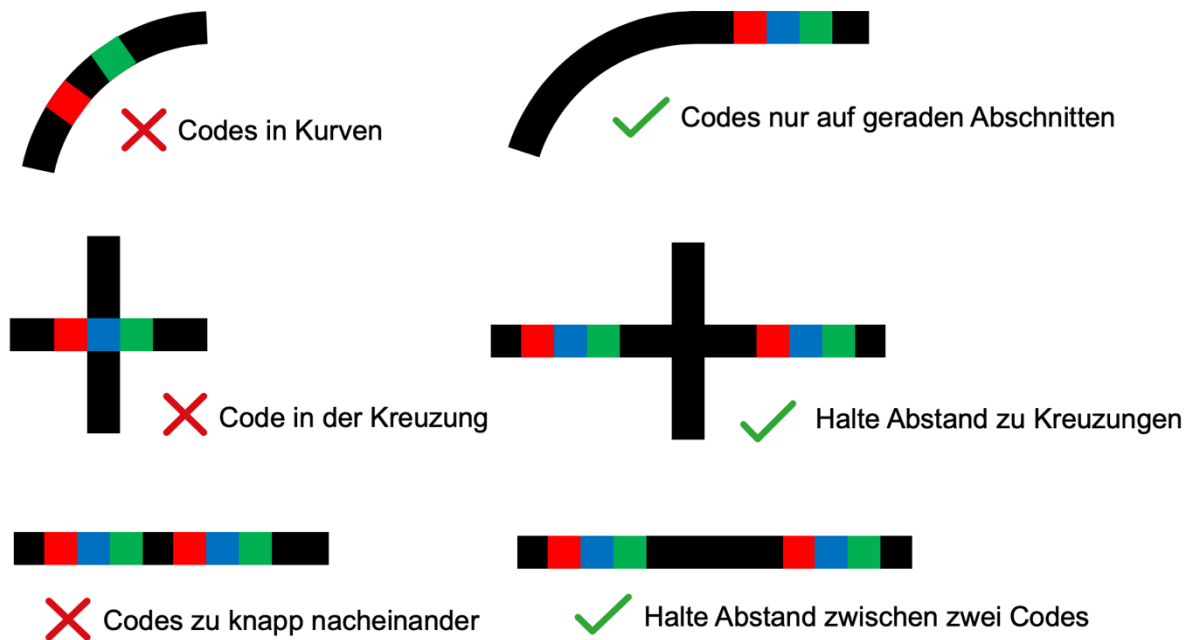
	Timer ein (30 sec)
	Timer aus
	Pause (3s)



## Farbcodes zeichnen

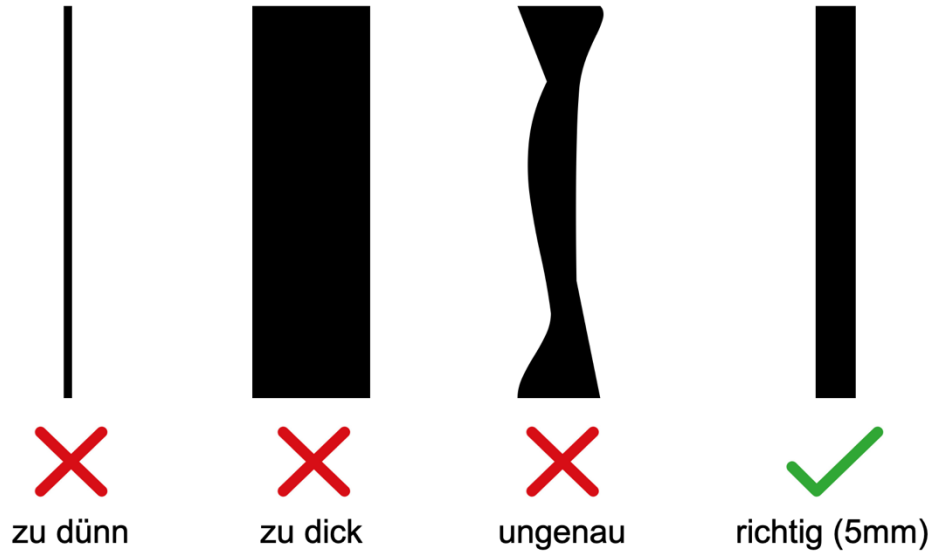


## Farbcodes platzieren





## Linien zeichnen



## Kurven zeichnen

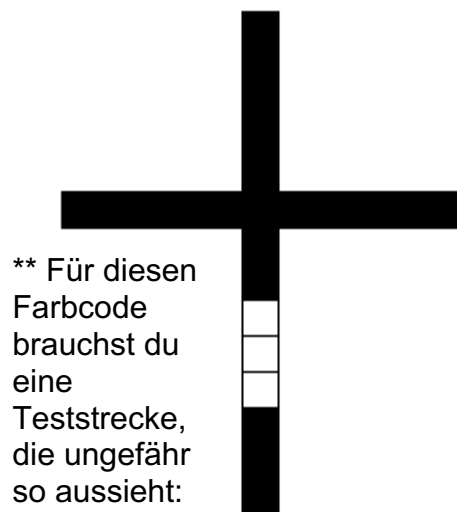




## Einheit 1

# Was macht welcher Farbcode? – Finde es heraus!

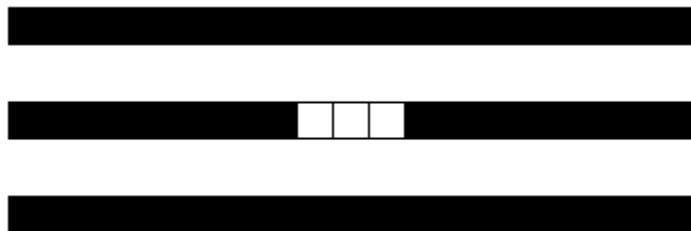
- ✓ Wie du bereits im Video gesehen hast, lässt sich der Ozobot mithilfe von **Farbcodes** steuern. **Finde nun heraus, wie der Ozobot auf die verschiedenen Farbcodes reagiert.**
- ✓ Manche Farbcodes beeinflussen die *Geschwindigkeit*, manche das *Verhalten bei Kreuzungen* und manche lassen den Ozobot „coole Moves“ ausführen.
- ✓ Nimm dir ein weißes Blatt Papier, Stifte (schwarz, blau, grün und rot) und probiere die verschiedenen Codes aus. Denke daran, dass die Codes nur dann vom Ozobot erkannt werden, wenn du sie sauber zeichnest.
- ✓ Für Farbcodes, die mit \*, \*\*, \*\*\* und \*\*\*\* gekennzeichnet sind, brauchst du spezielle Teststrecken. Beispiele dafür findest am Ende dieser Seite.
- ✓ Alle möglichen Lösungen findest du am Ende der nächsten Seite. Versuche also herauszufinden, welche Lösung (welches Verhalten) zu welchem Code gehört.
- ✓ **Wenn du herausgefunden hast, wie der Ozobot bei den Farbcodes reagiert, dann trage das Verhalten in die Farbcode-Tabelle auf der nächsten Seite ein.**
- ✓ Wenn du ein echter Experte bist, schaffst du die Aufgabe auch ohne die Lösungen (Klappe dazu die Tabelle mit den möglichen Lösungen um).



