



# iRobot Factory

## Handbuch

Gefördert durch:

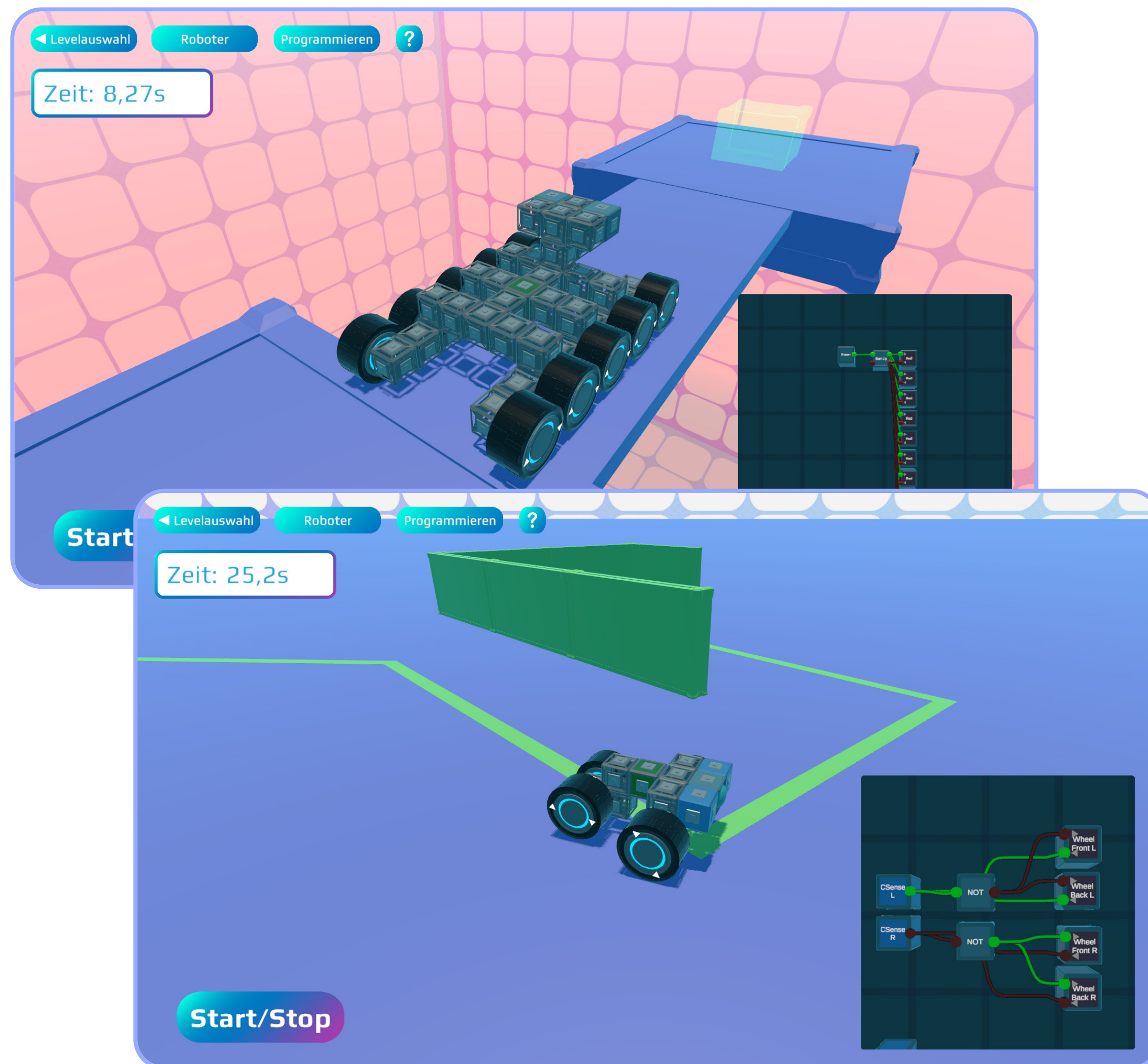
einfachgenial



Walter **Nebel** Stiftung

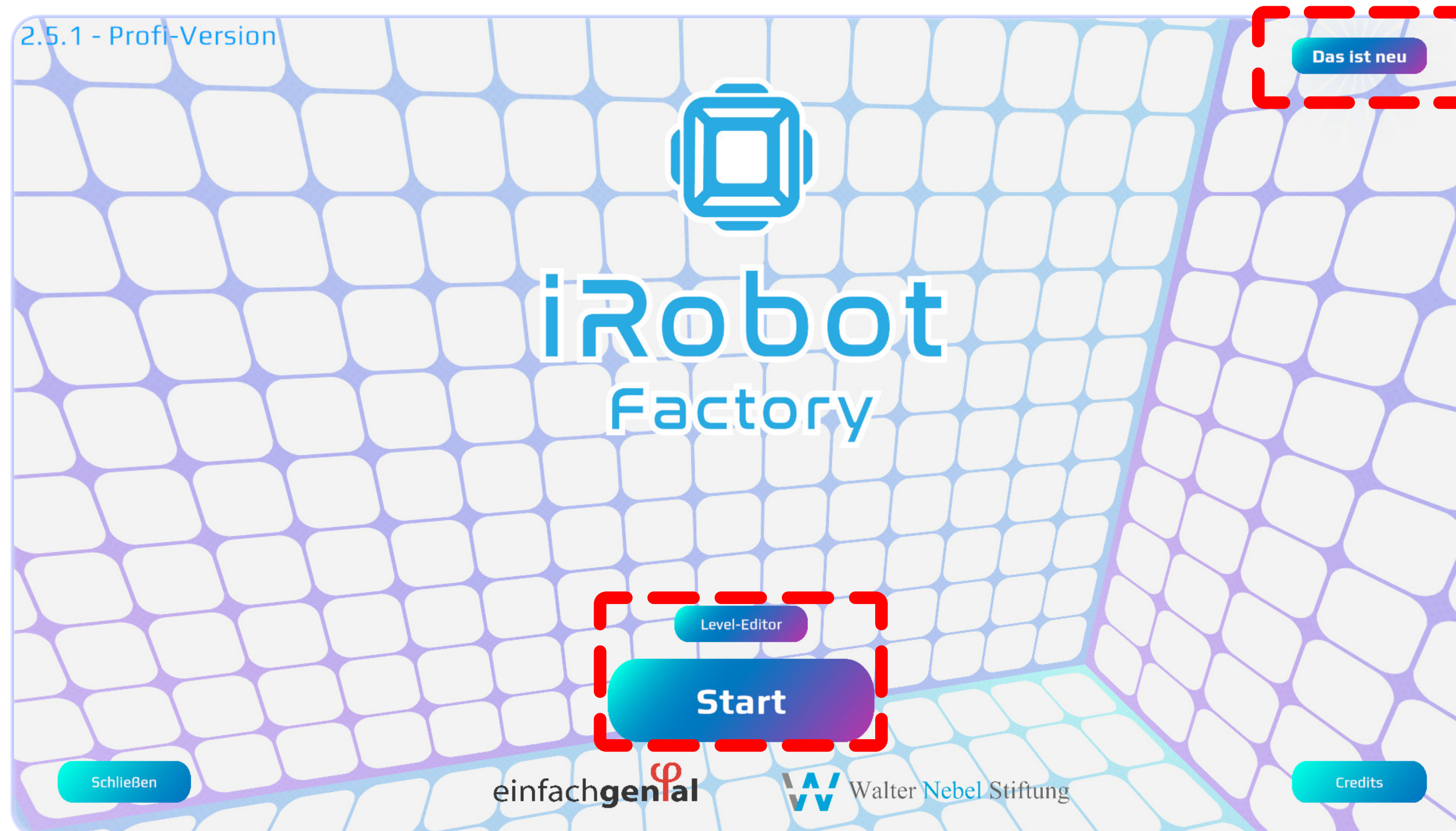
# iRobot Factory

## Worum geht's?



In **iRobot Factory** baust und programmierst du verschiedene Roboter, um mit ihnen unterschiedliche Parkours mit diversen Herausforderungen zu absolvieren.

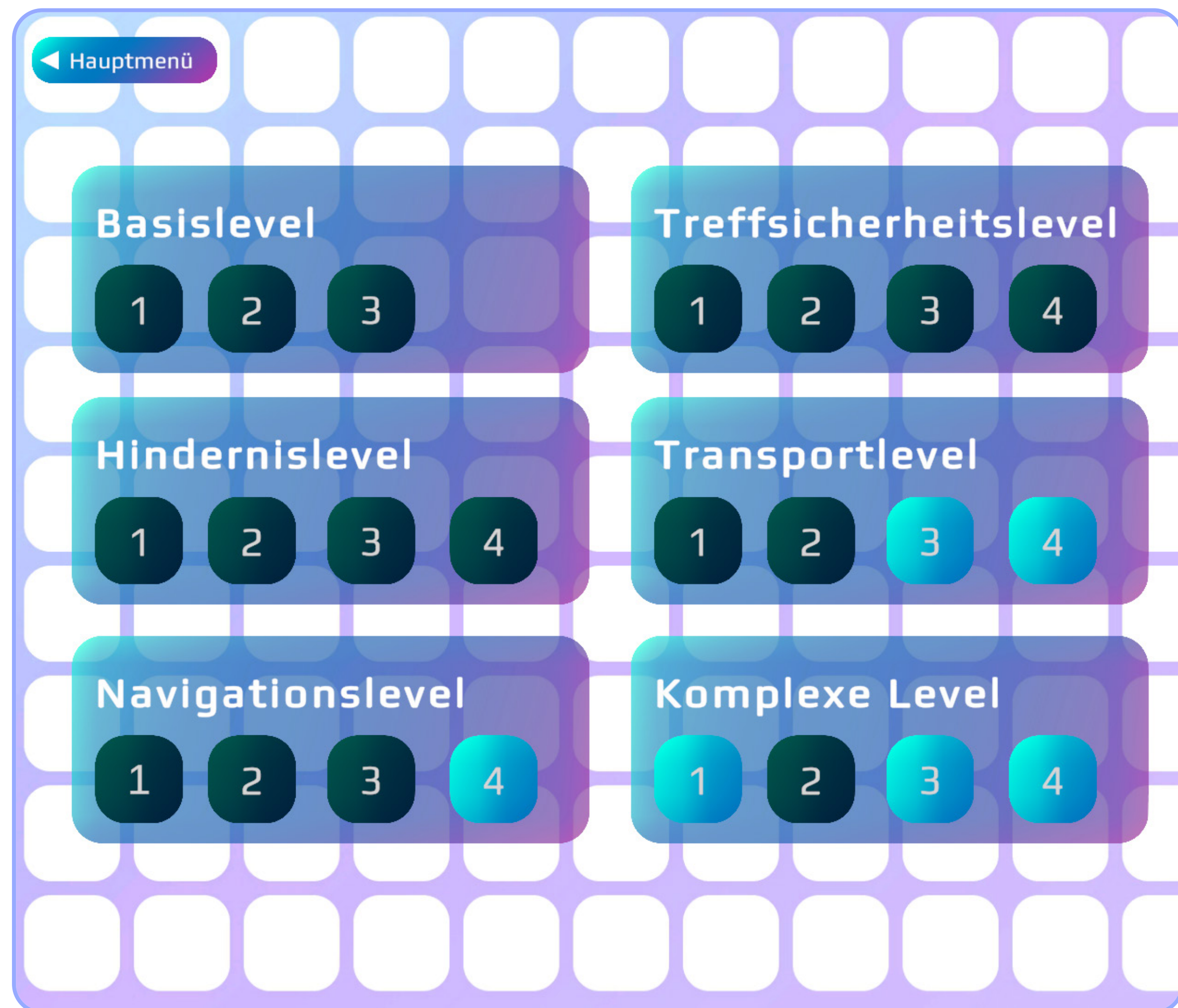
- Baue modulare Roboter!
- Erwecke deine Kreationen mittels einfacher Programmierung zum Leben!
- Stelle dich vielfältigen Herausforderungen und überwinde sie mit deiner ganz individuellen Lösung!



Über das Hauptmenü findest du zu allen Bereichen der iRobot Factory:

- **Start** bringt dich zur Levelauswahl und damit zum eigentlichen Spiel
- **Level-Editor** bringt dich zum Level-Editor, über den du eigene Parkours erstellen und dich an den Leveln anderer versuchen kannst  
(nur in der Profi-Version verfügbar)
- Unter **Das ist neu** findest du alle Informationen zu aktuellen Updates und Änderungen am Programm

Außerdem findest du hier natürlich die Credits und die Schließen-Taste zum Beenden der Anwendung



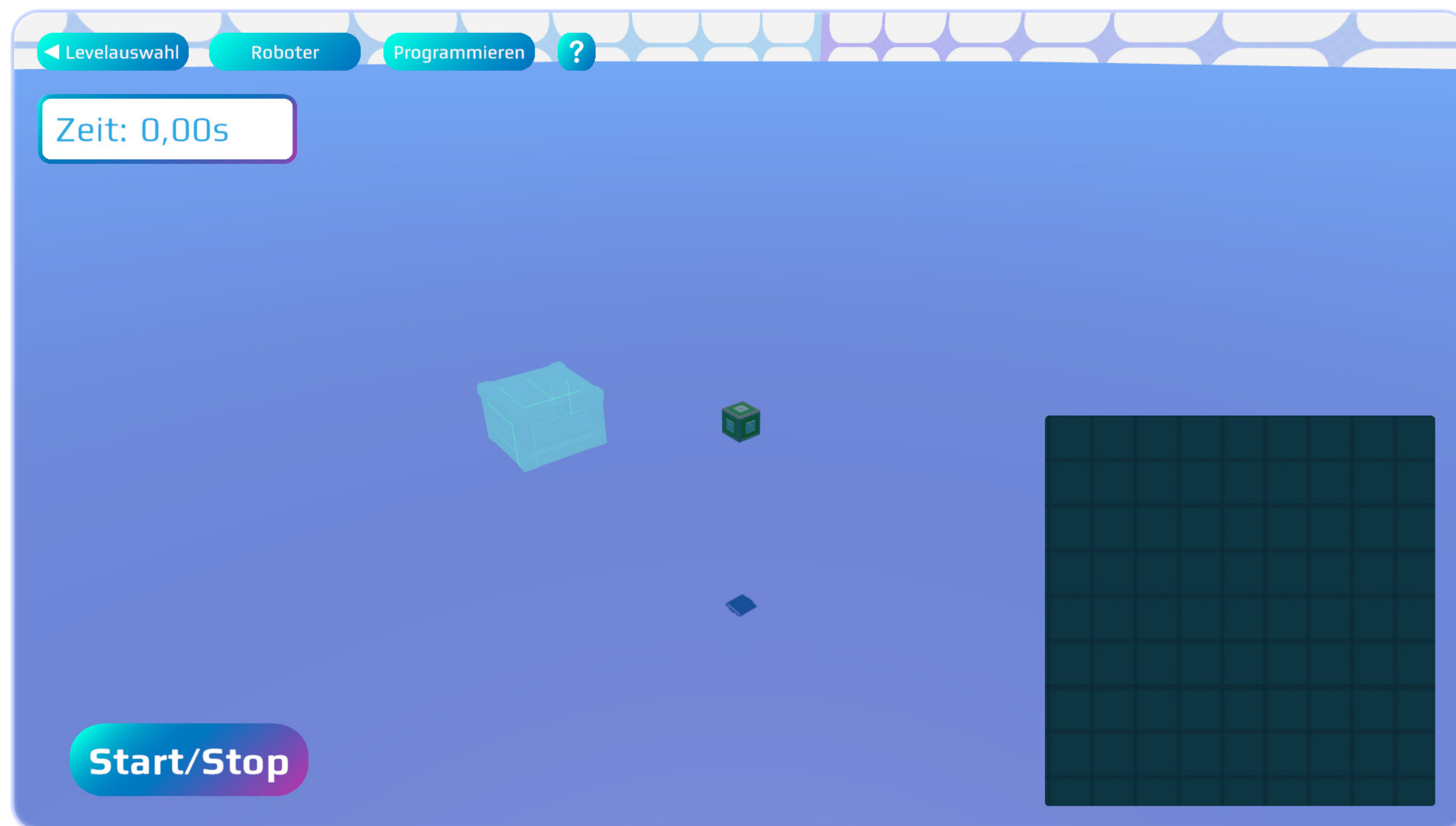
Über dieses Menü gelangst du zu den **einzelnen Leveln** der iRobot Factory.

Die Level hier sind nach verschiedenen Schwerpunkten und (grob) nach Schwierigkeit sortiert.

Bereits erfolgreich absolvierte Level sind dunkel hinterlegt.

**Klicke auf eines der Zahlenfelder** um in das entsprechende Level zu gelangen.

Die iRobot Factory merkt sich automatisch dein letztes verwendetes Roboter-Design, wenn du zwischen verschiedenen Leveln wechselst!



Lädst du ein Level, erwartet dich die Levelansicht, in der du den Parkour, den es zu meistern gilt, in Ruhe anschauen und dir überlegen kannst, welche Funktionen dein Roboter benötigen wird.

Natürlich haben alle vorgefertigten Parkours im Spiel eine vorgesehene Lösung; das heißt aber nicht, dass das die einzige Lösung sein muss.

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



Über die Felder oben gelangst du zu den Interfaces, mit denen du deinen Roboter für dieses Level anpassen kannst:

- **Roboter** bringt dich zur Werkstatt, in der du deinen Roboter modular umbauen kannst.
- **Programmieren** bringt dich zum Programmier-Interface, über welches du die funktionalen Teile deines Roboters grafisch programmieren – also ihr Verhalten bestimmen – kannst.

