

PLEDGE-ALGORITHMUS

Autor: Christophe Sahli
Version: 12. Januar 2013

PLEDGE-ALGORITHMUS

INHALT

| | |
|--------------------------------------|----|
| Einleitung | 2 |
| Voraussetzungen und Zielgruppe | 2 |
| Ablauf und Organisation | 3 |
| Lernziele | 4 |
| Lerninhalte | 5 |
| Papier und Bleistift | 5 |
| Der Pledge Algorithmus | 6 |
| Scratch | 8 |
| Zeichnen des Roboters | 8 |
| Zeichnen des Labyrinths | 9 |
| Programmierung des Roboters | 9 |
| Robotik mit Lego Mindstorm | 12 |
| Das Labyrinth Bauen | 12 |
| Bau Des Roboters | 13 |
| Programmierung des Algorithmus | 15 |
| Literaturverzeichnis | 19 |
| Anhang | 20 |

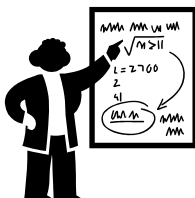
Einleitung

In dieser Unterrichtseinheit steht der Pledge-Algorithmus im Zentrum. Wie in (Berthold Vöcking, 2008) sehr schön beschrieben, geht es bei diesem Algorithmus darum, aus einem beliebigen Labyrinth im Dunkeln den Ausgang zu finden. Ein Labyrinth hat etwas Mystisches, Geheimnisvolles und seit Jahrtausenden beschäftigt es die Menschheit. Ist es überhaupt möglich aus einem Labyrinth mit einer geeigneten Strategie zu entkommen? Welche Bedingungen müssen gegeben sein, damit dies gelingen kann? Können wir einen fahrenden Roboter dazu bringen in einem unbekanntem Labyrinth völlig autonom den Ausgang zu finden?

Anhand des Pledge-Algorithmus, der einer von vielen Lösungsalgorithmen für Irrgärten darstellt, können zentrale Grundfertigkeiten im Bereich des Entwurfs, der Beurteilung und Umsetzung eines Algorithmus geschult werden, die immer wieder in den verschiedensten Bereichen der Informatik vorkommen. Der Themenbereich der Robotik ermöglicht es zudem, mit den Lernenden in einer im Schulalltag eher ungewohnten projektartigen Art und Weise zu arbeiten. Die starke Kopplung mit praktischen, den Ingenieurwissenschaften nahegelegenen Fähigkeiten und Fertigkeiten erlaubt es den Schülerinnen und Schülern Erfahrungen in einem Gebiet zu sammeln, was in den meisten anderen Fächern nur sehr selten möglich ist. Gerade dieser Aspekt motiviert viele Schülerinnen und Schüler besonders. Durch den starken Bezug zum Alltag, in dem computergesteuerte Maschinen dem Menschen ihren Dienst erweisen, ist es einfach mit den Schülern die erarbeiteten Informatiklösungen nach Sinn und möglichen Anwendungen zu hinterfragen. In der Interaktion werden wertvolle fachübergreifende Kompetenzen gefördert. Ein interdisziplinärer Austausch wäre nicht nur mit dem Fachgebiet der Geschichte in Zusammenhang mit dem Mysterium Labyrinth möglich. Auch die Naturwissenschaften (v.a. Physik, Mathematik und Biologie) und Bewegungswissenschaften haben mit der Robotik Gemeinsamkeiten, die durchaus angesprochen und bei Bedarf vertieft werden könnten.

Voraussetzungen und Zielgruppe

Diese Unterrichtseinheit richtet sich an Schülerinnen und Schüler der Mittel und Oberstufe. Es sind grundsätzlich keine Vorkenntnisse in der Programmierung notwendig.



¹Es ist jedoch sehr sinnvoll die Schülerinnen und Schüler, die noch nie programmiert haben, strukturiert und vollumfänglich in die Thematik der Programmierung einzuführen. Hier werden wir neben der einfachen, visuellen Programmiersprache Scratch noch die von Lego eigens entwickelte Programmieroberfläche benutzen. Eine Programmierung der Legoroboter mit Java² wäre durchaus denkbar, wird hier jedoch nicht weiter verfolgt. Pro Team werden zudem ein Computer mit installierter Scratch- und Lego Mindstorm NXT Software und ein Lego-Robotik-Bausatz benötigt. Neben Papier und Bleistift können Hartschaumplatten aus dem Baumarkt für den Bau eines Labyrinthes benutzt werden.

¹ Spezielle Anmerkungen für die Lehrperson werden jeweils mit diesem Piktogramm gekennzeichnet und enthalten didaktisch wichtige Hinweise für die Durchführung der jeweiligen Unterrichtseinheit.

² http://www.legorobotik.ch/index.php?inhalt_mitte=home/home.inc.php (12.01.2013)

Ablauf und Organisation

Die einzelnen Lektionen gliedern sich in thematisch zusammengehörende Teilbereiche, die in der Tabelle 1 – Ablauf und Organisation farblich hervorgehoben sind. Der zeitliche Aufwand richtet sich nach den Vorkenntnissen und der Grösse der Lerngruppe.

Der erste Teil „Papier und Bleistift“ ist von der Gruppengrösse und dem Vorwissen der Lernenden unabhängig. Zu Beginn werden die Lernenden von der Lehrperson in die Thematik eingeführt. Sie bearbeiten Arbeitsblätter und lösen Einzel- und Gruppenaufgaben. Falls bereits mit der Programmierumgebung Scratch gearbeitet wurde, verringert sich der Aufwand der zweiten Einheit „Scratch“ um eine Einführungslektion. Dasselbe gilt für die dritte Einheit Robotik. Je nachdem wie viele Lego-Robotik-Bausätze zur Verfügung stehen, werden kleinere oder grössere Teams gebildet. In dieser Einheit ist es gut möglich, dass den Schülerinnen und Schülern mehr Zeit eingeräumt wird, den Roboter im Bau zu optimieren und in der Programmierung zu verbessern. Wichtig dabei ist jedoch, dass die Gruppen stets zielorientiert und mit klaren Zeitvorgaben arbeiten. Die Präsentationen der Ergebnisse können je nach Klassengrösse in einer allenfalls in zwei Lektionen stattfinden.






| Titel der Unterrichtseinheit | Methode | Lektionen |
|--|---|-----------|
| Papier und Bleistift |  Einzel  Team | 1 |
| Scratch |  Team | 1-2 |
| Robotik Bau und Programmierung des Roboters |  Team | 6-8 |
| Präsentation der Ergebnisse |  Plenum | 1-2 |
| Geschätzter Aufwand Total | | 9-13 |

TABELLE 1 - ABLAUF UND ORGANISATION

Lernziele

Die Schülerinnen und Schüler können...

| Einheit | Ziele der jeweiligen Unterrichtseinheit |
|----------------------|---|
| Papier und Bleistift | <ul style="list-style-type: none">• eine klare Handlungsanweisung so deutlich formulieren, dass der Partner diese versteht und ohne nachzufragen umsetzen kann.• den Begriff Algorithmus erklären.• Beschreiben, mit welcher Strategie (Algorithmus) im Dunkeln aus einem Labyrinth herausgefunden werden kann.• einen Algorithmus testen und unter neuen Gegebenheiten verbessern.• den Algorithmus in einer geeigneten Form skizzieren.• Verstehen, weshalb der Pledge-Algorithmus für jedes <i>beliebige</i> Labyrinth funktioniert. |
| Scratch | <ul style="list-style-type: none">• einfache Programmierstrukturen (Schleifen) verstehen und anwenden.• eine Variable in der Programmierung korrekt anwenden.• in der Programmiersprache Scratch eine Lösung der Problemstellung programmieren und damit das Verständnis des im ersten Teil kennengelernten Algorithmus festigen.• im Team eine Rolle übernehmen und konstruktiv zur gemeinsamen Lösung eines Problems beitragen.• auf ein neues Problem angemessen reagieren und nach einer geeigneten Lösung suchen. |
| Robotik | <ul style="list-style-type: none">• einen funktionstüchtigen, einfachen Lego-Roboter bauen.• Sensoren und Aktoren gezielt einsetzen.• das Wissen aus den vorangegangenen Lerneinheiten in einer neuen Programmierumgebung anwenden.• anhand einer klaren Problemstellung einen eigenen Lösungsweg skizzieren und diesen danach in der Praxis, gemäss den bestehenden Vorgaben umsetzen.• besser im Team zusammenarbeiten.• besser kreativ mit den vorhandenen Mitteln nach geeigneten Lösungen suchen.• eher einschätzen welche praktischen Schwierigkeiten beim Bau eines Roboters auftreten können und wie man diese Probleme lösen kann. |
| Präsentation | <ul style="list-style-type: none">• ihre Ergebnisse der Gruppe präsentieren.• die Programmierung des Roboters erklären.• aufgetretene Probleme und deren Lösungen schildern. |

TABELLE 2 - UNTERRICHTSZIELE

LERNINHALTE

Papier und Bleistift

Nach einer kurzen Einführung in die Thematik der Unterrichtseinheit beginnt der erste Teil. Die Schülerinnen und Schüler sollen verschiedene Strategien/Algorithmen mit Stift und Papier ausprobieren und so eine eigene Lösung finden um aus einem Labyrinth zu entkommen. Dazu sind nach einander die Arbeitsblätter 1 bis 4 (siehe Anhang) zu verteilen, die didaktisch aufeinander aufbauen.