\* Dieser Farbcode muss am Ende einer Linie stehen



\*\*\* Für diesen Farbcode brauchst du eine Teststrecke, die ungefähr so aussieht:



\*\*\*\* Für diesen Farbcode brauchst du eine Teststrecke, die ungefähr so aussieht:





## Einheit 1

# Was macht welcher Farbcode? – Finde es heraus!

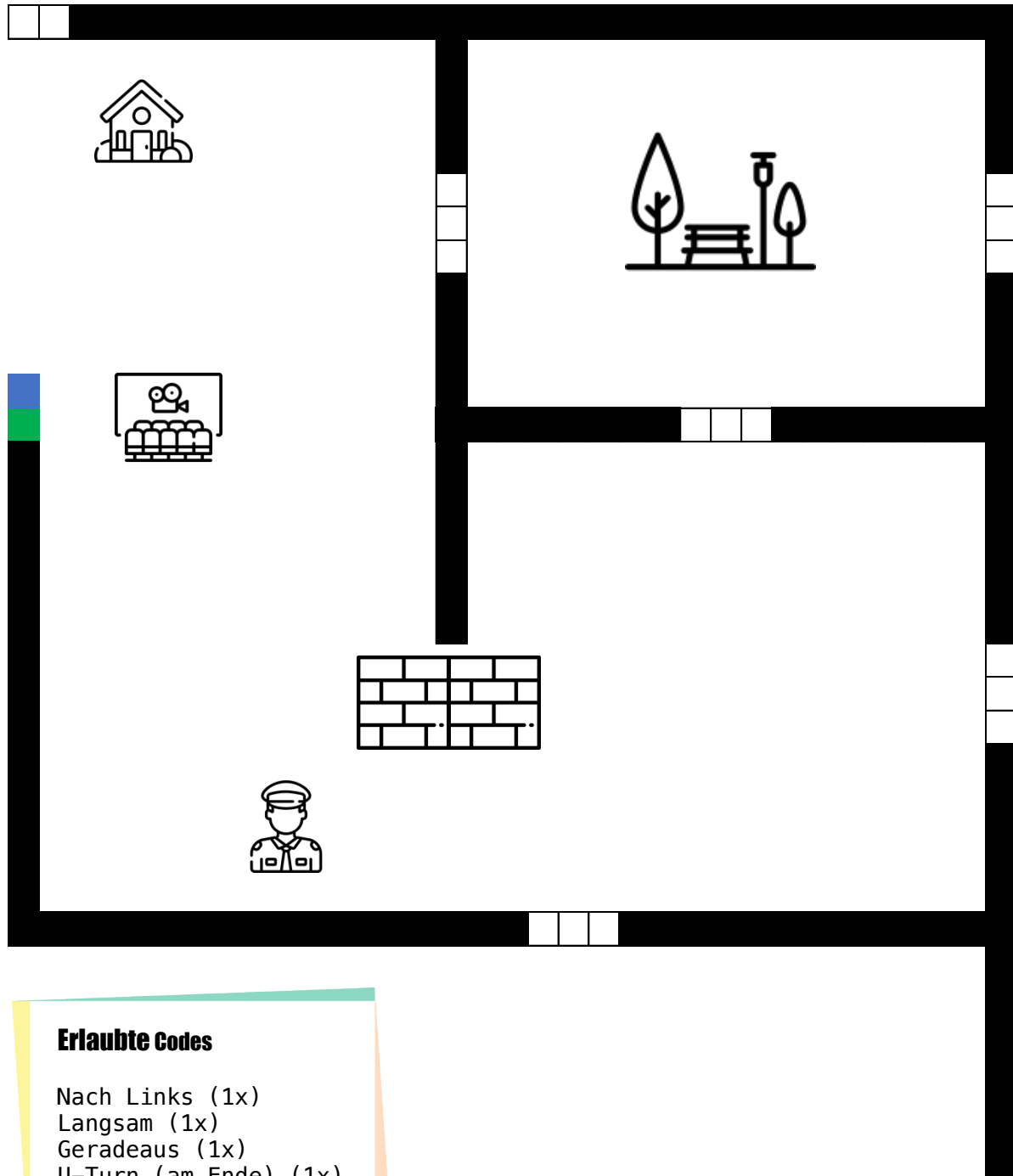
1	10
2	11  **
3  ***	12
4  *	13  ****
5  **	14
6  ***	15
7	16
8	17
9  **	18

## Mögliche Lösungen

Geschwindigkeit / Speed	Richtung / Direction	Moves
Schneckentempo	Nach Links	Tornado
Langsam	Nach Rechts	Zigzag
Normal	Geradeaus	Spin
Schnell	Line Jump Links	
Turbo	Line Jump Rechts	
Nitro-Boost	Line Jump Geradeaus	
Pause (3s)	U-Turn	
	U-Turn (am Ende)	