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



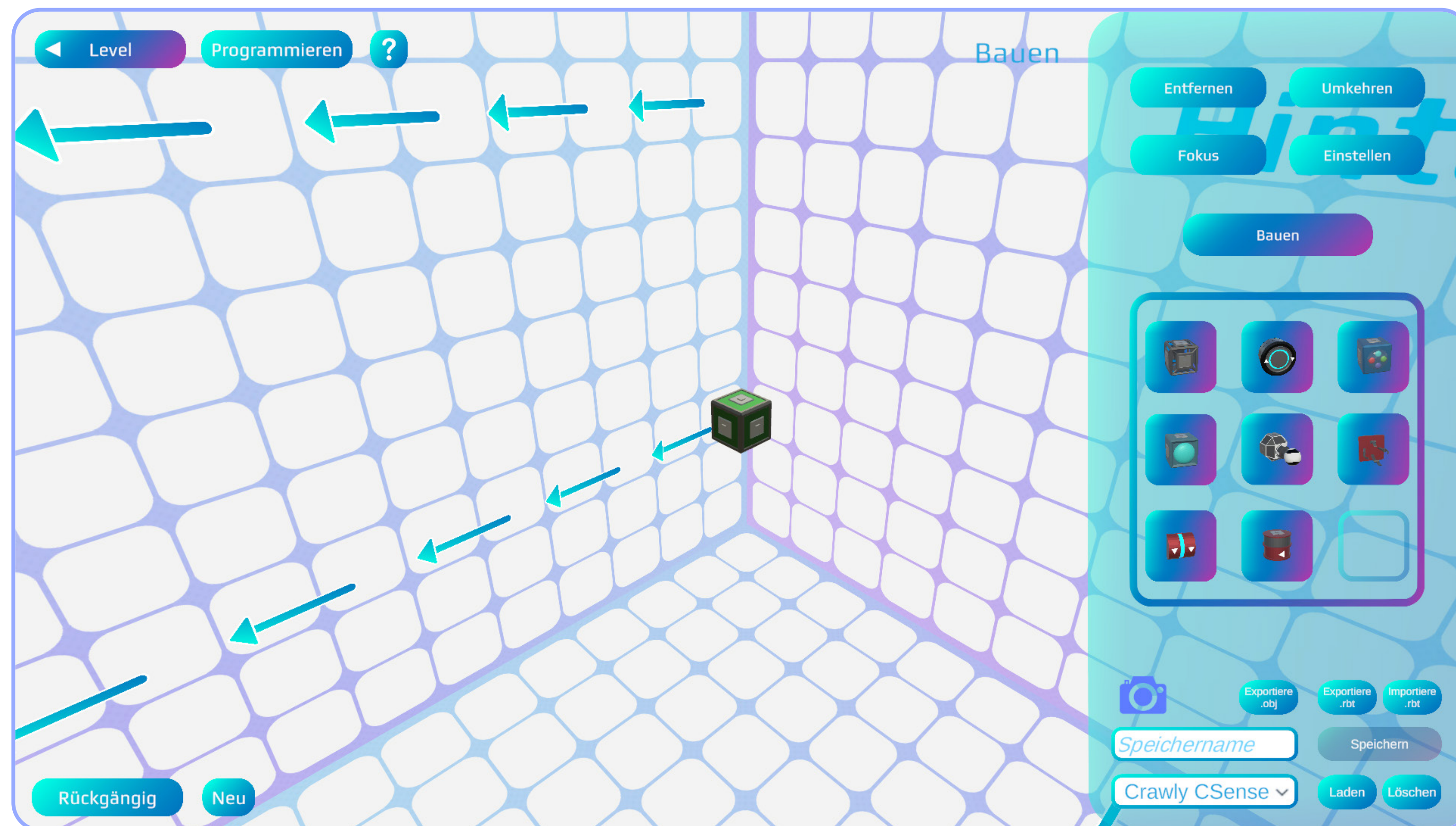
**Start/Stop** startet bzw. beendet die Simulation; sobald sie startet verhält sich dein Roboter entsprechend deiner Programmierung!

Der **Timer** misst die Zeit deines aktuellen Versuchs.

Das große Feld unten rechts ermöglicht dir das **Verfolgen deines Programmier-Graphen in Echtzeit**. Hier kannst du also sehen wie sich Einflüsse des Levels auf deinen Roboter bzw. sein programmiertes Verhalten auswirken – verhält er sich nicht wie erwartet siehst du vermutlich hier warum! (Manchmal musst du den zu sehenden Ausschnitt innerhalb des Fensters etwas verschieben.)

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



In diesem Modus baust du aus Einzelteilen deinen Roboter.

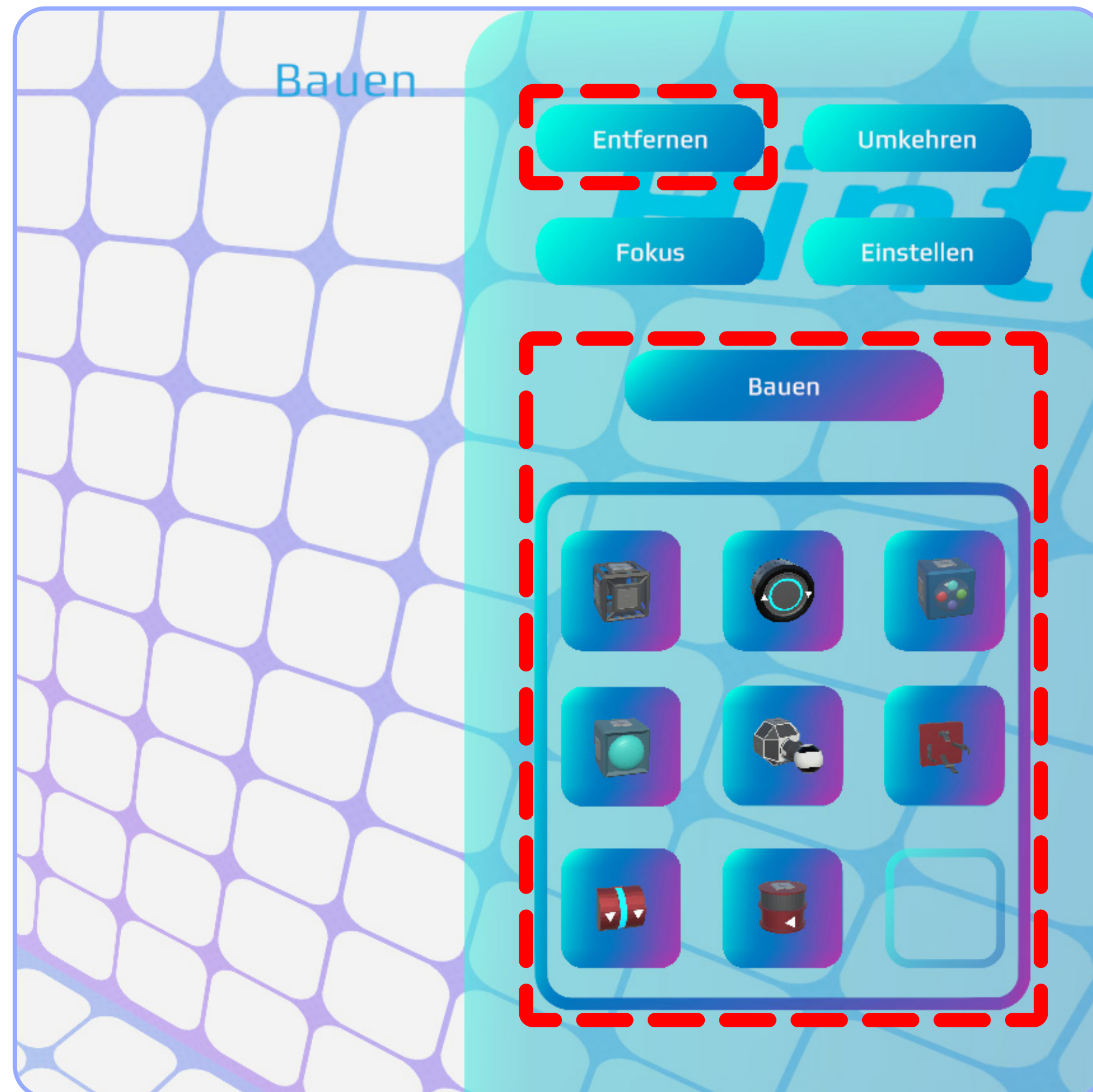
Ausgangspunkt dafür ist immer der **grüne** Grundstein.

Um diesen kannst du in einem **19x19** Blöcke großen Bereich verschiedene Teile **anbauen**; das heißt in alle Richtungen hast du vom Grundstein aus gesehen **9 Einheiten Platz**.

Beachte beim Bauen, dass Vorne, Hinten, Rechts und Links vom Spiel vordefiniert sind.

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



Dir stehen zum Bauen deines Roboters mehrere Werkzeuge zur Verfügung:

- Unter **Bauen** findest du alle dir für dieses Level zur Verfügung stehenden Teile. Um eines anzubauen, wähle das entsprechende Teil aus und dann auf die Seite eines bereits platzierten Bauteils, an der es angebaut werden soll. Du kannst sie auch mit **E** rotieren (bzw. um 90° drehen).\*
- Drücke auf **Entfernen** und danach auf das Teil, welches vom Roboter abgebaut werden soll, um es zu löschen. Beachte dabei, dass alle Teile, die keine Verbindung zum Grundstein haben, automatisch (mit)entfernt werden.

\*Für verschiedene Level steht dir auch eine unterschiedliche Auswahl an Bauteilen zur Verfügung. Um mehr über ein Bauteil zu erfahren, hovere einfach über dem entsprechenden Feld bzw. halte es gedrückt oder sieh auf den entsprechenden Infokarten (**türkis**) nach!

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

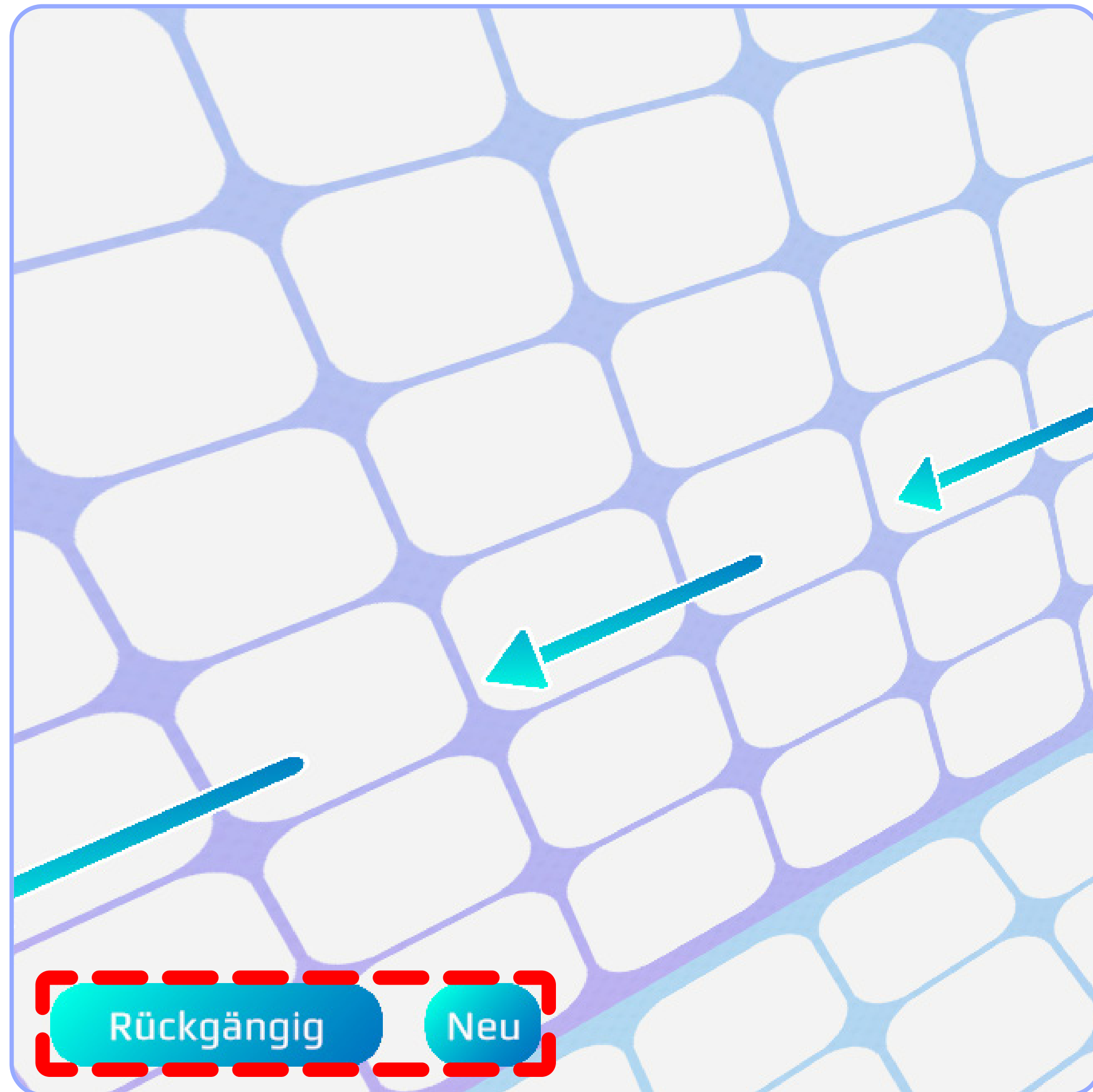
*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



- Bauteile wie Gelenke oder Räder können in ihrer Funktionsrichtung umgekehrt werden. Drücke dazu einfach **Umkehren** und dann auf das gewünschte funktionale Modul am Roboter.
- Alle funktionalen Bauteile können unabhängig voneinander bearbeitet werden. Nutze hierfür das Werkzeug **Einstellen** und wende es auf das Modul an, dessen Funktionsweise du anpassen möchtest. Beachte bitte, dass die meisten Bauteile mit einer Standardeinstellung versehen sind; Farbsensoren musst du allerdings immer manuell einstellen. (Denke daran, Teile zur besseren Unterscheidung individuell zu benennen!)
- **Fokus** lässt dich den Punkt, um den die Kamera rotiert, neu bestimmen. Wähle dazu einfach dieses Werkzeug aus und wähle dann das Bauteil an, welches der neue Ankerpunkt sein soll. Normaler-

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

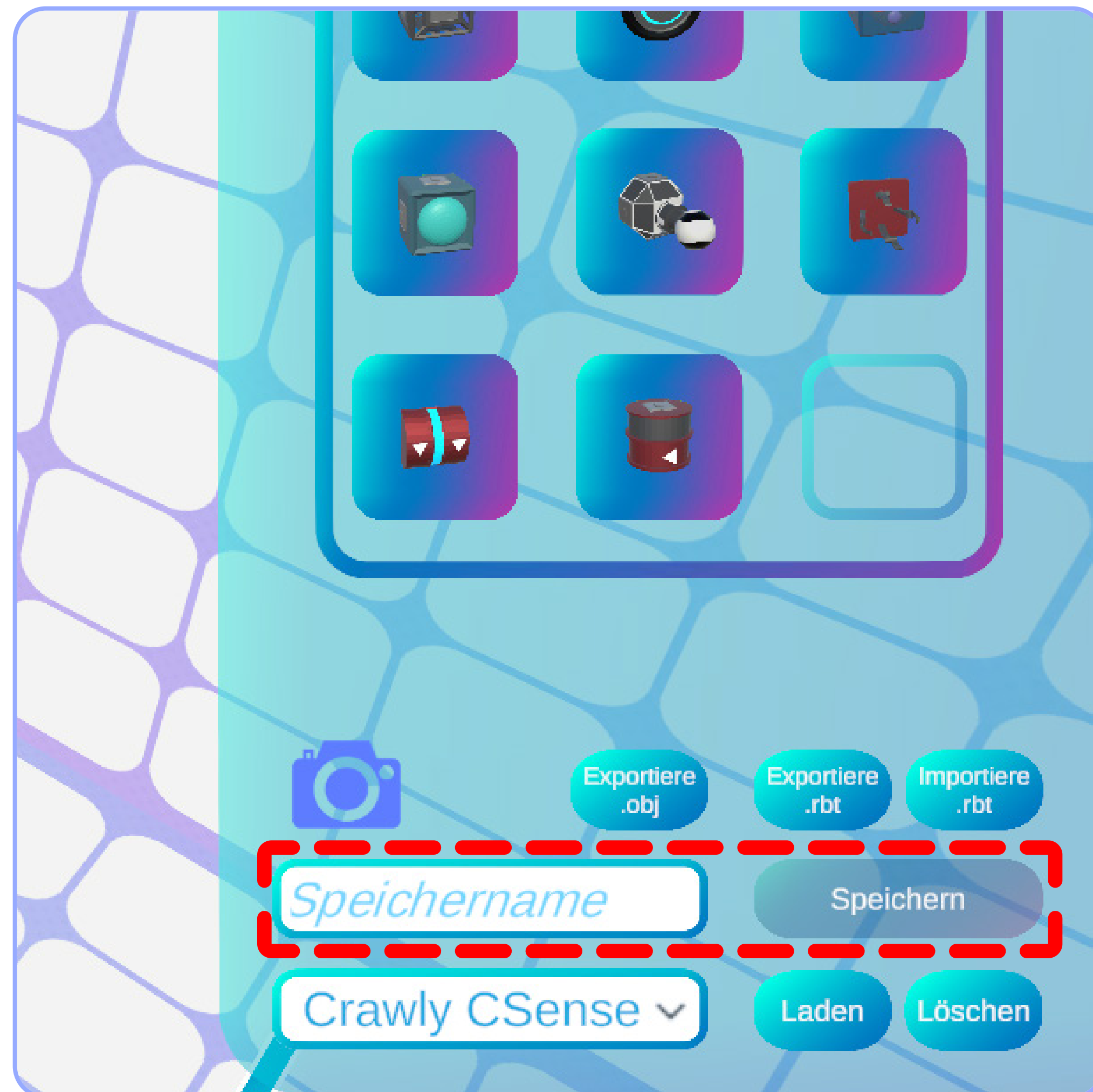
*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



Zusätzlich kannst du einfach deinen letzten vorangegangenen Bauschritt **rückgängig** machen oder gleich den Roboter auf seinen Grundstein zurücksetzen und damit **neu** anfangen.

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



In diesem Modus gibt es auch Optionen zum Auf-nehmen von **Screenshots** sowie zum Benennen und **Speichern** sowie zum **Export und Import** von Robotern. Du kannst ihn auch als 3D-Objekt exportieren.