Arbeitsblatt 1

Mögliche Lösung: Intuitiv wird wohl die erste Idee im Dunkeln sein, mit ausgestreckten Armen eine Wand zu suchen und dieser Wand zu folgen, bis man (hoffentlich) den Ausgang findet.

Arbeitsblatt 2

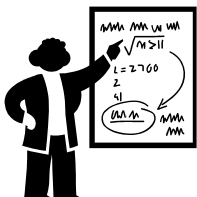
Mögliche Lösung: Folge der Wand nur so lange, bis du wieder in die alte Richtung läufst. Danach läufst du so lange geradeaus, bis du wieder auf eine Wand triffst.

Arbeitsblatt 3

Mögliche Lösung: Hier kommt der eigentliche Pledge-Algorithmus zum Zug. Damit der Ausgang gefunden wird, müssen die Drehungen gezählt werden. Der Roboter beginnt mit der Fahrt geradeaus und setzt den Drehungszähler auf 0. Beim ersten Hindernis dreht er um 90 Grad nach rechts und erhöht den Drehungszähler um +1. Jetzt folgt er dem Hindernis und passt bei jeder Richtungsänderung seinen Drehungszähler entsprechend an (Rechtsdrehung +1 und Linksdrehung -1). Sobald der Drehungszähler auf 0 steht, fährt der Roboter geradeaus, bis er auf das nächste Hindernis trifft.

Arbeitsblatt 4

Mit dem letzten Arbeitsblatt soll der Algorithmus weiter gefestigt werden.



In der Partnerarbeit werden die verschiedenen Sichtweisen (die des Roboters und die des Steuerungsprogramms) bewusstgemacht. In der Diskussion miteinander werden bestehende Lösungen getestet und neue gefunden. Die Lehrperson soll sich bewusst mit helfen zurückhalten, damit die Eigenleistung der Teams gefördert wird.

DER PLEDGE ALGORITHMUS

Anhand des folgenden Algorithmus, den angeblich ein damals zwölfjähriger Junge namens John Pledge gefunden haben soll, möchten wir zeigen, dass damit in jedem denkbaren ebenen Labyrinth in endlicher Zeit ein Ausweg aus dem Dunkeln gefunden werden kann, sofern es einen erreichbaren Ausgang gibt.

```
1  Setze Umdrehungszähler auf 0;  
2  repeat  
3    repeat  
4      Gehe geradeaus;  
5    until Wand erreicht;  
6    Drehe nach rechts;  
7    repeat  
8      Folge dem Hindernis;  
9    until Umdrehungszähler = 0;  
10 until ins Helle gelangt;
```

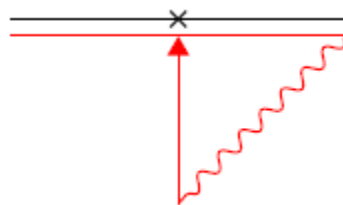
(Berthold Vöcking, 2008) S.77

Bei jeder Rechtsdrehung müssen wir den Umdrehungszähler um 1 erhöhen und bei einer Linksdrehung entsprechend um 1 erniedrigen (oder umgekehrt). Nur wenn der Umdrehungszähler auf 0 steht, kann und soll sich der Roboter vom Hindernis lösen. Genau so findet er in jedem Fall den Ausgang aus einem Labyrinth, sofern ein solcher existiert. Wie können wir das beweisen?

Das folgende Vorgehen für einen Beweis ist dem elektronischen Dokument im Anhang (Dr. Hanjo Täubing, 2012) entnommen.

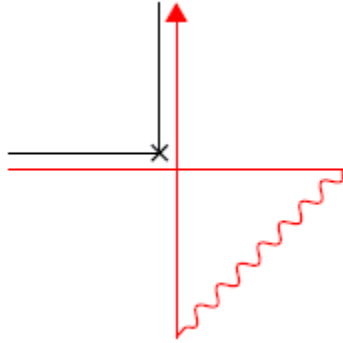
1. Zeige, dass es keine Kreuzung auf dem Weg durch das Labyrinth geben kann!
 - a. Wir führen einen Widerspruchsbeweis und nehmen dazu an: Es gibt eine Kreuzung.

i. Fall 1



Eine Kreuzung kann sich nicht mitten auf einer Wand befinden, da der Roboter nicht durch die Wand hindurchfahren kann. Er ist blockiert und muss abdrehen.

ii. Fall 2



Bei einem Kreuzungspunkt stehen folglich zwei Wände in Kontakt. Da unser Zähler auf 0 steht, wenn wir keine Wand berühren und uns von einem Hindernis lösen, ist dieser Fall ebenfalls ausgeschlossen, weil der Zähler nicht für beide Richtungen (waagrecht und senkrecht) 0 sein kann.

2. Zeige, dass sich der Roboter in einer Endlosschleife befindet!

- a. da es keine Kreuzungen geben und der Roboter nur in einer Richtung einer Wand entlang fahren kann, muss der Roboter demnach zweimal bei einer richtungsändernden Stelle durchkommen.

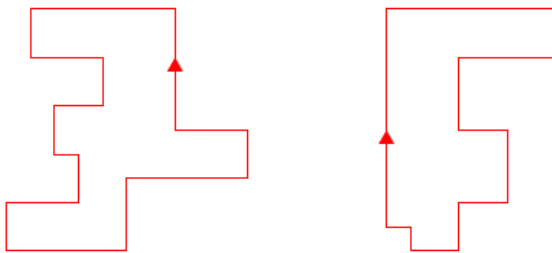
i. Fall 1

Der Roboter kommt an derselben Stelle, mit demselben Zählerstand in derselben Richtung durch. Folglich befindet er sich in einer Endlosschleife.

ii. Fall 2

Der Roboter kommt nie an derselben Stelle mit dem gleichen Zählerstand durch. Da der Zählerstand irgendwann 0 sein muss, damit sich der Roboter vom Hindernis lösen kann, kann er sich auf seinem weiteren Weg nie mehr vom Hindernis lösen. Folglich umkreist er das Hindernis und ist erneut in einer Endlosschleife gefangen.

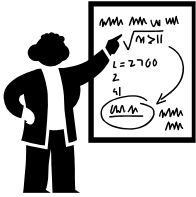
- b. Es gibt nur genau zwei mögliche Endlosschleifen



Da der Zähler sich bei jeder Umdrehung um 1 erhöht oder erniedrigt, nimmt er in einem Fall immer um 4 ab. Das ist nicht möglich! Im anderen Fall nimmt er immer um 4 zu. Das wiederum bedeutet, dass es sich um die äussere Begrenzung des Labyrinths handeln muss.

Scratch

Die Programmierumgebung Scratch³, die von der Lifelong Kindergarten Group am MIT Media Lab unterhalten wird, eignet sich bestens zur Simulation eines Roboters der in einem Labyrinth den Weg nach draussen finden muss.



Es ist empfehlenswert sich in einer Lektion mit der Programmieroberfläche vertraut zu machen um die wichtigsten Funktionen und Bausteine von Scratch kennenzulernen. Dazu eignet sich u.a. der sehr empfehlenswerte Lernparcours von Severin Küpfer (PHBern) auf der Webseite⁴ SwissEduc.ch. Auf eine eigentliche Einführung in Scratch wird hier verzichtet.

Für die Simulation eines Roboters in einem Labyrinth eignet sich folgendes Vorgehen:

ZEICHNEN DES ROBOTERS

Für das Zeichnen des Roboters in Scratch habe ich ein eigenes Objekt erstellt und im grösstmöglichen Zoombereich den Roboter 3x3 Pixel breit gezeichnet. An der Stirnseite rechts (Richtung 90 Grad) hat der Roboter ein gelbes Pixel. Dies simuliert einen ersten Berührungssensor. An der linken Seite des Roboters befindet sich ein blaues Pixel, das den zweiten Berührungssensor simuliert.