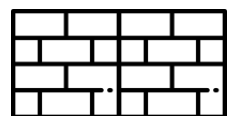


# Fahre ins Kino!



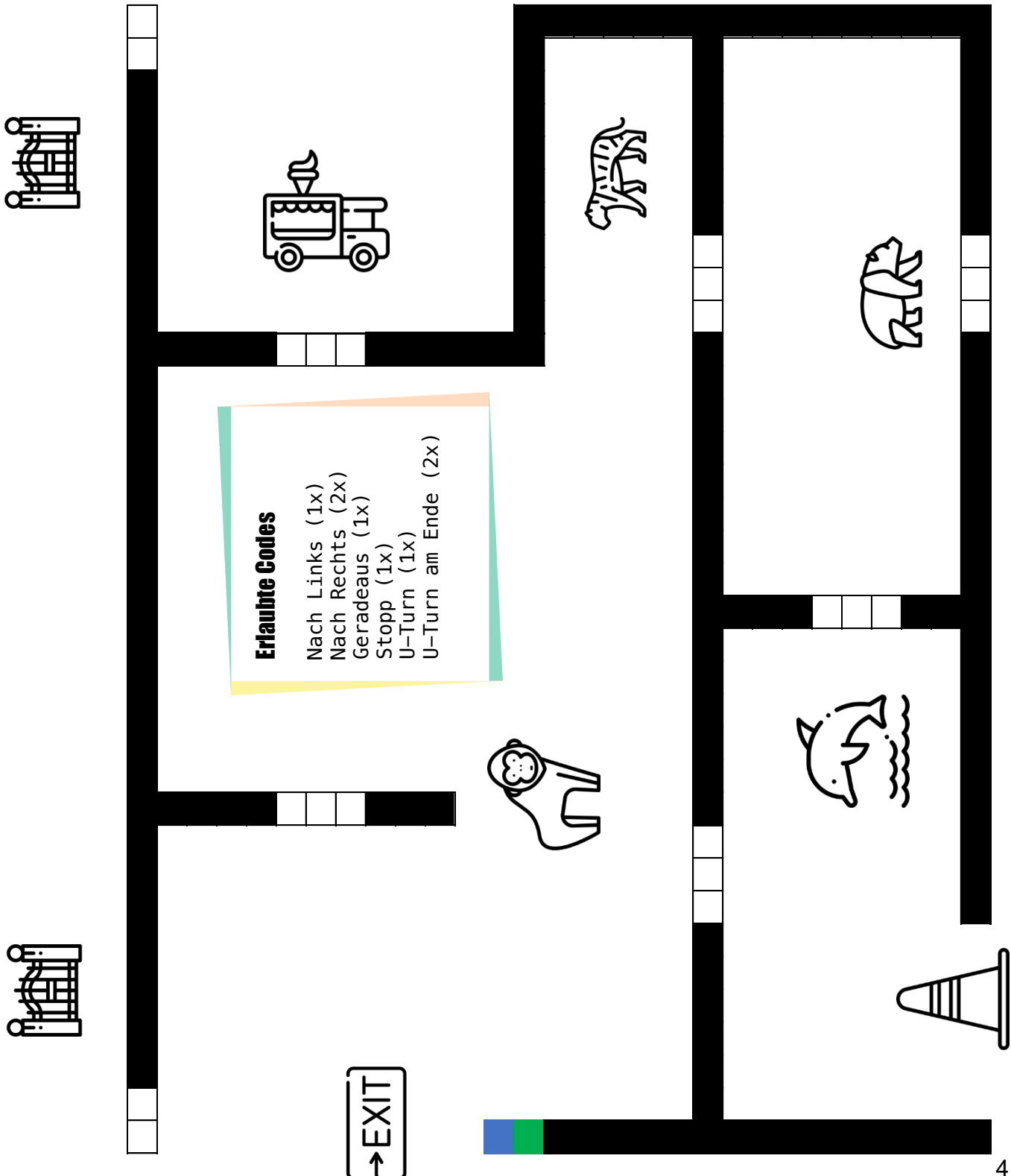
## Erlaubte Codes

Nach Links (1x)  
Langsam (1x)  
Geradeaus (1x)  
U-Turn (am Ende) (1x)  
Nach Rechts (2x)





# Ozobot im Zoo



Line Jump Rechts (1x)  
Line Jump Links (1x)  
Line Jump Geradeaus (1x)  
U-Turn (1x)  
U-Turn am Ende (1x)







## Einheit 3

# Geschwindigkeitsmessung

In dieser Übung geht es darum, die **Geschwindigkeit des Ozobot** zu bestimmen. Dabei ist es wichtig zu wissen, dass der Ozobot mit speziellen Farbcodes so gesteuert werden kann, dass er sich unterschiedlich schnell fortbewegt.

Die Aufgabe eurer Gruppe ist es, die **Geschwindigkeit des Ozobot im zugeteilten Geschwindigkeitsmodus (langsam, normal oder schnell)** zu berechnen und die berechneten Werte am Ende der Einheit in die folgende Tabelle einzutragen.

Modus	Geschwindigkeit in <i>m/s</i>
Langsam	
Normal	
Schnell	

## Berechnung der Geschwindigkeit

Wie ihr ja wahrscheinlich wisst, wird die Geschwindigkeit (abgekürzt durch  $v$ ), mit der Formel Weg ( $s$ ) durch Zeit ( $t$ ) berechnet. Die Geschwindigkeit ist also immer das Verhältnis des zurückgelegten Weges und der dafür benötigten Zeit.

$$v = \frac{s}{t}$$

Um die Geschwindigkeit des Ozobot berechnen zu können, benötigt ihr eine vorgegebene Strecke und die Zeit, die der Ozobot braucht, um diese Strecke abzufahren.

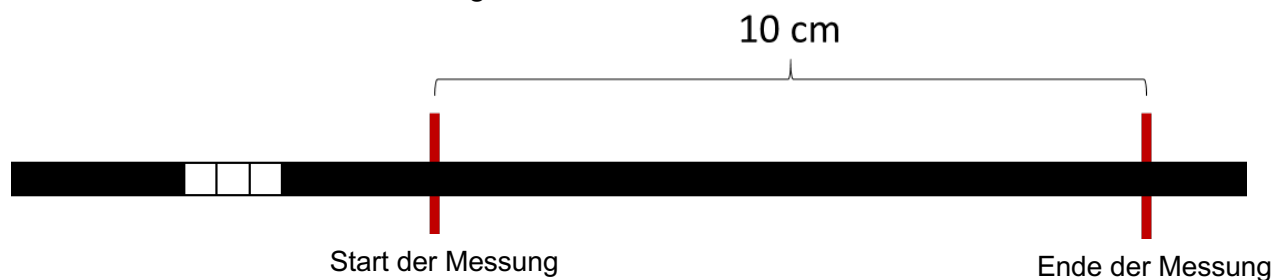
- ✓ Die Teststrecke muss eine definierte Länge haben.
- ✓ Mit einer Stoppuhr könnt ihr die Zeit messen, die der Ozobot für das Abfahren der Strecke benötigt.



### So gehst du am besten vor:

Damit ihr die unterschiedlichen Geschwindigkeiten des Ozobot messen könnt, müsst ihr am Beginn der Teststrecke auch den zugeteilten Speed-Farbcode eintragen. Da der Ozobot die Geschwindigkeit erst nach dem Lesen des Farbcodes ändert, solltet ihr die Zeitmessung erst mit etwas Abstand danach starten.