Möchtest du deinen Roboter – inkl. Programmierung – speichern, gib ihm einfach im Textfeld einen Namen und drücke dann auf **Speichern**.

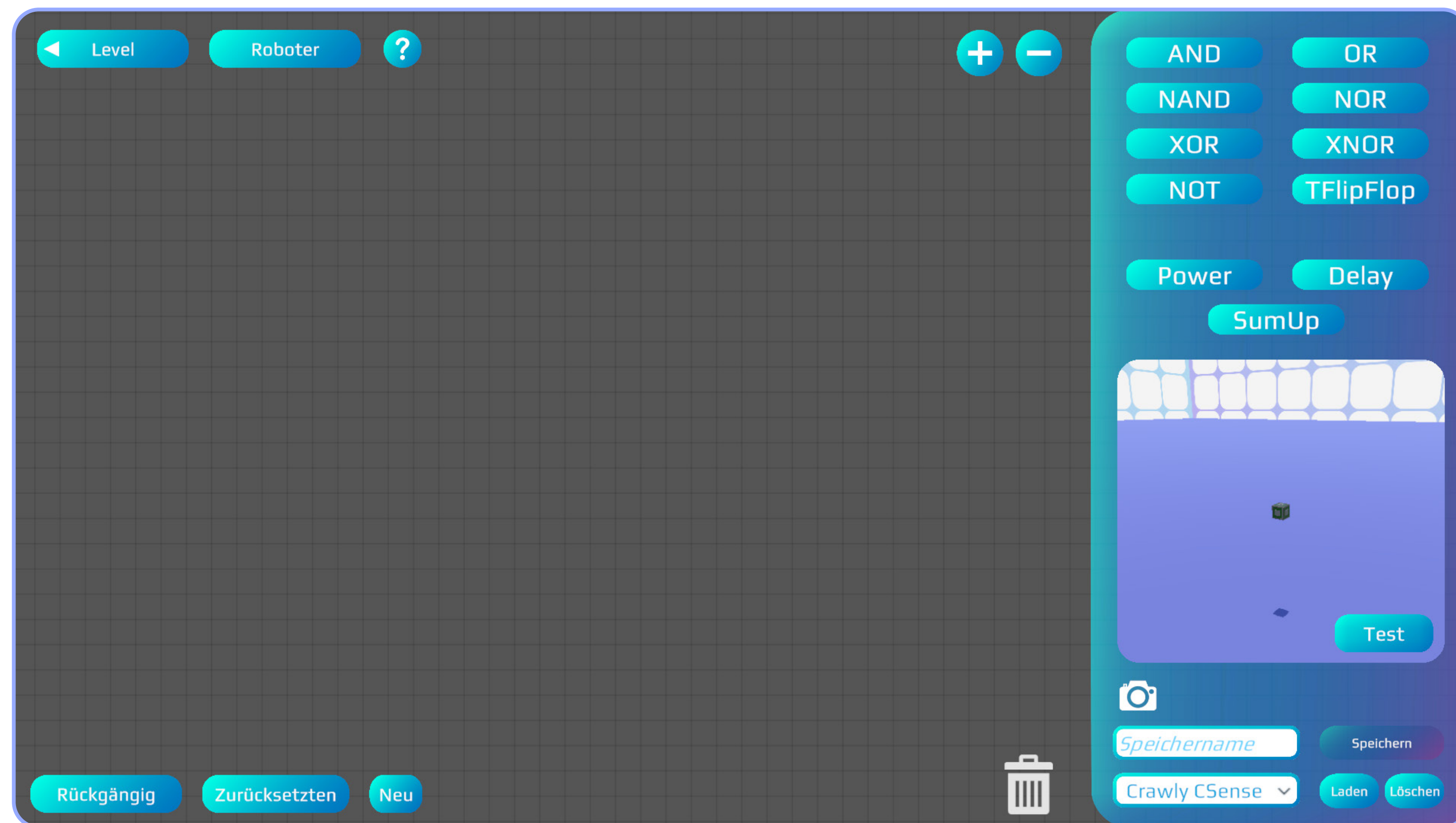
Bitte beachte, dass du bei gleichem Namen deinen alten gespeicherten Roboter überschreibst.

Natürlich kannst du auch ein existierendes Design löschen, wenn du es nicht mehr benötigst.

Für eine Erklärung zu Im- und Export von Robotern

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*

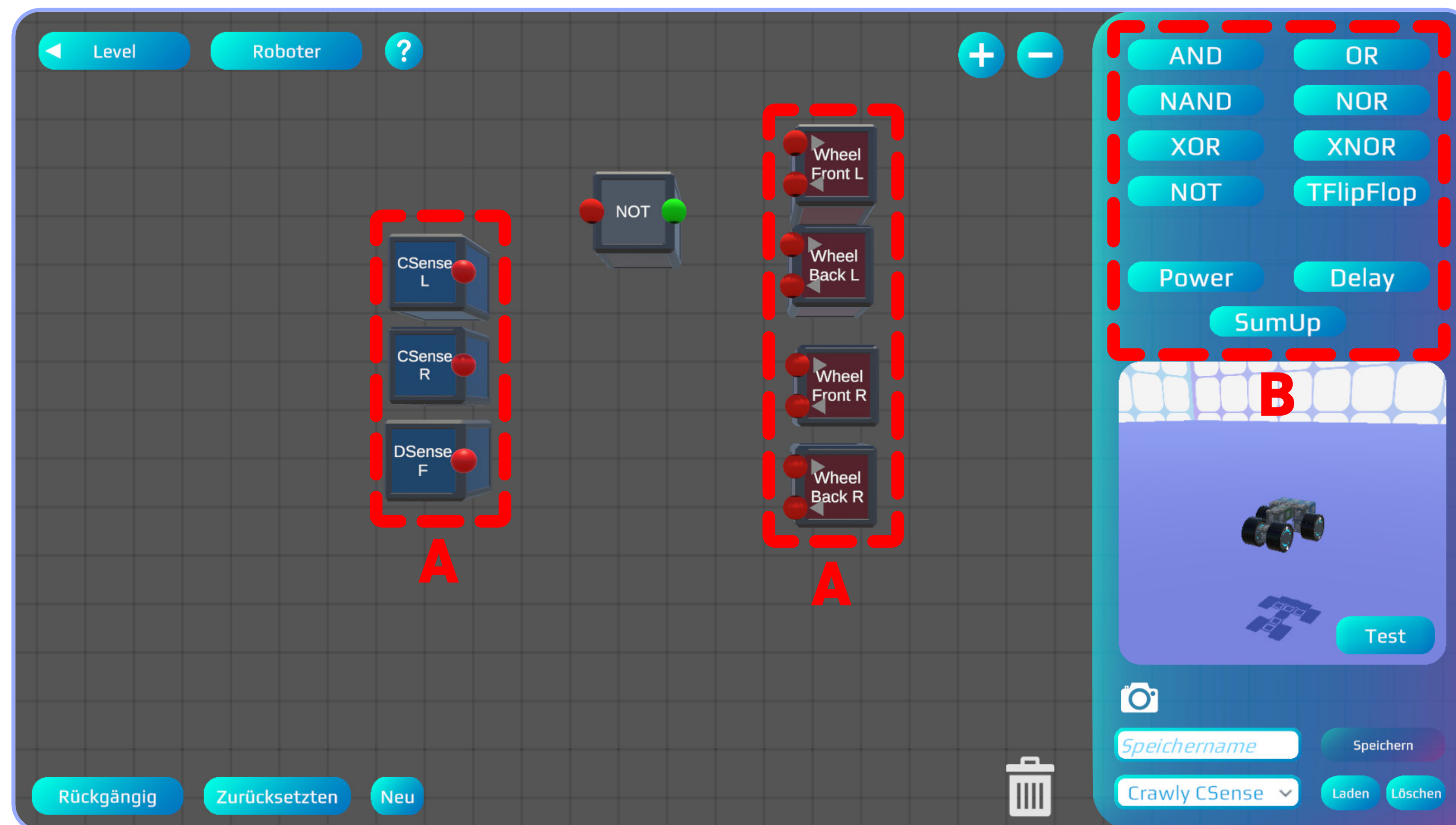


Hier im **Programmier-Interface** kannst du das Verhalten deines Roboters auf Basis der in ihm verbauten funktionalen Teile sowie einer Auswahl an Logikbausteinen festlegen.

Das Programmieren erfolgt grafisch: **Funktionsteile** und **Logikbausteine** existieren in diesem Interface als Baublöcke mit Ein- und/oder Ausgängen (die kleinen Kugeln links und rechts), die du entsprechend der gewünschten Funktionsweise verbinden musst.

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



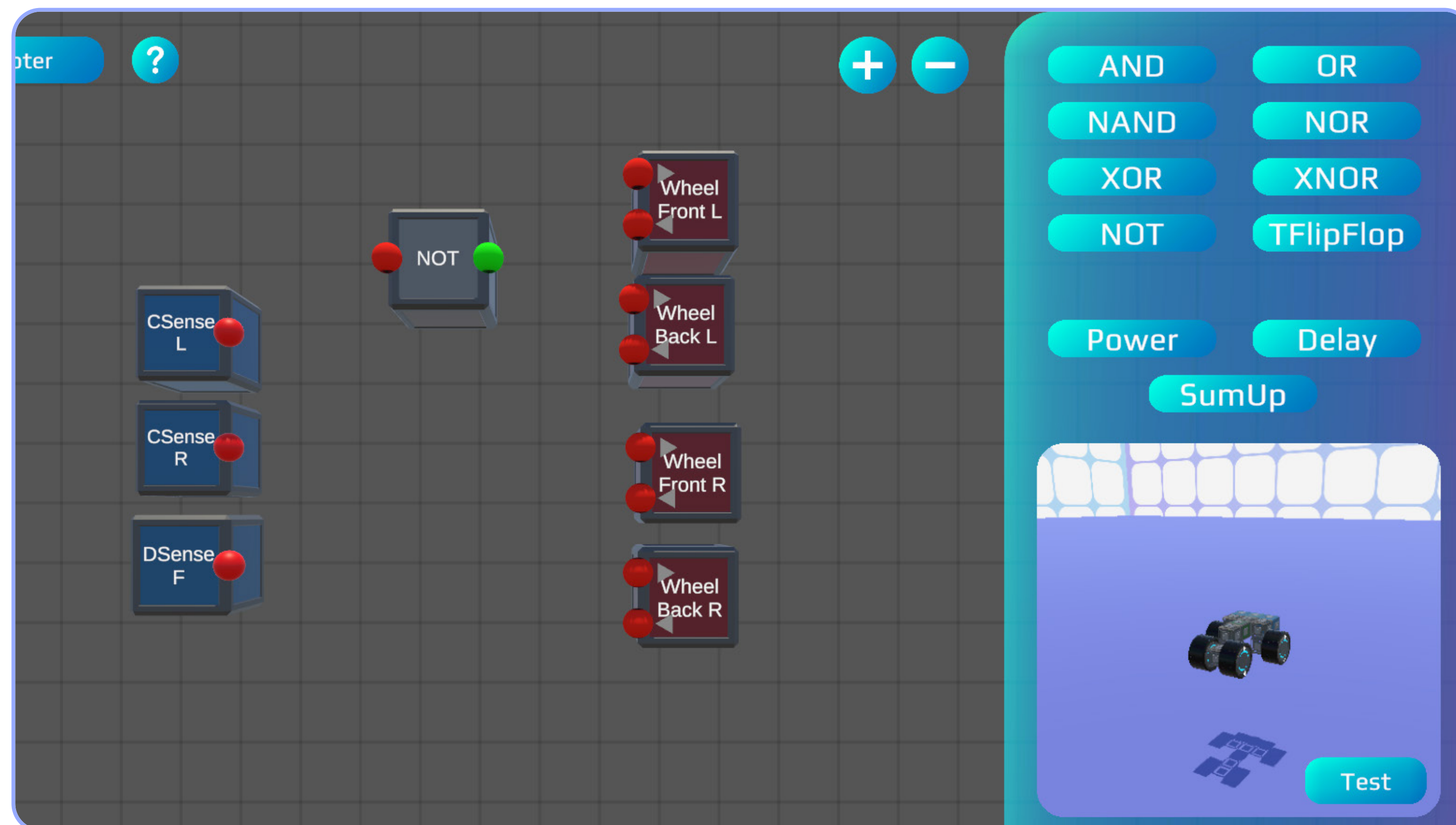
Die in deinem Roboter verbauten **funktionalen Teile (A)** werden automatisch als Programmierbausteine in dieses Interface importiert.

**Logikbausteine (B)** wählst du einfach aus und platzierst sie dann ebenso als Programmierblöcke im Graphen. (Wenn du eine genauere Erklärung der Verknüpfungen brauchst, schau einfach auf den entsprechenden Info-karten (**grün**) nach!)

Streng genommen sind **Power**, **SumUp** und **Delay** weder funktionale Roboterteile noch Logikbausteine; sie können aber in der iRobot Factory vieles vereinfachen bzw. sehr hilfreich sein.

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*

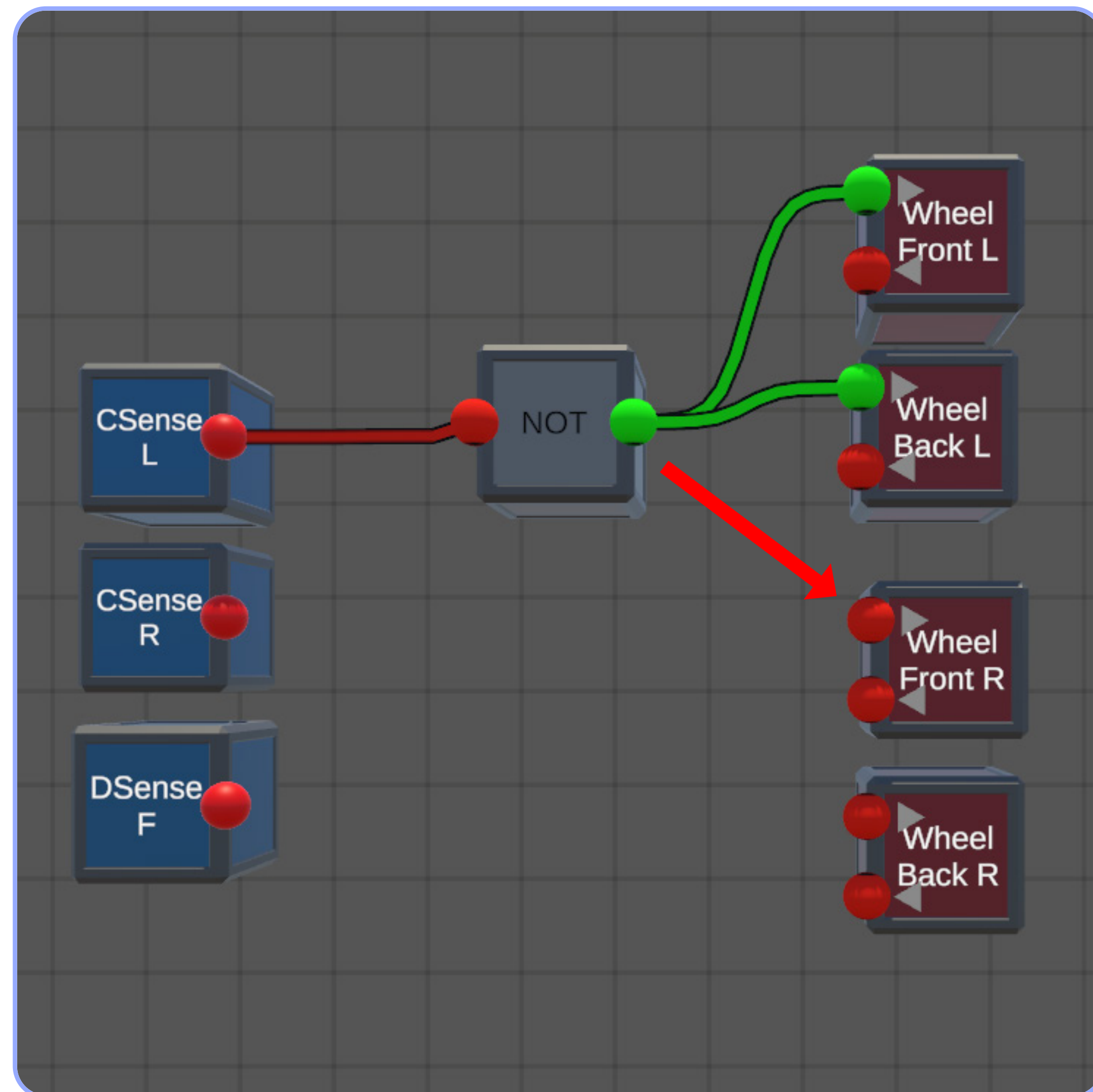


Um einen Block zu bewegen, halte ihn einfach gedrückt und verschiebe ihn nach Belieben; möchtest du einen Block löschen, bewege ihn einfach in den Papierkorb.

Dies löscht automatisch alle mit diesem Block bestehende Verbindungen.

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



Verbinden, die Teile in logische Abhängigkeit voneinander setzen und damit programmieren kannst du die einzelnen Programmierblöcke ganz einfach, in dem du einen **Ein- oder Ausgangspunkt gedrückt hältst** und die sich dann ergebende Linie zu einem anderen Ein- bzw. Ausgangspunkt ziehst.