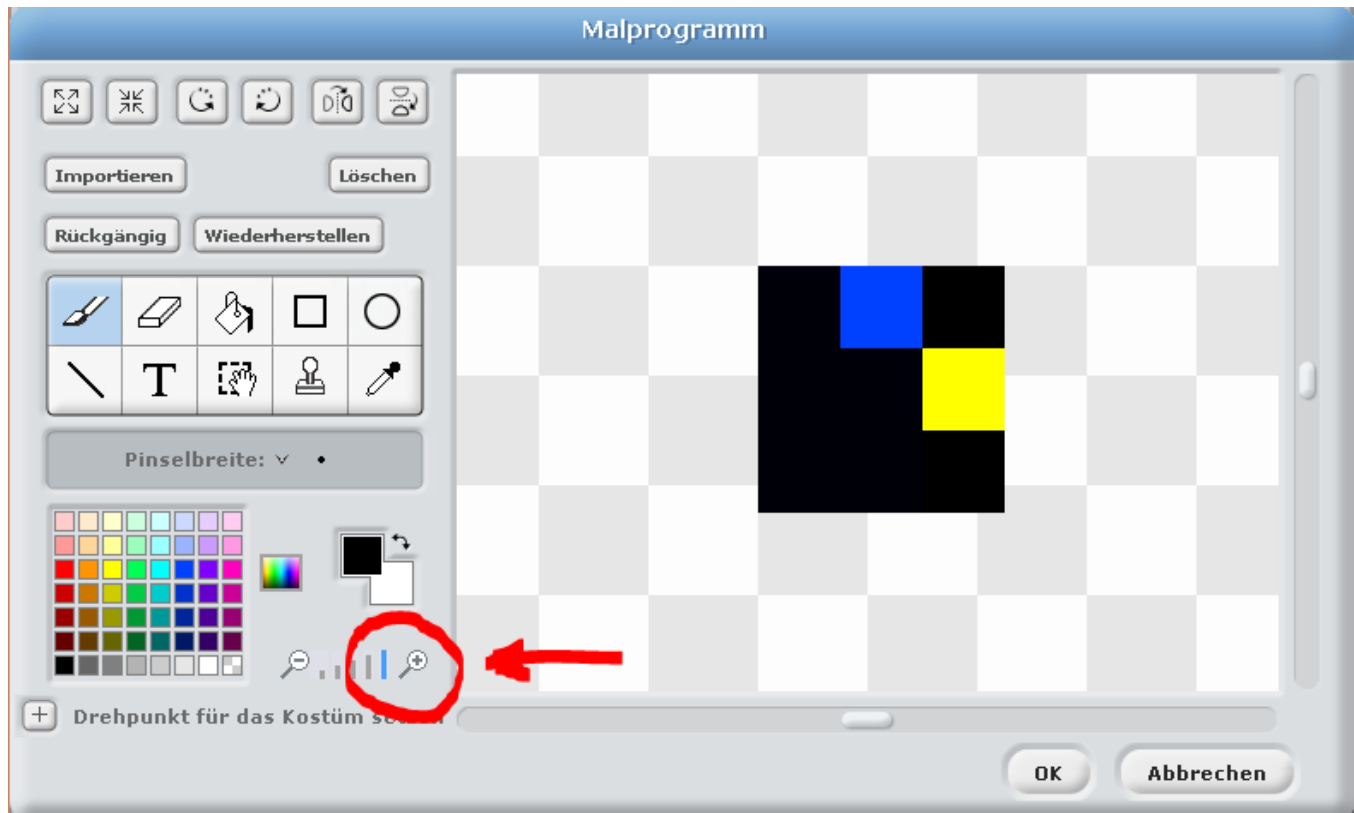


ABBILDUNG 1 - MALPROGRAMM IN SCRATCH

³ <http://scratch.mit.edu/> (12.01.2013)

⁴ http://www.swisseduc.ch/informatik/programmiersprachen/scratch_werkstatt/ (12.01.2013)

ZEICHNEN DES LABYRINTHS

Das Labyrinth muss in Scratch in Form eines Hintergrundes für die sogenannte Bühne gelöst werden. Hier kann schnell und einfach im Malprogramm auf einer weissen Fläche ein beliebiges Labyrinth gezeichnet werden. Besonders schön ist die Möglichkeit, das Labyrinth stetig ausbauen und erweitern zu können. So kann man einfach beginnen und die Reaktion des Roboters testen.

PROGRAMMIERUNG DES ROBOTERS

Die Umsetzung des Pledge-Algorithmus in Scratch ist mit den folgenden Bausteinen möglich:

Erster Teil

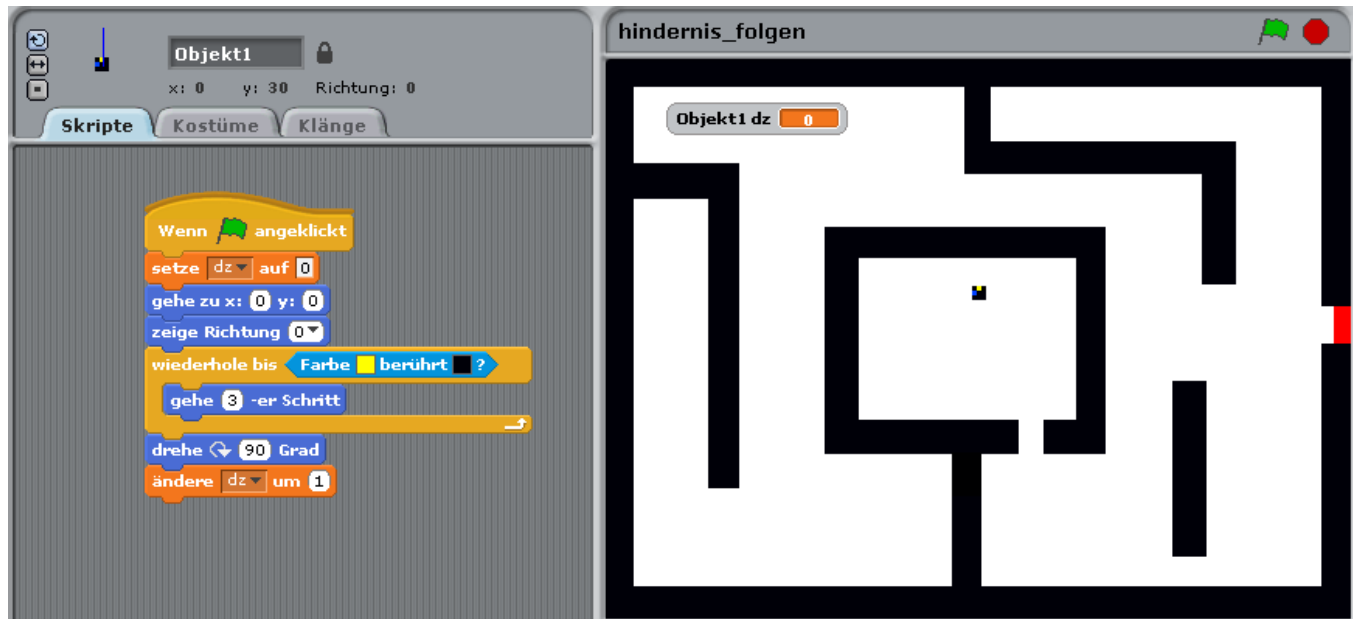


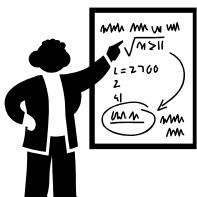
ABBILDUNG 2 - BÜHNE UND PROGRAMMTEIL

Erklärung

Der Programmcode ist durch die farblich voneinander abgehobenen Bausteine gut lesbar und beinahe selbsterklärend. Durch die erste Schleife mit Abbruchbedingung bewegt sich der Roboter solange vorwärts, bis der gelbe Farbpixel des Objekts mit der schwarzen Wand in Berührung kommt. Erst danach verlässt er die Schleife und dreht sich um 90 Grad (hier im Uhrzeigersinn) und zählt die Variable dz um 1 hoch.

Zweiter Teil

Der zweite Teil wird etwas komplizierter. Es stellt sich folgendes Problem: Wie kann der Roboter einem Hindernis folgen?



Im Unterricht kann der Lehrer je nach Einschätzung der Fähigkeiten der Lerngruppe die im Team arbeitenden Schülerinnen und Schüler ruhig eine eigene Lösung suchen lassen. Gute Lösungen können vom Team allen anderen präsentiert werden.

Eine mögliche Lösung könnte so aussehen wie in der Abbildung 3 - Fertiges Programm. Die einzelnen Programmteile sind kommentiert und entsprechen dem vorher besprochenen Pledge-Algorithmus.

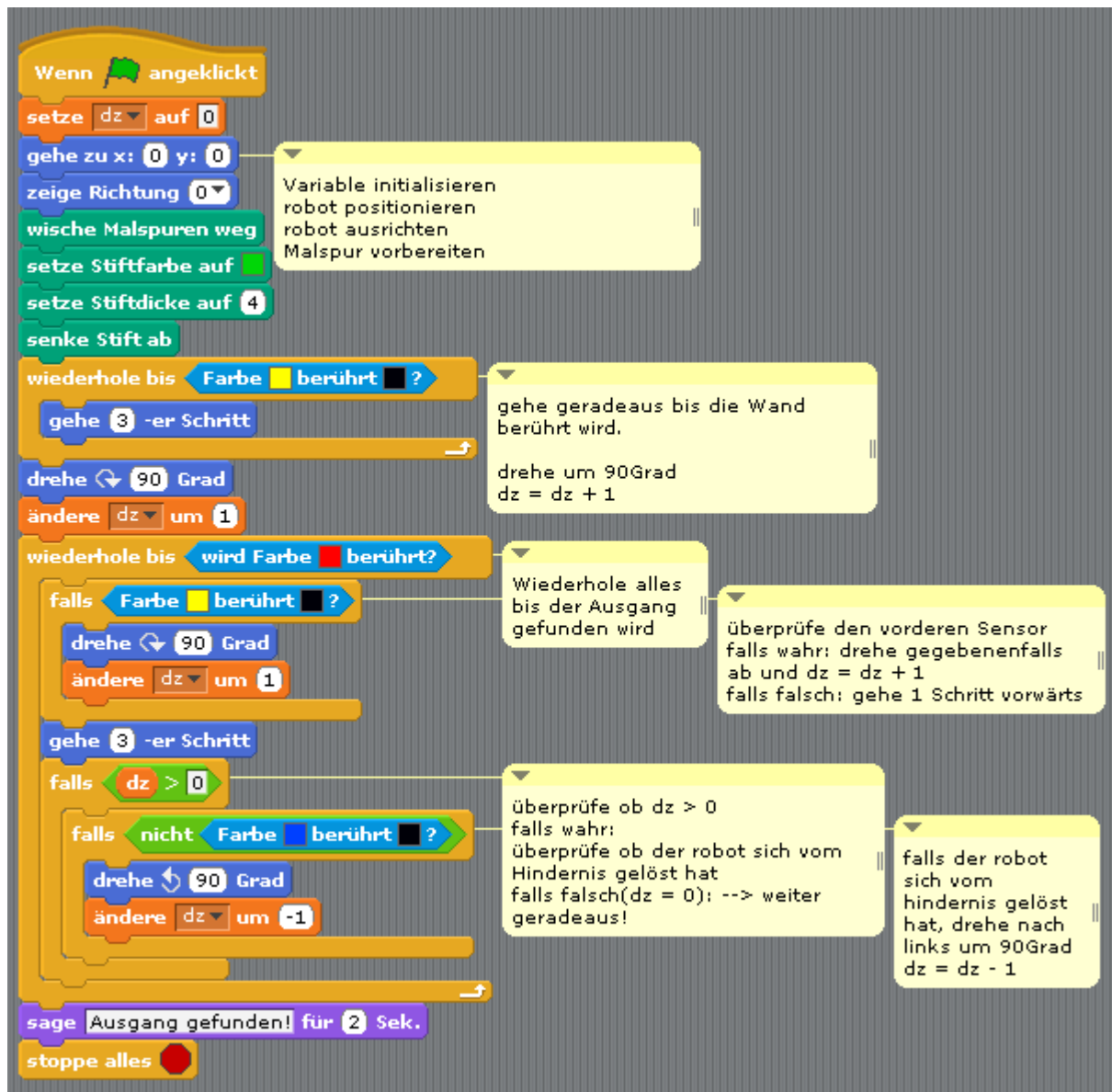
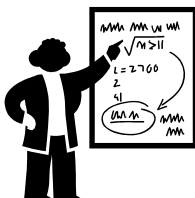