Eure Teststrecke könnte also folgendermaßen aussehen:



- ✓ Führt die Zeitmessung **mindestens 10-mal** durch, da ihr aufgrund eurer Reaktionszeit beim Stoppen der Zeit zu unterschiedlichen Ergebnissen kommt.
- ✓ Notiert alle 10 Messwerte (Zeiten).
- ✓ Berechnet anschließend den Durchschnitt (Summe dividiert durch Anzahl der Zeiten) der einzelnen Messungen, also die durchschnittlich benötigte Zeit.
- ✓ Berechnet nun die Geschwindigkeit nach der oben beschriebenen Formel und gebt das Ergebnis in  $m/s$  (Meter pro Sekunde) an.

### Beispielberechnung:

- ✓ Messen: Ihr habt eine  $10\text{ cm}$  lange Strecke gewählt und der Ozobot braucht dafür  $3,2\text{ Sekunden}$ .
- ✓ Einheiten umrechnen: Da ihr die Geschwindigkeit in der Einheit  $m/s$  angeben sollt, müsst ihr  $10\text{ cm}$  in  $m$  umrechnen. Das ergibt  $0,1\text{ m}$ .
- ✓ Formel: Nun könnt ihr die Werte in die Geschwindigkeitsformel einsetzen:

$$v = \frac{s}{t} = \frac{0,1}{3,2} = 0,03125\text{ m/s}$$

### Optional:

Probiert auch längere Teststrecken mit weniger Messungen aus und beantwortet die folgenden Fragen:

- ✓ Gelingt mit dieser Methode eine genauere Messung?
- ✓ Wenn ja, warum ist das so?



## Einheit 3 – Vertiefung

# Finde den schnelleren Weg!

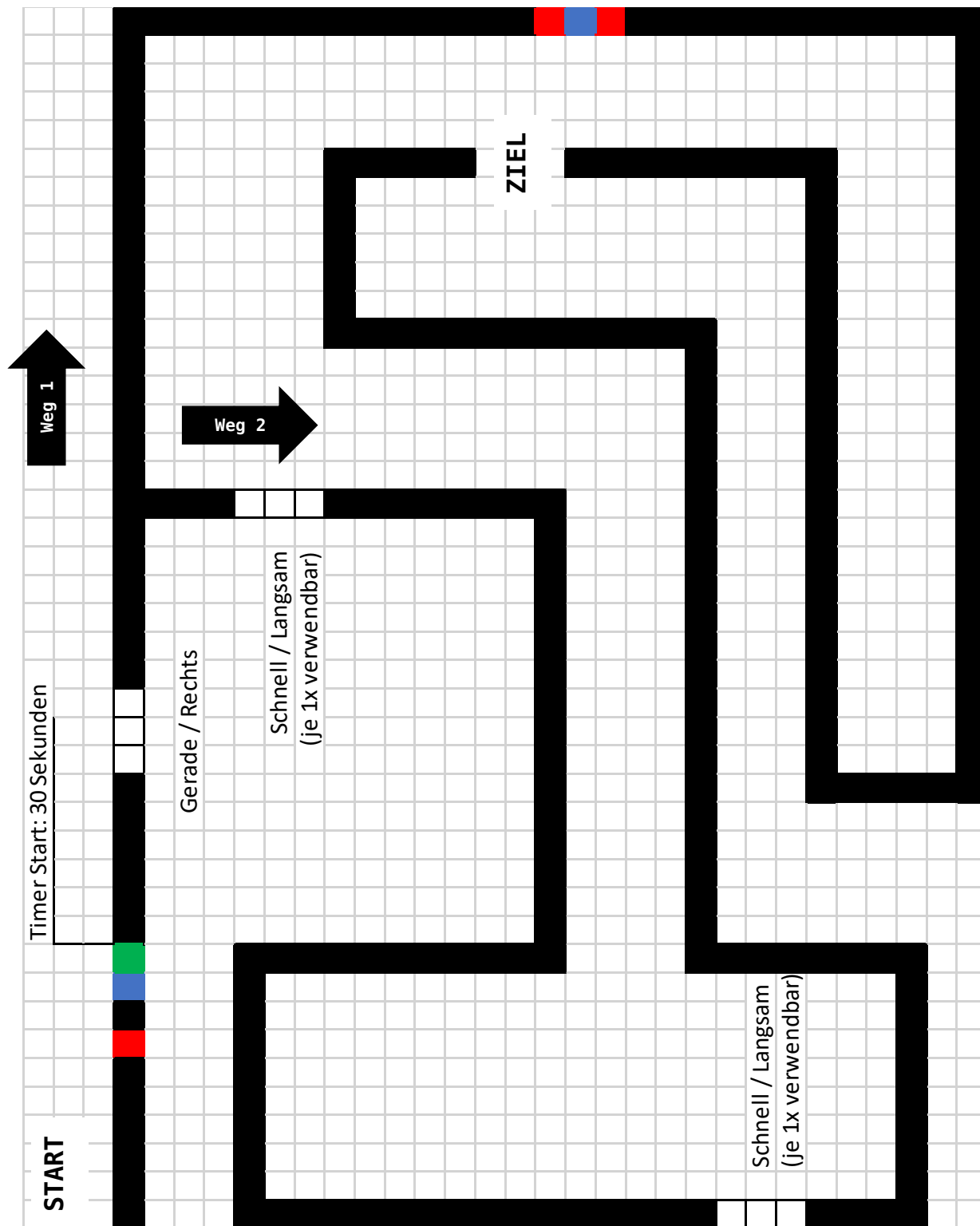
In der Aufgabe „Geschwindigkeitsmessung“ hast du herausgefunden, wie schnell der Ozobot fahren kann. Nun sollst du mit diesem Wissen eine weitere Aufgabe lösen. Bei dieser Aufgabe geht es darum, den Ozobot auf **schnellstem Weg** vom Start zum Ziel zu bringen. Dabei läuft die Zeit (ein sogenannter „Timer“) mit. Wählst du den falschen Weg, schafft es der Ozobot vor Ablauf des Timers nicht ins Ziel und bleibt stehen.