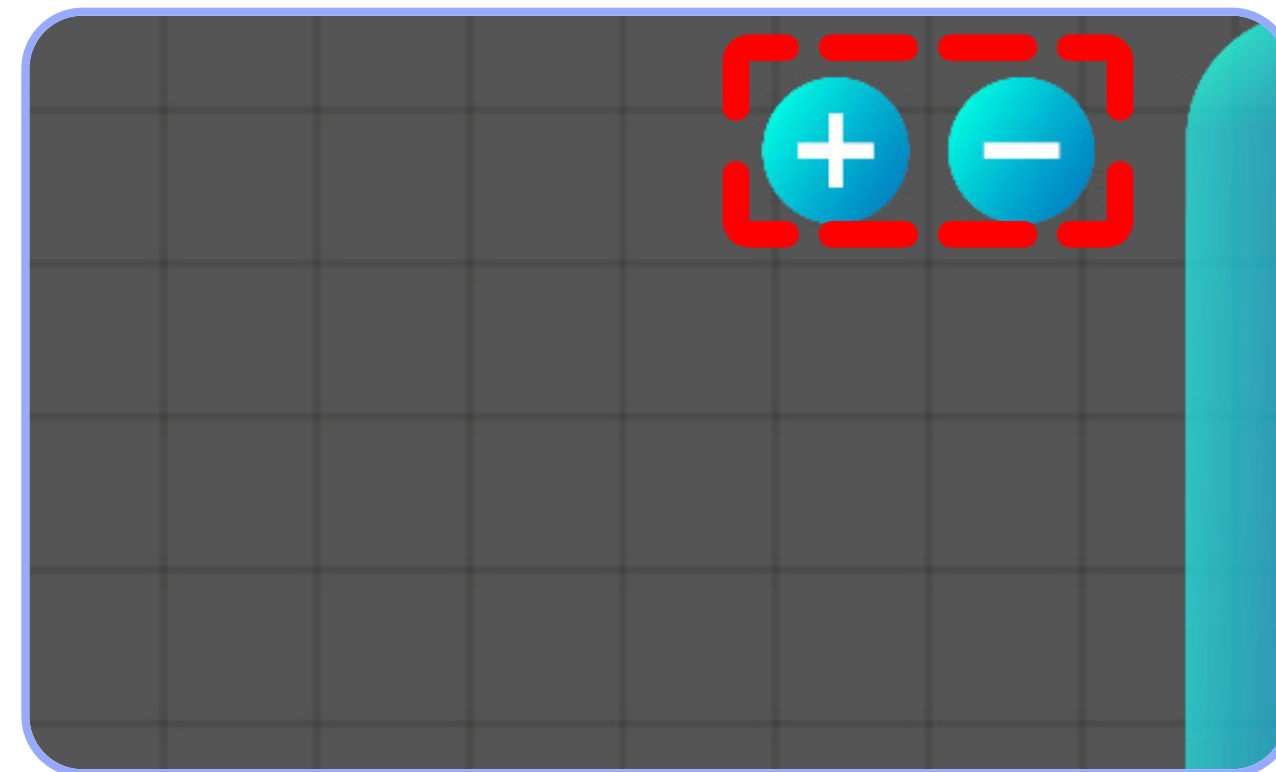
Hierbei ist allerdings wichtig, dass du immer **nur Ein- mit Ausgängen verbinden** kannst, wobei Eingänge immer links, Ausgänge immer rechts an einem Programmierblock zu finden sind.

Außerdem kann ein Ausgang mehrere, ein Eingang aber nur eine Verbindung haben – hier helfen dir AND, OR usw.!

**Ein gerade aktives Signal – sowohl in Punkt- als auch Linienform – leuchtet grün.**

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



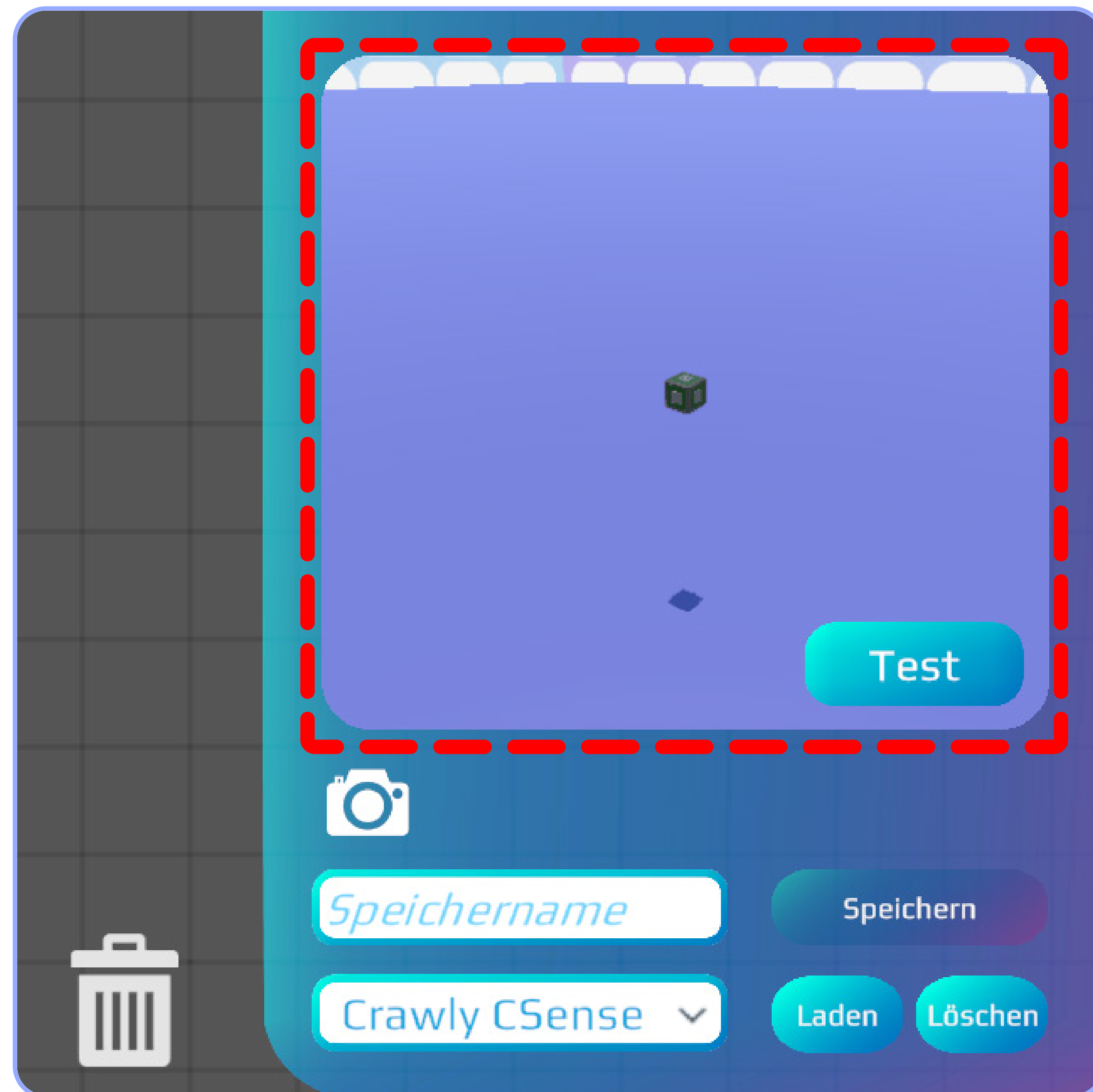
Mit +/- kannst du hier – neben den üblichen Optionen – auch stufenweise herein- und herauszoomen.



Wie im Baumodus hast du auch im Programmier-Interface die Optionen, deinen letzten Arbeitsschritt **rückgängig** zu machen oder alle nicht vom Roboter importierten Teile zu entfernen und **neu** zu beginnen. Zusätzlich kannst du hier allerdings über **Zurücksetzen** den Ausgangszustand des Programmiergraphen wiederherstellen. (Das kann v. A. beim Verwenden von **TFlipFlop** oder **Delay** hilfreich sein!)

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



Willst du sehen, wie sich dein Roboter mit der aktuellen Programmierung grundlegend verhält, kannst du ihn über das **Test**-Feld in einer leeren Umgebung ausprobieren.

Das kann vor Allem dann hilfreich sein, wenn du dir hinsichtlich des Ausgangszustands einer Logikschaltung unsicher bist oder nachsehen willst, ob sich bewegliche Teile so verhalten wie von dir gewünscht.

Auch hier im Programmier-Interface hast du die Möglichkeit, Screenshots zu machen und deinen Fortschritt zu **speichern**, einen vorherigen Stand zu laden oder ein Design zu löschen.

*Das ?-Feld gibt dir direkt in der Anwendung eine Erklärung über alle Elemente des aktuellen Interface!*

*Du kannst auch über den einzelnen Feldern hovern bzw. sie gedrückt halten, um einen entsprechenden Tooltip zu erhalten!*



Hier findest du noch einmal eine Übersicht über alle in iRobot Factory verfügbaren **Bauteile**:

- Grundstein **1**
- Baublock **2**
- Rad **3**
- Distanzsensor **4**
- Farbsensor **5**
- Ballmaschine **6**
- Greifer **7**
- Gelenk **8**
- Rotationsblock **9**
- Taktblock **10**

Für alle Details zu den entsprechenden Teilen schau einfach auf den folgenden Infokarten nach!



### Bild von Grundstein-Asset

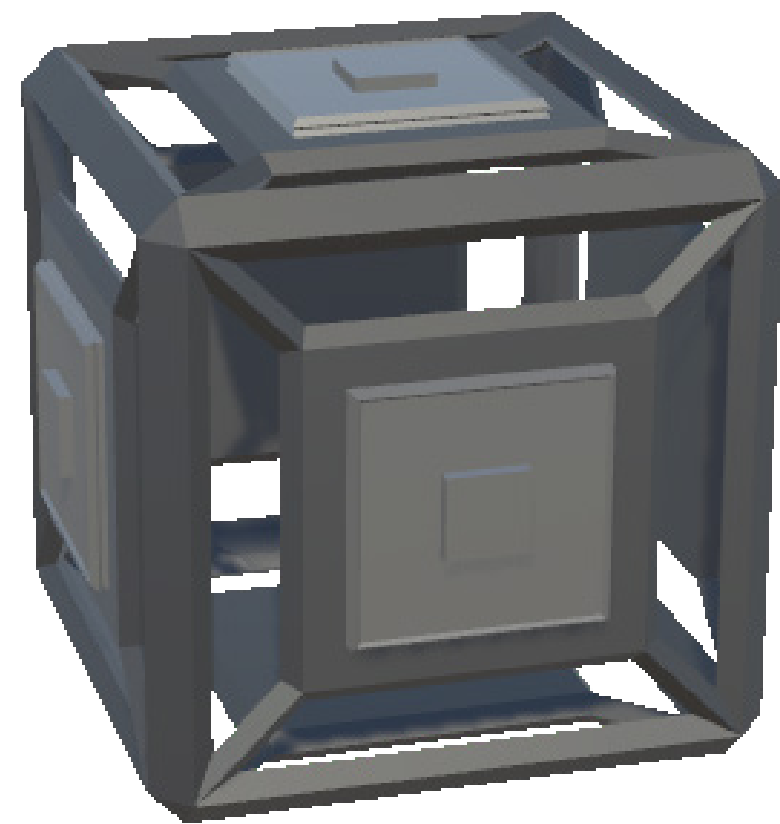
Der **Grundstein** dient als **Ausgangspunkt** für jeden Roboter der iRobot Factory; er ist von vornherein gegeben und **kann nicht gelöscht werden**.

**Alle anderen Teile des Roboters müssen eine Verbindung zum Grundstein aufweisen** – wird die Verbindung eines Elements zu ihm entfernt, wird das betreffende Element automatisch gleichfalls gelöscht.

Die Maße des Roboters sind in Abhängigkeit zum Grundstein zu sehen. Du kannst vom Grundstein aus **9 Blöcke in alle Richtungen** bauen – es ergibt sich also ein **19x19** Einheiten großer Raum, in welchem du Baublöcke platzieren kannst.

Beachte diese räumlichen Einschränkungen, wenn du anfängst zu bauen. Eventuell muss der Grundstein für dein Design nicht in der „intuitiven Mitte“ des Roboters platziert sein – z.B. wenn du einen besonders langen Greifarm bauen möchtest.

Der Grundstein dient außerdem als initialer Ankerpunkt für die Kamera.



Der **Baublock** ist dein Mittel, die funktionalen Teile deines Roboters zu verbinden und seine Dimensionen anzupassen.

Er verfügt über **keine besonderen Eigenschaften**, hat aber – wie alle anderen Bauteile auch – ein Eigengewicht, welches das Verhalten des Roboters oder von Robotersegmenten entsprechend beeinflusst.

Das lässt sich auch bewusst/gezielt nutzen – beispielsweise, wenn ein Greifarm deinen Roboter zu sehr aus dem Gleichgewicht bringt.



**Rad**-Module dienen deinem Roboter als primäres Fortbewegungswerkzeug.

Sie **können sich dabei in beide Richtungen bewegen** – je nach dem welcher ihrer Eingänge im Programmierinterface angesteuert wird. Du kannst ihre „Grundausrichtung“ auch im Baumodus umkehren.

Im Baumodus kannst du außerdem **einstellen, wie schnell bzw. mit wie viel Kraft das angewählte Rad sich drehen soll.**

Beachte – sowohl im Bau- als auch im Programmierinterface –, dass du jedes Rad einzeln einstellen bzw. ansprechen musst!

Das Rad ist zudem eines von zwei Bauteilen, dessen **Raumbedarf** 1x1x1 Einheiten überschreitet – es ist größer als der Standardblock und nimmt faktisch einen Raum von **3x3x1** ein.



Der **Distanzsensoren**-Block hat eine definierte, einfach erkennbare **Vorderseite, die in der Lage ist, Objekte zu erkennen**, die sich in einer geraden Linie vor ihr befinden **und in diesem Fall ein Signal auszugeben**, welches über das Programmierinterface verwertet werden kann.

Die Distanz, ab welcher der Distanzsensoren ein Signal ausgibt, kann – innerhalb einer gewissen maximalen Reichweite – von dir im Baumodus eingestellt werden.

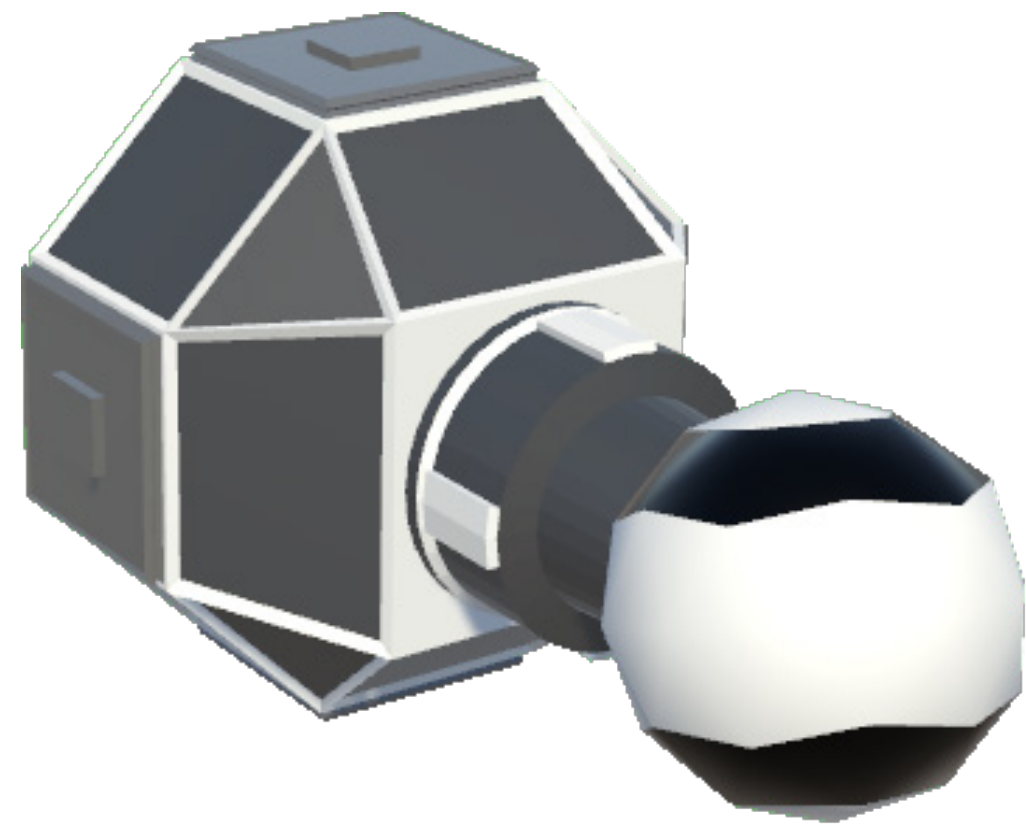
Der Distanzsensoren unterscheidet dabei nicht, was sich vor ihm befindet bzw. welche Eigenschaften das Objekt vor ihm hat – darin unterscheidet er sich vom Farbsensoren [5].



Der **Farbsensor** funktioniert praktisch wie der Distanzsensor [4] – **er ist in der Lage, Objekte, die sich in einer geraden Linie vor seiner gut identifizierbaren Vorderseite befinden, zu erkennen –, legt dabei aber zusätzlich Wert auf deren Farbe.**

Im Baumodus kannst du einstellen, auf welche Farben – **Rot, Grün, Cyan und Blau** stehen dabei zur Wahl – und ab welcher Distanz (wie beim Distanzsensor innerhalb einer Maximalreichweite) er reagieren soll. Objekte anderer Farbe werden von ihm ignoriert.

**Erkennt der Farbsensor also ein Objekt in einer der gewählten Farben, gibt er ein Signal aus,** welches im Programmierinterface verwertet werden kann.



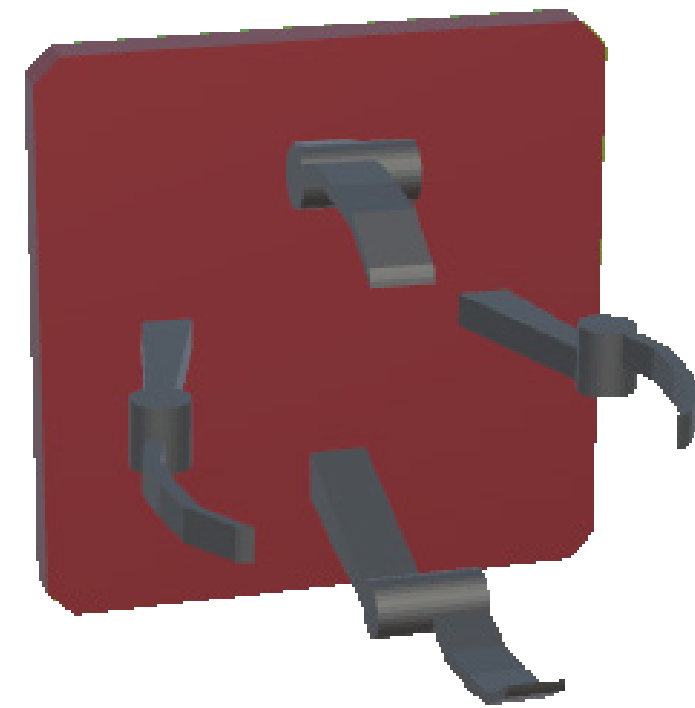
Die **Ballmaschine ist in der Lage, in vorgegebenen Intervallen Bälle zu verschießen**, mit denen du Ziele im Level treffen oder Physikobjekte anderweitig beeinflussen kannst.