ABBILDUNG 3 - FERTIGES PROGRAMM

Ich habe das Programm⁵ auf der Scratch Homepage in meinem Projektbereich veröffentlicht.



Die Schrittlänge 3 ergibt sich aus der Grösse des Objektes (3x3 Pixel). Auf der Bühne muss beachtet werden, dass das Objekt sich in Tat und Wahrheit in die Wand hinein bewegen kann.

⁵ <http://scratch.mit.edu/projects/csahli/3010703> (12.01.2013)

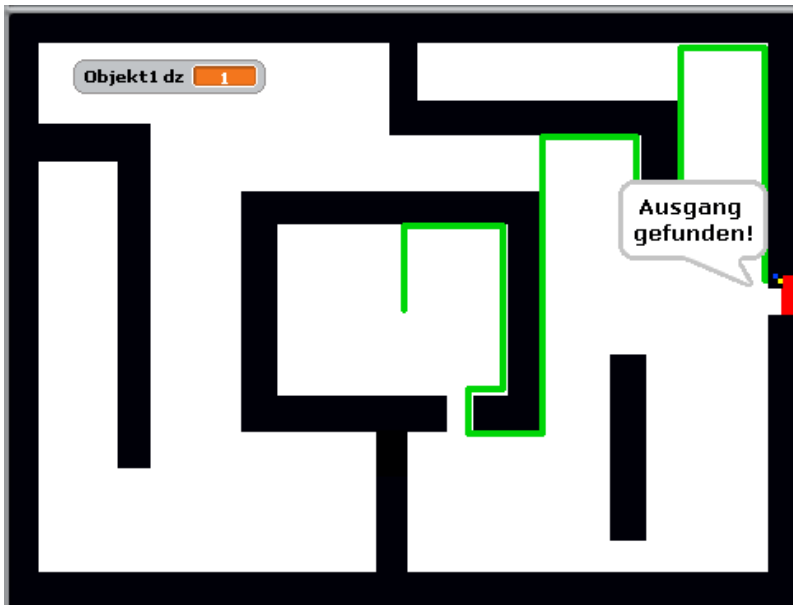


ABBILDUNG 4 - PRINTSCREEN BÜHNE

In Abbildung 4 - Printscreen Bühne ist der Weg des Roboters ersichtlich.

Robotik mit Lego Mindstorm

Im eigentlichen Hauptteil dieser Unterrichtseinheit kommen wir zur Hauptaufgabe:

Ein Lego-Mindstorm-Roboter soll so gebaut und programmiert werden, dass er autonom aus einem unbekanntem Labyrinth herausfindet. Die Platzierung und Ausrichtung des Roboters erfolgt zufällig an einer beliebigen Stelle innerhalb des Labyrinths.

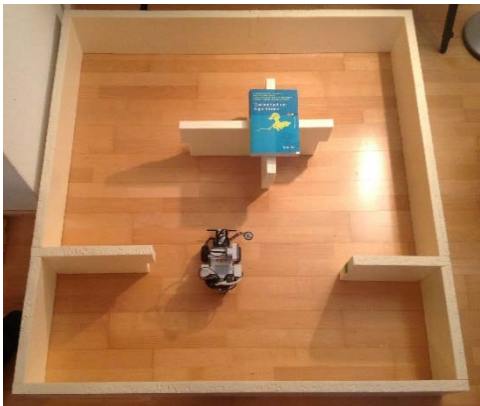
DAS LABYRINTH BAUEN



Für den Bau eines Labyrinths können Hartschaumplatten verwendet werden. Diese bekommt man für ca. CHF 10.- in der Abmessung 120 x 60 cm im Baumarkt. Ohne grossen Kraftaufwand können die Platten relativ einfach und zügig mit einem Teppichmesser, wie in *Abbildung 5 - Schneiden der Platten* ersichtlich ist, zugeschnitten werden. Damit die Teile einfach zusammengesteckt werden können, habe ich jeweils entsprechende Aussparungen von 3 x 10cm ausgeschnitten.



ABBILDUNG 5 - SCHNEIDEN DER PLATTEN

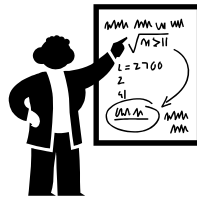


So ist speditiv ein einfaches Labyrinth, wie beispielsweise in *Abbildung 6 - Labyrinth* gezeigt wird, fertiggestellt.

ABBILDUNG 6 - LABYRINTH

BAU DES ROBOTERS

Für den Bau des Roboters genügt das LEGO MINDSTORMS Education NXT Base Set.



Als fahrtüchtigen Roboter beginnt man mit den Schülern am besten mit dem Grundmodell wie in Abbildung 7 - Grundmodell gezeigt. Dieses kann zu zweit oder zu dritt im Team Schritt für Schritt gemäss der beigelegten Bauanleitung zusammengebaut werden. Die Anleitung kann übrigens auch als PDF-Datei auf der beigelegten CD oder im Internet⁶ heruntergeladen werden.

ABBILDUNG 7 - GRUNDMODELL

Das Grundmodell dient auch dem Kennenlernen der Sensoren und Aktoren (Motoren) des Roboters. Je nach Erfahrung mit Lego Mindstorms muss die Lehrperson die verschiedenen Sensoren und deren Funktionsweise erklären.

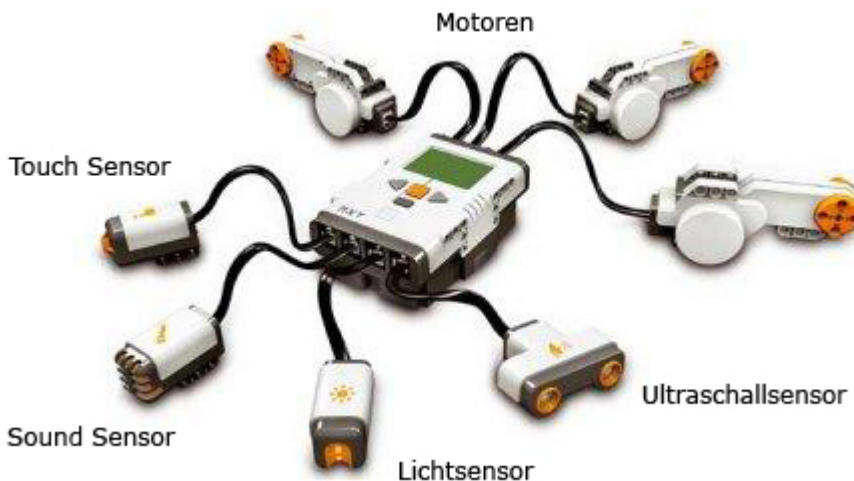
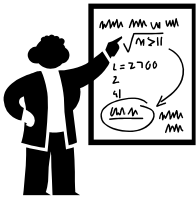


ABBILDUNG 8 - AKTOREN UND SENSOREN

Für den Bau des Roboters genügen 2 Motoren und 3 Sensoren (Licht-, Touch- und Ultraschallsensor).

⁶ <http://www.legorobotik.ch/index.php> (12.01.2013)



Im Team sollen die Schülerinnen und Schüler erst später gemeinsam besprechen, in welcher Form das Grundmodell abgeändert und weiter optimiert werden kann. Dabei kann die Lehrperson punktuell beraten, soll jedoch die Kreativität der Lernenden nicht einengen oder beeinflussen.

Damit die Form des Roboters seine Funktion unterstützt, habe ich, wie in Abbildung 9 - Roboter ersichtlich, gegenüber dem Grundmodell einige Optimierungen angebracht.