- ✓ Sieh dir Weg 1 bzw. Weg 2 genau an.
- ✓ Miss die Länge von Weg 1 bzw. Weg 2 (beginnend beim Start) und notiere dir die gemessenen Werte.
- ✓ Berechne nun die Zeit, die der Ozobot vom Start bis zum Ziel braucht,
  - wenn er Weg 1 wählt
  - wenn er Weg 2 wählt und bestimmte Streckenabschnitte mit unterschiedlicher Geschwindigkeit durchfährt.
- ✓ Achte darauf, dass bei Weg 2 die Reihenfolge, in der du die Speed-Farbcodes einsetzen kannst, bei der Berechnung berücksichtigt werden muss.
- ✓ Beantworte die folgenden Fragen:
  - Welcher Weg ist (rein rechnerisch) der schnellere – Weg 1 oder Weg 2?
  - Macht es (rein rechnerisch) einen Unterschied, in welcher Reihenfolge die Farbcodes bei Weg 2 eingesetzt werden?
- ✓ Füge die entsprechenden Farbcodes auf der Strecke ein und überprüfe dein Ergebnis im Praxistest mit dem Ozobot.

**Schaffst du es, den Ozobot vor Ablauf des Timers ins Ziel zu bringen?**



# Finde den schnelleren Weg!





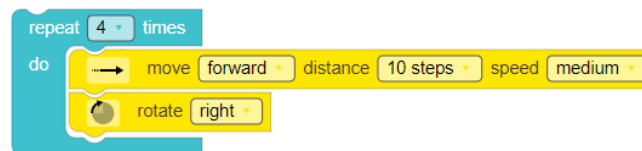
## Einheit 5

# Schleifen in Ozoblockly

In dieser Übung erfährst du, was in der Programmierung mit einer **Schleife** gemeint ist und warum es oft sehr sinnvoll ist, Schleifen in deinem Programm für den Ozobot zu verwenden. Wenn du das folgende *Ozoblockly*-Programm auf deinen Ozobot überträgst und ausführst, dann wird der Ozobot in Form eines Quadrates fahren.



Diesen Code kannst du allerdings auch wesentlich kürzer schreiben. Wie du oben siehst, **wiederholen** sich die beiden Blöcke „move“ und „rotate“ jeweils viermal. Und genau bei solchen Wiederholungen kommen Schleifen ins Spiel. Durch den Schleifenblock „repeat 4 times“ wird das Programm deutlich kürzer und übersichtlicher:



Die beiden Blöcke, die sich innerhalb der Schleife befinden, werden nun 4-Mal wiederholt und der Ozobot wird wieder in Form eines Quadrats fahren.

## Deine Aufgabe:

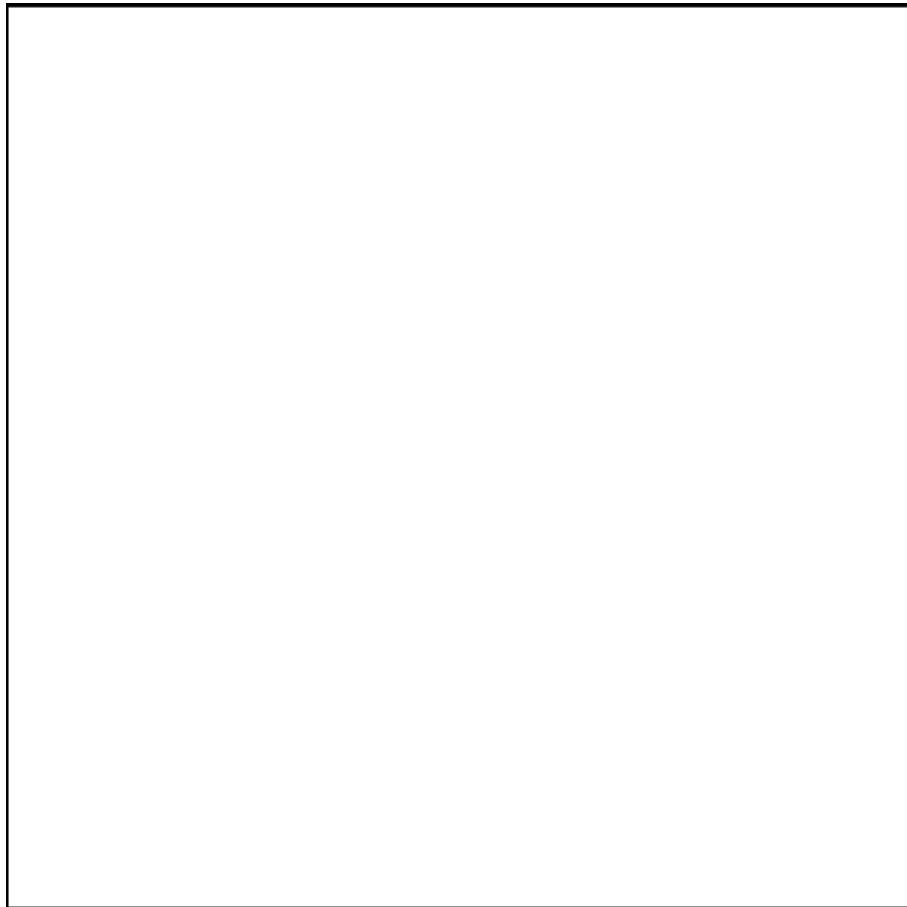
- ✓ Versuche nun mithilfe von *Ozoblockly* deinen Ozobot so zu programmieren, dass er einige der geometrischen Figuren auf den nächsten Seiten abfahren kann.
- ✓ Versuche Wiederholungen/wiederkehrende Muster in den geometrischen Figuren zu erkennen.
- ✓ Verwende für diesen Zweck Schleifen in deinem Ozoblockly-Programm.
- ✓ Übertrage deinen Code zum Testen auf den Ozobot und lasse den Ozobot die Figuren abfahren. (Hinweis: 1 „step“ entspricht etwa 1 cm)