Sie besteht dabei aus einem Grundblock und dem Ballsegment und verschießt Bälle immer in der Richtung, in die das Ballsegment von Grundblock her gesehen zeigt bzw. ausgerichtet ist.

Erhält die Ballmaschine im Programmierinterface ein Signal, feuert sie – wie beschrieben – in einem vorgegebenen Intervall Bälle ab.

Den Intervall, in welchem die Ballmaschine Projektile verschießt, kannst du im Baumodus einstellen (innerhalb gewisser Grenzen).

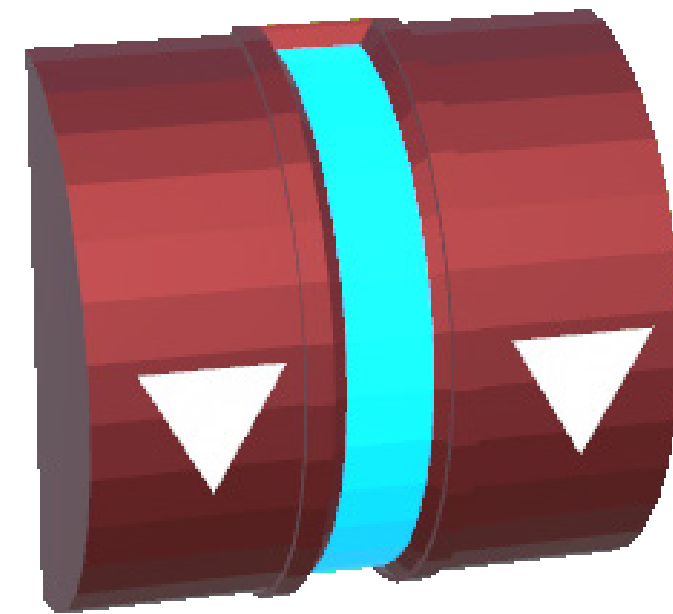
Die Ballmaschine nimmt theoretisch einen Raum von 1x2x1 Einheiten ein; auch mit ihr darf nichts überlagern.



Mit dem **Greifer** lassen sich **andere Physikobjekte aus der Levelumgebung greifen** und somit transportieren.

Er verfügt über **zwei simple Zustände**: geöffnet und geschlossen. Ist er **geöffnet**, kann er theoretisch ein vor bzw. in ihm – die Greifelemente ignorieren Kollision – befindliches Element greifen, wenn er geschlossen wird. Im **geschlossenen** Zustand hält der Greifer ein erfasstes Objekt fest – wobei hier die Physikeigenschaften des fraglichen Objekts zu berücksichtigen sind (ein größeres/schwereres Objekt ist entsprechend umständlicher zu transportieren). In diesem Zustand kann der Greifer keine neuen/zusätzlichen Objekte aufnehmen.

Der Greifer verfügt im Programmierinterface über zwei Eingänge, die dich diese Zustände nach Belieben einnehmen lassen können; er startet immer im offenen Zustand.

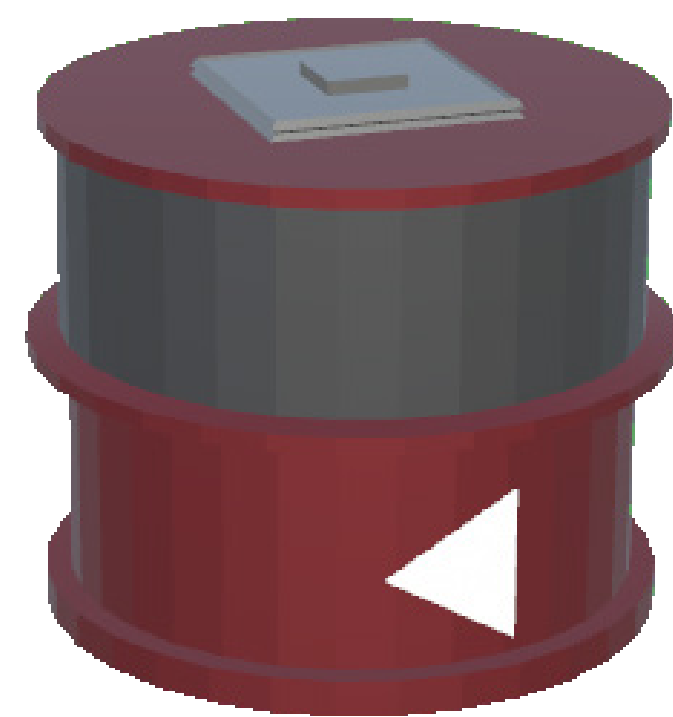


Das (Kipp-) **Gelenk** ermöglicht es, es selbst und an es angebaute Elemente ausgehend von ihrer Ausgangsposition zu kippen.

Im Baumodus kannst du die **Winkel, zwischen denen es wechseln können soll, einstellen**; ebenso die Kraft (und damit indirekt Geschwindigkeit), mit denen diese Bewegung vollzogen werden soll.

Im Programmiermodus verfügt das Gelenk entsprechend über zwei Eingänge, die die verschiedenen Kippzustände herbeiführen, wenn sie entsprechend ein Signal erhalten.

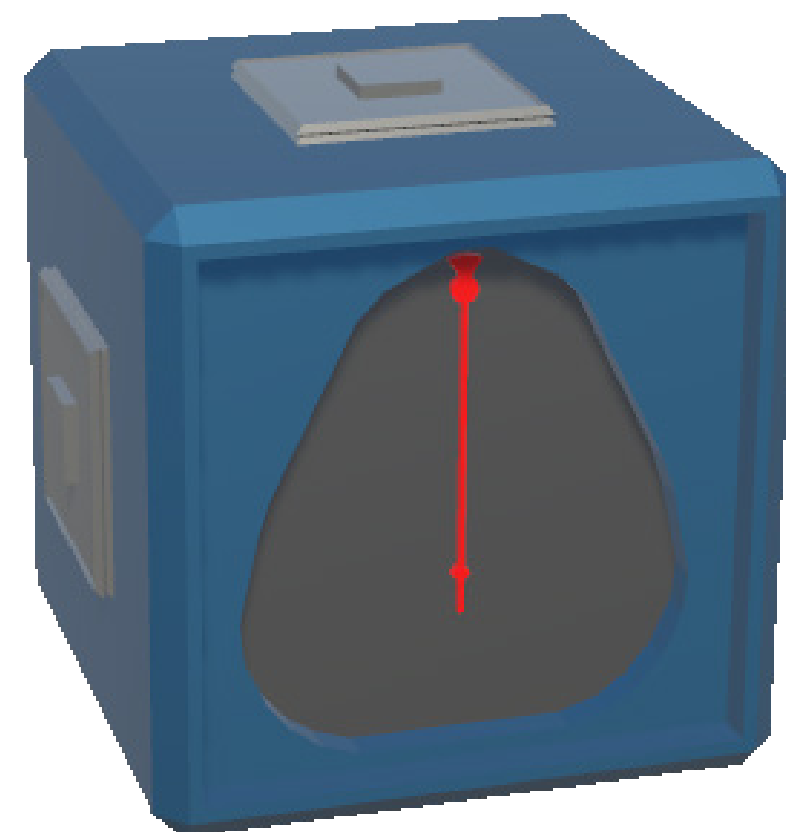
Du kannst das Element im Baumodus auch einfach umkehren, wenn du dich mit dem Einstellen vertan hast.



Der **Rotationsblock ist in der Lage, ihn und alle an ihn angebauten Elemente zu rotieren** – entweder ganz im Kreis oder innerhalb gewisser Winkelbeschränkungen, die du im Baumodus einstellen kannst; **für die meisten Anwendungen wirst du diese Limits benutzen** wollen. Wie beim Kippgelenk [8] lässt sich auch hier die Kraft und Geschwindigkeit, mit der die Rotationsbewegung ausgeführt werden soll, einstellen.

Beachte bei der Verwendung des Rotationsblocks, dass auch der Rest des Roboters (also das, woran du den Rotationsblock angebaut hast) von der Bewegung (dem Schwung) der rotierten Elemente **physikalisch korrekt mitbeeinflusst** werden kann!

Auch der Rotationsblock verfügt im Programmierinterface über zwei Eingänge, mit denen du die beiden Extremzustände (also die beiden eingestellten Winkel) herbeiführen bzw. die Rotationsrichtung ändern kannst.



Der **Taktblock** hat selbst **keine eigene physikalische Funktion**; er kann aber **für die Programmierung nützlich** sein!

Im Programmierinterface gibt der Taktblock **in einem vorgegebenen Intervall ein Signal von ebenfalls vorgegebener Länge ab**, wobei du sowohl den Intervall als auch die Signallänge – innerhalb gewisser Grenzen – frei einstellen kannst.

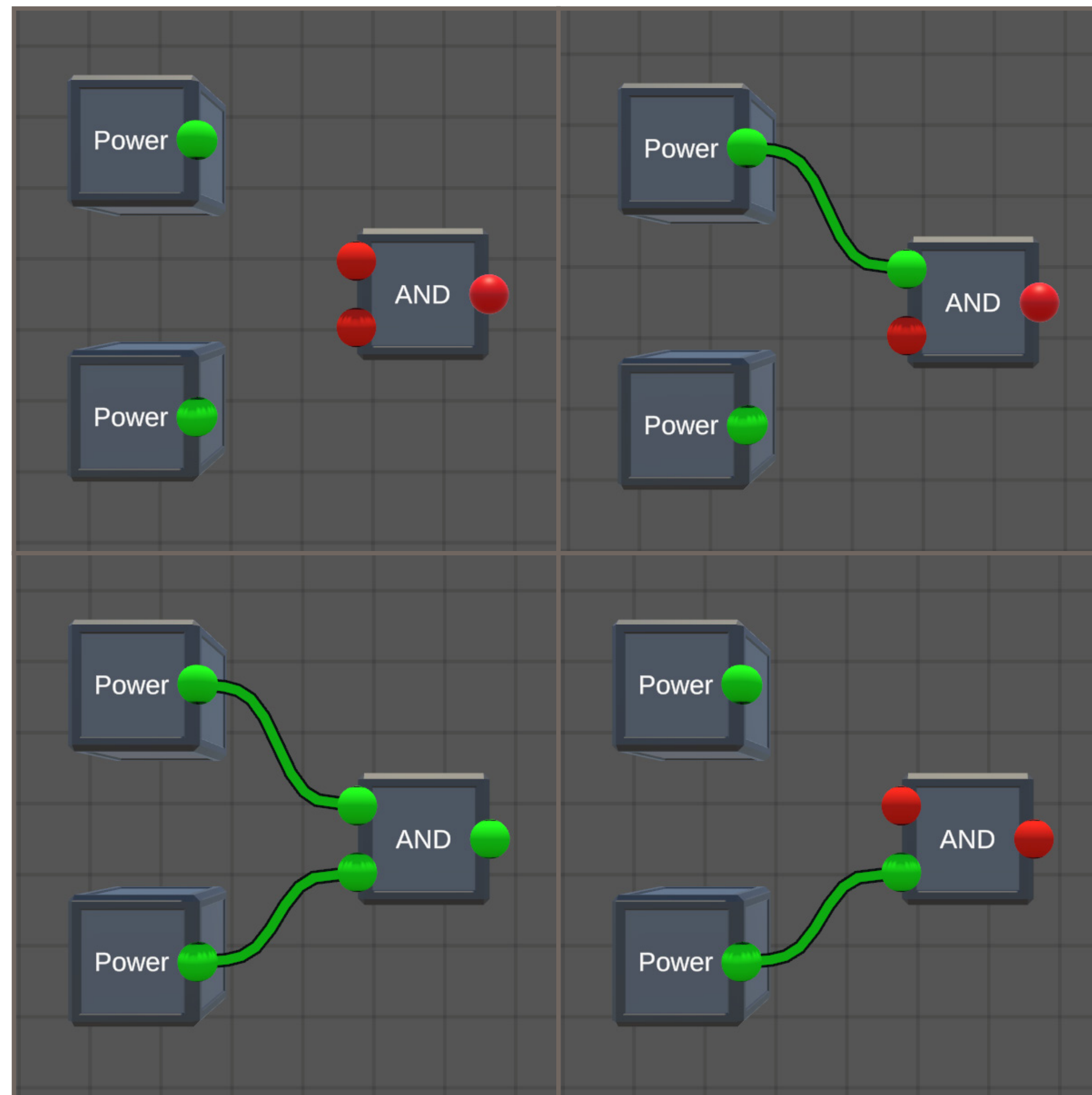
Beachte beim Bauen, dass du die Seite des Taktblocks, auf der das „Metronom“ zu sehen ist, nicht zubauen darfst!



Hier findest du noch einmal eine Übersicht über alle in iRobot Factory verfügbaren **logischen Verknüpfungen** bzw. **Programmierinterface-Bausteine**:

- AND 1
- OR 2
- NAND 3
- NOR 4
- XOR 5
- XNOR 6
- NOT 7
- TFlipFlop 8
- Power 9
- Delay 10
- SumUp 11

Für alle Details zu den entsprechenden Teilen schau einfach auf den folgenden Infokarten nach!