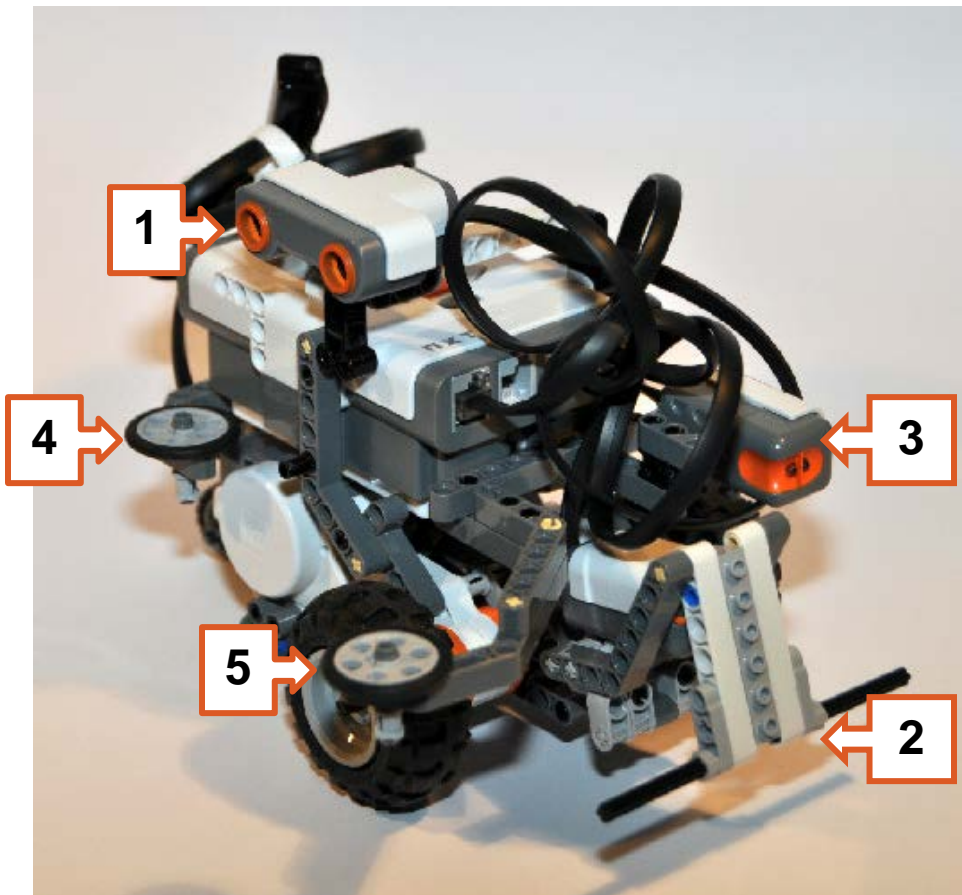


ABBILDUNG 9 - ROBOTER

1. Den Ultraschallsensor habe ich seitlich etwa 13cm ab Boden und 90 Grad versetzt zur Fahrtrichtung angebracht. Die Seitenwände des Labyrinths sind 20cm hoch.
2. An der Front befindet sich hinter einem Druckmechanismus (nach Bauanleitung) der Berührungssensor. Die zwei schwarzen Antennen dienen dem Roboter zur Ausrichtung an der flachen Wand.
3. Der Lichtsensor ist an einer u-förmigen Tragekonstruktion angebracht und zeigt nach vorn.
4. Das hintere kleine Rad verhindert beim Drehen ein unbeabsichtigtes Hängenbleiben an der übrigen Konstruktion.
5. Der Distanzhalter zur Wand ist ein überaus wichtiges Bauteil und verhindert, dass sich der Roboter bei seiner Fahrt dem Hindernis entlang zu nah zur Wand bewegen kann. Dies würde sonst das Wegdrehen von der Wand verunmöglichen.

PROGRAMMIERUNG DES ALGORITHMUS

Erster Teil

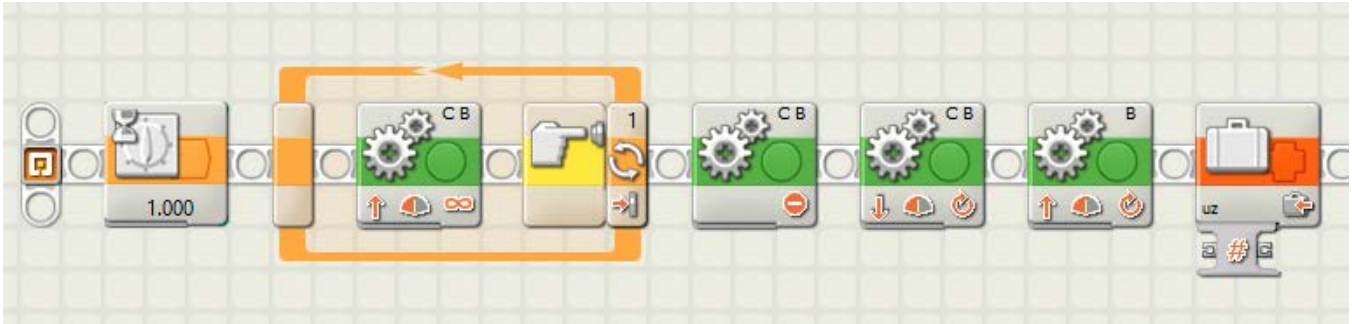


ABBILDUNG 10 - LEGOPROGRAMM TEIL 1

Erklärung

Der Roboter lässt nach 1 Sekunde Wartezeit beide Motoren mit gleicher Leistung vorwärts laufen, bis er durch ein Hindernis vom Drucksensor gestoppt wird. Danach fährt er ein Stück zurück und dreht mit seinem linken Motor eine Umdrehung vorwärts (90 Grad-Drehung nach rechts). Danach wird die Variable uz (Umdrehungszähler) auf den Wert 1 initialisiert.

Zweiter Teil

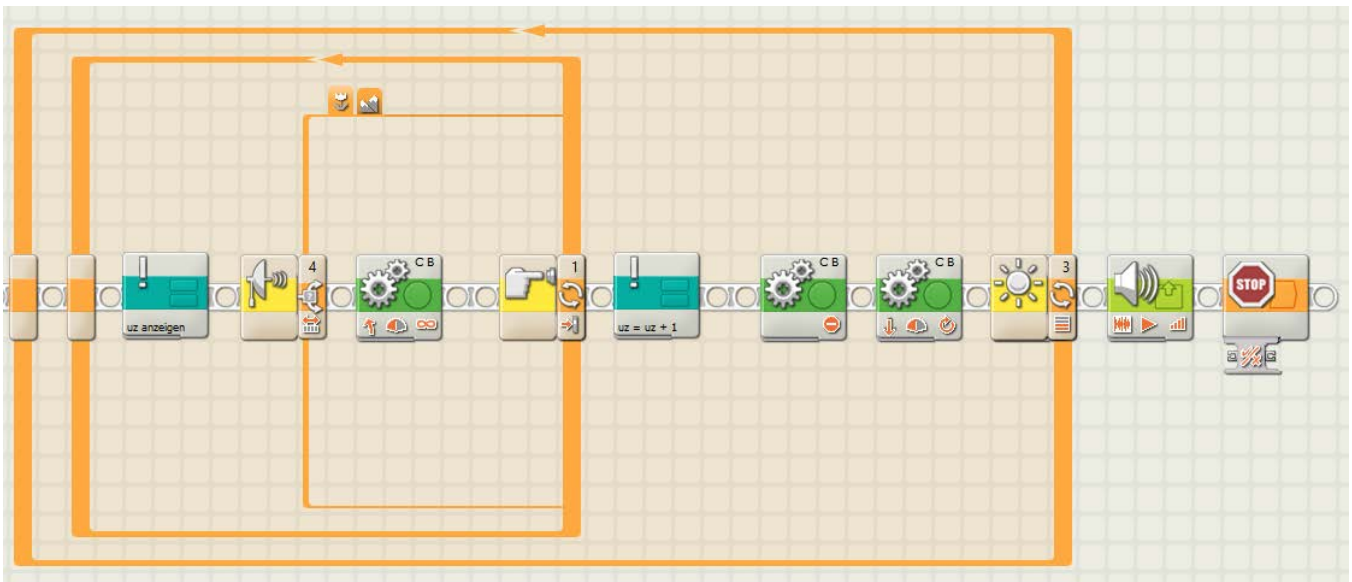
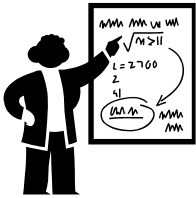


ABBILDUNG 11 - LEGOPROGRAMM TEIL 2

Wie in der Simulation mit Scratch beginnt nach dem ersten Teil eine Schleife, in der der innere Programmteil solange wiederholt wird, bis der Helligkeitssensor erkennt, dass der Roboter aus dem Labyrinth gefunden hat. Erst dann folgt der Klangbaustein, der als letzte Anweisung vor dem eigentlichen Programmende in Form einer Sprachausgabe auf die erfolgreiche Lösung der Aufgabe hinweist. Die Abbildung 11 - Legoprogramm Teil 2 zeigt die Programmstruktur mit der geschlossenen Ansicht der Schleifen. Dadurch sind einige Programmteile der Schleifen verborgen.