**Bonus: Fallen dir noch weitere Muster mit wiederkehrenden Elementen ein?**



## Einheit 5

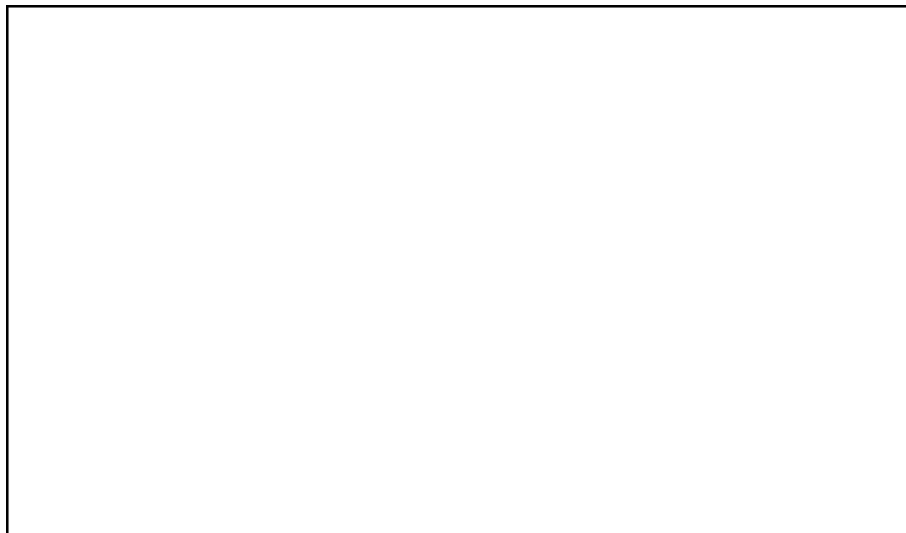
### Teste deinen Code hier – Objekt 1





## Einheit 5

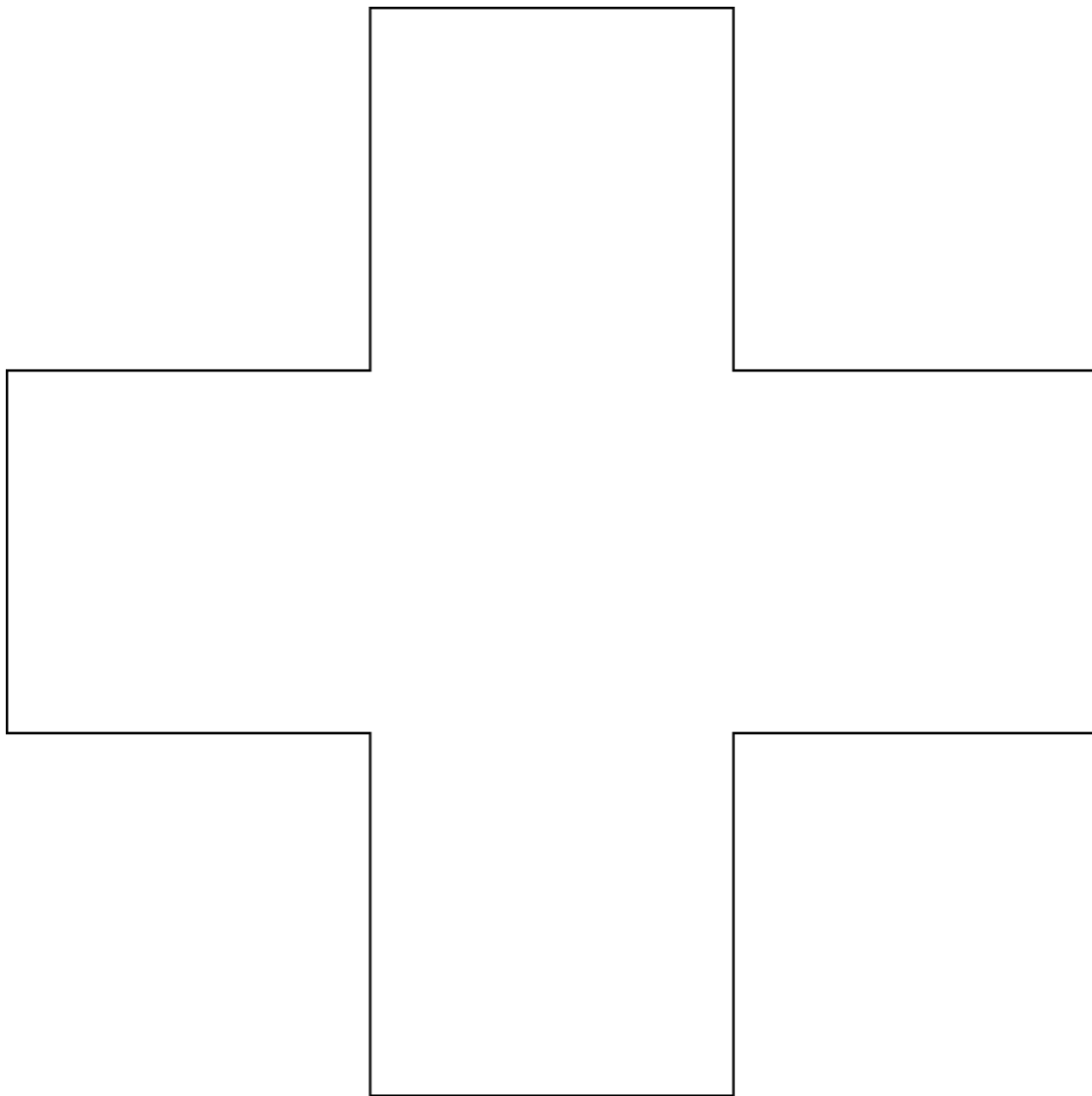
### Teste deinen Code hier – Objekt 2





## Einheit 5

### Teste deinen Code hier – Objekt 3

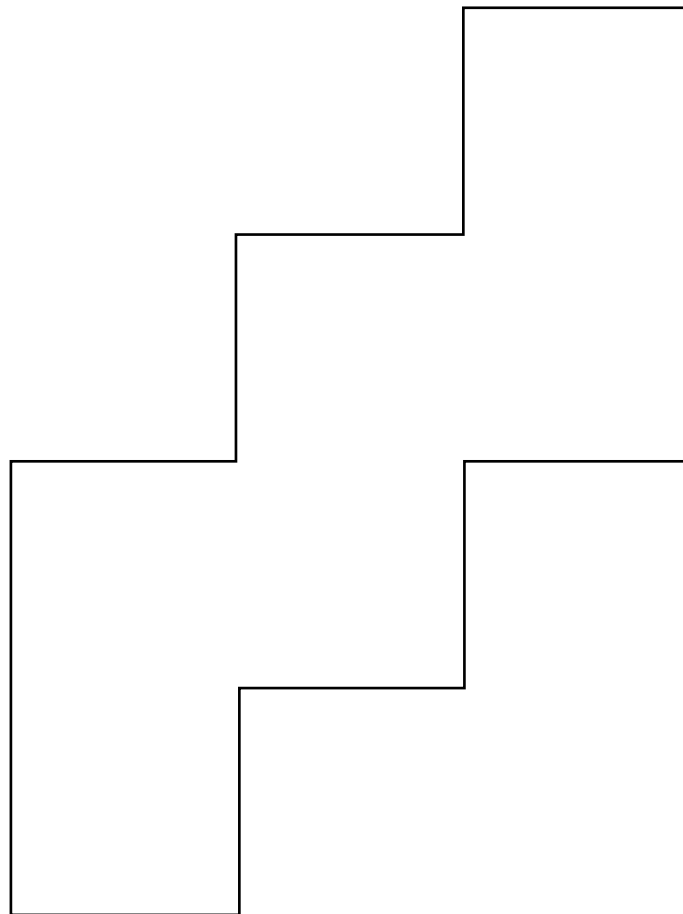