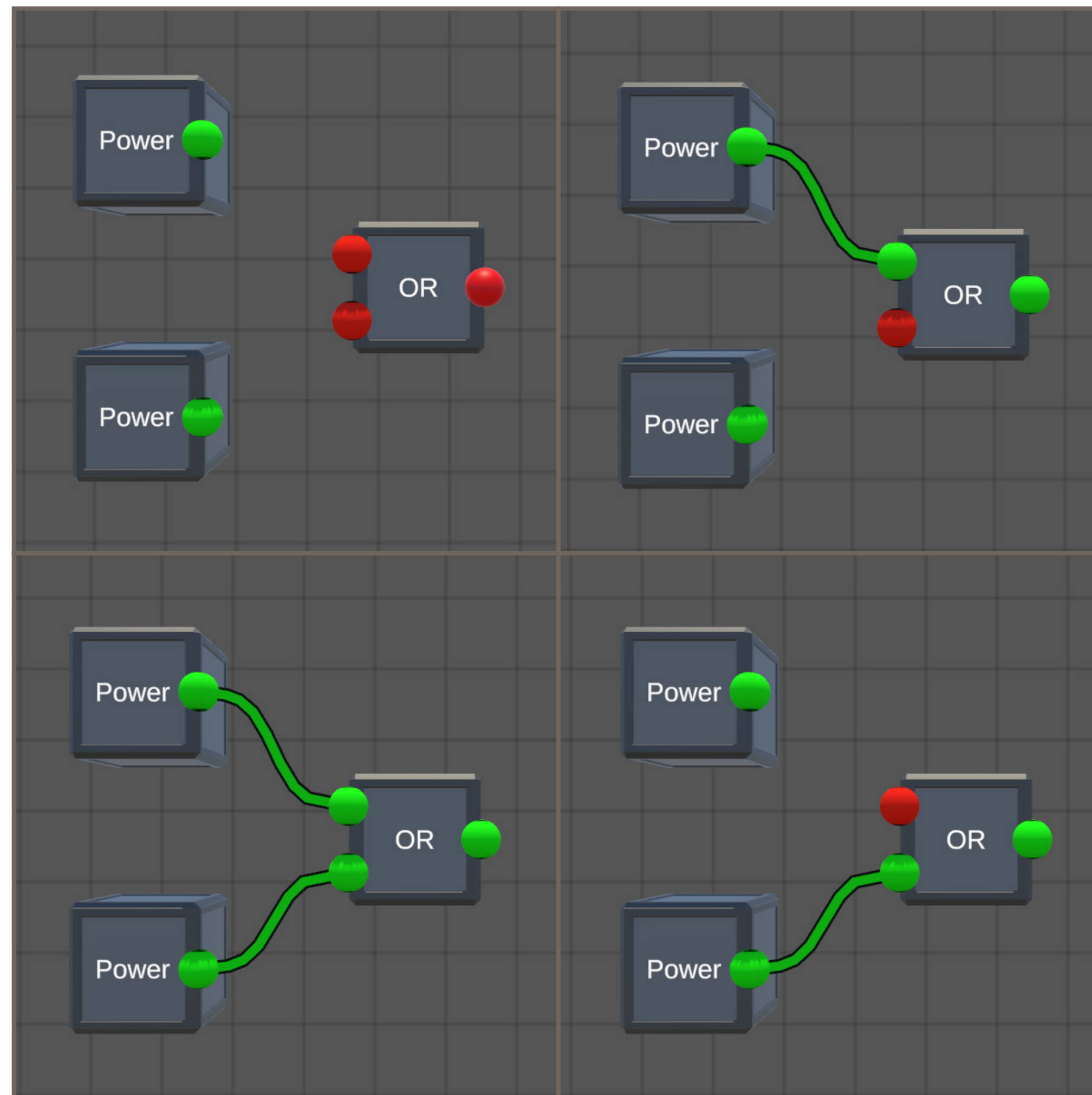


## AND

→ 2 Eingänge

→ 1 Ausgang

→ gibt ein Signal aus, wenn **beide** Eingänge ein Signal erhalten

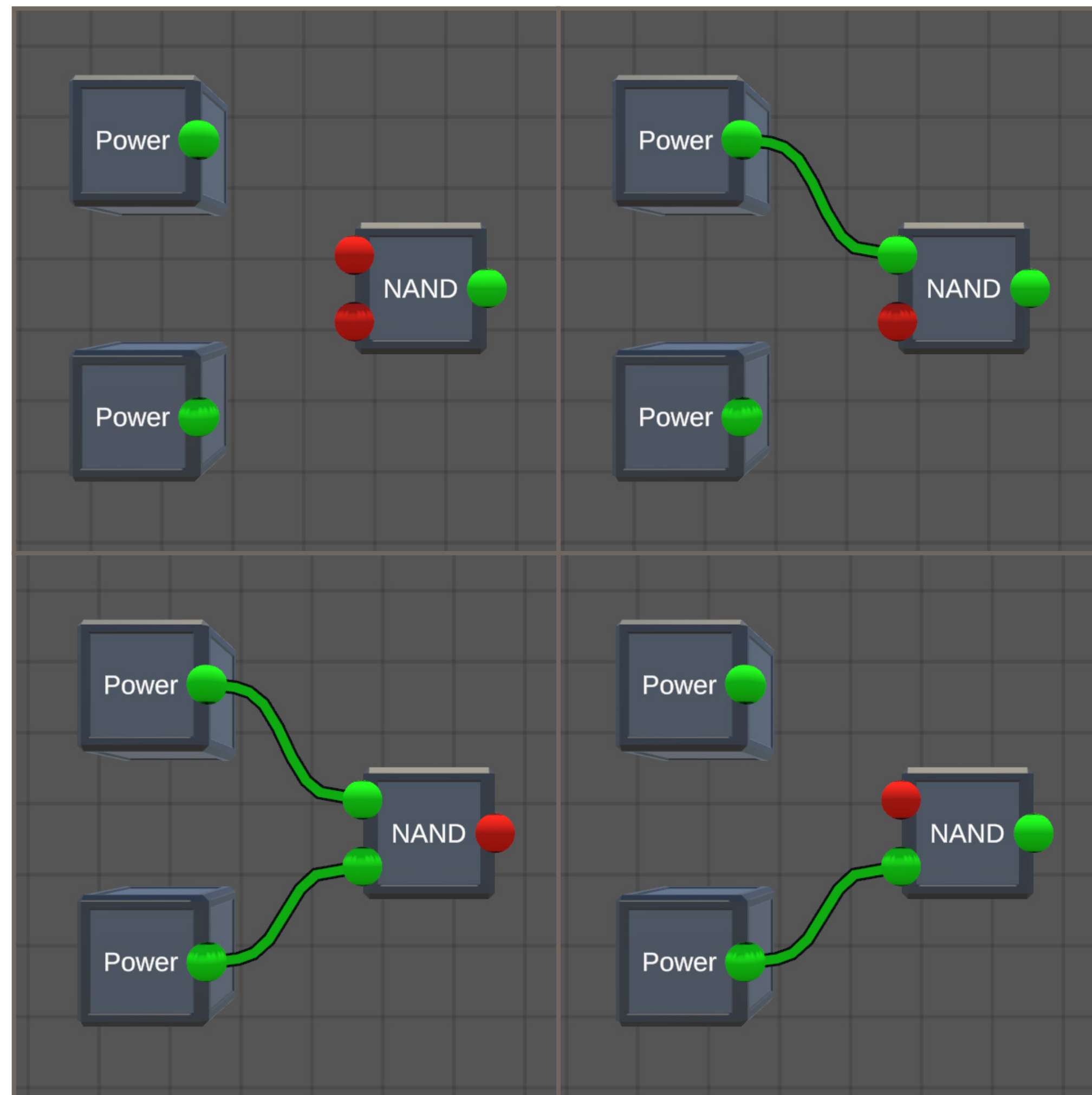


## OR

→ 2 Eingänge

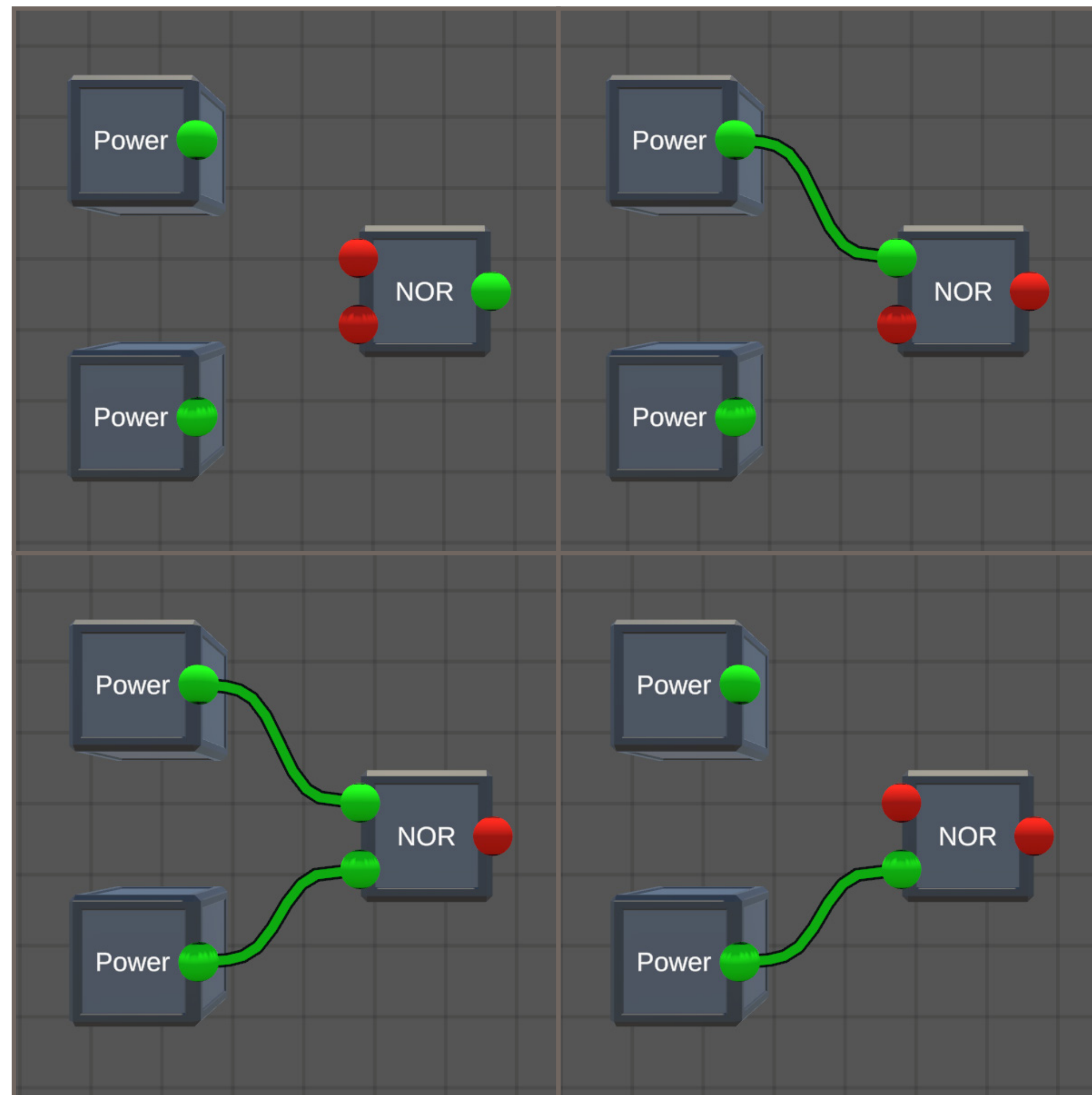
→ 1 Ausgang

→ gibt ein Signal aus, wenn **mindestens einer** der Eingänge ein Signal empfängt



## NAND

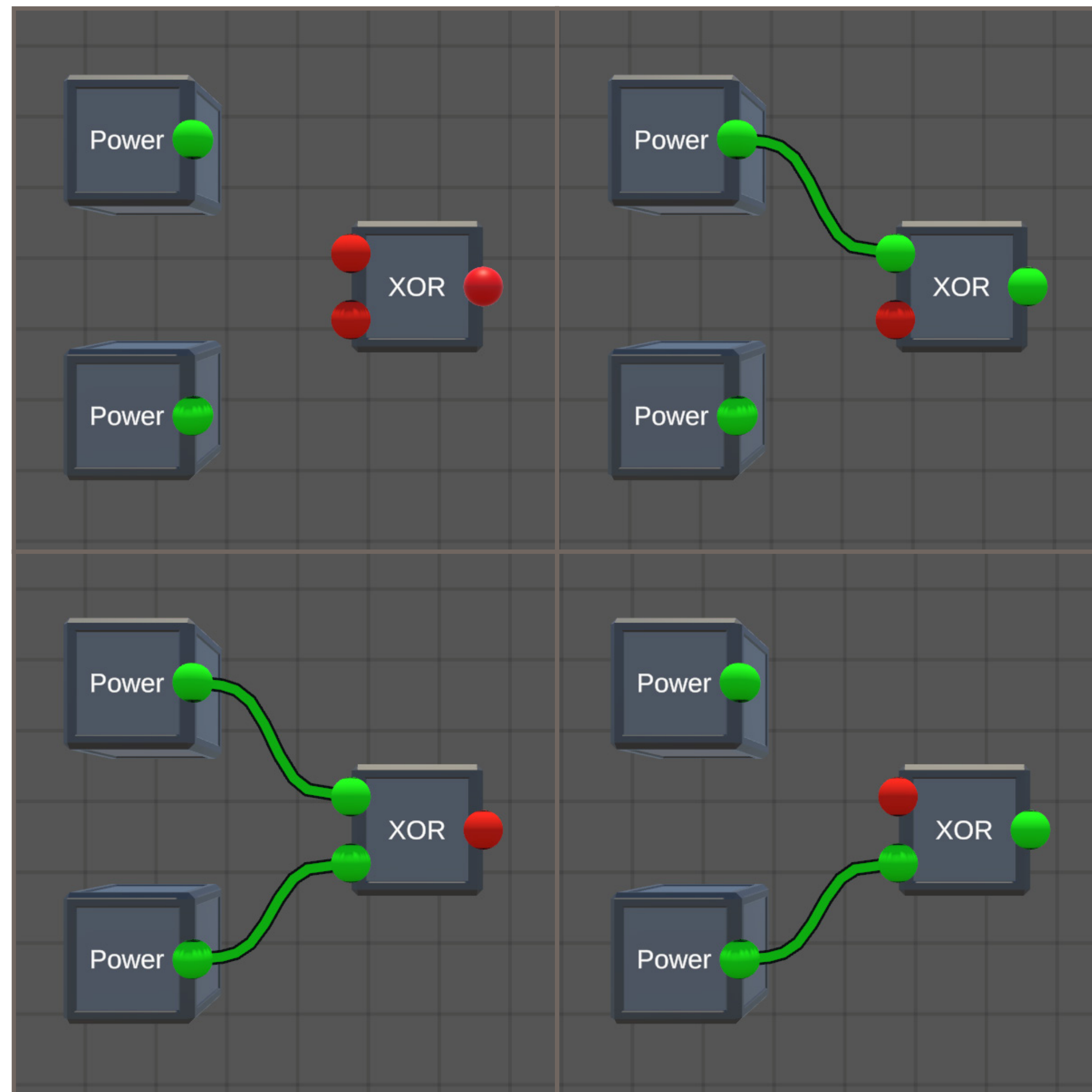
- 2 Eingänge
- 1 Ausgang
  
- gibt ein Signal aus, **solange nicht beide** Eingänge gleichzeitig ein Signal empfangen



## NOR

- 2 Eingänge
- 1 Ausgang

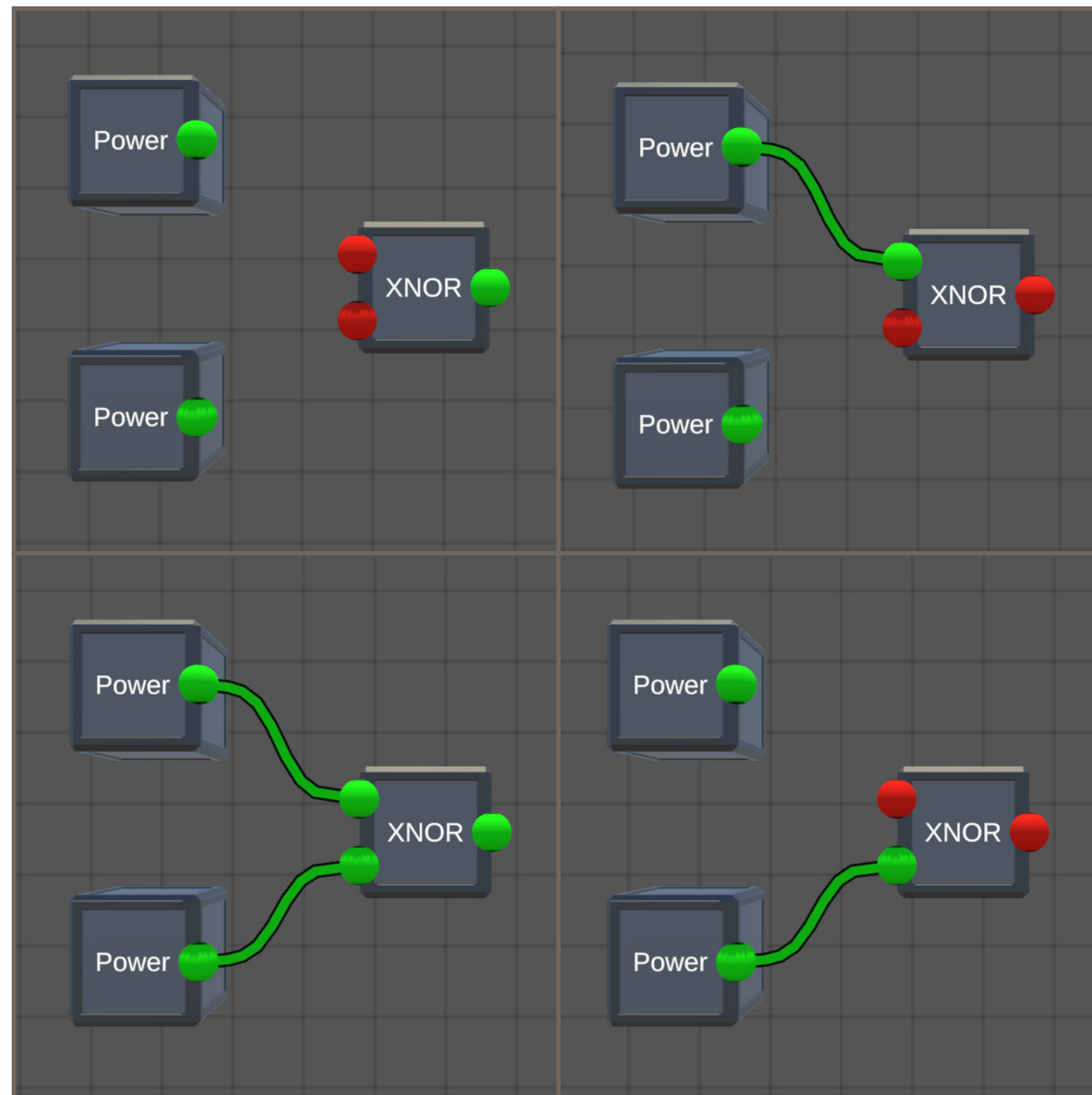
→ gibt ein Signal aus, **solange keiner der beiden** Eingänge ein Signal erhält



## XOR

- 2 Eingänge
- 1 Ausgang

→ gibt ein Signal ab, **solange genau einer** der Eingänge ein Signal erhält

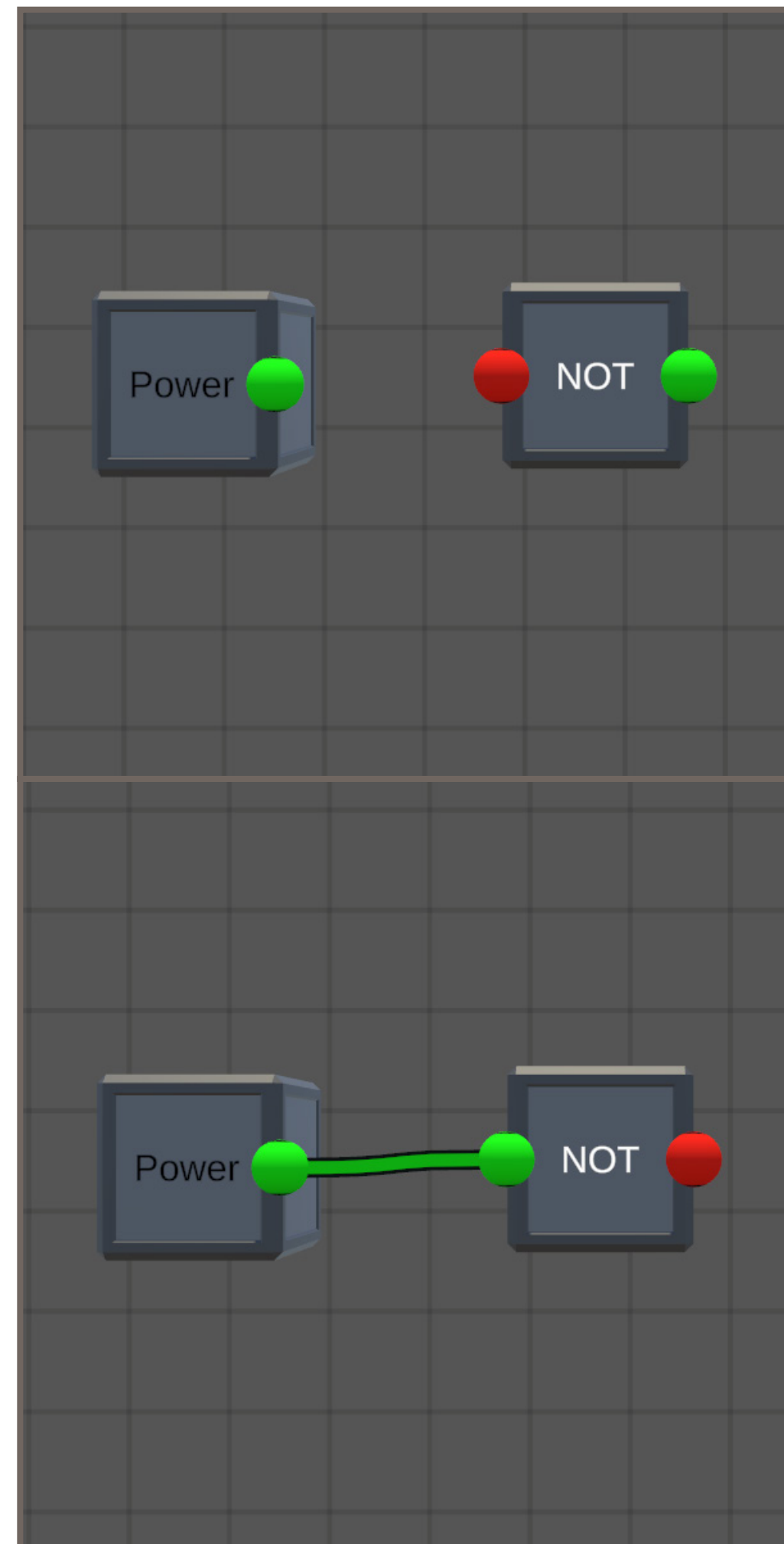


## XNOR

- 2 Eingänge
- 1 Ausgang

→ gibt ein Signal ab, solange **entweder von beiden gleichzeitig** oder **von keinem** der Eingänge ein Signal empfangen wird