Innerer Teil – Folge dem Hindernis

Damit der Roboter einem Hindernis folgen kann, benötigt er den Berührungssensor an der Front und den Ultraschallsensor an der Seite. Diese Aufgabe ist nicht ganz trivial, da es mit den Legomotoren schwierig ist geradeaus zu fahren. Nie sind beide Motoren genau gleich stark und kalibriert. So muss dazu eine eigene Strategie entwickelt werden, um diese Knacknuss zu lösen.



Für diesen Punkt muss genügend Zeit einberechnet werden! Es ist zu beachten, dass mehrere Ultraschallsensoren im Raum zu Fehlmessungen führen können.

Folge der Wand

Damit der Ultraschallsensor die Distanz zur Wand zuverlässig messen kann, darf der Sensor nicht zu hoch montiert werden. Auch der Anstellwinkel zur Wand muss entsprechend in die eingegebenen Grenzwerte miteinberechnet werden. Bei meinen Tests hat sich gezeigt, dass neben dem sehr nützlichen Distanzrad, das den Roboter sicher 5cm von der Wand fernhält, folgende Einstellungen zu einer erfolgreichen Wandverfolgung geführt haben:

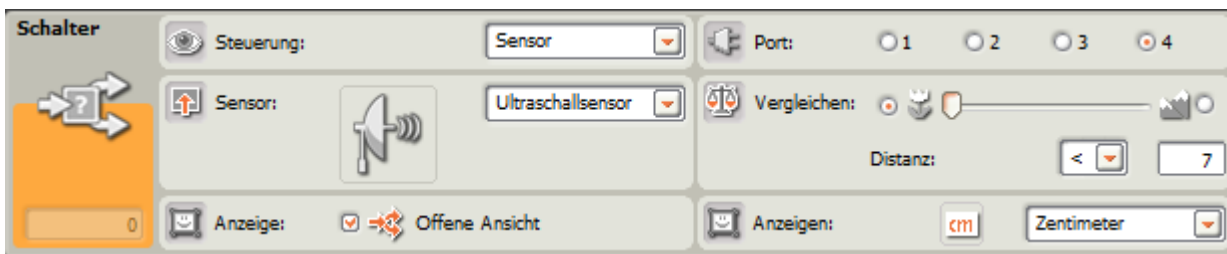


ABBILDUNG 12 - SCHALTEREINSTELLUNG

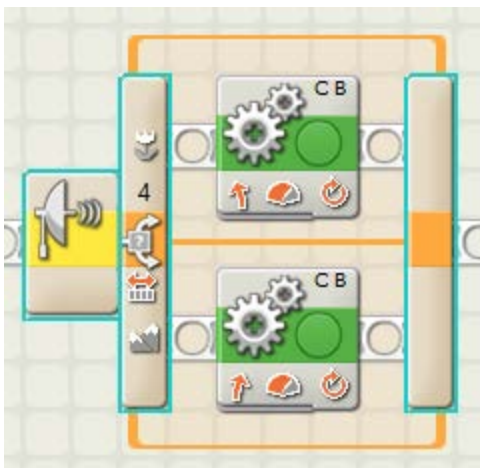


ABBILDUNG 13 - EINER WAND FOLGEN

Falls der Sensor einen Wert grösser als 7cm misst, ist der Roboter zu weit von der Wand entfernt und steuert ganz leicht gegen die Wand zu. Im anderen Fall wird er spätestens bei 5cm vom Distanzrad aufgehalten und der Roboter steuert ganz leicht von der Wand weg (Abbildung 13 - Einer Wand folgen). Das Distanzrad hat zudem den Zweck zu verhindern, dass der Roboter in eine Ecke fährt und daraus nicht mehr rauskommt. Dadurch ist sichergestellt, dass er die folgende 90 Grad Drehung ausführen kann, ohne mit seinem eigenen Heckteil anzuhängen.

Erkenne die Ecke oder eine Lücke und entscheide dem Hindernis zu folgen

Damit eine Ecke oder Lücke in der Wand erkannt wird, muss eine weitere Überprüfung der gemessenen Distanz erfolgen.

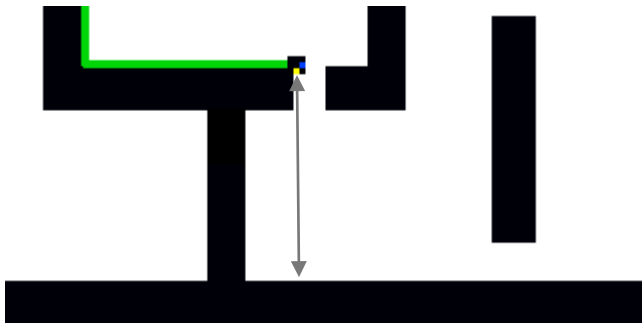


ABBILDUNG 14 - ECKE ODER LÜCKE ERKENNEN

Der Ultraschallsensor misst sprunghaft einen hohen Wert, sobald der Roboter einer Wand entlang fährt, die plötzlich endet (Situation in Abbildung 14 - Ecke oder Lücke erkennen). Sobald der gemessene Wert 7cm übersteigt und zugleich grösser als beispielsweise 20cm ist handelt es sich um eine Ecke oder Lücke in der Wand. Sofern der Drehungszähler nicht auf 0 steht, muss der Roboter dem Hindernis folgen. Dies kann mit zwei ineinander geschachtelten Schleifen mit je zwei Schaltern gelöst werden (Abbildung 15 - Geschachtelte Schleifen). Die äussere Schleife hier in der offenen Ansicht, entspricht der Abbildung 13 - Einer Wand folgen auf der vorhergehenden Seite.

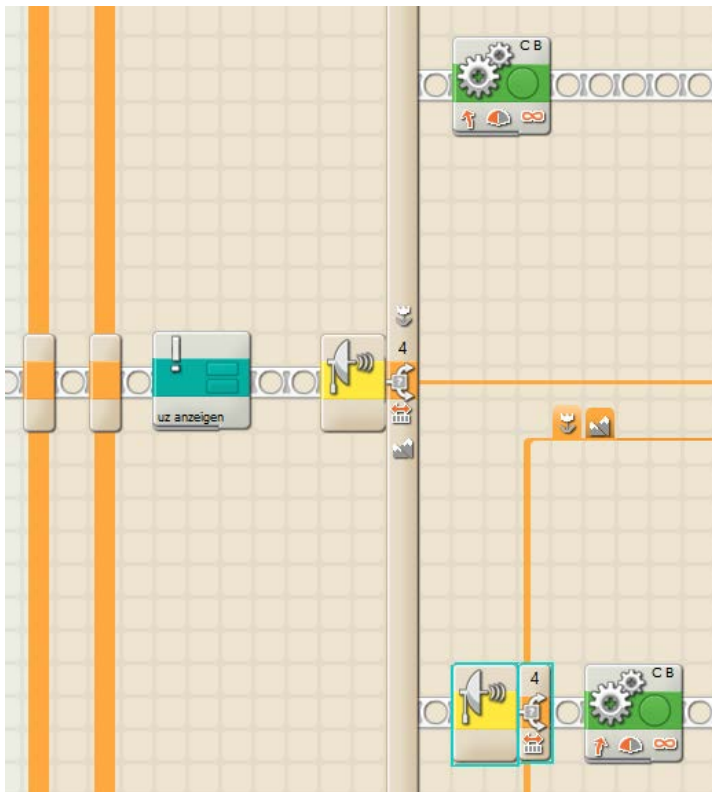


ABBILDUNG 15 - GESCHACHTELTE SCHLEIFEN

Die innere Schleife hingegen ist etwas komplexer aufgebaut (Abbildung 16 - Innere Schleife mit Logikschalter). Im Fall, dass der Ultraschallsensor einen höheren Wert als 20cm misst, wird zusätzlich mit einem Logikschalter überprüft, ob die Variable *uz* (Umdrehungszähler) grösser als 0 ist. Erst dadurch kann der Roboter den Pledge-Algorithmus korrekt umsetzen und sich bei Bedarf von der Wand lösen.