## Einheit 5

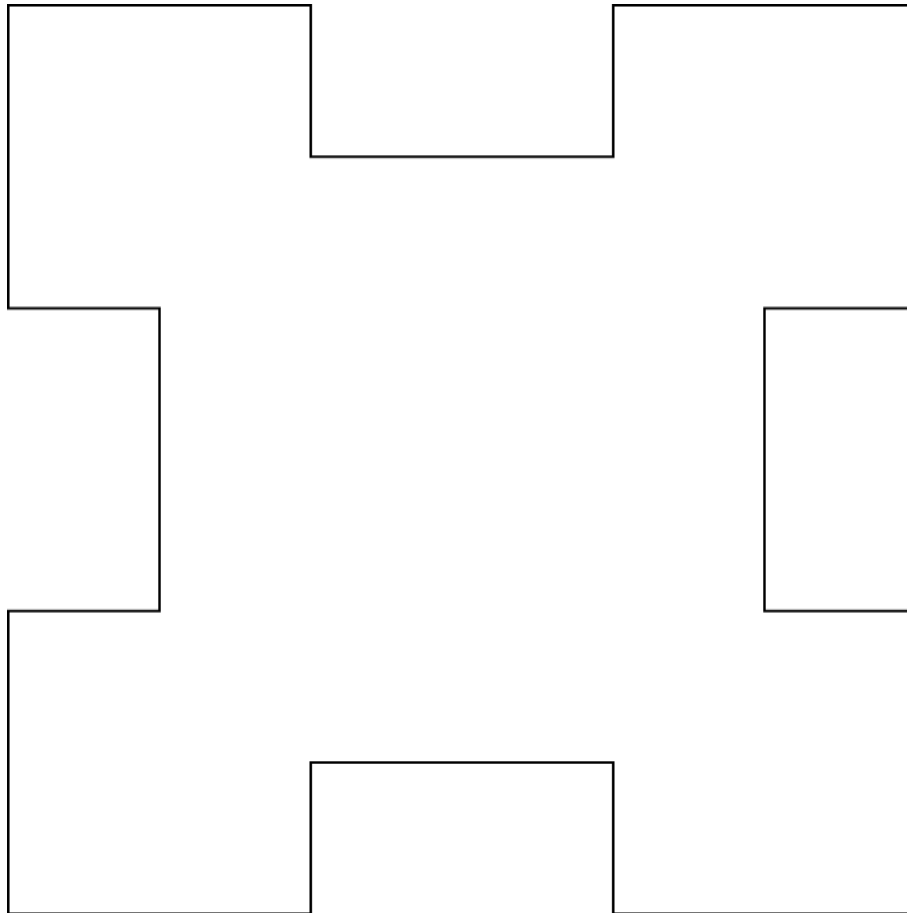
### Teste deinen Code hier – Objekt 4





## Einheit 5

### Teste deinen Code hier – Objekt 5

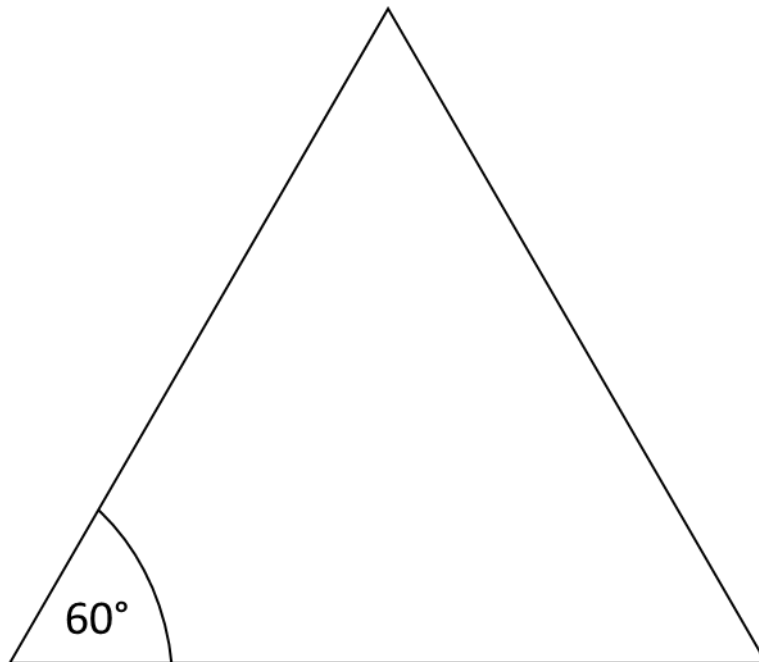




## Einheit 5

### Teste deinen Code hier – Objekt 6

Für diese Aufgabe musst du Ozoblockly mindestens in Level 4 (Advanced) verwenden.

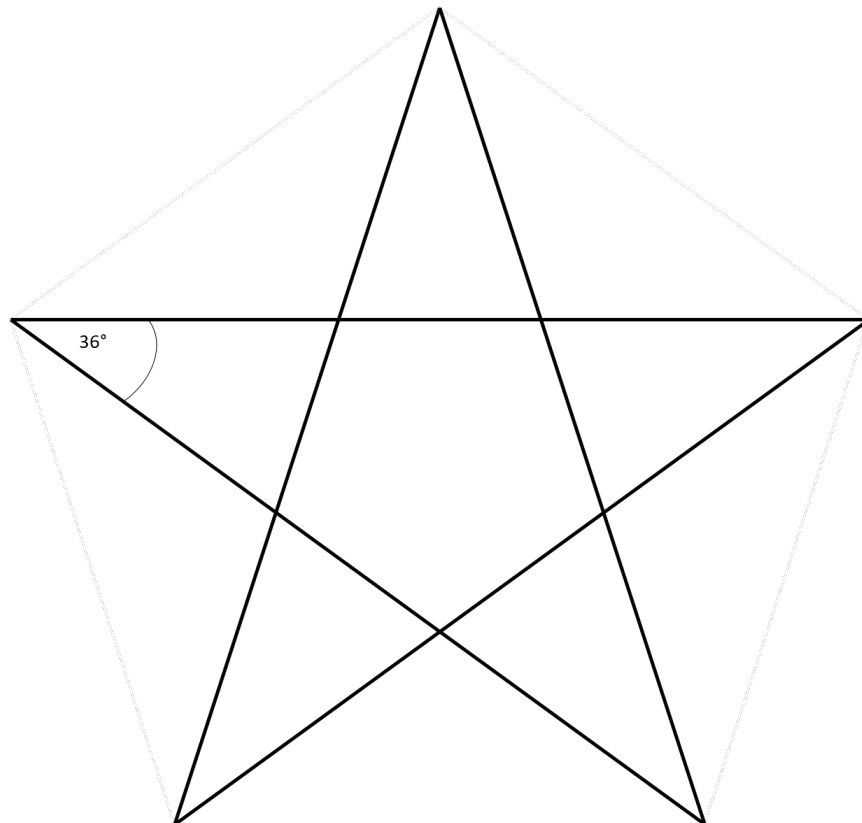




## Einheit 5

### Teste deinen Code hier – Objekt 7

Für diese Aufgabe musst du Ozoblockly mindestens in Level 4 (Advanced) verwenden.





## Einheit 6

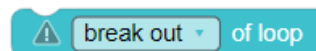
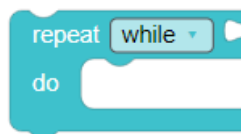
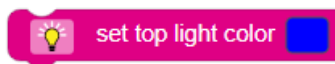
# Finde das Wasser!

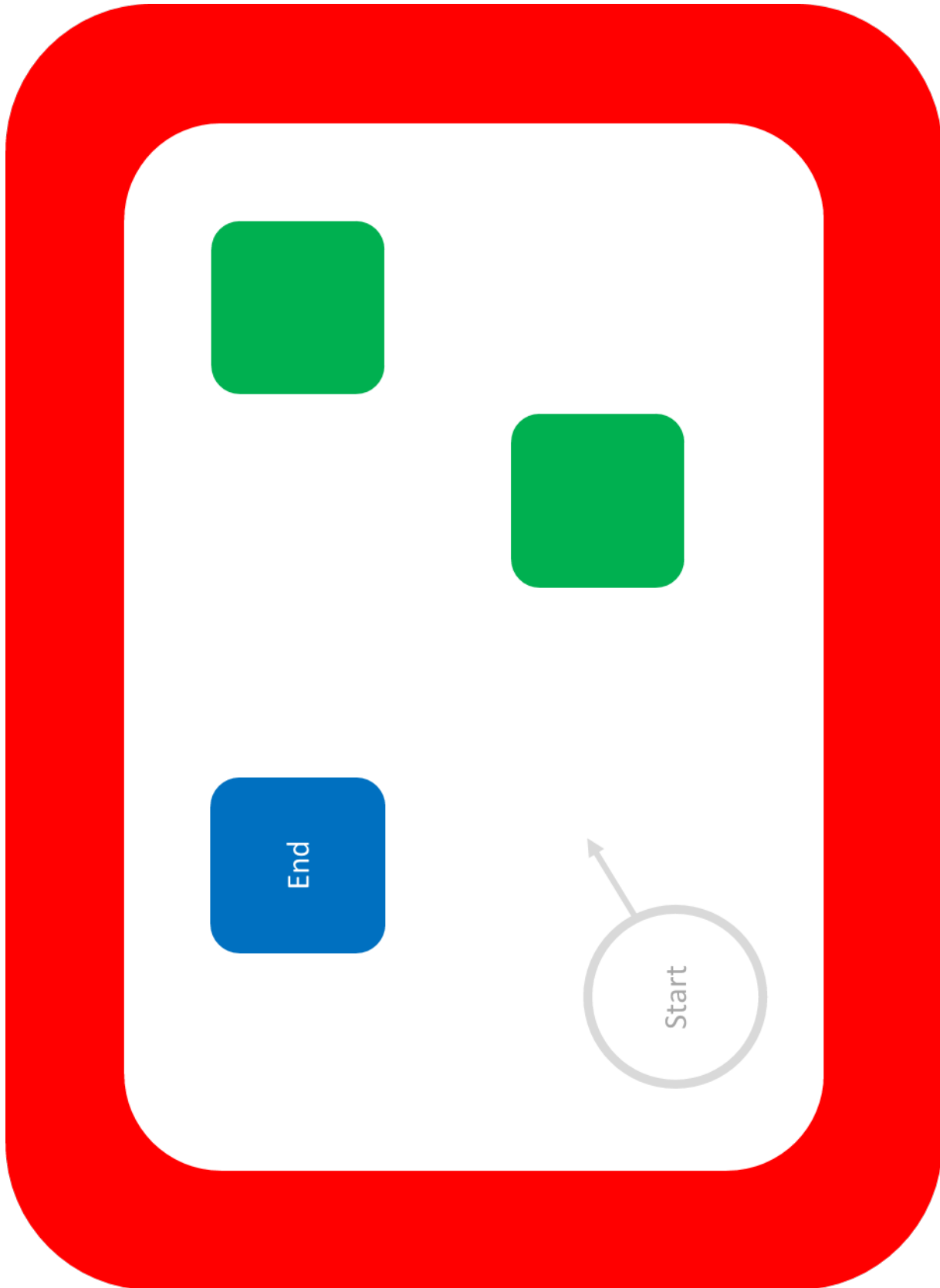
Auf der nächsten Seite findest du die Arena für diese Aufgabe.

- ✓ Der Ozobot soll sich innerhalb der Arena (rotes Feld) bewegen und nach der Wasserstelle (blauer Bereich) suchen.
- ✓ Dabei soll die LED des Ozobots stets in der Farbe des Untergrunds leuchten.
  - Überquert der Ozobot eine Grasfläche (grüne Fläche), so soll die LED des Ozobot grün leuchten.
  - Gelangt der Ozobot an den Rand, dann soll die LED des Ozobot rot leuchten und er soll seine Richtung ändern (Umdrehen/Drehen/...).
  - Findet der Ozobot die Wasserstelle, dann soll er nicht mehr weiterfahren und stehen bleiben.

## Tipps

Die folgenden Blöcke können für die Lösung der Aufgabe hilfreich sein.







## 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:

- Der Ozobot soll nach einer Minute stoppen.
- Der Ozobot soll nach 15 passiertten Kreuzungen stoppen.
- 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.

## Teste deinen Code hier!