**XNOR gibt dann ein Signal aus, wenn beide Eingänge den gleichen Zustand haben**

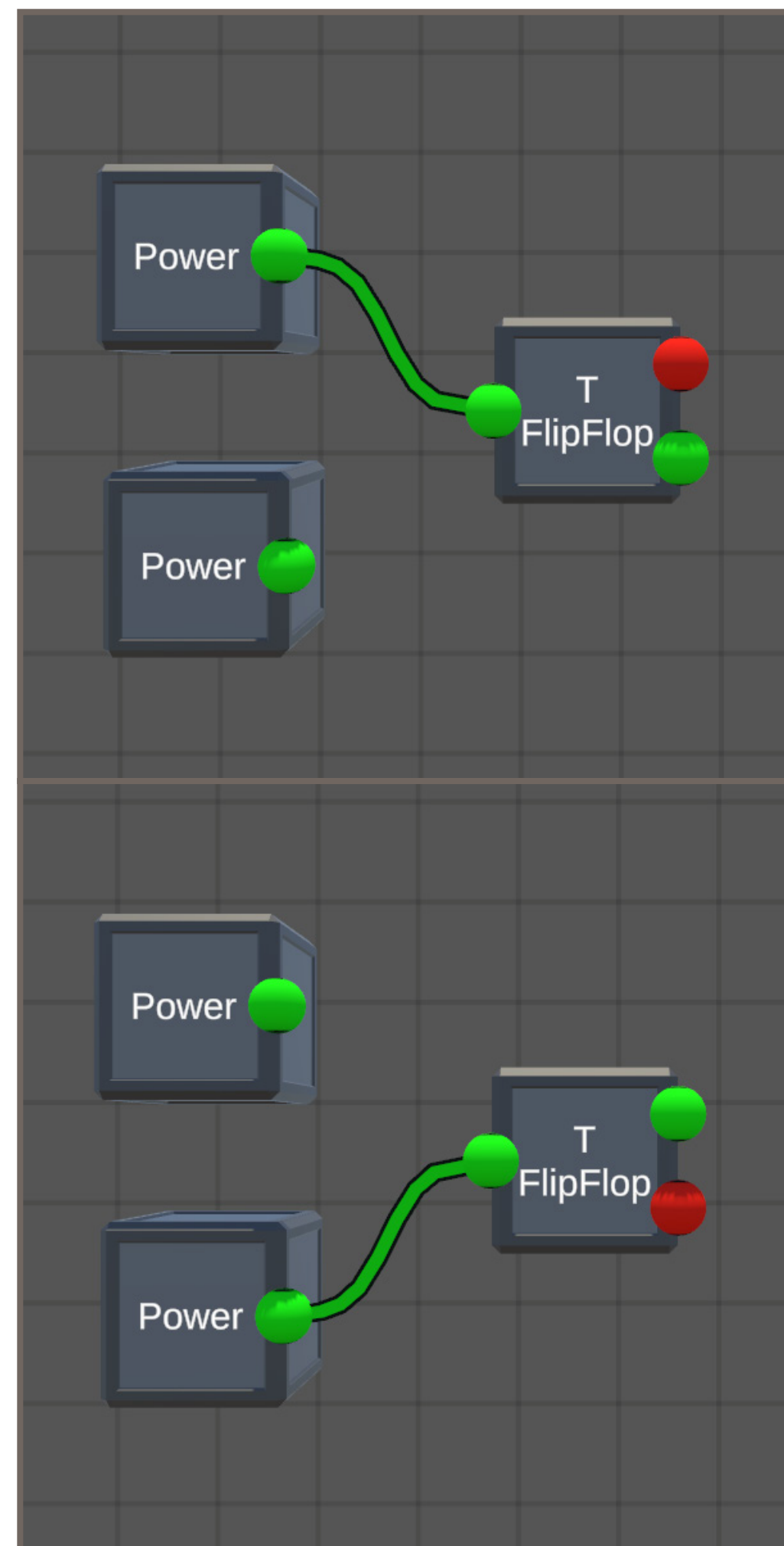


## NOT

- 1 Eingang
- 1 Ausgang

→ gibt ein Signal aus, **solange es kein** Signal empfängt

**NOT kehrt den Zustand um**



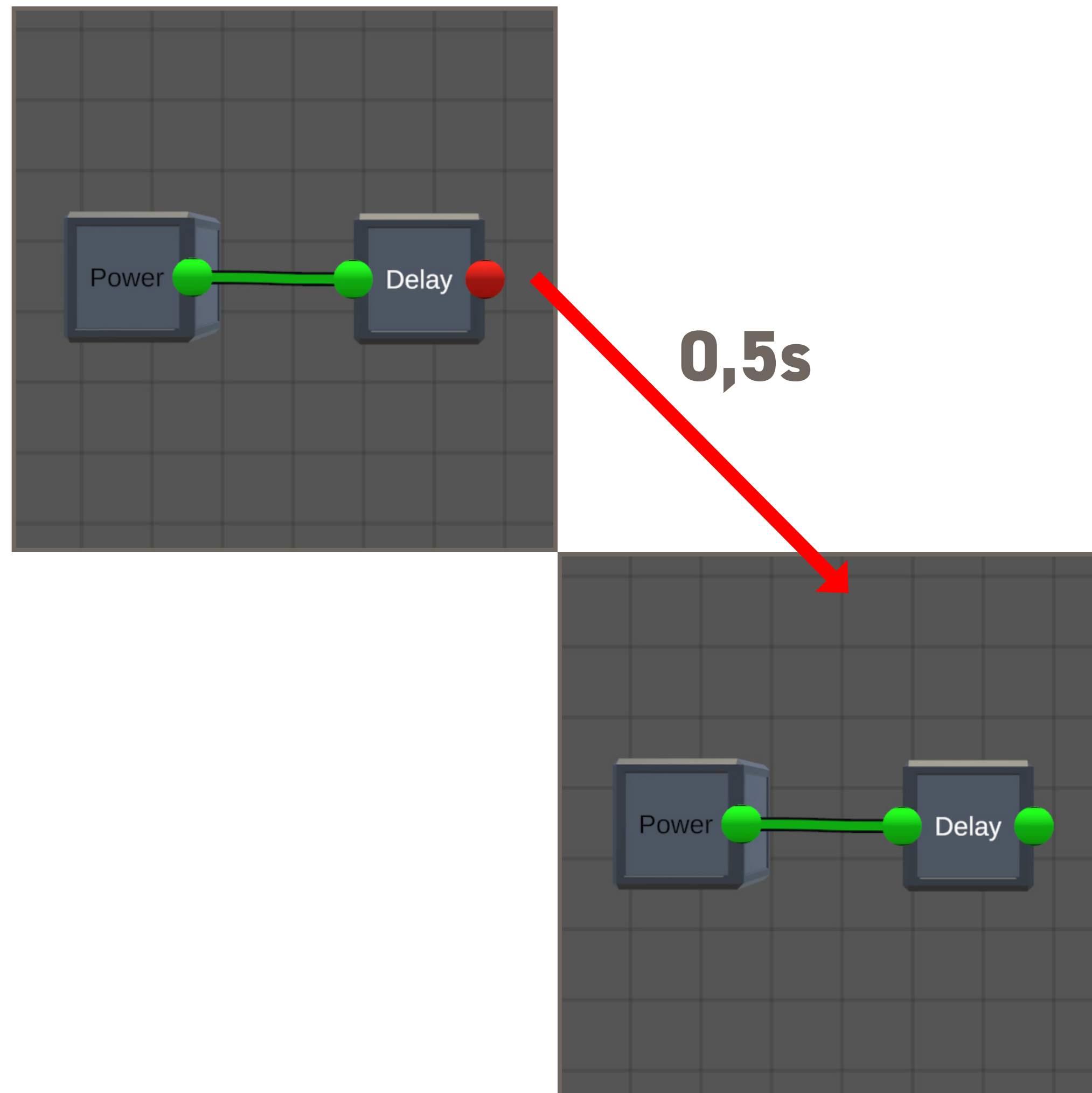
## TFlipFlop

- 1 Eingang
- 2 Ausgänge
- gibt **immer auf einem** der beiden Ausgänge ein Signal ab; **wechselt** den Ausgang, wenn es ein **neues Signal** bzw. ein Signal von einer neuen Quelle erhält



## Power

- 1 Ausgang
- gibt **immer** ein Signal aus  
kann so gut als Ausgangspunkt für Programmierungen oder als simpler Antrieb verwendet werden!

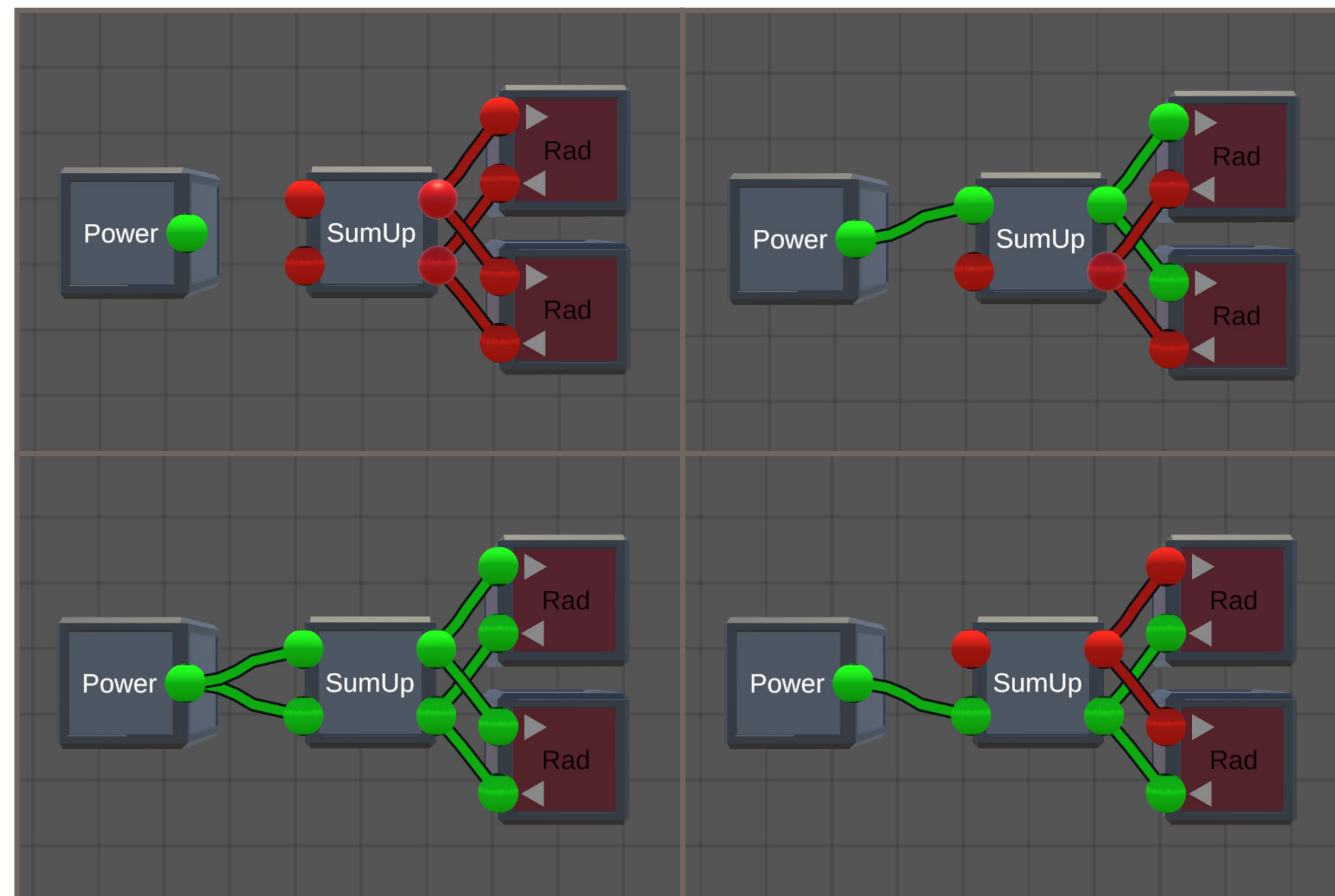


## Delay

- 1 Eingang
- 1 Ausgang

→ gibt Eingangszustand mit einer **halben Sekunde Verzögerung** aus

d. h. auch: wird ein bisher empfangenes Signal nicht mehr empfangen, wird weiterhin für eine halbe Sekunde ein Signal ausgegeben



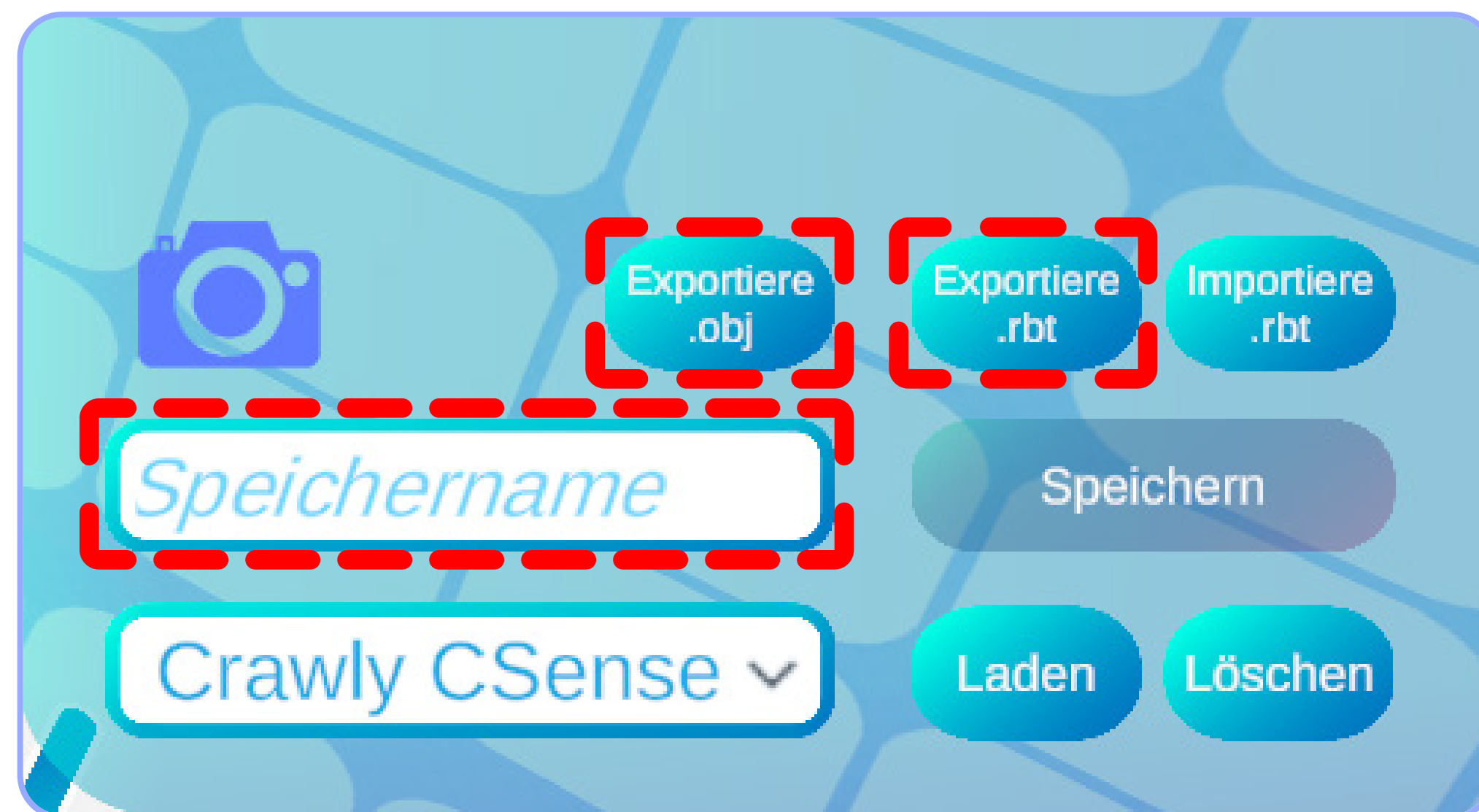
## SumUp

- 2 Eingänge
- 2 Ausgänge
- funktioniert in Kombination mit funktionalen Robotermodulen mit Signaleingängen!

- fasst alle Eingänge von zuvor ausgewählten funktionalen Modulen zusammen und lässt sie über die beiden Eingänge **als Gruppe ansteuern**

d. h. erhält der obere bzw. untere Eingang ein Signal, gibt er auch ein Signal am oberen bzw. unteren Ausgang an alle angeschlossenen Teile ein Signal ab

**SumUp verbindet sich automatisch mit den Eingängen aller vorher angewählten Programmierbausteine**



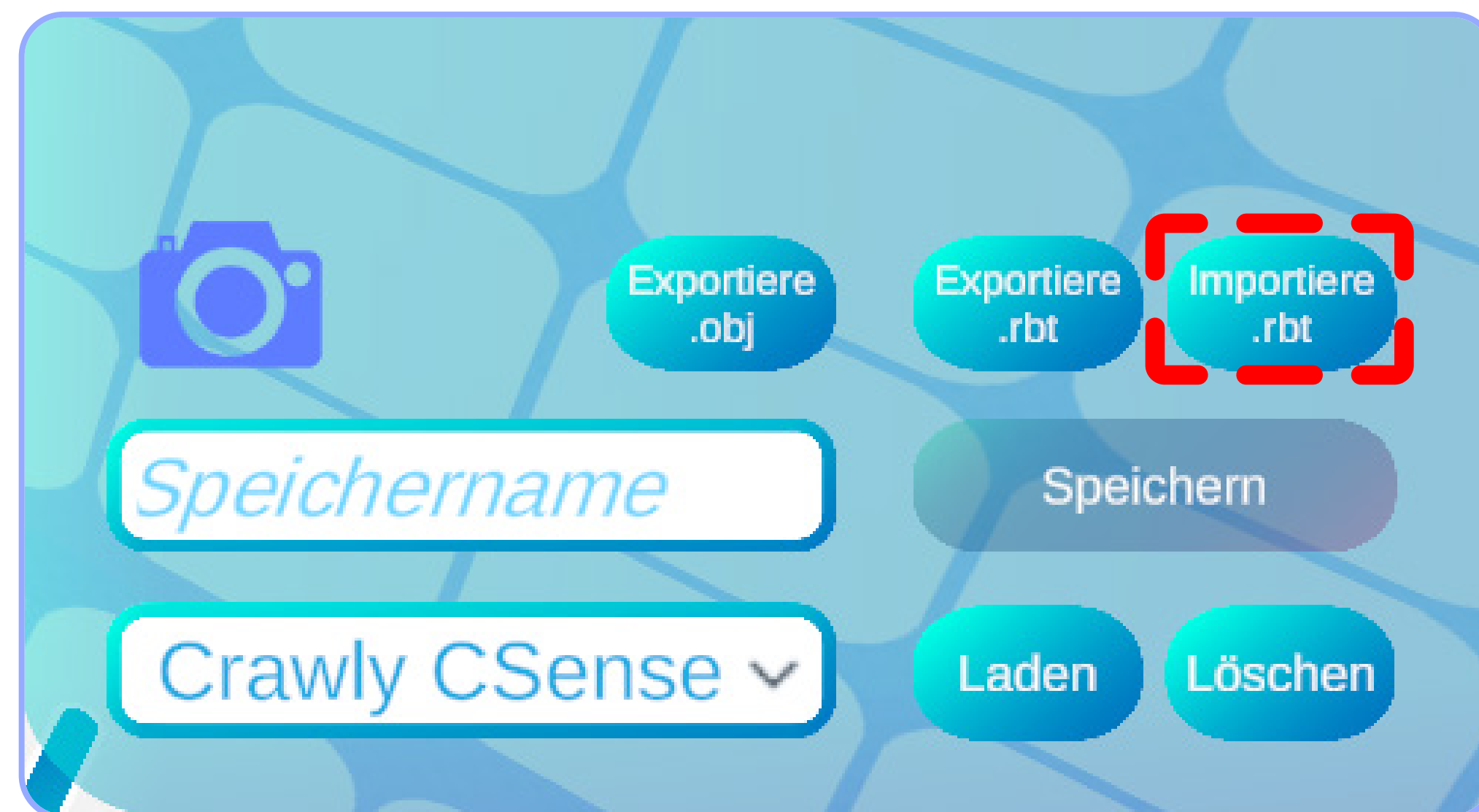
## Export

Willst du deinen Roboter exportieren, gib ihm einfach im dafür vorgesehenen Feld einen Namen, speichere ihn unter diesem und klicke dann entweder auf **Exportiere .rbt** oder **Exportiere .obj**.

- **.rbt** ist das Dateiformat, mit welchem die iRobot Factory **intern** arbeitet – möchtest du also einem anderen User dein Design zukommen lassen, nutze **.rbt**!
- **.obj** ist ein Standardformat zum Export von 3D-Objekten – geht es dir also nur um das **optische Design** deines Roboters, verwende **.obj**!

Tust du dies, öffnet sich ein Explorer-Fenster, in welchem du den Speicherort festlegen kannst. Wähle einen Ordner aus und klicke auf Speichern. Dann kannst du die exportierte Datei ganz nach Belieben versenden oder selbst weiterverwerten.

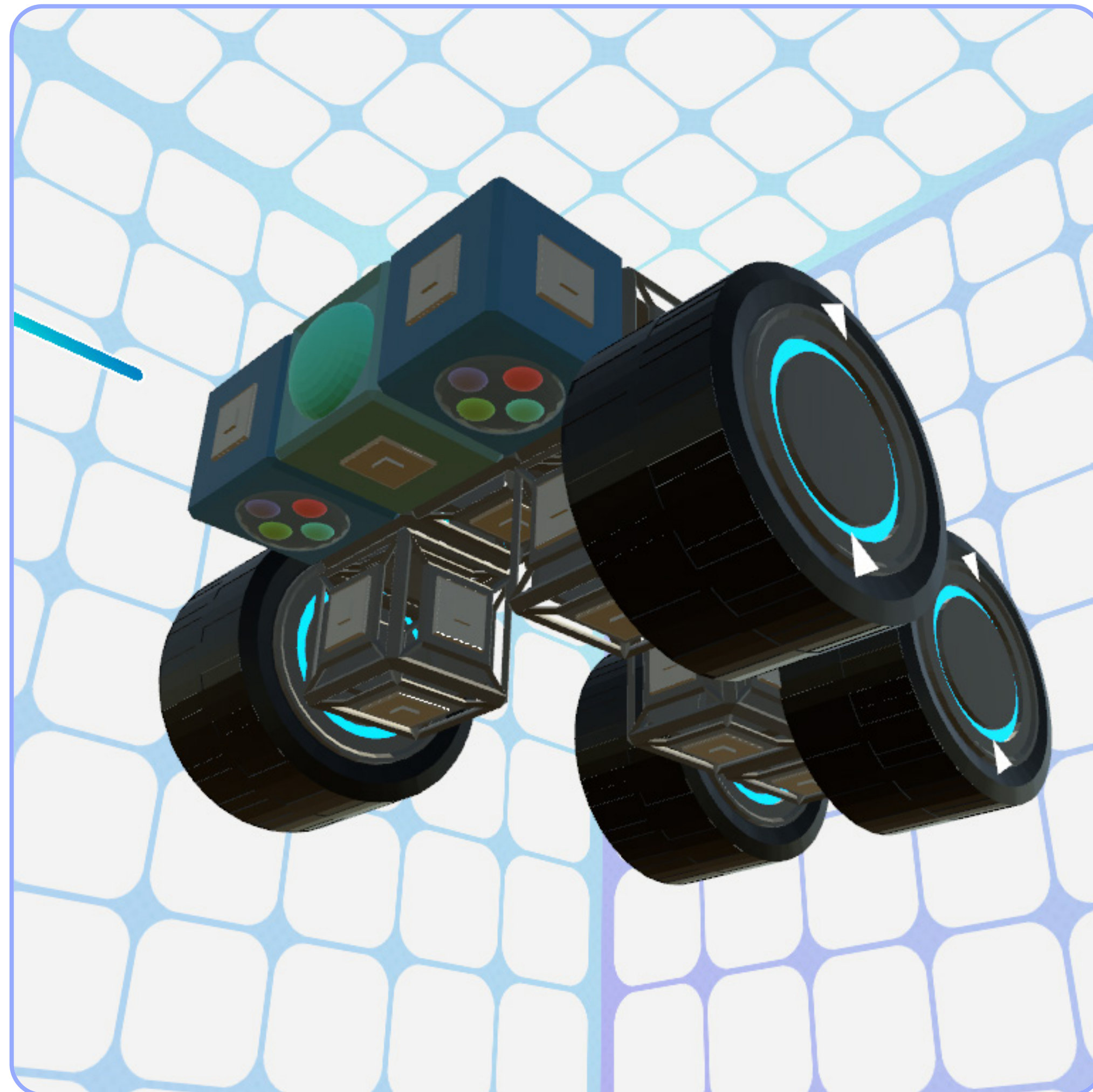
Um ein Design in der iRobot Factory weiterverwenden zu können, benötigst du das Dateiformat **.rbt**!



### Import

Zum Importieren eines fremden Designs in deine iRobot Factory benötigst du den entsprechenden Roboter als **.rbt-Datei**.

Ist diese auf deinem System vorhanden, klicke auf **Importiere .rbt**. Daraufhin öffnet sich ein Explorer-Fenster; wähle in diesem einfach die entsprechende .rbt-Datei aus und klicke auf Öffnen.



### **Lernziel:**

Verständnis der Funktionsweise von Farbsensoren sowie deren praktische Anwendungsmöglichkeiten

Möglichkeit der Nutzung von Farben der Umgebung zur Hindernisbewältigung

### **Voraussetzungen:**

Grundlegende Funktionalität der iRobot Factory ist den Anwendern geläufig; Logikgatter können korrekt verbaut werden.

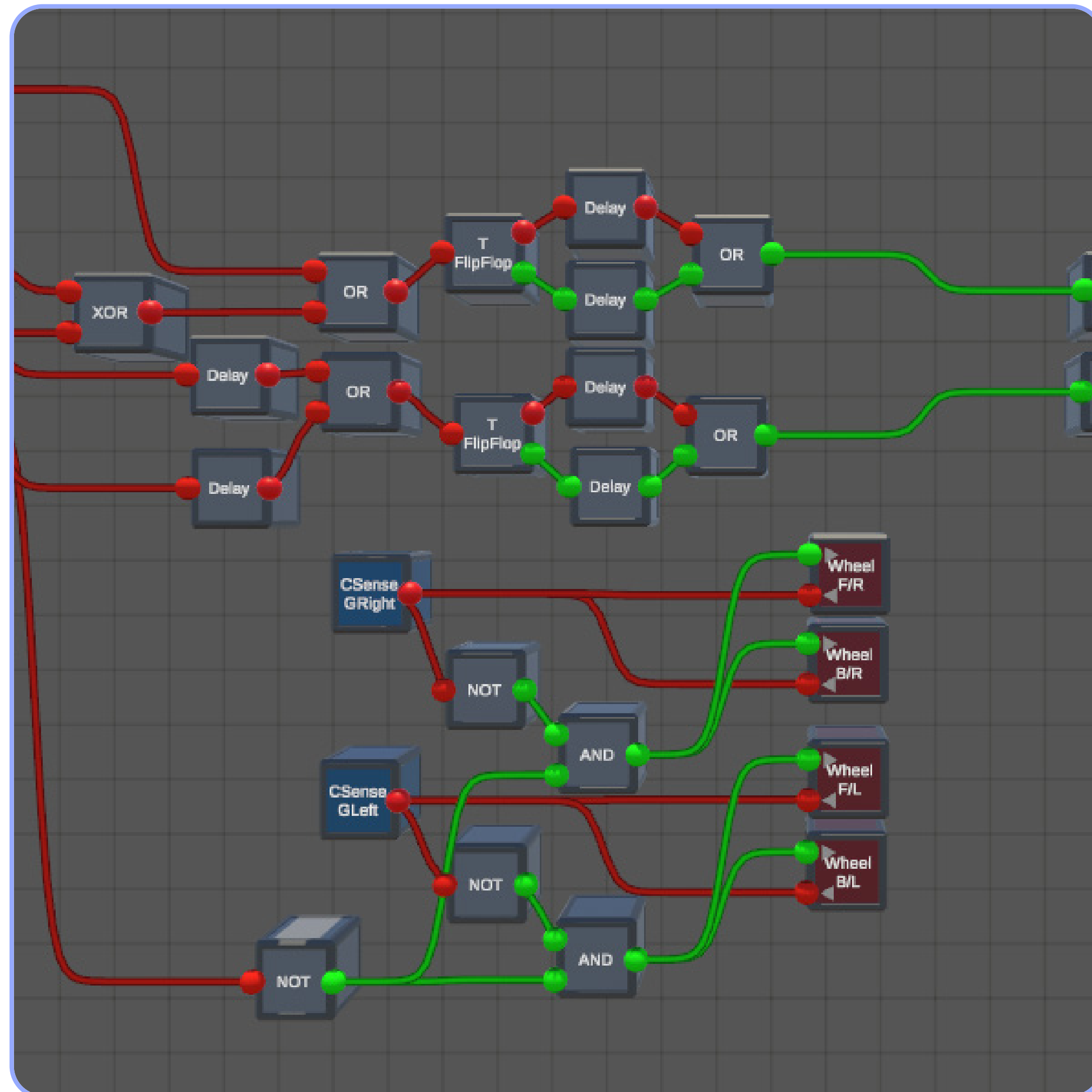
### **Level:**

Transport 2 [Anfänger]

Treffsicherheit 3 (möglichst wenig Bälle) [Anfänger]

Navigation 1 / Navigation 2 [Fortgeschrittene]

Transport 3 [Fortgeschrittene]

**Lernziel:**

Aufschlüsseln komplexer Probleme in kleinere Teilprobleme und eigenständiges Angehen/Lösen dieser.

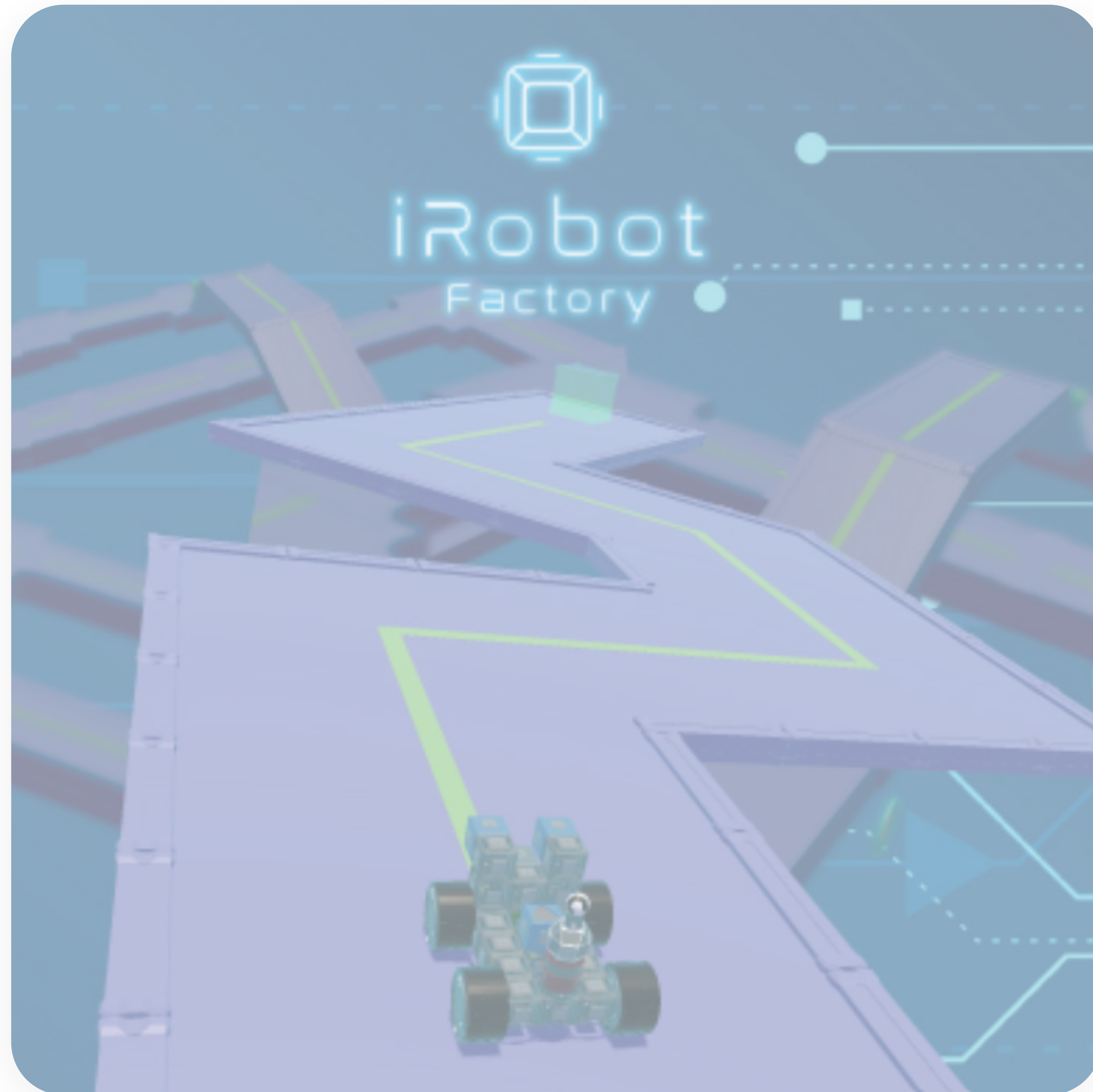
Ggf. Übertragung dieses Ansatzes auf Probleme der Anwender\*innen im echten Leben.

**Voraussetzungen:**

Die Funktionalität der iRobot Factory ist den Anwendern geläufig; Logikgatter & Funktionsteile können – bestenfalls eigenständig – korrekt verbaut werden.

**Level:**

Herausforderung 1 / Herausforderung 2  
[Fortgeschrittene]



Eine Komplettlösung (Walkthrough) für alle Level ist angedacht, aber noch in Bearbeitung.

Herausgegeben von:

**Einfach Genial gGmbH**

Vahrenwalder Str. 269 A

30179 Hannover

info@einfachgenial.org

0511/9666-811

Gefördert durch:

**Walter Nebel Stiftung**

einfachgenial

 Walter **Nebel** Stiftung