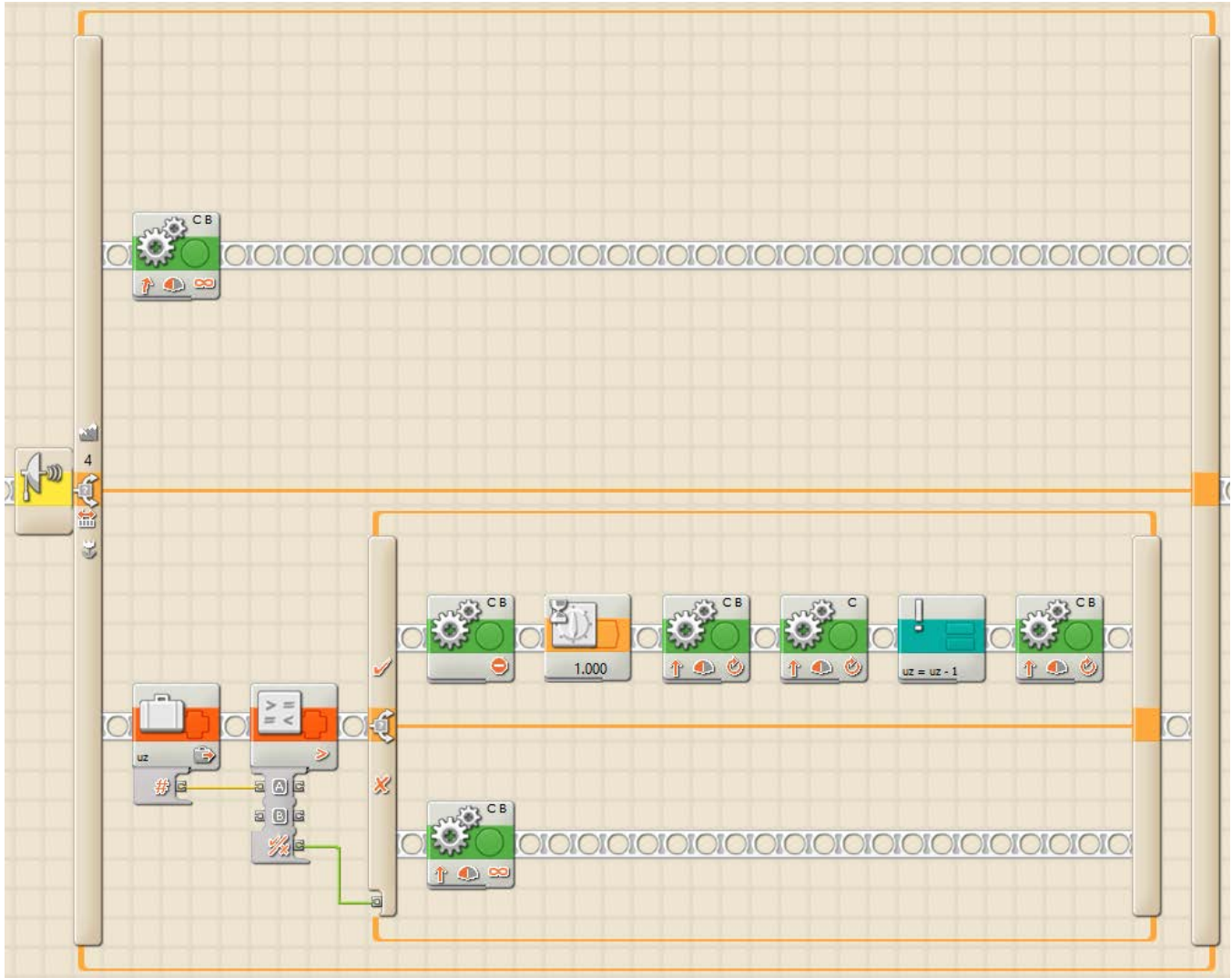


ABBILDUNG 16 - INNERE SCHLEIFE MIT LOGIKSCHALTER

Soweit ist unser Algorithmus fertig umgesetzt und der Roboter kann losfahren. Das entsprechende Video ist auf Youtube unter folgendem Link einsehbar:

<http://www.youtube.com/playlist?list=PLxpHOspRkl6yqH2Fjn89msmdlzHZbGXwF> (12.01.2013)

LITERATURVERZEICHNIS

Berthold Vöcking, H. A. (2008). *Taschenbuch der Algorithmen*. Leipzig: Springer.

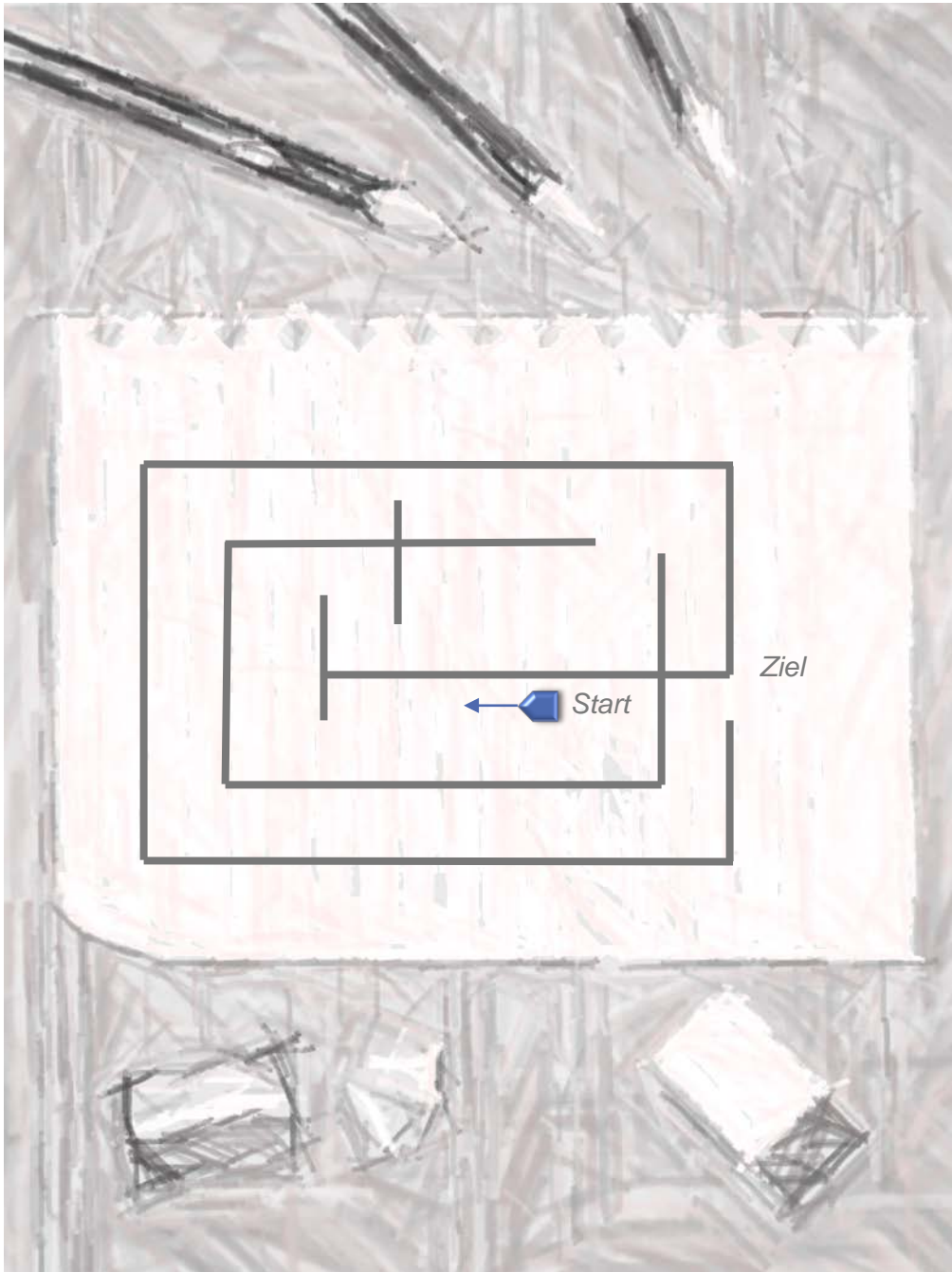
Dr. Hanjo Täubing, J. W. (4. Mai 2012). Grundlagen: Algorithmen und Datenstrukturen. Technische Universität München, Bayern, Deutschland.

Karsten Berns, D. S. (2010). *Programmierung mit LEGO MINDSTORMS NXT*. Stuttgart: Springer.

ANHANG

Arbeitsblatt Labyrinth 1

Wie findest Du im Dunkeln aus diesem Labyrinth?

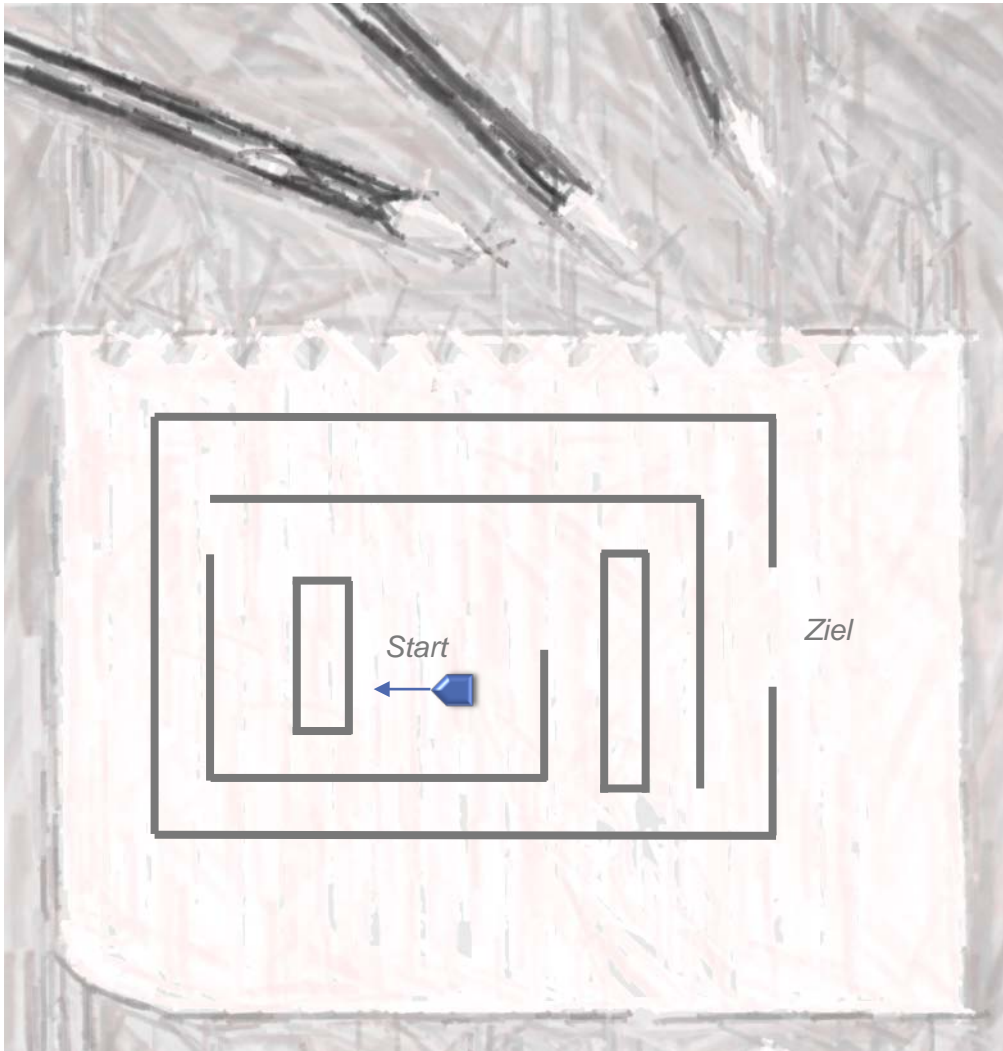


Beschreibe eine mögliche Lösungsstrategie in Form einer klaren Handlungsanweisung:

Arbeitsblatt Labyrinth 2

Eine mögliche Strategie für Labyrinth 1 lautet: „Gehe geradeaus bis du zu einer Wand kommst. Danach drehst du dich nach rechts und folgst der Wand solange, bis du den Ausgang findest.“

Teste diese Strategie in diesem Labyrinth.



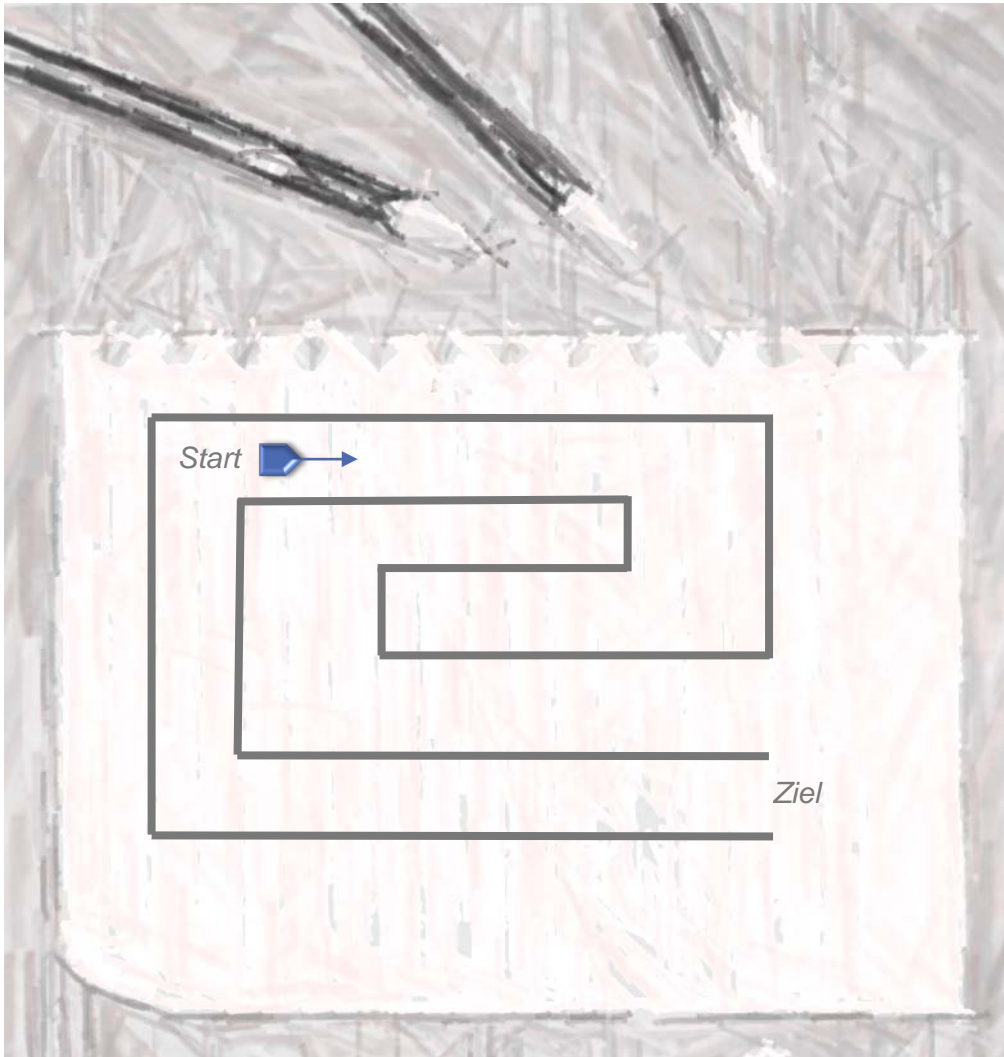
Weshalb funktioniert sie nicht?

Was musst du ändern, damit dein Roboter in diesem Labyrinth den Ausgang findet?

Arbeitsblatt Labyrinth 3

Eine mögliche Strategie für Labyrinth 2 lautet: „Gehe geradeaus, bis du zu einer Wand kommst. Danach drehst du dich nach rechts und folgst der Wand solange, bis du wieder in die ursprüngliche Richtung gehst. Danach gehst du so lange gerade aus, bis du auf das nächste Hindernis stößt.“

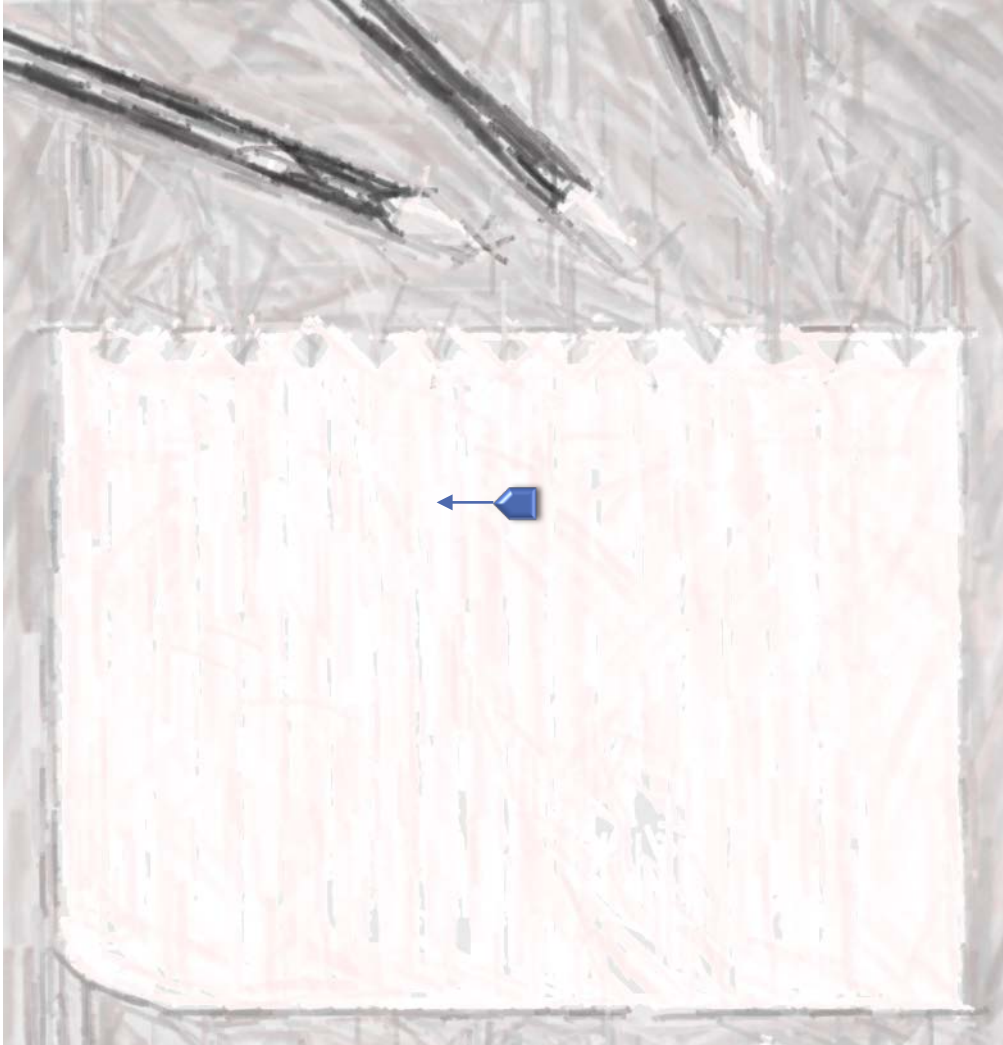
Probiere diese Strategie in diesem Labyrinth.



Weshalb funktioniert sie nicht und was musst du ändern, damit du in diesem Labyrinth den Ausgang findest?

Arbeitsblatt Partnerarbeit Labyrinth 4

Findest du einen Algorithmus, der in allen möglichen Fällen den Ausgang findet? Zeichne ein kompliziertes Labyrinth und teste damit den Algorithmus deines Mitschülers. Dabei soll dein Partner die Rolle des Computerprogramms, und du die des Roboters einnehmen. Wichtig ist dabei, dass das Computerprogramm die Welt des Roboters nicht sieht.



Dein Algorithmus:
